A survey of the cell-growth problem and some its variations [1]

Elena V. Konstantinova

Sobolev Institute of Mathematics, Russian Academy of Sciences,
Novosibirsk 630090, Russia
e_konsta@math.nsc.ru
and
Combinatorial and Computational Mathematics Center,
Pohang University of Science and Technology,
Pohang 790–784, The Republic of Korea
e_konsta@com2mac.postech.ac.kr

Abstract

A very brief survey of the main results concerning the cell–growth problem and its variations is given. The name stems from an analogy with an animal which, starting from a single cell of some specified basic polygonal shape, grows step by step in the plane by adding at each step a cell of the same shape to its periphery. The fundamental combinatorial problem concerning these animals is "How many animals with n cells are there?" This problem was included in the list of unsolved problems in the enumeration of graphs by Frank Harary in 1960. Despite serious efforts over the last 40 years, this problem is completely open. However, a few asymptotic results are known. For example, let $p(n)$ denote the number of polyominoes (square animals) having n cells. It was proved that $(p(n))^{1/n}$ tends to a limit Θ, which satisfies the following inequality: $3.87 < \Theta < 4.65$. The situation could hardly be worse, since the first digit of Θ is not even known...

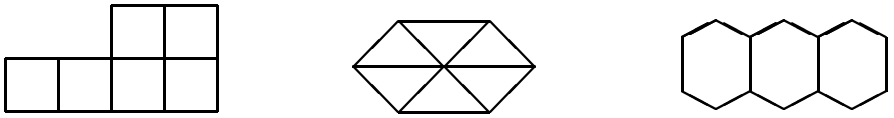
The difficulty of the classical cell–growth problem has led to the study of various restricted classes of polyominoes. Some variations of this problem are considered. Unsolved problems are stated. Chemical applications of this problem are mentioned too.

1. Classical cell–growth problem

Combinatorial problem known as *cell-growth problem* is stated as follows [1–7]. The name stems from an analogy with an *animal* which, starting from a single *cell* of some specified basic polygonal shape, grows step by step in the plane by adding at each step a cell of the same shape to its periphery. Thus if the basic shape is a square, the animals are the *polyominoes* (Fig.1a). If the basic shape is an equilateral triangle or a regular hexagon, we obtain triangular and hexagonal animals looking like those in Fig.1b and Fig.1c. Animals are defined as *simply–connected* ones if they have no *holes* and as *multiply–connected* ones otherwise. All animals presented in Fig.1 are simply–connected ones. The smallest multiply–connected polyomino is shown in Fig.2.

The fundamental combinatorial problem concerning these animals is *"How many animals with n cells are there?"* This problem was included in the list of unsolved problems in the enumeration of graphs by Harary in 1960 [8]. Polyominoes have the most long history, going to the start of the 20th century, but

[1]Supported by Com^{2}MaC-KOSEF, The Republic of Korea.



a *b* *c*

Figure 1. Simply-connected square (a) , triangular (b) and hexagonal (c) animals

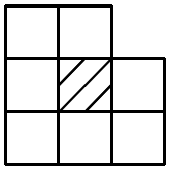
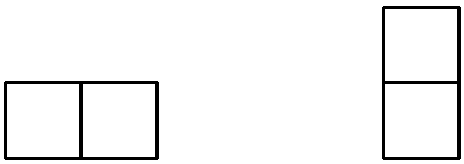


Figure 2. The smallest multiply-connected polyomino

they were popularized in the present era by Golomb [9–11] and by Gardner [12, 13] in his *Scientific American* columns "Mathematical Games". Another notable book on the subject is written by Martin [14]. There are a great many articles and problems concerning polyominoes to be found in the magazine *Recreational Mathematics* [15–20].

The answer on the main question "how many animals are there?" depends on how we distinguish animals. There are some distinguishing rules commonly used, and for each set there is a name for the animals.

Free animals are considered distinct if they have different shapes. Their orientation and location in the plane is no importance. For example, the two animals:



are the same free square animal since they differ only in orientation. We use *free(n)* to denote the number of free animals with n cells.

Fixed animals are considered distinct if they have different shapes or orientations. Thus two animals above are different fixed animals. We use *fixed(n)* to denote the number of fixed animals with n cells.

Originally the cell–growth problem was considered for the polyominoes. The most general discussion of polyominoes was done by Golomb [10], however the number of polyominoes was only briefly discussed. In 1962 Read [21] derived several theoretical results about the number of polyominoes. He presented a method for deriving generating functions to calculate the number of simply–connected and multiply–connected polyominoes, but these become intractable very quickly. He calculated *free(n)* only for n up to 10 and his value for $n = 10$ was incorrect.

Klarner [22, 23] found bounds for *free(n)* and *fixed(n)* polyominoes. The values seem to be growing exponentially, and indeed they have exponential bonds. It is easy to see that for each n,

$$\frac{fixed(n)}{8} \leq free(n) \leq fixed(n)$$

Eden [24] seems to have been the first person to give upper and lower bounds for *fixed(n)*. His bounds are

$$(3.14)^n < fixed(n) < 4^n,$$

for sufficiently large n. The proof of his upper found was questionable. Later these bounds were improved by Klarner and Rivest [25]. Using automata theory and building on earlier works of Eden, Klarner and Read they have shown

Theorem 1 [25]

$$\lim_{n\to\infty} (fixed(n))^{\frac{1}{n}} = \Theta \quad exists, \quad and \quad 3.87 < \Theta < 4.65. \tag{1}$$

Considerable effort has been expended to find a formula for the number of fixed polyominoes, with no success. Lunnon [26] has made the most successful previous enumeration. He computed the numbers of free, fixed and symmetric polyominoes up to 18 cells. Later Lunnon [27] computed the numbers of free and fixed triangular and hexagonal animals up to $n = 16$ and $n = 12$ respectively. The results are given in Table 1 and Table 2.

Table 1. The numbers of fixed and free triangular animals [27]

n	*fixed(n)*	*free(n)*
1	2	1
2	3	1
3	6	1
4	14	3
5	36	4
6	94	12
7	250	24
8	675	66
9	1838	160
10	5053	448
11	14.016	1186
12	39.169	3334
13	110.194	9235
14	311.751	26.166
15	886.160	73.983
16	2.529.260	211.297

Table 2. The numbers of fixed and free hexagonal animals [27]

n	*fixed(n)*	*free(n)*
1	1	1
2	3	1
3	11	3
4	44	7
5	186	22
6	814	82
7	3652	333
8	16.689	1448
9	77.359	6572
10	362.671	30.490
11	1.716.033	143.552
12	8.182.213	683.101

Table 3. The numbers of fixed and free polyominoes [28]

n	*fixed(n)*	*free(n)*
1	1	1
2	2	1
3	6	2
4	19	5
5	63	12
6	216	35
7	760	108
8	2725	369
9	9910	1285
10	36.446	4655
11	135.268	17.073
12	505.861	63.600
13	1.903.890	238.591
14	7.204.874	901.971
15	27.394.666	3.426.576
16	104.592.937	13.079.255
17	400.795.844	50.107.909
18	1.540.820.542	192.622.052
19	5.940.738.676	742.624.232
20	22.964.779.660	2.870.671.950
21	88.983.512.783	11.123.060.678
22	345.532.572.678	43.191.857.688
23	1.344.372.335.524	168.047.007.728
24	5.239.988.770.268	654.999.700.403

Redelmeier [28] enumerated all free and fixed polyominoes up to 24 cells. His algorithm, which produced the entries in Table 3 (and took over ten months of computer time to run), generates the fixed polyominoes one by one and counts them. The running time is (necessarily) exponential. At present, the computation of fixed(n) for $n > 30$ seems intractable.

Klarner [29] presented some unsolved problems arising in the cell–growth problem for polyominoes.

Problem 1. *Can the number of fixed animals with n cells be computed by a polynomial–time algorithm?*

A related problem concerns the constant Θ defined above.

Problem 2. *Is there a polynomial algorithm to find, for each n, an approximation* Θ_n *of* Θ *satisfying*

$$10^{-n} < |\Theta_n - \Theta| < 10^{-n+1}?$$

The lower–bound method of Klarner and Satterfield [30] gives an algorithm for approximating Θ from below that has exponential complexity; no such method is known for approximating Θ from above.

Problem 3. *Define some decreasing sequence* $\beta = (\beta_1, \beta_2, ...)$ *that tends to* Θ*, and give an algorithm to compute* β_n *for every n.*

It is known that $(fixed(n)^{1/n}) \leq \Theta$ for all n, and it seems that the ratios $\tau(n) = fixed(n+1)/fixed(n)$ increase for all n. If the latter is true, $\tau(n)$ would approach Θ from below. This gives two more unsolved problems:

Problem 4. *Show that* $(fixed(n))^{1/n} \leq (fixed(n+1))^{1/(n+1)}$ *for all* n

Problem 5. *Show that* $\tau(n) \leq \tau(n+1)$ *for all* n

Problem 6. *Is the generating function* $T(z) = \sum_{n=1}^{\infty} fixed(n) z^n$ *rational function?*
Is $T(z)$ *even algebraic?*

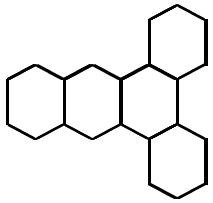
One can consider all of these problems for triangular and hexagonal animals.

So we gave the answer for the following question *"How many free and fixed animals with n cells are there?"*

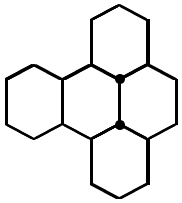
Actually one can say about another distinguishing rule among animals. We can ask *"How many simply–connected and multiply–connected animals with n cells are there?"*

Read [21] calculated the numbers of simply–connected and multiply–connected square animals up to $n = 10$. Later Trinajstić, etc., [31, 32] computed the numbers of simply–connected animals up to 10 cells and the numbers of multiply–connected animals with the only hole up to 10 cells.

The hexagonal animals which are also sometimes called *polyhexes* correspond to the structural formulas of planar polycyclic aromatic hydrocarbons [33–35]. That is the reason why polyhexes have found a big interest among chemists [36–51]. Moreover one more distinguishing rule among polyhexes was considered. For simply–connected hexagonal animals it was done the answer on the following question *"How many animals with n cells and i internal vertices are there?"* This classification is important for



cata-condensed hydrocarbon
a



peri-condensed hydrocarbon
b

Figure 3. Hexagonal animals depicting dibenzo[a,c]anthracene (a) and benzo[e]pyrene (b)

chemists because the hexagonal simply–connected animals without internal vertices correspond to the cata–condensed benzenoid hydrocarbons and the hexagonal simply–connected animals with internal vertices correspond to the peri–condensed benzenoid hydrocarbons (see Fig.3). The obtained results are given in Table 4.

The same classification was used for square and triangular animals by Konstantinova [52, 53]. The numbers of simply–connected square and triangular animals without and with internal vertices up to 11 and 13 cells correspondingly are given in Table 5 and Table 6.

Table 4. The numbers of simply–connected hexagonal animals with i internal vertices [31]

$n\backslash i$	0	1	2	3	4	5	6	7	8	9	10	*total*
1	1											1
2	1											1
3	2	1										3
4	5	1	1									7
5	12	6	3	1								22
6	36	24	14	4	3							81
7	118	106	68	25	10	3	1					331
8	411	453	329	144	67	21	9	1				1435
9	1489	1966	1601	825	396	154	55	15	4			6505
10	5572	8395	7652	4518	2340	1018	416	123	42	9	1	30086

Table 5. The numbers of simply–connected square animals with i internal vertices [52]

$n\backslash i$	0	1	2	3	4	5	*total*
1	1						1
2	1						1
3	2						2
4	4	1					5
5	11	1					12
6	27	7	1				35
7	82	21	4				107
8	250	90	21	2			363
9	815	334	89	9	1		1248
10	2685	1311	391	67	6		4460
11	9072	4978	1674	324	45	1	16094

Table 6. The numbers of simply-connected triangular animals with i internal vertices [53]

$n\backslash i$	0	1	2	3	*total*
1	1				1
2	1				1
3	1				1
4	3				3
5	4				4
6	11	1			12
7	23	1			24
8	62	4			66
9	148	11			159
10	405	38	1		444
11	1041	118	2		1161
12	2825	386	15		3226
13	7541	1189	54	1	8785

She also presented the numbers of multiply–connected square [52], triangular [53] and hexagonal [54] animals with respect to the type of holes.

All multiply–connected square animals with the fixed internal boundaries presented in Fig.4 were generated and enumerated. The obtained data for $n = 9, n = 10, n = 11$ are given in Table 7, Table 8 and Table 9 correspondingly. In these tables t is the type of an internal boundary (see Fig.4) and i is the number of internal vertices. The total numbers of multiply–connected square animals with $7 \leq n \leq 11$ are given in Table 10. The total numbers of simply– and multiply–connected square animals for $n \leq 11$ are given in Table 11. These data correspond to the numbers of free square animals [26, 28].

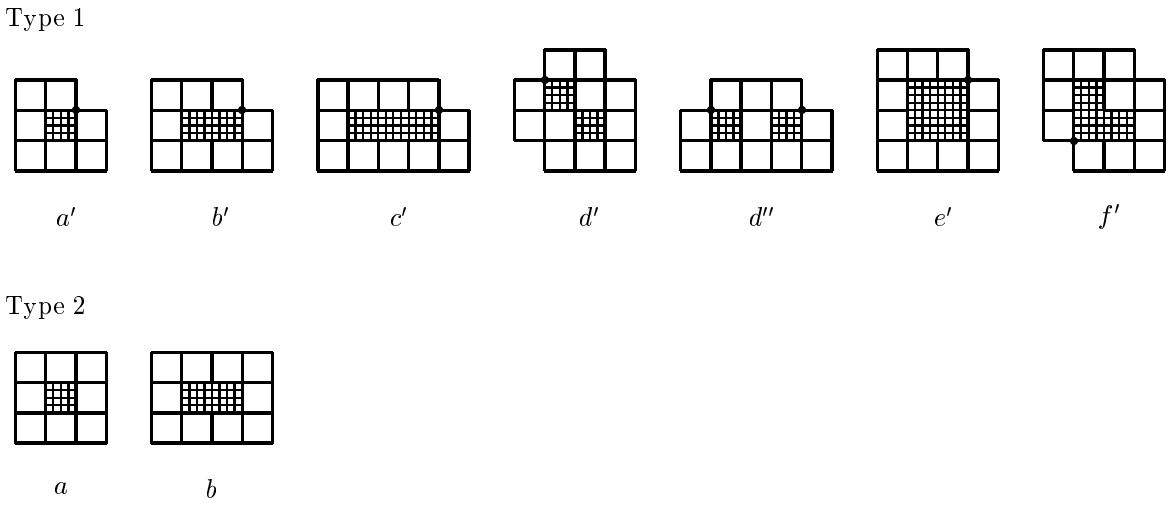


Figure 4. The different types of internal boundaries for multiply–connected square animals with up to 11 cells

Table 7. The numbers of multiply–connected square animals with $n = 9$

$i\backslash t$	a′	a	b′	*total*
0	31	2	1	34
1	3			3
total	34	2	1	37

Table 8. The numbers of multiply–connected square animals with $n = 10$

$i\backslash t$	a′	a	b′	b	*total*
0	132	14	12	1	159
1	34	1			35
2	1				1
total	167	15	12	1	195

Table 9. The numbers of multiply–connected square animals with $n = 11$

$i\backslash t$	a'	a	b'	b	c'	d'	d''	e'	f'	*total*
0	575	52	78	4	1	2	2	1	3	718
1	213	13	8							234
2	26	1								27
total	814	66	86	4	1	2	2	1	3	979

Table 10. The total numbers M of multiply–connected square animals

n	7	8	9	10	11
M	1	6	37	195	979

Table 11. The total numbers of simply– and multiply–connected square animals

n	1	2	3	4	5	6	7	8	9	10	11
data	1	1	2	5	12	35	108	369	1285	4655	17073

Moreover the diagrams of all multiply–connected square animals up to 11 cells are given in [52] and diagrams of all multiply–connected triangular animals up to 13 cells are given in [53].

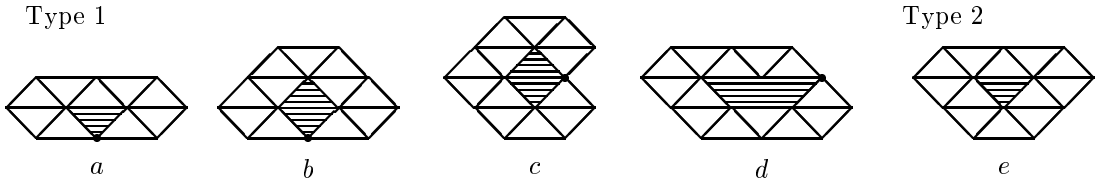


Figure 5. The different types of internal boundaries for multiply–connected triangular animals with up to 13 cells

All multiply–connected triangular animals with the fixed internal boundaries presented in Fig.5 were generated and enumerated. The obtained data are given in Table 12. In this table t is the type of an internal boundary, i is the number of internal vertices and n is the number of cells. The total numbers of simply– and multiply–connected triangular animals for $n \leq 13$ are given in Table 13. These data correspond to the numbers of free triangular animals [27].

Table 12. The number of multiply–connected triangular animals

i	0					1		
$n\backslash t$	a	b	c	d	e	a	total	
9	1						1	
10	4						4	
11	24	1					25	
12	100	5	1		1	1	108	
13	405	29	5	1	2	8	450	

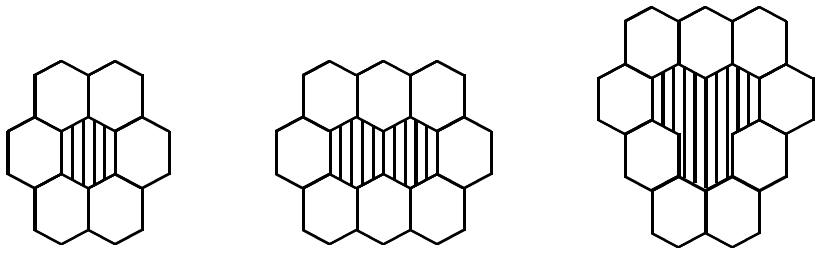
Table 13. The total number of simply– and multiply–connected triangular animals

n	1	2	3	4	5	6	7	8	9	10	11	12	13
data	1	1	1	3	4	12	24	66	160	448	1186	3334	9235

All multiply–connected hexagonal animals with the fixed internal boundaries presented in Fig.6 were generated and enumerated. The obtained data are given in Table 14. In this table t is the type of an internal boundary, i is the number of internal vertices and n is the number of cells. The total numbers of simply– and multiply–connected hexagonal animals for $n \leq 9$ are given in Table 15. These data correspond to the numbers of free hexagonal animals [27].

Table 14. The number of multiply–connected hexagonal animals

i	0			1		2	3	4	
$n\backslash t$	a	b	c	a	b	a	a	a	total
6	1								1
7	1			1					2
8	5	1		3		4			13
9	17	2	1	17	2	17	10	1	67



a *b* *c*

Figure 6. The different types of internal boundaries for multiply–connected hexagonal animals with up to 9 cells

Table 15. The total number of simply– and multiply–connected hexagonal animals

n	1	2	3	4	5	6	7	8	9
data	1	1	3	7	22	82	333	1448	6572

Sometimes more careful classification is used for simply–connected hexagonal animals [55]. *Unbranched tree–like polyhexes* have only two terminal cells (see Fig.1c). *Branched tree-like polyhexes* have more than two terminal cells (see Fig.3a). Unbranched tree–like polyhexes are the graph representations of unbranched cata–condensed benzenoid molecules, including helicenic species (non-embedded to the plane) and play a distinguished role in the theoretical chemistry of benzenoid hydrocarbons [35].

The unbranched tree–like polyhexes U_n were counted by Balaban and Harary [36]:

$$U_n = \begin{cases} \frac{1}{4}\,(3^{(n-2)/2}+1)^2, & \text{if } n \text{ is an even} \\ \frac{1}{4}\,(3^{n-2}+3^{(n-1)/2}+3^{(n-3)/2})^2, & \text{if } n \text{ is an odd} \end{cases} \tag{2}$$

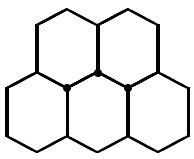
Later Dobrynin [56] computed and generated all these polyhexes up to 16 cells. Some variations of this problem were considered by Cyvin, etc., [57–60] for unbranched tree–like systems of congruent polygons.

Cyvin [60] formulated the following problem in mathematical chemistry. Let P is the polyhex, i is the number of internal vertices in P and n is the number of hexagons in P. Then a very useful relation for polyhexes holds:

$$i \le 2n - \lceil (12n-3)^{1/2} \rceil, \tag{3}$$

where $\lceil x \rceil$ is the smallest integer not smaller than x. The upper bound is realized in extremal animals [61] .

For example, for $n = 5$ extremal polyhex P looks like this one:

P 

Using the above formula we exactly have the following upper bound

$$i \leq 2 \cdot 5 - \lceil (12 \cdot 5 - 3)^{1/2} \rceil = 10 - \lceil \sqrt{57} \rceil = 10 - 7 = 3, \tag{4}$$

which is realized in polyhex P.

Let define the mono–q–polyhex as the planar graph embedded to the mono–q–hexagonal lattice which is similar to the hexagonal lattice; it consists of exactly one q–gon and otherwise hexagons.

Many hydrocarbons correspond to the mono–q–polyhex graphs, e.g., the (q)circulenes. (5)circulene, (6)circulene, (7)circulene have been synthesized and a synthesis of (8)circulene has been attempted.

Let h be the number of hexagons outside the unique q–gon. Then the following conjecture is proposed for mono–q–polyhexes:

Problem 7 [60]. *Show that*

$$i \leq 2h - \lceil (1/2)(8qh + q^2)^{1/2} - (q/2) \rceil$$

The upper bound is supposed to be realized in the appropriate extremal systems.

One more problem immediately arise here.

Problem 8. *To enumerate all mono–q–polyhex with i internal vertices and h hexagons*

So the main results concerning the cell–growth problem and some unsolved problems arising there are considered above. Actually there is a lot of variations for this combinatorial problem. We will consider some of them.

2. Cell–growth problem for non–embedding animals

The classical cell–growth problem was formulated for the animals embedded to the plane. The animals non–embedded to the plane were investigated in the several papers [3, 22, 23, 36, 41, 56, 57, 62, 63].

As was mentioned above unbranched tree–like polyhexes embedded and non–embedded to the plane were considered by Balaban, Harary and Read [3, 36]. Moreover they have obtained the formula (2) for the number of unbranched tree–like polyhexes embedded and non–embedded to the plane. Actually in [3] it was shown how, by making a fairly drastic change in the definition of hexagonal animals, it is possible to arrive at a combinatorial problem for which an explicit solution exists.

In [62] a variation of the cell–growth problem for so–called $n-$*clusters* was considered. Here is a more formal recursive definition. The graph which is a polygon of order n ($n-$gon) is an $n-$*cluster*, and if G is an $n-$cluster of order p then the graph of order $p + (n-2)$ obtained by identifying an edge of a new $n-$gon with an edge of G lying in exactly one $n-$gon is again an $n-$cluster. The example of 6–cluster is given in Figure 7. Thus three–like polyhexes enumerated in [3] form a subset of hexagonal clusters. The generating function for $n-$clusters was obtained and the results for $3 \leq n \leq 6$ are given in Table 16.

It was also mentioned that the enumeration of $n-$clusters can be viewed as the counting of dissections of a polygon. One can draw a cluster so that its perimeter (the set of outer edges) appears as a regular

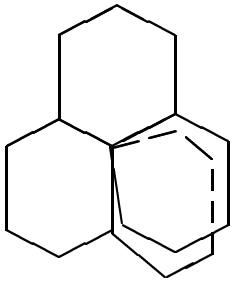


Figure 7. Hexagonal cluster

Table 16. The numbers of $n-$clusters, $3 \leq n \leq 6$, with h cells [62]

h	$n=3$	$n=4$	$n=5$	$n=6$
1	1	1	1	1
2	1	1	1	1
3	1	2	2	3
4	6	5	8	12
5	4	16	33	68
6	12	60	194	483
7	27	261	1196	3946
8	82	1243	8196	34.485
9	228	6257	58.140	315.810
10	733	32.721	427.975	2.984.570
11	2282	175.760	3.223.610	28.907.970
12	7528	963.900	24.780.752	285.601.251
13	24.834	5.374.400	193.610.550	2.868.869.733
14	83.898	30.385.256	1.534.060.440	29.227.904.840
15	285.357	173.837.631	12.302.123.640	301.430.074.416
16	983.244	1.004.867.079	99.699.690.472	3.141.985.563.575
17	3.412.420	5.861.610.475	815.521.503.060	33.059.739.636.198
18	11.944.614	34.469.014.515	6.725.991.120.004	
19	42.080.170	204.161.960.310	55.882.668.179.880	
20	149.197.152	1.217.145.238.485		
21	531.883.768	7.299.007.647.552		
22	1.905.930.975	44.005.602.441.840		
23	6.861.221.666			
24	24.806.004.996			
25	90.036.148.954			
26	327.989.004.892			
27	1.198.854.697.588			
28	4.395.801.203.290			
29	16.165.198.379.984			
30	59.609.171.366.325			

polygon. Then the cluster gives a dissection of the regular polygon into regions, each of which is an $n-$gon. The simplest of these dissection problems is the one for which $n = 3$, and concerns triangulations of the polygon. For some special cases this problem has been solved by Guy [64] and Motzkin [65]. In this connection see also the catalogue of sequences by Sloane [66], which corrects an error in Guy's list,

Table 17. The numbers of non–embedding polyhex NEP up to 10 hexagon [41]

n	1	2	3	4	5	6	7	8	9	10
NEP						1	8	71	542	3857

and also one in Motzkin's. Note also that the problem of counting clusters rooted at an exterior edge is equivalent to that of counting dissections of a fixed polygon, and has been considered in some detail by Motzkin. The case $n = 3$ for a fixed polygon is particularly well–known, having a history that extends all the way back to Euler, and gives rise to the ubiquitous Catalan number.

Some chemical enumerations of non-embedding animals take place too. Trinajstić, etc., [41] enumerated all simply–connected polyhexes non–embedded to the plane up to 10 hexagon. These polyhexes are the graph representations of cata–helicene and peri–helicene benzenoid hydrocarbons [55]. The data are shown in Table 17.

The following unsolved problems concerning non–embedding animals with n cells are here.

Problem 9. *To enumerate all simply–connected and multiply–connected non–embedding animals*

Problem 10. *To enumerate all non–embedding simply–connected animals with i internal vertices*

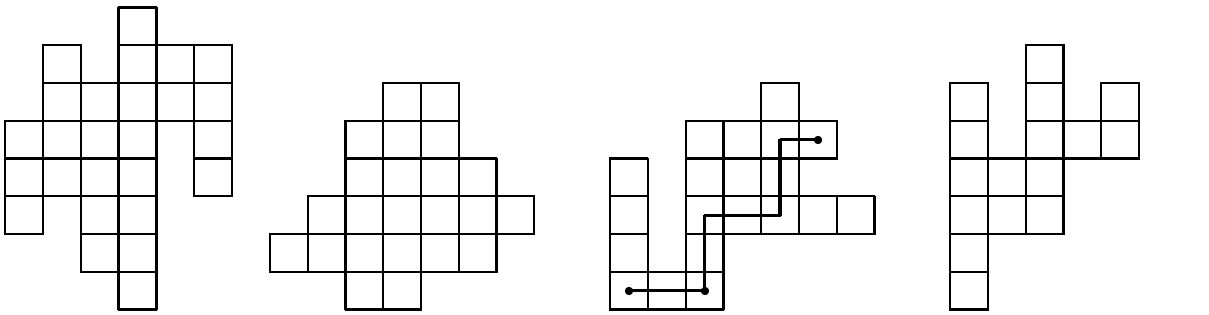
All previous considerations were dealing with the 2-dimensional case. Actually one can consider 3-dimensional case of the cell-growth problem and formulate the cell–growth problem for this case using, for example, cubes instead of squares:

Problem 11. *How many cubical animals are there?*

3. Variation of cell–growth problem: convex polyominoes

The difficulty of the classical cell–growth problem has led to the study of various restricted classes of polyominoes. Most of them can be defined by combining two notions: a geometric notion of *convexity*, and a notion of *directed growth*, which comes from statistical physics. Dhar [67, 68] presented the important example of the correspondence between the enumeration of *directed* polyominoes on a regular lattice in dimension D and the resolution of a gas–model in dimension $D - 1$.

A polyomino is said to be *vertically convex* (or *column–convex*) when its intersection with any vertical line is convex (see Fig.8). We can define similarly a notion of *horizontal* (or *row-*) convexity. A polyomino is *convex* if it is both vertically and horizontally convex. The *area* of a polyomino is the number of cells, and the *perimeter* is the length of the border. A polyomino is said to be *directed* when every it's cell can be reached from a distinguished cell, called a *root*, by a path that is contained in polyomino and has only North and East steps (see Fig.8).



Column–convex polyomino | Convex polyomino | Directed polyomino | Directed column–convex polyomino

Figure 8. Four main subclasses of polyominoes

Combining the two notions described above, one can already define four types of polyominoes, depending on whether they are only column–convex, or also row–convex, directed or not. Namely, here are the four main subclasses of polyominoes: column–convex polyominoes, convex polyominoes, directed and column–convex polyominoes, directed and convex polyominoes.

Usually the enumeration of these objects according to their perimeter and area is considered. Roughly, one can say that two kinds of generating functions occur, depending on the convexity properties of the class of polyominoes that is being enumerated. More precisely:

- the perimeter generating function for any usual convex polyominoes is an algebraic series, whereas the area generating function involves q-series; moreover, taking into account the perimeter (or the width and the height) when one already knows the area generating function is usually a rather easy task;

- the situation is different for families of column–convex polyominoes: the perimeter generating function and the area generating function are both algebraic; but the difficulty consists in taking into account simultaneously the two parameters.

Column–convex polyominoes apparently first appeared in Pólya's diary notes [69] and were independently introduced by Temperley [70]. The area generating function of these polyominoes was found on the spot [69, 70]. Klarner [22] has obtained the area generating function of row–convex polyominoes. He used the following method.

Let a composition of n with k parts is an ordered k-tuple $(a_1, \ldots, a_k)$ of positive integer with $a_1 + \ldots + a_k = n$ and let us assign to each composition a polyomino with n cells and with a horizontal strip of a_i cells in row i. Thus can be done in many ways, and the results are all row–column polyominoes. The examples of 6 row–convex polyominoes with 6 cells corresponding to the composition (3,1,2) of 6 are shown in Figure 9.

Since there are $(m+n-1)$ ways to form polyominoes with $(m+n)$ cells by placing a strip of n cells atop a strip of m cells, it follows that for each composition there are

$$(a_1 + a_2 - 1)(a_2 + a_3 - 1) \cdots (a_{k-1} + a_k - 1)$$

polyominoes with n cells having a strip of a_i cells in the ith row for each i.

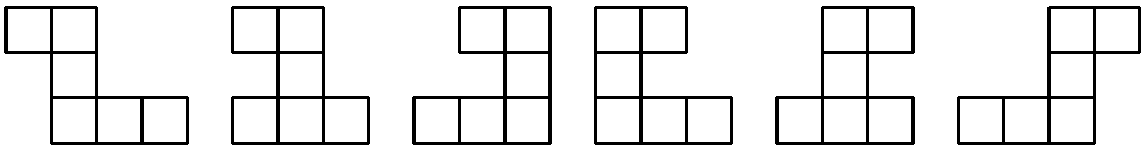


Figure 9. The 6 row–convex polyominoes with 6 cells corresponding to the composition (3,1,2) of 6

It follows that if $b(n)$ is the number of row–convex polyominoes with n cells, then

$$b(n) = \sum (a_1 + a_2 - 1)(a_2 + a_3 - 1) \cdots (a_{k-1} + a_k - 1),$$

where the sum extends over all compositions $(a_1, \ldots, a_k)$ of n into k parts, for all k. $b(n)$, and the area generating function $B(z) = \sum_{n=1}^{\infty} b(n) z^n$, are given by

Theorem 2 [22]

$$b(n+3) = 5b(n+2) - 7b(n+1) + 4b(n), \quad \text{and} \quad B(z) = \frac{z(1-z)^3}{1 - 5z + 7z^2 - 4z^3} \tag{5}$$

for $n = 2, 3, \ldots$, where $b(1) = 1, b(2) = 2, b(3) = 6, \ldots$.

Corollary 1.

$\lim_{n\to\infty} (b(n))^{\frac{1}{n}} = \beta$ where β is the largest real root of $z^3 - 5z^2 + 7z - 4 = 0$; $3.20 < \beta < 3.21$

More general case was considered by Bousquet–Mélou [71] in 1996 for directed column–convex polyominoes. Using 'Temperley methodology' [70] and building on her earlier works [72–79] she has obtained the generating function $V(x, y, n)$ in which the variables x, y, n mark horizontal and vertical edges of a perimeter and the number of cells

Theorem 3 [71]

$$V(x, y, n) = y^2 \frac{\sum_{i=1}^{\infty} \frac{x^{2i}(y^2-1)^{i-1} n^{i(i+1)/2}}{(n)_{i-1}(y^2 n)_{i-1}(y^2 n)_i}}{1 - \sum_{i=1}^{\infty} \frac{x^{2i}(y^2-1)^{i-1} n^{i(i+1)/2}}{(n)_i (y^2 n)_{i-1}(y^2 n)_i}} \tag{6}$$

The method which produced by formula (6) is markedly versatile. Besides the directed column–convex polyominoes, that method can be handle e.g. directed convex, convex, column–convex poliominoes and also some special classes such that parallelogram polyominoes [80]. Some common results for directed polyominoes on the triangular and hexagonal lattice was obtained in [81].

On the contrary, the perimeter generating function $G(x, y)$ of column–convex polyominoes remained unknown for many years after Pólya's and Temperley's works. At last Delest [82, 83] applied the DSV–methodology [84–88] and the computer algebra program MACSYMA to obtain a formula for $G(x, x)$. Subsequently, Brak, etc., [89] rederived the function $G(x, x)$ using the Temperley methodology and *Mathematica*. Thus it turned out that the formula given in [82] can be written in a simpler form. The

result of Brak was generalized to the case $x \neq y$ by Lin [90] and confirmed by Feretić [91]. The following remarkably simple formula for $G(x, y)$ takes place:

$$G(x,y) = (1-y^2)\left[1 - \frac{2\sqrt{2}}{3\sqrt{2} - \sqrt{1+x^2+\sqrt{(1-x^2)^2 - 16x^2y^2/(1-y^2)^2}}}\right] \tag{7}$$

The area and perimeter generating functions of column–convex polyominoes and directed column–convex polyominoes were obtained also by Brak, etc., [92], Delest and Dulucq [93], Feretić [94].

4. Some related topics

Some another results concerning the discussed topic can be found in [95–99] results related to the classical cell-growth problem can be found in [100–116]. Let us mention some of them.

The polyominoes' problem defined by two vectors has been proposed Navit in 1992 in the course of the seminar held at the Dipartimento di Sistemi e Informatica di Fireze, on September 1992, on the subject *Tiling the plane with a horizontal bar h_m and a vertical bar v_n*. It is the problem of establishing the existence of a polyomino with a given number of cells in every column and every row. The problem is solved by Lungo [109] for the following classes of polyominoes: directed column-convex, directed convex, and parallelogram. The problem is also solved in the class of convex polyominoes in a particular case. Also, for each of these classes an algorithm is defined which controls the existence of a polyomino for given vectors.

The following problem concerns polyominoes radically different from convex ones.

Problem 12. [**29**] *Find the smallest natural number n such that there exists a polyomino with n cells and with no row or column consisting of just a single strip of cells*

An example of a polyomino with 21 cells with this property is shown in Figure 10.

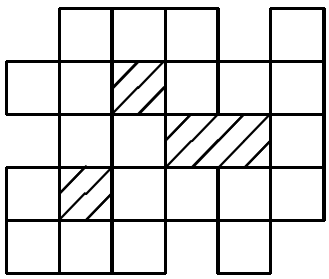


Figure 10. A polyomino with 21 cells and with no row or column a single strip of cells

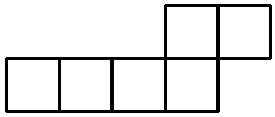


Figure 11. Snaky polyomino

Achievement games for polyominoes are frequently discussed in the literature [18–20,102,111,112,116]. For a given polyomino P two players A and B alternately mark the cells of the tessellation as game board. The player who first completes a copy of P with his marks wins the game. A polyomino P is called a winner if the first player A can win regardless of the moves made by B. Otherwise, P is called a loser.

For the triangular tessellation there are three winners and all other polyominoes are losers (see [20]). For the square tessellation 11 polyominoes are known to be winners. All others except one undecided polyomino, called Snaky (see Fig.11), are losers [111]. For the hexagonal tessellation all but five polyominoes with at most five cells are determined as winners or losers [116]. It may be remarked that for the five platonic solids as game boards all winners and losers are determined in [112].

Acknowledgements

The author thank the Com^{2}MaC at Pohang University of Science and Technology, The Republic of Korea, for its hospitality.

References

[1] F. Harary, Graphical enumeration problems, *In: Graph Theory and Theoretical Physics,* (chapter 1) 1–41, Academic Press, London, (1967).

[2] F. Harary, The cell–growth problem and its attempted solutions, *Beitr. Graphentheorie, Int. Kolloquium Manebach (DDR) 1967,* 49–60 (1968).

[3] F. Harary and R.C. Read, The enumeration of tree-like polyhexes, *Proc. Edinburgh Math. Soc.* **17** 1–15 (1970).

[4] F. Harary, *Graph Theory,* Addison-Wesley, Reading, MA, 2nd printing, (1971).

[5] F. Harary and E.M. Palmer, A survey of graphical enumeration problems, *Survey Combin. Theory,* Sympos. Colorado State Univ., Colorado 1971, 259–275 (1973).

[6] E.M. Palmer, Variations of the cell-growth problem, *In: Graph Theory and Applications, Proc. Conf. Western Michigan University, May 10–13 (1972),* 215–224, Berlin (1972).

[7] F. Harary and E.M. Palmer, *Graphical enumeration,* Academic Press, New York, (1973).

[8] F. Harary, Unsolved problems in the enumeration of graphs, *Publ. Math. Inst. Hungar. Acad. Sci.* **5** 1–20 (1960).

[9] S.W. Golomb, Checker boards and polyominoes, *Amer. Math. Mont.* **61** 675–682 (1954).

[10] S.W. Golomb, *Polyominoes,* Scribner, New York, (1965).

[11] S.W. Golomb, *Polyominoes: puzzles, patterns, problems, and packings,* 2nd, rev. and exp. ed., NJ: Princeton University Press, Princeton, (1994).

[12] M. Gardner, *Mathematics, magic and mystery,* Dover Publications, New York, (1956).

[13] M. Gardner, *New mathematical diversions,* Math. Assoc. Amer., Washington, (1995).

[14] G.E. Martin, *Polyominoes. A guide to puzzles and problems in tiling,* Math. Assoc. Amer., Washington, (1991).

[15] A.L. Clarke, Isoperimetrical polyominoes, *J. Recreational Math.* **13** 18–25 (1980).

[16] K. Scherer, Some new results on Y-pentominoes, *J. Recreational Math.* **12** 201–204 (1980).

[17] K. Scherer, Minimal fault-free rectangles packed with I_n-polyominoes, *J. Recreational Math.* **13** 4-6 (1980).

[18] F. Harary and C. Leary, Latin square achievement games, *J. Recreational Math.* **16** 241–246 (1984).

[19] F. Harary, Achievement and avoidance games on finite configurations, *J. Recreational Math.* **16** 182-187 (1984).

[20] F. Harary and H.Harborth, Achievement and avoidance games with triangular animals, *J. Recreational Math.* **18** 110–116 (1986).

[21] R.C. Read, Contributions to the cell-growth problem. *J. Can. Math.* **19** (1) 1–20 (1962).

[22] D.A. Klarner, Some results concerning polyominoes, *Fibonacci Quarterly* **3** 9–20 (1965).

[23] D.A. Klarner, Cell–growth problems, *Can. J. Math.* **19** (4) 851–863 (1967).

[24] M. Eden, A two–dimensional growth process, *Proceedings of the Fourth Berkely Symposium on Mathematical Statistics and Probability,* **4** 223–239 (1961).

[25] D.A. Klarner and R.L. Rivest, A procedure for improving the upper bound for the number of n-ominoes, *Can. J. Math.* **25** 585–602 (1973).

[26] W.F. Lunnon, Counting polyominoes, *In: Computers in number theory,* 347–372, Academic Press, London, (1971).

[27] W.F. Lunnon, Counting hexagonal and triangular polyominoes, *In: Graph theory and computing,* (Ed. by R.C. Read) 87–100, Academic Press, (1972).

[28] D.H. Redelmeier, Counting polyominoes: yet another attack, *Discrete Math.* **36** (2) 191–204 (1981).

[29] D.A. Klarner, Polyominoes, *In: Handbook of discrete and computational geometry,* (Ed. by J.E. Goodman et al.) 225–240, Boca Raton, FL: CRC Press Series on Discrete Mathematics and its Applications, (1997).

[30] D.A. Klarner and W. Satterfield, The number of width-k $n-$ominoes, *to appear*

[31] N. Trinajstić, Z. Jericević, J.V. Knop, W.R. Müller and K. Szymanski, Computer generation of isomeric structures, *Pure & Appl. Chem.* **55** 379–390 (1983).

[32] J.V. Knop, K. Szymanski, Z.Jericević and N.Trinajstić, On the total number of polyhexes, *MATCH* **16** 119–134 (1984).

[33] E. Clar. *The aromatic sextet,* Wiley, London, (1972).

[34] J.R. Dias, *Handbook of polycyclic hydrocarbons. Part A. Benzenoid hydrocarbons,* Elsevier, Amsterdam, (1987).

[35] I. Gutman and S.J. Cyvin, *Introduction to the theory of benzenoid hydrocarbons,* Springer–Verlag, Berlin, (1989).

[36] A.N. Balaban and F. Harary, Enumeration and proposed nomenclature of benzenoid cata-condensed polycyclic aromatic hydrocarbons, *Tetrahedron* **24** 2505–2516 (1968).

[37] A.T. Balaban, Ed., *Chemical applications of graph theory,* Academic Press, London, (1976).

[38] R.J. Wilson and L.W. Bieneke, Eds., *Applications of graph theory,* Academic Press, London, (1979).

[39] R.B. King, Ed., *Chemical applications of topology and graph theory,* Elsevier, Amsterdam, (1983).

[40] I. Stojmenović, R. Tošić and R. Doroslavčki, An algorithm for generating and counting hexagonal systems, *In: Proc. 6 Yugoslav Seminar on Graph Theory,* 189–198, Dubrovnik, (1985).

[41] J.V. Knop, W.R. Müller, K. Szymanski and N. Trinajstić, *Computer generation of certain classes of molecules,* SKTN, Zagreb, (1985).

[42] He Wenchen and He Wenjie, Generation and enumeration of planar polycyclic aromatic hydrocarbons, *Tetrahedron* **42** (19) 5291–5299 (1986).

[43] N. Trinajstić, *Mathematics and computational concepts in chemistry,* Horwood, Chichester, (1986).

[44] R.B. King and D.H. Rouvray, *Graph theory and topology in chemistry,* Elsevier, Amsterdam, (1987).

[45] G. Pólya and R.C. Read, *Combinatorial enumeration of groups, graphs and chemical compounds,* Springer–Verlag, New York, (1987).

[46] B.N. Cyvin, J. Brunvoll, S.J. Cyvin and I. Gutman, All-benzenoid systems: enumeration and classification of benzenoid hydrocarbons. VI, *MATCH* **23** 163–173 (1988).

[47] D.H. Rouvray, Ed., *Computational chemical graph theory,* Nova Science Publishers, Commack, NV, (1990).

[48] N. Trinajstić, S. Nikolić, J.V. Knop, Müller and K. Szymanski, *Computational chemical graph theory: characterization, enumeration and generation of chemical structures by computer methods,* Simon & Schuster, New York, (1991).

[49] N. Trinajstić,*Chemical graph theory,* CRC Press, Boca Raton, 2nd edition, (1992).

[50] B.N. Cyvin, J. Brunvoll, Rongsi Chen and S.J. Cyvin, Coronenic coronoids: a course in chemical enumeration, *MATCH* **29** 131–142 (1993).

[51] B.N. Cyvin, Fuji Zhang, Xiaofeng Guo, J. Brunvoll and S.J. Cyvin, On the total number of polyhexes with ten hexagons, *MATCH* **29** 143–163 (1993).

[52] E.V. Konstantinova, The constructive enumeration of square animals, preprint at Com^{2}MaC, POSTECH, 2000, no. 10, 55 pp. (http://com2mac.postech.ac.kr/resorce/pre00⊥ext.htm)

[53] E.V. Konstantinova, The constructive enumeration of triangular animals, preprint at Com^{2}MaC, POSTECH, 2000, no. 21, 33 pp. (http://com2mac.postech.ac.kr/resorce/pre00⊥ext.htm)

[54] E.V. Konstantinova, Enumeration and generation of animals, report at Com^{2}MaC (2001) 13pp.

[55] N. Trinajstić, On the classification of polyhexes, *J. Math. Chem.* **9** 373–380 (1992).

[56] A.A. Dobrynin, The effective algorithm of generation for graphs of unbranched hexagonal systems, *Vychisl. Sist.* **130** 3–38 (1989).

[57] B.N. Cyvin, J. Brunvoll, S.J. Cyvin and A.A. Dobrynin, Enumeration of unbranched catacondensed systems of congruent polygons, *Vychisl. Sist.* **155** 3–14 (1996).

[58] S.J. Cyvin, B.N. Cyvin and J. Brunvoll. Polycyclic conjugated hydrocarbons with arbitrary ring sizes, *J. Mol. Struct.* **300** 9–22 (1993).

[59] B.N. Cyvin, J. Brunvoll and S.J. Cyvin, Isomer enumeration of unbranched catacondensed polygonal systems with pentagons and heptagons, *MATCH* **34** 109–121 (1996).

[60] S.J. Cyvin, Generalization of extremal hexagonal animals (polyhexes), *J. Math. Chem.* **9** 389–390 (1992).

[61] F. Harary and H.Harborth, Heiko extremal animals, *J. Comb. Inf. Syst. Sci.* **1** 1-8 (1976).

[62] F. Harary, E.M. Palmer and R.C. Read, On the cell-growth problem for arbitrary polygons, *Discrete Math.* **11** (3–4) 371–389 (1975).

[63] R.E. Pippet and L.W. Beineke, An acyclic cell-growth problem, *In: Proc. Symp. Prague, 1974,* 441–454, Academ. Praha, (1975).

[64] R.K. Guy, Dissecting a polygon into triangles, *Bull. Malayan Math. Soc.* **5** 57–60 (1968).

[65] T. Motzkin, Relations between hyper–surface cross–ratios and a combinatorial formula for partitions of a polygon, for permanent preponderance, and for non–associative products, *Bull. Am. Math. Soc.* **54** 352–360 (1948).

[66] N.J.A. Sloane, *Handbook of integer sequences,* Academic Press, New York, (1973).

[67] D. Dhar, Exact solution of a directed site animals enumeration problem in three dimensions, Phys. Rev. Lett. **51** 853–856 (1983).

[68] D. Dhar, Equivalence of the two-dimensional directed site animal problem to Baxter's hard square lattice gas model, *Phys. Rev. Lett.* **49** 959–962 (1983).

[69] G. Pólya, On the number of certain lattice polygons, *J. Combin. Theory* **6** 102–105 (1969).

[70] H.N.V. Temperley, Combinatorial problems suggested by the statistical mechanics of domains and of rubber–like molecules, *Phys. Rev.* **103** 1–16 (1956).

[71] M. Bousquet–Mélou, A method for the enumeration of various classes of column-convex polygons, *Discrete Math.* **154** (1–3) 1–25 (1996).

[72] M. Bousquet–Mélou, Convex polyominoes and heaps of segments, *J. Phys. A, Math. Gen.* **25** (7) 1925–1934 (1992).

[73] M. Bousquet–Mélou, Une bijection entre les polyominos convexes diriges et les mots de Dyck bilateres. (A bijection between convex and directed polyominoes and the words of the bilateral Dyck language, (French), *Inform. Theor. Appl.* **26** (3) 205–219 (1992).

[74] M. Bousquet–Mélou and X.G. Viennot, Empilements de segments et q-enumeration de polyominos convexes diriges. (Heaps of segments and q-enumeration of directed convex polyominoes), (French), *J. Comb. Theory, Ser.A* **60** (2) 196–224 (1992).

[75] M. Bousquet–Mélou, q-enumeration de polyominos convexes. (q-enumeration of convex polyominoes), (French), *J. Comb. Theory, Ser. A* **64** (2) 265–288 (1993).

[76] M. Bousquet–Mélou, Codage des polyominos convexes et equations pour l'enumeration suivant l'aire. (Coding the convex polyominoes and equations for the enumeration according to the area), (French), *Discrete Appl. Math.* **48** (1) 21–43 (1994).

[77] M. Bousquet–Mélou, Polyominoes and polygons, *Contemp. Math.* **178** 55–70 (1994).

[78] M. Bousquet–Mélou and J.-M. Fedou, The generating function of convex polyominoes: the resolution of a q-differential system, *Discrete Math.* **137** (1–3) 53–75 (1995).

[79] M. Bousquet–Mélou and A.R. Conway, Enumeration of directed animals on an infinite family of lattices, *J. Phys. A, Math. Gen.* **29** (13) 3357–3365 (1996).

[80] R.A. Sulanke, Three recurrences for parallelogram polyominoes, *J. Difference Equ. Appl.* **5** (2) 155–176 (1999).

[81] M. Bousquet–Mélou, New enumerative results on two-dimensional directed animals, *Discrete Math.* **180** (1-3) 73–106 (1998).

[82] M.P. Delest, Generating functions for column-convex polyominoes, *J. Comb. Theory, Ser.A* **46** (1) 12–31 (1988).

[83] M. Delest, Enumeration of polyominoes using Macsyma, *Theor. Comput. Sci.* **79** 209-226 (1991).

[84] M.P. Delest and G. Viennot, Algebraic languages and polyominoes enumeration, *Lect. Notes Comput. Sci.* **154** 173–181 (1983).

[85] M.P. Delest, D. Gouyou-Beauchamps and B. Vauquelin, Enumeriation of parallelogram polyominoes with given bond and site perimeter, *Graphs Comb.* **3** 325–339 (1987).

[86] D. Kim, The number of convex polyominoes with given perimeter, *Discrete Math.* **70** (1) 47–51 (1988).

[87] M. Bousquet–Mélou, Convex polyominoes and algebraic languages, *J. Phys. A, Math. Gen. 25* **7** 1935–1944 (1992).

[88] M.P. Delest and J.M. Fedou, Enumeration of skew Ferrers diagrams, *Discrete Math.* **112** (1–3) 65–79 (1993).

[89] R. Brak, A.J. Guttmann and I.G. Enting, Exact solution of the row-convex polygon perimeter generating function, *J. Phys. A, Math. Gen.* **23** (12) 2319–2326 (1990).

[90] K.Y. Lin, Perimeter generating function for row–convex polygons on the rectangular lattice, *J. Phys. A* **23** 4703–4705 (1990).

[91] S. Feretić, A new way of counting the column-convex polyominoes by perimeter, *Discrete Math.* **180** (1-3) 173–184 (1998).

[92] R. Brak and A.J. Guttman, Exact solution of the staircase and row-convex polygon perimeter and area generating function, *J. Phys. A, Math. Gen.* **23** 4581–4588 (1990).

[93] M.P. Delest and S. Dulucq, Enumeration of directed column–convex animals by perimeter and area, *Croatia Chemica Acta* **66** 59–80 (1993).

[94] S. Feretić, An alternative method for q-counting directed column-convex polyominoes, *Discrete Math.* **210** (1–3) 55–70 (2000).

[95] A.J. Guttmann, On the number of lattice animals embeddable in the square lattice, *J. Phys. A* **15** 1987–1990 (1982).

[96] D. Gouyou-Beauchamps and G. Viennot, Equivalence of the two-dimensional directed animal problem to a one–dimensional path problem, *Adv. Appl. Math.* **9** (3) 334–357 (1988).

[97] V. Domocos, A combinatorial method for the enumeration of column-convex polyominoes, *Discrete Math.* **152** (1–3) 115–123 (1996).

[98] H. Harborth and C.Thuermann, Limited snakes of polyominoes, *Congr. Numerantium* **133** 211–218 (1998).

[99] P. Leroux, E. Rassart and A. Robitaille, Enumeration of symmetry classes of convex polyominoes in the square lattice, *Adv. Appl. Math.* **21** (3) 343–380 (1998)

[100] F. Harary and B. Manvel, Reconstruction of square-celled animals, *Bull. Soc. Math. Belgique* **24** 375–379 (1972).

[101] C. Berge, C.C. Chen, V. Chvatal and C.S. Seow, Combinatorial properties of polyominoes, *Combinatorica* **1** 217–224 (1981).

[102] M. Erickson and F. Harary, Picasso animal achievement games, *Bull. Malays. Math. Soc., II. Ser.* **6** 37–44 (1983).

[103] A. Fontaine and G.E. Martin, Polymorphic polyominoes, *Math. Mag.* **57** 275–283 (1984).

[104] S.W. Golomb, Polyominoes which tile rectangles, *J. Comb. Theory, Ser.A* **51** (1) 117–124 (1989).

[105] K.A. Dahlke, A heptomino of order 76, *J. Comb. Theory, Ser.A* **51** (1) 127–128 (1989).

[106] H. Harborth, Some mosaic polyominoes, *Ars Comb.* **29A** 5–12 (1990).

[107] H. Harborth and H. Weiss, Minimum sets of partial polyominoes, *Australas. J. Comb.* **4** 261–268 (1991).

[108] F. Maire, Polyominos and perfect graphs, *Inf. Process. Lett.* **50** (2) 57–61 (1994).

[109] A. Del Lungo, Polyominoes defined by two vectors, *Theor. Comput. Sci.* **127** (1) 187–198 (1994).

[110] L. Alonso and R. Cerf, The three dimensional polyominoes of minimal area, *J. Comb.* **3** 371-409 (1996).

[111] H. Harborth and M.Seemann, Snaky is an edge–to–edge loser, *Geombinatorics* **5** 132–136 (1996).

[112] Jens-P. Bode and H. Harborth, Achievement games on platonic solids, *Bull. Inst. Comb. Appl.* **23** 23–32 (1998).

[113] P. Duchon, Q-grammars and wall polyominoes, *Ann. Comb.* **3** (2–4) 311–321 (1999).

[114] M. Bousquet-Mélou, A.J. Guttmann, W.P. Orrick and A. Rechnitzer, Inversion relations, reciprocity and polyominoes, *Ann. Comb.* **3** (2–4) 223–249 (1999).

[115] Fuji Zhang and Xiaofeng Guo, The enumeration of several classes of hexagonal systems, *Acta Math. Appl. Sin., Engl. Ser.* **15** 65-71 (1999).

[116] Jens-P. Bode and H. Harborth, Hexagonal polyomino achievement, *Discrete Math.* **212** (1–2) 5–18 (2000).

Frank Zielen

Rigorose und Perturbative Konstruktion von ϕ^4-Trajektorien

Oktober 1998

Rigorose und Perturbative Konstruktion von ϕ^4-Trajektorien

Als Diplomarbeit vorgelegt von
Frank Zielen

Institut für Theoretische Physik I
Westfälische Wilhelms–Universität Münster

Oktober 1998

Inhaltsverzeichnis

Einführung

Im Jahre 1982 erhielt K. G. Wilson den Nobelpreis für Physik als Würdigung seiner Forschungsarbeit [Wil71, WK74] auf dem Gebiet der Renormierungsgruppe (RG). Dieser nichtperturbative Zugang zur Theorie kritischer Phänomene entwickelte sich in den letzten 25 Jahren zu einem machtvollen Werkzeug in der Statistischen Mechanik und der Quantenfeldtheorie (QFT).

Das Grundprinzip der RG ist der Skalenbegriff. Die physikalisch relevanten Größen, die Response- oder Greensfunktionen, erhält man durch Ableiten der erzeugenden Funktionale, die in der Statistik über die Zustandssumme und in der QFT über das Pfadintegral definiert werden.[1] In diesen spiegelt sich die meistens hohe und nicht selten unendliche Zahl von Freiheitsgraden des betrachteten Systems wider. Die Berechnung dieser hochdimensionalen Objekte (Integrale, Summen) führt man auf eine schrittweise, durch einen Skalenparameter organisierte Ausintegration von Freiheitsgraden zurück. Dies geschieht z.B. im kubisch diskretisierten, euklidischen Ortsraum durch eine Teilsummation über Würfel, deren Kantenlänge ein endliches, ganzzahliges Vielfaches der Gitterkonstante beträgt [GK84]. Eine äquivalente Möglichkeit bietet die Multiskalen-Zerlegung des Propagators einer Theorie in Impulsscheiben [BG95].

Die RG ist somit eine Skalentransformation, die eine Theorie, die über den Boltzmann-Faktor oder ein Wechselwirkungsfunktional definiert ist, auf eine effektive, ausgedünnte, gröbere Theorie abbildet. Ein wesentliches Merkmal dieser renormierten Theorie ist eine kleinere Korrelationslänge ξ'. Unter Anwendung der Blockspintransformation (BST) erkennt man den Zusammenhang[2]

$$\xi' = \frac{\xi}{L} \,. \tag{1}$$

[1] Im weiteren Verlauf verwenden wir die Begriffe Zustandssumme und Pfadintegral gleichwertig.

[2] $L \in \mathbb{N}_2$ ist ein Vielfaches der Raumgitterkonstante a und folglich La die Seitenlänge eines Blocks.

Diese Eigenschaft macht die RG zu einem idealen Untersuchungswerkzeug kritischer Phänomene. Kontinuierliche Phasenübergänge (PÜ) zeichnen sich am kritischen Punkt durch eine Nichtanalytizität in einer zweiten partiellen Ableitung des thermodynamischen Potentials aus. Die divergierende Korrelationslänge als wesentliches Merkmal eines PÜ's 2. Ordnung bleibt nach (1) unter Anwendung einer Renormierungsgruppentransformation (RGT) divergent: kritische Systeme sind (fast) skaleninvariant. Alle Theorien, die in Einzugsbereichen von kritischen Fixpunkten einer RGT liegen, besitzen dasselbe Verhalten wie die assoziierten Fixpunkte und bilden sogenannte Universalitätsklassen, die nur durch wenige Parameter wie die Raumdimension oder die lokalen Freiheitsgrade charakterisiert werden. Kritische Systeme lassen sich unabhängig von ihren mikroskopischen Wechselwirkungen beschreiben.[3]

Von einer auf dem Gitter diskretisierten QFT fordert man, daß ihre Korrelationsfunktionen, aus denen sich die physikalischen Größen, wie z.B. die Masse m des leichtesten Teilchens, bestimmen, im Kontinuumslimes (Gitterkonstante $\to 0$) endliche Werte annehmen. Der Zusammenhang

$$\xi = \frac{1}{am} \tag{2}$$

[GK84] bedingt für $a \to 0$ folglich die Divergenz der Korrelationslänge. Die Gitterfeldtheorie muß kritisch sein und eignet sich als Proband für die RG.

Die Erzeugung nicht lokaler Terme in der effektiven Theorie kompliziert die mathematische Behandlung der RGT erheblich. Aus diesem Grund arbeitet man mit hierarchischen Approximationen [Dys69, GK84, Por90], die ein ähnliches kritisches Verhalten wie die vollen Modelle aufweisen und lokalitätserhaltend sind. Sie dienen als vereinfachtes Versuchsfeld zur Entwicklung neuer RG-Strategien und repräsentieren eine Klasse von eigenständigen, untersuchungswürdigen Systemen der Statistischen Physik.

Eine Hauptanwendung der RG ist die konstruktive Behandlung der ϕ^4-artig gestörten skalaren, freien Feldtheorie. Nach einer Idee von C. Wieczerkowski parametrisiert man die durch RG-Iteration erzeugten Flüsse in der ϕ^4-Kopplung und konstruiert RG-invariante Kurven, die im freien Feld beginnen und tangential zur Störung liegen. RGT lassen sich somit einfach durch Entlangfahren der Trajektorie bestimmen.

Die Arbeit gliedert sich wie folgt:

Im ersten Kapitel geben wir eine Definition der RG auf dem Gitter und im hierarchischen Modell.

[3] Nur die Reichweite der Wechselwirkung ist noch von Belang.

Im zweiten Abschnitt liefern wir die rigorose Konstruktion der ϕ^4-Trajektorie in der hierarchischen Approximation. Das benutzte Verfahren basiert auf dem *construction mapping theorem*. Wir beweisen seine Anwendbarkeit für alle Dimensionen $2 < D < 4$ und behandeln den Sonderfall $D = 3$ explizit.

Im dritten Kapitel präsentieren wir eine perturbative Berechnung der ϕ^4_3-Kurve im Rahmen der RGT auf dem Gitter. Hierzu führen wir die Aufgabenstellung auf das bereits gelöste Problem im Kontinuum zurück.

Der vierte Abschnitt beschreibt den (nicht geglückten) Konstruktionsversuch der ϕ^4_4-Trajektorie im hierarchischen Modell.

Abschließend geben wir eine Zusammenfassung und präsentieren Ansätze und Ideen für eine weitere Behandlung des Themas. Anhänge über elementare, mathematische Notationen und Formeln vervollständigen das Bild der Arbeit.

Zur besseren Lesbarkeit des Inhaltsverzeichnisses sowie des weiteren Textes möchten wir bereits an dieser Stelle die wichtigsten Abkürzungen im Rahmen der Renormierungsgruppe präsentieren. Es stehen im folgenden RG für Renormierungsgruppe, T für Transformation, H für hierarchisch und G für Gitter, so daß z.B. eine Übersetzung des Kürzels HRGT keine Probleme bereiten dürfte. Desweiteren ergänzen wir Abkürzungen nicht um Fall spezifische Endungen.

Kapitel 1

Die RG

Wir beginnen dieses Kapitel mit der Erstellung eines Begriffs- und Formelapparates zur Behandlung skalarer Gitterfeldtheorien. Das Spin-Gitter-Modell ist das Demonstrationsobjekt der Statistischen Physik zur Untersuchung von PÜ und kritischen Phänomenen schlechthin. Prominentester Vertreter ist das D-dimensionale Ising-Modell, welches die spontane Magnetisierung eines Ferromagneten erklärt. Im Jahre 1925 bewies E. ISING, daß für $D = 1$ kein PÜ existiert. Die analytische Lösung L. ONSAGER's in zwei Dimensionen zeigt hingegen einen PÜ auf. Für $D = 3$ steht eine exakte Behandlung noch aus, aber die RG bewährt sich auch hier als ideales Werkzeug, z.B. zur Berechnung kritischer Exponenten. In dieser Arbeit legen wir die diskrete Darstellung einer skalaren QFT zugrunde.

Im nächsten Abschnitt geben wir dem Leser eine mathematische Definition der RG für Gitterfeldtheorien an die Hand. Diese stützt sich auf die BST, die von L. P. KADANOFF [Kad66] erdacht wurde. Dessen Urform basiert auf einem Ising-Gitter und Blockspins, die nur zwei unterschiedliche Werte annehmen. Unser Zugang findet sich z.B. in Arbeiten von K. GAWEDZKI und A. KUPIAINEN [GK84] oder C. WIECZERKOWSKI [Wie98].

Im Anschluß erarbeiten wir zwei Formen der RGT, die sich aus den zugrunde liegenden Modellklassen ableiten und nur die Transformation des Wechselwirkungsanteils beinhalten. Die eine bezieht sich auf Gittertheorien, deren freier Anteil durch den perfekten masselosen Propagator ν_{perf} beschrieben wird. Die andere ergibt sich für Systeme, deren kinetischer Part durch den hierarchischen Propagator ν_{hier} gegeben ist, und deren Potentiale lokal sind.

1.1 Grundlagen

Die Sprache der Teilchenphysik ist die Quantenfeldtheorie. Mit ihr ist es möglich, den für das Experiment so relevanten Streuquerschnitt zu berechnen, der die Fragen nach Reaktionswahrscheinlichkeiten und Zerfallslängen beantwortet. Die Streumatrix(-elemente) bestimmt man mit Hilfe der Reduktionsformel aus den amputierten Greensfunktionen [Ryd96], und die Korrelatoren gewinnt man durch Funktionalableitung aus dem von R. FEYNMAN erdachten und Nobelpreis gewürdigten Pfadintegral. Dieses Objekt wird durch eine Lagrangedichte oder die über die Lagrangedichte bestimmte Wirkung definiert. Lagrangedichte und Wirkungsfunktional nennen wir in Zukunft Theorie. In dieser werden die Teilchensorten durch die Dimensionalität der Felder und der assoziierten Algebra festgelegt. Wechselwirkungen können sowohl untereinander bestehen (inklusive Selbstwechselwirkungen) als auch von äußeren Quellen herrühren.

Zudem legen wir die euklidische Raum-Zeit in D Dimensionen (im allgemeinen $D = 2, 3, 4$) zugrunde. Die Konsequenz ist eine Vereinfachung der Rechnungen. Mittels Wick-Rotation lassen sich die Schwingerfunktionen (euklidische Greensfunktionen) in die physikalischen Wightmanfunktionen (auf dem Minkowski-Raum lebende n-Punkt-Funktionen) fortsetzen. Gegner des euklidischen Formalismus kritisieren, daß eine Integration der allgemeinen Relativitätstheorie unmöglich sei. Für die Fragestellungen der Hochenergiephysik ist das euklidische Pfadintegral jedoch besser geeignet als das minkowskische Pendant. Ein weiterer Bonuspunkt des Euklidischen Zugangs ist die formale Äquivalenz des Feynmanschen Integrals zur Zustandssumme in der klassischen Statistik. Durch sie entsteht eine Verbindung zu einem intensiv erforschten Zweig der Physik.

Als letztes diskretisieren wir die euklidische Raum-Zeit, indem wir statt des überabzählbaren $\mathbb{R}^D$ ein unendliches D-dimensionales kubisches Gitter mit der Gitterkonstanten a einführen.

Definition 1.1.1 (Das Gitter Λ)
Es seien $a \in \mathbb{R}_\star$ und

$$\Lambda(a) := a\mathbb{Z}^D \tag{1.1}$$
$$\Lambda(0) := \lim_{a\to 0} \Lambda(a) := \mathbb{R}^D \tag{1.2}$$

OBdA sei $a \in \mathbb{R}^+$, da $\Lambda(a) = \Lambda(-a)$. Der Gitterkontinuumslimes (1.2) ist formaler Natur.[1] Die Probleme eines Grenzübergangs zwischen abzählbaren

[1] In welcher Norm sollte dieser auch ausgeführt werden?

und überabzählbaren Mengen behandeln wir im Abschnitt 3.1.

Die Vorteile der Gitterfeldtheorien sind offensichtlich: Schon 20 Jahre vor der Entwicklung des Pfadintegrals wurden Untersuchungen von Spin-Gittern, z.B. Ising-, XY- oder Heisenbergmodelle, zur Beschreibung kritischer Phänomene betrieben. Darüber hinaus werden die Wohldefiniertheit des Pfadintegrals und unter entsprechenden Annahmen (z.B. lokalisierte Wirkungen) auch seine Berechnung (z.B. Faktorisierung) vereinfacht. Der entscheidende Faktor ist jedoch, daß die auf dem Gitter $\Lambda(a)$ definierte Theorien eine eingebaute Impulsbetragsobergrenze von $\frac{2\pi}{a}$ besitzen, die man auch *UV-cutoff* nennt. Für $a \to 0$ verschwindet diese Beschneidung. Die Verwendung eines räumlich begrenzten Gitters hätte eine Untergrenze des Impulsbetrags zur Folge. Die Theorie besäße einen *IR-cutoff*.

Ziel jeder diskretisierten QFT ist es, daß die Korrelationsfunktionen im Kontinuumslimes endlich bleiben.

In dieser Arbeit wollen wir die einfachste aller Feldtheorien betrachten: ein reelles skalares Feld, dessen kinetischer bzw. freier Anteil durch einen Propagator beschrieben wird, ergänzt um ein beliebiges Selbstwechselwirkungspotential. Der Urvater aller skalaren Feldtheorien ist die Klein-Gordon Gleichung, welche freie, ungeladene Spin-0 Teilchen beschreibt. Wir beginnen mit der Definition des Feldraumes:

Definition 1.1.2 (Der Konfigurationsraum $\mathcal{H}(a)$)

$$\mathcal{H}(a) := \left\{ \phi : \Lambda(a) \to \mathbb{R} \Big| \sum_{x \in \Lambda(a)} |\phi(x)| < \infty \right\} \tag{1.3}$$

Da $\Lambda(a)$ als endliches, kartesisches Produkt der abzählbaren Menge $\mathbb{Z}$ abzählbar ist, existiert ein Isomorphismus zwischen $\mathcal{H}(a)$ und dem Vektorraum der betragsintegrablen Folgen l_1. Folglich ist auch $\mathcal{H}(a)$ ein Vektorraum und die Bilinearform

Definition 1.1.3 (Skalarprodukt auf $\mathcal{H}(a)$)

$$(\phi, \psi) := \int_{\Lambda(a)} d^D x \; \phi(x)\psi(x) := \sum_{x \in \Lambda(a)} a^D \phi(x)\psi(x) \tag{1.4}$$

vervollständigt $\mathcal{H}(a)$ zu einem Hilbertraum, da $l_1 \subsetneq l_2$ und $\| \cdot \|_2 \leq \| \cdot \|_1$

[MV92].[2] Die Fourier-Transformation in den Impulsraum

$$\tilde{\phi}(p) = \int_{\Lambda(a)} d^D x \; e^{-ipx} \; \phi(x) \tag{1.5}$$

ist für alle $p \in \mathbb{R}^D$ wohldefiniert und gitterperiodisch bezüglich $\Lambda\left(\frac{2\pi}{a}\right)$, so daß wir uns auf die Brillouin-Zone

Definition 1.1.4 (Der Impulsraum)

$$\tilde{\Lambda}(a) := \left(-\frac{\pi}{a}, \frac{\pi}{a}\right]^D \tag{1.6}$$

beschränken können [MM94]. Der Impulskonfigurationsraum $\tilde{\mathcal{H}}(a)$ ist, wie die Bezeichnung schon andeutet, das Bild des Konfigurationsraumes $\mathcal{H}(a)$ unter Fourier-Transformation und auch ein Hilbertraum bezüglich des Skalarproduktes

$$(\tilde{\phi}, \tilde{\psi}) := \int_{\tilde{\Lambda}(a)} \frac{d^D p}{(2\pi)^D} \; \overline{\tilde{\phi}(p)} \; \tilde{\psi}(p) \; . \tag{1.7}$$

Die Rücktransformation in den Ortsraum schreibt sich als

$$\phi(x) = \int_{\tilde{\Lambda}(a)} \frac{d^D p}{(2\pi)^D} \; e^{ipx} \; \tilde{\phi}(p) \; . \tag{1.8}$$

Als letzter Komponente des Pfadintegrals begegnen wir dem Wirkungsfunktional $S(\phi) = \frac{1}{2}(\phi, \nu^{-1}\phi) + V(\phi)$. Die Eigenschaften des freien Propagators ν und der Wechselwirkung V notieren wir in folgendem

Satz 1.1.5 (Der freie Propagator und die Wechselwirkung)
Der freie Propagator $\nu \in L(\mathcal{H}(a))$ modulo Nullmoden[3] ist eine Kovarianz und invariant gegenüber der Poincaré-Gruppe auf dem Gitter $\Lambda(a)$.[4] Der Wechselwirkungsterm $Z = e^{-V}$ ist ein positives, reelles Funktional über $\mathcal{H}(a)$.

[2] Die kanonische Wahl $\mathcal{H}(a) \cong l_2$ ist natürlich auch möglich.

[3] Der physikalische Ausdruck „modulo Nullmoden“ entspricht dem Ausschluß des Operatorkerns $\mathrm{Ker}(\nu) = \{\phi \in \mathcal{H}(a) | \nu\phi = 0\}$ aus dem Integrationsgebiet $\mathcal{H}(a)$. Auf dem Quotientenraum $\mathcal{H}(a) \backslash \mathrm{Ker}(\nu)$ ist ν dann injektiv und somit invertierbar. Diese Reduzierung des Pfadintegrals ist möglich, da der Operatorkern des Propagators bezüglich des Gaußschen Maßes eine Nullmenge darstellt - für $\nu \to 0$ konvergiert der Exponentialfaktor gegen Null.

[4] Das ist die Menge aller Gittertranslationen, -rotationen und -spiegelungen.

Ziehen wir den kinetischen Anteil der Wirkung zum formalen Maß des Pfadintegrals $\prod_{x\in\Lambda(a)} d\phi(x)$, erhalten wir ein Gaußsches Maß (B.2), und das erzeugende Funktional der Gitter-Greensfunktionen schreibt sich als

$$G(J) = \frac{\int_{\mathcal{H}(a)} d\mu_\nu(\phi) Z(\phi) e^{(\phi,J)}}{\int_{\mathcal{H}(a)} d\mu_\nu(\phi) Z(\phi)} \ . \tag{1.9}$$

1.1.1 Propagatoren der RG

Die RG bildet eine nackte Wirkung auf eine effektive Theorie ab. Durch Iteration dieses Prozesses erhält man eine Folge von Propagatoren und Wechselwirkungspotentialen. In diesem Abschnitt charakterisieren wir die Eigenschaften von Propagatoren 1.1.5 anhand ihrer Fourier-Transformierten.

Aufgrund der Linearität lassen sich die Propagatoren über ihre Operatorkerne darstellen,[5] die auf $\Lambda(a)\times\Lambda(a)$ definiert sind. Aus der Gittertranslationsinvarianz folgt

$$\nu(x,y) = \int_{\tilde{\Lambda}(a)} \frac{d^D p}{(2\pi)^D} e^{ip(x-y)} \tilde{\nu}(p) \ . \tag{1.10}$$

Alle weiteren Eigenschaften charakterisieren wir über die Fourier-Transformierte $\tilde{\nu} : \tilde{\Lambda}(a) \to \mathbb{R}_0^+$.[6] Aus der Spiegelsymmetrie ($\tilde{\nu} \in \mathbb{Z}_2(\tilde{\Lambda}(a))$)[7] folgt, daß ν selbstadjungiert ist. Ein allgemeiner Propagator gemäß 1.1.5 sei durch

$$\tilde{\nu}(p)^{-1} = c_0 + c_1 p^2 + \sum_{\mu=1}^{D} O(p_\mu^4) \tag{1.11}$$

gegeben. Motiviert ist diese Darstellung durch die kanonische Diskretisierung des Standardpropagators der skalaren Feldtheorie $(-\triangle + m^2)^{-1}(x,y)$, dessen inverse Fourier-Transformierte [MM94]

$$\tilde{\nu}^{-1}(p) = 2a^{-2} \sum_{\mu=1}^{D} (1 - \cos(p_\mu a)) + m^2 = 4a^{-2} \sum_{\mu=1}^{D} \sin^2\left(\frac{p_\mu a}{2}\right) + m^2 \tag{1.12}$$

lautet. Zwei wichtige Eigenschaften für die Behandlung wechselwirkender Theorien sind die Beschränkungen

$$\begin{array}{rcll} \|\tilde{\nu}\|_1 & = & \int_{\tilde{\Lambda}(a)} \frac{d^D p}{(2\pi)^D} |\tilde{\nu}(p)| < \infty & \textit{UV-cutoff} \\ \|\tilde{\nu}\|_\infty & = & \sup_{p\in\tilde{\Lambda}(a)} |\tilde{\nu}(p)| < \infty & \textit{IR-cutoff} \end{array} \tag{1.13}$$

[5] $\nu\phi(x) = \int_{\Lambda(a)} d^D y \, \nu(x,y) \, \phi(y)$

[6] Der Bildbereich $\mathbb{R}_0^+$ macht ν (modulo Nullmoden) zu einem positiv definiten Operator. Mit $\tilde{\nu}(p) = \tilde{\nu}(-p)$ zeigt man, daß ν reell ist.

[7] und der Translationsinvarianz

Gelten diese nicht, spricht man von UV- bzw. IR-Divergenzen. Da sich die Integration in der $\|\cdot\|_1$ Norm für $a > 0$ auf ein Kompaktum beschränkt, ergeben sich für Propagatoren gemäß (1.11) mit $|c_0|+|c_2| > 0$ in Dimensionen $D > 2$ endliche Ausdrücke. Sie besitzen einen *UV-cutoff*. Propagatoren mit $c_0 = 0$ sind IR-divergent.[8]

Der gitterinterne *UV-cutoff* ermöglicht eine perturbative Behandlung gestörter Theorien, da die in den zusammenhängenden Greensfunktionen auftauchenden Schleifen-Integrale $\nu(x, x)$ nicht divergieren [Ryd96].

Eine störungstheoretische Behandlung nicht regularisierter Theorien wird durch Renormierung der nackten Kopplungen (1) oder das Einfügen von Countertermen in den Lagrangian (2) möglich [Ryd96]. Bei diesen Methoden entwickelt man die auftretenden Divergenzen in den Schleifen-Integralen. Durch die Annahme unendlicher, nackter (renormierter) Kopplungen erzeugt man endliche, physikalische Größen, wie z.B. die Masse (1), oder die zusätzlichen, divergenten Counterterme heben die Divergenzen des Propagators auf (2). Diese Form der RG ist äquivalent mit dem Zugang von K. Gawedzki und A. Kupiainen, den wir im nächsten Kapitel erläutern.

Man mag sich abschließend fragen, warum wir den Propagatorbegriff so allgemein halten? Im Kontinuum betreibt man skalare Feldtheorie mit dem masselosen oder massebehafteten Klein-Gordon Operator. Abweichungen von diesem Propagator-Standard können in das Wechselwirkungspotential geschrieben werden.[9] Da die Diskretisierung des Laplace-Operators nicht eindeutig ist, existiert nicht *der* Gitterpropagator des skalaren Feldes. Hierzu vergleiche man z.B. den kanonisch diskretisierten masselosen ($m = 0$) Propagator (1.12) mit der perfekten masselosen Kovarianz ν_{perf} (1.43). Unsere allgemeine Form (1.11) trägt diesem Umstand Rechnung und erlaubt darüber hinaus die Verwendung abstrakterer Propagatoren.

1.2 Die Idee der RG

In diesem Kapitel leiten wir die auf der BST basierende RGT für Gitterfeldtheorien her. Wir orientieren uns dabei an [GK84, Wie98] und beginnen mit einer Re-Definition des Pfadintegrals.

[8] Im Kontinuum ($a = 0$) sind obige Normen nicht äquivalent! Zwei Gegenbeispiele sind z.B. $\tilde{\nu}(p) = 1$ und $\tilde{\nu}(p) = |p|^{-(D-1)} e^{-|p|^2}$. Für $D = 1$ gilt jedoch, daß $\|\cdot\|_1$ schwächer ist.

[9] Im allgemeinen möchte man jedoch, daß $Z = 1$ weiterhin die ungestörte Theorie beschreibt.

Satz 1.2.1
Es sei H ein reelles Funktional über $\mathcal{H}(a)$ definiert durch

$$H(\phi) = \frac{\int d\mu_\nu(\zeta) Z(\phi+\zeta)}{\int d\mu_\nu(\zeta) Z(\zeta)} \; . \tag{1.14}$$

Dann gilt für alle $J \in \mathcal{H}(a)$

$$G(J) = e^{(J,\nu J)} H(\nu J) \; . \tag{1.15}$$

Eine einfache Substitution liefert den Beweis [Geh97]. In Zukunft arbeiten wir mit dem Funktional H, aus dem die Quelle J entfernt wurde. H ist das erzeugende Funktional der Korrelationsfunktionen, deren äußere Propagatoren trunkiert sind (amputierte Greensfunktionen).

Ein Trick zur Berechnung von $H(\phi)$ liegt nun in der Reorganisierung der Integration. Statt sofort „in einem Rutsch“ über alle möglichen Feldkonfigurationen aus $\mathcal{H}(a)$ zu summieren (diese Anzahl ist im übrigen überabzählbar, da die Spins aus $\mathbb{R}$ stammen), führen wir diese Aufgabe schrittweise aus.

Wir teilen das (unendlich große) Gitter $\Lambda(a)$ in kubische Blöcke der Seitenlänge La, wobei $L \in \mathbb{N}_{\geq 2}$ und wir L den Blockparameter nennen. Durch L führen wir eine Skala ein.

Nun zerlegen wir die Gesamtintegration. Dazu betrachten wir bzgl. $\mathcal{H}(a)$ den Untervektorraum $\mathcal{H}_0(a)$, dessen Elemente die Eigenschaft auszeichnet, daß ihre Blockspins (d.h. die Mittelwerte der Spins bzgl. der oben erklärten Blöcke) verschwinden. Betrachten wir desweiteren den Untervektorraum $\mathcal{H}_c(a)$, dessen Felder auf Blöcken konstant sind, so sind wir in der Lage, das Pfadintegral in einfacher Weise zu zerlegen.

$$\int_{\mathcal{H}(a)} \mathcal{D}[\psi] f(\psi) = \int_{\mathcal{H}_c(a)} \mathcal{D}[\phi] \int_{\mathcal{H}_o(a)} \mathcal{D}[\zeta] f(\phi+\zeta) \tag{1.16}$$

Die Integration über $\mathcal{H}_0(a)$ entspricht einer Ausintegration kurzreichweitiger Wechselwirkungen ($\leq La$), sogenannter Fluktuationen. Wir entfernen Impulse $|p| \in [\frac{2\pi}{aL}, \frac{2\pi}{a}]$ aus der Theorie. Dieses Verfahren läßt sich iterieren, wenn man weitere Blockungen mit den Skalenparametern L^2,L^3 usw. ausführt.

Es ist günstiger, den Feldkonfigurationsraum $\mathcal{H}_c(a)$ nach $\mathcal{H}(a)$ zu übertragen[10] und wiederum eine Blockung der Größe L vorzunehmen. Auf diese Weise gestaltet sich eine Iteration der BST wesentlich angenehmer. Im

[10] Diese Transformation ist möglich, da $\mathcal{H}_c(a) \cong \mathcal{H}(a)$.

nächsten Abschnitt kleiden wir dieses Vorgehen in ein mathematisches Gewand.

Das Prinzip der BST folgt Überlegungen im Ortsraum. Die schrittweise Blockung der Spins ist jedoch nichts anderes als die Multiskalen-Zerlegung des freien Propagators im Impulsraum [BG95, Geh97]. Dazu schreiben wir den Propagator ν als (unendliche) Summe über Fluktuationspropagatoren ν_k, deren Spektrum jeweils auf einer kompakten Impulsschale $a_{k+1} \leq |p| \leq a_k$ liegt. Es gelten die Randbedingungen $a_0 = \frac{2\pi}{a}$ und $\lim_{k\to\infty} a_k = 0$. Eine Integration über die regulären Kovarianzen ν_k ist problemlos. Das (nicht normierte) Pfadintegral zerfällt nach der Faltungsformel für Gaußsche Maße (B.14).

$$\int d\mu_{\sum_{k=0}^{N}\nu_k}(\phi)Z(\phi) = \int \prod_{k=0}^{N} d\mu_{\nu_k}(\zeta_k)Z(\phi+\sum_{k=0}^{N}\zeta_k) \tag{1.17}$$

An dieser Stelle erkennt man nun deutlich die Auswirkungen der Propagator *cutoff*'s (1.13). Die UV-Schranke liefert uns die Impulsobergrenze $\frac{2\pi}{a}$.[11] Im Falle einer IR-Divergenz, die aufgrund der Struktur (1.11) nur bei $p = 0$ auftreten kann, müssen wir uns mit der Untergrenze der Impulsscheiben a_N an den Pol herantasten und zur Berechnung des Funktionalintegrals über ν den IR-Limes $N \to 0$ ausführen. Ist $\tilde{\nu}$ in $p = 0$ regulär, benutzt man günstigerweise eine endliche (z.B. äquidistante) Multiskalen-Zerlegung des Propagators.

Die Berechnung des Pfadintegrals oder das Ausführen des IR-Limes entsprechen somit einer unendlichen Iteration von BST. Die korrespondierende Skalenzerlegung des Propagators wird durch den Blockparameter L in der Form $a_k = \frac{2\pi}{aL^k}$ organisiert. Dieses nichtperturbative Lösungsverfahren formulieren wir nun explizit.

1.2.1 Operatoren der RG

Wir beginnen mit der Definition des Blockmitteloperators B_L[12], welcher die (normierten) Blockspins berechnet und auf das gröbere Gitter $\Lambda(La)$ überträgt. Im folgenden sei $L \in \mathbb{N}_{\geq 2}$ vorausgesetzt.

[11] In der nichtdiskreten Theorie müßten wir den Propagator in eine Impulsfolge über $\mathbb{Z}$ zerlegen, und der UV-Limes entspräche der Untergrenze $-\infty$.

[12] Den Index notieren wir nur, wenn er benötigt wird.

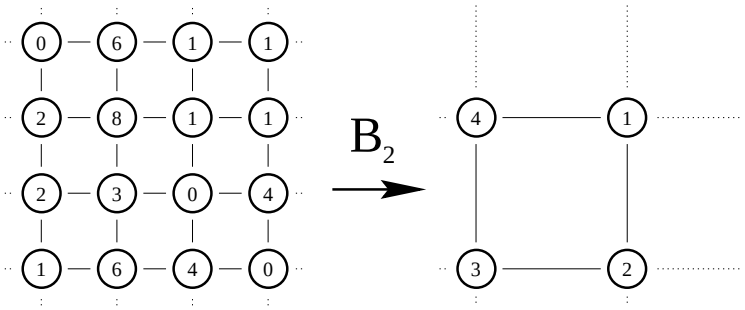


Abbildung 1.1: Die Wirkung des Blockmitteloperators B_L in 2 Dimensionen. Die Zahlen in den Kreisen sind die reellen Spins.

Definition 1.2.2 (Der Blockmitteloperator $B : \mathcal{H}(a) \to \mathcal{H}(La)$)

$$B(\phi)(x') = \frac{1}{\|\mathbb{B}(x')\|} \int_{\mathbb{B}(x')} d^D y \; \phi(y) \tag{1.18}$$

$$\mathbb{B}(x') = \left\{ y \in \Lambda(a) \mid La \left[\frac{y}{La} \right] = x' \right\} \tag{1.19}$$

$$\|\mathbb{B}(x')\| = (La)^D \tag{1.20}$$

Die Wirkung von B_L[13] verdeutlicht man sich am besten durch Abbildung 1.1. Obige Wahl der Normierung (Division durch das Blockvolumen liefert einen Faktor L^{-D}) scheint auf den ersten Blick willkürlich, doch wird erst auf diese Weise der adjungierte Operator $B_L^\dagger$ zu einem Re-Blockoperator.

Satz 1.2.3 (Der Blockoperator $B_L^\dagger : \mathcal{H}(La) \to \mathcal{H}(a)$)

$$B_L^\dagger(\phi')(x) = \phi' \left(La \left[\frac{x}{La} \right] \right) \tag{1.21}$$

Beweis:

$$(B_L^\dagger \phi', \phi) = \int_{\Lambda(a)} d^D x \; \phi' \left(La \left[\frac{x}{La} \right] \right) \phi(x)$$

[13] $\|\cdot\|$ beschreibt das Volumen einer Teilmenge des Gitters $\Lambda(a)$ (beliebige Vereinigung von Einheitsblöcken). Damit $\|\cdot\|$ zu einer Norm wird, muß man die Menge aller Teilmengen mit endlichem Volumen bzgl. der Äquivalenzrelation $A \sim B :\Leftrightarrow \|A\| = \|B\|$ auf den korrespondierenden Quotientenraum einschränken und eine Abbildung wie folgt definieren: $\forall A, B \exists C : \|C\| = \|A\| + \|B\|$. Inverse Elemente werden adjungiert (z.B. Farben + korrespondierende Verknüpfung). Eine skalare Multiplikation muß entsprechend erklärt werden. Der so konstruierte $\mathbb{R}$-Vektorraum ist isomorph zu $\mathbb{R}$.

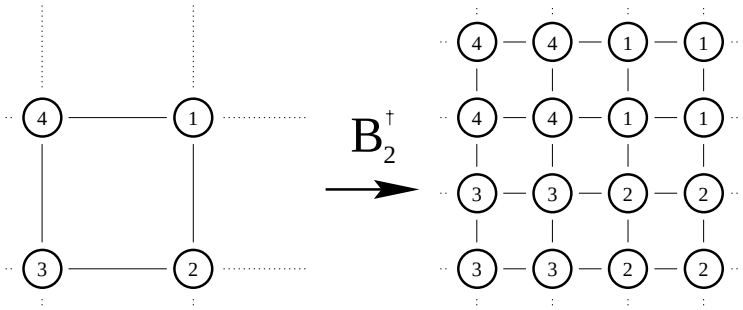


Abbildung 1.2: Die Wirkung des adjungierten Blockmitteloperators $B_L^\dagger$ in 2 Dimensionen

$$\begin{aligned} &= \frac{1}{\|\mathbb{B}(x')\|}\int_{\Lambda(La)} d^D x' \int_{\mathbb{B}(x')} d^D y\ \phi'\left(La\left[\frac{y}{La}\right]\right)\phi(y) \\ &= \int_{\Lambda(La)} d^D x' \phi'\,(x')\,\frac{1}{\|\mathbb{B}(x')\|}\int_{\mathbb{B}(x')} d^D y\ \phi(y) \\ &= (\phi', B\phi) \end{aligned}$$

□

Funktionen $B_L^\dagger(\phi')$ sind auf Blöcken $\mathbb{B}(x')$, $x' \in \Lambda(La)$ konstant. Dies verdeutlicht auch Abbildung 1.2. Als nächstes konstruieren wir einen Operator, mit dessen Hilfe man ein Gitter streckt bzw. staucht.

Definition 1.2.4 (Die Dilatationsoperatoren S und $S^\dagger$)
Es sei $\sigma \in \mathbb{R}$. Wir definieren und erhalten

$$S : \mathcal{H}(La) \to \mathcal{H}(a) \qquad S(\phi')(x) = L^\sigma \phi'(Lx) \tag{1.22}$$

$$S^\dagger : \mathcal{H}(a) \to \mathcal{H}(La) \qquad S^\dagger(\phi)(x') = L^{\sigma - D}\phi(\frac{x'}{L})\ . \tag{1.23}$$

Den Exponenten σ, den wir im folgenden skalierende Dimension nennen, lassen wir noch unbestimmt. Die Abbildungsvorschrift ist unabhängig von der Wahl des Definitionsbereiches und der zugehörigen Wertemenge, so daß S auch für $a = 0$ definiert ist und zu einer selbstabbildenden Kontinuumsfunktion wird.

Mit Hilfe der Operatoren B und S ist es nun möglich, einen auf $\mathcal{H}(a)$ selbstabbildenden Blockoperator abzuleiten, der das geblockte Feld bzgl. der Gitterkonstante wieder auf die ursprüngliche Größe reduziert bzw. ein Feld vergrößert und entstehende Blöcke mit ursprünglichen Eckwerten auffüllt. Wir definieren kanonisch:

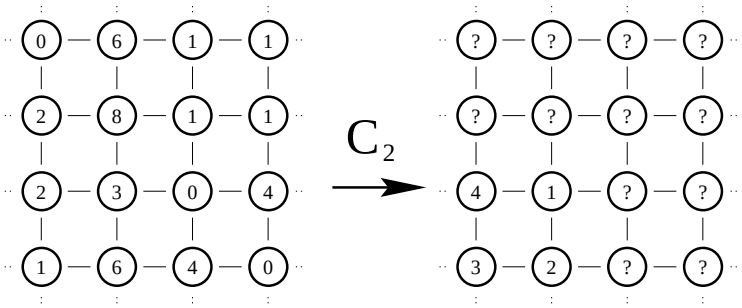


Abbildung 1.3: Die Wirkung des Blockmitteloperators C_L in 2 Dimensionen (ohne skalierenden Faktor)

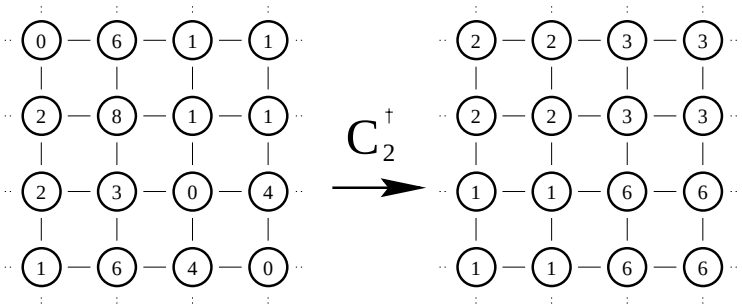


Abbildung 1.4: Die Wirkung des adjungierten Blockmitteloperators $C_L^\dagger$ in 2 Dimensionen (ohne skalierenden Faktor)

Definition 1.2.5 (Die Blockmitteloperatoren C und $C^\dagger$)

$$C : \mathcal{H}(a) \to \mathcal{H}(a) \quad C := S \circ B \quad \Rightarrow \tag{1.24}$$

$$C(\phi)(x) = \frac{L^\sigma}{\mathbb{B}(Lx)} \int_{\mathbb{B}(Lx)} d^D y \; \phi(y) \tag{1.25}$$

$$C^\dagger(\phi)(x) = L^{\sigma - D} \phi\left(a \left[\frac{x}{La}\right]\right) \tag{1.26}$$

Die Wirkungsweise der beiden Operatoren entnimmt man den Abbildungen 1.3 und 1.4. Man erkennt deutlich, daß $C^\dagger$ gerade die von uns definierte Aufgabe erfüllt, die in einem Feld gespeicherten Spins auf Blöcke zu verteilen. Wir wollen ein geblocktes Feld $C^\dagger\phi$ (manchmal auch nur ϕ) in Zukunft Hintergrund- oder Blockfeld nennen. Die Operatorkerne schreiben sich als

$$C(x,y) = L^{\sigma-D}\delta_{x,\left[\frac{y}{La}\right]a} \qquad C^\dagger(x,y) = C(y,x) = L^{\sigma-D} \sum_{z \in \mathbb{B}(Ly)} \delta_{x,z} \; . \tag{1.27}$$

Für die Komposition zweier Blockoperatoren gilt:

$$C_L C_{L'} = C_{LL'} \tag{1.28}$$

Für den Beweis setze man zur Vereinfachung $a = 1$ und $D = 1$. OBdA sei $x \in \mathbb{N}_0$. Es existieren die eindeutigen Darstellungen

$$\begin{array}{rcll} x & = & \sum_{k=0}^{\infty} (LL')^k \, x_k & x_k = 0, \ldots, LL' - 1 \\ x_0 & = & \sum_{k=0}^{\infty} (L')^k \, \tilde{x}_k & \tilde{x}_k = 0, \ldots, L' - 1 \, , \end{array} \tag{1.29}$$

aus denen man die Relationen

$$\left[\frac{x}{LL'}\right] = \sum_{k=0}^{\infty} (LL')^k \, x_{k+1} \tag{1.30}$$

$$\left[\frac{x}{L'}\right] = L \sum_{k=0}^{\infty} (LL')^k \, x_{k+1} + \underbrace{\sum_{k=0}^{\infty} (L')^k \, \tilde{x}_{k+1}}_{\leq \frac{x_0}{L'} < L} \tag{1.31}$$

ableitet. $\left[L^{-1} \left[(L')^{-1} \, x\right]\right]$ bestimmt sich mit Hilfe von (1.31) zu (1.30). Eine weitere wichtige Eigenschaft ist die Nichtinvertierbarkeit von C (und folglich auch $C^\dagger$) auf $\Lambda(a)$, da ganze Klassen von Feldern mit gleichen Blockmittelwerten existieren. Eine interessante Eigenschaft ist die Projektoreigenschaft von $L^{2(D-\sigma)} C C^\dagger$. Mit Hilfe der bildlichen Vorstellung ist diese Eigenschaft klar, sie rechnet sich jedoch auch leicht mittels (1.27) nach [Rol96]. Unter Benutzung der Blockmitteloperatoren C und $C^\dagger$ definieren wir die wesentlichen Objekte der RG auf dem Gitter.

Definition 1.2.6 (RGT-Operatoren)

$$\begin{array}{rcll} u & := & C \nu C^\dagger & \textit{geblockte Kovarianz} \\ A & := & \nu C^\dagger u^{-1} & \textit{A-Kern} \\ \Gamma & := & \nu - A u A^\dagger & \textit{Fluktuationskovarianz} \end{array} \tag{1.32}$$

u modulo Nullmoden (!) ist eine Kovarianz und invertierbar. Weitere Auskünfte erhalten wir über die Kerndarstellungen im Orts- und Impulsraum:

Satz 1.2.7

$$u(x,y) = L^{2\sigma} \int_{\mathbb{B}(Lx)} \frac{d^D z}{\|\mathbb{B}(Lx)\|} \int_{\mathbb{B}(Ly)} \frac{d^D w}{\|\mathbb{B}(Ly)\|} \nu(z,w) \tag{1.33}$$

$$\tilde{u}(p) = \sum_{Q \in p + \Lambda(\frac{2\pi}{a})_L} \left(\prod_{\mu=1}^{D} \frac{\sin^2\left(\frac{Q_\mu a}{2}\right)}{L^2 \sin^2\left(\frac{Q_\mu a}{2L}\right)} \right) L^{2\sigma - D} \tilde{\nu}\left(\frac{Q}{L}\right) \tag{1.34}$$

Es ist $\Lambda(\frac{2\pi}{a})_L := \Lambda(\frac{2\pi}{a})/\Lambda(\frac{2\pi L}{a}) = \left\{Q \in \Lambda\left(\frac{2\pi}{a}\right) \mid 0 \leq Q_\mu < \frac{2\pi L}{a}\right\}$. An der Produktdarstellung des Sinus [FL94]

$$\prod_{\mu=1}^{D} \frac{\sin^2\left(\frac{Q_\mu a}{2}\right)}{L^2 \sin^2\left(\frac{Q_\mu a}{2L}\right)} = \prod_{\mu=1}^{D} \prod_{n \in \mathbb{N}-L\mathbb{N}} \left\{1 - \left(\frac{Q_\mu a}{2\pi n}\right)^2\right\}^2 \tag{1.35}$$

erkennt man, daß sich Pole von ν auf u vererben, jedoch keine neuen Singularitäten entstehen. Es gilt die Abschätzung:

$$|\tilde{u}(p)| \leq \max_{P \in \Lambda(\frac{2\pi}{a})/\Lambda(\frac{2\pi L}{a})} L^{2\sigma} \left|\tilde{\nu}\left(\frac{p+P}{L}\right)\right| \tag{1.36}$$

A ist eine ausgeschmierte Version des Operators $C^\dagger$ und Rechts-Inverses zu C. Somit ist AC ein Projektor[14] auf dem Hilbertraum $\mathcal{H}(a)$ bezüglich des Skalarproduktes $< \phi, \psi > := (\phi, \nu^{-1}\psi)$ (da ν positiv ist, ist dies wirklich ein Skalarprodukt).

Die Fluktuationskovarianz Γ erfüllt die Eigenschaften einer Kovarianz. Mittels der Darstellung $\Gamma = (1 - AC)\nu(1 - AC)^\dagger$ folgt, daß Γ semidefinit ist. Schließen wir den Kern von $(1 - AC)^\dagger$ aus dem Pfadintegral aus, so ist Γ Kovarianz.

Ihren Namen verdankt die Fluktuationskovarianz der Eigenschaft

$$\Gamma C^\dagger = 0 \ , \tag{1.37}$$

d.h. sie verschwindet auf Hintergrundfeldern. Γ wirkt nur auf einen Fluktuationsanteil.

Leider zeigen wir in dieser Arbeit nicht, daß $\|\tilde{\Gamma}\|_1 < \infty$ - der Fluktuationspropagator also einen IR-*cutoff* besitzt. In der Kontinuumstheorie [Wie97d] definiert man Γ über die Impulsscheibenmethode, indem man den freien, masselosen Propagator $\frac{1}{p^2}$ mit einer L-abhängigen, exponentiellen IR- und UV-Abschneidefunktion versieht, die im wesentlichen Impulse mit $L^{-1} < |p| < 1$ herausfiltert: $\frac{1}{p^2}(e^{-p^2} - e^{-L^2p^2})$.

Schön wäre auch ein Beweis, der zeigt, daß die Fourier-Transformierte von Γ primär den durch $\frac{2\pi}{aL} \leq |p| \leq \frac{2\pi}{a}$ definierten, kompakten Träger besitzt.

[14] $(AC)^\dagger = \nu^{-1}AC\nu$

1.3 Herleitung der RGT

Mit Hilfe der Faltungsformel für Gaußsche Maße (B.14) und der Zerlegung der Kovarianz ν in einen geblockten Teil $AuA^\dagger$ und einen Fluktuationsanteil Γ läßt sich die geplante Aufteilung des Pfadintegrals problemlos durchführen:

$$\begin{aligned} \int d\mu_\nu(\phi)\; Z(\psi+\phi) &= \int d\mu_{AuA^\dagger}(\xi) \int d\mu_\Gamma(\zeta)\; Z(\psi+\xi+\zeta) \\ &\overset{\xi=A\phi}{=} \int d\mu_u(\phi) \int d\mu_\Gamma(\zeta)\; Z(\psi+A\phi+\zeta) \end{aligned} \tag{1.38}$$

Nun zerlegen wir Zähler und Nenner des erzeugenden Funktionals H und erhalten

$$H(A\phi) = \frac{\int d\mu_u(\zeta) R(Z)(\phi+\zeta)}{\int d\mu_u(\zeta) R(Z)(\zeta)} \tag{1.39}$$

$$R(Z)(\phi) = \frac{\int d\mu_\Gamma(\zeta) Z(A\phi+\zeta)}{\int d\mu_\Gamma(\zeta) Z(\zeta)}\,. \tag{1.40}$$

Unsere Teilintegration hat dazu geführt, daß wir in H nun eine effektive Theorie behandeln, die durch den Propagator u und die Wechselwirkung $R(Z)$ beschrieben wird. Die Transformation des Wechselwirkungsanteils nennen wir Gitter-RGT (GRGT). Die Wechselwirkungen, gemessen in der Korrelationslänge, sind in $R(Z)$ um den Faktor $\frac{1}{L}$ kurzreichweitiger.

Im folgenden präsentieren wir die beiden Theorieklassen, die wir in dieser Arbeit behandeln.

1.3.1 Die perfekte masselose Gitterkovarianz ν_{perf}

Wir konstruieren den Propagator ν_{perf}, der Fixpunkt der Abbildung $\nu \to u(\nu)$ ist. Auf diese Weise müssen wir nur noch die RGT des Potentials betrachten.

Mittels der Folge $\nu_0 := \nu$, $\nu_n := C\nu_{n-1}C^\dagger$ für $n \in \mathbb{N}$ und der Eigenschaft (1.28) formulieren wir eine unendliche Blockung der Kovarianz wie folgt:

$$\lim_{n\to\infty} \nu_n = \lim_{n\to\infty} C_{L^n}\nu C_{L^n}^\dagger = \lim_{L\to\infty} C_L \nu C_L^\dagger \tag{1.41}$$

Existiert dieser Limes, ν_{perf} genannt, ist er per Konstruktion ein Fixpunkt der Propagatortransformation. Unter der Voraussetzung

$$\sigma = \frac{D}{2} - 1 \tag{1.42}$$

fließen die IR-divergenten masselosen Kovarianzen (1.11) mit $c_0 = 0$ und $c_2 = 1$, für $L \to \infty$ in den freien, über Gitterkuben gemittelten Kontinuumspropagator

$$\nu_{perf}(x,y) = \int_{\mathbb{B}_E} \frac{d^D\overline{x}}{a^D} \int_{\mathbb{B}_E} \frac{d^D\overline{y}}{a^D} (-\triangle)^{-1}(x+\overline{x}, y+\overline{y}) \; . \tag{1.43}$$

Hierbei ist $\mathbb{B}_E = [0,a]^D$. ν_{perf} ist die perfekte, masselose Kovarianz.[15] Obwohl sie eine masselose Theorie beschreibt, können wir Massenkorrekturen problemlos in das Potential integrieren. Sie werden von der RGT (sofern $Z \neq 1$) sowieso generiert. In allen folgenden Gitter-Rechnungen wollen wir nur noch ν_{perf} benutzen und uns auf die Transformation des Boltzmann-Faktors beschränken.

1.3.2 Der hierarchische Propagator ν_{hier}

Ein praktisches Problem der RG ist das Auftreten nichtlokaler Terme in der effektiven Wechselwirkung $R(Z)$, selbst wenn Z lokal ist. Um dies zu vermeiden, führen wir den hierarchischen Propagator ν_{hier} ein, dessen Modelle unter RGT Lokalität bewahren. Der große Nachteil ist die fehlende Gittertranslationsinvarianz. Weitere Informationen finden sich z.B. in [PPW94, Por93].

Für $D > 2$ definieren wir

Definition 1.3.1 (Die hierarchische Kovarianz)

$$\nu_{hier}(x,y) = \gamma \sum_{n=0}^{\infty} L^{(2-D)n} \delta_{\left[\frac{x}{L^n a}\right],\left[\frac{y}{L^n a}\right]} \tag{1.44}$$

Natürlich erhalten wir nur für $\gamma \in \mathbb{R}^+$ eine positive Form. Die Delta-Funktion ist identisch eins, wenn x und y nach der n-ten Blockung im selben Hyperwürfel liegen. Der so definierte Kern ist für alle $x, y \in \Lambda(a)$ endlich, denn es gilt

$$\nu_{hier}(x,y) = \frac{\gamma}{a^D} \frac{L^{(2-d)N(x,y)}}{1 - L^{(2-d)}} \tag{1.45}$$

mit

$$N(x,y) = \min\left\{ n \in \mathbb{N}_0 \mid \left[\frac{x}{L^n a}\right] = \left[\frac{y}{L^n a}\right] \right\} \; . \tag{1.46}$$

[15] Sie ist in der Beziehung perfekt, das sie die optimale Diskretisierung des masselosen, inversen Klein-Gordon Operators $(-\triangle)^{-1}$ auf dem Gitter darstellt. Die physikalischen Vorhersagen sind unabhängig vom *cutoff*, also der Gitterkonstanten. Das Spektrum ist folglich exakt.

Der Beweis besteht aus einer einfachen Anwendung der geometrischen Reihe. Die Formulierung (1.45) findet sich auch bei [GK84]. Um die nicht vorhandene Gittertranslationsinvarianz zu zeigen, verschiebt man ein Gittertupel (x, y), das in einem L-Kubus liegt, so, daß x und y in disjunkten L-Würfeln leben. Für $|x-y| \gg 1$ gilt zwar mit großer Wahrscheinlichkeit $|x-y| \sim L^{N(x,y)}a$, so daß $\nu_{hier} \sim |x-y|^{2-d}$ und neben Translationsinvarianz ein ähnliches IR-Verhalten wie bei $-\triangle^{-1}$ vorliegt, aber die Nächste-Nachbar-Wechselwirkungen zerstören dieses Bild.

Man erkennt ferner, daß die hierarchische Kovarianz 1.3.1 über eine Multiskalen-Zerlegung definiert ist und dasselbe kritische Verhalten (UV-*cutoff*, IR-divergent) wie ein allgemeiner, masseloser ($c_0 = 0$) Propagator (1.11) aufweist.

Nun betrachten wir den geblockten Operator $C^\dagger \nu_{hier} C$. Mittels (1.27) berechnen wir

$$\begin{aligned} C^\dagger \nu_{hier} C(x,y) &= \int d^D z \int d^D w C(z,x) \nu_{hier}(z,w) C(w,y) \\ &= L^{2(\sigma-D)} \nu_{hier}(\left[\frac{x}{La}\right] a, \left[\frac{y}{La}\right] a) \\ &= \gamma L^{2(\sigma-D)} \sum_{n=0}^{\infty} L^{(2-d)n} \delta_{\left[\frac{x}{L^{n+1}a}\right],\left[\frac{x}{L^{n+1}a}\right]} \\ &= L^{2\sigma-D-2} \left(\nu_{hier}(x,y) - \gamma \delta_{x,y}\right) \end{aligned}$$

Bestimmen wir die skalierende Dimension zu

$$\sigma = 1 + \frac{D}{2} , \tag{1.47}$$

so erhalten wir folgende Zerlegung unseres hierarchischen Propagators:

$$\nu_{hier} = \gamma \mathrm{id} + C^\dagger \nu_{hier} C \tag{1.48}$$

Wir zerlegen das erzeugende Funktional H (1.14) mittels (B.14)[16] und erhalten als Pendant zu (1.38)

$$\int d\mu_{\nu_{hier}}(\phi) Z(\psi+\phi) = \int d\mu_{\nu_{hier}}(\phi) \int d\mu_{\gamma \mathrm{id}}(\zeta) Z(\psi + C^\dagger \phi + \zeta) . \tag{1.49}$$

Der hierarchische Propagator ist somit ein Fixpunkt des Kovarianzflusses, und wir müssen unser Augenmerk nur noch auf die effektive Wechselwirkung

[16] $C^\dagger \nu C$ ist (modulo Nullmoden) positiv.

legen. Wir benutzen die Ultralokalität des Fluktuationspropagators $\gamma\mathrm{id}$, indem wir lokale Wechselwirkugen betrachten. Es sei

$$Z(\phi) = \prod_{x\in\Lambda(a)} z(\phi(x)) \; . \tag{1.50}$$

Es folgt für die unnormierte RGT:

$$\begin{aligned} R(Z)(\phi) &:= \int d\mu_{\gamma\mathrm{id}}(\zeta) Z(C^\dagger\phi+\zeta) \\ &= \prod_{x\in\Lambda(a)} \prod_{z\in\mathbb{B}(Lx)} \mathcal{N} \int d\zeta(z) e^{-\frac{1}{2\gamma}\zeta(z)^2} z\left(L^{1-\frac{D}{2}}\phi(x)+\zeta(z)\right) \\ &= \prod_{x\in\Lambda(a)} \mathcal{N}' \left\{ \int d\mu_\gamma(\zeta) z\left(L^{1-\frac{D}{2}}\phi(x)+\zeta\right)\right\}^{L^d} \\ &= \prod_{x\in\Lambda(a)} \mathcal{N}'\mathcal{R}(z)(\phi(x)) \end{aligned} \tag{1.51}$$

Hierbei bezeichnet man die Abbildung $\mathcal{R}$ als (unnormierte) hierarchische RGT (HRGT). Die Normierungsfaktoren $\mathcal{N}$ und $\mathcal{N}'$ können aufgrund der unendlichen Dimensionalität des Integrals nicht explizit angegeben werden (da $\mathcal{H}(a)$ auf den l_1 eingeschränkt wurde) und werden durch die Normierungsbedingung an das Gaußsche Integral definiert. Für die normierte RGT folgt

$$R(Z)(\phi) = \prod_{x\in\Lambda(a)} \underbrace{\frac{\mathcal{R}(z)(\phi(x))}{\mathcal{R}(z)(0)}}_{\text{normierte HRGT}} \; . \tag{1.52}$$

1.4 Das Werkzeug RG

Wir haben in diesem Kapitel zwei Klassen von Theorien betrachtet, die durch die Propagatoren ν_{perf} und ν_{hier} charakterisiert werden. Beiden ist die Eigenschaft gemein, daß die Propagatoren Fixpunkte der Blockung sind, so daß nur noch die RGT des Potentials Z betrachtet werden muß.

Wie schon erwähnt ist die RG gut dazu geeignet, kritische Theorien zu betrachten. Die RGT ist eine Skalentransformation: Eine Theorie der Korrelationslänge ξ wird auf eine Theorie der Korrelationslänge $\frac{\xi}{L}$ abgebildet [GK84]. Ist die Theorie unkritisch, d.h. $\xi < \infty$, führt die Berechnung der Zustandssumme (= unendliche Iteration der RGT) zu einer Theorie der Korrelationslänge Null. Dies entspricht einer völlig unkorrelierten Phase - einer sog.

Hochtemperaturphase - von der man aufgrund der Ultralokalität eine Faktorisierung der Wirkung bzgl. des Gitters $\Lambda(a)$ erwartet.

Da man eine Gittertheorie jedoch so konstruiert, daß im Kontinuumslimes $a \to 0$ die Greensfunktionen endlich sind, behandelt man im allgemeinen kritische Gittertheorien mit divergierender Korrelationslänge, die unter RGT-Anwendung kritisch bleiben. Ein Fixpunkt der RGT ist somit kritisch ($\xi = \infty$) oder ultralokal ($\xi = 0$). Alle Theorien im Einzugsbereich eines Fixpunktes besitzen dieselben kritischen Eigenschaften wie der Fixpunkt selbst, d.h. divergierende oder endliche (bei einem ultralokalen Fixpunktpotential) Korrelationslängen. Die im Attraktionsbereich liegenden Theorien sind fast skaleninvariant.

Abschließend stellen wir noch einen fundamentalen Unterschied zwischen GRG und HRG heraus:

Satz 1.4.1 (Die Halbgruppeneigenschaft)
Die Menge der GRGT zu fester Dimension $\{R_L : L \in \mathbb{N}\}$ bildet bezüglich der Komposition eine abelsche Halbgruppe. Die Menge der HRGT besitzt diese Struktur nicht!

Beweis: Mit Definition 1.2.6 und der Fixpunkteigenschaft $u(\nu_{perf}) = \nu_{perf}$ erhält man die Relationen $A_L A_{L'} = A_{LL'}$ und $\Gamma_L + A_L \Gamma_{L'} A_L^\dagger = \Gamma_{LL'}$. Einsetzen in (1.40) liefert mit Substitution und Gaußscher Faltung (B.14) das gewünschte Resultat

$$R_L \circ R_{L'} = R_{LL'} \ . \tag{1.53}$$

Man beachte, daß das neutrale Element $R_1 := \mathrm{id}$ ergänzt werden muß, da die Fluktuationskovarianz für $L = 1$ verschwindet, und das Gaußsche Maß folglich nicht mehr definiert ist. Die Verletzung der Halbgruppeneigenschaft in der HRG mache man sich selbst klar.

Die Halbgruppeneigenschaft der GRGT ermöglicht eine alternative Berechnung des IR-Limes: statt einer unendlichen Iteration von GRGT schickt man den Skalenparameter gegen unendlich. Es gilt also für $L \in \mathbb{N}_2$:

$$\lim_{n\to\infty} R_L^n = \lim_{L\to\infty} R_L \tag{1.54}$$

Die Limes-Bildung über L ist praktischer, da die Berechnung einer RGT sehr komplex ist. Vor allem bei numerischen Untersuchungen bedeutet diese Art der Berechnung eine effiziente und effektive Ausnutzung von Rechnerkapazitäten.

Für den Beweis der Existenz von RG-Flüssen oder der Konstruierbarkeit von RG-Trajektorien stützt man sich jedoch auf einzelne RG-Schritte. Im

hierarchischen Modell ist dies sogar die einzige Möglichkeit. Dennoch spielt dort, wie wir noch sehen werden, eine geeignete Wahl von L eine große Rolle.

Kapitel 2

Rigorose Konstruktion der ϕ^4_D-Trajektorie im Hierarchischen Modell für $2 < D < 4$

Wir beginnen dieses Kapitel mit einer Zusammenstellung der wichtigsten Eigenschaften der HRGT. Vertiefende Informationen finden sich z.B. in [Rol96, GS96, Por93]. Nach der Definition der ϕ^4-Trajektorie gemäß C. Wieczerkowski berechnen wir diese perturbativ. Anschließend präsentieren wir eine formalisierte Version des ebenfalls von C. Wieczerkowski entwickelten nichtstörungstheoretischen Konstruktionsbeweis [Wie97a] und zeigen, daß wir mit Hilfe perturbativer Approximanten die ϕ^4-Kurve in jeder Dimension $2 < D < 4$ berechnen können. Schon in der Störungstheorie erkennt man, daß in bestimmten Dimensionen Resonanzen auftreten, die z.B. in $D = 3$ Dimensionen durch eine Doppelreihen-Entwicklung in Kopplung und logarithmierter Kopplung gelöst werden [RW]. Wir zeigen, daß das konstruktive Verfahren auch mit dieser perturbativen Näherung funktioniert. Abschließend stellen wir einige numerische Ergebnisse vor und diskutieren sie.

2.1 Grundlagen

Die HRGT ist eine nichtlineare Integraltransformation von reellwertigen, auf $\mathbb{R}$ definierten Funktionen (1.52). Die HRGT wird durch die Blockgröße L und die Dimension D bestimmt. Die Gitterkonstante a findet sich (implizit) im

Wechselwirkungsterm. OBdA sei $a = 1$. Im Rahmen des hierarchischen Modells wollen wir die Parameter L und D als kontinuierlich ansehen. Auf diese Weise können wir allgemeine Verhaltensmuster der HRGT besser studieren. Hier das zentrale Objekt dieses Kapitels:

Definition 2.1.1 (HRGT)
Es sei $L \in (1, \infty)$, $D \in (2, 4]$ *und* $\mathcal{V} \subseteq \mathcal{C}^0(\mathbb{R})$. *Dann erklären wir die HRGT* $\mathcal{R} \equiv \mathcal{R}_{L,D}$ *durch*

$$\mathcal{R} : \mathcal{V} \to \mathcal{V} \qquad \mathcal{R}(Z)(\phi) = \int d\mu_\gamma(\zeta) Z(\beta\phi + \zeta)^\alpha \; . \tag{2.1}$$

Hierbei sind $\alpha = L^D, \beta = L^{1-\frac{D}{2}}$, $\gamma = 1 - L^{2-D}$ *und* $\mathcal{V}$ *ein geeigneter Funktionenraum,*[1] *so daß* $\mathcal{R}$ *wohldefiniert ist.*

In obiger Integraltransformation ist $d\mu_\gamma(\zeta) = \frac{1}{\sqrt{2\pi\gamma}} e^{-\frac{\zeta^2}{2\gamma}} d\zeta$ das eindimensionale Gaußsche Maß über $\mathbb{R}$ mit Mittel Null und Kovarianz $\gamma > 0$, deren willkürliche Wahl zu $1-\beta^2$ im Kapitel über die Normalordnung erklärt wird. Im allgemeinen sind die Transformationen zu zwei Kovarianzen γ,γ' über die Beziehung

$$\mathcal{R}_\gamma(Z)(\phi) = \mathcal{R}_{\gamma'}\left(Z\left(\sqrt{\frac{\gamma}{\gamma'}}\,\cdot\right)\right)\left(\frac{\gamma'}{\gamma}\phi\right) \tag{2.2}$$

verknüpft.

OBdA seien L und D so gewählt, daß $\alpha \in \mathbb{N}$. Es folgt die Wohldefiniertheit von Z^α auch für nicht positive Funktionen.

Obige Form der RGT differiert von der Darstellung in (1.51). Mittels der Ähnlichkeitstransformation $\mathcal{U}(Z) = Z^\alpha$ erhalten wir die Beziehung[2]

$$\mathcal{R} = \mathcal{U}^{-1}\overline{\mathcal{R}}\mathcal{U}, \tag{2.3}$$

sofern wir uns auf nicht negative Z einschränken. Ist Z Fixpunkt von $\mathcal{R}$, so ist $\mathcal{U}(Z)$ Fixpunkt von $\overline{\mathcal{R}}$. Somit gestalten sich Untersuchungen an $\mathcal{R}$ und $\overline{\mathcal{R}}$ äquivalent.

Im weiteren Verlauf benutzen wir die nicht normierte RG. Der Vorteil der normierten Form $\mathcal{R}_0(Z)(\phi) := \frac{\mathcal{R}(Z)(\phi)}{\mathcal{R}(Z)(0)}$ liegt darin, daß Funktionen, die bezüglich der Äquivalenzrelation $Z \sim Z' :\Leftrightarrow \frac{Z}{Z'} \in \mathbb{R}^\star$ in derselben Nebenklasse liegen,

[1] $\mathcal{V}$ sei für den Beginn nicht den Beschränkungen des ersten Kapitels, z.B. Positivität, unterworfen.

[2] $\overline{\mathcal{R}}$ sei die HRGT mit externem Exponenten α gemäß (1.51)

auf dieselbe effektive Wechselwirkung abgebildet werden, so daß man sich auf die Repräsentantenmenge $Z(0) = 1$ beschränken kann. $Z = 0$ wird hierbei explizit auf Null abgebildet oder ausgeschlossen. Der Nachteil ist jedoch die kompliziertere Form der Transformation.

Drücken wir die Wechselwirkung Z durch ein Potential V aus, so schreibt sich die Transformation als

$$\mathcal{T} : \mathcal{W} \to \mathcal{W} \qquad V \mapsto -\ln \mathcal{R}\left(e^{-V}\right) \tag{2.4}$$

Die Wohldefiniertheit folgt daraus, daß $\mathcal{R}$ die Positivität einer Wirkung erhält. Entsprechendes gilt für die HRGT mit externem α. $\mathcal{W}$ muß natürlich gewisse Restriktionen erfüllen, damit $\mathcal{T}$ definiert und selbstabbildend ist, doch dazu später mehr.

Es stellt sich die Frage, für welche Untermengen von $\mathcal{C}^0(\mathbb{R})$ die HRGT nun wohldefiniert ist? Einen wichtigen Spezialfall liefert folgender

Satz 2.1.2
Für

$$\mathcal{V} \equiv \mathcal{V}_{Gauß} = \left\{ Z : \mathbb{R} \to \mathbb{R}^+ \mid Z(\phi) = Ae^{-\frac{b}{2}\phi^2} \;\; mit \;\; A \in \mathbb{R}^+, \;\; b \in \mathbb{R}_0^+ \right\}$$

ist die HRGT 2.1.1 wohldefiniert[3] *und es gilt:*

$$\forall Z \in \mathcal{V} : \quad \mathcal{R}(Z)(\phi) = A' e^{-\frac{b'}{2}\phi^2} \quad mit \quad A' = \frac{1}{\sqrt{1+\alpha\gamma b}} A^\alpha, \;\; b' = \frac{\alpha\beta^2 b}{1+\alpha\gamma b}$$

Beweis: Für $Z \in \mathcal{V}$ gilt:

$$\mathcal{R}(Z)(\phi) = \ldots = \frac{A^\alpha}{\sqrt{1+\alpha\gamma b}} e^{-\frac{1}{2}\frac{\alpha\beta^2 b}{1+\alpha\gamma b}\phi^2}$$

Da $\alpha, \beta, \gamma, A \in \mathbb{R}^+$ bzw. $b \in \mathbb{R}_0^+$, folgt $A' \in \mathbb{R}^+$ bzw. $b' \in \mathbb{R}_0^+$.

□

Eine massebehaftete Theorie ist unter der HRGT forminvariant. Interessant ist nun die Frage nach Fixpunkten, das heißt: Existieren $Z \in \mathcal{V}$, so daß $\mathcal{R}(Z) \equiv Z$? Eine Antwort liefert folgender

[3] Die HRGT ist natürlich auch für $b \in (-\frac{1}{\alpha\gamma}, \infty)$ definiert, wegen $b'\left((-\frac{1}{\alpha\gamma}, \infty)\right) = \left(-\infty, \frac{\beta^2}{\gamma}\right)$ allerdings nicht mehr selbstabbildend.

Satz 2.1.3 (Gaußsche Fixpunkte der HRGT)
Die HRGT über $\mathcal{V}_{Gauß}$ besitzt genau zwei Fixpunkte:

$$\begin{aligned} A_{UV} &= 1, & b_{UV} &= 0 & \Rightarrow \quad & Z_{UV}(\phi) = 1 \\ A_{QU} &= (\alpha\beta^2)^{\frac{1}{2(\alpha-1)}}, & b_{QU} &= \tfrac{\alpha\beta^2-1}{\alpha\gamma} & \Rightarrow \quad & Z_{QU}(\phi) = A_{QU} e^{-\frac{b_{QU}}{2}\phi^2} \end{aligned}$$

Hierbei bezeichnet der Index UV den ultravioletten bzw. trivialen, und das Kürzel QU den quadratischen bzw. Hochtemperaturfixpunkt.

Der Hochtemperaturfixpunkt entspricht einer völlig unkorrelierten Theorie. Die zugehörige Korrelationslänge ξ ist Null und die hierarchische Wechselwirkung folglich ultralokal [GS96][4]. Der triviale Fixpunkt entspricht einer freien Theorie, welche, wie im vollen Modell mit masselosem Propagator, einer kritischen Theorie oder einem PÜ zweiter Ordnung entspricht. Die zugehörige Korrelationslänge divergiert.

Neben diesen beiden Gaußschen Fixpunkten existieren noch weitere Fixpunkte: Zum einem die unphysikalische Theorie $Z = 0$. Zum anderen existieren noch die nichttrivialen Fixpunkte. In Abhängigkeit der Bifurkationsdimension $d_n = \frac{2n}{n-1}$ spalten sich vom trivialen Fixpunkt weitere Fixpunkte in Richtung des marginalen Eigenvektors in der Dimension d_n ab, so daß für $D \to 2$ unendlich viele nichttriviale Fixpunkte existieren. Die Untersuchung dieses interessanten Bifurkationsszenarios soll jedoch nicht Bestandteil der Arbeit sein.

Desweiteren bleiben symmetrische Funktionen unter Anwendung von $\mathcal{R}$ invariant, wie man leicht durch Einsetzen und die Substitution $\zeta \to -\zeta$ zeigt. Wir beschränken uns im weiteren auf symmetrische Boltzmann-Faktoren Z, da auch die Fixpunkte Z_{UV} und Z_{QU} symmetrisch sind. Ein weiterer Grund für diese Restriktion ist die Forderung, daß wir uns bei unseren Berechnungen in der sog. symmetrischen Phase befinden, in der Korrelationsfunktionen mit ungerader Argumentenanzahl verschwinden und der Vakuumzustand $|0\rangle$ eindeutig ist [MM94].

2.1.1 Die Banachräume $\mathcal{V}_{UV}$ und $\mathcal{V}_{QU}$

Nun wollen wir den Definitionsbereich $\mathcal{V}$ erklären. Um eine Iteration zu ermöglichen, muß $\mathcal{V}$ so gewählt sein, daß $\mathcal{R}$ selbstabbildend ist. Eine erste

[4] Die korrespondierenden Gitter-Wechselwirkungen sind natürlich immer lokal.

Wahl ist die Menge

$$\mathcal{V}_{UV} = \left\{ Z : \mathbb{R} \to \mathbb{R} \mid Z \in \mathcal{C}^0(\mathbb{R}), Z \in \mathbb{Z}_2(\mathbb{R}), \sup_{\phi \in \mathbb{R}} |Z(\phi)| < \infty \right\}, \quad (2.5)$$

die mit der Supremumsnorm zu einem Banachraum[5] ergänzt wird. Symmetrie (s.o) und Beschränktheit bleiben unter $\mathcal{R}$ erhalten, da

$$\sup_{\phi \in \mathbb{R}} |\mathcal{R}(Z)(\phi)| \leq \|Z\|_\infty^\alpha < \infty \, . \quad (2.6)$$

Die Stetigkeit von $\mathcal{R}(Z)$ folgt aus den Sätzen über Parameter-Integrale [For91]. Diese sind zwar nur für kompakte Intervalle formuliert, lassen sich aber problemlos auf $\mathbb{R}$ verallgemeinern, da das Kernstück der Beweise die gleichmäßige Stetigkeit der Integranden auf dem kartesischen Produkt von Integrationsgebiet und Gültigkeitsbereich der externen Variablen ist.

Beweis: Wählen wir $R \in \mathbb{R}^+$ beliebig, so ist

$$i(\zeta, \phi) := e^{-\frac{\zeta^2}{2\gamma}} Z^\alpha(\beta\phi + \zeta) \quad (2.7)$$

gleichmäßig stetig auf $\mathbb{R} \times [-R, R]$. Denn für alle $\epsilon > 0$ existieren positive G und δ mit $G > \delta$, so daß $|i(\zeta, \phi)| < \frac{\epsilon}{2}$ für (ζ, ϕ) mit $\zeta > G - \delta$, da Z beschränkt ist. Auf dem Kompaktum $[-G, G] \times [-R, R]$ ist die stetige Funktion i gleichmäßig stetig und

$$\forall (\zeta, \phi) \in \mathbb{R} \backslash [-G, G] \times [-R, R] \; \forall (\zeta', \phi') \in U_\delta(\zeta, \phi) \; : \; |i(\zeta', \phi') - i(\zeta, \phi)| < \epsilon$$

Folglich ist i auf ganz $\mathbb{R} \times [-R, R]$ gleichmäßig stetig und $\mathcal{R}(Z)$ auf $[-R, R]$ stetig. Da R beliebig ist, folgt $\mathcal{R}(Z) \in \mathcal{C}^0(\mathbb{R})$.

□

Auch die Stetigkeit der Transformation $\mathcal{R}$ zeigt man leicht:

$$\|\mathcal{R}(Z + \epsilon) - \mathcal{R}(Z)\|_\infty \leq \sum_{n=1}^{\alpha} \binom{\alpha}{n} \|Z\|_\infty^{\alpha - n} \|\epsilon\|_\infty^n \overset{\|\epsilon\|_\infty \to 0}{\longrightarrow} 0 \quad (2.8)$$

[5] Der Vektorraum der auf topologischen Räumen stetigen und beschränkten Abbildungen ist bezüglich der Supremumsnorm ein Banachraum [MV92]. Somit ist der abgeschlossene Unterraum $\mathcal{V}_{UV}$ auf natürliche Weise auch ein Banachraum. Beweis der Abgeschlossenheit: Sei $Z_n \in \mathcal{V}_{UV}$ eine Cauchy-Folge mit Grenzwert Z, so gilt $Z(\phi) = \lim Z_n(\phi) = \lim Z_n(-\phi) = Z(-\phi)$.

Einen weiteren invarianten Banachraum erhalten wir, wenn wir

$$\mathcal{V}_{QU} = \left\{ Z : \mathbb{R} \to \mathbb{R} \mid Z \in \mathcal{C}^0(\mathbb{R}), Z \in \mathcal{Z}_2(\mathbb{R}), \sup_{\phi \in \mathbb{R}} \left| \frac{Z(\phi)}{Z_{QU}(\phi)} \right| < \infty \right\} \quad (2.9)$$

durch die Norm

$$\| \cdot \|_{QU} : \mathcal{V}_{QU} \to \mathbb{R}_0^+ \qquad \|Z\|_{QU} = \sup_{\phi \in \mathbb{R}} \left| \frac{Z(\phi)}{Z_{QU}(\phi)} \right| \quad (2.10)$$

vervollständigen[6]. Auch dieser Raum ist invariant unter $\mathcal{R}$, denn

$$\sup_{\phi \in \mathbb{R}} \left| \frac{\mathcal{R}(Z)(\phi)}{Z_{QU}(\phi)} \right| \leq \sup_{\phi \in \mathbb{R}} \frac{1}{Z_{QU}(\phi)} \|Z\|_\infty^\alpha \mathcal{R}(Z_{QU})(\phi) = \|Z\|_\infty^\alpha \ . \quad (2.11)$$

Betrachtet man die Konstruktion von $\mathcal{V}_{QU}$, so erkennt man, daß $\mathcal{V}_{UV}$ im Grunde mit Hilfe des UV-Fixpunktes konstruiert wurde. Man beachte, daß $\mathcal{V}_{QU} \subsetneq \mathcal{V}_{UV}$, da z.B. Z_{UV} nicht in $\mathcal{V}_{QU}$ enthalten ist. In $\mathcal{V}_{QU}$ befinden sich nur Theorien, von denen man den Hochtemperaturfixpunkt abspalten kann, so daß der Rest beschränkt bleibt. Sprechen wir in Zukunft vom Theorieraum $\mathcal{V}$, so impliziert dies Gültigkeit für $\mathcal{V}_{UV}$ und $\mathcal{V}_{QU}$. In der Statistischen Physik oder der Feldtheorie werden Wechselwirkungsfunktionale jedoch über einen Boltzmann-Faktor erklärt, so daß wir uns letztendlich auf die konvexe, unter $\mathcal{R}$ invariante (2.4) Teilmenge

$$\mathcal{V}^+ = \left\{ Z \in \mathcal{V} \middle| Z(\mathbb{R}) \subseteq \mathbb{R}^+ \right\} \quad (2.12)$$

beschränken müssen. Dementsprechend folgt für den Raum der Potentiale[7]

$$\mathcal{W} = -\ln \mathcal{V}^+ \ . \quad (2.13)$$

Wir müssen noch anmerken, daß $\mathcal{W}$ keine Vektorraumstruktur besitzt, da z.B. das Inverse zu $-\ln Z_{QU}$ oder das neutrale Element im Falle von $\mathcal{V}_{QU}$ nicht enthalten sind.

2.1.2 Die Linearisierung $\mathcal{DR}$

Das Finden von Fixpunkten und zugehörigen Flüssen besitzt nicht nur physikalische Relevanz - es ist im allgemeinen der Beginn bei der Untersuchung

[6] Der Vollständigkeitsbeweis ist äquivalent zu dem Vorgehen bei der Supremumsnorm.

[7] Wir erklären die Wirkung einer Abbildung A auf eine Menge M durch $A(M) = \{A(m) | m \in M\}$.

nichtlinearer Systeme. Die hier vorgestellte Form der RGT ist diskret, sie läßt sich aber problemlos als Differentialgleichung mit kontinuierlichem Flußparameter L schreiben. Zur Untersuchung eines Fixpunktszenarios betrachtet man zuerst die am entsprechenden Fixpunkt linearisierte Transformation. Wir wollen in dieser Arbeit ϕ^4-artige Störungen der freien Theorie untersuchen. Somit linearisieren wir $\mathcal{R}$ bei $Z_{UV} = 1$ bzw. $\mathcal{T}$ bei $V_{UV} = 0$. Es folgt mit Hilfe von Definition (B.12):

$$\mathcal{DR}(Z)(\phi) := \frac{\partial}{\partial \epsilon}\Big|_{\epsilon=0} \mathcal{R}(Z_{UV} + \epsilon Z)(\phi) = \alpha \langle Z \rangle_{\gamma, \beta\phi} \tag{2.14}$$

Die Form von $\mathcal{DT}$ ist identisch. Mittels der Beziehung zwischen Normalordnung und Gauß-Integration - siehe hierzu auch die Anhänge (B.1) und (B.2) - rechnet man leicht nach, daß die normalgeordneten Monome über $\mathbb{R}$ Eigenfunktionen der linearisierten RGT sind. Es gilt:

$$\mathcal{DR}(P_{n,\nu})(\phi) = \alpha \beta^n P_{n,\beta^{-2}(\nu-\gamma)}(\phi) \tag{2.15}$$

Die Wahl von $\gamma = 1 - \beta^2$ bedingt eine normalordnende Kovarianz von $\nu = 1$, um die Eigenwertgleichung (2.15) zu lösen. Der Weg, ν zu fixieren und die Fluktuationskovarianz auf $\gamma = \nu(1-\beta^2)$ festzulegen, wird hier nicht verfolgt. Bezüglich des Hilbertraumes $L_2(\mathbb{R}, d\mu_{\nu=1}(\phi))$ bildet $\{P_{n,\nu=1}\}_{n \in \mathbb{R}_0}$ eine Basis. Allerdings ist dieser Raum kein geeigneter Definitionsbereich für die RGT selbst [GS96] - er ist „zu groß“.

Bezüglich $L_2(\mathbb{R}, d\mu_{\nu=1}(\phi))$ können wir jedoch Aussagen über das Fixpunktszenario machen. Stabile, unstabile und Zentrumsmannigfaltigkeiten werden von Eigenvektoren aufgespannt, deren Betrag kleiner, größer oder gleich eins ist. Schaut man sich eine unendliche Iteration der verschiedenen Eigenvektoren an, ist sofort klar, daß Objekte aus dem stabilen Unterraum auf Null - also den Fixpunkt - abgebildet werden. Dementsprechend werden Vektoren aus dem instabilen Unterraum bezüglich ihres Betrages divergieren, und eine Funktion aus der Zentrumsmannigfaltigkeit auf einer „Kugeloberfläche“ mit dem Radius des Vektorbetrags zu finden ist (Invarianz beim Eigenwert 1, alternierend beim Eigenwert -1, usw.).

Die Eigenwerte zu den normalgeordneten Polynomen $P_{n,\nu}$ sind

$$L^{D+n\left(1-\frac{D}{2}\right)}\,, \tag{2.16}$$

und die Dimensionen der Mannigfaltigkeiten D-abhängig. Siehe hierzu Abbildung 2.1. Man beachte, daß der Massenterm ($n = 2$) in jeder Dimension relevant (L^2) ist und der $:\phi^4:_\nu$ Eigenwert L^{4-D}, für $D < 4$ relevant, in vier Dimensionen marginal wird.

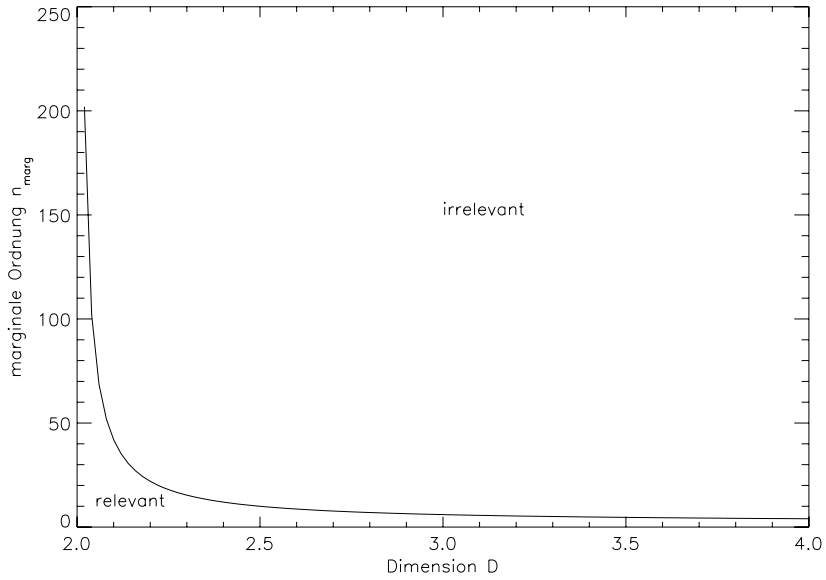


Abbildung 2.1: Mit Hilfe der Formel (2.16) berechnen wir die (reellen) Eigenwertordnungen n_{marg}, für die der Exponent verschwindet. Alle Monome $P_{n,\nu}$ mit $n > n_{marg}$ ($n < n_{marg}$) sind irrelevant (relevant). Schnittpunkte der Kurve mit der Funktionenschar $f_n(x) = n$, $n \in \mathbb{N}_0$ repräsentieren marginale Eigenvektoren.

Für alle Dimensionen $D \in (2, 4]$ mit $\frac{2D}{D-2} \notin \mathbb{N}_0$ handelt es sich bei Z_{UV} also um einen hyperbolischen Fixpunkt, so daß nach dem Hartman-Grobman-Theorem [GH86] ein Homöomorphismus existiert, der die stabilen/unstabilen Eigenräume bzgl. der linearisierten Transformation auf die tangential liegenden invarianten stabilen bzw. nicht stabilen Mannigfaltigkeiten bzgl. der nichtlinearen Transformation überträgt. Für den Fall $\frac{2D}{D-2} \in \mathbb{N}_0$ treten noch Zentrumsmannigfaltigkeiten auf, die nicht eindeutig sind [GH86][8]. Aus diesem Grund ist die Berechnung der Kurve in $D = 4$ komplizierter, da wir in einer marginalen Kopplung parametrisieren. Eine schöne Lösung wäre im übrigen das Finden dieses Homöomorphismus.

[8]Hier ist die nichtlineare Abbildung ein diffeomorphes Vektorfeld über dem $\mathbb{R}^n$. Schränkt man die HRGT jedoch auf polynomiale Potentiale ein und schaltet hinter die Abbildung einen geeigneten Projektor, fällt auch die HRGT in diese Gruppe.

2.2 Die ϕ^4-Trajektorie

Definition 2.2.1 (Die ϕ^4-Trajektorie)
Es sei $Z : \mathbb{R}_0^+ \to \mathcal{V}^+$ eine stetige Abbildung. Z heißt ϕ^4-Trajektorie genau dann, wenn:

1. $Z(0) = Z_{UV}$
2. $Z'(0) = -\frac{1}{4!}P_{4,1}$
3. $\exists \beta : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \in \mathcal{C}^0(R_0^+) : \mathcal{R}(Z(g)) = Z(\beta(g))$

Obige Definition liefert eine Kurve in $\mathcal{V}$, die im UV-Fixpunkt beginnt, in diesem die Steigung $P_{4,1}$ besitzt, also tangential zum normalgeordneten Monom 4. Grades liegt, und invariant unter der HRGT über $\mathcal{V}$ ist. Zu jeder ϕ^4-Trajektorie Z existiert also eine Reparametrisierungsfunktion β (*step-β-Funktion*), und wir sprechen in diesem Zusammenhang auch von einem skalierenden Paar (Z, β).

Schreibt man Z als formale Potenzreihe[9] in $g = 0$, so erhält man mittels der ersten beiden Eigenschaften aus 2.2.1 folgende Darstellung:

$$Z(g) = Z_{UV} - \frac{1}{4!}P_{4,1}g + O(g^2) \quad \Longleftrightarrow \quad Z(g) = e^{-(gP_{4,1}+O(g^2))}$$

$O(g^2)$ und β werden durch die Invarianzeigenschaft festgelegt.

Einige allgemeine Eigenschaften der β-Funktion lassen sich ohne ihre exakte Berechnung angeben: $\beta(0) = 0$,[10] denn würde dies nicht gelten, so wäre die Trajektorie zyklisch oder chaotisch. Diese beiden Fälle wollen wir ausschließen und gehen im folgenden davon aus, daß die ϕ^4-Trajektorie doppelpunktfrei ist.

Unter der Annahme, daß die RGT eine Halbgruppe bildet - nach Satz 1.4.1 trifft dies nur für die GRGT zu - vererbt sich als Folge der Injektivität der Trajektorie in g die Halbgruppeneigenschaft auf β:

$$Z\left(\beta_{L'} \circ \beta_L(g)\right) = \mathcal{R}_L \circ \mathcal{R}_{L'}(Z)(g) = \mathcal{R}_{L'L}(Z)(g) = Z\left(\beta_{L'L}\right)$$

$$\Leftrightarrow \quad \beta_{L'} \circ \beta_L(g) = \beta_{L'L} \tag{2.17}$$

[9] Zur Berechnung einer formalen Potenzreihe benötigen wir nur die $\mathcal{C}^\infty$ -Eigenschaft. Die Frage nach der Konvergenz stellt man zurück.

[10] Aus diesem Grund ist $b_0 = 0$ in (2.25).

Wir nennen (2.17) das Additionstheorem für β-Funktionen. Mit Hilfe der β_L-Funktion können wir die sog. *laufende Kopplung* über

$$g(L) := \beta_L(g) \tag{2.18}$$

definieren. Hierbei kann die Anfangskopplung g auf der rechten Seite beliebig gewählt werden. $g(L)$ gibt dann Auskunft darüber, auf welchem Punkt der invarianten Trajektorie man sich befindet, wenn man eine RGT mit Blockparameter L durchgeführt hat. Definieren wir nun noch die differentielle β-Funktion [Wie97b] über

$$\overline{\beta}(g) := \partial_L \beta_L(g)\Big|_{L=1} , \tag{2.19}$$

so erfüllt die laufende Kopplung die Gleichung

$$Lg'(L) = \overline{\beta}(g(L)) . \tag{2.20}$$

Beweis:

$$Lg'(L) = \partial_{L'} g(LL')\Big|_{L'=1} \overset{(2.17)}{=} \partial_{L'} \beta_{L'}(\beta_L(g))\Big|_{L'=1} = \overline{\beta}(g(L)) \tag{2.21}$$

Die differentielle β-Funktion gibt Auskunft über die Flußrichtung.

Im folgenden nehmen wir an, Z und β seien analytisch in g, so daß wir Störungstheorie betreiben dürfen. Zeigt sich letztendlich, daß Z und/oder β nicht konvergent sind,[11] so heißt dies lediglich, daß keine analytische Lösung existiert. Desweiteren wird sich herausstellen, daß die trunkierten Störungsreihen sehr gute Approximanten darstellen.

2.3 Störungstheorie

Wir wollen in diesem Kapitel das Potential V perturbativ bestimmen. Dazu definieren wir die parameterabhängige Funktion

$$F : [0,1] \to \mathbb{R} \quad F(t)(V)(\phi) := \mathcal{T}(tV)(\phi) \tag{2.22}$$

[11] Konvergenzradius gleich Null

unter der Annahme, daß sie für V und $\phi \in \mathbb{R}$ unendlich oft differenzierbar[12] und die Taylor-Entwicklung um 0 konvergent sind.[13] Es folgt:

$$\begin{aligned}\mathcal{T}(V)(\phi) &= F(1)(V)(\phi) \\ &= \sum_{m=0}^{\infty} \frac{(-1)^{m+1}}{m!} \frac{\partial^m}{\partial t^m} \ln\langle e^{\alpha V(\cdot)t}\rangle_{\gamma,\beta\phi}\Big|_{t=0} \\ &= \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m!} \langle [\alpha V;]^m\rangle^T_{\gamma,\beta\phi} \end{aligned} \tag{2.23}$$

Zur Definition der trunkierten Erwartungswerte lese man im Anhang (B.2). Die ϕ^4-Trajektorie und die Reparametrisierungsfunktion β schreiben wir als formale Potenzreihe in $g = 0$:[14]

$$V(\phi, g) = \sum_{r=1}^{\infty} V_r(\phi) g^r \tag{2.24}$$

$$\beta(g) = \sum_{r=1}^{\infty} b_r g^r \tag{2.25}$$

mit

$$V_1(\phi) =: \phi^4 : \tag{2.26}$$

Aufgrund der Multilinearität der trunkierten Erwartungswerte können wir $\mathcal{T}(V)$ nach Potenzen von g ordnen:

$$\begin{aligned} &\mathcal{T}(V)(\phi, g) \\ &= \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m!} \sum_{r_1=1}^{\infty} \cdots \sum_{r_m=1}^{\infty} g^{\sum_{i=1}^m r_i} \langle [\alpha V_{r_1}, \ldots, \alpha V_{r_m}]\rangle^T_{\gamma,\beta\phi} \\ &= \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m!} \sum_{r=m}^{\infty} g^r \sum_{\sum_{i=1}^r r_i = m} \langle [\alpha V_{r_1}, \ldots, \alpha V_{r_m}]\rangle^T_{\gamma,\beta\phi} \\ &= \sum_{r=1}^{\infty} \left\{ \sum_{m=1}^{r} \frac{(-1)^{m+1}}{m!} \sum_{\sum_{i=1}^m r_i = r} \langle [\alpha V_{r_1}, \ldots, \alpha V_{r_m}]\rangle^T_{\gamma,\beta\phi} \right\} g^r \end{aligned} \tag{2.27}$$

[12] Hierzu müssen die n-ten Ableitungen des RG-Integranden nach t auf $\mathbb{R} \times [0,1]$ gleichmäßig stetig sein. Aus diesem Grund ist z.B. eine Forderung $V \in \mathcal{W}$ zu schwach, da ein Term $V \exp(-\alpha V t)$ multipliziert mit dem Gaußschen Gewichtungsfaktor nicht mehr gleichmäßig stetig sein muß. Beispiel: $V(\phi) = e^{k\phi^2}$ mit $k > \frac{1}{2\gamma}$

[13] Es wird sich herausstellen, daß die Störungsreihe nicht konvergent ist. Aus diesem Grund wollen wir im folgenden lieber den Begriff der formalen Potenzreihe benutzen. Folglich handelt es sich auch bei $\mathcal{T}(V)$ in (2.23) nur um eine formale Potenzreihe.

[14] Wir geben die Entwicklungskoeffizienten ohne $\frac{1}{r!}$-Terme an, da dies eine formale Vereinfachung darstellt.

Ebenso ergibt sich

$$V(\phi, \beta(g)) = \sum_{r=1}^{\infty} \left\{ \sum_{m=1}^{r} \sum_{\sum_{i=1}^{m} r_i = r} \prod_{i=1}^{m} b_{r_i} V_m(\phi) \right\} g^r \ . \tag{2.28}$$

In erster Ordnung erhalten wir die Eigenwertgleichung

$$\alpha \left\langle V_1 \right\rangle_{\gamma, \beta\phi} = b_1 V_1(\phi), \tag{2.29}$$

die aufgrund der Anfangsbedingung (2.26) die eindeutige Lösung

$$b_1 = \alpha\beta^4 = L^{4-D} \tag{2.30}$$

besitzt. In höheren Ordnungen $r \geq 2$ gilt es

$$\alpha \left\langle V_r \right\rangle_{\gamma, \beta\phi} - b_1^r V_r(\phi) = b_r V_1(\phi) + L_r(\beta, V)(\phi) - K_r(V)(\phi) \tag{2.31}$$

zu lösen. Hierbei sind

$$L_r(\beta, V)(\phi) = \sum_{m=2}^{r-1} \sum_{\sum_{i=1}^{m} r_i = r} \prod_{i=1}^{m} b_{r_i} V_m(\phi) \ , \tag{2.32}$$

$$K_r(V)(\phi) = \sum_{m=2}^{r} \frac{(-1)^{m+1}}{m!} \sum_{\sum_{i=1}^{m} r_i = r} \left\langle [\alpha V_{r_1}, \ldots, \alpha V_{r_m}] \right\rangle_{\gamma, \beta\phi}^{T} \ . \tag{2.33}$$

Wir merken an, daß K_r und L_r unabhängig von V_r und b_r sind, so daß eine rekursive Lösung des Problems möglich ist. Da V_r analytisch und symmetrisch sein soll, stellen wir es durch eine Potenzreihe in normalgeordneten[15], geraden[16] Monomen dar:

$$V_r(\phi) = \sum_{n=0}^{\infty} V_{2n,r} : \phi^{2n} : \tag{2.34}$$

Aufgrund der Tatsache, daß es sich beim Startterm V_1 um eine endliche Reihe handelt ($V_{2n,1} = 0$ für $n \geq 3$), pflanzt sich diese Eigenschaft in den

[15] Eine Darstellung in der Basis $\{\phi^{2n}\}_{n \in \mathbb{N}_0}$ ist genauso gut möglich, doch vereinfacht die Benutzung der Eigenybasis $\{: \phi^{2n} :\}_{n \in \mathbb{N}_0}$ die Berechnungen erheblich.

[16] Wie man leicht nachrechnet, gilt: $P_{2n,\nu}(\phi) = \sum_{m=0}^{n} P_{2n,2m}(v) \phi^{2(n-m)}$: Es folgt also, daß $: \phi^{2n} : \in Lin\left(1, \phi^2, \ldots, \phi^{2n}\right)$ und ebenso $\phi^{2n} \in Lin\left(1, : \phi^2 :, \ldots, : \phi^{2n} :\right)$. Somit werden gerade Funktionen auch durch Potenzreihen in $: \phi^{2n} :$ repräsentiert.

Kumulanten fort. So ist K_r ein normalgeordnetes Polynom vom Grade $r+1$. Es gilt also für alle $n > r+1$:[17]

$$(P_{2n,1}, K_r)_1 = 0 \tag{2.35}$$

Dies zeigt man leicht mittels vollständiger Induktion. Im Induktionsschritt benutzt man

$$\begin{aligned} & \langle [\alpha V_{r_1}, \ldots, \alpha V_{r_m}] \rangle^T_{\gamma,\beta\phi} \qquad (2.36) \\ \overset{IV}{=} & \sum_{n_1=0}^{r_1+1} \cdots \sum_{n_m=0}^{r_m+1} \alpha^m V_{2n_1,r_1} \ldots V_{2n_m,r_m} \langle [P_{2n_1,1}, \ldots, P_{2n_m,1}] \rangle^T_{\gamma,\beta\phi} \\ = & \sum_{n_1=0}^{r_1+1} \cdots \sum_{n_m=0}^{r_m+1} \alpha^m V_{2n_1,r_1} \ldots V_{2n_m,r_m} \sum_{n=0}^{n_{max}} C_{n,n_1,\ldots,n_m}(\beta,\gamma) : \phi^n :_1 \ . \end{aligned}$$

Die genaue Form der Koeffizienten $C_{n,n_1,\ldots,n_m}(\beta,\gamma)$ ist unwichtig, essentiell ist hingegen die Bestimmungsformel für n_{max}:[18]

$$n_{max} = 2\sum_{i=1}^{m} n_i - 2(m-1) \leq 2\sum_{i=1}^{m}(r_i+1) - 2(m-1) = 2(r+1) \tag{2.37}$$

Die bislang noch unbeantwortete Frage nach der Konvergenz der Störungsreihe stellen wir ein wenig zurück. Physikalisch sinnvoll ist nur ein Konvergenzgebiet, das einen Quader $\mathbb{R} \times [0, g_{max})$ beinhaltet. Schon numerische Simulationen in [Rol96] sprachen gegen eine Konvergenz der perturbativen Trajektorie, ein weiteres Argument für die Divergenz liefern wir in Abschnitt 2.6.4.

2.3.1 Die lineare β-Funktion für $2 < D < 4$

Die Wahl der linearen β-Funktion

$$\beta(g) = L^{4-D} g \tag{2.38}$$

bedeutet $b_r = 0$ für alle $r \geq 2$ und (2.31) wird zu

$$\alpha \langle V_r \rangle_{\gamma,\beta\phi} - b_1^r V_r(\phi) = -K_r(V)(\phi) \ . \tag{2.39}$$

[17] Hierbei nutzen wir die Orthogonalitätsrelation der normalgeordneten Monome bezüglich des Skalarproduktes $(f,g)_\nu = \int d\mu_\nu(\zeta) f(\zeta) g(\zeta)$. Es gilt $(P_{n,\nu}, P_{m,\nu})_\nu = \nu^n n! \delta_{n,m}$.

[18] Diese Formel ist grafisch sofort klar: die Kontraktion mit maximaler Beinzahl erhält man, wenn $m-2$ Vertices jeweils 2 Beine und 2 Vertices jeweils 1 Bein opfern.

Mit Hilfe von (2.34) und (2.15) erhalten wir die Bestimmungsgleichung

$$\left\{1 - \frac{\alpha\beta^{2n}}{(\alpha\beta^4)^r}\right\} V_{2n,r} = \frac{1}{(2n)!(\alpha\beta^4)^r}(P_{2n,1}, K_r(V))_\gamma \ . \tag{2.40}$$

Durch diese Gleichung werden die Koeffizienten $V_{2n,r}$ eindeutig bestimmt. Im Falle

$$\left\{1 - \frac{\alpha\beta^{2n}}{(\alpha\beta^4)^r}\right\} = 0 \Leftrightarrow D - n(D-2) - r(4-D) = 0 \tag{2.41}$$

muß gewährleistet sein, daß die rechte Seite von (2.40) identisch Null ist - $V_{2n,r}$ wird zu einem frei wählbaren Parameter. Ansonsten sprechen wir von (n,r)-Resonanzen, die Trajektorie ist perturbativ nicht bestimmbar.[19] Für den Fall, daß $n > r+1$ und $r > 1$ folgt

$$D + n(2-D) - r(4-D) < 2(1-r) < 0 \ , \tag{2.42}$$

so daß mit (2.35) der polynomiale Ansatz

$$V_r(\phi) = \sum_{n=0}^{r+1} V_{2n,r} : \phi^{2n} : \tag{2.43}$$

gerechtfertigt ist. Stellt sich noch die Frage, wann Resonanzen auftreten. Zu fester Dimension D ergibt sich aus (2.41) die streng monoton fallende Folge

$$n_D(r) = \underbrace{\frac{4-D}{2-D}}_{<0} r + \underbrace{\frac{D}{D-2}}_{>0} \ , \tag{2.44}$$

die nach oben durch $n_D(2) = 3 - \frac{2}{D-2}$ beschränkt ist. Hieraus folgt, daß in $2 < D < \frac{8}{3}$ keine Resonanzen auftreten können. In $\frac{8}{3} \leq D < 3$ können nur eine Vakuumresonanz[20] $(n=0)$ und in $3 \leq D < 4$ eine Vakuumresonanz und/oder eine Massenresonanz[21] $(n=1)$ erscheinen. Resonante Terme treten für $D \to 4$ erst in immer höheren Ordnungen auf. Die Resonanz behafteten Dimensionen besitzen $D = 4$ als Häufungspunkt.

Man beachte noch, daß die lineare β-Funktion das Additionstheorem (2.17) erfüllt. Dies zeigt, daß die lineare Reparametrisierung auch für die volle RGT eine geeignete Wahl darstellt. Die nach (2.19) bestimmte differentielle β-Funktion lautet $\beta(g) = (4-D)g$.

[19] Zumindest nicht mit diesem Ansatz.

[20] Hierzu muß es eine Ordnung $r \in \mathbb{N}_2$ geben, so daß $D = \frac{4r}{1+r}$.

[21] Hierzu muß es eine Ordnung $r \in \mathbb{N}_2$ geben, so daß $D = 4 - \frac{2}{r}$.

Ferner verdeutlicht sich anhand der *step-β*-Funktion die Wirkung der RGT in $2 < D < 4$ Dimensionen: Ein RG-Schritt treibt uns auf der Trajektorie aus dem trivialen Fixpunkt heraus. Eine unendliche Iteration von RG-Schritten führt uns somit zur Fixpunkttheorie $Z(\infty)$, welche dieselben kritischen Eigenschaften besitzt wie alle Theorien $Z(g)$ mit $g > 0$. Eine Trajektorie mit streng monotoner Reparametrisierungsfunktion verbindet also immer die Fixpunkte $Z(0)$ und $Z(\infty)$, und die Flußrichtung ist eindeutig. Zur Berechnung von $Z(\infty)$ benötigen wir nur noch die Trajektorie selbst.

Die *step-β*-Funktion ist auf $\mathbb{R}_0^+$ definiert und umkehrbar. Wir erklären

$$\delta = \beta^{-1} : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \quad \delta(g) = \delta g \quad \text{mit} \quad \delta = L^{D-4} \tag{2.45}$$

und schreiben die Invarianzgleichung in eine Fixpunktgleichung um:

$$\mathcal{R} \times \delta^\star(Z)(g) := \mathcal{R}(\delta(g)) = Z(g) \tag{2.46}$$

2.3.2 Die kubische β-Funktion für $D = 4$

Die lineare β-Funktion verkommt in 4 Dimensionen zur Identität. Somit findet sich auf der Trajektorie für $g > 0$ überall dieselbe Theorie. Die Kurve reduziert sich zu einem „Punkt“.[22] Die Wirkung $Z(g > 0)$ ist ein weiterer Fixpunkt der RGT. Diese Unstetigkeit in $g = 0$ ist Motivator für den folgenden Beweis, daß für $D = 4$ kein skalierendes Paar mit linearer Reparametrisierungsfunktion existiert. Aus (2.29) folgt $b_1 = 1$ und (2.31) schreibt sich mit Hilfe des Ansatzes (2.43)[23] als

$$\left\{1 - \alpha\beta^{2n}\right\} V_{2n,r} = \frac{1}{(2n)!}(P_{2n,1}, K_r(V) - L_r(\beta, V) - b_r V_1)_1 \ . \tag{2.47}$$

(2.47) läßt sich für $n \in \{0, \ldots, r+1\} \setminus \{2\}$ eindeutig lösen.[24] Für $n = 2$ wird die linke Seite identisch Null und (2.47) bestimmt b_r zu

$$b_r = \frac{1}{4!}(P_{4,1}, K_r(V) - L_r(\beta, V))_1 \ . \tag{2.48}$$

Für $r = 2$ ergibt sich nach kurzer Rechnung unter Benutzung der Kumulantenformel (B.20)

$$b_2 = \frac{1}{4!}\left(P_{4,1}, -\frac{1}{2}\left\langle \alpha V_1; \alpha V_1 \right\rangle_{\gamma,\beta.}^T\right)_1 = -36(L^4 - 1) \ . \tag{2.49}$$

[22] Genauer gesagt zu zwei Punkten: $Z(0) = Z_{UV}$ und $Z(g > 0)$.

[23] Da L_r von der Ordnung $: \phi^{2r} :$ ist, folgt wie schon zuvor $V_{2n,r} = 0$ für $n > r + 1$.

[24] in Abhängigkeit von b_r

Diese Gleichung zeigt, daß die Wahl einer linearen β-Funktion nicht möglich ist. $V_{4,2}$ wird zu einem frei wählbaren Parameter. Es stellt sich die Frage, ob man nun $V_{4,2}$ so wählen kann, daß z.B. $b_3 = 0$. Allgemeiner formuliert: Determiniert eine willkürliche Wahl der $b_{r\geq 3}$ die freien Parameter $V_{4,r\geq 2}$? Für $r \geq 3$ gilt:

$$K_r(V)(\phi) = -\langle \alpha V_1; \alpha V_{r-1}\rangle^T_{\gamma,\beta\phi} + \tilde{K}_r(V)(\phi) \tag{2.50}$$

$$L_r(\beta, V)(\phi) = (r-1)b_2 V_{r-1}(\phi) + \tilde{L}_r(\beta, V)(\phi)\,, \tag{2.51}$$

wobei $\tilde{K}_r$ und $\tilde{L}_r$ nur aus $V_{\tilde{r}}$ mit $\tilde{r} < r-1$ bestehen. Es folgt für $r \geq 3$:

$$b_r = (3-r)\, b_2 V_{4,r-1} + \mathcal{N} \tag{2.52}$$

$\mathcal{N}$ besteht aus schon bekannten Größen. In dritter Ordnung fällt der erste Summand auf der rechten Seite weg, so daß b_3 noch bestimmt und $V_{4,2}$ ein freier Parameter ist. Es ergibt sich[25]

$$b_3 = 432 - 3456L^2 - 2592L^4 + 3456L^6 + 2160L^8 \tag{2.53}$$

Alle Koeffizienten $b_{r>3}$ setzen wir zu Null. Es folgt eine Determinierung der „freien“ Parameter $V_{4,r-1}$ entsprechend (2.52).[26] Die Störungsrechnung in $D = 4$ Dimensionen liefert somit ein skalierendes Paar bestehend aus der kubischen β-Funktion

$$\beta(g) = g - 36(L^4-1)g^2 + 432\left(1 - 8L^2 - 6L^4 + 8L^6 + 5L^8\right) g^3 \tag{2.54}$$

und dem Potential V, das bis auf den freien Parameter $V_{4,2}$ eindeutig bestimmt ist.

Wir wollen nun überprüfen, ob diese perturbativ bestimmte Reparametrisierungsfunktion als Grundstock für konstruktive Berechnungen geeignet ist. Als erstes stellen wir fest, daß β auf $\mathbb{R}$ streng monoton steigend ist, da die Diskriminante der 1. Ableitung für $L > 1$ echt kleiner Null ist und $\beta'(0) = 1 > 0$. Damit existiert die Umkehrfunktion $\delta := \beta^{-1}$, die wir zum Beispiel mittels des Satzes über implizite Funktionen berechnen können $(\delta(g) = g - b_2 g^2 + (2b_2^2 - b_3)g^3 + O(g^4))$. Die approximierte Umkehrfunktion 3. Grades liegt aber z.B. erst für $0 \leq g < 10^{-3}$ im richtigen 1. Quadranten. Natürlich läßt sich mit Hilfe der Cardanoschen Formeln, welche die drei Lösungen einer algebraischen kubischen Gleichung durch Radikale beschreiben, δ auch exakt bestimmen.

[25] b_3 wurde mittels *MapleV* und der Basis $\{\phi^{2n}\}_{n\in\mathbb{N}_0}$ berechnet. Dies ändert jedoch nichts.

[26] Dies kann man natürlich auch aus der entgegengesetzten Blickrichtung betrachten.

Es stellt sich die Frage nach Fixpunkten $\beta(g) = g$. Wir erhalten $g_{UV} = 0$ und

$$\overline{g} = \frac{1}{12\,(5L^4 + 8L^2 - 1)}\ . \tag{2.55}$$

Somit repräsentiert $Z(\phi, \overline{g})$ eine (perturbative) infrarote Fixpunkttheorie. Wir schließen den für das hierarchische Modell pathologischen Fall $L \to \infty$ aus.[27]

Daß ein Polynom vom Grade größer eins nicht die Kompositionseigenschaft für β-Funktionen (2.17) erfüllt, ist klar. Somit ist diese Funktion kein Kandidat für die volle RGT. Dessen perturbativ berechnete β-Funktion schreibt sich als [Wie97d]

$$\beta(g) = g - \frac{3\log(L)}{(4\pi)^2} g^2 + O(g^3)\ . \tag{2.56}$$

Durch die logarithmische L-Abhängigkeit der Koeffizienten ist Kompositionseigenschaft in der vollen RGT erfüllt. Man beachte jedoch, daß zur Berechnung von (2.56) die Parametrisierung $\hat{V}_{4,r} = \delta_{r,1}$ benutzt wurde. Allerdings merkte WIECZERKOWSKI in [Wie97c] an, daß auch im vollen Modell die Koeffizienten des kubischen Anteils universell sind und eine Reparametrisierung $b_{r>3} = 0$ der Trajektorie möglich ist.[28]

Das qualitative Verhalten ist jedoch für beide Modelle gleich, da der g^2-Term ein negatives Vorzeichen besitzt und somit für $0 \leq g \ll 1$ die Eigenschaft $\beta(g) \leq g$ folgt - Theorien mit kleinen Kopplungen laufen unter unendlicher RG-Iteration in den trivialen Fixpunkt.

Für das hierarchische Modell kann man konkret angeben, daß Theorien $V(0 < g < \overline{g})$ in den trivialen Fixpunkt ($\beta(g) < g$) und Punkte $V(g > \overline{g})$ nach $V(\infty)$ ($\beta(g) > g$) laufen, sofern diese Konvergenz existiert. Zur Verdeutlichung schaue man sich die differentielle β-Funktion (2.19) an, die die Kopplungsänderung in der Nähe von $L = 1$ widerspiegelt. In $D = 4$ ist die freie Theorie also attraktiv und der infrarote Fixpunkt repulsiv.[29]

2.4 Der Raum der Trajektorien

Im folgenden werden wir ob der einfacheren Notation immer das kartesische Produkt von Feld- und Kopplungsraum betrachten. Wir definieren über die

[27] Das hätte $\overline{g}(L) \to 0$ zur Folge.

[28] Diese Aussage bezieht sich jedoch auf die differentielle β-Funktion.

[29] Man beachte, daß in unserer Terminologie die Äquivalenzen IR-Fixpunkt = nicht-Gaußscher Fixpunkt, UV-Fixpunkt = trivialer Fixpunkt gelten. In [MM94] spricht man im Falle eines attraktiven/repulsiven Fixpunktes von einem IR-/UV-Fixpunkt.

Maximalkopplung $g_0 \in \mathbb{R}^+$ die Menge

$$\mathcal{P}_{g_0} = \mathbb{R} \times [0, g_0] \quad \text{mit} \quad \mathcal{P}_\infty = \lim_{g_0 \to \infty} \mathcal{P}_{g_0} = \mathbb{R} \times \mathbb{R}_0^+ \ . \tag{2.57}$$

In 2.1.2 haben wir den Funktionenraum $\mathcal{V}_{\text{Gauß}}$ kennengelernt und gesehen, daß die HRGT auf ihm wohldefiniert ist - eine Gauß-Funktion transformiert sich in eine Gauß-Funktion. Daß eine unter der HRGT invariante, aus Gauß-Funktionen bestehende Trajektorie existiert, zeigt folgender

Satz 2.4.1

$$Z_{QU} : \mathcal{P}_\infty \to \mathbb{R}^+ \qquad (\phi, g) \mapsto e^{a_{QU}(g) - \frac{b_{QU}(g)}{2}\phi^2}$$

mit

$$a_{QU}(g) = \frac{1 - \alpha^{-1}}{2} \sum_{n=1}^{\infty} \alpha^{-n} \ln\left(\frac{1 + (\delta^{-n} g)^\rho}{1 + g^\rho}\right), \qquad b_{QU}(g) = b_{QU} \frac{g^\rho}{1 + g^\rho}$$

$$\rho = \frac{2}{4 - D}$$

ist ein Fixpunkt der Abbildung $\mathcal{R} \times \delta^\star$.

Beweis: Z_{QU} ist eine in g parametrisierte Kurve in $\mathcal{V}_{\text{Gauß}}$, da $b_{QU}(\mathbb{R}_0^+) \subseteq \mathbb{R}_0^+$ und b_{QU} stetig ist. Bei a_{QU} zeigt man diese Eigenschaften mittels des Majorantenkriteriums. Es sei $g \in \mathbb{R}_0^+$ beliebig, aber fest:

$$\left|\alpha^{-n} \ln\left(\frac{1 + (\delta^{-n} g)^\rho}{1 + g^\rho}\right)\right| \leq \alpha^{-n} \ln\left(1 + \left(\frac{1}{L^2}\right)^n g^\rho\right) \leq \left(\frac{1}{\alpha}\right)^n \ln\left(1 + \frac{1}{L^2} g^\rho\right)$$

Aus dieser Abschätzung folgt $a_{QU}(\mathbb{R}_0^+) \subseteq \mathbb{R}$ und die gleichmäßige Konvergenz auf beliebigen kompakten Intervallen aus $\mathbb{R}_0^+$, und somit die Stetigkeit von a_{QU}.

Die Transformation $\mathcal{R} \times \delta^\star$ ist dadurch wohldefiniert. Es gilt noch zu beweisen, daß $\mathcal{R} \times \delta^\star(Z_{QU}) = Z_{QU}$. Nach Satz 2.1.2 gilt $\forall (\phi, g) \in \mathcal{P}_\infty$:

$$\mathcal{R} \times \delta^\star(Z_{QU}(\phi, g)) = \mathcal{R}(Z_{QU}(\phi, \delta g) = e^{a'_{QU}(g) - \frac{b'_{QU}(g)}{2}\phi^2} \tag{2.58}$$

mit

$$a'_{QU}(g) = \alpha a_{QU}(\delta g) - \frac{1}{2} \ln\left(1 + \alpha\gamma b_{QU}(\delta g)\right) \tag{2.59}$$

$$b'_{QU}(g) = \frac{\alpha\beta^2 b_{QU}(\delta g)}{1 + \alpha\gamma b_{QU}(\delta g)} \ . \tag{2.60}$$

Durch Einsetzen zeigt man $a'_{QU} \equiv a_{QU}$ und $b'_{QU} \equiv b_{QU}$. Eine mögliche Herleitung von b über die Hilfsfunktion $c = b^{-1}$ und die Bestimmung von a durch sukzessives Einsetzen in (2.59) findet der Leser in [Wie97a]. In dieser Arbeit wollen wir b jedoch mittels Störungsrechnung in g^ρ berechnen ($\rho \in \mathbb{R}^+$). Der Parameter ρ ermöglicht es, auch nicht unendlich oft differenzierbare Lösungen zu finden. Um den Vakuumterm a kümmern wir uns nicht, da er in der normierten Transformation sowieso bedeutungslos wird. Es sei also

$$b(g) = \sum_{k=1}^{\infty} b_k g^{\rho k} \ . \tag{2.61}$$

Wir tragen in diesem Ansatz der Bedingung Rechnung, daß die Kurve für $g = 0$ im UV-Fixpunkt beginnen soll. Mit Hilfe von $A = \alpha\beta^2 = L^2$, $B = \alpha\gamma$ und der Eigenschaft $\left|\frac{B}{A} b(g)\right| < 1$ schreibt sich (2.60) als[30]

$$b(\delta g) = \sum_{m=1}^{\infty} \frac{1}{B} \left(\frac{B}{A} b(g)\right)^m \ . \tag{2.62}$$

Mit der Vereinfachung $\tilde{b}_k = \frac{B}{A} b_k \Leftrightarrow \tilde{b} = \frac{B}{A} b$ ergibt sich analog zur Störungsrechnung in Kapitel 2.3 durch Koeffizientenvergleich für alle k:

$$A\tilde{b}_k \delta^{\rho k} = \sum_{m=1}^{k} \sum_{\substack{\sum_{i=1}^{m} n_i = k \\ n_i \in \mathbb{N}}} \tilde{b}_{n_1} \ldots \tilde{b}_{n_m} \tag{2.63}$$

In erster Ordnung ergibt sich, daß $\rho = \frac{2}{4-D}$ und $\tilde{b}_1$ ein freier Parameter ist. Alle übrigen $\tilde{b}_n$ lassen sich rekursiv bestimmen. Explizite Berechnungen der nächsten Ordnungen erhärten den Verdacht, daß für die übrigen Koeffizienten die Gleichung

$$\tilde{b}_k = \left(\frac{L^2}{1 - L^2}\right)^{k-1} \tilde{b}_1^k =: C^{k-1} \tilde{b}_1^k \tag{2.64}$$

gilt. Der Beweis erfolgt durch Einsetzen:

$$\tilde{b}_k \overset{(2.63)}{=} \frac{1}{L^{2(1-k)} - 1} \sum_{m=2}^{k} \sum_{\substack{\sum_{i=1}^{m} n_i = k \\ n_i \in \mathbb{N}}} \tilde{b}_{n_1} \ldots \tilde{b}_{n_m}$$

[30] Die Ungleichung $\left|\frac{B}{A} b(g)\right| < 1$ gilt ob der angenommenen Stetigkeit in $g = 0$ gewiß für kleine g. Das Resultat (2.66) mit $\tilde{b}_1 > 0$ erfüllt diese Relation sogar für alle $g \in \mathbb{R}_0^+$.

$$\begin{aligned}
&\overset{(2.64)}{=} C^{k-1}\tilde{b}_1^k \frac{C}{L^{2(1-k)}-1} \sum_{m=2}^{k} C^{-m} \sum_{\substack{\sum_{i=1}^{m} n_i = k \\ n_i \in \mathbb{N}}} 1 \\
&= C^{k-1}\tilde{b}_1^k \frac{C}{L^{2(1-k)}-1} \sum_{m=2}^{k} C^{-m} \frac{m}{k} \binom{k}{m} \\
&\overset{(\star)}{=} C^{k-1}\tilde{b}_1^k \frac{\partial_{C^{-1}} \left(1+C^{-1}\right)^k - 1}{L^{2(1-k)}-1} \\
&= C^{k-1}\tilde{b}_1^k \qquad (2.65)
\end{aligned}$$

In $(\star)$ wurde die Beziehung $\partial_x (1+x)^k = \sum_{m=1}^{k} m \binom{k}{m} x^{m-1}$ benutzt. Für den invarianten Massenterm ergibt sich somit:

$$b(g) = \frac{A}{BC} \frac{C\tilde{b}_1 g^\rho}{1 - C\tilde{b}_1 g^\rho} \qquad (2.66)$$

Da $C < 0$ folgt für jedes $\tilde{b}_1 > 0$, daß b auf $\mathbb{R}_0^+$ stetig ist. Unabhängig von der exakten Wahl gilt dann

$$\lim_{g\to\infty} b(g) = \frac{A}{BC} = \frac{A-1}{B} = b_{QU} \ . \qquad (2.67)$$

Für $\tilde{b}_1 = -C^{-1}$ erhalten wir $b = b_{QU}$.

□

Wir haben also mit Z_{QU} eine Trajektorie vorliegen, die im trivialen Fixpunkt beginnt ($Z_{QU}(\cdot, 0) = Z_{UV} \in \mathcal{V}_{\text{Gauß}}$) und in den Hochtemperaturfixpunkt läuft ($\lim_{g\to\infty} Z_{QU}(\cdot, g) = Z_{QU} \in \mathcal{V}_{\text{Gauß}}$). Man beachte aber, daß $Z_{QU}(\cdot, \mathbb{R}_0^+) \subsetneq \mathcal{V}_{\text{Gauß}}$, da z.B. $b_{QU}(\mathbb{R}_0^+) = [b_{UV}, b_{QU})$.

Arbeiten wir mit der normierten Transformation, so müssen wir den konstanten Term nicht beachten und erhalten die Gauß-Trajektorie

$$Z(\phi, g) = e^{-\frac{b_{QU}(g)}{2}\phi^2} \ . \qquad (2.68)$$

Wir merken noch an, daß $a_{QU}(g) = O(g^\rho)$ und $b_{QU}(g) = b_{QU} g^\rho + O\left((g^\rho)^2\right)$.

Es stellt sich die Frage, warum wir die perturbative b-Konstruktion der eleganten Methode von Wieczerkowski vorziehen? Wie man leicht sieht, ist die Gauß-Trajektorie für $D = 4$ nicht mehr definiert. Die beiden Fixpunkte, die durch die Kurve $Z_{QU}(g)$ verbunden werden, existieren aber dennoch. Für

eine nichtlineare δ-Funktion, die wir in 4 Dimensionen benutzen müssen (siehe 2.3.2), stellt die Störungstheorie jedoch ein mögliches Verfahren für die Konstruktion einer invarianten Massenkopplung unter der erweiterten RGT dar.

Jetzt konstruieren wir - analog zu (2.9) und (2.10) - mit Hilfe des quadratischen Fixpunktes der Transformation $\mathcal{R} \times \delta^\star$ einen Raum von Funktionenkurven bzw. Funktionen in zwei Variablen. Es sei $g_0 > 0$.

$$\begin{aligned} \mathcal{V}_{g_0} &= \Big\{ Z : \mathcal{P}_{g_0} \to \mathbb{R} \Big| Z(\cdot, g) \in \mathcal{C}^0(\mathbb{R}),\ Z(\phi, \cdot) \in \mathcal{C}^0([0, g_0]), \\ &\quad Z(\cdot, g) \in \mathbb{Z}_2(\mathbb{R})\, \forall g \in [0, g_0],\ \sup_{(\phi,g)\in\mathcal{P}_{g_0}} \left| \frac{Z(\phi, g)}{Z_{QU}(\phi, g)} \right| < \infty \Big\} \end{aligned} \tag{2.69}$$

Wir ergänzen die Abbildungen

$$g \in [0, g_0], \quad \|\cdot\|_g : \mathcal{V}_{g_0} \to \mathbb{R}_0^+ \qquad Z \mapsto \sup_{\phi\in\mathbb{R}} \left| \frac{Z(\phi, g)}{Z_{QU}(\phi, g)} \right| \tag{2.70}$$

$$|||\cdot|||_{g_0} : \mathcal{V}_{g_0} \to \mathbb{R}_0^+ \qquad Z \mapsto \sup_{g\in[0,g_0]} \|Z\|_g \ . \tag{2.71}$$

Der $\mathbb{R}$-Vektorraum $\mathcal{V}^{g_0}$ wird durch die Norm $|||\cdot|||_{g_0}$ zu einem Banachraum. Da für die lineare δ-Funktion die Eigenschaft $\delta(g) < g$ gegeben ist, ist die erweiterte RGT $\mathcal{R} \times \delta^\star$ auf $\mathcal{V}_{g_0}$ selbstabbildend. Schränken wir $\mathcal{V}_{g_0}$ auf Theorien zu einem festen Kurvenparameter g ein, so ist auf diesem Unterraum auch $\|\cdot\|_g$ eine Norm. Für $g = 0$ bzw. $g = \infty$ erhalten wir $\mathcal{V}_{UV}$ bzw. $\mathcal{V}_{QU}$. $\mathcal{V}_\infty$ stellt folglich eine Menge von Theorieräumen dar, deren Objekte zwischen $\mathcal{V}_{UV}$ und $\mathcal{V}_{QU}$ interpolieren und sich durch $Z_{QU}(g)$ abschätzen lassen.

Es sei $g_0^1 < g_0^2$. Werden die Funktionen aus $\mathcal{V}_{g_0^2}$ auf $\mathcal{P}_{g_0^1}$ eingeschränkt, so gilt $\mathcal{V}_{g_0^2} \subset \mathcal{V}_{g_0^1}$.

Korrespondierend zu (2.13) definieren wir aus der konvexen Teilmenge von $\mathcal{V}_{g_0}$, die aus den positiven Funktionen besteht, den Raum der Potentiale $\mathcal{W}_{g_0}$, auf dem die erweiterte Transformation $\mathcal{T} \times \delta^\star$ agiert.

Abschließend wollen wir noch einmal die Bedeutung der Transformation $\mathcal{R} \times \delta^\star$ herausstellen: Sie wirkt auf einem Raum von Kurven, und ihre Fixpunkte stellen unter der HRGT invariante Trajektorien dar.

2.5 Existenz und Konstruktion eines Fixpunktes

Ziel dieses Paragraphen ist es, Kriterien zu finden, die einen Punkt aus $\mathcal{V}_{g_0}$ zu einem approximativen Fixpunkt der Transformation $\mathcal{R} \times \delta^\star$ machen, so daß wir um diese Funktion eine Menge konstruieren können, in der gewiß ein Fixpunkt liegt. Der Banachsche Fixpunktsatz, welcher uns die Existenz dieses Fixpunktes beweist, liefert auch sogleich ein Konstruktionsverfahren desselbigen.

Zu Beginn zwei Definitionen, die uns das Leben leichter machen. Die erste vereinfacht uns die Handhabung der in diesem Abschnitt häufig auftretenden Indizes C, σ, g, die zweite bietet Transformationen, mit deren Hilfe wir $\mathcal{R} \times \delta^\star$ zerlegen können. Im folgenden sei $g_0 \in \mathbb{R}^+$ vorausgesetzt.

Definition 2.5.1

$$\mathcal{I} = \mathbb{R}^+ \times \mathbb{R}^+ \times [0, g_0], \qquad X^\alpha = (C_\alpha, \sigma_\alpha, g_\alpha) \in \mathcal{I} \tag{2.72}$$

Definition 2.5.2
Es sei $Z_1 \in \mathcal{V}_{g_0}$. Dann definiere

$$\triangle : \mathcal{V}_{g_0} \to \mathcal{V}_{g_0} \qquad Z \mapsto (\mathcal{R} \times \delta^\star - id)(Z) \tag{2.73}$$

$$\mathcal{R}_{Z_1} : \mathcal{V}_{g_0} \to \mathcal{V}_{g_0} \qquad Z \mapsto \mathcal{R} \times \delta^\star(Z_1 + Z) - \mathcal{R} \times \delta^\star(Z_1)\ . \tag{2.74}$$

Die Wohldefiniertheit der beiden Abbildungen folgt aus der von $\mathcal{R} \times \delta^\star$, welche sich mit Hilfe der obigen Definition auch darstellen läßt als:

$$Z_2 \in \mathcal{V}^{g_0} \Rightarrow \mathcal{R} \times \delta^\star(Z_1 + Z_2) = Z_1 + \triangle(Z_1) + \mathcal{R}_{Z_1}(Z_2) \tag{2.75}$$

Ferner liefert uns die Funktion $\triangle$ ein Maß für die Güte eines angenäherten Fixpunktes von $\mathcal{R} \times \delta^\star$, indem wir die Abweichungen in jedem Punkt mittels $\|\triangle(\cdot)\|_g$ oder die maximale Abweichung per $\||\triangle(\cdot)|\|_{g_0}$ berechnen.

Was nun einen beliebigen Punkt aus $\mathcal{V}_{g_0}$ zu einem angenäherten Fixpunkt macht, klärt folgende

Definition 2.5.3 (Der approximierte Fixpunkt)
$Z_1 \in \mathcal{V}_{g_0}$ ist ein approximierter Fixpunkt genau dann wenn gilt:

$$\exists X^1, X^\triangle \in \mathcal{I} \quad mit \quad \sigma_\triangle > \tfrac{D}{4-D} : \qquad \begin{aligned} Z_1 &\in \mathcal{V}^+_{g_0} \\ \|Z_1\|_g &\leq e^{C_1 g^{\sigma_1}} \quad \forall g \in [0, g_1] \\ \|\triangle(Z_1)\|_g &\leq C_\triangle g^{\sigma_\triangle} \quad \forall g \in [0, g_\triangle] \end{aligned}$$

Ein solcher approximierter Fixpunkt Z_1 hat die Eigenschaft, daß seine Trajektorie denselben Ursprung besitzt wie der exakte Fixpunkt, da $\|\triangle(Z_1)\|_0 = 0$. Je größer der Wert von $\sigma_\triangle$, desto mehr schmiegt sich die approximierte an die reale Fixpunkttrajektorie an (für $g \leq 1$). Die erste Forderung macht Z_1 zu einer physikalisch sinnvollen Fixpunktapproximante.[31] Die zweite Abschätzung fließt bei der Konstruktion eines Konus um Z_1 ein, innerhalb dessen nur positive, also durch Wirkungen realisierbare, Funktionen liegen.

Dieser Konus sei eine Menge von folgender Gestalt:

Definition 2.5.4 (Die approximierte Fixpunktumgebung)
Es sei $X^2 \in \mathcal{I}$ und $Z_1 \in \mathcal{V}_{g_0}$ ein approximierter Fixpunkt. Dann definiere

$$\mathcal{U}_{X^2}(0) = \left\{ Z_2 \in \mathcal{V}_{g_0} \mid \|Z_2\|_g \leq C_2 g^{\sigma_2} \ \ \forall g \in [0, g_2] \right\} \tag{2.76}$$

$$\mathcal{U}_{X^2}(Z_1) = Z_1 + \mathcal{U}_{X^2}(0)\,. \tag{2.77}$$

Es heißt nun eine Menge $\mathcal{U}_{X^2}(Z_1)$ Umgebung eines approximierten Fixpunktes, oder einfach approximierte Fixpunktumgebung, genau dann, wenn $X^2 \in \mathcal{I}$ so gewählt ist, daß

$$|Z_2(\phi, g)| \leq \frac{1}{2} Z_1(\phi, g) \ \ \forall (\phi, g) \in \mathcal{P}_{g_2} \tag{2.78}$$

Die Eigenschaft (2.78) gewährt, daß alle Wirkungen einer approximierten Fixpunktumgebung positiv sind.[32] Ferner sind oben definierte Mengen konvex und vollständig bzw. der Norm $\|\|\cdot\|\|_{g_2}$. Die letzte Eigenschaft wollen wir hier explizit zeigen:

Es sei $Z_n \in \mathcal{U}_{X^2}(0)$ eine Cauchy-Folge mit $Z = \lim Z_n$, d.h. für alle $\epsilon > 0$ existiert ein N, so daß $\|\|Z - Z_n\|\|_{g_2} < \epsilon$ für $n > N$. Es folgt direkt, daß für alle $(\phi, g) \in \mathcal{P}_{g_2}$ die Ungleichung $|Z(\phi, g)| < |Z_n(\phi, g)| + \epsilon$ gilt. Nehmen wir nun an, Z liegt nicht in $\mathcal{U}_{X^2}(0)$, so existiert ein (ϕ, g)-Tupel, so daß $Z(\phi, g) > \frac{1}{2} Z_1(\phi, g)$. Es existiert ein ϵ, mit dem auch

$$Z_n(\phi, g) > Z(\phi, g) - \epsilon > \frac{1}{2} Z_1(\phi, g)$$

gilt, was einen Widerspruch darstellt.

Im folgenden zeigen wir zwei Lemmata (inkl. eines Korollars), welche wir zum Beweis des darauffolgenden Satzes benötigen.

[31] Da wir den Fixpunkt durch eine unendliche Iteration der erweiterten RGT generieren werden, und $\mathcal{T} \times \delta^\star$ die Positivität erhält, müssen wir schon mit einer physikalischen Approximante starten.

[32] Statt $\frac{1}{2}$ hätten wir auch jede andere Zahl aus $(0, 1)$ benutzen können.

Lemma 2.5.5

$$\begin{aligned} & X^1, X^2 \in \mathcal{I} \quad mit \quad \sigma_2 > \frac{D}{4-D} \\ & Z_1 \in \mathcal{V}_{g_0} \quad mit \quad \|Z_1\|_g \leq e^{C_1 g^{\sigma_1}} \ \ \forall g \in [0, g_1] \\ \Rightarrow \exists \tilde{g} \in (0, g_0] \ \ & \forall Z_2 \in \mathcal{U}_{X^2}(0) : \|\mathcal{R}_{Z_1}(Z_2)\|_g \leq \frac{C_2}{2} g^{\sigma_2} \ \ \forall g \in [0, \tilde{g}] \end{aligned} \tag{2.79}$$

Beweis:

1. $\exists \tilde{g_1} \in (0, \min\{g_0, g_1\}] : e^{C_1 g^{\sigma_1}} \leq \tilde{C_1} := 1 + \frac{1}{2(\alpha-1)} \ \ \forall g \in [0, \tilde{g_1}]$
2. $\exists \tilde{g_2} \in (0, g_0] : C_2 (\delta g)^{\sigma_2} \leq \frac{1}{2(\alpha-1)} \ \ \forall g \in [0, \tilde{g_2}]$
3. $\tilde{g} := \min\{\tilde{g_1}, \tilde{g_2}\}$

Sei nun $Z_2 \in \mathcal{U}_{X^2}(0)$ und $(\phi, g) \in \mathcal{P}_{\tilde{g}}$:

$$\begin{aligned} & |\mathcal{R}_{Z_1}(Z_2)(\phi, g)| \\ = \ & \left| \mathcal{R} \times \delta^\star (Z_1 + sZ_2)(\phi, g) \Big|_0^1 \right| \\ \overset{(a)}{=} \ & \left| \int_0^1 ds \frac{\partial}{\partial s} \mathcal{R} \times \delta^\star (Z_1 + sZ_2)(\phi, g) \right| \\ \leq \ & \alpha \int_0^1 ds \int d\mu_\gamma(\zeta) \left\{ (|Z_1| + s|Z_2|)^{\alpha-1} |Z_2| \right\} (\beta\phi + \zeta, \delta g) \\ \overset{(b)}{\leq} \ & \alpha \int_0^1 ds \int d\mu_\gamma(\zeta) \left(\tilde{C_1} + s C_2 (\delta g)^{\sigma_2} \right)^{\alpha-1} C_2 (\delta g)^{\sigma_2} Z_{QU}^\alpha (\beta\phi + \zeta, \delta g) \\ \leq \ & \left\{ \sup_{s \in [0,1]} \alpha \left(\tilde{C_1} + s C_2 (\delta g)^{\sigma_2} \right)^{\alpha-1} C_2 (\delta g)^{\sigma_2} \right\} \mathcal{R} \times \delta^\star (Z_{QU})(\phi, g) \\ = \ & \alpha \left(\tilde{C_1} + C_2 (\delta g)^{\sigma_2} \right)^{\alpha-1} C_2 (\delta g)^{\sigma_2} Z_{QU}(\phi, g) \\ \overset{(c)}{\leq} \ & L^{D-(4-D)\sigma_2} e C_2 g^{\sigma_2} Z_{QU}(\phi, g) \\ \overset{(d)}{\leq} \ & \frac{1}{2} C_2 g^{\sigma_2} Z_{QU}(\phi, g) \ . \end{aligned}$$

Die partielle Ableitung des Integranden nach s (a) ist wohldefiniert, da er eine Verkettung stetig differenzierbarer Funktionen in s darstellt.[33] In der

[33] Vertauschung von Integration und Differentiation ist aufgrund gleichmäßiger Stetigkeit gewährleistet.

Umformung (b) durfte 1. wegen $\delta g \leq g$ benutzt werden. In (c) fließen 2. und $\forall x \in \mathbb{R}^+ : \left(1+\frac{1}{x}\right)^x \leq e$ ein. Damit (d) gilt, müssen wir

$$L \geq \exp\left(\frac{1+\ln 2}{(4-D)\sigma_2 - D}\right) \tag{2.80}$$

wählen.[34]

□

Es stellt sich die Frage nach dem maximalen Wert von $\tilde{g}$. Unabhängig von g_0 und g_1 stellen die Ungleichungen 1. und 2. natürliche Schranken dar und wir erhalten

$$g_{nat} = \min\left\{\left(\frac{\ln\left(1+\frac{1}{2(\alpha-1)}\right)}{C_1}\right)^{\frac{1}{\sigma_1}}, \frac{1}{\delta\left(2C_2(\alpha-1)\right)^{\frac{1}{\sigma_2}}}\right\} . \tag{2.81}$$

Da die Wahl von $\tilde{g}$ im obigen Lemma unabhängig von g_2 war, kann man sofort ein g_2 wählen, das die Abschätzungen von $\tilde{g}$ erfüllt. Wir erhalten somit folgenden wichtigen

Korollar 2.5.6
Es seien

$$X^1 \in \mathcal{I}, \quad Z_1 \in \mathcal{V}_{g_0} \quad mit \quad \|Z_1\|_g \leq e^{C_1 g^{\sigma_1}} \ \ \forall g \in [0, g_1] \tag{2.82}$$

$$\Rightarrow \exists X^2 \in \mathcal{I} \ \ \forall Z_2 \in \mathcal{U}_{X^2}(0) : \quad \|\mathcal{R}_{Z_1}(Z_2)\|_g \leq \frac{C_2}{2} g^{\sigma_2} \ \ \forall g \in [0, g_2] . \tag{2.83}$$

Hierbei sind $C_2 \in \mathbb{R}^+$ und $\sigma_2 \in \left(\frac{D}{4-D}, \infty\right)$ frei wählbar.

Lemma 2.5.7
Es seien

$$X^\triangle \in \mathcal{I}, \quad Z_1 \in \mathcal{V}_{g_0} \quad mit \quad \|\triangle(Z_1)\|_g \leq C_\triangle g^{\sigma_\triangle} \ \ \forall g \in [0, g_\triangle] \tag{2.84}$$

und

$$\tilde{C}_\triangle, \tilde{\sigma}_\triangle \in \mathbb{R}^+ \quad mit \quad \tilde{\sigma}_\triangle < \sigma_\triangle \tag{2.85}$$

$$\Rightarrow \exists \tilde{g}_\triangle \in (0, g_0] : \quad \|\triangle(Z_1)\|_g \leq \tilde{C}_\triangle g^{\tilde{\sigma}_\triangle} \ \ \forall g \in [0, \tilde{g}_\triangle] . \tag{2.86}$$

[34] Setzen wir $L > \exp\left(\frac{1}{(4-D)\sigma_2 - D}\right)$ voraus, so erhalten wir in Schritt (d) des Beweises statt $\frac{1}{2}$ einen Faktor $C(L) < 1$.

Beweis: Wähle $\tilde{g}_\triangle = \min\left\{\left(\frac{\tilde{C}_\triangle}{C_\triangle}\right)^{\frac{1}{\sigma_\triangle - \tilde{\sigma}_\triangle}}, g_\triangle\right\}$.

Korollar 2.5.6 und Lemma 2.5.7 bilden die Basis der Aussage, daß um einen approximierten Fixpunkt eine konvexe Menge konstruiert werden kann, auf der $\mathcal{R} \times \delta^\star$ selbstabbildend ist. Diese, für die spätere Anwendung des Banachschen Fixpunktsatzes benötigte, erste wichtige Eigenschaft protokolliert folgender

Satz 2.5.8
$Z_1 \in \mathcal{V}_{g_0}$ sei ein approximierter Fixpunkt $\Rightarrow$

$$\exists X^2 \in \mathcal{I} \quad mit \quad \sigma_2 < \sigma_\triangle : \quad \mathcal{R} \times \delta^\star : \mathcal{U}_{X^2}(Z_1) \to \mathcal{U}_{X^2}(Z_1) \tag{2.87}$$

Beweis: Es ist zu zeigen : $\exists X^2 \in \mathcal{I}\ \forall Z_2 \in \mathcal{U}_{X^2}(0)$:

$$\|\triangle(Z_1) + \mathcal{R}_{Z_1}(Z_2)\|_g \leq C_2 g^{\sigma_2}\ \ \forall g[0, g_2]$$

Nach Korollar 2.5.6 $\exists X^2 \in \mathcal{I}$ mit $\sigma_2 \in \left(\frac{D}{4-D}, \sigma_\triangle\right)$, so daß

$$\forall Z_2 \in \mathcal{U}_{X^2}(0) : \quad \|\mathcal{R}_{Z_1}(Z_2)\|_g \leq \frac{C_2}{2} g^{\sigma_2}\ \ \forall g \in [0, g_2]\ .$$

OBdA existiert nach Lemma 2.5.7 eine Transformation $X^\triangle \to X^2$, so daß

$$\|\triangle(Z_1)\|_g \leq \frac{C_2}{2} g^{\sigma_2}\ \ \forall g \in [0, g_2]\ .$$

□

Und auch die letzte benötigte Eigenschaft von $\mathcal{R} \times \delta^\star$ in einem

Satz 2.5.9
$Z_1 \in \mathcal{V}_{g_0}$ sei ein approximierter Fixpunkt $\Rightarrow$

$\exists X^2 \in \mathcal{I}$ so daß $\mathcal{R} \times \delta^\star : \mathcal{U}_{X^2}(Z_1) \to \mathcal{U}_{X^2}(Z_1)$ kontrahierend ist.

Beweis: Es ist zu zeigen:

$$\exists X^2 \in \mathcal{I} \quad \exists \lambda \in (0,1) \quad \forall Z = Z_1 + Z_2,\ Z' = Z_1 + Z_2' \in \mathcal{U}_{X^2}(Z_1) :$$

$$\|\|\mathcal{R} \times \delta^\star(Z) - \mathcal{R} \times \delta^\star(Z')\|\|_{g_0} \leq \lambda \|\|Z - Z'\|\|_{g_0}$$

Wähle X^2 wie in Satz 2.5.8

$$\begin{aligned}
& \left\| \left| \mathcal{R} \times \delta^\star(Z_1 + Z_2) - \mathcal{R} \times \delta^\star(Z_1 + Z_2') \right| \right\|_{g_0} \\
= \; & \left\| \left| \mathcal{R}_{Z_1}(Z_2) - \mathcal{R}_{Z_1}(Z_2') \right| \right\|_{g_0} \\
\overset{(1)}{=} \; & \left\| \left| \int_0^1 ds \frac{\partial}{\partial s} \mathcal{R}_{Z_1}(Z_2' - s(Z_2 - Z_2')) \right| \right\|_{g_0} \\
\leq \; & \int_0^1 ds \left\| \left| \frac{\partial}{\partial \epsilon} \mathcal{R}_{Z_1}(\underbrace{Z_2' + s(Z_2 - Z_2')}_{\nu}) + \epsilon(\underbrace{Z_2 - Z_2'}_{\omega}) \right|_{\epsilon=0} \right| \right\|_{g_0} \\
= \; & (\star)
\end{aligned}$$

Der Term in (1) ist stetig partiell nach s differenzierbar, da $\mathcal{U}_{X^2}(Z_1)$ konvex ist. Diese Eigenschaft bewahrt uns auch darauffolgend vor einem evtl. Verlassen des Definitionsbereiches von $\mathcal{R}_{Z_1}$, denn das Integral ist in diesem Fall als uneigentlich zu betrachten. Folglich ist $s \in (0,1)$ und es gilt:

$$\forall s \in (0,1) \;\; \exists \epsilon(s) > 0 \;\; \forall |\epsilon| < \epsilon(s) : \quad s + \epsilon \in [0,1]$$

Es folgt nun $\forall (\phi, g) \in \mathcal{P}_{g_0}$:

$$\begin{aligned}
& \left| \frac{\partial}{\partial \epsilon} \mathcal{R}_{Z_1}(\nu + \epsilon\omega)(\phi, g) \Big|_{\epsilon=0} \right| \\
= \; & \left| \frac{\partial}{\partial \epsilon} \int_0^1 d\tilde{s} \int d\mu_\gamma(\zeta) \alpha \left\{ (Z_1 + \tilde{s}(\nu + \epsilon\omega))^{\alpha-1} (\nu + \epsilon\omega) \right\} (\beta\phi + \zeta, \delta g) \Big|_{\epsilon=0} \right| \\
= \; & \left| \int_0^1 d\tilde{s} \int d\mu_\gamma(\zeta) \alpha \left\{ \omega(Z_1 + \tilde{s}\nu)^{\alpha-2} (Z_1 + \alpha\tilde{s}\nu) \right\} (\beta\phi + \zeta, \delta g) \right| \\
= \; & \left| \int d\mu_\gamma(\zeta) \alpha \left\{ \omega(Z_1 + \nu)^{\alpha-1} \right\} (\beta\phi + \zeta, \delta g) \right| \\
\overset{2.5.5}{\leq} \; & \frac{1}{2} |||\omega|||_{g_0} Z_{QU}(\phi, g)
\end{aligned}$$

Im letzten Schritt wurde darauf zurückgegriffen, daß Z_1 ein approximierter Fixpunkt ist, $\nu \in \mathcal{U}_{X^2}$ und $|\omega(\phi, g)| \leq |||\omega|||_{g_0} Z_{QU}(\phi, g) \;\; \forall (\phi, g) \in \mathcal{P}_{g_0}$. Die Rechnung läuft analog der aus Lemma 2.5.5.

Jetzt ist es uns vergönnt, die anfangs begonnene Rechnung fortzuführen. Da $\frac{1}{2}\|\omega\|_\infty Z_{QU}(\phi, g)$ unabhängig von s ist, erhalten wir $(\star) \leq \frac{1}{2}|||\omega|||_{g_0}$. Es ist also hier $\lambda = \frac{1}{2}$.

□

Mittels der Sätze 2.5.8, 2.5.9 und der Vollständigkeit der approximierten Fixpunktumgebung bezüglich der Norm-induzierten Metrik gelingt es uns nun, den fundamentalen Satz über die Existenz von Fixpunkten der Transformation $\mathcal{R} \times \delta^\star$ zu formulieren.

Satz 2.5.10 (Fixpunktsatz)
Sei $Z_1 \in \mathcal{V}_{g_0}$ ein approximierter Fixpunkt. Dann existiert ein $X^2 \in \mathcal{I}$, so daß $\mathcal{R} \times \delta^\star : \mathcal{U}_{X^2}(Z_1) \to \mathcal{U}_{X^2}(Z_1)$ genau einen Fixpunkt besitzt.

Beweis: Anwendung des Banachschen Fixpunktsatzes für metrische Räume, z.B. [Sma80].

Anmerkungen: Der Banachsche Fixpunktsatz liefert auch ein Verfahren zur Konstruktion des Fixpunktes, indem wir auf einen beliebigen Punkt der approximativen Fixpunktumgebung $\mathcal{R} \times \delta^\star$ iterativ anwenden, bis das Bild der Abbildung sich stabilisiert.

Am wichtigsten bei dem hier vorgestellten Verfahren ist die Wahl eines guten approximierten Fixpunktes, d.h. wir benötigen eine Abschätzung $\|\triangle(\cdot)\|_g \leq C_\triangle g^{\sigma_\triangle}$ mit großem $\sigma_\triangle$. Dieser Exponent wird jedoch nicht nur durch unsere Qualitätsansprüche bestimmt. Entscheidend ist die folgende Ungleichung, die für die Konstruktion von $\mathcal{R} \times \delta^\star$ gelten muß:

$$\frac{D}{4-D} < \sigma_\triangle \tag{2.88}$$

2.6 Approximierte Fixpunkte

Im letzten Paragraphen wurde gezeigt, daß man zur iterativen Berechnung unseres RG-Fixpunktes eine Starttrajektorie benötigt, welche den Bedingungen von Definition 2.5.3 genügt. Um die Konstruktion einer solchen Anfangskurve wollen wir uns in diesem Kapitel bemühen.

2.6.1 Interpolationsformeln

Zuvor wollen wir jedoch einige mathematische Hilfsmittel bereitstellen, die uns den Umgang mit der RGT erleichtern.

Satz 2.6.1
Es sei $t \in [0,1]$ und $\mathcal{V}$ ein Raum von analytischen Funktionen über $\mathbb{R}$, so daß

$$F_t : \mathcal{V} \to \mathcal{V} \qquad F_t(Z)(\phi) = \int d\mu_{t\gamma}(\zeta) Z(\phi + \zeta) \;\; \forall \phi \in \mathbb{R} \tag{2.89}$$

wohldefiniert ist. Dann ist die Abbildungsschar F stetig differenzierbar in ihrem Parameter t, und es gilt die Differentialgleichung

$$\left(\frac{\partial}{\partial t} - \frac{\gamma}{2} \frac{\partial^2}{\partial \phi^2} \right) F_t(Z)(\phi) = 0 \; . \tag{2.90}$$

Beweis: F_t ist in $(0,1]$ stetig partiell differenzierbar. Es gilt:

$$\begin{aligned}
\frac{\partial}{\partial t} F_t(Z)(\phi) &= \int \frac{\partial}{\partial t} d\mu_{t\gamma}(\zeta) Z(\phi + \zeta) \\
&= \int \frac{1}{2t} \left(\frac{\zeta^2}{t\gamma} - 1 \right) d\mu_{t\gamma}(\zeta) Z(\phi + \zeta) \\
&= \frac{\gamma}{2} \int \frac{\partial^2}{\partial \zeta^2} d\mu_{t\gamma}(\zeta) Z(\phi + \zeta) \\
&\overset{(*)}{=} \frac{\gamma}{2} \int d\mu_{t\gamma}(\zeta) \frac{\partial^2}{\partial \zeta^2} Z(\phi + \zeta) \\
&= \frac{\gamma}{2} \frac{\partial^2}{\partial \phi^2} F_t(Z)(\phi)
\end{aligned}$$

Die Umformung $(*)$ entspricht einer zweimaligen partiellen Integration, in der

$$\lim_{|\zeta| \to \infty} e^{-\frac{\zeta^2}{2t\gamma}} \frac{\partial}{\partial \zeta} Z(\phi + \zeta) = \lim_{|\zeta| \to \infty} Z(\phi + \zeta) \frac{\partial}{\partial \zeta} e^{-\frac{\zeta^2}{2t\gamma}} = 0$$

benutzt wurde. Um die Ableitung in $t = 0$ zu berechnen, bemühen wir deren Definition. Wir wollen noch bemerken, daß $F_0 = id$, da $d\mu_\gamma(\zeta) \overset{\gamma \to 0}{\longrightarrow} \delta(\zeta) d\zeta$. Daraus folgt für beliebige $Z \in \mathcal{V}$ und $\phi \in \mathbb{R}$:

$$\begin{aligned}
\lim_{t \to 0} \frac{1}{t} \left((F_t(Z)(\phi) - F_0(Z)(\phi) \right) &= \lim_{t \to 0} \frac{1}{t} \int d\mu_{t\gamma}(\zeta) \sum_{n=1}^{\infty} \frac{1}{n!} \frac{\partial^n}{\partial \phi^n} Z(\phi) \zeta^n \\
&= \lim_{t \to 0} \sum_{n=1}^{\infty} \frac{\gamma^n}{2^n n!} \frac{\partial^{2n}}{\partial \phi^{2n}} Z(\phi) t^{n-1} \\
&= \frac{\gamma}{2} \frac{\partial^2}{\partial \phi^2} Z(\phi)
\end{aligned}$$

Die Stetigkeit der Ableitung ist offensichtlich.

Der soeben bewiesene Satz 2.6.1 liefert die Grundlage für eine weitere Abbildung, die mittels des Parameters t zwischen der RGT $\mathcal{T} \times \delta^\star$ und der „Identität“ interpoliert.

Lemma 2.6.2
Definieren wir für $t \in [0,1]$

$$\mathcal{T}_t(V)(\phi, g) = -\log \int d\mu_{(1-t)\gamma}(\zeta) e^{-\alpha V(\beta\phi+\zeta, \delta g)} \,, \tag{2.91}$$

so ist diese Abbildungsschar stetig differenzierbar in ihrem Parameter t, und es gilt die Differentialgleichung[35]

$$\frac{\partial}{\partial t}\mathcal{T}_t(V)(\phi, g) = \frac{\gamma}{2\beta^2}\left\{\left(\frac{\partial}{\partial\phi}\mathcal{T}_t(V)(\phi, g)\right)^2 - \frac{\partial^2}{\partial\phi^2}\mathcal{T}_t(V)(\phi, g)\right\} \,. \tag{2.92}$$

Beweis: Wir haben die Angabe eines Definitions- und Wertebereiches bewußt ausgelassen, da wir die Eigenschaft der Differenzierbarkeit in $\mathcal{V}$ bzw $\mathcal{W}$ nicht involviert haben.[36] Man beachte aber, daß für eine Kovarianz γ RGT geeignete Wirkung (Potential) auch für eine RGT mit Kovarianz $\gamma' < \gamma$ geeignet ist. Die Wohldefiniertheit von $\mathcal{T} = \mathcal{T}_0$ überträgt sich also auf $\mathcal{T}_t$. Die stetige Differenzierbarkeit in t folgt aus 2.6.1. Die Differentialgleichung beweist man durch explizites Ableiten. Um dabei die Ergebnisse aus dem zuvor bewiesenen Satz benutzen zu können, schreiben wir $\mathcal{T}_t$ durch die Substitution $\frac{\zeta}{\beta} \to \zeta$ als

$$\mathcal{T}_t(V)(\phi, g) = -\log \int d\mu_{(1-t)\gamma\beta^{-2}}(\zeta) e^{-\alpha V(\beta(\phi+\zeta), \delta g)}$$

□

2.6.2 Baumgraphen

Betrachten wir die formale Störungsreihe

$$V^\infty(\phi, g) = \sum_{n=0}^{\infty} \sum_{r=\max\{1,n-1\}}^{\infty} V_{2n,r} g^r \phi^{2n} \,, \tag{2.93}$$

[35] Diese gilt natürlich nur für solche Potentiale, die zweimal stetig differenzierbar sind - und nur solche werden wir betrachten.

[36] Um die Vollständigkeit des Raumes $\mathcal{V}$ zu wahren, hätten wir eine Supremumsnorm, die auch die Ableitungen mit einschließt, benutzen müssen. Da wir die Interpolationsformel allerdings nur auf ganz bestimmte Potentiale, nämlich solche, die approximierte Fixpunkte generieren und die Differentiationseigenschaft aufweisen, anwenden, können wir die Definition von $\mathcal{V}$ bzw. $\mathcal{W}$ so allgemein halten, wie sie war.

deren Koeffizienten so bestimmt sind, daß V^∞ die Eigenschaften einer ϕ^4-Trajektorie bezüglich $\delta(g) = \delta g$ besitzt,[37] so erkennen wir, daß für $g \ll 1$ das Verhalten eines Feldes der Ordnung $2n$ primär durch den g-Summanden in kleinster Ordnung beschrieben wird. In diesem Abschnitt berechnen wir die korrespondierenden Leitkoeffizienten $V_{2n,\max\{n-1,1\}}$, die für $n \geq 2$ auch Baumgraphen genannt werden, indem wir aus dem mittels 2.6.2 interpolierten Potential (2.93) Differentialgleichungen ableiten und lösen.

Wir beginnen mit

$$\mathcal{T}_t(V^\infty)(\phi, g) = \sum_{n=0}^{\infty} \sum_{r=\max\{1,n-1\}}^{\infty} V_{2n,r}(t) g^r \phi^{2n} \; . \tag{2.94}$$

Da der Parameter t nur in der Kovarianz der Integraltransformation auftaucht, ist seine Wirkung durch die Gleichung (2.27) beschrieben.[38] Laut (2.35) sind die ϕ-abhängigen Koeffizienten zu gegebener Ordnung g^r von der Ordnung $r+1$. Folglich ist das interpolierte Potential forminvariant. Der nun t-abhängige Koeffizient $V_{2n,r}$ ist ein Polynom in t, da die Kovarianz polynomial in den normalgeordneten Monomen auftaucht. Es sei noch bemerkt, daß $\mathcal{T}_t$ keine Terme in g^0 generieren kann, da

$$\begin{aligned} & -\log \int d\mu_{\gamma(t)}(\zeta) e^{-V^\infty(\phi+\zeta,g)} = -\log \int d\mu_{\gamma(t)}(\zeta)\{1 + O(g)\} \\ = \; & -\log\{1 + O(g)\} = O(g) \; . \end{aligned}$$

Wir folgern folgenden

Satz 2.6.3 (Differentialgleichung der Baumgraphen)
Es seien $g_0 \in \mathbb{R}^+$, $t \in [0,1]$ und die formalen Potenzreihen

$$V^\infty, \mathcal{T}_t(V^\infty) : \mathcal{P}_{g_0} \to \mathbb{R}$$

definiert wie in (2.93) bzw. (2.94). Dann gilt

$$\dot{V}_{4,1}(t) = 0, \quad \dot{V}_{2,1}(t) = -6\frac{\gamma}{\beta^2} V_{4,1}(t), \quad \dot{V}_{0,1}(t) = -\frac{\gamma}{\beta^2} V_{2,1}(t) \; . \tag{2.95}$$

Ferner erhält man für $n \geq 3$ die Differentialgleichungen

$$\dot{V}_{2n,n-1}(t) = \frac{2\gamma}{\beta^2} \sum_{m=2}^{n-1} m(n+1-m) V_{2m,m-1}(t) V_{2(n+1-m),n-m}(t) \; . \tag{2.96}$$

[37] Diese Reihe unterscheidet sich von der in Kapitel 2.3 bestimmten Reihe nur dadurch, daß sie nicht in normalgeordneten Monomen organisiert ist.

[38] Man muß nur g durch δg und γ durch $(1-t)\gamma$ ersetzen.

Beweis: Um das Summieren in den folgenden Rechnungen zu vereinfachen, benutzen wir die Schreibweise

$$\mathcal{T}_t(V^\infty)(\phi, g) = \sum_{n=0}^{\infty} \sum_{r=n-1}^{\infty} V_{2n,r}(t) g^r \phi^{2n}$$

mit $V_{0,-1}(t) = V_{0,0}(t) = V_{2,0}(t) = 0$. Damit ergeben sich die Ableitungen

$$\begin{aligned} \frac{\partial}{\partial t}\mathcal{T}_t(V^\infty)(\phi, g) &= \sum_{n=0}^{\infty} \sum_{r=n-1}^{\infty} \dot{V}_{2n,r}(t) g^r \phi^{2n} \\ \frac{\partial^2}{\partial \phi^2}\mathcal{T}_t(V^\infty)(\phi, g) &= \sum_{n=0}^{\infty} \sum_{r=n}^{\infty} (2n+1)(2n+2) V_{2(n+1),r}(t) g^r \phi^{2n} \end{aligned}$$

und

$$\begin{aligned} & \left(\frac{\partial}{\partial \phi}\mathcal{T}_t(V^\infty)(\phi, g)\right)^2 \\ = & \sum_{n=0}^{\infty} \sum_{m=0}^{n} \sum_{r_1=m-1}^{\infty} \sum_{r_2=n-m-1}^{\infty} 4m(n-m) V_{2m,r_1}(t) V_{2(n-m),r_2}(t) g^{r_1+r_2} \phi^{2(n-1)} \\ \overset{(*)}{=} & \sum_{n=3}^{\infty} \sum_{m=2}^{n-1} \sum_{r_1=m-1}^{\infty} \sum_{r_2=n-m}^{\infty} 4m(n+1-m) V_{2m,r_1}(t) V_{2(n+1-m),r_2}(t) g^{r_1+r_2} \phi^{2n} \\ = & \sum_{n=3}^{\infty} \sum_{m=2}^{n-1} 4m(n+1-m) V_{2m,m-1}(t) V_{2(n+1-m),n-m}(t) g^{n-1} \phi^{2n} + O(g^n) \, . \end{aligned}$$

In $(*)$ haben wir ausgenutzt, daß ein Summand Null ist, falls r_1 oder r_2 nicht positiv sind. Wir dürfen uns somit auf $1 < m < n-1$ beschränken, und es folgt, daß für $n \leq 3$ alle Summanden verschwinden. Anschließend führt man noch eine Indexverschiebung der Form $n \to n-1$ durch.

Vergleichen wir nun mittels der Differentialgleichung aus Lemma 2.6.2 die Koeffizienten von $g^{\max\{1,n-1\}}\phi^{2n}$, erhalten wir die Behauptung.

□

Obwohl es auch möglich gewesen wäre, Differentialgleichungen für alle $V_{2n,r}(t)$ zu formulieren und durch das Lösen derselbigen die Koeffizienten zu bestimmen, benötigen wir im folgenden nur die Vorfaktoren der Form $V_{2n,\max\{1,n-1\}}(t)$. Wir weiten den Baumgraphenbegriff vom Beginn des Kapitels aus und erhalten eine

Definition 2.6.4 (Baumgraphen und Baumgraphenkoeffizienten)
Die Koeffizienten

$$b_{2n}(t) := V_{2n,\max\{1,n-1\}}(t) \tag{2.97}$$

heißen Baumgraphenkoeffizienten. Das interpolierte Potential der Form

$$V_B(t,\phi,g) := \sum_{n=0}^{\infty} b_{2n}(t) g^{\max\{1,n-1\}} \phi^{2n} \tag{2.98}$$

nennen wir dementsprechend Baumgraph oder Baumgraphenpotential, in manchen Fällen sprechen wir auch von der Baumgraphennäherung.

Das Baumgraphenpotential V_B genügt der Differentialgleichung, die sich aus der Interpolation 2.6.2 ergibt. Es hat jedoch einen Schönheitsfehler: $V_B(0,\cdot,\cdot)$ ist kein Fixpunkt der RGT $\mathcal{T}\times\delta^\star$. Wenn wir uns jedoch an unser eigentliches Vorhaben erinnern, das Konstruieren eines approximierten Fixpunktpotentials, so ist diese Eigenschaft der Baumgraphen belanglos, sofern sie den Bedingungen eines approximierten Fixpunktes genügen.

Wir erwähnen an dieser Stelle, daß der Baumgraph in seiner jetzigen Form natürlich nicht als Starttrajektorie für das vorgestellte Konstruktionsverfahren dienen kann, da man all seine Koeffizienten berechnen müßte, was ja formaler Störungstheorie entspräche. Ließe man das zu, könnten wir V^∞ sogleich perturbativ bestimmen und hätte die Fixpunktgleichung $\mathcal{T}\times\delta^\star(V^\infty) = V^\infty$ (formal) gelöst.[39]

Dennoch wollen wir im weiteren Verlauf dieses Kapitels mehr über die Baumgraphenkoeffizienten erfahren. Neben den in Satz 2.6.3 bestimmten Differentialgleichungen gehorchen sie noch einer weiteren Randbedingung:

$$T_1(V^\infty)(\phi,g) = \alpha V^\infty(\beta\phi,\delta g) \equiv \alpha T_0(V^\infty)(\beta\phi,\delta g) \tag{2.99}$$

Ein Koeffizientenvergleich liefert

$$\begin{array}{rcccl} b_{2n}(1) & = & (L^2)^{2-n} & b_{2n}(0) & \forall n \geq 2 \\ b_2(1) & = & \beta^{-2} & b_2(0) & \\ b_0(1) & = & \beta^{-4} & b_2(0) & . \end{array} \tag{2.100}$$

Da es sich bei V^∞ um eine ϕ^4-Trajektorie handelt, gilt ferner

$$b_4(0) = 1, \quad b_2(0) = -6, \quad b_0(0) = 3\ . \tag{2.101}$$

Mit Hilfe der Baumgraphendifferentialgleichung und den zuvor definierten Randbedingungen in (2.100) und (2.101) erarbeiten wir uns folgenden

[39] Wir werden jedoch eine explizite Formel für die Baumgraphenkoeffizienten herleiten.

Satz 2.6.5
Es sei V^∞ eine als formale Potenzreihe dargestellte ϕ^4-Trajektorie, deren Form Gleichung (2.93) genüge. Dann gilt für die Baumgraphen

$$b_0(t) = 3\left(\frac{\gamma}{\beta^2}\right)^2 t^2 + 6\frac{\gamma}{\beta^2}t + 3 \tag{2.102}$$

$$b_2(t) = -6\frac{\gamma}{\beta^2}t - 6 \tag{2.103}$$

$$b_4(t) = 1 \tag{2.104}$$

$$b_{2n,n\geq 3}(t) = B_{2n}\left\{\frac{\gamma(1-(1-L^{-2})t)}{\beta^2(1-L^{-2})}\right\}^{n-2} . \tag{2.105}$$

Hierbei ist die Folge $(B_{2n})_{n\geq 2}$ durch die rekursive Vorschrift

$$B_4 = 1 \tag{2.106}$$

$$B_{2n} = \frac{2}{2-n}\sum_{m=2}^{n-1} m(n+1-m)B_{2m}B_{2(n+1-m)} \tag{2.107}$$

gegeben.

Beweis: Die Herleitung der Baumgraphen $b_{2n}(t)$ mit $n \in \{0,1,2\}$ ist eine einfache Übung der Integrationstheorie. Beim Überprüfen der Randbedingungen beachte man, daß $\gamma = 1-\beta^2$. Durch explizites Einsetzten zeigt man, daß auch der Ansatz $b_{2n,n\geq 3}(t)$ den geforderten Randbedingungen genügt.

□

Ohne die Störungsreihe zu kennen oder berechnen zu müssen, ist es uns gelungen, eine Rekursionsformel für die $b_{2n}(t)$ zu finden. Da diesen, wie oben schon erwähnt, der Charakter von Leitkoeffizienten innewohnt, werden sie beim Führen von Abschätzungen vollständig ausreichen, um das Verhalten unserer Fixpunktkandidaten zu bestimmen. Es sei noch erwähnt, daß $b_{2n}(0)$ die exakten Baumgraphenkoeffizienten des Fixpunktpotentials V^∞ sind.

Abschließend noch ein

Lemma 2.6.6
Es sei $n \in \mathbb{N}_{\geq 2}$. Dann gilt:

$$B_{2n} = (-1)^n \left|B_{2n}\right| \tag{2.108}$$

Beweis über Induktion (nur Schritt):

$$B_{2n} = (-1)^n \frac{2}{n-2} \sum_{m=2}^{n-1} m(n+1-m) \left|B_{2m}\right| \left|B_{2(n+1-m)}\right| = (-1)^n \left|B_{2n}\right| \quad (2.109)$$

□

Aus (2.105) und Lemma 2.6.6 folgt, daß auch die t-abhängigen Baumgraphenkoeffizienten b_{2n} alternierend sind.

2.6.3 Explizite Formulierung der Baumgraphenkoeffizienten

Die rekursive Formulierung der Baumgraphenkoeffizienten (2.106) ist schön, aber nicht effizient. Aus diesem Grund machen wir uns in diesem Abschnitt auf die Suche nach einer expliziten Formel für die Leitkoeffizienten.

Wir betrachten die Hilfsfolge

$$c_m := \sqrt{2} m \left|B_{2m}\right| \quad (2.110)$$

und erhalten mit $a = 2^{\frac{3}{2}}$ und $b = 2^{\frac{1}{2}}$

$$c_2 = a \quad (2.111)$$

$$c_n = \frac{bn}{n-2} \sum_{m=2}^{n-1} c_m c_{n+1-m} \; . \quad (2.112)$$

Nun konstruieren wir die erzeugende Funktion

$$g(z) = \sum_{n=2}^{\infty} c_n z^n \; . \quad (2.113)$$

Unter der Annahme, daß g in $z = 0$ analytisch ist, erhalten wir mit Hilfe von (2.112)

$$z g'(z) - 2g(z) = b \left(\frac{g(z)^2}{z} \right)' z \; . \quad (2.114)$$

Diese nicht lineare Differentialgleichung besitzt die implizite Lösung (berechnet mit *MapleV*)

$$C z g(z) = (z + b g(z))^3 \quad (2.115)$$

Unter Verwendung der Anfangsbedingung (2.111) bestimmen wir die noch unbestimmte Konstante zu $C = \frac{1}{a}$. (2.115) erlaubt nun z.B. durch sukzessives Ableiten die Bestimmung der Koeffizienten c_n. Nach der Berechnung der ersten Ordnungen erahnen wir die Lösung

$$c_n = f_n a^{n-1} b^{n-2} \tag{2.116}$$

mit

$$\{f_n\}_{n\in\mathbb{N}_2} = \{1, 3, 12, 55, 273, 1428, 7752, 43263, 246675, \ldots\} \ . \tag{2.117}$$

Die bis zu einer gewissen Position berechnete Koeffizientenfolge $\{f_n\}$ finden wir[40] samt expliziter Formel in [Slo]. Wir erhalten

$$f_{n\geq 1} = \frac{1}{2n-1}\binom{3(n-1)}{n-1} \tag{2.118}$$

und beweisen nun für den Fall $a = b = 1$, daß (2.118) wirklich die Gleichungen (2.111) und (2.112) erfüllt. Den Fall $n = 2$ zeigt man durch Einsetzen, für die Rekursionsbeziehung betrachte man die implizite Gleichung

$$F(x) - F(x)^3 = x \ , \tag{2.119}$$

welche die Potenzreihe $F(x) = \sum_{n=1}^{\infty} f_n x^{2n-1}$ erfüllt.

Beweis: Die Gleichung (2.119) wird durch die Bürmann-Lagrangesche-Reihe $F(x) = \sum_{n=1}^{\infty} f_n x^n$ gelöst, deren Koeffizienten sich durch

$$f_n = \frac{1}{n}\mathrm{res}_0\left(\left(F - F^3\right)^{-n}\right) \tag{2.120}$$

bestimmen [HC64]. Bei der Residuenbestimmung ist die Laurentreihe in F gemeint. Mit Hilfe der geometrischen Reihe[41] erhalten wir

$$\begin{aligned}\left(\frac{1}{F-F^3}\right)^n &= \frac{1}{F^n}\frac{1}{(n-1)!}\frac{\partial^{n-1}}{\partial\left(F^2\right)^{n-1}}\frac{1}{1-F^2} \\ &= \frac{1}{(n-1)!}\sum_{k=n-1}^{\infty}\frac{k!}{(k-n+1)!}F^{2k-3n+2} \ . \end{aligned} \tag{2.121}$$

Für gerade n ist (2.121) residuenfrei. Ungerade Folgenglieder $n \to 2n-1$ lösen die Residuengleichung gemäß

$$2k - 3(2n-1) + 2 = -1 \Leftrightarrow k = 3(n-1) \ . \tag{2.122}$$

[40] nach langer Suche

[41] Da $F(0) = 0$,existiert ob der Stetigkeit ein $R > 0$, so daß $|F(x)| < 1$ für $|x| < R$.

Durch Einsetzen erhält man (2.118). □

(2.119) liefert nun nach einmaligem Differenzieren, Multiplikation mit F und nochmaliger Verwendung der Beziehung (2.119) die Gleichung

$$3xF(x)' - \left(F(x)^2\right)' = F(x) \;, \tag{2.123}$$

aus der die Rekursionsbeziehung der c_n folgt. Die Gültigkeit von (2.116) für beliebige $a, b \in \mathbb{R}$ folgt durch Einsetzen in (2.111), (2.112) und Ausnutzung der soeben gewonnenen Relation für $a = b = 1$. Für die Baumgraphenkoeffizienten erhalten wir

$$B_{2n} = (-1)^n \frac{2^{2n-3}}{n(2n-1)} \binom{3(n-1)}{n-1} \;. \tag{2.124}$$

Nachdem wir die f_n berechnet haben, können wir auch beweisen, daß der Ansatz einer konvergenten Potenzreihe g (2.113) gerechtfertigt war. Mittels des Quotientenkriteriums bestimmen wir den Konvergenzradius zu

$$\lim_{n\to\infty} \left| \frac{c_n}{c_{n+1}} \right| = \frac{4}{27}(ab)^{-1} = \frac{1}{27} > 0 \;. \tag{2.125}$$

Der Ansatz der erzeugenden Funktion g erweist sich letztendlich als überflüssig, da alle Resultate aus der Gleichung (2.119) ableitbar sind. Wir betrachten die Potenzreihe g jedoch als Experiment, das uns erste numerische und analytische Ideen schenkte, und deshalb einen berechtigten Platz in dieser Arbeit einnimmt.

2.6.4 Konvergenzgebiet der Baumgraphen

Abschließend diskutieren wir noch die Konvergenz der Baumgraphen V_B. Fassen wir sie als Potenzreihen in ϕ^2 auf, so ergibt sich für den kopplungsabhängigen Konvergenzradius mit Hilfe von (2.105) und (2.125)

$$R(t,g) = \begin{cases} \infty & g = 0 \\ \frac{1}{27} \frac{\beta^2(1-L^{-2})}{\gamma(1-(1-L^{-2})t)} |g|^{-1} & \text{sonst} \end{cases} \tag{2.126}$$

Dieses Resultat gilt auch für negative g. Dennoch erscheint es auf den ersten Blick sehr unbefriedigend, da für endliche Kopplungsparameter das (quadrierte) Feld beschränkt ist. Eine exakte Berechnung des Baumgraphenpotentials auf dem Rand des Konvergenzgebietes ergibt für $g > 0$:

$$V_B(t,\phi,g)\Big|_{\phi^2 = +R(t,g)} = \ldots + g^{-1} \left\{ \frac{\beta^2(1-L^{-2})}{\gamma(1-(1-L^{-2})t)} \right\}^2 \sum_{n=2}^{\infty} (-1)^n |B_{2n}| \left(\frac{1}{27}\right)^n \tag{2.127}$$

Die Punkte symbolisieren die für die Konvergenzbetrachtung unwichtigen ersten beiden Summanden. Die Konvergenz der Reihe (2.127) folgt aus dem Leibniz-Kriterium,[42] da

$$\left|\frac{B_{2(n+1)}}{B_{2n}}\right| = \frac{1}{6}\underbrace{\frac{3n-2}{n+1}}_{<3}\underbrace{\frac{3n-1}{2n+1}}_{<\frac{3}{2}} < 27\,, \tag{2.128}$$

und somit $(27)^{-n}|B_{2n}|$ streng monoton fallend ist. Unter Benutzung der Positivität folgt die Konvergenz gegen Null.

Für $\phi^2 = -R(t,g)$ erhalten wir (2.127) ohne den Faktor $(-1)^n$. Diese Reihe konvergiert nicht - ansonsten läge ein Widerspruch zum Konvergenzradius vor. Aufgrund der Entwicklung in ϕ^2 liegen auf der negativen reellen Achse nur imaginäre Feldvariablen ϕ, so daß wir unsere Betrachtungen auf $\mathbb{R}_0^+$ konzentrieren.

Für $\phi > \sqrt{R(t,g)}$ divergiert V_B. Möchte man mit einer Baumgraphenapproximante rechnen, die auch für Großfelder definiert ist, so muß man V_B in $\phi = \sqrt{R(t,g)}$ (n mal) stetig (differenzierbar) fortsetzen. Diesem Problem werden wir in dieser Arbeit jedoch nicht begegnen.

Unter der Annahme, daß das Konvergenzgebiet der perturbativen Trajektorie im Konvergenzbereich der Baumgraphen liegt, erhalten wir die Aussage, daß $V(\phi, g > 0)$ nicht für alle ϕ konvergiert. Dies ist direkt beweisbar, sofern ein ϕ^{2n}-Vertex die Gestalt $g^{\max(n-1,1)}(b_{2n}(0) + O(g))$ besitzt. Da die Reihe der Koeffizienten $\sum_{r\geq n} V_{2n,r} g^r$ jedoch höchstwahrscheinlich nicht konvergiert, dürfen wir sie nicht mit $O(g)$ identifizieren. Dennoch glauben wir, daß das Verhalten der Baumgraphenapproximante ein starkes Indiz für die Divergenz von V ist.

2.6.5 Das skalierende Potential

Wie generieren wir nun unseren approximierten Fixpunkt? Was liegt näher, als ihn von einem polynomialen Potential zu erzeugen, welches der Störungsreihe in der Ordnung s entspricht. Für kleine Kopplungskonstanten unterscheidet sich dieser Kandidat nur geringfügig von dem formal bestimmten perturbativen Potential. Dies liegt an der besonderen Struktur von V^∞, in der Felder der Ordnung $2n$ mit der Gewichtung $O(g^{\max\{1,n-1\}})$ einfließen.

Zuvor jedoch ein

[42] Es sei (a_n) eine monotone Nullfolge. Dann konvergiert die Reihe $\sum_n (-1)^n a_n$.

Lemma 2.6.7
Es seien $r, n \in \mathbb{N}_0$. Dann gilt:

$$\frac{1}{r!}\int_0^1 du(1-u)^r \frac{\partial^{r+1}}{\partial u^{r+1}} u^n = \begin{cases} 0 & n \leq r \\ 1 & n > r \end{cases} . \tag{2.129}$$

Beweis: Die Teilaussage für $n \leq r$ folgt aus $\frac{\partial^{r+1}}{\partial u^{r+1}} u^n = 0$. Den Part für $n > r$ zeigt man mit vollständiger Induktion über r und benutzt im Induktionsschritt partielle Integration.

□

Mittels obiger Integraltransformation gelingt es uns, einen Projektor zu konstruieren, der zu beliebig vorgegebenem $r \in \mathbb{N}_0$ das Polynom vom Grade r aus einer um den Nullpunkt entwickelten Potenzreihe entfernt. Dazu folgende

Definition 2.6.8
Es sei $r \in \mathbb{N}_0$ und

$$\mathcal{D} = \{f : U_f(0) \to \mathbb{R} \mid f \text{ analytisch in } 0\} .$$

Dann definiere den Projektor

$$\mathcal{P}^r : \mathcal{D} \to \mathcal{D} \qquad f(x) \mapsto \frac{1}{r!}\int_0^1 du(1-u)^r \frac{\partial^{r+1}}{\partial u^{r+1}} f(ux) \quad \forall x \in U_f(0) .$$

Obige Definition ist wohldefiniert, da Potenzreihen innerhalb ihres Konvergenzradius - und dort befinden wir uns während der Integration, da $|ux| < R(f)$ wegen $|u| \leq 1$ - gliedweise integriert und differenziert werden dürfen. Ferner ist die oben definierte Abbildung linear und es gilt $\mathcal{P}^r \circ \mathcal{P}^r = \mathcal{P}^r$.

Falls es sich beim Definitionsbereich $\mathcal{D}$ um Funktionen mehrerer Veränderlicher handelt, so wollen wir den Variablennamen, auf den der Projektor wirkt, als Index hinzufügen. Im Falle formaler Potenzreihen sei $\mathcal{P}^r$ ein formaler Projektor.

Mit Hilfe von $\mathcal{P}^r$ gewinnen wir nun auf einfachste Weise aus V^∞ ein polynomiales Potential.

Definition 2.6.9
Es sei $s \in 2\mathbb{N}_0 + 1$. Dann heißt

$$V^s(\phi, g) = (1 - \mathcal{P}_g^s)(V^\infty)(\phi, g) \tag{2.130}$$

skalierendes Potential in der Ordnung s.

Wir schränken uns bei dieser Definition sogleich auf die Potentiale ein, deren Großfeldverhalten die Wohldefiniertheit der Transformation $\mathcal{T} \times \delta^\star$ erhält. Für ungerade s sind die Baumgraphenkoeffizienten $b_{2(s+1)}(t)$ positiv (siehe Lemma 2.6.6), und folglich ist $e^{-V^s} \in \mathcal{W}_\infty$. Ferner werden wir V^s im weiteren Verlauf in der Form

$$V^s(\phi, g) = \sum_{n=0}^{s+1} g^{\max\{1,n-1\}} \lambda_{2n}(g) \phi^{2n} \tag{2.131}$$

notieren und bemerken noch, daß $b_{2n}(0) = \lambda_{2n}(0)$.

Eine grundlegende Eigenschaft des skalierenden Potentials ist die Erfüllung der Fixpunktgleichung für $\mathcal{T} \times \delta^\star$ bis zur Ordnung s in g. Man erkennt dies leicht, wenn man die Exponential- und Logarithmusfunktionen, die in der RGT auftauchen, als Reihen darstellt und beachtet, daß $V^\infty(\phi, g) = O(g)$.

$$\begin{aligned} \mathcal{T} \times \delta^\star(V^s)(\phi, g) &= \mathcal{T} \times \delta^\star \circ (1 - \mathcal{P}_g^s)(V^\infty)(\phi, g) \\ &= (1 - \mathcal{P}_g^s) \circ \mathcal{T} \times \delta^\star(V^\infty)(\phi, g) + O(g^{s+1}) \\ &= (1 - \mathcal{P}_g^s)(V^\infty)(\phi, g) + O(g^{s+1}) \\ &= V^s(\phi, g) + O(g^{s+1}) \end{aligned}$$

Wir vereinbaren noch, den Index g des Projektors in Zukunft nicht mehr anzugeben und formulieren einen

Satz 2.6.10
Es sei $s \in 2\mathbb{N}_0 + 1$ und V^s ein skalierendes Potential. Dann gilt:

$$(1 - \mathcal{P}^s) \circ \mathcal{T} \times \delta^\star(V^s) = V^s \tag{2.132}$$

2.6.6 Die Baumgraphenschranke

Bei V^s handelt es sich um ein Polynom in ϕ und g, das man erhält, wenn man aus der perturbativen Lösung der Fixpunktgleichung V^∞ den Part $O(g^{s+1})$ entfernt. Wir sprechen deshalb auch von einem trunkierten Potential, das in den einzelnen Ordnungen im Kopplungsparameter aus Summen trunkierter Erwartungswerte besteht. In diesem Paragraphen wollen wir nun zeigen, daß V^s durch $\left\| e^{-V^s} \right\|_g \leq e^{C_1 g^{\sigma_1}}$ abgeschätzt werden kann, und somit eine wichtige Voraussetzung erfüllt, um nach Definition 2.5.3 ein approximierter Fixpunkt zu sein. Die einzigen Größen, die für diesen *bound* bekannt sein müssen, sind die Baumgraphenkoeffizienten, welche wir in Kapitel 2.6.2 berechnet haben.

Wir werden im folgenden mit dem interpolierten skalierten Potential rechnen, das für $s \in 2\mathbb{N}+1$ als

$$V^s(t,\phi,g) = (1-\mathcal{P}^s)\,\mathcal{T}_t(V^\infty)(\phi,g) = \sum_{n=0}^{s+1} g^{\max\{1,n-1\}}\lambda_{2n}(g,t)\phi^{2n} \qquad (2.133)$$

definiert ist. Der Spezialfall $t=0$ liefert dann entsprechend Definition 2.6.9 das skalierende Potential. Das Polynom λ_{2n} erfüllt die Eigenschaft

$$\lambda_{2n}(g,t) = b_{2n}(t) + O(g)\ . \qquad (2.134)$$

Natürlich hängt auch $O(g)$ noch von t ab. Nun folgt eine rekursive Abschätzung des Potentials (2.133). Beginnend mit

$$\tilde{\lambda}^s_{2(s+1)}(g,t) := \lambda_{2(s+1)}(g,t), \qquad (2.135)$$

fahren wir für $n = s-1, s-3, \ldots, 2$ fort

$$\begin{aligned} & g^{n-1}\lambda_{2n}(g,t)\phi^{2n} + g^n\lambda_{2(n+1)}(g,t)\phi^{2(n+1)} + g^{n+1}\tilde{\lambda}^s_{2(n+2)}(g,t)\phi^{2(n+2)} \\ \geq\ & g^{n-1}\left\{\lambda_{2n}(g,t) - \frac{\lambda_{2(n+1)}(g,t)^2}{4\lambda_{2(n+2)}(g,t)}\right\}\phi^{2n} \\ =:\ & g^{n-1}\tilde{\lambda}^s_{2n}(g,t)\phi^{2n}\ , \end{aligned} \qquad (2.136)$$

um die Ungleichung

$$V^s(t,\phi,g) \geq g\lambda_0(g,t) + g\lambda_2(g,t)\phi^2 + g\tilde{\lambda}^s_4(g,t)\phi^4 \qquad (2.137)$$

zu erhalten. Jedes interpolierte skalierende Potential ungerader Ordnung besitzt also ein ϕ^4 Potential als untere Schranke. Diese Abschätzung ist allerdings nur sinnvoll, wenn wir zeigen können, daß $\tilde{\lambda}^s_4(g,t)$ positiv ist.[43] Die effektive ϕ^4-Kopplung $\tilde{\lambda}^s_4(g,t)$ ist ein Kettenbruch, der alle Koeffizienten höherer Feldordnungen in sich vereint. Es ergibt sich[44]

$$\tilde{\lambda}^s_4(g,t) = \lambda_4(g,t) - \frac{\lambda_6(g,t)^2|}{|4\lambda_8(g,t)} - \frac{\lambda_{10}(g,t)^2|}{|4\lambda_{12}(g,t)} - \ldots - \frac{\lambda_{2s}(g,t)^2|}{|4\lambda_{2(s+1)}(g,t)}\ . \qquad (2.138)$$

Wir können $\tilde{\lambda}^s_4$ zwar in einen „normalen“ Bruch umschreiben, doch ist es uns nicht möglich, die Funktionswerte oder singuläre Punkte zu berechnen, da wir die λ_{2n} nicht explizit bestimmt haben. Wir wissen aber, daß die λ_{2n} Polynome in g und t sind, folglich ist $\tilde{\lambda}^s_4$ eine rationale Funktion in (g,t).

[43] Ansonsten können wir nicht gegen $Z_{QU}(\phi,g)$ abschätzen.

[44] Wir verwenden hier die Darstellung von Kettenbrüchen gemäß [OL].

Wenn wir nun zeigen können, daß die effektive Kopplung für $g = 0$ durch eine positive Konstante $\tilde{C}$ nach unten beschränkt ist, so existieren $\tilde{g} > 0$ und $C > 0$ mit

$$\forall (g,t) \in [0,\tilde{g}] \times [0,1] \quad : \quad \tilde{\lambda}_4^s(g,t) > C \; . \tag{2.139}$$

Beweis: Da $\tilde{\lambda}_4^s$ als rationale Funktion nur endlich viele Null- und Polstellen besitzt, wählen wir $\overline{g} > 0$ so klein, daß das Rechteck $[0,\overline{g}] \times [0,1]$ null- und polstellenfrei ist. Es folgt Positivität. Die Existenz einer unteren Schranke leiten wir aus der gleichmäßigen Stetigkeit[45] ab, denn für alle $\epsilon > 0$ existiert ein Universal[46] $\delta > 0$, so daß für alle $(g_1,t_1),(g_2,t_2)$, die in Kreisen mit dem Durchmesser δ liegen[47], die Ungleichung $|\tilde{\lambda}_4^s(g_1,t_1) - \tilde{\lambda}_4^s(g_2,t_2)| < \epsilon$ erfüllt ist. Insbesondere folgt für $g_1 = 0$: $|\tilde{\lambda}_4^s(g_2,t_2)| > \tilde{C} - \epsilon$. Leider bietet die Argumentation über die gleichmäßige Stetigkeit keine quantitative Aussage über die maximale Kopplung $\tilde{g}$.

Es bleibt die Positivität für $g = 0$ zu zeigen. Mittels (2.134) und (2.105) erarbeiten wir

$$\begin{aligned} \tilde{\lambda}_4^s(0,t) &= b_4(t) - \frac{b_6(t)^2|}{|4b_8(t)} - \frac{b_{10}(t)^2|}{|4b_{12}(t)} - \ldots - \frac{b_{2s}(t)^2|}{|4b_{2(s+1)}(t)} \\ &= B_4 - \frac{B_6^2|}{|4B_8} - \frac{B_{10}^2|}{|4B_{12}} - \ldots - \frac{B_{2s}^2|}{|4B_{2(s+1)}} \end{aligned} \tag{2.140}$$

Man erkennt, daß die effektive ϕ^4-Wirkung $\tilde{\lambda}_4^s \equiv \tilde{\lambda}_4^s(0,t)$ unabhängig vom Interpolationsparameter ist. Die Berechnung der ersten 50 Schranken ist in Abbildung 2.2 dargestellt. Es stellt sich die Frage, ob diese Folge gegen eine positive reelle Zahl konvergiert. Diese Annahme wird durch die Grafik gestützt. In 199. Ordnung erhalten wir $\tilde{\lambda}_4^{199} \approx 0,7292155$ mit der absoluten Abweichung (zum Vorgänger) von $\approx 0,22\ 10^{-5}$ und dem relativen Fehler[48] $\approx 0,31\ 10^{-5}$. Allerdings ist das kein Indiz, denn auch „ganz viele kleine Dinge können etwas Großes bewirken“.[49] Für die Existenz der Baumschranke genügt es, die Relation $\tilde{\lambda}_4^s > 0$ für alle $s \in 2\mathbb{N}_0 + 1$ zu beweisen. Mit Hilfe der expliziten Formulierung der Baumgraphenkoeffizienten wird dies möglich. Aus (2.140) erhalten wir in Anlehnung an (2.136) die Rekursionsbeziehung

$$\tilde{\lambda}_{2(s+1)}^s = B_{2(s+1)} \tag{2.141}$$

[45] Jede stetige Funktion ist auf einem Kompaktum gleichmäßig stetig.

[46] unabhängig von (g,t)

[47] und natürlich auch im Rechteck $[0,\overline{g}] \times [0,1]$

[48] $r^s = 2\frac{\tilde{\lambda}_4^s - \tilde{\lambda}_4^{s-2}}{\tilde{\lambda}_4^s + \tilde{\lambda}_4^{s-2}}$

[49] Sinngemäßes Zitat von K. LANGMANN, welches er in der Mathematikvorlesung zur Veranschaulichung der Divergenz der harmonischen Reihe benutzte.

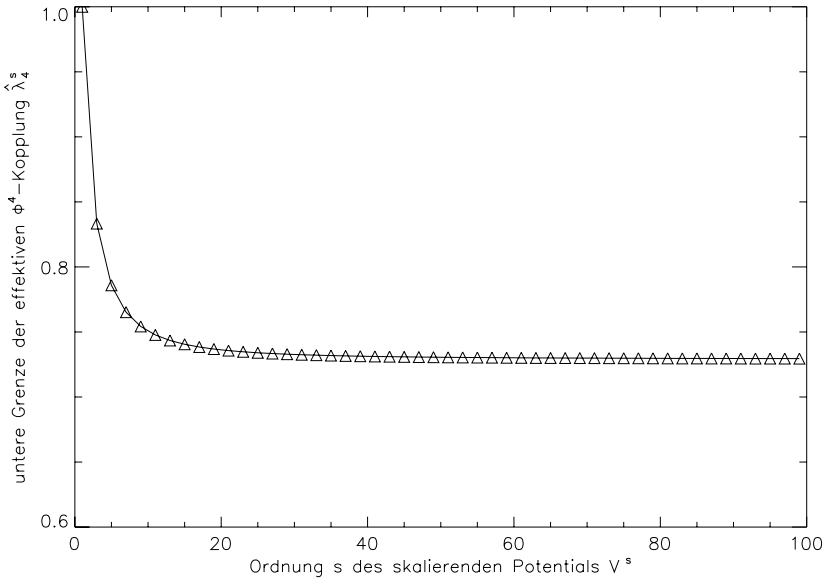


Abbildung 2.2: Diese Grafik unterstützt die Vermutung, daß die untere Grenze der effektiven ϕ^4-Kopplung $\tilde{\lambda}_4^s$ gegen eine positive, reelle Zahl konvergiert.

$$\tilde{\lambda}_{4n}^s = B_{4n} - \frac{B_{4n+2}^2}{4\tilde{\lambda}_{4(n+1)}^s} \quad n = 1, \ldots, \frac{s-1}{2} \; . \tag{2.142}$$

Zeigen wir nun induktiv für alle perturbativen Ordnungen $s \in 2\mathbb{N}_0 + 1$ die Ungleichung

$$0 < \tilde{\lambda}_{4n}^s \leq B_{4n} \quad n = 1, \ldots, \frac{s+1}{2}, \tag{2.143}$$

so sind wir fertig. Mit Hilfe des Lemmas 2.6.6 zeigt man leicht den Induktionsanfang ($n = \frac{s+1}{2}$) und die obere Schranke im Induktionsschritt. In der Abschätzung gegen Null nutzen wir ($n \geq 2$)

$$\begin{aligned} & 4\left|B_{2(2n)}\right|\left|B_{2(2n-2)}\right| - \left|B_{2(2n-1)}\right|^2 \\ = \; & \left|B_{2(2n-1)}\right|^2 \left\{ 4 \underbrace{\frac{2n-1}{2n}}_{\geq \frac{3}{4}} \underbrace{\frac{4n-3}{4n-1}}_{\geq \frac{5}{7}} \underbrace{\frac{4n-4}{4n-2}}_{\geq \frac{2}{3}} \underbrace{\frac{6n-3}{6n-6}}_{>1} \underbrace{\frac{6n-4}{6n-7}}_{>1} \underbrace{\frac{6n-5}{6n-8}}_{>1} - 1 \right\} \\ \geq \; & \frac{3}{7}\left|B_{2(2n-1)}\right|^2 > 0 \; . \end{aligned}$$

Aus (2.137) folgt nun, daß für alle s eine maximale Kopplung $g_s > 0$ und

eine Konstante $C_s > 0$ existieren, so daß für $g \in [0, g_s]$:

$$V^s(t, \phi, g) \geq g\lambda_0(g, t) + g\lambda_2(g, t)\phi^2 + C_s g\phi^4 \tag{2.144}$$

Was fehlt, ist die Abschätzung in der g-Norm. Die (Beträge der) stetigen Koeffizienten λ_0 und λ_2 seien auf $[0, g_s] \times [0, 1]$ durch C_0 und C_2 beschränkt. Definieren wir nun eine Funktion b durch

$$2\sqrt{C_s g}\, b(g) := \frac{b_{QU}(g)}{2} + C_2 g, \tag{2.145}$$

so gilt die Ungleichung

$$\begin{aligned} C_s g\phi^4 &= \left(\sqrt{C_s g}\phi^2 - b(g)\right)^2 + 2\sqrt{C_s g}\phi^2 b(g) - b(g)^2 \\ &\geq \frac{b_{QU}(g)}{2}\phi^2 + C_2 g\phi^2 - b(g)^2 \ . \end{aligned} \tag{2.146}$$

Aus (2.145) leiten wir mit Hilfe von[50] $b_{QU}(g) = O(g^\rho) \subsetneq O(g)$ die Beziehung $b(g) = O(g^{\frac{1}{2}})$ ab. Da auch $a_{QU}(g) = O(g^\rho) \subsetneq O(g)$ gilt, erhalten wir mit der Definition

$$\mathcal{T}_t^s = (1 - \mathcal{P}^s) \circ \mathcal{T}_t \tag{2.147}$$

die Repräsentation

$$e^{-\mathcal{T}_t^s(V^\infty)} \leq e^{C_0 g + b(g)^2 - a_{QU}(g)} Z_{QU}(\phi, g) = e^{O(g)} Z_{QU}(\phi, g) \ . \tag{2.148}$$

Mit $\sigma_1 \leq 1$ und eventueller Redefinition der maximalen Kopplung g_s folgt der

Satz 2.6.11
Es sei $s \in 2\mathbb{N}_0 + 1$. Dann existiert ein $X^1 = X^1(s) \in \mathcal{I}$, so daß für alle $g \in [0, g_1]$

$$\left\| e^{-\mathcal{T}_t^s(V^\infty)} \right\|_g \leq e^{C_1 g^{\sigma_1}} \tag{2.149}$$

Man setzt $t = 0$ und sieht, daß e^{-V^s} die zweite Eigenschaft eines approximierten Fixpunktes 2.5.3 erfüllt.

[50] Zur Notation: Wir fassen den Ausdruck $O(f(g))$ als eine Funktionenmenge auf, deren Elemente h die Eigenschaft innewohnt, für $g \to 0$ die Relation $\lim_{g\to 0} \frac{h(g)}{f(g)} < \infty$ zu erfüllen. $h = O(f(g))$ entspricht somit $h \in O(f(g))$.

2.6.7 Die Güte des skalierenden Potentials

Obiger Titel ist ein wenig irreführend, denn natürlich geht es in diesem Paragraphen um die Güte $\triangle$ der Funktion e^{-V^s}. Wir beginnen mit der Definition einer weiteren Interpolationsformel, die wir mit Hilfe von Lemma 2.6.2 konstruieren. Es seien $s \in \mathbb{N}$ und $t \in [0,1]$.

$$\mathcal{R}_t^s(V)(\phi,g) = \int d\mu_{\gamma t}(\zeta) e^{-\mathcal{T}_t^s(V)(\phi+\frac{\zeta}{\beta},g)} \tag{2.150}$$

Diese Abbildung ist für V^s mit $s \in 2\mathbb{N}_0 + 1$ wohldefiniert,[51] und es gelten

$$\mathcal{R}_0^s(V^s)(\phi,g) = e^{-\mathcal{T}_0^s(V^s)(\phi,g)} = e^{-(1-\mathcal{P}^s)\circ\mathcal{T}\times\delta^\star(V^s)(\phi,g)} = e^{-V^s(\phi,g)} \tag{2.151}$$

und

$$\begin{aligned}
\mathcal{R}_1^s(V^s)(\phi,g) &= \int d\mu_\gamma(\zeta) e^{-(1-\mathcal{P}^s)\circ\mathcal{T}_1(V^s)(\phi+\frac{\zeta}{\beta},g)} \\
&= \int d\mu_\gamma(\zeta) e^{-(1-\mathcal{P}^s)(\alpha V^s)(\beta\phi+\zeta,\delta g)} \\
&= \mathcal{R}\times\delta^\star(e^{-V^s})(\phi,g)\ .
\end{aligned} \tag{2.152}$$

Ferner wissen wir, daß $\mathcal{R}_t^s$ stetig differenzierbar in t ist, da es sich um eine Komposition von in t differenzierbaren Funktionen handelt (siehe hierzu auch Lemma 2.6.1). Eine Anwendung des Mittelwertsatzes bringt uns dann zu einer ersten Abschätzung der Güte $\triangle$.

$$\begin{aligned}
\left|\triangle\left(e^{-V^s}\right)(\phi,g)\right| &= \left|\mathcal{R}_1^s(V^s)(\phi,g) - \mathcal{R}_0^s(V^s)(\phi,g)\right| \\
&= \left|\frac{\partial}{\partial\chi}\mathcal{R}_\chi^s(V^s)(\phi,g)\Big|_{\chi\in(0,1)}\right| \\
&\le \sup_{t\in[0,1]}\left|\frac{\partial}{\partial t}\mathcal{R}_t^s(V^s)(\phi,g)\right|
\end{aligned}$$

Mit Hilfe der Substitution $\frac{\zeta}{\beta} \to \zeta$ und Anwendung des Satzes 2.6.1 erhalten wir

$$\begin{aligned}
\frac{\partial}{\partial t}\mathcal{R}_t^s(V^s)(\phi,g) &= \int d\mu_{\gamma\beta^{-2}t}(\zeta)\left\{\frac{\gamma}{2\beta^2}\frac{\partial^2}{\partial\phi^2} + \frac{\partial}{\partial t}\right\} e^{-\mathcal{T}_t^s(\phi+\zeta,g)} \\
&= \int d\mu_{\gamma\beta^{-2}t}(\zeta) e^{-\mathcal{T}_t^s(V^s)(\phi+\zeta,g)}\{\ldots\}
\end{aligned}$$

[51] da auch $\mathcal{T}_t^s$ für die skalierten Potentiale definiert ist

mit

$$\{\ldots\} = \frac{\gamma}{2\beta^2}\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi+\zeta,g)\right)^2 - \left(\frac{\partial}{\partial t}+\frac{\gamma}{2\beta^2}\frac{\partial^2}{\partial\phi^2}\right)\mathcal{T}_t^s(V^s)(\phi+\zeta,g)\,.$$

Für $s\to\infty$ ist $\{\ldots\}$ nach 2.6.2 identisch Null und somit $\triangle(e^{-V^\infty})=0$. Dies gilt für das skalierende Potential, welches einen Approximanten des echten Fixpunktpotentials darstellt, natürlich nicht. Wir berechnen

$$\begin{aligned}
\left(\frac{\partial}{\partial t}+\frac{\gamma}{2\beta^2}\frac{\partial^2}{\partial\phi^2}\right)\mathcal{T}_t^s(V^s)(\phi,g) &= (1-\mathcal{P}^s)\left(\frac{\partial}{\partial t}+\frac{\gamma}{2\beta^2}\frac{\partial^2}{\partial\phi^2}\right)\mathcal{T}_t(V^s)(\phi,g)\\
&= \frac{\gamma}{2\beta^2}(1-\mathcal{P}^s)\left(\frac{\partial}{\partial\phi}\mathcal{T}_t(V^s)(\phi,g)\right)^2\\
&= \frac{\gamma}{2\beta^2}(1-\mathcal{P}^s)^2\left(\frac{\partial}{\partial\phi}\mathcal{T}_t(V^s)(\phi,g)\right)^2\\
&= \frac{\gamma}{2\beta^2}(1-\mathcal{P}^s)\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2\\
&= \frac{\gamma}{2\beta^2}\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2 -\\
&\quad \frac{\gamma}{2\beta^2}\mathcal{P}^s\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2\,.
\end{aligned}$$

Man beachte, daß der Projektor $1-\mathcal{P}^s$ mit Differentialoperatoren in ϕ und t vertauscht, da er nur die Variable g angreift. Wir erkennen dies auch explizit in der Integraldarstellung des Projektors.

Die Abschätzungsformel der Güte vereinfacht sich also zu

$$\left|\triangle\left(e^{-V^s}\right)(\phi,g)\right| \le \sup_{t\in[0,1]}\frac{\gamma}{2\beta^2}\int d\mu_{\gamma\beta^{-2}t}(\zeta)e^{-\mathcal{T}_t^s(\phi+\zeta,g)}\{\ldots\} \tag{2.153}$$

mit

$$\{\ldots\} = \left|\mathcal{P}^s\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi+\zeta,g)\right)^2\right|\,. \tag{2.154}$$

Zur Behandlung des $\{\ldots\}$-Terms nutzen wir

$$\sum_{n=1}^{N}\sum_{m=1}^{N}a_{n,m} = \sum_{n=1}^{N}\sum_{m=1}^{n}a_{m,n+1-m} + \sum_{n=N+1}^{2N-1}\sum_{m=n+1-N}^{N}a_{m,n+1-m}\,. \tag{2.155}$$

Nun gilt es, die quadrierte Ableitung des interpolierten skalierenden Potentials zu bestimmen. Diese Aufgabe lösten wir schon im Beweis zur Baumgraphendifferentialgleichung 2.6.3. Allerdings wirkte dort nicht der Projektor $\mathcal{P}^s$. Aus diesem Grund ordnen wir die Summe hier in Potenzen von g.

$$\begin{aligned}
&\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2 \\
\overset{(2.155)}{=}\quad &\left\{\sum_{n=1}^{s+1}\sum_{m=1}^{n}+\sum_{n=s+2}^{2s+1}\sum_{m=n-s}^{s+1}\right\} 4m(n{+}1{-}m)g^{n-1}(\lambda_{2m}\lambda_{2(n+1-m)})(g,t)\phi^{2n} \\
=:\quad &\left\{\sum_{n=1}^{s+1}+\sum_{n=s+2}^{2s+1}\right\} g^{n-1}\tilde{\mu}_{2n}(g,t)\phi^{2n}
\end{aligned}$$

Die neu definierten $\tilde{\mu}_{2n}$ sind Polynome in (g,t). Folglich wirkt der Projektor $\mathcal{P}^s$, der alle Potenzen von kleinerer Ordnung als $s+1$ vernichtet, nur auf den ersten Summanden. Wir erhalten[52]

$$\mathcal{P}^s\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2 =: \sum_{n=1}^{s+1} g^{s+1}\mu_{2n}(g,t)\phi^{2n} + \sum_{n=s+2}^{2s+1} g^{n-1}\mu_{2n}(g,t)\phi^{2n}\ .$$

Sofern $g\in[0,\tilde{g}]$ gilt für den Betrag dieser Projektion

$$\begin{aligned}
&\left|\mathcal{P}^s\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2\right| \\
\leq\ & g^{\frac{s}{2}}\sum_{n=1}^{s+1} g^{\frac{s-n}{2}+1}\left|\mu_{2n}(g,t)\right|\left(g^{\frac{1}{4}}\phi\right)^{2n} + g^{\frac{s}{2}}\sum_{n=s+2}^{2s+1} g^{\frac{n-s}{2}-1}\left|\mu_{2n}(g,t)\right|\left(g^{\frac{1}{4}}\phi\right)^{2n} \\
\leq\ & Cg^{\frac{s}{2}}\sum_{n=1}^{2s+1}\left(g^{\frac{1}{4}}\phi\right)^{2n}\ . \qquad (2.156)
\end{aligned}$$

Hierbei ist

$$C := \max_{n,t,g} g^{\nu(n)\left(\frac{s-n}{2}+1\right)}\left|\mu_{2n}(g,t)\right| < \infty \qquad (2.157)$$

mit

$$\nu(n) := \begin{cases} +1 & 1\leq n\leq s+1 \\ -1 & s+2\leq n\leq 2s+1\ . \end{cases} \qquad (2.158)$$

Das Maximum C existiert, da $g^{\cdots}|\mu_{2n}|$ in $g=0$ regulär und die betrachtete Menge $\{1,\ldots,2s+1\}\times[0,1]\times[0,\tilde{g}]$ kompakt ist.

[52] Für $n\geq s+2$ gilt $\tilde{\mu}_{2n}=\mu_{2n}$, für $n\leq s+1$ überleben in $g^{n-1}\tilde{\mu}_{2n}(g,t)$ nur Summanden der Ordnung g^{s+1}. Diese werden mit abgespaltetem g^{s+1}-Term in μ_{2n} verwahrt.

Den bisher noch ungenutzten Exponentialfaktor $e^{-\mathcal{T}_t^s(\phi+\zeta,g)}$ schätzen wir mit (2.137) ab. Dann dominieren wir die in polynomialer Form auftretenden Felder, indem wir den Faktor $e^{-\frac{\tilde{C}_s}{2}g\phi^4}$ extrahieren und ausnutzen, daß für beliebiges $n \in \mathbb{N}_0$ und $\alpha > 0$ die Ungleichung

$$\sup_{x\in\mathbb{R}} x^n e^{-\alpha x^2} < \infty \tag{2.159}$$

gültig ist. Für alle $(\phi, g) \in \mathcal{P}_{\tilde{g}}$ gilt somit

$$e^{-\frac{\tilde{C}}{2}g\phi^4}\left|\mathcal{P}^s\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2\right| \leq Cg^{\frac{s}{2}}\sum_{n=1}^{2s+1}C_n =: Dg^{\frac{s}{2}} .$$

Analog zu (2.145) - (2.148) erhalten wir

$$\left|\triangle\left(e^{-V^s}\right)(\phi,g)\right| \leq: C_\triangle g^{\frac{s}{2}} Z_{QU}(\phi,g) . \tag{2.160}$$

Wählen wir s groß genug, so gelten sicher $\sigma_\triangle = \frac{s}{2} > \frac{D}{4-D}$ und folgender

Satz 2.6.12
Es sei $s \in 2\mathbb{N}_0 + 1$. Dann existiert ein $X^\triangle = X^\triangle(s) \in \mathcal{I}$ mit $\sigma_\triangle > \frac{D}{4-D}$, so daß für alle $g \in [0, g^{\sigma_\triangle}]$:

$$\left\|\triangle\left(e^{-V^s}\right)\right\|_g \leq C_\triangle g^\triangle \tag{2.161}$$

komplettiert gemeinsam mit Satz 2.6.11 die Aussage, daß in Dimensionen $2 < D < 4$ das skalierte Potential V^s mit $\frac{s}{2} > \frac{D}{4-D}$ gemäß der Definition 2.5.3 einen approximierten Fixpunkt darstellt. Wir schreiben diese Bedingung an s in der Form[53]

$$s \geq \left[\frac{4+D}{4-D}\right] . \tag{2.162}$$

Minimalen Berechnungsaufwand bietet also die perturbative Ordnung s, die die Gleichung in (2.162) erfüllt[54]. Veranschaulicht wird dies in der Grafik 2.3. Allerdings muß man beachten, daß diese Wahl die Größe des Blockparameters L beeinflußt (2.80). Obwohl bei $\sigma_\triangle = \frac{s}{2}$ für alle $D \in (2,4)$ die Ungleichung $(4-D)\sigma_2 - D > 0$ per Definition von σ erfüllt ist, und somit L in jeder Dimension finit ist, existieren kritische Dimensionen[55] $D_{n\geq 4} = 4\frac{n-1}{n+1}$, für die

$$\lim_{\substack{D\to D_n \\ D<D_n}} (4-D)\sigma_\triangle - D = 0 \tag{2.163}$$

[53]Wir benutzen, daß man den nähst größeren Integer einer Zahl x durch $[x+1]$ berechnet.
[54]Sollte dieses s gerade sein, müssen wir es natürlich um eins inkrementieren.
[55]Die Dimensionen $D_{n\geq 2} = 4\frac{2n-1}{2n+1}$ sind gerade die Dimensionen, bei denen zu einer Fixpunktapproximante höherer Ordnung übergegangen werden muß.

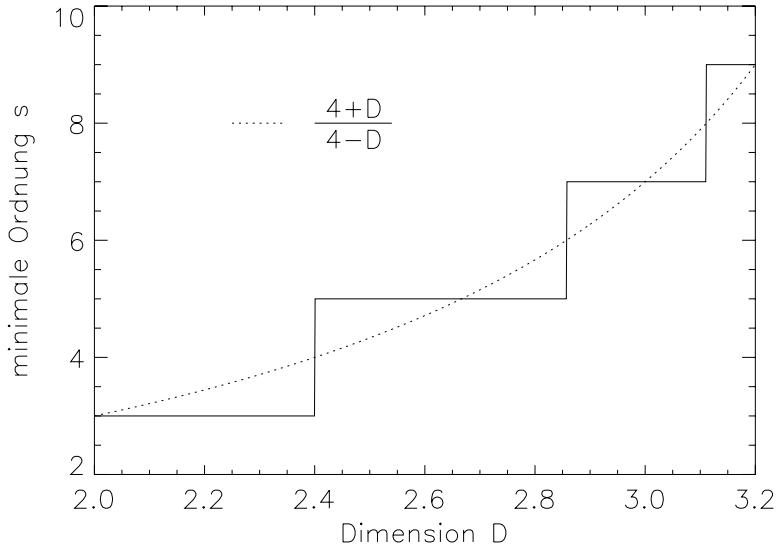


Abbildung 2.3: Diese Grafik veranschaulicht, welche Ordnung s die störungstheoretischen Fixpunktapproximanten in Abhängigkeit von der Dimension D mindestens besitzen müssen. In $D = 3$ Dimensionen muß Störungstheorie der Ordnung 7 betrieben werden.

gilt und folglich L divergiert. Dies veranschaulicht auch Abbildung 2.4. Man behebt dieses Problem jedoch, indem man zu einem approximierten Fixpunkt höherer Ordnung übergeht.

Eine andere Vorgehensweise läßt den Blockparameter konstant (z.B. $L = 2$) und betreibt Störungstheorie bis zu einer so hohen Ordnung s, daß die Abschätzung (d) im Beweis zu Lemma 2.5.5 ebenfalls gilt. Wir halten dieses Verfahren allerdings für ineffizient, da die Berechnung der störungstheoretischen Approximanten numerisch erfolgt.

2.7 Konstruktion der ϕ_3^4-Trajektorie

Bei der Konstruktion des approximierten Fixpunktes e^{-V^s} griffen wir, wie in Kapitel 2.6 zu sehen ist, auf Störungstheorie der Ordnung s zurück. Hierbei nahmen wir an, daß sich die Koeffizienten der Reihe V^s eindeutig rekursiv bestimmen lassen. In $D = 3$ Dimensionen treten allerdings zwei nicht lösbare

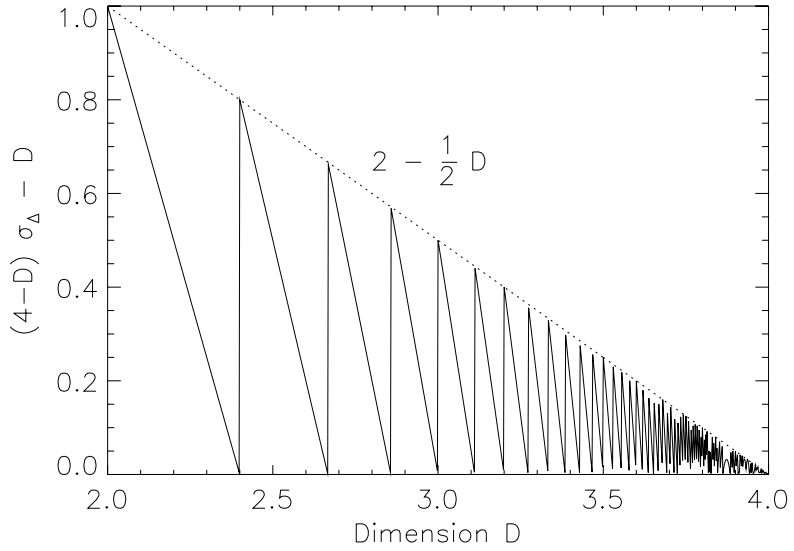


Abbildung 2.4: Der inverse Exponent der unteren Blockingparameterschranke konvergiert für die kritischen Dimensionen D_n linksseitig gegen Null.

Koeffizientengleichungen auf: die Massenresonanz $(1, 2)$ und die Vakuumresonanz $(0, 3)$.[56] Mit dem von ROLF und WIECZERKOWSKI erdachten Verfahren [RW] gelingt es jedoch, die Resonanzen bei einer perturbativen Lösung der erweiterten RG-Fixpunktgleichung zu eliminieren.

Die bisher nur in ihrer Kopplung g parametrisierten Trajektorien werden durch einen Parameter κ ergänzt, der über die Relation $\kappa = \kappa(g) = \log g$ explizit von g abhängt und somit der Reparametrisierung $\kappa \mapsto \log \kappa + \log \delta$ obliegt. Im folgenden sind die Koeffizienten $V^{\infty}_{2n,r}$ κ-abhängig, und wir erklären die formale Potenzreihe

$$V^{\infty}(\phi, g, \kappa) = \sum_{n=0}^{\infty} \sum_{r=\max\{1,n-1\}}^{\infty} V^{\infty}_{2n,r}(\kappa) g^r \phi^{2n} \tag{2.164}$$

mit

$$V_{2n,r}(\kappa) = \sum_{j=0}^{\left[\frac{r}{2}\right]} V^{\infty}_{2n,r,k} \kappa^j \ . \tag{2.165}$$

[56] Arbeitet man mit der normierten RGT, ist die Vakuumresonanz natürlich nicht vorhanden.

Der Ansatz einer oberen Grenze $\left[\frac{r}{2}\right]$ ermöglicht eine rekursive Berechnung der endlich vielen Koeffizienten $V^{\infty}_{2n,r,k}$ zu gegebener Ordnung in g und ϕ. Erweitern wir die Reparametrisierungsfunktion δ dann noch um den Parameter κ

$$\delta(g,\kappa) = (\delta g, \kappa + \log\delta) \;, \tag{2.166}$$

ist die RGT $\mathcal{T} \times \delta^\star$ für zwei-parametrige Potentiale erklärt, und wir bestimmen die Koeffizienten $V^{\infty}_{2n,g,\kappa}$ wiederum so, daß $\mathcal{T} \times \delta^\star(V^\infty) = V^\infty$ und die Randbedingungen einer ϕ^4-Trajektorie erfüllt sind. Beim Lösen der Fixpunktgleichung behandeln wir g und κ als unabhängige Variablen. Statt der beiden Resonanzen treten nun allerdings zwei frei wählbare Konstanten auf, zu denen verschiedenartig parametrisierte Kurven gehören. Sie werden i.a. zu Null gesetzt.

Aus V^∞ gewinnen wir wie zuvor die skalierenden Potentiale V^s. Natürlich bezieht sich der Projektor $\mathcal{P}^s$ immer noch auf die Kopplung g, denn sie ist ja nach der Substitution $\kappa = \log g$ der einzige Kurvenparameter. Aus diesem Grund gilt auch $O(\kappa) = O(\log g) = O(g^0)$, so daß logarithmische Terme bei Ordnungsbetrachtungen unberücksichtigt bleiben.

Desweiteren merken wir noch an, daß die skalierenden Potentiale V^s im Kurvenparameter g formal einmal stetig differenzierbar sind, da sie keine Terme der Form $\log g$ bzw. $g\log g$ enthalten. Diese Eigenschaft macht einen Zusatzparameter der Form $\kappa = \log g$ überhaupt erst sinnvoll, da er sonst für $g = 0$ nicht definiert wäre. Ferner sehen wir hier, wie wichtig es ist, daß unsere Transformation $\mathcal{T} \times \delta^\star$ die stetige Differenzierbarkeit des Argumentes erhält.

Wir wollen noch folgende Notation vereinbaren: Führen wir in Funktionen, die von (ϕ, g, κ) abhängen, die Substitution $\kappa = \log g$ aus, so verkürzen wir das Argument auf (ϕ, g), z.B. $V^{(s)}(\phi, g, \log g) \equiv V^{(s)}(\phi, g)$.

Satz 2.7.1

1. $\forall r \in \mathbb{N}\; \forall k \in \mathbb{N}_0\; \forall \epsilon \in (0, r]\; \exists g_0 \in \mathbb{R}^+\; \forall g \in [0, g_0]\;:\; \left|g^r \log^k g\right| \leq g^{r-\epsilon}$

2. $\forall r \in \mathbb{N}\; \forall k \in \mathbb{N}_0\; \exists g_0 \in \mathbb{R}^+\; \forall g \in [0, g_0]\;:\; \left|g^r \log^k g\right| \geq g^r$

Beweis: Alle Funktionen der Form $f_{\epsilon,k} : \mathbb{R}^+ \to \mathbb{R}_0^+ \quad g \mapsto |g^\epsilon \log^k g|$ mit $\epsilon > 0, k \in \mathbb{N}_0$ lassen sich mit $f_{\epsilon,k}(0) := 0$ stetig fortsetzen (L'Hospital). Folglich existiert ein $g_0 \in \mathbb{R}^+$, so daß $g^\epsilon |\log^k g| \leq 1 \;\; \forall g \in [0, g_0]$ und man zeigt (1). Für alle $g \in (0, e^{-1})$ gilt $|\log g| \geq 1$ und man erhält (2).

□

Aus dieser Abschätzung folgt

Korollar 2.7.2
Es seien $a \in \mathbb{R}$, $\epsilon \in (0,1)$ und $r \in \mathbb{N}, k \in \mathbb{N}_0$. Dann existiert ein $g_0 \in \mathbb{R}^+$, so daß $\forall g \in [0, g_0]$ gilt:

$$ag^r \log^k g \geq \begin{cases} (-1)^k a g^{r-\epsilon} & (a \geq 0 \wedge k \in 2\mathbb{N}+1) \vee (a < 0 \wedge k \in 2\mathbb{N}) \\ (-1)^k a g^r & \text{sonst} . \end{cases}$$

Mit Hilfe dieses Korollars gelingt es uns nun, das in g und $\log g$ entwickelte Potential nach unten gegen ein Potential in g abzuschätzen, dessen Struktur der in Kapitel 2.6 benutzten Form gleicht. Dies ist möglich, da die Baumgraphenkoeffizienten der Logarithmuskorrekturen verschwinden. Wir zeigen somit zunächst für $V^\infty = \mathcal{T}_0(V^\infty)$ die Relationen

$$\begin{aligned} V_{2n,n-1,0}(0) &= V_{2n,n-1}(0) \\ V_{2n,n-1,j}(0) &= 0 \quad \forall n \in \mathbb{N}_{\geq 2} \;\; \forall j \in \left\{1, \ldots, \left[\tfrac{n-1}{2}\right]\right\} . \end{aligned} \tag{2.167}$$

Beweis: Wir betrachten im folgenden Potentiale mit normalgeordneten Feldkomponenten. Es seien also

$$V(\phi, g, \kappa) = \sum_{r=1}^{\infty} \sum_{j=0}^{\infty} V_{r,j}(\phi) g^r \kappa^j$$

und

$$V_{r,j}(\phi) = \begin{cases} \sum_{n=0}^{r+1} V_{2n,r,j} : \phi^{2n} : & j \leq \left[\frac{r}{2}\right] \\ 0 & j > \left[\frac{r}{2}\right] . \end{cases}$$

Die Koeffizienten $V_{2(r+1),r,j}$ unterscheiden sich nicht von denen, die in einem nichtnormalgeordneten Potential auftauchen. Entwickeln wir $\mathcal{T} \times \delta^\star$ gemäß (2.27) in Kumulanten

$$\mathcal{T} \times \delta^\star(V)(\phi, g, \kappa) = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i!} \left\langle [\alpha V(\cdot, \delta g, \kappa + \log \delta);]^i \right\rangle^T_{\gamma, \beta\phi} ,$$

erhalten wir für V^∞ in beliebiger Ordnung (r, j) mit $j \leq \left[\frac{r}{2}\right]$

$$\begin{aligned} V^\infty_{r,j}(\phi) &\equiv \sum_{i=1}^{r} \sum_{t=j}^{\infty} \sum_{\sum_{k=1}^{i} s_k = r} \sum_{\sum_{k=1}^{i} j_k = t} A_{r,j,i,t} \left\langle \alpha V^\infty_{s_1,j_1}, \ldots, \alpha V^\infty_{s_i,j_i} \right\rangle^T_{\gamma,\beta\phi} \\ &=: \sum_{t=j}^{\infty} \alpha A_{r,j,1,t} \left\langle V^\infty_{r,j} \right\rangle_{\gamma,\beta\phi} + K_{r,j}(V^\infty)(\phi) . \end{aligned}$$

Die Koeffizienten bestimmen sich über

$$V_{2n,r,j} = \sum_{t=j}^{[\frac{r}{2}]} B_{2n,r,j,t} V_{2n,r,t} + \frac{1}{(2n)!} \left(P_{2n,1}, K_{r,j}(V^\infty)\right)_1$$

Hierbei ist $B_{2n,r,j,t} = \alpha\beta^{2n} A_{r,j,1,t} = \binom{t}{j} \alpha\beta^{2n}\delta^r (\log\delta)^{t-j}$. Nun zeigen wir die Behauptung des Lemmas per Induktion. Der Induktionsanfang $r = 1$ ist trivial, da der ϕ^4-Term in g^1 logarithmusfrei ist. Zur Bestimmung von $V^\infty_{2(r+1),r,t}$ benötigen wir $K_{r,j}(V)(\phi)$, welches eine Superposition von Kumulanten

$$\begin{aligned}
&\left\langle V^\infty_{s_1,j_1}, \ldots, V^\infty_{s_i,j_i} \right\rangle^T_{\gamma,\beta\phi} \\
&= \sum_{n_1=0}^{s_1+1} \cdots \sum_{n_i=0}^{s_i+1} V^\infty_{2n_1,s_1,j_1} \cdots V^\infty_{2n_i,s_i,j_i} \left\langle :\phi^{2n_1}:, \ldots, :\phi^{2n_i}: \right\rangle^T_{\gamma,\beta\phi} \\
&= \sum_{n_1=0}^{s_1+1} \cdots \sum_{n_i=0}^{s_i+1} \sum_{n=0}^{n_{max}} C_n^{n_1,\ldots,n_i} V^\infty_{2n_1,s_1,j_1} \cdots V^\infty_{2n_i,s_i,j_i} : \phi^n :
\end{aligned}$$

darstellt. Der bei obiger Konstruktion entstehende Vertex mit maximaler Beinzahl $: \phi^{n_{max}} :$ besitzt somit

$$n_{max} = (2n_1 - 1) + \sum_{k=2}^{i-1} (2n_k - 2) + (2n_i - 1) = 2(\sum_{k=1}^{i} n_k - i + 1)$$

Beine, und es folgt

$$n_{max} = 2(r+1) \quad \Leftrightarrow \quad \forall k \in \{1, \ldots, i\} \; : \; n_k = s_k + 1 \; .$$

Der $: \phi^{2(r+1)} :$-Anteil in $K_{r,j}(V^\infty)$ bestimmt sich somit zu

$$\propto \sum_{i=2}^{r} \sum_{t=j}^{\infty} \sum_{\sum_{k=1}^i s_k = r} \sum_{\sum_{k=1}^i j_k = t} V^\infty_{2(s_1+1),s_1,j_1} \cdots V^\infty_{2(s_i+1),s_i,j_i} \; .$$

Da nun aber $j \geq 1$ ist, existiert für alle $t \geq j$ ein $k \in \{1, \ldots, i\}$, so daß $j_k \geq 1$. Da aber $s_k < r$ (wegen $i > 1$), folgt nach Induktionsvoraussetzung $V_{2(s_k+1),s_k,j_k} = 0$. Somit ist

$$\int d\mu_1(\phi) : \phi^{2(r+1)} : K_{r,j}(V^\infty)(\phi) = 0 \; .$$

Das Gleichungssystem zur Bestimmung der $V_{2(r+1),r,j}$ reduziert sich für $j \geq 1$ zu

$$V^\infty_{2(r+1),r,j} = \sum_{t=j}^{[\frac{r}{2}]} B_{2(r+1),r,j,t} V^\infty_{2(r+1),r,t} \; .$$

Beginnend mit $j = [\frac{r}{2}]$ bestimmen sich die Koeffizienten rekursiv zu Null, da die Faktoren $B_{2(r+1),r,j,t} \neq 1$. Für $V_{2(r+1),r,0}$ erhalten wir die Bestimmungsgleichung (2.40) und folglich dieselben Baumgraphenkoeffizienten wie bei der Einfachentwicklung.

□

Obiges Lemma verallgemeinern wir nun für alle $t \in [0,1]$.

Satz 2.7.3
Es sei $\mathcal{T}_t(V^\infty)$ das interpolierte perturbative Fixpunktpotential mit t-abhängigen Koeffizienten. Dann sind

$$\forall n \in \mathbb{N}_{\geq 2} \quad : \quad V_{2n,n-1,0}(t) \equiv V_{2n,n-1}(t) = b_{2n}(t)$$

und

$$\forall n \in \mathbb{N}_{\geq 3} \;\; \forall j \in \left\{1, \ldots, \left[\frac{n-1}{2}\right]\right\} \quad : \quad V_{2n,n-1,j}(t) = 0 \; .$$

Beweis: Zu Beginn wollen wir die Frage behandeln, ob die Transformation $\mathcal{T}_t$ Terme der Form $g^r\kappa^l$ mit $l > [\frac{r}{2}]$ generieren kann. Es sei g^r durch g^{r_i} erzeugt, d.h. $g^r = \prod g^{r_i} = g^{\sum r_i}$. Dann gilt für den Exponenten der zugehörigen κ-Potenz $\sum l_i \leq \sum [\frac{r_i}{2}] \leq [r/2]$, und es ist gezeigt, daß V^∞ unter $\mathcal{T}_t$ formerhaltend ist.

Für die nun κ-abhängigen Baumgraphen mit $n \geq 2$ folgt aus Satz 2.6.3

$$\sum_{j=0}^{\left[\frac{n-1}{2}\right]} \dot{V}_{2n,n-1,j}(t)\kappa^j = \sum_{m=2}^{n-1} \sum_{j_2=0}^{\left[\frac{n-m}{2}\right]} \sum_{j_1=0}^{\left[\frac{m-1}{2}\right]} \alpha_{n,m} V_{2m,m-1,j_1}(t) V_{2(n+1-m),n-m,j_2}(t) \kappa^{j_1+j_2} \; ,$$

wobei $\alpha_{n,m} = \frac{2\gamma}{\beta^2} m(n+1-m)$. Da wir g und κ als unabhängig betrachten, lösen wir obige Gleichung durch Koeffizientenvergleich in κ. Man erkennt sofort, daß die Funktionen $V_{2n,n-1,0}(t)$ derselben Differentialgleichung gehorchen wie die Baumgraphenkoeffizienten $b_{2n}(t)$. Aus (2.167) ergibt sich

$$\mathcal{T}_0(V^\infty)(\phi, g, \kappa) = \alpha \mathcal{T}_1(V^\infty)(\beta\phi, \delta g, \log\delta + \kappa) \; ,$$

für $n \geq 2$ wiederum

$$V_{2n,n-1,0}(0) = (L^2)^{2-n} V_{2n,n-1,0}(1) \; ,$$

und der erste Teil des Satzes ist gezeigt. Der Beweis der zweiten Aussage folgt über Induktion: Für $n = 3$ gilt $\dot{V}_{6,2,1}(t) = 0$ und aufgrund obiger Forderung $V_{6,2,1}(t) = 0$. Im Induktionsschritt nutzen wir aus, daß die Indizes

j_1,j_2 nicht gleichzeitig Null und m,$n+1-m$ kleiner gleich $n-1$ sind, so daß also $\dot{V}_{2n,n-1,j\geq 1} = 0$. Mit oben geforderter Randbedingung erhalten wir die Behauptung.

□

Nennen wir den $(g, \log g)$-abhängigen Koeffizienten eines interpolierten Feldes ϕ^{2n} wieder $g^{n-1}\lambda_{2n}(g,t)$, so erhalten wir mit Hilfe des Satzes 2.7.3 für $n \in N_{\geq 2}$

$$\lambda_{2n}(g,t) = b_{2n}(t) + O(g) \; . \tag{2.168}$$

Man beachte, daß die Logarithmuskorrekturen in $O(g)$ mit g-Faktoren gepaart sind. Somit existiert zu gegebenem $s \in 2\mathrm{N}_0 + 1$ und $\epsilon \in (0,1)$ nach Lemma 2.7.2 ein $g_0 = g_0(s,\epsilon) \in \mathbb{R}^+$, so daß alle auftretenden $g\log g$-Terme durch g-Potenzen, deren Ordnung größer $n-1$ ist, ersetzt werden können. Da nur endlich viele Korrekturterme auftreten, ist die Existenz einer solchen Schranke für den Kopplungsparameter gesichert.

Für das abgeschätzte Potential gilt:

$$\lambda_{2n}(g,t) \geq b_{2n}(t) + O(g^{1-\epsilon}) =: \tilde{\lambda}_{2n}(g,t) \quad \Longrightarrow \quad \tilde{\lambda}_{2n}(0,t) = b_{2n}(t) \tag{2.169}$$

Nun werden wir zeigen, daß die in g und $\log g$ perturbativ entwickelten skalierenden Potentiale den Bedingungen eines approximierten Fixpunktes genügen. Auf dieselbe Weise wie im Beweis des Satzes 2.6.11 konstruieren wir eine untere Schranke für $\mathcal{T}_t^s(V^s)$. Somit erhalten wir sogar dasselbe $\tilde{C}$. Als obere Kopplungsparameterschranke g_1 wählen wir das Minimum der Grenzen, die 2.6.11 und 2.7.2 fordern. Es gilt somit für alle $(\phi, g) \in \mathcal{P}_{g_1}$:

$$\begin{aligned}
\mathcal{T}_t^s(V^s)(\phi,g) &= \sum_{n=0}^{s+1} g^{\max\{n-1,1\}} \lambda_{2n}(g,t)\phi^{2n} \\
&\geq \sum_{n=0}^{s+1} g^{\max\{n-1,1\}} \tilde{\lambda}_{2n}(g,t)\phi^{2n} \\
&\geq g\tilde{\lambda}_0(g,t) + g\tilde{\lambda}_2(g,t)\phi^2 + \tilde{C}g\phi^4
\end{aligned} \tag{2.170}$$

Hieraus folgern wir nun die Endlichkeit der g-Norm für $g \in [0, g_1]$.

Wir zeigen nun, daß auch die Güte $\triangle(V^s)$ den Bedingungen eines approximierten Fixpunktes genügt. Hierzu verfahren wir genauso wie im Kapitel 2.6.7. Die Abschätzung von $\mathcal{P}^s\left(\frac{\partial}{\partial\phi}\mathcal{T}_t^s(V^s)(\phi,g)\right)^2$ läuft problemlos, d.h

$$C := \max_{n,t,g} g^{\nu(n)\left(\frac{s-n}{2}+1\right)} \left|\mu_{2n}(g,t)\right| < \infty \; . \tag{2.171}$$

$L-1$	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
$100 g_{max}$	1.98541	2.35698	2.39793	2.40206	2.40247	2.40251

Tabelle 2.1: Die Funktion der durch die Baumgraphenschranke gegebenen Maximalkopplung ist fallend in L. Aus diesem Grund betrachten wir in dieser Tabelle $L \to 1$. Zur Berechnung wurde ein $g \log g$-Potential 7. Ordnung in $D = 3$ benutzt. ≈ 0.025 scheint eine obere Schranke für g zu sein.

Für $0 \leq n \leq s+1$ regularisiert $g^{\frac{s-n}{2}+1} = O(g^{\frac{1}{2}})$ eventuell in $\mu_{2n}(g,t)$ auftauchende Singularitäten, die durch alleinstehende Logarithmen verursacht werden. Die Terme mit $s+2 \leq n \leq 2s+1$ wurden nicht durch $\mathcal{P}^s$ beschnitten, und somit sind die Koeffizienten $\mu_{2n}(g,t)$ regulär in $g = 0$.

2.8 Numerische Ergebnisse

Ziel dieses Abschnittes war es, die ϕ^4-Trajektorie bis zu einer möglichst maximalen Kopplung g_{max} zu berechnen, um z.B. den Limes $g \to \infty$ zu bestimmen/abzuschätzen. Sollte er existieren, entspricht er einem Fixpunkt der RGT.

- Der Limes entspricht dem Hochtemperaturfixpunkt Z_{QU}. In diesem Fall hätten wir zwei RGT-invariante Trajektorien, die Z_{UV} und Z_{QU} verbinden, sich aber in ihren Ein-/Auslaufrichtungen unterscheiden (Ableitungen an den Stellen $g = 0$ und $g = \infty$). Es würde die Frage aufkommen, ob eine ganze Schar invarianter Trajektorien existiert, die den trivialen und den quadratischen Fixpunkt verbinden.

- Die ϕ^4-Trajektorie endet in einem nichttrivialen Fixpunkt. In diesem Fall liefert die Konstruktion der Kurve den Fixpunkt mit.

- Die Trajektorie endet nicht in einem Fixpunkt.

Die von uns benutzte Konstruktion ist nur für kleine Kopplungen g geeignet, da wir an vielen Stellen Restriktionen an g stellen müssen. Einen fundamentalen Einfluß hat die von uns benutzte Norm, die immer eine Relation zur Gauß-Trajektorie verlangt. Ziel der Optimierung dieses Konstruktionsschemas ist es, alle g-Abhängigkeiten zu extrahieren. Deshalb stellt auch die Größe von g_{max} ein Maß für die Qualität unseres Verfahrens dar.

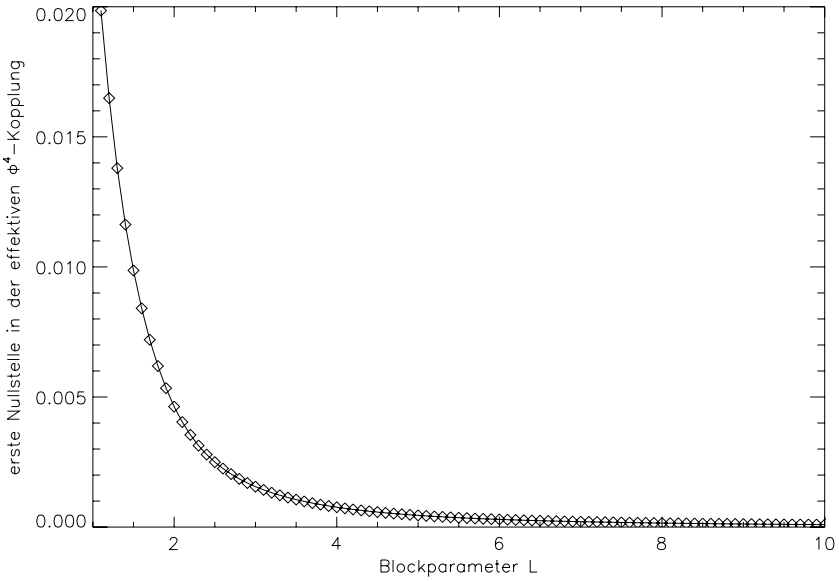


Abbildung 2.5: Die maximale Kopplung der Baumgraphenschranke ist streng monoton fallend in dem Blockparameter L.

Wir berechnen nun die Grenzkopplung in $D = 3$ Dimensionen, die sich aus der Baumgraphenschranke ergibt. Dazu bestimmen wir mittels Computeralgebra die Störungsreihe 7. Ordnung in g und $\log g$ und ermitteln numerisch die erste positive Nullstelle der g-abhängigen, effektiven ϕ^4-Wechselwirkung $\tilde{\lambda}_4^7(g, 0)$ (2.138). Wie man in der Abbildung 2.5 sieht, ist diese obere Schranke für die maximale Kopplung L abhängig, für $L \to 1$ scheint sie, wie Tabelle 2.8 zeigt, gegen einen Wert ≈ 0.024 zu konvergieren. Obwohl wir L beliebig klein machen können, müssen wir die Restriktion (2.80) beachten. In unserem Fall ($D = 3$ und $\sigma_\triangle = 3.5$) erhalten wir eine minimale Untergrenze von $L \approx 30$ und folglich eine sehr kleine maximale Kopplung. Nun mag man die perturbative Reihe bis zu Ordnungen > 7 bestimmen, doch dazu später mehr.

Eine weitere interessante Frage ist, inwieweit die Grenzkopplung g_{max} von der Dimension D abhängt. Hierzu haben wir Störungsrechnung in der minimalen Ordnung $s = \left[\frac{4+D}{4-D}\right]$ betrieben und bei konstantem $L = 2$ die erste positive Nullstelle der effektiven ϕ^4-Kopplung berechnet. Wir benutzten $D_n = 2 + \frac{n}{100}$ mit $n = 1, \ldots, 111$. Diese Wahl gewährleistet für alle $D_n \neq 3$ Resonanzfreiheit und folglich eine einfachere numerische Berechnung. Die Obergrenze $D_{111} = 3.11$ ergibt sich aus der begrenzten Rechnerleistung, da für $D = 3.12$

schon Störungstheorie bis zur 9. Ordnung betrieben werden muß.[57] Doch auch für den Bereich $\{D_n\}$ lassen sich interessante Aussagen machen. Für $2.01 \leq D_n \leq 2.85$[58] und $2.96 \leq D_n \leq 2.99$ ist die effektive Kopplung auf $\mathbb{R}_0^+$ nullstellenfrei, und die Baumgraphenabschätzung demnach für alle Kopplungen gültig. Bei $D_6 = 4\frac{6-1}{6+1} = 2.857...$ müssen wir von der Ordnung $s = 5$ zur Ordnung $s = 7$ übergehen (siehe hierzu auch Abbildung 2.3). Folge ist ein skalierendes Potential $V^{2.86}$, welches sich bis zur 5. Kopplungsordnung nur geringfügig von $V^{2.86}$ unterscheidet $((1 - \mathcal{P}^5)(V^{2.86} - V^{2.85}) \approx 0)$, aber zusätzliche Terme in g^6 und g^7 aufweist. Als Konsequenz besitzt $\tilde{\lambda}_4^7$ nun Nullstellen, deren Lage stetig von D abzuhängen scheint, siehe 2.6 oben. Dies ist jedoch nicht wahr, denn für 2.95 und 2.96 verändert die effektive Kopplung ihr Großkopplungsverhalten von $\lim_{g\to\infty} \tilde{\lambda}_4^7(D = 2.95) = -\infty$ zu $\lim_{g\to\infty} \tilde{\lambda}_4^7(D = 2.96) = \infty$. Dieser Prozeß kann nicht stetig verlaufen. Die aufgrund dieses Verhaltens bis dato sichere Nullstelle verschwindet wieder, da auch im Bereich endlicher Kopplungen Positivität vorliegt. Für $D = 3$ erhalten wir wieder ein endliches g_{max}, welches sich doch deutlich von den folgenden g-Grenzwerten, die nicht aus einer Doppelentwicklung gewonnen wurden, unterscheidet 2.6 unten. Auch das gegensätzliche Steigungsverhalten zwischen den Dimensionen 3.04 und 3.05 deutet eine Nichtdifferenzierbarkeit oder Unstetigkeit (evtl. in Form einer Singularität) an.

Wir erkennen somit, daß die Abhängigkeit $g_{max} = g_{max}(D)$ nicht von so einfacher Natur ist wie die Abhängigkeit von L. Auch wenn wir eine einheitliche Störungsordnung s benutzen, die selbstverständlich nur bis zu einer gewissen Dimension ausreicht, bleibt ein komplexes Gefüge zurück. Grund ist die Kettenbruchkonstruktion, die durch geringe Änderung der Koeffizienten $V_{2n,r}$ bzw. $V_{2n,r,k}$, Null- und Polstellen erzeugt/vernichtet und asymptotisches Verhalten ändert. Berechnen wir die Potentiale, die nur bis zur dritten oder fünften Ordnung bestimmt werden mußten, bis zur siebten Ordnung, so ändert sich der Grenzwert nicht.

Was läßt sich zu der Frage sagen, ob für $D \to 4$ die Kopplungsschranke endlich bleibt? Wenn wir beweisen können, daß $\tilde{\lambda}_4^s$ für $s \to \infty$ gegen eine positive reelle Zahl konvergiert, wissen wir auch um die Existenz einer positiven Grenzkopplung.

Zudem haben wir gesehen, wie gering die Parametrisierungslänge unser Trajektorie ist. Obwohl wir nur eine der vielen Restriktionen an g betrachtet

[57] Wir haben hierzu ein *MapleV*-Programm geschrieben. In einer Hochsprache wären Berechnungen höherer Ordnungen gewiß kein Problem gewesen, für die qualitative Betrachtung der dimensionalen Abhängigkeit reicht die 9. Ordnung jedoch aus.

[58] Für $s = 3,5$ verkommt $\tilde{\lambda}_4^s$ zu einer ganzrationalen Funktion.

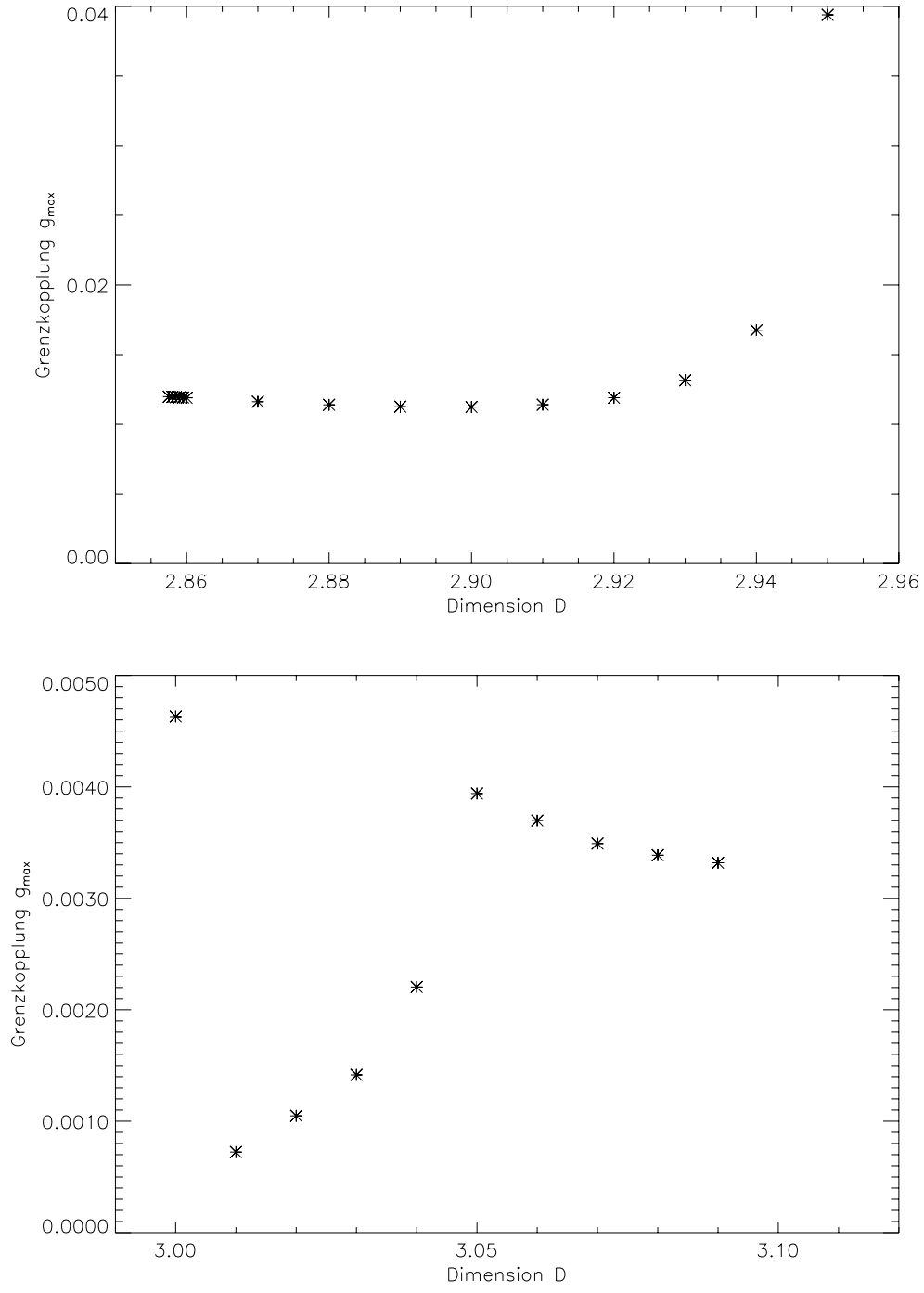


Abbildung 2.6: Erste positive Nullstellen der effektiven ϕ^4-Kopplung mit $s = \left[\frac{4+D}{4-D}\right]$. Bei $D = 3$ wurde eine Doppelreihenentwicklung benutzt.

haben, wissen wir, daß $g_0 \ll 1$. Es ist noch ein weiter Weg zu einer Konstruktion mit $g_0 = \infty$.

Kapitel 3

Perturbative Konstruktion der ϕ_3^4-Trajektorie auf dem Gitter

In diesem Abschnitt werden wir die ϕ_3^4-Trajektorie auf dem kubischen Gitter $\Lambda(a)$ perturbativ berechnen. Dazu benutzen wir viele Erkenntnisse aus der hierarchischen Approximation, deren Gültigkeit sich (formal) auf die GRG überträgt.

Wir beginnen mit der Konstruktion von Operatoren, die eine Interpolation zwischen Gitter- und Kontinuumsformulierung der RG ermöglichen [GK84, Wie98]. Mit diesen definieren wir die Gitterkerne der ϕ_3^4-Trajektorie über Kontinuumsfunktionen. Aus der Störungstheorie erhalten wir Bestimmungsgleichungen für die kontinuierlichen, gittertranslationsinvarianten Impulskerne der Kurve, die durch das Einfügen einer Gitterinterpolationsfunktion auf dieselbe Weise wie in [Wie97d, Wie97b] behandelt werden können. Wir geben ein explizites und ein implizites Verfahren zur Berechnung an. Die auch auf dem Gitter auftretenden Resonanzen werden wie schon im hierarchischen Modell durch Doppelentwicklung gelöst [Wie97b].

3.1 Der Operator $A^{(\infty)}$

Der kinetische Anteil unserer Theorie wird durch die perfekte, masselose Gitterkovarianz $\nu = \nu_{perf}$ beschrieben, die wir im Kapitel 1.3.1 hergeleitet haben. Die GRGT begegnet uns somit in der Form:

$$R(Z)(\phi) = \frac{\int d\mu_\Gamma(\zeta) Z(A\phi + \zeta)}{\int d\mu_\Gamma(\zeta) Z(\zeta)} \,. \tag{3.1}$$

Hierbei sind $A = \nu C^\dagger \nu^{-1}$ und $\Gamma = \nu - A\nu A^\dagger$. Da wir nicht - wie im hierarchischen Bild - über so mächtige mathematische Hilfsmittel (z.B. *contraction mapping*) verfügen, um den Raum der Wirkungen Z geeignet zu behandeln,[1] betrachten wir für perturbative Betrachtungen im folgenden die Transformation für Potentiale

$$\mathcal{T}(V)(\phi) = -\ln \left\langle e^{-V} \right\rangle_{\Gamma, A\phi} + \ln \left\langle e^{-V} \right\rangle_{\Gamma,0} \ . \tag{3.2}$$

Aufgrund der Normierung schränken wir uns OBdA wieder auf die Äquivalenzklasse[2] $V(0) = 0$ ein. Das freie Feld, also $V = 0$, ist trivialer Fixpunkt, und Linearisierung an diesem liefert analog (2.14)[3]

$$\mathcal{DT}(V)(\phi) = \langle V \rangle_{\Gamma, A\phi} - \langle V \rangle_{\Gamma,0} \ . \tag{3.3}$$

Als nächstes konstruieren wir einen Operator, mit dessen Hilfe wir ein Feld vom Kontinuum $\Lambda(0)$ auf das Gitter $\Lambda(a)$ transferieren können und *vice versa*. Benutzen wir als Ausgangsgitter für den Operator S nicht die Gitterkonstante a, sondern $\frac{a}{L^n}$ mit $n \in \mathbb{N}_0$, so erhalten wir $S_n : \mathcal{H}(\frac{a}{L^{n-1}}) \to \mathcal{H}(\frac{a}{L^n})$ mit $S_0 = S$. Diese Operatoren können wir hintereinanderschalten und definieren für $n \in \mathbb{N}$

$$S^n : \mathcal{H}(a) \to \mathcal{H}(\frac{a}{L^n}) \quad S^n(\phi)(x) = S_n \circ \ldots \circ S_1(\phi)(x) = L^{n\sigma}\phi(L^n x) \ . \tag{3.4}$$

Wir wollen einige Worte über den Fall $n \to \infty$ verlieren, der ein Gitterfeld auf das Kontinuum übertragen würde.[4] Für $x = 0$ gilt $S^n(\phi)(0) = L^{n\sigma}\phi(0)$. Da $\sigma = \frac{D}{2} - 1 > 0$ für $D \in (2, \infty)$, existiert $S^\infty(\phi)(0)$ nur, wenn $\phi(0) = 0$, und ist identisch Null. Da $\mathcal{H}(a)$ isomorph zum l_1 ist, folgt OBdA die Eigenschaft $|\phi(x)| < \frac{1}{|x|}$ für $|x| \to \infty$. Für $x \neq 0$ wählt man n groß genug, so daß $|S^n(\phi)(x)| < L^{(\sigma-1)n}\frac{1}{|x|} \leq \frac{1}{|x|}$ für $D \in (2, 4]$. Dies stimmt mutig, doch sollte man nicht vergessen, daß die Interpolation zwischen Gitter und Kontinuum nicht richtig funktioniert, da man z.B. einen Punkt des $\mathbb{R}^D$ mit irrationaler Komponente aus einem Gitter mit rationaler Gitterkonstante a nicht gewinnen kann. Würden wir unsere euklidische Raum-Zeit mit $\mathbb{Q}^D$ identifizieren und $a \in \mathbb{Q}$ fordern, wäre $S^{(n)}$ wohldefiniert und die Interpolation perfekt.

[1] Dieser Tatbestand stellt eher ein Unvermögen unsererseits dar, Funktionalräume mathematisch exakt zu fassen.

[2] bezüglich $V_1 \sim V_2 :\Leftrightarrow V_1 - V_2 \in \mathbb{R}$

[3] Man muß sich natürlich die Frage stellen, inwieweit eine Parameterableitung nach ϵ analog (3.3) gültig ist. Hier handelt es sich ja um eine unendliche Vertauschung von Integration und Differentiation. Wir wollen uns mit solchen Problemen in Zukunft nicht aufhalten und eine Wohldefiniertheit annehmen.

[4] Ein endliches Gitter würde nach unendlich vielen Stauchungen auf den Ursprung kontrahiert werden.

Wir definieren einen neuen A-Kern gemäß

$$A^{(n)} : \mathcal{H}(a) \to \mathcal{H}(\frac{a}{L^n}) \quad A^{(n)} := S^n A^n \ . \tag{3.5}$$

Da $A^n = \nu \left(C^\dagger\right)^n \nu^{-1}$, erfüllt $A^{(n)}$ die Aufgabe eines n-maligen Reblockens mit anschließender n-maliger Kontraktion. Desweiteren gilt die Relation

$$A^{(n)} A = S^n A^{n+1} = S_{n+1}^{-1} S^{n+1} A^{n+1} = S_{n+1}^{-1} A^{(n+1)} \ . \tag{3.6}$$

Definieren wir nun den Operator $A^{(\infty)} = \lim_{n\to\infty} A^{(n)}$ und gehen davon aus, daß dieser Grenzoperator existiert, so leitet sich aus (3.6) die *intertwiner* Eigenschaft

$$A^{(\infty)} A = S_\infty^{-1} A^{(\infty)} \tag{3.7}$$

ab. $A^{(\infty)}$, der Gitterfelder in Kontinuumsfelder transferiert, zeigt auf, daß dem Operator A im Kontinuum der Dilatationsoperator S_∞^{-1} entspricht. Die Möglichkeit, mittels des Operators $A^{(\infty)}$ vom Gitter ins Kontinuum zu wechseln, nutzen wir aus, um die Berechnung der ϕ^4-Trajektorie auf die Kontinuumsergebnisse zurückzuführen.

Nach einigem Rechenaufwand erhalten wir für den *intertwiner* $A^{(\infty)}$ die Impulsraumdarstellung

$$A^{(\infty)}(x,z) = \int_{\tilde{\Lambda}(a)} \frac{d^D p}{(2\pi)^D} \left[e^{iqx} \frac{\tilde{\chi}(q)}{q^2} \right]_{\Lambda(\frac{2\pi}{a})} (p)\ e^{-ipz} \frac{1}{\tilde{\nu}(p)} \tag{3.8}$$

mit

$$[F(q)]_{\Lambda(a)}\,(p) := \sum_{Q \in p + \Lambda(a)} F(Q) \tag{3.9}$$

und

$$\tilde{\chi}(p) = \prod_{\mu=1}^{D} \frac{\sin\left(\frac{p_\mu a}{2}\right)}{\frac{p_\mu a}{2}} e^{-i\frac{p_\mu a}{2}} \ . \tag{3.10}$$

Man zeigt leicht, daß $A^{(\infty)}$ invariant gegenüber den Symmetrieoperationen des Gitters ist.

3.2 Störungsrechnung auf dem Gitter

Im hierarchischen Bild waren die normalgeordneten Monome Eigenfunktionen der linearisierten RGT. Das ist auf dem Gitter nicht so. Wir erhalten

mit (B.13) und (B.2)

$$\begin{aligned} & : \phi(x_1) \dots \phi(x_n) :_\nu \\ \overset{\langle\cdot\rangle_{\Gamma, A\phi}}{\longrightarrow} \quad & : A\phi(x_1) \dots A\phi(x_n) :_{A\nu A^\dagger} \\ = \quad & \frac{\partial^n}{\partial J(x_1) \dots \partial J(x_n)} \exp\left\{ (\phi, A^\dagger J) - \frac{1}{2}(A^\dagger J, \nu A^\dagger J) \right\} \Bigg|_{J=0} . \end{aligned}$$

Die „Kettenregel“ für Funktionalableitungen

$$\frac{\partial}{\partial J(x)} A^\dagger J(y) = A(x,y) = \int d^D z A(x,z) \frac{\partial}{\partial (A^\dagger J)(z)} A^\dagger J(y)$$

liefert dann

$$\begin{aligned} & : \phi(x_1) \dots \phi(x_n) :_\nu \qquad (3.11) \\ \overset{\langle\cdot\rangle_{\Gamma, A\phi}}{\longrightarrow} \quad & \int_{\bigotimes \Lambda(a)} d^D z_1 \dots d^D z_n A(x_1, z_1) \dots A(x_n, z_n) : \phi(z_1) \dots \phi(z_n) :_\nu \ . \end{aligned}$$

Bei den A-Kernen handelt es sich um ausgeschmierte δ-Distributionen, die für $|x - y| \gg 1$ exponentiell abfallen. Wegen $\lim_{a \to 0} A(x,y) = \delta(x,y)$ werden die normalgeordneten Felder auf dem Kontinuum wieder zu Eigenfunktionen der RGT. Die „Eigenfunktionen“ der normierten RGT $\mathcal{DT}$ schreiben sich als

$$\overline{: \phi(x_1) \dots \phi(x_n) :_\nu} :=: \phi(x_1) \dots \phi(x_n) :_\nu - : \phi(x_1) \dots \phi(x_n) :_\nu \Big|_{\phi=0} . \qquad (3.12)$$

In Zukunft werden wir die normierten, normalgeordneten Monome mit der üblichen Darstellung ohne Überstrich identifizieren. Die normalordnende Kovarianz ν ergänzen wir nur in benötigten Fällen.

Wir wollen in diesem Kapitel wiederum ein skalierendes Paar bestimmen, das aus einer unter der RG-invarianten Potentialkurve V und einer Reparametrisierungsfunktion β besteht. Die zugehörigen formalen Potenzreihen definieren wir wie in (2.24) und (2.25). Für die Feldkomponenten setzten wir an:[5]

$$V_r(\phi) = \sum_{n=1}^{r+1} \frac{1}{(2n)!} \int_{\bigotimes \Lambda(a)} d^D x_1 \dots d^D x_{2n} V^{latt}_{2n,r}(x_1, \dots, x_{2n}) : \phi(x_1) \dots \phi(x_{2n}) : \qquad (3.13)$$

Die Funktionale V_r sind Taylor-Polynome $2(r+1)$. Grades in unendlich vielen Variablen $\{\phi(x)\}_{x \in \Lambda(a)}$. Die Wahl der normalgeordneten Darstellung ist auf

[5] Die explizite Darstellung von $V^{latt}_{4,1}$ und $V^{latt}_{2,1}$ geben wir später.

dem Gitter nicht zwingend, liefert aber beim Übergang zum Kontinuumsformalismus eine Basis aus Eigenfunktionen. Für eine rigorose Berechnung der ϕ^4-Trajektorie wird die lineare Hülle der normalgeordneten Monome zu groß sein, für eine perturbative Konstruktion, die sich wie schon in 2.3 auf die lineare Transformation reduziert, ist dieser Raum jedoch sehr gut geeignet.

Per Definition sind die V_r symmetrisch in ϕ.[6] Der Ansatz einer Obergrenze $r+1$ ist - wie schon in der hierarchischen Approximation (2.35) - eine Folge der Verwendung eines Feldpolynoms vierter Ordnung in der linearen Ordnung. Die weiteren Eigenschaften werden von dem Vertex $V^{latt}_{2n,r}$ bestimmt.[7] Da die normalgeordneten Monome invariant unter Permutation der Argumente sind (dies ist eine Folge der Symmetrie der Kovarianz ν), fordern wir dies auch für die Vertices:

$$\forall \pi \in S_{2n} \quad : \quad V^{latt}_{2n,r}(x_1, \ldots, x_{2n}) = V^{latt}_{2n,r}(\pi(x_1), \ldots, \pi(x_{2n})) \tag{3.14}$$

Desweiteren weisen die Vertices Gittersymmetrien auf, von denen wir besonders die Translationsinvarianz

$$\forall a \in \Lambda(a) \quad : \quad V^{latt}_{2n,r}(x_1, \ldots, x_{2n}) = V^{latt}_{2n,r}(x_1 - a, \ldots, x_{2n} - a) \tag{3.15}$$

betonen. Auf diese Weise gewähren wir die Invarianz des Potentialfunktionals gegenüber verschobenen, gedrehten oder gespiegelten Feldern.

Mittels des Operators $A^{(\infty)}$ erzeugen wir die Gitterkerne nun durch Kontinuumskerne. Grundgedanke ist die Rückführung des Problems auf die bereits in [Wie97d] gelöste perturbative Behandlung der diskreten RGT auf der euklidischen vierdimensionalen Raum-Zeit. Über $V^{cont}_{2n,r} : \bigotimes_{i=1}^{2n} \mathbb{R}^D \to \mathbb{R}$ definieren wir

$$V^{latt}_{2n,r}(x_1, \ldots, x_{2n}) = \int_{\bigotimes \mathbb{R}^D} d^D y_1 \ldots d^D y_{2n} V^{cont}_{2n,r}(y_1, \ldots, y_{2n}) \prod_{i=1}^{2n} A^{(\infty)}(y_i, x_i) \; . \tag{3.16}$$

Damit (3.14) gilt, muß auch $V^{cont}_{2n,r}$ unter Permutation der Argumente invariant sein. Die Forderung (3.15) und $A^{(\infty)}(x, y) = A^{(\infty)}(x - a, y - a)$ für $a \in \Lambda(a)$ bedingen die Translationsinvarianz der Kontinuumsvertices. Ferner existiere die Fourier-Transformierte, die sich als

$$\tilde{V}^{cont}_{2n,r}(p_1, \ldots, p_{2n}) = (2\pi)^D \sum_{Q \in \Lambda\left(\frac{2\pi}{a}\right)} \delta\left(\sum_{i=1}^{2n} p_i - Q\right) \hat{V}_{2n,r}(p_1, \ldots, p_{2n}) \tag{3.17}$$

[6] Man zeigt dies direkt mit Hilfe der Definition (B.2).

[7] Im weiteren Verlauf identifizieren wir die Begriffe Vertex, Vertexfunktion und Kern eines Vertex.

schreibt. Wir nennen $\hat{V}_{2n,r}$ einen reduzierten Impulskern. Man beachte, daß $\hat{V}_{2n,r}$ i.a. unter Permutation der Argumente nicht invariant ist, da für $\sum p_i \notin \Lambda(\frac{2\pi}{a})$ der Integrand verschwindet, und der reduzierte Impulskern somit beliebige Werte annehmen kann. Für Gesamtimpulse, die auf dem Impulsgitter $\Lambda(\frac{2\pi}{a})$ liegen, gilt jedoch die Vertauschungseigenschaft. Mit der Definition $\lim_{a\to 0} \Lambda(\frac{2\pi}{a}) = \{0\}$ werden auch die Kontinuumsvertices translationsinvariant. Für die assoziierten Kontinuumskerne im Ortsraum ergibt sich

$$\begin{aligned} & V^{cont}_{2n,r}(x_1,\ldots,x_{2n}) \qquad (3.18)\\ = & \sum_{Q\in\Lambda(\frac{2\pi}{a})} e^{iQx_{2n}} \int \frac{d^Dp_1}{(2\pi)^D}\ldots\frac{d^Dp_{2n}}{(2\pi)^D} e^{i\sum p_i(x_i-x_{2n})}\hat{V}_{2n,r}(p_1,\ldots,p_{2n})\ . \end{aligned}$$

Mittels (3.11)[8] erhält man

$$\begin{aligned} & V^{latt}_{2n,r}(x_1,\ldots,x_{2n}) \qquad (3.19)\\ \xrightarrow{\mathcal{DT}} & \int_{\bigotimes\Lambda(a)} d^Dy_1\ldots d^Dy_{2n} V^{latt}_{2n,r}(y_1,\ldots,y_{2n}) \prod_{i=1}^{2n} A(y_i,x_i)\\ \overset{(3.16)}{=} & \int_{\bigotimes\mathbb{R}^D} d^Dy_1\ldots d^Dy_{2n} V^{cont}_{2n,r}(y_1,\ldots,y_{2n}) \prod_{i=1}^{2n} A^{(\infty)}A(y_i,x_i)\\ \overset{(3.7)}{=} & \int_{\bigotimes\mathbb{R}^D} d^Dy_1\ldots d^Dy_{2n} V^{cont}_{2n,r}(y_1,\ldots,y_{2n}) \prod_{i=1}^{2n} L^{-\sigma}A^{(\infty)}\left(L^{-1}y_i,x_i\right)\\ \overset{L^{-1}y_i\to y_i}{=} & \int_{\bigotimes\mathbb{R}^D} d^Dy_1\ldots d^Dy_{2n} L^{2n(D-\sigma)} V^{cont}_{2n,r}(Ly_1,\ldots,Ly_{2n}) \prod_{i=1}^{2n} A^{(\infty)}(y_i,x_i) \end{aligned}$$

Man kann die Wirkung der linearen RGT auf eine Skalentransformation der Kontinuumskerne $V^{cont}_{2n,r}$, welche die Gitterkerne $V^{latt}_{2n,r}$ generieren, reduzieren.

$$V^{cont}_{2n,r}(x_1,\ldots,x_{2n}) \xrightarrow{\mathcal{DT}} L^{n(D+2)} V^{cont}_{2n,r}(Lx_1,\ldots,Lx_{2n}) \qquad (3.20)$$

Die Eigenfunktionen der linearisierten RGT werden durch homogene n-Punkt-Funktionen erzeugt. Ist k der Homogenitätsgrad, so ergibt sich der Eigenwert $L^{n(D+2)+k}$. Die einfachsten homogenen Kerne sind Produkte von δ-Distributionen, für die $k = -D$ ist. Ihnen entsprechen im Kontinuum normalgeordnete Produkte von Feldern, die auf dem Gitter durch A-Kerne verschmiert sind. Auch partielle Ableitungen von δ-Distributionen sind homogen

[8] Diese Gleichung gilt auch für die normierten, normalgeordneten Monome.

und liefern einen Faktor $\frac{1}{L}$ pro Ableitung nach $x_{i,j}$ ($j \in \{1, \ldots, D\}$). Im Kontinuum ist der entsprechende Eigenvektor ein normalgeordnetes Produkt aus Feldern und deren Ableitungen. Diese Elemente darf man bei der Suche nach relevanten, marginalen und irrelevanten (verschmierten) Eigenvektoren nicht vergessen.

Mit einer Definition der Impulskerne $V^{cont}_{2n,r}$ über unendlich oft differenzierbare reduzierte Kerne $\hat{V}_{2n,r}$ stellen wir sicher, daß die V_r eine Superposition aus (verschmierten) Feldern samt ihrer Ableitungen darstellen. Mehr hierzu in Kapitel 3.3.3

Im Impulsraum schreibt sich der transformierte Kern (3.20) wie folgt:

$$\begin{aligned}
& L^{n(D+2)} V^{cont}_{2n,r}(Lx_1, \ldots, Lx_{2n}) \\
\overset{Lp\to p}{=} \; & L^{D-n(D-2)} \int_{\bigotimes \mathbb{R}^D} \frac{d^D p_1 \ldots d^D p_{2n}}{(2\pi)^D \ldots (2\pi)^D} e^{i\sum_{i=1}^{2n} p_i x_i} \hat{V}_{2n,r}\left(\frac{p_1}{L}, \ldots, \frac{p_{2n}}{L}\right) \\
& \times (2\pi)^D \sum_{Q \in \Lambda\left(\frac{2\pi}{a}\right)} \delta\left(\sum_{i=1}^{2n} p_i - LQ\right) \\
= \; & L^{D-n(D-2)} \int_{\bigotimes \mathbb{R}^D} \frac{d^D p_1 \ldots d^D p_{2n}}{(2\pi)^D \ldots (2\pi)^D} e^{i\sum_{i=1}^{2n} p_i x_i} \hat{V}_{2n,r}\left(\frac{p_1}{L}, \ldots, \frac{p_{2n}}{L}\right) \\
& \times (2\pi)^D \sum_{Q \in \Lambda\left(\frac{2\pi}{a}\right)} \delta\left(\sum_{i=1}^{2n} p_i - Q\right) T\left(\sum_{i=1}^{2n} p_i\right) \qquad (3.21)
\end{aligned}$$

wobei

$$T : \mathbb{R}^D \to \mathbb{R} \qquad T\Big|_{\Lambda\left(\frac{2\pi}{a}\right)}(Q) = \begin{cases} 1 & Q \in \Lambda\left(\frac{2\pi}{a} L\right) \\ 0 & \text{sonst} \end{cases} \qquad (3.22)$$

Die ϕ^4-Trajektorie ist durch die vorgegebene lineare Kopplungsparameterordnung

$$V^{cont}_{2,1}(x_1, x_2) = 0 \qquad (3.23)$$

$$V^{cont}_{4,1}(x_1, x_2, x_3, x_4) = \prod_{i=2}^{4} \delta(x_1 - x_i) \qquad (3.24)$$

definiert. Diese Anfangsbedingung gilt auch für das Kontinuum. Die perturbative Behandlung der Gleichung $\mathcal{T}(V)(\phi, g) = V(\phi, \beta(g))$ verläuft wie in

Abschnitt 2.3. Die nach Kumulanten entwickelte RGT schreibt sich als[9]

$$\mathcal{T}(V)(\phi, g) = \sum_{r=1}^{\infty} \left\{ \sum_{m=1}^{r} \frac{(-1)^{m+1}}{m!} \sum_{\sum_{i=1}^{m} r_i = r} \langle [V_{r_1}, \ldots, V_{r_m}] \rangle_{\Gamma, A\phi}^{T} \right\} g^r \ . \quad (3.25)$$

$V(\phi, \beta(g))$ gleicht (2.28). Die in erster Ordnung entstehenden Eigenwertgleichungen legen ob (3.23) und (3.24) b_1 eindeutig fest. Der 4-Punkt-Vertex besitzt den Homogenitätsgrad $-3D$, so daß mit (3.20) wie schon in (2.30)

$$b_1 = L^{4-D} \quad (3.26)$$

folgt. Wir legen eine ganz bestimmte Parametrisierung der Trajektorie durch die Wahl der linearen β-Funktion fest (vgl. (2.38)), und erhalten mit den Ersetzungen $\alpha \Leftrightarrow 1$, $\gamma \Leftrightarrow \Gamma$, $\beta \Leftrightarrow A$ und $K_r(V) \Leftrightarrow -K_r(V)$ aus (2.39) für $r \geq 2$ die Bestimmungsgleichungen

$$\begin{aligned} & L^{n(D+2)-r(4-D)} V_{2n,r}^{cont}(Lx_1, \ldots, Lx_{2n}) - V_{2n,r}^{cont}(x_1, \ldots, x_{2n}) \\ = \ & L^{-r(4-D)} K(V)_{2n,r}^{cont}(x_1, \ldots, x_{2n}) \ . \end{aligned} \quad (3.27)$$

Hierbei ist $K(V)_{2n,r}^{cont}(x_1, \ldots, x_{2n})$ der erzeugende Kontinuumskern des $:\phi(x_1)\ldots\phi(x_{2n}):$-Anteils im vollständig bestimmten Term $K(V)_r$. (3.27) lösende Vertices erfüllen die (entsprechend substituierte) Gleichung (2.39), sind jedoch nicht eindeutig. Für die reduzierten Impulskerne ergibt sich

$$\begin{aligned} & L^{\sigma(n,r)} T \left(\sum_{i=1}^{2n} p_i \right) \hat{V}_{2n,r} \left(\frac{p_1}{L}, \ldots, \frac{p_{2n}}{L} \right) - \hat{V}_{2n,r}(p_1, \ldots, p_{2n}) \\ = \ & L^{-r(4-D)} \hat{K}(V)_{2n,r}(p_1, \ldots, p_{2n}) \end{aligned} \quad (3.28)$$

mit

$$\sigma(n, r) = D - n(D-2) - r(4-D) \ . \quad (3.29)$$

Der Grund für die Berechnung im Impulsraum liegt im besseren *power counting*. Wie wir noch sehen werden, bereitet die Berechnung der Vertices Probleme, deren Exponent $\sigma \geq 0$ ist. In der Impulsdarstellung ist deren Anzahl für $2 < D < 4$ endlich. Für $D = 3$ ergibt sich

$$\sigma(n, r) = 3 - n - r \ . \quad (3.30)$$

Da die Bestimmungsgleichungen erst für $r \geq 2$ gelten, und $\mathcal{T}$ normiert ist ($n > 0$), existiert kein relevanter ($\sigma > 0$) Vertex. Der Massenvertex $\hat{V}_{2,2}$ ist marginal ($\sigma = 0$) und alle übrigen Impulskerne sind irrelevant ($\sigma < 0$).

[9] Man beachte, daß es sich im folgenden immer um normierte Kumulanten handelt, d.h. $\langle [V_{r_1}, \ldots, V_{r_m}] \rangle_{\Gamma, A\phi}^{T} \hat{=} \langle [V_{r_1}, \ldots, V_{r_m}] \rangle_{\Gamma, A\phi}^{T} - \langle [V_{r_1}, \ldots, V_{r_m}] \rangle_{\Gamma, 0}^{T}$.

Man beachte die Äquivalenz von (3.29) und (2.42). Die HRG bestätigt ihren Ruf als Testfeld für das volle Modell.

Eine wichtige Eigenart der Störungstheorie auf dem Gitter ist, daß die Vertices $V_{2n,r}^{cont}$ gegenüber Kontinuumstranslationen nicht invariant sind, so daß eine äquivalente Behandlung gemäß [Wie97d] nicht möglich ist. Begründet liegt dies darin, daß die Operatoren A und Γ nur gittertranslationsinvariant sind und in die $K(V)_r$ einfließen.

Desweiteren wollen wir betonen, daß die mittels (3.28) bestimmten reduzierten Impulskerne zwar T-abhängig sind, durch die extrahierte δ-Distribution jedoch nur die Werte $\sum p_i \in \Lambda(\frac{2\pi}{a})$ zur ϕ^4-Trajektorie beitragen. Alle anderen Impulskonstellationen sind auf dem Gitter irrelevant und somit frei wählbar. Bei der Verwendung einer kontinuierlichen T-Funktion erübrigen sich jedoch Fallunterscheidungen im Definitionsbereich.

3.3 Explizite Berechnung reduzierter Impulskerne

Ziel dieses Paragraphen ist es, eine analytische Funktion T mit den Eigenschaften (3.22) zu finden, die eine explizite Berechnung der reduzierten Impulskerne $\hat{V}_{2n,r}$ ermöglicht.

Mit Hilfe der Funktion [Wal]

$$\hat{S} : \mathbb{R} \to \mathbb{R} \qquad \hat{S} = \begin{cases} \exp\left(-\frac{\left(\frac{a}{2\pi}x\right)^2}{1-\left(\frac{a}{2\pi}x\right)^2}\right) & |x| < \frac{2\pi}{a} \\ 0 & \text{sonst} \end{cases} \tag{3.31}$$

aus dem Raum der Testfunktionen[10] definieren wir

$$S : \mathbb{R} \to \mathbb{R} \qquad S(x) = \sum_{k\in\mathbb{Z}} \hat{S}\left(x + \frac{2\pi}{a}Lk\right) \tag{3.32}$$

$$T : \mathbb{R}^D \to \mathbb{R} \qquad T(p) = \prod_{\mu=1}^{D} S(p_\mu)\ . \tag{3.33}$$

Die auf diese Weise konstruierte T-Funktion ist unendlich oft differenzierbar und erfüllt die Eigenschaft (3.22). Siehe hierzu auch Abbildung 3.1.

[10] Das ist der Raum aller finiten (=kompakter Träger), unendlich oft differenzierbaren Funktionen über $\mathbb{R}$.

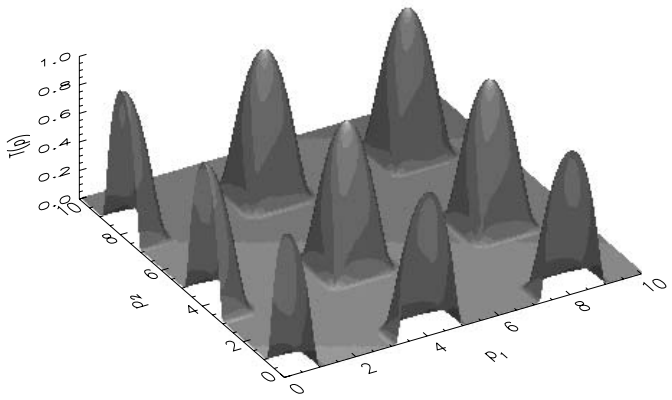


Abbildung 3.1: Eine 3-dimensionale Darstellung der T-Funktion für $D = 2$,$a = 2\pi$ und $L = 4$.

3.3.1 Berechnung der irrelevanten Kerne

Wir beginnen mit dem irrelevanten Fall, d.h. $\sigma := \sigma(n, r) < 0$. Gleichung (3.28)[11] wird dann von

$$\hat{V}(p_1, \ldots, p_{2n}) = -\sum_{k=0}^{\infty} L^{k\sigma} T_k(\sum_{i=1}^{2n} p_i) \hat{K}(V)(\frac{p_1}{L^k}, \ldots, \frac{p_{2n}}{L^k}) \tag{3.34}$$

eindeutig gelöst. Wir setzen voraus, daß

$$\hat{K}(V) \in \mathcal{C}^{\infty}(\mathbb{R}^{2nD}) \tag{3.35}$$

und benutzen die Definition

$$T_k(Q) := \prod_{m=0}^{k-1} T\left(\frac{Q}{L^m}\right) . \tag{3.36}$$

Beweis: (3.28) ist eine Gleichung der Form $\hat{V}(p) = F(\hat{V}(p/L))$. Durch sukzessives Einsetzen erhält man für $S \in \mathbb{N}$ die Aussage

$$\hat{V}(p) \quad = \quad -\sum_{k=0}^{S-1} L^{k\sigma} T_k(\sum_{i=1}^{2n} p_i) \hat{K}(V)(\frac{p_1}{L^k}, \ldots, \frac{p_{2n}}{L^k})$$

[11] Wir vernachlässigen im folgenden die Indizes $(2n, r)$ und den Vorfaktor des $\hat{K}(V)$-Anteils

$$+L^{S\sigma}T_S(\sum_{i=1}^{2n}p_i)\hat{V}(\frac{p_1}{L^S},\ldots,\frac{p_{2n}}{L^S})\ . \tag{3.37}$$

Man zeigt dies explizit mit dem Prinzip der vollständigen Induktion. Unter der Annahme, daß $\hat{V}$ stetig ist,[12], folgt aus $\sup_{\mathbb{R}^D}|T|=1$ und der Wahl eines genügend großen S

$$\left|L^{S\sigma}T_S(\sum_{i=1}^{2n}p_i)\hat{V}(\frac{p_1}{L^S},\ldots,\frac{p_{2n}}{L^S})\right| \leq 2L^{S\sigma}\left|\hat{V}(0,\ldots,0)\right| \overrightarrow{\scriptstyle s\to\infty}\ 0\ . \tag{3.38}$$

Der Beweis der Eindeutigkeit läuft analog, denn die Differenz $\hat{V}_D$ zweier Lösungen erfüllt die homogene Differenzengleichung, d.h. $\hat{K}(V)=0$, so daß man $\hat{V}_D=0$ erhält.

Desweiteren ist die Reihendarstellung (3.34) gleichmäßig konvergent, was man auf ähnliche Art und Weise wie zuvor mit Hilfe des Majorantenkriteriums beweist: Es sei $K\subset\bigotimes_{i=1}^{2n}\mathbb{R}^D$ kompakt. OBdA sei K eine Kugel, so daß für alle $(p_1,\ldots,p_{2n})\in K$ und $k\in\mathbb{N}_0$ die Relation $\frac{p_i}{L^k}\in K$ erfüllt ist. Dann gilt

$$\left|L^{k\sigma}T_k(\sum_{i=1}^{2n}p_i)\hat{K}(V)(\frac{p_1}{L^k},\ldots,\frac{p_{2n}}{L^k})\right| \leq (L^\sigma)^k\sup_K|\hat{K}(V)|\ . \tag{3.39}$$

Da die Summanden stetig sind, ist dies sogleich ein Beweis für die Stetigkeit der reduzierten Kerne. Mit Hilfe der Eigenschaft $\hat{V},T_k\in\mathcal{C}^\infty(\mathbb{R}^{2nD})$ und des 3. Vertauschungssatzes [For91] erarbeiten wir ebenso $\hat{V}\in\mathcal{C}^\infty(\mathbb{R}^{2nD})$.

3.3.2 $\hat{V}\in\mathcal{C}^\infty(\mathbb{R}^{2nD})$

Nun haben wir die Eigenschaft $\hat{V}\in\mathcal{C}^\infty(\mathbb{R}^{2nD})$ aus der Voraussetzung (3.35) abgeleitet. Daß dies gerechtfertigt war, zeigt man per Induktion über der Ordnung r. Aus der Gittertranslationsinvarianz der Vertices $V^{cont}_{2n,r}$ folgt für die assoziierten, reduzierten Impulskerne ($k\in\{1,\ldots,2n\}$ beliebig)

$$\hat{V}_{2n,r}(p_1,\ldots,p_{2n})=\int\prod_{\substack{i=1\\i\neq k}}^{2n}d^Dx_i\frac{1}{a^D}\int_{\left[-\frac{a}{2},\frac{a}{2}\right]^D}d^Dx_k e^{-i\sum_{i=1}^{2n}p_ix_i}V^{cont}_{2n,r}(x_1,\ldots,x_{2n})\ . \tag{3.40}$$

[12] Beweis folgt.

Diese Formel zeigt man unter Benutzung der Relation [Pur96]

$$\sum_{y\in\Lambda(a)} e^{-iyp} = \sum_{Q\in\Lambda\left(\frac{2\pi}{a}\right)} \frac{2\pi}{a}\delta(p-Q)\ . \tag{3.41}$$

Aus (3.40) erhält man in erster Ordnung[13]

$$\hat{V}_{2,1}(p_1,p_2) = 0 \tag{3.42}$$

$$\hat{V}_{4,1}(p_1,p_2,p_3,p_4) = \prod_{\mu=1}^{D} \mathrm{si}\left(\frac{ap_\mu}{2}\right)\Big|_{p=\sum_{i=1}^4 p_i}\ . \tag{3.43}$$

Da si $\in \mathcal{C}^\infty(\mathbb{R})$, ist der Induktionsanfang ($r = 1$) komplett. Man beachte, daß für $a = 0$ die Kontinuumseigenschaft $\hat{V}_{4,1}(p_1,\ldots,p_4) = 1$ folgt.[14]

Für den Induktionsschritt machen wir uns die Mühe, das Gitterpotential ins Kontinuum zu transferieren, um es dort von einer ebenfalls in das Kontinuum expandierten RGT bearbeiten zu lassen. Abschließend führen wir eine Gitterrücktransformation durch. Siehe hierzu auch [Wie98]. Diese Prozedur ermöglicht es uns, viele Ergebnisse der Kontinuumstheorie zu benutzen. Einziges Problem ist die nicht vorhandene Kontinuumstranslationsinvarianz der auf das Kontinuum gebrachten Fluktuationskovarianz und des kontinuierlichen Propagators. Ohne dieses Handicap könnte man die RG auf dem Gitter vollständig auf das gelöste Problem im Kontinuum zurückführen.

Es sei nun O ein Gitteroperator $\mathcal{H}(a) \to \mathcal{H}(a)$. Dann definieren wir den korrespondierenden Kontinuumsoperator $\mathcal{H}(0) \to \mathcal{H}(0)$ über

$$O^{(\infty)} = A^{(\infty)} O A^{(\infty)^\dagger}\ . \tag{3.44}$$

Wir erhalten somit $\nu^{(\infty)}$ und $\Gamma^{(\infty)}$, welche die Eigenschaft

$$\nu^{(\infty)} - \Gamma^{(\infty)} = S^{-1}\nu^{(\infty)}S^{-1^\dagger} \tag{3.45}$$

erfüllen. Die Fluktuationskovarianz ist eine nach Skalen zerlegte Kovarianz $\nu^{(\infty)}$. In dieser Gleichung finden wir das Grundprinzip der RG wieder.

Definieren wir nun die kontinuierliche RGT R^{cont} über $T^{(\infty)}$ und S^{-1} und die ϕ^4-Trajektorie V^{cont} analog zu (3.13), indem wir kontinuierliche Felder bezüglich der normalordnenden Kovarianz $\nu^{(\infty)}$ und durch die kontinuierlichen Impulskerne $\hat{V}^{cont}_{2n,r}$ erzeugte Ortskerne benutzen, so gilt

$$V(\phi) = V^{cont}(A^{(\infty)}\phi) \tag{3.46}$$

[13] $\mathrm{si}(x) = \frac{\sin(x)}{x}$

[14] In diesem Fall existiert nur $\sum p_i = 0$.

und folglich[15]

$$R(V)(\phi) = R^{cont}(V^{cont})(A^{(\infty)}\phi) \; . \tag{3.47}$$

Weitere Informationen finden sich bei [Wie98, GK84]. Aufgrund der Multilinearität der Kumulanten sind die $K_r(V)$ eine Superposition von trunkierten Erwartungswerten normalgeordneter Felder, deren Kerne Produkte der bereits berechneten $\mathcal{C}^\infty$-Kerne $V_{2n,r}$ sind. In die trunkierten Erwartungswerte fließen nun die normalordnende und die Fluktuationskovarianz ein. Nach WIECZERKOWSKI [Wie98] ergibt sich z.B. für $\nu^{(\infty)} = \nu_a^{(\infty)}$

$$\hat{\nu}_a^{(\infty)}(p_1,p_2) = -\left\{ \sum_{P\in\Lambda(\frac{2\pi}{a})} \frac{p_1^2 p_2^2}{(p_1+P)^2} \prod_{\mu=1}^{D} \frac{(p_1)_\mu (p_2)_\mu}{(p_1+P)^2{}_\mu} \right\}^{-1} \; . \tag{3.48}$$

Für $p_1 + p_2 \in \Lambda(\frac{2\pi}{a})$ ergibt sich nach Substitution von P die Symmetrie

$$\hat{\nu}_a^{(\infty)}(p_1,p_2) = \hat{\nu}_a^{(\infty)}(p_2,p_1) \; . \tag{3.49}$$

Der reduzierte Impulskern ist singulär. Dies ist nicht verwunderlich, da auch schon die perfekte masselose Gitterkovarianz für $p = 0$ divergent war. Für $a \to 0$ erhalten wir den freien Propagator $\nu(p_1,p_2) = -p_2^2$.

Diese infrarote Divergenz zerstört das Argument der unendlichen Differenzierbarkeit jedoch nicht. Im Kontinuum trat $\nu^{(\infty)}$ nur in Impulsraum-Faltungen mit $\Gamma^{(\infty)}$ auf. Die durch einen Exponential-*cutoff* regularisierte Fluktuationskovarianz vererbt ihre Beschränktheit in Null samt Differenzierbarkeit auf die Faltung, so daß die Kerne der Kumulanten unendlich oft differenzierbar bleiben [Wie97d].

Auf dem Gitter folgt mit (3.45), daß $\Gamma^{(\infty)}$ auf dem feineren Gesamtimpulsgitter $\Lambda(\frac{2\pi}{aL})$ lebt. Man berechnet

$$\Gamma^{(\infty)}(p_1,p_2) = \sum_{Q\in\Lambda(\frac{2\pi}{aL})} \delta(p_1+p_2+Q)\hat{\Gamma}^{(\infty)}(p_1,p_2) \tag{3.50}$$

mit[16]

$$\hat{\Gamma}^{(\infty)}(p_1,p_2) = T(p_1+p_2)\hat{\nu}_a^{(\infty)}(p_1,p_2) - L^2\hat{\nu}_a^{(\infty)}(Lp_1,Lp_2) \; . \tag{3.51}$$

[15] Das renormierte Potential ist ein Kontinuumspotential, dessen Impulskerne die renormierten Gitterkerne generieren.

[16] Man beachte, daß $\hat{\nu}_{La}^{(\infty)}(p_1,p_2) = L^2\hat{\nu}_a^{(\infty)}(Lp_1,Lp_2)$.

T sei eine $\mathcal{C}^\infty$-Funktion entsprechend (3.22) mit der Ersetzung $a \to La$. Für $p_1 + p_2 \in \Lambda(\frac{2\pi}{a})$ folgt:

$$\hat{\Gamma}^{(\infty)}(p_1, p_2) = (1 - L^2)\frac{1}{p_2^2} C_L(p_1, p_2) \tag{3.52}$$

C_L regularisiert den freien Propagator $\frac{1}{p_2^2}$. Leider können wir dies nicht zeigen. In Analogie zum Kontinuum, wo man den freien Propagator z.B. mit dem exponentiellen IR- und UV-Regulator

$$C_L^{cont}(p) = e^{-p^2} - e^{-(Lp)^2} = (L^2 - 1)p^2 + O(p^4) \tag{3.53}$$

versieht, sollte auch hier die Relation $C_L(p_1, p_2) = O(p_2)$ gelten. Ein weiteres Problem finden wir für $p_1 + p_2 \in \Lambda(\frac{2\pi}{La}) - \Lambda(\frac{2\pi}{a})$. Für diese Impulspaare verschwindet die T-Funktion, und es bleibt die divergente, reskalierte normalordnende Kovarianz zurück. Das Zeigen der Polfreiheit der Fluktuationskovarianz ist eine reizvolle Aufgabe, der wir uns hier jedoch nicht widmen wollen[17].

Wir gehen im weiteren davon aus, daß $\hat{\Gamma}^{(\infty)} \in \mathcal{C}^\infty(\mathbb{R}^D \times \mathbb{R}^D)$. Aus

$$K_2(V)(\phi) = -\left\langle V^{cont}, V^{cont} \right\rangle^T_{\Gamma^{(\infty)}, S^{-1}A^{(\infty)}\phi} \tag{3.54}$$

ergibt sich mit $u^{(\infty)}(x, y) = L^{2-D}\nu^{(\infty)}(\frac{x}{L}, \frac{y}{L})$ und der Kumulantenformel (B.20)

$$\begin{aligned} & K_{4,2}^{cont}(x_1, \ldots, x_4) \qquad (3.55) \\ = \; & 3L^4\delta(x_1 - x_2)\delta(x_3 - x_4)\Gamma^{(\infty)}(Lx_1, Lx_3)^2 \\ & +6L^4\delta(x_1 - x_2)\delta(x_3 - x_4)\Gamma^{(\infty)}(Lx_1, Lx_3)u^{(\infty)}(Lx_1, Lx_3)\ . \end{aligned}$$

$$\begin{aligned} & K_{6,2}^{cont}(x_1, \ldots, x_6) \qquad (3.56) \\ = \; & 20L^{6-D}\delta(x_1 - x_2)\delta(x_1 - x_3)\delta(x_4 - x_5)\delta(x_4 - x_6)\Gamma^{(\infty)}(\frac{x_1}{L}, \frac{x_4}{L})\ . \end{aligned}$$

Den aus drei Summanden bestehenden Massenvertex haben wir der Einfachheit halber weggelassen. Die reduzierten Impulskerne berechnen wir unter Benutzung von (3.40) zu

$$\begin{aligned} & \hat{K}_{4,2}(p_1, \ldots, p_4) \qquad (3.57) \\ = \; & L^{4-D} \sum_{S \in \Lambda(\frac{2\pi}{a}L)} \prod_{\mu=1}^{D} \mathrm{si}\left(\frac{aF_\mu}{2}\right)\Big|_{F = S - \sum_{i=1}^4 p_i} \sum_{Q \in \Lambda(\frac{2\pi}{a}L)} \int \frac{d^D p}{(2\pi)^D} \end{aligned}$$

[17] können

$$3\hat{\Gamma}^{(\infty)}(p,\frac{Q}{L}-p)\hat{\Gamma}^{(\infty)}(\frac{S-p_1-p_2}{L}-p,\frac{p_1+p_2-Q}{L}+p)$$
$$+6\hat{\Gamma}^{(\infty)}(p,\frac{Q}{L}-p)\hat{u}^{(\infty)}(\frac{S-p_1-p_2}{L}-p,\frac{p_1+p_2-Q}{L}+p)$$

und

$$\hat{K}_{6,2}(p_1,\ldots,p_6) \tag{3.58}$$
$$= 20L^{6-2D}\sum_{Q\in\Lambda(\frac{2\pi}{a}L)}\prod_{\mu=1}^{D}\mathrm{si}\left(\frac{aF_\mu}{2}\right)\Big|_{F=Q-\sum_{i=1}^{6}p_i}\hat{\Gamma}^{(\infty)}\left(\sum_{i=1}^{3}\frac{p_i}{L},\frac{Q}{L}-\sum_{i=1}^{3}\frac{p_i}{L}\right).$$

Für $a\to 0$ ergibt sich wieder das Kontinuumsresultat. Wir beweisen die $\mathcal{C}^\infty$-Eigenschaft für $\hat{K}_{6,2}$ und beginnen mit dem Zeigen von gleichmäßiger Konvergenz. Für jedes Kompaktum $K\subsetneq\mathbb{R}^{6D}$ existiert ein $R>0$, so daß $|\hat{\Gamma}^{(\infty)}|$ aufgrund seines ultravioletten *cutoffs* für $|Q|>R$ beschränkt ist. Dies gilt auch für alle partiellen Ableitungen.[18] Wir reduzieren unsere Betrachtungen somit auf die si-Reihe mit positiven Indizes und $a=D=1$. Die reskalierte Summe $x=\frac{a}{2}\sum p_i$ sei durch M beschränkt. OBdA seien $Ln\pi-M>0$ und L ungerade. Dann gilt unter Benutzung eines Additionstheorems:

$$\left|\sum_{n\in\mathbb{N}}\mathrm{si}(Ln\pi-x)\right|\le\left|\sum_{n\in\mathbb{N}}\frac{(-1)^{Ln}}{Ln\pi-x}\right|\le\sum_{n\in\mathbb{N}}\frac{(-1)^n}{Ln\pi-(-1)^nM}<\infty \tag{3.59}$$

Die Konvergenz der Majorante folgt aus dem Leibniz-Kriterium, und Stetigkeit liegt auf der Hand. Da si eine $\mathcal{C}^\infty$-Funktion ist, deren Ableitungen sich ebenso abschätzen lassen, folgt unendliche Differenzierbarkeit für den $\hat{K}_{6,2}$- Vertex.

Die erweiterte Faltung der 4-Punkt-Funktion $\hat{K}_{4,2}$ wirkt da schon etwas komplizierter. Für $a\to 0$ erhalten wir mit Hilfe der Identifikation $\hat{\Gamma}^{(\infty)}(p)=\hat{\Gamma}^{(\infty)}(-p)=\hat{\Gamma}^{(\infty)}(p,-p)$ das Kontinuumsresultat

$$\hat{K}_{4,2}^{cont}(p_1,\ldots,p_4)=\frac{L^{4-D}}{(2\pi)^D}\left\{3\hat{\Gamma}^{(\infty)}\star\hat{\Gamma}^{(\infty)}(\frac{p_1+p_2}{L})+6\hat{\Gamma}^{(\infty)}\star\hat{u}^{(\infty)}(\frac{p_1+p_2}{L})\right\}. \tag{3.60}$$

Hierbei bezeichnet $\star$ die Faltung. Die Berechnung von $\hat{K}_{4,2}^{cont}$ findet sich in [Wie97d]. Den Beweis der Analytizität auf dem Gitter bleiben wir schuldig.

3.3.3 Eigenschaft differenzierbarer Impulskerne $\hat{V}$

Wir betrachten in diesem Kapitel die kontinuierliche, gittertranslationsinvariante RGT und zeigen, daß unsere Wahl der Potentialdarstellung über

[18] Natürlich mit einem anderen R.

normalgeordnete Felder äquivalent zu einer Konstruktion ist, die auch normalgeordnete Produkte abgeleiteter Felder in die Superposition involviert. Neben der Gittertranslationsinvarianz bildet die unendliche Differenzierbarkeit der Impulskerne die Grundlage dieser Aussage.

Wir betrachten im folgenden $\hat{V}$ als Funktion von $p_1, \ldots, p_{2n-1}$ und Q und bilden die formale Taylor-Reihe.[19] Durch Einsetzen in (3.18) erhalten wir für einen Kontinuumskern

$$\begin{aligned} & V(x_1, \ldots, x_{2n}) \qquad (3.61) \\ = & \sum_k \frac{(-i)^{|k|}}{k!} \partial_p^k \hat{V}(0) \prod_{i=1}^{2n-1} \partial_{x_i}^{k_i} \delta(x_i - x_{2n}) \int_{\Lambda(a)} d^D y \partial_{x_{2n}}^{k_{2n}} \delta(x_{2n} - y) \, . \end{aligned}$$

Hierbei benutzen wir den Multiindex $k = k_{m,\mu}$ mit $m \in \{1, \ldots, 2n\}$ und $\mu \in \{1, \ldots, D\}$. $\partial_{x_i}^{k_i}$ steht somit für $\prod_{\mu=1}^{D} \partial_{x_{i,\mu}}^{k_{i,\mu}}$. Man beachte, daß $\partial_{p_{2n}}^{k_{2n}} = \partial_Q^{k_{2n}}$. Mit (3.61) und partieller Integration folgt

$$\begin{aligned} & \int_{\bigotimes_{i=1}^{2n} \mathbb{R}^D} \prod_{i=1}^{2n} d^D x_i V(x_1, \ldots, x_{2n}) : \phi(x_1) \ldots \phi(x_{2n}) :_{\nu^{(\infty)}} \\ = & \sum_k \frac{i^{|k|}}{k!} \partial_p^k \hat{V}(0) \int_{\Lambda(a)} d^D y : \partial_y^k \phi(y) :_{\nu^{(\infty)}} \quad . \qquad (3.62) \end{aligned}$$

Hierbei gilt $: \partial_y^k \phi(y) :=: \prod_{i=1}^{2n} \partial_{x_i}^{k_i} \phi(x_i) :_{x_i = y}$.

An dieser Darstellung sieht man sehr schön, daß die ϕ^4-Trajektorie durch die $\partial_p^k \hat{V}_{2n,r}(0)$ vollständig festgelegt ist und in der linearen Hülle der $: \partial_y^k \phi(y) :_{\nu^{(\infty)}}$, $y \in \Lambda(a)$ lebt.

Die Anfangsbedingungen (3.23) und (3.24) schreiben sich entsprechend:[20]

$$\hat{V}_{2,1}(0,0) = 0 \qquad (3.63)$$

$$\hat{V}_{4,1}(0,0,0,0) = 1 \qquad (3.64)$$

3.3.4 Berechnung der marginalen Kerne

Im marginalen Fall ($\sigma = 0$) erhält man aus der Bestimmungsgleichung (3.28) durch Einsetzen von $p = 0$ die Bedingung

$$\hat{K}(V)(0) = 0 \, . \qquad (3.65)$$

[19] Einen exakten Zugang bietet die Darstellung über Taylor-Polynom und Restglied.

[20] Für $|k| = 0$ ist $\hat{V}(p_1 = 0, \ldots, p_{2n-1} = 0, Q = 0) = \hat{V}(p_1 = 0, \ldots, p_{2n} = 0)$.

Wie wir im folgenden sehen werden, ist gerade diese Gleichung für $D = 3$ nicht erfüllt, wir beheben das Problem jedoch wie schon im hierarchischen Bild durch eine Doppelentwicklung.

Die Lösung von (3.28) bestimmt sich analog (3.37) zu[21]

$$\hat{V}(p_1, \ldots, p_{2n}) = \hat{V}(0) - \sum_{k=0}^{\infty} T_k(\sum_{i=1}^{2n} p_i)\hat{K}(V)(\frac{p_1}{L^k}, \ldots, \frac{p_{2n}}{L^k}) \; . \tag{3.66}$$

Sie ist, wie man leicht zeigt, bis auf die frei wählbare Konstante $\hat{V}(0)$ eindeutig. Desweiteren gilt $\hat{V} \in \mathcal{C}^{\infty}(\mathbb{R}^{2nD})$.

Beweis: Die unendliche Reihe in (3.66) ist gleichmäßig konvergent, denn für Impulse p_i aus einer beliebigen, kompakten Kugel $\overline{U_R(0)}$ gilt (Multiindex!):

$$\left| T_k(\sum_{i=1}^{2n} p_i)\hat{K}(V)(\frac{p_1}{L^k}, \ldots, \frac{p_{2n}}{L^k}) \right| \leq \left| \nabla\hat{K}(V)(\zeta)\frac{p}{L^k} \right| \leq \sup_{\overline{U_R(0)}} \hat{K}(V)\frac{R}{L^k} \tag{3.67}$$

Die Eigenschaft (3.65) ermöglicht die Extraktion der Majorante $\frac{1}{L^k}$. Direkte Folge der gleichmäßigen Konvergenz ist die Stetigkeit von $\hat{V}_{2n,r}$. Für partielle Ableitungen ergibt sich die Majorante direkt durch den Faktor L^{-k} der inneren Funktion $L^{-k}p$.

□

3.3.5 Berechnung der nicht-relevanten Kerne

Obwohl relevante Kerne in $D = 3$ Dimensionen nicht auftreten, wollen wir die Problemstellung kurz behandeln. Das vorgestellte Verfahren greift auch im marginalen Fall.

Wir gehen davon aus, daß $\hat{V}$ Taylor-entwickelbar ist. Unter Benutzung der Multiindexschreibweise erhalten wir durch das Bilden der partiellen Ableitung ∂_p^s der Differenzengleichung[22]

$$\sum_{k=0}^{|s|} L^{\sigma-k} \sum_{|v|=k, v\subseteq s} \left(\partial_p^v \hat{V}\right)\left(\frac{p}{L}\right)\left(\partial_p^{s-v} T\right)(p) - \partial_p^s \hat{V}(p) = \partial_p^s \hat{K}(V)(p) \; . \tag{3.68}$$

[21] Der marginale reduzierte Kern läßt sich für große Impulse logarithmisch abschätzen, d.h. $|\hat{V}(p)| \leq A + B\log|p|$ [Wie97b].

[22] Wir schreiben im folgenden $T(p)$ für $T(\sum_{i=1}^{2n} p_{i,1}, \ldots, \sum_{i=1}^{2n} p_{i,D})$. Da die innere Ableitung eins ist, treten beim Ableiten keine Probleme auf.

Da s und v Multiindizes sind, muß der Ausdruck $s - v$ mengentheoretisch interpretiert werden. Mit $p = 0$ folgt aus $T(0) = 1$ für $|s| \neq \sigma$

$$\partial_p^s \hat{V}(0) = \frac{1}{L^{\sigma-s} - 1} \left\{ \partial_p^s \hat{K}(V)(0) - \sum_{k=0}^{|s|-1} L^{\sigma-k} \sum_{|v|=k, v \subseteq s} \left(\partial_p^v \hat{V} \right)(0) \left(\partial_p^{s-v} T \right)(0) \right\}. \tag{3.69}$$

Für Ableitungen der Ordnung σ muß die Bedingung

$$\partial_p^s \hat{K}(V)(0) = \sum_{k=0}^{|s|-1} L^{\sigma-k} \sum_{|v|=k, v \subseteq s} \left(\partial_p^v \hat{V} \right)(0) \left(\partial_p^{s-v} T \right)(0) \tag{3.70}$$

erfüllt sein. Anders als im hierarchischen Modell tritt diese Forderung jedoch nicht nur bei den marginalen, sondern auch bei den relevanten (n, r)-Tupeln auf. $\partial_p^s \hat{V}(0)$ $(|s| = \sigma)$ wird in Folge der gewählten Parametrisierung (bis heute die lineare β-Funktion) frei wählbar.

Ein großer Nachteil im Vergleich zu den Lösungen (3.34) und (3.66) ist jedoch, daß wir keine Aussage darüber machen können, ob die Taylor-Reihe, die durch (3.69) definiert wird, konvergiert. Obwohl wir vermuten, daß der Faktor $L^{\sigma-k}$ für $s \gg \sigma$ die Konvergenz sicherstellen wird, ist es uns bis dato nicht gelungen zu zeigen, daß das Restglied für $s \to \infty$ verschwindet.

Durch die künstlich eingeführte Funktion T ist es nicht mehr möglich, wie in [Wie97d] durch partielles Ableiten den Grad L^σ der Differenzengleichung zu verringern *und* Forminvarianz zu wahren. Diese Eigenschaft ermöglicht eine Bestimmung der Taylor-Koeffizienten $\partial_p^k \hat{V}(0)$ für $|p| \leq \sigma$ und die Berechnung des Restgliedes durch (3.34) aus einer Bestimmungsgleichung mit renormierten $\sigma < 0$ ($T(p) = 1$ im Kontinuum).

Aus diesem Grund wollen wir abschließend ein letztes Verfahren zur Bestimmung der Impulsvertices vorstellen, welches sich auf [Wie97d] stützt, doch weniger explizit ist.

3.4 Implizite Berechnung reduzierter Impulskerne

Es sei wieder $K_r(V) \in \mathcal{C}^\infty(\mathbb{R}^{2nD})$ vorausgesetzt. Wir betrachten für beliebiges $Q \in \mathbb{R}^D$ die Menge

$$\mathcal{M}_{2n}(Q, \epsilon) = \left\{ (p_1, \ldots, p_{2n}) \middle| \| \sum_{i=1}^{2n} p_i - Q \|_2 \leq \epsilon \right\} \tag{3.71}$$

und definieren

$$U_{2n}(\epsilon) = \bigcup_{Q\in\Lambda\left(\frac{2\pi}{a}L\right)} \mathcal{M}_{2n}(Q,\epsilon) \tag{3.72}$$

$$N_{2n}(\epsilon) = \bigcup_{Q\in\Lambda\left(\frac{2\pi}{a}\right)} \mathcal{M}_{2n}(Q,\epsilon) - U_{2n}(\epsilon)\ . \tag{3.73}$$

Dann ist die T-Funktion gemäß (3.22) auf $U_{2n}(0)$ identisch eins und verschwindet für Impulse aus $N_{2n}(0)$. (3.28) gleicht somit auf $U_{2n}(0)$ der Bestimmungsgleichung in [Wie97d]. Wir erweitern diese Gleichung auf R^{2nD} und lösen sie gemäß Wieczerkowski. Die zugehörige Lösung sei $\hat{V}^{(W)}_{2n,r}$. Für $(p_1,\ldots,p_{2n} \in N_{2n}(\epsilon))$ ergibt sich direkt

$$\hat{V}_{2n,r}(p_1,\ldots,p_{2n}) = \hat{K}(V)_{2n,r}(p_1,\ldots,p_{2n})\ . \tag{3.74}$$

Wir suchen nun eine $\mathcal{C}^\infty$-Funktion $\hat{V}_{2n,r}$ mit den Eigenschaften

$$\hat{V}_{2n,r}\Big|_{U_{2n}(\epsilon)} = \hat{V}^{(W)}_{2n,r}\Big|_{U_{2n}(\epsilon)} \tag{3.75}$$

$$\hat{V}_{2n,r}\Big|_{N_{2n}(\epsilon)} = \hat{K}(V)_{2n,r}(p_1,\ldots,p_{2n})\Big|_{N_{2n}(\epsilon)}\ . \tag{3.76}$$

Es ist $\epsilon > 0$ beliebig. Hier zeigt sich wieder die fundamentale Eigenschaft einer Gittertheorie, durch Impulsvertices an Stellen, wo der Gesamtimpuls auf dem diskretisierten Impulsgitter liegt, vollständig definiert zu sein.

Daß die Konstruktion der Funktion $\hat{V}_{2n,r}$ möglich ist, wollen wir an einem vereinfachten Beispiel verdeutlichen: Wir reduzieren den Definitionsbereich auf $\mathbb{R}$ und redefinieren $U_{2n}(\epsilon) := (-\infty,-\epsilon]$ und $N_{2n}(\epsilon) := [\epsilon,\infty)$. Gelingt es uns nun, eine unendlich oft differenzierbare Funktion $S_\epsilon : \mathbb{R} \to \mathbb{R}$ mit den Eigenschaften $S_\epsilon(p) = 0$ für $p \le -\epsilon$ und $S_\epsilon(p) = 1$ für $p \ge \epsilon$ zu konstruieren, so besitzt die Abbildung

$$\hat{V}_{2n,r}(p) := (1 - S_\epsilon(p))\hat{V}^{(W)}_{2n,r}(p) + S_\epsilon(p)\hat{K}(V)_{2n,r}(p) \tag{3.77}$$

gerade die von uns gewünschten Eigenschaften.

Nehmen wir an der Testfunktion (3.31) die Substitution $\frac{2\pi}{a} = \epsilon$ vor, so haben wir eine Funktion $\hat{S}_\epsilon$ konstruiert, die außerhalb von $(-\epsilon,\epsilon)$ verschwindet. Man erhält $S_\epsilon \in \mathcal{C}^\infty(\mathbb{R})$ durch

$$S_\epsilon(p) = \frac{\int_{-\infty}^{p} \hat{S}_\epsilon(q)dq}{\int_{-\infty}^{\infty} \hat{S}_\epsilon(q)dq}\ . \tag{3.78}$$

3.5 Doppelreihenentwicklung in $D = 3$ Dimensionen

In $D = 3$ Dimensionen existieren neben dem marginalen ϕ^2-Vertex in 2. Ordnung nur irrelevante Impulskerne. Die Bestimmungsgleichung (3.65) wird von der Funktion $\hat{V}_{2,2}$ im Kontinuum nicht erfüllt [Wie97b]. Auch auf dem Gitter tritt die $(n, r) = (1, 2)$-Resonanz auf. Sie ist wiederum eine Folge der (willkürlichen) Parametrisierung $\beta(g) = Lg$. Wir beheben das Problem durch eine Doppelentwicklung.

Auf diese Weise erhalten wir eine störungstheoretische Behandlung analog der HRG in drei Dimensionen und bestimmen durch die Rückführung des Gitters auf das Kontinuum die ϕ^4-Trajektorie perturbativ gemäß [Wie97b].

Kapitel 4

Konstruktionsversuch der ϕ_4^4-Trajektorie im Hierarchischen Modell

In diesem Abschnitt präsentieren wir einige Ansätze der Konstruktion der ϕ_4^4-Trajektorie im hierarchischen Modell. Wir stützen uns dabei auf die Arbeiten [Por90] und [Alb91].

Im Kapitel 2 haben wir an vielen Stellen gesehen, daß die dort präsentierte Konstruktion nur für $2 < D < 4$ gültig ist. Wir wollen hier die wesentlichen Faktoren noch einmal aufführen.

- Die Benutzung einer linearen β- bzw. δ-Funktion (2.38), (2.45) ist unsinnig, da sie zur Identität verkommen. Ursache ist, daß die ϕ^4-Kopplung in $D = 4$ Dimensionen marginal ist. In der Störungstheorie erweist sich diese Wahl sogar als falsch (Abschnitt 2.3.2).

- Der Trajektorienraum $\mathcal{V}_{g_0}$ ist nicht konstruierbar, da keine Gauß-Trajektorie bezüglich der linearen δ-Funktion existiert (Kapitel 2.4).[1]

- Zur Berechnung der RG-Trajektorie benötigen wir skalierende Potentiale, die Polynome in Kopplung g und Feld ϕ darstellen. Für $D \to 4$ führt dies zur Verwendung der kompletten, formalen Störungsreihe (2.162). Diese ist praktisch nicht berechenbar und nicht konvergent (Abschnitt 2.6.4).

[1] Für $D \to 4$ gilt $Z_{QU}(0) = Z_{UV}$ und $Z_{QU}(g > 0) = Z_{QU}$ - die Trajektorie „verbindet“ also nicht mehr den trivialen Fixpunkt mit dem Hochtemperaturfixpunkt.

- Die Bedingung (2.88) bringt mit sich, daß eine (nichtstörungstheoretische) Fixpunktapproximante bereits ein Fixpunkt der RGT sein muß.

Wir verwenden im folgenden die RGT in der Form

$$\mathcal{R}(Z)(\phi) = \left\{ \langle Z \rangle_{\gamma,\beta\phi} \right\}^{\alpha} . \tag{4.1}$$

Diese Formulierung vermeidet eine α-Integration vor der Gaußschen Faltung. $\mathcal{T}$ sei die RGT für Potentiale. Die charakterisierenden Parameter bestimmen sich zu

$$\begin{array}{rcrcl} \alpha & = & L^4 & > & 1 \\ \beta & = & L^{-1} & < & 1 \\ \gamma & = & 1 - L^{-2} & < & 1 \, . \end{array} \tag{4.2}$$

Eine mögliche δ-Funktion ist das Inverse der in Abschnitt 2.3.2 bestimmten kubischen β-Funktion

$$\begin{aligned} \beta(g) &= \sum_{n=1}^{3} b_n g^n \qquad (4.3) \\ &= g - 36(L^4 - 1)g^2 + (432 - 3456L^2 - 2592L^4 + 3456L^6 + 2160L^8)g^3 \end{aligned}$$

Man beachte, daß die β-Funktion (4.3) mittels der Transformation $\tilde{\mathcal{T}} = U^{-1}\mathcal{T}U$ gewonnen wurde, wobei $U(V) = \alpha V$ ist. Aus der Eigenschaft, daß $(\tilde{V}, \beta)$ ein skalierendes Paar zu $\tilde{\mathcal{T}}$ ist, folgt, daß auch $(U(\tilde{V}), \beta)$ ein skalierendes Paar zu $\mathcal{T}$ darstellt. Wir dürfen die Trajektorie somit mit der obigen β-Funktion konstruieren.

Die störungstheoretische Konstruktion einer Gauß-Trajektorie gelingt nicht, da die δ-Funktion selbst eine unendliche Reihe ist. Setzt man $b(g)$ wie im Beweis zu 2.4.1 als Potenzreihe in g^ρ an, so läßt sich die Invarianzgleichung $b(g) = b'(\delta(g))$ (vgl. mit (2.60)) nicht mehr nach Potenzen in g ordnen. Für $\rho = 0$ ergibt sich z.B. nur die triviale Lösung.

4.1 Existenz der Trajektorie

Bei der perturbativen Behandlung im Kapitel 2.3 wurde uns der gravierende Unterschied zwischen einer Konstruktion in $D < 4$ und $D = 4$ Dimensionen aufgezeigt. Mit Hilfe der linearen δ-Funktion findet man in $D < 4$ für die Bestimmungsgleichungen sowohl in der hierarchischen (2.42) als auch in der Gitterapproximation (3.29) dieselben Exponenten. Diese zeigen auf, daß für

$2 < D < 4$ nur endlich viele relevante Vertices existieren. Für $D = 4$ ist σ nicht mehr ordnungsabhängig und der Massen- und ϕ^4-Vertex sind in jeder Ordnung der Störungstheorie relevant bzw. marginal.

Diese Eigenschaft korrespondiert mit den Ergebnissen der perturbativen Behandlung der skalaren Feldtheorie. Die dort auftauchende Gleichung für den oberflächlichen Divergenzgrad der Feynman-Graphen (*superficial degree of divergenz*) ist identisch mit (2.42) und (3.29). Man spricht von einer renormierbaren Theorie, wenn es möglich ist, die Divergenzen in jeder Ordnung durch Renormierung der nackten Kopplungen oder Einfügen von Countertermen zu beheben ($D = 4$). Eine Theorie heißt superrenormierbar, wenn in der gesamten Störungsreihe nur endlich viele divergente Graphen existieren ($D < 4$). Die Theorie ist gewiß nicht renormierbar, wenn der Divergenzgrad mit der Ordnung steigt.[2] Die Begriffe renormierbar und superrenormierbar übertragen sich in die Wilson-Renormierungsgruppe, wenn man sie auf die Anzahl der marginalen und relevanten Impulskerne bezieht.[3]

In $D = 4$ Dimensionen sehen wir uns also mit dem Problem konfrontiert, daß der Kopplungsterm, in welchem wir die Trajektorie parametrisieren, marginal ist. Dies bedeutet, daß wir keine pauschale Aussage darüber machen können, wie sich die ϕ^4-Kopplung unter der RGT verhält (wachsend oder fallend), da der ϕ^4-Vertex in der Zentrumsmannigfaltigkeit der RGT liegt.[4] Es ist somit völlig unklar, welche ϕ^4-Theorien, definiert durch

$$V_{\mu_0,g_0}(\phi) = \frac{1}{2}\mu_0\phi^2 + \frac{1}{4!}g_0\phi^4, \tag{4.4}$$

nach Z_{UV} fließen. Oder ob überhaupt ϕ^4-artig gestörte Theorien in die Universalitätsklasse der freie Theorie fallen.

Rigorose Aussagen sind also nur möglich, wenn wir Kontrolle über den RG-fluß bewahren. Hierzu betrachten wir einen durch das Potential

$$V(\phi) = \sum_{n=1}^{s} \frac{k_{2n}}{(2n)!}\phi^{2n} \tag{4.5}$$

[2]Es können jedoch in allen Graphen sog. primitive Divergenzen auftauchen. Die Konvergenz eines Feynman-Graphen ist genau dann gewährleistet, wenn die Summe aus oberflächlichem Divergenzgrad und Divergenzgrade in allen Untergraphen kleiner als Null ist (*Weinberg's Theorem*).

[3]Man beachte, das ein $2n$-Impulskern eine Potenzreihe eines ϕ^{2n}-Vertex samt seiner Ableitungen repräsentiert.

[4]Exakter: Die Zentrumsmannigfaltigkeit tangiert den Eigenraum zu $:\phi^4:$ in Z_{UV}.

erzeugten Boltzmann-Faktor. Dann bildet (4.1) die Theorie (4.5) auf die effektive Theorie

$$\mathcal{T}_0(V)(\phi) = \sum_{n=1}^{\infty} \frac{\tilde{k}_{2n}}{(2n)!}\phi^{2n} \tag{4.6}$$

ab - die Transformation generiert neue Wechselwirkungsterme. Hierbei ist

$$\tilde{k}_{2n} = \tilde{k}_{2n}(k_2, \ldots, k_{2s}) \tag{4.7}$$

OBdA kann man sogleich einen unendlich dimensionalen Raum von Kopplungen benutzen, und beginnt den ersten Iterationsschritt mit $k_{2n} = 0$ für $n > s$. Betrachtet man die trunkierte Transformation $\mathcal{R}_0^s$, der ein Polynomprojektor in ϕ vom Grad $2s$ nachgeschaltet ist, so erhalten wir eine Abbildung $\mathcal{R}_0^s : \mathbb{R}^s \to \mathbb{R}^s$ mit der Zuweisungsvorschrift (4.7). Die Flußgleichungen schreiben sich als

$$\tilde{k}_{2n}(k_2, \ldots, k_{2s}) = \frac{\partial^{2n}}{\partial\phi^{2n}}\mathcal{T}_0(V)(0) \; . \tag{4.8}$$

Mittels obiger Gleichung können wir die renormierten Kopplungen ganz bestimmter Feldterme explizit berechnen. In [Por90] betrachtet man Ausgangspotentiale der Form (4.4) und erhält

$$\mathcal{R}(e^{-V_{\mu_0,g_0}})(\phi) = e^{-V_{\mu_1,g_1}(\phi)} + H(\phi) \; . \tag{4.9}$$

Hierbei sind $\mu_1 = \mu_1(\mu_0, g_0)$ und $g_1 = g_1(\mu_0, g_0)$ über (4.8) berechnet und $H = H(\mu_0, g_0)$ über die Differenz $H := \mathcal{R}(e^{-V_{\mu_0,g_0}}) - e^{-V_{\mu_1,g_1}}$ definiert. Es folgt, daß $H(\phi) = O(\phi^6)$ ist, und so liegt die Erweiterung [Alb91] auf der Hand, sogleich von einem Startpotential der Form $e^{-V_{\mu_0,g_0}} + H_0$ mit $H_0(\phi) = H_{\mu_0,g_0}(\phi) = O(\phi^6)$ auszugehen. Entsprechend (4.9) erhält man

$$\begin{pmatrix} \mu_0 \\ g_0 \\ H_0 \end{pmatrix} \xrightarrow{\mathcal{R}} \begin{pmatrix} \mu_1(\mu_0, g_0, H_0) \\ g_1(\mu_0, g_0, H_0) \\ H_1 \end{pmatrix} . \tag{4.10}$$

Das weitere Vorgehen wollen wir hier nur kurz skizzieren. Das exakte Verfahren findet sich in den oben genannten Referenzen. Zur Untersuchung des Flußverhaltens taylort man die Gleichungen für die Massen- und die ϕ^4-Kopplung exakt (mit Restglied) an.[5] Konstruiert man einen Banachraum der Korrekturterme H über $H(\phi) = h(g^{\frac{1}{4}}\phi)$, so lassen sich die bei der

[5] Man substituiert die Kovarianz γ durch γt und entwickelt in t.

Entwicklung entstehenden Ableitungen von H mit Hilfe der gewichteten Supremumsnorm $\|h\| = \sup_{|\mathrm{Im}(x)|<C} |h(x)e^{cx^2}|$ abschätzen. Man erhält für den n-ten RG-Schritt Gleichungen der Form

$$\mu_{n+1} = L^2\{\mu_n + \frac{1}{2}\gamma g_n - \gamma\mu_n^2 - \frac{3}{2}\gamma^2 g_n\mu_n + \gamma^2\mu_n^3\} + O(g_n^{\frac{3}{2}}) \quad (4.11)$$

$$g_{n+1} = g_n - 4\gamma\mu_n g_n - \frac{7}{2}\gamma^2 g_n^2 + 10\gamma^2 g_n\mu_n^2 + O(g_n^{\frac{3}{2}}) \; . \quad (4.12)$$

Die abgeschätzten Ableitungen der Korrekturterme in H beinhalten Potenzen von $g_n^{\frac{1}{2}}$ und fließen in den Ausdruck $O(g_n^{\frac{3}{2}})$ ein. Das grundsätzliche Verhalten der effektiven Kopplungen unter $\mathcal{DT}$ ($g \to g$ und $\mu \to L^2\mu$) findet sich auch in der vollen Transformation. Das intuitive Gefühl, die Massenkopplung würde bei jeder Wahl von (μ_0, g_0) bei unendlicher Iteration ob des L^2 Faktors explodieren, wird z.B. in [GK84], [Por90] oder [Alb91] widerlegt:

Satz 4.1.1

$$\forall |g| \ll 1 \quad \exists \mu_c(g) = g^{\frac{1}{2}}O(g^{\frac{1}{2}}) \quad : \quad \lim_{n\to\infty} \mathcal{R}^n(e^{-V_{\mu_c,g}}) = Z_{UV} \quad (4.13)$$

Für die renormierte Massen- und ϕ^4-Kopplung bedeutet dies eine gleichmäßige Konvergenz gegen Null.[6] Die kritische Massenkopplung $\mu_c(g)$ muß aus der Menge $\bigcap_{n\in\mathbb{N}}[\alpha_n, \beta_n] = [\lim\alpha_n, \lim\beta_n]$ stammen[7]. Da man die genaue Struktur der Intervallgrenzen nicht kennt, erhalten wir nur einen Beweis für die Existenz einer kritischen Massenkopplung, aber nicht ihre explizite Abhängigkeit von g.

Für den effektiven, irrelevanten Term findet man in [Alb91]

$$h_{n+1}(\phi) = \frac{1}{L^2}\mathcal{L}_0(h_n) + g_n^{\frac{1}{2}}F + O(\|h_n\|^2, g_n^{\frac{1}{2}}\|h_n\|, g_n^{-\frac{1}{2}}\mu_n\|h_n\|, g_n, \mu_n) \; . \quad (4.14)$$

Arbeitet man mit der kritischen Masse, d.h. $\mu_0 = \mu_c(g_0)$, so gilt $O(g_n^{-\frac{1}{2}}\mu_n\|h_n\|) = O(g_n^{\frac{1}{2}}\|h_n\|)$. Benutzen wir nun noch, daß die Normen des linearen Operators $\mathcal{L}_0$ und der Funktion F beschränkt sind,[8] und wir L beliebig groß wählen dürfen, folgen $\|h_{n+1}\| < \|h_n\|$ und die Konvergenz des Restes h_n gegen Null.

[6] Die gleichmäßige Konvergenz ist für μ_c als Funktion von g zu verstehen. Er leitet sich aus den Eigenschaften $|g_n| < \frac{C}{n+1}$ und $|\mu_n + \frac{1}{2}\gamma\frac{L^2}{L^2-1}g_n| < Dg_n^{\frac{3}{2}}$ ab.

[7] Es gilt die Relation $[\alpha_{n+1}, \beta_{n+1}] \subsetneq [\alpha_n, \beta_n]$.

[8] Die Funktion F heißt bei Albuquerque h_0.

Mit Hilfe der Anfangstheorien (4.9) unter den Voraussetzungen von Satz 4.1.1 und additiver Wirkungen H_0 gelingt es uns, einen diskreten RG-fluß zu konstruieren, der in den trivialen Fixpunkt läuft. Es bleibt die offene Frage, ob diese Punktfolgen den Eigenschaften einer ϕ^4-Trajektorie genügen.

4.2 Konstruktionsversuch

Dieses Kapitel trägt seinen Namen nicht zu unrecht, da die vorgestellte Konstruktion nicht rigoros ist. Ursache ist die Verwendung eines nicht vollständigen Vektorraumes von Korrekturen. Komplettiert man diesen, so gilt es, weitere Abschätzungen zu beweisen, was uns bis dato nicht gelungen ist. Wir wollen das Verfahren dennoch vorstellen, um an entsprechenden Stellen auf Probleme, offene Fragen und Ideen einzugehen.

Wir verwenden im folgenden die RGT (4.1) mit einer δ-Funktion der Form

$$\delta(g) = g + O(g^2) \; . \tag{4.15}$$

Für unsere Konstruktion reicht auch[9] $\delta(g) = g + o(g)$, allerdings erfüllt das Inverse der kubischen β-Funktion (4.3) die Gleichung (4.15), so daß wir mit dieser Definition arbeiten werden. Desweiteren sei δ stetig differenzierbar. Auch diese Eigenschaft weist $\delta = \beta^{-1}$ nach dem Satz über die Ableitung der Umkehrfunktion ($\beta'(g) \neq 0$) auf. Die Reparametrisierung δ läßt sich nach oben und unten wie folgt abschätzen:

$$\forall C > 1 \;\; \exists g_0 \in \mathbb{R} \quad : \quad g \leq \delta(g) \leq Cg \tag{4.16}$$

Bei der Suche nach einer Funktion $Z(\phi, g)$, die einen Fixpunkt der erweiterten RGT $\mathcal{R} \times \delta^\star$ darstellt, teilen wir die Wirkungen analog dem Verfahren in $2 < D < 4$ Dimensionen in einen approximativen Fixpunkt Z_1 und einen Rest H. Es sei im folgenden

$$Z_1(\phi, g) = e^{-V_1(\phi,g)} := e^{-g:\phi^4:_1} = e^{-g(\phi^4 - 6\phi^2 + 3)} \; . \tag{4.17}$$

Es handelt sich hierbei um den linearen Anteil des perturbativen Fixpunktpotentials. Man beachte, daß Z_1 kein Fixpunkt der erweiterten linearisierten Transformation $\mathcal{DT}(Z_1)(\phi, \delta(g))$ ist. Die Wahl von (4.17) stellt auf dem ersten Blick einen Rückschritt dar, da dem Verfahren aus Kapitel 2 die Tendenz innewohnte, für $D \to 4$ immer bessere störungstheoretische Approximanten

[9] $f(g) \in o(g) \Leftrightarrow \lim_{g \to 0} \frac{f(g)}{g} = 0$

zu benutzen. Die lineare Näherung reichte dort nur für Probleme in $D < \frac{4}{3}$ [Wie97a]. [10] Dennoch kann diese Approximante als Testobjekt dienen, um uns fehlende Restriktionen aufzuzeigen.

Wir schreiben die Invarianzgleichung nach H um und erhalten

$$\mathcal{F}(H)(\phi, g) := \mathcal{R} \times \delta^\star(Z_1 + H)(\phi, g) - Z_1(\phi, g) \; . \tag{4.18}$$

Als nächstes konstruieren wir einen Banachraum, auf dem $\mathcal{F}$ selbstabbildend ist. Dazu benötigen wir die Konstanten

$$\lambda, g_0 \in \mathbb{R}^+ \; . \tag{4.19}$$

In Zukunft sei g_0 immer so gewählt, daß alle Abschätzungen gültig sind. Wir kennzeichnen das Auftreten solcher Re-Definitionen durch die Angabe eines g_0 über dem Relationszeichen, z.B. $A \overset{g_0}{\leq} B$. Sofern keine Mehrdeutigkeiten auftreten nennen wir alle Konstanten, die in Abschätzungen auftauchen, C, so daß z.B. eine Ungleichung der Form $x < 3C < C$ zulässig ist.

Der Banachraum der Korrekturen $\mathbb{B}$ bestehe aus reelwertigen Funktionen $H(\phi, g)$ über $\mathcal{P}_{g_0}$, die folgende Eigenschaften besitzen:

$$\begin{aligned}
H(\cdot, g) &\in \mathcal{C}^0(\mathbb{R}) && (4.20)\\
H(\phi, \cdot) &\in \mathcal{C}^1([0, g_0]) && (4.21)\\
H(\cdot, g) &\in \mathbb{Z}_2(\mathbb{R}) && (4.22)\\
H(\mathbb{R}, g) &\subseteq \mathbb{R} && (4.23)\\
H(\phi, 0) &= 0 && (4.24)\\
\partial_g H(\phi, 0) &= 0 && (4.25)\\
H(0, g) &= 0 && (4.26)
\end{aligned}$$

Hierbei gewährleisten (4.24) und (4.25), daß $Z_1 + H$ die Anfangsbedingungen einer ϕ^4-Trajektorie 2.2.1 erfüllt. (4.26) sorgt dafür, daß $Z_1 + H_1$ normiert ist. Fordern wir nun noch, daß

$$\max_{n=0,1} \sup_{g \in [0, g_0]} \sup_{\phi \in G^{(k)}(g)} |\partial_g^n H(\phi, g) e^{\lambda g \phi^2}| < \infty \; , \tag{4.27}$$

so wird $\mathcal{B}$ zu einem $\mathbb{R}$-Vektorraum und durch die Norm

$$\|H\| := \max_{n=0,1} \sup_{g \in [0, g_0]} \sup_{\phi \in \mathbb{R}} |\partial_g^n H(\phi, g) e^{\lambda g \phi^2}| \tag{4.28}$$

[10] Es gilt $\sigma_\triangle = \frac{1}{2}$.

komplettiert.

Für den weiteren Verlauf der Konstruktion sind die Selbstabbildungs- und Kontraktionseigenschaften von $\mathcal{F}$ essentiell. Es gelingt (uns) jedoch nicht, diese für Objekte $\partial_g F(H)$ zu zeigen. Das vorgestellte Verfahren ist somit nicht vollständig oder unmöglich. Wir sind nur in der Lage, die ϕ^4-Trajektorie bezüglich der „Norm“

$$\|H\| := \sup_{g\in[0,g_0]} \sup_{\phi\in\mathbb{R}} |H(\phi,g)e^{\lambda g\phi^2}|, \tag{4.29}$$

zu berechnen. Diese vervollständigt jedoch nicht den Raum $\mathbb{B}$, da dieser bezüglich der Kopplung g aus $\mathcal{C}^1$-Funktionen besteht. Die partielle Differenzierbarkeit der Korrekturterme ist jedoch eine notwendige Voraussetzung, denn

- die ϕ^4-Trajektorie ist per Definition in $g = 0$ differenzierbar. Es folgt Differenzierbarkeit in einer Umgebung von Null, die sich OBdA über $[0, g_0]$ erstreckt (Definition 2.2.1),
- die Konstruktion stützt sich auf eine Interpolation, die Differenzierbarkeit in g voraussetzt (4.35).

Desweiteren gewährleistet die Norm (4.29) natürlich nicht, daß die Ableitungen $\partial_g H$ beschränkt sind. Dies sind die wunden Punkte der Konstruktion.

Man prüft schnell nach, daß eine Funktion $\mathcal{F}(H)$ ebenfalls die Eigenschaften (4.20) bis (4.25) besitzt. So gilt z.B.

$$\begin{aligned} & \partial_g \mathcal{F}(H)(\phi,0) \\ = \; & -\alpha \left\langle Z_1(\cdot,0)\right\rangle_{\gamma,\beta\phi}^{\alpha-1} \int d\mu_\gamma(\zeta)\delta'(0) : (\beta\phi+\zeta)^4 : + : \phi^4 := 0 \; . \end{aligned} \tag{4.30}$$

Die Eigenschaft $\mathcal{F}(H)(0,g)$ folgt nur bei Verwendung der normierten RGT. Obwohl wir diese Transformation im Konstruktionsbeweis nicht benutzen, wollen wir OBdA gemäß (4.26) normierte H voraussetzen. Statt der Bedingung (4.27) zeigen wir die

4.3 Existenz eines invarianten Balls

Eine wichtige Voraussetzung zur Anwendung des Fixpunktsatzes von Banach ist die Selbstabbildungseigenschaft. Diese wollen wir nun bezüglich (4.29)

zeigen. Definieren wir für $\mu \in \mathbb{R}_0^+$ die abgeschlossene, konvexe Menge

$$B^{(\mu)} = \left\{ H \in B \middle| \|H\| \leq \mu \right\} \ , \tag{4.31}$$

so müssen wir die Existenz eines $\mu > 0$ zeigen, so daß

$$\mathcal{F} : B^{(\mu)} \to B^{(\mu)} \tag{4.32}$$

gilt. Die Wahl der RGT mit äußerem α, das OBdA eine natürliche Zahl sei, erlaubt folgende Zerlegung

$$\begin{aligned} & \mathcal{F}(\phi, g) \qquad (4.33) \\ = \ & \underbrace{\langle Z_1(\cdot, \delta(g)) \rangle_{\gamma,\beta\phi}^{\alpha} - Z_1(\phi, g)}_{=:\triangle(\phi,g)} + \sum_{k=1}^{\alpha} \binom{\alpha}{k} \langle Z_1(\cdot, \delta(g)) \rangle_{\gamma,\beta\phi}^{\alpha-k} \langle H(\cdot, \delta(g)) \rangle_{\gamma,\beta\phi}^{k} \ . \end{aligned}$$

Mit Hilfe der Sätze 4.5.1,4.5.2 und 4.5.3 gelingt es uns, fast alle Summanden aus (4.33) abzuschätzen. Es bleibt der Term

$$\alpha \langle Z_1(\cdot, \delta(g)) \rangle_{\gamma,\beta\phi}^{\alpha-1} \langle H(\cdot, \delta(g)) \rangle_{\gamma,\beta\phi} \ . \tag{4.34}$$

Dieser Ausdruck ist auch bei den Arbeiten von Pordt und Albuquerque problembehaftet. Da für $H \in \mathcal{B}$ die Eigenschaft $H(\phi, 0) = 0$ gilt, ergibt sich hier

$$\begin{aligned} \langle H(\cdot, \delta(g)) \rangle_{\gamma,\beta\phi} &= \int_0^1 ds \partial_s \langle H(\cdot, s\delta(g)) \rangle_{\gamma,\beta\phi} \\ &= \delta(g) \sup_{s\in[0,1]} \int d\mu_\gamma(\zeta) \, |\partial_g H(\beta\phi + \zeta, s\delta(g))| \\ &\overset{g_0}{\leq} Cg\|H\| \ . \end{aligned} \tag{4.35}$$

Es folgt

$$\left| \mathcal{F}(H)(\phi, g) e^{\lambda g \phi^2} \right| \leq C_\triangle g + \alpha C_1 g \|H\| + \sum_{k=2}^{\alpha} \binom{\alpha}{k} C_k \|H\|^k \overset{g_0}{\leq} \mu \ . \tag{4.36}$$

Es ist natürlich möglich, alle Terme mit Hilfe der Interpolation (4.35) abzuschätzen. Wir wollen dieses Hilfsmittel, das in der Norm (4.29) ungültig ist, jedoch nur dort benutzen, wo keine anderen Abschätzungen greifen.

Als zweiten Schritt gilt es die Ungleichung

$$\|\partial_g \mathcal{F}(H)\| \leq \mu \tag{4.37}$$

zu zeigen. Mit dieser Abschätzung haben wir uns bisher nicht sehr intensiv beschäftigt. Die Wahrscheinlichkeit, aus der differenzierten Abbildung $\mathcal{F}$ kleine Faktoren in g oder $\|H\|$ zu extrahieren, wirkt auf den ersten Blick gering. Unsere Skepsis begründet sich in folgendem Argument:

Ableiten nach g generiert einen Term (k=1)

$$\begin{aligned} & \alpha \left\langle Z_1(\cdot,\delta(g))\right\rangle_{\gamma,\beta\phi}^{\alpha-1} \partial_g \left\langle H(\cdot,\delta(g))\right\rangle_{\gamma,\beta\phi} \\ = \;& \alpha \left\langle Z_1(\cdot,\delta(g))\right\rangle_{\gamma,\beta\phi}^{\alpha-1} \delta'(g) \left\langle \partial_g H(\cdot,\delta(g))\right\rangle_{\gamma,\beta\phi} \;. \end{aligned} \tag{4.38}$$

Da $\delta'(g) = O(1)$, müssen wir $\left\langle \partial_g H(\cdot,\delta(g))\right\rangle_{\gamma,\beta\phi}$ analog (4.35) interpolieren. Das so entstehende $\partial_g^2 H$ findet sich jedoch nicht mehr in $\mathcal{B}$ und kann folglich nicht über die Norm abgeschätzt werden.

Eine Lösungsidee stellt die Einschränkung des Banachraumes auf $\mathcal{C}^\infty$-Funktionen in g dar. Auf diese Weise würde $|\partial_g^n H| \leq \|H\|$ für alle n gelten. Dann gilt es jedoch zu beweisen, daß alle Terme $|\partial_g^n \mathcal{F}(H)(\phi,g)|e^{-\lambda g\phi^2}$ durch μ beschränkt sind.

4.3.1 Ein anderer Weg

In [Por90] und [Alb91] tritt die Abschätzung des Problemterms (4.34) ebenfalls auf. Dort haben die Autoren den Vorteil, daß ihre Korrekturterme von der Ordnung $O(\phi^6)$ sind. Sie erreichen dies, weil der abgespaltene Summand (4.9) durch die renormierten Massen- und ϕ^4-Kopplungen (4.8) generiert wird. Dieses Vorgehen kommt für uns jedoch nicht in Frage, da wir die unter $\mathcal{R}\times\delta^\star$ invarianten, in g parametrisierten Kopplungen des ϕ^2- und ϕ^4-Vertex nicht kennen. Das *construction mapping* soll uns diese im Limes ja gerade erzeugen.

Dehnt man das Integrationsgebiet auf die g-abhängigen komplexen Streifen

$$G^{(k)}(g) = \left\{ \phi \in \mathbb{C} \,\middle|\, g^{\frac{1}{4}} |\mathrm{Im}\phi| < k \right\} \tag{4.39}$$

aus und betrachtet den Raum der dort holomorphen Funktion, so wird dieser, sofern man sich auf stetige Funktionen in g beschränkt, durch die Supremumsnorm (4.29) komplettiert. Die Einschränkung auf $G^{(k)}(g)$ ermöglicht die Abschätzung des Betrages von $e^{-g\phi^2}$ gegen eine Konstante.[11] Den Pro-

[11] Es liegt auf der Hand, im Komplexen mit der Norm $\|H\| = \sup_{\mathcal{P}_{g_0}} |H(\phi,g)e^{\lambda g^{\frac{1}{2}}\phi^2}|$ zu arbeiten. Mit ihr ist es möglich, Polynome in $g^{\frac{1}{4}}\phi$ zu dominieren. Bei unseren Rechnungen machte es jedoch keinen Unterschied, welche Norm wir benutzten.

blemterm spaltet man nun entsprechend

$$\begin{aligned} & \alpha \langle Z_1(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{\alpha-1} \langle H(\cdot,\delta(g))\rangle_{\gamma,\beta\phi} \\ = \; & \alpha \int_0^1 ds \partial_s \langle Z_1(\cdot,\delta(g))\rangle_{\gamma s,\beta\phi}^{\alpha-1} \langle H(\cdot,\delta(g))\rangle_{\gamma s,\beta\phi} \\ & + Z_1(\beta\phi,\delta(g))^{\alpha-1} H(\beta\phi,\delta(g)) \end{aligned} \tag{4.40}$$

auf. Zur Behandlung des zweiten Summanden taylort man $H(\beta\phi,\delta(g))e^{\frac{c}{2}(\beta\phi)^2}$ in der reskalierten Feldvariablen $\beta\phi$ an.[12]

$$\begin{aligned} H(\beta\phi,\delta(g))e^{\frac{c}{2}(\beta\phi)^2} \; = \; & \sum_{n=1}^{2} \frac{1}{(2n)!} \partial_\chi^{2n} H(\chi,\delta(g)) e^{\frac{c}{2}\chi^2}\Big|_{\chi=0} (\beta\phi)^{2n} \\ & + \frac{1}{6!} \partial_\chi^6 H(\chi,\delta(g)) e^{\frac{c}{2}\chi^2}\Big|_{\substack{\chi=\beta\phi s \\ s\in(0,1)}} (\beta\phi)^{2n} \end{aligned} \tag{4.41}$$

Die zusätzliche e-Funktion wird durch Multiplikation mit $e^{-\frac{c}{2}(\beta\phi)^2}$ wieder entfernt. Dieser Term dominiert dann auch die bei der Entwicklung entstehenden Potenzen in $\beta\phi$.

Die Elemente aus $\mathcal{B}$ besitzen im Gegensatz zu den Korrekturen in [Por90] und [Alb91] sehr wohl quadratische und quartische Anteile. Sie bereiten jedoch keine Probleme, denn es gilt:

$$\partial_\chi^2 H(\chi,\delta(g)) e^{\frac{c}{2}\chi^2}\Big|_{\chi=0} = \partial_\chi^2 H(\chi,\delta(g))\Big|_{\chi=0} \tag{4.42}$$

$$\partial_\chi^4 H(\chi,\delta(g)) e^{\frac{c}{2}\chi^2}\Big|_{\chi=0} = \partial_\chi^4 H(\chi,\delta(g))\Big|_{\chi=0} - 6c\partial_\chi^2 H(\chi,\delta(g))\Big|_{\chi=0} \tag{4.43}$$

Mittels der Cauchyschen Ungleichung und Benutzung eines Radius von $R = \frac{k}{2} g^{-\frac{1}{4}}$ ($\Rightarrow U_R(0) \subsetneq G^{(k)}(g)$) lassen sich (4.42) und (4.43) gegen $Cg^{\frac{1}{2}}\|H\|$ abschätzen. Da man auch die Ungleichung

$$Z_1(\beta\phi,\delta(g))^{\alpha-1} \overset{g_0}{\leq} C|e^{-\lambda g\phi^2}| \tag{4.44}$$

für komplexes $\phi \in G^{(k)}(g)$ herleiten kann (Satz 4.5.4), verläuft die Abschätzung des Betrages des Taylor-Polynoms durch $Cg^{\frac{1}{2}}\|H\||e^{-\lambda g\phi^2}|$ problemlos. Der vorangestellte Faktor α wird von der Wurzel in g dominiert.

Einziger Knackpunkt bei dieser Beweisführung ist das Restglied. Die Idee, die bei den Beweisen von Pordt und Albuquerque Anwendung findet, ist die Abschätzung durch L, da man durch das Monom $(\beta\phi)^6$ einen Faktor

[12] Man beachte (4.22) und (4.26).

$\alpha\beta^6 = L^{-2}$ extrahieren kann. Mit diesem kann man für große L die benötigten kleinen Vorfaktoren erzeugen. Im Restglied taucht nun allerdings die künstlich integrierte Exponentialfunktion explizit auf. Ihre Kopplungsfreiheit im Exponenten führt bei einer Cauchy-Abschätzung zu Problemen: Als Radius wählen wir $R = (1-\beta)kg^{-\frac{1}{4}}$. Auf diese Weise erfüllen wir $U_R(\beta\phi s) \subsetneq G^{(k)}(g)$. Damit folgt jedoch

$$\sup_{\psi\in\partial U_R(\beta\phi s)} \left| e^{-\lambda\delta(g)\psi^2+\frac{c}{2}\psi^2} \right| \xrightarrow{g\to 0} \infty \tag{4.45}$$

da $\mathrm{Re}^2(\phi)$ für $g \to 0$ divergiert.

Eine weitere Möglichkeit ist, die Taylor-Potenzen $\beta\phi$ nicht durch einen künstlichen Faktor zu dominieren, sondern mit Hilfe von Z_1. Da wir die Abschätzung (4.44) auch mit $\sqrt{g}$ statt g und 2λ statt λ führen können, siehe 4.5.4, gelingt eine Abschätzung der Form

$$\left| Z_1(\beta\phi, \delta(g))^{\alpha-1} \right| \overset{g_0}{\leq} C \left| e^{-\lambda g^{\frac{1}{2}}\phi^2} \right| \left| e^{-\lambda g\phi^2} \right| \ . \tag{4.46}$$

Wir vollführen also eine Taylor-Entwicklung des nackten $H(\beta\phi, \delta(g))$ und schätzen die Ableitungen mit der Cauchy-Formel ab. Mit diesem Verfahren gelingt es, das Restglied zu kontrollieren: Wir adjungieren den Cauchy-Faktor $g^{\frac{3}{2}}$ an ϕ^6 und dominieren durch $|e^{-\lambda g^{\frac{1}{2}}\phi^2}|$. Nun benötigen wir für die ϕ-Monome 2. und 4. Grades ebenfalls die Cauchy-Faktoren $g^{\frac{1}{2}}$ und g. Die g-Potenzen können somit nicht mehr die Vorfaktoren „klein machen“. Was bleibt, ist z.B. der Massenkoeffizient $8C\alpha\beta^2k^{-2} = 8CL^2k^{-2}$.

Nun mag man glauben, daß nur eine genügend große Wahl von k genügt, um die Vorfaktoren der $\|H\||e^{-\lambda g\phi^2}|$-Terme zu verringern. Tut man dies, muß man jedoch genaue Kenntnis über die L-Abhängigkeit der Koeffizienten besitzen. All unsere Berechnungen haben bisher ergeben, daß die Vorfaktoren in einem solchen Maße von L abhängen, daß eine Dominierung durch geeignete Wahl von k nicht möglich ist.

Das heißt jedoch nicht, daß es unmöglich ist.

4.4 Die Kontraktionseigenschaft

In diesem Abschnitt zeigen wir, daß eine Zahl $0 < q < 1$ existiert, so daß für alle $H_1, H_2 \in \mathbb{B}^{(\mu)}$ die Beziehung

$$\|\mathcal{F}(H_1) - \mathcal{F}(H_2)\| \leq q\|H_1 - H_2\| \tag{4.47}$$

erfüllt ist. Gemeinsam mit den Eigenschaften, daß $\mathcal{B}^{(\mu)}$ als abgeschlossene Untermenge des Banachraumes $\mathcal{B}$ vollständig und $\mathcal{F}$ nach (4.3) selbstabbildend sind, gewinnen wir wiederum aus dem *contraction mapping theorem* [Sma80] die Erkenntnis, daß $\mathcal{F}$ einen Fixpunkt in $\mathcal{B}^{(\mu)}$ besitzt. Man konstruiert diesen, indem man einen beliebigen Startpunkt aus $\mathcal{B}^{(\mu)}$ (z.B. 0) unendlich oft iteriert. Die ϕ^4-Trajektorie bestimmt sich zu

$$Z_1 + \lim_{n\to\infty} \mathcal{F}^n(0) \; . \tag{4.48}$$

Beweis:

$$\begin{aligned}
& |\mathcal{F}(H_1) - \mathcal{F}(H_2)| \\
\leq \; & \sum_{n=1}^{\alpha} \binom{\alpha}{n} \left| \langle Z_1(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{\alpha-n} \left\{ \langle H_1(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{n} - \langle H_2(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{n} \right\} \right| \\
= \; & \sum_{n=1}^{\alpha} \binom{\alpha}{n} \langle Z_1(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{\alpha-n} \left| \langle (H_1 - H_2)(\cdot,\delta(g))\rangle_{\gamma,\beta\phi} \right| \\
& \times \left| \sum_{k=0}^{n-1} \langle H_1(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{k} \langle H_2(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{n-1-k} \right| \; .
\end{aligned}$$

Für diese Rechnung nutzt man die Relation $x^n - y^n = (x-y)\sum_{k=0}^{n-1-k} x^k y^{n-1-k}$ und die Linearität der Gaußschen Mittelwertbildung aus. Das weitere Vorgehen hängt vom Summenindex n ab. Für alle n gilt

$$\left| \langle (H_1 - H_2)(\cdot,\delta(g))\rangle_{\gamma,\beta\phi} \right| \leq \|H_1 - H_2\| \; . \tag{4.49}$$

Im Falle $n = \alpha$ müssen wir sogar gegen $\|H_1 - H_2\| e^{-\lambda g\phi}$ abschätzen, da die Ungleichung $\langle Z_1(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{\alpha-n} \leq C e^{-\lambda g\phi^2}$ nur für $n \neq \alpha$ gilt.

Für den Betrag der inneren Summe in (4.49) existiert die obere Schranke

$$\left| \sum_{k=0}^{n-1} \langle H_1(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{k} \langle H_2(\cdot,\delta(g))\rangle_{\gamma,\beta\phi}^{n-1-k} \right| \leq n\mu^{n-1} \; , \tag{4.50}$$

die man durch entsprechende Wahl von μ für $n \neq 1$ beliebig klein machen kann.

Der Fall $n = 1$ bedarf einer Sonderbehandlung. Analog zu (4.35) arbeiten wir aus dem Ausdruck $\|H_1 - H_2\|$ einen g-Faktor heraus.

Die Abschätzung des Terms $|\partial_g \mathcal{F}(H)(\phi,g) e^{-\lambda g\phi^2}|$ steht noch aus. Wir rechnen jedoch mit ähnlichen Problemen wie in (4.3).

4.5 Abschätzungen

Die folgenden Abschätzungen sind zu einem großen Teil für reelle Felder hergeleitet. Sie bewahren ihre Gültigkeit jedoch auch auf dem komplexen Streifen (4.39). Fast alle auftauchenden Konstanten sind L-abhängig. Dies spielt jedoch keine Rolle, da L ein beliebiger, aber fester Parameter ist. Desweiteren sind die (vom Blockparameter abhängigen) Konstanten immer an Potenzen in g oder $\|H\|$ gekoppelt, so daß kleine g_0 oder μ die Koeffizienten dominieren.

Satz 4.5.1 (Die Güte der linearen Approximante)

$$\exists g_0 \in \mathbb{R}^+ \quad \forall (\phi, g) \in \mathcal{P}_{g_0} \quad : \quad |\triangle(\phi, g)| \leq C g^{\frac{1}{2}} e^{-\lambda g \phi^2}$$

Beweis: Wir definieren

$$R_s(\phi, g) = \left\{ \int d\mu_{\gamma s}(\zeta) e^{-\langle V_1(\cdot, \delta(g)) \rangle_{\gamma(1-s), \beta\phi+\zeta}} \right\}^{\alpha} =: r_s(\phi, g)^{\alpha} \; . \tag{4.51}$$

Diese Interpolation ist fast identisch mit (2.150). Sie beschneidet δ jedoch nicht in g und legt die RGT mit externem α zugrunde. Ferner leiten wir in diesem Abschnitt die Güte-Ungleichung über die Cauchy-Formel her. Es wäre aber genauso gut möglich, den Beweis mit kleinen Änderungen analog Kapitel 2.6.7 zu führen.

Aus den Eigenschaften $R_0(\phi, g) = Z_1(\phi, \delta(g))$ und $R_1 = \mathcal{R} \times \delta^\star(Z_1)$ folgt die Beziehung

$$\triangle(\phi, g) = \int_0^1 ds \partial_s \left\{ R_s(\phi, g) + Z_1(\phi, s\delta(g) + (1-s)g) \right\} \; . \tag{4.52}$$

Mit Hilfe der Relationen $|\delta(g) - g| \overset{g_0}{\leq} C_1 g^2$, $|s\delta(g) + (1-s)g| \overset{g_0}{\geq} 4C_2 g$ und $: \phi^4 := \frac{1}{2}(\phi^2 - 6)^2 + \frac{1}{2}\phi^4 - 15 \geq \frac{\phi^4}{2} - 15$ erhält man

$$\begin{aligned}
& \left| \int_0^1 ds \partial_s Z_1(\phi, s\delta(g) + (1-s)g) \right| \\
\leq \; & C_1 g^2 | : \phi^4 : | e^{-C_2 g \phi^4} e^{-C_2 g \phi^4 + 60 C_2 g} \\
\leq \; & C_1 g \left\{ (g^{\frac{1}{4}}\phi)^4 + 6 g^{\frac{1}{2}} (g^{\frac{1}{4}}\phi)^2 + 3g \right\} e^{-C_2 (g^{\frac{1}{4}}\phi)^4} e^{g(\frac{\lambda^2}{4C_2} + 60 C_2)} e^{-g\lambda\phi^2} \\
\overset{g_0}{\leq} \; & C g e^{-g\lambda\phi^2} \; .
\end{aligned} \tag{4.53}$$

Für die differenzierte Interpolierte gilt $|\alpha r_s^{\alpha-1}\partial_s r_s| \leq C|\partial_s r_s|$. Wir erhalten

$$\begin{aligned} |\partial_s r_s(\phi,g)| &\overset{\text{[Wie97a]}}{=} 2\gamma \int d\mu_{\gamma s}(\zeta) \left\{ \partial_{\beta\phi} e^{-\frac{1}{2}\langle V_1(\cdot,\delta(g))\rangle_{\gamma(1-s),\beta\phi+\zeta}} \right\}^2 \\ &= 2\gamma \int d\mu_{\gamma s}(\zeta) \left| \partial_\psi e^{-\frac{1}{2}\delta(g):\psi^4:_{1-\gamma(1-s)}} \right|^2_{\psi=\beta\phi+\zeta} . \end{aligned} \tag{4.54}$$

Die Funktion $\exp\{-\frac{1}{2}\delta(g) : \psi^4 :_{1-\gamma(1-s)}\}$ ist für $\psi \in \mathbb{C}$ eine ganze Funktion. Schätzen wir den Betrag der ψ-Ableitung über die Cauchysche Ungleichung mit einem Kreisradius $R = \delta(g)^{-\frac{1}{4}}$ ab[13] und benutzen [Por90]

$$\forall\psi \in \mathbb{R}\ \forall\chi \in \mathbb{C}\ \forall s \in [0,1]\ \exists a,b \in \mathbb{R}\ :\ \mathrm{Re} : (\phi+\chi)^4 :_{1-\gamma(1-s)} \geq \frac{\phi^2}{2} - a|\chi|^4 - b \tag{4.55}$$

so erhalten wir

$$\begin{aligned} (4.54) &\leq 2\gamma \int d\mu_{\gamma s}(\zeta)\delta(g)^{\frac{1}{2}} e^{-\frac{1}{2}\delta(g)(\beta\phi+\zeta)^2 + a + b\delta(g)} \\ &\overset{g_0}{\leq} C g^{\frac{1}{2}} e^{-\lambda g\phi^2} \end{aligned} \tag{4.56}$$

Die Gaußsche Integration der quadratischen Form führt man exakt aus (Satz 2.1.2) und schätzt dann ab.

□

Satz 4.5.2 (Potenzierte Gaußsche Erwartungswerte von Z_1)

$$\exists g_0, C \in \mathbb{R}^+\ \ \forall n \in \mathbb{N}\ \ \forall s \in [0,1]\ \ \forall(\phi,g) \in \mathcal{P}_{g_0}\ \ :\ \ \langle Z_1(\cdot,\delta(g))\rangle^n_{\gamma s,\beta\phi} \leq C e^{-\lambda g\phi^2}$$

Beweis: Man benutzt $\delta(g) \overset{g_0}{\geq} g$ (4.16) und[14] $: \psi^4 :_1 \geq \psi^4 - 6\psi^2 \geq r\psi^2 - \frac{1}{4}(r+6)^2$ für beliebiges, reelles r. Es folgt mit Re-Definition von C nach Gaußscher Integration

$$\langle Z_1(\cdot,\delta(g))\rangle_{\gamma s,\beta\phi} \leq C \int d\mu_{\gamma s}(\zeta) e^{-gr(\beta\phi+\zeta)^2} \leq C e^{-g\frac{2r\beta^2}{1+2\gamma s g r}\phi^2} . \tag{4.57}$$

Wählt man $2r\beta^2 > \lambda$, so existiert ein s-unabhängiges g_0, und die Behauptung des Satzes für $n = 1$ folgt. Für $n > 1$ gilt sie trivialerweise.

[13] Da der $\triangle$-Term für $g = 0$ verschwindet, sei OBdA $g \neq 0$. Desweiteren liegt jeder abgeschlossene Kreis um ψ im Holomorphiegebiet $\mathbb{C}$, und die Cauchysche Ungleichung $|f^{(n)}(\psi)| \leq \frac{n!}{R^n} \sup_{|\chi|=R} |f(\psi+\chi)|$ ist anwendbar.

[14] $\psi^4 = \left(\psi^2 - \frac{r+6}{2}\right)^2 + (r+6)\psi^2 - \frac{1}{4}(r+6)^2$

Satz 4.5.3 (Potenzierter Gaußscher Erwartungswert von H)

$$\forall n > \frac{1}{2}L^2 \exists g_0 \in \mathbb{R}^+ \forall s \in [0,1] \quad \forall (\phi, g) \in \mathcal{P}_{g_0} \quad : \quad \langle H(\cdot, \delta(g))\rangle^n_{\gamma s, \beta\phi} \leq \|H\|^n e^{-\lambda g \phi^2}$$

Beweis: Mit $\delta(g) \overset{g_0}{\geq} g$ (4.16) erhält man die Abschätzung

$$\left|\langle H(\cdot, \delta(g))\rangle_{\gamma s, \beta\phi}\right|^n \leq \|H\|^n e^{-\lambda g \frac{2n\beta^2}{1+2\gamma s \lambda g}\phi^2} \ . \tag{4.58}$$

Für kleine g_0 nähert sich der Nenner im Exponenten beliebig nahe der Eins. g_0 ist nicht vom Interpolationsparameter s abhängig, da $1 + 2\gamma s\lambda g$ in diesem monoton steigend ist und so g_0 für $s = 1$ bestimmt wird.

$$\frac{2n\beta^2}{1 + 2\gamma s \lambda g} \overset{g_0}{\geq} 1 \tag{4.59}$$

erfordert die notwendige Bedingung $2n > L^2$. Wir merken an, daß diese Eigenschaft für $n = \alpha$ gewiß erfüllt ist.

□

Satz 4.5.4 (Abschätzung von Z_1 mit komplexem Feld)

$$\exists C, g_0 \in \mathbb{R}^+ \ \ \forall g \in [0, g_0] \ \ \forall \phi \in G^{(k)}(g) \quad : \quad \left|Z_1(\beta\phi, \delta(g))^{\alpha-1}\right| \overset{g_0}{\leq} C \left|e^{-\lambda g \phi^2}\right|$$

Beweis: Für alle $\phi \in G^{(k)}(g \neq 0)$ und $\zeta, r \in \mathbb{R}$ gilt:[15]

$$\begin{aligned} \mathrm{Re}\left(:(\beta\phi + \zeta)^4 :_1\right) &\geq \left\{2r - 6\beta^2 \mathrm{Im}^2(\phi) - 6\right\}(\beta \mathrm{Re}(\phi) + \zeta)^2 - r^2 \\ &\geq \left\{2r - 6\beta^2 k^2 g^{-\frac{1}{2}} - 6\right\}(\beta \mathrm{Re}(\phi) + \zeta)^2 - r^2 \end{aligned} \tag{4.60}$$

Vergessen wir nicht $\delta(g) \overset{g_0}{\geq} g$ und $g^{\frac{1}{2}} \overset{g_0=1}{\geq} g$, so führt die Wahl von $r = Cg^{-\frac{1}{2}}$ mit $(2C - 6\beta^2 k^2 - 6g^{\frac{1}{2}}) > \lambda(\alpha - 1)^{-1}$ zum Ziel.

□

[15] Man benutzt hierbei $\mathrm{Re}^4(z) \geq 2r\mathrm{Re}^2(z) - r^2$ für alle komplexen z und reellen r.

Zusammenfassung und Ausblick

Ziel dieser Arbeit war die Konstruktion der ϕ^4-Trajektorie in zwei verschiedenen Modellen (HRG/GRG). Dabei hätten die Behandlungsmethoden unterschiedlicher nicht sein können: Der rigorosen Beweisführung in der hierarchischen Approximation stand die Störungstheorie auf dem Gitter gegenüber - eine Konsequenz der wesentlich höheren Komplexität des „großen Bruders“.

Betrachtet man den Berechnungsaufwand in der HRG

- Konstruktion eines geeigneten Banachraumes
- Beweis von Invarianz- und Kontraktionseigenschaften durch Norm-Abschätzungen
- Suche nach geeigneten approximierten Fixpunkt-Trajektorien
- Berechnung der Approximanten durch Störungstheorie,

so liegt es auf der Hand, daß die in der GRG oder der kontinuierlichen RG zu lösenden Probleme um ein Vielfaches komplexer sind. Zudem ist die Mathematik einer reellen oder komplexen Veränderlichen besser verstanden und erforscht als das Gebiet des Pfadintegrals [GJ81].[16] Dies macht die Konstruktion in der hierarchischen Approximation jedoch nicht trivial.

Das wesentliche Defizit der Berechnungen im 2. Kapitel besteht in der Beschränkung der Kopplung g durch g_0. Obwohl dieser Tatbestand gegenüber der Störungstheorie, deren perturbative Reihe nur für $g = 0$ konvergiert, einen großen Vorteil ausmacht, sind wir von dem eigentlichen Ziel, dem Ausführen des Limes $g \to \infty$, noch weit entfernt. Es gilt also das Verfahren in der Hinsicht zu verbessern, daß alle g-abhängigen Abschätzungen unabhängig von einer Maximalkopplung g_0 werden. C. Wieczerkowski hat

[16] Dieses Zitat soll die Verdienste der Autoren J. Glimm und A. Jaffe nicht schmälern, sondern betonen!

sich dieses Problems angenommen und präsentiert in der Überarbeitung von [Wie97a] ein allgemeineres Lösungsprinzip.

Eine sinnvolle Erweiterung des Verfahrens ist sicherlich der Beweis, daß auch Baumgraphenpotentiale approximierte Fixpunkte generieren. Da sie die Baumschranke trivialerweise erfüllen, muß man nur noch die Güte-Abschätzung zeigen. Der Vorteil von Baumgraphen liegt auf der Hand: sie sind mit der Koeffizientenformel (2.124) in jeder Ordnung explizit berechenbar. Desweiteren sind die Kettenbrüche, die bei der Berechnung einer Obergrenze für die Baumgraphenschranke entstehen, einfacherer Natur.

Eine weitere interessante Aufgabe stellt die Bestimmung des Grenzwertes der effektiven ϕ^4-Untergrenze $\tilde{\lambda}_4^\infty$ dar.

Das weitere Vorgehen bei der Behandlung des Problems in $D = 4$ Dimensionen ist klar: es gilt die Konstruktion zu komplettieren. Das Manko unseres Verfahrens ist, daß der approximierte Fixpunkt nicht „gut genug“ ist. Die Korrekturterme enthalten quadratische Felder und ϕ^4-Anteile. Da wir in vier Dimensionen nicht wie in $D < 4$ mittels eines $\delta < 1$ große Terme dominieren können, müssen wir die RGT-internen Parameter α und β benutzen. Die linearisierte RGT zeigt uns (Eigenwerte), daß dieses Vorhaben nur für Feldpotenzen der Ordnung sechs gelingt. Aus diesem Grund sehen wir die besten Chancen in einer Aufteilung des Potentialraumes: Ein zweidimensionaler Raum für Massen- und ϕ^4-Kopplung und ein „Restraum“- in der Hoffnung, daß die Behandlung des endlich dimensionalen, nicht trivialen Problems (eine relevante und eine marginale Richtung) lösbar ist.

Auf dem Gitter gelang die perturbative Behandlung mittels der T-Funktion, die es uns ermöglichte, die Ideen der Kontinuumslösung [Wie97d, Wie97b] zu adaptieren. Ein großer Nachteil ist die fehlende Berechnung des reduzierten Γ-Kernes, den man für eine allgemeine Form der Beweise benötigt. Aus diesem Grund ist auch die Berechnung der zweiten Ordnung noch nicht beendet.

Allerdings stellt sich die Behandlung der Kontinuumstheorie in drei Dimensionen auch nicht so einfach dar wie in $D = 4$. So ist es uns z.B. nicht gelungen, die Faltungen, die sich bei der Berechnung der Impulskontinuumskerne in zweiter Ordnung ergeben, vgl. z.B. (3.60), explizit zu lösen. Numerisch kann man jedoch zeigen, daß der Impulskern $\hat{K}_{2,2}^{cont}(0)$ nicht verschwindet und eine Doppelentwicklung in g und $\log g$ notwendig ist.

Anhang A

Notation

$\mathbb{N}_k$	$= \{k, k+1, \ldots\}$
$\mathbb{Z}_k$	$= \mathbb{Z}/k\mathbb{Z} = \{\overline{0}, \ldots, \overline{k-1}\}$, $\mathbb{Z}_0 = \mathbb{Z}$
$M_\star$	Die Gruppe M reduziert um das neutrale Element. Bei einem Körper bezieht sich dies auf die Addition.
$[(x_1, \ldots, x_n)]$	$= ([x_1], \ldots, [x_2])$
$\mathcal{C}^n(U)$	Menge der auf[1] U n-mal stetig differenzierbaren Funktionen
$\mathbb{Z}_2(U)$	Menge der Abbildungen $Z : U \to \mathbb{R}$ mit $Z(\phi) = Z(-\phi)$ für alle $\phi \in U$
Kovarianz	Reeller[2], symmetrischer, positiv definiter Operator[3]
$S_\epsilon(U)$	$= U \times i(-\epsilon, \epsilon)$ mit $U \subseteq \mathbb{R}$
$\mathcal{O}(U)$	Die Menge der auf dem Bereich[4] $U \subseteq \mathbb{C}$ holomorphen Funktionen
l_p, $\|\cdot\|_p$	$l_p = \left\{(a_n) \middle\vert (\sum_n \lvert a_n\rvert^p)^{\frac{1}{p}} < \infty\right\}$ ist bezüglich der Norm $\|(a_n)\|_p = (\sum_n \lvert a_n\rvert^p)^{\frac{1}{p}}$ ein Banachraum.
$L(\mathcal{H}(a))$	$= \{f : \mathcal{H}(a) \to \mathcal{H}(a) \vert f \text{ linear}\}$

[1] OBdA sei U offen. Ansonsten definieren wir $\mathcal{C}^n(U) = \bigcap_{U \subset \tilde{U} offen} \mathcal{C}^n(\tilde{U})$.

[2] O reell $\Leftrightarrow O(x, y) \in \mathbb{R}$ für alle x, y

[3] auch positiver Operator oder Operator > 0

[4] offene, nichtleere Teilmenge

Anhang B

Formelsammlung

Wir geben eine kurze Zusammenfassung der wichtigsten Erkenntnisse über die Normalordnung und das Gaußsche Maß. Weitere Informationen finden sich z.B. in [GJ81, Geh97, GS96, Rol96].

B.1 Normalordnung

Die Normalordnung begegnet uns in vielen Bereichen der Physik. Normalgeordnete Ausdrücke liefern häufig die kanonische Formulierung eines Problems. In der Feldtheorie steht der Normalordnungsbegriff im allgemeinen für das Entfernen der divergierenden Nullpunktsenergie aus den Zuständen. Im Kontext der RG erweisen sich die normalgeordneten Monome als Basis von Eigenvektoren bezüglich der linearisierten RGT am trivialen Fixpunkt.

Wir betrachten das erzeugende Funktional

$$: e^{(\phi,J)} :_\nu = e^{(\phi,J)-\frac{1}{2}(J,\nu J)} \; . \tag{B.1}$$

ϕ und J seien Elemente eines Hilbertraumes - in unserem Fall Felder auf dem Kontinuum $\mathbb{R}^D$ oder dem Gitter $\Lambda(a)$, ν eine beliebige Kovarianz. Ein bezüglich ν normalgeordnetes polynomiales Funktional wird über die Funktionalableitung [MM94] definiert.[1]

$$: \phi(x_1)^{m_1} \dots \phi(x_n)^{m_n} :_\nu \tag{B.2}$$

[1] Auf dem Gitter muß die Funktionalableitung nicht definiert werden, da man die Feld-Spins $\phi(x)$, $x \in \Lambda(a)$ explizit mit den üblichen Differentiationsregeln für eindimensionale reellwertige Funktionen ableiten kann.

$$= \frac{\partial^{|m|}}{\partial J(x_{1,1}) \dots J(x_{1,m_1}) \dots J(x_{n,1}) \dots J(x_{n,m_n})} : e^{(\phi,J)} :_\nu \Big|_{\substack{J=0 \\ x_{i,j}=x_i}} \quad \text{(B.3)}$$

Man sieht sofort, daß die Normalordnung linear ist und erhält z.B.

$$: \phi(x_1) :_\nu = \phi(x_1) \quad \text{(B.4)}$$

$$: \phi(x_1)\phi(x_2) :_\nu = \phi(x_1)\phi(x_2) - v(x_1, x_2) \; . \quad \text{(B.5)}$$

Die Normalordnung von Polynomen $\mathbb{R} \to \mathbb{R}$ erhält man, wenn man auf dem Gitter $\Lambda(1)$ arbeitet, die ultralokale Kovarianz $\nu(x,y) = \nu\delta(x,y)$ benutzt und $\phi = \phi(x)$ setzt. Die normalgeordneten Monome sind reskalierte Hermite-Polynome:

$$P_{n,\nu}(\phi) =: \phi^n :_\nu = \left(\frac{\nu}{2}\right)^{\frac{n}{2}} H_n\left(\frac{\phi}{\sqrt{2\nu}}\right) \quad \text{(B.6)}$$

Eine weitere Definition der normalgeordneten Polynome, die auch auf beliebige Funktionale ausgeweitet werde kann, ergibt sich aus der Anwendung des Differentialoperators $\exp\left\{-\frac{1}{2}\left(\frac{\partial}{\partial\phi}, \nu\frac{\partial}{\partial\phi}\right)\right\}$ auf das polynomiale Funktional. Die Äquivalenz zeigt man durch Berechnung von (B.1), indem man den Normalordnungsoperator auf $\exp(\phi, J)$ anwendet.

Mit $J(x) = J\delta(x - x_1)$ folgt die „Vereinfachung“

$$: \phi(x_1)^n := \frac{\partial^n}{\partial J^n} \exp\left\{J\phi(x_1) - \frac{1}{2}\nu(x_1,x_1)J^2\right\}\Big|_{J=0} . \quad \text{(B.7)}$$

Für n-Punkt-Funktionen setzten wir $J(x) = \sum_{m=1}^n J_m\delta(x - x_m)$ und betrachten entsprechende Mehrfachableitungen. Mit Hilfe dieses Tricks zeigt man z.B. leicht die Fusionsformel [Rol96]

$$\begin{aligned} & : \phi(x_1)^{n_1} :_\nu : \phi(x_2)^{n_2} :_\nu \\ = & \sum_{m=0}^{\min\{n_1,n_2\}} m! \binom{n_1}{m}\binom{n_2}{m} \nu(x_1,x_2)^m : \phi(x_1)^{n_1-m}\phi(x_2)^{n_2-m} :_\nu \; . \end{aligned} \quad \text{(B.8)}$$

B.2 Gaußsche Maße

Für n-dimensionale Kovarianzen γ ist das Gaußsche Maß auf dem $\mathbb{R}^n$

$$d\mu_\gamma(\phi) := \frac{d^n\phi}{\sqrt{(2\pi)^n \det(\gamma)}} e^{-\frac{1}{2}(\phi,\gamma^{-1}\phi)} \quad \text{(B.9)}$$

wohldefiniert und auf eins normiert. $(\cdot,\cdot)$ bezeichnet das euklidische Skalarprodukt des $\mathbb{R}^n$. Äquivalent dazu ist die Definition über die charakteristische Funktion

$$\int d\mu_\gamma(\phi) e^{(\phi,J)} := e^{\frac{1}{2}(J,\gamma J)} \quad \forall J \in \mathbb{R}^n \ . \tag{B.10}$$

Sollte $n \to \infty$ oder der zugrunde liegende Raum gar von überabzählbarer Dimension sein, d.h. es existiert keine abzählbare Basis, wie z.B. beim Kontinuums-Pfadintegral, so wollen wir Gleichung $B.10$ als alleinige Definition betrachten. Dabei muß dann die Wohldefiniertheit des Exponenten gewährleistet sein. Im abzählbar unendlich dimensionalen Fall kann man z.B. das Maß auf den Folgenraum l_2 einschränken und fordern, daß die Kovarianz in der Operatornorm finit ist. Es folgt

$$\|(J,\gamma J)\|_2 \leq \|\gamma\| \, \|J\|_2^2 < \infty \ . \tag{B.11}$$

Da aber schon die physikalische Kovarianz $\frac{1}{p^2}$ unbeschränkt ist, müssen wir zur Regularisierung häufig künstliche *cutoffs* einführen, die nach einer Berechnung wieder entfernt werden.

Zitieren wir den Satz von Bochner [Geh97], so folgt die Wohldefiniertheit der Definition (B.10) aus der Eigenschaft, daß $e^{-\frac{1}{2}(J,\gamma J)}$ als in J stetige, positiv (semi)definite Funktion Fourier-Transformierte eines endlichen, positiven Maßes ist.[2]

Mit Hilfe des Gaußschen Maßes definieren wir die Integraltransformation

$$\langle Z \rangle_{\gamma,\phi} := \int d\mu_\gamma(\zeta) Z(\phi+\zeta) \tag{B.12}$$

und nennen sie das Gaußsche Mittel mit Kovarianz γ und Mittel ϕ.

Zwischen (B.12) und der Normalordnung besteht ein fundamentaler Zusammenhang:

$$\langle :\, Z(\cdot)\, :_\nu \rangle_{\gamma,\phi} =: Z(\phi) :_{\nu-\gamma} \ . \tag{B.13}$$

Für polynomiale Funktionale beweist man diesen Zusammenhang, indem man das Gaußsche Mittel der erzeugenden Funktion $: e^{(\phi,J)} :$ berechnet.

Für zwei Kovarianzen γ_1,γ_2 gilt die Faltungsformel für Gaußsche Maße:

$$\int d\mu_{\gamma_1+\gamma_2}(\zeta) Z(\phi+\zeta) = \int d\mu_{\gamma_1}(\zeta_1) d\mu_{\gamma_2}(\zeta_2) Z(\phi+\zeta_1+\zeta_2) \tag{B.14}$$

[2]Man erhält (B.10) mit der Substitution $J \to iJ$.

Eine weitere Abbildung stellen die trunkierten Erwartungswerte[3], auch Kumulanten genannt, dar, definiert über[4]

$$\langle [O_1;\ldots;O_n]\rangle^T_{\gamma,\beta\phi} := \frac{\partial^n}{\partial\lambda_1\ldots\partial\lambda_n} \ln\left\langle e^{\sum_{i=1}^n \lambda_i O_i(\cdot)}\right\rangle_{\gamma,\beta\phi}\Bigg|_{\lambda_i=0} . \tag{B.15}$$

Einfache Rechnungen liefern

$$\langle [O]\rangle^T = \langle O\rangle \tag{B.16}$$

$$\langle [O_1;O_2]\rangle^T = \langle O_1O_2\rangle - \langle O_1\rangle\langle O_2\rangle . \tag{B.17}$$

Hierbei ist (B.17) nichts anderes als die Korrelationsfunktion der Operatoren O_1 und O_2. Eine wichtige Eigenschaft der Kumulanten ist ihre Multilinearität, d.h.

$$\langle [\ldots,\lambda_1O_1+\lambda_2O_2,\ldots]\rangle^T = \lambda_1\langle [\ldots,O_1,\ldots]\rangle^T + \lambda_2\langle [\ldots,O_2,\ldots]\rangle^T , \tag{B.18}$$

die direkt aus der Linearität von (B.12) folgt.

Für die störungstheoretische Behandlung der RG benötigt man in 2. Ordnung folgende Kumulantenformel:

$$\begin{aligned} &\langle :\phi(x_1)^{n_1}:_\nu, :\phi(x_2)^{n_2}:_\nu\rangle^T_{\gamma,\psi} \qquad\qquad (B.19)\\ = &\sum_{m=1}^{\min(n_1,n_2)} m!\binom{n_1}{m}\binom{n_2}{m} :\psi(x_1)^{n_1-m}:_{\nu-\gamma} :\psi(x_2)^{n_2-m}:_{\nu-\gamma} \gamma(x_1,x_2)^m . \end{aligned}$$

Eine andere Darstellung dieser Kumulante findet man, wenn man die Formeln (B.17), (B.8) und (B.13) benutzt. Es ergibt sich

$$\begin{aligned} &\langle :\phi(x_1)^{n_1}:_\nu, :\phi(x_2)^{n_2}:_\nu\rangle^T_{\gamma,\psi}\\ = &\sum_{m=1}^{\min(n_1,n_2)} m!\binom{n_1}{m}\binom{n_2}{m} :\psi(x_1)^{n_1-m}\psi(x_2)^{n_2-m}:_{\nu-\gamma}\\ &\times\sum_{l=1}^{m}(-1)^{l+1}\binom{m}{l}\nu(x_1,x_2)^{m-1}\gamma(x_1,x_2)^m . \end{aligned} \tag{B.20}$$

[3] Die Definition ist unabhängig von der Art der Mittelwertbildung und der Operatoren.

[4] $\langle [O;]^n\rangle^T = \langle \underbrace{[O;\ldots;O]}_{n \text{ mal}}\rangle^T$

Abbildungsverzeichnis

Literaturverzeichnis

[Alb91] P. Albuquerque. La liberté asymptotique du modèle ϕ_4^4 dans l'approximation hiérarchique et le théorème de la variété centrale. Diplomarbeit, Universität Genf, 1991.

[BG95] G. Benfatto und G. Gallavotti. *Renormalization Group.* Princeton University Press, 1995.

[Dys69] F. J. Dyson. Nonexistence of Spontaneous Magnetization in a One-Dimensional Ising Ferromagnet. *Commun. Math. Phys.*, 12, 1969.

[FL94] W. Fischer und I. Lieb. *Funktionentheorie.* vieweg, 1994.

[For91] O. Forster. *Analysis 2.* vieweg studium 31, 1991.

[Geh97] B. Gehrmann. Störungstheoretische und numerische Berechnung von Renormierungsgruppen-Fixpunkten und kritischen Exponenten. Diplomarbeit, Westfälische Wilhelms-Universität Münster, 1997.

[GH86] J. Guckenheimer und P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* Springer Verlag, 1986.

[GJ81] J. Glimm und A. Jaffe. *Quantum Physics.* Springer-Verlag, 1981.

[GK84] K. Gawedzki und A. Kupiainen. Asymptotic freedom beyond perturbation theory. In K. Osterwalder und R. Stora, Editoren, *critical phenomena, random systems, gauge theories*, Seiten 185–293. Les Houches, 1984.

[GS96] J. Göttker-Schnetmann. Analytische und numerische Untersuchungen hierarchischer Renormierungsgruppenfixpunkte am Beispiel O(N)-invarianter Modelle. Diplomarbeit, Westfälische Wilhelms-Universität Münster, 1996.

[HC64] A. Hurwitz und R. Courant. *Funktionentheorie*, Kapitel Die Umkehrung der analytischen Funktionen. Springer-Verlag, 1964.

[Kad66] L. P. Kadanoff. *Physics*, 2, 1966.

[MM94] I. Montvay und G. Münster. *Quantum Fields on a Lattice.* Cambridge University Press, 1994.

[MV92] R. Meise und D. Vogt. *Einführung in die Funktionalanalysis.* Vieweg Verlag, 1992.

[OL] H.-H. Ostmann und H. Liermann. *Grundzüge der Mathematik I*, Kapitel Zahlentheorie.

[Por90] A. Pordt. *Convergent Multigrid Polymer Expansions and Renormalization for Euclidean Field Theory.* Doktorarbeit, II. Institut für Theoretische Physik, Universität Hamburg, 1990.

[Por93] A. Pordt. Renormalization Theory for Hierarchical Models. *Helv. Phys. Acta*, 66:105–154, 1993.

[PPW94] K. Pinn, A. Pordt und C. Wieczerkowski. Algebraic Computation of Hierarchical Renormalization Group Fixed Points and their ϵ-Expansions. *J. Statist. Phys.*, 77(977), 1994.

[Pur96] H.-G. Purwins. *Angewandte Physik I*, Kapitel III Signalanalyse, §1 Fourier-Transformation. Vorlesungsskript, 1995/1996.

[Rol96] J. Rolf. Störungstheoretische und numerische Methoden zur Beschreibung von Renormierungsgruppenfixpunkten und -trajektorien. Diplomarbeit, Westfälische Wilhelms-Universität Münster, 1996.

[RW] J. Rolf und C. Wieczerkowski. The Hierarchical ϕ^4-Trajectory by Perturbation Theory in a Running Coupling and its Logarithm. *hep-lat/9508031.*

[Ryd96] L. H. Ryder. *Quantum Field Theory.* Cambridge University Press, 1996.

[Slo] N. Sloane. Sloane's On-Line Encyclopedia of Integer Sequences. *http://www.research.att.com/~njas/sequences/eisonline.html.*

[Sma80] D.R. Smart. *Fixed Point Theorems.* Cambridge University Press, 1980.

[Wal] W. Walter. *Einführung in die Theorie der Distributionen*, Kapitel §1 IV. Beispiele, Seiten 3–4.

[Weg] F. J. Wegner. The Critical State, General Aspects. In C. Domb und M. S. Grenn, Editoren, *Phase Transitions and Critical Phenomena*, volume 6.

[Wie97a] C. Wieczerkowski. Construction of the hierarchical ϕ^4-trajectory. überarbeitete Version *hep-lat/9809050*, 1997.

[Wie97b] C. Wieczerkowski. Renormalized $g - \log(g)$ double expansion for the invariant ϕ^4-trajectory in three dimensions. *Nucl. Phys.*, B(506):468–482, 1997.

[Wie97c] C. Wieczerkowski. Running Coupling Expansion for the Renormalized ϕ^4_4-Trajectory from Renormalization Invariance. *J. Statist. Phys.*, 89(5/6), 1997.

[Wie97d] C. Wieczerkowski. The renormalized ϕ^4_4 trajectory by perturbation theory in the running coupling (I). The discrete renormalization group. *Nuclear Physics*, B(488):441–465, 1997.

[Wie98] C. Wieczerkowski. The renormalized ϕ^4_3-trajectory in the block spin renormalization group by perturbation theory in a running coupling. unveröffentlicht, 1998.

[Wil71] K. G. Wilson. Renormalization Group and Critical Phenomena. I+II. *Physical Review*, 1971.

[WK74] K. G. Wilson und J. Kogut. The renormalizaion group and the ϵ expansion. *Physics Letters*, C(12):75–200, 1974.

Danke!

Ich möchte allen danken, die mich beim Erstellen dieser Arbeit unterstützt haben. Das interessante Thema stellte mir Christian Wieczerkowski, der immer ein offenes Ohr für meine Ideen und Probleme hatte. Kreative Diskussionen führte ich mit den Teilnehmern der *sci.math*-newsgroup. Mein besonderer Dank gilt Robert Israel, Edward C. Hook und Robin Chapman. Meinen Mitstreitern aus Zimmer 411 - Bernd, Katrin und Jimmy - danke ich für die freundschaftliche Atmosphäre in unserem Büro. Für das sorgfältige Korrekturlesen richtet sich mein Dank an meine Freundin Claudia, meinen Vater, Bernd, Martin und Johannes. Für die Bereitstellung seines Druck-Accounts danke ich Christoph.

Ach ja, mein besonderer Dank gilt allen, die ich vergessen habe und es verdient hätten, hier zu stehen.

Hiermit versichere ich, daß ich diese Arbeit ohne fremde Hilfe verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Münster, im Oktober 1998

POLYOMINOES AND ANIMALS: SOME RECENT RESULTS[1]

M. DELEST[2]
LaBRI[3]
UNIVERSITE BORDEAUX I[4]

Abstract. We give a survey of recent works relating algebraic languages and formal power series with the enumeration of polyominoes (and animals). More precisely, encoding these structures with words yields new exact results.

1 - INTRODUCTION

Let Ω be a class of combinatorial objects. Let us suppose that they are enumerated by the integer a_n according to the value n of some parameter p. Let us further suppose that the corresponding generating function $f(t) = \Sigma_{n\geq 0}\, a_n\, t^n$ is *algebraic*.

M.P. Schützenberger's methodology in [48, 49] consisting in first constructing a bijection between the objects Ω and the words of an *algebraic language*, accounts for the explanation for the algebraic nature of the generating function. Let ω be an object in Ω. Then the parameter p of ω turns to be a number of letters in the corresponding word coding of ω. This methodology was first, illustrated by R. Cori [16], then by R. Cori and B. Vauquelin [17] about Tutte formulas on planar maps. The reader will find an introduction to the topic in [10, 30] and a synthesis by X. Viennot in [52]. Recently this method has been effectively used to code and count *polyominoes* which can be described as a finite connected union of *cells* (unit squares) in the plane $\mathbb{N}\times\mathbb{N}$; see [31] for instance. A polyomino is displayed in Figure 1.

---

[1] This work was partially supported by the "PRC de Mathématiques et Informatique".

[2] e-mail: maylis@geocub.greco-prog.fr

[3] Laboratoire Bordelais de Recherche en Informatique, Unité Associée au Centre National de la Recherche Scientifique n°726.

[4] Département d'Informatique, U.F.R. De Mathématiques et Informatique, 351 Cours de la Libération, 33405 TALENCE CEDEX, FRANCE.

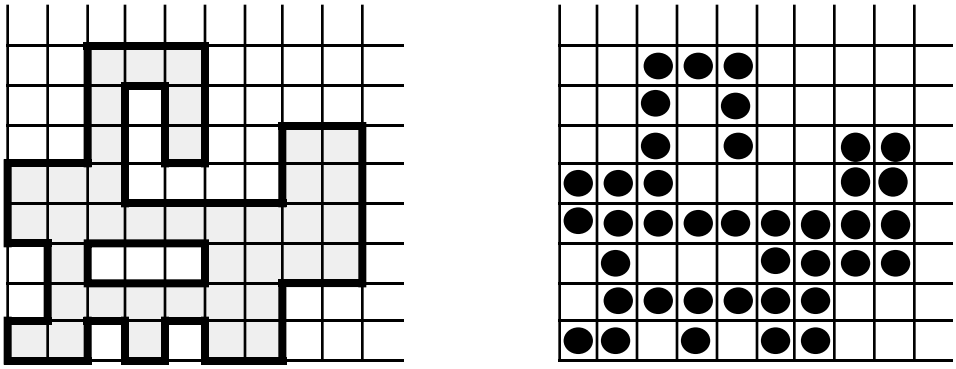


Figure 1. A polyomino and an associated animal.

The most often studied parameters are the *perimeter* which is the length of the border of the polyomino and the *area* , which is the number of cells.

Counting polyominoes is a problem in combinatorics which more often than not remains unsolved. Yet, some exact formulas dependent on one parameter only (e.g. either the perimeter or the area) are proved for some particular types of polyominoes. The reader is referred to [37, 38] for examples . But all the research on polyominoes so far has led one to believe that it is a harder problem when it comes to solve the distribution for two parameters at the same time (e.g. both the perimeter and the area).

This problem is also well-known in statistical physics. Usually physicists consider animals instead of polyominoes, an equivalent object obtained by taking the center of each elementary cell (see Figure 1).

We give below several examples in which Schützenberger's methodology has solved open problems in the field of polyominoes.

We shall begin with a brief review of the problems about polyominoes and also introducing the methodology. We shall end it with new features.

2 - POLYOMINOES AND ANIMALS

Studying polyominoes has a long set of problems. Their study is connected to partition problems, but the first book on this subject is due to S. Golomb [31] in 1965. It was preceded by some papers of M.Gardner in 1958 in the Scientific American [29]. See also the nice paper of Klarner: "My life among polyominoes" [37]. There are two classes of problems when dealing with polyominoes. The first one aims at enumerating them according to the perimeter and/or the area, and the second at spanning the plane with a set of polyominoes having a given area [32, 53].

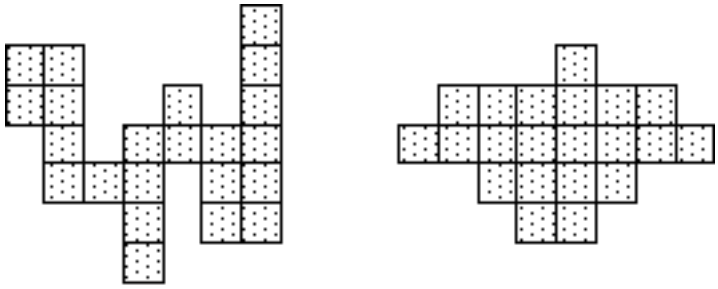


Figure 2. Some kind of polyominoes.

This does not lead to enumeration problems but rather to an algorithms allowing us to obtain a polyomino by spanning it with a smaller ones [5, 6], by superimposing rectangles [53]. Here are some possible applications:

- design of VLSI [14], the shadow of a VLSI circuit is a polyomino,
- storage of images [1, 13], the periphery is a polyomino.

We are interested in enumerating polyominoes. Generally speaking, only asymptotic results are known, the latest ones being Guttman's [34]. Thus, many people take particular polyominoes into account in order to get some approaches to the general problem. To describe particular cases, let us define a *column* (resp. *row*) of a polyomino as the intersection with an infinite vertical (resp. horizontal) strip of cells. A polyomino is a *column-convex* (resp. *row-convex)* if every column (resp. row) is connected. It is *convex* if it is both row- and column-convex. See the examples in Figure 2.

An *animal* is a set of points of $\mathbb{N}\times\mathbb{N}$ such that every pair of points of the animal can be connected by a path (sequence of points) included in the animal and having elementary steps North, East, South and West. Animals are related to the percolation problem and a lot of results have been published on this subject [45bis]. Physicists attempt to find some relations for the number a_n of animals having area or perimeter n. They look for asymptotic results in the form $a_n \approx \mu^n n^{-\theta}$. The exponent θ is called the *universality class* of the model and n the connecting constant.

Recently, the interest was in *directed* animals. They are related to some gas lattice models. An animal is said to be *directed*, if it contains a set of s points (called roots or source points) lying on the line x+y=s-1, such that any other point in the animal can be reached from one of the roots, by a path making only North or East steps in the lattice plane within the animal (see Figure 3). The surprising result was that exact results can be found for this class of animals [41, 25, 26, 36]. See [51] for a survey.

Note that polyominoes are obtained from animals by placing a unit square with vertices at integer points for each point of the animal. Thus, we shall say that a polyomino is directed if the associated animal is directed and in the following we shall

only use the word polyomino. Let P be a polyomino. The enumeration is made according to the following parameters:

- the *bond* perimeter $\mathfrak{p}(P)$, that is the length of the border of the polyomino,
- the *site* perimeter $\mathfrak{s}(P)$, that is the number of squares (resp. unit cells) outside and adjacent to the boundary of the polyomino (resp. animal),
- the *area* $\mathfrak{a}(P)$, that is the number of squares (resp. unit cells) of the polyomino (resp. animal).

3 - SCHUTZENBERGER'S METHODOLOGY

Let $X = \{x_1, x_2, ..., x_k\}$ be an alphabet. We denote by X^* the free monoid generated by X, that is, the set of words (finite sequences of letters from X). The *empty word* is denoted by ε. The number of occurrences of the letter x in the word w is denoted by $|w|_x$, the length (number of letters) of w by $|w|$. Let Ω be a class of objects for which a parameter π is to be studied. The Schützenberger's methodology is based upon four steps:

1- code the objects of Ω by the words of an algebraic language $\mathfrak{A}$ preserving π,

2- write out a non-ambigous grammar $\mathfrak{G}$ generating the language $\mathfrak{A}$,

3- solve the algebraic system associated to $\mathfrak{G}$ in commutative variables getting a generating function $\mathfrak{F}$ (or a functional equation) for the language $\mathfrak{A}$,

4- compute using $\mathfrak{F}$ an exact formula or an asymptotic expression for the number of objects in Ω having a given value for the studied parameter π.

For example, let Ω be the class of stack polyominoes. A *stack polyomino* S is a convex polyomino given by two paths η and λ from (0,0) to (k,0). The path η makes only East steps. In the first part λ makes only North and East steps, then after an East step makes only South and East steps (see Figure 4).

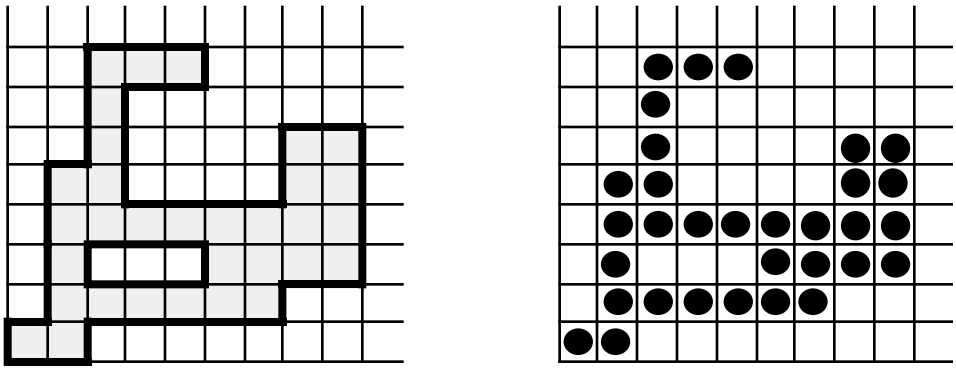


Figure 3. A directed polyomino and an associated animal.

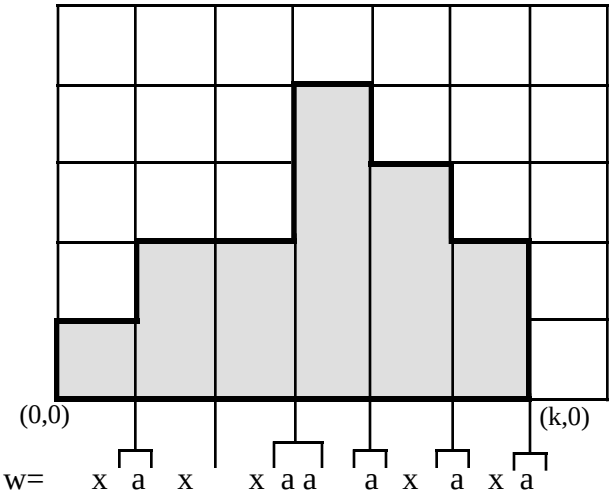


Figure 4. A stack polyomino and its coding.

The first step consists in coding the stack polyominoes using a word w of $\{x,a\}^*$ such that

(i) w is in $(x+a)^*$,
(ii) $|w|_a$ is even.

These words constitute the language $\mathfrak{L}$. This coding is immediately obtained translating the path λ: each East step is translated by the letter x excepting the middle one and each North or South step by a letter a, excepting the first and last (see figure 6). Then, we have $\mathbf{p}(S) = |w|_a + 2|w|_x + 4$.

In the second step, we write the non-commutative system of equations associated with the previous language

$$L = a\,L\,a\,L_1 + x\,L + \varepsilon \qquad (1)$$

$$L_1 = \varepsilon + x\,L_1 \qquad (2)$$

where

$$L = \sum_{w \in \mathfrak{L}} w \ .$$

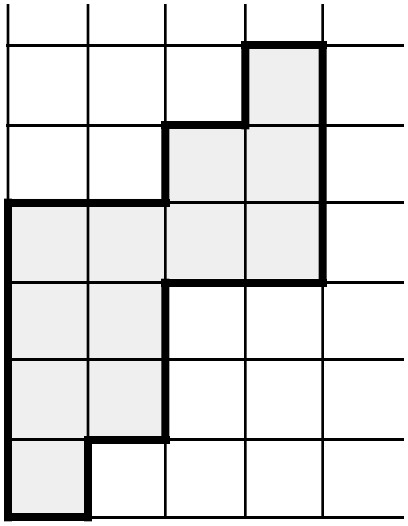
The first equation means that a non-empty word w in $\mathfrak{L}$ has the form w=xw' with w' in $\mathfrak{L}$ or $w=aw_1aw_2$ with w_1 in $\mathfrak{L}$ and w_2 in $\{x\}^*$.

In the third step, by commuting the variables, we get the commutative image of L

$$l(x,a) = \frac{1 - x}{(1-x)^2 - a^2} \ .$$

This function enumerates the stack polyominoes according to its height and width. From it, one can easily prove as an example of the fourth step, the following

Proposition 1. *The number of stack polyominoes whose perimeter is* $2p+4$ *is the Fibonacci number* F_{2p}.



Perimeter 20
Site perimeter 16
Area 12

Figure 5. A parallelogram polyomino.

4 - ENUMERATION OF POLYOMINOES USING THIS METHODOLOGY

Knuth asked the question: what is the number of convex polyominoes [38]? In 1984, Delest and Viennot enumerated these according to the perimeter [24]. They show that the number of convex polyominoes whose perimeter is 2n+8 is given by

$$p_4 = 1,\ p_6 = 2,$$

$$\text{for } n \geq 0,\ p_{2n+8} = (2n+11)\ 4^n - 4(2n+1)\binom{2n}{n}.$$

This result was recently found again by Enting and Guttmann [35] and Lin and Chang [40]. On the other hand, according to the area, there is only an asymptotic result [34]

$$g_p = 2.67564\ (2.30914)^n .$$

Following this work, since 1984, we investigated several kinds of polyominoes which are related to some properties of convexity. Firstly, we examine the *parallelogram polyominoes* which are defined by two non-intersecting paths begining and ending at the same points and making only North and East steps (see Figure 5). The number of such polyominoes with perimeter 2n+2 is known to be the Catalan number C_n . We proved [23] that the number of such polyominoes having perimeter 2n and site perimeter 2n-k is

$$C_{n,k} = \frac{2}{k+2}\binom{n-2}{k}\binom{n}{k+1}.$$

For column-convex polyominoes, it was well-known [36bis] that the generating function according to the area was rational. In [18], the generating function according to the bond perimeter is proved to be algebraic. Its expression needs a full page of formulas.

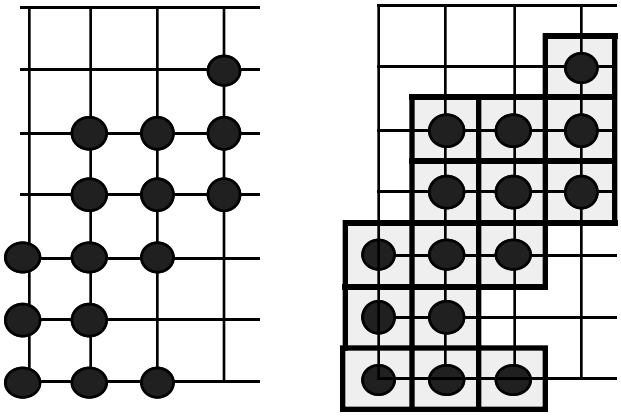


Figure 6. A fully diagonal compact animal
and the associated polyomino.

For directed animals, the first study was made by Dhar, Phani and Barma [27]. Exact results were proved successively by Dhar [26], Hakim and Nadal [36], and finally using combinatorics by Gouyou-Beauchamps and Viennot [33] and very recently by Betrema and Penaud[11]. Finally the following results are known:

- the number of directed animals having area n is

$$a_n = \sum_{i=0}^{n-1} \binom{n-1}{i} \binom{i}{\lfloor i/2 \rfloor},$$

- the number of directed animals having area n+1 with compact source is 3^n .

But no exact result concerning the perimeter is known.

In the case of directed column-convex animals [19], one can find exact results for the three parameters. The most surprising result was that the number of those having an area n is the Fibonacci number of rank 2(n-1). For fully diagonal compact animals [19], (i.e. directed and with diagonals compact (see figure 6)), V. Privman and N.M. Svrakic [47] gave the generating function according to the area. In [20], we gave it according to the two perimeters.

Also the most surprising result was that the number of such polyominoes having one root and a site perimeter equal to n+1 is

$$d_n = \frac{1}{2n+1} \binom{3n}{n}.$$

This number is the number of ternary trees having n internal nodes. This result has also been recently proved by Penaud [44].

In fact in a lot of cases, exact results according to the perimeter are well-known and according to the area there is only asymptotic or no result. In other cases, the situation is just the opposite. For example, the generating function for column-convex polyominoes according to the area is rational but the one according to the perimeter is algebraic. This has set us wondering. We have noted that the bijection between

polyominoes and algebraic languages preserve the parameter area when the coding is made according to the perimeter. Thus, since 1987, we search for some methods relating area and perimeter in polyominoes enumeration. We shall show in the last paragraph an extension of the Schützenberger methodology allowing to deduce the generating function according to the two parameters together. But first, we give one more simple example.

5 - ANOTHER EXAMPLE: THE PARALLELOGRAM POLYOMINOES

In this section, we explain how to get a coding for parallelogram polyominoes preserving the four parameters [24, 18]. A *path* is a sequence of points in $\mathbb{N}\times\mathbb{N}$. A *step* of a path is a pair of two consecutive points in the path. A *Dyck path* is a path $w = (s_0,s_1,...,s_{2n})$ such that $s_0 = (0,0)$, $s_{2n} = (2n,0)$, having only steps North-East $(s_i=(x,y),s_{i+1}=(x+1,y+1))$ or South-East $(s_i=(x,y),s_{i+1}=(x+1,y-1))$. A *peak* (resp. *trough*) is a point s_i such that the step (s_{i-1},s_i) is North-East (resp. South-East) and the step (s_i,s_{i+1}) is South-East (resp. North-East). The *height* $h(s_i)$ of a point s_i is its ordinate.

A *Dyck word* is a word $w \in \{x, \bar{x}\}^*$ satisfying the following two conditions:

(i) $|w|_x = |w|_{\bar{x}}$,

(ii) for every factorization $w = uv$, $|u|_x \geq |u|_{\bar{x}}$.

Classically, a Dyck path having length 2n is coded by a Dyck word of length 2n, $w=x_1\ldots x_{2n}$: each North-East (resp. South-East) step (s_{i-1}, s_i) corresponds to the letter $x_i = x$ (resp. $x_i = \bar{x}$). The peaks (resp. troughs) of a Dyck path correspond with the factors $x\,\bar{x}$ (resp $\bar{x}\,x$) of the associated Dyck word. The Dyck path shown in Figure 7 is coded by the Dyck word

$$w = x\,x\,x\,x\,\bar{x}\,\bar{x}\,x\,\bar{x}\,\bar{x}\,\bar{x}\,x\,x\,\bar{x}\,x\,x\,\bar{x}\,\bar{x}\,\bar{x}\ .$$

A parallelogram polyomino P can be defined by the two sequences of integers $(a_1,\ldots, a_n)$ and $(b_1,\ldots, b_{n-1})$, where a_i is the number of cells belonging to the i^{th} column and (b_i+1) is the number of cells adjacent to columns i and i+1.

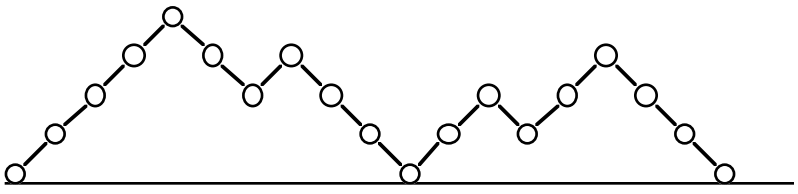


Figure 7. A Dyck path.

The Dyck word $\mu(P)$ is the Dyck word associated to the Dyck path having n peaks, whose heights (resp. troughs) are $a_1,\ldots, a_n$ (resp. $b_1,\ldots, b_{n-1}$). Note that μ associates the parallelogram polyomino of figure 5 to the Dyck path of figure 6. It is very easy to prove that μ is a bijection preserving the four parameters:

- if $\mathbf{p}(P) = 2n+2$ then $|\mu(P)| = 2n$,
- if $\mathbf{s}(P) = k$ then $|\mu(P)| - |\mu(P)|_{\bar{x}\bar{x}x} - |\mu(P)|_{\bar{x}xx} = k$,
- if $\mathbf{a}(P) = r$ then the sum of the heights of the peaks in $\mu(P)$ is r,
- if the width of P is h then $\mu(P)$ has h factors $x\bar{x}$.

In the second step we write the non commutative equation associated to the language

$$D = x\,\bar{x} + x\,D\,\bar{x} + x\,\bar{x}\,D + x\,D\,\bar{x}\,D.$$

From this equation, taking the commutative image, it is easy to prove that the number of such polyominoes having a bond perimeter 2n+2 is the Catalan number

$$C_n = \frac{1}{n+1}\binom{2n}{n}.$$

Let us explain now how to get the generating function according to the bond perimeter and the width. First marking with a letter, say t, every factor $x\,\bar{x}$, gives

$$D = x\,t\,\bar{x} + x\,D\,\bar{x} + x\,t\,\bar{x}\,D + x\,D\,\bar{x}\,D. \qquad (3)$$

From this point, take the commutative image and apply the morphism $\eta(x)=\eta(\bar{x})=x$, $\eta(t)=t$.Then we get the equation in commutative variables

$$d(x,t) = x^2\,t + x^2\,d(x,t) + x^2\,t\,d(x,t) + x^2\,d^2(x,t)$$

in which

$$d(x,t) = \sum_{h\geq 0}\sum_{n\geq 0} d_{n,h}\; x^{2n}\,t^h$$

and $d_{n,h}$ is the number of parallelogram polyominoes having width h and perimeter 2n+2. From this it is easy to deduce that

$$d_{n,h} = \frac{1}{n}\binom{n}{h}\binom{n}{h-1}.$$

We will show in the last paragraph a transformation which permits us to take into account the area using the last equation (3).

6 - q-SERIES AND COMPILING

Let P be a polyomino and let us suppose that it is coded by a word w such that $|w|$ is the perimeter of P. Let $\mathbf{q}(w)$ be $q^{\mathbf{a}(P)}$ where $\mathbf{a}(P)$ is the area of P. Let us consider the formal power series

$$\sum_{w \in L} \mathcal{Q}(w)\ w.$$

Taking the commutative image, we get an enumerating function which turns to be a series in two variables

$$f(x;q) = \sum_{n \geq 0} \sum_{p \geq 0} f_{n,p}\ x^n\ q^p$$

in which $f_{n,p}$ is the number of polyominoes whose perimeter is 2n and area is p. Note that, such a generating function is related to q-series in combinatorics. There is a vast literature on q-calculus and q-series. A nice introduction to the subject can be found in the paper of D. Foata [28]. We just give, here, few features

The q-analogue of an integer n is the polynomial

$$[n] = 1 + q + q^2 + \ldots + q^{n-1},$$

and the q-analogue of n factorial is

$$[n]! = \prod_{i=1}^{n} [i]\ .$$

In some way, a q-series is a series s in $\mathbb{C}[[X,q]]$,

$$s(x\ ;\ q) = \sum_{n \geq 0} \alpha_n\ (q)\ x^n$$

where $\alpha_n(q)$ is some function in $\mathbb{C}[[q]]$ in which the classical q-analog [n] comes up. The recent book by G.E. Andrews [2] introduces one to some applications of q-calculus to number theory and physics. A very fruitful way of getting some combinatorial interpretation of q-analogues of classical numbers is by replacing the ordinary counting of the corresponding objects by q-counting. If C is a set of objects, the cardinality of C is

$$\left|C\right| = \sum_{x \in C} 1\ .$$

A q-counting of the elements of C will be the formal power series

$$\left|C\right|_q = \sum_{x \in C} q^{s(x)}$$

where s is a statistics on the elements of C.

Just what we need now is to have a mean relating grammars to q-series. In other words knowing the word coding the polyomino, we must construct its translation which is a word "shuffled" with letter q.

In computer science, the compiler theory, more precisely the attribute grammars which were introduced by Knuth [39], permits to associate a translation to a word of an algebraic language. The interest of the method is that every translation is defined locally on every rule (every monomial) of the grammar (equations). Thus the problem of finding recurrences on a polyomino according to the area is transformed in a very local problem on some particular configurations of the polyomino.

7 - q-GRAMMARS AND ENUMERATION

In [22], we define what we call a q-grammar. For short, just consider that we associate to every monomial of a non commutative equation a translation function τ called *attribute*. Then the pair (S,τ) where S is the non commutative system of equations is called a q-grammar . The q-analogue of the enumerating function L (denoted by ${}^{q}L$) is the series in $\mathbb{B}\langle\langle X \cup \{q\}\rangle\rangle$ defined by

$$ {}^{q}L = \sum_{w \in L} \tau(w). $$

The attribute τ is such that if we substitute to each q the value 1 then we merely get the word w. In many cases, $\tau(w)$ will appear as a shuffle of the word w and a word of $\{q\}^*$. Similarly, the function ${}^{1}L$ is merely the enumerating function of L.

The commutative image of the series ${}^{q}L$ is the series over $X \cup \{q\}$ defined by

$$ {}^{q}l(X) = \sum_{i_1 \geq 0, \ ..., \ i_k \geq 0} \lambda_{i_1, \ ..., \ i_k}(q) \, x_1^{i_1} \ ... \ x_k^{i_k} \ . $$

The coefficient $\lambda_{i_1,..., i_k}(q)$ is in $\mathbb{C}[\{q\}]$ and often rational in q in our examples. The series ${}^{q}l(X)$ is clearly a q-series. Therefore, it ends up with being a natural way of relating a q-series to an algebraic ordinary generating function. Now we give two very simple examples. First in the case of stack polyominoes, we write the associated attribute to each monomial of the system of equations (1) and (2).

$$ \tau(L) = q^{|\tau(L)|} x \ a \ \tau(L) \ a \ \tau(L_1), \text{ (associated to } L \rightarrow a \ L \ a \ L_1) $$
$$ \tau(L) = q \ x \ \tau(L), \qquad \text{(associated to } L \rightarrow x \ L) $$
$$ \tau(L) = \varepsilon \ , \qquad \text{(associated to } L \rightarrow \varepsilon) $$
$$ \tau(L_1) = q \ x \ \tau(L_1), \qquad \text{(associated to } L_1 \rightarrow x \ L_1) $$
$$ \tau(L_1) = \varepsilon. \qquad \text{(associated to } L_1 \rightarrow \varepsilon) $$

From [22], it can be easily proved that ${}^{q}l(x,a)$ is a solution of the system

$$ {}^{q}l(x,a) = qx \ {}^{q}l(x,a) + a^2 \ {}^{q}l(xq,a) \ {}^{q}l_1(x,a) + \varepsilon \ , $$
$$ {}^{q}l_1(x,a) = qx \ {}^{q}l_1(x,a) + \varepsilon. $$

By solving this system, we get the following

Proposition 2. *The number of stack polyominoes having perimeter* $2p+2$ *and area* n *is the coefficient of* $x^p q^n$ *in the* q*-series*

$$ S(x;q) = \sum_{k \geq 0} \frac{x^{k+1} \, q^{k+1}(1-xq^{k+1})}{\prod_{i=1}^{k+1}(1-xq^i)^2} \ . $$

Using the equation (3), it is also easy to deduce a q-equation for parallelogram polyomino. First, we write the associated attribute to each monomial

$$\tau(D) = q\ x\ t\ \bar{x}\ , \qquad \text{(associated to } D \rightarrow x\ t\ \bar{x})$$

$$\tau(D) = q^{|\tau(L)|}\ t\ x\ \tau(D)\ \bar{x}, \qquad \text{(associated to } D \rightarrow x\ D\ \bar{x})$$

$$\tau(D) = q\ x\ t\ \bar{x}\ \tau(D)\ , \qquad \text{(associated to } D \rightarrow x\ t\ \bar{x}\ D)$$

$$\tau(D) = q^{|\tau(D)|}\ t\ x\ \tau(D)\ \bar{x}\ \ \tau(D). \text{ (associated to } D \rightarrow x\ D\ \bar{x}\ \ D)$$

From this, we deduced [21]

Theorem 11. *The number of skew Ferrers diagrams having area n and p columns is the coefficient of* $t^p q^n$ *in the q-series*

$$_qs(t) = (1-q)\ \varphi_0\left(\frac{qt}{(1-q)^2}\right)$$

where $\varphi_0(x)$ *is the quotient of two basic Bessel functions*

$$\varphi_0(x) = \frac{{}_qI_1(x)}{{}_qI_0(x)}\ ,$$

in which the basic Bessel function is defined by

$${}_qI_\nu(x) = \sum_{n=0}^{\infty} \frac{(-1)^n\ q^{\binom{n+\nu}{2}}\ x^{n+\nu}}{[n]![n+\nu]!}$$

Recently, using this method, M. Bousquet-Mélou [12] has given a generating function for convex polyominoes according to the area.

8 - CONCLUSION

We give in Figure 8, a table of authors on polyominoe enumeration which is due to Delest, Penaud and Viennot and pictured in [42]. A remarkable fact of all these codings with words is that they are very efficient on planar pictures and especially for polyominoes. An interest of these coding is the interplay between Computer Science, Combinatorics and Physics. Finally we note that most of the results were obtained using symbolic calculus (especially MAPLE from Waterloo University) and using also the book of N.J. Sloane [50].

REFERENCES

[1] E. AHRONOVITZ et M.HABIB, CICC: Un logiciel de compression d'images par codes de contours, Rapport de l'Ecole de Mines, (1986) Saint-Etienne.

[2] G.E. ANDREWS, *q-Séries: their development and application in analysis, number theory, combinatorics, physics, and computer algebra,* AMS, Library of congress Cataloging-in-Publication Data (1986).

Polyomino	Perimeter	Area
Stacks	Exercise	Euler 1748, Gauss 1863 Sylvester 1884 Temperley 1952, 1956 Wright 1968, Derrida, Nadal 1984
Parallelogram	Polya 1969 Kreweras 1970 Delest, Gouyou-Beauchamps, Vauquelin 1987 (site and bond)	(particular case of *quasi-partitions:* Auluck 1951, Andrews 1981)Polya 1969, Gessel 1980 Delest, Fedou 1988 (area and width)
Directed convex	Chang, Lin 1988 (Width and length) Bousquet-Mélou 1990	Bousquet-Mélou,Viennot 1990 (area, width and length)
Convex	Delest, Viennot 1984 Kim, Stanton 1988 Enting, Guttmann 1988,1989 Chang, Lin 1988 Lin 1988 (width and length)	(*asymptotic results:* Klarner, Rivest 1974, Bender 1974) Bousquet-Mélou (area, width and length)
Column-convex	Delest 1987	Klarner 1965, 1967 Stanley 1978, 1986 Delest 1987 (area + width) Privman, Forgacs 1987 Privman, Svrakic 1989 (area and length)
Directed Column-convex	Delest, Dulucq 1987 (site and bond)	Delest, Dulucq 1987 Barcucci, Pinzani, Rodella 1990
Fully-diagonal Compact	Delest, Fédou 1988 (site and bond) Penaud 1990	Bhat, Bhan, Singh 1988 Privman, Svrakic 1988
Directed		Nadal, Derrida, Vannimenus 1982 Hakim, Nadal 1982 Dhar, Phani, Barma 1982 Dhar 1982, 1983 Viennot 1985 Gouyou-Beauchamps, Viennot 1988 (area and width) Betrema, Penaud 1990

Figure 8. Exact enumeration of polyominoes.

[3] E. BARCUCCI, R. PINZANI, E. RODELLA, Some properties of binary search networks, Research rapport 1/90, Dipartimento di Sistemi e Informatica, Université de Florence, 1990.

[4] R. J. BAXTER, Exactly solved models in statistical mechanics, Academic Press, New-York, 1982.

[5] D. BEAUQUIER et M. NIVAT, Tiling with polyominoes, rapport LITPn° 88-66, Université Paris VII, 1988 .

[6] D. BEAUQUIER et M. NIVAT, Tiling the plane with one polyomino, rapport , LITP n° 88-66, Université Paris VII, 1989 .
[7] E. BENDER, Convex n-ominoes, Discrete Math 8 (1974), 219-226.
[8] E. BENDER, Asymptotic methods in enumeration, SIAM review (1974), 485-515.
[9] C. BERGE, C.C. CHEN, V. CHVATAL, C.S. SEOW, Combinatorial properties of polyominoes, Combinatorica 3 (1981), 217-224.
[10] J. BERSTEL et C. REUTENAUER, *Les séries rationnelles et leurs langages*, Masson, Paris, 1984.
[11] J. BETREMA, J.G. PENAUD, Animaux et arbres guingois, rapport LaBRI n°90-60, Université de Bordeaux I.
[12] M. BOUSQUET-MELOU, Codage des polyominos convexes et équations pour l'énumération selon l'aire, soumis à publication.
[13] R. CEDERBERG, On the coding, processing and display of binary images, Linköping Studies in Science and Technology, Dissertation n°57, Linköping, Sweden, 1980.
[14] S. CHAIKEN, D.J. KLEITMAN, M. SAKS, J. SHEARER, Covering regions by rectangles, SIAM J. Allg. Disc. Meth., 2 (1981), 394-410.
[15] J.H.CONWAY et J.C.LAGARIAS, Tiling with polyominoes and combinatorial group theory, J.C.T. A 53 (1990), 183-208.
[16] R. CORI, *Un code pour les graphes planaires et ses applications*, Astérisque, Soc. Math. France n° 27 (1975).
[17] R. CORI et B. VAUQUELIN, Planars maps are well labeled trees, Can J. Math. 33 (1981), 1023-1042.
[18] M.P. DELEST, Generating functions for column-convex polyminoes, J.C.T A, 48 (1988) 12-31.
[19] M.P. DELEST, S. DULUCQ, Enumeration of directed column-convex animals with given perimeter and area, rapport LaBRI n° 86-15, Université de Bordeaux I.
[20] M.P.DELEST, J.M. FEDOU, Exact formulas for fully compact animals, rapport LaBRI n° 89-06.
[21] M.P.DELEST, J.M. FEDOU, Enumeration of Skew Ferrers diagrams, to appear in Discrete Math.
[22] M.P.DELEST, J.M. FEDOU, Enumeration of polyominoes using attribute grammars, to appear in Proceedings Workshop on attribute grammars 1990, INRIA, Paris.
[23] M. DELEST, D. GOUYOU-BEAUCHAMPS ET B. VAUQUELIN, Enumeration of parallelogram polyominos with given bond and site parameter, Graphs and Combinatorics, 3 (1987) 325-339.
[24] M.P.DELEST, G.VIENNOT, Algebraic langages and polyominoes enumeration, Theor. Comp.Sci. 34 (1984), 169-206 North-Holland.
[25] D. DHAR, Equivalence of the two-dimensional directed animal problem to Baxter hard-square lattice-gas model, Phys. Rev Lett. 49 (1982), 959-962.
[26] D. DHAR, Exact solution of a directed-site animals enumeration in 3 dimensions, Phys. Rev Lett. 59 (1983), 853-856.
[27] D. DHAR, M.K. PHANI, M. BARMA, Enumeration of directed site animals on two-dimensional lattices, J.Phys.A : Math Gen.15, (1982), L 279 -L 284.
[28] D. FOATA, Aspects combinatoires du calcul des q-séries, Compte rendu du Séminaire d'Informatique Théorique LITP année 1980-1981,Universités Paris VI Paris VII, 37-53.
[29] M. GARDNER, Mathematical games, Scientific American, 1958, Sept. 182-192, Nov 136-142.
[30] J. GOLDMAN, Formal langages and enumeration, J. of Comb. Th. A 24 (1978),318-338.
[31] S.GOLOMB, *Polyominoes*, Scribner, New York, (1965).
[32] S. GOLOMB, Polyominoes Which Tile Rectangles, J. of Comb. Th., A 51,117-124, (1989).
[33] D. GOUYOU-BEAUCHAMPS, X.G. VIENNOT, Equivalence of the two dimensional directed animal problem to a one-dimensional path problem, Advances in Applied Mathematics 9, 334-357 (1988).

[34] A.J. GUTTMANN, On the number of lattice animals enbedable in the square lattice, J. Phys. A:Math. Gen. 15 (1982), 1987-1990.
[35] A.J. GUTTMANN, I.G. ENTING, The number of convex polygons on the square and honeycomb lattices, J. Phys. A:Math. Gen. 21 (1988), 467-474.
[36] V. HAKIM, J.P. NADAL, Exact result for 2D directed lattice animals on a strip of finite with, J. Phys. A: Math. Gen. 16 (1983), L 213-L 218.
[36bis] D.A. KLARNER, Some results concerning polyominoes, Fibonacci Quart. 3 (1965), 9-20.
[37] D.A. KLARNER, My life among polyominoes, in *The Mathematical Gardner*, 243-262, Wadsworth, Belmont CA, 1981.
[38] D.A. KLARNER, R.L. RIVEST, Asymptotic bounds for the number of convex n-ominoes, Discrete Maths 8 (1974), 31-40.
[39] D.E. KNUTH, Semantics of context-free langages, Math. Sys. Th. 2, 127-145.
[40] K.Y. LIN, S.J. CHANG, Rigourous results for the number of convex polygons on the square and honeycomb lattices, J. Phys. A: Math. Gen. 21 (1988) 2635-2642..
[41] J.P. NADAL, B.DERRIDA, J. VANNIMENUS, Directed lattices animals in 2 dimension: numerical and exact results, J. Physique 43 (1982), 1561.
[42] J.P. NADAL, B.DERRIDA, J. VANNIMENUS, Directed diffusion-controlled aggregation versus directed animals, preprint (1983).
[43] G. POLYA, On the number of certain lattice polygons, J. Comb. Theory, 6 (1969) 102-105.
[44] J.G. PENAUD, Animaux dirigés diagonalement convexes et arbres ternaires, rapport LaBRI n°90-62, Université de Bordeaux I.
[45] J.G. PENAUD, Arbres et Animaux, Mai 1990, Université de Bordeaux I.
[46] V. PRIVMAN, G. FORGACS, Exact solution of the partially directed compact lattice animal model, J. Phys. A: Math Gene. 20 (1987) L543-547.
[47] V. PRIVMAN, N. M. SVRAKIČ, Exact generating function for fully directed compact lattice animals, Physical Review Letters vol. 60, n° 12 (1988) 1107-1109.
[48] M.P.SCHÜTZENBERGER, Certain elementary families of automata, Proc. Symp. on Mathematical Theory of Automata (Polytechnic Institute of Brooklyn,1962) pp. 139-153.
[49] M.P.SCHÜTZENBERGER, Context-free langages and pushdown automata, Information and Control 6 (1963), 246-264.
[50] N.J. SLOANE, *A handbook of integer sequences*, Academic Press, New-York, 1979.
[51] X.G.VIENNOT, Problèmes combinatoires posés par la physique statistique, Séminaire Bourbaki n° 626, 36ème année, in Astérisque n°121-122 (1985) 225-246 Soc. Math. France.
[52] X.G. VIENNOT, Enumerative combinatorics and algebraic langages, Proceedings FCT'85, ed. L. Budach, Lecture Notes in Computer Science n°199, Springer-Verlag, Berlin, 1985, 450-464.
[53] H.A.J. WIJSHOFF, J. VAN LEEUWEN, Arbitrary versus periodic storage schemes and tessellations of the plane using one type of polyomino, Information and Control, 62 (1984), 1-25.

PATTERN AVOIDANCE IN INVOLUTIONS

ELIZABETH WULCAN

ABSTRACT. This work concerns pattern avoidance in involutions. We give a complete solution for the number of involutions avoiding one or two classical 3-patterns, mainly by relating these to well known combinatorial structures such as Dyck paths and Young tableaux. The results for single 3-patterns were previously obtained by Simion and Schmidt. However, we give new proofs in most cases. We also give some results for the number of involutions avoiding generalised patterns.

CONTENTS

1. INTRODUCTION

Classically a k-pattern p is a permutation of $[k] = \{1, 2, \ldots, k\}$ and a permutation π of $[n]$ is said to have an occurrence of p if π has a subword whose letters are in the same relative order as the letters of p. If π has no occurrences of p, we say that π avoids p. For example $\pi = 52134$ avoids $p = 132$ whereas $\pi = 41253$ has two occurrences of p (the subwords 153 and 253).

In the last decades there have been plenty of articles written on the subject of patterns and in particular on pattern avoidance. One of the earliest results worth mentioning is found in Knuth [7], where it is established that for all 3-patterns p, the number of permutations of $[n]$ that avoid p equals the nth Catalan number. In Simion and Schmidt [10], multiavoidance, that is when two or more patterns are simultaneously avoided, was considered and a full solution for the case of double avoidance was given. Simion and Schmidt also treated pattern-avoiding involutions, the topic of this work. Indeed, the results of Section 3, which concern the six classical 3-patterns, are all proven in [10]. However, we give new proofs of some of the results.

As a further development of the concept of patterns, Babson and Steingrímsson [3] introduced generalised patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation for the pattern to occur. Avoidance of generalised patterns has been studied by, for example, Claesson [1], Kitaev [5], [6] and Claesson and Mansour [2]. In Section 4 we give some results for involutions avoiding generalised patterns.

Finally, in Section 5 we investigate double avoidance and give a complete solution for the number of involutions avoiding any two classical 3-patterns.

2. Preliminaries

Before starting the investigation on pattern-avoiding involutions we introduce the main concepts that will be used in this work. To start with, an *alphabet* X is a nonempty set of *letters* and a *word* over X is a finite sequence of letters from X. We denote the *empty word*, that is the word with no letters, by ϵ. Let $x = x_1x_2\cdots x_n$ be a word over X. A *subword* of x is a word $v = x_{i_1}x_{i_2}\cdots x_{i_k}$, where $1 \leq i_1 \leq i_2 \leq \cdots \leq i_k \leq n$. A *segment* is a word $v = x_ix_{i+1}\cdots x_{i+k}$. We define the *length of* x, denoted by $|x|$, to be the number of elements in x.

2.1. **Permutations.** Let $[n] = \{1, 2, \ldots, n\}$. A *permutation* π of $[n]$ is a bijection from $[n]$ to $[n]$. However, we sometimes refer to permutations of a subset A of $[n]$. This should be interpreted as a bijection from A to A. There are several different notations for the permutations, suitable for different purposes. A permutation π is usually seen as the word

$$\pi = \pi(1)\pi(2)\cdots\pi(n).$$

Another way of writing the permutation is given by the two line (or French) notation

$$\pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ a_1 & a_2 & \ldots & a_n \end{pmatrix}.$$

This means that $1 \mapsto a_1$, $2 \mapsto a_2$ et cetera, hence the permutation is unaffected by rearrangement of the columns, which makes it easy to find the inverse of π. Indeed

$$\pi^{-1} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ 1 & 2 & \dots & n \end{pmatrix}.$$

Rearranging the top line in increasing order gives π^{-1} as a word in the bottom line.

We will also use a third notation, the cycle form, where the letters in $[n]$ are grouped together in cycles. A cycle $(a_1 a_2 \cdots a_k)$ means that $a_i \mapsto a_{i+1}$ for $i < k$ and that $a_k \mapsto a_1$. Fixed points, that is those i for which $i \mapsto i$, are conventionally omitted. As will be shown in the example below, the cycle notation is generally not unique.

We denote the set of permutations of $[n]$ by $\mathcal{S}_n$.

Example 1. Consider the permutation

$$\pi = \begin{cases} 1 \to 3 \\ 2 \to 4 \\ 3 \to 6 \\ 4 \to 2 \\ 5 \to 5 \\ 6 \to 1 \end{cases}$$

We write it as the word

$$\pi = 346251,$$

or in the two line notation;

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 6 & 2 & 5 & 1 \end{pmatrix},$$

from which we get the inverse of π as

$$\pi^{-1} = \begin{pmatrix} 3 & 4 & 6 & 2 & 5 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 4 & 1 & 2 & 5 & 3 \end{pmatrix}.$$

The permutation π could be written in cycle form as

$$\pi = (136)(24)$$

but we also have

$$\pi = (24)(136) = (361)(24) = (42)(613).$$

This shows that the cycle notation is not unique. Note that the fixed point 5 is not written out.

2.2. **Involutions.** An *involution* is a permutation that is its own inverse. Thus an involution consists of cycles of length 1 or 2. We let $\mathcal{I}_n$ denote the set of all involutions of $[n]$.

2.3. **Generalised patterns.** A *generalised k-pattern* p is a word of length k consisting of all the elements of $[k]$, in which two letters may or may not be separated by a dash. Consider $\pi = a_1a_2\cdots a_n$ in $\mathcal{S}_n$. We say that the subword $v = v_1v_2\cdots v_k$ is a *p-subword* of π if the v_i's are in the same relative order as the p_i's and two adjacent letters of v are adjacent in π whenever the corresponding letters of p are not separated by a dash. We also refer to v as an *occurrence of p*. If π has no occurrences of p, we say that π *avoids* p or that π is *p-avoiding*. We define $\mathcal{S}_n(p)$ and $\mathcal{I}_n(p)$ to be the set of p-avoiding permutations and involutions in $\mathcal{S}_n$, respectively, and more generally we let $\mathcal{S}_n(A) = \bigcap_{p\in A}\mathcal{S}_n(p)$, just as $\mathcal{I}_n(A) = \bigcap_{p\in A}\mathcal{I}_n(p)$. It is convenient to regard the pattern p as a function from $\mathcal{S}_n$ to $\mathbb{N}$ where $p\pi$ is defined as the number of p-subwords of π. Thus π is p-avoiding if and only if $p\pi = 0$.

Usually the term pattern refers to the type of patterns p_1-p_2-$\cdots$-p_k with dashes between each pair of adjacent letters, that is, no attention is paid to whether the letters of the permutation are adjacent or not. Those patters were the first to be defined and studied and we therefore call them classical patterns.

Example 2. Regarded as a permutation statistic (a function from $\mathcal{S}_n$ to $\mathbb{N}$), the pattern (1-2-3) counts the number of increasing subsequences of length 3. For example, the longest increasing sequence of the permutation 21543 is of length two and consequently 21543 avoids (1-2-3).

The pattern (21) counts *descents* in a permutation, that is the number of i's such that $a_i > a_{i+1}$, just as (12) counts the *ascents*, the number of i's such that $a_i < a_{i+1}$.

The pattern $p = (1\text{-}32)$ counts the subwords of the form a_i-a_ja_{j+1} such that $a_i < a_{j+1} < a_j$. The permutation 25431 has two occurrences of p, namely 254 and 243.

2.4. **Young tableaux.** A *Young tableau P of shape $(n_1, n_2, \dots n_m)$* is an arrangement of n distinct integers as an array of m left-justified rows, with n_i elements in row i, where $n_1 \geq n_2 \geq n_m \geq 0$ and $n_1+n_2+\cdots+n_m = n$. The entries of the rows and the columns must be ordered increasingly from left to right and from top to bottom, respectively. We write $P_{i,j}$ for the element in row i and column j.

Example 3. We have that

$$P = \begin{array}{|c|c|c|c|}\hline 1 & 3 & 4 & 9 \\ \hline 2 & 5 \\ \cline{1-2} 6 & 7 \\ \cline{1-2} 8 \\ \cline{1-1}\end{array}$$

is a Young tableau of shape $(4, 2, 2, 1)$ and that $P_{3,2} = 7$.

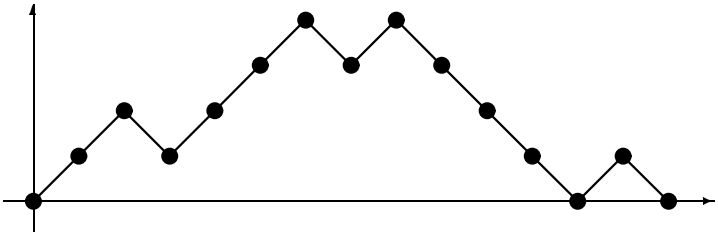


FIGURE 1. The Dyck path in Example 5

2.5. **Inversion tables.** Given a permutation $\pi = a_1 a_2 \cdots a_n$, we let $t(\pi) = (t_1, t_2, \ldots, t_n)$, where $t_i = |\{j : j > i, a_j < a_i\}|$. That is, the ith entry of t is the number of letters following the ith letter of π that are smaller than the ith letter.

A pair (a_i, a_j) is called an *inversion* of the permutation π if $i < j$ and $a_i > a_j$. Accordingly, t defined above is called the *inversion table* of π, since it gives a measure of the number of inversions that each letter of π causes.

It is easy to see that a permutation is uniquely determined by its inversion table, for a demonstration see for example Stanley [12].

Example 4. Consider $\pi = 1327654$. The corresponding inversion table is $t = (0, 1, 0, 3, 2, 1, 0)$, because there is no element smaller than 1 and there is exactly one element to the right of 3, namely 2, that is smaller than 3 et cetera.

2.6. **Dyck paths.** A *Dyck path of length 2n* is a lattice path from (0,0) to $(0, 2n)$ that consists of steps (1,1) and (1,-1) and that never goes below the x-axis. Denoting the steps (1,1) and (1,-1) by u (for up) and d (for down), a Dyck path can be written as a word over the alphabet $\{u, d\}$. The number of Dyck paths of length $2n$ is the nth *Catalan number* $C_n = \frac{1}{n+1}\binom{2n}{n}$. We denote the set of Dyck paths of length $2n$ by $\mathcal{D}_n$.

Example 5. The Dyck path of length $2 \cdot 7$ in Figure 3 is coded by the word $uuduuududdddud$.

3. Pattern avoiding involutions

We start our work on pattern-avoiding involutions by investigating the avoidance of the six classical 3-patterns. For each such pattern we generate and study $\mathcal{I}_n(p)$, when n is small. When counting these involutions we obtain the first elements of the sequences that are presented in Table 1. Our aim is to show that the results are indeed true for all n.

For odd n, when $n/2$ is not an integer, it is natural to consider $\binom{n}{n/2}$ as $\binom{n}{\lfloor n/2 \rfloor}$ or $\binom{n}{\lceil n/2 \rceil}$, since the binomial $\binom{n}{k}$ coefficients are defined only for

p	$\lvert\mathcal{I}_n(p)\rvert$
(1–2–3)	$\binom{n}{n/2}$
(1–3–2)	$\binom{n}{n/2}$
(2–1–3)	$\binom{n}{n/2}$
(2–3–1)	2^{n-1}
(3–1–2)	2^{n-1}
(3–2–1)	$\binom{n}{n/2}$

TABLE 1. Classical patterns

integer n and k. However, $\binom{n}{\lfloor n/2\rfloor} = \binom{n}{n-\lfloor n/2\rfloor} = \binom{n}{\lceil n/2\rceil}$, so there should be no ambiguities concerning the interpretation. Let $\binom{n}{n/2} := \binom{n}{\lfloor n/2\rfloor}$.

It is observed that the involutions that avoid (2–3–1) are exactly the same as those that avoid (3–1–2), at least for small n. On the other hand we see that although $|\mathcal{I}_n(p)| = \binom{n}{n/2}$ for four different patterns p, there are no two distinct patterns p and q of these, such that $\mathcal{I}_n(p) = \mathcal{I}_n(q)$. The reader may convince himself of this by studying $\mathcal{I}_n(p)$ for small n.

3.1. **Avoiding p, when p is not an involution.** We consider the case when the pattern p itself is not an involution. As noticed above an involution avoids (2–3–1) if and only if it avoids (3–1–2). In this section this will be shown to follow from the fact that the patterns are inverses of each other. First, however, we show that $\mathcal{I}_n(2\text{–}3\text{–}1)$ is counted by 2^{n-1}.

Proposition 6. *The number of involutions of* $[n]$ *that avoid* (2–3–1) *is* 2^{n-1}.

We give a general description of the elements of $\mathcal{I}_n(2\text{–}3\text{–}1)$. Note that, if $\pi = a_1a_2\cdots a_n$ is a permutation of $[n]$, where n is in position k, then π avoids (2–3–1) if and only if it can be written as $\pi = \sigma n\tau$, where $\sigma = a_1a_2\cdots a_{k-1}$ is a (2–3–1)-avoiding permutation of $[k-1]$ and $\tau = a_{k+1}a_{k+2}\cdots a_n$ is a (2–3–1)-avoiding permutation of $\{k,\dots n-1\}$. Furthermore, if π is an involution we see that since n is in position k, the letter k must be in position n, and the only (2–3–1)-avoiding permutation τ of $\{k,\dots n-1\}$ ending with k is $\tau = (n-1)(n-2)\cdots(k+1)k$, that is, these letters must be in decreasing order. Indeed, all other possible τ's will contain at least one ascent ij, where $i<j$, and ijk will then form a (2–3–1)-subword. Hence every π in $\mathcal{I}_n(2\text{–}3\text{–}1)$ is of the form $\sigma n(n-1)\cdots(k+1)k$ where σ is in $\mathcal{I}_{k-1}(2\text{–}3\text{–}1)$. Such a π can be written explicitly as

$$\pi = k_1\cdots 1k_2\cdots(k_1+1)k_3\cdots(k_{\ell-1}+1)n\cdots(k_\ell+1).$$

In other words, the involutions can be considered as divided into segments, such that

(a) each letter in segment i is smaller than every letter in segment $(i+1)$,
(b) the elements in each segment are in decreasing order.

In order to show that $|\mathcal{I}_n(2\text{-}3\text{-}1)| = 2^{n-1}$ we give four proofs, where we construct bijections from $\mathcal{I}_n(2\text{-}3\text{-}1)$ to different sets that are known to be counted by 2^{n-1}.

First proof. Let B_n be the collection of binary strings of length n. Given a binary string $x = x_1x_2\cdots x_{n-1}$ in B_{n-1}, a permutation $\pi = a_1a_2\cdots a_n$ in $\mathcal{S}_n$ is constructed inductively by letting $\pi_0 = 1$ and then, if $\pi_i = \sigma i\tau$, by letting

$$\begin{aligned} \pi_{i+1} &= \sigma i\tau(i+1), \text{ if } x_i = 0 \\ \pi_{i+1} &= \sigma i(i+1)\tau, \text{ if } x_i = 1. \end{aligned}$$

That is, the permutation π is built up by successively placing each of the elements $1,\ldots,n$ either as the last element or just before the largest element already placed. This procedure defines a mapping

$$\begin{aligned} \Phi_n : B_{n-1} &\to \mathcal{S}_n, \\ x &\mapsto \pi. \end{aligned}$$

Denote the image of B_{n-1} by A_n. Then A_n consists of all permutations of the form

$$\sigma n(n\text{-}1)(n\text{-}2)\ldots(k+1)k, \text{ where } \sigma \in A_{k-1},$$

and is easily seen to coincide with $\mathcal{I}_n(2\text{-}3\text{-}1)$, according to the description above. Since Φ_n is clearly injective we have a one-to-one correspondence between the binary strings of length $(n-1)$ and $\mathcal{I}_n(2\text{-}3\text{-}1)$, hence $|\mathcal{I}_n(2\text{-}3\text{-}1)| = |B_{n-1}| = 2^{n-1}$. □

Example 7. Consider the binary string $x = 010111 \in B_6$. Then Φ_7 maps x to $\pi = 1327654$, via π_i, for $i = 0,\ldots,6$, where

$$\begin{aligned} \pi_0 &= 1 \\ \pi_1 &= 12, \text{ since } x_1 = 0 \\ \pi_2 &= 132, \text{ since } x_2 = 1 \\ \pi_3 &= 1324, \text{ since } x_3 = 0 \\ \pi_4 &= 13254, \text{ since } x_4 = 1 \\ \pi_5 &= 132654, \text{ since } x_5 = 1 \\ \pi = \pi_6 &= 1327654, \text{ since } x_6 = 1. \end{aligned}$$

Second proof. In this proof we show the one-to-one correspondence between $\mathcal{I}_n(2\text{-}3\text{-}1)$ and the binary strings of length $(n-1)$ by constructing

a mapping Ψ_n from T_n to B_{n-1}. Here T_n is the set of inversion tables $t = (t_1, t_2, \dots, t_n)$ defined from $\pi = a_1 a_2 \cdots a_n \in \mathcal{I}_n(2\text{–}3\text{–}1)$ as

$$t_i := |\{j : j > i, a_j < a_i\}|.$$

That is, the ith entry of t is the number of letters following the ith letter of π that are smaller than the ith letter. From the appearance of $\mathcal{I}_n(2\text{–}3\text{–}1)$ it follows that the elements in T_n will be of the form

$$(k_1, k_1 - 1, \dots, 1, 0, \dots, 0, k_2, k_2 - 1, \dots, 1, 0, k_\ell, k_\ell - 1, \dots, 1, 0).$$

For example, a decreasing sequence $a_i a_{i+1} \dots a_{i+k}$ of length $(k+1)$ will give rise to the segment $(t_i, t_{i+1}, \dots, t_{i+k}) = (k, (k-1), \dots, 1, 0)$ in the corresponding inversion table $t(\pi)$.

The mapping

$$\begin{array}{rcl} \Psi_n : T_n & \to & B_{n-1} \\ t = (t_1, t_2, \dots, t_n) & \mapsto & x = x_1 x_2 \cdots x_{n-1} \end{array}$$

is now defined by

$$x_i = \begin{cases} 0 \text{ if } t_i = 0, \\ 1 \text{ if } t_i \neq 0. \end{cases}$$

It is easy to see that Ψ_n is invertible, when restricted to $(2\text{–}3\text{–}1)$-avoiding involutions. The inverse mapping is given by

$$t_i = \begin{cases} 0, \text{ if } x_i = 0, \\ s, \text{ where } (s-1) \text{ is the number of 1's following } x_i, \text{ if } x_i = 1. \end{cases}$$

A permutation is uniquely determined by its inversion table. Hence there is a one-to-one correspondence between B_{n-1} and $\mathcal{I}_n(2\text{–}3\text{–}1)$ via the inversion tables $\{T_n\}$, and $|\mathcal{I}_n(2\text{–}3\text{–}1)| = 2^{n-1}$. □

Example 8. Consider $\pi = 1327654$ from Example 7. The corresponding inversion table is $t = (0, 1, 0, 3, 2, 1, 0)$, according to Example 4. Now Ψ_7 maps $(0, 1, 0, 3, 2, 1, 0)$ onto 0101110, which is exactly the binary string x, given by the mapping Φ_7 in the first proof.

Third proof. Denote the set of subsets of $[n]$ by $\mathcal{P}_n$. We construct π in $\mathcal{I}_n(2\text{–}3\text{–}1)$ from A in $\mathcal{P}_{n-1}$ by letting the letter i be immediately preceded by a larger letter, if and only if i is in A. Because of the appearance of the elements in $\mathcal{I}_n(2\text{–}3\text{–}1)$ there is only one choice of the larger letter to precede i, namely $(i+1)$, and this algorithm for constructing π from A therefore clearly defines a bijection. Indeed, the segment $(i\ +\ k\ )(i\ +\ k\ -\ 1) \cdots\ i$ is contained in π if and only if i, $(i+1)$, …, $(i+k)$ are in A. Hence there is a one-to-one correspondence between $\mathcal{P}_{n-1}$ and $\mathcal{I}_n(2\text{–}3\text{–}1)$, so $|\mathcal{I}_n(2\text{–}3\text{–}1)| = |\mathcal{P}_{n-1}| = 2^{n-1}$. □

Example 9. Let $A = \{2, 4, 5, 6\}$. The corresponding π is 1327654. Indeed, the letter 2 is the smallest letter that is in A, and accordingly the smallest letter to be preceded by a larger letter. From this we conclude

that 1 is a fixed point and, since 3 is not in A, the decreasing sequence ending with 2 must start with 3. The letter 4 is in A as well as 5 and 6, and hence π must contain the segment 7654.

We also see from this example how to get from π to A. Considering $\pi = 1327654$ we find that exactly the letters 2, 4, 5 and 6 are preceded by larger letters, hence $A = \{2, 4, 5, 6\}$.

Porism 10. *The number of involutions in $\mathcal{I}_n(2\text{-}3\text{-}1)$ with exactly k descents is $\binom{n-1}{k}$.*

Proof. Consider the bijection from $\mathcal{P}_{n-1}$ to $\mathcal{I}_n(2\text{-}3\text{-}1)$ defined in the third proof above. A $(2\text{-}3\text{-}1)$-avoiding involution is constructed from A in $\mathcal{P}_{n-1}$ by letting i be preceded by a larger letter if and only if i is in A. Hence the number of elements in A counts the descents of π. Since there are $\binom{n-1}{k}$ ways of choosing k letters out of $[n-1]$, the result follows. □

Finally, we give a proof by showing a one-to-one correspondence between $\mathcal{I}_n(2\text{-}3\text{-}1)$ and a certain type of Dyck paths, that are easily counted.

Fourth proof (of Proposition 6). Claesson [1] gives a proof of the well-known result that $\mathcal{S}_n(2\text{-}1\text{-}3)$ is counted by the nth Catalan number, in which he defines recursively a bijective mapping Φ from $\mathcal{S}_n(2\text{-}1\text{-}3)$ to the set of Dyck paths of length $2n$. We mimic his proof and construct a mapping Φ from $\mathcal{S}_n(2\text{-}3\text{-}1)$ to the Dyck paths of length $2n$.

Consider $\pi = a_1 a_2 \cdots a_n$ in $\mathcal{S}_n(2\text{-}3\text{-}1)$ with the letter n in position k. According to the discussion on page 6 we can write $\pi = \sigma n \tau$, where $\sigma = a_1 a_2 \cdots a_{k-1}$ is a $(2\text{-}3\text{-}1)$-avoiding permutation of $[k-1]$ and $\tau = a_{k+1} a_{k+2} \cdots a_n$ is a $(2\text{-}3\text{-}1)$-avoiding permutation of $\{k+1, k+2, \ldots, n-1\}$.

Denoting the empty word by ϵ, we define $\Phi(\pi)$ recursively by

$$\Phi(\pi) = \begin{cases} \epsilon, \text{ if } \pi = \epsilon, \\ u\,(\Phi \circ \mathrm{proj})(\sigma)\,d\,(\Phi \circ \mathrm{proj})(\tau), \text{ otherwise.} \end{cases}$$

Here, $\mathrm{proj}(x)$ denotes the *projection* of the word $x = x_1 x_2 \cdots x_n$, where $x_i \in \mathbb{N}$ and $x_i \neq x_j$, onto $\mathcal{S}_n$, defined by

$$\mathrm{proj}(x) = a_1 a_2 \cdots a_n, \text{ where } a_i = |\{j \in [n].x_i \geq x_j\}|.$$

For example $\mathrm{proj}(265) = 132$.

It is easy to see that Φ is invertible and hence a bijection.

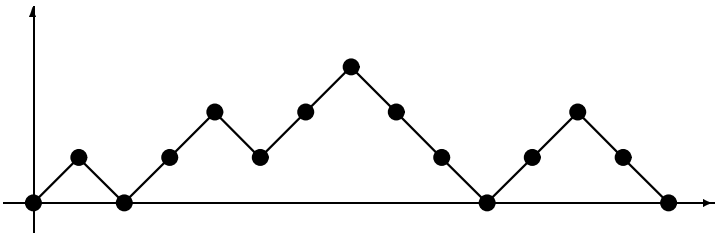


FIGURE 2. The Dyck path in Example 11

Example 11. Consider $\pi = 7312645 \in \mathcal{S}_7(2\text{-}3\text{-}1)$. The corresponding Dyck path is given by:

$$\begin{aligned}
\Phi(\pi) &= u\Phi(\epsilon)d\Phi(312645)\\
&= u\epsilon du\Phi(312)d\Phi(12)\\
&= u\epsilon duu\Phi(\epsilon)d\Phi(12)du\Phi(1)d\Phi(\epsilon)\\
&= u\epsilon duu\epsilon du\Phi(1)d\Phi(\epsilon)duu\Phi(\epsilon)d\Phi(\epsilon)d\epsilon\\
&= u\epsilon duu\epsilon duu\Phi(\epsilon)d\epsilon d\epsilon duu\epsilon d\epsilon d\epsilon\\
&= u\epsilon duu\epsilon duu\epsilon d\epsilon d\epsilon duu\epsilon d\epsilon d\epsilon\\
&= uduuduuddduudd.
\end{aligned}$$

When restricted to involutions we have that τ, and accordingly $\Phi(\tau)$, is determined by the position of n. In fact, the decreasing sequence starting with n, that is $n\tau$, where $\tau = (n-1)\cdots k$, corresponds to the Dyck path

$$\begin{aligned}
\Phi(\tau) &= u\Phi(\epsilon)d\Phi((n-1)(n-2)\cdots k)\\
&= udu\Phi(\epsilon)d\Phi((n-2)(n-3)\cdots k)\\
&\vdots\\
&= udud\cdots ud.
\end{aligned}$$

The image of $\mathcal{I}_n(2\text{-}3\text{-}1)$, denoted by D_n^*, will therefore be

$$D_n^* = \{uD_{n-k}^* dud\cdots ud\}.$$

That is, a Dyck path in D_n^* ends with a tail of the form $udud\cdots ud$, preceding which, there are no returns to the x-axis. Removing the tail, the down-step just before it and the first up-step of the Dyck path yields a path with the same properties. Note that D_n^* is the set of Dyck paths, in which a down-step is immediately followed by at most one up-step. As a consequence the peaks as well as the valleys are of decreasing height. An illustration of a typical Dyck path in D_n^* is given in Example 12 below.

From the construction, the number of D_n^* satisfies the recursion

$$|D_n^*| = \sum_{i=1}^{n-1} |D_i^*|,$$

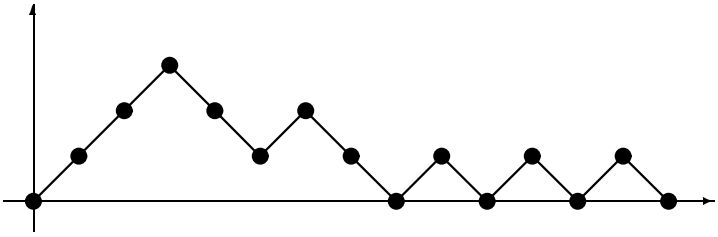


FIGURE 3. The Dyck path in Example 12

and since $D_1^* = \{ud\}$, we have $|D_1^*| = 1$, so $|D_n^*| = 2^{n-1}$. □

Example 12. Let us return to the (2-3-1)-avoiding involution $\pi = 1327654$ from Example 7. We have that π corresponds to the Dyck path:

$$\begin{aligned}
\Phi(\pi) &= u\Phi(132)d\Phi(321) \\
&= uu\Phi(1)d\Phi(1)du\Phi(\epsilon)d\Phi(21) \\
&= uuu\Phi(\epsilon)ddu\Phi(\epsilon)ddu\epsilon du\Phi(\epsilon)d\Phi(1) \\
&= uuu\epsilon ddu\epsilon ddu\epsilon du\epsilon du\Phi(\epsilon)d\Phi(\epsilon) \\
&= uuu\epsilon ddu\epsilon ddu\epsilon du\epsilon du\epsilon d\epsilon \\
&= uuuddudduddududud.
\end{aligned}$$

Proposition 13. *The number of involutions of* $[n]$ *that avoid* (3-1-2) *is* 2^{n-1}.

For the proof we need the following lemma.

Lemma 1. *Let* p *be a pattern in* $\mathcal{S}_k$. *Then* $\mathcal{I}_n(p) = \mathcal{I}_n(p^{-1})$.

Proof. Consider the involution π written in two line notation;

$$\pi = \begin{pmatrix} 1 & 2 & \dots & n \\ a_1 & a_2 & \dots & a_n \end{pmatrix}.$$

Suppose that the subword

$$v = \begin{pmatrix} i_1 & i_2 & \dots & i_k \\ a_{1_1} & a_{i_2} & \dots & a_{i_n} \end{pmatrix}$$

forms an occurrence of p. Then

$$v^{-1} = \begin{pmatrix} a_{1_1} & a_{i_2} & \dots & a_{i_n} \\ i_1 & i_2 & \dots & i_k \end{pmatrix}$$

is a p^{-1}-subword, contained in

$$\pi^{-1} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ 1 & 2 & \dots & n \end{pmatrix}.$$

But since π is an involution, we have that $\pi = \pi^{-1}$, so π contains also the p^{-1}-subword v^{-1}. Hence we have an occurrence of p^{-1} if and only if we have an occurrence of p. □

Example 14. Let p be the pattern

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}.$$

Then

$$q = p^{-1} = \begin{pmatrix} 2 & 4 & 1 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

is the inverse of p. Let π be the involution

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 3 & 2 & 9 & 1 & 6 & 8 & 7 & 4 \end{pmatrix}.$$

The letters

$$v = \begin{pmatrix} 2 & 4 & 5 & 8 \\ 3 & 9 & 1 & 7 \end{pmatrix}$$

form an occurrence of the pattern p. Accordingly,

$$v^{-1} = \begin{pmatrix} 3 & 9 & 1 & 7 \\ 2 & 4 & 5 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 3 & 7 & 9 \\ 5 & 2 & 8 & 4 \end{pmatrix}$$

forms a q-subword.

Proof of Proposition 13. We have that (3–1–2) is the inverse of (2–3–1). The result then follows immediately from Proposition 6 and Lemma 1. □

We conclude this section by an application of Proposition 6 to a certain set of pattern avoiding permutations. Claesson [1] shows that involutions of $[n]$ are in one-to-one correspondence with permutations of $[n]$ that avoid (1–23) and (1–32). For the proof he constructs a bijection Φ between $\mathcal{I}_n$ and $\mathcal{S}_n(1\text{–}23, 1\text{–}32)$, which we describe below.

The standard form of a permutation π is defined by writing π in cycle notation and requiring that

(a) each cycle is written with its least element first
(b) the cycles are written in decreasing order with respect to their first elements.

The corresponding permutation $\widehat{\pi} = \Phi(\pi)$ is obtained from π in standard form by erasing the brackets separating the cycles. Since involutions consist of cycles of length one or two, each permutation $\widehat{\pi}$ in $\mathcal{S}_n(1\text{–}23, 1\text{–}32)$ is obtained from exactly one involution, and Φ is therefore a bijection.

Corollary 15. *Involutions of* $[n]$ *that avoid* (2–3–1) *are in one-to-one correspondence with permutations in* $[n]$ *that avoid* (1–23), (1–32), (13–2) *and* (3–214). *Hence*

$$|\mathcal{S}_n(1\text{–}23, 1\text{–}32, 13\text{–}2, 3\text{–}214)| = |\mathcal{I}_n(2\text{–}3\text{–}1)| = 2^{n-1}.$$

Proof. Claesson [1] proves the one-to-one correspondence between $\mathcal{S}_n(1\text{–}23, 1\text{–}32)$ and $\mathcal{I}_n$, so what is left to prove is that, given a (2–3–1)-avoiding involution π we have that $\Phi(\pi)$ avoids (13–2) and (3–214) and vice versa.

To show that $\Phi(I_n(2-3-1)) \subseteq S_n(1-23, 1-32, 13-2, 3-214)$, assume that $\widehat{\pi}$ in $\mathcal{S}_n(1\text{–}23, 1\text{–}32)$ contains a (13–2)-subword. Then there exists a segment of $\widehat{\pi}$ of the form

$$a_1 a_3 \cdots a_2, \quad \text{where } a_1 < a_2 < a_3.$$

Since the cycles of involutions in standard form are of maximum length two and are written with their least element first, $\widehat{\pi}$ necessarily corresponds to an involution π containing the cycle (a_1, a_3). It also follows that the letter a_2 must be contained in a cycle $(\tilde{a}, a_2)$, where $\tilde{a} < a_1$, for otherwise a_2 would precede a_1 in $\widehat{\pi}$. We now have that

$$a_2 \cdots a_3 \cdots \tilde{a} \cdots a_1, \quad \text{where } \tilde{a} < a_1 < a_2 < a_3,$$

is a segment of π, so π contains the (2–3–1)-subword $a_2 a_3 a_1$.

Assume instead that $\widehat{\pi}$ has an occurrence of (3–214), that is $\widehat{\pi}$ contains the segment

$$a_3 \cdots a_2 a_1 a_4, \quad \text{where } a_1 < a_2 < a_3 < a_4.$$

Then $(a_1 a_4)$ must be a 2-cycle of π. The letters a_2 and a_3 can either be fixed points or contained in 2-cycles.

Assuming that a_2 and a_3 both are fixed points implies that π contains a segment of the form

$$a_4 \cdots a_2 \cdots a_3 \cdots a_1, \quad \text{where } a_1 < a_2 < a_3 < a_4.$$

Here $a_2 a_3 a_1$ forms a (2–3–1)-subword.

If a_3 is a fixed point while a_2 is not, then a_2 will be contained in a cycle $(\tilde{a_2}, a_2)$ where $a_1 < \tilde{a_2} < a_2$, once again resulting in the (2–3–1)-subword $a_2 a_3 a_1$ of π.

Finally we assume that a_3 is contained in a 2-cycle $(\tilde{a_3}, a_3)$, where $\tilde{a_3} > a_2$ (or $\tilde{a_3} > \tilde{a_2}$, if there is a cycle $(\tilde{a_2}, a_2)$). We then get the following possible segments of π:

$$\begin{aligned}
&a_4 \cdots a_2 \cdots \tilde{a_2} \cdots a_3 \cdots \tilde{a_3} \ldots a_1, \text{ where } \tilde{a_3} < a_3, \\
&a_4 \cdots a_2 \cdots \tilde{a_2} \cdots \tilde{a_3} \cdots a_3 \ldots a_1, \text{ where } a_3 < \tilde{a_3} < a_4, \\
&a_4 \cdots a_2 \cdots \tilde{a_2} \cdots \tilde{a_3} \cdots a_1 \ldots a_3, \text{ where } \tilde{a_3} > a_4.
\end{aligned}$$

In all cases we get an occurrence of (2–3–1). Hence it follows that

$$\Phi(I_n(2-3-1)) \subseteq S_n(1-23, 1-32, 13-2, 3-214).$$

To show the converse, that is

$$\mathcal{S}_n(1-23, 1-32, 13-2, 3-214) \subseteq \Phi(I_n(2-3-1)),$$

we consider $\pi \in I_n(2\text{-}3\text{-}1)$. There are essentially two different ways of constructing a (2-3-1)-subword out of three letters a_1, a_2 and $a_3 \in [n]$ such that $a_1 < a_2 < a_3$. Either we get an involution of the form

$$\dots(a_3b_3)\dots(a_2b_2)\dots(a_1b_1)\dots,$$

where

$$a_1 < a_2 < a_3,\ b_2 < b_3 < b_1,\ a_1 < b_1$$

or an involution of the form

$$\dots(b_2a_2)\dots(b_3a_3)\dots(a_1b_1)\dots$$

where

$$a_1 < a_2 < a_3,\ b_2 < b_3 < b_1,\ a_2 > b_2,\ a_3 > b_3.$$

Consider the first case. Without loss of generality we let $a_3 \leq b_3$ and $a_2 \leq b_2$. The special cases when a_2 and a_3 are fixed points are given by letting a_2 and a_3 be equal to b_2 and b_3 respectively. Consider the cycle $(ij) = (b_1a_1)$. Clearly $i < j$. Let $(k\ell)$ be the cycle to the left of (ij) ($k = \ell$ denotes the case when k is a fixed point). If $\ell < j = b_3$, then ℓij forms a (3-214)-subword of the corresponding permutation $\Phi(\pi)$, because ℓ is clearly larger than i. Otherwise let $(ij) = (k\ell)$ and repeat the above arguments until a (3-214)-subword is obtained. This is guaranteed to happen, since if we have gone through all cycles between (b_2a_2) and (b_1a_1), then with b_2 as ℓ we have that $\ell = b_2 < b_3$.

Considering the second case, without loss of generality we let $a_1 \leq b$. The case when a_1 is a fixed point is denoted by $a_1 = b_1$. The subword $(b_2a_2a_1)$ will now form an occurrence of (13-2) since $b_2 < a_1 < a_2$.

This proves that

$$S_n(1-23, 1-32, 13-2, 3-214) \subseteq \Phi(I_n(2-3-1)).$$

Hence

$$|S_n(1-23, 1-32, 13-2, 3-214)| = |(I_n(2-3-1)|.$$

□

3.2. **Avoiding (2-1-3) or (1-3-2).** We introduce a couple of results that will be used in the proof of Proposition 18. First we present a well-known property of the patterns in $\mathcal{S}_3$.

Proposition 16. *Let p be a pattern in $\mathcal{S}_3$. Then $|\mathcal{S}_3(p)| = C_n$, where $C_n = \frac{1}{n+1}\binom{2n}{n}$ is the nth Catalan number.*

One way of proving Proposition 16 is to construct a bijection between the pattern avoiding permutations of $[n]$ and the set of Dyck paths of length $2n$, that are known to be counted by the nth Catalan number. Such a bijection for the case when $p = (2\text{-}3\text{-}1)$ is actually presented in the fourth proof of Proposition 6 on page 9.

Next we consider a consequence of the fact that an involution is its own inverse.

Lemma 2. *Let p be an involution of $[k]$ and π a permutation of $[n]$. Then π avoids the pattern p if and only if π^{-1} avoids p.*

Proof. Consider π written in two line notation:

$$\pi = \begin{pmatrix} 1 & 2 & \dots & n \\ a_1 & a_2 & \dots & a_n \end{pmatrix}.$$

Suppose that we have an occurrence of p as the subword

$$v = \begin{pmatrix} i_1 & i_2 & \dots & i_k \\ a_{1_1} & a_{i_2} & \dots & a_{i_n} \end{pmatrix}.$$

Since p is an involution, we have that $p^{-1} = p$ and

$$v^{-1} = \begin{pmatrix} a_{1_1} & a_{i_2} & \dots & a_{i_n} \\ i_1 & i_2 & \dots & i_k \end{pmatrix}$$

forms a p-subword contained in

$$\pi^{-1} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ 1 & 2 & \dots & n \end{pmatrix}.$$

Hence π avoids p if and only if π^{-1} avoids p. □

Example 17. Let p be the 5-pattern

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 1 & 4 & 2 \end{pmatrix}.$$

Clearly p is an involution. Now consider the permutation

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 4 & 9 & 1 & 2 & 6 & 7 & 3 & 8 \end{pmatrix}.$$

The subword

$$v = \begin{pmatrix} 1 & 3 & 5 & 7 & 8 \\ 5 & 9 & 2 & 7 & 3 \end{pmatrix}$$

forms an occurrence of p and accordingly

$$\pi^{-1} = \begin{pmatrix} 5 & 4 & 9 & 1 & 2 & 6 & 7 & 3 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 5 & 8 & 2 & 1 & 6 & 7 & 9 & 3 \end{pmatrix},$$

contains the p-subword

$$v = \begin{pmatrix} 5 & 9 & 2 & 7 & 3 \\ 1 & 3 & 5 & 7 & 8 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 5 & 7 & 9 \\ 5 & 8 & 1 & 7 & 3 \end{pmatrix}.$$

Proposition 18. *The number of involutions of $[n]$ that avoid (2–1–3) is the nth central binomial coefficient $\binom{n}{n/2}$.*

Proof. First we give a general description of the elements in $\mathcal{I}_n(2\text{–}1\text{–}3)$. If $\pi = a_1a_2\cdots a_n$ is a permutation of $[n]$ with the letter 1 in position k, then π avoids (2–1–3) if and only if it can be written as $\pi = \sigma 1 \tau$, where $\sigma = a_1a_2\cdots a_{k-1}$ is a (2–1–3)-avoiding permutation of $\{n, (n-1), \ldots, (n-k+2)\}$ and $\tau = a_{k+1}a_{k+2}\cdots a_n$ is a (2–1–3)-avoiding permutation of $\{2, 3, \ldots, (n-k+1)\}$. That is, the letters preceding 1 must all be larger than the ones following 1, and clearly all segments of π must be (2–1–3)-avoiding.

When constructing a (2–1–3)-avoiding involution, π, there are essentially two different ways of positioning the letter 1. Either it can be placed as the first letter a_1, in which case $\sigma = \epsilon$, the empty word, or it can be placed in the second half of the word, that is in position k where $k \geq \frac{n}{2}+1$. Namely, σ, if nonempty, consists of the $(k-1)$ largest letters of $[n]$, in particular k, that is the first letter of π, because 1 is the kth letter, must be one of the $(k-1)$ largest letters, so $k \geq \frac{n}{2}+1$.

Let us now consider the permutation τ. In the first case, when 1 is a fixed point, τ is merely a (2–1–3)-avoiding involution of $\{2, 3, \ldots, n\}$. In the second case though, the letters following 1, in positions larger than k, will all be smaller than k, so an arbitrary permutation of $\{2, 3, \ldots, (n-k+1)\}$ will do as τ as long as it avoids (2–1–3). We notice that the first $(n-k+1)$ letters of π are uniquely determined by τ since the letters of τ must all be contained in 2-cycles (i, a_i), where $i \leq (n-k+1)$. Hence $\pi = a_1a_2\cdots a_n$ can be written as

$$\pi = k\tau^{-1}\rho 1 \tau,$$

where τ^{-1} is the inverse of τ seen as a bijection from $\{2, 3, \ldots, (n-k+1)\}$ to $\{k, (k+1), \ldots n\}$ and where $\rho = a_{n-k+2}a_{n-k+3}\cdots a_{k-1}$. To make sure that π is (2–1–3)-avoiding we must check that τ^{-1} avoids (2–1–3) whenever τ does, but this is exactly what is said in Lemma 2. Finally ρ, must be a (2–1–3)-avoiding involution of $\{(n-k+2), (n-k+1), \ldots, (k-1)\}$, on which we recursively repeat the arguments above.

The next step of the proof is to derive an expression for the number of (2–1–3)-avoiding involutions from the above description of them. Let, for the sake of simplicity, $|\mathcal{I}_n(2\text{–}1\text{–}3)|$ be denoted by A_n. With 1 in position k, where $k \geq \frac{n}{2}+1$, the number of possible τ's is the $(n-k)$th Catalan number C_{n-k}, according to Proposition 16. Independently of τ there are A_{2k-n-2} ways of choosing ρ, so the number of (2–1–3)-avoiding involutions with $a_k = 1$ is $A_{2k-n-2}C_{n-k}$. Moreover, there are A_{n-1} possible (2–1–3)-avoiding involutions with 1 as a fixed point. Thus

$$A_n = A_{n-1} + \sum_{k=\lfloor \frac{n}{2} \rfloor + 1}^{n} A_{2k-n-2}C_{n-k}$$

and $A_n = 0$ if $n \leq 0$. This recursion is satisfied by the central binomial coefficients [11], thus we conclude that $|\mathcal{I}_n(2\text{–}1\text{–}3)| = A_n = \binom{n}{n/2}$. □

We now turn to avoidance of (1-3-2). For this purpose we introduce the trivial bijections on permutations.

3.2.1. *Trivial bijections.* Let $\pi = a_1a_2\cdots a_n \in \mathcal{S}_n$. We define the *reverse* of π as $R(\pi) := a_n\cdots a_2a_1$, and the *complement* of π by $C(\pi)(i) = n+1-\pi(i)$, where $i \in [n]$. These bijections from $\mathcal{S}_n$ to itself and their composition $C\circ R$ are called *trivial*. Let Φ be a trivial bijection and let π be in $\mathcal{S}_n(p)$. Then the permutation $\Phi(\pi)$ avoids the pattern $\Phi(p)$ and consequently the number of permutations avoiding $R(p)$, $C(p)$ or $R\circ P(p)$ is the same as the number of permutations avoiding the pattern p. Note that the reverse of the generalised pattern $(a_1\text{-}a_2a_3\text{-}a_4a_5)$ is $(a_5a_4\text{-}a_3a_2\text{-}a_1)$. Also the dashes are "reversed".

Example 19. Let $p = 534621$. It is clear that p avoids (1-3-2). The reverse of π, $R(\pi) = 125435$, the complement of π, $C(\pi) = 243156$ and their composition, $R\circ C(\pi) = 651342$ then avoid $R(p) = (2\text{-}3\text{-}1)$, $C(p) = (3\text{-}1\text{-}2)$ and $R\circ C(p) = (2\text{-}1\text{-}3)$, respectively.

Lemma 3. *The composition $C\circ R$, restricted to $\mathcal{I}_n$, is a bijection from $\mathcal{I}_n$ to itself.*

Proof. Let π be in $\mathcal{I}_n$. Then π consists of cycles of length 1 and 2, that is, $\pi(j) = k$ whenever $\pi(k) = j$. The case when j is a fixed point is denoted by $k = j$. Let (j,k) be a cycle of π, then

$$\begin{aligned} R(\pi)(n+1-j) &= \pi(n+1-(n+1-j)) = \pi(j) = k, \\ C\circ R(\pi)(n+1-j) &= n+1-R(\pi)(n+1-j) = n+1-k. \end{aligned}$$

Likewise $C\circ R(\pi)(n+1-k) = n+1-j$ which shows that $(n+1-j,\ n+1-k)$ is a 2-cycle of $C\circ R(\pi)$. Hence $C\circ R(\pi)$ is an involution, so $C\circ R(\mathcal{I}_n) = \mathcal{I}_n$. □

Proposition 20. *The number of involutions of $[n]$ that avoid (1-3-2) is the nth central binomial coefficient $\binom{n}{n/2}$.*

Proof. Let π be in $\mathcal{I}_n(2\text{-}1\text{-}3)$. The permutation $C\circ R(\pi)$ is in $\mathcal{I}_n$ by Lemma 3 and it is clear from above that $C\circ R(\pi)$ avoids $C\circ R(2\text{-}1\text{-}3) = (1\text{-}3\text{-}2)$. Since $C\circ R$ is a bijection from $\mathcal{I}_n$ to $\mathcal{I}_n$ it follows that

$$C\circ R : \mathcal{I}_n(2\text{-}1\text{-}3) \to \mathcal{I}_n(1\text{-}3\text{-}2)$$

is injective, thus $|\mathcal{I}_n(2\text{-}1\text{-}3)| \le |\mathcal{I}_n(1\text{-}3\text{-}2)|$. In order to show the converse, note that $C\circ R$ is its own inverse and hence $C\circ R(C\circ R(p)) = p$. An application of the same argument to $C\circ R(2\text{-}1\text{-}3) = (1\text{-}3\text{-}2)$ implies the desired inequality $|\mathcal{I}_n(1\text{-}3\text{-}2)| \le |\mathcal{I}_n(2\text{-}1\text{-}3)|$. Thus, it follows that $|\mathcal{I}_n(2\text{-}1\text{-}3)| = |\mathcal{I}_n(1\text{-}3\text{-}2)|$. □

3.3. **Avoiding p, when p is an increasing or decreasing sequence.** This section concerns avoidance of to the two remaining 3-patterns, (1–2–3) and (3–2–1). Although we have not found any direct relation between $\mathcal{I}_n(1\text{–}2\text{–}3)$ and $\mathcal{I}_n(3\text{–}2\text{–}1)$, it is possible to give almost analogous proofs for them being counted by $\binom{n}{n/2}$ by using the RSK algorithm for Young Tableaux, as will be seen below. We start however with a combinatorial proof for (3–2–1)-avoidance, based on work by Kitaev and Claesson.

In Kitaev [5], which concerns multiavoidance of 3-patterns without internal dashes, it is shown that the permutations of $[n]$ that simultaneously avoid (123), (132) and (213) are counted by the central binomial coefficients. We will use this result to conclude that the number of (3–2–1)-avoiding involutions of $[n]$ is $\binom{n}{n/2}$. Thus we have to establish a relation between $\mathcal{I}_n(3\text{–}2\text{–}1)$ and $\mathcal{S}_n(123, 132, 213)$.

Lemma 4. *Involutions of* $[n]$ *that avoid* (3–2–1) *are in one-to-one correspondence with permutations of* $[n]$ *that avoid* (123), (132) *and* (213). *Hence*

$$|\mathcal{I}_n(3\text{–}2\text{–}1)| = |\mathcal{S}_n(123, 132, 213)|.$$

Proof. Claesson [1] gives a proof that there is a one-to-one correspondence between $\mathcal{I}_n$ and $\mathcal{S}_n(1\text{–}23, 1\text{–}32)$ by constructing the bijection Φ, which is described in connection to Corollary 15, on page 12. Furthermore, he observes that the dashes in the patterns are immaterial for the proof and accordingly $\mathcal{S}_n(123, 132) = \mathcal{S}_n(1\text{–}23, 1\text{–}32)$. We show that Φ restricted to the (3–2–1)-avoiding involutions gives exactly the permutations that avoid (123), (132) and (213).

To show that $\mathcal{S}_n(123, 132, 213) \subseteq \Phi(\mathcal{I}_n(3\text{–}2\text{–}1))$, let π be an involution of $[n]$ and let $\widehat{\pi}$ be the corresponding permutation in $\mathcal{S}_n(123, 132)$. Assume that $\widehat{\pi}$ contains a (213)-subword. There then exists a segment of $\widehat{\pi}$ of the form

$$a_2 a_1 a_3, \text{ where } a_1 < a_2 < a_3.$$

Since the cycles in the standard form are of maximum length two and are written in decreasing order with their least elements first, the only possibility for a_3 to follow a_1 is that (a_1, a_3) is a cycle of π. The letter a_2 is either a fixed point or contained in the 2-cycle $(\widetilde{a_2}, a_2)$, where $a_1 < \widetilde{a_2} < a_2$. Thus π contains either the segment

$$a_3 \cdots a_2 \cdots a_1, \text{ where } a_1 < a_2 < a_3$$

or

$$a_3 \cdots \tilde{a_2} \cdots a_2 \cdots a_1, \text{ where } a_1 < \tilde{a_2} < a_2 < a_3,$$

where $a_3 a_2 a_1$ forms a (3–2–1)-subword in both cases.

In order to show that $\Phi(\mathcal{I}_n(3\text{–}2\text{–}1)) \subseteq \mathcal{S}_n(123, 132, 213)$ we assume that there is an occurrence of (3–2–1) in π, that is, π contains a segment

$$a_3 \cdots a_2 \cdots a_1, \text{ where } a_1 < a_2 < a_3.$$

There are essentially three different ways of constructing this out of a_1, a_2 and a_3.

First, we consider the case when π, written in cycle notation, is of the form

$$\cdots(a_1b_1)\cdots(b_2a_2)\cdots(b_3a_3)\cdots,$$

where

$$a_1 < a_2 < a_3 \text{ and } b_3 < b_2 < b_1.$$

Let $a_1 = b_1$ denote the case when a_1 is a fixed point. Consider the cycle $(ij) = (b_3a_3)$. Clearly $i < j$. Let $(k\ell)$ be the cycle to the left of (ij) ($k = \ell$ denotes the case when k is a fixed point). If $\ell < j = a_3$, then ℓij forms a (213)-subword of the corresponding permutation $\Phi(\pi)$. Otherwise let $(ij) = (k\ell)$ and repeat the above reasoning. We realize that this procedure will cause a (213)-subword to be formed as ℓij. Indeed, if we have gone through all cycles between a_2 and b_3, then with a_2 as ℓ it will be true that $i < \ell < j$, because ℓ is smaller than j ($j \geq a_3 > a_2 = \ell$) and since the cycles are written in decreasing order it follows that ℓ is larger than i.

The next possibility is that π is of the form

$$\cdots(a_1b_1)\cdots(a_2b_2)\cdots(a_3b_3)\cdots,$$

where

$$a_1 < a_2 < a_3 \text{ and } b_3 < b_2 < b_1.$$

Let $a_3 = b_3$ denote the special case when a_3 is a fixed point. By setting $(ij) = (a_2b_2)$, letting $(k\ell)$ be the cycle to the left of (ij) and repeating the arguments from the first case we get an occurrence of (213) in the corresponding permutation $\widehat{\pi}$. Indeed, the fact that b_1 is smaller than b_2 and consequently smaller than every j and also clearly larger than i guarantees that ℓij will form a (213)-subword for some ℓ, i and j.

Finally we consider π, when π is of the form

$$\cdots(a_1b_1)\cdots(a_2b_2)\cdots(b_3a_3)\cdots,$$

where

$$a_1 < a_2 < a_3 \text{ and } b_3 < b_2 < b_1.$$

The special case when a_2 is a fixed point is denoted by $a_2 = b_2$. A (213)-subword is obtained by letting $(ij) = (b_3a_3)$ and once again applying the above arguments.

This proves that $\mathcal{S}_n(123, 132, 213) = \mathcal{I}_n(2\text{–}3\text{–}1)$. □

We are now prepared to conclude the following result.

Proposition 21. *The number of involutions of $[n]$ that avoid $\mathcal{I}_n(3-2-1)$ is the nth central binomial coefficient $\binom{n}{n/2}$.*

Proof. This follows immediately from Lemma 4 and the fact that $|\mathcal{S}_n(123, 132, 213)| = \binom{n}{n/2}$, shown by Kitaev in [5]. □

3.3.1. *Young tableaux and involutions.* Knuth [8] proves that the number of involutions of $[n]$ is the same as the number of Young tableaux that can be formed from $[n]$. In his proof he constructs a Young tableau from an involution by inserting the letters of the involution into an originally empty Young tableau, using an algorithm I. Together with its inverse D, for deleting elements from a tableau, I is called the *RSK algorithm*, after its creators; Robinson, Schenstedt and Knuth.

Given a Young tableau P and an integer x that is not in P, algorithm I creates a new tableau P' that contains x in addition to its original elements. The tableau P' has the same shape as P except for a new entry added to one of the rows. When inserting the element x into P, it is first compared to the elements in the first row of P. If x is larger than all elements in the first row it is placed as the last element in that row and the algorithm terminates, otherwise it is placed in the position of the smallest element larger than x. This element x' is then inserted into the next row in the same way. The procedure is repeated until an element x' is inserted as the last element of a row.

Example 22. We illustrate the insertion algorithm I by an example. Suppose that we want to insert 4 into the Young tableau P, where

$$P = \begin{array}{|c|c|c|c|}\hline 1 & 3 & 6 & 7 \\ \hline 2 & 9 \\ \cline{1-2} 5 \\ \cline{1-1} 8 \\ \cline{1-1}\end{array} \quad .$$

First, the 4 will be placed in the entry occupied by 6, since 6 is the smallest element larger than 4 in the first row.

$$\begin{array}{|c|c|c|c|}\hline 1 & 3 & 4 & 7 \\ \hline 2 & 9 \\ \cline{1-2} 5 \\ \cline{1-1} 8 \\ \cline{1-1}\end{array}$$

Element 6 is then moved down to the second row where it displaces 9.

$$\begin{array}{|c|c|c|c|}\hline 1 & 3 & 4 & 7 \\ \hline 2 & 6 \\ \cline{1-2} 5 \\ \cline{1-1} 8 \\ \cline{1-1}\end{array}$$

Finally 9 will be placed as the last element in the third row, since the row contains no element larger than 9, and the procedure terminates. The tableau P has now been transformed into P', where

$$P' = \begin{array}{|c|c|c|c|}\hline 1 & 3 & 4 & 7 \\ \hline 2 & 6 \\ \cline{1-2} 5 & 9 \\ \cline{1-2} 8 \\ \cline{1-1}\end{array} \quad .$$

Note that P' has the same shape as P except for the new square, containing 9.

With P' and the position of the entry added when inserting x, it is possible to get back to P by running algorithm I backwards. More generally, given a Young tableau Q and indices (s,t) such that $y = Q_{st}$ is the rightmost element in row s and that column t has no entries below y, algorithm D transforms Q into a Young tableau Q' with no element in position (s,t) but otherwise of the same shape as Q. An element x is then deleted from Q. The method starts by removing the element y from row s and inserting it into row $s-1$ where it displaces the largest element smaller than y. This element y' is in turn moved up to row $s-2$. This procedure continues until an element is removed from the first row. If we apply algorithm D to the tableau P' and the indices of the entry that makes the difference in shape between P' and P, we end up with the original tableau P and the element x. Likewise, if we start with a Young tableau Q and indices (s,t) and apply algorithm D we get a tableau Q' and an element z. Inserting z into Q' according to I will get us back to Q. In this sense the algorithms I and D are inverses of each other.

Example 23. We want to transform the Young tableau P' from example 22 back to its original form P. The entry that makes the difference between the shape of P' and that of P has the indices $(3,2)$, so we start by removing the element in this position, that is 9. The element 9 is inserted into the second row in the position of 6, since 6 is the largest element smaller than 9. Finally 6 replaces element 4 in the first row and we get back to P, with 4 as the deleted element.

$$P' = \begin{array}{|c|c|c|c|}\hline 1&3&4&7\\\hline 2&6\\\cline{1-2} 5&9\\\cline{1-2} 8\\\cline{1-1}\end{array} \quad \rightarrow \quad \begin{array}{|c|c|c|c|}\hline 1&3&4&7\\\hline 2&9\\\cline{1-2} 5\\\cline{1-1} 8\\\cline{1-1}\end{array} \quad \rightarrow \quad \begin{array}{|c|c|c|c|}\hline 1&3&6&7\\\hline 2&9\\\cline{1-2} 5\\\cline{1-1} 8\\\cline{1-1}\end{array} \quad = \; P \; .$$

By considering a permutation written in two line notation, Knuth constructs a mapping from $\mathcal{S}_n$ to the set of ordered pairs of Young tableaux (P,Q) formed from the elements $\{1,2,\ldots,n\}$, where P and Q have the same shape. This is done by inserting the elements one by one into an initially empty Young tableau, partly by using algorithm I. This mapping is shown to be invertible, so there is a one-to-one correspondence between $\mathcal{S}_n$ and the set of ordered pairs (P,Q), where P and Q are as above.

Next, Knuth shows that if the permutation

$$\pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ a_1 & a_2 & \ldots & a_n \end{pmatrix}$$

corresponds to the ordered pair of tableaux (P, Q), then the inverse permutation

$$\pi^{-1} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ 1 & 2 & \dots & n \end{pmatrix}$$

corresponds to (Q, P). Hence, since the involutions are the permutations that are their own inverses they correspond to pairs of tableaux (P, P), and therefore the number of tableaux that can be formed from $[n]$ equals the number of involutions of length n. For a detailed proof we refer to [8].

A consequence of the tableau-constructing method based on algorithm I is that the number of rows in the resulting Young tableau P corresponds to the length of the longest decreasing sequence of the permutation. Indeed, for the algorithm not to terminate before the kth row, the element inserted into row i, where $i \leq k$, has to be smaller than the largest element of the row. That is, an element x that causes a movement down to the kth row must have been preceded by a smaller element in the involution (now in the first row), that in turn must have been preceded by an even smaller element (in the second row) et cetera, that is the involution must contain a decreasing sequence of length k. On the other hand, if we let $a_{i_k} \dots a_{i_1}$ denote the lexicographically smallest decreasing sequence of length k, it is easy to realize that when a_{i_1} has been inserted into the first row, element a_{i_j} will be in row j for each j. Hence the Young tableau will have k rows exactly when the longest decreasing sequence is of length k. In particular $\mathcal{I}_n(3\text{–}2\text{–}1)$ will be in one-to-one correspondence with the Young tableaux with at most two rows. It is known that the number of Young tableaux with two or less rows is the nth central binomial coefficient. For a proof see for example Lundin [9]. This therefore gives another proof of Proposition 21.

As the length of the longest decreasing sequence of the involution determines the number of rows, the length of the longest increasing sequence equals the number of columns. This can be seen from the construction by arguments similar to those above. The set of (1–2–3)-avoiding involutions will therefore be in one-to-one correspondence with the Young tableaux with two or less columns. Taking the transpose of a Young tableau; $P_{ij} \mapsto P_{ji}$, that is reflecting in the NW-SE diagonal, clearly gives a bijection from the tableaux with k rows to the tableaux with k columns. Thus the number of Young tableaux with at most two columns is indeed the nth central binomial coefficient. This proves the following proposition.

Proposition 24. *The number of involutions avoiding (*1–2–3*) is the nth central binomial coefficient* $\binom{n}{n/2}$.

4. Involutions avoiding generalised 3-patterns

So far our work has concerned avoidance of classical patterns. In this section we extend the study to include all generalised 3-patterns.

We start our investigation by counting the pattern-avoiding involutions of $[n]$, when n is small ($n \leq 10$). The results are presented in Table 2.

p	$\lvert\mathcal{I}_n(p)\rvert$	p	$\lvert\mathcal{I}_n(p)\rvert$	p	$\lvert\mathcal{I}_n(p)\rvert$	p	$\lvert\mathcal{I}_n(p)\rvert$
(1-2-3)	$\binom{n}{n/2}$	(1-23)	A_n	(12-3)	A_n	(123)	B_n
(1-3-2)	$\binom{n}{n/2}$	(1-32)	A_n	(13-2)	$\binom{n}{n/2}$	(132)	C_n
(2-1-3)	$\binom{n}{n/2}$	(2-13)	$\binom{n}{n/2}$	(21-3)	A_n	(213)	C_n
(2-3-1)	2^{n-1}	(2-31)	2^{n-1}	(23-1)	2^{n-1}	(231)	D_n
(3-1-2)	2^{n-1}	(3-12)	2^{n-1}	(31-2)	2^{n-1}	(312)	D_n
(3-2-1)	$\binom{n}{n/2}$	(3-21)	E_n	(32-1)	E_n	(321)	B_n.

TABLE 2. Generalised patterns

Here:

$$\begin{aligned} A_n &= 1, 2, 3, 6, 11, 23, 46, 100, 213, 481, \ldots \\ B_n &= 1, 2, 3, 7, 15, 38, 97, 271, 778, 2371, \ldots \\ C_n &= 1, 2, 3, 6, 12, 28, 66, 172, 458, 1305, \ldots \\ D_n &= 1, 2, 4, 8, 17, 39, 94, 241, 646, 1821, \ldots \\ E_n &= 1, 2, 3, 6, 11, 23, 47, 103, 225, 513, \ldots \end{aligned}$$

Further we consult the On-Line Encyclopedia of Integer Sequences [11] for information about the obtained sequences of $|\mathcal{I}_n(p)|$. However, except for the well-known $\binom{n}{n/2}$ and 2^{n-1}, none of them can be found in [11]. Still the enumeration of $A_n, \ldots, E_n$ is of some interest for comparison reasons. For each row in the table there is a hierarchy amongst the patterns. Namely, an occurrence of a one-dash pattern, $(x\text{-}yz)$ or $(xy\text{-}z)$, is a special case of an occurrence of the classical two-dash pattern $(x\text{-}y\text{-}z)$, and an occurrence of the zero-dash pattern (xyz) implies an occurrence of the one-dash patterns. This hierarchy induces a partial ordering of $\mathcal{I}_n(p)$ with respect to inclusion. Accordingly

$$\begin{aligned} \mathcal{I}_n(x\text{-}y\text{-}z) &\subseteq \mathcal{I}_n(x\text{-}yz) \subseteq \mathcal{I}_n(xyz), \\ \mathcal{I}_n(x\text{-}y\text{-}z) &\subseteq \mathcal{I}_n(xy\text{-}z) \subseteq \mathcal{I}_n(xyz), \end{aligned}$$

which implies that

$$\begin{aligned} |\mathcal{I}_n(x\text{-}y\text{-}z)| &\leq |\mathcal{I}_n(x\text{-}yz)| \leq |\mathcal{I}_n(xyz)|, \\ |\mathcal{I}_n(x\text{-}y\text{-}z)| &\leq |\mathcal{I}_n(xy\text{-}z)| \leq |\mathcal{I}_n(xyz)|. \end{aligned}$$

Taking a look at the fourth row of Table 2 above, a consequence of $|\mathcal{I}_n(2\text{-}3\text{-}1)| = |\mathcal{I}_n(2\text{-}31)| = |\mathcal{I}_n(23\text{-}1)|$ is seen to be that $\mathcal{I}_n(2\text{-}3\text{-}1) =$

$\mathcal{I}_n(2\text{–}31) = \mathcal{I}_n(23\text{–}1)$, that is an involution avoids (2–3–1) if and only if it avoids (2–31), which is in turn avoided if and only if (23–1) is avoided. However, for $n \geq 5$, the sequence D_n indicates the existence of involutions that avoid (231) even though they may contain (2–3–1)-subwords. This is in fact the case for $\pi = 52431$.

Proposition 25. *The number of involutions that avoid p, when p is equal to* (2–13) *or* (13–2)*, is* $\binom{n}{n/2}$*. Hence*

$$|\mathcal{I}_n(2\text{–}13)| = |\mathcal{I}_n(13\text{–}2)| = \binom{n}{n/2}.$$

For the proof we use a consequence of the proof of Proposition 20.

Porism 26. *[of Proposition 20] For a generalised pattern p we have that $|\mathcal{I}_n(p)| = |\mathcal{I}_n(C \circ R(p))|$.*

Proof. Without loss of generality, the pattern $p = (2\text{–}1\text{–}3)$ in the proof of Proposition 20 could be replaced by any generalised pattern. □

Proof of Proposition 25. In Claesson [1] it is shown that a permutation π avoids (2–13) if and only if it avoids (2–1–3). In particular this is true when π is an involution. Thus, recalling from Proposition 6 that the (2–1–3)-avoiding involutions are counted by $\binom{n}{n/2}$, we obtain the desired result in the first case. An application of Porism 26 to $p = (13\text{–}2) = C \circ R(2\text{–}13)$ then proves the remaining part. □

Proposition 27. *An involution avoids p, where p is one of the patterns* (2–31)*,* (31–2)*,* (23–1) *or* (3–12)*, if and only if it avoids* (2–3–1)*. Hence*

$$|\mathcal{I}_n(2\text{–}31)| = |\mathcal{I}_n(31\text{–}2)| = |\mathcal{I}_n(23\text{–}1)| = |\mathcal{I}_n(3\text{–}12)| = 2^{n-1}.$$

Proof. Claesson[1] partitions the twelve one dash patterns into three equidistributed classes, with respect to the patterns considered as permutation statistics. This is done on the basis of their behaviour under actions of the trivial bijections.

As mentioned in the proof of Proposition 25, Claesson [1] shows that a permutation avoids (2–13) if and only if it avoids (2–1–3). Due to the properties of the trivial bijections, the corresponding results are true for all patterns in the (2–13) class, that is (2–31), (13–2) and (31–2). In particular, an involution avoids (2–31) if and only if it avoids (2–3–1) and avoidance of (31–2) is equivalent to avoidance of (3–1–2), which in turn, by Lemma 1, is equal to (2–3–1)-avoidance.

Concerning the two remaining patterns we give a proof by describing the pattern-avoiding involutions. Let $\pi = a_1 a_2 \cdots a_n$ be a (23–1)-avoiding involution with k as the first letter. The initial segment $\sigma = a_1 a_2 \cdots a_k$ of π is easily seen to be determined by k. Indeed, the letter 1 must clearly be in position k and since no ascents are allowed to precede 1, the only possibility is to let σ consist of the k smallest letters in decreasing

order. In the same way, the $(k+1)$st letter fixes the subsequent segment, and so forth. This procedure results in an involution of the form that was used to describe $\mathcal{I}_n(2\text{-}3\text{-}1)$ in the proof of Proposition 6. Hence $\mathcal{I}_n(23\text{-}1) \subseteq \mathcal{I}_n(2\text{-}3\text{-}1)$ and since the converse inclusion obviously holds we conclude that $\mathcal{I}_n(23\text{-}1) = \mathcal{I}_n(2\text{-}3\text{-}1)$.

By similar arguments the description of $\mathcal{I}_n(2\text{-}3\text{-}1)$ is easily seen to fit also a $(3\text{-}21)$-avoiding involution, so $\mathcal{I}_n(23\text{-}1) = \mathcal{I}_n(2\text{-}3\text{-}1)$. The details are left to the reader.

We recall from Proposition 6 that the $(2\text{-}3\text{-}1)$-avoiding involutions are counted by 2^{n-1}. Thus the second part of the proposition follows accordingly. □

5. Multiavoidance of 3-patterns among involutions

We devote this final section to the case of multiavoidance, that is when two or more patterns are simultaneously avoided. This was first systematically studied for classical 3-patterns by Simion and Schmidt [10] but has recently been extended to generalised patterns, for instance by Claesson [1], Kitaev [5], [6] and Claesson and Mansour [2].

Consider $\mathcal{I}_n(p_1, \dots p_k)$, where p_i are 3-patterns. Allowing the patterns p_i to be generalised and the number of them, k, to vary, provides us with a huge amount of different restrictions to investigate, even though many of them are not of much interest. Here we limit ourselves to the case of two classical 3-patterns, denoted p and q.

As in the study of generalised patterns we start by counting the involutions of $[n]$ that avoid the pair of patterns p and q, when n is small. The result is presented in Table 3, where a certain cell represents the number of involutions that avoid simultaneously the row and the column pattern.

$p\backslash q$	(1-2-3)	(1-3-2)	(2-1-3)	(2-3-1)	(3-1-2)	(3-2-1)
(1-2-3)		A_n	A_n	n	n	B_n
(1-3-2)			A_n	n	n	C_n
(2-1-3)				n	n	C_n
(2-3-1)					2^{n-1}	D_{n+1}
(3-1-2)						D_{n+1}
(3-2-1)						

Table 3. Double avoidance of classical patterns

Here:

$$\begin{aligned} A_n &= 1,2,2,4,4,8,8,16,16,\dots \\ B_n &= 1,2,2,2,0,0,0,0,0,\dots \\ C_n &= 1,2,2,3,3,4,4,\dots \\ D_n &= 1,1,2,3,5,8,13,21,\dots \end{aligned}$$

Note the simplicity of the sequences above, compared to those treated earlier in this work. Also note that the sequence D_n is the well known *Fibonacci numbers*.

First we consider the "simplest" sequence $B_n = 1,2,2,2,0,0,0,\dots$, which counts the involutions that avoid (1-2-3) and (3-2-1). By studying the involutions of length at most 4 it is easy to verify the first 4 B_n's. To realize that an involution of length larger than 4 must have a decreasing or an increasing subsequence of length 3, we recall from the theory behind the proofs of Proposition 21 and 24, that the RSK algorithm gives a bijection between the Young tableaux with n elements and the set of involutions of $[n]$, where the number of rows and columns of the Young tableau equal the length of the longest increasing and decreasing subsequence, respectively. It is easy to see that a Young tableau with $r \cdot c + 1$ elements must have a row containing $r+1$ elements or a column with $c+1$ elements and we conclude that all Young tableaux with $5 = 2 \cdot 2 + 1$ or more elements must have a row or a column with at least 3 elements.

An apparently different approach is to use one of the famous results in combinatorics, proved by Erdős and Szekeres in 1935.

Theorem. *(Erdős-Szekerez) Let $A = (a_1, \dots, a_n)$ be a sequence of n different real numbers. If $n \geq sr + 1$ then either A has an increasing subsequence of $s+1$ terms or a decreasing subsequence of $r+1$ terms (or both).*

For a proof, see for example [4]. From the theorem it follows immediately that an involution of $[n]$, where $n \geq 5$, must have a decreasing or increasing sequence of length 3. However, to conclude this we use the same argument as above, namely that $5 = 2 \cdot 2 + 1$. In fact, what we implicitly do above is to prove the Erdős-Szekeres Theorem in the case of integer a_i, via the RSK-algorithm.

Let us continue with the case when one of the avoided patterns is (2-3-1) or (3-1-2). As pointed out in Lemma 1, we have that $\mathcal{I}_n(2\text{-}3\text{-}1) = \mathcal{I}_n(3\text{-}1\text{-}2)$, hence $\mathcal{I}_n(p, 2\text{-}3\text{-}1) = \mathcal{I}_n(p, 3\text{-}1\text{-}2)$, and consequently it suffices to consider either of those sets. Also we conclude the obvious result that $|\mathcal{I}_n(2\text{-}3\text{-}1, 3\text{-}1\text{-}2)| = |\mathcal{I}_n(2\text{-}3\text{-}1)| = |\mathcal{I}_n(3\text{-}1\text{-}2)| = 2^{n-1}$.

Proposition 28. *We have that*

$$\begin{aligned}|\mathcal{I}_n(1\text{-}2\text{-}3, 2\text{-}3\text{-}1)| &= |\mathcal{I}_n(1\text{-}2\text{-}3, 3\text{-}1\text{-}2)| = \\ |\mathcal{I}_n(1\text{-}3\text{-}2, 2\text{-}3\text{-}1)| &= |\mathcal{I}_n(1\text{-}3\text{-}2, 3\text{-}1\text{-}2)| = \\ |\mathcal{I}_n(2\text{-}1\text{-}3, 2\text{-}3\text{-}1)| &= |\mathcal{I}_n(2\text{-}1\text{-}3, 3\text{-}1\text{-}2)| = n.\end{aligned}$$

Proof. We recall the description of $\mathcal{I}_n(2\text{-}3\text{-}1) = \mathcal{I}_n(1\text{-}3\text{-}2)$ from the proof of Proposition 6;

$$\mathcal{I}_n(2\text{-}3\text{-}1) = \{k_1 \cdots 1 k_2 \cdots (k_1+1) k_3 \cdots (k_{\ell-1}+1) n \cdots (k_\ell+1)\}.$$

That is the involutions can be considered as consisting of segments, such that

(a) all letters in segment i are smaller than all letters in segment $(i+1)$,
(b) the elements in a segment are in decreasing order.

Let $\pi = a_1 a_2 \cdots a_n$ be such an involution. We want to investigate what happens when we add the restriction to avoid p, where p is one of the patterns in $\{(1\text{-}2\text{-}3), (1\text{-}3\text{-}2), (2\text{-}1\text{-}3)\}$.

The avoidance of (1-2-3) limits the number of segments of π to two. Indeed, because of property (a) above, if π has more than two segments, a (1-2-3)-subword will be formed as $a_{i_1} a_{i_2} a_{i_3}$, where a_{i_1}, a_{i_2} and a_{i_3} can be arbitrarily chosen from the first, second and third segment respectively. Thus it follows that π in $\mathcal{I}_n(1\text{-}2\text{-}3, 2\text{-}3\text{-}1)$ is of the form

$$\pi = k \cdots 1 n \cdots (k+1),$$

that is, the involution π is uniquely determined by the choice of k, hence

$$|\mathcal{I}_n(1\text{-}2\text{-}3, 2\text{-}3\text{-}1)| = |\mathcal{I}_n(1\text{-}2\text{-}3, 3\text{-}1\text{-}2)| = n.$$

When the patterns (1-3-2) and (2-3-1) are to be simultaneously avoided, π can not have any "peaks". No letter a_i can be both preceded and succeeded by smaller letters. Consequently, if the letter 1 is in position k, then π must consist of the k smallest letters in decreasing order, followed by the letters that are larger than k in increasing order. To use the above notation, all segments except for the first one contain only one letter. Accordingly $\mathcal{I}_n(1\text{-}3\text{-}2, 2\text{-}3\text{-}1)$ consists of all permutations π of the form $\pi = k(k-1)\cdots 1(k+1)\cdots n$. Again the choice of k fixes the remaining involution, so it follows that

$$|\mathcal{I}_n(1\text{-}3\text{-}2, 2\text{-}3\text{-}1)| = |\mathcal{I}_n(1\text{-}3\text{-}2, 3\text{-}1\text{-}2)| = n.$$

Likewise, avoiding the patterns (2-1-3) and (3-1-2) implies that there can not be any "valleys", so, if the letter n is in position $(k+1)$, it must be preceded by the k smallest letters in increasing order and followed by the larger letters in decreasing order. This time each segment but the first one consists of a single letter, thus an involution π in $\mathcal{I}_n(2\text{-}1\text{-}3, 3\text{-}1\text{-}2)$

can be written $\pi = 12\cdots kn(n-1)\cdots(k+1)$, from which we conclude that

$$|\mathcal{I}_n(2\text{-}1\text{-}3, 2\text{-}3\text{-}1)| = |\mathcal{I}_n(2\text{-}1\text{-}3, 3\text{-}1\text{-}2)| = n,$$

since each involution is fully determined by k. □

Proposition 29. *We have that*

$$|\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)| = |\mathcal{I}_n(3\text{-}2\text{-}1, 3\text{-}1\text{-}2)| = F_{n+1},$$

where F_n denotes the nth Fibonacci number.

Note that, as in the proof of Proposition 28, it suffices to study either $\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$ or $\mathcal{I}_n(3\text{-}2\text{-}1, 3\text{-}1\text{-}2)$ since the two sets are indeed the same.

Consider π in $\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$. Being a $(2\text{-}3\text{-}1)$-avoiding involution, π can be described as consisting of segments, within which the letters are decreasingly ordered, according to the above characterization of $\mathcal{I}_n(2\text{-}3\text{-}1)$. Furthermore, the avoidance of $(3\text{-}2\text{-}1)$ implies that the decreasing sequences must be of length at most two, so π consists of fixed points and 2-cycles of consecutive letters. Hence

$$\pi = \cdots(k_i)\cdots(k_j, k_j+1)\cdots$$

gives a description of π in cycle form.

We prove that the involutions of the above form are counted by the Fibonacci numbers, first by combining two of the proofs of Proposition 6 with well known properties of the Fibonacci numbers and then by recursively constructing $\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$ from $\mathcal{I}_{n-1}(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$ and $\mathcal{I}_{n-2}(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$.

First proof. We begin with a proof that refers to the first proof of Proposition 6, in which a bijection Φ_n from the binary strings of length $(n-1)$, to $\mathcal{I}_n(2\text{-}3\text{-}1)$ is constructed. Given a binary string $x = x_1\cdots x_{n-1}$ in B_{n-1}, the corresponding involution is recursively built up from $[n]$ by considering the letters x_i, one at at time. We recall that $x_i = 1$ causes an inversion to be formed as the letter i is placed before $(i-1)$, whereas $x_i = 0$ implies that i is placed as the last element so far. From the construction it is easily seen that π contains a decreasing subsequence of length larger than 3 whenever x has two consecutive 1's and conversely that an x with no two consecutive 1's maps to an involution of the form

$$\cdots(k_i)\cdots(k_j, k_j+1)\cdots.$$

Hence there is a one-to-one correspondence between $\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$ and the binary strings of length $n-1$ with no consecutive 1's, which are known to be counted by F_{n+1}. For a reference, see for example [11]. Thus it follows that

$$|\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)| = |\mathcal{I}_n(3\text{-}2\text{-}1, 3\text{-}1\text{-}2)| = F_{n+1}.$$

□

Second proof. Next we relate to the third proof of Proposition 6, in which a bijection between $\mathcal{P}_{n-1}$, the subsets of $[n-1]$, and $\mathcal{I}_n(2\text{-}3\text{-}1)$ is defined. Let A be in $\mathcal{P}_{n-1}$. The corresponding involution is constructed from A by letting i be preceded by a larger letter if and only if i belongs to A. From the appearance of $\mathcal{I}_n(2\text{-}3\text{-}1)$ we see that there is only one choice of the larger letter preceding i, namely $i+1$. Thus, π has an occurrence of $(3\text{-}2\text{-}1)$ if and only if A contains two or more consecutive integers. It is well known that the number of subsets of $[n]$ with no consecutive integers is the nth Fibonacci number, see for instance [11]. Thus the result follows. □

Third proof. Finally we give a proof by induction. Recall that the Fibonacci numbers are defined by

$$F_n = F_{n-1} + F_{n-2}, \text{ where } F_0 = 0, F_1 = 1.$$

We will now show that the number of $(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$-avoiding involutions of $[n]$ satisfies the same recursion. Let π be such an involution. From the above description of $\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)$ as consisting only of fixed points and 2-cycles of consecutive letters we see that the letter n will be either a fixed point or contained in the cycle $(n-1, n)$. This gives us two ways of recursively constructing $\mathcal{I}_n(p, q)$ from $\mathcal{I}_{n-1}(p, q)$ and $\mathcal{I}_{n-2}(p, q)$ (for convenience we let the patterns $(3\text{-}2\text{-}1)$ and $(2\text{-}3\text{-}1)$ be denoted by p and q). Either n is added to a (p, q)-avoiding involution of $[n-1]$ or the cycle $((n-1)n)$ is added to a (p, q)-avoiding involution of $[n-2]$. Hence

$$\begin{aligned}\mathcal{I}_n(p,q) &= \{b_1 \cdots b_{n-1} n,\ b_1 \cdots b_{n-1} \in \mathcal{I}_{n-1}(p,q)\} \cup \\ &\quad \{c_1 \cdots c_{n-2} n(n-1),\ c_1 \cdots c_{n-2} \in \mathcal{I}_{n-2}(p,q)\},\end{aligned}$$

so

$$|\mathcal{I}_n(p,q)| = |\mathcal{I}_{n-1}(p,q)| + |\mathcal{I}_{n-2}(p,q)|.$$

Since $\mathcal{I}_0(p, q) = 0$ and $\mathcal{I}_1(p, q) = 1$, we conclude that

$$|\mathcal{I}_n(3\text{-}2\text{-}1, 2\text{-}3\text{-}1)| = |\mathcal{I}_n(3\text{-}2\text{-}1, 3\text{-}1\text{-}2)| = F_{n+1}.$$

□

Proposition 30. *We have that*

$$|\mathcal{I}_n(1\text{-}3\text{-}2, 3\text{-}2\text{-}1)| = |\mathcal{I}_n(2\text{-}1\text{-}3, 3\text{-}2\text{-}1)| = \lfloor n/2 \rfloor + 1.$$

Proof. Consider $\mathcal{I}_n(2\text{-}1\text{-}3, 3\text{-}2\text{-}1)$. We recall from the proof of Proposition 18 that a permutation π, with 1 in position k, avoids $(2\text{-}1\text{-}3)$ if and only if it can be written as $\sigma 1 \tau$, where $\sigma = a_1 a_2 \cdots a_{k-1}$ is a $(2\text{-}1\text{-}3)$-avoiding permutation of $\{n, (n-1), \ldots, (n-k+2)\}$ and $\tau = a_{k+1} a_{k+2} \cdots a_n$ is a $(2\text{-}1\text{-}3)$-avoiding permutation of $\{2, 3, \ldots, (n-k+1)\}$. Furthermore we recall that, when π is an involution, the letter 1 can be either a fixed point or in position k, where $k \geq n/2$. Let us investigate the latter case. Clearly, the letter k is in position 1. In order to avoid $(3\text{-}2\text{-}1)$, the remaining σ must consist of letters larger than k in increasing order. We

realize that this leads to absurdity whenever $k > n/2$ (since there are not enough larger letters). Thus, k must simultaneously be larger than or equal to $n/2$ and less then or equal to $n/2$, which is possible for integer k only when n is even. Then $k = n/2$, which determines π to be equal to $n/2 \cdots n1 \cdots (n/2-1)$. When 1 is a fixed point we can recursively apply the above reasoning to τ, so that an involution in $\mathcal{I}_n(2\text{–}1\text{–}3, 3\text{–}2\text{–}1)$ can be written as $\pi = 12\cdots(n-2k-1)(n-2k)\rho$. Here ρ is the $(2\text{–}1\text{–}3, 3\text{–}2\text{–}1)$-avoiding involution of $\{(n-2k), (n-2k+1), \ldots, n\}$ in which the smallest letter is in the middle position. Thus, these involutions are fully characterized by the choice of k, where k has to be less than or equal to $n/2$, hence

$$|\mathcal{I}_n(2\text{–}1\text{–}3, 3\text{–}2\text{–}1)| = \lfloor n/2 \rfloor + 1.$$

For the (1–3–2)- and (3–2–1)-avoiding involutions the proposition can be proved in a similar way, for which we omit the details. Let $\pi = a_1 a_2 \cdots a_n$ be such an involution. The letter n can either be a fixed point or, if n is even, in position $n/2$, in which case it determines the rest of π. By recursively repeating the arguments to the segment $a_1 a_2 \cdots a_{n-1}$ when n is a fixed point, we see that an $(1\text{–}3\text{–}2, 3\text{–}2\text{–}1)$-avoiding involution π can be written as $\rho(2k)(2k+1)\cdots n$, where ρ is the $(1\text{–}3\text{–}2, 3\text{–}2\text{–}1)$-avoiding involution of $[2k-1]$, in which the largest letter is in the middle position. Again the involutions are uniquely determined by the choice of k, hence the result follows. □

Proposition 31. *We have that*

$$\begin{aligned} |\mathcal{I}_n(1\text{–}2\text{–}3, 1\text{–}3\text{–}2)| &= |\mathcal{I}_n(1\text{–}2\text{–}3, 2\text{–}1\text{–}3)| = \\ |\mathcal{I}_n(1\text{–}3\text{–}2, 2\text{–}1\text{–}3)| &= 2^{\lfloor n/2 \rfloor}. \end{aligned}$$

Proof. We start with the case of (1–2–3)- and (2–1–3)-avoiding involutions of $[n]$. Note that the largest letter, n, has to be in position 1 or 2, because otherwise n will be preceded by two smaller letters that are either ordered as (1–2) or (2–1), causing occurrences of (1–2–3) and (2–1–3) respectively. On the other hand if n is the first (or second) letter, there can not be any (1–2–3)- or (2–1–3)-subwords containing 1 (or 2) or n, since n can not act as a 1 or a 2, as well as 1 (or 2) in position n will not do as a 3. Therefore, letting (1–2–3) and (2–1–3) be denoted by p and q, we can recursively construct $\mathcal{I}_n(p,q)$ from $\mathcal{I}_{n-2}(p,q)$, according to

$$\begin{aligned} \mathcal{I}_n(p,q) &= \{n a_2 \cdots a_{n-1} 1,\ \ \mathrm{proj}(a_2 \cdots a_{n-1}) \in \mathcal{I}_{n-2}(p,q)\} \cup \\ &\quad \{a_1 n a_3 \cdots a_{n-1} 2,\ \ \mathrm{proj}(a_1 a_3 \cdots a_{n-1}) \in \mathcal{I}_{n-2}(p,q)\}. \end{aligned}$$

Hence we get the recursion formula

$$|\mathcal{I}_n(p,q)| = 2 \cdot |\mathcal{I}_{n-2}(p,q)|, \text{ where } \mathcal{I}_1(p,q) = 1 \text{ and } \mathcal{I}_2(p,q) = 2,$$

from which it follows that

$$|\mathcal{I}_n(1\text{–}2\text{–}3, 1\text{–}3\text{–}2)| = 2^{\lfloor n/2 \rfloor}.$$

Next we consider (1–2–3)- and (2–1–3)-avoidance. This is similar to the above case, but now with the letter 1 playing the role of n. For an involution π to be in $\mathcal{I}_n(1\text{–}2\text{–}3, 2\text{–}1\text{–}3)$, the 1 can be placed either as the last or the penultimate letter of π. As above, none of the two corresponding cycles $(1, n)$ or $(1, n-1)$ can possibly contribute to the formation of (1–2–3)- or (2–1–3)-subwords. So, letting p and q denote the patterns (1–2–3) and (2–1–3) respectively, we see that $\mathcal{I}_n(p, q)$ can be recursively constructed from $\mathcal{I}_{n-2}(p, q)$ as

$$\begin{aligned} \mathcal{I}_n(p,q) \;=\; & \{na_2\cdots a_{n-1}1,\ \mathrm{proj}(a_2\cdots a_{n-1}) \in \mathcal{I}_{n-2}(p,q)\} \cup \\ & \{(n-1)a_2\cdots a_{n-2}1a_n,\ \mathrm{proj}(a_2\cdots a_{n-2}a_n) \in \mathcal{I}_{n-2}(p,q)\}. \end{aligned}$$

This will once again result in the recursion

$$|\mathcal{I}_n(p, q)| = 2 \cdot |\mathcal{I}_{n-2}(p, q)|,$$

with initial conditions $\mathcal{I}_1(p, q) = 1$ and $\mathcal{I}_2(p, q) = 2$. Thus the result follows.

Finally we turn to the (1–3–2)- and (2–1–3)-avoiding involutions of $[n]$. Let π be such an involution. From the proof of Proposition 18 we recall that, in order to avoid (2–1–3), the letter 1 must be in position $k \geq n/2+1$, or it is a fixed point. However, the simultaneous avoidance of (1–3–2) precludes the latter alternative in all cases except the identity permutation $\pi = 12\cdots n$. Assume therefore that the letter 1 is in position k. According to the proof of Proposition 18, π can be written as $\sigma 1 \tau$ where τ is a (2–1–3)-avoiding permutation of $\{2, \ldots, (n-k+1)\}$. We realize that the only choice of τ that makes π (1–3–2)-avoiding is in fact $\tau = 23\cdots(n-k+1)$, which corresponds to the initial segment $a_2\cdots a_k$ of π. We can then write $\pi = k(k+1)\cdots n\rho 12\cdots(n-k+1)$, where ρ is a (1–3–2)-avoiding involution of $\{n-k+2, n-k+1, \ldots k-1\}$, that is $\mathrm{proj}(\rho) \in \mathcal{I}_{n-2k}(1\text{–}3\text{–}2, 2\text{–}1\text{–}3)$. Accordingly, with p and q denoting (1–3–2) and (2–1–3) respectively, we can construct $\mathcal{I}_n(p, q)$ from $\{\mathcal{I}_{n-2k}(p, q)\}$, where

$k \leq n/2$. We have that

$$\begin{aligned}\mathcal{I}_n(p,q) &= \{na_2\cdots a_{n-1}1,\ \ \mathrm{proj}(a_2\cdots a_{n-1}) \in \mathcal{I}_{n-2}(p,q)\} \cup \\ &\quad \{(n-1)na_3\cdots a_{n-2}12,\ \ \mathrm{proj}(a_3\cdots a_{n-2}) \in \mathcal{I}_{n-4}(p,q)\} \cup \\ &\quad \vdots \\ &\quad \{k(k+1)\cdots na_{n-k+2}\cdots a_{k-1}12\cdots(n-k+1), \\ &\quad\ \mathrm{proj}(a_{n-k+2}\cdots a_{k-1}) \in \mathcal{I}_{n-2(n-k+1)}(p,q)\} \cup \\ &\quad \vdots \\ &\quad 12\cdots n.\end{aligned}$$

Thus $|\mathcal{I}_n(p,q)|$ satisfies the recursion

$$|\mathcal{I}_n(p,q)| = \sum_{k=1}^{\lfloor n/2 \rfloor} |\mathcal{I}_{n-2k}(p,q)|$$

and, since $|\mathcal{I}_1(p,q)| = 1$ and $|\mathcal{I}_2(p,q)| = 2$, we conclude that

$$|\mathcal{I}_n(1\text{–}3\text{–}2, 2\text{–}1\text{–}3)| = 2^{\lfloor n/2 \rfloor}.$$

□

Acknowledgement

I would like to thank my supervisor Einar Steingrímsson for the support and encouragement I have received during the writing of this masters thesis and also for teaching me combinatorics. I would also like to thank Sverker Lundin for showing me how to use Mathematica in my work with pattern avoidance.

References

[1] A. Claesson Generalised Pattern Avoidance *European J. Combin*, 22:961-9, 2001
[2] A. Claesson and T. Mansour. Enumerating Permutations Avoiding a Pair of Babson-Steingrímsson Patterns. Preprint, Chalmers University of Technology
[3] E. Babson and E. Steingrímsson. Generalized permutation patterns and a classification of the Mahonian statistics. *Sém. Lothar. Combin.*, 44:Art. B44b, 18 pp. (electronic), 2000.
[4] S. Jukna. *Extremal Combinatorics With Applications in Computer Science* Springer-Verlag, 2001
[5] S. Kitaev Multi-Aviodance of Generalised Patterns. Preprint, Chalmers University of Technology.
[6] S. Kitaev Generalised Pattern Avoidance with Additional Restrictions. Preprint, Chalmers University of Technology.
[7] D. E. Knuth. *The art of computer programming. Vol. 1: Fundamental algorithms.* Addison-Wesley Publishing Co., 1969.
[8] D. E. Knuth. *The art of computer programming. Vol. 3: Sorting and Searching.* Addison-Wesley Publishing Co., 1973.
[9] S. Lundin: Young-Tablåer och mönsterundvikande, Master's thesis, Chalmers University of Technology, 2001

[10] R. Simion and F. W. Schmidt. Restricted permutations. *European J. Combin.*, 6(4):383–406, 1985.

[11] N. J. A. Sloane and S. Plouffe. *The encyclopedia of integer sequences.* Academic Press Inc., San Diego, CA, 1995. Also available online: http://www.research.att.com/~njas/sequences/.

[12] R. P. Stanley. *Enumerative combinatorics. Vol. I.* Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, CA, 1986.

Matematik, Chalmers tekniska högskola och Göteborgs universitet, S-412 96 Göteborg, Sweden

E-mail address: wulcan@math.chalmers.se

Mathematical Chats Between Two Physicists

Aviezri S. Fraenkel

To Martin Gardner — The Master of recreational mathematics

The Luncheon Chat

Joyce is a physicist doing statistical mechanics, and Gill a nuclear physicist specializing in particle interactions. While relaxing with their cups of coffee after a tasty enjoyable light lunch at the TEX (TasteEnjoyrelaχ) — the *Sciences Club* of the University — they began to chat about some common aspects of their specialties.

Gill: The interaction between elements such as particles, nucleons, spins, etc. that are "close" to one another is common to our two disciplines. I wonder whether a lesson can be learned by viewing these phenomena in a unified manner.

Joyce: Hmm...a nice idea. I think that to do this we need some abstract model that reflects the basic common properties of these interactions, and that is amenable to mathematical analysis, such as working with two elements 1 and 0, that form a field called by those pompous mathematicians the *Galois field* of two elements, GF(2).

G: Yes, GF(2) has the advantage that $1 = -1$, so the rule $1 + 1 = 0$ in this field is the same as the annihilation rule of particles and spins: $1 - 1 = 0$. We have of course $0 + 1 = 1 + 0 = 1$ and $0 + 0 = 0$, as well as $1 + 1 = 0$. These addition rules are also known as *Nim sum* or *Xor* — *exclusive or*. Furthermore, to model interactions that are not necessarily neighboring vertices on a grid, it seems best to have a directed graph $G = (V, E)$ — that mathematicians, always tending to succinctness, call *digraph* for short — on whose vertices V

[†] Aviezri S. Fraenkel is a scholar and computicianeer — computer scientist, mathematician and engineer, who worked on the design of one of the earliest digital computers and has fathered the Responsa Retrieval Project.

the "particles" 0 and 1 are initially distributed. Selecting a particle on a vertex u, it is complemented as well as all its neighbors along edges directed away from u.

J: What you describe is a system called *cellular automata* by those inflated logicians, mathematicians and computer scientists, a manifestation of which is the *Merlin Magic Square* game manufactured by Parker Brothers (but Arthur-Merlin games are something else again). Quite a bit is known about such solitaire games. Anyway, a huge literature has been accumulating on cellular automata. A small example, intersecting with solitaire games, is [Gol91], [Pel87], [Sto89], [Sut88], [Sut89], [Sut90], [Sut95]. Incidentally, related but different solitaires are *chip firing games*, see e.g., [BL92], [Lóp97], [Big99].

What seems more attractive and new is to transform these solitaire games into two-player games, where the player first achieving 0s on all the non-leaf vertices wins and the opponent loses. If there is no last move, the outcome is a draw. Moreover, this version will appeal to many of my colleagues who have turned their attention to biology, such as protein folding, where the main aim is to tinker with nature, in order to achieve some doubtful benefits such as designing specialized medicines and genetic engineering (alias tinkering)...For want of a better name, we might call them *Cellata* games, since it reminds me both of the Italian cuisine that I just enjoyed, and of cellular automata.

G (taking a paper napkin and beginning to draw on it): I like your idea, and I share your belief that it appears to be new and interesting. In most of the solitaire games you have mentioned, *any* order of the moves produces the same result. To promote your suggestion of tinkering, I think it's then best to permit the players to select only an *occupied* vertex, i.e., a vertex occupied by a 1. So a move in the game consists of selecting an occupied vertex and *firing* it, i.e., complementing it together with all its directed neighbors. The player making the last move wins. If there is no last move, the outcome is a draw...the order of the moves is then definitely important, unlike in those solitaires. ...Here now is a suggested game on two components with an initial $0,1$-distribution, where 1s are indicated by $\star$s (Figure 1) and vertices occupied by 0s remain unlabeled. As a gentleman, I'm used to "Ladies First" etiquette, so I graciously offer you to move first.

J (pulling a PalmCrash from her handbag and hammering away furiously on its buttons): You propose to play a *sum* of games, i.e., a move consists of selecting a component and firing an occupied vertex on it. The player making the last move in the entire digraph wins, and her opponent loses...it seems to me that your gentlemanly gesture is all but gallant. It is indeed patronizing, since whatever I'll do from this position, you can win. I'll therefore add to your two components two simplified versions, namely deleting vertices 5 and 6 on the two components, with $\star$s as indicated (Figure 3). Under these circumstances I accept your offer to make the first move in the sum consisting of all the 4 components.

G (blushing): Well...I really hadn't expected you to find out so soon...I see that on the game consisting of the four components you can win by making an appropriate move.... Since it seems that both of us understand the win/lose

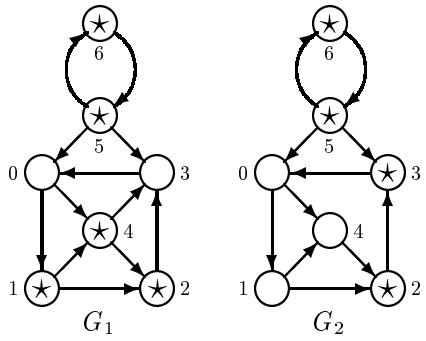


Figure 1: A two-player game $G_1 + G_2$ on cellular automata. A move consists of selecting a vertex v marked with a $\star$ and "firing" it. Once fired, the $\star$ is removed, and $\star$s are placed on every vertex v points to. If two $\star$s appear at a vertex, both are annihilated. Two players play by taking turns firing a vertex. The first player unable to move loses, and the opponent wins. If there is no last move, the outcome is a draw. The result of firing vertex 4 in G_1 is shown in G_1 of Figure 2.

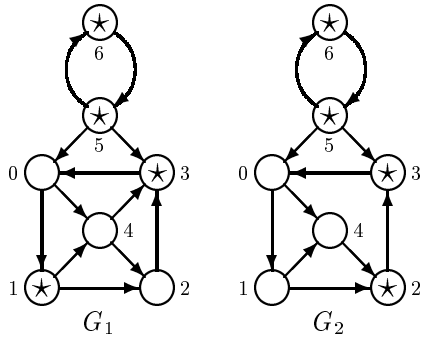


Figure 2: Game $G1 + G2$ from Figure 1 after one move.

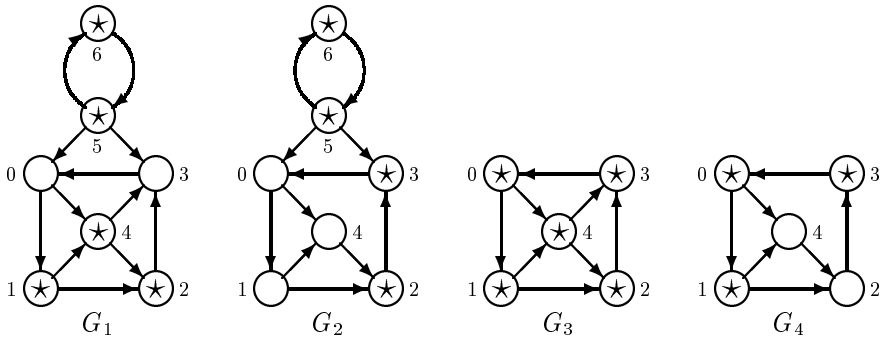


Figure 3: Adding two more components, G_3 and G_4.

positions of this game, I suggest to play the same game with the small change of adjoining a $\star$ on vertex 0 of G_1.

J (consulting her PalmCrash once more and then rising): Alright, the initial position is now a draw. Since we seem to have mastered also the draw positions, it's time to head back to our offices and do some serious physics...such as deciding the computational complexity of Cellata games.

A Conversation in Joyce's Office

The next day, Professor Gill Andrin strolled over to Professor Joyce Prato's office.

Gill: Good morning Joyce, I was wondering how you found me out so quickly yesterday when I offered you to play first on Figure 1.

Joyce: Hi Gill, I'm already used to your tricks. When I saw that you proposed to play on two components of a game that obviously has cycles, I assumed that you had computed the generalized Sprague-Grundy function γ for the game [Smi66], [Con76, Ch. 11], [FY86]; otherwise you would hardly be able to beat a sharp opponent and be so smug about it. (Walking over to the whiteboard.) I suspected that γ is *additive* (also called *linear*) on the digraph $G = (\mathbf{V}, \mathbf{E})$, induced by the given groundgraph $G = (V, E)$, where $\mathbf{V}$ is the collection of all subsets of vertices from $V = (z_1, \ldots, z_n)$. That is, $\gamma(\mathbf{u}) \oplus \gamma(\mathbf{v}) = \gamma(\mathbf{u} \oplus \mathbf{v})$ whenever either $\gamma(\mathbf{u}) < \infty$ or $\gamma(\mathbf{v}) < \infty$. The $\oplus$ denotes Nim sum, and every $\mathbf{w} \in \mathbf{V}$ is an n-dimensional binary vector with 1s precisely in locations i where z_i is an occupied vertex in G. I proved linearity with the aid of my PalmCrash. This enabled me to compute γ very easily.

G: Congratulations. But how could you possibly prove linearity with the aid of a computer?

J: I took lots of examples, and it always confirmed linearity. There was no counterexample at all.

G: Hmm...Is this a standard method of proof in statistical mechanics?

J: Well, I don't need the formal proofs of those highbrow mathematicians. I perceive truth when I meet it.

G: It appears that you have been a little hard on mathematicians, especially yesterday. Many phenomena are counterintuitive. I concur with the mathematicians that proofs of claims are necessary, though the precise notion of "proof" might be debatable. Of course one might formulate a *conjecture*, and base further results on it.

To come back to our Cellata game, $\gamma(u)$, when finite, is the smallest nonnegative integer not appearing among the *options* (direct followers) of vertex u...instead of using n-dimensional vectors to denote vertices of $\mathbf{V}$, it will now be more convenient to denote them by $n_1 \dots n_k$, where $z_{n_1}, \dots, z_{n_k}$ are the occupied vertices of V. Thus you presumably noticed that on G_1, $\gamma(4) = 0$, since it has the as yet unlabeled option 23, that has the option Φ, the configuration with no $\star$s, for which obviously $\gamma(\Phi) = 0$. Similarly, $\gamma(02) = \gamma(13) = 0$. Using linearity, we then get

$$\mathbf{V}_0 = \{\Phi, 4, 02, 13, 024, 134, 0123, 01234\},$$

where $\mathbf{V}_i$ is the subset of $\mathbf{V}$ on which γ assumes the value i ($i < \infty$). In fact, γ is a homomorphism from $\mathbf{V}^f$ (the linear subspace of the vector space $\mathbf{V}$ on which γ is finite) onto $\mathrm{GF}(2)^t$ for some nonnegative integer t with kernel $\mathbf{V}_0$ and quotient space $\mathbf{V}^f/V_0 = \{\mathbf{V}_i : 0 \le i < 2^t\}$, and $\dim(\mathbf{V}^f) = t + \dim(\mathbf{V}_0)$. We have $\mathbf{V}^\infty = \mathbf{V} \setminus \mathbf{V}^f$, where $\mathbf{V}^\infty$ is the subset on which $\gamma = \infty$. For G_1, $\gamma(23) = 1$, since its only options are $\{\Phi, 02\} \subseteq \mathbf{V}_0$. Also $\gamma(56) = 2$. We thus get the cosets

$$\mathbf{V}_1 = 23 \oplus \mathbf{V}_0 = \{23, 234, 03, 12, 034, 124, 01, 014\},$$

$\mathbf{V}_2 = 56 \oplus \mathbf{V}_0$, $\mathbf{V}_3 = 0356 \oplus \mathbf{V}_0$, $\dim \mathbf{V}_0 = 3$, $\dim \mathbf{V}^f = 5$, $t = 2$.

For G_2 we get

$$\mathbf{V}_0 = \{\Phi, 1, 02, 34, 012, 134, 0234, 01234\},$$

$\mathbf{V}_1 = 23 \oplus \mathbf{V}_0$, $\mathbf{V}_2 = 56 \oplus \mathbf{V}_0$, $\mathbf{V}_3 = 0356 \oplus \mathbf{V}_0$, $\dim \mathbf{V}_0 = 3$, $\dim \mathbf{V}^f = 5$, $t = 2$.

It follows that the γ-value on G_1 is $\gamma(56) \oplus (124) = 2 \oplus 1 = 3$, and also on G_2 we have a γ-value of 3. Their Nim sum is thus 0, which means that whoever moves from this position loses. Is this how you figured things out?

J: Precisely. For G_3 and G_4 that I adjoined to the game, we have $\mathbf{V}_0$, $\mathbf{V}_1$ as for G_1 and G_2 respectively, but $\dim \mathbf{V}_0 = 3$, $\dim \mathbf{V}^f = 4$, $t = 1$. Therefore on G_3, $\gamma(01234) = 0$ and on G_4, $\gamma(013) = \gamma\big((23) \oplus (012)\big) = 1$. Thus firing vertex 0 on G_4, results in 34, with $\gamma(34) = 0$. This is a winning move, since γ now vanishes on the entire digraph.

G: Yes. By adjoining a $\star$ at vertex 0 in G_1, we get $\gamma(012456) = \infty$, so the sum of the four components has also γ-value infinity, and the outcome is now a draw, as you said. I better leave now, as I got to teach my Graduate Mesoscopic Physics course.

J: Enjoy — bye.

The Truncated Chat in the Faculty Room

Joyce and Gill met again next day in the Faculty room where doughnuts, cookies, coffee and tea were served in anticipation of an important gathering.

Joyce (moving to the whiteboard): I thought it would be interesting to change the rules, a particular case of which would be to fire the selected vertex u and complement precisely any *two* of its options in the groundgraph if $d_{\rm out}(u) \geq 2$; and complement all the options of u if $d_{\rm out}(u) \leq 2$. (I'm now using the terminology "firing" in a new sense: complementing the selected vertex and some subset of its options.) I conjecture that additivity holds also for this game. The digraph $G(s)^2$ I would like to play this game on depends on a parameter $s \in \mathbb{Z}^+$. It has vertex set $\{x_1, \ldots, x_s, y_1, \ldots, y_s\}$, and edges:

$$\begin{array}{rcll} F(x_i) & = & y_i & \text{for } i = 1, \ldots, s, \\ F(y_k) & = & \{y_i \colon 1 \leq i < k\} \cup \{x_j \colon 1 \leq j \leq s \text{ and } j \neq k\} & \text{for } k = 1, \ldots, s. \end{array}$$

As an example, I'm drawing $G(4) = G(4)^2$ on the board (Figure 4). Suppose we play on $G(7)$, and place 1s precisely on the 8 vertices $x_7, y_1, \ldots, y_7$. Can you figure out the nature of this position?

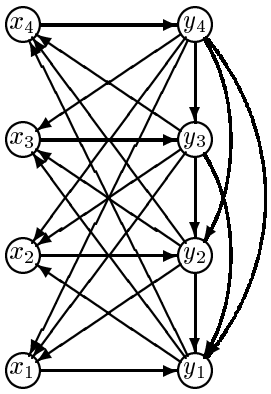


Figure 4: Playing on a parametrized digraph.

Gill: (fingering the knobs of the WallComp next to the whiteboard): Before doing that, why didn't you consider the G^1 version, i.e., firing an occupied vertex (in your new sense of "firing"), and complementing precisely *one* of its options?

J: Well, this would be a pure particle physics game without much appeal to statistical mechanics, and it was *you* who had suggested to consider a unified approach. Besides, this special case was solved in [Fra74], [FY76], [FY82], where a polynomial strategy was formulated. The misère version was analyzed in [Fer84].

G: I expected you to say this, but it gave me time to think about the question you asked me...I concur with your conjecture about additivity. It seems that

though the groundgraph $G(s)$ has no leaf, the game-graph $G(s)$ has no γ-value ∞. It also appears that any collection of x_i is in $\mathbf{V}_0$. The value of the y_i seem to be more tricky...I think that $\gamma(y_i)$ =the ith *odious* number, where the odious numbers are those positive integers whose binary representations have an odd number of 1-bits. Incidentally, odious numbers arise in the analysis of other games, such as Grundy's game, Kayles, Mock Turtles, Turnips. See [BCG82]. They arose earlier in a certain two-way splitting of the nonnegative integers [LM59] (but without this odious terminology...). More information about this and over 54,000 other integer sequences is available on-line from

`http://www.research.att.com/~njas/sequences/`

thanks to the mathematician Neil Sloane, who probably contributed more to a larger number of mathematicians than any other mathematician!

For the position concocted by you, $\gamma(x_7y_1\dots y_7) = 0\oplus 1\oplus 2\oplus 4\oplus 7\oplus 8\oplus 11\oplus 13 = 14$. Thus the player to move can win: either by firing y_7 and complementing y_1, y_2, or by firing y_6 and complementing y_1, y_3, or by firing y_5 and complementing y_2 and y_3.

J: Very nice...suppose we take an identical clone of $G(s)^2$, and begin with precisely the same initial configuration, but change the rule for the clone: complement the selected vertex u together with any *three* of its options if $d_{\rm out}(u)\geq 3$; and all the options of u if $d_{\rm out}(u)\leq 3$. We better call this new clone $G(s)^3$, to distinguish it from $G(s)^2$. Can the first player win also here?

G (moving to within reach of both the whiteboard and the WallComp): Let's see...on the clone, all collections of an even number of x_i are in $\mathbf{V}_0$; and $\mathbf{V}^f$ consists precisely of all collections of an *even* number of 1s...we seem to have $\gamma(x_jy_j)$ = smallest nonnegative integer not the Nim sum of at most three $\gamma(x_iy_i)$ for $i<j$. Thus $\gamma(x_7y_1\dots y_7) = 0\oplus 1\oplus 2\oplus 4\oplus 8\oplus 15\oplus 16\oplus 32 = 48$. So firing y_7 and complementing y_6 and any two of the x_i ($i<7$) is a winning move. ...Incidentally, the sequence $\{1,2,4,8,15,16,32,51,\dots\}$ appears also in Neil's Encyclopædia, and has been used in [BCG82] for a special case of the game "Turning Turtles".

J: How about playing the sum of $G(s)^2$ and $G(s)^3$ with the same given initial position on both clones?

G: That's easy. The value of the sum is simply the Nim sum of their γ-values which is $14\oplus 48$. To win we have to move in $G(s)^3$ to a position with γ-value 14. There is a unique winning move of changing the γ-value 32 to 30. This is affected by firing y_7 and complementing y_6, y_5 and y_1...I hear in the corridor the President talking with the Cabinet Minister of Science approaching...we better adjourn before we'll have to explain to the minister that we are playing a game.

J (moving to the WallComp): Not before we briefly summarize where we stand...We still should address the question of the computational complexity of Cellata games...and yes, I concede that it would be nice to prove additivity formally for the family of all Cellata games...In these games, is every draw position necessarily such that *every* move from it leads to another draw? This is the case for all the games we considered, but it would be nice to provide a

case where this doesn't hold...We played *impartial* games. How about playing a sort of *partizan* game on, say, $G(s)^3$ and $G(s)^2$ simultaneously, i.e., one player follows the $G(s)^3$ rules and her opponent the $G(s)^2$ rules?...I think I can see some interesting applications in fields other than physics. Incidentally, the case of $G(s)^4$, where a vertex on G_s is fired and any *four* of its options are complemented, seems to give rise to the sequence $1, 2, 4, 8, 16, 31, 32, 64, 103, \ldots$. It is the sequence $\gamma(x_j y_j)$ defined as the smallest nonnegative integer not the Nim sum of at most four earlier terms. This sequence was not in the Encyclopædia of integers, so I just sent a message, via the WallComp, to your latest mathematics hero Neil Sloane, together with the fact that it appears in Table 3, Chapter 14 of [BCG82]. Note that the strategy of our Cellata games on just Figure 4 alone subsumes and unifies that of a battery of games there...I just noted that Sloane has added the new sequence into his Encyclopædia.

If it wouldn't be for our own University President who seeks to elicit more money from this narrow-minded minister, I'd proudly tell the latter that we are playing a game, followed by a quote from the founder of our *Sciences Club*:

"*...A third purpose of this book is to have fun. Indeed, pleasure has probably been the main goal all along. But I hesitate to admit it, because computer scientists want to maintain their image as hard-working individuals who deserve high salaries. Sooner or later society will realise that certain kinds of hard work are in fact admirable even though they are more fun than just about anything else.*" ([Knu93b, p. iii], see also [Knu77].)

Bibliography

[BCG82] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for your Mathematical Plays* (volumes I and II). Academic Press, London, 1982. Translated into German: *Gewinnen, Strategien für Mathematische Spiele* by G. Seiffert, Foreword by K. Jacobs, M. Reményi and Seiffert, Friedr. Vieweg & Sohn, Braunschweig (four volumes), 1985.

[Big99] N. L. Biggs. Chip-firing and the critical group of a graph. *Journal of Algebraic Combinatorics*, 9(1):25–45, 1999.

[BL92] A. Björner and L. Lovász. Chip-firing games on directed graphs. *Journal of Algebraic Combinatorics*, 1(4):305–328, 1992.

[Con76] John H. Conway. *On Numbers and Games.* Academic Press, London/New York, 1976. Translated into German: *Über Zahlen und Spiele* by Brigitte Kunisch, Friedr. Vieweg & Sohn, Braunschweig, 1983.

[Fer84] Thomas S. Ferguson. Misère annihilation games. *Journal of Combinatorial Theory. Series A*, 37:205–230, 1984.

[Fra74] Aviezri S. Fraenkel. Combinatorial games with an annihilation rule. In J. P. LaSalle, editor, *The Influence of Computing on Mathemati-*

cal Research and Education (*Proc. Symp. Appl. Math., Vol. 20, Univ. Montana, 1973*), pages 87–91. American Mathematical Society, Providence, RI, 1974.

[FY76] A. S. Fraenkel and Y. Yesha. Theory of annihilation games. *Bulletin of the American Mathematical Society*, 82(5):775–777, 1976.

[FY82] A. S. Fraenkel and Y. Yesha. Theory of annihilation games — I. *Journal of Combinatorial Theory. Series B*, 33(1):60–86, 1982.

[FY86] A. S. Fraenkel and Y. Yesha. The generalized Sprague–Grundy function and its invariance under certain mappings. *Journal of Combinatorial Theory. Series A*, 43(2):165–177, 1986.

[Gol91] Eric Goles. Sand piles, combinatorial games and cellular automata. *Mathematics and its Applications*, 64:101–121, 1991.

[Knu77] D. E. Knuth. Are toy problems useful? *Popular Computing*, 5:3–10, 1977.

[Knu93b] D. E. Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. ACM Press, New York, 1993.

[LM59] J. Lambek and L. Moser. On some two way classifications of integers. *Canadian Mathematical Bulletin*, 2:85–89, 1959.

[Lóp97] C. M. López. Chip firing and the Tutte polynomial. *Annals of Combinatorics*, 1(3):253–259, 1997.

[Pel87] D. H. Pelletier. Merlin's magic square. *American Mathematical Monthly*, 94(2):143–150, 1987.

[Smi66] C. A. B. Smith. Graphs and composite games. *Journal of Combinatorial Theory*, 1:51–81, 1966. Reprinted in slightly modified form in: *A Seminar on Graph Theory* (F. Harary, ed.), Holt, Rinehart and Winston, New York, NY, 1967.

[Sto89] D. L. Stock. Merlin's magic square revisited. *Amererican Mathematical Monthly*, 96(7):608–610, 1989.

[Sut88] K. Sutner. On σ-automata. *Complex Systems*, 2(1):1–28, 1988.

[Sut89] K. Sutner. Linear cellular automata and the Garden-of-Eden. *Mathematical Intelligencer*, 11(2):49–53, 1989.

[Sut90] K. Sutner. The σ-game and cellular automata. *American Mathematical Monthly*, 97(1):24–34, 1990.

[Sut95] K. Sutner. On the computational complexity of finite cellular automata. *Journal of Computer and System Science*, 50(1):87–97, 1995.

RICE UNIVERSITY

On Eliminating Square Paths in a Square Lattice

by

Nikki L. Williams

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Arts

Approved, Thesis Committee:

Nathaniel Dean, Chairman
Associate Professor of Computational and
Applied Mathematics

Richard A. Stong
Professor of Mathematics

Richard A. Tapia
Noah Harding Professor of Computational
and Applied Mathematics

Yin Zhang
Associate Professor of Computational and
Applied Mathematics

Houston, Texas

April, 2000

Abstract

On Eliminating Square Paths in a Square Lattice

by

Nikki L. Williams

Removing the minimum number of vertices or points from a square lattice such that no square path exists is known as the square path problem. Finding this number as the size of the lattice increases is not so trivial. Results provided by Erdös-Pósa and Bienstock-Dean provides an upper bound for eliminating all cycles from a planar graph but sheds little light on the case of the square lattice. This paper provides several values for the minimum number of vertices needed to be removed such that no square path exists.

Acknowledgments

I would like to thank the consistent support and advice of many people.

First and foremost I have to thank God for his grace and allowing me to make it thus far. He deserves all the honor and the praise.

I am very grateful for my advisor Dr. Nathaniel Dean who introduced me to this problem. Thanks Nate for hanging in there with me. You are indeed "The Best Advisor in the Whole World." I also want to sincerely thank Dr. Richard Tapia for his constant support, advice, time, and teaching me how to dance. I am extremely grateful to Dr. Yin Zhang and Dr. Richard Stong for much guidance during this process.

Much thanks to the National Defense Science and Engineering Graduate Fellowship for supporting me. Also much gratitude to the Andrew W. Mellon Foundation and the SSRC/Mellon Graduate Fellows.

Thanks to Dr. Robert Bixby, Dr. Cassandra McZeal, Dr. Jennifer Rich, Tim Redl, Sripriya Venkataraman, and Melisa Ramos for all that you have done.

Also, I want to acknowledge Dr. Rhonda Hughes and Dr. Sylvia Bozeman for constantly inspiring me to reach my goals. You both are true role models. Thanks Dr. Pamela J. Williams for taking me in and showing me the way.

I also want to thank the following: Bible Book Club (BBC) members; Lilly Grove Missionary Baptist Church and Rev. Terry K. Anderson for your prayers and encouragement; and Silver Hill Baptist Church.

Sincere thanks to the following individuals who have had a major impact in my life academically and spiritually: Nikeya C. Harper (Rin-Tin-Tin) for being my soror, my sister, and my true friend; Donald Williams for your advice; Ronald Session for your

prayers long talks, and testimonies; Dragon for believing me when I really needed it most; Illya Hicks for constantly pushing me; Boo; and Rayzov Sonlight for being virtually who you are.

Lastly, I must thank my family: Thanks to my mommy, Mary R. Williams, for her constant support, advice, and wisdom. You have always believed that I could reach the stars and never once failed to tell me. Thanks mom! Thanks to my dad, Jimmie L. Williams, for his continuous support and teaching me perseverance. Thanks to my sister, LaTanya C. Williams, for everything! Words just can not explain! Thanks to my aunt, Marjorie Rich, for all that you have done for me over the years. Your kind works never go unappreciated. Thanks to the rest of my family and friends for all that you have done for me.

Contents

Illustrations

Tables

Chapter 1

Introduction

Consider an $n \times n$ square grid (also called a lattice) with vertices colored either black or white. A *path* is a chain of edges such that the end vertex of one edge is the beginning vertex of the next edge and no vertices are repeated, except possibly the beginning is the end. A *square path* is a closed path in the shape of a square with sides parallel to the edges of the lattice. Define $M(n)$ to be the minimum number of black points needed for an $n \times n$ square lattice so that every square path has at least one black point. We seek to find $M(n)$ for any given n. This is known as the Square Path Problem [5]. For example, $M(2) = 1$. For the single point or 1×1 lattice we define $M(1) = 0$. Even though these examples are obvious, finding $M(n)$ is not so trivial as n increases.

According to the 1988 editors of *Mathematics Magazine* [7, 4] the Square Path Problem (SPP), which is one of three problems posed by Morris [5], has not been solved. Several web and library searches using key words, such as Hamiltonian square path, square cycle, square path, square circuit, square lattice, lattice packing, path packing, and packing, were done in order to obtain information on the problem. Some of these searches returned nothing while others returned articles that were not related to the SPP.

However, the literature search did reveal one related problem. If we require the black points in the SPP to cover not only square paths (or circuits) but also circuits of any shape, then this new value is referred to as $\tau(n)$. We say a graph is *planar* if it can be drawn in the plane such that there are no edge crossings. Thus, the grid in the SPP is planar. This related problem seeks to find the minimum number of

vertices to be removed from a planar graph such that no circuit exists. Dean refers to this minimum number as $\tau(n)$. In relation to the Square Path problem discussed in this work, $\tau(n)$ provides an upper bound for $M(n)$. Several upper bounds for n up to 14 are included in the Results chapter.

Bienstock-Dean [1] consider covering points of a planar graph with a minimum number of faces. The Erdös-Pósa theorem [3] on independent circuits in graphs can be applied when graphs with a specific embedding are considered. Erdös-Pósa define a family of cycles in a graph *independent* if they are pairwise vertex-disjoint.

The SPP can be transformed into a node covering problem in a bipartite graph. Let A,B be a bipartition of our graph. Then the set A contains a node for every vertex in the lattice, and the set B contains a node for every square path. An edge joins a vertex in A to a vertex in B if a node in the set A is a black point. This problem can be stated as follows: Find the minimum cardinality set S of nodes in the set A such that every node in the set B is adjacent to a member of S.

Chapter 2

Square and Non-square Lattices

2.1 Bounds for M(n)

Erickson [2] showed that

$$\lim_{n\to\infty} \frac{M(n)}{n^2}$$

exists and that

$$\frac{M(n)}{(n-1)^2} \geq \frac{2}{7}.$$

He also replicated a pattern to show that $\leq 2/7$ of the points of the $n \times n$ lattice need to be black. Thus,

$$\frac{2}{7}(n-1)^2 \leq M(n) \leq \frac{2}{7}n^2.$$

Proof of lowerbound (Erickson): Let B be a black point in the lattice, and suppose S is a 2×2 square path that passes through B. We will assign B a "credit" of 1/k if S passes through exactly k black points. Let T(B) be the sum of all the credits assigned to B as S varies over all 2×2 squares that pass through B.

Note that the sum of T(B) as B varies over all black points in the square array is $(n-1)^2$ since each of the 2×2 arrays contributes 1 to the total.

It is clear that the sum of $T(B) \leq 1$ if B is a corner point, and $T(B) \leq 2$ if B is on the outer edge. Suppose that B is a point in the interior of the lattice. It lies on exactly four 2×2 square paths, and there must be at least one black point on the 3×3 square path surrounding B. Thus, for such a B, $T(B) \leq 7/2$.

Thus, in all cases, $T(B) \leq 7/2$, and so $(7/2)M(n) \geq (n-1)^2$, or equivalently, $M(n) \geq 2(n-1)^2/7$. □

Hence,

$$\lim_{n\to\infty} \frac{M(n)}{n^2} = \frac{2}{7}.$$

2.2 Trivial values for M(n)

In order to find a general formula for $M(n)$, values for n small were easily computed. The following results were used to obtain more information about $M(n)$.

Theorem 2.1 $M(2) = 1$.

Theorem 2.2 $M(3) = 2$. Moreover, if one black point is a corner point, then the other is the center.

Figure 2.1 Two configurations to show $M(3) = 2$.

Note that Figure 2.1 shows that $M(3) = 2$ does not have a unique solution. Thus, there might be several optimal configurations.

Theorem 2.3 $M(4) = 4$.

Proof Since a 4×4 lattice contains four distinct 2×2 lattices and $M(2) = 1$, then $M(4) \geq 4$. Choosing the four points indicated in Figure 2.2 eliminates all square paths, and so $M(4) = 4$. □

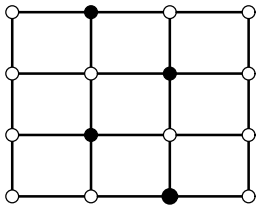


Figure 2.2 $M(4) = 4$

2.3 Non-square Lattices

Proving $M(n)$ for a specific n can be difficult as n increases. One possible method involves considering non-square lattices which partition the lattice into several regions. This technique allowed $M(n)$ to be determined for larger values of n.

Suppose we are given an $a \times b$ lattice or rectangle. Then we let $M(a, b)$ denote the minimum number of points to be removed from an $a \times b$ size rectangle such that no square path exists. Note that $M(a, b) = M(b, a)$. The following results are trivial.

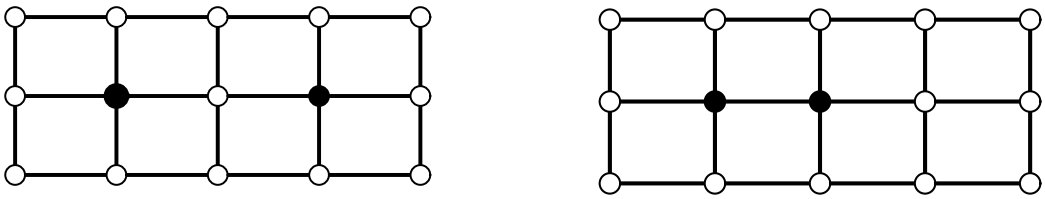
Lemma 2.1 $M(2, 3) = 1$.

Lemma 2.2 $M(3, 4) = 2$, and the solution is unique.

The following results are used in the next section to prove cases for square lattices.

Lemma 2.3 $M(3,5) = 3$.

Proof Since a 3×3 lattice is contained in a 3×5 rectangle, then $M(3,5) \geq 2$. Suppose $M(3,5) = 2$. Since we want to remove a minimum number of vertices, then we want to choose points that eliminate as many square paths as possible. Choosing the points in Figure 2.3(a) covers the eight distinct 2×2 square paths. However, a 3×3 square path exists containing the center point. If we were to choose the two center points indicated in Figure 2.3(b), then there is a 2×2 square path not covered. Thus, at least one more black point is needed. Hence, $M(3,5) \geq 3$. In fact, the configurations in Figure 2.4 show that $M(3,5) \leq 3$. □



(a) $M(3,5) \geq 2$ (b) $M(3,5) \geq 2$

Figure 2.3 Two subfigures

Lemma 2.4 $M(4,5) = 4$, and the solution is unique.

Proof Since a 5×4 contains a 4×4 square lattice and $M(4) = 4$, then $M(4,5) \geq 4$. In fact, Figure 2.5 shows that $M(4,5) = 4$ by choosing the four corners of the inner 2×3 rectangle.

Figure 2.4 Two configurations showing that $M(3,5) \leq 3$

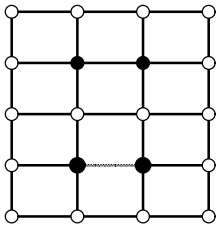


Figure 2.5 $M(4,5) = 4$

It's easy to see that there are only 4 solutions for $M(2,5) = 2$. When we partition the 4×5 lattice into two 2×5 lattices, the only solutions that avoid a square path are combined as shown in Figure 2.5. Thus, the solution is unique. □

Lemma 2.5 $M(3,7) = 4$.

Proof We can divide the 3×7 rectangle into a 3×3 lattice and a 3×4 rectangle. Since $M(3,4) = 2$ and $M(3) = 2$, then $M(3,7) \geq 4$. But, Figure 2.6 shows that $M(3,7) \leq 4$. □

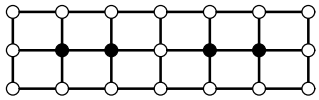


Figure 2.6 $M(3,7) = 4$

Lemma 2.6 $M(4,7) = 6$.

Proof Since we can divide the 4×7 rectangle into a 4×4 lattice and a 3×4 rectangle, then $M(4,7) \geq 6$. But, Figure 2.7 shows that $M(4,7) \leq 6$. □

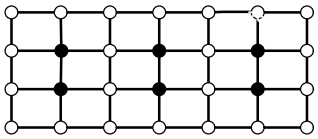


Figure 2.7 $M(4,7) = 6$

Lemma 2.7 $M(5,6) = 7$.

Proof The 5×6 lattice can be partitioned into two 3×3 lattices and three 2×2 lattices which are all pairwise disjoint. Hence, $M(5,6) \geq 1+1+1+2+2 = 7$. Figure 2.8 shows that $M(5,6) \leq 7$. □

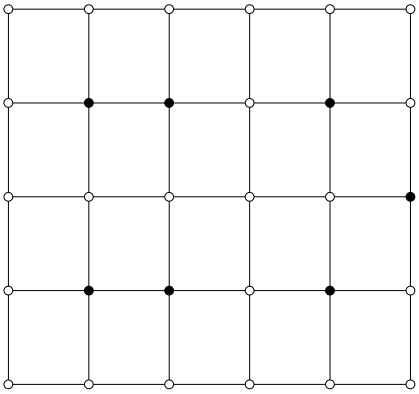


Figure 2.8 $M(5,6) = 7$

2.4 Square Lattices

Recall that the original problem seeks to remove the minimum number of vertices on a square lattice such that no square path exists. Our approach will be for us use the results from the previous section to prove values of $M(n)$. We begin with $n = 5$.

Theorem 2.4 $M(5) = 6$.

Proof

Since the 5×5 lattice contains a 4×4 lattice, then $M(5) \geq 4$. Suppose all square paths can be covered by five black points. Divide the 5×5 lattice into four 2×3 rectangles as in Figure 2.9, and label the regions I, II, III, and IV such that the middle point is not included in any or the 2×3 rectangles.

Case 1: Center is a black point.

If the center is a black point, then there is one point from each region. We choose the inner middle point so that all the square paths in that region are covered. Consider

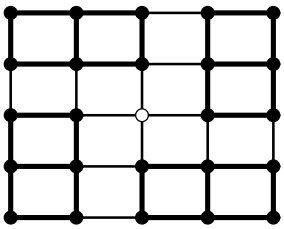


Figure 2.9 Divide into 2×3 rectangles and center

region IV. See Figure 2.10. If we choose the point 1 or 2 instead of a, then we have the 2×2 square path that contains a and the cornerpoint 5 of the 5×5 lattice. If we choose 3 or 5 instead of a, then we have a 2×2 square path between regions I and IV. If we choose point 4, then we have the 2×2 lattice in IV. But, notice that the 5×5 square path is not covered, a contradiction.

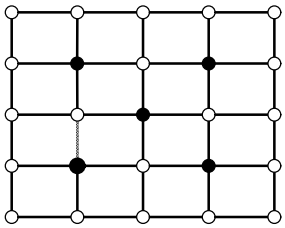


Figure 2.10 Center selected

Case 2: Center is not a black point.

If the center is not black, then there is a region that contains two black points, say region I. Using the same strategy from Case 1 we can choose the points in regions II,

III, and IV. See Figure 2.11. Regardless of how the two points are selected in region I, there are two 3×3 square paths that contain the center in regions II and III, a contradiction.

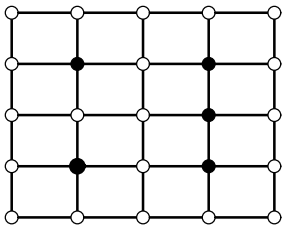


Figure 2.11 center point not selected

Thus, $M(5) > 5$. In fact, Figure 2.12 shows that $M(5) \leq 6$.

□

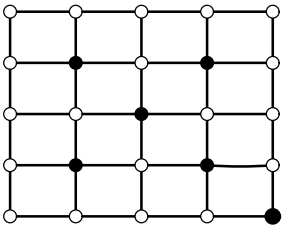


Figure 2.12 $M(5) = 6$

Theorem 2.5 $M(6) = 9$

Proof Since the 6×6 lattice contains 9 disjoint 2×2 lattices, then $M(6) \geq 9$. In fact, Figure 2.13 shows that $M(6) \leq 9$. □

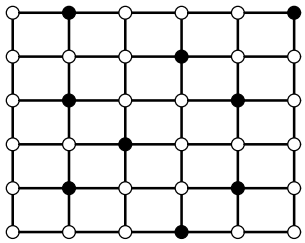


Figure 2.13 $M(6) = 9$

Theorem 2.6 $M(7) = 12$.

Proof Divide the 7×7 lattice into four 3×4 rectangles such that the center point of the lattice is not included in any 3×4 lattice. Recall that $M(3,4) = M(4,3) = 2$, and the solution is unique.

Suppose our 7×7 lattice has exactly 11 black points such that no square path exists.

Case 1 Center point is a black point.

If the center point is a black point, then the remaining 10 points are in each of the 3×4 lattices.

Case 1.1 One 3×4 lattice has 4 black points.

WLOG assume that II has 4 black points. The remaining six black points are in the three 3×4 lattices each containing two black points. Then there exists a 2×2 square path between III and IV regardless of how the four black points are arranged in II, a contradiction. See Figure 2.14.

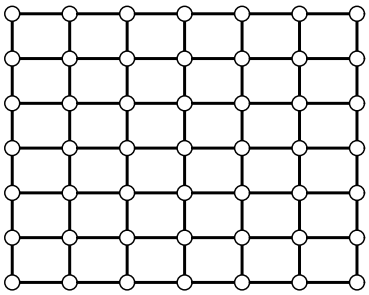


Figure 2.14 11 black points for $M(7)$

Case 1.2 Two 3×4 lattices have 3 black points.

Case 1.2.a

If the two 3×4 lattices that contain 3 black points are I and II (or any two 3×4 lattices that are not horizontal), then a 2×2 square path exists regardless of how the 3 black points in each of these two 3×4 lattices are arranged, a contradiction. See Figure 2.15.

Case 1.2.b

Suppose the two 3×4 lattices that contain 3 black points are in II and IV. WLOG consider region II. The black points in III and I are fixed according to Lemma 2.2. See Figure 2.16.

Notice that the 2×2 lattices labeled 1 and 5 and the 3×3 lattice labeled 8 are disjoint, and so at least $1+1+2 = 4$ points are required to cover them, a contradiction.

Case 2 Center point is not a black point.

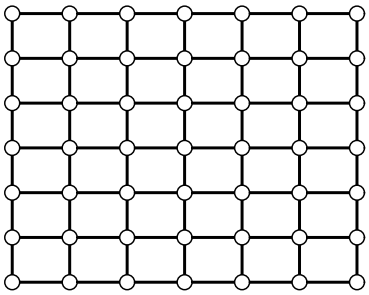


Figure 2.15 11 black points for $M(7)$

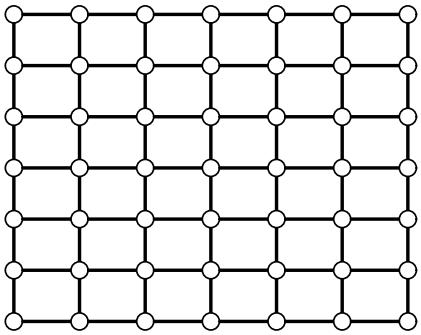


Figure 2.16 11 black points for $M(7)$

Case 2.1 One 3×4 contains 5 black points.

The result is similar to Case 1.1 which contains a square path, a contradiction. See Figure 2.17.

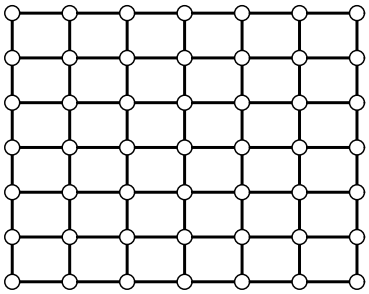


Figure 2.17 11 black points for $M(7)$

Case 2.2 One 3×4 contains 4 black points and another has 3 black points.

Case 2.2.a

If the lattices are not diagonal, for example quadrants II, III, then WLOG assume III has 4 black points and II has 3 black points. Then regardless of how these points in II and III are chosen, a square path exists between I and IV, a contradiction. See Figure 2.18.

Case 2.2.b

If the lattices are diagonal, for example regions II, IV, then WLOG assume II has 4 black points and IV has 3 black points. We note as in Case 1.2.b, that we must eliminate the 2×2 lattices labeled 4 and 5 and the 3×3 lattice labeled 8 with only 3 points. This is impossible, because they are disjoint. See Figure 2.19.

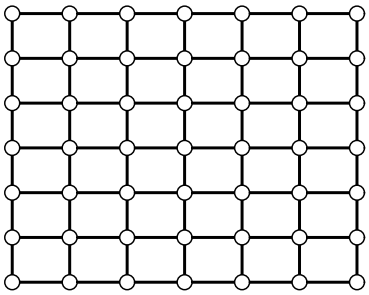


Figure 2.18 11 black points for $M(7)$

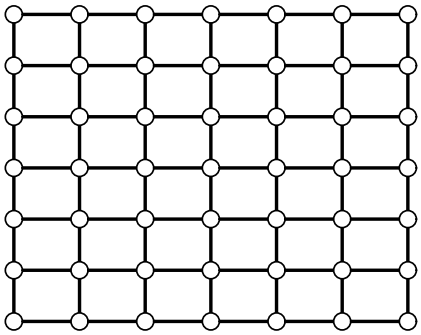


Figure 2.19 11 black points for $M(7)$

Case 2.3 Three 3×4 rectangles contain 3 black points.

Assume WLOG that I,II, and III contain 3 black points and IV contains 2 black points. Since the center is not included and IV contains 2 fixed points, then we can extend III to a 4×4 lattice. We know a 4×4 lattice requires at least 4 black points. But, III is only allowed 3 black points. Thus, a square path exists, a contradiction. See Figure 2.20.

Thus, $M(7) \geq 12$. But, we can in fact show that $M(7) \leq 12$. See Figure 2.21. Thus, $M(7) = 12$. □

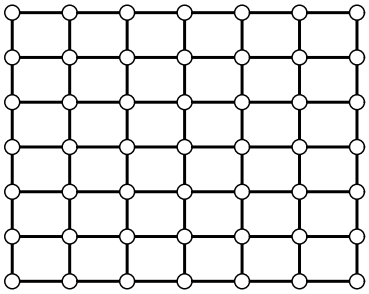


Figure 2.20 11 black points for $M(7)$

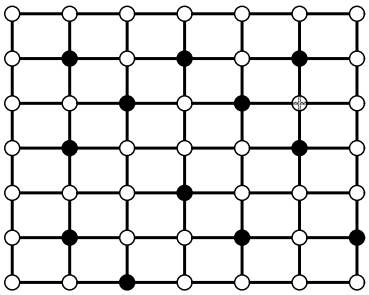


Figure 2.21 $M(7) = 12$

Theorem 2.7 $M(8) = 16$

Proof Since the 8×8 lattice contains 16 disjoint 2×2 lattices, then $M(8) \geq 16$. In fact, Figure 2.22 shows that $M(8) \leq 16$. □

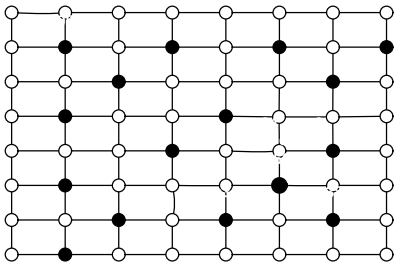


Figure 2.22 $M(8) = 16$

Theorem 2.8 $M(9) = 20$

Proof Suppose 19 black points is enough to cover a 9×9 lattice. We can divide the lattice into the center and four regions of size 4×5. Since $M(4,5) = 4$, then each of these regions contains at least four black points. Table 2.1 lists the five possibilities:

Table 2.1 Possibilities for 9×9 lattice

case	center	I	II	III	IV
1.1	1	4	4	4	6
1.2a	1	4	5	4	5
1.2b	1	5	5	4	4
2.1	0	4	4	4	7
2.2a	0	4	4	5	6
2.2b	0	4	5	4	6
2.3	0	4	5	5	5

Case 1 Center point is a black point.

Case 1.1 Assume region IV contains six black points. Recall that the configuration for $M(4,5)$ is unique from Lemma 2.4. Regardless of how the six nodes are placed, we will have a 2×2 square path between regions that contain only four black points as indicated by Figure 2.23.

Case 1.2 Have 5,5,4,4 black points in the regions

Case 1.2.a The regions that contain five black points are on diagonal, say II and IV. Since the configuration of black points in regions I and III are fixed by Lemma 2.4 to contain no points from their perimeters and the four 2×2 lattices indicated are disjoint, at least four black points of IV are needed to eliminate these square paths. This leaves only one black point to eliminate all square paths in the remaining 4×3 of IV which is impossible. See Figure 2.24.

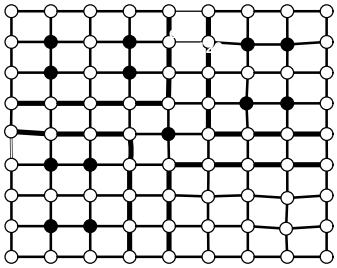


Figure 2.23 Case 1.1 of Proof of Theorem 2.8

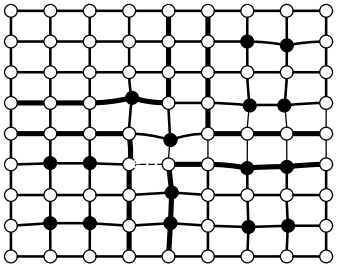


Figure 2.24 Case 1.2.a of Proof of Theorem 2.8

Case 1.2.b The regions that contain five black points are not on diagonal. This case breaks down like Case 1.1 since will have a 2×2 square path between regions that only contain four black points.

Case 2 Center is not a black point

Case 2.1 This case is also similar to Case 1.1.

Case 2.2 We have 4,4,5,6 black points in the regions.

Case 2.2.a The regions that contain four black points are not diagonal

This is also similar to Case 1.1.

Case 2.2.b The regions that contain four black points are diagonal, say I and III.

Since the center is not black and the selection of black points in I and III is fixed (Lemma 2.4), region IV can be extended to a 5×6 lattice without adding more black points, i.e., $M(5,6) \leq 6$ contradicting Lemma 2.7. See Figure 2.25.

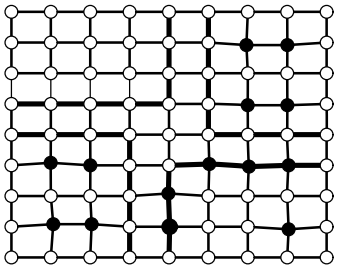


Figure 2.25 Case 2.2.b of Proof of Theorem 2.8

Case 2.3 We have 4,5,5,5 black points in the regions.

As in Case 2.2.b, region IV can be extended to a 5×5 without adding more black points. Hence, $M(5) \leq 5$, contradicting Theorem 2.4. See Figure 2.26.

Thus, $M(9) \geq 20$. In fact, Figure 2.27 gives an optimal configuration that uses exactly 20 black points. Thus, $M(9) = 20$. □

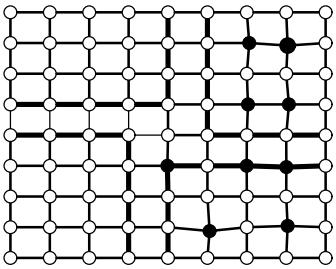


Figure 2.26 Case 2.3 of Proof of Theorem 2.8

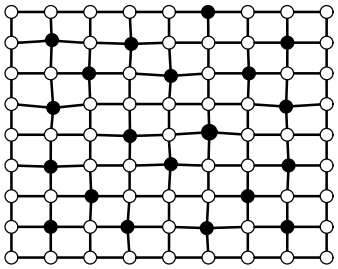


Figure 2.27 $M(9) = 20$ of Proof of Theorem 2.8

Chapter 3

Binary Integer Programming Formulation

The SPP can be modeled as a {0,1}-integer programming problem or a binary integer programming (BIP) problem. In the formulation we assign a variable to every point in the square lattice.

Let L be an n$\times$ n lattice. For each point $x_i \in$ L define

$$x_i = \begin{cases} 1 & \text{if } i \text{ is a black point,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

where $i = 1, ..., n^2$

The variables are indexed over the total number of vertices in the lattice. The points are assigned the value 1 if they are black and 0 if not black. We want to minimize the total number of black points in the square lattice subject to the constraint that every square path contains a black point. Thus, the SPP can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_i x_i \\ \text{subject to} \quad & \textstyle\sum_{i \in S} x_i \geq 1, for\ each\ 2 \times 2\ square\ S \\ & \textstyle\sum_{i \in S} x_i \geq 1, for\ each\ 3 \times 3\ square\ S \\ & \qquad \vdots \\ & \textstyle\sum_{i \in S} x_i \geq 1, for\ each\ n \times n\ square\ S \\ & x_i = \{0, 1\} \end{aligned} \tag{3.2}$$

These constraints require at least one black point on every square path, and the last constraint forces the variables to be binary. CPLEX version 6.0.1 was employed to

solve the BIP formulation and obtain values for $M(n)$. The following result describes the growth of the formulation which indicates how efficiently values for $M(n)$ were obtained from the BIP formulation.

Theorem 3.1 As n increases, the number of inequalities in the BIP formulation grows cubically.

Proof There are

$(n-1)^2$ 2×2 equations, for $n \geq 2$

$(n-2)^2$ 3×3 equations, for $n \geq 3$

$(n-3)^2$ 4×4 equations, for $n \geq 4$

$\vdots$

1 $n \times n$ equation generated

and taking the sum gives

$$(n-1)^2 + (n-2)^2 + \ldots + 1^2 = \sum_{i=1}^{n-1} i^2 = \frac{n(n-1)(2n-1)}{6}$$

$\Rightarrow O(n^3)$ growth. □

Since the rapid growth prevented an efficient solution, there was a need to add more constraints to the BIP so that CPLEX considers a minimum feasible region. This led to the following results.

Theorem 3.2 There exists an optimal solution that contains no corner point.

Proof Suppose we have an optimal solution with one corner point. Then removal of this corner point yields a square path. We can replace this point by an adjacent

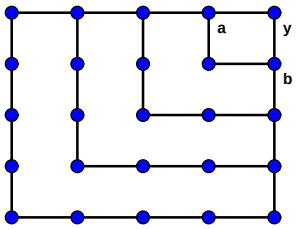


Figure 3.1 no corner point

point, say either a or b, and still obtain an optimal solution, a contradiction. See Figure 3.1.

□

This result can be formulated as $x_1 + x_n + x_{n^2-n+1} + x_{n^2} = 0$ which is equivalent to saying no corners are black.

Theorem 3.3 There exists an optimal solution that contains no boundary contains two consecutive black points.

Proof Suppose we have two consecutive black points on a boundary. We can force one of these to be white by making its perpendicular neighbor black and still cover the same square paths. Figure 3.2 becomes Figure 3.3. □

3.1 Unproven Claims for $M(14)$

The following conjectures were used to add more constraints to the BIP for the 14×14 case.

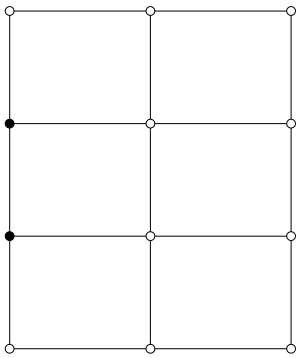


Figure 3.2 2 consecutive black points on boundary

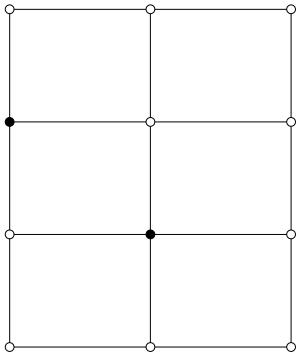


Figure 3.3 No consecutive black points on boundary

Conjecture 1 There does not exist an optimal solution that contains a black point on each of the four distinct boundaries.

Since the 14×14 square must be covered and if Conjecture 1 holds, then the optimal configuration for the 14×14 case has either 1, 2, or 3 points on the boundary.

Conjecture 2 There exists an optimal solution that contains no black points on one of the boundaries.

Argument for Conjecture 2 Suppose there exists an optimal solution with 3 boundaries containing exactly one black point. Consider the 12×12 region which requires 38 black points. WLOG, let 3 of these black points share the boundary with the 3 black points from the 14×14 lattice such that there is no overlap in squares covered thus allowing a possible optimal solution. See Figure 3.4. Then we need at least 53 black points since the 12×12 region requires 38, the 3×5 region and the 3×7 region requires 3 and 4 black points respectively. Also, the seven 2×3 regions require 1 each and the two 2×2 regions may be in a 2×3 form so only require at least one black point. But, the configuration in Figure 3.5 requires 52 black points which is one less black point than the $38 + 3 + 4 + 7 + 1 = 53$, a contradiction.

End of Argument

If Conjecture 2 holds, then it can be formulated for the BIP as $x_1 + x_2 + \ldots + x_n = 0$, i.e., first row of the lattice has no black points.

Remark Since the lattice can be rotated 90,180, or 270 degrees any boundary may be considered the first row of the lattice.

Conjecture 3 There exists an optimal solution such that two nonadjacent boundaries have no black points.

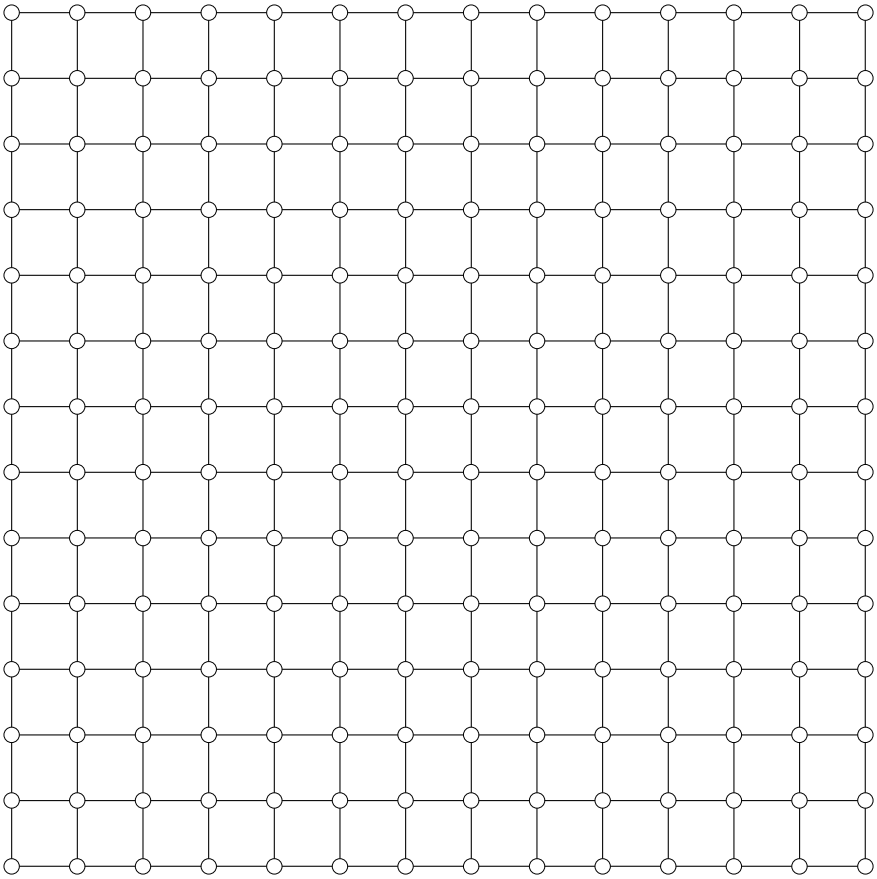


Figure 3.4 3 black points on the boundary

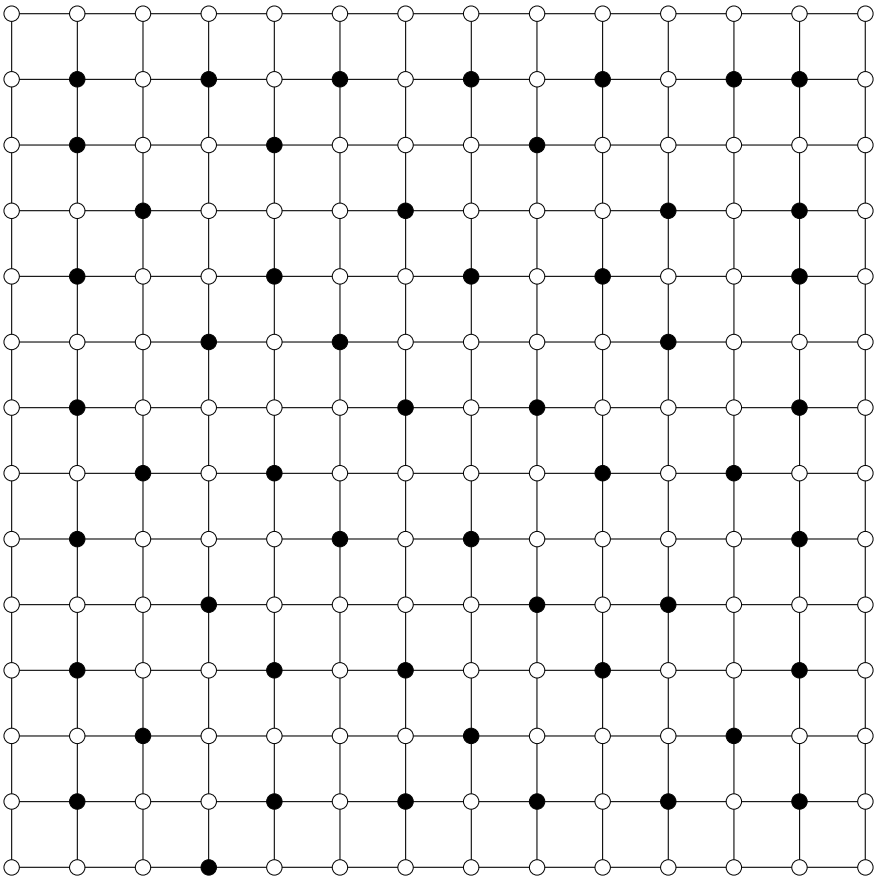


Figure 3.5 $M(14) = 52$

Argument for Conjecture 3 Moving the black point in row 2, column 2 in Figure 3.5 up to the top boundary gives us a configuration that requires 52 black points. Moving this black point up still covers the same square paths. This solution has two black points on the boundary that are on nonadjacent boundaries.

End of Argument

If Conjectures 2 and 3 hold, then clearly Conjecture 4 follows.

Conjecture 4 There exists an optimal solution such that three of the four boundaries do not contain a black point.

The following conjecture holds if all of the above are true.

Conjecture 5 There exists an optimal solution such that the points in positions $n+2$, $2n-1$, n^2-n-1 are black points.

Argument for Conjecture 5 Since there exist an optimal solution such that three of the four boundaries do not contain any black points, then it follows that the points labeled a and b must be black since the boundaries adjacent to the corner nodes do not contain a black point. See Figure 3.6. This forces either c or e and d or f to be black points. Since there is an optimal solution that contains a black point on the outer $n \times n$ boundary, then we can force say e to be a black point. Thus, this requires d to be a black.

End of Argument

The new constraints contributed significantly in the improvement for the time to obtain $M(n)$. For n=12, the original BIP in (wherever it is located) required 3111.67 seconds while the new BIP took only 1157.18 seconds *. With the additional constraints the final BIP required only 182.53 seconds. These computational

*All computations were obtained from a Sun Sparc Ultra 30 Model with 256M memory.

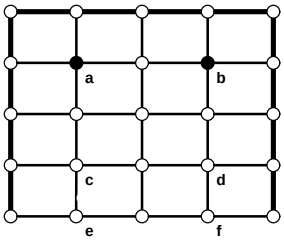


Figure 3.6 No black points on one boundary.

results quantify how the final mathematical formulation is significantly better than the original. All computed values for $M(n)$ can be found in the Results chapter.

Chapter 4

A Similar Problem

One related problem to the SPP is the problem that seeks to find the minimum number of vertices to remove from a planar graph G such that no circuits exist. Dean refers to this minimum value as $\tau(G)$ in general and $\tau(n)$ for the SPP. The graphs are planar in both of these problems. In relation to to the Square Path problem discussed in this work, $\tau(n)$ provides an upper bound for $M(n)$. Bienstock-Dean consider covering points of a planar graph with a minimum number of faces. The Erdös-Pósa theorem on independent circuits in graphs can be applied when we consider graphs with a specific embedding.

Several upper bounds for $\tau(n)$ for n up to 14 are included in the Results chapter.

The following is a conjecture about the bounds of $\tau(n)$.

Conjecture 1

$$\frac{1}{3}(n-1)^2 \leq \tau(n) \leq \frac{1}{3}n^2$$

The following figures show an example of $\tau(n)$ where n=4. Figure 4.2 illustrates the new lattice after the black points indicated in Figure 4.1 have been removed.

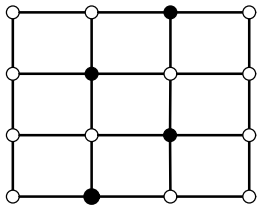


Figure 4.1 $\tau(4) = 4$

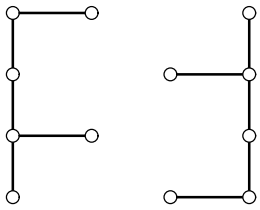


Figure 4.2 No circuits in $\tau(4) = 4$

Chapter 5

Results

Table 5.1 indicates proven results for the non-square lattice.

Table 5.1 Results for $M(a, b)$

$M(2,3) = M(3,2) = 1$
$M(3,4) = M(4,3) = 2$
$M(3,5) = M(5,3) = 3$
$M(5,4) = M(4,5) = 4$
$M(3,7) = M(7,3) = 4$
$M(4,7) = M(7,4) = 6$
$M(5,6) = M(6,5) = 7$

Table 5.2 indicates proven and computed values for $M(n)$ and $\tau(n)$. Note that an underlined value provides an upper bound and a "*" indicates values computed via CPLEX and not proven theoretically. Also note that "**" indicates that constraints that were not proven theoretically were added to the BIP to obtain the indicated solution.

Table 5.2 Results for both $M(n)$ and $\tau(n)$

n	2	3	4	5	6	7	8	9	10	11	12	13	14
$M(n)$	1	2	4	6	9	12	16	20	26*	31*	38*	44*	52**
$\tau(n)$	1	2	4	6	<u>10</u>	<u>13</u>	<u>19</u>	<u>24</u>	<u>32</u>	<u>38</u>	<u>47</u>	<u>56</u>	<u>64</u>

5.1 Computational Results

The table below will include the amount of time it took to solve the BIP with the original formulation and also with the new formulation which includes the new constraints mentioned in chapter 3. The computational results were obtained from a Sun Sparc Ultra 30 Model with 256M memory using CPLEX version 6.0.1 with the exception of the new results for $n = 12, 13, 14$. These results were run on four processors and used an unpublished version of CPLEX and are indicated by * in the table. We might expect the case for $n = 15$ to take at least 2 days if run on multiple processors. Otherwise, it may take about six days since the lack of memory forced us to abort the run after 175057.52 sec which is roughly 48.63 hours. The upper and lower bounds for this problem at the time of abortion was 62 and 58, respectively. Storing only necessary information in CPLEX should eventually lead us to the solution for $n = 15$.

Table 5.3 Computational Results

n	$M(n)$	Old BIP (sec)	New BIP (sec)
3	2	0.00	0.0
4	4	0.02	0.0
5	6	0.02	0.0
6	9	0.02	0.0
7	12	0.13	0.7
8	16	0.15	0.11
9	20	3.78	2.93
10	26	40.92	17.45
11	31	568.42	313.87
12	38	3111.67	182.53*
13	44	--	19506.30*
14	52	--	26206.80**

We want to stress the importance of having more constraints to define the feasible region of the BIP. As mentioned earlier, adding the new constraints decreases the running time. Finding more constraints should direct us to solutions for larger n.

5.2 Configurations for Larger n

The following figures show configurations obtained by CPLEX.

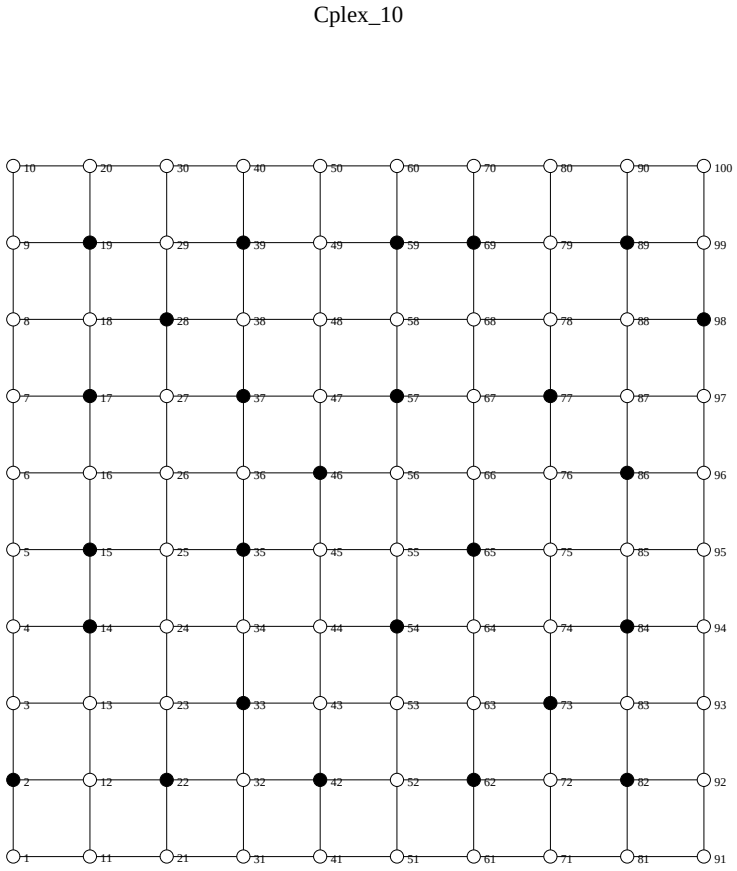


Figure 5.1 $M(10)$

Cplex_11

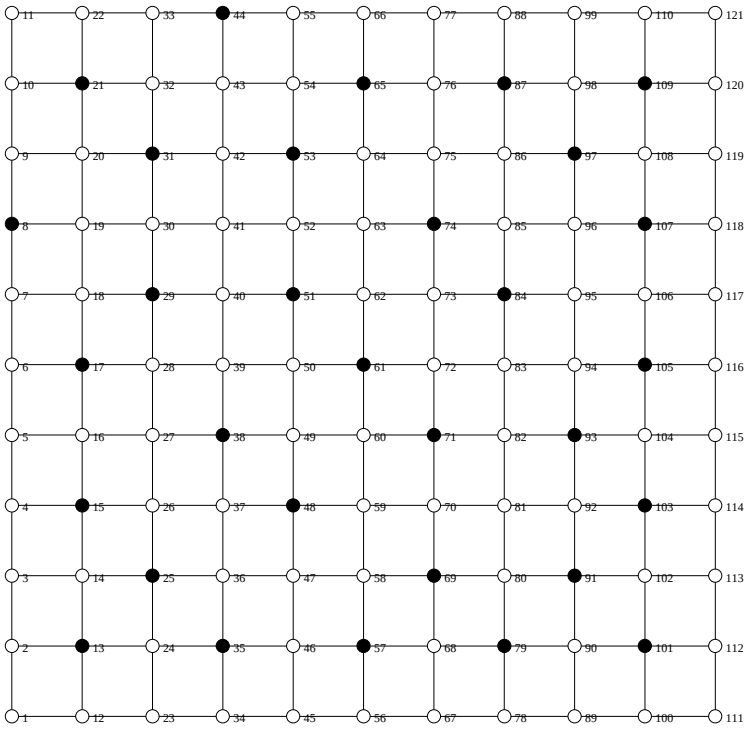


Figure 5.2 $M(11)$

Cplex_12

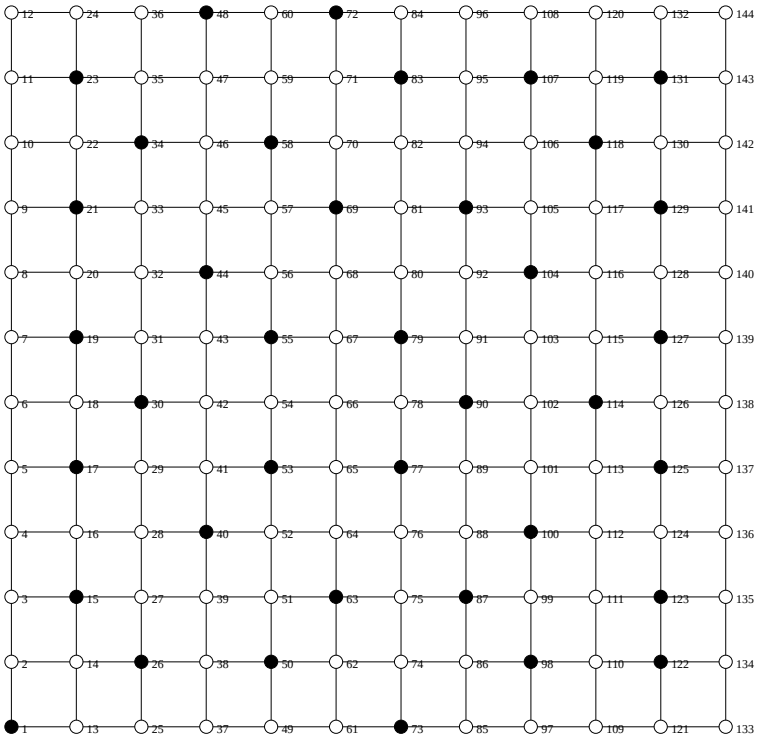


Figure 5.3 $M(12)$

Cplex_13

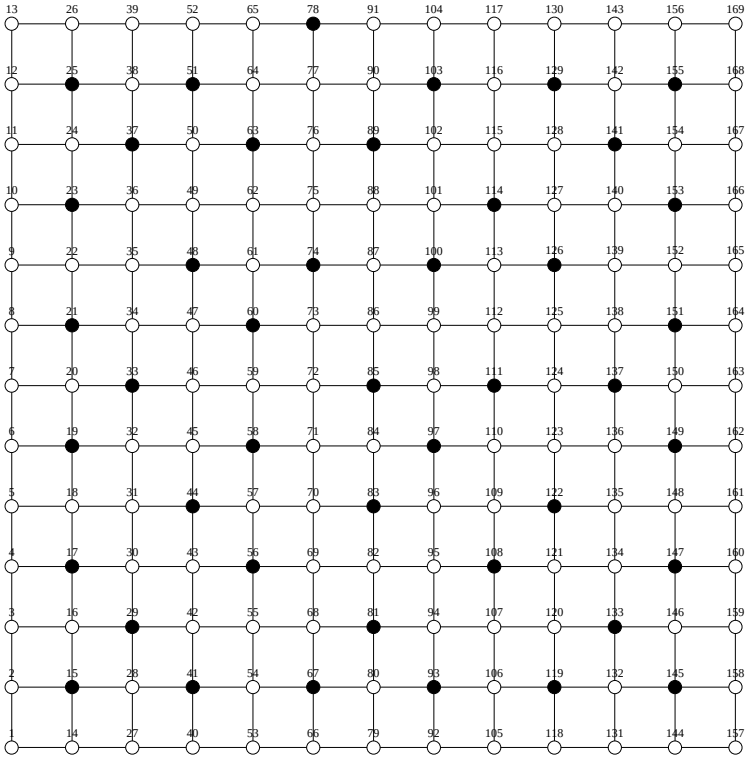


Figure 5.4 $M(13)$

5.3 Closed Form Attempt

In an attempt to obtain insight for a closed form expression or formula for $M(n)$, the computed values from the CPLEX solution of the BIP formulation were fed into Sloane's On-Line Encyclopedia of Integer Sequences [6], but no formula was found. However, there exists a formula for even values of n due to Kimberling [6]. Kimberling describes this sequence as the index of 5^n within the sequence of numbers of the form $2^i 5^j$. For example, the first nine terms of this sequence are $\underline{1}$, 2, 4, $\underline{5}$, 8, 10, 16, 20, $\underline{25}$ and the underlined terms are the first, fourth and ninth terms of the sequence. These indices are indeed the values of $M(n)$ for $n \geq 2$ with n even. Even though this provides more information for a formula for the SPP, a general formula for any n is still desired.

5.4 Future Work

Adding more constraints to the BIP as well as taking advantage of symmetry should aid in providing a a formula for $M(n)$ efficiently. Also passing known bounds to CPLEX for $M(n)$ and using tricks in CPLEX should decrease the running time compared to the time for n=14. Proving more values for $M(n)$ should eventually help in obtaining a general formula for $M(n)$. This work can be investigated further.

Bibliography

[1] D. Bienstock and N. Dean, On Obstructions to Small Face Covers in Planar Graphs, *Journal of Combinatorial Theory, Series B* **55** (1992), 163-189.

[2] D. Erickson, "Solution 1296", *Mathematics Magazine*, **62** (1988), 142.

[3] P. Erdös and L. Pósa, On Independent Circuits Contained in a Graph, *Canadian Journal of Mathematics* **17** (1965), 347-352.

[4] L. Larson, personal communication, 1999.

[5] H.C. Morris, "Proposal 1296", *Mathematics Magazine*, **61** (1988), 11 5.

[6] `http://akpublic.research.att.com/~njas/sequences/index.html`

[7] P. Zorn, personal communication, 1999.

Enumération des 2-arbres k-gonaux

Gilbert Labelle, Cédric Lamathe, Pierre Leroux

RÉSUMÉ : *Dans ce travail*[1]*, nous généralisons les 2-arbres en remplaçant les triangles par des quadrilatères, des pentagones ou des polygones à k côtés (k-gones), où $k \geq 3$ est fixe. Cette généralisation, aux 2-arbres k-gonaux, est naturelle et est étroitement liée dans le cas planaire aux arbres cellulaires. Notre objectif est le dénombrement, étiqueté et non étiqueté, des 2-arbres k-gonaux selon le nombre n de k-gones. Nous donnons des formules explicites dans le cas étiqueté, et, dans le cas non étiqueté, des formules de récurrence et des formules asymptotiques.*

ABSTRACT: *In this paper*[1]*, we generalize 2-trees by replacing triangles by quadrilaterals, pentagons or k-sided polygons (k-gons), where $k \geq 3$ is given. This generalization, to k-gonal 2-trees, is natural and is closely related, in the planar case, to some specializations of the cell-growth problem. Our goal is the enumeration, labelled and unlabelled, of k-gonal 2-trees according to the number n of k-gons. We give explicit formulas in the labelled case, and, in the unlabelled case, recursive and asymptotic formulas.*

1 Introduction

L'espèce des arbres bidimensionnels, ou 2-arbres, a été bien étudiée dans la littérature. Voir par exemple [4] et [2, 3]. Essentiellement, un 2-arbre est un graphe simple connexe constitué de triangles qui sont liés entre eux par les arêtes de manière arborescente, c'est-à-dire sans former de cycles (de triangles). Dans [5], Harary et al. ont énuméré une variante des arbres cellulaires (relié au "cell-growth problem"), à savoir des 2-arbres k-gonaux plans et planaires[2], dans lesquels les triangles ont été remplacés par des quadrilatères, des pentagones ou des polygones à k côtés (k-gones), où $k \geq 3$ est fixe. De tels 2-arbres, bâtis sur des k-gones, sont appelés 2-arbres k-gonaux. Cette généralisation apparaît naturellement et le but de ce travail est l'énumération des 2-arbres k-gonaux libres, c'est-à-dire vus comme graphes simples, sans question de planarité. La figure 1 a) propose un exemple de 2-arbres k-gonal, dans le cas où $k = 4$.

Nous disons qu'un 2-arbre k-gonal est *orienté* si ses arêtes sont orientées de façon telle que chaque k-gone forme un cycle orienté, voir la figure 1 b). Notons par $\mathcal{A}$ et par $\mathcal{A}_o$ les espèces des 2-arbres k-gonaux et des 2-arbres k-gonaux orientés respectivement. Pour ces deux espèces, nous utilisons les symboles $-$, $\diamond$ et $\underline{\diamond}$ en exposant pour indiquer que les structures ont été pointées en une arête, en un polygone, et en un polygone muni d'une arête distinguée, respectivement.

Notre objectif est le dénombrement, étiqueté et non étiqueté, des 2-arbres k-gonaux selon le nombre n de k-gones. Nous donnons des formules explicites dans le cas étiqueté, et dans le cas non étiqueté, des formules de récurrence et des formules asymptotiques. Pour cela, nous adaptons l'approche de Fowler et al. dans [2, 3] qui correspond au cas $k = 3$. En particulier, les 2-arbres sont étiquetés aux k-gones.

[1] Avec l'appui du FCAR (Québec) et du CRSNG (Canada)

[2] Au sens où toutes les faces, à part la face externe, sont des k-gones

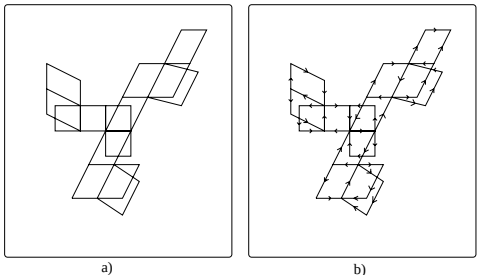


Figure 1: Un 2-arbre 4-gonal non orienté et orienté

La principale difficulté à cette extension vient, comme on le verra, du cas où k est pair.

Les deux premières étapes sont assez directes. Il s'agit d'étendre le théorème de dissymétrie au cas k-gonal et de caractériser l'espèce $B = \mathcal{A}^{\rightarrow}$ des 2-arbres k-gonaux munis d'une arête distinguée et orientée, à l'aide d'une équation fonctionnelle de type lagrangien. Le premier résultat est une extension immédiate du cas $k = 3$ et la démonstration est omise.

Théorème 1.1. THÉORÈME DE DISSYMÉTRIE. *Les espèces $\mathcal{A}$ et $\mathcal{A}_o$ des 2-arbres k-gonaux orientés et non orientés respectivement satisfont les isomorphismes d'espèces suivants :*

$$\mathcal{A}_o^{-} + \mathcal{A}_o^{\diamond} = \mathcal{A}_o + \mathcal{A}_o^{\underline{\diamond}}, \tag{1}$$

$$\mathcal{A}^{-} + \mathcal{A}^{\diamond} = \mathcal{A} + \mathcal{A}^{\underline{\diamond}}. \tag{2}$$

Dans la prochaine section, nous caractérisons l'espèce $B = \mathcal{A}^{\rightarrow}$ et nous en donnons ses propriétés. Par la suite, nous exprimons les diverses espèces pointées qui apparaissent dans le théorème de dissymétrie en fonction de l'espèce B et nous en déduisons les résultats énumératifs désirés pour les espèces $\mathcal{A}_o$ et $\mathcal{A}$. Le cas orienté, plus simple, est traité d'abord, dans la section 3. Le cas non orienté, suit, dans la section 4, en distinguant les deux cas de parité de k, pour le dénombrement non étiqueté. Enfin, les résultats asymptotiques sont présentés dans la section 5.

2 L'espèce $B = \mathcal{A}^{\rightarrow}$

L'espèce $B = \mathcal{A}^{\rightarrow}$ joue un rôle fondamental dans l'étude des 2-arbres k-gonaux.

Théorème 2.1. *L'espèce $B = \mathcal{A}^{\rightarrow}$ des 2-arbres k-gonaux pointés en une arête orientée satisfait l'équation (isomorphisme) fonctionnelle suivante :*

$$B = E(XB^{k-1}), \tag{3}$$

où E représente l'espèce des ensembles.

Preuve. On décompose une $\mathcal{A}^{\rightarrow}$-structure en un ensemble de *pages*, c'est-à-dire en sous-graphes maximaux qui partagent un seul k-gone avec l'arête distinguée. Pour chaque page, l'orientation de l'arête pointée permet alors de définir un ordre et une orientation sur les $k-1$ arêtes restantes du polygone possédant cette arête, selon la figure 2 a) pour le cas impair, et b) pour le cas pair. Ces arêtes étant orientées, on peut alors y accrocher des B-structures. On en déduit alors l'équation (3). ■

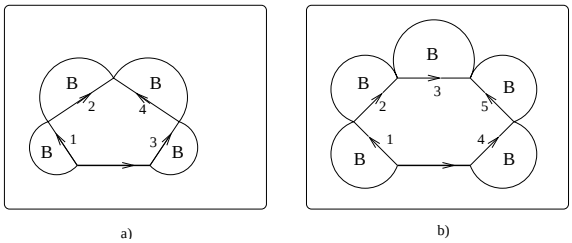


Figure 2: Une page orientée a) $k = 5$ b) $k = 6$

On peut relier simplement l'espèce $B = \mathcal{A}^{\rightarrow}$ à celle des arborescences (arbres enracinés), A, caractérisée par l'équation fonctionnelle $A = XE(A)$, où X est ici l'espèce des sommets. En effet de (3), on déduit successivement

$$(k-1)XB^{k-1} = (k-1)XE((k-1)XB^{k-1}), \tag{4}$$

sachant que $E^m(X) = E(mX)$, et, par unicité,

$$(k-1)XB^{k-1} = A((k-1)X). \tag{5}$$

Finalement, on obtient l'expression suivante pour l'espèce B en fonction de l'espèce des arborescences :

Proposition 2.2. *L'espèce $B = \mathcal{A}^{\rightarrow}$ des 2-arbres k-gonaux pointés en une arête orientée vérifie*

$$B = \sqrt[k-1]{\frac{A((k-1)X)}{(k-1)X}}. \tag{6}$$

Proposition 2.3. *Les nombres $a_n^{\rightarrow}$, $a_{n_1,n_2,\ldots}^{\rightarrow}$, et $b_n = \widetilde{a}_n^{\rightarrow}$ de 2-arbres k-gonaux pointés en une arête orientée et ayant n k-gones, respectivement étiquetés, laissés fixes par une permutation de $\mathbb{S}_n$ de type cyclique $1^{n_1}2^{n_2}\ldots$, et non étiquetés, satisfont les relations suivantes :*

$$a_n^{\rightarrow} = ((k-1)n+1)^{n-1} = m^{n-1}, \tag{7}$$

où $m = (k-1)n+1$ est le nombre d'arêtes,

$$a_{n_1,n_2,\ldots}^{\rightarrow} = \prod_{i=1}^{\infty}(1+(k-1)\sum_{d|i} dn_d)^{n_i-1}(1+(k-1)\sum_{\substack{d|i\\ d<i}} dn_d), \tag{8}$$

et

$$b_n = \frac{1}{n}\sum_{1\le j\le n}\sum_{\alpha}(|\alpha|+1)b_{\alpha_1}b_{\alpha_2}\ldots b_{\alpha_{k-1}}b_{n-j}, \qquad b_0 = 1, \tag{9}$$

la deuxième somme étant prise sur les $(k-1)$-uplets d'entiers $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{k-1})$ tels que $|\alpha|+1$ divise l'entier j, où $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_{k-1}$.

Preuve. Les formules (7) et (8) s'obtiennent en spécialisant avec $\mu = (k-1)^{-1}$ les formules suivantes, données par Fowler et al. dans [2, 3],

$$\left(\frac{A(x)}{x}\right)^{\mu} = \sum_{n\geq 0} \mu(\mu+n)^{n-1}\frac{x^n}{n!}, \tag{10}$$

$Z_{\left(\frac{A(X/\mu)}{X/\mu}\right)^{\mu}} =$

$$\sum_{n_1,n_2,\ldots} \frac{x_1^{n_1}x_2^{n_2}\ldots}{1^{n_1}n_1!2^{n_2}n_2!\ldots}\prod_{i=1}^{\infty}(1+\frac{1}{\mu}\sum_{d|i} dn_d)^{n_i-1}(1+\frac{1}{\mu}\sum_{d|i,d<i} dn_d). \tag{11}$$

La formule (7) peut également se voir directement par une adaptation de la bijection de Prüfer. Pour obtenir la récurrence (9), il suffit de prendre la dérivée logarithmique de l'équation

$$\widetilde{B}(x) = \exp\left(\sum_{i\geq 1}\frac{x^i\widetilde{B}^{k-1}(x^i)}{i}\right), \tag{12}$$

où $\widetilde{B}(x) = \sum_{n\geq 0} b_n x^n$, qui découle de la relation (3). ■

La suite des nombres $\{b_n\}$, pour $k = 2, 3, 4, 5$, est répertoriée dans l'encyclopédie des suites d'entiers [11] et l'équation (3), dans l'encyclopédie des structures combinatoires [6]. Le comportement asymptotique des nombres b_n est analysé, notamment en fonction de k, dans la section 5.

3 Cas orienté

Commençons par déterminer les espèces pointées qui apparaissent dans le théorème de dissymétrie. Ces relations sont assez immédiates et la démonstration est laissée au lecteur.

Proposition 3.1. *Les espèces $\mathcal{A}_o^-$, $\mathcal{A}_o^\diamond$, et $\mathcal{A}_o^{\underline{\diamond}}$ sont caractérisées par les isomorphismes suivants*

$$\mathcal{A}_o^- = B, \quad \mathcal{A}_o^\diamond = XC_k(B), \quad \mathcal{A}_o^{\underline{\diamond}} = XB^k, \tag{13}$$

où $B = \mathcal{A}^{\rightarrow}$ et C_k représente l'espèce des cycles (orientés) de longueur k.

Le théorème de dissymétrie permet d'exprimer la série génératrice ordinaire $\widetilde{a}_o(x)$ des 2-arbres k-gonaux orientés non étiquetés, en termes des espèces pointées,

$$\widetilde{a}_o(x) = \widetilde{a}_o^-(x) + \widetilde{a}_o^\diamond(x) - \widetilde{a}_o^{\underline{\diamond}}(x), \tag{14}$$

et par la proposition 3.1, nous pouvons alors exprimer $\widetilde{a}_o(x)$ en fonction de $\widetilde{B}(x) = \widetilde{a}^{\rightarrow}(x)$.

Proposition 3.2. *La série génératrice ordinaire $\widetilde{a}_o(x)$ de l'espèce des 2-arbres k-gonaux orientés non étiquetés est donnée par l'expression*

$$\widetilde{a}_o(x) = \widetilde{B}(x) + \frac{x}{k}\sum_{\substack{d|k\\ d>1}} \phi(d)\widetilde{B}^{\frac{k}{d}}(x^d) - \frac{k-1}{k}x\widetilde{B}^k(x). \tag{15}$$

Corollaire 3.3. *Les nombres $a_{o,n}$ et $\widetilde{a}_{o,n}$ de 2-arbres k-gonaux orientés étiquetés et non étiquetés, sur n k-gones sont donnés par*

$$a_{o,n} = ((k-1)n+1)^{n-2} = m^{n-2}, \quad n \geq 2, \tag{16}$$

$$\widetilde{a}_{o,n} = b_n - \frac{k-1}{k}b_{n-1}^{(k)} + \frac{1}{k}\sum_{\substack{d|k\\ d>1}} \phi(d) b_{\frac{n-1}{d}}^{(\frac{k}{d})}, \tag{17}$$

où $b_i^{(j)} = \displaystyle\sum_{i_1+\cdots+i_j=i} b_{i_1}b_{i_2}\ldots b_{i_j}$, représente le coefficient de x^i dans la série $\widetilde{B}^j(x)$, avec $b_r^{(j)} = 0$ si r est non entier ou négatif.

Preuve. Pour le cas étiqueté, il suffit de remarquer que $a_n^{\rightarrow} = ma_{o,n}$. Dans le cas non étiqueté, l'équation (17) s'obtient directement de (15). ■

4 Cas non orienté

Dans le cas non orienté, le nombre a_n de 2-arbres k-gonaux étiquetés sur n polygones satisfait $2a_n = a_{o,n} + 1$, puisque le seul 2-arbre k-gonal orienté étiqueté laissé fixe par changement d'orientation pour un nombre de polygones donné, est celui dont les polygones partagent tous une arête commune. On obtient

Proposition 4.1. *Le nombre a_n de 2-arbres k-gonaux étiquetés sur n polygones est donné par*

$$a_n = \frac{1}{2}\left(m^{n-2}+1\right), \qquad n \geq 2, \tag{18}$$

où $m = (k-1)n+1$.

Pour le dénombrement non étiqueté des 2-arbres k-gonaux (non orientés), nous allons considérer certaines espèces quotients de la forme $F/\mathbb{Z}_2$, où F est une espèce de structures "orientées" et $\mathbb{Z}_2 = \{1, \tau\}$, est un groupe dont l'action de τ sur les F-structures est de renverser l'orientation. Une structure d'une telle espèce quotient consiste alors en une orbite $\{s, \tau\cdot s\}$ de F-structures selon l'action de $\mathbb{Z}_2$.

Par exemple, les diverses espèces pointées de 2-arbres k-gonaux , a^-, $a^\diamond$ et $a^{\underline{\diamond}}$, s'expriment comme espèces quotients des espèces de 2-arbres k-gonaux orientés correspondantes :

$$a^- = \frac{a^{\rightarrow}}{\mathbb{Z}_2}, \quad a^\diamond = \frac{a_o^\diamond}{\mathbb{Z}_2} = \frac{XC_k(B)}{\mathbb{Z}_2}, \quad a^{\underline{\diamond}} = \frac{a_o^{\underline{\diamond}}}{\mathbb{Z}_2} = \frac{XB^k}{\mathbb{Z}_2}. \tag{19}$$

Pour le dénombrement non étiqueté de telles espèces quotients, on utilise la formule suivante qui est évidente :

$$(F/\mathbb{Z}_2)^{\sim}(x) = \frac{1}{2}(\widetilde{F}(x) + \widetilde{F}_\tau(x)), \tag{20}$$

où $\widetilde{F}_\tau(x) = \sum_{n\geq 0} |\mathrm{Fix}_{\widetilde{F}_n}(\tau)|x^n$ est la série génératrice des F-structures non étiquetées laissées fixes par l'action de τ, c'est-à-dire par changement d'orientation. Toutefois, le calcul de ces séries $\widetilde{F}_\tau(x)$ est assez complexe et il est avantageux de différencier en deux cas selon la parité de k.

4.1 Cas k impair

On peut remarquer, en observant les figures 2 a) et b), que dans tout k-gone contenant l'arête pointée (mais non orientée), d'une $\mathcal{A}^-$-structure, il est possible d'orienter les $k-1$ autres arêtes, dans la direction s'éloignant de l'arête pointée comme dans la figure 2 a), lorsque k est impair, mais qu'il restera une arête ambiguë si k est pair. Ce phénomène permet d'introduire des espèces squelettes, lorsque k est impair, en analogie avec l'approche de Fowler et al. [2, 3] où $k = 3$. Ce sont les espèces à deux sortes $Q(X,Y)$, $S(X,Y)$ et $U(X,Y)$, où X représente la sorte des k-gones et Y celle des arêtes orientées, définies par les figures 3 a), b) et c), où $k = 5$. En analogie avec le cas $k = 3$, on a les propositions suivantes.

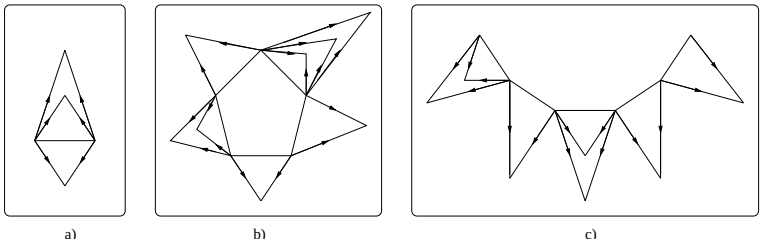


Figure 3: Espèces squelettes a) $Q(X,Y)$, b) $S(X,Y)$ et c) $U(X,Y)$

Proposition 4.2. *Les espèces squelettes Q, S et U admettent des expressions en termes d'espèces quotients :*

$$Q(X,Y) = E(XY^2)/\mathbb{Z}_2,\ S(X,Y) = C_k(E(XY^2))/\mathbb{Z}_2,\ U(X,Y) = (E(XY^2))^k/\mathbb{Z}_2. \tag{21}$$

Proposition 4.3. *Lorsque k est impair, $k \geq 3$, on a les expressions suivantes pour les espèces pointées de 2-arbres k-gonaux , où $B = \mathcal{A}^{\rightarrow}$:*

$$\mathcal{A}^- = Q(X, B^{\frac{k-1}{2}}), \quad \mathcal{A}^\circ = S(X, B^{\frac{k-1}{2}}), \quad \mathcal{A}^{\underline{\circ}} = U(X, B^{\frac{k-1}{2}}). \tag{22}$$

Dans le but d'obtenir des formules d'énumération, il faut préalablement calculer les séries indicatrices de cycles des espèces Q, S et U.

Proposition 4.4. *Les séries indicatrices de cycles des espèces $Q(X,Y)$, $S(X,Y)$ et $U(X,Y)$ sont données par la formule*

$$Z_Q = \frac{1}{2}\left(Z_{E(XY^2)} + q\right), \tag{23}$$

$$Z_S = \frac{1}{2}\left(Z_{C_k(E(XY^2))} + q\cdot(p_2\circ Z_{E(XY^2)})^{\frac{k-1}{2}}\right), \tag{24}$$

$$Z_U = \frac{1}{2}\left(Z_{(E(XY^2))^k} + q\cdot(p_2\circ Z_{E(XY^2)})^{\frac{k-1}{2}}\right), \tag{25}$$

où $q = h\circ(x_1y_2 + p_2\circ(x_1\frac{y_1^2-y_2}{2}))$, p_2 représente la fonction somme de puissances de degré deux, h la fonction symétrique homogène et $\circ$, la composition pléthystique.

Preuve. La formule (23) et la méthode utilisée se trouvent dans [2, 3]. Il s'agit de dénombrer les $F(X,Y)$-structures colorées non étiquetées laissées fixes par τ. Dans le cas de S, on doit laisser fixe une $C_k(E(XY^2))$-structure colorée. Pour cela le cycle de base de longueur k doit posséder au moins un axe de symétrie passant par le milieu d'un des côtés. On peut voir que lorsqu'une telle structure possède plusieurs axes de symétrie, le choix d'un axe est arbitraire. De part et d'autre de l'axe de symétrie, chaque $E(XY^2)$-structure colorée doit avoir son image miroir; ce qui contribue pour un terme de $(p_2\circ Z_{E(XY^2)})^{\frac{k-1}{2}}$. Ensuite, la structure attachée à l'arête distinguée doit être globalement laissée fixe, ce qui donne le facteur q. Le raisonnement est très similaire pour l'espèce U. ■

Combinant le théorème de dissymétrie, les équations (23), (24), (25) et les lois de substitution de la théorie des espèces, on obtient les séries génératrices des types de l'espèce des 2-arbres k-gonaux .

Proposition 4.5. *Soit $k \geq 3$ impair. La série génératrice ordinaire $\widetilde{a}(x)$ des 2-arbres k-gonaux non étiquetés est donnée par*

$$\widetilde{a}(x) = \frac{1}{2}\left(\widetilde{a}_o(x) + \exp\Big(\sum_{i\geq 1}\frac{1}{2i}(2x^i\widetilde{B}^{\frac{k-1}{2}}(x^{2i}) + x^{2i}\widetilde{B}^{k-1}(x^{2i}) - x^{2i}\widetilde{B}^{\frac{(k-1)}{2}}(x^{4i}))\Big)\right). \tag{26}$$

Corollaire 4.6. *Pour $k \geq 3$ impair, le nombre $\widetilde{a}_n$ de 2-arbres k-gonaux non étiquetés sur n k-gones satisfait la récurrence suivante*

$$\widetilde{a}_n = \frac{1}{2n}\sum_{j=1}^{n}\Big(\sum_{l|j} l\omega_l\Big)\Big(\widetilde{a}_{n-j} - \frac{1}{2}\widetilde{a}_{o,n-j}\Big) + \frac{1}{2}\widetilde{a}_{o,n}, \quad \widetilde{a}_k[0] = 1, \tag{27}$$

où, pour tout $n \geq 1$,

$$\omega_n = 2b_{\frac{n-1}{2}}^{(\frac{k-1}{2})} + b_{\frac{n-2}{2}}^{(k-1)} + b_{\frac{n-2}{4}}^{(\frac{k-1}{2})}, \tag{28}$$

et $b_i^{(j)}$ est défini au corollaire 3.3.

4.2 Cas k pair

Le cas où k est pair est plus délicat. Dans le but d'exprimer les séries génératrices ordinaires des types des trois espèces a^-, $a^\diamond$ et $a^{\underline{\diamond}}$, nous appliquons la formule (20) aux formules (19). Pour l'espèce a^-, on a

$$\widetilde{a}^-(x) = \frac{1}{2}(\widetilde{a}^{\rightarrow}(x) + \widetilde{a}^{\rightarrow}_{\tau}(x)), \tag{29}$$

où $\widetilde{a}^{\rightarrow}_{\tau}(x) = \sum_{n\geq 0} |\mathrm{Fix}_{\widetilde{a}^{\rightarrow}_n}(\tau)|x^n$ est la série génératrice des 2-arbres k-gonaux pointés en une arête orientée, non étiquetés, laissés fixes par changement d'orientation. Il faut donc calculer $\widetilde{a}^{\rightarrow}_{\tau}(x)$. Pour cela, introduisons quelques espèces auxiliaires. La première, notée a_{TS}, est l'espèce des 2-arbres k-gonaux pointés en une arête orientée et dont toutes les pages attachées autour de cette arête sont verticalement symétriques, sans symétries croisées (voir plus loin); on dira *totalement symétriques*. On peut caractériser cette espèce par l'équation fonctionnelle suivante

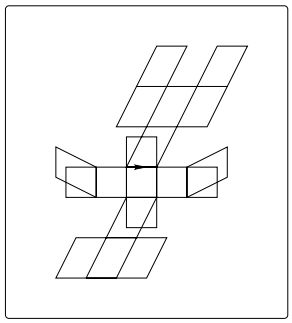


Figure 4: Une structure de l'espèce a_{TS}

(voir figure 4),

$$a_{\mathrm{TS}} = E(X \cdot X^2_{=} < B^{\frac{k-2}{2}} > \cdot a_{\mathrm{TS}}) = E(P_{\mathrm{TS}}), \tag{30}$$

où $X^2_{=} < F >$ représente l'espèce des couples de F-structures isomorphes et P_{TS} est l'espèce des *pages totalement symétriques*. Cette équation se traduit au niveau des séries génératrices des types par

$$\widetilde{a}_{\mathrm{TS}}(x) = \exp\left(\sum_{i\geq 1} \frac{1}{i} x^i \widetilde{B}^{\frac{k-2}{2}}(x^{2i})\widetilde{a}_{\mathrm{TS}}(x^i)\right). \tag{31}$$

Proposition 4.7. *Les nombres $\beta_n = |\widetilde{a}_{\mathrm{TS}}[n]|$, de a_{TS}-structures non étiquetées sur n polygones satisfont la récurrence*

$$\beta_n = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{d|i} d\omega_d\right)\beta_{n-i},\ n \geq 1 \quad \beta_0 = 1, \tag{32}$$

où

$$\omega_n = \sum_{\substack{i+j=n-1 \\ i\ pair}} b_{\frac{i}{2}}^{(\frac{k-2}{2})}\beta_j\,.$$

Preuve. Il suffit de prendre la dérivée logarithmique de l'expression (31). ■

Passons maintenant à l'introduction des deux espèces $P_{\rm CR}$ et $P_{\rm M}$, des *paires de pages croisées* et des *pages mixtes*. Une paire de pages *croisées* est, par définition, une paire de pages orientées (des $\mathcal{A}^{\rightarrow}$-structures comportant une seule page) de la forme $\{s, \tau\cdot s\}$ avec s et $\tau\cdot s$ non isomorphes. La figure 5 a) montre une structure de cette espèce. Une page *mixte* est une page symétrique possédant une (ou plusieurs) symétrie de type croisée. Une telle structure est dessinée en figure 5 b). On peut alors exprimer ces deux espèces l'une en fonction de l'autre, comme suit

$$\begin{aligned} P_{\rm CR} &= \Phi_2 < XB^{k-1} - (P_{\rm TS} + P_{\rm M}) >, & (33)\\ P_{\rm M} &= X\cdot X_{=}^2 < B^{\frac{k-2}{2}} > \cdot \mathcal{A}_{\rm TS}\cdot E_+(P_{\rm CR} + P_{\rm M}), & (34)\end{aligned}$$

où $\Phi_2 < F >$ représente l'espèce des paires de F-structures de la forme $\{s, \tau\cdot s\}$ et E_+ est l'espèce des ensembles non vides. Passant aux séries génératrices des types, il vient

$$\begin{aligned} \widetilde{P}_{\rm CR}(x) &= \frac{1}{2}\big(x^2\widetilde{B}^{k-1}(x^2) - \widetilde{P}_{\rm TS}(x^2) - \widetilde{P}_{\rm M}(x^2)\big), & (35)\\ \widetilde{P}_{\rm M}(x) &= x\widetilde{B}^{\frac{k-2}{2}}(x^2)\widetilde{\mathcal{A}}_{\rm TS}(x)\left(\exp\left(\sum_{i\geq 1}\frac{1}{i}(\widetilde{P}_{\rm CR}(x^i) + \widetilde{P}_{\rm M}(x^i))\right) - 1\right). & (36)\end{aligned}$$

Après manipulations et la prise de la dérivée logarithmique de (36), on obtient les nombres $\widetilde{P}_{{\rm CR},n}$ et $\widetilde{P}_{{\rm M},n}$ de pages croisées et mixtes respectivement sur n polygones

$$\begin{aligned} \widetilde{P}_{{\rm CR},n} &= b^{(k-1)}_{\frac{n-2}{2}} - \widetilde{P}_{{\rm TS},\frac{n}{2}} - \widetilde{P}_{{\rm M},\frac{n}{2}}, & (37)\\ \widetilde{P}_{{\rm M},n} &= \sum_{i=1}^{n}\left(\sum_{d|i}\varepsilon_d\right)c_{n-i} + f_n, & (38)\end{aligned}$$

où

$$\begin{aligned} \varepsilon_n &= \frac{k-2}{2}b^{(k-1)}_{n-1} + \widetilde{P}_{{\rm TS},n} + \widetilde{P}_{{\rm CR},n} + \widetilde{P}_{{\rm M},n}, & (39)\\ c_n &= \widetilde{P}_{{\rm M},n} + \sum_{i+j=n-1} b^{(\frac{k-2}{2})}_{\frac{1}{2}}\widetilde{a}_{{\rm TS},j}, & (40)\\ f_n &= \sum_{i+j=n-1} b^{(\frac{k-2}{2})}_{\frac{1}{2}}\widetilde{a}_{{\rm TS},j} + 2\sum_{i+j+l=n-2} b^{(\frac{k-4}{2})}_{\frac{1}{2}} j b_{\frac{j}{2}}\widetilde{a}_{{\rm TS},l} \\ &\qquad + \sum_{i+j=n-1} j b^{(\frac{k-2}{2})}_{\frac{1}{2}}\widetilde{a}_{{\rm TS},j}. & (41)\end{aligned}$$

Notons par $\widetilde{\mathcal{A}}_{\rm S}(x)$ la série génératrice des $\mathcal{A}^{\rightarrow}$-structures non étiquetées symétriques. On a alors (voir figure 6)

$$\begin{aligned} \widetilde{\mathcal{A}}_{\rm S}(x) &= E(P_{\rm TS} + P_{\rm CR} + P_{\rm M})^{\sim}(x), & (42)\\ &= \exp\left(\sum_{i\geq 1}\frac{1}{i}(\widetilde{P}_{\rm TS}(x^i) + \widetilde{P}_{\rm CR}(x^i) + \widetilde{P}_{\rm M}(x^i))\right). & (43)\end{aligned}$$

On en déduit alors une récurrence pour le nombre $\alpha_n = \widetilde{a}_{\mathrm{S},n}$ de 2-arbres k-gonaux pointés en une arête laissés fixes par changement d'orientation.

$$\alpha_n = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{d|i} d\omega_d\right)\alpha_{n-i}, \quad \alpha_0 = 1, \tag{44}$$

où

$$\omega_k = \widetilde{P}_{\mathrm{TS},k} + \widetilde{P}_{\mathrm{CR},k} + \widetilde{P}_{\mathrm{M},k}.$$

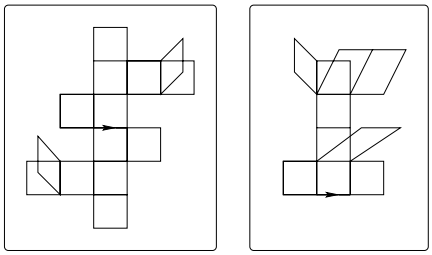


Figure 5: Une paire de pages croisées et une page mixte

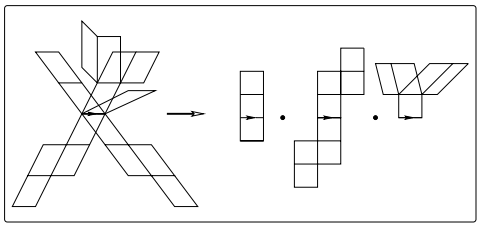


Figure 6: Décomposition d'une $\mathcal{A}^{\rightarrow}$-structure fixée sous τ

Proposition 4.8. *Si k est un entier pair, $k \geq 4$, alors le nombre de 2-arbres k-gonaux pointés en une arête (non orientée) sur n k-gones est donné par*

$$\widetilde{a}_n^{-} = \frac{1}{2}(b_n + \alpha_n). \tag{45}$$

Passons maintenant à l'espèce $\mathcal{A}^{\diamond}$ des 2-arbres k-gonaux pointés en un k-gone possédant une arête distinguée. On trouve

$$\widetilde{a}^{\diamond}(x) = \frac{1}{2}\left(\widetilde{a}_o^{\diamond}(x) + \widetilde{a}_{o,\tau}^{\diamond}(x)\right), \quad \text{où} \quad \widetilde{a}_{o,\tau}^{\diamond}(x) = x\,\widetilde{a}_{\mathrm{S}}^{2}(x)\,\widetilde{B}^{\frac{k-2}{2}}(x^2), \tag{46}$$

puisque une $\mathcal{A}_o^{\diamond}$-structure non étiquetée τ-symétrique possède un axe de symétrie qui est, en fait, la médiatrice de l'arête distinguée dans le polygone pointé, et, qui est donc aussi naturellement la médiatrice de l'arête opposée à celle pointée.

Les structures attachées à ces deux arêtes sont donc symétriques, d'où le terme $(\widetilde{a}_{\mathrm{S}}(x))^2$; ensuite, de part et d'autre de l'axe, les B-structures que l'on y attache doivent s'échanger par paire, soit une contribution d'un facteur $\widetilde{B}(x^2)$ pour chacune des $\frac{k-2}{2}$ paires. On en déduit alors une expression du nombre de $\underline{a^{\diamond}}$-structures non étiquetées $\widetilde{a}^{\diamond}_n$,

$$\widetilde{a}^{\diamond}_{\underline{n}} = \frac{1}{2}\left(\widetilde{a}^{\diamond}_{o,n} + \sum_{i+j=n-1} \alpha^{(2)}_i \cdot b^{(\frac{k-2}{2})}_{\underline{j}}\right), \tag{47}$$

où $\alpha^{(2)}_i = [x^i]\widetilde{a}^2_S(x)$.

Procédons de façon similaire pour l'espèce $\underline{a^{\diamond}}$, des 2-arbres k-gonaux pointés en un polygone. Une nouvelle fois, nous utilisons la relation (20), qui donne

$$\widetilde{a}^{\diamond}(x) = \frac{1}{2}\left(\widetilde{a}^{\diamond}_o(x) + \widetilde{a}^{\diamond}_{o,\tau}(x)\right). \tag{48}$$

Remarquons d'abord que pour qu'une $a^{\diamond}_o$-structure soit laissée fixe par changement d'orientation, elle doit comporter au moins un axe de symétrie, qui peut être de deux types :

1. un axe passant par le milieu de deux arêtes opposées, ou
2. un axe passant par deux sommets opposés,

du polygone pointé. Le dénombrement se fait en orientant d'abord l'axe de symétrie. On trouve

$$\widetilde{a}^{\diamond}_{o,\tau}(x) = \frac{x}{2}\widetilde{a}^2_{\mathrm{S}}(x)\widetilde{B}^{\frac{k-2}{2}}(x^2) + \frac{x}{2}\widetilde{B}^{\frac{k}{2}}(x^2), \tag{49}$$

où le premier terme correspond à une symétrie de type 1, et le deuxième, de type 2. Les structures qui possèdent les deux symétries sont précisément celles qui sont comptées une demi fois dans chacun des deux termes. Le théorème de dissymétrie donne donc, pour $k \geq 4$ pair,

$$\begin{aligned}\widetilde{a}(x) &= \frac{1}{2}\widetilde{a}_o(x) + \frac{1}{2}\widetilde{a}_{\mathrm{S}}(x) + \frac{1}{2}\widetilde{a}^{\diamond}_{o,\tau}(x) - \frac{1}{2}\widetilde{a}^{\diamond}_{o,\tau}(x), \\ &= \frac{1}{2}\widetilde{a}_o(x) + \frac{1}{2}\widetilde{a}_{\mathrm{S}}(x) + \frac{x}{4}(\widetilde{B}^{\frac{k}{2}}(x^2) - \widetilde{a}^2_{\mathrm{S}}(x)\widetilde{B}^{\frac{k-2}{2}}(x^2)),\end{aligned} \tag{50}$$

où $\widetilde{a}_o(x)$ est donné par (15) et $\widetilde{a}_{\mathrm{S}}(x)$ par (43).

Théorème 4.9. *Si $k \geq 4$ est pair, le nombre de 2-arbres k-gonaux non étiquetés sur n k-gones est donné par*

$$\widetilde{a}_n = \frac{1}{2}\widetilde{a}_{o,n} + \frac{1}{2}\alpha_n + \frac{1}{4}b^{(\frac{k}{2})}_{\frac{n-1}{2}} - \frac{1}{4}\sum_{i+j=n-1} \alpha^{(2)}_i \cdot b^{(\frac{k-2}{2})}_{\underline{j}}, \tag{51}$$

avec

$$b^{(m)}_l = [x^l]\widetilde{B}^m(x), \quad \alpha^{(2)}_i = [x^i]\widetilde{a}^2_{\mathrm{S}}(x).$$

5 Dénombrement asymptotique

Grâce au théorème de dissymétrie et aux diverses équations combinatoires qui lui sont associées, le dénombrement asymptotique des 2-arbres k-gonaux (étiquetés ou non) dépend essentiellement de celui des B-structures où B est l'espèce auxiliaire caractérisée par l'équation combinatoire (3). Dans le cas étiqueté, la situation est triviale puisque l'on dispose des formules closes simples (7), (16) et (18). Dans le cas non étiqueté, la situation est vraiment plus délicate puisque la série $\widetilde{B}(x)$ est caractérisée par l'équation fonctionnelle complexe (12).

Voici quelques notations préliminaires à l'énoncé du résultat principal de la présente section. Si $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_\nu)$ est un partage d'un entier n en ν parts, on écrit $\lambda \vdash n$, $n = |\lambda|$, $\nu = l(\lambda)$, $m_i(\lambda) = |\{j : \lambda_j = i\}|$ = nombre de parts de taille i dans λ. De plus, on pose

$$\sigma_i(\lambda) = \sum_{d|i} d m_d(\lambda), \quad \sigma_i^*(\lambda) = \sum_{d|i,\, d<i} d m_d(\lambda) \tag{52}$$

$$\widehat{\lambda} = 1 + |\lambda| + l(\lambda), \quad \widehat{z}(\lambda) = 2^{m_1(\lambda)} m_1(\lambda)! 3^{m_2(\lambda)} m_2(\lambda)! \ldots . \tag{53}$$

On a le résultat suivant.

Proposition 5.1. *Posons $p = k - 1$ et $\widetilde{B}(x) = \sum b_n(p) x^n$. Alors*

i) $b_n(p)$ est un polynôme en p de degré $n-1$, $n \geq 1$,

ii) il existe des constantes α_p et β_p telles que

$$b_n(p) \sim \alpha_p \beta_p^n n^{-\frac{3}{2}}, \qquad \text{pour } n \to \infty. \tag{54}$$

De plus, $\alpha_p = \alpha(\xi_p) = \dfrac{1}{\sqrt{2\pi}} \dfrac{1}{(p\xi_p)^{\frac{1}{p}} p} \left(1 + \dfrac{p\xi_p \omega'(\xi_p)}{\omega(\xi_p)}\right)^{\frac{1}{2}}$ et $\beta_p = \dfrac{1}{\xi_p}$, où ξ_p est la plus petite racine de l'équation

$$\xi = \frac{1}{ep} \omega^{-p}(\xi), \tag{55}$$

où $\omega(x)$ est la série (absolument convergente au voisinage de ξ_p) donnée par (58). On a le développement convergent

$$\xi_p = \sum_{n=1}^{\infty} \frac{c_n}{p^n}, \tag{56}$$

où les coefficients c_n sont des constantes, indépendantes de p, données explicitement par

$$c_n = \sum_{\lambda \vdash n} \frac{e^{-\widehat{\lambda}}}{\widehat{\lambda}\widehat{z}(\lambda)} \prod_{i \geq 1} (\sigma_i(\lambda) - \widehat{\lambda})^{m_i(\lambda)-1} (\sigma_i^*(\lambda) - \widehat{\lambda}), \tag{57}$$

lorsque λ parcourt l'ensemble des partages de n.

Preuve. La partie *i)* de l'énoncé découle immédiatement de la formule explicite (8). Pour la partie *ii)* qui affirme l'existence des constantes α_n et β_n, on s'inspire de l'approche de Fowler et al. pour les 2-arbres ($k = 3$) en utilisant le théorème classique de Bender. Posons, pour simplifier $b(x) = \widetilde{B}(x)$. Alors, grâce à (12), $y = b(x)$ satisfait la relation

$$y = e^{xy^p}\omega(x), \quad \text{où} \quad \omega(x) = e^{\frac{1}{2}x^2b^p(x^2)+\frac{1}{3}x^3b^p(x^3)+\cdots} \tag{58}$$

Par le théorème de Bender, appliqué à la fonction $f(x,y) = y - e^{xy^p}\omega(x)$, on doit chercher un couple (ξ_p, τ_p) solution du système

$$f(x,y) = 0 \qquad \text{et} \qquad f_y(x,y) = 0. \tag{59}$$

Ceci équivaut à dire que ξ_p est solution de (55) et que $p\xi_p\tau_p^p = 1$. Les formules explicites (56) et (57) s'obtiennent en appliquant préalablement l'inversion de Lagrange à l'équation $\xi = zR(\xi)$ où $z = \frac{1}{ep}$ et $R(t) = \omega^{-p}(t)$, pour obtenir

$$\xi_p = \xi = \sum_{n\geq 0}\frac{a_n}{n!}\left(\frac{1}{ep}\right)^n, \quad \frac{a_n}{n!} = \frac{1}{n}[t^{n-1}]\omega^{-np}(t). \tag{60}$$

Ensuite, pour évaluer explicitement $\omega^{-np}(x)$, on utilise la version de Labelle [7] de la formule d'inversion de Good pour les séries indicatrices en tenant compte de (6) et en remarquant que

$$\omega^{-np}(x) = e^{-n(\frac{x_2}{2}+\frac{x_3}{3}+\cdots)} \circ Z_A(x_1, x_2, \ldots)|_{x_i:=px^i}, \tag{61}$$

où $A = XE(A)$ est l'espèce des arborescences. ■

Dans le cas orienté non pointé, une méthode similaire basée sur l'équation (15), mène à

$$\widetilde{a}_{o,n} \sim \overline{\alpha}_p\beta_p^n n^{-\frac{5}{2}}, \quad \text{où} \quad \overline{\alpha}_p = 2\pi p(p\xi_p)^{\frac{2}{p}}\alpha_p^3. \tag{62}$$

Enfin, une analyse fine de la formule (51) montre que

$$\widetilde{a}_n \sim \frac{1}{2}\widetilde{a}_{o,n}. \tag{63}$$

La table 1 donne, à 20 décimales, les constantes ξ_p, α_p et $\beta_p = \frac{1}{\xi_p}$ pour $p = 1, \ldots, 5$.

p	ξ_p	α_p	β_p
1	0.3383218568 9920769520	1.3003121246 8216843599	2.955765285651994974715
2	0.177099522303285617693	0.349261381742311443973	5.646542616232949712893
3	0.119674100436145452060	0.191997258649948899321	8.356026879295995368276
4	0.090334539604383047938	0.131073637348549764379	11.069962877759326312419
5	0.072539192528125499910	0.099178841365021748147	13.785651110084685198930

Table 1 : Valeurs numériques de ξ_p, α_p et β_p, $p = 1, \ldots, 5$.

Voici les premières valeurs des constantes universelles c_n apparaissant dans (56), pour $n = 1, \ldots 5$.

$$c_1 = \frac{1}{\mathrm{e}} = 0.36787944117144232160, \tag{64}$$

$$c_2 = -\frac{1}{2}\frac{1}{\mathrm{e}^3} = -0.02489353418393197149, \tag{65}$$

$$c_3 = \frac{1}{8}\frac{1}{\mathrm{e}^5} - \frac{1}{3}\frac{1}{\mathrm{e}^4} = -0.00526296958802571004, \tag{66}$$

$$c_4 = -\frac{1}{48}\frac{1}{\mathrm{e}^7} + \frac{1}{\mathrm{e}^6} - \frac{1}{4}\frac{1}{\mathrm{e}^5} = 0.00077526788594593923, \tag{67}$$

$$c_5 = \frac{1}{384}\frac{1}{\mathrm{e}^9} - \frac{4}{3}\frac{1}{\mathrm{e}^8} + \frac{49}{72}\frac{1}{\mathrm{e}^7} - \frac{1}{5}\frac{1}{\mathrm{e}^6} = 0.00032212622183609932. \tag{68}$$

Remarque 5.1. *Les calculs de cette section sont également valables pour le cas où $k = 2$ et $p = 1$, correspondant aux arborescences ordinaires (de Cayley) définies par l'équation $A = XE(A)$. Dans ce cas, la constante de croissance $\beta = \beta_1$, dans (54), est connue sous le nom de constante d'Otter (voir [10]). Il est intéressant de noter que cette constante prend la forme explicite $\beta = \frac{1}{\xi_1}$, avec*

$$\xi_1 = \sum_{n \geq 1} c_n. \tag{69}$$

Il est à noter que lorsque $k = 3$, nous retrouvons les résultats asymptotiques obtenus par Fowler et al. dans [2, 3].

References

[1] F. Bergeron, G. Labelle, and P. Leroux, *Combinatorial Species and tree-like structures*, Encyclopedia of Mathematics and its Applications, vol. 67, Cambridge University Press, (1998).

[2] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *Specifying 2-trees*, Proceedings FPSAC'00, Moscou, 26-30 juin 2000, 202–213.

[3] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *The Specification of 2-trees*, Advances in Applied Mathematics, 28, 145–168, (2002).

[4] F. Harary and E. Palmer, *Graphical Enumeration*, Academic Press, New York, (1973).

[5] F. Harary, E. Palmer and R. Read, *On the cell-growth problem for arbitrary polygons*, Discrete Mathematics, 11, 371–389, (1975).

[6] INRIA, *Encyclopedia of combinatorial structures.* http://algo.inria.fr/encyclopedia/index.html.

[7] G. Labelle, *Some new computational methods in the theory of species*, Combinatoire énumérative, Proceedings, Montréal, Québec, Lectures Notes in Mathematics, vol. 1234, Springer-Verlag, New-York/Berlin, 160–176, (1985).

[8] G. Labelle, C. Lamathe and P. Leroux, *Développement moléculaire de l'espèce des 2-arbres planaires*, Proceedings GASCom01, 41–46, (2001).

[9] G. Labelle, C. Lamathe and P. Leroux, *A classification of plane and planar 2-trees*, preprint CO/0202052, submitted.

[10] R. Otter, *The number of trees*, Annals of Mathematics, 49, 583–599, (1948).

[11] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, (1995).

Gilbert Labelle, Cédric Lamathe, Pierre Leroux
LaCIM
Université du Québec à Montréal
Case Postale 8888, succursale centre-ville
H3C 3P8 Montréal
{gilbert, lamathe, leroux}@math.uqam.ca

Refined Upper and Lower Bounds for 2-SUM

Alexander C. Chan *
University of Maryland

William I. Gasarch†
University of Maryland

Clyde P. Kruskal‡
University of Maryland

Abstract

We prove upper and lower bounds on the time complexity of solving the 2-SUM problem: given a set of numbers, are there two of them that sum to zero? Our basic models are the linear decision tree and the degree-d algebraic decision tree. Our bounds are more precise than is common for this field and allow us to observe that 2-SUM is strictly harder than sorting in the linear decision tree model.

1 Introduction, Upper Bound, and Model

The 3-SUM problem is: Given n numbers, do any *three* of them sum to zero? This problem is important in Computational Geometry [BBG94, Eri96, GO95, ORou94] because if 3-SUM requires $\Omega(n^2)$ steps (which is likely [Eri96, ES95]) then the following problems (and others) also require $\Omega(n^2)$ steps:

1. Given n points in the plane, determine if some three of them that are co-linear [GO95].
2. Given n triangles, compute the area of their union [GO95].

We study a related problem, namely 2-SUM: Given n numbers, do any *two* of them sum to zero? We first show that this problem can be solved with $O(n \log n)$ linear comparisons and requires $\Omega(n \log n)$ queries on a d-ADT (see definition below). We refine the constants on both the upper and lower bounds. The mathematics used for the refined lower bound is of interest and may be useful on other problems such as Element Distintness and Two-list Element Distinctness (discussed in Section 6.) In addition our results show that 2-SUM is *strictly harder* than sorting in the linear decision tree model, which is clearly of interest.

To clarify our model we exhibit a well-known upper bound. First sort the numbers in nondecreasing order. Let $i = 1$, $j = n$, and $s := x_i + x_j$. Test $s \leq 0$. If $s \leq 0$ then we need to test $s \geq 0$, otherwise we do not. If $s = 0$ then we are done. if $s < 0$ then $i = i + 1$, and if $s > 0$ then $j := j - 1$. In either case $s = x_i + x_j$ and repeat until $j < i$. In effect we are keeping pointers at each end of the sorted list and moving them toward each other in a fashion that guarantees that if there are two numbers that sum to zero, we will find them. Otherwise we determine that no two (distinct) numbers sum to zero. The entire algorithm requires sorting and then at most $2n - 2$ comparisons. Since the initial sorting can be done in $n \lg n - 1.329n$ comparisons [FJ59, HL69, Knu73, Man79], this yields an *upper bound* of

*Dept. of C.S., U. of MD, College Park, MD 20742. (alexchan@cs.umd.edu).

†Dept. of C.S. and Inst. for Adv. Comp. Studies, U. of MD, College Park, MD 20742. Supported in part by NSF grant CCR-97-32692 (gasarch@cs.umd.edu).

‡Dept. of C.S. U. of MD, College Park, MD 20742. (kruskal@cs.umd.edu).

$n \lg n - 1.329n + 2n \leq n \lg n + 0.67n$ comparisons. We will give an improved upper bound later in this paper.

The algorithm above uses comparisons and questions of the form "$x + y = 0$?" The latter is viewed as asking "$x + y \leq 0$?" and "$-x - y \leq 0$?" Hence the natural basic operation to consider is the 2-ary linear comparisons: questions of the form "$ax + by \leq c$?". A more general model would allow questions of the form "$\sum_{i=1}^{n} a_i x_i$ COMP b?" where COMP is one of $\{<, \leq, =, \geq, >\}$. In this notation x_i is an input value and a_i and b are constants. An even more general model would allow, for some fixed d, a comparison between a polynomial in $x_1, \ldots, x_n$ of degree d and a constant.

A sequential algorithm involving linear comparisons is represented as a *linear decision tree*, (henceforth an LDT) a finite rooted binary tree with a linear comparison at each node and YES and NO edges from each node to its children. The input $(x_1, \ldots, x_n)$ to an LDT determines a path from the root (first linear comparison) to a leaf by the outcome of the linear comparisons encountered. The answer at the leaf node is the output of the LDT. A sequential algorithm involving comparisons between polynomials of degree d and constants is called an *algebraic decision tree of degree d* (henceforth a d-ADT) and is defined similarly.

If T is an LDT or d-ADT then let $\text{ht}(T)$ be the height of the tree, i.e., the maximum number of internal nodes on a path from the root to a leaf. In Section 4 we show that if T is a d-ADT for 2-SUM then $\text{ht}(T) \geq \Omega(n \log n)$. In Section 5 we refine this lower bound.

Our final results are

1. 2-SUM can be computed with $n \lg n + 0.351n + O(\lg n)$ linear queries.
2. 2-SUM requires $n \lg n - 0.92n - \Omega(\lg n)$ linear queries.
3. 2-SUM requires $\frac{0.38n \lg n - 0.96n}{d} - \Omega(\lg n)$ queries on a d-ADT.

2 Preliminaries

The following proposition, due to Dobkin and Lipton [DL79] and Ben-Or [Ben83] can be used to obtain lower bounds.

Definition 2.1 A set $X \subseteq R^n$ is *connected* if for all points $x, y \in X$ there is a path from x to y that is entirely inside X. Let $\mathcal{E} \subseteq R^n$. X is a *connected component of* $\mathcal{E}$ if $X \subseteq \mathcal{E}$, X is connected, and no superset of X is contained in $\mathcal{E}$.

Proposition 2.2 *Let $\mathcal{E}$ be the union of N connected components in R^n.*

1. *[DL79] If T is an LDT for determining membership in $\mathcal{E}$ then*

$$\text{ht}(T) \geq \lg N.$$

2. *[Ben83] If T is a d-ADT for determining membership in $\mathcal{E}$ then*

$$\text{ht}(T) \geq \frac{0.38 \lg N - 0.61n}{d}.$$

(This is obtained by looking at Ben-Or's paper more carefully than is commonly done. The constant 0.38 is a close upper bound for $\frac{1}{1+\lg 3}$. The constant 0.61 is a close lower bound for $\frac{\lg 3}{1+\lg 3}$.)

We would like to cast 2-SUM as a decision problem in this framework. Let

$$\mathcal{E}_n = \{(x_1, \ldots, x_n) \in R^n : (\forall i \neq j)[x_i + x_j \neq 0]\}.$$

Clearly $\mathcal{E}_n \subseteq R^n$ is the set of all inputs to 2-SUM which answer NO. Hence we can obtain a lower bound on 2-SUM by counting the connected components of $\mathcal{E}_n$. Let $\#\mathcal{E}_n$ denote this number.

3 A Refined Upper Bound

Theorem 3.1 *There is an algorithm for 2-SUM that takes $n \lg n + 0.351n + O(\lg n)$ comparisons.*

Proof: The basic idea is to partition the numbers into two sets, the negative and nonnegative numbers, and look for a number in the larger set whose additive complement is in the smaller set. The only minor hitch is that we need to check for the special case of two occurrences of 0 in the set of nonnegative numbers.

Let $S = \{a_1, a_2, \ldots, a_n\}$. Compare all numbers to 0, putting them into two sets: L, those less than 0, and G, those greater than or equal 0. Let X be the set with fewer elements, and let x be its size. Sort X, which takes time at most $x \lg x - \alpha x + \frac{1}{2} \lg x + \beta$, where $\alpha = 2 - \lg 3 + \lg e - \lg \lg e \approx 1.329$ and $\beta < 3.3$ [FJ59, HL69, Knu73, Man79].

If $X = G$ then check if the list starts with two 0's; if so, we are done. If $X = L$ append a 0 to the end of L. For every element $a \in S - X$ look for $-a$ in X. (It takes two matches to succeed for 0.) The searches take at most $(n - x)(\lceil \lg(x + 1) + 1 \rceil)$ comparisons using binary search.

The total number of comparison steps for this algorithm is at most

$$\begin{aligned} & n + x \lg x - \alpha x + \frac{1}{2} \lg x + \beta + 2 + (n - x)(\lceil \lg(x + 1) + 1 \rceil) \\ \leq \; & n + x \lg x + \frac{1}{2} \lg(x + 1) - \alpha x + \beta + 2 + (n - x)(\lg(x + 1) + 2) \\ \leq \; & (n + \frac{1}{2}) \lg(x + 1) - (2 + \alpha)x + 3n + \beta + 2 \end{aligned} \tag{1}$$

To maximize, take the derivative and set to 0.

$$\frac{(n + \frac{1}{2}) \lg e}{x + 1} - (2 + \alpha) = 0 \implies x = \frac{(n + \frac{1}{2}) \lg e}{2 + \alpha} - 1$$

Substituting back into (1) the total number of steps is at most

$$\begin{aligned} & (n + \frac{1}{2}) \lg(\frac{(n + \frac{1}{2}) \lg e}{2 + \alpha}) - (2 + \alpha) \left(\frac{(n + \frac{1}{2}) \lg e}{2 + \alpha} - 1 \right) + 3n + \beta + 2 \\ = \; & n \lg(n + \frac{1}{2}) + \frac{1}{2} \lg(n + \frac{1}{2}) + (3 - \lg e - \lg(2 + \alpha) + \lg \lg e)n - \alpha + \beta \\ \leq \; & n \lg n + 0.351n + \frac{1}{2} \lg n + O(1) \end{aligned}$$

∎

4 An Easy Lower Bound

The following lower bound, while easy, does not seem to be in the literature.

Theorem 4.1 *If T is a d-ADT for $\mathcal{E}_n$ then* $\mathrm{ht}(T) \geq \Omega(n \log n)$.

Proof: We assume n is even. The case of n odd is similar. Let $n = 2m$. We show that $\mathcal{E}_n$ has at least $m! = \Omega(n \log n)$ connected components.

Let $\sigma \in S_m$. Let A_σ be the set of all $(x_1, \ldots, x_m, y_1, \ldots, y_m)$ such that

$$-x_m < y_{\sigma(1)} < -x_{m-1} < y_{\sigma(2)} < \cdots < -x_1 < y_{\sigma(m)} < 0 < x_1 < x_2 < \cdots < x_m.$$

It is easy to see that each A_σ is a connected component of $\mathcal{E}_n$. Since there are $m!$ such components we are done. ∎

5 A Refined Lower Bound

The set $\mathcal{E}_n$ can be expressed as a disjoint union of nonempty open sets. The number of these open sets is the number of connected components of $\mathcal{E}_n$. We call these open sets *the cells of* $\mathcal{E}_n$. We need to count them.

To this end we introduce (undirected) threshold graphs. Our goal is to establish a bijection between the cells of $\mathcal{E}_n$ and the labeled threshold graphs on n vertices. Since threshold graphs have been enumerated, we have a count of the cells of $\mathcal{E}_n$.

Definition 5.1 [Stan] A *threshold graph* may be defined recursively as follows:

1. The empty graph is a threshold graph.
2. If G is a threshold graph, then so is the disjoint union of G with a one-vertex graph.
3. If G is a threshold graph, then so is the (edge) complement of G.
4. No other graph is a threshold graph.

Note 5.2 If graph G has an isolated vertex v, then G is a threshold graph iff $G - v$ is a threshold graph, by Definition 5.1, condition 2.

Notation 5.3 We denote the number of threshold graphs on n vertices by $t(n)$.

Notation 5.4 If $G = (V, E)$ is a graph and $v \in V$ then $G - \{v\}$ is the graph

$$(V - \{v\}, E - \{\{u, v\} : u \in V\}).$$

Lemma 5.5 *If $n \geq 2$ then $\#\mathcal{E}_n = t(n)$.*

Proof: Let $I = \{(i,j) : 1 \leq i < j \leq n\}$. Let E_n be a mapping from I to $\{<, >\}$. Throughout the proof we view E_n as an unordered set of inequalities of the form $x_i + x_j < 0$ or $x_i + x_j > 0$. For each E_n let O^{E_n} be the open set $\{(x_1, \ldots, x_n) : (x_i + x_j)\ E_n[i,j]\ 0\}$. Note that $\mathcal{E}_n$ is the disjoint union of O^{E_n}'s over all possible E_n's. Unfortunately many of the O^{E_n} are empty. We map all the E_n such that $O^{E_n} \neq \emptyset$ onto the set of all threshold graphs on n vertices.

We define a function $\mathbf{P}$ that will, given a map E_n such that $O^{E_n} \neq \emptyset$, return a threshold graph on n vertices. Given E_n such that $O^{E_n} \neq \emptyset$, let $\mathbf{P}(G)$ be the graph on n vertices that has edge (i,j) iff $E_n[i,j]$ is $>$. We show that $\mathbf{P}$ is 1-1 and maps onto the set of threshold graphs.

Range of $\mathbf{P}$ *is Threshold Graphs:* For $n = 2$ this is easy. Suppose (by way of contradiction) that $n > 2$ is the smallest value such that there exists an E_n such that $O^{E_n} \neq \emptyset$ and $\mathbf{P}(E_n)$ is not a threshold graph. We show that n is not minimal. $\mathbf{P}(E_n)$ does not have an isolated vertex: assume it had an isolated vertex i. Then E_n thinks $x_i + x_j < 0$ for all j. Hence if E'_{n-1} is E_n without any of the inequalities that mention x_i then $O^{E'_{n-1}} \neq \emptyset$ and $\mathbf{P}(E'_{n-1})$ is not a threshold graph, contradicting the minimality of n. Similarly, the complement of $\mathbf{P}(E_n)$ cannot have an isolated vertex. Let $(x_1, \ldots, x_n) \in O^{E_n}$. Let i-min be such that $x_{i\text{-min}} = \min_{1 \leq i \leq n} x_i$. Let i-max be such that $x_{i\text{-max}} = \max_{1 \leq i \leq n} x_i$.

If $x_{i\text{-min}} + x_{i\text{-max}} > 0$ then $x_{i\text{-max}} + x_i > 0$ for all $1 \leq i \leq n$ and i-max is an isolated vertex in the complement of $\mathbf{P}(E_n)$. If $x_{i\text{-min}} + x_{i\text{-max}} < 0$ then $x_{i\text{-min}} + x_i < 0$ for all $1 \leq i \leq n$ and i-min is an isolated vertex in $\mathbf{P}(E_n)$. So $x_{i\text{-min}} + x_{i\text{-max}} = 0$ which implies $(x_1, \ldots, x_n) \notin \mathcal{E}_n$. Contradiction.

One-to-one: If $E_n \neq E'_n$ then they differ on some (i,j). Hence the graphs $\mathbf{P}(E_n)$ and $\mathbf{P}(E'_n)$ differ on (i,j)

Onto: The $n = 2$ case is easy. Suppose for $n \geq 2$, $\mathbf{P}$ maps onto threshold graphs on n vertices. Consider G, a threshold graph on $n+1$ vertices. Either G or $\overline{G}$ has an isolated vertex i such that $G - \{i\}$ is a threshold graph. We assume it is G, the other case is similar. Renumber so that $i = n+1$. Let E_n be such that $\mathbf{P}(E_n) = G - \{n+1\}$ and $O^{E_n} \neq \emptyset$. Let E'_{n+1} be $E_n \cup \{x_{n+1} + x_i > 0 : 1 \leq i \leq n\}$. Clearly $O^{E'_{n+1}} \neq \emptyset$ and $\mathbf{P}(E'_{n+1}) = G$.

The bijection establishes the desired result. ∎

We now need a good approximation for $t(n)$.

Theorem 5.6 $t(0) = t(1) = 1$. $(\forall n \geq 2)[t(n) = 2 + \sum_{i=2}^{n-1} \binom{n}{i} t(i)]$.

Proof: Let $s(n)$ denote the number of threshold graphs with no isolated vertex. Since a threshold graph is a choice of isolated vertices, $I \subseteq \{1, \ldots, n\}$, plus a threshold graph with no isolated vertex on the remaining vertices, $\{1, \ldots, n\} - I$, $t(n) = \sum_{i=0}^{n} \binom{n}{i} s(i)$. Observe from Definition 5.1 that a threshold graph with $n \geq 2$ vertices has no isolated vertex if and only if its complement has an isolated vertex. Hence $t(n) = 2s(n)$ for $n \geq 2$. For $n < 2$ we calculate $s(1) = 0$ and $s(0) = t(0) = t(1) = 1$. Combining the two equations for $t(n)$ we obtain, for $n \geq 2$,

$$\begin{aligned}
t(n) &= \binom{n}{0} s(0) + \binom{n}{1} s(1) + \binom{n}{n} s(n) + \sum_{i=2}^{n-1} \binom{n}{i} s(i) \\
t(n) &= 1 + 0 + \frac{t(n)}{2} + \frac{1}{2} \sum_{i=2}^{n-1} \binom{n}{i} t(i) \\
2t(n) &= 2 + t(n) + \sum_{i=2}^{n-1} \binom{n}{i} t(i) \\
t(n) &= 2 + \sum_{i=2}^{n-1} \binom{n}{i} t(i)
\end{aligned}$$

∎

We need to estimate $t(n)$. To do this we need the following lemma which easily follows from the Remainder theorem for Taylor series. We include an elementary proof for simplicity and completeness.

Lemma 5.7 *For all $s \in R$ and $s \in N$, $\sum_{i=1}^{n-2} \frac{s^i}{i!} \geq e^s - 1 - \frac{s^{n-1}}{(n-1)!}e^s$ and $\sum_{i=1}^{n-2} \frac{s^i}{i!} \leq e^s - 1$*

Proof:

$$\begin{aligned}
\sum_{i=1}^{n-2} \frac{s^i}{i!} &= e^s - 1 - \sum_{i=n-1}^{\infty} \frac{s^i}{i!} \\
&= e^s - 1 - \frac{s^{n-1}}{(n-1)!} \sum_{i=n-1}^{\infty} \frac{s^{i-n+1}(n-1)!}{i!} \\
&= e^s - 1 - \frac{s^{n-1}}{(n-1)!} \sum_{j=0}^{\infty} \frac{s^j (n-1)!}{(n+j-1)!} \\
&\geq e^s - 1 - \frac{s^{n-1}}{(n-1)!} \sum_{j=0}^{\infty} \frac{s^j}{j!} \\
&= e^s - 1 - \frac{s^{n-1}}{(n-1)!} e^s
\end{aligned}$$

$$\sum_{i=1}^{n-2} \frac{s^i}{i!} = e^s - 1 - \sum_{i=n-1}^{\infty} \frac{s^i}{i!} \leq e^s - 1.$$

∎

Theorem 5.8

1. $t(n) = O(\frac{n!}{(\ln 2)^n})$.
2. $\lg t(n) = n \lg n - n \lg(\ln 2) + \Theta(\lg n) \geq n \lg n - 0.92n$.

Proof: We prove that, for all $n \geq 2$, $t(n) \geq a\frac{n!}{s^n} + bn$ by constructive (mathematical) induction, deriving values for the constants a, b and s.
Base cases: To satisfy the $n = 2$ cases we need the following constraint on a, b, s

$$a\frac{2}{s^2} + 2b \leq 2;$$

Induction step: We may assume that $n \geq 3$ and that the inequality is true for all natural numbers less than n. Then

$$\begin{aligned}
t(n) &= 2 + \sum_{i=2}^{n-1} \binom{n}{i} t(i) \\
&\geq 2 + \sum_{i=2}^{n-1} \binom{n}{i} \left[a\frac{i!}{s^i} + bi\right] \qquad \text{by the induction hypothesis} \\
&= 2 + a\sum_{i=2}^{n-1} \binom{n}{i} \frac{i!}{s^i} + b\sum_{i=2}^{n-1} \binom{n}{i} i \\
&= 2 + a\sum_{i=2}^{n-1} \frac{n!}{(n-i)!i!} \frac{i!}{s^i} + b\sum_{i=2}^{n-1} n\binom{n-1}{i-1} \\
&= 2 + a\sum_{i=2}^{n-1} \frac{n!}{s^i(n-i)!} + bn\sum_{i=2}^{n-1} \binom{n-1}{i-1} \\
&= 2 + a\sum_{i=1}^{n-2} \frac{n!}{s^{n-i}i!} + bn\sum_{i=1}^{n-2} \binom{n-1}{i} \\
&= 2 + a\frac{n!}{s^n}\sum_{i=1}^{n-2} \frac{s^i}{i!} + bn[2^{n-1} - 2]
\end{aligned}$$

$$\begin{aligned}
&\geq 2 + a\frac{n!}{s^n}\left[e^s - 1 - \frac{s^{n-1}}{(n-1)!}e^s\right] + bn[2^{n-1} - 2] \quad \text{by Lemma 5.7} \\
&= 2 + a\frac{n!}{s^n}[e^s - 1] - ae^s\frac{n}{s} + bn[2^{n-1} - 2] \\
&= 2 + a\frac{n!}{s^n} - 2a\frac{n}{s} + bn[2^{n-1} - 2] \quad \text{setting } e^s - 1 = 1, \text{ or } s = \ln 2 \sim 0.69
\end{aligned}$$

We need a, b such that

$$2 + a\frac{n!}{s^n} - 2a\frac{n}{s} + bn[2^{n-1} - 2] \geq a\frac{n!}{s^n} + bn$$

hence

$$2 - 2a\frac{n}{s} + bn[2^{n-1} - 3] \geq 0$$

This is hardest to satisfy when n is small. Since $n \geq 3$ we only have to satisfy the $n = 3$ case which is

$$2 - \frac{6a}{s} + 3b \geq 0.$$

Arithmetic shows that $a = \frac{s^2}{3} \sim 0.16$ and $b = 0.6$. will satisfy this constraint and the one from the base case.

We prove that, for all $n \geq 2$, $t(n) \leq a\frac{n!}{s^n} + bn$ by constructive (mathematical) induction, deriving values for the constants a, b and s.

Base cases: To satisfy the $n = 2$ cases we need the following constraint on a, b, s

$$a\frac{2}{s^2} + 2b \geq 2;$$

Induction step: We may assume that $n \geq 3$ and that the inequality is true for all natural numbers less than n. Then

$$\begin{aligned}
t(n) &\leq 2 + a\frac{n!}{s^n}\sum_{i=1}^{n-2}\frac{s^i}{i!} + bn[2^{n-1} - 2] \quad \text{similar to algebra in other case} \\
&\leq 2 + a\frac{n!}{s^n}[e^s - 1] + bn[2^{n-1} - 2] \quad \text{by Lemma 5.7} \\
&= 2 + a\frac{n!}{s^n} + bn[2^{n-1} - 2] \quad \text{setting } e^s - 1 = 1, \text{ or } s = \ln 2 \sim 0.69
\end{aligned}$$

We need b such that

$$2 + a\frac{n!}{s^n} + bn[2^{n-1} - 2] \leq a\frac{n!}{s^n} + bn$$

hence

$$2 + bn[2^{n-1} - 3] \leq 0.$$

This is hardest to satisfy when n is small (we will be taking $b < 0$). Since $n \geq 3$ we only have to satisfy the $n = 3$ case which is $2 + 3b \leq 0$. Arithmetic shows that $a = 2.5s^2 \sim 1.2$ and $b = -0.7$. will satisfy this constraint and the one from the base case.

Using Stirling's formula we get: $t(n) = \Theta\left(\sqrt{n}\left(\frac{n}{e\ln 2}\right)^n\right)$. Hence

$\lg t(n) = n\lg n - (\lg(e\ln 2))n + \Theta(\lg n) \geq n\lg n - 0.92n + \Theta(\lg n)$. ∎

Note 5.9 Computer evidence seems to indicate that $t(n) = 0.442695(\frac{n!}{(\ln 2)^n} + O(1)$. Note that 0.442695 looks suspiciously like $(\ln 2)^{-1} - 1$.

Theorem 5.10

1. *If T is an LDT that solves 2-SUM then*

$$\mathrm{ht}(T) \geq n \log n - 0.92n + 0.5 \log n + \Omega(1).$$

2. *If T is a d-ADT that solves 2-SUM then*

$$\mathrm{ht}(T) \geq \frac{0.38n \lg n - 0.96n + 0.19 \lg n}{d}.$$

Proof: This follows from Proposition 2.2 and Theorem 5.8. ∎

Corollary 5.11 *On the LDT model 2-SUM is strictly harder than sorting.*

Proof: By Theorem 5.10 2-SUM requires at least $n \lg n - 0.92n + 0.5 \lg n - 2.30$. Sorting can be done with $n \lg n - 1.329n$ comparisons [FJ59, HL69, Knu73, Man79]. The conclusion follows. ∎

6 The Two List Element Distinctness Problem

The algorithm in Theorem 3.1 can be phrased informally as (1) split the list into two groups X and Y, and (2) see if an element of X is the negation of some element of Y. We can study part (2) in isolation.

Definition 6.1 The *two list element distinctness problem* (TLED henceforth) is the problem of determining membership in

$$TLED_{p,q} = \{[X = (x_1, \ldots, x_p)], [Y = (y_1, \ldots, y_p)] : (\forall i, j)[x_i \neq y_j]\}.$$

Theorem 6.2 *The TLED problem can be solved in $n \lg n - 0.64n + O(\log n)$ comparisons.*

Proof sketch: Assume $p \leq q$. Sort X. For every element of Y look for it in X via binary search. We omit the algebra. ∎

We have a partial result on lower bounds. We try to show that $TLED_{p,q}$ can be expressed as the disjoint union of a certain number of nonempty open sets. We have that number as a recurrence, but not in closed form.

If A and B are sets then $A < B$ means that every element of A is less than every element of B. Consider the set

$$\{(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) : \{x_2, x_4\} < \{y_3, y_4\} < \{x_1\} < \{y_1, y_2\} < \{x_3\}\}.$$

This is an open subset of $TLED_{4,4}$. More generally, $TLED_{p,q}$ can be partitioned into disjoint nonempty open set of this type. Let $c(p, q)$ be the number of such sets that begin with an a subset of $\{x_1, \ldots, x_p\}$. It is easy to show that $c(1, q) = 1$, $c(p, 0) = 1$, and $c(p, q) = \sum_{i=1}^{p} \binom{p}{i} c(q, p - i)$. With some manipulation one can obtain $c(p, q) = \sum_{i=0}^{p-1} \binom{p}{i} c(i + 1, q - 1)$. Using these recurrences and Propostion 2.2 we obtain that a computable lower bound for $TLED_{p,q}$ is $\lg(c(p, q) + c(q, p))$.

7 Open Problems

Several open problems suggest themselves.

1. Obtain tighter upper and lower bounds for 2-SUM for both the LDT and d-ADT models.

2. The *Element Distinctness Problem* (henceforth ED) is the following: given $(x_1, \ldots, x_n)$ determine if there exists $i \neq j$ such that $x_i = x_j$. ED can be solved in $n \lg n - 0.33n + O(1)$ comparisons (first sort then compare adjacent elements). It is know that ED requires $\Omega(n \log n)$ queries on a d-ADT and other models [Ben83, BLY92, GK94, Lop94]. It is easy to show that ED requires $\lg n! = n \lg n - 1.44n + \Omega(1)$ operations on an LDT. It is unknown how ED compares with sorting and 2-SUM. We conjecture that ED is harder than sorting but easier than 2-SUM.

3. The lower bound for 3-SUM using an LDT is $\Omega(n \lg n)$, far from the upper bound of $O(n^2)$. Jeff Erickson has proven a $\Omega(n^2)$ lower bound for 3-SUM [Eri96] if linear comparisons are restricted to be of the form "$ax_i + bx_j + cx_k$ COMP d?". Unfortunately the upper bound of $O(n^2)$ uses 4-ary linear comparisons for which his result does not apply. The open problem here is to obtain better lower bound for 3-SUM on 4-ary LDT's, LDT's, and d-ADT's.

8 Acknowledgements

Sloane's On-line Encyclopedia of Integer Sequences [Slo97] was used to discover the combinatorial relationship between 2-SUM and threshold graphs. Thanks to Richard Stanley, Clara Chan (no relation), and Christos Athanasiadis for providing enumerative combinatorics answers and references. Thanks to Jason Howald and Ashley Reiter for inspired algebra. Thanks to Samir Khuller and David Mount for entertaining the odd idea from time to time. Thanks to Andrew Lee for proofreading.

References

[Ben83] M. Ben-Or. Lower bounds for algebraic computation trees. *Proceedings of the 15th Annual ACM Symposium on the Theory of Computing*, 80-86, 1983.

[BLY92] A. Björner, L. Lovász, and Andrew C.C. Yao. Linear decision trees: Volume estimates and topological bounds. *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, 170-177, 1992.

[BBG94] S. Block, J. Buss, and J. Goldsmith. How hard are n^2-hard problems. *SIGACT News*, 25(2):83–85, 1994.

[DL79] David P. Dobkin and Richard J. Lipton. On the complexity of computations under varying sets of primitives. *Journal of Computer and System Sciences*, 18:86-91, 1979.

[Eri96] Jeffrey G. Erickson. *Lower Bounds for Fundamental Geometric Problems.* Ph.D. Dissertation, UC-Berkeley, 1996.
`http://www.cs.duke.edu:80/~jeffe/pubs/thesis.html`

[ES95] J. Erickson and R. Seidel. Better lower bounds on detecting affine and spherical degeneracies. *Discrete Computational Geometry*, 13:41–57, 1995. Earlier version in FOCS93. Erratum in Disc. Comp. Geom. in 1997.

[FJ59] L. Ford and S. Johnson. A tournament problem. *American Mathematical Monthly*, 66:387-389, 1959.

[GK94] Dima Grigoriev and Marek Karpinski. Lower bound for randomized linear decision tree recognizing a union of hyperplanes in generic position. *Research Report*, 85114-CS, University of Bonn, 1994.

[GO95] Gajen and Overmars. On a class of $O(n^2)$ problems in comp. geom. *Comp. Geom. Theory Appl.*, 5:165-185, 1995.

[HL69] F. Hwang and S. Lin. An analysis of ford-johnson's sorting algorithm. In *Proceedings of the third annual Princeton Conf. on Inform. Sci. and Systems*, pages 292–296, 1969.

[Knu73] Donald E. Knuth. *Sorting and Searching,* vol. 3 of *The Art of Computer Programming*. Addison-Wesley, 1973.

[Lop94] Alex Lopez-Ortiz. New lower bounds for element distinctness on a one-tape turing machine. *Information Processing Letters*, 51:311-314, 1994.
`http://daisy.uwaterloo.ca/~alopez-o/papers.html`

[Man79] G. K. Manacher. The ford-johnson sorting algorithm is not optimal. *Journal of the ACM*, 26(3):441–456, July 1979.

[ORou94] J. O'Rourke. Computational geometry column 22. *SIGACT News*, 25(1):32–33, 1994.

[Slo97] N.J.A. Sloane. *Sloane's On-Line Encyclopedia of Integer Sequences*, AT&T, 1997.
`http://www.research.att.com/~njas/sequences/`

[Stan] Richard P. Stanley. *Enumerative Combinatorics*, vol. 2, Wadsworth and Brooks/Cole, due in 1998.
`http://www-math.mit.edu/~rstan/ec/ec.html`

Twelve countings with rooted plane trees

Martin Klazar

Department of Applied Mathematics of Charles University
Malostranské náměstí 25
118 00 Praha 1
Czech Republic
klazar@kam.ms.mff.cuni.cz

Abstract

The average number of (1) antichains, (2) maximal antichains, (3) chains, (4) infima closed sets, (5) connected sets, (6) independent sets, (7) maximal independent sets, (8) brooms, (9) matchings, (10) maximal matchings, (11) linear extensions, and (12) drawings in (of) a rooted plane tree on n vertices is investigated. Using generating functions we determine the asymptotics and give some explicit formulae and identities. In conclusion we discuss the extremal values of the above quantities and pose some problems.

1 Rooted plane trees

A *rooted plane tree*, a classical enumerative structure, is a quadruple $T = (r, V, E, L)$ such that

- (V, E) is a nonempty finite directed tree, as usual V is the *vertex set* and E is the *edge set*,
- where all edges are directed away from the *root* $r \in V$,
- and $L = \{(\{w : \ vw \in E\}, <_v) : \ v \in V\}$ is a collection of $|V|$ linear orders.

We call the elements of the set $ch(v) = \{w : \ vw \in E\}$ *children* of v, v is their *parent*. A *leaf* is a vertex with no child. Rooted plane trees will be called shortly *trees*. A tree T is visualized by embedding it in the plane (see Figure 1) so that the root is at the lowest position, all edges are straight segments directed up, and the orders $<_v$ coincide with the natural left-right order.

By $\mathcal{T}$ we denote the collection of all substantially different trees and by $\mathcal{T}_n$ the collection of those having n vertices. The aim of the paper is, given a *weight* $w : \mathcal{T} \to \{0, 1, 2, \ldots\}$, to count the total weight $w(n) = \sum_{T \in \mathcal{T}_n} w(T)$ of trees on n vertices. We consider twelve combinatorial weights w and for the first ten of them we determine the *generating function*

$$F_w(x) = \sum_{\mathcal{T}} w(T) x^{|V(T)|} = \sum_{n \geq 1} w(n) x^n.$$

For the eleventh and twelfth weight n stands for $|E|$ and the *exponential generating function* will be determined.

For instance, setting $w(T) = 1$ for all T one gets the celebrated *Catalan function*

$$C = C(x) = \sum_{n \geq 1} |\mathcal{T}_n| x^n = \sum_{n \geq 1} c_{n-1} x^n = \frac{1}{2} \left(1 - \sqrt{1 - 4x}\right) = x + x^2 + 2x^3 + 5x^4 + 14x^5 + 42x^6 + \ldots$$

counting the number of trees on n vertices. $c_n = \frac{1}{n+1}\binom{2n}{n}$ is the nth *Catalan number*. Catalan function satisfies the quadratic equation $C^2 - C + x = 0$.

What are the weights? Mostly the numbers of subsets of V or E with special properties. The first four of them appear by understanding a tree T as a poset. The standard partial ordering $(V, \leq)$ is defined by $u \leq v$ iff u lies on the path joining r and v. A *chain* in T is then a subset $X \subset V$ of pairwise comparable vertices. On the contrary an *antichain* X consists of mutually incomparable vertices. A tree with n vertices may have as many as $2^n - 1$ nonempty chains and as few as $2n - 1$. As for the antichains, there may be as few as n and as many as 2^{n-1} of them. These are extremes but what is going on in average? One would expect that in average antichains are much more numerous than chains, is this really the case? How fast the average numbers grow? Seeking answers to this sort of questions and led by the joy of counting by generating functions we investigated twelve weights of this kind. Our arguments are more or less standard but, except for w_8 and w_{11} which we discuss later, we failed to find any reference to results of this type in [5], [8], and [12], or to localize the sequences $\{w(n)\}_{n\geq 1}$ in [11].

We need to review some more definitions. We say that $X \subset V$ is *infima closed* (in a tree T) if X contains with any two vertices $u, v \in X$ also the merging point of the paths joining r and u, and r and v (i.e., the infimum $u \wedge v$). Six weights arise from graph-theoretical considerations. A set $X \subset V$ is *independent* if $uv \in E$ for no two $u, v \in X$. A set $X \subset V$ is *connected* if any two vertices of X can be joined by an undirected path lying completely in X. A *matching* $X \subset E$ is a set of pairwise disjoint edges. A *broom* $X \subset E$ is a set of pairwise intersecting edges, all directed up. Single vertex is also a broom. Two more weights arise from the concept of drawing trees. Suppose $T = (r, V, E, L)$ is a tree. A *simple drawing* of T is a permutation of edges $(e_1, e_2, \ldots, e_{|E|})$ of T such that $r \in e_1$ and, for any $i = 2, \ldots, |E|$, e_i intersects some of the edges $e_1, e_2, \ldots, e_{i-1}$. A *drawing* of T is a sequence of trees $(T_1, T_2, \ldots, T_n)$, $n = |V|$, such that $T_n = T$ and T_{i-1} arises from T_i by deleting a leaf of T_i.

Now we list the weights. Maximality is meant to inclusion and maximal sets are nonempty by definition. For a given tree T, $w_1(T)$ is the number of nonempty antichains in T, $w_2(T)$ is the number of maximal antichains, $w_3(T)$ is the number of nonempty chains, $w_4(T)$ counts the number of nonempty infima closed sets, $w_5(T)$ counts nonempty connected sets, $w_6(T)$ counts all independent sets (including $\emptyset$), $w_7(T)$ counts maximal independent sets, $w_8(T)$ counts the number of brooms in T, $w_9(T)$ counts matchings (including $\emptyset$), $w_{10}(T)$ counts maximal matchings, $w_{11}(T)$ is the number of simple drawings of T, and $w_{12}(T)$ is the number of drawings of T.

The paper is organized as follows. In the next section we summarize the results — explicit formulae or equations for generating functions, asymptotics — for the first ten weights. In Section 3 we give proofs or sketches of proofs to these results. Applications of the Lagrange inversion formula to the weights w_6, w_7, and w_9 are given in Section 4. In particular, we derive a closed formula for $w_6(n)$. Weights w_{11} and w_{12} are handled in Section 5. In Section 6 we give some concluding comments and open problems, and we determine $\max_{T\in\mathcal{T}_n} w_2(T)$.

2 Subset countings — results

First we list the closed formulae for the generating functions F_1, F_2, F_3, F_4, F_5, and F_8, $F_i(x) = \sum_{n\geq 1} w_i(n)x^n$.

$$F_1(x) = \frac{1 + \sqrt{1-4x} - \sqrt{2}\sqrt{\sqrt{1-4x} + 1 - 10x}}{4} \tag{1}$$

$$F_2(x) = \frac{3 - 2x - \sqrt{1-4x} - \sqrt{2}\sqrt{(1+2x)\sqrt{1-4x} + 1 - 8x + 2x^2}}{4} \tag{2}$$

$$F_3(x) = \frac{x\ \left(1 + 3\sqrt{1-4x}\,\right)}{4\left(1 - \frac{9}{2}x\right)} \quad F_4(x) = \frac{\left(1 + \sqrt{1-4x}\,\right)\left(1 - \sqrt{3 - 2/\sqrt{1-4x}}\,\right)}{4} \tag{3}$$

$$F_5(x) = \frac{x}{x - C^2(x)} F_1(x) = \frac{1}{8}\left(1 + \frac{1}{\sqrt{1-4x}}\right)\left(1 + \sqrt{1-4x} - \sqrt{2}\sqrt{1 + \sqrt{1-4x} - 10x}\,\right) \tag{4}$$

$$F_8(x) = \frac{x}{2(1-4x)} + \frac{x}{2\sqrt{1-4x}} \tag{5}$$

The four functions F_6, F_7, F_9, and F_{10} satisfy the following algebraic equations.

$$F_6^3 - 2F_6^2 + (1+2x)F_6 + x^2 - 2x = 0 \tag{6}$$

$$F_7^4 - 3F_7^3 + (3+x)F_7^2 - (1+x)^2 F_7 - x^3 + x^2 + x = 0 \tag{7}$$

$$F_9^4 - 3F_9^3 + (3+x)F_9^2 - (1+2x)F_9 + x^2 + x = 0 \tag{8}$$

$$\begin{aligned} F_{10}^7 - (6+x)F_{10}^6 + (15+6x)F_{10}^5 + (x^2 - 15x - 20)F_{10}^4 - (2x^2 - 20x - 15)F_{10}^3 - \\ -(15x+6)F_{10}^2 + (2x^2 + 6x + 1)F_{10} + x^4 - x^2 - x = 0 \end{aligned} \tag{9}$$

In the first order asymptotics we use the notation $f(n) \sim g(n)$ for $\lim_{n\to\infty} f(n)/g(n) = 1$.

$$w_1(n) \sim \frac{1}{\sqrt{15\pi}}\ \frac{1}{n\sqrt{n}}\ \left(\frac{25}{4}\right)^n \quad w_2(n) \sim 0.16584\ n^{-3/2}\ (4.80261)^n \quad w_3(n) \sim \frac{1}{9}\ \left(\frac{9}{2}\right)^n \tag{10}$$

$$w_4(n) \sim \frac{5}{16}\ \sqrt{\frac{5}{6\pi}}\ \frac{1}{n\sqrt{n}}\ \left(\frac{36}{5}\right)^n \quad w_5(n) \sim \frac{4}{3}\ w_1(n) \sim \frac{4}{3\sqrt{15\pi}}\ \frac{1}{n\sqrt{n}}\ \left(\frac{25}{4}\right)^n \tag{11}$$

$$w_6(n) \sim \frac{4}{9\sqrt{3\pi}}\ \frac{1}{n\sqrt{n}}\ \left(\frac{27}{4}\right)^n \quad w_7(n) \sim \frac{\sqrt{5731 - 4635/\sqrt{17}}}{256\sqrt{\pi}}\ \frac{1}{n\sqrt{n}}\ \left(\frac{107 + 51\sqrt{17}}{64}\right)^n \tag{12}$$

$$w_8(n) \sim \frac{1}{8}\ 4^n \tag{13}$$

$$w_9(n) \sim \frac{\sqrt{5 - 1/\sqrt{13}}}{4\sqrt{6\pi}}\ \frac{1}{n\sqrt{n}}\ \left(\frac{70 + 26\sqrt{13}}{27}\right)^n \quad w_{10}(n) \sim 0.12075\ n^{-3/2}\ (5.22159)^n \tag{14}$$

The constants in the asymptotics of w_2 and w_{10} are just approximations but, as we shall see in the next section, in principle we can give closed algebraic expressions for them as well. Numerically the asymptotics read as follows. $w_1(n) \sim 0.14567\ n^{-3/2}\ 6.25^n$, $w_2(n) \sim 0.16584\ n^{-3/2}\ 4.80261^n$, $w_3(n) \sim 0.11111\ 4.5^n$, $w_4(n) \sim 0.16095\ n^{-3/2}\ 7.2^n$, $w_5(n) \sim 0.19423\ n^{-3/2}\ 6.25^n$, $w_6(n) \sim 0.14477\ n^{-3/2}\ 6.75^n$,

$w_7(n) \sim 0.14958\ n^{-3/2}\ 4.95747^n$, $w_8(n) \sim 0.125\ 4^n$, $w_9(n) \sim 0.12514\ n^{-3/2}\ 6.06460^n$, $w_{10}(n) \sim 0.12075\ n^{-3/2}\ 5.22159^n$. A remarkable fact is that all the ten linear constants lie in the interval $(0.1, 0.2)$.

In the left table below we list the first eight values $w_i(n), n = 1, 2, \ldots, 8$, for each $i = 1, 2, \ldots, 10$. For this and other heavy calculations we used MATHEMATICA and MAPLE. For $i = 1, 2, 3, 4, 5, 8$ we took directly the generating function. For $i = 6, 7, 9, 10$ we started with $F_i(0) = 0$ and then, differentiating the equation, we applied the relations $w_i(n) = F_i^{(n)}(0)/n!$. For $i = 6, 7, 9$ one can apply alternatively the Lagrange inversion formula — see Section 4. In the right table we sort the weights by their exponential growth rates.

w_1	1	2	7	19	131	625	3099	15818
w_2	1	2	5	15	50	178	663	2553
w_3	1	3	12	51	222	978	4338	19323
w_4	1	3	13	63	326	1769	9964	57843
w_5	1	3	12	52	236	1109	5366	26639
w_6	2	3	10	42	198	1001	5304	29070
w_7	1	2	4	13	44	164	636	2559
w_8	1	3	11	42	163	638	2510	9908
w_9	1	2	6	23	98	447	2134	10530
w_{10}	1	1	4	12	44	175	718	3052

w_8	brooms	4^n
w_3	chains	4.5^n
w_2	max. antichains	4.80261^n
w_7	max. ind. sets	4.95747^n
w_{10}	max. matchings	5.22159^n
w_9	matchings	6.06460^n
w_1	antichains	6.25^n
w_5	connected sets	6.25^n
w_6	independent sets	6.75^n
w_4	infima closed sets	7.2^n

We conclude the section with a few comments. Note the relation between w_1 and w_5. From (5) it follows at once a closed formula for $w_8(n)$, see (24). In Section 4 we derive a closed formula (29) for $w_6(n)$ and a nice recurrent formula (30) for $w_7(n)$. Expressions and equations (1)–(9) yield effective procedures calculating for a given n the numbers $w_i(n)$, $1 \leq i \leq 10$. A natural question is whether one can calculate effectively, given a tree T, the numbers $w_i(T)$. This turns out to be possible for each of the weights, in the next section we give the corresponding recurrent relations.

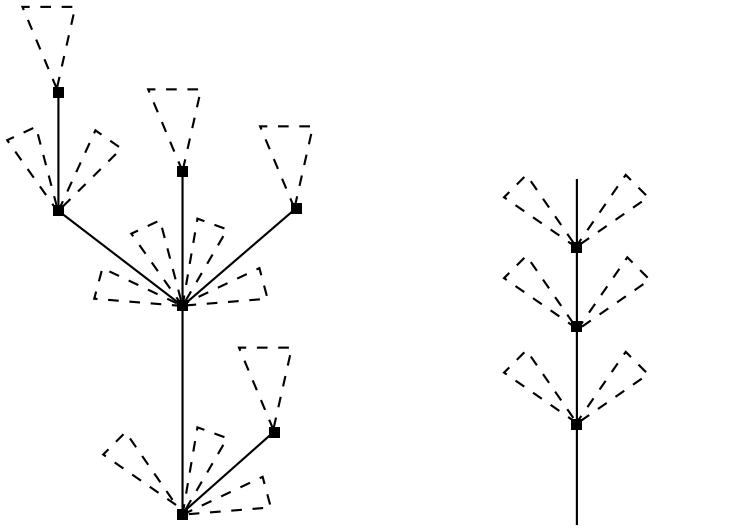
Thus, indeed, the average tree has asymptoticly much more antichains than chains in spite the tendency shown by the first nine values. For $n \geq 10$ we have, in accordance with the asymptotics, $w_1(n) > w_3(n)$. Even maximal antichains beat asymptoticly chains but now $w_2(n) < w_3(n)$ for $n = 2, 3, \ldots, 99$. Only from 100 vertices on the asymptotics prevails and the average tree starts to have more maximal antichains than chains.

3 Subset countings — proofs

Let $T = (r, V, E, L)$ be a tree and $v \in V$ be a vertex. A *subtree* T_v of T rooted in v is the subtree spanned by the upset $\{x \in V : \ x \geq v\}$. A *degree* $deg(v)$ of v is the number $|ch(v)|$ of children of v. A *principal subtree* of T is a subtree T_v such that $v \in ch(r)$. T is determined uniquely by the list $ps(T) = (T_v : \ v \in ch(r))$ of its principal subtrees. A *singleton* s is the trivial one vertex tree. Let us remind the Catalan function C satisfying $C^2 - C + x = 0$, see Section 1.

To determine the generating function F_w we use arguments of two kinds. In the *recurrence argument* we take the decomposition $ps(T) = (T_1, T_2, \ldots, T_k)$ and find, for a weight w, the recurrent relation that transforms the list $(w(T_1), w(T_2), \ldots, w(T_k))$ into the number $w(T)$. The relation can be often translated to an equation for F_w. This way we obtain both the *individual count* (the recurrence for $w(T)$) and the *collective count* (the function F_w that counts $w(n)$). An alternative approach via another decomposition is indicated in the concluding section.

The *extension argument* is basically counting in two ways. We count the number of extensions of a fixed set $X \subset V$ with a special property to a tree. See Figure 1. Draw a tree $T = (r, V, E, L)$ in the plane. The *gaps* of $v \in V$ are the wedge-shaped areas into which the edges incident with v split v's neighborhood. Thus v has $deg(v) + 1$ gaps. All gaps of all vertices form the set $g(T)$ with $2|V| - 1$ elements. In the *gap extension* we take a tree $T \in \mathcal{T}_m$ and into each gap $g \in g(T)$ we insert a tree T_g. The root $r(T_g)$ and the vertex of g are identified. A moment of thought reveals that the number of choices for which a tree from $\mathcal{T}_n$ arises is the coefficient at x^n in $x^m(C(x)/x)^{2m-1}$. In the *edge extension* we mark on a fixed oriented edge $e \in \mathcal{T}_2$ from top to bottom $k \geq 0$ points $p_1, \ldots, p_k$ and we put a tree T_i to the left and a tree U_i to the right of p_i, identifying p_i with the roots $r(T_i)$ and $r(U_i)$. A tree from $\mathcal{T}_n$ (we do not count the endpoints of e) is obtained for $[x^n] \sum_{k\geq 0}(C^2/x)^k = [x^n]\, x/(x - C^2)$ choices. Here and further on $[x^n]\, f$ denotes the coefficient at x^n in the power series f. In the *l edges extension* we extend this way independently l edges. While saying nothing about the individual count this method is usually more elegant than the recurrence argument.



gap extension edge extension

Figure 1: Extensions.

1 Antichains by extension. Consider an antichain $X \subset V(T)$ and the tree T^* spanned by the downset $\{v \in V(T) :\ v \leq x \in X\}$. Obviously T is a gap extension of T^* and therefore

$$F_1(x) = \sum_{m\geq 1} c_{m-1} x^m \left(\frac{C(x)}{x}\right)^{2m-1} = \frac{x}{C(x)} \sum_{m\geq 1} c_{m-1} \left(\frac{C^2(x)}{x}\right)^m = \frac{C(C^2(x)/x)}{C(x)/x}.$$

The rest is a matter of simplifications.

Antichains by recurrence. For singleton we have $w_1(s) = 1$. For a nonsingleton T with $ps(T) = (T_1, T_2, \ldots, T_k)$ we have the recurrence

$$w_1(T) = \prod_{i=1}^{k} (1 + w_1(T_i)) \tag{15}$$

whose proof is immediate. It translates to $F_1 = x \sum_{k\geq 0}(F_1 + C)^k = x/(1 - F_1 - C)$ which simplifies to $F_1^2 + (C - 1)F_1 + x = 0$. Solving this we get again the formula (1).

2 Maximal antichains by recurrence. Similarly to (15) we get $w_2(s) = 1$ and, $ps(T) = (T_1, T_2, \ldots, T_k)$,

$$w_2(T) = 1 + \prod_{i=1}^{k} w_2(T_i). \tag{16}$$

This translates to $F_2 = C + x\sum_{k\geq 1} F_2^k = C + xF_2/(1-F_2)$, i.e. to $F_2^2 + (x - C - 1)F_2 + C = 0$. The quadratic formula yields (2).

3 Chains by extension. Consider a chain $X = (x_1, \ldots, x_m) \subset V$ in T and think of the x_{i-1}–x_i path as an edge, $i = 1, \ldots, m$, $x_0 = r$. Then T is a gap extension and m edges extension of X. Hence

$$F_3(x) = \sum_{m\geq 1} x^m \left(\frac{C(x)}{x}\right)^{2m-1} \left(\frac{x}{x - C^2(x)}\right)^m = \frac{xC(x)}{x - 2C^2(x)}.$$

After further simplifications we obtain the formula for F_3 in (3).

Chains by recurrence. The recurrence for chains is $w_3(s) = 1$, $ps(T) = (T_1, T_2, \ldots, T_k)$,

$$w_3(T) = 1 + 2\sum_{i=1}^{k} w_3(T_i). \tag{17}$$

Consider the generating function

$$G(x, y) = \sum_{\mathcal{T}} x^{w_3(T)} y^{|V(T)|}.$$

Then (17) reads as

$$G(x, y) = xy \sum_{k\geq 0} G(x^2, y)^k = \frac{xy}{1 - G(x^2, y)}.$$

Clearly $G(1, y) = C(y)$ and $F_3(y) = G_x(1, y)$. Taking the partial derivative by x of the equation for G and evaluating it at $(1, y)$ we find

$$F_3(y) = y\frac{1 - C(y) + 2F_3(y)}{(1 - C(y))^2} \quad \text{that solves as } F_3(y) = y\frac{1 - C(y)}{(1 - C(y))^2 - 2y}.$$

Simplifications lead again to the formula in (3).

4 Infima closed sets by extension. Consider a nonempty infima closed set $X \subset V(T)$, $|X| = m$. By replacing all u–v paths, $u, v \in X$, not containing other vertices of X by an edge we produce a tree T^* on m vertices. Clearly T is a gap and m edges extension of T^*, in the same way as for chains. Only now we are extending all trees on m vertices, not only the path. Thus

$$F_4(x) = \sum_{m\geq 1} c_{m-1} x^m \left(\frac{C(x)}{x}\right)^{2m-1} \left(\frac{x}{x - C^2(x)}\right)^m = \frac{x}{C(x)}\, C\left(C^2(x)/(x - C^2(x))\right).$$

Simplifications lead to the formula in (3).

For the sake of completeness we mention the recurrent formula. Let $ps(T) = (T_1, \ldots, T_k)$. Then $w_4(s) = 1$,

$$w_4(T) = \sum_{i=1}^{k} w_4(T_i) + \prod_{i=1}^{k} (1 + w_4(T_i)). \tag{18}$$

5 Connected sets by extension. Consider a connected set $X \subset V$. It is easy to see that T is a gap and (one) edge extension of X. The edge corresponds to the path $r(T)$–$r(X)$. Thus the additional factor $x/(x - C^2(x))$ in (4) compared to antichains.

Again, given a T, we can effectively calculate $w_5(T)$:

$$w_5(T) = \sum_{v \in V} w_1(T_v) \tag{19}$$

where T_v is the subtree rooted in v.

6 Independent sets by recurrence. We need an auxiliary weight $z(T)$ counting $\emptyset$ and the independent sets in T not containing r. Let $ps(T) = (T_1, \ldots, T_k)$. A moment of thought reveals that $z(s) = 1$, $w_6(s) = 2$,

$$z(T) = \prod_{i=1}^{k} w_6(T_i) \text{ and } w_6(T) = \prod_{i=1}^{k} w_6(T_i) + \prod_{i=1}^{k} z(T_i). \tag{20}$$

Translated to generating functions,

$$F_z = x \sum_{k \geq 0} F_6^k = \frac{x}{1 - F_6} \text{ and } F_6 = x \sum_{k \geq 0} F_z^k + x \sum_{k \geq 0} F_6^k = \frac{x}{1 - F_z} + \frac{x}{1 - F_6}. \tag{21}$$

Eliminating F_z from the system we get the cubic equation (6).

7 Maximal independent sets by recurrence. So far we always calculated the number at a vertex from the numbers at its children, now we need to consider also the numbers at grandchildren. We define two auxiliary weights t and q. Let $t(T) = \#$ of ind. sets in T not containing r which are maximal or extendable only by the root r. Further $q(s) = 1$, and $q(T) = t(T_1)t(T_2)\ldots t(T_k)$ where $ps(T) = (T_1, \ldots, T_k)$. Then $w_7(s) = t(s) = q(s) = 1$ and, $ps(T) = (T_1, \ldots, T_k)$,

$$t(T) = \prod_{i=1}^{k} w_7(T_i) \text{ and } w_7(T) = \prod_{i=1}^{k} t(T_i) + \prod_{i=1}^{k} w_7(T_i) - \prod_{i=1}^{k} (w_7(T_i) - q(T_i)). \tag{22}$$

The first equality is easy — to take an r-free ind. set in T extendable at most by r is the same as to take a max. ind. set in each T_i. In the second equality in (22) we count first by the product $\prod t(T_i)$ the number of max. ind. sets containing the root. To take a max. ind. set in T not containing r is the same as to take a max. ind. set in each T_i, not all of them avoiding $r(T_i)$. There are $q(T_i)$ max. ind. sets in T_i containing $r(T_i)$. This gives the rest of the second equation. (22) expressed in generating functions is

$$F_t = \frac{x}{1 - F_7} \text{ and } F_7 = \frac{x}{1 - F_t} + \frac{x}{1 - F_7} - \frac{x}{1 - F_7 + x/(1 - F_t)} \tag{23}$$

because the generating function corresponding to q is $x/(1 - F_t)$. The elimination of F_t yields the quartic (7).

8 Brooms by extension. Fix a broom B with m vertices in a tree T. T is a gap extension and one edge extension (as for connected sets) of B and therefore

$$F_8(x) = \frac{x}{x - C^2(x)} \sum_{m \geq 1} x^m \left(\frac{C(x)}{x}\right)^{2m-1} = \frac{x^2}{C.(x - C^2)} \frac{C^2/x}{1 - C^2/x} = \frac{x^2 C}{(x - C^2)^2} = \frac{x^2 C}{(2x - C)^2} =$$

$$= \frac{1}{1 - 4x} \frac{x^2 C}{C - x} = \frac{x}{1 - 4x} \frac{x}{C} = \frac{x}{1 - 4x} \frac{1 + \sqrt{1 - 4x}}{2} = \frac{x}{2(1 - 4x)} + \frac{x}{2\sqrt{1 - 4x}}.$$

It is easy to extract the coefficient by the binomial formula. On the other hand clearly $w_8(T) = \sum_{v \in V} 2^{deg(v)}$ and we have the identity

$$w_8(n) = \sum_{T \in \mathcal{T}_n} \sum_{v \in V(T)} 2^{deg(v)} = \frac{4^{n-1} + \binom{2n-2}{n-1}}{2}. \tag{24}$$

In our derivation we used only that for any $m \geq 1$ there is exactly one broom on m vertices. Thus more generally:

Theorem 3.1 *Suppose $\mathcal{S} \subset \mathcal{T}$ is a family of trees such that $|\mathcal{S} \cap \mathcal{T}_n| = 1$ for any $n \geq 1$. Let $w(T)$ count the total number of ways to embed a member of $\mathcal{S}$ into T. Then $w(n) = \sum_{T \in \mathcal{T}_n} w(T) = w_8(n) = (4^{n-1} + \binom{2n-2}{n-1})/2$.*

If $\mathcal{S}$ is the family of all paths we obtain the identity

$$\sum_{T \in \mathcal{T}_n} |\{(u,v) \in V(T) \times V(T) : \ u \text{ and } v \text{ are comparable in } T\}| = 4^{n-1}$$

because the left hand side is $2w_8(n) - nc_{n-1}$. We remark that a quantity similar to w_8, namely the average vertex altitude, was counted by D. E. Knuth, see [8].

9 Matchings by recurrence. We set $z(T)$ to be the number of matchings in T not covering the root, the empty set included. Let $ps(T) = (T_1, \ldots, T_k)$. Then $z(s) = w_9(s) = 1$,

$$z(T) = \prod_{i=1}^{k} w_9(T_i) \text{ and } w_9(T) = \prod_{i=1}^{k} w_9(T_i).\left(1 + \sum_{i=1}^{k} \frac{z(T_i)}{w_9(T_i)}\right). \tag{25}$$

The first relation follows from the fact that a matching in T avoiding r arises simply by taking in each T_i either a matching or the empty set. In the second relation we add the numbers of matchings using the edge $r(T)r(T_i)$. To translate this to generating functions we use the identity $\sum_{k \geq 0}(k+1)x^k = 1/(1-x)^2$. Thus

$$F_z = \frac{x}{1-F_9} \text{ and } F_9 = \frac{x}{1-F_9} + \frac{xF_z}{(1-F_9)^2}.$$

Eliminating F_z we obtain the quartic equation (8).

10 Maximal matchings by recurrence. From technical reasons we set $w_{10}(s) = 1$. Consider two auxiliary weights z and q. $z(s) = 0$ and $z(T)$ counts the number of max. matchings in T covering the root, $q(s) = 1$ and $q(T) = w_{10}(T_1)w_{10}(T_2)\ldots w_{10}(T_k)$ where $ps(T) = (T_1, \ldots, T_k)$. Then $z(s) = 0$ and $q(s) = w_{10}(s) = 1$,

$$z(T) = \prod_{i=1}^{k} w_{10}(T_i).\sum_{i=1}^{k} \frac{q(T_i)}{w_{10}(T_i)} \text{ and } w_{10}(T) = z(T) + \prod_{i=1}^{k} z(T_i). \tag{26}$$

In the first relation we count the number of max. matchings using the edge $r(T)r(T_i)$. Those arise by taking a max. matching in each $T_j, j \neq i$, (or $\emptyset$ if $T_j = s$, that's why we set $w_{10}(s) = 1$) and an $r(T_i)$-free matching in T_i (or $\emptyset$ if $T_i = s$) extendable eventually only by some edge going up from $r(T_i)$. Such matchings are counted by $q(T_i)$. In the second relation we add to $z(T)$ the number of max. matchings avoiding $r(T)$. Algebraicly,

$$F_z = \frac{xF_q}{(1-F_{10})^2} \text{ and } F_{10} = F_z + \frac{x}{1-F_z} \text{ where } F_q = \frac{x}{1-F_{10}}.$$

From this one obtains the relation $F_{10} = x^2/(1-F_{10})^3 + x/(1-x^2/(1-F_{10})^3)$ which simplifies to the equation of degree 7 in (9).

The asymptotics of the numbers $\mathbf{w_1(n), \ldots, w_{10}(n)}$. We start with the simple cases and proceed to more complicated ones. Catalan numbers have the asymptotics

$$c_n \sim \frac{4^n}{n\sqrt{\pi n}}. \tag{27}$$

This follows by Stirling formula.

$\mathbf{w_8(n)}$. The asymptotics (13) for $w_8(n)$ is immediate from (24).

When F_i is given by square roots the next theorem of Bender, p. 496 in [2], is useful. We need also binomial and Stirling formulae and basic concepts of analytic functions.

Theorem 3.2 *Let $A(x) = \sum a_n x^n$, $B(x) = \sum b_n x^n$, and $C(x) = A(x)B(x) = \sum d_n x^n$ be three power series, and let A and B have radii of convergence $\alpha > \beta \geq 0$. Suppose $b_{n-1}/b_n \to \beta$ as $n \to \infty$, and $A(\beta) \neq 0$. Then*

$$d_n \sim A(\beta) b_n.$$

$\mathbf{w_3(n)}$. For $F_3(x)$ we use Theorem 3.2 with $A(x) = x(1 + 3\sqrt{1-4x})/4$, $B(x) = 1/(1 - 9x/2)$, $\alpha = 1/4$, $\beta = 2/9$, and $A(2/9) = 1/9$. The asymptotics (10) for $w_3(n)$ follows.

$\mathbf{w_4(n)}$. To obtain the asymptotics (11) for $w_4(n)$ we write $F_4(x) = (1 + \sqrt{1-4x})/4 - A(x)B(x)$ where

$$A(x) = \frac{\sqrt{5}}{4} \frac{1 + \sqrt{1-4x}}{\sqrt{(3\sqrt{1-4x} + 2)\sqrt{1-4x}}} \text{ and } B(x) = \sqrt{1 - \frac{36x}{5}}.$$

Theorem 3.2 is applied with $\alpha = 1/4$, $\beta = 5/36$, and $A(5/36) = (5/8)\sqrt{5/6}$. The coefficient b_n in $B(x) = \sum b_n x^n = (1 - 36x/5)^{1/2}$ can be estimated by means of binomial and Stirling formulae.

$\mathbf{w_1(n)}$. We observe that the expresion under the big radical in (1) determines a function that is analytic in the 1/4 circle and that is nonzero there except for the simple zero 4/25. Thus we can write $F_1(x) = (1 + \sqrt{1-4x})/4 - A(x)B(x)$ with $B(x) = \sqrt{1 - 25x/4}$ and $A(x)$ a function analytic in the 1/4 circle. Further, $A(4/25) = 2/\sqrt{15}$. Theorem 3.2 implies the first asymptotics in (10).

$\mathbf{w_5(n)}$. Here $A(x) = (1/2)(1 + 1/\sqrt{1-4x})$, $B(x) = F_1(x)$, $\alpha = 1/4$, $\beta = 4/25$, and $A(4/25) = 4/3$. The second asymptotics in (11) follows.

$\mathbf{w_2(n)}$. The expression under the big radical in (2) is analytic in the 1/4 circle and is nonzero there except for the simple zero $\beta = 0.20821\ldots$ (the only real root of $x^3 - 4x^2 + 20x - 4$). Thus we have again $F_5(x) = (3 - 2x - \sqrt{1-4x})/4 - A(x)B(x)$ with $B(x) = \sqrt{1 - x/\beta}$ and $A(x)$ a function analytic in the 1/4 circle. One can calculate that

$$A(\beta) = \sqrt{\frac{\beta}{2}} \sqrt{\frac{3\beta}{\sqrt{1-4\beta}} - \beta + 2}.$$

The second asymptotics in (10) is obtained.

To resolve the remaining cases when F_i satisfies an equation of degree > 2 we use the following result, found on p. 502 in [2].

Theorem 3.3 *A power series $f(x) = \sum a_n x^n$ with nonnegative coefficients satisfying $F(x, f(x)) = 0$ and two real numbers $\alpha > 0$ and $\beta > a_0$ are given. Suppose that*

(a) *for some $\delta > 0$, $F(x, y)$ is analytic whenever $|x| < \alpha + \delta$, $|y| < \beta + \delta$,*

(b) *$F(\alpha, \beta) = F_y(\alpha, \beta) = 0$,*

(c) *$F_x(\alpha, \beta) \neq 0$ and $F_{yy}(\alpha, \beta) \neq 0$, and*

(d) *if (κ, λ) is another solution of the system in (b) then $|\kappa| > \alpha$ or $|\lambda| > \beta$.*

Then

$$a_n \sim \sqrt{\frac{\alpha F_x(\alpha, \beta)}{2\pi F_{yy}(\alpha, \beta)}} \frac{1}{n\sqrt{n}} \left(\frac{1}{\alpha}\right)^n. \tag{28}$$

This is exactly what we need but the difficulty is that the theorem is incorrect, as pointed out by Canfield [3]. However, the conclusion (28) still holds if we can present positive reals (α, β), $f(\alpha) = \beta$, such that (01) (α, β) lies inside the analyticity domain of F (i.e., (a) holds), (02) the condition (c) holds, (03) α is the radius of convergence of $f(x)$, and (04) $f(x)$ has no other singularity on the boundary than α.

We know, by implicit function theorem, that the pair (α, β) we look for (as well as and any other singularity on the boundary) is hidden among the solutions of the simultaneous equations (b). In general it may be difficult to determine which solution is the right one or even to find all solutions. Therefore several conditions for F making (α, β) unique or localizing it among the solutions were proposed, see [9] and [10], p. 1162–3.

For the four functions F_6, F_7, F_9, and F_{10} we can always find (α, β) meeting the conditions (01)–(04). Indeed, $F(x, y)$ is a bivariate polynomial, thus analytic everywhere, and it is not too difficult to find all solutions of the algebraic system (b). Notice that $c_{n-1} \le w_i(n) \le 2^n c_{n-1}$. By (27) we know that the radius of convergence of any $F_i(x)$, $i = 1 \dots 10$, lies in $[1/8, 1/4]$. In all four cases there is only one (complex) solution (α, β) such that $1/8 \le |\alpha| \le 1/4$. Thus (01)–(04) holds and (28) is true.

$\mathbf{w_6(n)}$. $F_6(x)$ satisfies the cubic equation (6). The system (b) has four solutions: $(0, 1)$ (with multiplicity 3) and $(\alpha, \beta) = (4/27, 5/9)$. Plugging in the formula (28) we obtain the first bound in (12).

$\mathbf{w_7(n)}$. The equation for $F_7(x)$ is given by (7). The solutions of (b) are: $(0, 1)$ (with multiplicity 4), $((-51\sqrt{17} - 107)/512, (33 - 7\sqrt{17})/128)$, and $(\alpha, \beta) = ((51\sqrt{17} - 107)/512, (33 + 7\sqrt{17})/128)$. The second bound in (12) follows.

$\mathbf{w_9(n)}$. The equation for $F_9(x)$ is (8). The solutions of (b) are: $(0, 1)$ (multiplicity 2), $((-13\sqrt{13} - 35)/72, (1 - \sqrt{13})/12)$, and $(\alpha, \beta) = ((13\sqrt{13} - 35)/72, (1 + \sqrt{13})/12)$. The first bound in (14) follows.

$\mathbf{w_{10}(n)}$. $F_{10}(x)$ satisfies (9). The system (b) has 12 solutions: $(0, 1)$ (multiplicity 8), $(-0.26689 \pm 0.51782i, 0.01231 \pm 0.40950i)$, $(11.67188, 8.47407)$, and $(\alpha, \beta) = (0.19151, 0.38840)$. The four y solutions different from 1 are roots of the quartic $248y^4 - 2204y^3 + 912y^2 - 389y + 137$. x appears in $F_{yy}(x, y) = 0$ only in the second degree. Thus α and β still express in radicals. The second bound in (14) follows.

4 Applications of the LIF

The generating functions F_6, F_7, F_9, and F_{10} satisfy an algebraic equation of degree > 2. Such an equation is often very hard, if not impossible, to solve explicitly. Nevertheless, sometimes we can find easily the inverse to the solution. Then the *Lagrange inversion formula* applies.

Theorem 4.1 (LIF) *Suppose $f(x)$ is a power series with $[x^0]f = 0$ and $[x^1]f \neq 0$. Then*

$$[x^n]f(x)^{<-1>} = n^{-1}[x^{n-1}](f(x)/x)^{-n}.$$

For more details see [14], [10] (p. 1106), and [7] (p. 1032).

Theorem 4.2 *Let $n \ge 1$. Recall that $w_6(n)$ is the total number of all independent sets in all $T \in \mathcal{T}_n$ (the empty set counted) and $z(n)$ is the number of those avoiding the root. Then*

$$w_6(n) = \frac{1}{n-1}\binom{3n-3}{n} \text{ and } z(n) = \frac{1}{n}\binom{3n-2}{n-1}. \tag{29}$$

Proof. We start with $z(n)$. Eliminating F_6 from (21) we obtain $F_z(1 - F_z)^2 = x$. Thus $F_z(x)^{<-1>} = x(1 - x)^2$. The formula for $z(n)$ follows readily by the LIF.

To determine $w_6(n)$ we observe that

$$3xF_6' - 2F_6 - 4xF_z' + 2F_z = 0.$$

This is not difficult to check by means of the relations (21). We leave the straightforward calculations to the reader as an exercise. In terms of coefficients:

$$(3n-2)w_6(n) = (4n-2)z(n).$$

Substituting the formula for $z(n)$ we finish the proof. □

Theorem 4.3 *Let $n \geq 1$. Recall that $w_7(n)$ is the total number of all maximal independent sets in all $T \in \mathcal{T}_n$ and $t(n)$ is the number of independent sets avoiding the root and extendable at most by it. Then*

$$t(n) = \frac{1}{n}\sum_{k=0}^{n-1}(-1)^k\binom{n+k-1}{k}\binom{3n-k-2}{n-k-1} = \frac{1}{n}\sum_{k=0}^{\lfloor (n-1)/2\rfloor}\binom{2n-2-2k}{n-1-2k}\binom{n+k-1}{k}$$

and

$$w_7(n) = t(n+1) - \sum_{k=2}^{n} t(k).w_7(n-k+1). \tag{30}$$

Proof. Eliminating F_7 from (23) we find that $F_t(1-F_t)(1-F_t^2) = F_t(1+F_t)(1-F_t)^2 = x$. Thus $F_t(x)^{<-1>} = x(1-x)(1-x^2) = x(1+x)(1-x)^2$. The LIF yields the formula for $t(n)$. The recurrence for $w_7(n)$ follows from the relation $F_t(1-F_7) = x$. □

As to the values of w_9, the LIF helps here too. $F_9(x)^{<-1>}$ is easily found by solving (8) for x. We obtain a more comfortable way to calculate $w_9(n)$ (instead of taking derivatives) but no nice explicit formula seems to arise here. The details are omitted. We did not succed in applying the LIF to w_{10}.

5 Drawing countings

The calculations for the weights w_{11} and w_{12} are more elegant when the main parameter n is $|E|$ rather than $|V|$. We use exponential instead of ordinary generating function. We determine

$$F_i(x) = \sum_{n\geq 0}\frac{w_i(n)}{n!}\,x^n$$

where $i = 11, 12$ and in $w_i(n) = \sum_T w_i(T)$ we sum over the trees with n edges.

A simple drawing $(e_1, e_2, \ldots, e_n)$ of a tree T with n edges is a way of planting T from the root. To look on it differently consider the vertices $(v_1, v_2, \ldots, v_n)$ where v_i is the endpoint of e_i. Obviously, $(r, v_1, \ldots, v_n)$ is a linear extension of the tree as a poset. And vice versa, any linear extension determines a simple drawing of T. Thus $w_{11}(T)$ is the number of linear extensions of T. This notion and the results below (Theorems 5.1 and 5.2) seem be frequently rediscovered, as we learned after proving the theorems.

Theorem 5.2 is close in statement and proof to Lemma 2.1 in [1]. Theorem 5.1 is proved, in a more complicated manner, in [13]. Another proof of Theorem 5.1, much the same as the one below, can be found in [6]. There the authors point to the thesis [4] as to an older reference for this result and mention that R. P. Stanley proved it before as well. We join in and include, for the readers convenience, our (independent) proofs. As to the notation, $(2n-1)!!$ stands, as usual, for $1.3.5\ldots(2n-1)$. For triple and quadruple factorials see [6]!!

Theorem 5.1 *Let $n > 0$. Then*

$$w_{11}(0) = 1,\ w_{11}(n) = (2n-1)!!\ \textit{and}\ F_{11}(x) = \frac{1}{\sqrt{1-2x}}. \tag{31}$$

Proof. So $w_{11}(T)$ counts the labelings of vertices by $0, 1, \ldots, n$ such that the label of u is smaller than that of v whenever $u < v$. Thus r is always labeled by 0. Clearly $w_{11}(0) = 1$. For $T \in \mathcal{T}_{n+1}$, $n \geq 1$, in any of the labelings n sits at a leaf l and deleting l we get a proper labeling of a $T^* \in \mathcal{T}_n$. From each labeled T^* we can get, adding l back, exactly $2n-1$ different labeled T's since each T^* has $2n-1$ gaps to place l. Hence $w_{11}(n) = (2n-1).w_{11}(n-1)$ and we obtain the first formula in (31). The second formula follows from the first one after rewriting $(2n-1)!!$ as $n!\binom{2n}{n}/2^n$. □

The asymptotics

$$w_{11}(n) \sim \sqrt{2}\ \left(\frac{2n}{e}\right)^n$$

follows by Stirling formula.

We show now how to perform for w_{11} the individual count.

Theorem 5.2 *Recall that T_v stands for the subtree of T rooted in $v \in V$. We abbreaviate $|V(T_v)|$ by $|T_v|$. Then, for a tree T with $|V| = n+1$ vertices,*

$$w_{11}(T) = \frac{(n+1)!}{\prod_{v \in V} |T_v|} = \frac{n!}{\prod_{v \in V, v \neq r} |T_v|}. \tag{32}$$

Proof. By induction on the height of T. Clearly $w_{11}(s) = 1$. For a nonsingleton tree T with $ps(T) = (T_1, T_2, \ldots, T_k)$ we have

$$w_{11}(T) = \binom{n}{|T_1|\ |T_2| \ldots |T_k|} \prod_{i=1}^{k} w_{11}(T_i)$$

because for each of the choices $\{1, 2, \ldots, n\} = X_1 \cup X_2 \cup \ldots \cup X_k$, $|X_i| = |T_i|$, X_i mutually disjoint, of the sets of labels for vertices $V(T_i)$ (r is labeled by 0) we have exactly $\prod w_{11}(T_i)$ labelings. Plugging in the formulae for $w_{11}(T_i)$ and canceling the factorials we get (32). □

The counting of $w_{12}(n)$ is more interesting. Note that $w_{12}(T)$ counts different ways to plant T from its root too but "different" has other meaning compared to w_{11}. For instance, if T_0 is the V-shaped tree on 5 vertices then $w_{11}(T_0) = 6$ but $w_{12}(T_0) = 4$. The key fact is that the insertion of a new leaf in T in different gaps may produce the same tree. More precisely:

Lemma 5.3 *Suppose T has $n \geq 1$ edges and l leaves. Adding the new leaf in all $2n+1$ gaps yields $2n+1-l$ new different trees with $n+1$ edges, l of them have l leaves and $2n+1-2l$ have $l+1$ leaves.*

Proof. Consider the trees $X = \{T_g :\ g \in g(T)\}$ where T_g arises by adding the new leaf in the gap g. T_g and T_h coincide iff g and h share the same vertex v and all edges between g and h going up from v lead to leaves. Thus $|X| = 2n+1-c$ where c is the number of gaps whose left edge leads to a leaf. Clearly $c = l$. The number of leaves does not change iff we add the new leaf to a leaf and then we produce l new trees. Otherwise the number of leaves increases by one. □

Theorem 5.4

$$F_{12}(x) = \sum_{\mathcal{T}} \frac{w_{12}(T)}{|E(T)|!} x^{|E(T)|} = \sum_{n \geq 0} \frac{w_{12}(n)}{n!} x^n = \frac{1}{\sqrt{2e^{-x} - 1}}. \tag{33}$$

Proof. Consider the bivariate exp. gen. function ($l(T)$ is the number of leaves of T)

$$F^*(x,y) = \sum_{T \in \mathcal{T}} \frac{w_{12}(T)}{|E(T)|!} x^{|E(T)|} y^{l(T)} = 1 + xy + \frac{x^2 y}{2} + \frac{x^2 y^2}{2} + \dots$$

Lemma 5.3 translates to generating functions as

$$\int_x (y \frac{\partial}{\partial y} + 2xy \frac{\partial}{\partial x} + y - 2y^2 \frac{\partial}{\partial y}) F^* = F^* - 1.$$

This yields the partial differential equation

$$(\frac{1}{y} - 2x) \frac{\partial F^*}{\partial x} + (2y - 1) \frac{\partial F^*}{\partial y} = F^*. \tag{34}$$

(34) is of the type $a(x,y)F_x + b(x,y)F_y = f(x,y,F)$ that reduces to two ordinary diff. equations. We review briefly the standard resolution and apply it to (34). First one solves the equation

$$\frac{dy}{dx} = \frac{b(x,y)}{a(x,y)} \tag{35}$$

which gives the system of *characteristic curves* $\{y_c(x) : c \in D\}$ (D is a set of real parameters). Along each of the curves F turns into a univariate function $F_c(x) = F(x, y_c(x))$ that satisfies

$$\frac{dF_c}{dx} = \frac{f(x, y_c(x), F_c(x))}{a(x, y_c(x))} \tag{36}$$

(this follows by the chain rule for partial derivatives). The value of F at a point $p = (x_0, y_0)$ is then $F_c(x_0)$ where $c = c(p)$ is chosen so that y_c goes through p.

The equation (35) becomes for (34)

$$\frac{dy}{dx} = \frac{2y - 1}{1/y - 2x}$$

which is an exact equation $(1/y - 2x)dy + (1 - 2y)dx = 0$. Solving it in a standard way we get the following equation for characteristic curves:

$$y\, e^{(1-2y)x} = c. \tag{37}$$

(36) turns into a separated variables equation

$$\frac{dF_c^*}{dx} = \frac{y_c'}{2y_c - 1} F_c^*$$

whose solution is $F_c^*(x) = d(c).\sqrt{2y_c(x) - 1}$. From (37) we have $y_c(0) = c$ and from $F_c^*(0) = 1$ we get $d(c) = 1/\sqrt{2c - 1}$. Thus $F_c^*(x) = \sqrt{2y_c(x) - 1}/\sqrt{2c - 1}$ and, using (37),

$$F^*(x,y) = \sqrt{\frac{2y - 1}{2y.e^{x(1-2y)} - 1}}\ .$$

Specializing $y = 1$ we obtain (33). $\square$

Setting in (34) $y = 1/2$ we get for $g(x) = F^*(x, 1/2)$ the ord. diff. equation $2(1-x)g' = g$, thus $(g(0) = 1)$ $g(x) = 1/\sqrt{1-x}$. Hence

$$2^n \sum_{T \in \mathcal{T}_{n+1}} w_{12}(T) \left(\frac{1}{2}\right)^{l(T)} = (2n-1)!!. \tag{38}$$

Let $k(T)$ stand for the number of nonleaves of T. By (38) the sum $\sum w_{12}(T).2^{k(T)-1}$ over all trees with n edges gives the same result as the sum $\sum w_{11}(T)$.

The function $F_{12}(x)$ satisfies $F_{12}(x)'.(2-e^x) = F_{12}(x)$. This provides us with the simple recurrence $w_{12}(0) = 1$,

$$w_{12}(n+1) = w_{12}(n) + \sum_{i=1}^{n} w_{12}(i).\binom{n}{i-1}. \tag{39}$$

The first few numbers are

$$\{w_{12}(n)\}_{n \geq 0} = \{1,\ 1,\ 2,\ 7,\ 35,\ 226,\ 1787,\ 16717,\ 180560,\ 2211181,\ \ldots\}.$$

To determine the asymptotics we proceed as in Section 3. The function $2e^{-x} - 1$ is entire and nonzero, except for the simple zeros $\log 2 + 2k\pi i$. Thus we write $F_{12}(x) = (1 - x/\log 2)^{-1/2} A(x)$ where $A(x)$ is analytic in the $((\log 2)^2 + 4\pi^2)^{1/2}$ circle and $A(\log 2) = 1/\sqrt{\log 2}$. By Theorem 3.2

$$w_{12}(n) = n!\ [x^n] F_{12}(x) \sim n!\ \frac{1}{\sqrt{\pi n \log 2}} \left(\frac{1}{\log 2}\right)^n \sim \sqrt{\frac{2}{\log 2}} \left(\frac{n}{e \log 2}\right)^n. \tag{40}$$

6 Concluding remarks

1 An alternative decomposition. In all recurrence arguments we used the decomposition $ps(T) = (T_1, T_2, \ldots, T_k)$. However, one can use the decomposition $T = (T_1, T^*)$ where T_1 is the subtree rooted in the leftmost child of r and T^* is the rest. In some cases this leads to easier derivations of equations for generating functions. On the other hand this decomposition is not well suited to do the indiviual count.

We advice the reader to try some individual counts by the formulae (15)–(20), (22), (25), (26), and (32). For instance, to calculate $w_1(T)$ one writes 1 to each leaf of T and then, by (15), recursively assignes to each vertex v the product of by 1 increased numbers assigned to v's children. Then $w_1(T)$ is the number assigned to r. By such calculations we were motivated to some of the problems stated below.

2 The weight w_{12}. The individual count for the weights w_i, $i = 1, 2, \ldots, 11$ can be done by the (recurrent) formulae (15)–(20), (22), (25), (26), and (32) ($w_8(T)$ can be easily calculated from the definition). The question is how to calculate efficiently for any given T the number $w_{12}(T)$. It would be also interesting to give direct combinatorial proofs and interpretations to (39) and (38).

3 Extremal weight values. We define, for $i = 1, 2, \ldots, 12$,

$$m_i(n) = \min w_i(T) \text{ and } M_i(n) = \max w_i(T)$$

where for $i = 1, 2, \ldots, 10$ the extremum is taken over $\mathcal{T}_n$ and for $i = 11, 12$ over $\mathcal{T}_{n+1}$. In many cases it is easy to determine the extremal value. It is trivial that $m_1(n) = n$ (path), $M_1(n) = 2^{n-1}$ (broom), $m_2(n) = 2$ (broom), $m_3(n) = 2n-1$ (broom), $M_3(n) = 2^n - 1$ (path), $M_4(n) = 2^n - 1$

(path), $m_7(n) = 2$ (broom), $m_8(n) = 2n - 1$ (path), $M_8(n) = 2^{n-1}$ (broom), $m_9(n) = n - 1$ (broom), $m_{11}(n) = 1$ (path), $M_{11}(n) = n!$ (broom), and $m_{12}(n) = 1$ (path).

It is not difficult to show that $m_5(n) = \binom{n}{2} + n$ (path), $M_5(n) = 2^{n-1} + n - 1$ (broom), $M_6(n) = 2^{n-1}$ (broom), and $(n \geq n_0)$ $m_{10}(n) = n - 1$ (broom). Now we determine $M_2(n)$.

Theorem 6.1 *Let $n = 1 + 3m + i > 2$, $i \in \{0, 1, 2\}$. Denote by $\mathcal{U}_n \subset \mathcal{T}_n$ the set of trees whose nonroot vertices have only the degrees 1 or 0 and which have only the branches with 3 edges and either 0, 1 or 2 branches with 2 edges or 1 branch with 4 edges. Then*

$$w_2(T) = M_2(n) \text{ for any } T \in \mathcal{U}_n \text{ and } w_2(T) < M_2(n) \text{ for any } T \in \mathcal{T}_n \backslash \mathcal{U}_n \text{ where}$$

$$M_2(n) = 1 + 3^m \text{ for } i = 0, \ = 1 + 3^m + 3^{m-1} \text{ for } i = 1, \quad \text{and} \ = 1 + 2.3^m \text{ for } i = 2.$$

Proof. Suppose T has a nonroot vertex v with $deg(v) = l \geq 2$. Denote by u the parent of v and by x_i the children of v. The tree T^* arises from T by cutting the edge joining v and x_l and joining x_l to u. We write a_i for $w_2(T_{x_i})$, a for the product of a_i's, and b for the product $\prod w_2(T_t)$ where t runs through the children of u different from v ($b = 1$ if there is no such child). By (16)

$$w_2(T_u) = 1 + (1 + a)b = 1 + b + ab \leq 1 + a_l b + ab = 1 + (1 + a_1 \ldots a_{l-1})a_l b = w_2(T_u^*).$$

Thus $w_2(T) \leq w_2(T^*)$, the equality holds iff x_l is a leaf. Applying repeatedly the transformation we change T into a tree U with the same number of vertices, with no nonroot vertex of degree > 1, and with w_2 at least as large. Let $d_1, d_2, \ldots, d_k$ stand for the number of edges of the branches of U. It holds $w_2(U) = 1 + d_1 d_2 \ldots d_k$ and $d_1 + d_2 + \ldots + d_k = |V(T)| - 1$. We reduced our problem to a well known riddle asking what is the maximum product of a collection of positive integers with fixed sum. The answer follows by easy splitting arguments and is described above — the maximum is achieved exactly when all d_i's equal to 2 or 3 and there is as many 3's as possible, two 2's may be traded for one 4. The trees U with such d_i's form the set $\mathcal{U}_n$. We see that $w_2(T) = w_2(U)$ implies $T = U$ or $d_i = 1$ for some i. But $d_i = 1$ implies that the maximum product is not attained. Therefore the inequality is strict for the trees outside $\mathcal{U}_n$. □

The problem is to determine the remaining extremal values $m_4(n), m_6(n), M_7(n), M_9(n), M_{10}(n)$, and $M_{12}(n)$ or to give some bounds on them. To single some of them out: what is $m_4(n)$ and what are the trees with few infima closed sets? What is $M_{12}(n)$ and what are the trees with many drawings? For $\varepsilon > 0$ fixed and n large we have the bounds

$$\frac{1 - \varepsilon}{4\sqrt{\log 2}} \, \frac{1}{n} \left(\frac{1}{\log 16} \right)^n n! < M_{12}(n) \leq n!$$

The upper bound is trivial and the lower bound follows by the averaging argument from (27) and (40). The problem is how to improve these bounds. The remaining undetermined extremal values can be estimated in a similar way.

4 Two more problems. Is there any tree T different from s for which $w_1(T) = w_3(T)$, i.e., has the same number of chains and antichains? Are there infinitely many of them? We define the *height* of a positive integer m as the minimum height of a tree T such that $w_1(T) = m$. Are there numbers with arbitrary large height? Similarly for w_2.

Acknowledgment

The author thanks the referee for bringing in his attention the reference [1].

References

[1] M. D. Atkinson, The complexity of orders, in I. Rival, ed., *Algorithms and order*, Dordrecht, Kluwer 1989.

[2] E. Bender, Asymptotic methods in enumeration, *Siam Review* **16** (1974), 485–515; Errata **18** (1976), 292.

[3] E. R. Canfield, Remarks on an asymptotic method in combinatorics, *J. Combinatorial Th. Ser. A* **37** (1984), 348–352.

[4] W. Y. C. Chen, Ph.D. thesis, M.I.T., Cambridge, MA (1991).

[5] L. Comtet, *Advanced Combinatorics*, D. Reidel Publishing Company, Dordrecht, 1974.

[6] I. M. Gessel, B. E. Sagan, and Y. Yeh, Enumeration of trees by inversions, *J. Graph Theory* **19** (1995), 435–459.

[7] I. M. Gessel and R. P. Stanley, Algebraic enumeration, in: *Handbook of Combinatorics*, edited by R. L. Graham, M. Grötschel, and L. Lovász, North-Holland, 1995.

[8] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, J. Wiley, New York, 1983.

[9] A. Meir and J. W. Moon, On an asymptotic method in enumeration, *J. Combinatorial Th. Ser. A* **51** (1989), 77–89.

[10] A. M. Odlyzko, Asymptotic enumeration methods, in: *Handbook of Combinatorics*.

[11] N. J. A. Sloane and collaborators, On-line Encyclopedia of Integer Sequences, email: `sequences@research.att.com, superseeker@research.att.com`.

[12] R. Stanley, *Enumerative Combinatorics I*, Wadsworth & Brooks/Cole Advanced Books & Software, Monterey CA, 1986.

[13] Wen-Chin Chen and Wen-Chun Ni, Heap-ordered trees, 2-partitions and continued fractions, *Europ. J. Combinatorics* **15** (1994), 513–517.

[14] H. S. Wilf, *Generatingfunctionology*, Academic Press, New York, 1994.

TILTING AND COTILTING FOR QUIVERS OF TYPE $\tilde{A}_n$

ASLAK BAKKE BUAN AND HENNING KRAUSE

ABSTRACT. Tilting and cotilting modules are classified for the completed path algebra of a quiver of type $\tilde{A}_n$ with linear orientation. This classification problem arises naturally in the classification of cotilting modules over certain associative algebras [5]. The combinatorics of the collection of all tilting and cotilting modules is described in terms of Stasheff associahedra.

INTRODUCTION

Throughout we fix a field k. We consider the completion $k[\![\Delta]\!]$ of the path algebra of the following quiver.

$$\Delta\colon\ 1 \rightleftarrows 2 \longrightarrow 3 \longrightarrow \cdots \longrightarrow n$$

More precisely, $k[\![\Delta]\!] = \varprojlim k[\Delta]/\mathfrak{m}^i$ where $\mathfrak{m}$ denotes the ideal of the path algebra $k[\Delta]$ which is generated by all arrows in Δ. In this paper we classify all finitely presented tilting modules and all locally finite cotilting modules over $k[\![\Delta]\!]$. The initial motivation for this project is to complete the classification of all cotilting modules over a tame hereditary algebra [5], which includes the classification of all cotilting modules for quivers of type $\tilde{A}_n$ having non-linear orientation. To this end we are interested in cotilting objects of certain Grothendieck categories which we call tubes.

Let $\mathcal{C}$ be an abelian Grothendieck category which is a k-category and has a generating set of finite length objects. We say that $\mathcal{C}$ is a *tube* if the full subcategory $\operatorname{fin}\mathcal{C}$ formed by the finite length objects has the following properties:

- $\operatorname{Hom}(X,Y)$ and $\operatorname{Ext}^1(X,Y)$ have finite k-dimension for all $X, Y \in \operatorname{fin}\mathcal{C}$;
- $\operatorname{fin}\mathcal{C}$ has *Serre duality*, that is, there is an equivalence $\tau\colon \operatorname{fin}\mathcal{C} \to \operatorname{fin}\mathcal{C}$ and a natural isomorphism $D\operatorname{Ext}^1(X,Y) \cong \operatorname{Hom}(Y,\tau X)$ for all $X, Y \in \operatorname{fin}\mathcal{C}$, where $D = \operatorname{Hom}_k(-,k)$;
- there are only finitely many isomorphism classes of simple objects in $\operatorname{fin}\mathcal{C}$.

Note that the Auslander-Reiten quiver of $\operatorname{fin}\mathcal{C}$ has the shape of a tube [14] provided that $\mathcal{C}$ is connected; this explains the terminology. The number of simple objects in $\mathcal{C}$ is called the *rank* of $\mathcal{C}$. Tubes arise in the category of regular modules over a tame hereditary algebra, but also as subcategories of other abelian categories, see for instance [2, 10]. We shall use that a tube of rank n is equivalent to the category of locally finite $k[\![\Delta]\!]$-modules. Recall that a module is *locally finite* if it is a filtered colimit of finite length modules.

Next we recall the definition of a cotilting object [6] for any Grothendieck category $\mathcal{C}$. To this end we fix an object T in $\mathcal{C}$. We let $\operatorname{Prod} T$ denote the category of all direct summands in any product of copies of T. The object T is called *cotilting object* if the following holds:

(C1) the injective dimension of T is at most 1;
(C2) $\operatorname{Ext}^1(T^\alpha, T) = 0$ for every cardinal α;
(C3) there is an exact sequence $0 \to T_1 \to T_0 \to Q \to 0$ with each T_i in $\operatorname{Prod} T$ for some injective cogenerator Q.

By definition, two cotilting objects T and T' are equivalent if $\operatorname{Prod} T = \operatorname{Prod} T'$. Let us mention a result from [5] which motivates the classification of cotilting objects.

For any locally finite Grothendieck category $\mathcal{C}$, there exists a bijection between the set of torsion pairs $(\mathcal{T}, \mathcal{F})$ for the category $\operatorname{fin}\mathcal{C}$ such that $\mathcal{F}$ generates $\operatorname{fin}\mathcal{C}$, and the set of equivalence classes of cotilting objects in $\mathcal{C}$.

Our first result describes the structural properties of an arbitrary cotilting object in a tube.

Theorem A. *Let T be an object in a tube of rank n satisfying $\operatorname{Ext}^1(T, T) = 0$.*

(1) *T decomposes uniquely into a coproduct of indecomposable objects having local endomorphism rings.*
(2) *T is a cotilting object if and only if the number of pairwise non-isomorphic indecomposable direct summands of T equals n.*

The classification of cotilting objects in a tube of rank n is the same as the classification of locally finite cotilting modules over $k[\![\Delta]\!]$. Note that $k[\![\Delta]\!]$ is a noetherian algebra over a complete local ring which is of *artinian type*, that is, each non-zero locally finite module has a non-zero artinian direct summand. For this class of algebras we have the following.

Theorem B. *Let Λ be a noetherian algebra over a complete local ring which is of artinian type. Then the duality between Λ- and Λ^{op}-modules induces a bijection between the equivalence classes of finitely presented Λ-tilting modules and the equivalence classes of locally finite Λ^{op}-cotilting modules.*

This result extends the bijection between finitely presented tilting and cotilting modules over artin algebras. It would be interesting to see a general correspondence between tilting and cotilting modules which does not depend on finiteness conditions on the algebra.

The second part of this paper is devoted to the classification of all finitely presented tilting modules over $k[\![\Delta]\!]$. It is somewhat surprising that all of them are induced from tilting modules over the path algebra of the following quiver.

$$\Gamma:\ 1 \longrightarrow 2 \longrightarrow 3 \longrightarrow \cdots \longrightarrow n$$

The collection of all $k[\Gamma]$-tilting modules is best described in terms of the Stasheff associahedron of dimension $n-1$. Another connection between representations of Dynkin quivers and generalized associahedra is discussed in [12].

Theorem C. *The isomorphism classes of faithful and basic partial $k[\Gamma]$-tilting modules correspond bijectively to the faces of the Stasheff associahedron of dimension $n-1$. This correspondence identifies the tilting modules with the vertices, and it identifies the Hasse diagram of the lattice of all tilting modules with the 1-skeleton of the Stasheff associahedron. Therefore the lattice of tilting modules is a Tamari lattice.*

The collection of all faithful partial $k[\![\Delta]\!]$-tilting modules is obtained by glueing together n copies of a Stasheff associahedron of dimension $n-1$. This leads to a combinatorial structure which seems to be new; it is discussed in an appendix which is independent from the rest of this paper. It turns out that the tilting modules are parametrized by integer sequences as follows.

Theorem D. *The map sending a $k[\![\Delta]\!]$-module X to the sequence $(a_1, \dots, a_n)$ where a_i denotes the number of composition factors of $X/\operatorname{rad} X$ isomorphic to the simple with support $i \in \Delta$, induces a bijection between the isomorphism classes of finitely presented basic $k[\![\Delta]\!]$-tilting modules and the sequences $(a_1, \dots, a_n)$ of non-negative integers satisfying $\sum_i a_i = n$.*

Acknowledgements. Work on this project started while both authors were visiting the "Senter for Høyere Studier" in Oslo. We would like to thank this institution for its generous support and its hospitality. Also, we are grateful to Bill Crawley-Boevey for suggesting the completed path algebra as the right set-up to study tubes, and we thank Kiyoshi Igusa for pointing out the relevance of the Stasheff associahedra (cf. the formula B.1). The idea for our classification of cotilting objects is based on a combinatorial formula. The On-Line Encyclopedia of Integer Sequences [19] produced this formula from the input $1, 3, 10, 35$, which are the number of cotilting objects in tubes of rank $1, 2, 3, 4$.

1. Cotilting versus tilting

Let Λ be an associative R-algebra over a commutative ring R. We denote by $\operatorname{Mod}\Lambda$ the category of (right) Λ-modules and $\operatorname{mod}\Lambda$ denotes the full subcategory formed by the finitely presented Λ-modules. In this section we establish a connection between cotilting objects for the category of locally finite Λ-modules and tilting modules over Λ^{op}. We need to fix some notation and terminology.

Recall that a Λ-module is *locally finite* if it is a filtered colimit of finite length modules. The full subcategory formed by the locally finite Λ-modules is denoted by $\operatorname{Fin}\Lambda$. In addition, we consider the full subcategories given by the noetherian Λ-modules (written as $\operatorname{noeth}\Lambda$), the artinian Λ-modules (written as $\operatorname{art}\Lambda$), and the finite length Λ-modules (written as $\operatorname{fin}\Lambda$).

Next we recall the definition of a finitely presented tilting module. A module $T \in \operatorname{mod}\Lambda$ is a *tilting module* if

(T1) the projective dimension of T is at most 1;
(T2) $\operatorname{Ext}^1_\Lambda(T,T)=0$;
(T3) there is an exact sequence $0\to\Lambda\to T_0\to T_1\to 0$ with each T_i in $\operatorname{add}T$.

A tilting module is called *basic* if each indecomposable direct summand occurs exactly once in a direct sum decomposition. Two finitely presented tilting modules T,T' are *equivalent* if $\operatorname{add}T=\operatorname{add}T'$.

Throughout this section we assume that Λ is a noetherian R-algebra and that R is a complete local ring R. Let I be the injective envelope of $R/\operatorname{rad}R$. The functor $D=\operatorname{Hom}_R(-,I)\colon \operatorname{Mod}R\to\operatorname{Mod}R$ induces functors between $\operatorname{Mod}\Lambda$ and $\operatorname{Mod}\Lambda^{\mathrm{op}}$ which become dualities on appropriate subcategories.

Lemma 1.1. *The functor D induces inverse dualities* $\operatorname{noeth}\Lambda\to\operatorname{art}\Lambda^{\mathrm{op}}$ *and* $\operatorname{art}\Lambda^{\mathrm{op}}\to\operatorname{noeth}\Lambda$.

We do not give the proof of this lemma but refer instead to [1, Section I.5] for basic facts about algebras over complete local rings.

The following characterization of a tilting module is classical. Bongartz proved it for finite dimensional algebras [3], but the same proof works in our setting. We denote for any module X by $\delta(X)$ the number of pairwise non-isomorphic indecomposable direct summands of X.

Lemma 1.2. *A finitely presented Λ-module T is a tilting module if and only if the following holds:*

(1) *the projective dimension of T is at most* 1*;*
(2) $\operatorname{Ext}^1_\Lambda(T,T)=0$*;*
(3) $\delta(T)=n$ *where n denotes the number of simple Λ-modules.*

Moreover, each module satisfying (1) and (2) is a direct summand of a tilting module.

Next recall from [8] that an object X in a locally finite Grothendieck category is *endofinite* if $\operatorname{Hom}(C,X)$ has finite length as $\operatorname{End}(X)$-module for each finite length object C. All we need to know about endofinite objects is collected in the following lemma.

Lemma 1.3. (1) *Every endofinite object decomposes into indecomposable objects with local endomorphism rings.*
(2) *A finite coproduct of endofinite objects is endofinite, and all coproducts of a fixed endofinite object are endofinite.*
(3) *If X is indecomposable and endofinite, then* $\operatorname{Add}X=\operatorname{Prod}X$.

Proof. See [7, Section 3] and [8, Section 3.6] □

Lemma 1.4. *Each artinian Λ-module is an endofinite object in* $\operatorname{Fin}\Lambda$.

Proof. Let X be artinian and C of finite length. One checks that $\operatorname{Hom}_{\Lambda^{\mathrm{op}}}(DX,DC)$ has finite length as a $\operatorname{End}_{\Lambda^{\mathrm{op}}}(DX)$-module, for instance by induction on the composition length of C. Then apply the duality, to see that $\operatorname{Hom}_\Lambda(C,X)$ is of finite length over $\operatorname{End}_\Lambda(X)$. □

We say that the algebra Λ is of *artinian type* if each non-zero locally finite Λ-module has a non-zero direct summand which is artinian. Note that 'artinian type' is equivalent to 'finite representation type' in case Λ is artinian.

Proposition 1.5. *Suppose Λ is of artinian type. Let X be a locally finite Λ-module satisfying* $\operatorname{id} X \leq 1$ *and* $\operatorname{Ext}^1_\Lambda(X, X) = 0$.

(1) *X decomposes into a coproduct of indecomposable modules with local endomorphism rings.*
(2) *$\delta(X) \leq n$ where n is the number of simple Λ-modules.*
(3) $\operatorname{Ext}^1_\Lambda(X', X) = 0$ *for every product $X' = X^\alpha$ taken in* $\operatorname{Fin}\Lambda$.

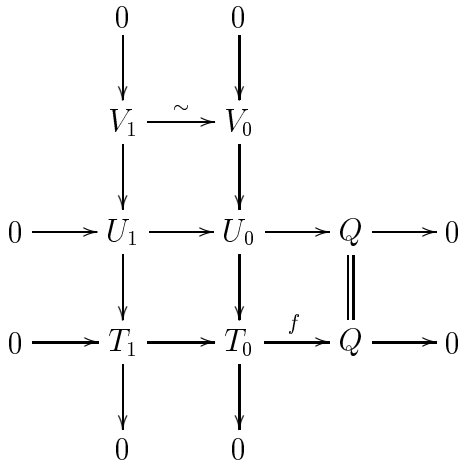
Proof. Up to isomorphism, X has only a finite number of indecomposable artinian direct summands. This follows from Lemma 1.2, using the duality D. Label the indecomposables $X_1, \ldots, X_p$. Using Zorn's lemma, we find a maximal direct summand X' of X which is a coproduct of modules in $\{X_1, \ldots, X_p\}$. Clearly, $X' = X$ since Λ is of artinian type, and X is endofinite by Lemmas 1.3 and 1.4. Now all assertions follow from the properties of endofinite objects. □

Lemma 1.6. *Suppose Λ is of artinian type. Let T be a cotilting object in* $\operatorname{Fin}\Lambda$. *Then there exists an exact sequence* $0 \to T_1 \to T_0 \to D(\Lambda^{\mathrm{op}}) \to 0$ *such that T and $T_0 \amalg T_1$ are equivalent cotilting objects and each T_i belongs to* $\operatorname{art}\Lambda \cap \operatorname{Prod} T$.

Proof. We write $Q = D(\Lambda^{\mathrm{op}})$ and note that Q is an injective cogenerator for the category $\operatorname{Fin}\Lambda$. Next observe that for indecomposable objects X and Y in $\operatorname{Fin}\Lambda$, we have that $\operatorname{Hom}_\Lambda(X, Y)$ is finitely generated as $\operatorname{End}_\Lambda(X)$-module. This is because X and Y are artinian by our assumption, and we have the duality $\operatorname{art}\Lambda \to \operatorname{noeth}\Lambda^{\mathrm{op}}$.

Now choose an exact sequence $0 \to U_1 \to U_0 \to Q \to 0$ with $U_i \in \operatorname{Prod} T$. We know from Proposition 1.5 that T decomposes into a coproduct of indecomposable objects and only finitely many isoclasses occur. We find therefore a map $f\colon T_0 \to Q$ such that $U_0 \to Q$ factors through f and T_0 decomposes into finitely many indecomposables from $\operatorname{Prod} T$. In particular, $T_0 \in \operatorname{art}\Lambda$. We may assume that f is minimal, that is every endomorphism $g\colon T_0 \to T_0$ with $f \circ g = f$ is an isomorphism. Note that f factors through $U_0 \to Q$ since $\operatorname{Ext}^1_\Lambda(T_0, U_1) = 0$. Thus $U_0 \cong T_0 \amalg V_0$ for some object V_0, and

we obtain the following commutative diagram.



We conclude that $U_1 \cong T_1 \amalg V_1$. In particular, each T_i belongs to $\operatorname{art}\Lambda \cap \operatorname{Prod} T$. It remains to show that $T_0 \amalg T_1$ is a cotilting object which is equivalent to T. However, this follows from our construction, using for instance Proposition 3.1 in [5]. □

Lemma 1.7. *Let $Y \in \operatorname{Mod}\Lambda$ be artinian. Then the class of modules X satisfying $\operatorname{Ext}^1_\Lambda(X,Y) = 0$ is closed under taking products.*

Proof. We can decompose $Y = Y' \amalg Y''$ such that Y' is injective and $Y'' = D\operatorname{Tr} Z$ for some $Z \in \operatorname{mod}\Lambda^{\mathrm{op}}$. Now use the Auslander-Reiten formula $\operatorname{Ext}^1_\Lambda(-, D\operatorname{Tr} Z) = D\operatorname{\underline{Hom}}_\Lambda(Z,-)$ (see [1, Proposition I.3.4]). Note that every map $Z \to \prod_i P_i$ into a product of projectives factors through a projective since $\prod_i P_i$ is flat. □

Lemma 1.8. *Let $T \in \operatorname{mod}\Lambda^{\mathrm{op}}$ be a tilting module. Then DT is a Λ-cotilting module.*

Proof. Let $T \in \operatorname{mod}\Lambda^{\mathrm{op}}$ be a tilting module. The conditions on T for a tilting module translate via the duality D into the conditions on DT for a cotilting module. More precisely, (C1) and (C3) follow immediately from (T1) and (T3). Condition (C2) follows from (T2), using Lemma 1.7. Thus DT is a cotilting module. □

Lemma 1.9. *Let $\mathcal{A}$ be any abelian Grothendieck category and $\mathcal{A}'$ be a localizing subcategory. If T is a cotilting object in $\mathcal{A}$ and belongs to $\mathcal{A}'$, then T is also a cotilting object in $\mathcal{A}'$.*

Proof. We use the well-known fact that in any Grothendieck category, T is a cotilting object if and only if $\operatorname{id} T \leq 1$ and $\operatorname{Cogen} T = {}^{\perp}T$, where $\operatorname{Cogen} T$ is the class of subobjects of products of copies of T, and ${}^{\perp}T$ is the class of objects X satifying $\operatorname{Ext}^1(X,T) = 0$.

Now assume that T is a cotilting object in $\mathcal{A}$. Clearly, $\operatorname{id} T \leq 1$ holds in $\mathcal{A}'$ because this is equivalent to $\operatorname{Ext}^2(-,T) = 0$. The inclusion functor $\mathcal{A}' \to \mathcal{A}$ has a right adjoint which preserves products. This implies that the condition $\operatorname{Cogen} T = {}^{\perp}T$ carries over from $\mathcal{A}$ to $\mathcal{A}'$ as well. In fact, $\operatorname{Cogen}_{\mathcal{A}'} T = \mathcal{A}' \cap \operatorname{Cogen}_{\mathcal{A}} T$. Thus T is a cotilting object in $\mathcal{A}'$. □

Theorem 1.10. *Let Λ be of artinian type. Then the following conditions are equivalent for a locally finite Λ-module X:*

(1) *X is a cotilting object in* $\operatorname{Mod}\Lambda$.
(2) *X is a cotilting object in* $\operatorname{Fin}\Lambda$.
(3) $\operatorname{Prod} X = \operatorname{Prod} DT$ *in* $\operatorname{Mod}\Lambda$ *for some finitely presented Λ^{op}-tilting module T.*
(4) $\operatorname{Prod} X = \operatorname{Prod} DT$ *in* $\operatorname{Fin}\Lambda$ *for some finitely presented Λ^{op}-tilting module T.*

Moreover, the assignment $T \mapsto DT$ induces a bijection between the equivalence classes of finitely presented Λ^{op}-tilting modules and the equivalence classes of locally finite Λ-cotilting modules.

Proof. (1) $\Rightarrow$ (2): First observe that the locally finite Λ-modules form a localizing subcategory in $\operatorname{Mod}\Lambda$. Now apply Lemma 1.9.

(2) $\Rightarrow$ (3): Let X be a cotilting object for the category $\operatorname{Fin}\Lambda$. Then X is equivalent to an artinian cotilting object by Lemma 1.6, which is of the form DT for some tilting module $T \in \operatorname{mod}\Lambda^{\mathrm{op}}$. The proof shows that every indecomposable direct summand of DT is a direct summand of X. Thus $\operatorname{Prod} DT \subseteq \operatorname{Prod} X$. On the other hand, $\operatorname{Ext}^1_\Lambda(X^\alpha, X) = 0$ for every product X^α taken in $\operatorname{Mod}\Lambda$, by Lemma 1.7, since X decomposes into a coproduct of artinian objects. We know from Lemma 1.8 that DT is a cotilting Λ-module, and combining this with $\operatorname{Prod} DT \subseteq \operatorname{Prod} X$, we obtain $\operatorname{Prod} DT = \operatorname{Prod} X$, for instance by Proposition 3.1 in [5].

(3) $\Rightarrow$ (4): This follows from the fact that the right adjoint of the inclusion $\operatorname{Fin}\Lambda \to \operatorname{Mod}\Lambda$ preserves products.

(4) $\Rightarrow$ (1): The module DT is a cotilting module by Lemma 1.8. The assumption on X implies that it decomposes into indecomposables, and the isomorphism classes which appear are precisely those appearing in a decomposition of DT. This follows essentially from Proposition 1.5. Thus X is a cotilting module since we know it for DT. □

Remark 1.11. The category of locally finite Λ-modules is usually not closed under taking products. However, one checks easily for two locally finite tilting modules T and T', that $\operatorname{Prod} T = \operatorname{Prod} T'$ in $\operatorname{Mod}\Lambda$ if and only if $\operatorname{Prod} T = \operatorname{Prod} T'$ in $\operatorname{Fin}\Lambda$.

2. Tubes

Let $\mathcal{C}$ be a tube of rank n and suppose that $\mathcal{C}$ is connected, that is, any decomposition $\mathcal{C} = \mathcal{C}_1 \amalg \mathcal{C}_2$ into abelian categories implies $\mathcal{C}_1 = 0$ or $\mathcal{C}_2 = 0$. Note that any tube decomposes into finitely many connected tubes. In this section we exhibit some basic properties of $\mathcal{C}$ and establish an equivalence between $\mathcal{C}$ and the category of locally finite $\tilde{\Lambda}_n$-modules.

First we recall the classification of finite length objects which is well-known: each indecomposable object is uniserial and uniquely determinded by its socle and its composition length. For each simple object S and each $n \in \mathbb{N}$, we denote by $S[n]$ the object with socle S and composition length n. We obtain a chain of monomorphisms

$$S = S[1] \longrightarrow S[2] \longrightarrow \cdots$$

and denote by $S[\infty]$ the *Prüfer object* $\varinjlim S[n]$ which is independent of the choice of maps. Note that each Prüfer object is indecomposable injective.

Lemma 2.1. *Every non-zero object in $\mathcal{C}$ has an indecomposable direct factor, and every indecomposable object is of the form $S[n]$ for some simple S and some $n \in \mathbb{N} \cup \{\infty\}$.*

Proof. We use the fact that for each simple S and each $n \in \mathbb{N}$ the natural map

$$S[n] \longrightarrow S[n+1] \amalg S[n]/S$$

is left almost split. Now let X be a non-zero object, and fix a non-zero map $f\colon S \to X$ for some simple S. Let $n \geq 1$ be the maximal number such that there is a factorization

$$f\colon S \longrightarrow S[n] \xrightarrow{f'} X$$

so that f' is a monomorphism. We claim that f' splits. If $n = \infty$, then this is clear since $S[\infty]$ is injective. Assume $n < \infty$ and f' does not split. Then f' factors through the left almost split map starting in $S[n]$. The composite $S[n] \to S[n]/S \to X$ kills S. Therefore f factors through the natural map $S \to S[n+1]$. The corresponding map $S[n+1] \to X$ kills S by our choice of n and this is a contradiction. We conclude that f splits. □

Denote by $\tilde{\Lambda}_n$ the completion of the path algebra of the following quiver.

$$1 \rightleftarrows 2 \longrightarrow 3 \longrightarrow \cdots \longrightarrow n$$

The center of $\tilde{\Lambda}_n$ contains a copy of the ring $k[\![t]\!]$ of power series. The generator of this copy corresponds to the sum $\sum_{i=1}^{n} \gamma_i$ where γ_i is the path of length n starting and ending in the vertex i. Note that $\tilde{\Lambda}_n$ is finitely generated over R so that $\tilde{\Lambda}_n$ is a noetherian algebra over a complete local ring.

Lemma 2.2. *The endomorphism ring of $\coprod_{S \text{ simple}} S[\infty]$ is isomorphic to $\tilde{\Lambda}_n$.*

Proof. Number the simples $S_1, \ldots, S_n$ such that there are epimorphisms $\pi_i\colon S_i[\infty] \to S_{i+1}[\infty]$ with simple kernel for each i modulo n. The π_i generate the endomorphism ring of $S_1 \amalg \ldots \amalg S_n$ and we get an isomorphism onto $\tilde{\Lambda}_n$ by sending π_i to the arrow $i \to i+1$. □

Proposition 2.3. *The category $\mathcal{C}$ is equivalent to the category of locally finite $\tilde{\Lambda}_n$-modules.*

Proof. The category $\operatorname{art}\mathcal{C}$ is abelian and $Q = \coprod_{S \text{ simple}} S[\infty]$ is an injective cogenerator. Moreover, each object $X \in \operatorname{art}\mathcal{C}$ admits an injective copresentation $0 \to X \to I_0 \to I_1$ with each $I_i \in \operatorname{add} Q$. It follows that the opposite category is equivalent to the category of finitely presented modules over $\operatorname{End}(Q)^{\mathrm{op}}$ via the functor $\operatorname{Hom}(-, Q)$. Composing this functor with the duality $\operatorname{noeth}\tilde{\Lambda}_n^{\mathrm{op}} \to \operatorname{art}\tilde{\Lambda}_n$ induces an equivalence $F\colon \operatorname{art}\mathcal{C} \to \operatorname{art}\tilde{\Lambda}_n$. This induces an equivalence $\mathcal{C} \to \operatorname{Fin}\tilde{\Lambda}_n$ by sending $X = \varinjlim X_\alpha$ to $\varinjlim FX_\alpha$ since every object in $\mathcal{C}$ is a filtered colimit of finite length objects. □

Using the equivalence between $\mathcal{C}$ and the category of locally finite $\tilde{\Lambda}_n$-modules, we obtain from Theorem 1.10 the following correspondence between tilting and cotilting objects.

Corollary 2.4. *The algebra $\tilde{\Lambda}_n$ is of artinian type. Therefore there are, up to equivalence, canonical bijections between*

(1) *cotilting objects in a tube of rank n,*
(2) *locally finite cotilting modules over $\tilde{\Lambda}_n$,*
(3) *finitely presented tilting modules over $\tilde{\Lambda}_n$.*

Proof. The bijections are established in Theorem 1.10. All we need to show is that $\tilde{\Lambda}_n$ is of artinian type. However, this follows from Lemma 2.1. □

3. Tilting for quivers of type A_n

We fix a quiver of type A_n with linear orientation

$$1 \longrightarrow 2 \longrightarrow 3 \longrightarrow \cdots \longrightarrow n$$

and denote by Λ_n its path algebra over the field k. For each $i \in \{1, \ldots, n\}$, let P_i be the indecomposable projective Λ_n-module having as a k-basis all paths ending in the vertex i. Let $\mathcal{I}(n)$ denote the set of intervals $[i, j]$ in $\mathbb{Z}$ with $0 \leq i < j \leq n$. Each indecomposable Λ_n-module is of the form $M_{[i,j]} = P_j/\mathrm{rad}^{j-i} P_j$, and we write $M_X = \coprod_{I \in X} M_I$ for any $X \subseteq \mathcal{I}(n)$. It easy to compute $\mathrm{Ext}^1_{\Lambda_n}(-,-)$ and we obtain the following.

Lemma 3.1. $\mathrm{Ext}^1_{\Lambda_n}(M_I, M_J) = 0 = \mathrm{Ext}^1_{\Lambda_n}(M_J, M_I)$ *if and only if the intervals I and J are compatible, that is, $I \subseteq J$ or $J \subseteq I$ or $I \cap J = \emptyset$.*

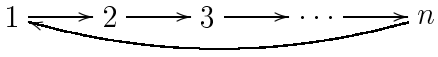
We denote for each module M by $\mathrm{top}\, M$ the factor $M/\mathrm{rad}\, M$, and $\dim M$ denotes the sequence $(a_1, \ldots, a_n)$ where a_i is the number of composition factors of M isomorphic to the simple $P_i/\mathrm{rad}\, P_i$. The classification of the Λ_n-tilting modules is well-known [4].

Proposition 3.2. *The map sending a Λ_n-module M to* $\dim(\mathrm{top}\, M)$ *induces a bijection between the set of isomorphism classes of basic tilting modules over Λ_n and the set of sequences $(a_1, \ldots, a_n)$ of non-negative integers such that $\sum_i a_i = n$ and $\sum_{i \leq p} a_i \leq p$ for all $1 \leq p \leq n$.*

Proof. Lemma 3.1 reduces the classification of tilting modules to the classification of subsets $X \subseteq \mathcal{I}(n)$ of cardinality n such that all elements in X are pairwise compatible. Now everything follows from Lemma A.1 since we have for $X \subseteq \mathcal{I}(n)$ that $\mathrm{top}\, X = \dim(\mathrm{top}\, M_X)$. □

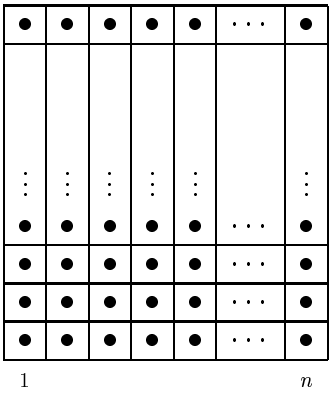
4. Tilting for quivers of type $\tilde{A}_n$

We fix a quiver of type $\tilde{A}_{n-1}$ with linear orientation



and denote by $\tilde{\Lambda}_n$ the completion of its path algebra over the field k. In this section the finitely presented tilting modules over $\tilde{\Lambda}_n$ are classified. We present two approaches: a reduction to the classification of the Λ_n-tilting modules, and a reduction to the classification of the $\tilde{\Lambda}_{n-1}$-tilting modules.

For each $i \in \{1, \dots, n\}$, we denote by P_i the indecomposable projective $\tilde{\Lambda}_n$-module corresponding to the vertex i. The following figure shows the indecomposable $\tilde{\Lambda}_n$-modules. The dots in the ith column represent the factors of P_i, with $P_i/\mathrm{rad}^{j+1} P_i$ sitting on top of $P_i/\mathrm{rad}^j P_i$ for $1 \leq j < \infty$.



The cyclic group C_n of order n acts in an obvious way on $\tilde{\Lambda}_n$ and therefore on $\operatorname{mod} \tilde{\Lambda}_n$. For each $g \in C_n$ and $X \in \operatorname{mod} \tilde{\Lambda}_n$, we denote by X^g the translate of X.

4.1. **Classification via tilting modules over Λ_n.** Consider the embedding of quivers $A_n \to \tilde{A}_{n-1}$ which sends the vertex $i \in A_n$ to $i \in \tilde{A}_{n-1}$. This induces an embedding $\Lambda_n \to \tilde{\Lambda}_n$.

Lemma 4.1. *The functor*

$$F\colon \operatorname{mod} \Lambda_n \longrightarrow \operatorname{mod} \tilde{\Lambda}_n, \quad X \mapsto X \otimes_{\Lambda_n} \tilde{\Lambda}_n$$

has the following properties:

(1) *F is faithful, exact, and preserves indecomposability.*
(2) $\operatorname{Ext}^1_{\Lambda_n}(X, Y) \cong \operatorname{Ext}^1_{\tilde{\Lambda}_n}(FX, FY)$ *for all* $X, Y \in \operatorname{mod} \Lambda_n$.
(3) $X \in \operatorname{mod} \tilde{\Lambda}_n$ *belongs to the image of F iff* $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(X, P_n) = 0$.
(4) $X \in \operatorname{mod} \Lambda_n$ *is tilting iff FX is tilting.*

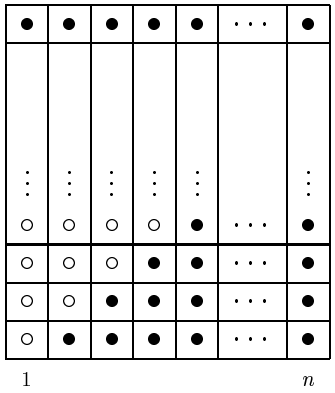
Proof. (1) This is straightforward. Note that $\tilde{\Lambda}_n$ is projective as a Λ_n-module.

(2) Use the Auslander-Reiten formula $\operatorname{Ext}^1_\Lambda(X, Y) = D \operatorname{Hom}_\Lambda(Y, D \operatorname{Tr} X)$. Note that F commutes with the Auslander-Reiten translate $D \operatorname{Tr}$.

(3) An indecomposable module belongs to the image if and only if it is of the form $P_i/\mathrm{rad}^j P_i$ with $j < i$. Now use again the Auslander-Reiten formula.

(4) A Λ-module X is a tilting module if and only if $\operatorname{Ext}^1_\Lambda(X, X) = 0$ and X has n pairwise non-isomorphic indecomposable summands. Thus (4) follows from (1) and (2). □

The following figure describes the image of F.



Proposition 4.2. *A $\tilde{\Lambda}_n$-module is a tilting module if and only if it is isomorphic to $(FT)^g$ for some $g \in C_n$ and some Λ_n-tilting module T. For fixed $g \in C_n$, two Λ_n-tilting modules T and T' are equivalent if and only if $(FT)^g$ and $(FT')^g$ are equivalent.*

Proof. We apply Lemma 4.1. There it is shown that F preserves tilting modules. Now suppose that $T \in \operatorname{mod} \tilde{\Lambda}_n$ is a tilting module. Then T has at least one indecomposable projective summand because every module X of finite length satisfying $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(X, X) = 0$ has at most $n-1$ pairwise non-isomorphic indecomposable summands. Let $T = T' \amalg T''$ and choose $g \in C_n$ such that $(T')^g \cong P_n$. Then T^g belongs to the image of F by Lemma 4.1, since $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(T^g, P_n) = 0$. Let $T^g = FX$. Then X is a tilting module, again by Lemma 4.1, and $T = (FX)^{g^{-1}}$. This completes the proof. □

Corollary 4.3. *The map sending a $\tilde{\Lambda}_n$-module M to $\dim(\operatorname{top} M)$ induces a bijection between the set of isomorphism classes of basic tilting modules over $\tilde{\Lambda}_n$ and the set of sequences $(a_1, \ldots, a_n)$ of non-negative integers such that $\sum_i a_i = n$.*

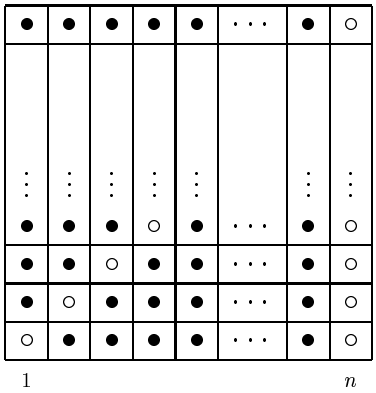
Proof. This follows from the classification of tilting modules over Λ_n in Proposition 3.2, using that $\operatorname{top}(FM) \cong F(\operatorname{top} M)$. □

4.2. **Classification via tilting modules over $\tilde{\Lambda}_{n-1}$.** Let

$$\tilde{\Lambda}_n = P_1 \amalg \ldots \amalg P_{n-1} \amalg P_n \longrightarrow P_1 \amalg \ldots \amalg P_{n-1} \amalg P_1 = P$$

be the map sending $(x_1, \ldots, x_{n-1}, x_n)$ to $(x_1, \ldots, x_{n-1}, \rho(x_n))$ with $\rho\colon P_n \to P_1$ being the monomorphism with simple cokernel. The composition of the induced map $\operatorname{Hom}_{\tilde{\Lambda}_n}(\tilde{\Lambda}_n, \tilde{\Lambda}_n) \to \operatorname{Hom}_{\tilde{\Lambda}_n}(\tilde{\Lambda}_n, P)$ with the inverse of the isomorphism $\operatorname{Hom}_{\tilde{\Lambda}_n}(P, P) \to \operatorname{Hom}_{\tilde{\Lambda}_n}(\tilde{\Lambda}_n, P)$ induces a ring homomorphism $\phi\colon \tilde{\Lambda}_n \to \operatorname{End}_{\tilde{\Lambda}_n}(P)$. Clearly, $\operatorname{End}_{\tilde{\Lambda}_n}(P)$ is Morita equivalent to $\tilde{\Lambda}_{n-1}$, and restriction of scalars along ϕ induces a fully faithful functor $\phi_*\colon \operatorname{mod} \tilde{\Lambda}_{n-1} \to \operatorname{mod} \tilde{\Lambda}_n$ with inverse $\phi^*\colon \operatorname{mod} \tilde{\Lambda}_n \to \operatorname{mod} \tilde{\Lambda}_{n-1}$ induced by $P \otimes_{\tilde{\Lambda}_n} -$. Note that ϕ is a universal localization in the sense of Schofield [15], making the arrow $n \to 1$ in $\tilde{\Lambda}_n$, hence the map $\rho\colon P_n \to P_1$ in $\operatorname{mod} \tilde{\Lambda}_n$, invertible. In particular, the image of ϕ_* is the full subcategory of modules X in $\operatorname{mod} \tilde{\Lambda}_n$ with $\operatorname{Hom}_{\tilde{\Lambda}_n}(S_1, X) = 0$

and $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(S_1, X) = 0$, since $S_1 = \operatorname{Coker}\rho$. The following figure illustrates the image of ϕ_*.



The embedding $\operatorname{mod}\tilde{\Lambda}_{n-1} \to \operatorname{mod}\tilde{\Lambda}_n$ via ϕ_* is not appropriate for our purpose; we need a slight modification. To this end we consider the full subcategory $\mathcal{X}$ of modules X in $\operatorname{mod}\tilde{\Lambda}_n$ satisfying $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(X, S_1) = 0$ and $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(S_1, X) = 0$, which in addition have no direct summand isomorphic to S_1. We denote by $I\colon \mathcal{X} \to \operatorname{mod}\tilde{\Lambda}_n$ the inclusion functor.

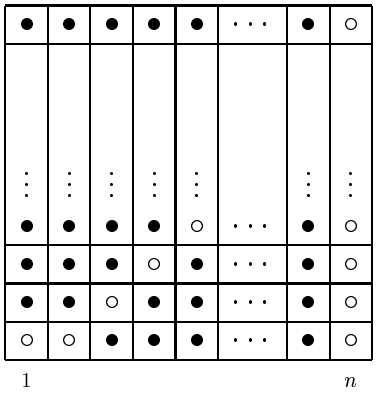
Lemma 4.4. *The functor $\phi^* \circ I\colon \mathcal{X} \to \operatorname{mod}\tilde{\Lambda}_{n-1}$ is an equivalence.*

Proof. The functor ϕ^* is a left adjoint of the embedding ϕ_*. Denoting by $\mathcal{Y}$ the image of ϕ_*, we see that the composite $\phi_* \circ \phi^*$ leaves almost all indecomposables in $\mathcal{X}$ unchanged, except the indecomposables $X \in \mathcal{X}$ with $\operatorname{soc} X = S_1$, which are sent to $X/\operatorname{soc} X$. Thus the following diagram commutes.

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{\sim} & \mathcal{Y} \\ \downarrow{\scriptstyle I} & & \downarrow \\ \operatorname{mod}\tilde{\Lambda}_n & \xrightarrow{\phi_* \circ \phi^*} & \operatorname{mod}\tilde{\Lambda}_n \end{array}$$

The assertion follows by composing $\phi_* \circ \phi^*$ with ϕ^*, since $\phi^* \circ \phi_* = \mathrm{id}_{\operatorname{mod}\tilde{\Lambda}_{n-1}}$. □

We denote by $G = I \circ (\phi^* \circ I)^{-1}$ the composite of I with an inverse of $\phi^* \circ I$. The following figure illustrates the image of G.



Lemma 4.5. *The functor* $G\colon \operatorname{mod}\tilde{\Lambda}_{n-1} \to \operatorname{mod}\tilde{\Lambda}_n$ *has the following properties:*

(1) G *is fully faithful.*

(2) $\operatorname{Ext}^1_{\tilde{\Lambda}_{n-1}}(X,Y) \cong \operatorname{Ext}^1_{\tilde{\Lambda}_n}(GX,GY)$ *for all* $X, Y \in \operatorname{mod}\tilde{\Lambda}_{n-1}$.

(3) $X \in \operatorname{mod}\tilde{\Lambda}_n$ *belongs to the image of* G *iff* $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(X,S_1) = 0 = \operatorname{Ext}^1_{\tilde{\Lambda}_n}(S_1,X)$ *and no direct summand of* X *is isomorphic to* S_1.

Proof. (1) and (3) follow immediately from the definition of G and Lemma 4.4. To prove (2) one uses the Auslander-Reiten formula. □

Proposition 4.6. *Let* $n > 1$. *A* $\tilde{\Lambda}_n$*-module is a tilting module if and only if it is either projective or isomorphic to* $(S \amalg GT)^g$ *for some* $g \in C_n$, *some* $\tilde{\Lambda}_{n-1}$*-tilting module* T, *and some non-zero* $S \in \operatorname{add} S_1$. *For fixed* $g \in C_n$, *two* $\tilde{\Lambda}_{n-1}$*-tilting modules* T *and* T' *are equivalent if and only if* $(S_1 \amalg GT)^g$ *and* $(S_1 \amalg GT')^g$ *are equivalent.*

Proof. Let $T \in \operatorname{mod}\tilde{\Lambda}_n$ be a tilting module and suppose for simplicity that T is basic. Let $\dim(\operatorname{top} T) = (a_1, \ldots, a_n)$. Suppose first $a_i \neq 0$ for all i. We claim that in this case T is projective. In fact, T has a projective indecomposable direct summand, say P_i, since there is no tilting module of finite length. We have $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(P_{i+1}/U, P_i) \neq 0$ for all proper factors P_{i+1}/U of P_{i+1}. Thus P_{i+1} is a summand of T. Proceeding by induction, we see that T is projective. Now assume $a_n = 0$ and $a_1 \neq 0$. It is easily checked that this implies $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(T,S_1) = 0$ and $\operatorname{Ext}^1_{\tilde{\Lambda}_n}(S_1,T) = 0$. Thus T has a decomposition $T = T' \amalg S_1$ with $T' = GU$ for some module $\tilde{\Lambda}_{n-1}$-module U, by Lemma 4.5. Moreover, U is a tilting module. Thus any non-projective $\tilde{\Lambda}_n$-tilting module is of the form $(S \amalg GU)^g$ for some $g \in C_n$, some $\tilde{\Lambda}_{n-1}$-tilting module U, and some non-zero $S \in \operatorname{add} S_1$. The converse of this statement is an immediate consequence of Lemma 4.5. This completes the proof. □

5. The collection of all tilting modules

In this section we study the collection of all tilting modules over a fixed algebra Λ. We assume that $\operatorname{mod}\Lambda$ is a Krull-Schmidt category. Thus it is sufficient to study basic

tilting modules. Recall that an object is *basic*, if each indecomposable direct summand occurs exactly once in a direct sum decomposition. Let T and U be finitely presented tilting modules. One defines

$$T \leq U \iff T^\perp \subseteq U^\perp$$

where $T^\perp = \{X \in \operatorname{mod}\Lambda \mid \operatorname{Ext}^1_\Lambda(T, X) = 0\}$. This defines a partial ordering on the set of isomorphism classes of basic tilting modules which we denote by $\mathcal{T}(\Lambda)$.

There is an alternative description of this partial ordering because $T^\perp = \operatorname{Gen} T$ where $\operatorname{Gen} T$ denotes all factors of finite coproducts of copies of T in $\operatorname{mod}\Lambda$.

Lemma 5.1. *Suppose every indecomposable Λ-module is uniserial, that is, the lattice of submodules forms a chain. Then $T \leq U$ if and only every indecomposable summand of T is a factor of some indecomposable summand of U.*

The poset $\mathcal{T}(\Lambda)$ has been studied by various authors. Recent work of Happel and Unger [11] describes the Hasse diagram of this poset in terms of a graph defined by Riedtmann and Schofield [13].

We are also interested in the set $\mathcal{S}(\Lambda)$ of isomorphism classes of finitely presented Λ-modules which are faithful, basic, and selforthogonal. Recall that a module X is *selforthogonal* if $\operatorname{Ext}^1_\Lambda(X, X) = 0$. For $X, Y \in \mathcal{S}(\Lambda)$ we define $X \leq Y$ if X is isomorphic to a direct summand of Y.

From now on we fix $n \geq 1$ and assume that Λ is the completed path algebra of a quiver of type A_n or $\tilde{A}_{n-1}$ with linear orientation. Thus $\Lambda = \Lambda_n$ or $\Lambda = \tilde{\Lambda}_n$. The combinatorial analysis of $\mathcal{T}(\Lambda)$ is based on the description of the indecomposable Λ-modules via intervals. To each interval I in $\mathcal{I}(n)$ or $\tilde{\mathcal{I}}(n)$ we assign the indecomposable M_I. This is by definition the factor of the projective P_i of composition length $l-1$ where $i = \sup I$ and $l = \operatorname{card} I$. Note that the M_I provide a complete list of indecomposable Λ-modules. In order to describe $\mathcal{T}(\Lambda)$, we use the Tamari lattice $\mathcal{C}(n)$ and its variation $\tilde{\mathcal{C}}(n)$, which are defined and discussed in the appendix.

Theorem 5.2. *The assignment $X \mapsto M_X = \coprod_{I \in X} M_I$ induces isomorphisms*

$$\mathcal{C}(n) \xrightarrow{\sim} \mathcal{T}(\Lambda_n) \quad \text{and} \quad \tilde{\mathcal{C}}(n) \xrightarrow{\sim} \mathcal{T}(\tilde{\Lambda}_n)$$

of partially ordered sets.

Proof. The fact that both maps are well-defined bijections follows from the classification of the tilting modules for Λ_n in Proposition 3.2, and for $\tilde{\Lambda}_n$ in Corollary 4.3. For the partial ordering in $\mathcal{T}(\Lambda)$ we use the description given in Lemma 5.1. The lemma given below translates the factor relation between indecomposable Λ-modules into a relation between the corresponding intervals. The relation $I \twoheadrightarrow J$ between intervals is precisely the one used for the definition of the partial ordering on $\mathcal{C}(n)$ and $\tilde{\mathcal{C}}(n)$. Thus both maps respect the poset structure and the proof is complete. □

Lemma 5.3. *Let $I, J \in \mathcal{I}(n)$ or $I, J \in \tilde{\mathcal{I}}(n)$.*

(1) *There is a monomorphism $M_I \to M_J$ if and only if $I \rightarrowtail J$.*

(2) *There is an epimorphism $M_I \to M_J$ if and only if $I \twoheadrightarrow J$.*

Proof. Clear. □

Next we describe the cover relation in $\mathcal{T}(\Lambda)$. This is based on the analysis of $\mathcal{C}(n)$ and $\tilde{\mathcal{C}}(n)$ in the appendix.

Proposition 5.4. *Let $T, T' \in \mathcal{T}(\Lambda)$. Then T covers T' or T' covers T if and only if T and T' have precisely $n-1$ indecomposable direct summands in common.*

Proof. For Λ_n apply Lemma A.4, and for $\tilde{\Lambda}_n$ use Proposition B.2 to reduce from $\tilde{\mathcal{C}}(n)$ to $\mathcal{C}(n)$. □

Proposition 5.5. *For $T, T' \in \mathcal{T}(\Lambda)$ the following are equivalent:*

(1) *T covers T'.*
(2) *There are decompositions $T = T_0 \amalg X$ and $T' = T_0' \amalg X$ such that T_0 and T_0' are indecomposable with a monomorphism $T_0 \to X$ and an epimorphism $X \to T_0'$.*
(3) *There are decompositions $T = T_0 \amalg X$ and $T' = T_0' \amalg X$ such that T_0 and T_0' are indecomposable with a monomorphism $T_0 \to X_0$ and an epimorphism $X_0 \to T_0'$ for some indecomposable summand X_0 of X.*

Proof. Apply Lemma A.5 and Proposition B.2. □

We end this section with a description of $\mathcal{S}(\Lambda)$ which is the analogue of our results on $\mathcal{T}(\Lambda)$. We refer to the appendix for the definitions of $\mathcal{B}(n)$ and $\tilde{\mathcal{B}}(n)$.

Theorem 5.6. *The assignment $X \mapsto M_X = \coprod_{I \in X} M_I$ induces isomorphisms*

$$\mathcal{B}(n) \xrightarrow{\sim} \mathcal{S}(\Lambda_n) \quad \text{and} \quad \tilde{\mathcal{B}}(n) \xrightarrow{\sim} \mathcal{S}(\tilde{\Lambda}_n)$$

of partially ordered sets.

Proof. First observe that a Λ_n-module is faithful if and only if the indecomposable projective of maximal dimension appears as a direct summand. Thus a subset $X \subseteq \mathcal{I}(n)$ corresponds to a faithful and selforthogonal module M_X if and only if X belongs to $\mathcal{B}(n)$. This follows from Lemma 3.1.

Now let $\Lambda = \tilde{\Lambda}_n$. Observe that a $\tilde{\Lambda}_n$-module is faithful if and only if there is a non-zero projective direct summand. Thus every faithful selforthogonal module lies, up to a cyclic permutation, in the image of F, by Lemma 4.1. Note that $F(M_X) = M_{\pi^*(X)}$ for each $X \in \mathcal{B}(n)$. Thus F commutes with the embedding $\mathcal{B}(n) \to \tilde{\mathcal{B}}(n)$. We conclude that $X \mapsto M_X$ induces an isomorphism $\tilde{\mathcal{B}}(n) \to \mathcal{S}(\tilde{\Lambda}_n)$. □

Appendix A. Stasheff associahedra

Fix an integer $n \geq 1$. The *Stasheff associahedron* of dimension $n-1$ is a convex polyhedron whose faces are indexed by the meaningful bracketings of a string of $n+1$ letters [17, 20]. We shall identify the Stasheff associahedron with its poset of faces. This can be described as follows. Let $\mathcal{I}(n)$ be the set of intervals $[i, j] = \{i, i+1, \ldots, j\}$ in $\mathbb{Z}$ with $0 \leq i < j \leq n$. Two intervals I, J are said to be *compatible* if $I \subseteq J$ or

$J \subseteq I$ or $I \cap J = \emptyset$. Denote by $\mathcal{B}(n)$ the set of all subsets $X \subseteq \mathcal{I}(n)$ such that $[0,n] \in X$ and all intervals in X are pairwise compatible. The set $\mathcal{B}(n)$ is ordered by inclusion. In fact, $\mathcal{B}(n)$ is a lattice and we identify it with the lattice of faces of the Stasheff associahedron of dimension $n-1$ by identifying an interval $[i,j]$ with the bracketing $x_0 \dots (x_i \dots x_j) \dots x_n$ of the string $x_0 \dots x_n$. This identification is order reversing, that is, $X \subseteq Y$ in $\mathcal{B}(n)$ if and only if the face corresponding to X contains the face corresponding to Y. In particular, a set $X \in \mathcal{B}(n)$ of cardinality p corresponds to a face of dimension $n-p$. Note that the cardinality of a set in $\mathcal{B}(n)$ is bounded by n.

A *vertex* of $\mathcal{B}(n)$ is by definition an element in $\mathcal{B}(n)$ having cardinality n. The set of vertices of $\mathcal{B}(n)$ is denoted by $\mathcal{C}(n)$. Let us give an alternative description of the set of vertices. To this end define $\operatorname{top} X$ for each $X \subseteq \mathcal{I}(n)$ to be the sequence $(a_1, \dots, a_n)$ with $a_p = \operatorname{card}\{I \in X \mid \sup I = p\}$ for $1 \leq p \leq n$.

Lemma A.1. *The map sending $X \in \mathcal{C}(n)$ to $\operatorname{top} X$ induces a bijection between $\mathcal{C}(n)$ and the set of sequences $(a_1, \dots, a_n)$ of non-negative integers such that $\sum_i a_i = n$ and $\sum_{i \leq p} a_i \leq p$ for all $1 \leq p \leq n$. In particular, the cardinality of $\mathcal{C}(n)$ equals the Catalan number $C(n) = \frac{1}{n+1}\binom{2n}{n}$*

Proof. Identifying $X \in \mathcal{C}(n)$ with a bracketing of a string $x_0 \dots x_n$, the sequence $\operatorname{top} X = (a_1 \dots, a_n)$ represents the positions of the closing brackets. Clearly, $\operatorname{top} X$ satisfies $\sum_i a_i = n$ and $\sum_{i \leq p} a_i \leq p$ for all p. Moreover, each bracketing is determined by this data. □

We define the following relations on the set $\mathcal{I}(n)$ of intervals:

$$I \rightarrowtail I' \iff \inf I = \inf I' \text{ and } \operatorname{card} I \leq \operatorname{card} I';$$
$$I \twoheadrightarrow I' \iff \sup I = \sup I' \text{ and } \operatorname{card} I \geq \operatorname{card} I'.$$

Given subsets X and X' of $\mathcal{I}(n)$, we define $X \rightarrowtail X'$ if for each $I \in X$ there exists $I' \in X'$ with $I \rightarrowtail I'$. Analogously, $X' \twoheadrightarrow X$ if for each $I \in X$ there exists $I' \in X'$ with $I' \twoheadrightarrow I$.

Lemma A.2. *The set $\mathcal{C}(n)$ is partially ordered via*

$$X' \geq X \iff X' \twoheadrightarrow X.$$

Proof. Transitivity is clear. Now suppose $X \geq X' \geq X$. Both sets have cardinality n. The assumption implies that all intervals in $X \cup X'$ are pairwise compatible. Thus $X = X'$. □

Remark A.3. Let $X, X' \in \mathcal{C}(n)$. Then one can show that $X' \twoheadrightarrow X$ if and only if $X' \rightarrowtail X$.

It turns out that $\mathcal{C}(n)$ is in fact a lattice, which appears as *Tamari lattice* in the literature [18, 16]. The Tamari lattice can be described in many ways via the known bijections between families of Catalan objects. Our description seems to be new. It is

related to the usual definition via the covering relation in $\mathcal{C}(n)$. Recall that an element x in a poset covers another element x' if $\{y \mid x \geq y \geq x'\} = \{x, x'\}$.

Lemma A.4. *Let $X, X' \in \mathcal{C}(n)$. Then X covers X' or X' covers X if and only if $X \cap X'$ has cardinality $n-1$.*

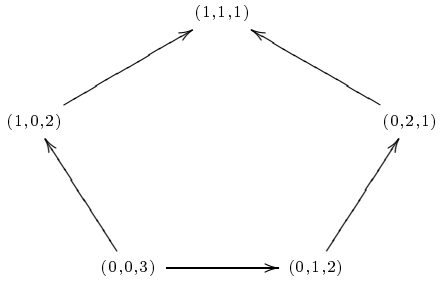
Lemma A.5. *Let $X, X' \in \mathcal{C}(n)$ and $Y = X \cap X'$. Then the following are equivalent:*

(1) *X covers X'.*
(2) *Y has cardinality $n-1$ and $X \rightarrowtail Y \twoheadrightarrow X'$.*
(3) *There are intervals I, I' such that $X = Y \cup \{I\}$ and $X' = Y \cup \{I'\}$. Moreover, $I \cup I' \in Y$ and $I \rightarrowtail (I \cup I') \twoheadrightarrow I'$.*

The proofs of Lemma A.4 and Lemma A.5 are elementary, but rather technical and therefore omitted. A key observation is the following. Given $I \in X \in \mathcal{C}(n)$ with $I \neq [0, n]$, there exists $I' \in X \setminus \{I\}$ such that either $I \rightarrowtail I'$ or $I' \twoheadrightarrow I$.

Corollary A.6. *The Hasse diagram of the Tamari lattice $\mathcal{C}(n)$ equals the 1-skeleton of the Stasheff associahedron $\mathcal{B}(n)$.*

The following figure shows the Hasse diagram of $\mathcal{C}(3)$.



Appendix B. Circular associahedra

Fix an integer $n \geq 1$. We need some notation. Given $X \subseteq \mathbb{Z}$ and $z \in \mathbb{Z}$, we define $X + z = \{x + z \mid x \in X\}$. This definition extends to subsets $X \subseteq 2^{\mathbb{Z}}$ and $X \subseteq 2^{(2^{\mathbb{Z}})}$.

Let $\mathcal{I}$ be the set of possibly infinite intervals $I \subseteq \mathbb{Z}$ with $\sup I < \infty$. Two intervals I and J are said to be *n-equivalent* if there exists $z \in \mathbb{Z}$ such that $J = I + zn$. We denote by $\tilde{\mathcal{I}}(n)$ the set of equivalence classes of n-equivalent intervals from $\mathcal{I}$. Next consider the projection

$$\pi \colon \mathbb{Z} \longrightarrow \{0, 1, 2, 3, \dots\}, \quad z \mapsto \begin{cases} z & \text{if } z \geq 0, \\ 0 & \text{if } z < 0. \end{cases}$$

This induces an injective map $\pi^* \colon \mathcal{I}(n) \to \tilde{\mathcal{I}}(n)$ which takes $I \in \mathcal{I}(n)$ to the equivalence class of $\pi^{-1}(I)$. We define $\tilde{\mathcal{B}}(n)$ to be the set of subsets of $\tilde{\mathcal{I}}(n)$ which are of the form $\pi^*(X) + z$ for some $X \in \mathcal{B}(n)$ and some $z \in \mathbb{Z}$. Thus we have an injective map

$$\mathcal{B}(n) \longrightarrow \tilde{\mathcal{B}}(n), \quad X \mapsto \pi^*(X),$$

and viewing this as an identification, we get

$$\tilde{\mathcal{B}}(n) = \bigcup_{p=0}^{n-1} \mathcal{B}(n) + p.$$

We note that $\tilde{\mathcal{B}}(n)$ is partially ordered by inclusion.

A *vertex* of $\tilde{\mathcal{B}}(n)$ is by definition an element in $\tilde{\mathcal{B}}(n)$ having cardinality n. The set of vertices of $\tilde{\mathcal{B}}(n)$ is denoted by $\tilde{\mathcal{C}}(n)$. Each $X \in \tilde{\mathcal{B}}(n)$ is a set of equivalence classes of intervals in $\mathbb{Z}$. Thus we can define $\operatorname{top} X = (a_1, \dots, a_n)$ with $a_p = \operatorname{card}\{I \in X \mid p \equiv \sup I \pmod n\}$ for $1 \leq p \leq n$. Note that for each $I \in \tilde{I}(n)$, the values $\inf I$ and $\sup I$ are well-defined modulo n.

Lemma B.1. *The map sending $X \in \tilde{\mathcal{C}}(n)$ to $\operatorname{top} X$ induces a bijection between $\tilde{\mathcal{C}}(n)$ and the set of sequences $(a_1, \dots, a_n)$ of non-negative integers such that $\sum_i a_i = n$. In particular, the cardinality of $\tilde{\mathcal{C}}(n)$ equals $\binom{2n-1}{n-1}$.*

Proof. We use the embedding $\mathcal{C}(n) \to \tilde{\mathcal{C}}(n)$ via π^* and the description of $\mathcal{C}(n)$ via integer sequences in Lemma A.1. Given a sequence $(a_1, \dots, a_n)$, there is a cyclic permutation $(a_k, \dots, a_{k-1})$ such that $\sum_{i=1}^{p} a_{k+i-1} \leq p$ for all $1 \leq p \leq n$. Thus each sequence is of the form $\operatorname{top} X$ for some $X \in \tilde{\mathcal{C}}(n)$. On the other hand, two elements X, X' in $\mathcal{C}(n)$ get identified in $\tilde{\mathcal{C}}(n)$ after a cyclic permutation, that is $\pi^*(X') = \pi^*(X) + p$ for some p, if and only if $\operatorname{top} X'$ is a cyclic permutation of $\operatorname{top} X$. □

We define the following relations on the set $\tilde{\mathcal{I}}(n)$ of intervals:

$$\begin{aligned} I \rightarrowtail I' &\iff \inf I = \inf I' \pmod n \text{ and } \operatorname{card} I \leq \operatorname{card} I'; \\ I \twoheadrightarrow I' &\iff \sup I = \sup I' \pmod n \text{ and } \operatorname{card} I \geq \operatorname{card} I'. \end{aligned}$$

As in Section A, this induces relations $X \rightarrowtail X'$ and $X \twoheadrightarrow X'$ for subsets X, X' of $\tilde{\mathcal{I}}(n)$. Moreover, one obtains a partial ordering on the set $\tilde{\mathcal{C}}(n)$ via

$$X' \geq X \iff X' \twoheadrightarrow X.$$

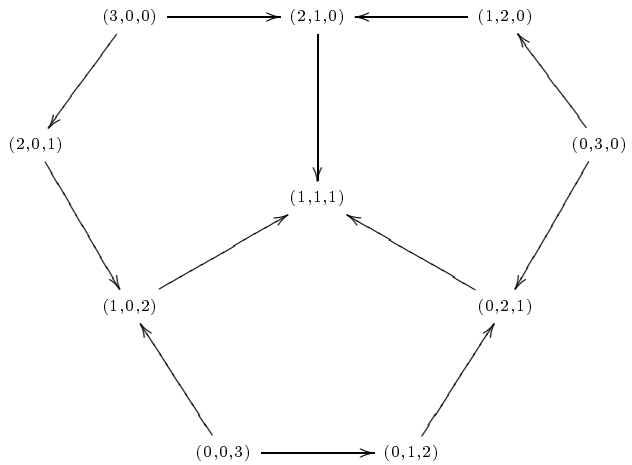
Next we describe the poset structure of $\tilde{\mathcal{C}}(n)$. We use two approaches: a description via $\mathcal{C}(n)$ and a description via $\tilde{\mathcal{C}}(n-1)$. It is convenient to identify each element X in $\mathcal{C}(n)$ or $\tilde{\mathcal{C}}(n)$ with the integer sequence $\operatorname{top} X$. We define $\mathbf{1} = (1, \dots, 1)$ and for each $i \in \{1, \dots, n\}$ we denote by $\mathbf{0}_i$ the sequence $(a_1, \dots, a_n)$ with $a_i = n$ and $a_j = 0$ for $j \neq i$.

Proposition B.2. *The poset $\tilde{\mathcal{C}}(n)$ has the following properties:*

1. $\mathbf{1}$ *is the unique maximal element.*
2. $\{\mathbf{0}_i \mid 1 \leq i \leq n\}$ *is the set of minimal elements.*
3. *Each set of elements has a supremum.*
4. *The natural embedding $\mathcal{C}(n) \to \tilde{\mathcal{C}}(n)$ induces an isomorphism of posets between $\mathcal{C}(n)$ and the interval $[\mathbf{0}_n, \mathbf{1}]$.*

Proof. The assertions follow from some elementary properties of the embedding $\mathcal{C}(n) \to \tilde{\mathcal{C}}(n)$. This embedding sends X to $\pi^*(X)$ and we observe that $\operatorname{top} X = \operatorname{top} \pi^*(X)$. Moreover $X \leq Y$ in $\mathcal{C}(n)$ if and only if $\pi^*(X) \leq \pi^*(Y)$. Finally, we note that each $X \in \tilde{\mathcal{C}}(n)$ contains at least one infinite interval, say I with $i = \sup I$, and this implies $\mathbf{0}_i \leq X$. □

The following figure shows the Hasse diagram of $\tilde{\mathcal{C}}(3)$.



Proposition B.3. *Let $n > 1$. The map*

$$\tilde{\mathcal{C}}(n-1) \longrightarrow \tilde{\mathcal{C}}(n), \quad (a_1, \ldots, a_{n-1}) \mapsto (a_1 + 1, a_2, \ldots, a_{n-1}, 0)$$

induces an isomorphism of posets onto its image. Moreover, the image is interval closed.

Proof. The assertion follows from an explicit description of the embedding $\tilde{\mathcal{C}}(n-1) \to \tilde{\mathcal{C}}(n)$. The map sends $X \in \tilde{\mathcal{C}}(n-1)$ to $\alpha(X) \cup \{S\}$, where S is the n-equivalence class of the interval $[0, 1]$, and $\alpha \colon \tilde{\mathcal{I}}(n-1) \to \tilde{\mathcal{I}}(n)$ sends the $n-1$-equivalence class of an interval $I \subseteq \mathbb{Z}$ with $\sup I \in \{1, \ldots, n-1\}$ to the n-equivalence class of the interval $I' \subseteq \mathbb{Z}$ with $\sup I' = \sup I$ and

$$\operatorname{card} I' = \begin{cases} \operatorname{card} I & \text{if } \operatorname{card} I \leq \sup I, \\ 1 + \operatorname{card} I & \text{if } \operatorname{card} I > \sup I. \end{cases}$$

Note that $I \twoheadrightarrow J$ if and only if $\alpha(I) \twoheadrightarrow \alpha(J)$. □

Corollary B.4. *Viewing the injective maps $\mathcal{C}(n) \to \tilde{\mathcal{C}}(n)$ and $\tilde{\mathcal{C}}(n-1) \to \tilde{\mathcal{C}}(n)$ as identifications, we have*

$$\tilde{\mathcal{C}}(n) = \bigcup_{p=0}^{n-1} \mathcal{C}(n) + p \qquad \text{and} \qquad \tilde{\mathcal{C}}(n) \setminus \{\mathbf{1}\} = \bigcup_{p=0}^{n-1} \tilde{\mathcal{C}}(n-1) + p.$$

The first equation says that the poset $\tilde{\mathcal{C}}(n)$ is the union of n copies of the Tamari lattice $\mathcal{C}(n)$. Kiyoshi Igusa pointed out to us that this fact can be expressed numerically by the following inclusion-exclusion formula. Note that the cardinality of $\tilde{\mathcal{C}}(n)$ is $\binom{2n-1}{n-1}$, wheras the cardinality of $\mathcal{C}(n)$ is the Catalan number $C(n) = \frac{1}{n+1}\binom{2n}{n}$.

$$\binom{2n-1}{n-1} = \sum_{i=1}^{n} (-1)^{i-1} \frac{n}{i} \sum_{n_1+\ldots+n_i=n} C(n_1)C(n_2)\ldots C(n_i) \tag{B.1}$$

Note that all n_j in this formula are positive integers. We do not know whether the Hasse diagram of $\tilde{\mathcal{C}}(n)$ arises as the 1-skeleton of a polytope.

References

[1] Auslander, M.: *Functors determined by objects*, in: Representation theory of algebras, ed. R. Gordon, Marcel Dekker (1978), 1–244.

[2] van den Bergh, M. and Reiten, I.: *Noetherian hereditary abelian categories with Serre duality*, J. Amer. Math. Soc. **15** (2002), 295-366.

[3] Bongartz, K.: *Tilted algebras*, in: Representations of algebras (Puebla, 1980), Lecture Notes in Math. **903**, Springer, Berlin-New York (1981), 26–38.

[4] Bongartz, K. and Gabriel, P.: *Covering spaces in representation-theory*, Invent. Math. **65** (1981/82), 331–378.

[5] Buan, A. and Krause, H.: *Cotilting modules over tame hereditary algebras*, Pacific J. Math., to appear.

[6] Colpi, R.: *Tilting in Grothendieck categories*, Forum Math. **11** (1999), 735-759.

[7] Crawley-Boevey, W. W.: *Modules of finite length over their endomorphism ring*, in: Representations of algebras and related topics, eds. S. Brenner and H. Tachikawa, London Math. Soc. Lec. Note Series **168** (1992), 127–184.

[8] Crawley-Boevey, W. W.: *Locally finitely presented additive categories*, Comm. Algebra **22** (1994), 1644–1674.

[9] Gabriel, P.: *Des catégories abéliennes*, Bull. Soc. Math. France **90** (1962), 323–448.

[10] Happel, D.: *A characterization of hereditary categories with a tilting object*, Invent. Math. **144** (2001), 381–398.

[11] Happel, D. and Unger, L.: *On a partial order of tilting modules*, preprint.

[12] Marsh, R.; Reineke, M. and Zelevinski, A.: *Generalized associahedra and quiver representations*, preprint.

[13] Riedtmann, C. and Schofield, A.: *On a simplicial complex associated with tilting modules*, Comment. Math. Helv. **66** (1991), 70–78.

[14] Ringel, C. M.: *Finite dimensional hereditary algebras of wild representation type*, Math. Z. **161** (1978), 235–255.

[15] Schofield, A.: *Representations of rings over skew fields*, London Math. Soc. Lec. Note Series **92** (1985).

[16] Stanley, R. P.: *Enumerative combinatorics, Volume 2*, Cambridge Univ. Press (1999).

[17] Stasheff, J. D.: *Homotopy associativity of H-spaces I*, Trans. Amer. Math. Soc. **138** (1963), 275–292.

[18] Tamari, D.: *The algebra of bracketings and their enumeration*, Nieuw Arch. Wisk. **10** (1962), 131–146.

[19] The On-Line Encyclopedia of Integer Sequences, `www.research.att.com/~njas/sequences/`

[20] Ziegler, G. M.: *Lectures on Polytopes*, Springer Verlag, New York (1995).

Aslak Bakke Buan, Institutt for matematiske fag, NTNU, N-7491 Trondheim, Norway
E-mail address: aslakb@math.ntnu.no

Henning Krause, Department of Pure Mathematics, University of Leeds, Leeds LS2 9JT, United Kingdom
E-mail address: henning@maths.leeds.ac.uk

Cluster Algebras, Somos Sequences and Exchange Graphs

Gregg Musiker
musiker@fas.harvard.edu
(617) 493–2447

Supervised by Richard Stanley of the Massachusetts Institute of Technology.

A thesis presented to the Department of Mathematics
in partial fulfillment of the requirements
for the degree of Bachelor of Arts with Honors

Harvard University
Cambridge, Massachusetts
April 1, 2002

Abstract

In this thesis, we will investigate the theory of cluster algebras, a recently created combinatorial theory that is still developing. Cluster algebras are not only intrinsically interesting, but have useful applications to the theory of Somos sequences and Laurent polynomials, generalized associahedra and many other fields. We will concentrate on an axiomatic development of cluster algebras, motivating them by their aforementioned applications. We will end with several open problems and conjectures. This exposition will utilize semisimple Lie algebras and root systems; however, the necessary results from these mathematical areas will be presented here and developed as needed. This should be accessible to anyone familiar with graph theory and recurrence relations.

Contents

1 Introduction

This thesis surveys the work of Sergey Fomin and Andrei Zelevinksy in the development of cluster algebras. Let us spend a moment explaining the significance of this theory. Their theory of cluster algebras is a unifying framework

which has produced more and more applications the more it is developed. Prior to the theory of cluster algebras, the sequences Somos-4 and Somos-5 had been proven to be integer sequences by several people including Janice Malouf and George Bergman [10]. The integrality of Somos-6 and Somos-7 had been proven by Raphael Robinson [10]. However, the method of cluster algebras provides a unified proof for the integrality of Somos-4 through Somos-7 as well as the integrality of a number of other sequences as described in [8]. More importantly, cluster algebras hint at a deep connection between this solution in the area of Laurent polynomial theory and their solution to a problem concerning the explicit factorization of totally positive matrices into elementary Jacobi matrices [6, 25]. Connections between cluster algebras and algebraic topological objects such as the associahedron have also been discovered more recently [3]. Though the theory has surprising applications, Zelevinsky (personal communication) has stated that he is most excited by the intrinsic beauty and elegance of the theory; they are an interesting object of study in their own right.

Fomin and Zelevinsky were motivated to create cluster algebras based on empirical properties of the dual canoncial bases found in total positivity theory. We will discuss this connection more in the appendix, however, our main focus will be the applications to Somos sequences and the properties of exchange graphs.

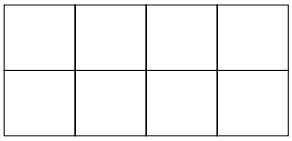
Acknowledgements. I am deeply indebted to my thesis advisor, Richard Stanley, for all of his guidance. For all of his suggestions and editorial suggestions, I have great gratitude. Also I very much appreciate Sergey Fomin's and Andrei Zelevinsky's willingness to discuss their theory more in depth with me. Sergey Fomin helped me better understand the beautiful geometry behind the theory of cluster algebras, and Zelevinsky showed me the intrinsic elegance of this developing theory. This year, I have been involved with a research group, REACH (Research Experiences in Algebraic Combinatorics at Harvard). My experience during this project has been invaluable for the completion of this exposition. I wish to thank all of the members of REACH, especially Jim Propp, who's encouragement as the group leader and expertise assisted me greatly. His insights concerning Somos sequences were particularly helpful. In addition, I am very thankful for the aid of David Speyer, another member of REACH, and his insights about Laurent polynomials. Lastly, I would like to thank my friends Eiichi Miyasaka and Harvey Wun for their editorial and technical support.

2 Laurentness and Somos Sequences

Consider the sequence $f_n = \frac{f_{n-1}^2+1}{f_{n-2}}$ ($f_n f_{n-2} = f_{n-1}^2 + 1$). At first glance, this sequence appears to be a sequence of non-integral rational numbers, even if one lets $f_0 = f_1 = 1$. However, after computing several terms of the sequence, one finds that $f_2 = 2, f_3 = 5, f_4 = 13, f_5 = 34, \ldots$. Not only are these all integers, but they are every other Fibonacci number. One might believe this pattern continues despite the denominator in the recursion.

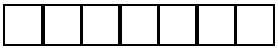
In fact this pattern will continue. There is a trivial proof by induction, but for our purposes, a proof by combinatorial interpretation is more edifying.

We define $G_{m,n}$ to be the $m \times n$ grid graph where there are m vertices in each column and each row has n vertices.



The grid graph $G_{3,5}$.

Let f_n be the number of perfect matchings in $G_{2,2(n-1)}$ such as

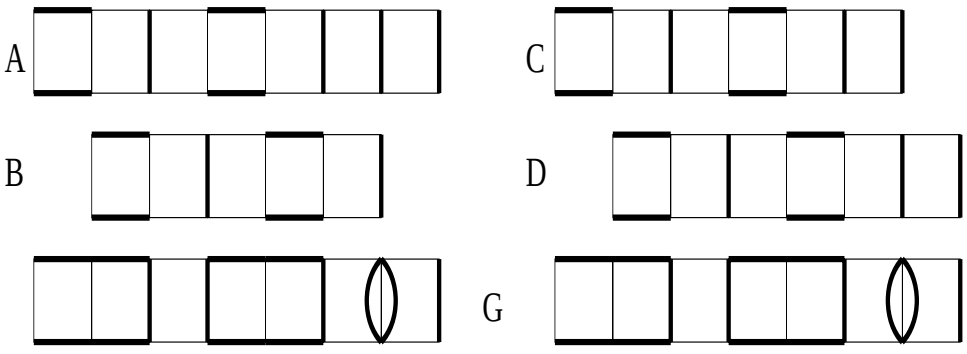


for $n = 4$. By convention we will set $f_0 = f_1 = 1$ and one can readily check that $f_2 = 2$. One can show using Eric Kuo's technique of *graphical condensation* that f_n satisfies the recurrence $f_n f_{n-2} = f_{n-1}^2 + 1$ [17].

The following proof is from [20] based on [17]. Consider the set of ordered pairs $(A, B) \in T_n \times T_{n+2}$ where T_n is the set of perfect matchings of $G_{2,2(n-1)}$. Since $|T_n| = f_n$, the number of such pairs is exactly $f_n f_{n+2}$. Similarly the set of pairs (C, D) from $T_{n+1} \times T_{n+1}$ will have cardinality f_{n+1}^2.

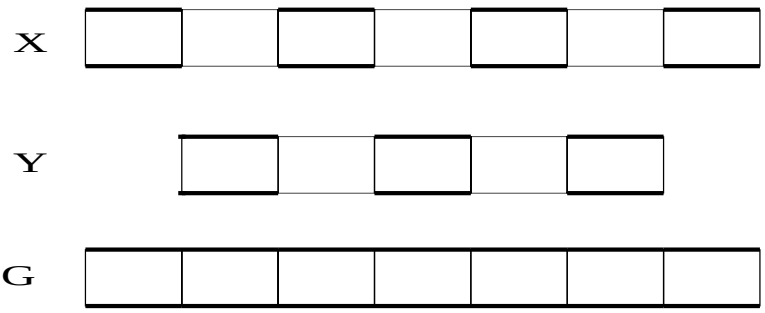
Lemma 1 *There is a bijection from $T_n \times T_{n+2} - (X, Y)$ to $T_{n+1} \times T_{n+1}$ where X and Y are specific instances of perfect matchings as pictured below.*

Sketch of Proof. One can superimpose a matching A and a matching B onto a $2 \times 2(n+1)$ grid graph G with distinguished edge set M_{AB} (allowing double edges) so that the matching B is centered on G. Each vertex of G (except those on the outer boundary) will have two distinguished edges emanating from it. Similarly one can superimpose matchings C and D onto the same $2 \times 2(n+1)$ grid graph G with distinguished edge set M_{CD} where C is left-justified and D is right-justified with respect to G.



An example of such a superposition.

Notice that the double matching on graph G can be decomposed into the matchings (A, B) or the matchings (C, D). This decomposition is not unique. However the number of decomposition into (A, B) is the same as the number of decompositions into a pair (C, D).

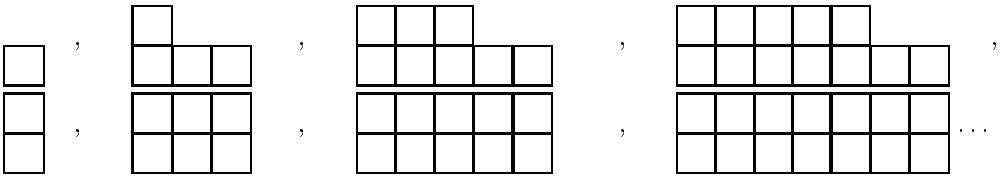


An undecomposable pair.

The correspondence between decompositions will be valid for all pairs of matchings but one, (X, Y). For the pair (X, Y), it is not possible to superimpose (X, Y) together on G and then decompose it into two matchings of left- and right- justified graphs. For all other pairs, there is a bijection (counting multiplicities) between the two decompositions, hence there is a bijection between $T_n \times T_{n+2} - (X, Y)$ and $T_{n+1} \times T_{n+1}$.

Since f_n is a function that counts an actual object, it is clear that f_n must be a nonnegative integer for all $n \geq 1$.

Similar techniques work for sequences such as $g_n g_{n-3} = g_{n-1} g_{n-2} + 1$ where $g_0 = g_1 = g_2 = 1$, $\{g_n : n \geq 3\} = 2, 3, 7, 11, 26, 41, 97, 153, \ldots$. In fact, this counts the number of perfect matchings of the family of graphs:



Ira Gessel [11] noticed that the sequence $\{g_{2n}\} = (1, 3, 11, 41, \ldots)$ appeared on Neil Sloane's website, the Encyclopedia of Integer Sequences [22]. On this site, the sequence was noted to have the combinatorial interpretation of counting domino tilings of a $3 \times 2(n-1)$ rectangle, which implies it counts the number of perfect matchings of $G_{3,2(n-1)}$. Eric Kuo noted that the terms $\{g_{2n+1}\} = (1, 2, 7, 26, 97, \ldots)$ counting the number of "mutilated" $3 \times 2(n-1)$ grid graphs [16]. By mutilated $3 \times 2(n-1)$ grid graphs, we mean graphs resembling the ones in the top row of the previous figure, i.e. they are $3 \times 2(n-1)$ grid graphs where the rightmost two vertices in the top row, along with their incident edges, have been removed.

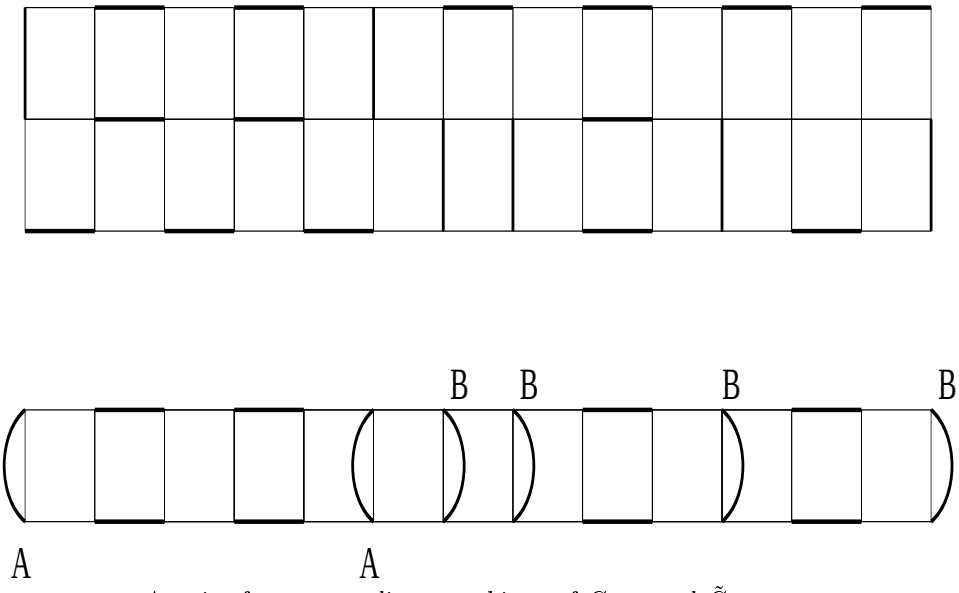
The following is an original proof of a direct bijection between the perfect matchings of $G_{3,2(n-1)}$ and the perfect matchings of $\tilde{G}_{2,2(n-1)}$, a $2 \times 2(n-1)$ grid multi-graph where each vertical edge has been replaced with two vertical edges (labeled A and B) and the vertical edges are paired off so that each pair of consecutive vertical edges in the matching use the same label.

The bijection is as follows: whenever a vertical edge appears in a matching M of $G_{3,2(n-1)}$, it will either be an edge from the 2nd row to the 3rd row, or the 1st row to the 2nd. If it is from the 2nd to the 3rd, then the corresponding matching of the $2 \times 2(n-1)$ grid multi-graph $\tilde{G}_{2,2(n-1)}$ has the vertical edge labeled A in the corresponding column. If it is from the 1st to the 2nd, use the edge labeled B.

Claim 1 *Once these vertical edges have been specified there is a unique choice of horizontal edges that will complete M to a perfect matching.*

Claim 2 *Each consecutive pair of vertical edges will be in the same row.*

These claims are easily verified by studying the possible perfect matchings of $G_{3,2(n-1)}$.



A pair of corresponding matchings of $G_{3,14}$ and $\tilde{G}_{2,14}$.

Furthermore, the number of perfect matchings in $\tilde{G}_{2,2(n-1)}$ is the same as the weighted number of perfect matchings in $G_{2,2(n-1)}$ where we give a matching that uses m vertical edges weight 2^m. Let $\tilde{f}_n$ be the number of perfect matchings in $\tilde{G}_{2,2(n-1)}$.

Using graphical condensation, one can show that just as f_n satisfies the recurrence $f_n f_{n-2} = f_{n-1}^2 + 1$, $\tilde{f}_n$ satisfies the recurrence $\tilde{f}_n \tilde{f}_{n-2} = \tilde{f}_{n-1}^2 + 2$.

In fact if $f_{n,w}$ is the weighted number of perfect matchings in $G_{2,2(n-1)}$ where we give a matching that uses m vertical edges weight w^m, then $f_{n,w}f_{n-2,w} = f_{n-1,w}^2 + w$. Consequently, every other term of the sequence g_n ($g_{2n} = 1, 3, 11, 41, \ldots$) satisfies the recurrence $g_{2n}g_{2n-4} = g_{2n-2}^2 + 2$.

2.1 Somos Sequences

So far we've seen two rational recurrences give rise to integer sequences. What about the sequence

$$s_n s_{n-4} = s_{n-1}s_{n-3} + s_{n-2}^2$$

where $s_1 = s_2 = s_3 = s_4 = 1$? This sequence is called Somos-4 where a general Somos-k sequence is a sequence of the form $S_nS_{n-k} = S_{n-1}S_{n-k+1} + S_{n-2}S_{n-k+2} + \ldots$. Such sequences were discovered by Michael Somos while he was studying recurrences resembling relations found among elliptic functions. More can be found about Somos sequences in David Gale's article [10] or Jim Propp's website [19]. Somos-4 is in fact a sequence of positive integers, however assigning a combinatorial interpretation to s_n (like in the case of f_n or g_n) was an open problem until recently.[1]

As mentioned in the introduction, one way to prove the integrality for the sequence Somos-4 involves using cluster algebras. Fomin and Zelevinsky in fact can prove a much more general result using their technique [8]. Before describing the use of cluster algebras to prove Laurentness, we will consider a simpler problem based on the work of David Speyer, an example which is also a special case of the Laurent phenomenon discussed in [8]. The following is David Speyer's proof from an email to REACH [23].

Consider a sequence x_n that satisfies the recurrence $x_n x_{n-2} = p(x_{n-1})$ for $n \geq 3$ where $p(t)$ is a univariate polynomial.

Definition 1 A *Laurent polynomial* over the variables $x_1, \ldots, x_n$ is a finite sum of terms where the variables $x_1^{\pm 1}, \ldots, x_n^{\pm 1}$ appear rather than just $x_1, \ldots, x_n$ as in the case of a polynomial.

Another way to think of a Laurent polynomial is as a rational function in $x_1, \ldots, x_n$ where the denominator consists of a single monomial. Let $R = \mathbb{Q}[x_1^{\pm 1}, x_2^{\pm 1}]$ be the ring of Laurent polynomials in the variables x_1 and x_2 with coefficients in $\mathbb{Q}$.

$$\text{When is } x_n \in R \ \ \forall\ n \geq 1?$$

Proposition 1 *As long as $p(0) \neq 0$, all of the $x_n \in R$ if and only if $p(t) = c \cdot t^{\deg p} \cdot p\left(\frac{p(0)}{t}\right)$ for some $c \in \mathbb{Q}$.*

[1] As of March, 2002, this was solved by members of REACH in work to be written up. Like f_n and g_n, the combinatorial interpretation of s_n involves perfect matchings of a family of graphs. Bousquet-Mélou and West also just recently found a combinatorial interpretation using an earlier suggestion from Jim Propp.

Proof. Assume $x_n \in R \ \ \forall\ n \geq 1$ but that $p(t) \ \neq \ c \cdot t^{\deg p} \cdot p\left(\frac{p(0)}{t}\right) \ \forall \ \ c \ \in \mathbb{Q}$. Then,

$$x_5 = \frac{p\left(\frac{p\left(\frac{p(x_2)}{x_1}\right)}{x_2}\right)}{\frac{p(x_2)}{x_1}}$$

is Laurent only if $x_1 \cdot p\left(\frac{p\left(\frac{p(x_2)}{x_1}\right)}{x_2}\right) \equiv 0 \mod p(x_2)$ in R. Using $p(x_2) \equiv 0$, this requirement reduces to $x_1 \cdot p\left(\frac{p(0)}{x_2}\right) \equiv 0 \mod p(x_2)$. Since x_1 and x_2 are units in R, we obtain $p(x_2) \Big| \left(x_1^{k_1} x_2^{k_2} \cdot p\left(\frac{p(0)}{x_2}\right)\right)$ for some choices of k_1, k_2. The variable x_1 does not appear in $p(x_2)$ so k_1 must equal 0. Furthermore, the degrees (in terms of x_2) of $p(x_2)$ and $x_2^{k_2} \cdot p\left(\frac{p(0)}{x_2}\right)$ only match if $k_2 = \deg p$. However, in this case, $p(x_2) | x_2^{\deg p} p\left(\frac{p(0)}{x_2}\right)$ implies there exists a $c \in \mathbb{Q}$ such that $p(t) = \ c \cdot t^{\deg p} \cdot p\left(\frac{p(0)}{t}\right)$, a contradiction.

Now assume $p(t) \ = \ c \cdot t^{\deg p} \cdot p\left(\frac{p(0)}{t}\right)$ for some $c \in \mathbb{Q}$. $x_3 = \frac{p(x_2)}{x_1}$ and $x_4 = \frac{p\left(\frac{p(x_2)}{x_1}\right)}{x_2}$ are in R along with x_1 and x_2. From this base case of four elements we will inductively show that all $x_n \in R$.

Claim 3 *Suppose $x_{n+1}, x_{n+2}, x_{n+3}$, and $x_{n+4} \in R$. Then $x_{n+5} \in R$.*
By the defining recurrence of the sequence, $x_{n+2}x_{n+4} = p(x_{n+3})$ which is equivalent to $p(0) \mod x_{n+3}$. The term $p(0)$ is nonzero and rational therefore $p(0)$ is a unit which implies that x_{n+2} and x_{n+4} are also units. Now we can divide freely by x_{n+2} and x_{n+4}. Thus

$$p(x_{n+4}) \equiv p\left(\frac{p(0)}{x_{n+2}}\right) \equiv \frac{1}{c x_{n+2}^{\deg p}} p(x_{n+2}) \equiv$$

$$\frac{1}{c x_{n+2}^{\deg p}} x_{n+1} x_{n+3} \equiv 0 \mod x_{n+3}.$$

Consequently $x_{n+5} = \frac{p(x_{n+4})}{x_{n+3}} \in R$. Given this claim, $x_n \in R \ \ \forall\ n \geq 1$. $\square$

We will later show this result holds if we let $R = A[x_1^{\pm 1}, x_2^{\pm 1}]$ where A is any unique factorization domain. In particular, we could allow A to be $\mathbb{Z}$ and the first two terms x_1, x_2 to be 1. In this case we recover integrality. Thus Laurentness is a more general condition than integrality. Fomin and Zelevinsky's result concerns the question of whether or not all terms of a sequence are Laurent polynomials in terms of the k initial terms. Thus they are able to prove that a sequence satisfying the Somos-4 recurrence $x_0 x_4 = x_1 x_3 + x_2^2$ is a sequence of Laurent polynomials in the initial four terms. In the case that $x_1 = x_2 = x_3 = x_4 = 1$, we get that Somos-4 is a sequence of integers. To understand their proof, we will now introduce the theory of cluster algebras.

2.2 Fomin and Zelevinsky's Definitions

Unless otherwise noted, the material from this section is directly from or based on [7]. Fomin and Zelevinsky define a *cluster algebra* $\mathcal{A}$ as "a commutative ring with unit and no zero divisors, equipped with a distinguished family of generators called *cluster variables*" [7, pg. 1]. The cluster algebra is a (non-disjoint) union of a distinguished collection of subsets called *clusters*. Each of the subsets in this collection have equal size, and this size is known as the rank of $\mathcal{A}$. For every cluster $X = \{x_1, \ldots, x_n\} \subset \mathcal{A}$ in a cluster algebra $\mathcal{A}$ of rank n there exist n clusters $Y_1, Y_2, \ldots, Y_n \subset \mathcal{A}$ adjacent to X. These clusters are adjacent to X because X and each $Y_i = \{x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n\}$ are related by a *binomial exchange relation*

$$x_i y_i = M_i(X) + M_i(Y_i), \tag{1}$$

where $M_i(X)$ and $M_i(Y_i)$ are two relatively prime monomials in the $n-1$ variables $X - \{x_i\}$. For example, the monomial[2] $M_i(X)$ is given by

$$M_i(X) = c_i(X) \prod_{1 \leq j \leq n, j \neq i} x_j(X)^{b_{ij}(X)}$$

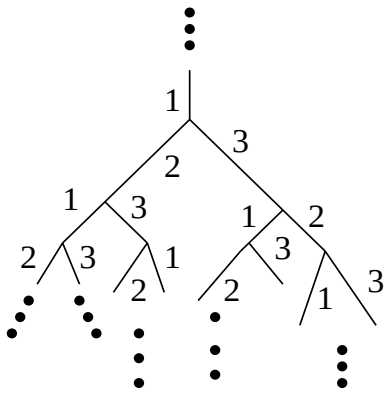
and for a general cluster C, the associated monomial is

$$M_i(C) = c_i(C) \prod_{z \in C} z^{b_{i,z}(C)}.$$

Furthermore, one can switch between any two clusters of $\mathcal{A}$ by a series of such exchanges. Besides the condition that $M_i(C)$ cannot depend on the i^{th} variable of the cluster C, the choice of a family of monomials is restricted by specific axioms which we will explain after some initial definitions.

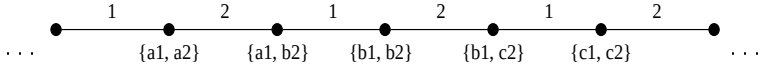
For any cluster algebra of rank n we define an n-regular graph called an *exchange graph* whose vertices are the different clusters, and whose edges correspond to the exchanges between two clusters. If $\mathcal{A}$ is a rank 1 cluster algebra, the only possible exchange graph is a 1-regular graph consisting of two vertices. When $n \geq 2$ and there are no relations between the variables of the various clusters, this graph will be an *exchange tree*, an infinite n-degree graph such that each of the n edges coming out of a given vertex have a unique label out of $\{1, \ldots, n\}$.

[2] Fomin and Zelevinsky allow the coefficients $c_i(X)$ to be chosen from a torsion-free multiplicative abelian group $\mathbb{P}$. However, for the remainder of this exposition, we will assume all of the $c_i(X)$'s are 1, i.e. that $\mathbb{P} = \{1\}$.



An exchange tree for a rank 3 cluster algebra.

Also, an exchange tree $\mathbb{T}$ for a rank 2 cluster algebra is a line.



Notice that whenever $\{x_1, y_2\}$ connects to $\{w_1, z_2\}$ via an edge labeled 1, then $y_2 = z_2$ and if an edge labeled 2 connects them, $x_1 = w_1$. Furthermore, we can define an *exchange pattern* β, a family of exchange binomials $\{B_t\}$, so that $x_1 w_1 = B(y_2)$ for some $B \in \beta$ when edge 1 connects them and $y_2 z_2 = B'(x_1)$ for $B' \in \beta$ when edge 2 connects them. Here I emphasize that the dependent variable of the binomial is determined by the edge label.

One can more formally define the possible exchange patterns that can be associated to a cluster algebra. Here we will assume that the exchange graph is an undirected tree of degree n. We will let $\mathcal{T}$ be the set of vertices in the exchange tree, and will use the notation $E_i(t, t')$ to signify that vertices t and t' are connected by an edge labeled i.

Then if $E_i(t, t')$ we will let the exchange binomial associated with this edge be $M_i(t) + M_i(t')$ where we will let the vertices t, t' stand for the associated clusters. For $\mathcal{A}$ to be a cluster algebra, the exchange pattern $\{M_i(t) : i \in \{1, \ldots, n\},\ t \in \mathcal{T}\}$ must satisfy the following axioms:

$$\text{If } E_j(t_1, t_2), \text{ then } x_i(t_1) = x_i(t_2) \text{ when } i \neq j, \quad (2)$$

$$\text{and } x_j(t_1)x_j(t_2) = M_j(t_1) + M_j(t_2). \quad (3)$$

$$\text{For } t_1 \in \mathcal{T},\ x_j \not| M_j(t_1). \quad (4)$$

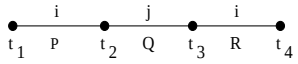
$$\text{If } E_i(t_1, t_2) \text{ and } x_i | M_j(t_1) \text{ then } x_i \not| M_j(t_2). \quad (5)$$

$$\text{If } E_i(t_1, t_2) \text{ and } E_j(t_2, t_3) \text{ then } x_j | M_i(t_1) \text{ if and only if } x_i | M_j(t_2). \quad (6)$$

$$\text{Suppose } E_i(t_1, t_2),\ E_j(t_2, t_3) \text{ and } E_i(t_3, t_4). \text{ Then } \frac{M_i(t_3)}{M_i(t_4)} = \left(\frac{M_i(t_2)}{M_i(t_1)}\right)\Big|_{x_j \leftarrow M_0/x_j} \quad (7)$$

$$\text{where } M_0 = (M_j(t_2) + M_j(t_3))|_{x_i = 0}.$$

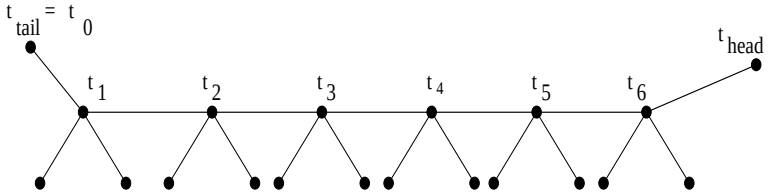
Axiom (7) is the most significant axiom. Axiom (7) will uniquely determine how to propagate the binomial exchanges. Letting $P = M_i(t_1) + M_i(t_2)$, $Q = M_j(t_2) + M_j(t_3)$ and $R = M_i(t_3) + M_i(t_4)$, axiom (7) implies that whenever



appears in the exchange graph or tree, then the exchange binomials P, Q and R satisfy the condition that there exists a Laurent monomial L and nonnegative integer b such that $L \cdot Q_0^b \cdot P = R|_{x_j \leftarrow \frac{Q_0}{x_j}}$ where $Q_0 = Q|_{x_i \leftarrow 0}$ [8, pgs. 8-9].

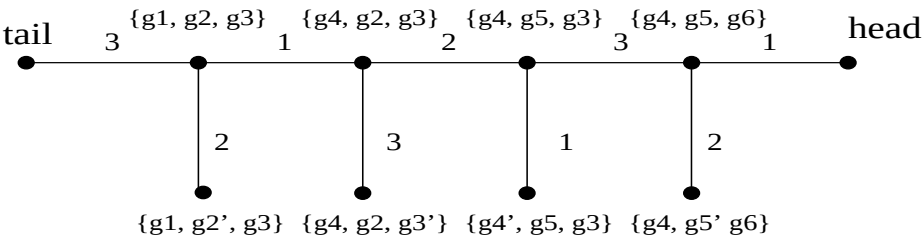
A Laurent monomial is a fraction consisting of a monomial over another monomial and the notation $P|_{x_i \leftarrow a}$ signifies the evaluation of polynomial P where a has been substituted for the variable x_i.

Exchange trees have another graph structure embedded in them: graphs $\mathbb{T}_{m,n}$ which Fomin and Zelevinsky call caterpillars. A caterpillar $\mathbb{T}_{m,n}$ for $m \geq 2$ is defined as a tree with a spine of m vertices of degree n and $m(n-2)+2$ vertices of degree 1. Of the degree-1 vertices, $m(n-2)$ of them will be referred to as feet and the remaining two (which must emanate from the extremities of the spine) will be called the head and the tail.



A caterpillar $\mathbb{T}_{6,4}$.

The clusters on the spine could represent a recursive sequence that we would like to propagate. For example, consider the sequence $g_n = \frac{g_{n-1}g_{n-2}+1}{g_{n-3}}$ Then the associated caterpillar $\mathbb{T}_{4,3}$ would look like



where $g_4g_1 = g_2g_3 + 1$, $g_5g_2 = g_3g_4 + 1$ and $g_6g_3 = g_4g_5 + 1$ are the exchange relations corresponding to the edges of the spine. The legs have different relations and our goal is to show that there are ways to define binomial exchanges

corresponding to the leg edges that keep the caterpillar consistent with axioms (2-7) thereby making it part of the exchange graph for a cluster algebra.

To uphold these axioms in the above caterpillar, the polynomial relations associated with the legs will be $g_2'g_2 = g_1 + g_3, g_3'g_3 = g_2 + g_4,\ \ g_4'g_4 = g_3 + g_5$, and $g_5'g_5 = g_4 + g_6$.

Notice now that if we start with the cluster $\{g_1, g_2, g_3\}$ and then travel along edge 1, then edge 2 and edge 1, we get to

$\{\frac{g_2g_3+1}{g_1}, g_2, g_3\} \to \{\frac{g_2g_3+1}{g_1}, \frac{(g_2g_3+1)g_3+g_1}{g_1g_2}, g_3\}$

$\to \{\frac{g_1}{g_2g_3+1}\frac{(g_2g_3+1)g_3+g_1+g_1g_2g_3}{g_1g_2}, \frac{(g_2g_3+1)g_3+g_1}{g_1g_2}, g_3\} = \{\frac{g_1+g_3}{g_2}, \frac{(g_2g_3+1)g_3+g_1}{g_1g_2}, g_3\}$

$= \{g_4', g_5, g_3\}$

We will study this example in more depth later. As of now, it is noteable that a priori one might expect g_4' to be more complicated then $g_4 = \frac{g_2g_3+1}{g_1}$ just as g_5 is more complicated than g_2 but it is in fact still a Laurent polynomial. It is just as simple if not simpler than g_4. Furthermore, even though g_5 is more complicated, it also is a Laurent polynomial in the variables x_1, x_2 and x_3. One could construct similar caterpillars with larger spines, $\mathbb{T}_{m,3}$ for arbitrarily large m, and allow the exchange binomial $xy + 1$ to be associated to all of the edges of the spine. We thereby would extend the sequence of g_n and g_n'. It is natural to ask: *Will all of the g_n and g_n' turn out to be Laurent polynomials in terms of the initial variables?* One can answer this in the affirmative and we will find this is the corollary of a more general result.

2.3 The Caterpillar Lemma

The following results and proofs come from [7] and [8]. The motivation for the Caterpillar Lemma is the following observation by Fomin and Zelevinsky.

> One of the main structural features of cluster algebras established in the present paper is the following *Laurent phenomenon*: any cluster variable x viewed as a rational function in the variables of any given cluster is in fact a Laurent polynomial. This property is quite surprising: in most cases, the numerators of these Laurent polynomials contain a huge number of monomials, and the numerators for x moves into the denominator when we compute the cluster variable x' obtained from x by an exchange (1). The magic of the Laurent phenomenon is that, at every stage of the recursive process, a cancellation will inevitably occur, leaving a single monomial in the denominator [7, pg. 3].

Theorem 1 *In a cluster algebra, any cluster variable is expressed in terms of any given cluster as a Laurent polynomial with coefficients in $\mathbb{Z}$.* [3]

[3] The statement of this theorem differs from Fomin and Zelevinsky's formulation in the fact that Fomin and Zelevinsky allow the cluster variables to be written in terms of coefficients from the group ring $\mathbb{Z}\mathbb{P}$ but since we previously set $\mathbb{P} = \{1\}$ we only allow for integer coefficients.

Remark 1 Fomin and Zelevinsky conjecture that all of the cluster variables can be expressed using *nonnegative* integer coefficients.

To prove this theorem, we will prove a generalization from [8]. First, we will need to generalize our definition of exchange pattern.

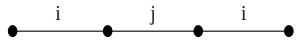
Definition 2 Let $\mathbb{A}$ be a unique factorization domain, and assume that a nonzero polynomial $P \in \mathbb{A}[x_1, \ldots, x_n]$ that does not depend on x_k is associated with every edge such that $E_k(t, t')$ in the exchange tree $\mathbb{T}$. This will be called a *generalized exchange pattern.*

These generalized exchange patterns are analogous to the exchange patterns relying on binomials and $x_k(t)x_k(t') = P(x(t))$.

We will label the vertices on the spine of a caterpillar, $\mathbb{T}_{m,n}$, t_1 through t_m and label the tail t_{tail} as t_0.

Lemma 2 *(Caterpillar Lemma) Assume that that a generalized exchange pattern on $\mathbb{T}_{m,n}$ satisfies the following conditions:*

- *For any edge labeled k, the associated exchange polynomial P does not depend on x_k, and is not divisible by any $x_i \in \{x_1, \ldots, x_n\}$.*
- *If two consecutive edges have P and Q associated to them,* 

then the polynomials P and $Q_0 = Q|_{x_i \leftarrow 0}$ are coprime elements of $\mathbb{A}[x_1, \ldots, x_n]$.
- *If three consecutive edges have P, Q and R associated to them,* 

then there exists a nonnegative integer b and Laurent monomial L coprime with P with coefficients in $\mathbb{A}$ such that $L \cdot Q_0^b \cdot P = R|_{x_j \leftarrow \frac{Q_0}{x_j}}$ where $Q_0 = Q|_{x_i \leftarrow 0}$.

If those conditions are satisfied, then for every $i \in \{1, \ldots, n\}$, $t \in \mathbb{T}_{m,n}$, $x_i(t)$ is a Laurent polynomial in $X(t_0) = \{x_1(t_0), \ldots, x_n(t_0)\}$ with coefficients in $\mathbb{A}$.

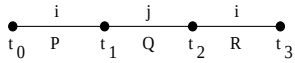
Remark 2 As mentioned previously, this third axiom resembles axiom (7) except now P, Q and R are allowed to be polynomials rather than just binomials.

Proof. For every $t \in \mathbb{T}_{m,n}$, let

$$\mathcal{L}(t) = \mathbb{A}[x_1(t)^{\pm 1}, \ldots, x_n(t)^{\pm 1}]$$

be the Laurent polynomial ring of the cluster $X(t)$ with coefficients in $\mathbb{A}$. We will treat $\mathcal{L}(t)$ as a subring of the field of rational functions of $\mathbb{A}(X(t_0))$.

It suffices to prove that every cluster $X(t) \in \mathcal{L}(t_0) = \mathcal{L}_0$. Since $\mathcal{L}_0$ is a unique factorization domain, elements have a gcd defined up to units of $\mathbb{A}$. We will prove all $X(t) \in \mathcal{L}_0$ by induction on the size of the spine, m. The case $m = 1$ is trivial so we can assume there exists an M such that for all $m \leq M$, the caterpillar lemma is true. Now assume $m \geq 2$. We will prove that $X(t_{head}) \in \mathcal{L}_0$ and be done since t_{head} will be the vertex of the caterpillar furthest from t_0. We will assume $E_i(t_0, t_1)$ and $E_j(t_1, t_2)$. Letting $t_3 \in \mathbb{T}_{m,n}$ be the vertex so that $E_i(t_2, t_3)$, we have



$X(t_1) \cup X(t_2) \cup X(t_3) = X(t_0) \cup \{x_i(t_1), x_j(t_2), x_i(t_3)\}$ and similar to Speyer's proof,

$$x_i(t_1) = \frac{P(x_j(t_0))}{x_i(t_0)}$$

and

$$x_j(t_2) = \frac{Q\left(\frac{P(x_j(t_0))}{x_i(t_0)}\right)}{x_j(t_0)}$$

are clearly in $\mathcal{L}_0$. We now must show that all of the clusters $X(t_1), X(t_2)$ and $X(t_3)$ are contained in $\mathcal{L}_0$ as subsets. This it suffices to prove:

$$\begin{aligned} x_i(t_3) &\in \mathcal{L}_0, & (8)\\ \gcd\big(x_i(t_1), x_j(t_2)\big) &= 1, & (9)\\ \gcd\big(x_i(t_1), x_i(t_3)\big) &= 1. & (10) \end{aligned}$$

By the third axiom stated in the lemma (previously axiom 7), $R\left(\frac{Q(0)}{x_j(t_0)}\right) = L\big(x_j(t_0)\big)Q(0)^b P\big(x_j(t_0)\big)$ where $L\big(x_j(t_0)\big) = L|_{x_j \leftarrow x_j(t_0)}$.

$$x_i(t_3) = \frac{R\left(\frac{Q(x_i(t_1))}{x_j(t_0)}\right)}{x_i(t_1)} = \frac{R\left(\frac{Q(x_i(t_1))}{x_j(t_0)}\right) - R\left(\frac{Q(0)}{x_j(t_0)}\right)}{x_i(t_1)} + \frac{R\left(\frac{Q(0)}{x_j(t_0)}\right)}{x_i(t_1)}.$$

The polynomial $Q(x_i(t_1))$ minus its constant term $Q(0)$ is divisible by $x_i(t_1)$ and extending this property we obtain

$$\frac{R\left(\frac{Q(x_i(t_1))}{x_j(t_0)}\right) - R\left(\frac{Q(0)}{x_j(t_0)}\right)}{x_i(t_1)} \in \mathcal{L}_0 \quad \text{and}$$

$$\frac{R\left(\frac{Q(0)}{x_j(t_0)}\right)}{x_i(t_1)} = \frac{L\big(x_j(t_0)\big)Q(0)^b P\big(x_j(t_0)\big)}{x_i(t_1)} = L\big(x_j(t_0)\big)Q(0)^b x_i(t_0) \in \mathcal{L}_0,$$

thus (8) is true. $x_j(t_2) = \frac{Q\left(x_i(t_1)\right)}{x_j(t_0)} \equiv \frac{Q(0)}{x_j(t_0)} \mod x_j(t_0)$ and $x_i(t_0), x_j(t_0)$ are invertible in $\mathcal{L}_0$ so $\gcd\big(x_i(t_1), x_j(t_2)\big) = \gcd\big(P(x_j(t_0)), Q(0)\big) = 1$ by the the second axiom of generalized exchange patterns. Thus (9) is proved.

To prove (10), we use the fact that

$$x_i(t_3) = \frac{R\left(\frac{Q(x_i(t_1))}{x_j(t_0)}\right) - R\left(\frac{Q(0)}{x_j(t_0)}\right)}{x_i(t_1)} + L\big(x_j(t_0)\big)Q(0)^b x_i(t_0)$$

and taking the limit $x_i(t_1) \to 0$, and applying calculus, we arrive at the equality

$$x_i(t_3) \equiv R'\left(\frac{Q(0)}{x_j(t_0)}\right) \cdot \frac{Q'(0)}{x_j(t_0)} + L(x_j(t_0)Q(0)^b x_i(t_0) \mod x_i(t_1).$$

Since $\gcd\big(L(x_j(t_0)Q(0)^b, P(x_j(t_0))\big) = 1$ thus $\gcd\big(x_i(t_1), x_i(t_3)\big) = 1$.

By the inductive step, the subset $X(t_{head})$ is contained in both $\mathcal{L}(t_1)$ and $\mathcal{L}(t_3)$ since the length of the spine between t_{head} and t_1 or t_3 is less than the distance to t_0. Thus for any $x \in X(t_{head})$, $x = \frac{f_1}{x_i(t_1)^a} = \frac{f_3}{x_j(t_2)^b x_i(t_3)^c}$ for some $f_1, f_3 \in \mathcal{L}_0$ and nonnegative integers a, b, c. By (10), the denominators are relatively prime, hence $x \in \mathcal{L}_0$. $\square$

This Lemma is a generalization of Theorem 1 for the following reasons, as explained in [7].

- $\mathbb{T}_{m,n}$ can be embedded in $\mathbb{T}_n$.
- We are allowing polynomials with appropriate restrictions instead of binomials.
- We are allowing any unique factorization domain $\mathbb{A}$ instead of just $\mathbb{Z}$.

In the special case of $k = 2$ and $\mathbb{A} = \mathbb{Q}$, we get the condition $L_1 \cdot P\left(\frac{P(0)}{t_2}\right) \equiv P(t_2) \mod t_3,\ L_1 \in \mathbb{Q}[t_2^{\pm 1}]$, which exactly matches Speyer's result. However, unlike David Speyer's proof, the converse of the caterpillar lemma does not hold.

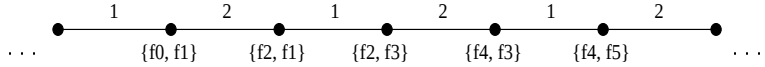
2.4 Sample Proofs for Laurentness of Sequences

The following proofs are based on proofs given in [8]. More details have been included below. Consider the sequence $f_n f_{n-2} = f_{n-1}^2 + 1$ from section 2. We can show that for all $n \geq 1$, f_n is a Laurent polynomial where only a monomial of the form $f_0^a f_1^b$ appears in the denominator.

Proof. We can create the following sequence of clusters

$$\{f_0, f_1\}, \{f_2, f_1\}, \{f_2, f_3\}, \{f_4, f_3\}, \ldots$$

and make an exchange tree of rank 2 using the clusters as vertices and label edges with an alternating pattern of 1 and 2. We will use the exchange binomial $P(t) = t^2 + 1$ for all edges. Thus $f_2 f_0 = P(f_1), f_3 f_1 = P(f_2)$, etc.



By the caterpillar lemma, it suffices to show that such a choice of an exchange pattern is consistent with the axioms of a cluster algebra. We have an alternating pattern $\bullet\!\!-\!\!\overset{1}{-}\!\!\bullet\!\!-\!\!\overset{2}{-}\!\!\bullet\!\!-\!\!\overset{1}{-}\!\!\bullet$ thus axiom (7) requires there exists a Laurent monomial $L = c \cdot t^d$ s.t.

$$P(t) = L \cdot P\left(\frac{P(0)}{t}\right)$$

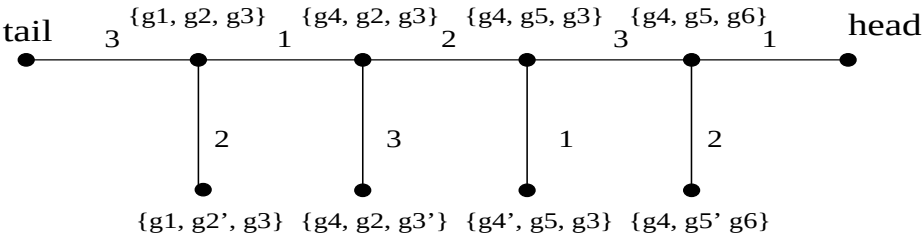
In fact, letting $L = t^2$,

$$L \cdot \left(\frac{0^2 + 1}{t}\right)^2 + 1 = 1 + t^2 = P(t).$$

We can similarly prove that $\{g_n\}$ from section 2 is a Laurent sequence. We build the clusters

$$\{g_0, g_1, g_2\}, \{g_3, g_1, g_2\}, \{g_3, g_4, g_2\}, \{g_3, g_4, g_5\}, \ldots$$

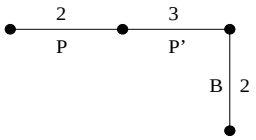
and the caterpillar



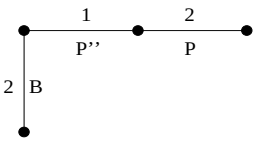
where each edge on the spine will have the exchange binomial $P(x, y) = xy + 1$. However, now it is important to specify the domain variables: letting n be an arbitrary integer with the property $n \equiv 0 \mod 3$ an edge labeled 1 will use g_{n+2} and g_n as domain variables, an edge labeled 2 will use g_{n+1} and g_n, and an edge labeled 3 will use g_{n+1} and g_{n+2}.

From each vertex on the spine, there will be three edges stemming from it, but only two of those edges will be contained in the spine. To show $\{g_n\}$ is Laurent, it suffices to show that the remaining edges, the *legs* can be chosen so that the axioms of a cluster algebra are satisfied.

Specifically, we must be able to choose a binomial B such that



with $P = P(g_{n+3}, g_{n+1})$, $P' = P(g_{n+1}, g_{n+2})$ satisfies axiom (7) for some Laurent monomial $L = c \cdot x^a y^b$ and



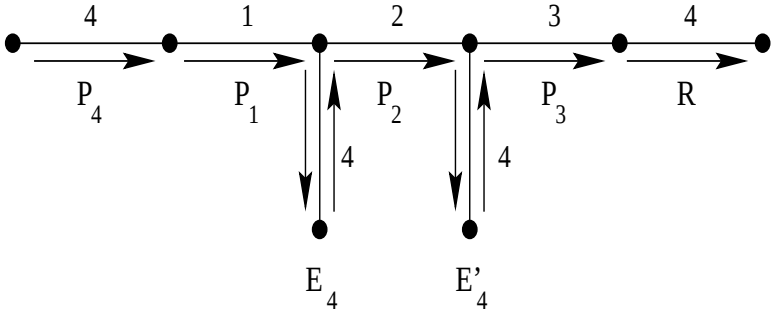
with $P'' = P(g_{n+2}, g_{n+3})$ satisfies axiom (7) for some Laurent monomial $L' = c' \cdot x^{a'} y^{b'}$.

The first relation uniquely determines B. In this particular example $B = x + y$ and plugging this B into the second relation we find that the second relation is indeed satisfied. Thus we conclude $\{g_n\}$ is a Laurent sequence.

Assigning $f_0 = f_1 = 1$ or $g_0 = g_1 = g_2 = 1$ we can conclude that the sequences f_n and g_n are integer sequences. For these two examples we already knew this since we have a combinatorial interpretation of f_n or g_n. However, the beauty of this Cluster Algebra method is that it can prove Laurentness even when there is no combinatorial interpretation of a sequence. The next proofs will demonstrate Laurentness for Somos-4, Somos-5, Somos-6, and Somos-7.

2.4.1 Proof of Laurentness for Several Somos Sequences

For the Somos-4 sequence s_n we create a spine of clusters of size 4 such that each cluster only contains a window $s_{n+1}, s_{n+2}, s_{n+3}, s_{n+4}$ for some n. We build the caterpillar starting arbitrarily at a vertex between edges labeled 4 and 1 on the spine, and create the associated leg with edge label 4. We then continue back onto the spine and find the exchange polynomial associated with the next edge labeled 4. This edge will also be a leg of the caterpillar. Finally, by using the spine edge labeled 3, we conclude that this series of exchanges is consistent with the axioms of the caterpillar lemma. Thus Somos-4 is a Laurent sequence, a sequence of Laurent polynomials in the first four terms.



A caterpillar for the Somos-4 sequence.

$$\begin{array}{rcll} P_4 & = & x_1x_3 + x_2^2 & P_1 = x_2x_4 + x_3^2 \\ E_4 & = & x_3^3 + x_2^2x_1 & P_2 = x_3x_1 + x_4^2 \\ E_4' & = & x_3x_2^2 + x_1^3 & P_3 = x_4x_2 + x_1^2 \\ R & = & x_1x_3 + x_2^2 & \square \end{array}$$

The summary of the calculations for Somos-5, Somos-6, and Somos-7 are below. They are also calculated constructing associated caterpillars according to axiom (7). Since P_5, P_6 and P_7 respectively equal R for each calculation, these sequences are also Laurent sequences. The program **Maple** was used for the calculations.

$$
\begin{array}{rcll}
P_5 &=& x_1x_4 + x_2x_3 & P_1 = x_2x_5 + x_3x_4 \\
E_5 &=& x_4^2 + x_2x_1 & P_2 = x_3x_1 + x_4x_5 \\
E_5' &=& x_4^2x_2 + x_1^2x_3 & P_3 = x_4x_2 + x_5x_1 \\
E_5'' &=& x_3x_4 + x_1^2 & P_4 = x_5x_3 + x_1x_2 \\
R &=& x_1x_4 + x_2x_3 & \square
\end{array}
$$

$$
\begin{array}{rcl}
P_6 &=& x_1x_5 + x_2x_4 + x_3^2 \\
P_1 &=& x_2x_6 + x_3x_5 + x_4^2 \\
E_6 &=& x_5^2x_3 + x_5x_4^2 + x_4x_2x_1 + x_3^2x_1 \\
P_2 &=& x_3x_1 + x_4x_6 + x_5^2 \\
E_6' &=& x_5^2x_3x_2 + x_5x_4^2x_2 + x_1^2x_4x_3 + x_1x_4x_5^2 + x_3^2x_1x_2 \\
P_3 &=& x_4x_2 + x_5x_1 + x_6^2 \\
E_6'' &=& x_4x_1x_2^2 + x_4x_5x_3^2 + x_1^2x_4x_3 + x_2x_1^2x_5 + x_5^2x_3x_2 \\
P_4 &=& x_5x_3 + x_6x_2 + x_1^2 \\
E_6''' &=& x_5x_3^2 + x_1^2x_3 + x_4x_2x_5 + x_2^2x_1 \\
P_5 &=& x_6x_4 + x_1x_3 + x_2^2 \\
R &=& x_1x_5 + x_2x_4 + x_3^2 \quad \square
\end{array}
$$

$$
\begin{array}{rcl}
P_7 &=& x_1x_6 + x_2x_5 + x_3x_4 \\
P_1 &=& x_2x_7 + x_3x_6 + x_4x_5 \\
E_7 &=& x_3x_6^2 + x_6x_5x_4 + x_5x_2x_1 + x_3x_4x_1 \\
P_2 &=& x_3x_1 + x_4x_7 + x_5x_6 \\
E_7' &=& x_3x_6^2x_2 + x_6x_5x_4x_2 + x_5^2x_1x_6 + x_1^2x_3x_5 + x_4x_2x_1x_3 \\
P_3 &=& x_4x_2 + x_5x_1 + x_6x_7 \\
E_7'' &=& x_4x_2x_1 + x_6^2x_2 + x_6x_5x_3 + x_1^2x_5 \\
P_4 &=& x_5x_3 + x_6x_2 + x_7x_1 \\
E_7''' &=& x_2x_1x_5x_3 + x_2^2x_1x_6 + x_6^2x_2x_4 + x_6x_5x_3x_4 + x_1^2x_5x_4 \\
P_5 &=& x_6x_4 + x_7x_3 + x_1x_2 \\
E_7'''' &=& x_6x_3x_4 + x_1^2x_4 + x_6x_5x_2 + x_2x_1x_3 \\
P_6 &=& x_7x_5 + x_1x_4 + x_2x_3 \\
R &=& x_1x_6 + x_2x_5 + x_3x_4 \quad \square
\end{array}
$$

Notice that the axioms of cluster algebras require all exchange polynomials to be binomials, and accordingly, the exchange polynomials associated with the legs for Somos-4 and Somos-5 are binomials. On the other hand, for Somos-6 and

Somos-7, the exchange polynomials for the edges of the spine are not binomials (they are trinomials) and the number of terms in the exchange polynomials of the legs is not bounded by three.

Unlike the previous Somos sequences, Somos-8, $\tilde{S}_n$, is not a Laurent sequence. If one attempts to use the caterpillar lemma, one finds that R is such a large polynomial that it would be cumbersome to include it in this exposition. Thus $R \neq P_8$. This alone does not suffice to show that Somos-8 is not Laurent. However using the initial conditions $\tilde{S}_1 = \cdots = \tilde{S}_8 = 1$ and applying the recurrence $\tilde{S}_n\tilde{S}_{n+8} = \tilde{S}_{n+1}\tilde{S}_{n+7} + \tilde{S}_{n+2}\tilde{S}_{n+6} + \tilde{S}_{n+3}\tilde{S}_{n+5} + \tilde{S}_{n+4}^2$, one finds the first 18 terms are $1, 1, 1, 1, 1, 1, 1, 1, 4, 7, 13, 25, 61, 187, 775, 5827, 14815, 420514/7$. Thus Somos-8 is not a Laurent sequence.

3 Exchange Graphs

Another beautiful application of cluster algebra theory is the construction of exchange graphs. In the initial definition of cluster algebras, we are given a collection of clusters and exchange relations between the variables in the form of exchange binomials. We saw that one could construct an n-regular tree $\mathbb{T}_n$ where the clusters are the vertices. In practice this graph $\mathbb{T}_n$ need not be a tree, it could have cycles or it could even be finite. Any exchange graph must be n-regular [7, pg. 27]. Before delving into the theory of exchange graphs, we will discuss some background material concerning semisimple Lie algebras and root systems.

Lie algebras and root systems are significant to the theory of cluster algebras because these structures appear to help classify cluster algebras. One can classify a special class of Lie algebras, known as semisimple Lie algebras, according to their associated root systems and reflection groups. As we will see later on, the machinery of root systems will allow us to classify the semisimple Lie algebras according to Cartan matrices. This famous classification is known as the Cartan-Killing classification. We will see that to each Cartan matrix, we can associate an exchange matrix, and lastly each exchange matrix will uniquely determine a cluster algebra. We will explicitly use the theory of semisimple Lie algebras and root systems to classify the exchange graphs of some low-rank cluster algebras of *finite type*. By finite type, I refer to a cluster algebra whose exchange graph has a finite number of vertices.

3.1 Lie Algebras

The following background material is from Fulton and Harris' text, *Representation Theory* [9].

Definition 3 A Lie group is a group which is also a $\mathbb{C}_\infty$ smooth manifold. In this group, the composition operator $\circ : G \times G \to G$ and the inverse operator $^{-1} : G \to G$ are both differentiable [9, pg. 93].

A fundamental example of a Lie group is the general linear group $GL_n\mathbb{R}$, the group of invertible $n \times n$ real matrices [9, pg. 95].

Definition 4 A Lie algebra, $\mathfrak{g}$, is a vector space and a accompanying skew-symmetric bilinear map $[\ ,\ ] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ satisfying *Jacobi's identity* [9, pg. 108]

$$[X,[Y,Z]] + [Y,[Z,X]] + [Z,[X,Y]] = 0.$$

Definition 5 The subspace of $\mathfrak{g}$ such that $Z(\mathfrak{g}) = \{X \in \mathfrak{g} : [X,Y] = 0]$ for all $Y \in \mathfrak{g}\}$ is defined to be the *center* of $\mathfrak{g}$ [9, pg. 121].

Remark 3 A Lie Algebra and Lie Group can be associated to each other via the exp and derivative maps [9, pgs. 104-120], however this explicit correspondence is not needed for the rest of this exposition.

3.1.1 The Classification of Semisimple Lie Algebras

Semisimple Lie algebras are those Lie algebras that are reducible as direct products of simple Lie algebras. There are four infinite families of simple Lie algebras, $A_n(n \geq 1), B_n(n \geq 2), C_n(n \geq 2)$, and $D_n(n \geq 3)$. There are also several exceptional Lie Algebras, E_6, E_7, E_8, F_4, and G_2. (Page 326 Fulton and Harris) The families have nice representations as fundamental matrix algebras, which correspond to important matrix Lie groups.

$$\begin{aligned}
(A_n) &\leftrightarrow \mathfrak{sl}_{n+1}\mathbb{C} \\
(B_n) &\leftrightarrow \mathfrak{so}_{2n+1}\mathbb{C} \\
(C_n) &\leftrightarrow \mathfrak{sp}_{2n}\mathbb{C} \\
(D_n) &\leftrightarrow \mathfrak{so}_{2n}\mathbb{C}
\end{aligned}$$

It turns out these Lie algebras can be better understood using root systems. Before defining root systems, we first need to discuss reflection groups. That background material comes from Humphreys' text *Reflection Groups and Coxeter Groups* [12, pgs. 5-11, 39].

3.2 Reflection Groups and Root Systems

Given a real Euclidean Vector space V with a positive definite symmetric bilinear form $\langle\ ,\ \rangle$, a *reflection* is a linear map s_α on V that sends a nonzero vector α to its negative while fixing the hyperplane H_α orthogonal to α. We may write a reflection as the formula

$$s_\alpha(\beta) = \beta - \frac{2\langle\alpha,\beta\rangle}{\langle\alpha,\alpha\rangle}\alpha.$$

A finite group generated by such maps is a *finite reflection group*. Such a group is in fact a subgroup of the orthogonal group $\mathbf{O}(V)$ since reflections preserve the length of elements. It is customary to denote a reflection group by W.

We define a *root system* Φ as a set of vectors that satisfy the conditions

$$\Phi \cap \mathbb{R}\alpha = \{\alpha, -\alpha\},\ \forall \alpha \in \Phi,$$

$$s_\alpha(\Phi) = \Phi,\ \forall \alpha \in \Phi.$$

The reflection group W associated to a root system is the group generated by the s_α for $\alpha \in \Phi$. Consequently the set of vectors Φ are fixed under the action of W.

A root system Φ is called *crystallographic* if $2\langle\alpha,\beta\rangle/\langle\beta,\beta\rangle \in \mathbb{Z}$ for all $\alpha, \beta \in \Phi$. We will actually need such a condition for the group generated by the reflections $s_\alpha(\alpha \in \Phi)$ to be a Weyl group so that the root system will be associated to a semisimple Lie algebra.

So far, it is unclear which root systems are more natural than others. To define specific types of root systems that are more canonical, we need a total ordering on the vectors of V. A total ordering on a real vector space V is a transitive relation $<$ such that the following additional conditions hold:

$$\text{for every distinct pair } a \neq b \in V, \text{ either } a < b \text{ or } b < a \text{ but not both;}$$

$$a < b \Rightarrow a + c < b + c;$$

$$a < b \in \mathbb{R}, c \in \mathbb{R} - 0 \Rightarrow ca < cb \text{ if } c > 0 \text{ and } cb < ca \text{ if } c < 0.$$

One can construct a total ordering on V in many ways, the easiest example is lexicographical ordering: suppose $v_1, \ldots v_n$ is a basis for V then $a_1v_1 + \cdots a_nv_n < b_1v_1 + \cdots b_nv_n$ if and only if $a_1 = b_1, a_2 = b_2, \ldots, a_k = b_k$, and $a_{k+1} < b_{k+1}$ where k can be zero and $a_i, b_i \in \mathbb{R}$ for all $i \in \{1, \ldots, n\}$.

We can call a vector λ *positive* if it is larger than the zero vector under the chosen total ordering of V. A *positive system* is a subset Π of a root system Φ where all of the constituent vectors are positive. A reflection of α is also a reflection of $-\alpha$ so one can also construct a *negative system* $-\Pi$ where all of the roots are negative. Φ is the disjoint union of Π and $-\Pi$.

Definition 6 A subset Δ of Φ is a *simple system* with *simple roots* as elements if Δ is a basis for the $\mathbb{R}$-span of Φ in V, and each $\alpha \in \Phi$ can be written as a $\mathbb{R}_{\geq 0}$- or $\mathbb{R}_{\leq 0}$-linear combination of elements of Δ.

Proposition 2 *For every root system Φ, there is a unique positive system Π that contains a unique simple system Δ*

For a proof, see [12, pgs. 8-9].

Definition 7 The *rank* of a reflection group W is the cardinality of the simple system contained in a root system associated with W. Note that even though the choice of the root system is not unique, the cardinality of Δ is invariant of the choice.

Simple root systems are so fundamental because W is actually generated by the reflections s_α for $\alpha \in \Delta$ for any simple system Δ. These reflections are called *simple* reflections.

Definition 8 A *reduced word* for $w \in W$ is a product of simple reflections equal to w so that the number of constituent reflections is minimal.

3.2.1 Simple Root Systems for Simple Lie Algebras

The following is from [12, pgs. 41-42] and outlines part of the classification of semisimple Lie algebras according to their root system.

$(A_n, n \geq 1)$ Let V be the hyperplane in $\mathbb{R}^{n+1}$ where all the coordinates add up to 0. Let Φ be the set of vectors $v \in V \cap \mathbb{Z}\epsilon_1 + \mathbb{Z}\epsilon_1 + \cdots + \mathbb{Z}\epsilon_{n+1}$ such that $|v| = \sqrt{2}$. Here $\mathbb{Z}\epsilon_1 + \mathbb{Z}\epsilon_1 + \cdots + \mathbb{Z}\epsilon_{n+1}$ is the standard unit integer lattice of $\mathbb{R}^{n+1}$. More explicitly, $\Phi = \{\epsilon_i - \epsilon_j : 1 \leq i \neq j \leq n+1\}$. The associated simple system is

$$\Delta = \{\epsilon_i - \epsilon_{i+1} : 1 \leq i \leq n\}.$$

The associated reflection group W is the symmetric group on $n+1$ letters, S_{n+1} that permutes the ϵ_i.

$(B_n, n \geq 2)$ Let $V = \mathbb{R}^n$, Φ be the set of vectors $v \in \mathbb{Z}\epsilon_1 + \mathbb{Z}\epsilon_1 + \cdots + \mathbb{Z}\epsilon_n$ such that $|v| = 1$ or $\sqrt{2}$. $\Phi = \{\pm\epsilon_i\} \cup \{\pm\epsilon_i \pm \epsilon_j : 1 \leq i \neq j \leq n\}$. The associated simple system is

$$\Delta = \{\epsilon_i - \epsilon_{i+1} : 1 \leq i \leq n-1\} \cup \{\epsilon_n\}.$$

The associated reflection group W is the semidirect product of S_n with $(\mathbb{Z}/2\mathbb{Z})^n$.

$(C_n, n \geq 2)$ is B_n's dual and its simple system is

$$\Delta = \{\epsilon_i - \epsilon_{i+1} : 1 \leq i \leq n-1\} \cup \{2\epsilon_n\}.$$

$(D_n, n \geq 4)$ Let $V = \mathbb{R}^n$, Φ be the set of vectors $v \in \mathbb{Z}\epsilon_1 + \mathbb{Z}\epsilon_1 + \cdots + \mathbb{Z}\epsilon_n$ such that $|v| = \sqrt{2}$. $\Phi = \{\pm\epsilon_i \pm \epsilon_j : 1 \leq i \neq j \leq n\}$. The associated simple system is

$$\Delta = \{\epsilon_i - \epsilon_{i+1} : 1 \leq i \leq n-1\} \cup \{\epsilon_{n-1} + \epsilon_n\}.$$

The associated reflection group W is the semidirect product of S_n with $(\mathbb{Z}/2\mathbb{Z})^{n-1}$.

(G_2) Let V be the hyperplane in $\mathbb{R}^3$ where the coordinates sum to 0. Let Φ be the set of vectors $v \in V \cap \mathbb{Z}\epsilon_1 + \mathbb{Z}\epsilon_1 + \cdots + \mathbb{Z}\epsilon_n$ such that $|v| = \sqrt{2}$ or $\sqrt{6}$. $\Phi = \{\pm(\epsilon_i - \epsilon_j) : 1 \leq i \neq j \leq 3\} \cup \{\pm(2\epsilon_i - \epsilon_j - \epsilon_k)\}$ where (i, j, k) is a permutation of $(1, 2, 3)$. The associated simple system is

$$\Delta = \{\epsilon_1 - \epsilon_2, -2\epsilon_1 + \epsilon_2 + \epsilon_3\}.^4$$

Letting the simple roots associated with a given simple Lie algebra as $\{\alpha_i\}$, to each of these simple Lie algebras we can associate a Cartan matrix [14, pg. 111].

[4] Since they are not relevant to our later discussion of cluster algebras, I will omit a description of the simple root systems for the other simple Lie algebras.

Definition 9 A *Cartan Matrix* is a matrix where the (i,j)th entry is $\langle \alpha_i, \alpha_j \rangle$.

One can show that $\langle \alpha, \beta \rangle = 2\frac{||\beta||}{||\alpha||}\cos\theta_{\alpha,\beta}$ where $\theta_{\alpha,\beta}$ is the angle between α and β. Thus all of the diagonal entries of a Cartan matrix will be 2 [14, pg. 114]. Furthermore, the off-diagonal entries will be less than or equal to zero [21, pg. 34]. Since semisimple Lie algebras are reducible, one can also associate a Cartan matrix to them. If a semisimple Lie algebra S is the direct product of $S_1 \times \cdots \times S_n$ where S_i is a simple Lie algebra with associated Cartan matrix M_i, then the Cartan matrix associated to S is the direct sum $M_1 \oplus \cdots \oplus M_n$.

3.3 Cluster Algebras and Root Systems

This section, as well as the next (3.4) comes from [7]. Root systems can be used to analyze Cluster Algebras. Specifically, let the exchange binomial associated to edge j between vertices t and t' be $M_j(t) + M_j(t')$. Then we let b_{ij} be the exponent of x_i in the expression $\frac{M_j(t)}{M_j(t')}$. It follows that

$$M_j(t) = \prod_{i:b_{ij}(t)>0} x_i^{b_{ij}(t)} \tag{11}$$

$$M_j(t') = \prod_{i:b_{ij}(t)<0} x_i^{-b_{ij}(t)}. \tag{12}$$

and $B(t) = (b_{ij}(t)$ will be a $n \times n$ integer matrix associated to vertex t. We will call such a matrix an *exchange matrix* associated to vertex t of cluster algebra $\mathcal{A}$.

In other words, $B(t)$ encodes the exponents of the exchange binomials for all of the edges stemming from vertex t. All of the exchange binomials have positive exponents but in an effort to differentiate between the binomials $x_1^2x_2 + x_3^3x_4$ and $x_1^2x_3^3 + x_2x_4$, the encoding of the exponents which appear in the second monomial are given a negative sign. Note, by axiom (5), x_i cannot divide both monomials. Axiom (6) implies that B is forced to be *sign-skew symmetric*, i.e. $b_{ij} = b_{ji} = 0$ or b_{ij} and b_{ji} have opposite signs.

Fomin and Zelevinsky also define a family of *matrix mutation* functions $\{\mu_i\}$, so that $\mu_k(B) = B' = (b'_{ij})$, where

$$b'_{ij} = -b_{ij} \quad \text{if } i = k \text{ or } j = k \tag{13}$$

$$= b_{ij} \quad \text{if } b_{ik}b_{kj} \leq 0 \tag{14}$$

$$= b_{ij} + b_{ik}b_{kj} \quad \text{if } b_{ik}, b_{kj} > 0 \tag{15}$$

$$= b_{ij} - b_{ik}b_{kj} \quad \text{if } b_{ik}, b_{kj} < 0. \tag{16}$$

Proposition 3 *A family of $n \times n$ integer matrices $(B(t))_{t\in\mathbb{T}_n}$ corresponds to an exchange pattern if and only if*

- *$B(t)$ is sign-skew-symmetric for all $t \in \mathbb{T}_n$.*
- *If there is an edge labeled k connecting vertices t and t', then $B(t') = \mu_k(B(t))$.*

Proof. First, let us assume that the family of matrices $(B(t))$ corresponds to an exchange pattern. Then by axiom (4), $b_{jj} = 0$ otherwise $x_j | M_j(t)$. Likewise, axiom (6) implies that $B(t)$ will be sign-skew symmetric.

The equality $b_{ik} = b'_{ik}$ stems from definitions (11) and (12) which is a formal way of saying that the ordering of the monomials in the exchange binomial depends on whether you are traveling from t to t' or t' to t.

If $j \neq k$, applying axiom (7) to the edge labeled k between vertices t and t' along with the two edges emanating from t and t' labeled with j, we see that

$$\prod_i x_i^{b'_{ij}} = \prod_i x_i^{b_{ij}} |_{x_k \leftarrow M/x_k} \tag{17}$$

for $M = \prod_{i:b_{ik}b_{jk}<0} x_i^{|b_{ik}|}$. Considering the exponents on the left-hand-side and right-hand-side of x_k in (17), we see that $b'_{kj} = -b_{kj}$. Comparing the exponents on the left-hand-side and right-hand-side of x_i in (17) for arbitrary i completes the proof.

Assuming that we have a family of sign-skew-symmetric matrices subject to matrix mutation as above, it is clear that the corresponding exchange binomials will obey the axioms of an exchange pattern, axioms (2-7). □

Once an exchange matrix B is defined for a given vertex (cluster) of the exchange graph, axiom (7) will uniquely define matrix mutation μ and all the exchange matrices associated to each vertex of the exchange graph.[5] To each of these exchange matrices we can associate a (generalized) Cartan matrix $A = A(B) = (a_{ij})$ of the same size where

$$\begin{array}{c} a_{ij} = 2 \text{ if } i = j \text{ and} \\ -|b_{ij}| \text{ if } i \neq j. \end{array} \tag{18}$$

These generalized Cartan matrices appear in the theory of Kac-Moody algebras. Fomin and Zelevinsky note that there seems to be a relation between cluster algebra with exchange matrix M and a Kac-Moody algebra with generalized Cartan matrix M' when M and M' are associated as in (18) [7, pgs. 15-16]. In general it is hard to prove that a given choice of B will force the whole family of $B(t)$ to be sign-skew-symmetric. However, the following condition implies that the whole family will in fact be sign-skew-symmetric. For more details on the proof, see [7, pg. 15].

Definition 10 A matrix B is called *skew-symmetrizable* if there exists a diagonal matrix D s.t. DB is skew-wymmetric.

There are other kinds of matrices that will work, but for the purposes of this exposition, we will restrict our attention to exchange matrices that are skew-symmetrizable. In fact, many of the following examples will only require the matrices to be skew-symmetric.

[5] The exchanges are uniquely defined since we have assumed all of the monomial coefficients are 1.

3.3.1 The Rank 2 Case

Let $\mathbb{T}_n$ be a 2-regular tree whose vertices are labeled t_m for $m \in \mathbb{Z}$ where an edge labeled $m \mod 2$ joins vertices t_m and t_{m+1}. Let the cluster associated with vertex t_m be $\{x_m, x_{m+1}\}$. By theorem 1, all of the succeeding cluster variables can be rewritten in terms of a Laurent polynomial of two initial cluster variables (x_1, x_2) after completing a series of exchanges according to the exchange binomials. Let

$$x_m = \frac{P_m(x_1, x_2)}{x_1^{d_1(m)} x_2^{d_2(m)}},$$

where P_m is a polynomial with coefficients in $\mathbb{Z}$ not divisible by x_1 or x_2 and $d_1, d_2 \in \mathbb{Z}$.

Corollary 1 *The only possible exchange patterns for a rank 2 cluster algebra correspond to a family of matrices with the form*

$$B(t_m) = (-1)^m \begin{bmatrix} 0 & b \\ -c & 0 \end{bmatrix}$$

for integers b and c of like sign.

Furthermore, these exchange matrices will correspond to an exchange pattern that alternates between the two binomials $x^b + 1$ and $1 + x^c$.

Proof. This is a corollary of Proposition 3 restricted to the rank 2 case.
The corresponding generalized Cartan matrix is

$$A(B(t)) = \begin{bmatrix} 2 & -b \\ -c & 2 \end{bmatrix}.$$

A root system with basis of simple roots $\{\alpha_1, \alpha_2\}$ corresponds to this Cartan matrix. Let $W(A)$ be the reflection group generated by the two simple roots

$$s_1 = \begin{bmatrix} -1 & b \\ 0 & 1 \end{bmatrix}, \; s_2 = \begin{bmatrix} 1 & 0 \\ c & -1 \end{bmatrix}$$

$s_1^2 = s_2^2 = 1$ so the possible reduced words $w \in W$ are

$$w_1(m) = s_1 s_2 s_1 \cdots s_{(m \mod 2)} \text{ or } w_2(m) = s_2 s_1 s_2 \cdots s_{(m+1 \mod 2)}.$$

After perusing our list of rank 2 semisimple Lie algebras and their corresponding root systems, we see that W finite $\Leftrightarrow bc \leq 3$. Further study reveals that the rank 2 cluster algebras of finite type can be summarized in the following table.

Lie Algebra	Cartan Matrix	Representative Exchange Matrix	# Clusters
$A_1 \times A_1$	$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	4
A_2	$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	5
B_2	$\begin{bmatrix} 2 & -1 \\ -2 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -2 & 0 \end{bmatrix}$	6
C_2	$\begin{bmatrix} 2 & -2 \\ -1 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 \\ -1 & 0 \end{bmatrix}$	6
G_2	$\begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 3 \\ -1 & 0 \end{bmatrix}$	8
$G_2^\vee$	$\begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -3 & 0 \end{bmatrix}$	8

3.4 The Formalism Behind Exchange Graphs

Fomin and Zelevinsky define two clusters t and t' to be $\mathcal{M}$-*equivalent* if there is a permutation $\sigma \in S_n$ such that $x_i(t') = x_{\sigma(i)}(t)$ for all $i \in \{1, \ldots . n\}$ and if $E_{\sigma(j)}(t, t_1)$ along with $E_j(t', t'_1)$ implies $M_j(t') = M_{\sigma(j)}(t)$ and $M_j(t'_1) = M_{\sigma(j)}(t_1)$. In other words, the two clusters are composed of a permutation of the same variables.

Let one follow a path on the tree $\mathbb{T}_n$ associated with a particular exchange pattern where the edges alternate $i, j, i, j, \ldots$. If $t \equiv_{\mathcal{M}} t'$ after a sequence of such steps, then we see that $\mathbb{T}_n$ has a cycle. By the analysis of the rank 2 case, the only cycles will be of length $4, 5, 6$ or 8. All other paths of alternating edges will be infinite.

The type A_2 case where the exchange graph is a pentagon is exceptional since the number of clusters is odd. Starting at cluster $\{x_1, y_1\}$, the polynomial exchanges $x_i x_{i+1} = y_i + 1$, $y_i y_{i+1} = x_{i+1} + 1$ leads to the clusters

$$\{x_1, y_1\} \to \{\frac{y_1 + 1}{x_1}, y_1\} \to \{\frac{y_1 + 1}{x_1}, \frac{x_1 + y_1 + 1}{x_1 y_1}\} \to \{\frac{x_1 + 1}{y_1}, \frac{x_1 + y_1 + 1}{x_1 y_1}\}$$

$$\to \{\frac{x_1 + 1}{y_1}, x_1\} \to \{y_1, x_1\}.$$

Each edge will not have a precise edge label since $x_6 = y_1$ and $y_6 = x_1$ implies that changing the 2nd variable of the cluster $\{x_6, y_6\}$ is equivalent to changing the 1st variable of $\{x_1, y_1\}$ even though these two clusters are $\mathcal{M}$-equivalent to each other.

Considering the example from section 2, $\{g_n\}$, one can append the caterpillar to get the full exchange graph for this cluster algebra. It turns out not to be an infinite tree, but instead two infinite rows of pentagons. We will call this graph $\mathcal{G}_{5,2}$. From earlier analysis, we know that the edges emanating from a vertex on the spine is associated with a cyclic transformation of the exchange polynomials $1 + x_2x_3$, $x_1 + x_3$, and $x_1x_2 + 1$. We encode these exchanges as the exchange matrix

$$B(t_0)) = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}.$$

Applying the matrix mutation functions $\{\mu_i\}$ to this starting matrix, we obtain

$$\mu_1(B(t_0)) = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \tag{19}$$

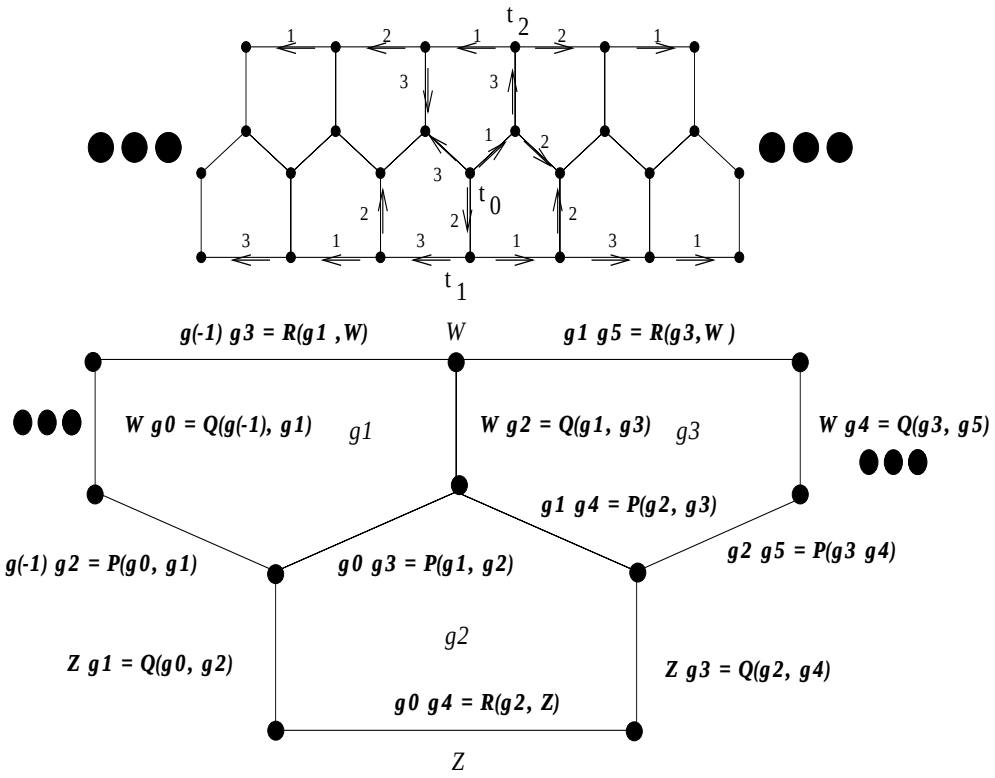
$$B(t_1) = \mu_2(B(t_0)) = \begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & -1 \\ -2 & 1 & 0 \end{bmatrix} \tag{20}$$

$$\mu_3(B(t_0)) = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \tag{21}$$

$$B(t_2) = \mu_3\mu_1(B(t_0)) = \begin{bmatrix} 0 & -2 & 1 \\ 2 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \tag{22}$$

Notice that $\mu_1(B(t_0))$ and $\mu_3(B(t_0))$ are just cyclic transformations of $B(t_0)$. This symmetry arises since the exchange polynomials associated with every vertex on the spine are cyclic transformations of each other. Edge 2, on the other hand, will lead one to a vertex off the spine, $B(t_1)$. One notices that $B(t_2)$, the result of traveling along edge 1 on the spine, followed by edge 3 leads one to another vertex off the spine, one whose exchange polynomials are cyclic transformations of $B(t_1)$'s exchange polynomials. Since $B(t_0)$'s three principal 2×2 submatrices are the exchange graph for a cluster algebra of type A_2, this implies that a walk along a sequence of adjoining edges alternatively labeled (either $1, 2, 1, 2, \ldots$ or $1, 3, 1, 3, \ldots$ or $2, 3, 2, 3, \ldots$) will be a cycle of length 5. Thus the associated exchange graph will have three adjoining pentagons at each vertex. Furthermore $B(t_1)$ and $B(t_2)$ contain exactly one principal 2×2 submatrix of the form $\begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix}$ thus the vertices t_1 and t_2 sit on infinite lines disjoint from the original spine. Based on the cyclic symmetry between $B(t_1)$ and $B(t_2)$, we are able to deduce that the pentagons must interlock in such a pattern to allow travel in both directions to be cyclically symmetric. Thus we find that the exchange graph consists of three spines, where the original spine associated

with g_n is the middle spine. We use axiom (7) and the exchange matrices to compute the exchange binomials associated with the additional edges.



Exchange graph $\mathcal{G}_{5,2}$ for the sequence g_n with a close-up.

Each vertex is bordered by three bounded (or unbounded) regions. The cluster variables of each vertex are represented by these three regions. $P(x,y) = xy+1$, $Q(x,y) = x+y$ and $R(x,y) = x^2+y$. The edge labels assume one is starting from vertex t_0 and traveling outward. Since subgraphs are pentagons, each edge does not have a precise edge label.

3.5 New Recurrences for Old Sequences

Notice that if we start with the recurrence $g_n g_{n-3} = g_{n-1}g_{n-2}+1$, and build the exchange graph which has the corresponding exchange binomials on it spine, we get an exchange graph with two additional spines which satisfy the recurrences $g_{2n+2}g_{2n-2} = g_{2n}^2 + z$ and $g_{2n+1}g_{2n-3} = g_{2n-1}^2 + w$ respectively. If $g_0 = g_1 = g_2 = 1$, then $z = 2$ and $w = 3$. Our combinatorial interpretation of g_n as the number of perfect matchings for a family of graphs had previously revealed the extra recurrence $g_{2n+2}g_{2n-2} = g_{2n}^2 + 2$.

Based on numerical evidence and these exchange graphs, the author conjectures that there is a one-to-many surjective map between perfect matchings in $2\times 2(n-1)$ grid graphs and perfect matchings in mutilated $3\times 2(n-1)$ grid graphs where a matching with m pairs of horizontal edges map to 3^m perfect matchings of a mutilated $3 \times 2(n-1)$ grid graph. Thus, the sequence g_n can be split into two alternating subsequences where the terms $g_{2n} = 1, 3, 11, 41, \ldots$ satisfy the recurrence $g_{2n}g_{2n-4} = g_{2n-2}^2 + 2$ and the terms $g_{2n+1} = 1, 2, 7, 26, 97, \ldots$ satisfy the recurrence $g_{2n+1}g_{2n-3} = g_{2n-1}^2 + 3$. The cluster algebra method has given an alternate way to uncover and prove these recurrences without presupposing knowledge of the combinatorial objects the integer sequence counts, and without explicit bijections. Thus the exchange graph method for discovering new recurrences provides a method for discovering new recurrences for a sequence even where the combinatorial interpretation is unknown.

3.6 Three-dimensional Exchange Graphs

We noticed for cluster algebras of rank 2 that the only possible exchange graphs are an infinite line, a square, a pentagon, a hexagon, or an octagon. Likewise, cluster algebras of rank 3 will either be of infinite type or finite type. The graph $\mathcal{G}_{5,2}$ is a nice example of an exchange graph for a cluster algebra of infinite type. A 3-degree tree is another possibility for a rank 3 cluster algebra of infinite type.

As illustrated in section 3.3, Fomin and Zelevinsky [7] illustrate that it appears possible to classify cluster algebras in terms of corresponding semisimple Lie algebras. However, it is unclear whether or not all cluster algebras of finite-type correspond to semisimple Lie algebras. If they do not correspond to semisimple Lie algebras, perhaps they correspond to Kac-Moody algebras [7, pg. 16]. For the case of rank 2 cluster algebras of finite type, the classification can be completed only using semisimple Lie algebras, as explained in section 3.3.1 and explained more thoroughly in [7]. In the following pages, some cluster algebras of higher ranks will be classified, though this exposition will only hint at some patterns since a complete classification is still an open problem. We will refer to a cluster algebra as type S if one of the clusters (vertices) has an exchange matrix associated to the Cartan matrix for the semisimple Lie algebra S. The cluster algebras of infinite type are hard to classify, but for rank 3 cluster algebras of finite type, some possible exchange graphs will correspond to the Lie algebras $A_1 \times A_1 \times A_1$, $A_2 \times A_1$, $B_2 \times A_1$, $G_2 \times A_1$, A_3 or B_3.

The one-dimensional exchange graph corresponding to A_1 is a line between two points ($\mathbb{Z}/2\mathbb{Z}$), the one for $A_1 \times A_1$ is a square, and $A_1 \times A_1 \times A_1$ has a cube as its exchange graph. Such evidence motivates the following result which seems not to have appeared in the literature before.

Proposition 4 *In general, the cluster algebra of type A_1^n has an n-cube as its exchange graph.*

Proof. The semisimple Lie algebra A_1^n has the diagonal matrix $2I_n$ as its Cartan matrix. Consequently, the associated exchange matrix is N, the matrix

of all zeros. So the cluster algebra of type A_1^n (which will be of rank n) contains at least one cluster $X = \{x_1, \ldots, x_n\}$ with the binomial exchange relations:

$$\begin{aligned} x_1 y_1 &= 1 \\ x_2 y_2 &= 1 \\ &\cdots \\ x_n y_n &= 1 \end{aligned}$$

which means all the adjacent clusters must look like

$$Y_i = \{x_1, \ldots, x_{i-1}, \frac{1}{x_i}, x_{i+1}, \ldots, x_n\}.$$

Furthermore, $\mu_i(N) = N$ for all i so each cluster of $\mathcal{A}$ will look be of the form

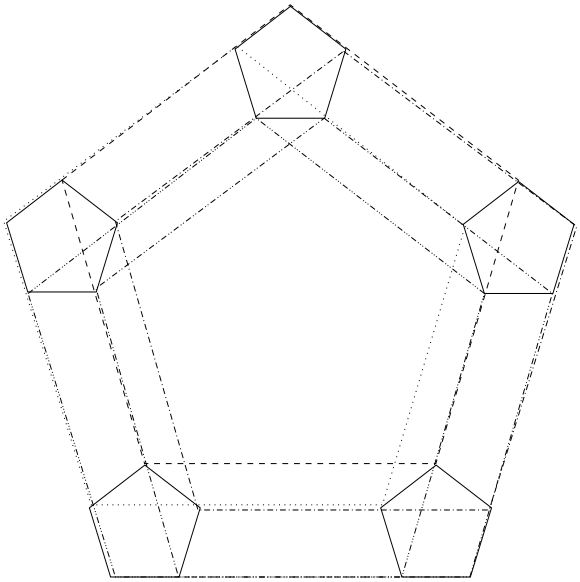
$$\{x_1^{\epsilon_1}, \ldots, x_n^{\epsilon_n}\}$$

where $\epsilon_i = \pm 1$, and each exchange changes the sign of exactly one ϵ_i. The corresponding exchange graph is an n-cube. $\square$

Proposition 5 *We can generalize this result. Let G_X be the exchange graph for a cluster algebra of type X. Then a cluster algebra of type $X \times A_1$ has $G_X \times \mathbb{Z}/2\mathbb{Z}$ as its exchange graph. This is a graph consisting of two copies of G_X where vertex $(v, 0)$ is connected to vertex $(v', 1)$ if and only if $v = v'$.*

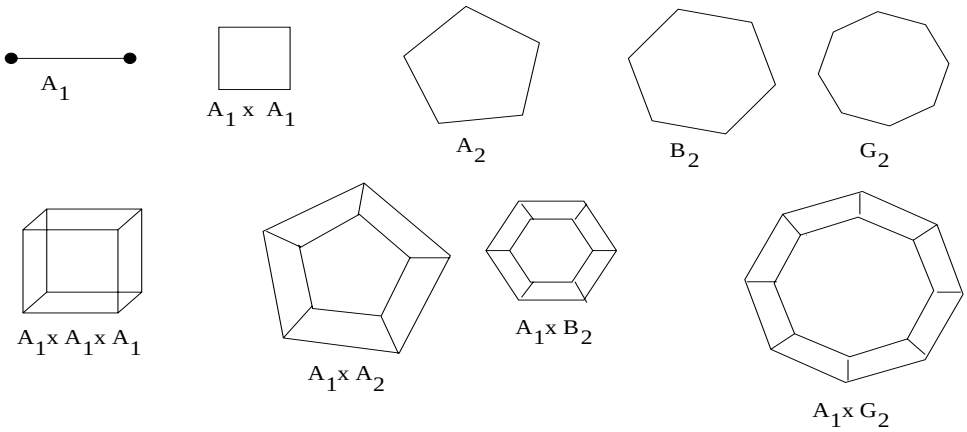
Proof. The proof is analogous. The exchange matrix associated with $X \times A_1$ will be M_X, the exchange matrix associated with X, with an extra row and column of zeros. This corresponds to adding an edge corresponding to the exchange $xy = 1$ to one of the vertices of G_X. Let $M_X \oplus 0$ be the corresponding exchange matrix for a cluster algebra of type $X \times A_1$. For $i \neq n$, $\mu_i(M_X \oplus 0) = \mu_i(M_X) \oplus 0$ and $\mu_n(M_X \oplus 0) = M_X \oplus 0$ which means we have added an nth variable to each cluster, and an edge at every vertex which sends x_n to its reciprocal. This addition will force the exchange graph to be $G_X \times \mathbb{Z}/2\mathbb{Z}$ where each vertex of G_X has been replaced by two vertices $\{(v, x_n), (v, \frac{1}{x_n})\}$ connected by an edge. $\square$

Similarly, we conjecture that the rank 4 cluster algebra $A_2 \times A_2$ would have an exchange graph $\mathbb{Z}/5\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$, a pentagonal graph for each vertex is blown up to a pentagon.



The conjectured exchange graph for the rank 4 cluster algebra of type $A_2 \times A_2$.

Getting back to the low rank cases, the following are all of the exchange graphs for rank 1 or rank 2 cluster algebras of finite type. Several rank 3 cluster algebras have also been included.



Exchange graphs for some low rank cluster algebras of finite type.

Representative exchange matrices associated with each of these exchange graphs are:

$$
\begin{aligned}
A_1 &\leftrightarrow \begin{bmatrix} 0 \end{bmatrix} \\
A_1 \times A_1 &\leftrightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
A_2 &\leftrightarrow \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
B_2 &\leftrightarrow \begin{bmatrix} 0 & 1 \\ -2 & 0 \end{bmatrix} \\
G_2 &\leftrightarrow \begin{bmatrix} 0 & 1 \\ -3 & 0 \end{bmatrix} \\
A_1 \times A_1 \times A_1 &\leftrightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
A_2 \times A_1 &\leftrightarrow \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
B_2 \times A_1 &\leftrightarrow \begin{bmatrix} 0 & 1 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
G_2 \times A_1 &\leftrightarrow \begin{bmatrix} 0 & 1 & 0 \\ -3 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

A cluster algebra of type A_3 is the simplest rank 3 cluster algebra of finite type whose exchange graph cannot be described as the direct product of lower rank graphs. To construct its exchange graph we complete the following procedure.

3.7 A_3's Exchange Graph

First we note that A_3 has the associated Cartan matrix

$$
\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}.
$$

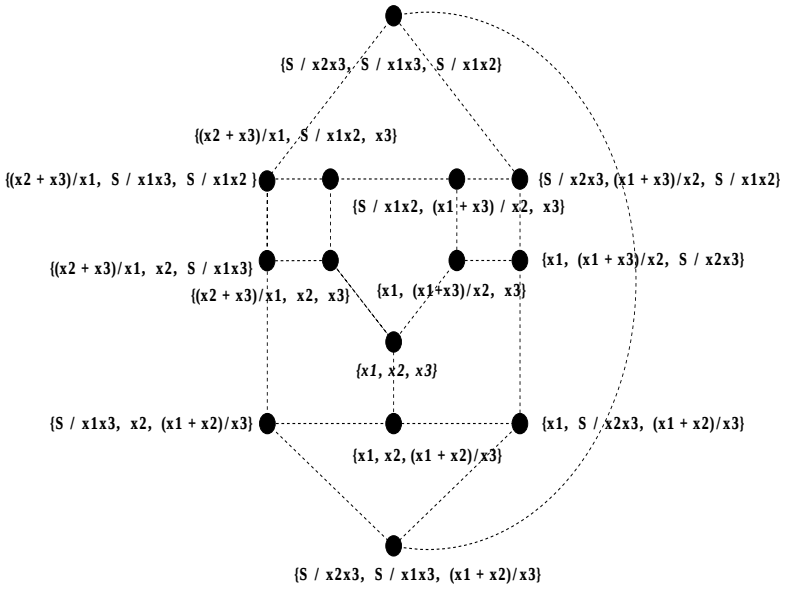
We will consider the cluster algebra where one of the clusters has

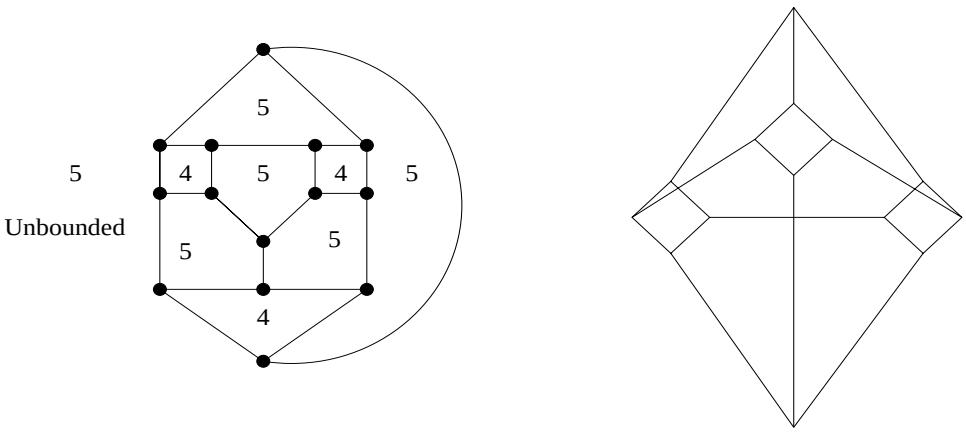
$$
M = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}
$$

as its exchange matrix. Mutating M as defined in section 3.3, we find

$$
\begin{aligned}
M_1 = \mu_1(M) &= \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \\
M_2 = \mu_2(M) &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \\
M_3 = \mu_3(M) &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}.
\end{aligned}
$$

M_1, M_2 and M_3 each are cyclic transformations of each other, and they each have two principal submatrices that are exchange graphs of type A_2 and one principal submatrix that is of type $A_1 \times A_1$. Furthermore, all of M's principal submatrices were of type A_2. From this, we deduce that the vertex associated with M lies at the junction of three pentagons, and the vertices associated with M_1, M_2 and M_3 each are at the junction of two pentagons and a rectangle. We continue to apply the matrix mutation functions μ_1, μ_2 and μ_3 to M_1, M_2, M_3 and beyond, and at the same time apply the corresponding exchange relations to the initial cluster $\{x_1, x_2, x_3\}$ (which is associated with exchange matrix M). We find that the following graph characterizes all of the clusters of this cluster algebra where $S = x_1 + x_2 + x_3$.





Two representations of the exchange graph for the cluster algebra of type A_3.

This exchange graph can be pictured in three dimensions as two tetrahedra glued together where all of the corners at the adjoining faces have been rubbed down to make square faces. An easy way to see this is that if one shrinks the faces of size 4 in the planar version of the graph (left-hand side) down to points, one is left with six faces of size 3 which forms two adjoined tetrahedra. I am grateful to Curtis T. McMullen for noticing this three dimensional characterization of this graph.

This graph also turns out to be the three dimensional *associahedron* [3]. Notice it has 14 vertices, the A_2 graph had 5 vertices, and the A_1 graph had 2 vertices. These are the Catalan numbers $C_2,\ C_3,\ C_4$ where $C_n = \frac{1}{n+1}\binom{2n}{n}$ and this is not a coincidence. In fact, each of these vertices correspond to a triangulation of a hexagon and there are $C_4 = 14$ ways of doing this (there are C_n triangulations of the $(n+2)$-gon). Another interpretation is that the polytope's vertices correspond to the ways you can associatively write a product, thus the name associahedron [18].

Fomin and Zelevinsky have a more general result that all exchange graphs for an A_n-type cluster algebra are n-dimensional associahedra. Similarly they have a result that all exchange graphs for a B_n-type (C_n-type) cluster algebra are n-dimensional cyclohedra. See [2] or [24] for details about the cyclohedron. In fact, they define the families of exchange graphs for cluster algebras of finite type associated with simple Lie algebras to be polytopes that they call *generalized associahedra* [3]. It appears that this would extend to a classification of many cluster algebras of finite type, namely a cluster algebra is determined by its exchange graph, and some possible exchange graphs would be direct products (as graphs) of the generalized associahedra.[6]

[6] As mentioned earlier, it is unclear if all cluster algebras would correspond to semisimple Lie algebras. Furthermore, we have restricted the definition of cluster algebra in this exposition by not allowing coefficients other than 1. If one allows coefficients from $\mathbb{P}$, an abelian group without torsion, the classification would be even more complicated. Additionally, we have not mentioned anything about the classification of cluster algebras of infinite type, such as the cluster algebra defined by the sequence g_n with $\mathcal{G}_{5,2}$ as its exchagne graph.

4 Open Problems

The caterpillar lemma is great for proving that certain sequences are Laurent sequences. However, the converse does not hold, and it cannot prove definitively that a sequence x_n is not Laurent. Is there a way to refine the condition to make this lemma an exact criterion? Or is there at least a way to determine for what kinds of sequences the caterpillar method will fail [4]?

The caterpillar lemma can prove that a sequence is Laurent, which in turn proves that the sequence is an integer sequence given that the first several terms are 1. However, this cannot determine whether the coefficients of the resulting Laurent polynomials are all nonnegative. It appears they are (Remark 1) and this would allow one to conclude the more powerful result that the sequence would be a sequence of nonnegative integers. Hence it could count combinatorial objects. Finding an explicit combinatorial interpretation for a sequence proves that sequence indeed consists of nonnegative integers. Is there a more universal way to assign such interpretations? The nonnegativity condition is also important because it appears that the dual canonical basis, the original motivation for the development of cluster algebras, should only involve nonnegative coefficients. This will be discussed briefly in the appendix.

Type A_n exchange graphs were identified as associahedra, and B_n (C_n) exchange graphs as cyclohedra in [3]. In this same article, Fomin and Zelevinsky ask about the structure of D_n exchange graphs. It would also be significant to classify exchange graphs of infinite type. Perhaps this can be done for rank 3 or at least for rank 3 cluster algebras which have exchange graphs that Zelevinsky refers to as tame, i.e. they are highly symmetrical like $\mathcal{G}_{5,2}$.

In particular, Fomin and Zelevinsky are pursuing a more complete classification of all cluster algebras (or at least tame ones) which would be analogous to the classification of semisimple Lie algebras or Kac-Moody algebras. Zelevinsky explained (personal communication) that seeing patterns and connections to Laurent sequences and associahedra help them develop insight as to patterns in the classification.

After a more explicit classification of cluster algebras has been formulated, one could explore the theory of Laurent polynomials and recurrence relations in more depth. In particular, we saw how the exchange graph $\mathcal{G}_{5,2}$ helped reveal secondary recurrences that the sequence $\{g_n\}$ satisfies. In practice one should be able to do this for other sequences, and the classification of cluster algebras and exchange patterns would correspond to a classification of families of recurrences where two recurrences R_1 and R_2 would be in the same family if any sequence that satisfies R_1 must also satisfy recurrence R_2.

5 Appendix: Fomin and Zelevinsky's Motivation for the Development of Cluster Algebras

The inspiration for the development of cluster algebras came from Fomin and Zelevinsky's study of the dual canonical basis of Quantum groups. In the following paragraphs, we will summarize the results and conjectures that led Fomin and Zelevinsky to create a new algebraic structure. First, we will recall some notation and results from [25]. We let $U^+ = U_q(\mathfrak{n}) \subseteq U_q(\mathfrak{g})$ be the subalgebra of the quantized universal enveloping algebra generated by elements E_i and let $R(w_0)$ be the set of *reduced words* for the permutation $w_0 = (n\ n-1\ \cdots 3\ 2\ 1)$. We will not give a more precise definition of E_i. For such a definition, see [25, pg. 8].

Remark 4 For every $\overline{i} \in R(w_0)$ and $t \in \mathbb{Z}^m_{\geq 0}$, there is a unique element $b = b_{\overline{i}}(t)$ of U^+ such that b and $b - p^{(t)}_{\overline{i}}$ is a linear combination of the elements of $\mathcal{B}_{\overline{i}}$, a basis with coefficients in $q^{-1}\mathbb{Z}[q^{-1}]$. It turns out $\mathcal{B}_{\overline{i}}$ is not dependent on the choice of $\overline{i}$ thus we let $\mathcal{B}$ be the *canonical basis*.

We will not define the elements $p^{(t)}_{\overline{i}}$, see [25, pg. 8] for a defintion. The importance of Remark 4 and the related notation is that it allows a definition of a *canonical basis* to make sense. Prior to their formulation of cluster algebras, the canonical basis, which is due to G. Lusztig [13], had been a main object of study for Fomin and Zelevinsky.

For example, Fomin and Zelevinsky describe a more explicit parameterization of the canonical basis $\mathcal{B}$ in [1] and [5]. Then to study more of the algebraic structure of $\mathcal{B}$, they investigated the *dual canonical basis* $\mathcal{B}^{dual}$ in the ring of regular functions $\mathbb{C}[N]$ where N is the maximal unipotent subgroup of the group under investigation. After many examples, a pattern hinting at an underlying algebraic structure emerged [25]. The properties of these algebraic structures were axiomatized as the theory of cluster algebras. Zelevinsky explains

> The dual canonical basis $\mathcal{B}^{dual}$ was constructed explicitly in several small rank cases. ... In all of these cases, $\mathcal{B}^{dual}$ consists of certain monomials in a distinguished family of generators. ... The monomials that constitute $\mathcal{B}^{dual}$ are defined by not allowing certain pairs of generators to appear together. In each case, the product of every two "incompatible" generators can be expressed as the sum of two allowed monomials [25, pg. 12].

In addition to the theory of dual canonical bases, Lusztig generalized the concept of totally positive[7] matrices to total positivity in any reductive group G.

Lusztig related the theory of total positivity back to the dual canonical basis. He showed that the elements of the dual canonical basis in $\mathbb{C}[G]$ take positive values. It is this connection that motivated remark 1, i.e. Fomin and Zelevinsky's

[7] A matrix is considered *totally positive* if all of its minors are positive. Such matrices are important in the study of differential equations and Polya frequency sequences [15].

conjecture that not only are the cluster variables Laurent polynomials in any other cluster but are Laurent polynomials with nonnegative coefficients. If such a conjecture was true, a more explicit correspondence between the dual canonical basis and cluster algebras might be possible. In fact Fomin and Zelevinsky conjecture that any coordinate ring $\mathbb{C}[G]$ or $\mathbb{C}[G/N]$ can be characterized as a cluster algebra assuming one is free to use coefficients in $\mathbb{P}$ other than 1 [7].

References

[1] A. Berkenstein, S. Fomin and A. Zelevinsky, Parametrizations of Canonical bases and totally positive matrices, *Adv. Math.* **122** (1996), 49-149.

[2] R. Bott and C. Taubes, On the self-linking of knots. Topology and physics, *J. Math. Phys.* **35** (1994), no. 10, 5247-5287.

[3] F. Chapoton, S. Fomin and A. Zelevinsky, Polytopal realizations of generalized associahedra, *Canadian Mathematical Bulletin*, To Appear.

[4] S. Fomin, The Laurent Phenomenon. Lecture given at *Northeastern Geometry-Algebra-Singularities-Combinatorics Seminar*, February 25, 2002.

[5] S. Fomin and A. Zelevinsky, Double Bruhat Cells and Total Positivity, *Journal of the American Mathematical Society* **12** (April 1999), no 2, 335-380.

[6] S. Fomin and A. Zelevinsky, Total Positivity: tests and parametrizations, *Math. Intelligencer* **22** (2000), no 1, 22-33.

[7] S. Fomin and A. Zelevinsky, Cluster Algebras I: Foundations, *Journal of the AMS* **15** (2002), 497-529.

[8] S. Fomin and A. Zelevinsky, The Laurent Phenomenon, *Adv. in Applied Math.* **28** (2002), 119-144.

[9] W. Futon and J. Harris, *Representation Theory: A First Course*, Springer-Verlag, 1991.

[10] D. Gale, The strange and surprising saga of the Somos sequences, *Math. Intelligencer* **13** (1991), no. 1, 40-43.

[11] I. Gessel, e-mail, October 25, 1999.

[12] J. Humphreys, *Reflection Groups and Coxter Groups*, Cambridge University Press, 1990.

[13] G. Lusztig, *Introduction to quantum groups*, Birkhäuser, Boston, 1993.

[14] R. Kane, *Reflection Groups and Invariant Theory*, Canadian Mathematical Society, 2001.

[15] S. Karlin, *Total Positivity: Volume 1*, Stanford University Press, 1968.

[16] E. Kuo, e-mail, October 28, 1999.

[17] E. Kuo, Applications of graphical condensation for enumerating matchings and tilings, (Preprint: February 27, 2001)

[18] C. W. Lee, The associahedron and triangulations of the n-gon, *European J. Combin.* **10** (1989), no. 6, 551-560.

[19] J. Propp, The Somos Sequence Site, www.math.wisc.edu/~propp/somos.html

[20] J. Propp, Lecture given in Mathematics 192: Algebraic Combinatorics, Harvard University, December 11, 2001.

[21] J. P. Serre, Trans. G. A. Jones, *Complex Semisimple Lie Algebras*, Springer-Verlag, 1966.

[22] N. J. Sloane, The Online Encyclopedia of Integer Sequences, www.research.att.com/~njas/sequences/

[23] D. Speyer, e-mail, December 12, 2001.

[24] J. D. Stasheff, From operads to "physically" inspired theories, *Contemp. Math.* **202** (1997), 53-81.

[25] A. Zelevinsky, From Littlewood coefficients to cluster algebras in three lecutres, (Preprint: December 6, 2001)

N° d'ordre: 1416

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par Olivier GUIBERT

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Combinatoire des permutations à motifs exclus en liaison avec mots, cartes planaires et tableaux de Young

---

Soutenue le : 15 Décembre 1995

Après avis de : MM. Dominique Gouyou-Beauchamps Rapporteurs
Renzo Pinzani

Devant la Commission d'examen formée de :

MM.	Robert Cori	Professeur	Président
	Jean-Guy Penaud	Professeur	Rapporteur
	Serge Dulucq	Professeur	Examinateurs
	Dominique Gouyou-Beauchamps	Professeur	
	Renzo Pinzani	Professeur	
	Christophe Reutenauer	Professeur	
	Timothy Walsh	Professeur	

– 1995 –

A Florence.

J'exprime mon immense gratitude à Robert Cori dont les cours ont été à l'origine de l'intérêt que je porte à l'informatique théorique et à la combinatoire. Je lui en suis profondément reconnaissant et je le remercie de me faire l'honneur de présider ce jury.

Je suis très touché de l'attention portée par Dominique Gouyou-Beauchamps et Renzo Pinzani à ce mémoire en qualité de rapporteur. Je leur en suis extrêmement reconnaissant et je leur adresse mes plus vifs remerciements.

J'ai eu le plaisir de rencontrer Christophe Reutenauer et Timothy Walsh durant mon séjour à Montréal. Je suis très heureux de pouvoir les compter parmi les membres du jury.

Il m'est agréable de remercier Jean-Guy Penaud d'avoir su prolonger mon intérêt pour la combinatoire. Il me fait la gentillesse d'accepter de participer au jury.

J'adresse mes plus sincères remerciements à Serge Dulucq qui a dirigé mon travail. Je lui sais tout particulièrement gré de sa confiance, son attention, sa disponibilité et de la liberté qu'il m'a accordée. Que ces quelques lignes expriment tout le respect que je lui porte.

Que soit ici remercié Xavier Viennot qui, par son enthousiasme, rend communicative sa passion pour la combinatoire bijective. Mes remerciements s'adressent également aux autres membres des équipes combinatoires énumérative et algorithmique du LaBRI, pour leur soutien et leurs conseils renouvelés.

Je remercie Pierre Leroux qui m'a chaleureusement accueilli au LaCIM, quatre mois durant, dans le cadre de la coopération franco-québécoise. Mes remerciements vont aussi aux autres membres de ce laboratoire, et tout particulièrement à Srećko Brlek qui aura été bien plus que mon responsable durant cette période, me faisant notamment connaître la patinoire de Saint-Jean de Matha.

Un très grand merci à Sophie Gire ; elle sait toute l'amitié que j'ai pour elle.

A ces remerciements, j'associe les doctorants qui m'ont accompagné durant la préparation de cette thèse.

Table des Matières

Introduction

Le travail que nous présentons ici a pour thème la Combinatoire des permutations, et porte plus particulièrement sur l'énumération de permutations à motifs exclus, c'est à dire de permutations pour lesquelles certaines sous-suites (sous-mots) d'un type donné sont interdites.

Un ensemble particulier de permutations à motifs exclus, les permutations triables par deux passages consécutifs dans une pile, nous amène à résoudre plusieurs conjectures portant sur l'énumération de certaines classes de mots et de tableaux de Young standard.

La plupart des résultats sont obtenus en mettant en correspondance les ensembles considérés avec des objets classiques en Combinatoire, notamment certaines familles de cartes planaires.

Avant-propos

L'un des centres d'intérêt important en Combinatoire des mots [71] est constitué par la recherche et l'analyse de régularités dans les mots, et de façon duale la recherche de mots ne comportant pas certaines régularités.
Ces propriétés de régularités, à rechercher ou à éviter, s'expriment souvent en terme de facteurs ou de sous-mots.

Ainsi, A. Thue [100, 101, 71, 8] a été le premier à chercher à mettre en évidence des mots ne comportant pas certaines régularités, et ce plus particulièrement les mots sans facteur chevauchant et les mots sans carré. Par exemple, le mot de Thue-Morse *abbabaabbaababba* . . ., qui ne comporte pas de facteurs se chevauchant (et est donc sans cube), permet d'obtenir un mot sans carré sur un alphabet à trois lettres.
D'autre part, la présence de sous-mots particuliers dans un mot est lié au principe de régularité illustré par le théorème de B.L. Van der Waerden [106]. Tout mot suffisamment long sur un alphabet fini contient une même lettre à des positions satisfaisant une progression arithmétique. Ce résultat, depuis sa démonstration, a suscité de nombreux travaux dans divers domaines [71].

L'intérêt porté aux sous-mots s'est également manifesté dans diverses directions, comme l'atteste le chapitre 6 de M. Lothaire [71]. Par exemple, dans l'ensemble partiellement ordonné construit en considérant la relation d'ordre partiel "être sous-mot de", tout ensemble de mots deux à deux incomparables sur un alphabet fini est lui-même fini. Toutefois, il existe de tels ensembles de mots deux à deux incomparables arbitrairement grands. Ce résultat, dû à G. Higman

[58], a été maintes fois redécouvert.

Ainsi, de nombreux travaux se situant dans divers domaines ont porté sur les mots comportant ou excluant des facteurs ou des sous-mots particuliers.

Ceux que nous présentons ici ont pour cadre les permutations à motifs exclus, c'est à dire les permutations ne comportant pas certaines sous-suites d'un type donné. Par exemple, les permutations n'admettant pas de sous-suite croissante de longueur supérieure à k sont les permutations excluant le motif identité $12\ldots(k+1)$ en correspondance avec les paires de tableaux de Young standard de même forme et dont la plus grande part de la partition est au plus égale à k.

La plupart des travaux sur le sujet ont eu pour objectif d'énumérer des ensembles spécifiques de permutations à motifs exclus. Certains travaux ont toutefois permis d'obtenir des résultats plus généraux, comme par exemple ceux d'A. Regev [79] qui a donné une expression pour le comportement asymptotique du nombre de permutations ne comportant pas de motif $12\ldots k$, et de P. Erdös et G. Szekeres [34] qui ont montré qu'aucune permutation d'ordre supérieur à $l.m$ ne peut exclure simultanément les motifs $12\ldots(l+1)$ et $(m+1)m\ldots1$. Il est à noter que ce dernier résultat, antérieur à la correspondance de Robinson-Schensted [83, 89], en est une conséquence immédiate.

Les autres travaux portent pour la plupart sur l'énumération des permutations ne comportant pas un ou plusieurs motifs de forme donnée. Citons dans ce cadre les travaux de R. Simion et F.W. Schmidt [95] où les motifs interdits correspondent à des permutations de S_3 (permutations ayant 3 éléments) et ceux de J. West [114] où les deux motifs exclus appartiennent à S_3 et S_4.

D'autres auteurs ont pour leur part étudié certains ensembles de permutations qui peuvent être caractérisées en terme de permutations à motifs exclus.
Par exemple, les permutations vexillaires considérées par A. Lascoux et M.P. Schützenberger [69] sont les permutations ne comportant pas de motif de type 2143, c'est à dire de sous-suite $jilk$ avec $i<j<k<l$. Rappelons qu'une permutation est vexillaire si et seulement si les partitions correspondant aux tables d'inversion (ou codes de Lehmer) de cette permutation et de son inverse sont conjuguées. Notons qu'une formule d'énumération existe pour ces permutations et résulte d'un article d'I.M. Gessel [42] et d'un travail de J. West [110] consacré aux permutations à motifs exclus qui montre que les permutations ne comportant pas le motif 2143 et celles interdisant le motif 1234 sont en bijection.
De même, S. Gire [45] a caractérisé les permutations de Baxter [4] comme étant celles excluant simultanément les motifs $25\overline{3}14$ (c'est à dire les sous-suites de type 2413 ne faisant pas elles-mêmes partie de sous-suites de type 25314) et $41\overline{3}52$. Les permutations de Baxter ont fait l'objet de plusieurs travaux [15, 74, 108, 30] ayant pour but d'établir une formule d'énumération tenant compte de différentes distributions de ces permutations.

Ce sont également les permutations de Baxter, mais alternantes cette fois-ci, qui interviennent dans un article de R. Cori, S. Dulucq et X. Viennot [18] mettant en correspondance les mélanges

de deux mots de systèmes de parenthèses bien formés et les couples de tels mots de parenthèses. En fait, il n'existe que peu de résultats concernant l'énumération des permutations alternantes à motifs exclus, et il en est de même pour le dénombrement des involutions à motifs exclus.
Cependant, les involutions ne comportant pas le motif $12\ldots(k+1)$ sont en correspondance, par l'algorithme de Robinson-Schensted [83, 89], et d'après un résultat de M.P. Schützenberger [92], avec les tableaux de Young standard de hauteur au plus k. Parmi les travaux ayant porté sur l'énumération de ces tableaux, citons ceux d'A. Regev [79] qui donne le comportement asymptotique du nombre de tels tableaux de hauteur au plus k, ainsi qu'une formule exacte très classique en Combinatoire (nombres de Motzkin) pour ceux de hauteur au plus 3. Pour sa part, D. Gouyou-Beauchamps [49] a obtenu combinatoirement des formules remarquables (notamment le produit de deux nombres de Catalan) pour de tels tableaux de hauteur au plus 4 et 5.

De nombreuses suites de nombres, très classiques en Combinatoire, apparaissent dans des problèmes d'énumération de permutations à motifs exclus. C'est le cas des nombres de Pell, des coefficients binomiaux centraux, des nombres de Motzkin ou encore des nombres de Schröder qui sont obtenus par l'interdiction de motifs particuliers. Certains de ces résultats ont été obtenus par J. West [110, 112] ou par S. Gire [45]. De même, suite aux travaux de D.E. Knuth [62], les permutations excluant un motif quelconque correspondant à une permutation de S_3 sont énumérées par les nombres de Catalan. En effet, D.E. Knuth s'est intéressé aux permutations triables par passage dans une pile et a montré qu'elles correspondent exactement aux permutations ne comportant pas de sous-suite de type 231. Du fait de leur nombre, ces permutations sont parfois appelées permutations de Catalan.
En considérant l'une des généralisations possibles de ce problème de tri, J. West [110, 113] a montré que les permutations triables par deux passages consécutifs dans une pile sont exactement les permutations ne comportant pas de motif 2341 et $3\overline{5}241$. Il conjecturait également une remarquable formule pour l'énumération de ces permutations, conjecture démontrée par D. Zeilberger [119]. Par la suite, S. Dulucq, S. Gire et J. West [28] et S. Dulucq, S. Gire et O. Guibert [27] ont mis en correspondance permutations triables par deux passages consécutifs dans une pile, permutations non séparables (excluant simultanément les motifs 2413 et $41\overline{3}52$) et cartes planaires pointées non séparables, ces cartes ayant été énumérées par W.T. Tutte [105] et leur nombre correspondant exactement à la formule conjecturée par J. West.

L'intérêt porté aux cartes planaires remonte au célèbre problème des quatre couleurs et les travaux de W.T. Tutte [102, 103, 104, 105] sur l'énumération des cartes planaires avaient pour objectif de déterminer le nombre de cartes planaires 4-coloriables et de le comparer au nombre de cartes planaires. Ces travaux ont eu ensuite de nombreux développements, notamment sous l'impulsion de R. Cori [16], de R. Cori et J. Richard [19] et de D. Arquès [1].
Or, les connexions entre cartes planaires et permutations à motifs exclus ne se limitent pas uniquement aux cartes planaires pointées non séparables et permutations non séparables. Ainsi, comme nous le montrons dans ces travaux, les permutations non séparables alternantes sont en

bijection avec les cartes planaires cubiques pointées non séparables dénombrées par W.T. Tutte [103]. D'autre part, la correspondance de R. Cori, S. Dulucq et X. Viennot [18] entre mots du mélange de deux mots de parenthèses et couples de mots de parenthèses, qui fait intervenir les permutations de Baxter alternantes (et donc des permutations à motifs exclus), résout un problème posé par R.C. Mullin [76] sur les cartes planaires. En effet, les mots du mélange de deux mots de parenthèses codent les cartes planaires pointées cubiques hamiltoniennes tandis que les couples de mots de parenthèses sont en bijection avec les cartes planaires pointées triangulaires hamiltoniennes.

Présentation générale de nos travaux

En Combinatoire, plusieurs approches ou méthodes sont possibles pour énumérer une famille d'objets combinatoires, et leur emploi dépend du contexte du problème.
Parmi les plus classiques, citons la méthodologie due à M.P. Schützenberger [91, 93], qui a montré l'existence d'une étroite relation entre problèmes d'énumération en Combinatoire et classification de langages en théorie des langages. Cette méthode permet d'obtenir une équation algébrique dont est solution la série génératrice des objets considérés et la formule de Lagrange permet d'obtenir une expression pour le nombre d'objets de taille n. Quand cette méthode ne s'applique pas, le recours aux équations avec opérateurs [16, 19] permet dans certains cas d'aboutir au résultat.
D'autres approches sont possibles, comme la théorie des espèces de structures dont A. Joyal [60] a montré l'efficacité pour le traitement combinatoire des séries formelles. Le livre de F. Bergeron, G. Labelle et P. Leroux [7] expose cette méthodologie dans ses moindres détails.
Une autre méthode consiste à trouver une bijection entre les objets considérés et une autre famille d'objets pour laquelle des résultats d'énumération sont connus ou plus simples à établir. A propos de la formule des équerres dénombrant les tableaux de Young standard d'une forme donnée, D.E. Knuth [63] estime que toute formule simple devrait avoir une explication naturelle. Depuis, de nombreux travaux ont tenté d'obtenir une telle explication pour cette formule des équerres, usant notamment de preuves probabilistique [52] et combinatoires [80, 40, 117, 78, 64].

Dans les travaux que nous présentons ici, nous aurons recours à ces méthodes bijectives, et à l'utilisation de la méthode des arbres de génération utilisée pour la première fois de manière explicite par F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15] dans leurs travaux sur l'énumération des permutations de Baxter [4].
Cette méthode consiste à considérer un arbre de génération des objets étudiés, obtenu par certaines règles de croissance de ces objets, et à caractériser cet arbre par un système de réécriture où chaque règle de réécriture décrit une règle de croissance. Des équations de récurrence peuvent alors être déduites de ce système de réécriture et permettre, dans certains cas, d'obtenir une formule d'énumération pour les objets étudiés. Notons que lorsque deux familles d'objets

combinatoires ont des arbres de génération caractérisés par le même système de réécriture, ces deux arbres sont isomorphes et cela induit une bijection entre ces objets.
F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15] se sont étonné que, jusqu'alors, personne n'ait utilisé cette méthode des arbres de génération. Depuis, et plus particulièrement suite à la thèse de J. West [110], cette approche a inspiré de nombreux travaux portant principalement sur l'énumération de permutations à motifs exclus [53, 45, 28, 27, 97, 114, 112].
Notons également qu'E. Barcucci, A. Del Lungo, E. Pergola et R. Pinzani [3] ont utilisé une démarche similaire pour obtenir de nouvelles équations fonctionnelles dont sont solutions les séries génératrices de plusieurs objets classiques en combinatoires tels certaines classes d'arbres.

Afin de faciliter nos recherches sur le dénombrement de permutations à motifs exclus, nous avons développé un logiciel, dénommé *forbid*, qui met en œuvre cette méthode des arbres de génération dans le cas des permutations, des permutations alternantes et des involutions à motifs exclus. De plus, ce logiciel donne la distribution de ces objets suivant la plupart des paramètres classiques sur les permutations.

C'est notamment grâce au logiciel *forbid* et à cette méthode des arbres de génération que nous avons pu caractériser les arbres de génération des permutations et involutions excluant le motif identité.
Cela nous a également permis de prouver combinatoirement que plusieurs ensembles de permutations à motifs exclus sont énumérés par des formules classiques en Combinatoire. Par exemple, certains sont énumérés par les nombres de Pell tandis que d'autres sont dénombrés par les coefficients binomiaux centraux. De même, nous avons montré que des ensembles de permutations à motifs exclus sont en bijection avec les arbres 1-2, prolongeant ainsi les travaux de S. Gire [45] sur de tels ensembles énumérés par les nombres de Motzkin, et ceux de J. West [110, 112] et de S. Gire [45] sur ceux énumérés par les nombres de Schröder. Ensuite, nous avons mis en bijection plusieurs objets combinatoires avec les permutations de Baxter et un autre ensemble de permutations à motifs exclus.

Nous avons prolongé ce premier travail sur les permutations de Baxter et avons mis en évidence une nouvelle correspondance (voir également S. Dulucq et O. Guibert [30]) entre ces permutations et certains triplets de chemins deux à deux disjoints. Cette correspondance unifie les preuves combinatoires pour l'énumération des permutations de Baxter de X. Viennot [108] et des permutations de Baxter alternantes de R. Cori, S. Dulucq et X. Viennot [18]. De plus, elle nous permet d'affiner plusieurs formules d'énumération connues. Ainsi, nous donnons notamment une interprétation combinatoire d'une formule due à C.L. Mallows [74] qui elle-même précise celle de F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15].

A mi-chemin entre les permutations de Baxter qui excluent les motifs $25\overline{3}14$ et $41\overline{3}52$ et les permutations excluant les motifs 2413 et 3142 (énumérées par les nombres de Schröder [112]) se trouvent les permutations non séparables ne comportant pas les motifs 2413 et $41\overline{3}52$. En effet, les motifs 2413 et 3142 s'obtiennent à partir des motifs respectivement $25\overline{3}14$ et $41\overline{3}52$ en

supprimant l'élément barré.
S. Dulucq, S. Gire et J. West [28] ont montré que ces permutations non séparables sont directement en bijection avec les cartes planaires pointées non séparables par isomorphisme de leurs arbres de génération. Nous avons montré (voir également S. Gire [45] et S. Dulucq, S. Gire et O. Guibert [27]) que ces permutations sont également en bijection avec les permutations triables par deux passages consécutifs dans une pile. La réunion de ces travaux [28, 27] constitue donc une preuve de la conjecture de J. West [110, 113] sur l'énumération de ces permutations.

D.E. Knuth s'est intéressé à plusieurs algorithmes de tri et, en particulier, a considéré les permutations triables par passage dans une pile, montrant qu'il s'agissait des permutations ne comportant pas le motif 231. Dans cet algorithme, la pile ne peut contenir à tout instant que des entiers allant en croissant à partir du sommet; ainsi, dans un certain sens, la pile vérifie une condition dite de type "tour de Hanoï" par référence au problème du même nom. La généralisation de ce problème considérée par J. West [110, 113] consiste à imposer cette contrainte sur les piles à chaque passage des éléments de la permutation.

Pour sa part, S. Gire [45] s'est intéressée, non plus aux seules permutations, mais également aux mouvements de deux piles placées en série lorsqu'elles sont traversées par la permutation identité. Le langage obtenu est le langage de Yamanushi codant les tableaux de Young standard rectangulaires de hauteur 3.
Elle a conjecturé des formules d'énumération pour trois langages (l'un d'entre-eux prenant en compte la contrainte "tour de Hanoï") correspondant à des restrictions sur ces tableaux, notant que ces formules dénombraient également les permutations de Baxter alternantes, les permutations de Baxter et les cartes planaires cubiques pointées non séparables, cartes énumérées par W.T. Tutte [103]. Nous avons prouvé combinatoirement ces conjectures (voir également S. Dulucq et O. Guibert [29] pour deux d'entre-elles). Nous remarquons ainsi de nouveau les liens étroits existant entre permutations de Baxter (alternantes ou non) et permutations non séparables (alternantes ou non) : les permutations non séparables alternantes sont en bijection avec les cartes planaires cubiques pointées non séparables et un quatrième langage se dégageant naturellement des trois considérés par S. Gire est directement en correspondance avec les permutations non séparables. Il est intéressant de constater que les permutations non séparables, apparues pour résoudre le problème des permutations triables par deux passages consécutifs dans une pile, interviennent de nouveau dans ce travail sur les mots de piles.

Il est également surprenant d'observer qu'une même formule dénombre les permutations de Baxter alternantes étudiées par R. Cori, S. Dulucq et X. Viennot [18] et les involutions excluant le motif 54321 considérées par D. Gouyou-Beauchamps [49]. Toutefois, les preuves combinatoires apportées par ces auteurs sont différentes et ne permettent pas de mettre directement en bijection ces ensembles de permutations alternantes et d'involutions à motifs exclus. C'est l'une des raisons qui a motivé l'intégration du traitement des permutations alternantes et involutions à motifs exclus au logiciel *forbid*.

Plan détaillé de la thèse

Cette thèse s'articule en deux parties. La première partie, constituée des chapitres 1 à 3, expose les objets, résultats classiques, méthodes et outils utilisés. La seconde partie, allant des chapitres 4 à 7, détaille les résultats que nous avons obtenus. L'annexe A constitue un catalogue des résultats que nous connaissons à ce jour sur les permutations à motifs exclus.

Le premier chapitre présente les objets étudiés dans cette thèse.
Il s'agit des permutations, et plus particulièrement des permutations à motifs exclus, permutations pour lesquelles certaines sous-suites sont interdites. Quelques bijections classiques et résultats généraux complètent cette présentation.
Nous consacrons la fin de ce chapitre à des rappels de bijections classiques mettant en correspondance des objets fréquemment étudiés en Combinatoire. Dans le même temps, nous donnons des formules les dénombrant. Nous rencontrerons par la suite ces objets que nous mettrons alors en bijection avec certaines de permutations à motifs exclus.

Le deuxième chapitre expose la méthode des arbres de génération d'objets combinatoires.
Il s'agit de construire un arbre infini dont les sommets sont les objets de l'ensemble étudié de telle manière qu'il contienne au niveau n tous les objets de taille n, chacun apparaissant une fois et une seule. Chaque objet est ensuite remplacé dans l'arbre par une étiquette qui le caractérise, de sorte que l'arbre ainsi étiqueté correspond à l'arbre de dérivation d'un système de réécriture. Ainsi, la donnée d'une part d'un axiome correspondant à l'étiquette de la racine et d'autre part d'un ensemble de règles de réécriture précisant les étiquettes de tous les fils d'un sommet d'une étiquette donnée, permet de caractériser l'arbre de génération des objets considérés.
Cette méthode présente plusieurs intérêts. Tout d'abord, lorsque les arbres de génération de deux ensembles d'objets combinatoires se caractérisent par le même système de réécriture, ils sont isomorphes et induisent ainsi une bijection entre les deux ensembles considérés. Ensuite, il est possible, à partir d'un système de réécriture quelconque, d'obtenir des récurrences permettant d'énumérer l'ensemble d'objets combinatoires étudié. Ce sont les deux principales applications de cette méthode, même si d'autres utilisations s'avèrent possibles, comme la génération aléatoire par exemple.

La méthode des arbres de génération peut bien évidemment s'appliquer aux permutations à motifs exclus et être programmée. C'est ce qu'illustre le troisième chapitre.
L'arbre de génération des permutations s'obtient en insérant l'élément $n+1$ dans une permutation d'ordre n. Nous privilégions cette façon de faire croître les permutations, même s'il existe d'autres possibilités [45, 97] qui conduisent à des arbres de génération des permutations éventuellement différents. Par contre, nous considérons d'autres familles de permutations, les involutions et les permutations alternantes. Ainsi, nous obtenons deux arbres de génération des involutions, c'est à dire deux façons de faire croître les involutions, et un arbre de génération des permutations alternantes.
Nous avons développé le logiciel *forbid* qui permet de construire les arbres de génération de l'une

quelconque des familles de permutations, pour un ou plusieurs motifs exclus choisi par l'utilisateur. *forbid* fournit alors des informations sur l'arbre de génération, sur l'ensemble étudié, et des distributions de cet ensemble selon des paramètres classiques sur les permutations.

En utilisant la méthode des arbres de génération, et avec l'aide du logiciel *forbid*, nous avons montré que plusieurs ensembles de permutations à motifs exclus sont énumérés par des suites classiques en Combinatoire. Ces résultats figurent dans le quatrième chapitre.
Tout d'abord, nous obtenons que les arbres de génération de trois ensembles de permutations à motifs exclus, énumérés par les nombres de Pell, se caractérisent tous par le même système de réécriture.
Ensuite, nous mettons onze ensembles de permutations à motifs exclus en bijection avec les mots du Grand Dyck dénombrés par les coefficients binomiaux centraux, quatre systèmes de réécriture différents étant nécessaires pour établir cette bijection. De plus, nous obtenons un autre ensemble de permutations à motifs exclus ayant même formule d'énumération.
S. Gire [45] a exhibé un système de réécriture caractérisant les arbres de génération des arbres 1-2 (suivant le nombre d'arêtes) et d'un ensemble de permutations à motifs exclus, objets énumérés par les nombres de Motzkin. Nous montrons que ce système de réécriture caractérise également les arbres de génération des buissons, d'un autre ensemble de permutations à motifs exclus et de deux ensembles d'involutions à motifs exclus.
J. West [110, 112] a caractérisé le système de réécriture de l'arbre de génération d'un ensemble de permutations à motifs exclus énuméré par les nombres de Schröder et S. Gire [45] a montré que ce système de réécriture caractérise les arbres de génération d'un autre ensemble de permutations à motifs exclus et des arbres 1-2 (suivant le nombre de sommets internes). Nous prouvons que le même système de réécriture caractérise les arbres de génération de huit autres ensembles de permutations à motifs exclus.
Enfin, nous caractérisons les systèmes de réécriture des tableaux et paires de tableaux de Young standard de hauteur bornée, c'est à dire des involutions et permutations excluant le motif identité.

Le cinquième chapitre prouve combinatoirement une conjecture de J. West [110, 113].
La conjecture de J. West consistait à relier combinatoirement les permutations triables par deux passages consécutifs dans une pile et les cartes planaires pointées non séparables. S. Dulucq, S. Gire et J. West [28] ont établi une partie de cette conjecture en mettant en correspondance permutations non séparables et cartes planaires pointées non séparables. Notre travail, également présenté dans la thèse de S. Gire [45] et dans l'article de S. Dulucq, S. Gire et O. Guibert [27], consiste en une bijection entre permutations triables par deux passages consécutifs dans une pile et permutations non séparables, et utilise de nouveau la méthode des arbres de génération des permutations.
Après avoir rappelé le résultat général, nous détaillons l'une des quatre correspondances de la bijection, c'est à dire l'un des quatre systèmes de réécriture et les deux arbres de génération des

permutations ainsi caractérisés. Ensuite, nous montrons qu'un autre ensemble de permutations à motifs exclus se relie combinatoirement à la bijection.

Le sixième chapitre est entièrement consacré aux permutations de Baxter [4].
S. Gire [45] a caractérisé le système de réécriture de l'arbre de génération des permutations de Baxter. Nous montrons dans un premier temps que ce système de réécriture caractérise également les arbres de génération d'autres objets combinatoires.
Ensuite, nous établissons une bijection entre permutations de Baxter et triplets de chemins deux à deux disjoints, différente de celle de X. Viennot [108]. Nous obtenons combinatoirement une nouvelle formule dénombrant les permutations de Baxter qui généralise celles de F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15] et de C.L. Mallows [74]. De plus, cette bijection traduit naturellement la propriété d'alternance, nous permettant de retrouver et d'affiner le résultat de R. Cori, S. Dulucq et X. Viennot [18] sur l'énumération des permutations de Baxter alternantes.

Le septième chapitre résout trois conjectures de S. Gire [45].
S. Gire a considéré trois restrictions différentes de l'ensemble des mots de piles, conjecturant les formules d'énumération des trois langages ainsi obtenus. Nous établissons tout d'abord une bijection entre le premier de ces langages et l'ensemble des permutations de Baxter alternantes, permutations dénombrées combinatoirement par R. Cori, S. Dulucq et X. Viennot [18]. Or, les deux derniers langages étudiés par S. Gire sont tous deux des restrictions différentes du premier. En considérant ces deux restrictions dans la bijection obtenue pour le premier langage, nous avons caractérisé les deux ensembles de permutations résultants : il s'agit des permutations de Baxter et des permutations non séparables alternantes. Nous mettons alors en bijection ces permutations non séparables alternantes et les cartes planaires cubiques pointées non séparables, cartes dénombrées par W.T. Tutte [103]. De plus, nous établissons une bijection entre permutations non séparables et mots appartenant à l'intersection des deuxième et troisième langages considérés par S. Gire. Enfin, nous nous sommes intéressés à d'autres restrictions naturelles de l'ensemble des mots de piles et avons ainsi obtenu plusieurs autres résultats d'énumération.

Chapitre 1

Généralités

Dans ce chapitre, nous rappelons quelques notions classiques sur les permutations et les permutations à motifs exclus, et présentons quelques objets combinatoires classiques.

1.1 Permutations

Une *permutation* $\pi = \pi(1)\pi(2)\dots\pi(n)$ sur $[n] = \{1, 2, \dots, n\}$, notée sous forme de mot, est une bijection de $[n]$ dans $[n]$.
S_n désigne l'ensemble des $n!$ permutations sur $[n]$.

Définition 1.1 *Pour toute permutation π de S_n, nous notons*

- π^* *la permutation* miroir *de π définie par $\pi^*(i) = \pi(n+1-i)$ pour tout $i \in [n]$*
- π^c *la permutation* complémentaire *de π définie par $\pi^c(i) = n+1-\pi(i)$ pour tout $i \in [n]$*
- π^{-1} *la permutation* inverse *de π vérifiant $\pi^{-1}(i) = j \iff \pi(j) = i$ pour tout $i, j \in [n]$*

Sur l'ensemble des permutations de S_n, nous serons amenés à nous intéresser aux paramètres suivants.

Définition 1.2 *Etant donnée une permutation π de S_n, nous appelons*

- descente *un indice $i \in [n-1]$ tel que $\pi(i) > \pi(i+1)$;*
  le nombre de descentes de π est noté $desc(\pi)$
- montée *un indice $i \in [n-1]$ tel que $\pi(i) < \pi(i+1)$;*
  le nombre de montées de π est noté $mont(\pi)$
- descente inverse *un indice $i \in [n-1]$ tel que $\pi^{-1}(i) > \pi^{-1}(i+1)$;*
  le nombre de descentes inverses de π est noté $descinv(\pi)$

- montée inverse *un indice* $i \in [n-1]$ *tel que* $\pi^{-1}(i) < \pi^{-1}(i+1)$;
  le nombre de montées inverses de π *est noté* $montinv(\pi)$

- maximum à gauche *un élément* $\pi(i)$ *tel que* $\pi(i) > \pi(j)$ *pour tout* $1 \leq j < i$;
  le nombre de maxima à gauche de π *est noté* $maxg(\pi)$

- maximum à droite *un élément* $\pi(i)$ *tel que* $\pi(i) > \pi(j)$ *pour tout* $i < j \leq n$;
  le nombre de maxima à droite de π *est noté* $maxd(\pi)$

- minimum à gauche *un élément* $\pi(i)$ *tel que* $\pi(i) < \pi(j)$ *pour tout* $1 \leq j < i$;
  le nombre de minima à gauche de π *est noté* $ming(\pi)$

- minimum à droite *un élément* $\pi(i)$ *tel que* $\pi(i) < \pi(j)$ *pour tout* $i < j \leq n$;
  le nombre de minima à droite de π *est noté* $mind(\pi)$

Exemple 1.3 *La permutation* 761254893 *possède* 4 *descentes (indices* 1,2,5,8*),* 4 *montées (indices* 3,4,6,7*),* 4 *descentes inverses (indices* 3,4,5,6*),* 4 *montées inverses (indices* 1,2,7,8*),* 3 *maxima à gauche (éléments* 7,8,9*),* 2 *maxima à droite (éléments* 3,9*),* 3 *minima à gauche (éléments* 7,6,1*),* 3 *minima à droite (éléments* 3,2,1*).*

Propriété 1.4 *Pour toute permutation* π *de* S_n*, nous avons* $desc(\pi) + mont(\pi) = n - 1$ *et les relations décrites par la figure 1.1.*

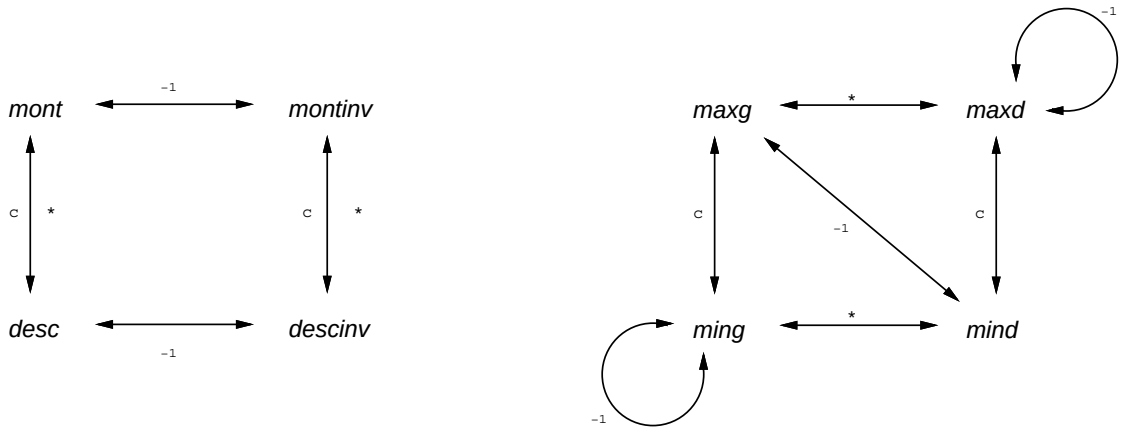


Figure 1.1 Effet des trois bijections classiques sur les paramètres montées/descentes et les minima/maxima.

Nous désignons par I_n l'ensemble des *involutions* sur $[n]$, c'est à dire les permutations π de S_n vérifiant $\pi = \pi^{-1}$. Un élément i d'une involution π est un *point fixe* si et seulement si $\pi(i) = i$.

Exemple 1.5 *La permutation* 62583174 *est une involution de* I_8 *ayant* 2 *points fixes (éléments* 2,7*).*

Nous désignons par $\widehat{S}_n$ l'ensemble des *permutations alternantes* sur $[n]$, c'est à dire les permutations π de S_n vérifiant $\pi(2i-1) < \pi(2i) > \pi(2i+1)$ pour tout $i \in [\lfloor \frac{n}{2} \rfloor]$.

Exemple 1.6 *La permutation* 381429675 *est une permutation alternante de* $\widehat{S}_9$.

Plusieurs bijections classiques relient les permutations à d'autres objets combinatoires. Nous en citons deux (voir figure 1.2) que nous utiliserons par la suite.

- La correspondance de Robinson-Schensted [83, 89], dont X. Viennot [107] a donné une interprétation géométrique, met en bijection les permutations et les paires de tableaux de Young standard de même forme.

- La construction suivante (voir notamment [98]) permet d'obtenir un arbre binaire (complet) croissant à partir d'une permutation (alternante) donnée.

  $$abc(u) = \lhd\, abc(v), x, abc(w) \,\rhd \text{ où } u = vxw, x = \min\{u_i : u = u_1 u_2 \dots u_p\}$$

  Cette construction est bijective ; il suffit de lire la projection en ordre infixe de l'étiquetage des sommets de l'arbre binaire croissant pour obtenir la permutation.

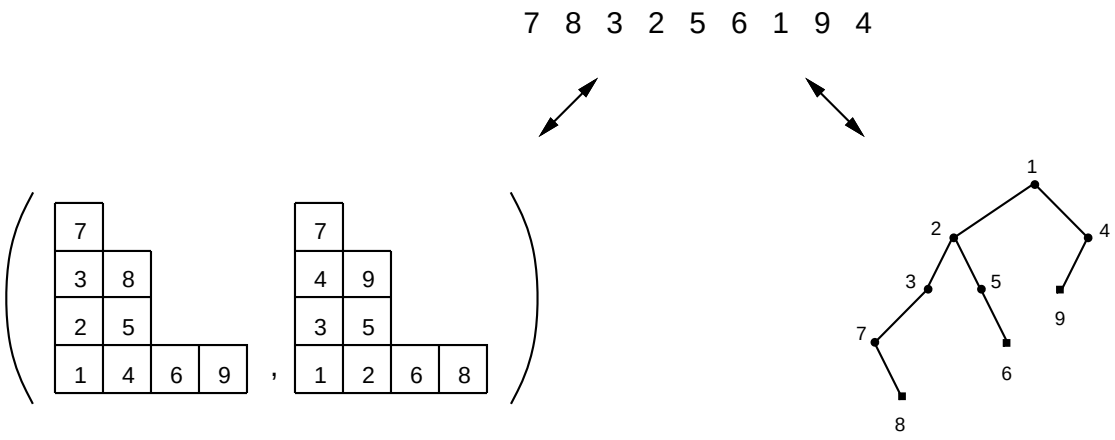


Figure 1.2 Permutation en bijection avec une paire de tableaux de Young standard et un arbre binaire croissant.

1.2 Permutations à motifs exclus

Nous allons maintenant introduire les *permutations à motifs exclus*, c'est à dire les permutations pour lesquelles certaines sous-suites (sous-mots) sont interdites.

Définition 1.7 *Une permutation* π *de* S_n *contient une* sous-suite *de* type τ *appartenant à* S_k *si et seulement s'il existe une suite d'indices* $1 \leq i_{\tau(1)} < i_{\tau(2)} < \cdots < i_{\tau(k)} \leq n$ *tels que* $\pi(i_1) < \pi(i_2) < \cdots < \pi(i_k)$.

Nous notons $S_n(\tau)$ l'ensemble des permutations de S_n qui ne contiennent pas de sous-suite de type τ.

Exemple 1.8 *La permutation* 761254893 *appartient à* $S_9(2413)$ *car aucune de ses sous-suites de longueur* 4 *n'est de type* 2413*, mais n'appartient pas à* $S_9(3142)$ *notamment car la sous-suite* $\pi(2)\pi(4)\pi(7)\pi(9) = 6283$ *est de type* 3142*.*

Définition 1.9 *Une* permutation barrée $\overline{\tau}$ *sur* $[k]$ *est une permutation de* S_k *ayant un élément distingué (nous parlons de permutations p-barrées lorsque p éléments sont distingués).*
Nous notons τ *la permutation sur* $[k]$ *identique à* $\overline{\tau}$ *mais sans distinction d'élément, et* $\widetilde{\tau}$ *la permutation sur* $[k-1]$ *correspondant au type de la sous-suite composée des éléments non distingués de* $\overline{\tau}$*.*
Une permutation π *de* S_n *contient une sous-suite de type* $\overline{\tau}$ *si et seulement si* π *contient une sous-suite de type* $\widetilde{\tau}$ *qui ne fait pas elle-même partie d'une sous-suite de type* τ*.*

Nous notons $S_n(\overline{\tau})$ l'ensemble des permutations de S_n qui ne contiennent pas de sous-suite de type $\overline{\tau}$.

Exemple 1.10 *La permutation* $\pi = 761254893$ *appartient à* $S_9(41\overline{3}52)$ *car toutes les sous-suites de type* 3142 *font partie de sous-suites de type* 41352*. Par exemple, la sous-suite* $\pi(2)\pi(4)\pi(8)\pi(9) = 6293$ *de type* 3142 *fait partie de la sous-suite* $\pi(2)\pi(4)\pi(6)\pi(8)\pi(9) = 62493$ *de type* 41352*.*
Par contre, π *n'appartient pas à* $S_9(21\overline{3}4)$ *notamment car la sous-suite* $\pi(2)\pi(4)\pi(7) = 628$ *de type* 213 *ne fait pas partie d'une sous-suite de type* 2134*.*

Par la suite, nous emploierons le terme *motif* pour désigner une permutation, une permutation barrée ou une permutation p-barrée.

Notation 1.11 *Etant donné un ensemble de motifs* $\{\tau_1, \tau_2, \ldots, \tau_p\}$*,* $S_n(\tau_1, \tau_2, \ldots, \tau_p)$ *désigne l'ensemble des permutations appartenant à* $S_n(\tau_1) \cap S_n(\tau_2) \cap \ldots \cap S_n(\tau_p)$*.*
De même, $I_n(\tau_1, \tau_2, \ldots, \tau_p)$ *et* $\widehat{S}_n(\tau_1, \tau_2, \ldots, \tau_p)$ *désignent respectivement les ensembles des involutions et permutations alternantes excluant simultanément les motifs* $\tau_1, \tau_2, \ldots, \tau_p$*.*

Propriété 1.12 *(J. West [110]) Pour tout ensemble de motifs* $\{\tau_1, \tau_2, \ldots, \tau_p\}$*, nous avons* $\pi \in S_n(\tau_1, \tau_2, \ldots, \tau_p) \iff \pi^* \in S_n(\tau_1{}^*, \tau_2{}^*, \ldots, \tau_p{}^*) \iff \pi^c \in S_n(\tau_1{}^c, \tau_2{}^c, \ldots, \tau_p{}^c) \iff \pi^{-1} \in S_n(\tau_1{}^{-1}, \tau_2{}^{-1}, \ldots, \tau_p{}^{-1})$*.*

Corollaire 1.13 *Pour tout ensemble de motifs* $\{\tau_1, \tau_2, \ldots, \tau_p\}$*, nous avons*

- $\pi \in I_n(\tau_1, \tau_2, \ldots, \tau_p) \iff \pi^{*c} \in I_n(\tau_1{}^{*c}, \tau_2{}^{*c}, \ldots, \tau_p{}^{*c})$
  $\pi \in I_n(\tau_1, \tau_2, \ldots, \tau_p) \iff \pi \in I_n(\tau_1{}^{-1}, \tau_2{}^{-1}, \ldots, \tau_p{}^{-1})$
- $\pi \in \widehat{S}_{2k}(\tau_1, \tau_2, \ldots, \tau_p) \iff \pi^{*c} \in \widehat{S}_{2k}(\tau_1{}^{*c}, \tau_2{}^{*c}, \ldots, \tau_p{}^{*c})$
  $\pi \in \widehat{S}_{2k+1}(\tau_1, \tau_2, \ldots, \tau_p) \iff \pi^* \in \widehat{S}_{2k+1}(\tau_1{}^*, \tau_2{}^*, \ldots, \tau_p{}^*)$

1.3 Rappels sur quelques objets combinatoires classiques

Nous rappelons ici quelques correspondances classiques entre arbres binaires, mots de parenthèses et polyominos parallélogrammes, chacune de ces familles étant énumérée par les nombres de Catalan. Nous terminons ce paragraphe en rappelant une interprétation combinatoire des coefficients binomiaux, des nombres de Motzkin et de Schröder, interprétation que nous utiliserons par la suite.

1.3.1 Arbres binaires, mots de parenthèses et polyominos parallélogrammes

La figure 1.3 illustre les correspondances entre arbres binaires, mots de parenthèses et polyominos parallélogrammes que nous allons présenter maintenant.

Le langage des *mots de parenthèses* (ou systèmes de parenthèses bien formés ou encore mots de Dyck) est le langage $P_{x,\overline{x}} = \{w \in \{x, \overline{x}\}^* : |w|_x = |w|_{\overline{x}}; \forall w = w'w'', |w'|_x \geq |w'|_{\overline{x}}\}$.
Il est fréquent de représenter un mot de parenthèses w de $P_{x,\overline{x}}$ par un chemin de Dyck ω. Le $i^{ème}$ pas du chemin ω est un pas unitaire Nord-Est ou Sud-Est selon que la $i^{ème}$ lettre du mot est respectivement x ou $\overline{x}$.

Nous notons A_n l'ensemble des *arbres binaires complets* ayant $2n+1$ sommets (n sommets internes et $n+1$ feuilles).

Une bijection classique [62] met en correspondance les arbres binaires complets ayant $2n+1$ sommets et les arbres binaires ayant n sommets. Elle consiste à supprimer simultanément toutes les feuilles de l'arbre binaire complet pour obtenir l'arbre binaire. Nous notons $complété(a)$ l'arbre binaire complet obtenu à partir de l'arbre binaire a.

Une bijection entre un arbre binaire complet a de A_n et un mot de parenthèses de $P_{x,\overline{x}}$ de longueur $2n$ est définie par le codage suivant.

$$code(a) = \begin{cases} \varepsilon & \text{si } a \text{ est réduit à un sommet} \\ x\ code(gauche(a))\ \overline{x}\ code(droit(a)) & \text{sinon} \end{cases}$$

Notons $a^\star$ le miroir de l'arbre binaire a obtenu en échangeant les sous-arbres gauche et droit de chaque sommet.

Un *polyomino parallélogramme* [21] est la donnée, sur le réseau carré, de deux chemins disjoints ayant même origine et même extrémité finale, et n'empruntant que des pas unitaires Nord et Est.

M. Delest et X. Viennot [21] ont donné une correspondance entre polyominos parallélogrammes et chemins de Dyck (ou mot de parenthèses). Celle-ci est obtenue en associant à un polyomino parallélogramme deux suites d'entiers, la première correspondant au nombre de cellules de chaque colonne, la seconde indiquant le nombre de cellules en contact pour chaque paire de colonnes consécutives. Ces deux suites d'entiers sont alors les hauteurs des pics (facteurs $x\overline{x}$) et les hauteurs augmentées d'une unité des creux (facteurs $\overline{x}x$) du chemin de Dyck (ou mot de parenthèses) correspondant.

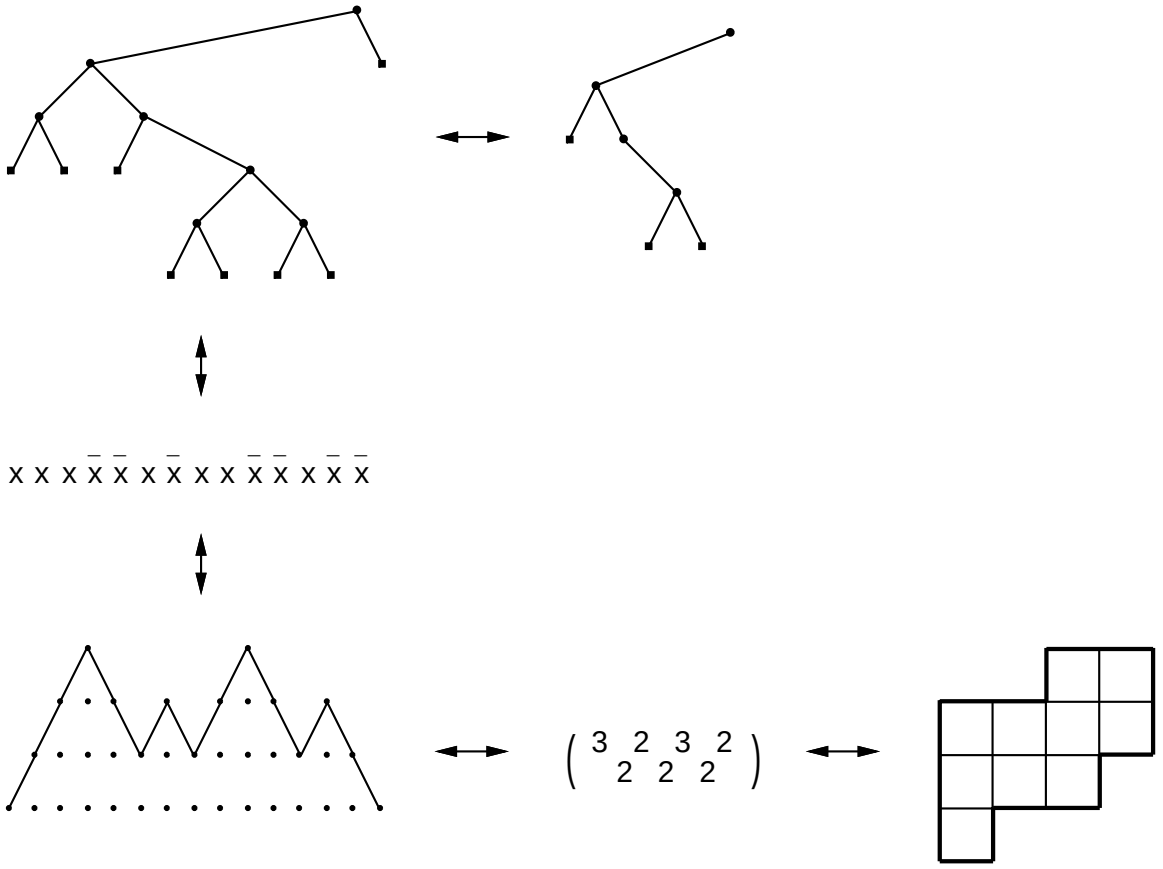


Figure 1.3 Mot de parenthèses en bijection avec un arbre binaire complet, un arbre binaire, un chemin de Dyck, deux suites d'entiers et un polyomino parallélogramme.

1.3.2 Nombres de Catalan

Le nombre de mots de parenthèses de longueur $2n$ est donné par le $n^{ème}$ nombre de Catalan $c_n = \frac{(2n)!}{(n+1)!n!}$.

Nous serons amenés par la suite à considérer certaines distributions classiques sur les nombres de Catalan. Nous les rappelons ici, et donnons leur interprétation sur les mots de parenthèses.

$c_n = \sum_{k=1}^{n} c_{n,k}^{<\alpha>}$ où $c_{n,k}^{<\alpha>} = \binom{2n-k-1}{n-1} - \binom{2n-k-1}{n}$ (nombres de Delannoy [36] ou nombres de scrutins [41] ou distribution α [65]).
Les nombres $c_{n,k}^{<\alpha>}$ énumèrent les mots de parenthèses w de longueur $2n$ de la forme $w = x^k\overline{x}w'$ de même que les mots de parenthèses w de longueur $2n$ comportant k facteurs premiers $w = w_1w_2\dots w_k$ où $w_i \in xP_{x,\overline{x}}\overline{x}$ pour tout $i \in [k]$. Ce résultat est immédiat en considérant le miroir de l'arbre binaire complet codé par un mot de parenthèses.

$c_n = \sum_{k=1}^{n} c_{n,k}^{<\beta>}$ où $c_{n,k}^{<\beta>} = \begin{vmatrix} \binom{n-1}{k-1} & \binom{n-1}{k} \\ \binom{n}{k-1} & \binom{n}{k} \end{vmatrix} = \frac{1}{n}\binom{n}{k}\binom{n}{k-1} = \frac{1}{k}\binom{n-1}{k-1}\binom{n}{k-1}$ (nombres de Narayana [77] ou distribution β [65]).

Les nombres $c_{n,k}^{<\beta>}$ énumèrent les mots de parenthèses w de longueur $2n$ ayant k facteurs $x\overline{x}$, c'est à dire le nombre de chemins de Dyck ayant k pics.

1.3.3 Coefficients binomiaux, nombres de Motzkin et nombres de Schröder

Le langage $GD_{z,\overline{z}} = \{w \in \{z,\overline{z}\}^* : |w|_z = |w|_{\overline{z}}\}$ est parfois appelé langage du Grand Dyck.

Le nombre de mots de $GD_{z,\overline{z}}$ de longueur $2n$ est donné par le $n^{ème}$ coefficient binomial central $\binom{2n}{n}$.

Un *arbre 1-2* est un arbre dessiné (ou ordonné) et enraciné dans lequel chaque sommet possède au plus deux fils.

Le langage $P_{x,\overline{x}} \sqcup\!\sqcup \{y\}^*$ désigne le langage de Motzkin (où $\sqcup\!\sqcup$ est le symbole du produit de mélange).

Nous utiliserons par la suite les deux codages suivants (voir figure 1.4), appelés préfixe et suffixe, d'un arbre 1-2 (ou d'un arbre binaire complet) a possédant n arêtes par un mot de Motzkin de $P_{x,\overline{x}} \sqcup\!\sqcup \{y\}^*$ (ou un mot de parenthèses de $P_{x,\overline{x}}$) de longueur n.

$$préfixe(a) = \begin{cases} \varepsilon \\ y\ préfixe(sous\_arbre\_central(a)) \\ x\ préfixe(sous\_arbre\_gauche(a))\ \overline{x}\ préfixe(sous\_arbre\_droit(a)) \end{cases}$$

suivant que la racine de a soit respectivement une feuille, un point simple, un point double.

$$suffixe(a) = \begin{cases} \varepsilon \\ suffixe(sous\_arbre\_central(a))\ y \\ suffixe(sous\_arbre\_gauche(a))\ x\ suffixe(sous\_arbre\_droit(a))\ \overline{x} \end{cases}$$

suivant que la racine de a soit respectivement une feuille, un point simple, un point double.

Clairement, pour un arbre binaire complet a, nous avons $préfixe(a) = code(a)$.

Remarquons que le codage préfixe du miroir d'un arbre 1-2 correspond au miroir et complémentaire du codage suffixe de l'arbre 1-2. Sur l'exemple illustré par la figure 1.4, nous avons $préfixe(a^\star) = suffixe(a)^{*c} = xxyxx\overline{x}y\overline{x}xx\overline{xxx}y\overline{x}yyxyx\overline{xx}y$.

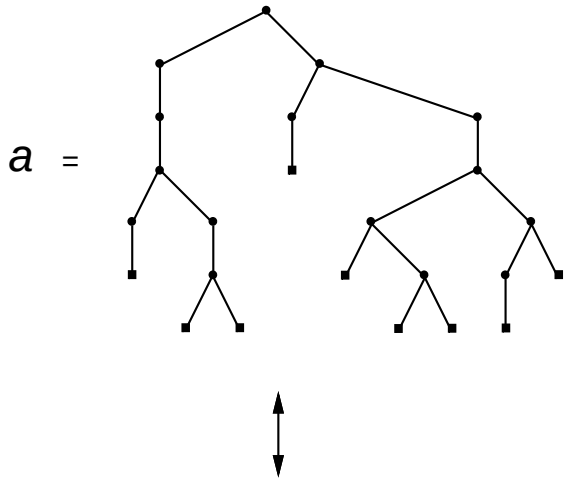
De plus, nous avons la propriété suivante. Soit s un sommet interne d'un arbre binaire complet a de numéro i dans l'ordre infixe, en ne numérotant que les sommets internes. Alors, le $i^{ème}$ $\overline{x}$ [resp. x] du codage préfixe [resp. suffixe] de a correspond à l'arête droite [resp. gauche] de s.

Un *buisson* [23] est un arbre dessiné et enraciné dans lequel aucun sommet, sauf éventuellement la racine, ne possède qu'un seul fils.

Les buissons possédant n arêtes sont en correspondance (voir figure 1.5) avec les mots de longueur $2n$ du langage $P_{x,\overline{x}} \backslash \{\{x,\overline{x}\}^* xxw\overline{xx}\{x,\overline{x}\}^* : w \in P_{x,\overline{x}}\}$. Cette bijection résulte du codage classique d'un arbre par un parcours préfixe où une arête est codée par la lettre x [resp. $\overline{x}$] lors de sa première [resp. seconde] visite.

Les arbres 1-2 et les buissons possédant n arêtes sont énumérés par le $n^{ème}$ nombre de Motzkin $\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$.

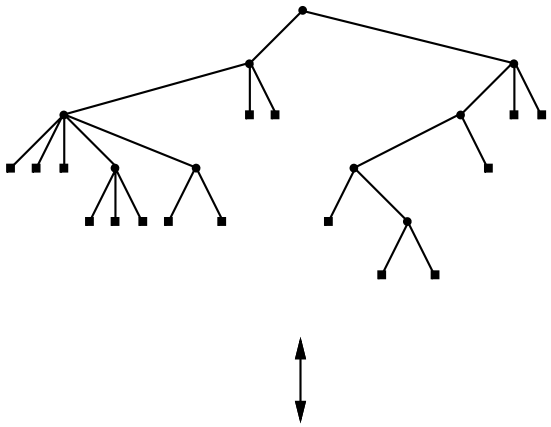
Le nombre d'arbres 1-2 possédant n sommets internes est donné par le $n^{ème}$ nombre de Schröder $\sum_{i=0}^{n} \binom{n+i}{n-i} c_i$.



préfixe(a) = x y y x y $\bar{x}$ y x $\bar{x}\bar{x}$ x y $\bar{x}$ y x x $\bar{x}$ x $\bar{x}\bar{x}$ x y $\bar{x}$

suffixe(a) = y x x $\bar{x}$ y $\bar{x}$ y y x y x x x $\bar{x}\bar{x}$ x y x $\bar{x}\bar{x}$ y $\bar{x}\bar{x}$

Figure 1.4 Codages préfixe et suffixe d'un arbre 1-2.



x x x $\bar{x}$ x $\bar{x}$ x $\bar{x}$ x x $\bar{x}$ x x $\bar{x}$ x x x $\bar{x}$ x x $\bar{x}\bar{x}\bar{x}$ x x $\bar{x}\bar{x}$ x x x x x $\bar{x}$ x x $\bar{x}$ x x $\bar{x}\bar{x}$ x x $\bar{x}$ x x $\bar{x}\bar{x}$

Figure 1.5 Un buisson.

Chapitre 2

Arbre de génération d'une famille d'objets combinatoires

En Combinatoire, lorsque nous considérons une famille d'objets combinatoires, il est naturel de s'intéresser à la façon dont peuvent croître ces objets dans le but de les caractériser et de les énumérer. Cette approche se retrouve en particulier dans le cadre de la théorie des espèces de structures [60, 7] et des grammaires d'objets [32, 33].

Nous abordons ici cette problématique en considérant la notion d'arbre de génération d'une famille d'objets combinatoires. Un tel arbre peut être défini lorsque chaque objet de la famille est obtenu de manière unique à partir d'un autre objet plus petit (de la même famille) par une certaine règle de croissance, et lorsqu'il existe un unique objet de taille minimale. Ainsi, à chaque sommet de l'arbre correspond un objet de la famille.

A travers un exemple simple dans ce chapitre, et d'autres plus complexes dans les suivants, nous montrons qu'à de nombreux objets combinatoires peuvent être associés un arbre de génération. Pour chacun d'eux, la règle de croissance des objets permet de caractériser l'arbre de génération par un système de réécriture duquel il est possible de déduire des équations de récurrence. Par la suite, nous utiliserons à de nombreuses reprises le fait que, lorsque deux arbres de génération sont caractérisés par le même système de réécriture, ils sont isomorphes et induisent une bijection entre les deux familles d'objets combinatoires sous-jacentes.

Afin d'illustrer cette présentation de la méthode des arbres de génération, nous utilisons les mots de parenthèses comme exemple de référence.

2.1 Arbre de génération

Soit $E = \cup_{n\geq 0} E_n$ un ensemble d'objets combinatoires où E_n désigne l'ensemble des objets de taille n. Supposons que E_0 soit réduit à un seul objet, le générateur.
L'*arbre de génération* de l'ensemble E est un arbre pour lequel

- chaque objet de E_n apparaît une fois et une seule sur un sommet de niveau n,
- les arêtes de l'arbre reliant les sommets au niveau n à ceux au niveau $n+1$ correspondent aux règles de génération permettant de construire les objets de E_{n+1} à partir des objets de E_n.

Exemple 2.1 *Un mot de parenthèses w non vide se factorise de manière unique sous la forme $w = w_1 w_2 \dots w_p$ où les w_i sont des mots de parenthèses non vides et p est maximal (les w_i sont alors des mots de parenthèses premiers). Un tel mot w permet d'engendrer $p+1$ mots de parenthèses, à savoir les mots $x w_1 w_2 \dots w_{i-1} \overline{x} w_i w_{i+1} \dots w_p$ pour tout $i \in [p+1]$. Partant du mot vide, nous obtenons ainsi tous les mots de parenthèses, ce qui conduit à l'arbre de génération des mots de parenthèses donné par la figure 2.1.*
Notons que ces règles de génération correspondent exactement aux opérateurs d et D introduits par R. Cori [16].

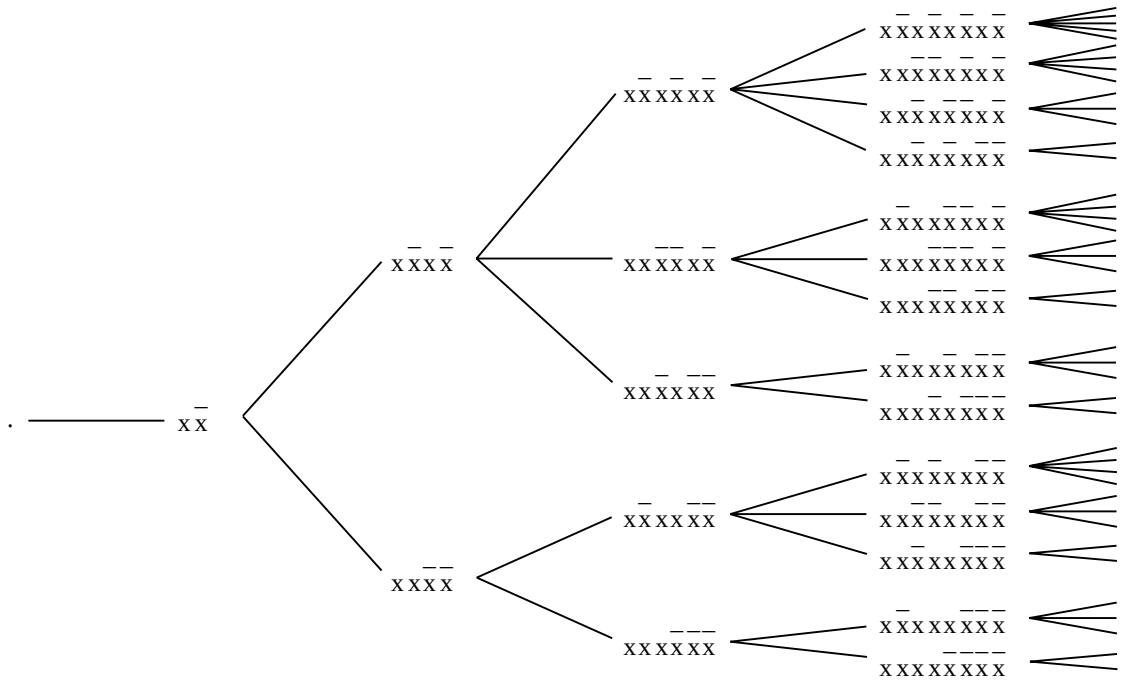


Figure 2.1 Arbre de génération des mots de parenthèses.

Remarque 2.2

(i) Il est possible d'étendre la définition d'arbre de génération à la notion de forêt d'arbres de génération. Ceci correspondrait au cas où l'ensemble considéré possèderait plusieurs générateurs (c'est à dire plusieurs objets minimaux).

(ii) Il serait également envisageable d'autoriser des arêtes reliant deux sommets situés aux niveaux n et $n+k$ (avec $k>1$) dans le cas où les règles de génération des objets auraient pour effet de les faire croître de k unités.

2.2 De l'arbre de génération au système de réécriture

Un arbre de génération d'un ensemble d'objets combinatoires peut être caractérisé par un *système de réécriture*. Celui-ci est obtenu en associant à chaque objet (sommet de l'arbre) une étiquette et en déduisant des règles de génération des objets un ensemble de règles de réécriture pour ces étiquettes.
Le système de réécriture caractérisant l'arbre de génération considéré se compose alors d'un axiome (étiquette initiale correspondant à l'objet générateur) et d'un ensemble de règles de réécriture. Nous le représentons ainsi.

$$\begin{cases} étiquette\_initiale \\ étiquette_1 \rightsquigarrow étiquette_1\_fils_1, étiquette_1\_fils_2, \ldots, étiquette_1\_fils_{t_1} \\ étiquette_2 \rightsquigarrow étiquette_2\_fils_1, étiquette_2\_fils_2, \ldots, étiquette_2\_fils_{t_2} \\ \qquad\qquad \vdots \\ étiquette_r \rightsquigarrow étiquette_r\_fils_1, étiquette_r\_fils_2, \ldots, étiquette_r\_fils_{t_r} \end{cases}$$

où $étiquette_i\_fils_1, étiquette_i\_fils_2, \ldots, étiquette_i\_fils_{t_i}$ sont les t_i étiquettes obtenues à partir de l'étiquette $étiquette_i$ pour tout $i \in [r]$.

Exemple 2.3 *Le système de réécriture*

$$\begin{cases} (0) \\ (p) \rightsquigarrow (p+1),(p),\ldots,(1) \end{cases}$$

dont l'arbre de dérivation est donné figure 2.2 caractérise l'arbre de génération des mots de parenthèses donné dans l'exemple 2.1. En effet, pour cette famille d'objets, l'étiquette d'un mot de parenthèses correspond à la taille de sa factorisation en mots premiers.

Un arbre de génération d'un ensemble d'objets combinatoires peut être caractérisé par un système de réécriture dans lequel certains termes des étiquettes correspondent à la valeur de certains paramètres associés à ces objets. Ces termes, qui ne sont pas forcément tous nécessaires à la caractérisation de l'arbre, peuvent être toutefois utiles pour considérer une distribution particulière des objets.

Exemple 2.4 *Le système de réécriture*

$$\begin{cases} (0,-1) \\ (p,c) \rightsquigarrow (p+1,c+1),(p,c),(p-1,c),\ldots,(1,c) \end{cases}$$

caractérise l'arbre de génération des mots de parenthèses. Il est tel qu'à chaque mot de parenthèses w correspond une étiquette (p,c) où p est le nombre de facteurs premiers de w et c est le nombre de facteurs $\overline{x}x$ de w (nombre de creux du chemin de Dyck correspondant). Pour cela, il est nécessaire d'associer l'étiquette $(0,-1)$ au mot vide.

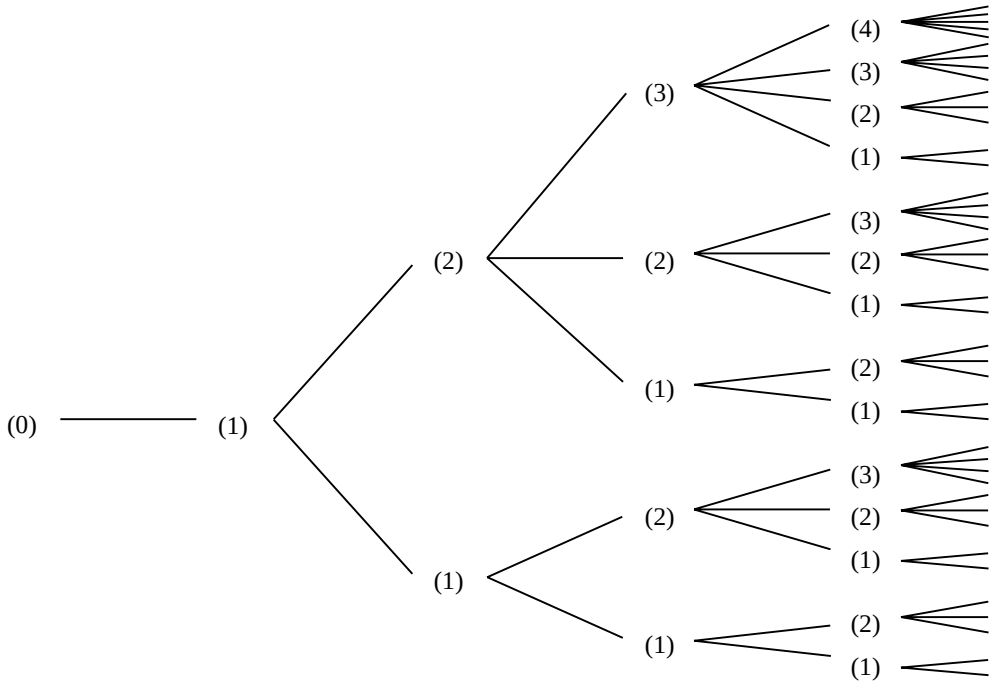


Figure 2.2 Arbre de dérivation du système de réécriture caractérisant l'arbre de génération des mots de parenthèses.

Remarque 2.5 *Deux arbres de génération caractérisés par le même système de réécriture sont isomorphes (symbolisé par $\cong$). Cet isomorphisme induit une bijection entre les deux ensembles d'objets correspondants qui transporte tous les paramètres associés aux étiquettes du système de réécriture.*

Exemple 2.6 *Considérons un nouvel arbre de génération des mots de parenthèses basé sur la construction suivante. Partant d'un mot de parenthèses w codant un chemin de Dyck de hauteur initiale p, $w = x^p\overline{x}w'$, nous engendrons $p+1$ mots de parenthèses, à savoir les mots $x^i\overline{x}x^{p+1-i}\overline{x}w'$ pour tout $i \in [p+1]$. Clairement, nous obtenons ainsi, en itérant cette construction, chaque mot de parenthèses une fois et une seule. L'arbre de génération correspondant est caractérisé par le système de réécriture*

$$\begin{cases} (0) \\ (p) \quad \rightsquigarrow \quad (1),(2),\ldots,(p+1) \end{cases}$$

identique à celui de l'exemple 2.3.

Nous en déduisons que la distribution des mots de parenthèses suivant le nombre de facteurs premiers est identique à la distribution de ces mêmes mots suivant leur hauteur initiale ; il s'agit des nombres de Delannoy.

2.3 Du système de réécriture aux récurrences

A partir d'un système de réécriture caractérisant un arbre de génération d'objets combinatoires, il est toujours possible d'établir des équations de récurrence pour le nombre de ces objets en fonction de leur taille et des paramètres de l'étiquette.

Exemple 2.7 *Considérons le système de réécriture caractérisant l'arbre de génération des mots de parenthèses donné dans l'exemple 2.3*

$$\begin{cases} (0) \\ (p) \quad \rightsquigarrow \quad (p+1),(p),\ldots,(1) \end{cases}$$

Soit $c_{n,(p)}$ le nombre d'étiquettes (p) obtenues au niveau n de l'arbre de génération et c_n le nombre total d'étiquettes à ce même niveau. Ainsi, d'après ce que nous avons vu précédemment, $c_{n,(p)}$ est le nombre de mots de parenthèses de longueur $2n$ ayant exactement p facteurs premiers.

Nous déduisons du système de réécriture les récurrences suivantes.

$$\begin{cases} c_{0,(0)} & = & 1 & \\ c_{1,(1)} & = & 1 & \\ c_{n,(1)} & = & \sum_{k=1}^{n-1} c_{n-1,(k)} & \text{pour tout } n > 1 \\ c_{n,(p)} & = & \sum_{k=p-1}^{n-1} c_{n-1,(k)} & \text{pour tout } n > 1 \text{ et pour tout } p \in [2,n] \\ c_n & = & \sum_{p=1}^{n} c_{n,(p)} & \text{pour tout } n \geq 1 \end{cases}$$

Nous pouvons alors vérifier que $c_{n,(p)} = \binom{2n-p-1}{n-1} - \binom{2n-p-1}{n}$ pour tout $1 \leq p \leq n$ et que $c_n = \frac{(2n)!}{(n+1)!n!}$ pour tout $n \geq 0$.

2.4 Arbres de génération et génération aléatoire

La méthode des arbres de génération est directement utilisable pour la génération aléatoire d'objets combinatoires.

En effet, tirer de façon uniforme et équiprobable l'un des éléments de taille n de l'ensemble étudié correspond à choisir aléatoirement un chemin partant de la racine et aboutissant à l'un des sommets du niveau n de l'arbre de génération. Pour cela, il est nécessaire de connaître, pour chaque sommet de l'arbre de génération, son nombre de descendants au niveau n. Ainsi, pour un sommet donné, il sera possible de choisir équiprobablement l'un de ses fils. Ce calcul est donc très similaire de celui à réaliser pour résoudre les récurrences du système de réécriture caractérisant l'arbre de génération : les règles de réécriture sont identiques et l'étiquette d'initialisation à prendre en compte est celle du sommet considéré.

Nous pouvons alors en déduire un algorithme en temps linéaire qui tire uniformément et équiprobablement l'un des éléments de l'ensemble étudié. Toutefois, cet algorithme manipule de grands nombres, et nous nous retrouvons confrontés aux mêmes problèmes que ceux présents dans la méthode de T. Hickey et J. Cohen [57] pour la génération aléatoire de mots d'un langage engendré par une grammaire algébrique.

Exemple 2.8 *Afin de générer aléatoirement un mot de parenthèses, de façon uniforme et équiprobable, il est nécessaire de calculer* $c_n^{r,(e)}$ *le nombre de sommets au niveau* n *issus d'un sommet d'étiquette* (e) *du niveau* r*. Pour cela, nous devons résoudre les récurrences du système de réécriture*

$$\begin{cases} (e) \\ (p) \quad \rightsquigarrow \quad (1),(2),\ldots,(p+1) \end{cases}$$

qui correspond au système de réécriture des mots de parenthèses de l'exemple 2.6 (seule l'étiquette d'initialisation est modifiée). Nous pouvons vérifier que $c_n^{r,(e)} = \binom{2n-2r+e}{n-r} - \binom{2n-2r+e}{n-r-1}$ *pour tout* $1 \leq e \leq r \leq n$ *et plus précisément que le nombre de sommets ayant pour étiquette* (p)*, pour tout* $p \in [n]$*, est* $c_{n,(p)}^{r,(e)} = \binom{2n-2r+e-p-1}{n-r-1} - \binom{2n-2r+e-p-1}{n-r+e}$*. Nous en déduisons l'algorithme de génération aléatoire d'un mot de parenthèses illustré par la figure 2.3.*

<u>Entrée</u> : $n \geq 1$
<u>Sortie</u> : un mot de parenthèses w de longueur $2n$

$w' \longleftarrow \varepsilon$
$e \longleftarrow 1$
pour r variant de 1 à $n-1$
    choisir le fils d'étiquette (p) $(p \in [e+1])$ avec la probalité $\frac{c_n^{r+1,(p)}}{c_n^{r,(e)}}$
    $w' \longleftarrow x^{e+1-p}\overline{x}w'$
    $e \longleftarrow p$
retourner $x^e\overline{x}w'$

Figure 2.3 Algorithme de génération aléatoire d'un mot de parenthèses.

Chapitre 3

Arbres de génération de permutations : le logiciel forbid

Dans ce chapitre, nous nous intéressons aux arbres de génération des permutations et de deux familles particulières : les involutions et les permutations alternantes. Nous montrons comment ces arbres peuvent être construits et caractérisons chacun d'eux par un système de réécriture. De manière plus générale, nous considérons les arbres de génération de ces familles de permutations lorsque certains motifs sont interdits.

Ensuite, nous présentons un logiciel, baptisé *forbid* et développé en langage C, pour l'obtention de n'importe quelle famille de permutations (permutations, involutions, permutations alternantes) excluant un ou plusieurs motifs. Nous décrivons les différents états de sortie que permet ce logiciel : arbre de génération, ensemble des permutations, distributions de ces permutations suivant la plupart des paramètres classiques sur les permutations, ...

3.1 Arbres de génération de permutations à motifs exclus

3.1.1 Arbre de génération des permutations

Définition 3.1 *(J. West [110])* L'arbre de génération des permutations *excluant l'ensemble de motifs* $\{\tau_1, \tau_2, \ldots, \tau_p\}$ *vérifie*

- *sa racine est constituée par la permutation* 1 *de* S_1,
- *les fils d'une permutation* π *de* $S_n(\tau_1, \tau_2, \ldots, \tau_p)$ *sont toutes les permutations appartenant à* $S_{n+1}(\tau_1, \tau_2, \ldots, \tau_p)$ *obtenues en insérant l'élément* $n+1$ *dans* π, *c'est à dire les permutations* $(n+1)\pi(1)\pi(2)\ldots\pi(n)$, $\pi(1)(n+1)\pi(2)\ldots\pi(n)$, ..., $\pi(1)\pi(2)\ldots\pi(n)(n+1)$.

L'arbre de génération des permutations excluant les motifs de $\{\tau_1, \tau_2, \ldots, \tau_p\}$ est noté $T(\tau_1, \tau_2, \ldots, \tau_p)$.

Remarque 3.2 *D'autres constructions pour l'arbre de génération des permutations sont également possibles comme l'a proposé S. Gire [45]. Par exemple, il est possible d'insérer l'élément* 1 *dans la permutation* π *dont tous les éléments ont préalablement été incrémentés d'une unité, ou bien d'insérer les éléments* e *appartenant à* $[n+1]$ *devant la permutation* π *dont tous les éléments supérieurs ou égaux à* e *ont préalablement été incrémentés d'une unité, ...*

L'arbre de génération de toutes les permutations (voir figure 3.1) est caractérisé par le système de réécriture

$$\begin{cases} (2) \\ (t) \rightsquigarrow \underbrace{(t+1),(t+1),\ldots,(t+1)}_{t \text{ fois}} \end{cases}$$

Ce système de réécriture vérifie qu'à chaque permutation de S_n correspond l'étiquette $(n+1)$.

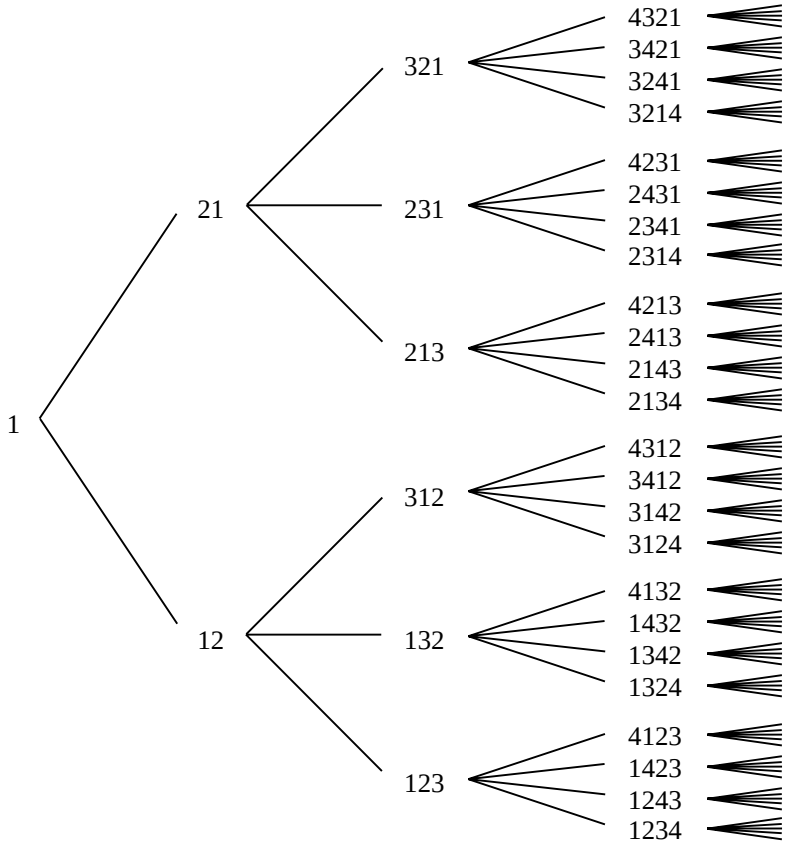


Figure 3.1 Arbre de génération de toutes les permutations.

Pour un ensemble de motifs exclus, l'arbre de génération $T(\tau_1, \tau_2, \ldots, \tau_p)$ est donc un sous-arbre de l'arbre de génération de toutes les permutations. Par exemple, la figure 3.2 représente l'arbre de génération des permutations $T(312)$ jusqu'au niveau 4.

Définition 3.3 *(J. West [110], O. Guibert [53], S. Gire [45]) Etant donnée une permutation* π *de* $S_n(\tau_1, \tau_2, \ldots, \tau_p)$*, un* site *est soit une position entre deux éléments consécutifs* $\pi(i)$ *et* $\pi(i+1)$

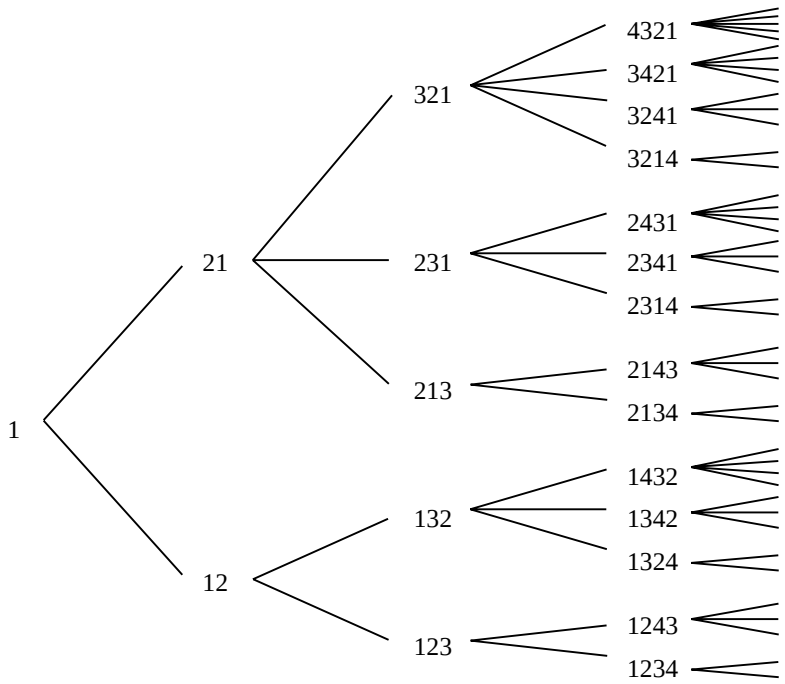


Figure 3.2 $T(312)$, l'arbre de génération des permutations excluant le motif 312.

pour tout $i \in [n-1]$, *soit la position à gauche de* $\pi(1)$, *soit la position à droite de* $\pi(n)$.
Un site est dit actif *si l'insertion de l'élément* $n+1$ *dans cette position de la permutation* π *donne une permutation appartenant à* $S_{n+1}(\tau_1, \tau_2, \ldots, \tau_p)$ *; le site est dit* inactif *dans le cas contraire. Un site est dit* temporairement inactif *s'il est inactif pour l'insertion de l'élément* $n+1$ *dans cette position de la permutation* π *et qu'existe* $k > 1$ *tel que l'insertion de l'élément* $n+k$ *dans cette position de la permutation* π' *obtenue en insérant les éléments de* $[n+1, n+k-1]$ *dans* π *rend ce site actif; le site inactif est dit* définitivement inactif *dans le cas contraire.*

Nous représentons par les symboles $\diamond$, . *et* $\bullet$ *les sites respectivement actif, temporairement inactif et définitivement inactif.*

Exemple 3.4 *La permutation* $\diamond 2.1 \diamond 5.4 \diamond 6 \diamond 3 \bullet$ *de* $S_6(2314, \overline{4}2513)$ *possède* 4 *sites actifs car* $\{7215463, 2175463, 2154763, 2154673\} \subset S_7(2314, \overline{4}2513)$, 2 *sites temporairement inactifs car* $2715463 \notin S_7(2314, \overline{4}2513)$ *mais* $72815463 \in S_8(2314, \overline{4}2513)$ *et* $2157463 \notin S_7(2314, \overline{4}2513)$ *mais* $72158463 \in S_8(2314, \overline{4}2513)$, *et* 1 *site définitivement inactif car pour tout* $e > 6$ *la sous-suite* $563e$ *est de type* 2314.

La propriété suivante nous sera particulièrement utile pour établir certains résultats des chapitres suivants.

Propriété 3.5 *(O. Guibert [53], S. Gire [45]) Soient* τ *et* $\overline{\beta}$ *deux permutations respectivement non barrée et barrée sur* $[k]$, β *et* $\widetilde{\beta}$ *étant obtenus à partir de* $\overline{\beta}$ *conformément à la définition 1.9. Alors, nous avons les propriétés suivantes.*

(i) *Les $\tau^{-1}(k)-1$ premiers sites et les $k-\tau^{-1}(k)$ derniers sites de toute permutation π appartenant à $S_n(\tau)$ sont actifs.*

(ii) *Les $\widetilde{\beta}^{-1}(k-1)-1$ premiers sites et les $k-1-\widetilde{\beta}^{-1}(k-1)$ derniers sites de toute permutation π appartenant à $S_n(\overline{\beta})$ sont actifs.*

(iii) *Etant donnée une permutation π appartenant à $S_n(\tau)$, un site inactif de π reste définitivement inactif.*

(iv) *Etant donnée une permutation π appartenant à $S_n(\overline{\beta})$, un site inactif de π reste définitivement inactif si l'élément distingué est $k-1$ et si les éléments k et $k-1$ sont consécutifs dans un ordre quelconque dans β, ou bien si l'élément distingué est différent de $k-1$.*

(v) *Soient π une permutation appartenant à $S_n(\tau)$ et π' une permutation obtenue en insérant l'élément $n+1$ dans π. Alors π' n'appartient pas à $S_{n+1}(\tau)$ si et seulement si elle admet une sous-suite $\pi'(l_1)\pi'(l_2)\ldots\pi'(l_k)$ de type τ telle qu'il existe i et j appartenant à $[k]$ vérifiant $\pi'(l_i)=n+1$, $\tau(i)=k$ et $\pi'(l_j)=n$, $\tau(j)=k-1$.*

Remarque 3.6 *Soit $\overline{\beta}$ une permutation barrée, β et $\widetilde{\beta}$ étant obtenus à partir de $\overline{\beta}$ conformément à la définition 1.9. Soit x l'élément distingué de $\overline{\beta}$ et posons $i=\beta^{-1}(x)$.*
Si $\beta(i-1)=x\pm1$ ou $\beta(i+1)=x\pm1$, alors $S_n(\overline{\beta})=S_n(\widetilde{\beta})$.

Preuve Par définition, si la permutation π appartient à $S_n(\widetilde{\beta})$, π appartient également à $S_n(\overline{\beta})$. Supposons maintenant que π n'appartienne pas à $S_n(\widetilde{\beta})$, c'est à dire que π possède une sous-suite $\widetilde{\sigma}$ de type $\widetilde{\beta}$. Posons $k=|\beta|$. Si $\widetilde{\sigma}$ ne fait pas elle-même partie d'une sous-suite de type β, π n'appartient pas à $S_n(\overline{\beta})$. Sinon, parmi toutes les sous-suites $\sigma' y \sigma''$ de type β avec $\widetilde{\sigma}=\sigma'\sigma''$ et $|\sigma'|=i-1$, choisissons celle dont l'élément y est situé le plus à droite [resp. gauche] possible dans π si $\beta(i-1)$ vaut $x+1$ ou $x-1$ [resp. sinon], de sorte que la sous-suite $\sigma'(1)\sigma'(2)\ldots\sigma'(i-2)y\sigma''$ [resp. $\sigma' y\sigma''(2)\sigma''(3)\ldots\sigma''(k-i)$] est de type $\widetilde{\sigma}$ mais ne fait pas elle-même partie d'une sous-suite de type β : π n'appartient donc pas à $S_n(\overline{\beta})$. □

Il est naturel de se demander si l'arbre de génération d'un ensemble de permutations à motifs exclus peut être fini. Ceci ne peut se produire que dans un seul cas.

Propriété 3.7 *Soit $\{\tau_1,\tau_2,\ldots,\tau_p\}$ un ensemble de permutations non barrées. Alors, $S_n(\tau_1,\tau_2,\ldots,\tau_p)$ est vide à partir d'un certain rang n si et seulement s'il existe $l,m\geq 0$ tels que $\{12\ldots(l+1),(m+1)m\ldots1\}\subseteq\{\tau_1,\tau_2,\ldots,\tau_p\}$.*

Preuve Seul le motif $12\ldots r$ [resp. $s(s-1)\ldots1$] permet d'exclure toutes les permutations $12\ldots n$ [resp. $n(n-1)\ldots1$] pour tout $n\geq r$ [resp. s]. Or, P. Erdös et G. Szekeres [34] ont montré que $S_n(12\ldots(l+1),(m+1)m\ldots1)$ est vide pour tout $n>l.m$. □

Ce résultat se généralise aux permutations p-barrées. En effet, exclure le motif identité [resp. miroir de l'identité] sur $[k]$ avec p éléments distingués revient à exclure le motif identité [resp. miroir de l'identité] sur $[k-p]$. Par exemple, $S_n(1\overline{23}\overline{45}67\overline{89},43\overline{2}1)=S_n(1234,321)$.

3.1.2 Arbre de génération des involutions

Définition 3.8 *L'*arbre de génération des involutions *excluant l'ensemble de motifs* $\{\tau_1, \tau_2, \ldots, \tau_p\}$ *obtenu par la* méthode dite des points fixes *vérifie*

- *sa racine est constituée par l'involution* 1 *de* I_1,
- *les fils de l'involution* π *de* $I_n(\tau_1, \tau_2, \ldots, \tau_p)$ *sont les involutions appartenant à* $I_{n+1}(\tau_1, \tau_2, \ldots, \tau_p)$ *suivantes : involutions* $\pi(1)\pi(2)\ldots\pi(i-1)(n+1)\pi(i+1)\pi(i+2)\ldots\pi(n)i$, *pour chaque point fixe* i *de* π, *et l'involution* $\pi(1)\pi(2)\ldots\pi(n)(n+1)$.

L'arbre de génération de toutes les involutions par la méthode des points fixes (voir figure 3.3) est caractérisé par le système de réécriture

$$\begin{cases} (1) & \\ (p) & \rightsquigarrow \underbrace{(p-1),(p-1),\ldots,(p-1)}_{p \text{ fois}},(p+1) \end{cases}$$

Ce système de réécriture est tel qu'à chaque involution correspond une étiquette (p) où p est le nombre de points fixes de cette involution.

A partir de ce système de réécriture, nous obtenons l'équation de récurrence suivante.

$$\begin{cases} |I_{1,1}| & = & 1 \\ |I_{n,p}| & = & |I_{n-1,p-1}| + (p+1).|I_{n-1,p+1}| \end{cases}$$ pour tout $0 \leq p \leq n$, p et n ayant même parité

où $I_{n,p}$ est l'ensemble des involutions sur $[n]$ ayant p points fixes.

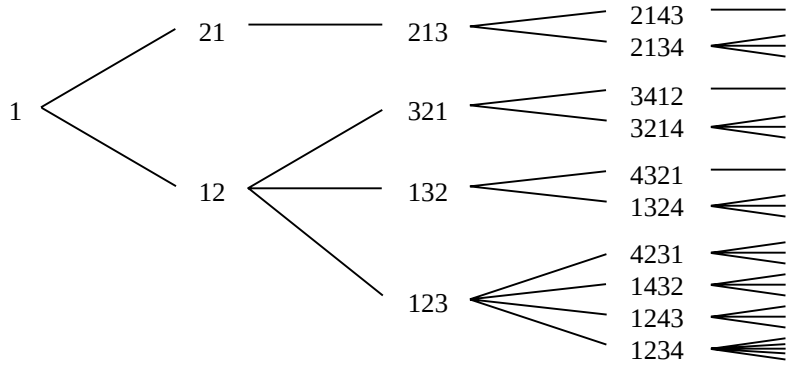


Figure 3.3 Arbre de génération de toutes les involutions par la méthode des points fixes.

Une seconde construction permet d'obtenir un autre arbre de génération des involutions. Cette construction est basée sur la formule de récurrence classique pour les involutions. Notons que J.S. Beissinger [5] donne une construction équivalente sur les tableaux de Young standard.

Définition 3.9 *L'*arbre de génération des involutions *excluant l'ensemble de motifs* $\{\tau_1, \tau_2, \ldots, \tau_p\}$ *obtenu par la* méthode dite récurrente *vérifie*

- *sa racine est constituée par l'involution* ε *de* I_0,

- *les fils de l'involution* π *de* $I_n(\tau_1, \tau_2, \ldots, \tau_p)$ *sont l'involution* $\pi(1)\pi(2)\ldots\pi(n)(n+1)$ *lorsqu'elle appartient à* $I_{n+1}(\tau_1, \tau_2, \ldots, \tau_p)$ *et les involutions* $\pi^+(1)\pi^+(2)\ldots\pi^+(i-1)(n+2)$ $\pi^+(i)\pi^+(i+1)\ldots\pi^+(n)i$ *pour tout* $i \in [n+1]$ *si elles appartiennent à* $I_{n+2}(\tau_1, \tau_2, \ldots, \tau_p)$, *où* $\pi^+(j) = \pi(j)$ *[resp.* $\pi(j)+1$*] si* $\pi(j) < i$ *[resp.* $\geq i$*].*

L'arbre de génération de toutes les involutions par la méthode récurrente (voir figure 3.4) est caractérisé par le système de réécriture

$$\begin{cases} (0) & & \\ (n) & \rightsquigarrow & (n+1) \\ & \overset{2}{\rightsquigarrow} & \underbrace{(n+2), (n+2), \ldots, (n+2)}_{n+1 \text{ fois}} \end{cases}$$

Ce système de réécriture est tel qu'à chaque involution de I_n correspond l'étiquette (n) .
A partir de ce système de réécriture, nous obtenons l'équation de récurrence suivante.

$$\begin{cases} |I_0| = |I_1| = 1 \\ |I_n| = |I_{n-1}| + (n-1).|I_{n-2}| \quad \text{pour tout } n \geq 2 \end{cases}$$

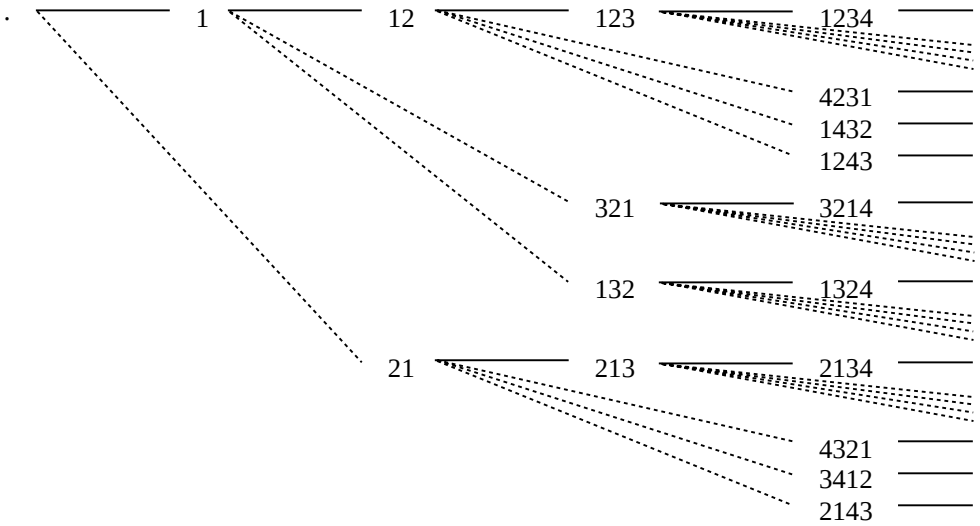


Figure 3.4 Arbre de génération de toutes les involutions par la méthode récurrente.

La propriété suivante nous sera utile dans certaines démonstrations ultérieures.

Propriété 3.10 *Soit* τ *une permutation non barrée sur* $[k]$. *Nous avons les propriétés suivantes.*

(i) Si $\pi \in I_n(\tau)$ *et* $\tau(k) \neq k$, *alors* $\pi(1)\pi(2)\ldots\pi(n)(n+1) \in I_{n+1}(\tau)$.

(ii) Un site inactif qui ne permet pas d'engendrer une involution sur $[n+2]$ *à partir d'une involution de* $I_n(\tau)$ *par la méthode récurrente reste définitivement inactif.*

Preuve

- (i). En effet, l'élément $n+1$ qui est inséré ne peut pas correspondre à l'élément k du motif exclu τ.

- (ii). L'insertion ne change pas l'ordre des éléments précédents.

□

3.1.3 Arbre de génération des permutations alternantes

Définition 3.11 *L'*arbre de génération des permutations alternantes *excluant l'ensemble de motifs* $\{\tau_1, \tau_2, \ldots, \tau_p\}$ *vérifie*

- *sa racine est constituée par la permutation alternante* ε *de* $\widehat{S}_0$,

- *les fils d'une permutation alternante* π *de* $\widehat{S}_n(\tau_1, \tau_2, \ldots, \tau_p)$ *sont les permutations alternantes appartenant à* $\widehat{S}_{n+1}(\tau_1, \tau_2, \ldots, \tau_p)$ *suivantes : permutations alternantes* $\pi^+(1)\pi^+(2)\ldots\pi^+(n)i$ *pour tout* $i \in \{1, 2, \ldots, \pi(n)\}$ *[resp.* $\{\pi(n)+1, \pi(n)+2, \ldots, n+1\}$*] si* n *est pair [resp. impair], où* $\pi^+(j) = \pi(j)$ *[resp.* $\pi(j)+1$*] si* $\pi(j) < i$ *[resp.* $\geq i$*].*

L'arbre de génération de toutes les permutations alternantes (voir figure 3.5) est caractérisé par le système de réécriture

$$\begin{cases} (1,0) \\ (x,y) \rightsquigarrow (y+i, x+1-i) \text{ pour tout } i \in [x] \end{cases}$$

Ce système de réécriture est tel qu'à chaque permutation alternante π de $\widehat{S}_n$ correspond une étiquette (x, y) vérifiant $x + y = n + 1$ et x [resp. y] $= \pi(n)$ si n est pair [resp. impair].

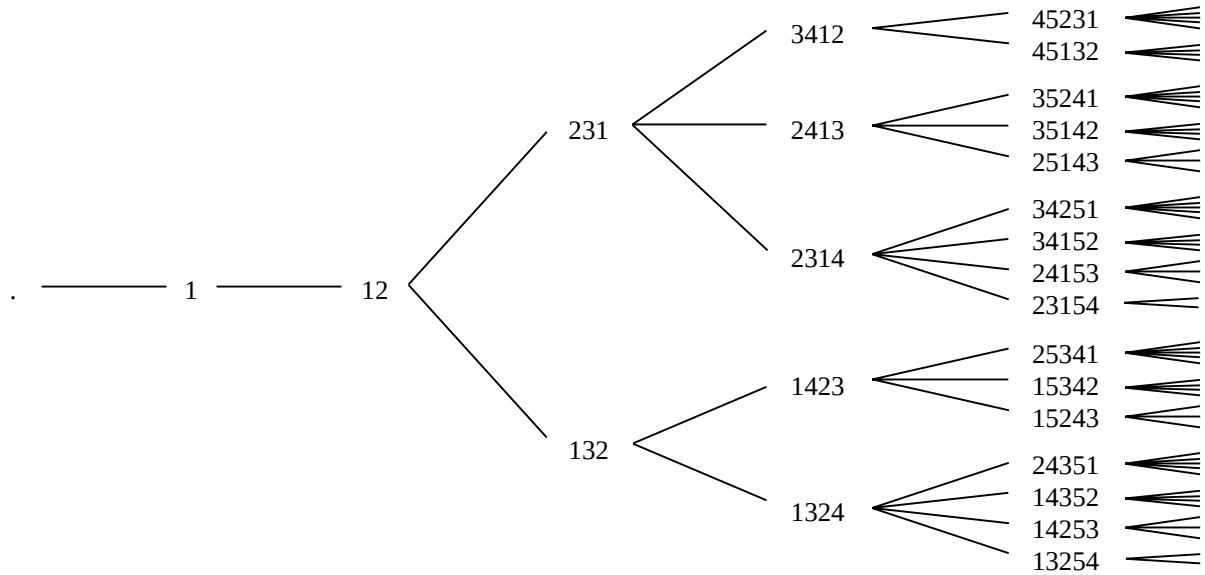


Figure 3.5 Arbre de génération de toutes les permutations alternantes.

3.2 Le logiciel *forbid*

Nous nous limitons ici à l'aspect utilisation du logiciel *forbid* puisque les principaux algorithmes, comme par exemple l'implémentation de certaines propriétés pour l'amélioration des performances, ont déjà été exposés dans [53].

forbid est un logiciel d'aide à la recherche sur les permutations à motifs exclus pour lequel de nombreuses options sont possibles. Nous présentons ici celles que nous pensons être les plus utiles.

forbid construit, à partir d'une liste de motifs exclus fournie par l'utilisateur, l'arbre de génération (jusqu'à un certain niveau) de la famille désirée. Cette famille peut être celle des permutations, celle des involutions (les deux méthodes de construction de l'arbre de génération des involutions sont possibles) ou celle des permutations alternantes.
Différents états de sortie, au choix de l'utilisateur, peuvent être obtenus. Un premier groupe permet d'obtenir l'arbre de génération sous diverses formes : arbre ASCII, fichier destiné au logiciel de visualisation de graphes *CABRI*, règles de réécriture (pour éventuellement deviner le système de réécriture), ...Un deuxième groupe fournit l'ensemble des permutations à motifs exclus et le couple de tableaux de Young standard correspondant par l'algorithme de Robinson-Schensted. Un troisième groupe procure les distributions de cet ensemble de permutations à motifs exclus suivant de nombreux paramètres (minima ou maxima à gauche ou à droite, montées, excédences, indices des éléments extrêmes, cycles, points fixes, ...).

Nous adoptons les notations syntaxiques "Backus Normal Form" pour présenter *forbid*.

$\lhd type \rhd$	désigne une expression du type indiqué
$\lhd type \rhd *$	désigne une suite de $\lhd type \rhd$ de longueur quelconque
$\lhd type \rhd +$	désigne une suite de $\lhd type \rhd$ de longueur non nulle
::=	indique que les expressions gauche et droite sont équivalentes
`Mot`	représente la chaîne de caractères `Mot`
[ ]	signale que ce qui est entre crochets est optionnel
{}	exprime un choix unique parmi les objets (séparés par des virgules) entre accolades

La syntaxe de la commande `forbid` est la suivante.

`forbid` $\lhd familles \rhd$ $\lhd sorties \rhd$ [+]$\lhd niveau\_max \rhd$ $\lhd motif\_exclu \rhd *$ [−`r`[+] $\lhd perm\_racine \rhd$ $*$] [− `f` [$\lhd fichier \rhd$]]

- `forbid` : nom du programme exécutable
- $\lhd familles \rhd$ ::= {`p`, `a`, `i`, `I`}+ où chaque caractère correspond à une famille différente
  - `p` pour les permutations
  - `a` pour les permutations alternantes
  - `i` pour les involutions par la méthode récurrente
  - `I` pour les involutions par la méthode des points fixes
- $\lhd sorties \rhd$ ::= {`a`, `Y`, `p`, `c`, `f`, `r`, $*$, `t`, `s`, `m`, `e`, `i`, `y`, `x`}+ où chaque caractère correspond à une sortie différente

- $[+]\lhd niveau\_max \rhd$: limite la hauteur de l'arbre de génération, construit jusqu'au niveau $niveau\_max$ (absence du symbole +) ou jusqu'aux niveaux $k + niveau\_max$ pour chaque permutation racine de $\lhd perm\_racine \rhd$ sur $[k]$ (présence du symbole +)
- $\lhd motif\_exclu \rhd$: un motif (permutation p-barrée, barrée ou non barrée) à exclure
- $\lhd perm\_racine \rhd$: une permutation non barrée, racine de l'arbre construit
- `-f` $[\lhd fichier \rhd]$: spécifie le fichier (l'écran par défaut) devant recevoir les résultats

Exemple 3.12 *La commande*

```
forbid p am +3 236-4-15 2413 41-352 -r 1 321 12 -f resultat
```

demande au logiciel forbid, parmi l'ensemble des permutations excluant simultanément les motifs $236\overline{41}5$, 2413 *et* $41\overline{3}52$ *les permutations issues de* 1, 321 *et* 12 *dans l'arbre de génération des permutations, et respectivement sur* [4], [6] *et* [5] *au plus. La sortie, redirigée vers le fichier* `resultat`, *se compose de l'arbre ASCII (sortie* `a`*) et des distributions selon le nombre de montées et montées inverses (sortie* `m`*).*

Une page d'aide est donnée par le logiciel *forbid*; il suffit pour cela de taper la commande `forbid` sans option.

Nous présentons maintenant en détail chacune des sorties du logiciel *forbid*, en les illustrant avec l'ensemble des permutations excluant le motif 312.

- L'option `a` fournit l'arbre de génération sous forme d'un fichier texte ASCII.
  Dans ce cas, les permutations à motifs exclus sur $[n]$ sont représentées sur une même colonne. Chaque permutation est donnée avec ses sites actifs (`_`) et inactifs (`.`), et est suivie de son nombre de fils (entre parenthèses).

  Exemple 3.13 *Arbre ASCII de* $S_n(312)$.

```
Commande : forbid p a 4 312 -r 1 -f exemple
_1_(2)     _2_1_(3)     _3_2_1_(4)     _4_3_2_1_(5)
                                       .3_4_2_1_(4)
                                       .3.2_4_1_(3)
                                       .3.2.1_4_(2)
                        .2_3_1_(3)     .2_4_3_1_(4)
                                       .2.3_4_1_(3)
                                       .2.3.1_4_(2)
                        .2.1_3_(2)     .2.1_4_3_(3)
                                       .2.1.3_4_(2)
                .1_2_(2)     .1_3_2_(3)     .1_4_3_2_(4)
                                            .1.3_4_2_(3)
                                            .1.3.2_4_(2)
                        .1.2_3_(2)     .1.2_4_3_(3)
                                       .1.2.3_4_(2)
```

- L'option `Y` fournit l'ensemble des permutations à motifs exclus.
  Chaque ligne éditée correspond à une permutation (dont les éléments sont séparés par un caractère blanc). De plus, pour chaque permutation obtenue, cette option fournit la paire de tableaux de Young standard de même forme (codés par des mots de Yamanushi) lui correspondant par l'algorithme de Robinson-Schensted.

  Exemple 3.14 *$S_n(312)$.*

```
Commande : forbid p Y 4 312 -r 1 -f exemple
1               1           1
2 1             12          12
3 2 1           123         123
4 3 2 1         1234        1234
3 4 2 1         1231        1123
3 2 4 1         1231        1213
3 2 1 4         1231        1231
2 3 1           121         112
2 4 3 1         1213        1123
2 3 4 1         1211        1112
2 3 1 4         1211        1121
2 1 3           121         121
2 1 4 3         1212        1212
2 1 3 4         1211        1211
1 2             11          11
1 3 2           112         112
1 4 3 2         1123        1123
1 3 4 2         1121        1112
1 3 2 4         1121        1121
1 2 3           111         111
1 2 4 3         1112        1112
1 2 3 4         1111        1111
```

- L'option `p` fournit le mot parenthésé codant l'arbre de génération obtenu, où figure notamment le nombre de sites actifs.

  Exemple 3.15 *Mot parenthésé de $S_n(312)$.*

```
Commande : forbid p p 4 312 -r 1 -f exemple
(((23)(234))((23)(234)(2345)))
```

- L'option `c` fournit le fichier destiné au logiciel *CABRI* correspondant à l'arbre de génération obtenu, en vue de sa visualisation.

Un fichier texte, décrivant l'arbre de génération, est créé. Le logiciel *CABRI* reconnait alors ce dernier et le dessine. Chaque permutation est donnée avec ses sites actifs (`_`) et inactifs (`.`).

Exemple 3.16 *La commande* `forbid p c 3 312 -r 1 -f exemple_cabri` *crée le fichier* `exemple_cabri` *à destination du logiciel CABRI pour* $S_n(312)$.

- L'option `r` fournit toutes les règles de réécriture obtenues, en éliminant les redondances. L'arbre de génération est parcouru en regardant localement le nombre de fils t d'un sommet et pour chacun de ses fils, leur nombre de fils, respectivement $t_1, t_2, \ldots, t_t$. La règle de réécriture ainsi obtenue est $t \rightsquigarrow t_1, t_2, \ldots, t_t$. Toutes les règles de réécriture rencontrées sont éditées par ordre croissant sur t et sur les t_i.

Exemple 3.17 *Règles de réécriture de* $S_n(312)$.

```
Commande : forbid p r 10 312 -r 1 -f exemple
#fils --> #fils de tous les successeurs
 2    -->    2  3
 3    -->    2  3  4
 4    -->    2  3  4  5
 5    -->    2  3  4  5  6
 6    -->    2  3  4  5  6  7
 7    -->    2  3  4  5  6  7  8
 8    -->    2  3  4  5  6  7  8  9
 9    -->    2  3  4  5  6  7  8  9 10
10    -->    2  3  4  5  6  7  8  9 10 11
#regles differentes comptees = 9
```

- L'option `*` fournit la valeur de plusieurs paramètres sur les permutations obtenues. Cette sortie est donnée sous forme tabulaire, où chaque colonne correspond à un paramètre différent et chaque ligne correspond à une permutation.
  Les paramètres considérés sont les suivants.
  - `n` l'ordre de la permutation
  - `fils` le nombre de fils de la permutation correspondante dans l'arbre de génération
  - `sig` le nombre de minima à gauche (ou saillants inférieurs à gauche),
    `sid` le nombre de minima à droite (ou saillants inférieurs à droite),
    `ssg` le nombre de maxima à gauche (ou saillants supérieurs à gauche),
    `ssd` le nombre de maxima à droite (ou saillants supérieurs à droite)
  - `mont` le nombre de montées,
    `mont-1` le nombre de montées inverses

- `exc>=` le nombre d'excédences larges (éléments supérieurs ou égaux à leurs indices),
  `exc>` le nombre d'excédences strictes (éléments supérieurs à leurs indices)
- `pi-1(n)` l'indice du plus grand élément,
  `pi-1(1)` l'indice de l'élément 1,
  `pi(n)` le dernier élément,
  `pi(1)` le premier élément
- `cycles` le nombre de cycles,
  `+lgcyc` la longueur du plus long cycle,
  `ptfix` le nombre de points fixes
- `invers` le nombre d'inversions
- `major` l'index du major (somme des indices des descentes) de la permutation,
  `major-1` l'index du major de la permutation inverse

Exemple 3.18 *Les valeurs de ces paramètres pour $S_n(312)$ sont donnés par la figure 3.6.*

- Les options `t`, `s`, `m`, `e`, `i`, `y`, `x` fournissent les distributions de l'ensemble des permutations à motifs exclus selon un paramètre particulier.
  Chaque distribution est donnée sous forme tabulaire, avec en abscisse l'ordre des permutations et en ordonnée les valeurs du paramètre. De plus, la première ligne totalise chaque colonne, énumérant ainsi les permutations obtenues selon leur ordre.
  Les différents paramètres sont les suivants.
  - `t` le nombre de fils
  - `s` le nombre de minima ou maxima à gauche ou à droite (ou saillants)
  - `m` le nombre de montées et de montées inverses
  - `e` le nombre d'excédences strictes et larges
  - `i` les indices et valeurs relativement à n et 1
  - `y` le nombre de cycles et la longueur du plus long cycle
  - `x` le nombre de points fixes

Exemple 3.19 *Distribution de $|S_n(312)|$ suivant le nombre de maxima. Nous reconnaissons les premiers termes des nombres de Narayana et de Delannoy (distributions des nombres de Catalan).*

```
Commande : forbid p s 10 312 -r 1 -f exemple
Perm. : #saillants superieurs gauches sur les ordonnees
Sn= =1 =2 =5 =14 =42 =132 =429 =1430 =4862 =16796
10:  -  -  -   -   -    -    -     -     -      1
 9:  -  -  -   -   -    -    -     -     1     45
 8:  -  -  -   -   -    -    -     1    36    540
 7:  -  -  -   -   -    -    1    28   336   2520
 6:  -  -  -   -   -    1   21   196  1176   5292
 5:  -  -  -   -   1   15  105   490  1764   5292
 4:  -  -  -   1  10   50  175   490  1176   2520
 3:  -  -  1   6  20   50  105   196   336    540
 2:  -  1  3   6  10   15   21    28    36     45
 1:  1  1  1   1   1    1    1     1     1      1
    1: 2: 3:  4:  5:   6:   7:    8:    9:    10: [n]
Perm. : #saillants superieurs droits sur les ordonnees
Sn= =1 =2 =5 =14 =42 =132 =429 =1430 =4862 =16796
10:  -  -  -   -   -    -    -     -     -      1
 9:  -  -  -   -   -    -    -     -     1      9
 8:  -  -  -   -   -    -    -     1     8     44
 7:  -  -  -   -   -    -    1     7    35    154
 6:  -  -  -   -   -    1    6    27   110    429
 5:  -  -  -   -   1    5   20    75   275   1001
 4:  -  -  -   1   4   14   48   165   572   2002
 3:  -  -  1   3   9   28   90   297  1001   3432
 2:  -  1  2   5  14   42  132   429  1430   4862
 1:  1  1  2   5  14   42  132   429  1430   4862
    1: 2: 3:  4:  5:   6:   7:    8:    9:    10: [n]
```

```
Commande : forbid p * 4 312 -r 1 -f exemple
n fils sig sid ssg ssd mont mont-1 exc>= exc> pi-1(n) pi-1(1) pi(n) pi(1) cycles +lgcyc ptfix invers major major-1
1 2    1   1   1   1   0    0      1     0    1       1       1     1     1      1      1     0      0     0
2 3    2   1   1   2   0    0      1     1    1       2       1     2     1      2      0     1      1     1
3 4    3   1   1   3   0    0      2     1    1       3       1     3     2      2      1     3      3     3
4 5    4   1   1   4   0    0      2     2    1       4       1     4     2      2      0     6      6     6
4 4    3   1   2   3   1    1      2     2    2       4       1     3     1      4      0     5      5     3
4 3    3   1   2   2   1    1      3     2    3       4       1     3     2      3      1     4      4     3
4 2    3   2   2   1   1    1      3     1    4       3       4     3     3      2      2     3      3     3
3 3    2   1   2   2   1    1      2     2    2       3       1     2     1      3      0     2      2     1
4 4    2   1   2   3   1    1      3     2    2       4       1     2     2      3      1     4      5     4
4 3    2   1   3   2   2    2      3     3    3       4       1     2     1      4      0     3      3     1
4 2    2   2   3   1   2    2      3     2    4       3       4     2     2      3      1     2      2     1
3 2    2   2   2   1   1    1      2     1    3       2       3     2     2      2      1     1      1     1
4 3    2   2   2   2   1    1      2     2    3       2       3     2     2      2      0     2      4     4
4 2    2   3   3   1   2    2      3     1    4       2       4     2     3      2      2     1      1     1
2 2    1   2   2   1   1    1      2     0    2       1       2     1     2      1      2     0      0     0
3 3    1   2   2   2   1    1      2     1    2       1       2     1     2      2      1     1      2     2
4 4    1   2   2   3   1    1      3     1    2       1       2     1     3      2      2     3      5     5
4 3    1   2   3   2   2    2      3     2    3       1       2     1     2      3      1     2      3     2
4 2    1   3   3   1   2    2      3     1    4       1       4     1     3      2      2     1      2     2
3 2    1   3   3   1   2    2      3     0    3       1       3     1     3      1      3     0      0     0
4 3    1   3   3   2   2    2      3     1    3       1       3     1     3      2      2     1      3     3
4 2    1   4   4   1   3    3      4     0    4       1       4     1     4      1      4     0      0     0
```

Figure 3.6 Les paramètres générés par l'option * de *forbid*.

Chapitre 4

Permutations à motifs exclus énumérées par quelques suites classiques

Afin d'illustrer cette méthode consistant à construire l'arbre de génération d'une famille d'objets combinatoires, nous avons considéré quelques classes de permutations à motifs exclus pour lesquels nous obtenons des résultats nouveaux. Pour cela, le logiciel *forbid* nous a été d'une grande utilité, nous permettant souvent de deviner des formules d'énumération et parfois nous suggérant des bijections par isomorphisme d'arbres de génération.

L'ensemble des résultats obtenus vient compléter le catalogue (voir annexe A) déjà important relatif à l'énumération d'ensembles de permutations à motifs exclus auquel de nombreux auteurs ont contribué.

La particularité des résultats obtenus ici réside dans le fait que les formules d'énumération obtenues sont toutes très classiques en Combinatoire.
Ainsi, nous mettons en évidence des ensembles de permutations à motifs exclus dont les formules d'énumération sont proches des nombres de Catalan [13] tels les coefficients binomiaux centraux qui énumèrent les mots du langage que nous appelons Grand Dyck et les nombres de Motzkin [75] et de Schröder [90] qui dénombrent en particulier les arbres 1-2 selon deux paramètres différents.

Dans un premier temps, nous mettons en correspondance trois ensembles de permutations à motifs exclus et montrons qu'ils sont énumérés par les nombres de Pell satisfaisant la formule de récurrence $p_n = 2p_{n-1} + p_{n-2}$ $(p_1 = 1, p_2 = 2)$.

Ensuite, nous mettons en évidence de nombreux ensembles de permutations à motifs exclus en bijection avec les mots du Grand Dyck et donc énumérés par les coefficients binomiaux centraux $\binom{2n}{n}$.

De plus, nous complétons les travaux de S. Gire [45] sur les nombres de Motzkin $\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$ et ceux de J. West [110, 112] et S. Gire [45] sur les nombres de Schröder $\sum_{i=0}^{n} \binom{n+i}{n-i} c_i$.
En particulier, nous montrons que les arbres de génération d'un ensemble de permutations et de deux ensembles d'involutions à motifs exclus sont caractérisés par le même système de réécriture, qui lui-même caractérise un arbre de génération des arbres 1-2 suivant le nombre de sommets. Nous procédons de même pour d'autres ensembles de permutations en les reliant à ces mêmes arbres, mais considérés cette fois-ci distribués suivant leur nombre de sommets internes.

Pour terminer, nous nous intéressons aux tableaux de Young standard de hauteur bornée ainsi qu'à des paires de tels tableaux. Ceci revient à considérer respectivement les involutions et les permutations excluant le motif identité. Nous donnons les systèmes de réécriture caractérisant les arbres de génération de ces objets.

4.1 Nombres de Pell

Les nombres de Pell, dont les premières valeurs sont $1, 2, 5, 12, 29, 70, \ldots$, apparaissent dans un exercice proposé par G.L. Alexanderson [59].

Nous allons montrer que ces nombres énumèrent trois ensembles de permutations à motifs exclus que nous mettons en correspondance par isomorphisme de leurs arbres de génération.

Théorème 4.1 *Les ensembles de permutations à motifs exclus* $S_n(123, 2143, 3214)$, $S_n(213, 1234, 1243)$ *et* $S_n(132, 2341, 3241)$ *sont en bijection et sont énumérés par le* $n^{ème}$ *nombre de Pell donné par*

$$p_n = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n}{2k+1} 2^k$$

défini par la formule de récurrence $p_n = 2p_{n-1} + p_{n-2}$ $(p_1 = 1, p_2 = 2)$.

Pour établir ce résultat, nous montrons que les arbres de génération $T(123, 2143, 3214)$, $T(213, 1234, 1243)$ et $T(132, 2341, 3241)$ sont isomorphes en les caractérisant par le même système de réécriture.

Proposition 4.2

- *Le système de réécriture* $\mathcal{S}_{Pell}$ *(voir figure 4.1) caractérisant les arbres de génération* $T(123, 2143, 3214)$, $T(213, 1234, 1243)$ *et* $T(132, 2341, 3241)$ *est*
$$\begin{cases} (2) \\ (2) \rightsquigarrow (2), (3) \\ (3) \rightsquigarrow (2), (2), (3) \end{cases}$$
- *Les étiquettes* (2) *et* (3) *du système de réécriture* $\mathcal{S}_{Pell}$ *correspondent au nombre de sites actifs d'une permutation de l'un quelconque des trois arbres de génération.*

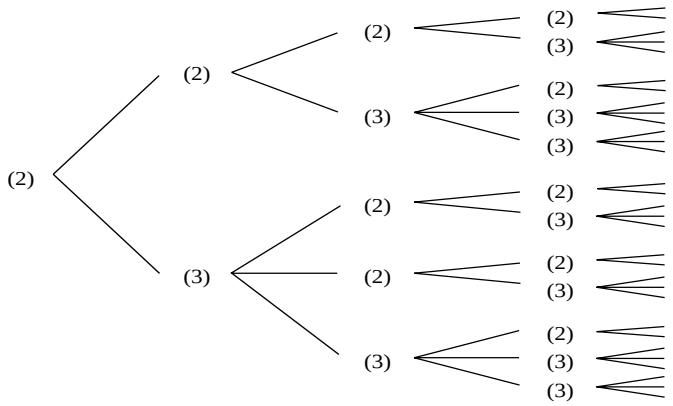


Figure 4.1 Arbre de dérivation du système de réécriture $\mathcal{S}_{Pell}$.

Lemme 4.3 *Les permutations π de $S_n(123, 2143, 3214)$ ont les trois premiers sites actifs si $\pi(1) = n$, les deux premiers sites actifs sinon.*

Preuve Clairement, compte-tenu de la forme des motifs exclus, les deux premiers sites de π sont toujours actifs.

Supposons qu'il y ait un site actif autre que l'un des trois premiers. Alors, la permutation $\pi(1)\pi(2)\pi(3)\ldots(n+1)\ldots$ appartient à $S_{n+1}(123, 2143, 3214)$. Le motif 123 étant interdit, nous devrions avoir $\pi(3) < \pi(2) < \pi(1)$ et ainsi la sous-suite $\pi(1)\pi(2)\pi(3)(n+1)$ serait de type 3214, ce qui est interdit. S'il y a 3 sites actifs, alors $\pi(1)\pi(2)(n+1)\ldots \in S_{n+1}(123, 2143, 3214)$. Si $\pi(1) < n$, alors soit la sous-suite $\pi(1)\pi(2)(n+1)$ est de type 123 si $\pi(1) < \pi(2)$, soit la sous-suite $\pi(1)\pi(2)(n+1)n$ est de type 2143 sinon. A l'inverse, si $\pi(1) = n$, le troisième site est actif d'après la forme des motifs exclus. □

Les deux lemmes suivants se démontrent de façon analogue.

Lemme 4.4 *Les permutations π de $S_n(213, 1234, 1243)$ ont les trois premiers sites actifs si $\pi(2) = n$, les deux premiers sites actifs sinon.*

Lemme 4.5 *Les permutations π de $S_n(132, 2341, 3241)$ ont les premier, deuxième et dernier sites actifs si $\pi(1) = n$, les premier et dernier sites actifs sinon.*

Preuve de la proposition 4.2. Clairement, la permutation 1 engendre les permutations 12 et 21 et admet donc 2 fils dans chacun des trois arbres de génération.

D'après les lemmes précédents, une permutation π de $S_n(123, 2143, 3214)$ [resp. $S_n(213, 1234, 1243)$ et $S_n(132, 2341, 3241)$] telle que $\pi^{-1}(n) = 1$ [resp. 2 et 1] permet d'engendrer 3 permutations, chacune d'elles en engendrant respectivement 3,2,2 [resp. 2,3,2 et 3,2,2]. Par contre, si $\pi^{-1}(n) \neq 1$ [resp. 2 et 1], la permutation π n'engendre que 2 permutations, chacune d'elles en engendrant respectivement 3,2 [resp. 2,3 et 3,2]. □

Preuve du théorème 4.1. Soit maintenant p_n le nombre de permutations de $S_n(123, 2143, 3214)$, $S_n(213, 1234, 1243)$ ou $S_n(132, 2341, 3241)$.

D'après le système de réécriture $\mathcal{S}_{Pell}$, nous avons $p_{n,(2)} = p_{n-1,(2)} + 2p_{n-1,(3)}$ et $p_{n,(3)} = p_{n-1,(2)} + p_{n-1,(3)}$

où $p_{n,(t)}$ est le nombre d'étiquettes (t) au niveau n $(t = 2, 3)$. Ainsi, nous obtenons que $p_n = p_{n,(2)} + p_{n,(3)} = 2(p_{n-1,(2)} + p_{n-1,(3)}) + p_{n-1,(3)} = 2p_{n-1} + p_{n-2}$, avec $p_1 = 1$ et $p_2 = 2$.
Nous pouvons également déduire du système de réécriture $\mathcal{S}_{Pell}$ que $\begin{pmatrix} p_{n,(2)} & p_{n,(3)} \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}^{n-1}$. Alors, nous avons que $p_n = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n}{2k+1} 2^k$ pour tout $n \geq 1$ avec $p_{n,(2)} = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-1}{2k} 2^k$ et $p_{n,(3)} = \sum_{k=0}^{\lfloor \frac{n-2}{2} \rfloor} \binom{n-1}{2k+1} 2^k$. □

4.2 Coefficients binomiaux centraux

Les coefficients binomiaux centraux, dont les premières valeurs sont $1, 2, 6, 20, 70, 252, \ldots$, apparaissent lors de l'énumération de divers objets combinatoires, comme par exemple dans le cas des polyominos convexes dirigés comptés suivant le périmètre [14, 109].

Nous montrons que onze ensembles de permutations à motifs exclus sont en correspondance avec les mots du Grand Dyck, par isomorphisme de leurs arbres de génération (il est cependant nécessaire d'exhiber quatre systèmes de réécriture différents), et sont donc énumérés par les coefficients binomiaux centraux. De plus, nous prouvons analytiquement qu'il en est de même pour un autre ensemble de permutations à motifs exclus.

Théorème 4.6 *Les ensembles de permutations à motifs exclus* $S_n(1234, 1243, 1423, 4123)$, $S_n(1324, 1342, 1432, 4132)$, $S_n(2134, 2143, 2413, 4213)$, $S_n(2314, 2413, 3142, 3241)$, $S_n(1234, 1324, 2134, 2314)$, $S_n(1234, 2134, 2314, 3124)$, $S_n(1324, 2134, 2314, 3124)$, $S_n(1324, 2134, 3124, 3214)$, $S_n(1324, 2314, 3124, 3214)$, $S_n(1342, 2341, 3142, 3241)$ *et* $S_n(1324, 1342, 2314, 2341)$, *sont en bijection avec les mots de* $GD_{z,\overline{z}}$ *de longueur* $2n-2$, *et sont donc énumérés par le* $(n-1)^{ème}$ *coefficient binomial central*

$$\binom{2n-2}{n-1}$$

La même formule énumère les permutations de $S_n(1342, 2341, 2431, 3241)$.

Le schéma général de la preuve qui suit est donné par la figure 4.2 où les ensembles regroupés seront caractérisés par le même système de réécriture.

Proposition 4.7

- *Le système de réécriture* $\mathcal{S}_{GrandDyck1}$ *(voir figure 4.3) caractérisant les arbres de génération* $T(2134, 2143, 2413, 4213)$, $T(1234, 1243, 1423, 4123)$ *et* $T(1324, 1342, 1432, 4132)$ *est*
$$\begin{cases} (1,2) \\ (p,t) \rightsquigarrow (1,t+1), (2,t+1), \ldots, (p+1,t+1), \underbrace{(0), (0), \ldots, (0)}_{t-p-1 \text{ fois}} \end{cases}$$
- *L'étiquette* (p,t) *du système de réécriture* $\mathcal{S}_{GrandDyck1}$ *correspondant à une permutation* π *sur* $[n]$ *de l'un quelconque des trois arbres de génération vérifie*

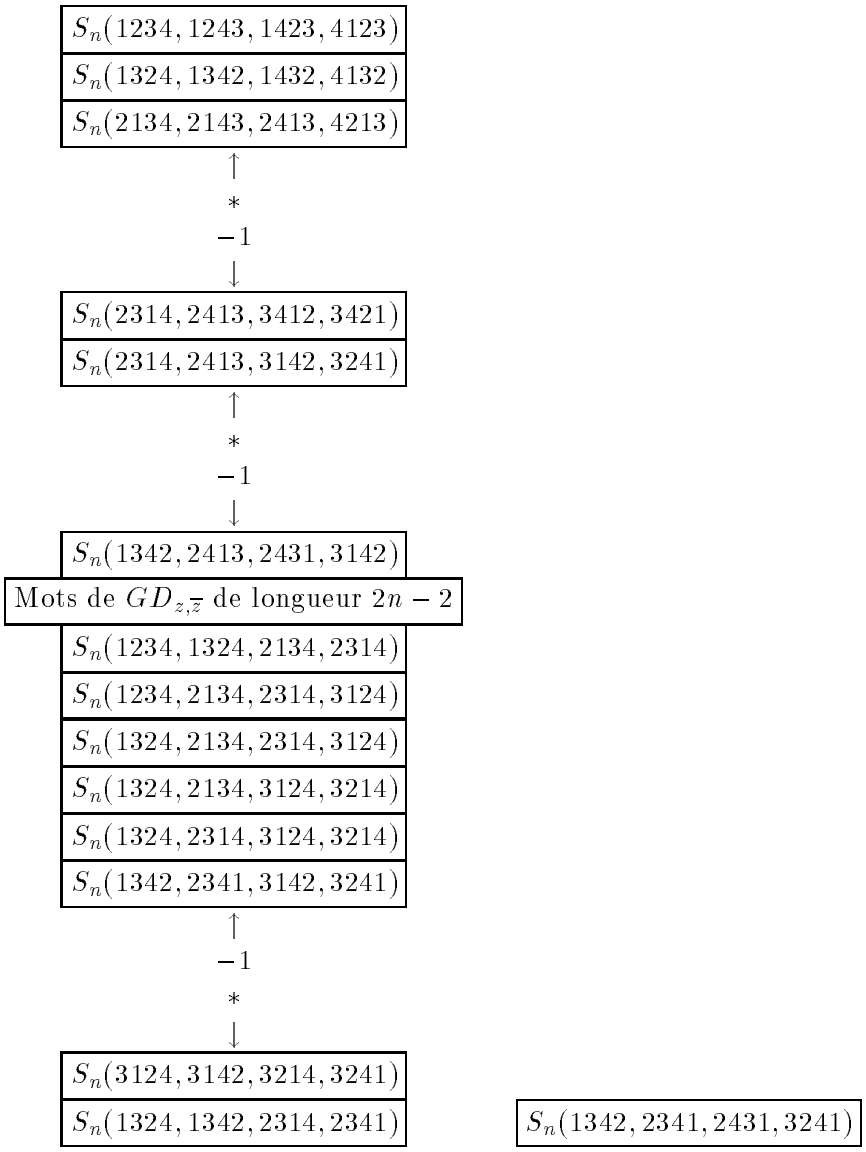


Figure 4.2 Schéma des bijections entre mots de $GD_{z,\overline{z}}$ et plusieurs ensembles de permutations à motifs exclus.

- $p = \pi^{-1}(n)$ *pour* $\pi \in T(2134, 2143, 2413, 4213)$,
- $p = \min\{i : \pi(i) < \pi(i+1)\}$ *(ou* $p = n$ *si* $\pi = n(n-1)\ldots1$*) pour* $\pi \in T(1234, 1243, 1423, 4123)$,
- $p = maxd(\pi)$ *pour* $\pi \in T(1324, 1342, 1432, 4132)$,
- t *est le nombre de sites actifs de* π.

L'étiquette (0) *indique pour sa part que la permutation correspondante n'a aucun site actif.*

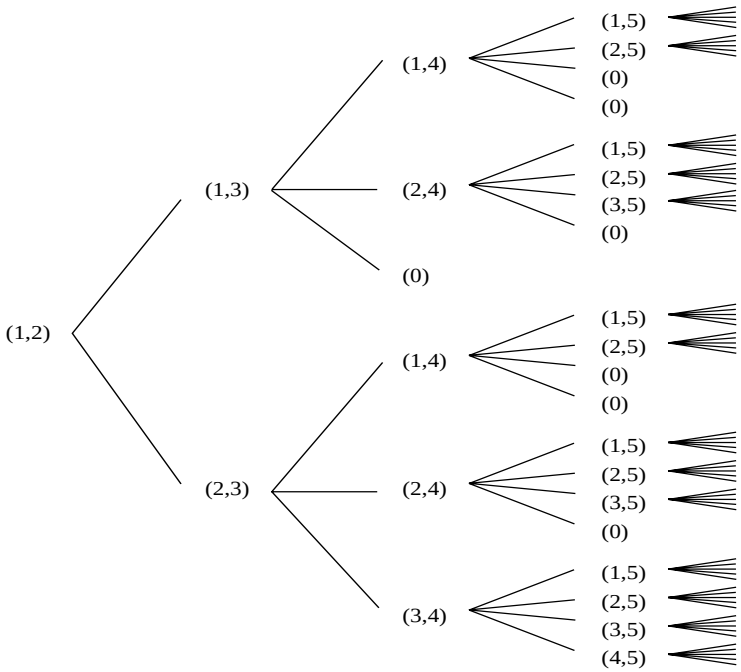


Figure 4.3 Arbre de dérivation du système de réécriture $\mathcal{S}_{GrandDyck1}$.

Lemme 4.8 *Les permutations de* $S_n(2134, 2143, 2413, 4213)$ *[resp.* $S_n(1234, 1243, 1423, 4123)$ *et* $S_n(1324, 1342, 1432, 4132)$*] ont soit tous leurs sites actifs, soit aucun lorsqu'elles contiennent une sous-suite de type* 213 *[resp.* 123 *et* 132*].*

Preuve Il suffit de remarquer que $\{2134, 2143, 2413, 4213\} = \{213 \sqcup\!\sqcup 4\}$ [resp. $\{1234, 1243, 1423, 4123\} = \{123 \sqcup\!\sqcup 4\}$ et $\{1324, 1342, 1432, 4132\} = \{132 \sqcup\!\sqcup 4\}$]. □

Preuve de la proposition 4.7. Compte-tenu du lemme précédent, nous pouvons utiliser les travaux de J. West [110] qui a montré que les arbres de génération $T(213)$, $T(123)$ et $T(132)$ sont caractérisés par le système de réécriture $\begin{cases} (1) \\ (f) \quad \rightsquigarrow \quad (1), (2), \ldots, (f+1) \end{cases}$ pour lequel l'interprétation de l'étiquette f est identique à celle que nous donnons pour le paramètre p de l'étiquette (p, t) dans notre système de réécriture (en fait, $f = p + 1$).

Soit π une permutation appartenant à l'arbre de génération $T(2134, 2143, 2413, 4213)$ [resp. $T(1234, 1243, 1423, 4123)$ et $T(1324, 1342, 1432, 4132)$].

Si π appartient à $T(213)$ [resp. $T(123)$ et $T(132)$], son étiquette est (p,t) et elle engendre $p+1$ permutations appartenant à $T(213)$ [resp. $T(123)$ et $T(132)$] ayant un site actif supplémentaire : leurs étiquettes sont donc $(1,t+1),(2,t+1),\ldots,(p+1,t+1)$. De plus, π engendre $t-(p+1)$ permutations n'appartenant pas à $T(213)$ [resp. $T(123)$ et $T(132)$]; elles ont donc toutes (0) pour étiquette.
Dans le cas où la permutation π n'appartient pas à $T(213)$ [resp. $T(123)$ et $T(132)$], son étiquette est (0) et elle n'engendre aucune permutation.

Remarquons que la permutation 1 a bien pour étiquette $(1,2)$. □

Proposition 4.9

- *Le système de réécriture $\mathcal{S}_{GrandDyck2}$ (voir figure 4.4) caractérisant les arbres de génération $T(2314,2413,3412,3421)$ et $T(2314,2413,3142,3241)$ est*
$$\begin{cases} (A,2) & \\ (A,t) & \rightsquigarrow \quad (B,3),(B,4),\ldots,(B,t+1),(A,t+1) \\ (B,t) & \rightsquigarrow \quad (B,3),(B,4),\ldots,(B,t),(C,t),(A,t) \\ (C,t) & \rightsquigarrow \quad (C,2),(C,3),\ldots,(C,t+1) \end{cases}$$
- *L'étiquette (X,t) du système de réécriture $\mathcal{S}_{GrandDyck2}$ correspondant à une permutation π sur $[n]$ de l'un quelconque des deux arbres de génération vérifie*
  - *$X=A$ si le dernier site est actif et $\pi(n)=n$,*
  - *$X=B$ si le dernier site est actif et $\pi(n)\neq n$,*
  - *$X=C$ si le dernier site est inactif,*
  - *t est le nombre de sites actifs de π.*

Lemme 4.10 *Pour une permutation π de $S_n(2314,2413,3412,3421)$, nous avons les propriétés suivantes.*

(i) Les premier et avant-dernier sites et celui situé à gauche de n sont actifs.

(ii) Le dernier site est actif si et seulement si π appartient à $S_n(231)$.

(iii) Tous les sites à droite de n jusqu'à l'antépénultième sont inactifs.

(iv) Les autres sites sont actifs si et seulement si tous les éléments situés à leur gauche sont inférieurs à tous ceux situés à leur droite et à la gauche de n.

(v) L'insertion de l'élément $n+1$ dans l'un des sites actifs à gauche de n laisse dans le même état tous les sites situés à gauche de $n+1$ dans la permutation obtenue.

Preuve

- (i) résulte de la forme des motifs exclus. En particulier, l'avant-dernier site est toujours actif car seul le motif 2314 pourrait l'inactiver par une sous-suite $\pi(i_1)\pi(i_2)\pi(i_3)$ de type 231 formée sur les $n-1$ premiers éléments de π; mais alors la sous-suite $\pi(i_1)\pi(i_2)\pi(i_3)\pi(n)$ est de l'un des types 2413, 3412 ou 3421, ces motifs étant tous exclus.

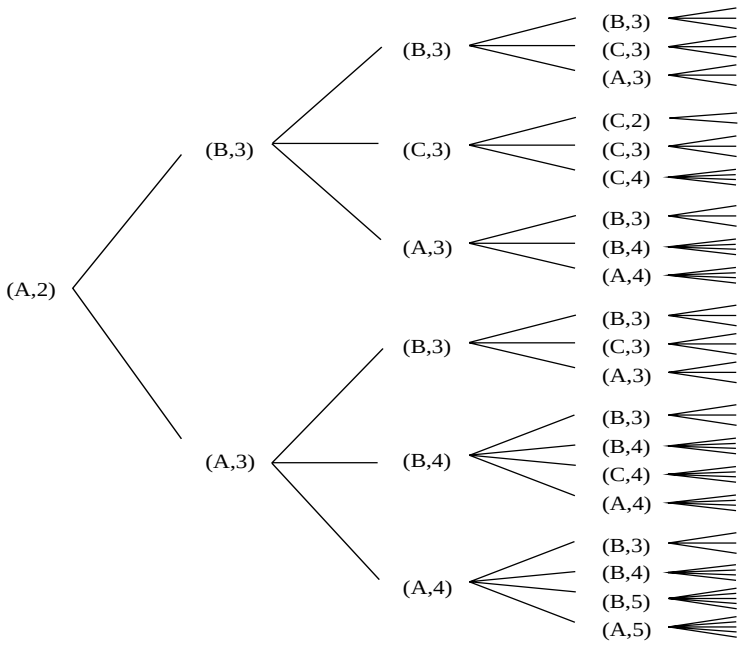


Figure 4.4 Arbre de dérivation du système de réécriture $\mathcal{S}_{GrandDyck2}$.

- (ii) résulte du motif 2314 interdit et de la forme des trois autres.
- (iii) permet d'éviter les motifs 3412 et 3421 à partir de la sous-suite $n\pi(n-1)\pi(n)$.
- (iv) résulte de la forme du motif exclu 2413.
- (v) résulte de la forme des motifs exclus.

□

Pour des raisons analogues, nous obtenons le résultat suivant.

Lemme 4.11 *Pour une permutation π de $S_n(2314, 2413, 3142, 3241)$, nous avons les propriétés suivantes.*

(i) Le premier site et ceux entourant n sont actifs.

(ii) Le dernier site est actif si et seulement si π appartient à $S_n(231)$.

(iii) Les autres sites situés à droite de n sont inactifs.

(iv) Les autres sites sont actifs si et seulement si tous les éléments situés à leur gauche sont inférieurs à tous ceux situés à leur droite et à la gauche de n.

(v) L'insertion de l'élément $n+1$ dans l'un des sites actifs à gauche de n laisse dans le même état tous les sites situés à gauche de $n+1$ dans la permutation obtenue.

Preuve de la proposition 4.9. Clairement, dans les deux arbres de génération des permutations, la permutation 1 doit être étiquetée $(A, 2)$ (elle engendre les deux permutations 12 et 21).
Soit π une permutation de $S_n(2314, 2413, 3412, 3421)$ [resp. $S_n(2314, 2413, 3142, 3241)$] d'étiquette (X, t).

- $X = A$.
  - Insertion dans l'un des sites actifs autre que le dernier.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [t-1]$.
    Le dernier site reste actif.
    L'étiquette de γ est alors $(B, i+2)$.
  - Insertion dans le dernier site.
    L'étiquette de la permutation ainsi obtenue est alors $(A, t+1)$.
- $X = B$.
  - Insertion dans l'un des sites actifs autre que les deux derniers sites actifs.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [t-2]$.
    Le dernier site reste actif.
    L'étiquette de γ est alors $(B, i+2)$.
  - Insertion dans l'avant-dernier des sites actifs.
    La sous-suite $n(n+1)\pi(n)$ est de type 231 et inactive le dernier site.
    L'étiquette de la permutation ainsi obtenue est alors (C, t).
  - Insertion dans le dernier site.
    Le site à gauche de $\pi(n)$ [resp. à droite de n] s'inactive car la sous-suite $n(n+2)\pi(n)(n+1)$ est de type 2413.
    L'étiquette de la permutation ainsi obtenue est alors (A, t).
- $X = C$.
  - Insertion dans l'un des sites actifs autre que le dernier site actif.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [t-1]$.
    A droite de $n+1$, seul l'avant-dernier site [resp. le site à droite de $n+1$] reste actif.
    L'étiquette de γ est alors $(C, i+1)$.
  - Insertion dans le dernier site actif.
    Tous les sites actifs à gauche de n restent actifs et les deux sites entourant $n+1$ sont actifs.
    L'étiquette de la permutation ainsi obtenue est alors $(C, t+1)$.

□

Proposition 4.12 *Un arbre de génération des mots de $GD_{z,\overline{z}}$ (voir figure 4.5) est*

$$\begin{cases} \varepsilon & & \\ \varepsilon & \rightsquigarrow & \overline{z}z, \\ & & z\overline{z} \\ x^l\overline{x}w' & \rightsquigarrow & x^i\overline{x}x^{l+1-i}\overline{x}w' \text{ pour tout } i \in [0, l+1] \end{cases}$$

où ε est le mot vide, $l > 0$, $x \in \{z, \overline{z}\}$ et $\overline{\overline{z}} = z$.

Preuve Cet arbre de génération est construit de manière analogue à celui obtenu pour les mots de parenthèses en s'intéressant à leur distribution suivant la hauteur initiale (exemple 2.6). □

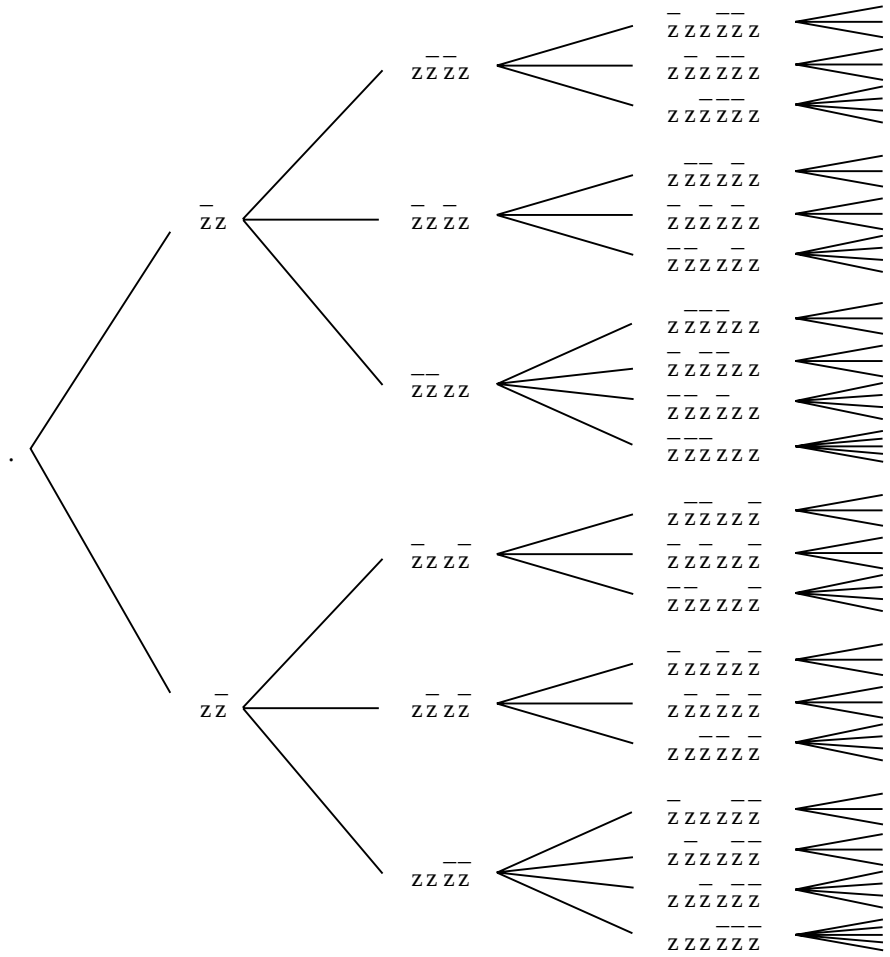


Figure 4.5 Arbre de génération des mots de $GD_{z,\overline{z}}$.

Proposition 4.13

- *Le système de réécriture $\mathcal{S}_{GrandDyck3}$ (voir figure 4.6) caractérisant cet arbre de génération des mots de $GD_{z,\overline{z}}$ et les arbres de génération $T(1342, 2413, 2431, 3142)$, $T(1234, 1324, 2134, 2314)$, $T(1234, 2134, 2314, 3124)$, $T(1324, 2134, 2314, 3124)$, $T(1324, 2134, 3124, 3214)$, $T(1324, 2314, 3124, 3214)$ et $T(1342, 2341, 3142, 3241)$ est*
  $\begin{cases} (0) \\ (l) \quad \rightsquigarrow \quad (1), (1), (2), \ldots, (l+1) \end{cases}$
- *L'étiquette (l) du système de réécriture $\mathcal{S}_{GrandDyck3}$ est telle que*
  - *le mot de $GD_{z,\overline{z}}$ est de la forme $w = x^l \overline{x} w'$ avec $x \in \{z, \overline{z}\}$ et $\overline{\overline{z}} = z$,*
  - *$l+2$ est le nombre de sites actifs d'une permutation de l'un quelconque des sept arbres de génération.*

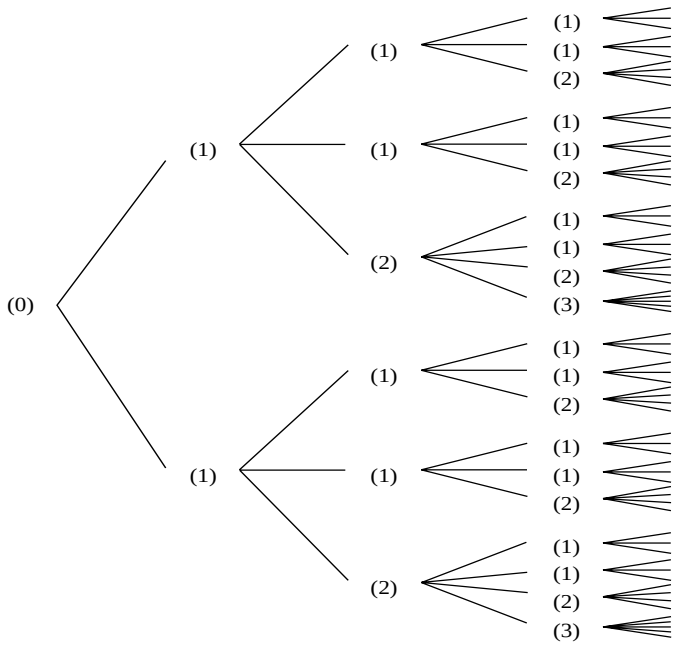


Figure 4.6 Arbre de dérivation du système de réécriture $\mathcal{S}_{GrandDyck3}$ caractérisant notamment les mots de $GD_{z,\overline{z}}$.

Lemme 4.14 *Pour une permutation de $S_n(1342, 2413, 2431, 3142)$, nous avons les propriétés suivantes.*

(i) Les premier et dernier sites sont actifs.

(ii) Les autres sites actifs sont soit tous situés à gauche de n, soit tous situés à droite de n.

(iii) Un site est actif si et seulement si tous les éléments situés à sa gauche sont soit inférieurs, soit supérieurs à tous ceux situés à sa droite.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii). Supposons qu'un site situé à gauche [resp. droite] de n soit actif, excepté le premier [resp. dernier]. Alors, tous les sites à droite [resp. gauche] de n sauf le dernier [resp. premier] sont inactifs à cause de la sous-suite $\pi(1)n(n+1)\pi(n)$ [resp. $\pi(1)(n+1)n\pi(n)$] de type 1342 [resp. 2431].
- (iii).
  Considérons un site actif autre que les premier et dernier. Si l'élément n est situé à droite [resp. gauche] de ce site, alors, pour éviter les motifs 2413 et 2431 [resp. 3142 et 1342], tous les éléments à gauche de ce site doivent être inférieurs [resp. supérieurs] à tous ceux à sa droite.
  Soit $k \in [n]$ tel que les k premiers [resp. derniers] éléments de π forment une permutation de $[k]$ et donc que les $n-k$ derniers [resp. premiers] éléments de π forment une permutation de $[k+1, n]$. L'activation du site situé à droite du $k^{\hat{e}me}$ [resp. $(n-k)^{\hat{e}me}$] élément de π ne peut créer que des sous-suites de type 1234, 1324, 2134, 2314, 3124, 3214, 1243, 2143, 1423, 1432, 4123, 4132, 4213,

4231, 4312, 4321 [resp. 1234, 1324, 2134, 2314, 3124, 3214, 2341, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312, 4321] dont aucune n'est à exclure.

□

Lemme 4.15 *Si une permutation π appartenant à $S_n(1234, 1324, 2134, 2314)$, [resp. $S_n(1234, 2134, 2314, 3124)$, $S_n(1324, 2134, 2314, 3124)$, $S_n(1324, 2134, 3124, 3214)$, $S_n(1324, 2314, 3124, 3214)$] a $l+2$ sites actifs dans l'arbre de génération correspondant, alors ses $l+2$ premiers sites sont actifs et le type la sous-suite $\pi(1)\pi(2)\dots\pi(l+1)$ appartient à $S_{l+1}(123, 132, 213, 231) = \{(l+1)l\dots1, (l+1)l\dots312\}$ [resp. $S_{l+1}(123, 213, 231, 312) = \{(l+1)l\dots1, 1(l+1)l\dots2\}$, $S_{l+1}(132, 213, 231, 312) = \{12\dots(l+1), (l+1)l\dots1\}$, $S_{l+1}(132, 213, 312, 321) = \{12\dots(l+1), 23\dots(l+1)1\}$, $S_{l+1}(132, 231, 312, 321) = \{12\dots(l+1), 2134\dots(l+1)\}$].*
De plus, pour une permutation de $S_n(1324, 2314, 3124, 3214)$, le site situé à droite de n est actif.

Preuve Les motifs exclus étant des permutations de S_4 ayant 4 pour dernier élément, lorsqu'un site est inactif, tous ceux à sa droite sont également inactifs. De plus, les trois premiers sites sont actifs.
R. Simion et F.W. Schmidt [95] ont montré que $|S_n(\tau_1, \tau_2, \tau_3, \tau_4)| = 2$ pour tout $n \geq 2$ lorsque $\tau_1, \tau_2, \tau_3, \tau_4$ sont des permutations de S_3 différentes deux à deux et vérifiant $\{123, 321\} \not\subset \{\tau_1, \tau_2, \tau_3, \tau_4\}$. Il est alors aisé de caractériser les deux permutations à motifs exclus des ensembles correspondants. □

Lemme 4.16 *Une permutation de $S_n(1342, 2341, 3142, 3241)$ vérifie les propriétés suivantes.*

(i) Les premier, deuxième et dernier sites sont actifs.

(ii) Les autres sites à droite de n sont inactifs.

(iii) L'insertion de l'élément $n+1$ dans l'un des sites actifs à gauche de n laisse dans le même état tous les sites situés à gauche de $n+1$ dans la permutation obtenue.

Preuve

- (i) et (iii) résultent de la forme des motifs exclus.
- (ii). Autrement, la sous-suite $n\pi(\pi^{-1}(n)+1)(n+1)\pi(n)$ serait de type 3142 ou 3241 et la sous-suite $\pi(1)n(n+1)\pi(n)$ serait de type 1342 ou 2341.

□

Preuve de la proposition 4.13. Pour les mots de $GD_{z,\overline{z}}$, le résultat se déduit immédiatement de la proposition 4.12 définissant l'arbre de génération de ces mots.
Considérons maintenant le cas des permutations à motifs exclus. Clairement, l'étiquette de la permutation 1 doit être (0) dans tous les cas. Soit π une permutation de l'un quelconque des arbres de génération des permutations ayant pour étiquette (l).

- $\pi \in S_n(1342, 2413, 2431, 3142)$.
  Il nous suffit de considérer le cas où les l sites actifs autres que les premier et dernier sites sont à gauche de n. En effet, dans le cas contraire, l'étude de π^* se ramène au cas considéré.

- Insertion dans le premier site.
  L'étiquette de la permutation ainsi obtenue est alors (1).
- Insertion dans l'un des sites actifs autre que les premier et dernier sites.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [2, l+1]$.
  L'étiquette de γ est alors $(i-1)$.
- Insertion dans le dernier site.
  L'étiquette de la permutation ainsi obtenue est alors $(l+1)$.

- $\pi \in S_n(1234, 1324, 2134, 2314)$.
  - Insertion dans le premier site.
    L'étiquette de la permutation ainsi obtenue est alors $(l+1)$.
  - Insertion dans le deuxième site.
    L'étiquette de la permutation ainsi obtenue est alors (1).
  - Insertion dans l'un des sites actifs autre que les deux premiers sites.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [3, l+2]$.
    L'étiquette de γ est alors $(i-2)$.

- $\pi \in S_n(1234, 2134, 2314, 3124)$.
  Le type des $l+1$ premiers éléments de π est $(l+1)l \dots 1$ [resp. $1(l+1)l \dots 2$].
  - Insertion dans le premier site.
    L'étiquette de la permutation ainsi obtenue est alors $(l+1)$ [resp. (1)].
  - Insertion dans le deuxième site.
    L'étiquette de la permutation ainsi obtenue est alors (1) [resp. $(l+1)$].
  - Insertion dans l'un des sites actifs autre que les deux premiers sites.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [3, l+2]$.
    L'étiquette de γ est alors $(i-2)$ [resp. $(i-2)$].

- $\pi \in S_n(1324, 2134, 2314, 3124)$.
  Le type des $l+1$ premiers éléments de π est $12 \dots (l+1)$ [resp. $(l+1)l \dots 1$].
  - Insertion dans le premier site.
    L'étiquette de la permutation ainsi obtenue est alors (1) [resp. $(l+1)$].
  - Insertion dans le deuxième site.
    L'étiquette de la permutation ainsi obtenue est alors (1) [resp. (1)].
  - Insertion dans l'un des sites actifs autre que les deux premiers sites.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [3, l+2]$.
    L'étiquette de γ est alors $(i-1)$ [resp. $(i-2)$].

- $\pi \in S_n(1324, 2134, 3124, 3214)$.
  Le type des $l+1$ premiers éléments de π est $12 \dots (l+1)$ [resp. $23 \dots (l+1)1$].
  - Insertion dans le premier site.
    L'étiquette de la permutation ainsi obtenue est alors (1) [resp. (1)].

  - Insertion dans l'un des sites actifs autre que les premier, avant-dernier et dernier sites actifs.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [2, l]$.
    L'étiquette de γ est alors $(i-1)$ [resp. $(i-1)$].
  - Insertion dans l'avant-dernier site actif.
    L'étiquette de la permutation ainsi obtenue est alors (l) [resp. $(l+1)$].
  - Insertion dans le dernier site actif.
    L'étiquette de la permutation ainsi obtenue est alors $(l+1)$ [resp. (l)].

- $\pi \in S_n(1324, 2314, 3124, 3214)$.
  - Insertion dans le premier site.
    L'étiquette de la permutation ainsi obtenue est alors (1).
  - Insertion dans l'un des sites actifs autre que le premier site.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [2, l+2]$.
    L'étiquette de γ est alors $(i-1)$.

- $\pi \in S_n(1342, 2341, 3142, 3241)$.
  - Insertion dans le premier site.
    L'étiquette de la permutation ainsi obtenue est alors (1).
  - Insertion dans l'un des sites actifs autre que le premier site.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [2, l+2]$.
    L'étiquette de γ est alors $(i-1)$.

□

Proposition 4.17

- *Le système de réécriture $\mathcal{S}_{GrandDyck4}$ (voir figure 4.7) caractérisant les arbres de génération $T(3124, 3142, 3214, 3241)$ et $T(1324, 1342, 2314, 2341)$ est*

$$\begin{cases} (A,2) \\ (A,t) \rightsquigarrow (B,2),(B,3),\ldots,(B,t-1),(A,t+1),(A,t+1) \\ (B,t) \rightsquigarrow (B,2),(B,3),\ldots,(B,t+1) \end{cases}$$

- *L'étiquette (X,t) du système de réécriture $\mathcal{S}_{GrandDyck4}$ correspondant à une permutation π de l'un quelconque des deux arbres de génération vérifie*
  - *$X = A$ si tous les sites sont actifs,*
  - *$X = B$ sinon,*
  - *t est le nombre de sites actifs de π.*

Lemme 4.18 *Une permutation π de $S_n(3124, 3142, 3214, 3241)$ vérifie les propriétés suivantes.*

(i) Tous les sites sont actifs si et seulement si $\pi^{-1}(n) \geq n-1$.

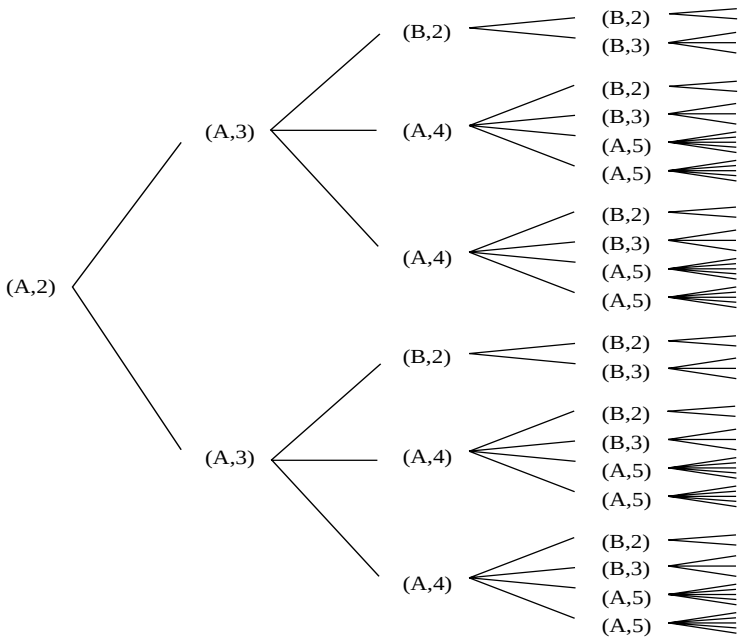


Figure 4.7 Arbre de dérivation du système de réécriture $\mathcal{S}_{GrandDyck4}$.

(ii) *Si $\pi^{-1}(n) \leq n-2$, tous les sites situés à gauche et à droite de $\pi(\pi^{-1}(n)+1)$ sont respectivement actifs et inactifs.*

Preuve D'après la forme des motifs exclus, le site situé à droite de n est actif et aucun des motifs exclus ne peut inactiver le dernier site si $\pi(n-1) = n$.
Ensuite, si $\pi^{-1}(n) \leq n-2$, le site à droite de $\pi(\pi^{-1}(n)+1)$ est inactif car autrement la sous-suite $n\pi(\pi^{-1}(n)+1)(n+1)\pi(n)$ serait de type 3142 ou 3241.
Enfin, tous les sites à droite d'un site inactif sont également inactifs. En effet, d'une part cela est immédiat s'il est inactivé par 3124 ou 3214, et, d'autre part, s'il est inactivé à cause d'une sous-suite $\pi(i_1)\pi(i_2)(n+1)\pi(i_3)$ de type 3142 [resp. 3241], alors la sous-suite $\pi(i_1)\pi(i_2)\pi(i_3)(n+1)$ serait de type 3124 [resp. 3214]. □

Lemme 4.19 *Pour une permutation π de $S_n(1324, 1342, 2314, 2341)$, soit $p \in [0, n-1]$ l'élément maximum tel que $\pi(1)\pi(2)\ldots\pi(p) = n(n-1)\ldots(n+1-p)$. Alors, nous avons les propriétés suivantes.*

(i) *Tous ses sites sont actifs si et seulement si $\pi(n) = n-p$.*

(ii) *Si $t = \pi^{-1}(n-p)$ appartient à $]p+1, n[$, seuls les t premiers sites de π sont actifs.*

Preuve Tout d'abord, tous les sites à droite d'un site inactif sont également inactifs. En effet, d'une part c'est évident s'il est inactivé par 1324 ou 2314, et, d'autre part, s'il est inactivé à cause d'une sous-suite $\pi(i_1)\pi(i_2)(n+1)\pi(i_3)$ de type 1342 [resp. 2341], alors la sous-suite $\pi(i_1)\pi(i_2)\pi(i_3)(n+1)$ serait de type 1324 [resp. 2314].

Ensuite, tous les sites à gauche de $n-p$ sont actifs. Autrement, soit il existerait une sous-suite $\pi(i_1)\pi(i_2)\pi(i_3)(n{+}1)$ de type 1324 ou 2314 avec $i_3 < \pi^{-1}(n-p)$ de sorte que, comme $i_1 > p$, la sous-suite $\pi(i_1)\pi(i_2)\pi(i_3)(n-p)$ serait du même type, soit il existerait une sous-suite $\pi(i_1)\pi(i_2)(n{+}1)\pi(i_3)$ de type 1342 ou 2341 avec $i_2 < \pi^{-1}(n-p)$ de sorte que, comme $i_1 > p$, les sous-suites $\pi(i_1)\pi(i_2)(n-p)\pi(i_3)$ et $\pi(i_1)\pi(i_2)\pi(i_3)(n-p)$ seraient respectivement du même type et de type 1324 ou 2314.
Enfin, les sites à droite de $n{-}p$ sont inactifs si $\pi^{-1}(n{-}p) < n$ puisque la sous-suite $\pi(p{+}1)(n{-}p)(n{+}1)\pi(n)$ est de type 1342 ou 2341 ; par contre, le dernier site est actif si $\pi(n) = n-p$. □

Preuve de la proposition 4.17. La permutation 1 ayant ses deux sites actifs dans les deux arbres de génération des permutations, son étiquette est $(A, 2)$.
Soit π une permutation de $S_n(1342, 2341, 3142, 3241)$ [resp. $S_n(1324, 1342, 2314, 2341)$] d'étiquette (X, t).

- $X = A$.
  - Insertion dans l'un des $n-1$ premiers [resp. du deuxième à l'avant-dernier] sites de π.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ [resp. $(i+1)^{ème}$] site de π, pour tout $i \in [t-2] = [n-1]$.
    L'étiquette de γ est alors $(B, i+1)$.
  - Insertion dans l'avant-dernier [resp. premier] site de π.
    L'étiquette de la permutation ainsi obtenue est alors $(A, t+1)$.
  - Insertion dans le dernier site de π.
    L'étiquette de la permutation ainsi obtenue est alors $(A, t+1)$.
- $X = B$.
  - Insertion dans l'un des $t-1$ premiers [resp. du deuxième au $t^{ème}$] sites de π.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ [resp. $(i+1)^{ème}$] site de π, pour tout $i \in [t-1]$.
    L'étiquette de γ est alors $(B, i+1)$.
  - Insertion dans le dernier [resp. premier] site actif de π.
    L'étiquette de la permutation ainsi obtenue est alors $(B, t+1)$.

□

Proposition 4.20

- *Le système de réécriture $\mathcal{S}_{GrandDyck5}$ (voir figure 4.8) caractérisant l'arbre de génération $T(1342, 2341, 2431, 3241)$ est*

$$\begin{cases} (0,2) \\ (p,t) \quad \rightsquigarrow \quad (t-p-1, t-p+1), \underbrace{(0,2),(0,2),\ldots,(0,2)}_{p \text{ fois}}, (p,p+3),(p,p+4),\ldots,(p,t+1) \end{cases}$$

- *L'étiquette (p,t) du système de réécriture $\mathcal{S}_{GrandDyck5}$ correspondant à une permutation π de l'arbre de génération vérifie*
  - *$p = 0$ si $\pi(1) = 1$,*
  - *$p+1$ est le nombre de sites actifs situés à gauche du dernier élément de π inférieur à $\pi(1)$ sinon,*

– *t est le nombre de sites actifs de π.*

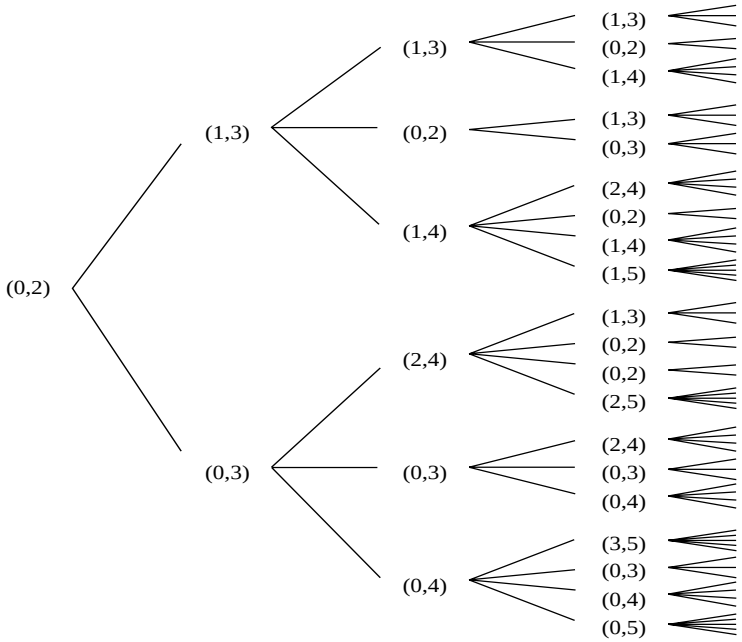


Figure 4.8 Arbre de dérivation du système de réécriture $\mathcal{S}_{GrandDyck5}$.

Lemme 4.21 *Une permutation π de $S_n(1342, 2341, 2431, 3241)$ vérifie les propriétés suivantes.*

(i) Les premier et dernier sites sont actifs.

(ii) Si $\pi(1) \neq n$, tous les sites à droite de n jusqu'à l'avant-dernier sont inactifs.

(iii) A droite du dernier élément de π inférieur à $\pi(1) \neq 1$, un site est actif si et seulement si tous les éléments à sa gauche sont inférieurs à tous ceux à sa droite.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii). Autrement, la sous-suite $\pi(1)n(n+1)\pi(n)$ serait de type 1342 ou 2341.
- (iii).
  Si un tel site est inactif, en raison d'une sous-suite $\pi(i)\pi(j)(n+1)\pi(l)$ de type $1342, 2341, 3241$, ou d'une sous-suite $\pi(j)(n+1)\pi(k)\pi(l)$ de type 2431, alors $\pi(j) > \pi(l)$ et $\pi(j)$ [resp. $\pi(l)$] est à gauche [resp. droite] du site inactif.
  Soient $\pi(j)$ et $\pi(l)$ deux éléments respectivement à gauche et à droite du site considéré, avec $\pi(j) > \pi(l)$; comme $\pi(l) > \pi(1)$, le site est inactif car la sous-suite $\pi(1)\pi(j)(n+1)\pi(l)$ serait de type 1342.

□

Preuve de la proposition 4.20. L'étiquette de la permutation 1 doit être $(0, 2)$.
Soit π une permutation de $S_n(1342, 2341, 2431, 3241)$ d'étiquette (p, t).

- Insertion dans le premier site.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le premier site de π.
  Le deuxième site de γ est actif. Les p sites actifs de π, du deuxième au $(p+1)^{ème}$, deviennent inactifs pour γ à cause de la sous-suite $n\pi(1)(n+1)e$ de type 3241 où e est le dernier élément inférieur à $\pi(1)$. Les $t-p-1$ derniers sites actifs de π restent actifs dans γ.
  L'étiquette de γ est alors $(t-p-1, t-p+1)$.
- Insertion dans l'un des sites actifs à gauche du dernier élément inférieur à $\pi(1)$ autre que le premier site.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [2, p+1]$.
  Les sites à gauche de $n+1$ dans γ, excepté le premier, sont inactifs car la sous-suite $\pi(1)(n+2)(n+1)e$ est de type 2431 où e est le dernier élément inférieur à $\pi(1)$.
  L'étiquette de γ est alors $(0, 2)$.
- Insertion dans l'un des sites actifs à droite du dernier élément inférieur à $\pi(1)$.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [p+2, t]$.
  Les sites à gauche de $n+1$ dans γ sont inchangés car aucune sous-suite n'est de type 2431.
  L'étiquette de γ est alors $(p, i+1)$.

□

Preuve du théorème 4.6. Il résulte des propositions 4.7, 4.9, 4.13 et 4.17 et des opérations de symétrie par miroir et inverse, comme le montre la figure 4.2. Il nous reste à prouver (analytiquement) que $|S_n(1342, 2341, 2431, 3241)| = \binom{2n-2}{n-1}$. Nous déduisons du système de réécriture $\mathcal{S}_{GrandDyck5}$ les récurrences suivantes.

$$\begin{cases} g_{n,(0,2)} & = & \sum_{u=3}^{n} \sum_{q=1}^{u-2} q.g_{n-1,(q,u)} & \text{pour tout } n > 1 \\ g_{n,(t-2,t)} & = & \sum_{u=0}^{n+1-t} g_{n-1,(u,u+t-1)} & \text{pour tout } n > 2 \text{ et } t > 2 \\ g_{n,(p,t)} & = & \sum_{u=t-1}^{n} g_{n-1,(p,u)} & \text{pour tout } n > 2 \text{ et } 0 \leq p < t-2 \\ g_n & = & \sum_{t=2}^{n+1} \sum_{p=0}^{t-2} g_{n,(p,t)} & \text{pour tout } n \geq 1 \\ g_{1,(0,2)} & = & 1 & \\ g_{1,\neq(0,2)} & = & 0 & \end{cases}$$

Des calculs simples permettent de vérifier que $g_{n,(p,t)} = \binom{2n-2-t}{n-3}$ pour tout $n \geq 3$ et $0 \leq p \leq t-2 \leq n-1$ et que $g_n = \binom{2n-2}{n-1}$ pour tout $n \geq 1$. □

4.3 Nombres de Motzkin

Les nombres de Motzkin [75], dont les premières valeurs sont $1, 2, 4, 9, 21, 51, \ldots$, apparaissent dans de nombreux travaux en Combinatoire Enumérative comme par exemple ceux de R. Donaghey [23, 24], R. Donaghey et L.W. Shapiro [25], J. Riordan [82], P. Hanlon [55].

Dans le contexte de nos travaux, S. Gire [45] a montré que les permutations de $S_n(321, 3\overline{1}42)$ sont énumérées par les nombres de Motzkin en établissant une correspondance par isomorphisme d'arbres de génération entre ces permutations et les arbres 1-2 distribués suivant le nombre d'arêtes.

Nous prolongeons ici ces résultats de S. Gire en montrant qu'un autre ensemble de permutations et que deux ensembles d'involutions à motifs exclus sont caractérisés par le même système de réécriture que les arbres 1-2.

Enfin, nous nous intéressons aux permutations vexillaires introduites par A. Lascoux et M.P. Schützenberger [69]. Ces permutations sont exactement celles pour lesquelles les partitions associées aux tables d'inversion (ou codes de Lehmer) de la permutation et de son inverse sont conjuguées. J. West [110] a montré que ces permutations, qui excluent le motif 2143 [72], sont en correspondance avec celles excluant le motif 1234, ce qui a permis de déduire des travaux d'I.M. Gessel [42] une formule les énumérant. Ici, nous conjecturons que les involutions vexillaires sont énumérées par les nombres de Motzkin.

Théorème 4.22 *Les ensembles de permutations à motifs exclus $S_n(321, 3\overline{1}42)$, $S_n(231, 4\overline{1}32)$, et les ensembles d'involutions à motifs exclus $I_n(3412)$, $I_n(4321)$, $I_n(1234)$, sont en correspondance avec les arbres 1-2 ayant n arêtes et les buissons ayant $n-1$ arêtes, et sont donc énumérés par le $n^{ème}$ nombre de Motzkin*

$$\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$$

De plus, les ensembles $I_n(2143)$ et $I_n(1243)$ sont en bijection.

Conjecture 4.23 *Les involutions vexillaires sur $[n]$ sont énumérées par le $n^{ème}$ nombre de Motzkin et nous avons plus précisément*

$$|I_n(2143)| = |I_n(1432)| = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$$

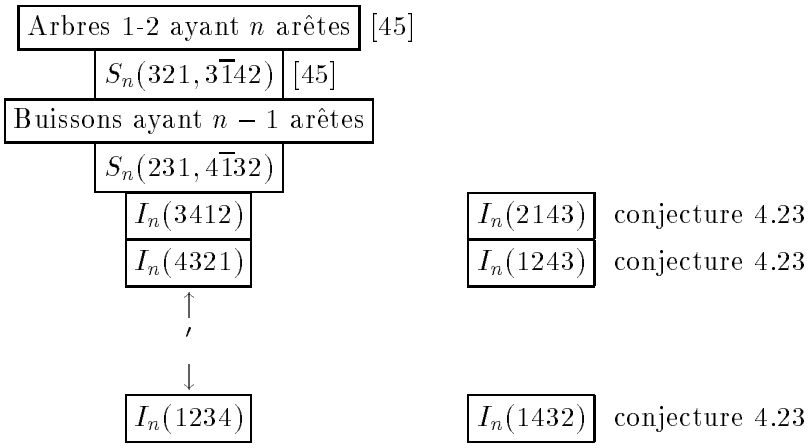


Figure 4.9 Schéma des correspondances entre arbres 1-2, buissons et plusieurs ensembles de permutations et involutions à motifs exclus.

Le schéma général de la preuve qui suit est donné par la figure 4.9 où les ensembles regroupés seront caractérisés par le même système de réécriture.

Proposition 4.24 *(S. Gire [45])*

- *Le système de réécriture $\mathcal{S}_{Motzkin1}$ (voir figure 4.10) caractérisant l'arbre de génération [45] des arbres 1-2 suivant le nombre d'arêtes et l'arbre de génération $T(321, 3\overline{1}42)$ est*
$$\begin{cases} (2) \\ (t) \quad \rightsquigarrow \quad (1),(2),\ldots,(t-1),(t+1) \end{cases}$$
- *L'étiquette (t) du système de réécriture $\mathcal{S}_{Motzkin1}$ correspond au nombre de*
  - *sommets de la branche droite ayant au plus un fils (sommets non doubles) pour l'arbre 1-2,*
  - *sites actifs d'une permutation de l'arbre de génération $T(321, 3\overline{1}42)$.*

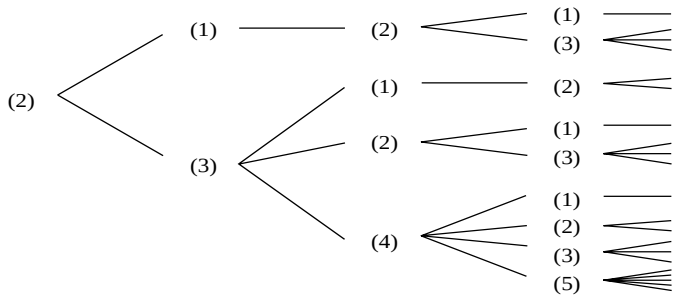


Figure 4.10 Arbre de dérivation du système de réécriture $\mathcal{S}_{Motzkin1}$.

Proposition 4.25

- *Le système de réécriture $\mathcal{S}_{Motzkin1}$ caractérise l'arbre de génération $T(231, 4\overline{1}32)$ et les arbres de génération des involutions par la méthode des points fixes de $I_n(3412)$ et de $I_n(4321)$.*
- *L'étiquette (t) du système de réécriture $\mathcal{S}_{Motzkin1}$ correspond au nombre de*
  - *sites actifs dans le cas des permutations de $S_n(231, 4\overline{1}32)$,*
  - *points fixes actifs pour les involutions de $I_n(3412)$ et $I_n(4321)$.*

Lemme 4.26 *Une permutation π de $S_n(231, 4\overline{1}32)$ vérifie les propriétés suivantes.*

(i) Le dernier site est actif.

(ii) Le $k^{ème}$ site est actif si et seulement si $\pi(k) = k$ et tous les éléments à sa gauche [resp. droite] sont inférieurs [resp. supérieurs] à k.

Preuve

- (i) résulte de la forme des motifs exclus.

- (ii).
  Si le $k^{ème}$ site est actif, comme le motif 231 est interdit, nous avons $\{\pi(k), \pi(k+1), \ldots, \pi(n)\} = [k, n]$. Si $\pi(k) > k$, alors la permutation obtenue en activant le site situé à gauche de $\pi(k)$ contiendrait la sous-suite $(n+1)\pi(k)k$ de type 321 ne faisant pas elle-même partie d'une sous-suite de type 4132.
  Si $\pi(k) = k$ et $\{\pi(1), \pi(2), \ldots, \pi(k-1)\} = [k-1]$, le motif 231 ne peut clairement pas inactiver le $k^{ème}$ site. Il en est de même pour le motif $4\overline{1}32$ car toute sous-suite $(n+1)\pi(i_1)\pi(i_2)$ de type 321 avec $k < i_1$ fait partie d'une sous-suite $(n+1)k\pi(i_1)\pi(i_2)$ de type 4132.

□

Lemme 4.27 *Une involution π de $I_n(3412)$ [resp. $I_n(4321)$] vérifie les propriétés suivantes.*

(i) L'involution $\pi(1)\pi(2)\ldots\pi(n)(n+1)$ appartient à $I_{n+1}(3412)$ [resp. $I_{n+1}(4321)$].

(ii) Un point fixe j [resp. i] de π est actif si et seulement s'il n'existe pas de cycle (i,k) [resp. (j,k)] de π avec $i < j < k$.

Preuve

- (i) résulte de la forme du motif exclu.
- (ii). En effet, l'activation du point fixe j [resp. i] engendrerait la sous-suite $k(n+1)ij$ [resp. $(n+1)kji$] de type 3412 [resp. 4321].

□

Remarque 4.28 *Une correspondance directe entre les involutions de $I_n(3412)$ et les mots du langage de Motzkin $P_{x,\overline{x}} \sqcup\!\sqcup \{y\}^*$ s'obtient en codant un point fixe par la lettre y et le début et la fin d'un cycle par respectivement les lettres x et $\overline{x}$, deux cycles ne pouvant se chevaucher.*

Preuve de la proposition 4.25. Tout d'abord, la permutation ou l'involution 1 a clairement pour étiquette (2).
Soit π une permutation ou une involution de l'un quelconque des arbres de génération ayant pour étiquette (t).

- $\pi \in S_n(231, 4\overline{1}32)$.
  - Insertion dans l'un des sites actifs autre que le dernier site.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [t-1]$.
    L'étiquette de γ est alors (i).
  - Insertion dans le dernier site.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.
- $\pi \in I_n(3412)$ [resp. $I_n(4321)$].
  - Transformation d'un point fixe actif en un cycle.
    Soit γ l'involution obtenue en transformant le $i^{ème}$ point fixe actif de π en un cycle avec l'élément $n+1$, pour tout $i \in [t-1]$.
    L'étiquette de γ est alors (i) [resp. $(t-i)$].

– Ajout d'un point fixe.
L'étiquette de l'involution ainsi obtenue est alors $(t+1)$ [resp. $(t+1)$].

□

Proposition 4.29 *Un arbre de génération des mots de* $P_{x,\overline{x}}\backslash\{\{x,\overline{x}\}^* x x w \overline{x}\overline{x} \{x,\overline{x}\}^* : w \in P_{x,\overline{x}}\}$ *codant les buissons (voir figure 4.11) est*

$$\begin{cases} x\overline{x}x\overline{x} \\ w_1 w_2 \ldots\ldots w_t \quad \rightsquigarrow \quad w_1 w_2 \ldots w_{i-1} x w_i w_{i+1} \ldots w_t \overline{x} \text{ pour tout } i \in [t-1] \cup \{t+1\} \end{cases}$$

où les w_i *sont des mots de parenthèses premiers, pour tout* $i \in [t]$.

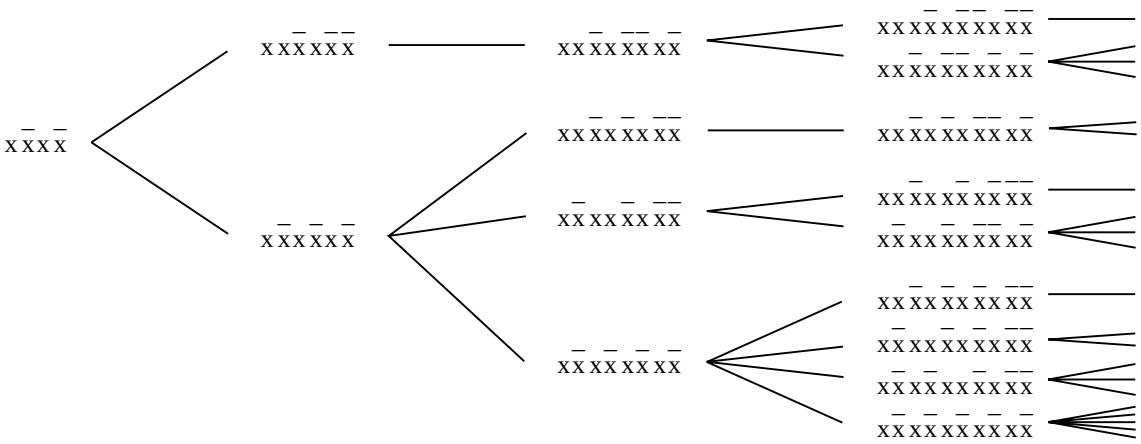


Figure 4.11 Arbre de génération des mots codant les buissons.

Cet arbre de génération des buissons, du fait de sa construction, est clairement caractérisé par le système de réécriture $\mathcal{S}_{Motzkin1}$.

Nous allons maintenant caractériser les ensembles d'involutions $I_n(2143)$ et $I_n(1243)$ à l'aide des deux systèmes de réécriture suivants.

Lemme 4.30 *Le système de réécriture* $\mathcal{S}_{Motzkin2}$ *(voir figure 4.12) donné par*

$$\begin{cases} (1) & & \\ (t) & \rightsquigarrow & (t+1) \\ & \overset{2}{\rightsquigarrow} & (2,t+1),(3,t+1),\ldots,(t+1,t+1) \\ (p,t) & \rightsquigarrow & (p,p) \\ & \overset{2}{\rightsquigarrow} & (2,t+1),(3,t+1),\ldots,(p+1,t+1),(p,t),(p,t-1),\ldots,(p,p+1) \end{cases}$$

et le système de réécriture $\mathcal{S}_{Motzkin2'}$ *(voir figure 4.13) défini par*

$$\begin{cases} (1) & \\ (1) & \rightsquigarrow \quad (2) \\ & \overset{2}{\rightsquigarrow} \quad (3) \\ (t) & \rightsquigarrow \quad (t,t) \\ & \overset{2}{\rightsquigarrow} \quad (t+2),(2,t+1),(3,t+1),\ldots,(t,t+1) \text{ pour tout } t \geq 2 \\ (p,t) & \rightsquigarrow \quad (p,p) \\ & \overset{2}{\rightsquigarrow} \quad (p+1,t+1),(2,t+1),(3,t+1),\ldots,(p,t+1),(p,t),(p,t-1),\ldots,(p,p+1) \end{cases}$$

sont équivalents dans le sens où les arbres de dérivation qu'ils engendrent possèdent des étiquettes identiques à chaque niveau.

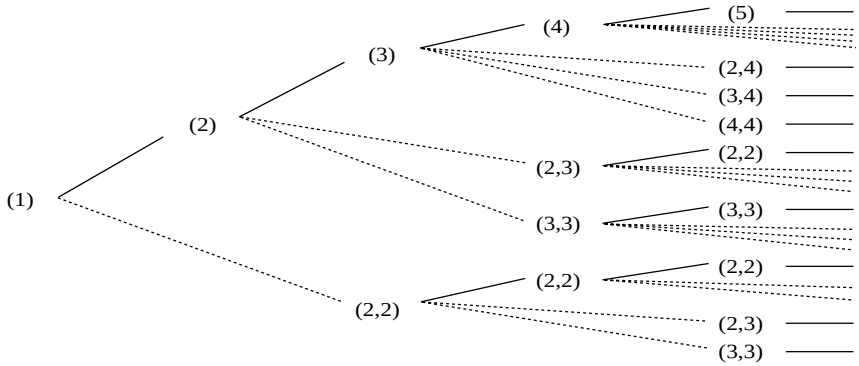


Figure 4.12 Arbre de dérivation du système de réécriture $\mathcal{S}_{Motzkin2}$.

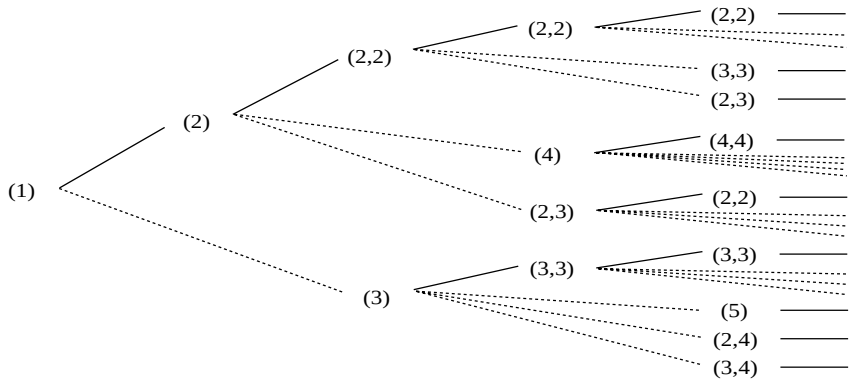


Figure 4.13 Arbre de dérivation du système de réécriture $\mathcal{S}_{Motzkin2'}$.

Preuve La transformation de $\mathcal{S}_{Motzkin2}$ en $\mathcal{S}_{Motzkin2'}$ s'effectue en remplaçant pour tout $x \geq 1$

- la règle $(x+1) \rightsquigarrow (x+2)$ par la règle $(x) \overset{2}{\rightsquigarrow} (x+2)$,
- la règle $(x) \overset{2}{\rightsquigarrow} (x+1,x+1)$ par la règle $(x+1) \rightsquigarrow (x+1,x+1)$.

Les autres règles sont conservées. □

Proposition 4.31

- *L'arbre de génération des involutions de $I_n(2143)$ [resp. $I_n(1243)$] par la méthode récurrente est caractérisé par le système de réécriture $\mathcal{S}_{Motzkin2}$ [resp. $\mathcal{S}_{Motzkin2'}$].*
- *Les étiquettes (t) et (p,t) du système de réécriture $\mathcal{S}_{Motzkin2}$ [resp. $\mathcal{S}_{Motzkin2'}$] correspondant à une involution π sur $[n]$ de l'arbre de génération vérifient*
  - *$p = \min\{d : \pi(d-1) > \pi(d)\}$ [resp. $\min\{m : \pi(m-1) < \pi(m)\}$] pour $\pi \in I_n(2143)$ [resp. $I_n(1243)$] et $\pi \neq 12\ldots n$ [resp. $n(n-1)\ldots 1$],*
  - *t est le nombre de sites actifs de π, c'est à dire le nombre d'involutions respectivement de $I_{n+2}(2143)$ [resp. $I_{n+2}(1243)$] obtenues à partir de π.*

Lemme 4.32 *Une involution π de $I_n(2143)$ [resp. $I_n(1243)$] d'étiquette (t) ou (p,t) définie conformément à la proposition 4.31 vérifie les propriétés suivantes.*

(i) L'involution $\pi(1)\pi(2)\ldots\pi(n)(n+1)$ appartient à $I_{n+1}(2143)$ [resp. $I_{n+1}(1243)$].

(ii) Les sites de l'involution $12\ldots n$ [resp. $n(n-1)\ldots 1$], qui appartient à $I_n(2143)$ [resp. $I_n(1243)$] pour tout $n \geq 0$, sont tous actifs.

(iii) L'involution $\pi \neq 12\ldots n$ [resp. $n(n-1)\ldots 1$] a pour étiquette (p,t). Ses p premiers sites sont actifs et le dernier site est inactif.

(iv) Si $\pi \neq 12\ldots n$ [resp. $n(n-1)\ldots 1$] a pour étiquette (p,t), l'ajout du point fixe $(n+1)$ ou l'insertion d'un cycle $(k, n+2)$ avec $k > p$ dans l'un des sites actifs de π inactive, dans l'involution γ ainsi obtenue, tous les sites situés entre $\gamma(p)$ et $n+1$ ou entre $\gamma(p)$ et $n+2$. Si $\pi \neq 12\ldots n$ [resp. $n(n-1)\ldots 1$], l'insertion d'un cycle $(k, n+2)$ dans l'un des sites actifs de π laisse dans le même état tous les sites situés à droite de $n+2$ dans l'involution γ ainsi obtenue.

Preuve

- (i) et (ii) résultent de la forme du motif exclu.
- (iii).
  - Etudions l'involution sur $[n+2]$ obtenue en insérant le couple $(i, n+2)$ avec $i \in [p]$ en incrémentant d'une unité les éléments de π supérieurs ou égaux à i.
    Les éléments à gauche de l'élément $n+2$ forment toujours une sous-suite croissante [resp. décroissante] ce qui interdit à $n+2$ de jouer le rôle de 4 dans le motif 2143 [resp. 1243].
    Par définition, nous avons $\pi(1) < \pi(2) < \cdots < \pi(p-1) > \pi(p)$ [resp. $\pi(1) > \pi(2) > \cdots > \pi(p-1) < \pi(p)$]. Comme π est une involution, nous avons $\pi^{-1}(1) < \pi^{-1}(2) < \cdots < \pi^{-1}(p-1) > \pi^{-1}(p)$ [resp. $\pi^{-1}(1) > \pi^{-1}(2) > \cdots > \pi^{-1}(p-1) < \pi^{-1}(p)$]. Cela interdit à i de jouer le rôle de 3 dans le motif 2143 [resp. 1243] puisque tous les éléments inférieurs à i forment une sous-suite croissante [resp. décroissante].

- Le dernier site est inactif car autrement la sous-suite $\pi(1)\pi(p)(n+2)(n+1)$ serait de type 2143 [resp. 1243].

- (iv).
  - En effet, la sous-suite $\gamma(1)\gamma(p)(n+1)$ ou $\gamma(1)\gamma(p)(n+2)$ de type 213 [resp. 123] et inactive les sites à droite de $\gamma(p)$ et ceux à gauche de $n+1$ ou $n+2$.
  - Si le $l^{ème}$ site de γ avec $l > k$ est inactif, le $(l-1)^{ème}$ site de π l'est également. En effet, le seul cas à considérer serait l'existence d'une sous-suite $\gamma^+(i_1)\gamma^+(i_2)(n+4)k$ ou $\gamma^+(i_1)\gamma^+(i_2)(n+3)l$ avec $i_2 < l$ ou $i_2 < k$ qui soit de type 2143 [resp. 1243], où $\gamma^+(j) = \gamma(j)$ ou $\gamma(j)+1$ selon que $\gamma(j) < l$ ou $\geq l$. Dans ce cas, la sous-suite $\gamma(i_1)\gamma(i_2)(n+2)(l-1)$ serait du même type.

□

Preuve de la proposition 4.31. Clairement, l'étiquette de l'involution ε est (1).
Soit π une involution de l'un des arbres de génération par la méthode récurrente.

- $\pi \in I_n(2143)$.
  - $\pi = 12\ldots n$ a pour étiquette (t) avec $t = n+1$.
    - Ajout d'un point fixe engendrant l'involution $12\ldots(n+1)$.
      L'étiquette de l'involution ainsi obtenue est alors $(t+1)$.
    - Insertion dans l'un des sites actifs.
      Soit γ l'involution obtenue en insérant le cycle $(i, n+2)$ dans le $i^{ème}$ site de π, pour tout $i \in [t]$.
      L'étiquette de γ est alors $(i+1, t+1)$.
  - $\pi \neq 12\ldots n$ a pour étiquette (p,t).
    - Ajout d'un point fixe engendrant l'involution $\pi(1)\pi(2)\ldots\pi(n)(n+1)$.
      L'étiquette de l'involution ainsi obtenue est alors (p,p).
    - Insertion dans l'un des p premiers sites.
      Soit γ l'involution obtenue en insérant le cycle $(i, n+2)$ dans le $i^{ème}$ site de π, pour tout $i \in [p]$.
      L'étiquette de γ est alors $(i+1, t+1)$.
    - Insertion dans l'un des $t-p$ derniers sites actifs.
      Soit γ l'involution obtenue en insérant le cycle $(i, n+2)$ dans le $i^{ème}$ site actif de π, pour tout $i \in [p+1, t]$.
      L'étiquette de γ est alors $(p, p+1+t-i)$.

- $\pi \in I_n(1243)$.
  - $\pi = \varepsilon$ a pour étiquette (1).
    Elle permet d'engendrer les involutions 1 d'étiquette (2) et 21 d'étiquette (3).
  - $\pi = n(n-1)\ldots 1$ a pour étiquette (t) avec $t = n+1$, pour tout $n > 0$.
    - Ajout d'un point fixe engendrant l'involution $n(n-1)\ldots 1(n+1)$.
      L'étiquette de l'involution ainsi obtenue est alors (t,t).
    - Insertion dans le premier site (engendrant l'involution $(n+2)(n+1)\ldots 1$).
      L'étiquette de l'involution ainsi obtenue est alors $(t+2)$.
    - Insertion dans l'un des sites actifs autre que le premier site.
      Soit γ l'involution obtenue en insérant le cycle $(i, n+2)$ dans le $i^{ème}$ site de π, pour tout

$i \in [2, t]$.
L'étiquette de γ est alors $(i, t+1)$.

- $\pi \neq n(n-1)\ldots1$ a pour étiquette (p, t).
  - Ajout d'un point fixe engendrant l'involution $\pi(1)\pi(2)\ldots\pi(n)(n+1)$.
    L'étiquette de l'involution ainsi obtenue est alors (p, p).
  - Insertion dans le premier site (engendrant l'involution $(n+2)(\pi(1)+1)(\pi(2)+1)\ldots(\pi(n)+1)1$).
    L'étiquette de l'involution ainsi obtenue est alors $(p+1, t+1)$.
  - Insertion dans l'un des p premiers sites autre que le premier site.
    Soit γ l'involution obtenue en insérant le cycle $(i, n+2)$ dans le $i^{ème}$ site de π, pour tout $i \in [2, p]$.
    L'étiquette de γ est alors $(i, t+1)$.
  - Insertion dans l'un des $t-p$ derniers sites actifs.
    Soit γ l'involution obtenue en insérant le cycle $(i, n+2)$ dans le $i^{ème}$ site actif de π, pour tout $i \in [p+1, t]$.
    L'étiquette de γ est alors $(p, p+1+t-i)$.

□

Preuve du théorème 4.22. Il résulte des propositions 4.24, 4.25, 4.29 et 4.31.
De plus, les ensembles d'involutions $I_n(4321)$ et $I_n(1234)$ sont en bijection puisque les tableaux de Young standard correspondant à ces involutions par l'algorithme de Robinson-Schensted sont transposés l'un de l'autre. □

4.4 Nombres de Schröder

Les nombres de Schröder [90], dont les premières valeurs sont $1, 2, 6, 22, 90, 394, \ldots$, sont liés à l'énumération de nombreux objets combinatoires classiques. Ils apparaissent dans les travaux de D.E. Knuth [62], G. Kreweras [66], D.G. Rogers [84], D.G. Rogers et L.W. Shapiro [86, 87], D. Gouyou-Beauchamps et B. Vauquelin [50], L.W. Shapiro et A.B. Stephens [94]. Notons que, parfois, les nombres $1, 3, 11, 45, 197, \ldots$ sont également appelés nombres de Schröder.

Récemment, J. West [110, 112] a montré que les permutations excluant les deux motifs 2413 et 3142 sont énumérés par les nombres de Schröder. S. Gire [45] a pour sa part prolongé ce résultat en montrant que ces permutations sont en correspondance avec celles excluant les deux motifs 3124 et 3214 et avec les arbres 1-2 distribués suivant le nombre de sommets internes. En fait, les arbres de génération de ces trois ensembles sont caractérisés par le même système de réécriture.
Nous montrons que huit nouveaux ensembles de permutations à motifs exclus ont des arbres de génération également caractérisés par ce système de réécriture, et sont donc dénombrés par les nombres de Schröder.

Théorème 4.33 *Les ensembles de permutations à motifs exclus* $S_n(1234, 2134)$, $S_n(1324, 2134)$, $S_n(1324, 2314)$, $S_n(2134, 3124)$, $S_n(2314, 3124)$, $S_n(1342, 2341)$, $S_n(3142, 3241)$, $S_n(3412, 3421)$,

$S_n(2413,3142)$ et $S_n(3124,3214)$ sont en bijection avec les arbres 1-2 ayant $n-1$ sommets internes et sont donc énumérés par le $(n-1)^{ème}$ nombre de Schröder

$$\sum_{i=0}^{n-1} \binom{n-1+i}{n-1-i} c_i$$

Proposition 4.34 *(J. West [110, 112])*

- *Le système de réécriture $\mathcal{S}_{Schröder}$ (voir figure 4.14) caractérisant l'arbre de génération $T(2413,3142)$ est*
$$\begin{cases} (2) \\ (t) \quad \rightsquigarrow \quad (3),(4),\ldots,(t+1),(t+1) \end{cases}$$
- *L'étiquette (t) du système de réécriture $\mathcal{S}_{Schröder}$ correspond au nombre de sites actifs de la permutation associée dans l'arbre de génération.*

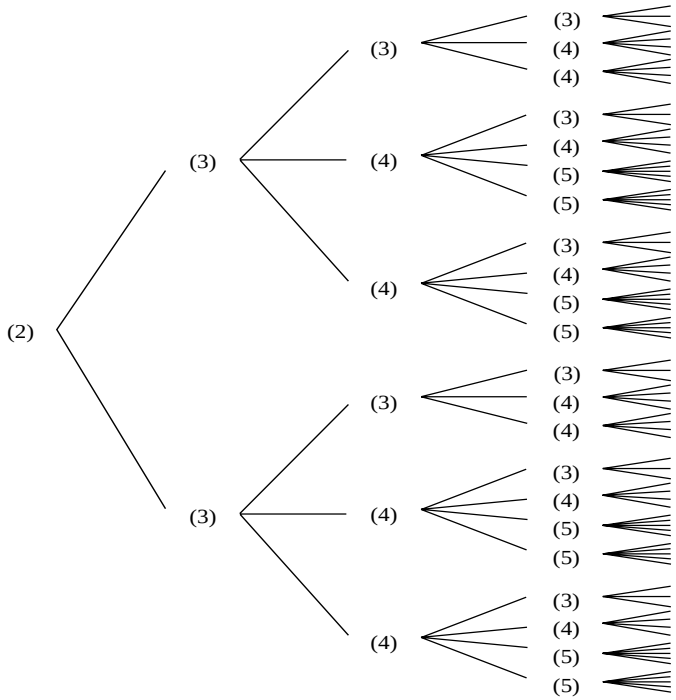


Figure 4.14 Arbre de dérivation du système de réécriture $\mathcal{S}_{Schröder}$ caractérisant notamment les arbres 1-2 suivant le nombre de sommets internes.

Proposition 4.35 *(S. Gire [45])*

- *Le système de réécriture $\mathcal{S}_{Schröder}$ caractérise l'arbre de génération [45] des arbres 1-2 suivant le nombre de sommets internes et l'arbre de génération $T(3124,3214)$.*
- *L'étiquette (t) du système de réécriture $\mathcal{S}_{Schröder}$ correspond au nombre de*

- *mots premiers du mot de Schröder (ou mot de Motzkin) codant l'arbre 1-2 augmenté de 2,*
- *sites actifs de la permutation associée dans l'arbre de génération.*

Proposition 4.36

- *Le système de réécriture $\mathcal{S}_{Schröder}$ caractérise les arbres de génération $T(1234,2134)$, $T(1324,2134)$, $T(1324,2314)$, $T(2134,3124)$, $T(2314,3124)$, $T(1342,2341)$, $T(3142,3241)$ et $T(3412,3421)$.*
- *L'étiquette (t) du système de réécriture $\mathcal{S}_{Schröder}$ correspond au nombre de sites actifs de la permutation associée pour l'un quelconque des huit arbres de génération.*

Lemme 4.37 *Pour une permutation de $S_n(1234,2134)$, $S_n(1324,2134)$, $S_n(1324,2314)$, $S_n(2134,3124)$, $S_n(2314,3124)$, $S_n(1342,2341)$, $S_n(3142,3241)$, $S_n(3412,3421)$, l'insertion de l'élément $n+1$ dans l'un des sites actifs laisse dans le même état tous les sites situés à gauche de $n+1$ dans la permutation obtenue.*

Preuve En effet, tous les motifs exclus sont des permutations de S_4 pour lesquelles l'élément 3 est situé à gauche de l'élément 4. □

Lemme 4.38 *Si une permutation π appartenant à $S_n(1234,2134)$ [resp. $S_n(1324,2134)$, $S_n(1324,2314)$, $S_n(2134,3124)$, $S_n(2314,3124)$] a t sites actifs dans l'arbre de génération correspondant, alors ses t premiers sites sont actifs et le type la sous-suite $\pi(1)\pi(2)\dots\pi(t-1)$ appartient à $S_{t-1}(123,213)$ [resp. $S_{t-1}(132,213)$, $S_{t-1}(132,231)$, $S_{t-1}(213,312)$, $S_{t-1}(231,312)$]. De plus, pour une permutation de $S_n(1324,2314)$ ou $S_n(2314,3124)$, le site situé à droite de n est actif.*

Preuve Ce résultat se déduit clairement de la forme des motifs exclus dont le dernier élément 4 est aussi le plus grand élément. □

Remarque 4.39

(i) $S_n(123,213) = \{\pi : \forall \pi(i) \in [2,n], |\{j : j<i, \pi(j)<\pi(i)\}| \in \{0,1\}\}$.

(ii) $S_n(132,213) = \{\pi : \exists\ 0 = e_0 < e_1 < \cdots < e_l = n$ *tel que*
$\pi = (e_{l-1}+1)(e_{l-1}+2)\dots e_l(e_{l-2}+1)(e_{l-2}+2)\dots e_{l-1}\dots(e_0+1)(e_0+2)\dots e_1\}$.

(iii) $S_n(132,231) = \{\pi : 1 \le j < i = \pi^{-1}(1) < k \le n, \pi(j) > \pi(j+1)$ *et* $\pi(k-1) < \pi(k)\}$.

(iv) $S_n(213,312) = \{\pi : 1 \le j < i = \pi^{-1}(n) < k \le n, \pi(j) < \pi(j+1)$ *et* $\pi(k-1) > \pi(k)\}$.

(v) $S_n(231,312) = \{\pi : \exists\ 0 = e_0 < e_1 < \cdots < e_l = n$ *tel que*
$\pi = e_1(e_1-1)\dots(e_0+1)e_2(e_2-1)\dots(e_1+1)\dots e_l(e_l-1)\dots(e_{l-1}+1)\}$.

Preuve R. Simion et F.W. Schmidt [95] ont montré que $|S_n(123,213)| = |S_n(132,213)| = |S_n(132,231)| = |S_n(213,312)| = |S_n(231,312)| = 2^{n-1}$ pour tout $n \geq 1$. Afin de n'étudier que les cas utiles, remarquons que $\{132,231\} = \{213,312\}^c$ et $\{132,213\} = \{231,312\}^*$. Ensuite, notons que les arbres de génération des permutations $T(123,213)$, $T(132,213)$, $T(132,231)$ sont caractérisés par le système de réécriture $\begin{cases} (2) & \\ (2) & \rightsquigarrow \quad (2),(2) \end{cases}$ pour lequel l'étiquette correspond au nombre de sites actifs; plus précisément, les deux sites actifs sont respectivement les deux premiers, le premier et celui à droite de n, le premier et le dernier. Nous en déduisons alors la caractérisation des permutations à motifs exclus des ensembles correspondants. □

Lemme 4.40 *Une permutation de $S_n(1342,2341)$ vérifie les propriétés suivantes.*

(i) Les deux premiers et le dernier sites sont actifs.

(ii) Si $\pi(1) \neq n$, les sites à droite de n jusqu'à l'avant-dernier sont inactifs.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii). Autrement, la sous-suite $\pi(1)n(n+1)\pi(n)$ serait de type 1342 ou 2341.

□

Lemme 4.41 *Une permutation de $S_n(3142,3241)$ vérifie les propriétés suivantes.*

(i) Les deux premiers sites, celui situé à droite de n et le dernier sont actifs.

(ii) Tous les autres sites à droite de n sont inactifs.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii). Autrement, la sous-suite $n\pi(\pi^{-1}(n)+1)(n+1)\pi(n)$ serait de type 3142 ou 3241.

□

Lemme 4.42 *Une permutation de $S_n(3412,3421)$ vérifie les propriétés suivantes.*

(i) Le premier et les deux derniers sites sont actifs.

(ii) Les sites à droite de n jusqu'à l'antépénultième sont inactifs.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii). Autrement, la sous-suite $n(n+1)\pi(n-1)\pi(n)$ serait de type 3412 ou 3421.

□

Preuve de la proposition 4.36. La permutation 1 ayant ses deux sites actifs, pour chacun des ensembles considérés, son étiquette est (2).
Soit π une permutation de l'un quelconque des arbres de génération des permutations ayant pour étiquette (t).

- $\pi \in S_n(1234, 2134)$.
  - Insertion dans l'un des deux premiers sites.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.
  - Insertion dans l'un des sites actifs autre que les deux premiers sites.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [3, t]$.
    L'étiquette de γ est alors (i).

- $\pi \in S_n(1324, 2134)$.
  Soit $k = \pi(j) = \max\{\pi(1), \pi(2), \ldots, \pi(t-1)\}$.
  - Insertion dans le premier site ou dans le site situé à droite de k.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.
  - Insertion dans l'un des sites à gauche de k autre que le premier site.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [2, j]$.
    L'étiquette de γ est alors $(i+1)$.
  - Insertion dans l'un des sites actifs situé à droite de k autre que le premier d'entre eux.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [j+2, t]$.
    L'étiquette de γ est alors (i).

- $\pi \in S_n(1324, 2314)$.
  - Insertion dans le premier site ou dans le dernier site actif.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.
  - Insertion dans l'un des sites actifs autre que les premier et dernier sites actifs.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [2, t-1]$.
    L'étiquette de γ est alors $(i+1)$.

- $\pi \in S_n(2134, 3124)$.
  Soit $k = \pi(j) = \max\{\pi(1), \pi(2), \ldots, \pi(t-1)\}$.
  - Insertion dans l'un des sites à gauche de k autre que le dernier d'entre eux.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [j-1]$.
    L'étiquette de γ est alors $(i+2)$.
  - Insertion dans l'un des sites entourant k.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.
  - Insertion dans l'un des sites actifs à droite de k autre que le premier d'entre eux.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site de π, pour tout $i \in [j+2, t]$.
    L'étiquette de γ est alors (i).

- $\pi \in S_n(2314, 3124)$.
  Reprenons les notations de la remarque 4.39 selon lesquelles la sous-suite $\pi(1)\pi(2)\ldots\pi(t-1)$ est

de type $\tau = e_1(e_1-1)\dots(e_0+1)e_2(e_2-1)\dots(e_1+1)\dots e_l(e_l-1)\dots(e_{l-1}+1)$ avec $0 = e_0 < e_1 < \dots < e_l = t-1$.

- Insertion dans le premier site ou dans l'un des sites actifs correspondant à une montée, autre que le dernier site actif.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $(e_j+1)^{ème}$ site de π (c'est à dire dans le site situé à gauche de e_{j+1} relativement à τ), pour tout $j \in [0, l-1]$.
  L'étiquette de γ est alors $(e_{j+1}+2)$.
- Insertion dans l'un des sites actifs correspondant à une descente autre que le dernier site actif.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $(e_j+i)^{ème}$ site de π (c'est à dire dans l'un des sites situés entre e_{j+1} et e_j+1 relativement à τ), pour tout $j \in [0, l-1]$ et pour tout $i \in [2, e_{j+1}-e_j]$.
  L'étiquette de γ est alors (e_j+i+1).
- Insertion dans le dernier site actif.
  L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.

Notons que les insertions dans les $t-1$ premiers sites engendrent chacune des étiquettes $(3), (4), \dots, (t+1)$ exactement une fois puisque les étiquettes générées vont de e_0+3 à e_l+2.

- $\pi \in S_n(1342, 2341)$.
  - Insertion dans le premier ou dernier site.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.
  - Insertion dans l'un des sites actifs autre que les premier et dernier sites.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [2, t-1]$.
    L'étiquette de γ est alors $(i+1)$.

- $\pi \in S_n(3142, 3241)$.
  - Insertion dans l'un des sites actifs autre que les deux derniers sites actifs.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [t-2]$.
    L'étiquette de γ est alors $(i+2)$.
  - Insertion dans l'avant-dernier des sites actifs ou dans le dernier site.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.

- $\pi \in S_n(3412, 3421)$.
  - Insertion dans l'un des sites actifs autre que les deux derniers sites.
    Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [t-2]$.
    L'étiquette de γ est alors $(i+2)$.
  - Insertion dans l'un des deux derniers sites.
    L'étiquette de la permutation ainsi obtenue est alors $(t+1)$.

□

Preuve du théorème 4.33. Il résulte des propositions 4.34, 4.35 et 4.36. □

4.5 Systèmes de réécriture pour les tableaux de Young standard bornés

L'algorithme de Robinson-Schensted [83, 89] donne une correspondance entre permutations sur $[n]$ et paires de tableaux de Young standard de même forme $\lambda \vdash n$. Cette correspondance met en évidence de nombreuses propriétés et permet en particulier la lecture de certains paramètres des permutations sur les tableaux associés.
Par exemple, la longueur de la plus longue sous-suite croissante [resp. décroissante] d'une permutation est la longueur [resp. hauteur] des tableaux correspondants [89]. Rappelons que des résultats plus fins, caractérisant exactement la forme des tableaux, ont été obtenus par C. Greene [51]. Pour sa part, M.P. Schützenberger [92] a montré que lorsque la permutation considérée est une involution, les deux tableaux obtenus par l'algorithme de Robinson-Schensted sont identiques.

Ainsi, s'intéresser aux paires de tableaux de même forme et aux tableaux de Young standard de longueur [resp. hauteur] au plus k revient à considérer respectivement les permutations et les involutions excluant le motif identité $12\ldots(k+1)$ [resp. miroir de l'identité $(k+1)k\ldots1$]. En effet, l'exclusion de ces motifs consiste à interdire des sous-suites croissantes [resp. décroissantes] de longueur $k+1$.

Ce problème, que nous abordons ici sous une forme purement combinatoire, a été considéré dans la thèse d'I. Schur. Depuis, de nombreux auteurs se sont intéressés au dénombrement des paires de tableaux de Young standard de hauteur bornée [73, 62, 85, 95, 42, 110] et des tableaux de Young standard de hauteur bornée [79, 47, 48, 49, 99, 118, 37]. Nous pouvons en particulier citer les résultats asymptotiques obtenus par A. Regev [79], ainsi que les conjectures énoncées par F. Bergeron, L. Favreau et D. Krob [6].

Notre apport consiste ici à établir des systèmes de réécriture caractérisant les arbres de génération des permutations de $S_n(12\ldots(k+1))$ et des involutions de $I_n(12\ldots(k+1))$, et à en déduire des équations de récurrence dans le cas des permutations de $S_n(12\ldots(k+1))$. Malheureusement, leur complexité ne nous a pas permis de pouvoir les exploiter.

Avant de présenter notre contribution à ce problème, rappelons tout d'abord les résultats connus dans ce cadre.

$\lvert S_n(123)\rvert = c_n$	P.A. MacMahon [73]
$\lvert S_n(1234)\rvert = 2\sum_{k=0}^{n}\binom{2k}{k}\binom{n}{k}^2\frac{3k^2+2k+1-n-2kn}{(k+1)^2(k+2)(n-k+1)}$	I.M. Gessel [42]
$\lvert I_n(123)\rvert = \binom{n}{\lfloor\frac{n}{2}\rfloor}$	
$\lvert I_n(1234)\rvert = \sum_{i=0}^{\lfloor\frac{n}{2}\rfloor}\binom{n}{2i}c_i$	A. Regev [79]
$\lvert I_n(12345)\rvert = c_{\lfloor\frac{n+1}{2}\rfloor}.c_{\lceil\frac{n+1}{2}\rceil}$	D. Gouyou-Beauchamps [49]
$\lvert I_n(123456)\rvert = 6\sum_{i=0}^{\lfloor\frac{n}{2}\rfloor}\binom{n}{2i}.c_i.\frac{(2i+2)!}{(i+2)!(i+3)!}$	D. Gouyou-Beauchamps [49]

4.5.1 Paires de tableaux de Young standard de hauteur bornée

J. West [110] s'est intéressé aux arbres de génération $T(123)$ et $T(1234)$. Nous généralisons ses résultats en donnant une caractérisation de l'arbre de génération $T(12\ldots(k+1))$, et ce pour tout entier k.

Définition 4.43 *Nous appelons $i^{ème}$* suite des minima *d'une permutation π la suite constituée des minima à gauche de la sous-suite obtenue en supprimant de π les éléments des $i-1$ premières suites des minima.*
Nous notons p_i l'indice dans σ du premier élément constituant la $i^{ème}$ suite des minima de σ, avec $p_i = n+1$ si si cette suite est vide, et adoptons par convention le fait que $p_0 = 0$.

Exemple 4.44 *Les première, deuxième et troisième suites des minima de la permutation $\pi =$ 547298136 sont respectivement les suites* 5421, 73 *et* 986. *Nous avons alors $p_0 = 0$, $p_1 = 1 = \pi^{-1}(5)$, $p_2 = \pi^{-1}(7) = 3$, $p_3 = \pi^{-1}(9) = 5$ et $p_i = 10$ pour tout $i \geq 4$.*

Proposition 4.45 *Soit π une permutation de S_n admettant exactement k suites des minima. Alors, la plus longue sous-suite croissante de π est exactement de longueur j.*

Preuve Soit y un élément de la $(q+1)^{ème}$ suite des minima de π. Alors, il existe un élément x appartenant à la $q^{ème}$ suite des minima tel que $x < y$ et $\pi^{-1}(x) < \pi^{-1}(y)$. Ainsi, π possède une sous-suite croissante de longueur k. De plus, deux éléments d'une sous-suite croissante de π ne peuvent appartenir à la même $i^{ème}$ suite des minima de π. □

Remarque 4.46 *La preuve précédente nous permet de constater que la suite $\pi(1)\pi(2)\ldots\pi(p_j)$ contient une sous-suite croissante de longueur j pour tout $j \in [k]$.*

Proposition 4.47

- *Pour tout entier positif k, l'arbre de génération $T(12\ldots(k+1))$ est caractérisé par le système de réécriture*
$$\begin{cases} \underbrace{(2,2,\ldots,2)}_{k-1 \text{ fois}} \\ (p_2,p_3,\ldots,p_k) \rightsquigarrow (p_2+1,p_3+1,\ldots,p_k+1), \\ \qquad\qquad (p_2,p_3,\ldots,p_{i-1},s,p_{i+1}+1,p_{i+2}+1,\ldots,p_k+1) \\ \qquad\qquad\qquad \text{pour tout } i \in [2,k] \text{ et pour tout } s \in [p_{i-1}+1,p_i] \end{cases}$$
- *A une permutation π de $S_n(12\ldots(k+1))$ correspond, par ce système de réécriture, une étiquette $(p_2,p_3,\ldots,p_k)$ dans laquelle p_i est l'indice du premier élément de la $i^{ème}$ suite des minima de π (voir définition 4.43).*

Exemple 4.48 *La permutation $\pi =$ 6254713 possède les trois suites des minima* 621, 543 *et* 7. *Ainsi, π n'appartient pas à $S_7(123)$ mais a pour étiquette $(3,5)$ dans $T(1234)$ et pour étiquette $(3,5,8,8)$ dans $T(123456)$.*

Lemme 4.49 *Pour une permutation d'étiquette $(p_2, p_3, \ldots, p_k)$ dans l'arbre de génération $T(12\ldots(k+1))$, seuls les p_k premiers sites sont actifs.*

Preuve Ce résultat est une conséquence de la proposition 4.45. □

Preuve de la proposition 4.47. Clairement, l'étiquette de la permutation 1 est $(2, 2, \ldots, 2)$ par définition des étiquettes.
Soit π une permutation de l'arbre de génération $T(12\ldots(k+1))$ ayant pour étiquette $(p_2, p_3, \ldots, p_k)$.

- Insertion dans le premier site.
  L'étiquette de la permutation ainsi obtenue est alors $(p_2+1, p_3+1, \ldots, p_k+1)$.
- Insertion dans l'un des p_k premiers sites autre que le premier site.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $s^{ème}$ site de π, pour tout $s \in [2, p_k]$.
  Alors, il existe $i \in [2, k]$ tel que $p_{i-1} < s \leq p_i$.
  L'étiquette de γ est alors $(p_2, p_3, \ldots, p_{i-1}, s, p_{i+1}+1, p_{i+2}+1, \ldots, p_k+1)$.

□

Corollaire 4.50 *Soit $P(n; p_2, p_3, \ldots, p_k)$ le nombre de permutations de $S_n(12\ldots(k+1))$ ayant pour étiquette $(p_2, p_3, \ldots, p_k)$ où $2 \leq p_2 < p_3 < \cdots < p_k \leq n+1$, et posons $P(n; p_2, p_3, \ldots, p_j, n+1, n+1, \ldots, n+1) = P(n; p_2, p_3, \ldots, p_j)$.*
Nous avons alors les équations de récurrence suivantes.

$$\begin{cases} P(1;) = 1 \\ P(n; p_2, p_3, \ldots, p_k) = \\ \qquad P(n-1; p_2-1, p_3-1, \ldots, p_k-1) \\ \quad + \quad P(n-1; p_2, p_3, \ldots, p_{k-1}) \\ \quad + \quad \sum_{i=2}^{k} \sum_{s=p_i}^{p_{i+1}-2} P(n-1; p_2, p_3, \ldots, p_{i-1}, s, p_{i+1}-1, p_{i+2}-1, \ldots, p_k-1) \\ |S_n(12\ldots(k+1))| - |S_n(12\ldots k)| = \sum_{2 \leq p_2 < p_3 < \cdots < p_k \leq n} P(n; p_2, p_3, \ldots, p_k) \end{cases}$$

Par exemple, nous avons $P(n; p_2) = \binom{2n-p_2}{n-1} - \binom{2n-p_2}{n}$ pour tout $p_2 \in [2, n]$. Aussi, avant même de connaître $P(n; p_2, p_3, \ldots, p_k)$ dans le cas général, il serait bien évidemment utile d'avoir une formule pour $P(n; p_2, p_3)$, problème étudié par J. West [110].

Conjecture 4.51 *Le nombre de permutations π de $S_n(1234)$ telles que $\pi(1) < \pi(2) < \pi(3) = n$ est*

$$P(n; 2, 3) = \sum_{l=1}^{n-2} \binom{n-3}{l-1} \sum_{m=0}^{l-1} \frac{\binom{l+1}{m}.\binom{l+1}{m+1}.\binom{l+1}{m+2}}{\binom{l+1}{1}.\binom{l+1}{2}}$$

Remarquons que $\sum_{m=0}^{l-1} \frac{\binom{l+1}{m}.\binom{l+1}{m+1}.\binom{l+1}{m+2}}{\binom{l+1}{1}.\binom{l+1}{2}}$ dénombre les permutations de Baxter ayant l éléments [15, 108].

4.5.2 Tableaux de Young standard de hauteur bornée

Nous donnons ici une caractérisation de l'arbre de génération des involutions excluant le motif $12\ldots(k+1)$, arbre obtenu par la méthode récurrente. Cet arbre de génération est très semblable à celui donnant les permutations excluant ce même motif.

Proposition 4.52

- *Pour tout entier $k \geq 2$, l'arbre de génération des involutions de $I_n(12\ldots(k+1))$ obtenu par la méthode récurrente est caractérisé par le système de réécriture*

$$\begin{cases}
(0;) & & \\
(n;\underbrace{p_1,p_2,\ldots,p_j}_{0\leq j\leq k-2} & \rightsquigarrow & (n+1;p_1,p_2,\ldots,p_j,n+1) \\
 & \overset{2}{\rightsquigarrow} & (n+2;p_1,p_2,\ldots,p_{i-1},s,p_{i+1}+1,p_{i+2}+1,\ldots,p_j+1) \\
 & & \quad \textit{pour tout } i\in[j] \textit{ et pour tout } s\in[p_{i-1}+1,p_i] \\
 & \overset{2}{\rightsquigarrow} & (n+2;p_1,p_2,\ldots,p_j,s) \textit{ pour tout } s\in[p_j+1,n+1] \\
(n;p_1,p_2,\ldots,p_{k-1}) & \rightsquigarrow & (p_1,p_2,\ldots,p_{k-1},n+1) \\
 & \overset{2}{\rightsquigarrow} & (n+2;p_1,p_2,\ldots,p_{i-1},s,p_{i+1}+1,p_{i+2}+1,\ldots,p_{k-1}+1), \\
 & & \quad \textit{pour tout } i\in[k-1] \textit{ et pour tout } s\in[p_{i-1}+1,p_i] \\
 & & (p_1,p_2,\ldots,p_{k-1},s) \textit{ pour tout } s\in[p_{k-1}+1,n+1] \\
(p_1,p_2,\ldots,p_k) & \overset{2}{\rightsquigarrow} & (p_1,p_2,\ldots,p_{i-1},s,p_{i+1}+1,p_{i+2}+1,\ldots,p_k+1) \\
 & & \quad \textit{pour tout } i\in[k] \textit{ et pour tout } s\in[p_{i-1}+1,p_i]
\end{cases}$$

- *A une involution π de $I_n(12\ldots(k+1))$ correspond, par ce système de réécriture, une étiquette dans laquelle p_i est l'indice du premier élément de la $i^{ème}$ suite des minima de π (voir définition 4.43).*

Exemple 4.53 *L'involution $\pi = 57381624$ possède les trois suites des minima 531, 762 et 84. Ainsi, π n'appartient pas à $I_8(123)$, mais a pour étiquette $(1,2,4)$ dans le cas de $I_8(1234)$ et a pour étiquette $(8;1,2,4)$ dans le cas de $I_8(12345)$ et de $I_8(1234567)$.*

Lemme 4.54 *Pour une involution π de $I_n(12\ldots(k+1))$, nous avons les propriétés suivantes.*

(i) Si π possède j suites des minima, où $j\in[0,k-1]$, tous ses sites sont actifs.
En adoptant les définitions de la proposition 4.52, π a pour étiquette $(n;p_1,p_2,\ldots,p_j)$.

(ii) Si π possède exactement k suites des minima, seuls les p_k premiers sites engendrant des involutions de I_{n+2} sont actifs.
En adoptant les définitions de la proposition 4.52, π a pour étiquette $(p_1,p_2,\ldots,p_k)$.

Preuve

- (i). L'involution $\pi(1)\pi(2)\ldots\pi(n)(n+1)$ appartient à $I_{n+1}(12\ldots(k+1))$ et toutes les involutions de I_{n+2} engendrées à partir de π appartiennent à $I_{n+2}(12\ldots(k+1))$ car π appartient à $I_n(12\ldots(j+1))$.
- (ii). D'après la proposition 4.45, π possède une sous-suite σ de type $12\ldots k$. L'involution $\pi(1)\pi(2)\ldots\pi(n)(n+1)$ n'appartient donc pas à $I_{n+1}(12\ldots(k+1))$. De plus, comme $\pi(p_k)$ appartient à σ, toutes les insertions à droite de $\pi(p_k)$ engendrent des involutions de I_{n+2} contenant au moins une sous-suite de type $12\ldots(k+1)$. Par contre, aucune des insertions à gauche de $\pi(p_k)$ n'engendre d'involutions de I_{n+2} contenant une sous-suite de type $12\ldots(k+1)$.

□

Preuve de la proposition 4.52. Clairement, l'étiquette de l'involution ε de $I_0(12\ldots(k+1))$ est $(0;)$. Cherchons les étiquettes des fils d'une involution π de $I_n(12\ldots(k+1))$ dans l'arbre de génération obtenu par la méthode récurrente.

- Ajout d'un point fixe engendrant l'involution $\pi(1)\pi(2)\ldots\pi(n)(n+1)$.
  - π a pour étiquette $(n;p_1,p_2,\ldots,p_j)$ avec $j\in[0,k-2]$.
    L'étiquette de l'involution ainsi obtenue est alors $(n+1;p_1,p_2,\ldots,p_j,n+1)$.
  - π a pour étiquette $(n;p_1,p_2,\ldots,p_{k-1})$.
    L'étiquette de l'involution ainsi obtenue est alors $(p_1,p_2,\ldots,p_{k-1},n+1)$.
- Insertion dans l'un des sites situés à gauche du premier élément de la dernière suite des minima engendrant une involution de I_{n+2}.
  - π a pour étiquette $(n;p_1,p_2,\ldots,p_j)$ avec $j\in[0,k-1]$.
    Soit γ l'involution obtenue en insérant l'élément $n+2$ dans le $s^{ème}$ site de π, pour tout $s\in[p_j]$, et en incrémentant d'une unité tous les éléments de π supérieurs ou égaux à s.
    Alors, il existe $i\in[j]$ tel que $p_{i-1}<s\leq p_i$.
    L'étiquette de γ est alors $(n+2;p_1,p_2,\ldots,p_{i-1},s,p_{i+1}+1,p_{i+2}+1,\ldots,p_j+1)$.
  - π a pour étiquette $(p_1,p_2,\ldots,p_k)$.
    Soit γ l'involution obtenue en insérant l'élément $n+2$ dans le $s^{ème}$ site de π, pour tout $s\in[p_k]$, et en incrémentant d'une unité tous les éléments de π supérieurs ou égaux à s.
    Alors, il existe $i\in[k]$ tel que $p_{i-1}<s\leq p_i$.
    L'étiquette de γ est alors $(p_1,p_2,\ldots,p_{i-1},s,p_{i+1}+1,p_{i+2}+1,\ldots,p_k+1)$.
- Insertion dans l'un des sites situés à droite du premier élément de la dernière suite des minima engendrant une involution de I_{n+2}.
  - π a pour étiquette $(n;p_1,p_2,\ldots,p_j)$ avec $j\in[0,k-2]$.
    Soit γ l'involution obtenue en insérant l'élément $n+2$ dans le $s^{ème}$ site de π, pour tout $s\in[p_j+1,n+1]$, et en incrémentant d'une unité tous les éléments de π supérieurs ou égaux à s.
    L'étiquette de γ est alors $(n+2;p_1,p_2,\ldots,p_j,s)$.
  - π a pour étiquette $(n;p_1,p_2,\ldots,p_{k-1})$.
    Soit γ l'involution obtenue en insérant l'élément $n+2$ dans le $s^{ème}$ site de π, pour tout $s\in[p_{k-1}+1,n+1]$, et en incrémentant d'une unité tous les éléments de π supérieurs ou égaux à s.
    L'étiquette de γ est alors $(p_1,p_2,\ldots,p_{k-1},s)$.

□

Chapitre 5

Permutations triables par deux passages consécutifs dans une pile

D.E. Knuth [62] s'est intéressé aux permutations qui peuvent être triées par passage dans une pile. Il les a caractérisées comme étant exactement les permutations excluant le motif 231 et celles-ci, du fait de leur nombre, sont parfois appelées permutations de Catalan.

Dans ce problème du tri d'une permutation par passage dans une pile, nous constatons que la pile ne doit contenir à tout instant que des entiers allant en croissant à partir du sommet de pile. Ainsi, dans un certain sens, la pile vérifie une condition que nous pouvons qualifier de contrainte de type "tour de Hanoï" par référence au problème du même nom.

Parmi les extensions possibles du problème considéré par D.E. Knuth, J. West [110, 113] s'est intéressé à l'énumération des permutations triables par plusieurs passages consécutifs dans une pile, celle-ci devant à tout instant obéir à cette condition "tour de Hanoï". Il a donné une caractérisation de l'ensemble des permutations sur $[n]$ triables par deux passages consécutifs dans une pile en terme de permutations à motifs exclus : il s'agit de $S_n(2341, 3\overline{5}241)$. De plus, il en a conjecturé la formule d'énumération $\frac{2.(3n)!}{(2n+1)!(n+1)!}$ dont les premières valeurs sont $1, 2, 6, 22, 91, 408, \ldots$

Une première preuve de cette conjecture, basée sur la résolution d'une récurrence très complexe avec des outils de calcul formel, a été donnée par D. Zeilberger [119].
D'autre part, S. Dulucq, S. Gire et J. West [28, 45], en utilisant la méthode des arbres de génération, ont établi une correspondance entre les permutations de $S_n(2413, 41\overline{3}52)$ dites permutations non séparables et l'ensemble des cartes planaires pointées non séparables ayant $n+1$ arêtes dont le nombre est exactement $\frac{2.(3n)!}{(2n+1)!(n+1)!}$ comme l'a montré W.T. Tutte [105]. De plus, cette bijection fait correspondre aux paramètres degré de la face distinguée et nombre de sommets de ces cartes les paramètres nombre de maxima à droite et descentes de ces permutations. Or, W.G. Brown [11] d'une part, et W.G. Brown et W.T. Tutte [12] d'autre part, ont donné des formules exprimant la distribution de ces cartes suivant ces deux paramètres.

Ainsi, la conjecture de J. West pouvant se ramener à trouver une correspondance entre permutations triables par deux passages consécutifs dans une pile et cartes planaires pointées non séparables, le résultat de S. Dulucq, S. Gire et J. West constitue une première étape vers une preuve combinatoire de cette conjecture. La seconde étape, que nous allons reprendre partiellement dans ce chapitre, a été proposée par S. Dulucq, S. Gire et O. Guibert [27, 45].

Signalons également qu'I.P. Goulden et J. West [46] ont récemment établi une nouvelle correspondance entre permutations triables par deux passages consécutifs dans une pile et cartes planaires pointées non séparables.

Cette seconde étape vers une preuve combinatoire de la conjecture de J. West, en partie devinée grâce au logiciel *forbid*, établit une correspondance entre les permutations sur $[n]$ triables par deux passages consécutifs dans une pile $S_n(2341, 3\overline{5}241)$ et les permutations non séparables de $S_n(2413, 41\overline{3}52)$, en utilisant la méthode des arbres de génération des permutations. Toutefois, la correspondance n'est pas directe et il nous est nécessaire de passer par quatre systèmes de réécriture différents (et donc sept autres ensembles de permutations à motifs exclus). Par contre, les deux paramètres considérés sur les cartes planaires pointées non séparables se transportent sur tous ces ensembles, de sorte que nous obtenons des formules d'énumération raffinant la formule conjecturée par J. West et démontrée par D. Zeilberger.

Comme la plupart des bijections intermédiaires reliant ces deux ensembles de permutations à motifs exclus sont détaillées dans la thèse de S. Gire [45], seule l'une des quatre correspondances est présentée ici.

Ensuite, nous relions, en utilisant toujours la même méthode, un nouvel ensemble de permutations à motifs exclus à ceux issus de notre correspondance.

Finalement, nous énonçons une conjecture dans laquelle nous proposons deux nouveaux ensembles de permutations à motifs exclus pour lesquels nous pensons qu'ils ont la même formule d'énumération que celle des permutations triables par deux passages consécutifs dans une pile.

5.1 Des permutations 2-triables aux permutations non séparables

Pour des raisons de simplicité, nous employons le terme de permutations 2-triables pour désigner les permutations triables par deux passages consécutifs dans une pile.

Soient $Piles_n = S_n(2341, 3\overline{5}241)$ l'ensemble des permutations 2-triables sur $[n]$ et $NSép_n = S_n(2413, 41\overline{3}52)$ l'ensemble des permutations sur $[n]$ que nous qualifions de non séparables. Désignons par NS_n l'ensemble des cartes planaires pointées non séparables ayant n arêtes.

Théorème 5.1 *(S. Dulucq, S. Gire, O. Guibert et J. West [27, 28]) Le nombre de permutations*

sur $[n]$ triables par deux passages consécutifs dans une pile est

$$|Piles_n| = \frac{2.(3n)!}{(2n+1)!(n+1)!}$$

En fait, en mettant en commun les résultats obtenus par S. Dulucq, S. Gire et J. West [28, 45] et S. Dulucq, S. Gire et O. Guibert [27, 45], nous obtenons des résultats beaucoup plus précis comme le laisse deviner le schéma général de notre preuve donné figure 5.1. Ce résultat, dont nous déduisons plusieurs formules pour la distribution des permutations 2-triables, est le suivant.

Théorème 5.2 *(S. Dulucq, S. Gire, O. Guibert et J. West [27, 28]) Les ensembles*
$\{\pi \in Piles_n : maxd(\pi) = i, desc(\pi) = j\}$,
$\{\pi \in S_n(3241, \overline{2}4153) : maxg(\pi) = i, montinv(\pi) = j\}$,
$\{\pi \in S_n(2413, \overline{4}2315) : ming(\pi) = i, descinv(\pi) = j\}$,
$\{\pi \in S_n(3412, \overline{2}4531) : maxg(\pi) = i, montinv(\pi) = j\}$,
$\{\pi \in S_n(3142, 45\overline{3}12) : ming(\pi) = i, desc(\pi) = j\}$,
$\{\pi \in NSép_n : maxd(\pi) = i, desc(\pi) = j\}$ *et*
$\{c \in NS_{n+1} :$ *le degré de la face distinguée de* c *est* $i+1, c$ *a* $j+2$ *sommets*$\}$
sont en correspondance.

Utilisant les résultats de W.G. Brown [11] et de W.G. Brown et W.T. Tutte [12] sur les cartes planaires pointées non séparables, nous en déduisons les formules d'énumération suivantes qui peuvent être étendues aux autres ensembles de permutations à motifs exclus.

Corollaire 5.3 *(S. Dulucq, S. Gire, O. Guibert et J. West [27, 28]) Le nombre de permutations sur $[n]$ triables par deux passages consécutifs dans une pile ayant i maxima à droite est*

$$\frac{i+1}{(2n-i+1)!} \sum_{j=i+1}^{\min\{n+1,2i+2\}} \frac{(3i-2j+2)(2j-i-1)(j-2)!(3n-j-i+1)!}{(n-j+1)!(j-i-1)!(j-i)!(2i-j+2)!}$$

Corollaire 5.4 *(S. Dulucq, S. Gire, O. Guibert et J. West [27, 28]) Le nombre de permutations sur $[n]$ triables par deux passages consécutifs dans une pile ayant j descentes est*

$$\frac{(2n-j-1)!(n+j)!}{(2n-2j-1)!(n-j)!(2j+1)!(j+1)!}$$

Corollaire 5.5 *(S. Dulucq, S. Gire, O. Guibert et J. West [27, 28]) Le nombre de permutations sur $[n]$ triables par deux passages consécutifs dans une pile ayant $k+1$ maxima à droite et k descentes est*

$$\frac{1}{n}\binom{n}{k}\binom{n}{k-1}$$

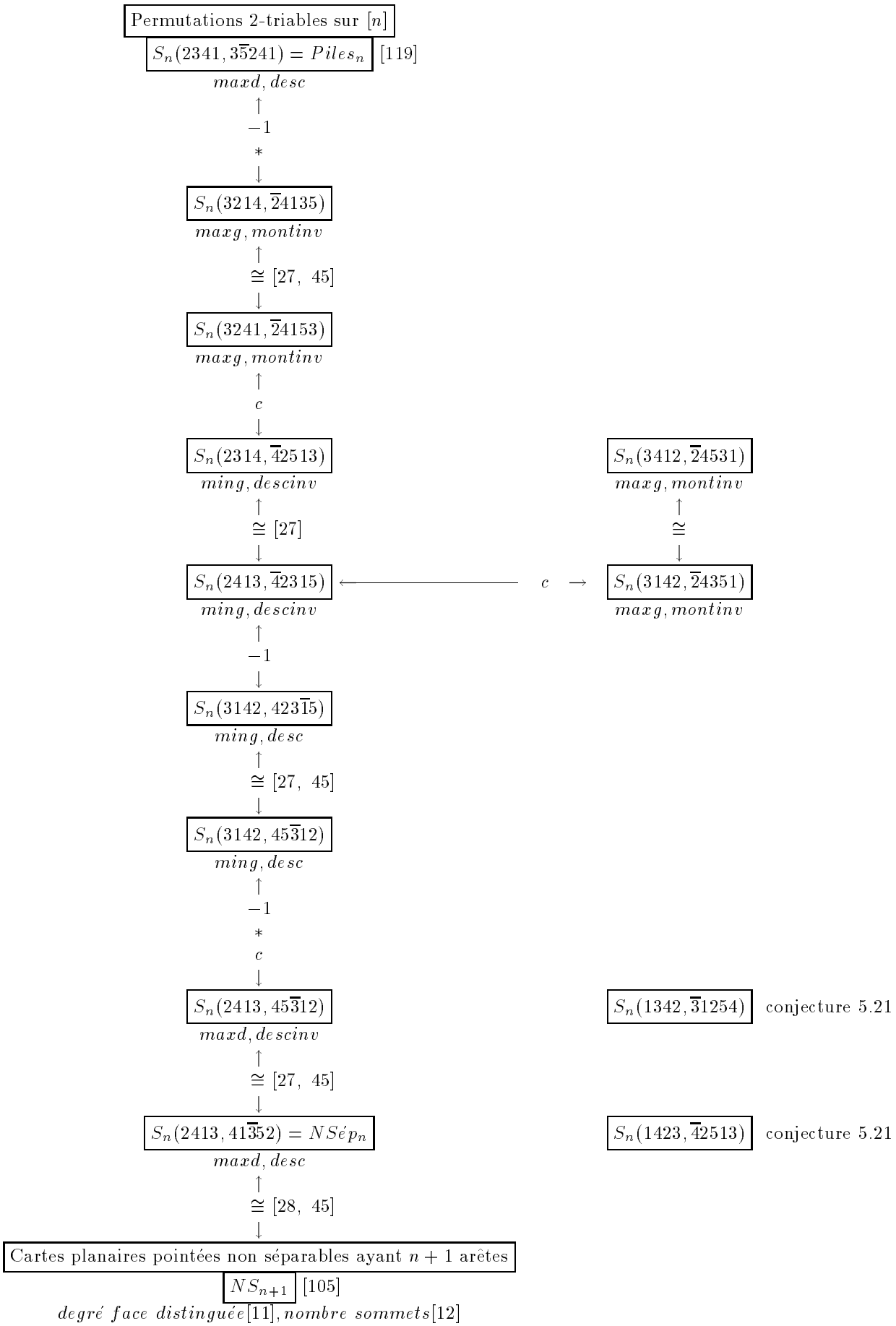


Figure 5.1 Schéma de la preuve de la conjecture de J. West.

Toutes les correspondances par isomorphisme d'arbres de génération mettant en bijection les permutations 2-triables et les cartes planaires pointées non séparables sont détaillées dans la thèse de S. Gire [45], exceptée la bijection entre $S_n(2314, \overline{4}2513)$ et $S_n(2413, \overline{4}2315)$ que nous précisons ici.

Ensuite, nous relions l'ensemble $S_n(3412, \overline{2}4531)$ à l'un des ensembles en bijection avec les permutations 2-triables sur $[n]$, toujours par isomorphisme des arbres de génération.

5.2 La correspondance entre $S_n(2314, \overline{4}2513)$ et $S_n(2413, \overline{4}2315)$

Théorème 5.6 *Les ensembles de permutations à motifs exclus $S_n(2314, \overline{4}2513)$ et $S_n(2413, \overline{4}2315)$ sont en correspondance.*
De plus, ils vérifient $|\{\pi \in S_n(2314, \overline{4}2513) : ming(\pi) = i, descinv(\pi) = j\}| = |\{\pi \in S_n(2413, \overline{4}2315) : ming(\pi) = i, descinv(\pi) = j\}|$.

Nous allons prouver ce résultat en montrant que les arbres de génération de ces ensembles de permutations sont isomorphes et que la bijection induite conserve les deux paramètres minima à gauche et descentes inverses.

Proposition 5.7

- *Le système de réécriture $\mathcal{S}_{DeuxPiles1}$ (voir figure 5.2) caractérisant l'arbre de génération $T(2314, \overline{4}2513)$ [resp. $T(2413, \overline{4}2315)$] est*

$$\begin{cases} (2|1;1;0;) \\ (x|maxg;ming;dinv;t_1,t_2,\ldots,t_{maxg-1}) \rightsquigarrow \\ \quad (x+1+\sum_{k=1}^{maxg-1} t_k|1;ming+1;dinv+1;) \\ \quad (x+1+\sum_{k=i}^{maxg-1} t_k|i;ming;dinv+1;t_1,t_2,\ldots,t_{i-1}) \text{ pour tout } i \in [2,maxg] \\ \quad (maxg+2|maxg+1;ming;dinv;t_1,t_2,\ldots,t_{maxg-1},j-1) \text{ pour tout } j \in [x-maxg] \end{cases}$$

- *L'étiquette $(x|maxg;ming;dinv;t_1,t_2,\ldots,t_{maxg-1})$ du système de réécriture $\mathcal{S}_{DeuxPiles1}$ correspondant à une permutation π de l'un quelconque des deux arbres de génération vérifie*
  - *x est le nombre de sites actifs de π,*
  - *$maxg = maxg(\pi)$,*
  - *$ming = ming(\pi)$,*
  - *$dinv = descinv(\pi)$,*
  - *t_i, pour tout i appartenant à $[maxg-1]$, est le nombre d'éléments de π situés à droite [resp. gauche] d'un site temporairement inactif relativement à $\overline{4}2513$ [resp. $\overline{4}2315$] et appartenant à l'intervalle $]g_{i-1}, g_i[$ (où g_i est le $i^{ème}$ maximum à gauche de π et $g_0 = 0$).*

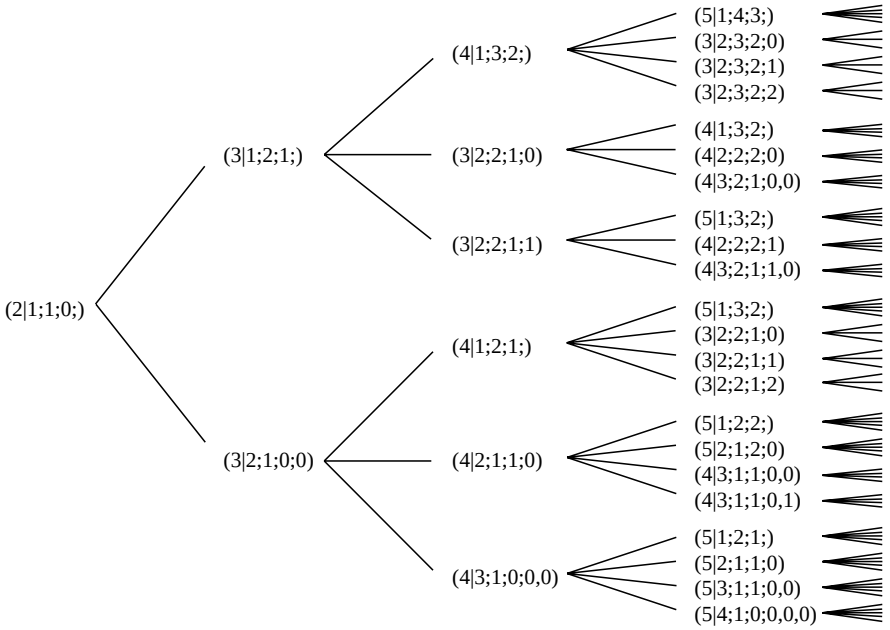


Figure 5.2 Arbre de dérivation du système de réécriture $\mathcal{S}_{DeuxPiles1}$.

Exemple 5.8 *La permutation* $\pi = {}_\diamond 2.1_\diamond 5.4_\diamond 6_\diamond 3_\bullet$ *appartenant à* $S_6(2314, \overline{4}2513)$ *a pour étiquette* $(4|3;2;3;1,1)$. *En effet, elle possède* 4 *fils dans* $T(2314, \overline{4}2513)$, 3 *maxima à gauche (éléments* 2,5,6*),* 2 *minima à gauche (éléments* 2,1*),* 3 *descentes inverses (indices* 1,3,4*). De plus, à droite des sites temporairement inactifs, nous trouvons les éléments* 1 *et* 4*, le premier étant inférieur à* $g_1 = 2$ *et le dernier étant compris entre* $g_1 = 2$ *et* $g_2 = 5$.

La permutation $\pi = {}_\diamond 5_\diamond 6_\diamond 4.2_\bullet 1.3.$ *appartenant à* $S_6(2413, \overline{4}2315)$ *a pour étiquette* $(3|2;4;3;3)$. *En effet, elle possède* 3 *fils dans* $T(2413, \overline{4}2315)$, 2 *maxima à gauche (éléments* 5,6*),* 4 *minima à gauche (éléments* 5,4,2,1*),* 3 *descentes inverses (indices* 1,3,4*). De plus, les* 3 *éléments* 4,1 *et* 3 *situés à gauche des sites temporairement inactifs sont inférieurs à* $g_1 = 5$.

Les figures 5.3 et 5.4 présentent respectivement les arbres de génération $T(2314, \overline{4}2513)$ et $T(2413, \overline{4}2315)$.

L'introduction des paramètres nombre de minima à gauche et nombre de descentes inverses dans les étiquettes du système de réécriture $\mathcal{S}_{DeuxPiles1}$ n'est pas nécessaire à la caractérisation des arbres de génération $T(2314, \overline{4}2513)$ et $T(2413, \overline{4}2315)$. Ces paramètres nous permettent seulement d'obtenir un raffinement des formules d'énumération.

Lemme 5.9 *En reprenant les notations de la proposition 5.7, une permutation* π *de* $S_n(2314, \overline{4}2513)$ *[resp.* $S_n(2413, \overline{4}2315)$*] vérifie les propriétés suivantes.*

(i) Le premier site et ceux entourant n *sont actifs.*

(ii) Tout élément e *situé entre* g_i *et* g_{i+1} *dans* π *vérifie* $g_{i-1} < e < g_i$.

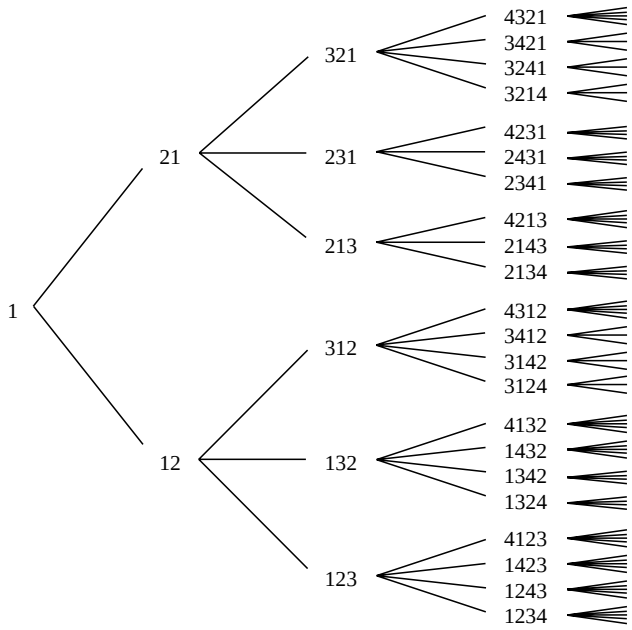


Figure 5.3 Arbre de génération $T(2314, \overline{4}2513)$.

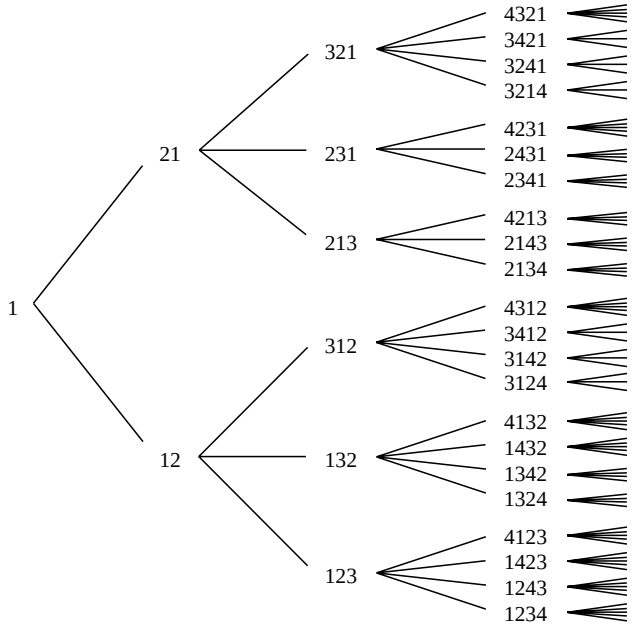


Figure 5.4 Arbre de génération $T(2413, \overline{4}2315)$.

(iii) Tous les éléments à droite de n et situés avant le dernier des sites actifs sont supérieurs à g_{maxg-1}.

(iv) A gauche de n, tout site situé à gauche d'un élément qui n'est pas un maximum à gauche est inactif.

(v) Tout site situé à gauche d'un maximum à gauche est actif.

(vi) Un site actif inactivé par $\overline{4}2513$ [resp. $\overline{4}2315$] ne l'est que temporairement.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii). Par définition d'un maximum à gauche, nous avons $e < g_i$. Supposons maintenant que $e < g_{i-1}$. Alors, la sous-suite $g_{i-1}g_ieg_{i+1}$ est de type 2314, interdite pour $T(2314, \overline{4}2513)$, et ne fait pas partie d'une sous-suite de type 42315 pour $T(2413, \overline{4}2315)$ car il n'existe pas d'élément à gauche de g_{i-1} qui soit compris entre g_i et g_{i+1}.
- (iii). Si tel n'était pas le cas, pour un élément e situé tel qu'indiqué, la sous-suite $g_{maxg-1}ne(n+1)$ serait de type 2314, interdite pour $T(2314, \overline{4}2513)$, et ne ferait pas partie d'une sous-suite de type 42315 pour $T(2413, \overline{4}2315)$.
- (iv). Soit e un élément situé à gauche de n qui ne soit pas un maximum à gauche. Alors, il existe $e' > e$ tel que $\pi^{-1}(e') < \pi^{-1}(e)$. La sous-suite $e'(n+1)en$ est de type 2413, interdite pour $T(2413, \overline{4}2315)$, et ne fait pas partie d'une sous-suite de type 42513 pour $T(2314, \overline{4}2513)$.
- (v). Supposons que le site situé à gauche de g_i ne soit pas actif.
  Si le site est inactivé par une sous-suite de type 2314 ou $\overline{4}2315$, tous les sites à sa droite seraient également inactifs, ce qui est en contradiction avec le fait que le site à droite de n est toujours actif d'après (i).
  Si le site est inactivé par une sous-suite de type 2413 ou $\overline{4}2513$, l'élément de π jouant le rôle de 2 dans le motif serait inférieur ou égal à g_{i-1} d'après (ii) et l'élément de π jouant le rôle de 1 dans le motif serait situé à droite du dernier site actif d'après (ii) et (iii), ce qui est en contradiction avec le fait que le site à droite de n est toujours actif d'après (i).
- (vi). Par exemple, la permutation $(n+1)\pi(1)\pi(2)\dots\pi(n)$ réactive un tel site.

□

Lemme 5.10 *En reprenant les notations de la proposition 5.7, une permutation π de $S_n(2314, \overline{4}2513)$ vérifie les propriétés suivantes.*

(i) A droite de n, tous les sites actifs sont consécutifs à partir de n.

(ii) A droite de n, tous les sites inactifs le restent définitivement.

(iii) A gauche de n, tous les sites inactifs ne le sont que temporairement.

(iv) $t_i = |\{\pi(\pi^{-1}(g_i)+1), \pi(\pi^{-1}(g_i)+2), \dots, \pi(\pi^{-1}(g_{i+1})-1)\}| = \pi^{-1}(g_{i+1}) - \pi^{-1}(g_i) - 1.$

Preuve

- (i) et (ii). Le motif $\overline{4}2513$ ne peut pas rendre inactif un site situé à droite de n et un site inactivé par 2314 impose que tous les sites à sa droite le restent définitivement.
- (iii). Le motif 2314 ne peut pas rendre inactif un site situé à gauche de n.
- (iv). C'est une conséquence des propriétés (ii) et (iv) du lemme 5.9 et des cas (ii) et (iii) précédents.

□

Lemme 5.11 *En reprenant les notations de la proposition 5.7, une permutation π de $S_n(2413, \overline{4}2315)$ vérifie les propriétés suivantes.*

(i) A gauche de n, tous les sites inactifs le restent définitivement.

(ii) Tous les sites à droite de n et situés à gauche du dernier des sites actifs sont actifs ou définitivement inactifs.

(iii) Soient $\pi(k)$ et $\pi(l)$ deux éléments situés à la gauche de deux sites temporairement inactifs et appartenant respectivement à $]g_{i-1}, g_i[$ et $]g_{j-1}, g_j[$. Alors, $k < l \Longrightarrow i \geq j$.

Preuve

- (i) et (ii). Tous les sites à la droite d'un site inactivé par $\overline{4}2315$ sont inactifs. Cependant, un site à droite du dernier site actif peut être définitivement inactif.
- (iii). Tous les éléments e situés à droite de $\pi(k)$ sont inférieurs à g_i pour éviter que la sous-suite $g_i n \pi(k) e$ ne soit de type 2413.

□

Preuve de la proposition 5.7. Par définition, l'étiquette de la permutation 1 est $(2|1;1;0;)$.
Soit π une permutation de $S_n(2314, \overline{4}2513)$ [resp. $S_n(2413, \overline{4}2315)$] ayant pour étiquette $(x|maxg; ming; dinv; t_1, t_2, \ldots, t_{maxg-1})$.

- Insertion dans le premier site.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le premier site de π.
  Tous les sites temporairement inactifs dans π s'activent dans γ, $n+1$ jouant le rôle du $\overline{4}$ dans $\overline{4}2513$ [resp. $\overline{4}2315$].
  L'étiquette de γ est alors $(x + 1 + \sum_{k=1}^{maxg-1} t_k | 1; ming + 1; dinv + 1;)$.
- Insertion dans l'un des sites actifs à gauche de n autre que le premier site.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [2, maxg]$.
  Tous les sites temporairement inactifs dans π dont l'élément situé à droite [resp. gauche] est supérieur à g_{i-1} s'activent dans γ, $n+1$ jouant le rôle du $\overline{4}$ dans $\overline{4}2513$ [resp. $\overline{4}2315$]. A l'inverse, tous les sites temporairement inactifs dans π dont l'élément situé à droite [resp. gauche] est inférieur à g_{i-1} sont inchangés dans γ.
  Les autres sites restent inchangés.
  L'étiquette de γ est alors $(x + 1 + \sum_{k=i}^{maxg-1} t_k | i; ming; dinv + 1; t_1, t_2, \ldots, t_{i-1})$.

- Insertion dans l'un des sites actifs à droite de n.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $(maxg+j)^{\grave{e}me}$ [resp. $(x+1-j)^{\hat{e}me}$] site actif de π, pour tout $j \in [x-maxg]$.
  Pour $T(2314, \overline{4}2513)$, les sites à droite de $n+1$ dans γ, sauf le premier, sont définitivement inactivés par la sous-suite $n(n+1)\gamma(\gamma^{-1}(n+1)+1)(n+2)$ de type 2314. Pour $T(2413, \overline{4}2315)$, les $j-1$ derniers sites actifs de π deviennent temporairement inactifs dans γ ; de plus, les éléments situés à leur gauche sont supérieurs à g_{maxg-1}. Rappelons que pour $T(2314, \overline{4}2513)$ [resp. $T(2413, \overline{4}2315)$], les sites compris entre n et l'élément à gauche de $n+1$ dans γ sont tous temporairement [resp. définitivement] inactifs.
  Les autres sites sont inchangés.
  L'étiquette de γ est alors $(maxg+2|maxg+1; ming; dinv; t_1, t_2, \ldots, t_{maxg-1}, j-1)$.

□

5.3 La correspondance entre $S_n(3142, \overline{2}4351)$ et $S_n(3412, \overline{2}4531)$

Théorème 5.12 *Les ensembles de permutations à motifs exclus $S_n(3142, \overline{2}4351)$ et $S_n(3412, \overline{2}4531)$ sont en correspondance.*
De plus, ils vérifient $|\{\pi \in S_n(3142, \overline{2}4351) : maxg(\pi) = i, montinv(\pi) = j\}| = |\{\pi \in S_n(3412, \overline{2}4531) : maxg(\pi) = i, montinv(\pi) = j\}|$.

Afin d'établir ce résultat, nous introduisons les notations suivantes.

Notation 5.13 *Soit π une permutation de $S_n(3142, \overline{2}4351)$ ou $S_n(3412, \overline{2}4531)$. Nous convenons des notations suivantes.*

- *x est le nombre de sites actifs de π dans l'arbre de génération des permutations correspondant,*
- *xg [resp. xd] est le nombre de sites actifs à gauche [resp. droite] de n,*
- $i = \begin{cases} 1 & si\ \pi^{-1}(n) < \pi^{-1}(n-1)\ et\ n > 1 \\ 0 & sinon \end{cases}$
- *$c = xd - i - 1$ est le nombre de sites actifs à droite de n exceptés le dernier et éventuellement le premier d'entre eux si i vaut 1.*

Lemme 5.14 *Une permutation de $S_n(3142, \overline{2}4351)$ vérifie les propriétés suivantes.*

(i) Le dernier site et celui à droite de n sont actifs.

(ii) A droite de n, seuls les sites à droite des maxima à droite peuvent être actifs.

Preuve

- (i) résulte de la forme des motifs exclus.

- (ii). Soit e un élément situé à droite de n qui ne soit pas un maximum à droite, c'est à dire qu'il existe au moins un élément e' à sa droite qui lui est supérieur. Alors, la sous-suite $ne(n+1)e'$ est de type 3142.

□

Lemme 5.15 *Une permutation de $S_n(3412, \overline{2}4531)$ vérifie les propriétés suivantes.*

(i) Les deux derniers sites sont actifs.

(ii) Tous les éléments à droite du premier des sites actifs à droite de n décroissent dans π.

(iii) Tous les sites actifs à droite de n sont consécutifs et situés complètement à droite.

(iv) Si $i = 0$, tous les éléments à droite de n décroissent.

(v) Si $i = 0$, tous les sites à droite de n sont actifs.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii). Aucune sous-suite $n(n+1)e_1e_2$ ne doit être de type 3412 où $n+1$ est inséré dans le premier des sites actifs à droite de n.
- (iii). Soit e un élément à droite de n et à droite d'un site actif. Alors, pour tout couple d'éléments e_1 et e_2 vérifiant $\pi^{-1}(n) < \pi^{-1}(e_1) < \pi^{-1}(e_2)$, la sous-suite $n(n+1)e_1e_2$ de type 3421 doit elle-même faire partie d'une sous-suite $e'n(n+1)e_1e_2$ de type 24531. De même, la sous-suite $e(n+1)e_1e_2$ de type 3421 fait partie d'une sous-suite $e'e(n+1)e_1e_2$ de type 24531. Le site à droite de e est donc actif.
- (iv). Autrement, il existerait une sous-suite $(n-1)ne_1e_2$ de type 3412 (où e_2 serait un maximum à droite contrairement à e_1).
- (v). Supposons que la permutation γ obtenue en insérant l'élément $n+1$ dans le site à droite de n n'appartienne pas à $S_{n+1}(3412, \overline{2}4531)$. L'élément $n+1$ de γ doit jouer le rôle du plus grand élément des motifs exclus, ce qui rend impossible d'interdire le motif 3412 d'après (iv). Considérons la sous-suite $e(n+1)e_1e_2$ de type 3421 interdite de sorte qu'aucun élément à gauche de e n'appartienne à $]e_2, e_1[$. e est différent de n car sinon la sous-suite ene_1e_2 de π devrait faire partie d'une sous-suite de type 24531. Enfin, si e vaut n, il existe une sous-suite $a(n-1)ne_1e_2$ de type 24531, du même type que la sous-suite $an(n+1)e_1e_2$. Ainsi, dans chaque cas, nous avons obtenu une contradiction.

□

Notation 5.16 *Conformément aux notations 5.13, nous convenons de celles-ci pour tout $k \in [c]$.*

- *Soit π une permutation de $S_n(3142, \overline{2}4351)$.*
  Notons d_j le $j^{ème}$ maximum à droite de π lue de droite à gauche. d_{j_k} désigne le maximum à droite situé juste à gauche du $(x-k)^{ème}$ site actif de π.
  Alors, a_k est l'élément le plus à gauche des éléments de π appartenant à $]d_{j_k-1}, d_{j_k}[$.

Lorsque $i = 0$ et $k = c$, $d_{j_c} = n$ et alors a_c est soit l'élément $n-1$, soit un élément à sa gauche. Dans tous les autres cas, la sous-suite $nd_{j_k}(n+1)d_{j_k-1}$ de type 3241 doit faire elle-même partie d'une sous-suite de type 24351 et impose l'existence d'un élément de la définition de a_k.
Nous en déduisons également que a_k est toujours situé à gauche de n.

- *Soit π une permutation de $S_n(3412, \overline{2}4531)$.*
  a_k est l'élément le plus à gauche des éléments de π appartenant à $]\pi(n-k+1), \pi(n-k)[$.

  Notons que nous avons toujours $\pi(n-k+1) < \pi(n-k)$ d'après les propriétés (ii) et (iv) du lemme 5.15 pour i valant respectivement 1 et 0.

  Lorsque $i = 0$ et $k = c$, $\pi(n-c) = n$ et alors a_c est soit l'élément $n-1$, soit un élément à sa gauche. Dans tous les autres cas, la sous-suite $n(n+1)\pi(n-k)\pi(n-k+1)$ de type 3421 doit faire elle-même partie d'une sous-suite de type 24531 et impose l'existence d'un élément de la définition de a_k.
  Nous en déduisons également que a_k est toujours situé à gauche de n.

Nous allons maintenant prouver le théorème 5.12 en montrant que les arbres de génération de ces ensembles de permutations sont isomorphes et que la bijection induite conserve les deux paramètres maxima à gauche et nombre de montées inverses.

Proposition 5.17 *En reprenant les notations 5.13 et 5.16, nous obtenons le résultat suivant.*

- *Le système de réécriture $\mathcal{S}_{DeuxPiles2}$ (voir figure 5.5) caractérisant les arbres de génération $T(3142, \overline{2}4351)$ et $T(3412, \overline{2}4531)$ est*

$$\begin{cases} (2|1;0;0;) \\ (x|w;minv;i;n_1,n_2,\ldots,n_c) \rightsquigarrow \\ \quad (l+1+k|w_1w_2\ldots w_{l-1}1;minv;1;n_1,n_2,\ldots,n_{k-1}) \text{ pour tout } l \in [xg] \\ \quad\quad \text{où } k \text{ est tel que } s_{k-1} < l \leq s_k \\ \quad (x+1|w1;minv+1;0;n_1,n_2,\ldots,n_c,xg-s_c) \text{ si } i=1 \\ \quad (x+1|w0^{c+i-k}1;minv+1;0;n_1,n_2,\ldots,n_k) \text{ pour tout } k \in [0,c] \end{cases}$$

  avec $s_k = \sum_{h=1}^{k} n_h$ et $s_{c+1} = xg = x-c-i-1$.

- *L'étiquette $(x|w;minv;i;n_1,n_2,\ldots,n_c)$ du système de réécriture $\mathcal{S}_{DeuxPiles2}$ correspondant à une permutation π sur $[n]$ de l'un quelconque des deux arbres de génération vérifie*

  - *$w = w_1w_2\ldots w_{xg}$ vérifie, pour tout $l \in [xg]$,*
    $w_l = \begin{cases} 1 & \text{si l'élément à droite du } l^{ème} \text{ site actif est un maximum à gauche} \\ 0 & \text{sinon} \end{cases}$
    Ainsi, $maxg(\pi) = \sum_{l=1}^{xg} w_l$,
  - *$minv = montinv(\pi)$,*

– n_k *est le nombre de sites actifs situés entre* a_{k-1} *et* a_k, *pour tout* $k \in [c]$ *(situés à gauche de* a_1 *pour* $k = 1$*).*

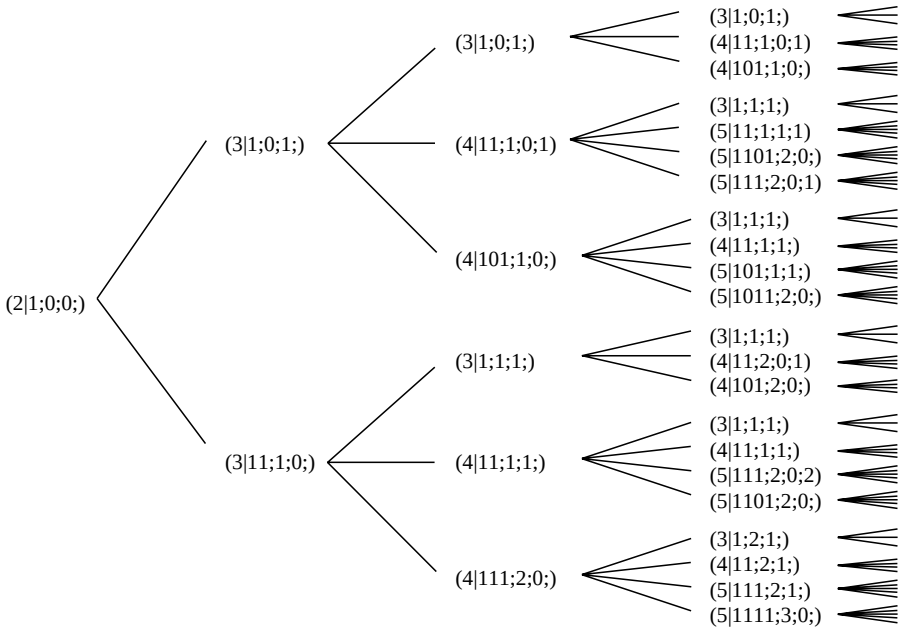


Figure 5.5 Arbre de dérivation du système de réécriture $\mathcal{S}_{DeuxPiles2}$.

Exemple 5.18 *La permutation* $\pi = {}_\diamond 3_\diamond 7_\diamond 9_\diamond 8_\diamond 4_\bullet 6_\bullet 5_\diamond 1_\bullet 2_\diamond$ *appartenant à* $S_9(3142,\overline{2}4351)$ *a pour étiquette* $(7|111;4;1;1,1)$. *En effet, elle possède* 7 *fils dans* $T(3142,\overline{2}4351)$, 3 *maxima à gauche (éléments* 3,7,9*) qui correspondent aux* 3 *sites actifs à gauche de* 9, 4 *montées inverses (indices* 1,3,4,7*),* $i = 1$, $xg = 3$, $xd = 4$ *et* $c = 2$. *Les maxima à droite sont* $d_1 = 2$, $d_2 = 5 = d_{j_1}$, $d_3 = 6$, $d_4 = 8 = d_{j_2}$ *et* $d_5 = 9$. *Nous avons* $a_1 = 3$ *(*3 *est l'élément le plus à gauche des éléments de* π *appartenant à* $]d_1, d_2[$*),* $a_2 = 7$ *(*7 *est l'élément le plus à gauche des éléments de* π *appartenant à* $]d_3, d_4[$*) et* $n_1 = 1$ *(un seul site est actif à gauche de* a_1*),* $n_2 = 1$ *(un seul site est actif entre* a_1 *et* a_2*).*

La permutation $\pi = {}_\diamond 2_\diamond 5_\bullet 3_\diamond 4_\diamond 1_\diamond$ *appartenant à* $S_5(3412,\overline{2}4531)$ *a pour étiquette* $(5|11;2;1;1)$. *En effet, elle possède* 5 *fils dans* $T(3412,\overline{2}4531)$, 2 *maxima à gauche (éléments* 2,5*) qui correspondent aux* 2 *sites actifs à gauche de* 5, 2 *montées inverses (indices* 2,3*),* $i = 1$, $xg = 2$, $xd = 3$ *et* $c = 1$. *Nous avons* $a_1 = 2$ *(*2 *est l'élément le plus à gauche des éléments de* π *appartenant à* $]\pi(5), \pi(4)[$*) et* $n_1 = 1$ *(un seul site est actif à gauche de* a_1*).*

Les paramètres w et $minv$ ont été ajoutés à l'étiquette du système de réécriture $\mathcal{S}_{DeuxPiles2}$ afin d'obtenir les formules d'énumération de $S_n(3142,\overline{2}4351)$ et de $S_n(3412,\overline{2}4531)$ suivant le nombre de maxima à gauche et le nombre de montées inverses.

Lemme 5.19 *Pour une permutation* π *de* $S_n(3142,\overline{2}4351)$, *nous avons les propriétés suivantes compte-tenu des notations 5.13 et 5.16.*

(i) Les deux premiers sites et celui à gauche de n sont actifs.

(ii) Un site inactif le reste définitivement.

(iii) L'insertion de l'élément $n+1$ dans l'un des sites actifs laisse dans le même état tous les sites situés à gauche de $n+1$ dans la permutation obtenue.

(iv) Les sites entourant un maximum à gauche sont actifs.

(v) $\pi^{-1}(a_k) < \pi^{-1}(a_{k'})$, pour tout $1 \leq k < k' \leq c$.

(vi) Tout élément situé à gauche de a_k est inférieur à d_{j_k-1}, pour tout $k \in [c]$.

(vii) Tout élément situé entre a_k et d_{j_k} est supérieur à a_k, pour tout $k \in [c]$.

Preuve

- (i), (ii) et (iii) résultent de la forme des motifs exclus.
- (iv). Les motifs 3142 et $\overline{2}4351$ ne peuvent pas rendre inactif le site à gauche d'un maximum à gauche g car les sous-suites $e_1e_2(n+1)e_3$ et $e_1e_2ge_3$ sont du même type respectivement 3142 et 3241.
  Similairement, le site à droite d'un maximum à gauche g ne peut pas être inactivé par 3142 ou $\overline{2}4351$ car les sous-suites $e_1e_2(n+1)e_3$ et $e_1e_2ge_3$ sont du même type respectivement 3142 ou 3241.
- (v). Autrement, la sous-suite $a_{k'}a_kd_{j_{k'}}d_{j_{k'}}$ serait de type 3142.
- (vi). Soit e un élément situé à gauche de a_k. $e < d_{j_k}$ car sinon la sous-suite $ea_knd_{j_k}$ serait de type 3142.
- (vii). Soit e un élément situé entre a_k et d_{j_k}. $e > d_{j_k-1}$ car sinon la sous-suite $a_ked_{j_k}d_{j_k-1}$ serait de type 3142, et $e \notin]d_{j_k-1}, a_k[$ car sinon la sous-suite $a_ked_{j_k}d_{j_k-1}$ serait de type 3241 mais ne ferait pas elle-même partie d'une sous-suite de type 24351, par définition de a_k.

□

Lemme 5.20 *Pour une permutation π de $S_n(3412, \overline{2}4531)$, nous avons les propriétés suivantes compte-tenu des notations 5.13 et 5.16.*

(i) Le premier site et celui à gauche de n sont actifs.

(ii) Un site inactif le reste définitivement.

(iii) L'insertion de l'élément $n+1$ dans l'un des sites actifs laisse dans le même état tous les sites situés à gauche de $n+1$ dans la permutation obtenue.

(iv) Le site à gauche de tout maximum à gauche est actif.

(v) $\pi^{-1}(a_k) < \pi^{-1}(a_{k'})$, pour tout $1 \leq k < k' \leq c$.

(vi) Tout élément situé à gauche de a_k est inférieur à $\pi(n-k+1)$, pour tout $k \in [c-1]$.

(vii) Tout élément situé entre a_{k+1} et $\pi(n-k-1)$ est supérieur à $\pi(n-k+1)$, pour tout $k \in [c-1]$.

Preuve

- (i), (ii) et (iii) résultent de la forme des motifs exclus.
- (iv). Les motifs 3412 et $\overline{2}4351$ ne peuvent pas rendre inactif le site à gauche d'un maximum à gauche g car les sous-suites $e_1(n+1)e_2e_3$ et $e_1ge_2e_3$ sont du même type respectivement 3412 et 3421.
- (v). Autrement, la sous-suite $a_{k'}\pi(n-k')\pi(n-k)\pi(n-k+1)$ serait de type 3421 mais ne ferait pas elle-même partie d'une sous-suite de type 24531, par définition de a_k.
- (vi). Soit e un élément situé à gauche de a_k. $e < \pi(n-k)$ car sinon la sous-suite $en\pi(n-k)\pi(n-k+1)$ serait de type 3421 ne ferait pas partie d'une sous-suite de type 24531, par définition de a_k.
- (vii). Soit e un élément situé entre a_{k+1} et $\pi(n-k-1)$. $e > \pi(n-k+1)$ car sinon la sous-suite $a_ka_{k+1}e\pi(n-k+1)$ serait de type 3412.

□

Preuve de la proposition 5.17. L'étiquette de la permutation 1 est $(2|1;0;0;)$.
Soit π une permutation de $S_n(3142, \overline{2}4351)$ [resp. $S_n(3412, \overline{2}4531)$] ayant pour étiquette $(x|w; minv; i; n_1, n_2, \ldots, n_c)$.

- Insertion dans l'un des sites actifs à gauche de n.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $l^{ème}$ site actif de π, pour tout $l \in [xg]$.
  Alors, il existe $k \in [c+1]$ tel que l'élément $n+1$ a été inséré entre a_{k-1} et a_k, avec $a_{c+1} = n$ et $d_{c+i} = n$.
  - Pour $T(3142, \overline{2}4351)$.
    Les sites situés entre a_k et d_{j_k} sont inactifs dans γ car la sous-suite $(n+1)a_k(n+2)d_{j_k}$ est de type 3142. Le site à droite de d_{j_k} est inactif dans γ car la sous-suite $(n+1)d_{j_k}(n+2)d_{j_k-1}$ est de type 3241 et tous les éléments à gauche de $n+1$ sont inférieurs à d_{j_k-1}. Les k derniers sites actifs le restent dans γ. Ceci est également vrai pour $k = c+1$ et $i = 0$.
  - Pour $T(3412, \overline{2}4531)$.
    Le site situé entre $\pi(n-k-1)$ et $\pi(n-k)$ est inactif dans γ étant donné que la sous-suite $(n+1)(n+2)\pi(n-k)\pi(n-k+1)$ est de type 3421 et que tous les éléments à gauche de $n+1$ sont inférieurs à $\pi(n-k+1)$. Les $k+1$ derniers sites actifs le restent dans γ.

  L'étiquette de γ est alors $(l+1+k|w_1w_2\ldots w_{l-1}1; minv; 1; n_1, n_2, \ldots, n_{k-1})$.
- Si i vaut 1, insertion dans le premier site actif à droite de n.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $(xg+1)^{ème}$ site actif de π, c'est à dire dans le site situé à droite de n [resp. $\pi(n-c-1)$].
  Tous les sites actifs le restent dans γ.
  - Pour $T(3142, \overline{2}4351)$, $d_{c+1} = n+1$ et $d_c = n-1$.
  - Pour $T(3412, \overline{2}4531)$, deux cas sont à envisager.
    * $\pi(n-c-1) < \pi(n-c)$. Alors, aucun élément e à gauche de n ne peut être supérieur à $\pi(n-c)$ pour éviter que la sous-suite $en\pi(n-c-1)\pi(n-c)$ ne soit de type 3412.

* $\pi(n-c-1) > \pi(n-c)$. Montrons que tous les éléments à gauche de n sont inférieurs à $\pi(n-c)$. Le site inactif à gauche de $\pi(n-c-1)$ impose l'existence d'un élément e vérifiant $\pi^{-1}(n) \leq \pi^{-1}(e) < n-c-1$ tel que la sous-suite $\sigma = e\pi(n-c-1)\pi(n-c)$ de type 321 ne fait pas partie d'une sous-suite de type 2431 : e n'est pas situé à gauche de n à cause du site actif à gauche de n, $\pi(n-c-1)$ fait partie de σ car le site entre $\pi(n-c-1)$ et $\pi(n-c)$ est actif, et $\pi(n-c)$ fait partie de σ car tous les sites à droite de $\pi(n-c)$ sont actifs. Tous les éléments à gauche de e sont donc supérieurs à $\pi(n-c-1)$ ou inférieurs à $\pi(n-c)$. Or, aucun élément e' à gauche de n n'est supérieur à $\pi(n-c-1)$ pour éviter que la sous-suite $e'n\pi(n-c-1)\pi(n-c)$ de type 3421 ne fasse partie d'une sous-suite de type 24531.

  Alors, $a_{c+1} = n$ et n_{c+1} est le nombre de sites actifs compris entre a_c et n.
  L'étiquette de γ est alors $(x+1|w1; minv+1; 0; n_1, n_2, \ldots, n_c, x-c-2-\sum_{h=1}^{c} n_h)$.

- Insertion dans l'un des sites actifs à droite de n autre que le premier si i vaut 1.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $(x-k)^{\grave{e}me}$ site actif de π, pour tout $k \in [0, c]$.
  Tous les sites actifs le restent dans γ.
  L'étiquette de γ est alors $(x+1|w0^{c+i-k}1; minv+1; 0; n_1, n_2, \ldots, n_k)$.

□

5.4 A propos des permutations de $S_n(1342, \overline{3}1254)$ et de $S_n(1423, \overline{4}2513)$

Nous proposons la conjecture suivante, vérifiée à l'aide du logiciel *forbid* jusqu'à l'ordre 14.

Conjecture 5.21

$$|S_n(1342, \overline{3}1254)| = |S_n(1423, \overline{4}2513)| = \frac{2.(3n)!}{(2n+1)!(n+1)!}$$

Nous avons constaté que les arbres de génération $T(1342, \overline{3}1254)$ et $T(1423, \overline{4}2513)$, ainsi que tous ceux correspondant aux permutations obtenues en considérant les opérations miroir, complément et inverse, ne satisfont à l'un des systèmes de réécriture des ensembles de permutations à motifs exclus que nous avons déjà mis en correspondance avec les permutations 2-triables.

De plus, aucune des distributions précédemment obtenues n'apparait pour ces deux ensembles de permutations à motifs exclus en termes de minima/maxima à gauche/droite et montées.

Chapitre 6

Permutations de Baxter

Faisant suite à une conjecture d'E. Dyer, G. Baxter [4] a mis en évidence une classe particulière de permutations en étudiant les points fixes de fonctions continues commutant par composition. Ces permutations, dites depuis permutations de Baxter, peuvent être définies en termes de motifs exclus. Plus précisément, elles vérifient les deux conditions suivantes : pour tout $1 \leq i < j < k < l \leq n$,

$$\text{si } \pi(i)+1 = \pi(l) \text{ et } \pi(j) > \pi(l) \text{ alors } \pi(k) > \pi(l),$$
$$\text{si } \pi(l)+1 = \pi(i) \text{ et } \pi(k) > \pi(i) \text{ alors } \pi(j) > \pi(i).$$

Ainsi, sur quatre éléments, seules les permutations 2413 et 3142 ne sont pas des permutations de Baxter.

F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15] ont montré, de manière analytique, que le nombre de permutations de Baxter sur $[n]$ est donné par la formule $\sum_{m=0}^{n-1} \frac{\binom{n+1}{m}\cdot\binom{n+1}{m+1}\cdot\binom{n+1}{m+2}}{\binom{n+1}{1}\cdot\binom{n+1}{2}}$ dont les premières valeurs sont $1, 2, 6, 22, 92, 422, \ldots$, après avoir deviné cette formule avec le concours du logiciel de calcul formel *MACSYMA*.

Plus tard, C.L. Mallows [74] a donné une interprétation plus fine de ce résultat en montrant que cette sommation correspondait à la distribution des permutations de Baxter suivant leur nombre de montées. De plus, il donne une nouvelle formule pour ces permutations où seul le paramètre m possède une interprétation (nombre de montées) : $\sum_{m=0}^{n-1}\sum_{s=1}^{n}\sum_{i=1}^{n}\binom{n+1}{m+1}\frac{s.i}{n.(n+1)}\left[\binom{n-s-1}{n-m-2}\binom{n-i-1}{m-1} - \binom{n-s-1}{n-m-1}\binom{n-i-1}{m}\right]$.

X. Viennot [108] a donné une preuve combinatoire de la formule obtenue par F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman en établissant une correspondance entre les permutations de Baxter et certains tableaux semi-standard pour lesquels une formule d'énumération est connue, correspondance qui repose sur un certain nombre de bijections classiques [39, 21, 43].

D'autre part, R. Cori, S. Dulucq et X. Viennot [18, 26] lors de la résolution d'un problème posé par R.C. Mullin [76] à propos de l'énumération de certaines familles de cartes planaires, ont établi une correspondance entre le langage produit de mélange (ou shuffle) de deux mots de parenthèses et les couples d'arbres binaires complets. Parmi les divers objets mis en œuvre dans

cette correspondance, apparaissent naturellement les permutations de Baxter alternantes. Ils en déduisent que le nombre de telles permutations est $c_n.c_n$ et $c_{n+1}.c_n$.

L'un des objectifs de ce chapitre est de fournir une preuve combinatoire unifiant les résultats de X. Viennot sur les permutations de Baxter [108] et ceux de R. Cori, S. Dulucq et X. Viennot sur les permutations de Baxter alternantes [18], tout en donnant une interprétation combinatoire de la formule de C.L. Mallows [74].

Avant cela, nous restons dans le contexte des permutations à motifs exclus. En effet, S. Gire [45] a montré que l'ensemble des permutations de Baxter sur $[n]$ est exactement $S_n(25\overline{3}14, 41\overline{3}52)$. Elle a également donné un système de réécriture caractérisant l'arbre de génération de ces permutations.

Nous montrons que les arbres de génération de divers objets considérés par la suite, tels que les permutations excluant simultanément les motifs $21\overline{3}54$ et $41\overline{3}52$ et triplets de chemins deux à deux disjoints, sont caractérisés par le système de réécriture donné par S. Gire pour les permutations de Baxter.

Ensuite, nous considérons une nouvelle famille d'arbres, les arbres binaires jumeaux. Ceux-ci sont obtenus en prenant les deux arbres binaires croissant et décroissant associés à une permutation et en oubliant leurs étiquetages. Cette application surjective est bijective lorsque les permutations considérées sont les permutations de Baxter. Par exemple, les permutations 2413 et 3412 donnent le même couple d'arbres binaires, mais seule la permutation 3412 est une permutation de Baxter.
Partant de cette caractérisation des permutations de Baxter en termes d'arbres binaires jumeaux [29], nous mettons en correspondance permutations de Baxter et triplets de chemins deux à deux disjoints dans un huitième de plan (ces triplets de chemins correspondent à des polyominos parallélogrammes jumeaux) et retrouvons ainsi les chemins obtenus par X. Viennot [108], ce qui nous permet de conclure. De plus, dans ces différentes bijections, un certain nombre de paramètres sont transportés.

Ainsi, nous déduisons des travaux d'I.M. Gessel et X. Viennot [43, 44] un déterminant 3×3 donnant le nombre de permutations de Baxter sur $[n]$ distribuées suivant cinq paramètres. Des cas particuliers nous permettent de retrouver les formules de F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15] et de C.L. Mallows [74] pour laquelle nous montrons que les paramètres m, i et s correspondent aux nombres de montées, de minima et maxima à gauche des permutations de Baxter.

De plus, dans la bijection que nous donnons entre permutations de Baxter et triplets de chemins deux à deux disjoints, le caractère alternant des permutations correspond au fait que le second chemin est en escalier. Ainsi apparait naturellement un couple de chemins de Dyck. Nous en déduisons que le nombre de permutations de Baxter alternantes sur $[2n+e]$ ($e = 0$ ou 1) est $c_{n+e}.c_n$, et affinons ensuite ce résultat.

6.1 Permutations de Baxter et triplets de chemins deux à deux disjoints

Définition 6.1 *Une permutation π de S_n est une* permutation de Baxter *si et seulement si, pour tout entier $p \in [n-1]$, π se factorise de manière unique sous la forme*

$$\pi = \pi' p \overset{<}{\pi}\overset{>}{\pi} (p+1)\pi'' \quad ou \quad \pi = \pi'(p+1) \overset{>}{\pi}\overset{<}{\pi} p\pi''$$

où tous les éléments de $\overset{<}{\pi}$ [resp. $\overset{>}{\pi}$] sont inférieurs à p [resp. supérieurs à $p+1$].
Nous notons $Baxter_n$ l'ensemble des permutations de Baxter sur $[n]$.

Exemple 6.2 *La permutation* 4236571 *appartient à $Baxter_7$. Par exemple, pour $p = 4$ nous avons $\overset{<}{\pi} = 23$ et $\overset{>}{\pi} = 6$, et pour $p = 1$ nous avons $\overset{>}{\pi} = 3657$ et $\overset{<}{\pi} = \emptyset$.*

Proposition 6.3 *(S. Gire [45]) $Baxter_n = S_n(25\overline{3}14, 41\overline{3}52)$.*

Remarquons que, d'après la forme des motifs exclus, si π appartient à $Baxter_n$, alors π^*, π^c et π^{-1} appartiennent également à $Baxter_n$.

Définition 6.4 *Nous désignons par $\widehat{Baxter}_{2n+e}$ l'ensemble des permutations de Baxter alternantes sur $[2n+e]$ où $e \in \{0,1\}$.*

Exemple 6.5 *La permutation* 2834176(11)9(10)5 *appartient à $\widehat{Baxter}_{11}$.*

Définition 6.6 *Soit $T_{n,m}$ l'ensemble des* triplets de chemins deux à deux disjoints *(voir figure 6.1) allant respectivement des 3 points de coordonnées $(0,n-1),(1,n),(2,n+1)$ aux 3 points de coordonnées $(m,m),(m+1,m+1),(m+2,m+2)$ en empruntant seulement des pas Est et Nord.*
Nous posons $T_n = \cup_{m=0}^{n-1} T_{n,m}$.

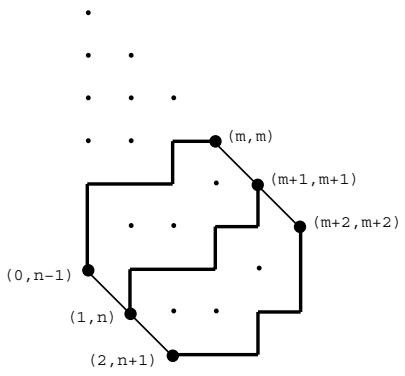


Figure 6.1 Un triplet de chemins deux à deux disjoints appartenant à $T_{7,3}$.

Ces triplets de chemins deux à deux disjoints sont un cas particulier de chemins considérés par I.M. Gessel et X. Viennot [43]. En effet, ils ont montré que les k-uplets de chemins deux à deux disjoints empruntant des pas Est et Nord dans un huitième de plan sont énumérés par

le déterminant d'une matrice carrée $k \times k$ à coefficients binômiaux. D'autre part, ils exhibent une correspondance entre ces chemins et les tableaux de Young semi-standard, tableaux dans lesquels les entiers sont non décroissant en colonne et strictement croissant en ligne. De plus, une formule d'énumération pour ces tableaux est connue depuis les travaux de J.B. Remmel et R. Whitney [81].

Ces résultats ont conduit X. Viennot [108] à donner une preuve combinatoire pour l'énumération des permutations de Baxter, en établissant une correspondance entre ces permutations et les triplets de chemins deux à deux disjoints. Cette correspondance se décompose en une bijection entre permutations et histoires de Laguerre [39], une bijection entre mots de Motzkin 2-colorés et polyominos parallélogrammes [21], et finalement une bijection entre chemins deux à deux disjoints et tableaux de Young semi-standard [43]. Il en déduit le résultat suivant.

Proposition 6.7 *(X. Viennot [108]) Le nombre de permutations de Baxter sur $[n]$ ayant m montées est égal au nombre de triplets de chemins deux à deux disjoints de $T_{n,m}$ et est donné par le déterminant*

$$\begin{vmatrix} \binom{n-1}{m} & \binom{n}{m} & \binom{n+1}{m} \\ \binom{n-1}{m+1} & \binom{n}{m+1} & \binom{n+1}{m+1} \\ \binom{n-1}{m+2} & \binom{n}{m+2} & \binom{n+1}{m+2} \end{vmatrix} = \frac{\binom{n+1}{m}.\binom{n+1}{m+1}.\binom{n+1}{m+2}}{\binom{n+1}{1}.\binom{n+1}{2}}$$

6.2 Un système de réécriture unique pour engendrer ces objets

Nous montrons que les permutations de Baxter, les permutations excluant simultanément les motifs $21\overline{3}54$ et $41\overline{3}52$ et les triplets de chemins deux à deux disjoints sont tous en correspondance. En effet, il est possible de caractériser leurs arbres de génération par le même système de réécriture que celui établi par S. Gire pour les permutations de Baxter. De plus, ces ensembles présentent une même triple distribution suivant certains paramètres que nous préciserons.

Proposition 6.8 *(S. Gire [45])*

- *Le système de réécriture $\mathcal{S}_{Baxter}$ (voir figure 6.2) caractérisant l'arbre de génération des permutations de Baxter $T(25\overline{3}14, 41\overline{3}52)$ est*

$$\begin{cases} (0,1,1) \\ (m,g,d) \rightsquigarrow (m,1,d+1),(m,2,d+1),\ldots,(m,g,d+1), \\ \qquad\qquad (m+1,g+1,d),(m+1,g+1,d-1),\ldots,(m+1,g+1,1) \end{cases}$$

- *L'étiquette (m,g,d) du système de réécriture $\mathcal{S}_{Baxter}$ correspondant à une permutation π de l'arbre de génération vérifie $m = mont(\pi)$, $g = maxg(\pi)$ et $d = maxd(\pi)$.*

Pour démontrer cette proposition, S. Gire utilise le résultat suivant.

Lemme 6.9 *(S. Gire [45]) Un site situé à gauche [resp. droite] de l'élément n d'une permutation de Baxter sur $[n]$ est actif si et seulement si l'élément à sa droite [resp. gauche] est un maximum à gauche [resp. droite].*

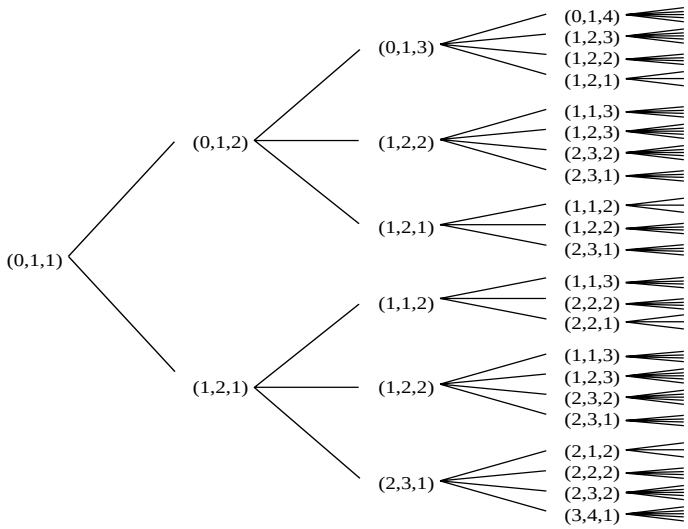


Figure 6.2 Arbre de dérivation du système de réécriture $\mathcal{S}_{Baxter}$.

Le paramètre m de l'étiquette du système de réécriture $\mathcal{S}_{Baxter}$ n'est pas nécessaire à la caractérisation de l'arbre de génération mais a été ajouté afin d'obtenir des raffinements dans nos formules d'énumération.

Propriété 6.10 *Une permutation π de $Baxter_n$ vérifie $mont(\pi) = montinv(\pi)$.*

Preuve En effet, d'après le lemme 6.9, le paramètre m peut aussi bien être le nombre de montées que de montées inverses dans l'arbre de génération des permutations de Baxter. □

Proposition 6.11

- *Le système de réécriture $\mathcal{S}_{Baxter}$ caractérise l'arbre de génération $T(21\overline{3}54, 41\overline{3}52)$.*
- *L'étiquette (m, g, d) du système de réécriture $\mathcal{S}_{Baxter}$ correspondant à une permutation π de l'arbre de génération vérifie $m = montinv(\pi)$, $g = maxg(\pi)$ et $d = maxd(\pi)$.*

Lemme 6.12 *Une permutation de $S_n(21\overline{3}54, 41\overline{3}52)$ vérifie les propriétés suivantes.*

(i) Les deux premiers sites, le dernier et celui situé à droite de n sont actifs.

(ii) Un site est actif si et seulement si il est situé à droite d'un maximum à gauche ou à droite.

Preuve

- (i) résulte de la forme des motifs exclus.
- (ii).
  Soit e un maximum à gauche [resp. droite] autre que n, le motif $41\overline{3}52$ [resp. $21\overline{3}54$] ne pouvant pas interdire un site à gauche [resp. droite] de n. Supposons que le motif $21\overline{3}54$ [resp. $41\overline{3}52$] interdise le site à droite de e, c'est à dire qu'existent e_1 et e_2 à gauche de e [resp. e_1 entre n et e et e_2 à

droite de e] tels que la sous-suite $e_1e_2(n+1)n$ [resp. $ne_1(n+1)e_2$] soit de type 2143 [resp. 3142]. Alors, $e_1e_2e(n+1)n$ [resp. $ne_1e(n+1)e_2$] est de type 21354 [resp. 41352].
Réciproquement, soit e un élément qui ne soit ni un maximum à gauche, ni un maximum à droite. Alors, existent e_1 et e_2 respectivement à gauche et à droite de e, tous deux supérieurs à e. La sous-suite $e_1e(n+1)e_2$ est de type 2143 ou 3142 sans qu'aucun élément ne soit situé entre e et $n+1$.

□

Preuve de la proposition 6.11. L'étiquette de la permutation 1 est bien $(0, 1, 1)$.
Soit π une permutation de l'arbre de génération des permutations ayant pour étiquette (m, g, d).

- Insertion dans l'un des sites actifs à gauche de n.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $i^{ème}$ site actif de π, pour tout $i \in [g]$.
  L'étiquette de γ est alors $(m, i, d+1)$.
- Insertion dans l'un des sites actifs à droite de n.
  Soit γ la permutation obtenue en insérant l'élément $n+1$ dans le $(g+d+1-j)^{ème}$ site actif de π, pour tout $j \in [d]$.
  L'étiquette de γ est alors $(m+1, g+1, j)$.

□

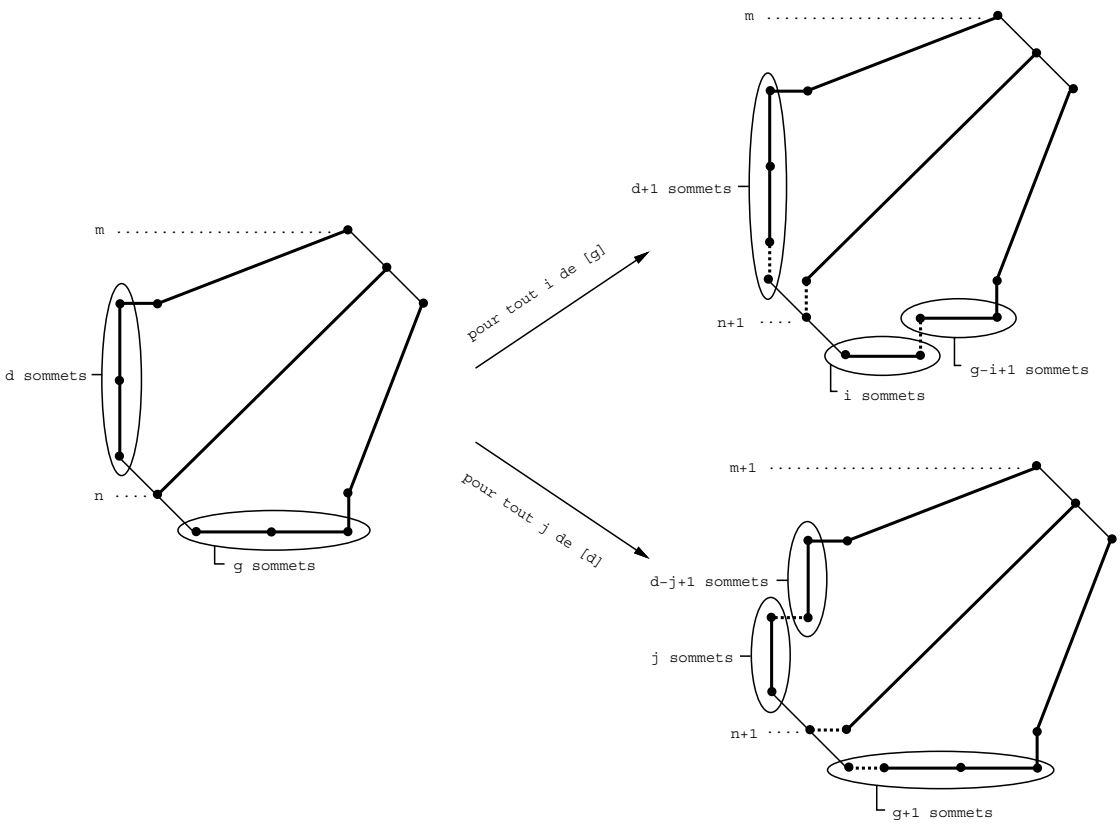


Figure 6.3 Règles de construction des triplets de chemins deux à deux disjoints.

Proposition 6.13

- *Le système de réécriture $\mathcal{S}_{Baxter}$ caractérise un arbre de génération des triplets de chemins deux à deux disjoints (voir figure 6.4) obtenu en appliquant les règles de construction décrites par la figure 6.3.*
- *L'étiquette (m, g, d) du système de réécriture $\mathcal{S}_{Baxter}$ associée à un triplet de chemins deux à deux disjoints de $T_{n,m}$ est telle que $d-1$ [resp. $g-1$] soit exactement le nombre de pas Nord [resp. Est] initiaux du premier [resp. dernier] chemin.*

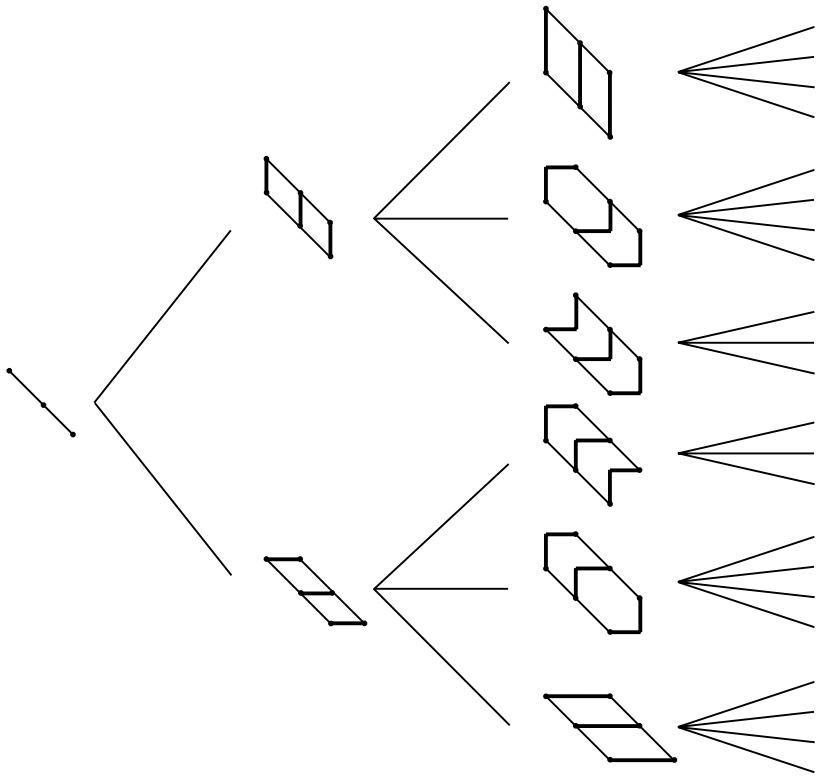


Figure 6.4 Arbre de génération des triplets de chemins deux à deux disjoints.

Preuve Un raisonnement par induction permet de montrer que tout triplet de chemins deux à deux disjoints peut être obtenu en appliquant ces règles de construction. En effet, soit $(\omega_1, \omega_2, \omega_3)$ un triplet de chemins deux à deux disjoints de $T_{n,m}$. Considérons le premier pas de ω_2. S'il s'agit d'un pas Nord [resp. Est], sa suppression ainsi que la suppression du premier pas Nord [resp. Est] de ω_1 [resp. ω_3] et la suppression du premier pas de ω_3 [resp. ω_1] donne un triplet de chemins deux à deux disjoints appartenant à $T_{n-1,m}$ [resp. $T_{n-1,m-1}$].

De plus, d'après les règles de construction, chaque triplet de chemins deux à deux disjoints n'est obtenu qu'une seule fois.

Il est immédiat de constater, compte-tenu des règles de construction, que cet arbre de génération est caractérisé par le système de réécriture $\mathcal{S}_{Baxter}$. □

6.3 Une correspondance entre permutations de Baxter et triplets de chemins

Nous donnons ici une nouvelle correspondance entre permutations de Baxter et triplets de chemins deux à deux disjoints permettant d'unifier les preuves combinatoires de X. Viennot [108] sur l'énumération des permutations de Baxter et celle de R. Cori, S. Dulucq et X. Viennot [18] pour les permutations de Baxter alternantes.

Cette correspondance se compose de deux bijections, la première reliant les permutations de Baxter et les arbres binaires jumeaux, la seconde reliant les arbres binaires jumeaux et les triplets de chemins deux à deux disjoints.

Définition 6.14 *L'ensemble des* arbres binaires jumeaux J_n *(voir figure 6.5) est l'ensemble*

$$J_n = \{(a_1, a_2) : \textit{complété}(a_1), \textit{complété}(a_2) \in A_n \text{ et}$$
$$\Theta(\textit{code}(\textit{complété}(a_1))) = \Theta^c(\textit{code}(\textit{complété}(a_2)))\}$$

où Θ *consiste en l'étiquetage des feuilles gauches [resp. droites] d'un arbre binaire (une fois complété) par la lettre* 0 *[resp.* 1*] excepté les deux feuilles extrêmes et* Θ^c *est identique à* Θ *modulo l'échange des lettres* 0 *et* 1*.*
Plus formellement, Θ *est l'application surjective de* $P_{z,\overline{z}}$ *dans* $\{0,1\}^*$ *définie par* $\Theta(z^l\overline{z}w_{l+2}w_{l+3}\dots w_{2n}) = \Theta(\overline{z}w_{l+2})\Theta(w_{l+2}w_{l+3})\dots\Theta(w_{2n-1}w_{2n})$ *avec* $\Theta(zz) = \Theta(\overline{z}z) = \varepsilon$, $\Theta(z\overline{z}) = 0$, $\Theta(\overline{z}\overline{z}) = 1$.

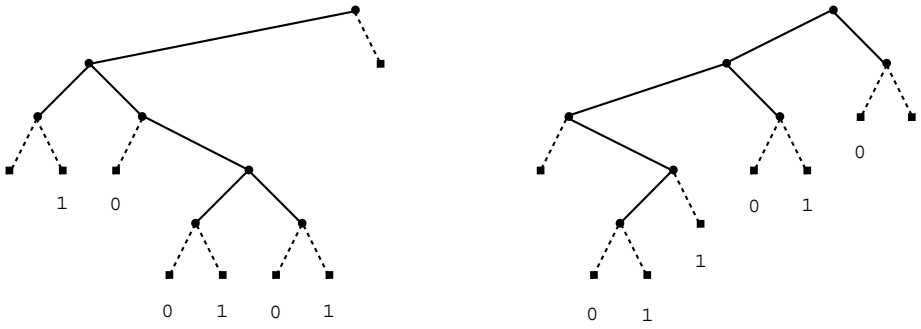


Figure 6.5 Deux arbres binaires jumeaux.

Exemple 6.15 *La figure 6.5 représente deux arbres binaires jumeaux de* J_7 *avec leurs étiquetages sur* $\{0,1\}$*, et dont les codes des arbres binaires complétés sont respectivement* $zzz\overline{z}\overline{z}z\overline{z}zz\overline{z}\overline{z}z\overline{z}\overline{z}$ *et* $zzz\overline{z}zz\overline{z}\overline{z}\overline{z}z\overline{z}\overline{z}z\overline{z}$*.*

6.3.1 La bijection entre permutations de Baxter et arbres binaires jumeaux

Théorème 6.16 *(S. Dulucq et O. Guibert [29]) Il existe une bijection* Ψ *(voir figure 6.6) entre permutations de Baxter et arbres binaires jumeaux.*

$$\begin{array}{rccc} \Psi & : & Baxter_n & \longrightarrow & J_n \\ & & \pi & \longmapsto & (a_1, a_2) \end{array}$$

De plus, à une permutation de Baxter ayant m montées, i minima à gauche et s maxima à gauche correspond par Ψ deux arbres binaires jumeaux dont le premier arbre binaire possède m arêtes droites et i sommets sur sa branche gauche et dont le second arbre binaire possède s sommets sur sa branche gauche.

L'application Ψ et son inverse sont définies de la façon suivante.

- Ψ consiste, pour une permutation π appartenant à $Baxter_n$, à construire ses arbres binaires croissant et décroissant. Les deux arbres binaires a_1 et a_2 sont ces deux arbres dépouillés de l'étiquetage de leurs sommets.

- L'application inverse Ψ^{-1} peut être décrite par l'algorithme suivant opérant sur un couple d'arbres binaires jumeaux (a_1, a_2) ayant n sommets.

  pour k variant de n à 1, répéter le processus suivant :
  considérant l'ordre infixe sur les sommets de a_1 et a_2, soit i l'ordre de la racine de a_2
  étiqueter k le $i^{ème}$ sommet (une feuille) f de a_1
  si f est une feuille gauche
  alors soit s le dernier sommet de la branche gauche du sous-arbre droit de a_2
  greffer le sous-arbre gauche de a_2 sur le sommet s
  sinon soit s le dernier sommet de la branche droite du sous-arbre gauche de a_2
  greffer le sous-arbre droit de a_2 sur le sommet s
  supprimer la racine de a_2
  supprimer la feuille f de a_1

  Au cours de cet algorithme, l'étiquetage croissant des sommets de a_1 est réalisé et la permutation π est alors obtenue en projetant en ordre infixe cet étiquetage.

Exemple 6.17 *La figure 6.6 illustre l'application Ψ en présentant une permutation de Baxter, ses arbres binaires croissant et décroissant, et les arbres binaires jumeaux correspondants.*
La figure 6.7 présente l'application Ψ^{-1} en déroulant chaque étape de l'algorithme précédent (où les $i^{èmes}$ sommets de a_1 et a_2 sont visualisés entre crochets, la feuille f de a_1 étiquetée k est mise en évidence, le sommet s est encadré et les arêtes de a_1 et a_2 à supprimer sont représentées en gras), l'arbre binaire croissant ainsi obtenu, et la permutation de Baxter correspondante.

Preuve du théorème 6.16.

- Considérons une permutation π et ses arbres binaires croissant et décroissant dépouillés de leurs étiquetages. Supposons que ces deux arbres binaires ne soient pas jumeaux. Alors, il existe pour chacun des deux arbres binaires complétés une feuille (autre que la première ou la dernière) ayant

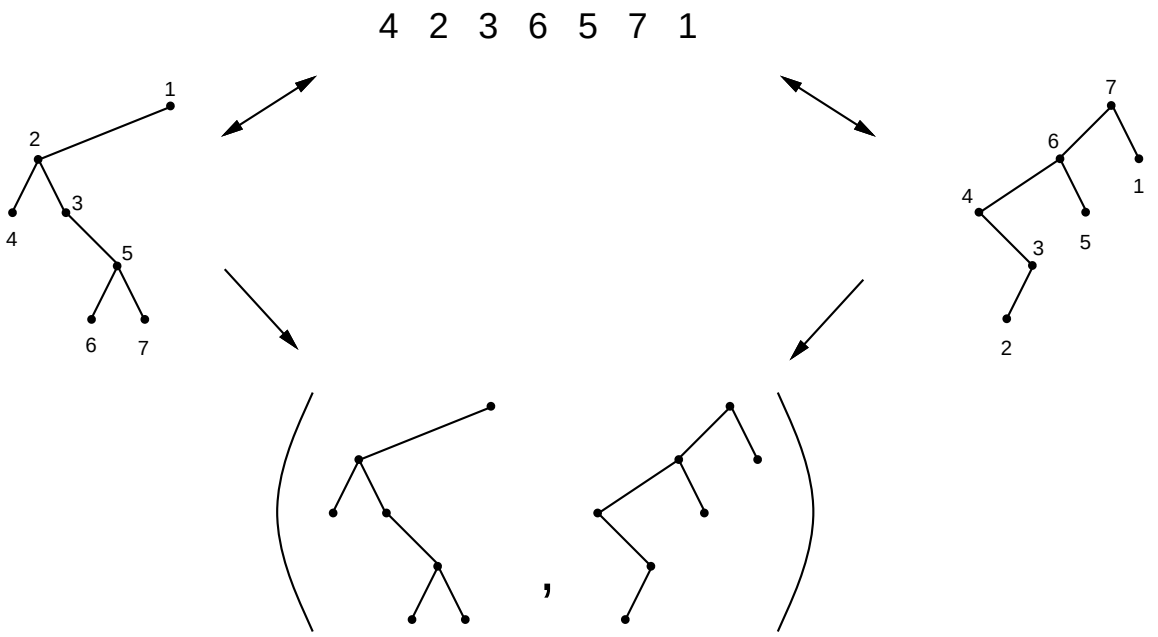


Figure 6.6 L'application Ψ, d'une permutation de Baxter aux arbres binaires jumeaux.

même rang impair relativement à l'ordre infixe sur tous les sommets et même orientation gauche ou droite dans les deux arbres. Supposons que cette même orientation soit par exemple la gauche. Le sommet père de cette feuille correspond dans les deux arbres binaires croissant et décroissant à un même élément de la permutation π. Ce même élément serait donc soit une feuille, soit un point simple à droite des arbres binaires croissant et décroissant associés à π. Ainsi, il correspondrait dans le cas de l'arbre binaire croissant à un pic (montée suivie d'une descente) ou une double-montée de π, et il correspondrait dans le cas de l'arbre binaire décroissant à un creux (descente suivie d'une montée) ou une double-descente de π. Ceci est bien évidemment impossible.

- L'application inverse Ψ^{-1} associe une et une seule permutation de Baxter π à un couple d'arbres binaires jumeaux (a_1, a_2).
  - L'application Ψ^{-1} est bien définie.
    - f est une feuille. Sinon, pour que les deux arbres soient jumeaux, il serait nécessaire d'avoir la relation $i = k = 1$, vraie seulement si les arbres sont réduits à un sommet.
    - f ayant un père dans a_1, la racine de a_2 a une arête d'orientation contraire.
    - A chaque étape, les nouveaux arbres binaires obtenus sont jumeaux.
  - Ψ^{-1} reconstitue un à un l'étiquetage croissant d'un arbre binaire, ce qui code une et une seule permutation.
  - La permutation obtenue est bien une permutation de Baxter.
    A l'étape k, pour tout $k \in [2, n]$, l'application Ψ^{-1} supprime une feuille gauche [resp. droite] f correspondant à l'élément k situé à gauche [resp. droite] d'un maximum à gauche [resp. droit] d'une permutation de S_{k-1}. Compte-tenu de la proposition 6.8 et du lemme 6.9, nous en déduisons par induction que la permutation obtenue est une permutation de Baxter.

□

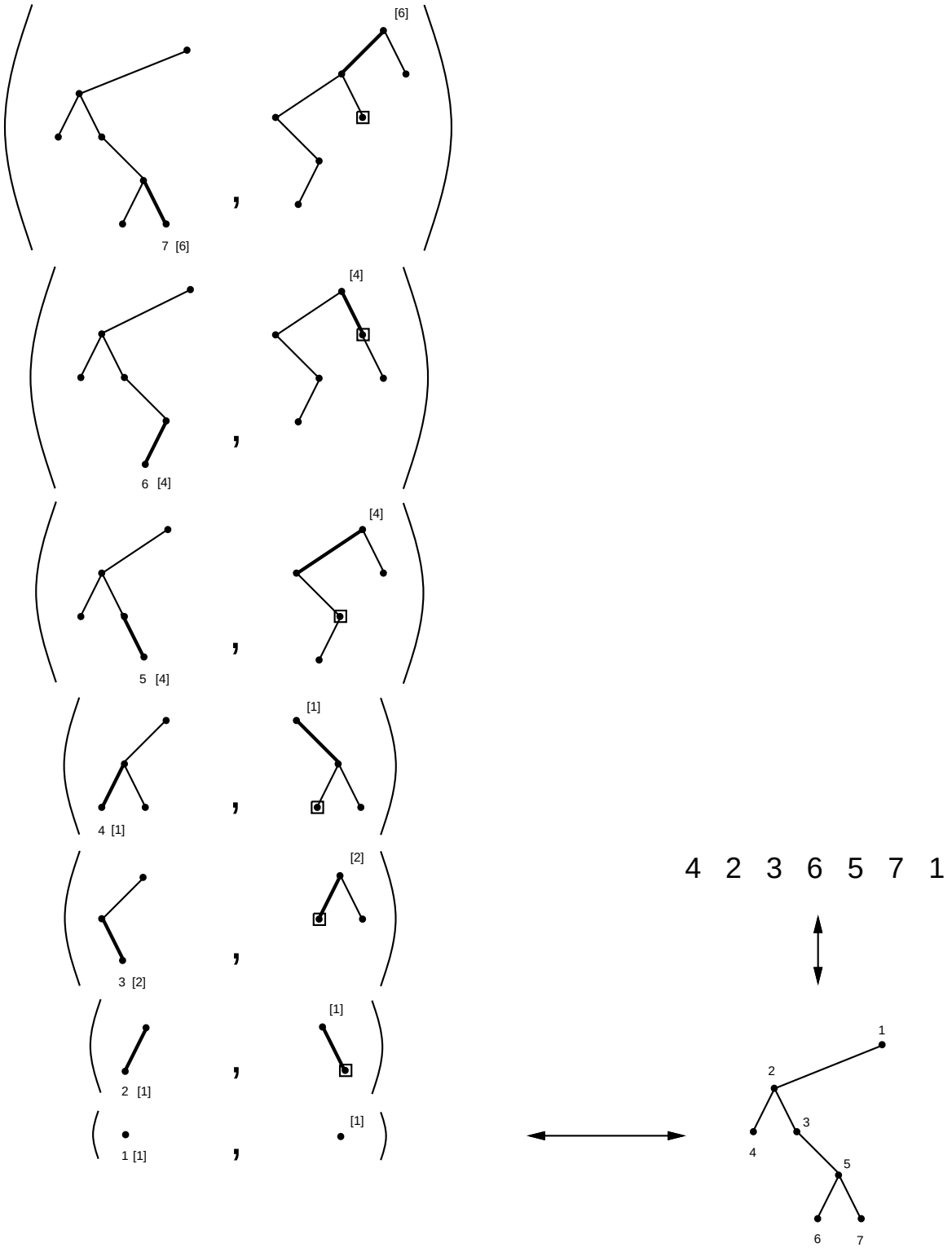


Figure 6.7 L'application inverse Ψ^{-1}, de deux arbres binaires jumeaux à une permutation de Baxter.

Propriété 6.18 *Soit π une permutation de Baxter telle que $\Psi(\pi) = (a_1, a_2)$.*
Alors, $\Psi(\pi^*) = (a_1^\star, a_2^\star)$ et $\Psi(\pi^c) = (a_2, a_1)$.

Preuve D'une part, l'arbre binaire croissant [resp. décroissant] d'une permutation σ quelconque est exactement le miroir de l'arbre binaire croissant [resp. décroissant] de σ^*. D'autre part, les arbres binaires croissant et décroissant d'une permutation σ quelconque sont échangés (et leurs étiquetages complémentés) lorsque nous considérons la permutation σ^c. □

6.3.2 La bijection entre arbres binaires jumeaux et triplets de chemins

Théorème 6.19 *Il existe une bijection Γ (voir figure 6.9) entre arbres binaires jumeaux et triplets de chemins deux à deux disjoints.*

$$\begin{array}{rccc} \Gamma & : & J_n & \longrightarrow & T_n \\ & & (a_1, a_2) & \longmapsto & t \end{array}$$

De plus, à deux arbres binaires jumeaux dont le premier arbre binaire possède m arêtes droites et i sommets sur sa branche gauche et dont le second arbre binaire possède s sommets sur sa branche gauche correspond par Γ un triplet de chemins deux à deux disjoints allant des 3 points de coordonnées $(1, n-i), (1, n), (s+1, n)$ respectivement aux 3 points de coordonnées $(m, m), (m+1, m+1), (m+2, m+2)$ en empruntant des pas Est et Nord.

Lemme 6.20 *(M. Delest et X. Viennot [21]) Il existe une bijection (voir figure 6.8) entre arbres binaires ayant n sommets, m arêtes droites et dont la branche gauche contient k sommets et couples de chemins disjoints allant des points de coordonnées $(1, n-k)$ et $(1, n)$ respectivement aux points de coordonnées (m, m) et $(m+1, m+1)$ en empruntant des pas Est et Nord.*

Preuve La bijection originale [21] entre arbres binaires et polyominos parallélogrammes, détaillée dans la sous-section 1.3.1, peut être vue sous une autre forme. En effet, elle revient à coder un arbre binaire complété en le parcourant suivant l'ordre préfixe par deux chemins deux à deux disjoints. Le premier chemin est obtenu en codant les arêtes internes (qui ne supportent pas les feuilles) gauche [resp. droite] par un pas Nord [resp. Est]; le second chemin est obtenu en codant les feuilles (exceptées les deux extrêmes) gauche [resp. droite] par un pas Est [resp. Nord]. □

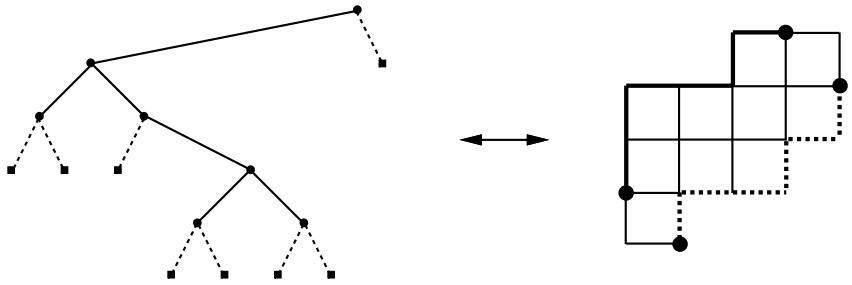


Figure 6.8 Arbre binaire complété et polyomino parallélogramme en bijection.

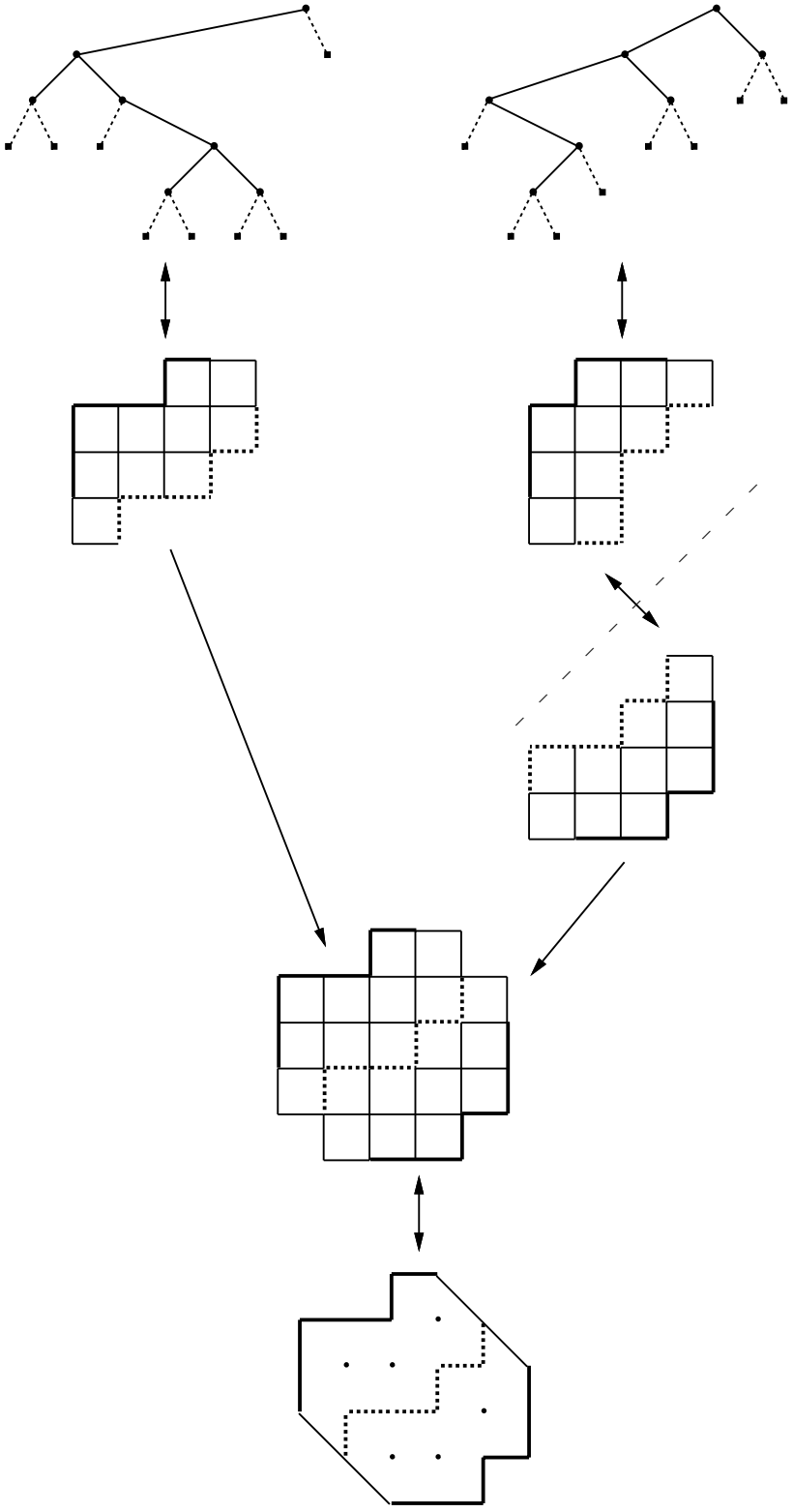


Figure 6.9 La bijection Γ entre deux arbres binaires jumeaux et un triplet de chemins deux à deux disjoints.

Preuve du théorème 6.19. Tout d'abord, d'après le lemme 6.20, à deux arbres binaires jumeaux correspondent deux couples de chemins disjoints (ou deux polyominos parallélogrammes). D'après la définition du caractère jumeau de deux arbres et la correspondance entre arbres binaires et couples de chemins disjoints (preuve du lemme 6.20), ces deux couples de chemins ont leurs seconds chemins complémentaires. Une symétrie par rapport à la diagonale du second couple permet de coller ces deux couples de chemins, donnant ainsi un triplet de chemins deux à deux disjoints.
Cette construction est clairement réversible. □

6.4 Enumération des permutations de Baxter

En composant les deux bijections Ψ et Γ des théorèmes 6.16 et 6.19, nous avons obtenu une correspondance entre permutations de Baxter et triplets de chemins deux à deux disjoints qui transporte plusieurs paramètres, nous permettant de raffiner les formules d'énumération connues sur ces objets.

6.4.1 Permutations de Baxter

En utilisant les résultats d'I.M. Gessel et X. Viennot [43, 44] sur l'énumération de chemins deux à deux disjoints, nous obtenons des formules d'énumération pour les permutations de Baxter qui précisent celles déjà connues, nous permettant en particulier d'avoir une interprétation naturelle de la formule de C.L. Mallows [74].

Théorème 6.21 *Le nombre de permutations de Baxter sur $[n]$ ayant m montées, i minima à gauche et s maxima à gauche est*

$$\binom{n+1}{m+1}\frac{s.i}{n.(n+1)}\left[\binom{n-s-1}{n-m-2}\binom{n-i-1}{m-1}-\binom{n-s-1}{n-m-1}\binom{n-i-1}{m}\right]$$

Preuve Le nombre de permutations de Baxter sur $[n]$ ayant m montées, i minima à gauche et s maxima à gauche est donné par le déterminant

$$\begin{vmatrix} \binom{n-1-i}{m-1} & \binom{n-1}{m-1} & \binom{n-1-s}{m-s-1} \\ \binom{n-1-i}{m} & \binom{n-1}{m} & \binom{n-1-s}{m-s} \\ \binom{n-1-i}{m+1} & \binom{n-1}{m+1} & \binom{n-1-s}{m-s+1} \end{vmatrix}$$

En effet, ces permutations de Baxter correspondent aux triplets de chemins deux à deux disjoints allant des 3 points de coordonnées $(1,n-i),(1,n),(s+1,n)$ respectivement aux 3 points de coordonnées $(m,m),(m+1,m+1),(m+2,m+2)$. Or, par une preuve en tout point identique à celle décrite dans l'article d'I.M. Gessel et X. Viennot [43] (la même involution permet d'obtenir le même résultat bien qu'ici nous n'ayons pas un mineur de déterminant binômial) ou plus simplement en spécialisant le résultat général exposé dans leur autre article [44], nous obtenons que le nombre de tels chemins est donné par ce déterminant, dont le calcul fournit la formule annoncée et due à C.L. Mallows [74]. □

Remarque 6.22 *La formule du théorème 6.21 dénombre les sommets au niveau n ayant pour étiquette (m, s, i) dans l'arbre de dérivation du système de réécriture $\mathcal{S}_{Baxter}$. En particulier, elle s'applique aux permutations excluant simultanément les motifs $21\overline{3}54$ et $41\overline{3}52$ et donne leur distribution suivant les nombres m de montées inverses, s de maxima à gauche et i de maxima à droite.*

Preuve Soit π une permutation de $Baxter_n$ ayant m montées, i minima à gauche et s maxima à gauche. Alors, la permutation π^{-1c*}, elle-même une permutation de Baxter sur $[n]$, possède m montées, i maxima à droite, s maxima à gauche d'après les propriétés 1.4 et 6.10. Les propositions 6.8 et 6.13 nous permettent de conclure. □

Notre correspondance nous permet de retrouver le résultat dû à F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15], démontré ensuite combinatoirement par X. Viennot [108].

Corollaire 6.23 *Le nombre de permutations de Baxter sur $[n]$ ayant m montées est*

$$\frac{\binom{n+1}{m}.\binom{n+1}{m+1}.\binom{n+1}{m+2}}{\binom{n+1}{1}.\binom{n+1}{2}}$$

Preuve Ces permutations correspondent à des triplets de chemins deux à deux disjoints allant des 3 points de coordonnées $(0, n-1), (1, n), (2, n+1)$ respectivement aux 3 points de coordonnées (m, m), $(m+1, m+1), (m+2, m+2)$. Or, le nombre de tels chemins est donné par le déterminant [43, 108]

$$\begin{vmatrix} \binom{n-1}{m} & \binom{n-1}{m-1} & \binom{n-1}{m-2} \\ \binom{n-1}{m+1} & \binom{n-1}{m} & \binom{n-1}{m-1} \\ \binom{n-1}{m+2} & \binom{n-1}{m+1} & \binom{n-1}{m} \end{vmatrix}$$

□

En fait, nous obtenons une énumération plus fine des permutations de Baxter en considérant deux paramètres supplémentaires.

Définition 6.24 *Etant donnée une permutation π de S_n, considérons les paramètres suivants.*

- $md(\pi) = mont(\pi(i)\pi(i+1)\dots\pi(n)) \in [n]$
  avec $i = \max\{j : \exists k \geq 2, \pi(j) < \pi(j+k) < \pi(j+1) < \pi(j+2) < \cdots < \pi(j+k-1)\}$
  en considérant que $\pi(0) = -1$ *et* $\pi(n+1) = 0$.

- $dd(\pi) = desc(\pi(i)\pi(i+1)\dots\pi(n)) \in [n]$
  avec $i = \max\{j : \exists k \geq 2, \pi(j) > \pi(j+k) > \pi(j+1) > \pi(j+2) > \cdots > \pi(j+k-1)\}$
  en considérant que $\pi(0) = n+2$ *et* $\pi(n+1) = n+1$.

Remarquons que $md(\pi)$ [resp. $dd(\pi)$] vaut un de plus que le nombre d'arêtes droites situées à droite de la dernière arête gauche de l'arbre binaire croissant [resp. décroissant] de π parcouru dans l'ordre infixe.

Théorème 6.25 *Le nombre de permutations de Baxter π sur $[n]$ ayant m montées, i minima à gauche, s maxima à gauche et telles que $p = md(\pi)$ et $q = dd(\pi)$ (voir figure 6.10) est donné par le déterminant*

$$\begin{vmatrix} \binom{n-1-i-p}{m-p} & \binom{n-1-p}{m-p} & \binom{n-1-s-p}{m-s-p} \\ \binom{n-1-i}{m} & \binom{n-1}{m} & \binom{n-1-s}{m-s} \\ \binom{n-1-i-q}{m} & \binom{n-1-q}{m} & \binom{n-1-s-q}{m-s} \end{vmatrix}$$

Preuve Une lecture plus fine des bijections Ψ et Γ permet de constater que ces deux paramètres supplémentaires sur les permutations de Baxter sont effectivement transportés sur les triplets de chemins deux à deux disjoints. □

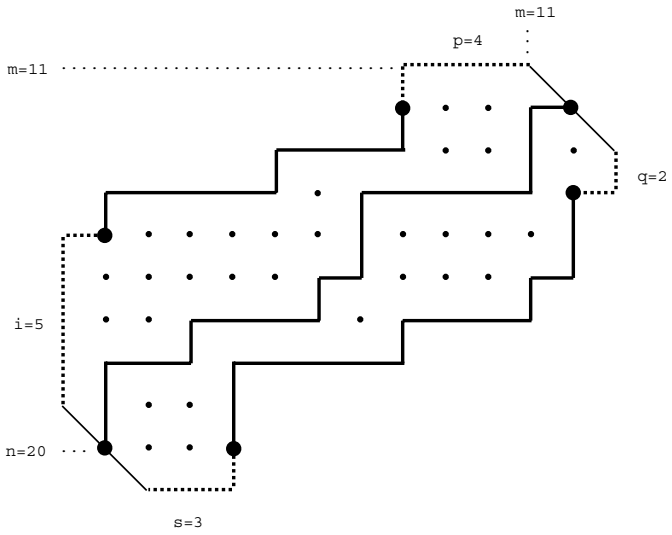


Figure 6.10 Les cinq paramètres considérés sur les triplets de chemins deux à deux disjoints.

6.4.2 Permutations de Baxter alternantes

Nous retrouvons et précisons ici un résultat dû à R. Cori, S. Dulucq et X. Viennot [18].

Théorème 6.26 *Le nombre de permutations de Baxter alternantes sur $[2n+e]$ où $e \in \{0,1\}$ ayant i minima à gauche et s maxima à gauche est*

$$\frac{i}{n+e-i}\binom{2(n+e)-i-1}{n+e}\cdot\frac{s-1}{n-s+1}\binom{2n-s}{n}$$

Corollaire 6.27 *Le nombre de permutations de Baxter alternantes sur $[2n+e]$ où $e \in \{0,1\}$ est*

$$c_{n+e}.c_n$$

Preuve Rappelons tout d'abord qu'à une permutation alternante correspond un arbre binaire croissant (ou décroissant) quasi-complet. En particulier, à une permutation de Baxter alternante sur $[2n]$

correspond un arbre binaire croissant [resp. décroissant] complet auquel il manque la feuille gauche [resp. droite] extrême tandis qu'à une permutation de Baxter alternante sur $[2n+1]$ correspond un arbre binaire croissant complet auquel il manque les deux feuilles (gauche et droite) extrêmes et un arbre binaire décroissant qui lui est complet. De plus, un tel couple d'arbres binaires quasi-complets, du fait de cette quasi-complétude, constitue un couple d'arbres binaires jumeaux. Ainsi, les permutations de Baxter alternantes sont en bijection avec les couples d'arbres binaires complets, et nous obtenons le résultat annoncé. □

Exemple 6.28 *La figure 6.11 illustre la correspondance entre permutations de Baxter alternantes et triplets de chemins deux à deux disjoints, et permet de constater que le deuxième chemin a une forme fixée (en escalier). Nous retrouvons naturellement par ce passage les chemins de Dyck.*

Preuve du théorème 6.26. Ce résultat est analogue à celui du théorème 6.21. En effet, aux permutations de Baxter alternantes sur $[2n+e]$ ayant i minima à gauche et s maxima à gauche correspondent deux chemins de Dyck débutant exactement par i et $s-1$ pas montants (ou deux arbres binaires jumeaux dont les branches gauches ont respectivement i et $s-1$ sommets). Ces objets sont énumérés par les nombres de Delannoy ce qui nous donne le résultat. □

En fait, nous affinons encore ces résultats en considérant les paramètres de la définition 6.24.

Théorème 6.29 *Le nombre de permutations de Baxter alternantes π sur $[2n+e]$ où $e \in \{0,1\}$ ayant i minima à gauche, s maxima à gauche et telles que $p = md(\pi)$ et $q = dd(\pi)$ est*

$$\left[\binom{2n+e-i-p-1}{n-p} - \binom{2n+e-i-p-1}{n-i-p}\right] \cdot \left[\binom{2n+e-s-q-1}{n-s} - \binom{2n+e-s-q-1}{n-1}\right]$$

Preuve C'est une conséquence directe du théorème 6.25 pour les permutations de Baxter alternantes. La formule se déduit de l'énumération des chemins de Dyck selon les hauteurs initiale et finale. □

La série génératrice des permutations de Baxter alternantes n'est pas algébrique, comme l'a montré D. Gouyou-Beauchamps [48, 49]. Toutefois, signalons que le système de réécriture

$$\begin{cases} (1,1) \\ (x,y) \quad \rightsquigarrow \quad (1,x+1),(2,x+1),\ldots,(y,x+1) \end{cases}$$

caractérise l'arbre de génération des permutations de Baxter alternantes. L'étiquette (x,y) correspondant à une permutation π sur $[2n+e]$ avec $e \in \{0,1\}$ de l'arbre de génération vérifie $x = maxd(\pi)$ et $y = mind(\pi)$ dans le cas $e = 0$, $x = mind(\pi)$ et $y = maxd(\pi)$ lorsque $e = 1$.

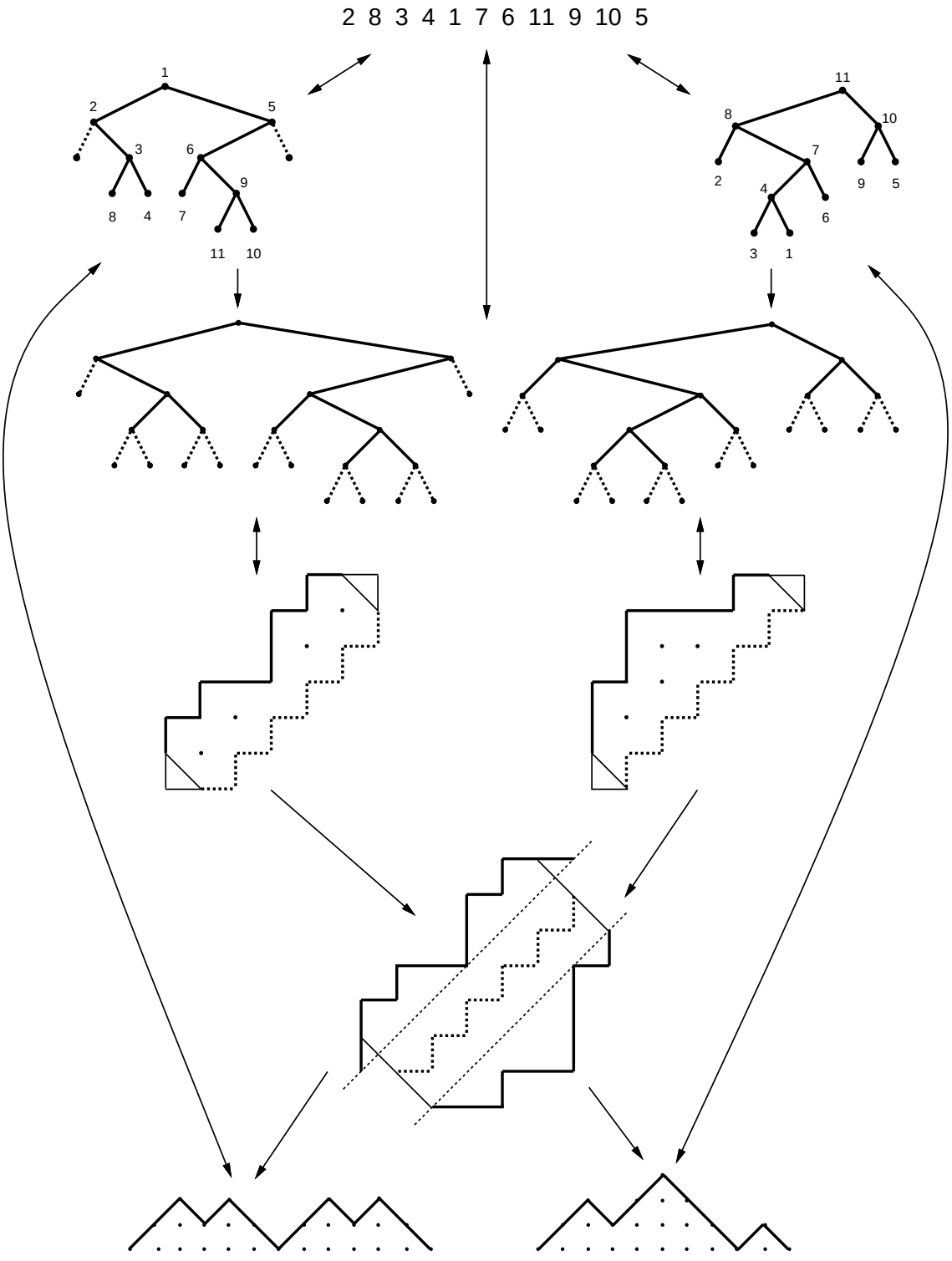


Figure 6.11 D'une permutation de Baxter alternante au triplet de chemins deux à deux disjoints : deux chemins de Dyck.

Chapitre 7

Mots de piles et tableaux de Young standard rectangulaires

Parmi les généralisations naturelles de l'algorithme de tri au moyen d'une pile considéré par D.E. Knuth [62], J. West [110, 113] s'est intéressé aux permutations triables par plusieurs passages consécutifs dans une pile, celle-ci devant satisfaire à une condition dite de type "tour de Hanoï", c'est à dire vérifier qu'à tout instant les entiers croissent à partir du sommet de la pile.

S. Gire [45] a étudié un problème voisin. Elle considère un ensemble de k piles placées en série et s'intéresse à leurs mouvements lorsque la permutation identité les traverse.
Les mots du langage $Y_n^{(k)} = \{f \in \{1,2,\ldots,k+1\}^* : \forall i \in [k+1], |f|_i = n; \forall i \in [k], \forall f = f'f'', |f'|_i \geq |f'|_{i+1}\}$ codent exactement les mouvements des k piles lorsque la permutation $12\ldots n$ les traverse (voir figure 7.1). Ce langage $Y_n^{(k)}$ code également les tableaux de Young standard [116] rectangulaires de hauteur $k+1$ et de longueur n, c'est à dire de forme $\lambda = (n, n, \ldots, n)$ partition de l'entier $(k+1).n$. Nous déduisons de la formule des équerres [38] que $|Y_n^{(k)}| = ((k+1).n)! \prod_{i=0}^{k} \frac{i!}{(n+i)!}$.

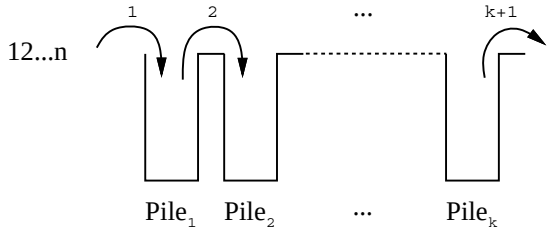


Figure 7.1 Mots de piles.

Dans le cas d'une seule pile ($k = 1$), les mots de $Y_n^{(1)}$ sont les mots de parenthèses et il y a une correspondance immédiate entre ces mots et les permutations 1-triables.

Ici, nous nous intéressons aux cas de deux piles ($k = 2$) et aux objets correspondant aux mouvements de ces deux piles que sont les tableaux de Young standard rectangulaires $3 \times n$, au nombre de $\frac{2(3n)!}{(n+2)!(n+1)!n!}$.

Définition 7.1 *Notons $\mathcal{A} = \{1, 2, 3\}$ l'alphabet des mots associés aux mouvements de deux piles, et $Y = \{f \in \mathcal{A}^* : |f|_1 = |f|_2 = |f|_3; \forall f = f'f'', |f'|_1 \geq |f'|_2 \geq |f'|_3\}$ le langage des mots codant les mouvements des piles (ensemble des mots de piles) correspondant aux tableaux de Young standard rectangulaires de hauteur 3. Posons $Y_n = Y_n^{(2)} = \{f \in Y : |f| = 3n\}$.*

Le fait d'imposer certaines restrictions sur les piles (par exemple qu'elles vérifient une condition de type "tour de Hanoï") se traduit simplement par certaines restrictions sur ces tableaux de Young standard.

Nous montrons en particulier que le nombre de tableaux de Young standard rectangulaires $3 \times n$ n'ayant pas deux entiers consécutifs sur la deuxième ligne est donné par le carré du $n^{ème}$ nombre de Catalan, résultat à rapprocher de ceux obtenus par D. Gouyou-Beauchamps [48, 49] à propos de l'énumération de tableaux de Young standard de hauteur au plus 4 et par L. Favreau [37] sur les tableaux oscillants de hauteur au plus 2.

Nous obtenons les résultats suivants, les trois premiers ayant été conjecturés par S. Gire [45].

Théorème 7.2 *Le nombre de mots du langage $C_n = Y_n \setminus \{\mathcal{A}^*22\mathcal{A}^*\}$ (ensemble des mots de piles sans facteur 22) codant les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur la deuxième ligne est*

$$c_n.c_n$$

Théorème 7.3 *Le nombre de mots du langage $B_n = Y_n \setminus \{\mathcal{A}^*22\mathcal{A}^*, \mathcal{A}^*11\mathcal{A}^*, \mathcal{A}^*33\mathcal{A}^*\}$ (ensemble des mots de piles sans facteur $22, 11, 33$) codant les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur une même ligne est égal au nombre de permutations de Baxter de $S_n(25\overline{3}14, 41\overline{3}52)$ donné par*

$$\sum_{m=0}^{n-1} \frac{\binom{n+1}{m}.\binom{n+1}{m+1}.\binom{n+1}{m+2}}{\binom{n+1}{1}.\binom{n+1}{2}}$$

Théorème 7.4 *Le nombre de mots du langage $H_n = Y_n \setminus \{f = f'2g2f'' : g \in Y\}$ (ensemble des mots de piles vérifiant la condition "tour de Hanoï") codant les tableaux de Young standard rectangulaires non séparables de hauteur 3 et de longueur n est égal au nombre de cartes planaires cubiques pointées non séparables ayant $2n$ sommets de CNS_{2n} donné par [103]*

$$\frac{2^n.(3n)!}{(2n+1)!(n+1)!}$$

Théorème 7.5 *Le nombre de mots du langage $P_n = Y_n \setminus \{f = f'2g2f'' : g \in Y\} \setminus \{\mathcal{A}^*11\mathcal{A}^*, \mathcal{A}^*33\mathcal{A}^*\}$ (ensemble des mots de piles vérifiant la condition "tour de Hanoï" et sans*

facteur $11,33$*) codant les tableaux de Young standard rectangulaires non séparables de hauteur* 3 *et de longueur* n *n'ayant pas deux entiers consécutifs sur une même ligne est égal au nombre de cartes planaires pointées non séparables ayant* $n+1$ *arêtes de* NS_{n+1} *donné par [105]*

$$\frac{2.(3n)!}{(2n+1)!(n+1)!}$$

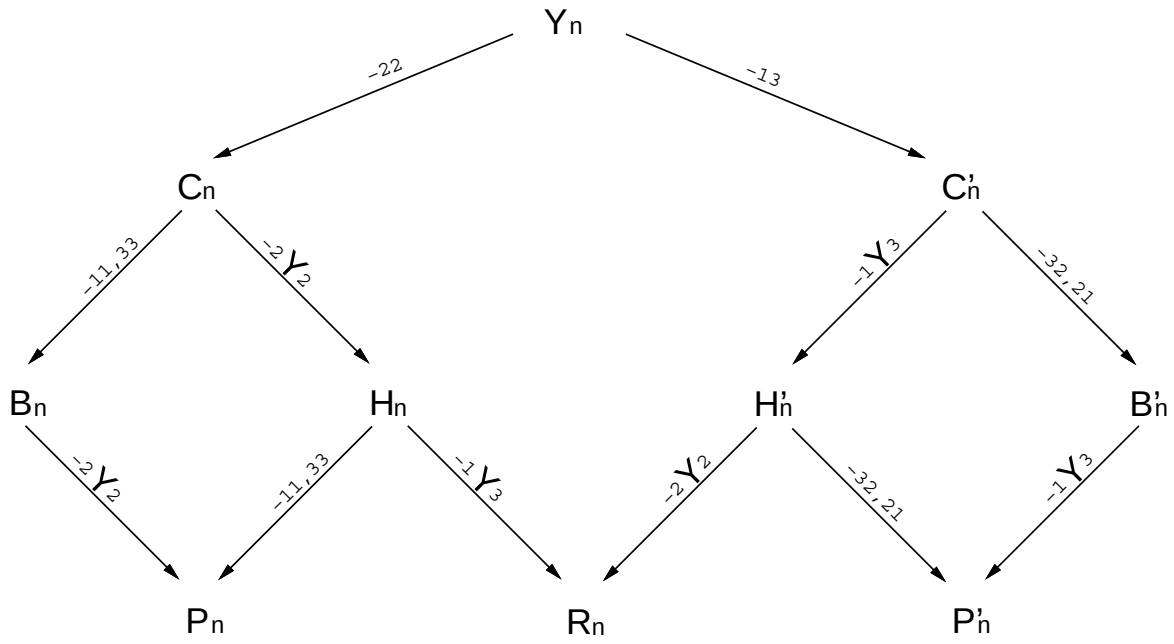


Figure 7.2 Schéma des restrictions sur le langage Y_n.

Le schéma de la figure 7.2 présente les différentes restrictions apportées au langage Y_n codant les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n (ensemble des mots de piles) auxquelles nous nous sommes intéressés.
La notation `-22` signifie que nous considérons les mots de piles ne comportant pas le facteur 22 et la notation `-2Y2` indique que nous interdisons tout facteur de la forme $2g2$ où $g \in Y$.

Nous pouvons remarquer, à partir de la définition du langage C_n [resp. C'_n], que pour tout mot f de C_n [resp. C'_n], le nombre de facteurs $11,33,13$ [resp. $32,21,22$] de f détermine exactement le nombre de chacun des autres facteurs de longueur deux de f.

Tandis que les langages C_n, B_n, H_n et P_n apparaissant dans la partie gauche de la figure 7.2 correspondent à des restrictions naturelles sur les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n, les langages C'_n, B'_n, H'_n et P'_n traduisent des restrictions sur une famille particulière d'arbres 1-2 que nous définissons maintenant et pour laquelle nous énonçons quelques propriétés.

Définition 7.6 *Un* arbre 1-2 filiforme *[resp.* arbre 1-2 filiforme non séparable*] (voir figure 7.3) est un arbre 1-2*

- *ayant autant de points simples que de points doubles (condition* **C1***),*
- *vérifiant qu'à tout instant du parcours préfixe, il y a au moins autant de points simples que de points doubles (condition* **C2***),*
- *tel qu'un sommet fils unique ne peut être une feuille (condition* **C3***) [resp. ne possédant aucun sommet fils unique racine d'un arbre 1-2 filiforme (condition* **C3'***)].*

Notons F_n [resp. $\overline{F}_n$] l'ensemble des arbres 1-2 filiformes [resp. arbres 1-2 filiformes non séparables] ayant n points simples.

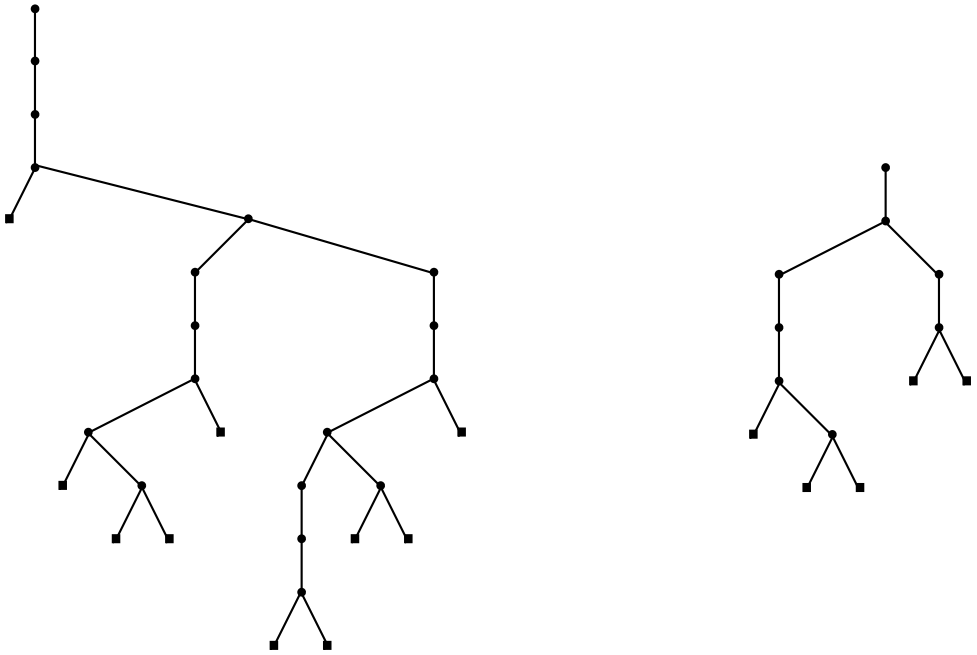


Figure 7.3 Un arbre 1-2 filiforme appartenant à F_9 et un arbre 1-2 filiforme non séparable appartenant à $\overline{F}_4$.

Les mots du langage C'_n [resp. H'_n] (ensemble des mots de piles sans facteur 13 [resp. sans facteur $1g3$ où $g \in Y$]) sont les codages préfixes sur $P_{2,3} \sqcup\!\sqcup \{1\}^*$ des arbres 1-2 filiformes de F_n [resp. $\overline{F}_n$].
Par exemple, les mots 11123211223233311221123323 3 de C'_9 et 121123233123 de H'_4 codent les arbres 1-2 filiformes respectivement de F_9 et de $\overline{F}_4$ illustrés par la figure 7.3.

Propriété 7.7 *Les conditions* **C1** *et* **C3'** *pour un arbre 1-2 suffisent à définir les arbres 1-2 filiformes non séparables.*

Preuve Soit a un arbre 1-2 filiforme non séparable de $\overline{F}_n$ codé par le mot f de H'_n et supposons que a ne respecte pas la condition **C2**. Alors, f admet un facteur droit $f' \in P_{1,2} \sqcup\!\sqcup \{3\}^*$ précédé d'une lettre 1. Si $f' \notin Y$, alors f' admet un facteur gauche $f'' \in P_{2,3} \sqcup\!\sqcup \{1\}^*$ suivi d'une lettre 3. Si $f'' \notin Y$, alors f''

admet un facteur droit $f'''' \in P_{1,2} \sqcup\!\sqcup \{3\}^*$ précédé d'une lettre 1, ce qui nous ramène à l'étape initiale de notre raisonnement. Par induction, nous obtenons alors un facteur f' de f appartenant à Y et précédé de la lettre 1, ou un facteur $1f'$ de f contenant un facteur de la forme $1g3$ où $g \in Y$. Dans les deux cas, ceci est en contradiction avec la condition **C3'**. □

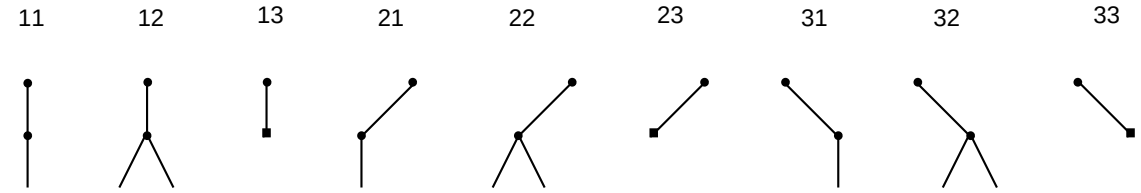
Propriété 7.8 *Le miroir d'un arbre 1-2 filiforme non séparable est également un arbre 1-2 filiforme non séparable.*

Preuve En effet, l'opération miroir sur un arbre 1-2 filiforme ne peut violer que la condition **C2**, et non les conditions **C1** et **C3'**. □

Remarquons que l'ensemble des arbres 1-2 filiformes n'est pas clos par l'opération miroir.

Corollaire 7.9 *Soit a un arbre 1-2 filiforme non séparable de $\overline{F}_n$ et $a^\star$ son miroir codés respectivement par les mots f et $f^\star$ de H'_n (ensemble des mots de piles sans facteur $1g3$ où $g \in Y$). Alors, nous avons en particulier $|f|_{32} = |f^\star|_{22}$, $|f|_{21} = |f^\star|_{31}$ et $|f|_{22} = |f^\star|_{32}$.*

Preuve



Une fois considéré le tableau ci-dessus représentant tous les facteurs de longueur deux sur un arbre 1-2 filiforme, il suffit de constater comment l'opération miroir agissant sur un arbre transforme ces facteurs. □

Nous rappelons maintenant la définition d'une famille de cartes planaires considérée par W.T. Tutte [103] qui interviendra dans la preuve du théorème 7.4.

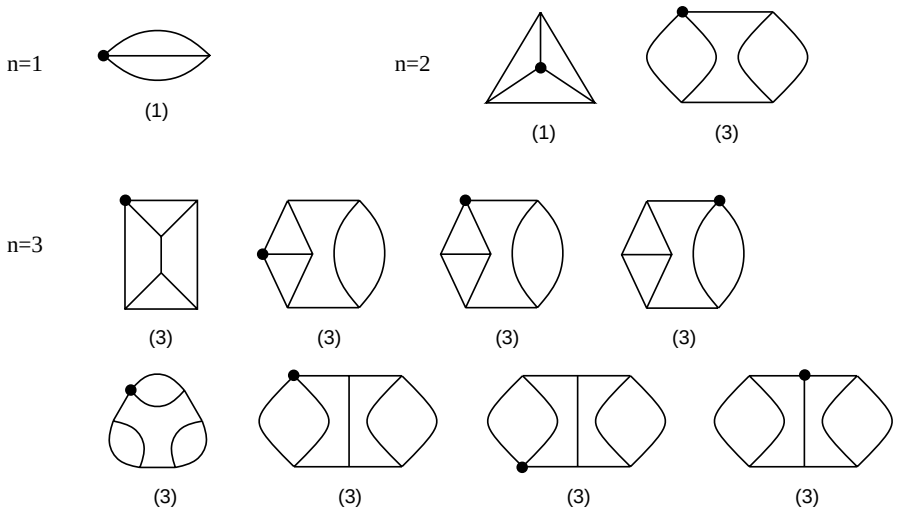


Figure 7.4 Les premières cartes planaires cubiques non séparables (voir exemple 7.11).

Définition 7.10 *Une* carte planaire cubique pointée non séparable *(voir figure 7.4) est une carte planaire sans point d'articulation dont tous les sommets sont de degré trois et pour laquelle un brin est pointé.*
Nous notons CNS_{2n} l'ensemble des cartes planaires cubiques pointées non séparables ayant $2n$ sommets.

Exemple 7.11 *La figure 7.4 présente les cartes planaires cubiques pointées non séparables ayant 2, 4 et 6 sommets, les nombres entre parenthèses indiquant le nombre de cartes différentes obtenues en pointant l'un des brins du sommet distingué repéré par* $\bullet$.

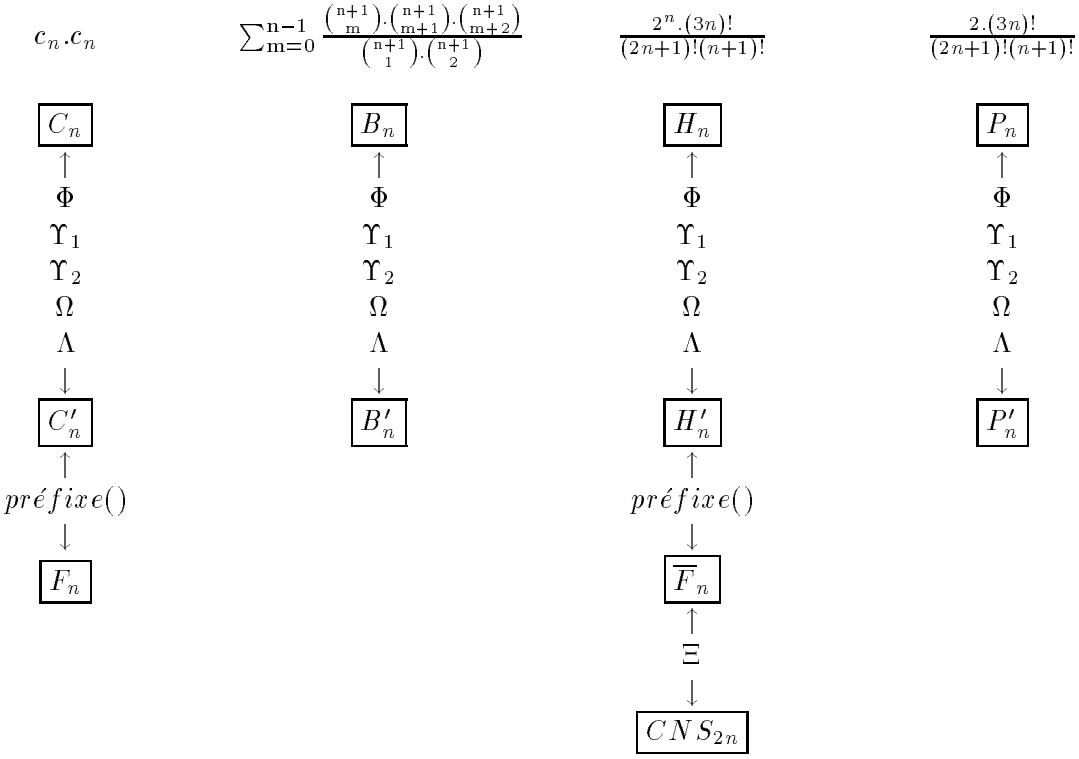


Figure 7.5 Schéma général des correspondances reliant les ensembles des mots de piles.

La figure 7.5 présente brièvement les quatre correspondances reliant C_n, B_n, H_n, P_n respectivement à C'_n, B'_n, H'_n, P'_n que nous allons mettre en évidence par la suite.
Parmi les objets intermédiaires qui seront mis en jeu, nous retrouverons des ensembles de permutations à motifs exclus déjà rencontrés dans les chapitres 5 et 6. Il s'agira des permutations non séparables (excluant simultanément les motifs 2413 et $41\overline{3}52$) et des permutations de Baxter (excluant simultanément les motifs $25\overline{3}14$ et $41\overline{3}52$), alternantes ou non. Signalons également que plusieurs familles de cartes, comme les cartes planaires pointées non séparables, sont reliées aux objets que nous considérons.

Enfin, nous étudierons plusieurs autres restrictions apportées au langage Y_n des mots de piles. Par exemple, nous montrerons que le langage R_n (voir figure 7.2) est directement en

correspondance avec l'ensemble des arbres ternaires complets ayant n sommets internes.

7.1 Tableaux de Young standard rectangulaires de hauteur 3 n'ayant pas deux entiers consécutifs sur la deuxième ligne

Nous établissons maintenant les résultats suivants qui précisent le théorème 7.2.

Théorème 7.12 *Les mots du langage C_n (ensemble des mots de piles sans facteur 22) codant les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur la deuxième ligne sont en correspondance avec les mots du langage C'_n (ensemble des mots de piles sans facteur 13) codant les arbres 1-2 filiformes ayant n points simples. Ils sont dénombrés par*

$$|C_n| = |C'_n| = c_n.c_n$$

Proposition 7.13 *Les mots de $\{f \in C_n : |f|_{11} = n_1, |f|_{33} = n_2, |f|_{13} = n_3\}$ codant les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur la deuxième ligne et possédant n_1 couples d'entiers consécutifs sur la première ligne, n_2 couples d'entiers consécutifs sur la troisième ligne, n_3 couples d'entiers consécutifs situés sur les première et troisième lignes sont en correspondance avec les mots de $\{f' \in C'_n : |f'|_{32} = n_1, |f'|_{21} = n_2, |f'|_{22} = n_3\}$ codant les arbres 1-2 filiformes ayant n points simples et possédant n_1 points doubles fils droits, n_2 points simples fils gauches, n_3 points doubles fils gauches.*

Corollaire 7.14 *Le nombre de tableaux de Young standard rectangulaires de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur la deuxième ligne et possédant i couples d'entiers consécutifs situés sur les première et deuxième lignes et j couples d'entiers consécutifs situés sur les deuxième et troisième lignes est égal au nombre d'arbres 1-2 filiformes ayant n points simples et possédant i fils uniques et j feuilles gauches donné par*

$$|\{f \in C_n : |f|_{12} = i, |f|_{23} = j\}| = |\{f' \in C'_n : |f'|_{12} = i, |f'|_{23} = j\}| =$$

$$\frac{1}{n^2}\binom{n}{i}\binom{n}{i-1}\binom{n}{j}\binom{n}{j-1}$$

Nous présentons successivement les bijections Φ, Υ, Ω et Λ qui conduisent à ces résultats.

7.1.1 Tableaux $3 \times n$ n'ayant pas deux entiers consécutifs sur la deuxième ligne et mélanges de deux mots de parenthèses

Définition 7.15 *Soit $M = \{\alpha \in P_{a,\overline{a}} \sqcup\!\sqcup \; P_{b,\overline{b}} : \forall \alpha = \alpha' b \alpha'', |\alpha'|_a > |\alpha'|_{\overline{a}}\}$ le langage produit de mélange (ou shuffle) de deux langages de parenthèses, et notons $M_{2n} = \{\alpha \in M : |\alpha| = 2n\}$.*

Lemme 7.16 *Il existe une bijection Φ entre mots du produit de mélange de deux mots de parenthèses et mots de piles sans facteur 22. Celle-ci est donnée par le morphisme*

$$\begin{array}{ccccc} \Phi & : & M_{2n} & \longrightarrow & C_n \\ & & \alpha & \longmapsto & f \end{array} \quad \textit{défini par} \quad \begin{cases} \Phi(a) = 1 \\ \Phi(b) = 21 \\ \Phi(\overline{a}) = 23 \\ \Phi(\overline{b}) = 3 \end{cases}$$

Preuve Soient $\alpha \in M_{2n}$ et $f = \Phi(\alpha)$; nous avons alors

- $|\alpha| = 2n, |\alpha|_a = |\alpha|_{\overline{a}}, |\alpha|_b = |\alpha|_{\overline{b}} \Longrightarrow |f|_1 = |f|_2 = |f|_3 = n$,
- $\forall \alpha = \alpha'\alpha'', |\alpha'|_a \geq |\alpha'|_{\overline{a}}, |\alpha'|_b \geq |\alpha'|_{\overline{b}}$ et $\forall \alpha = \alpha' b \alpha'', |\alpha'|_a > |\alpha'|_{\overline{a}} \Longrightarrow \forall f = f'f'', |f'|_1 \geq |f'|_2 \geq |f'|_3$.

De plus, l'ensemble $\{1, 21, 23, 3\}$ constituant un code préfixe, l'application réciproque de Φ est clairement définie. □

Exemple 7.17 *Le mot $a\overline{a}aab\overline{a}\overline{b}b\overline{a}\overline{b}$ de M_{10} est en correspondance par Φ avec le mot* 123112123321233 *de C_5.*

7.1.2 Mélanges de deux mots de parenthèses, permutations de Baxter alternantes et couples d'arbres binaires complets

Afin de résoudre un problème posé par R.C. Mullin [76], R. Cori, S. Dulucq et X. Viennot [18, 26] ont établi le résultat suivant.

Lemme 7.18 *(R. Cori, S. Dulucq et X. Viennot [18]) Il existe une bijection Υ (voir figure 7.7) entre mots du produit de mélange de mots de parenthèses, permutations de Baxter alternantes et couples d'arbres binaires complets.*

$$\begin{array}{ccccccc} \Upsilon & : & M_{2n} & \longrightarrow & \widehat{Baxter}_{2n} & \longrightarrow & A_n \times A_n \\ & & \alpha & \longmapsto & \pi & \longmapsto & (a_1, a_2) \end{array}$$

Ainsi, ces trois familles d'objets sont énumérées par le carré du $n^{ème}$ nombre de Catalan.

La première bijection, notée Υ_1, met en correspondance un mot du produit de mélange de mots de parenthèses α de M_{2n} et une permutation de Baxter alternante π de $\widehat{Baxter}_{2n}$. Partant de l'arbre binaire complet réduit à trois sommets, c'est à dire deux feuilles libres et un sommet interne étiqueté 1, elle consiste en l'application séquentielle des opérateurs correspondant aux lettres $\alpha_2, \alpha_3, \ldots, \alpha_{2n}$ du mot α. Ces opérateurs (voir figure 7.6) agissent sur un arbre binaire complet croissant de la manière suivante :

- opérateur a : étiqueter la feuille gauche libre la plus à droite et lui greffer deux arêtes,
- opérateur b : étiqueter la feuille droite libre la plus à gauche et lui greffer deux arêtes,
- opérateur $\overline{b}$: étiqueter la feuille gauche libre la plus à droite,
- opérateur $\overline{a}$: étiqueter la feuille droite libre la plus à gauche.

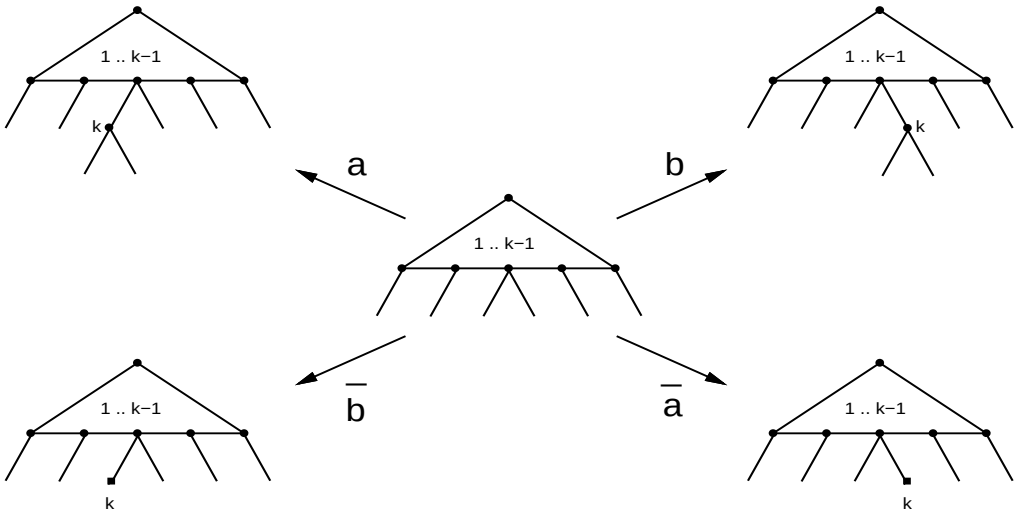


Figure 7.6 Les quatre opérateurs de la bijection Υ_1.

A l'issue de l'application des opérateurs, nous obtenons un arbre binaire complet croissant dont la projection infixe est la permutation π de $\widehat{Baxter}_{2n}$.

La deuxième bijection, notée Υ_2, consiste à prendre respectivement les arbres binaires complets croissant et décroissant de π en oubliant leurs étiquetages. Notons que R. Cori, S. Dulucq et X. Viennot [18] présentent différemment la construction du second arbre.

Exemple 7.19 *La figure 7.7 présente la bijection Υ appliquée au mot de l'exemple 7.17.*

Un examen attentif des bijections Φ et Υ nous conduit à la propriété suivante.

Propriété 7.20 *Soit f un mot de C_n (ensemble des mots de piles sans facteur 22) et une factorisation quelconque $f = f'f''f'''$ telle que $f'' = x2y$ ou $f'' = xy$ avec x et y appartenant à $\{1,3\}$. Soient α le mot du produit de mélange de deux mots de parenthèses tel que $\Phi(\alpha) = f$, π la permutation de Baxter alternante en bijection avec α par Υ_1 et les arbres binaires complets croissant et décroissant associés à π. Soit $p = |f'x|_1 + |f'x|_3$.*
Alors, le tableau de la figure 7.8 indique quels sont les liens entre le facteur f'' de f, le facteur $\alpha_p\alpha_{p+1}$ de α, les positions de p et $p+1$ dans π ainsi que dans les arbres binaires complets croissant $abc(\pi)$ et décroissant $abd(\pi)$ associés.

Voici quelques commentaires supplémentaires pour une lecture plus aisée du tableau de la figure 7.8. La première colonne correspond aux huit possibilités pour le facteur f'', la deuxième colonne donne les seize facteurs possibles $\alpha_p\alpha_{p+1}$ sur l'alphabet $\{a, \overline{a}, b, \overline{b}\}$, la troisième colonne regroupe pour la permutation π les positions de p et $p+1$ et les facteurs se trouvant dans l'intervalle correspondant (définition 6.1 des permutations de Baxter), les quatrième et cinquième colonnes précisent les positions respectives de p et $p+1$ dans les arbres binaires complets croissant et décroissant associés à π en indiquant le rang (ordre infixe) des sommets supportant ces étiquettes.

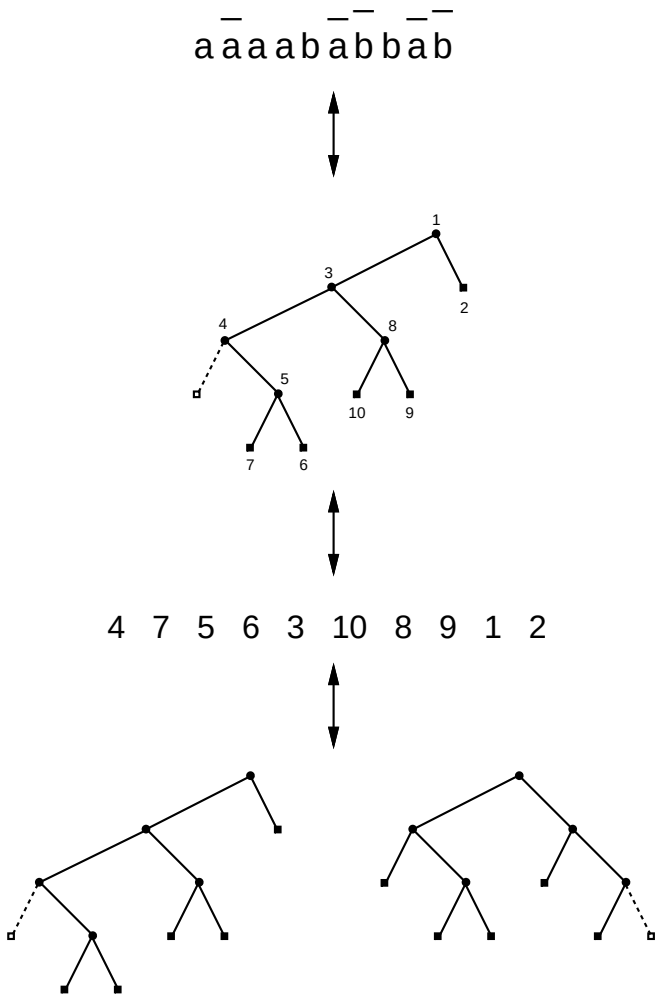


Figure 7.7 La bijection Υ entre un mot du produit de mélange de deux mots de parenthèses, une permutation de Baxter alternante et son arbre binaire croissant, et un couple d'arbres binaires complets.

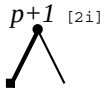
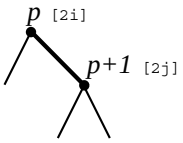
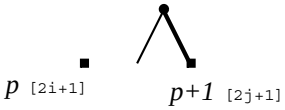
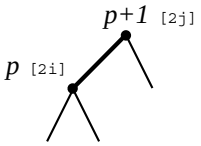
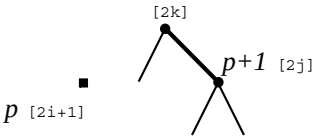
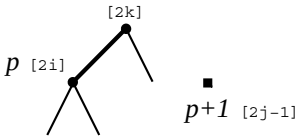
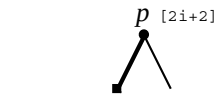
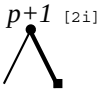
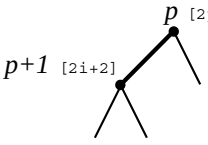
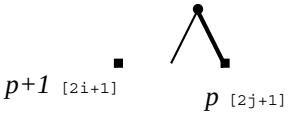
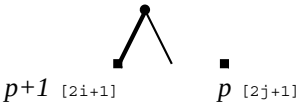
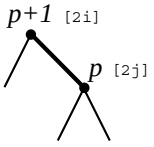
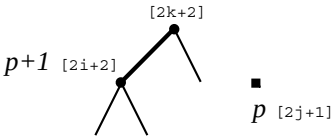
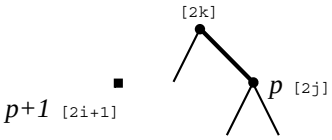
f''	$\alpha_p\alpha_{p+1}$	π	$abc(\pi)$	$abd(\pi)$
123	$a\overline{a}$ ou $b\overline{a}$	$\pi' p(p+1)\pi''$		
121	ab ou bb	$\pi' p \overset{\geq}{\pi} (p+1)\pi''$		
323	$\overline{a}\overline{a}$ ou $\overline{b}\overline{a}$	$\pi' p \overset{\leq}{\pi} (p+1)\pi''$		
321	$\overline{a}b$ ou $\overline{b}b$	$\pi' p \overset{\leq\geq}{\pi} (p+1)\pi''$		
13	$a\overline{b}$ ou $b\overline{b}$	$\pi'(p+1)p\pi''$		
11	aa ou ba	$\pi'(p+1) \overset{\geq}{\pi} p\pi''$		
33	$\overline{a}\overline{b}$ ou $\overline{b}\overline{b}$	$\pi'(p+1) \overset{\leq}{\pi} p\pi''$		
31	$\overline{a}a$ ou $\overline{b}a$	$\pi'(p+1) \overset{\geq\leq}{\pi} p\pi''$		

Figure 7.8 Relations liant le mot f de C_n, le mot α de M_{2n}, la permutation π de $\widehat{Baxter}_{2n}$ se correspondant par Φ et Υ.

Preuve Toutes ces relations se déduisent directement du morphisme Φ et des opérateurs de la bijection Υ_1. Les seuls points qui demandent une attention particulière sont, pour l'arbre binaire décroissant $abd(\pi)$, le cas où $\alpha_{p+1} = b$ [resp. a] qui imposent à p d'être l'extrémité d'une arête gauche [resp. droite] ce qui correspond à l'interdiction du motif $41\overline{3}52$ [resp. $25\overline{3}14$], motifs absents dans les permutations de Baxter. □

Remarque 7.21 *La composition des bijections Φ et Υ est équivalente à la construction décrite par la figure 7.9 qui permet de mettre directement en correspondance les mots de C_n (ensemble des mots de piles sans facteur 22) et les arbres binaires complets croissant et décroissant d'une permutation de Baxter alternante sur $[2n]$, et donc avec les couples d'arbres binaires complets de $A_n \times A_n$.*

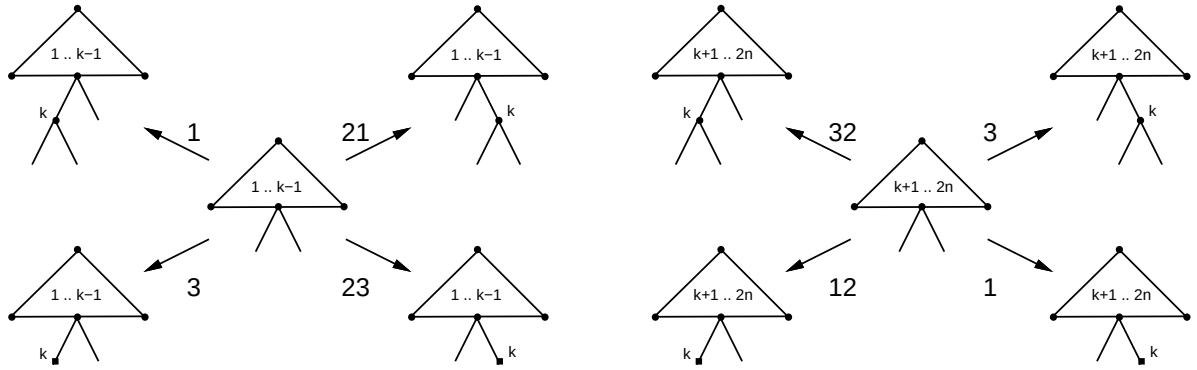


Figure 7.9 Les opérateurs permettant de construire directement les arbres binaires complets croissant et décroissant d'une permutation de Baxter alternante depuis un mot de piles sans facteur 22.

Cette construction est directement inspirée de celle de R. Cori, S. Dulucq et X. Viennot [18]; en particulier, la construction de l'arbre binaire croissant est identique. L'arbre binaire complet croissant [resp. décroissant] de la permutation de Baxter alternante π s'obtient en lisant les facteurs $1, 21, 3, 23$ [resp. $32, 3, 12, 1$] du mot f de C_n parcouru de gauche à droite [resp. de droite à gauche] et en appliquant l'opérateur correspondant; la figure 7.9 illustre l'opération à effectuer pour le $k^{ème}$ facteur rencontré, avec k allant de 1 à $2n$ [resp. de $2n$ à 1].

7.1.3 Couples d'arbres binaires complets et arbres 1-2 filiformes

Définition 7.22 *Nous désignons par Ω le codage des couples d'arbres binaires complets défini par*

$$\begin{array}{rcccc} \Omega & : & A_n \times A_n & \longrightarrow & P_{2,3} \times P_{1,2} \\ & & (a_1, a_2) & \longmapsto & (suffixe(a_1), préfixe(a_2)) \end{array}$$

où $préfixe$ et $suffixe$ sont les codages des arbres binaires complets définis dans la sous-section 1.3.3.

Exemple 7.23 *Au couple d'arbres binaires complets de l'exemple 7.19 (voir figure 7.7) correspond par* Ω *le couple de mots de parenthèses* $(2233223323, 1121221212)$.

Lemme 7.24 *Il existe une bijection* Λ *entre mots de piles sans facteur* 13 *et couples d'arbres binaires complets de même taille. Celle-ci est donnée par le morphisme*

$$\Lambda \ : \ \begin{array}{ccc} C'_n & \longrightarrow & P_{2,3} \times P_{1,2} \\ f' & \longmapsto & (a,b) \end{array} \quad \textit{défini par} \begin{cases} \Lambda(1) = (\varepsilon, 1) \\ \Lambda(2) = (2,2) \\ \Lambda(3) = (3, \varepsilon) \end{cases}$$

Preuve Λ est clairement une application de C'_n vers $P_{2,3} \times P_{1,2}$ (codant $A_n \times A_n$).
Réciproquement, soit $(a,b) \in P_{2,3} \times P_{1,2}$ avec $|a| = |b| = 2n$. Alors, a et b se factorisent de manière unique en $a = 23^{k_1}23^{k_2}\ldots23^{k_n}$ et $b = 1^{l_1}21^{l_2}2\ldots1^{l_n}2$. Le mot $f' = 1^{l_1}23^{k_1}1^{l_2}23^{k_2}\ldots1^{l_n}23^{k_n}$ appartient à C'_n et vérifie $\Lambda(f') = (a,b)$. □

Exemple 7.25 *Le couple de mots de parenthèses* $(2233223323, 1121221212)$ *de* $P_{2,3} \times P_{1,2}$ *de l'exemple 7.23 est en bijection par* Λ *avec le mot* 112123321233123 *de* C'_5.

Remarque 7.26 *Notons que le même morphisme* Λ *mettrait en bijection les mots de piles sans facteur* 31 *et les couples d'arbres binaires complets de même taille. Dans ce cas, son application réciproque consisterait à placer, entre deux lettres* 2 *successives, le bloc de lettres* 3 *à la droite du bloc de lettres* 1.

Preuve du théorème 7.12. Clairement, la composition des bijections Φ, Υ_1, Υ_2, Ω et Λ met en correspondance les langages C_n et C'_n, avec les couples d'arbres binaires complets de $A_n \times A_n$ ce qui nous permet d'en déduire la formule d'énumération. □

Preuve de la proposition 7.13. Soit f un mot de C_n tel que $|f|_{11} = n_1$, $|f|_{33} = n_2$, $|f|_{13} = n_3$.
Soient α, π, (u,v) , (a,b) et f' appartenant respectivement à M_{2n}, $\widehat{Baxter}_{2n}$, $P_{x,\overline{x}} \times P_{y,\overline{y}}$ (codant $A_n \times A_n$), $P_{2,3} \times P_{1,2}$ et C'_n en correspondance avec f successivement par les bijections Φ, Υ_1, Υ_2, Ω et Λ.
Nous déduisons de ces bijections et de la propriété 7.20 les relations suivantes.

- $|\alpha|_{aa} + |\alpha|_{ba} = n_1$, $|\alpha|_{\overline{a}\overline{b}} + |\alpha|_{\overline{b}\overline{b}} = n_2$, $|\alpha|_{a\overline{b}} + |\alpha|_{b\overline{b}} = n_3$.
- $|\{p \in [2n-1] : \pi = \pi'(p+1) \overset{>}{\pi} p\pi''\}| = n_1$,
  $|\{p \in [2n-1] : \pi = \pi'(p+1) \overset{<}{\pi} p\pi''\}| = n_2$,
  $|\{p \in [2n-1] : \pi = \pi'(p+1)p\pi''\}| = n_3$.
- $|\{i \in [n-1] : u = u'\overline{x}\overline{x}u'', v = v'\overline{y}\overline{y}v''; |u'\overline{x}|_{\overline{x}} = |v'\overline{y}|_{\overline{y}} = i\}| = n_1$,
  $|\{i \in [n-1] : u = u'x\overline{x}u'', v = v'\overline{y}yv''; |u'x|_{\overline{x}} = |v'\overline{y}|_{\overline{y}} = i\}| = n_2$,
  $|\{i \in [n-1] : u = u'x\overline{x}u'', v = v'\overline{y}\overline{y}v''; |u'x|_{\overline{x}} = |v'\overline{y}|_{\overline{y}} = i\}| = n_3$.
- $|\{i \in [n-1] : a = a'32a'', b = b'22b''; |a'3|_2 = |b'2|_2 = i\}| = n_1$,
  $|\{i \in [n-1] : a = a'22a'', b = b'21b''; |a'2|_2 = |b'2|_2 = i\}| = n_2$,
  $|\{i \in [n-1] : a = a'22a'', b = b'22b''; |a'2|_2 = |b'2|_2 = i\}| = n_3$.
- $|f'|_{32} = n_1$, $|f'|_{21} = n_2$, $|f'|_{22} = n_3$.

Ceci nous assure donc le résultat. □

Preuve du corollaire 7.14. Nous déduisons de la propriété 7.20 et de la distribution des arbres binaires complets selon le nombre de sommets et de feuilles gauches (ou droites), donnée par les nombres

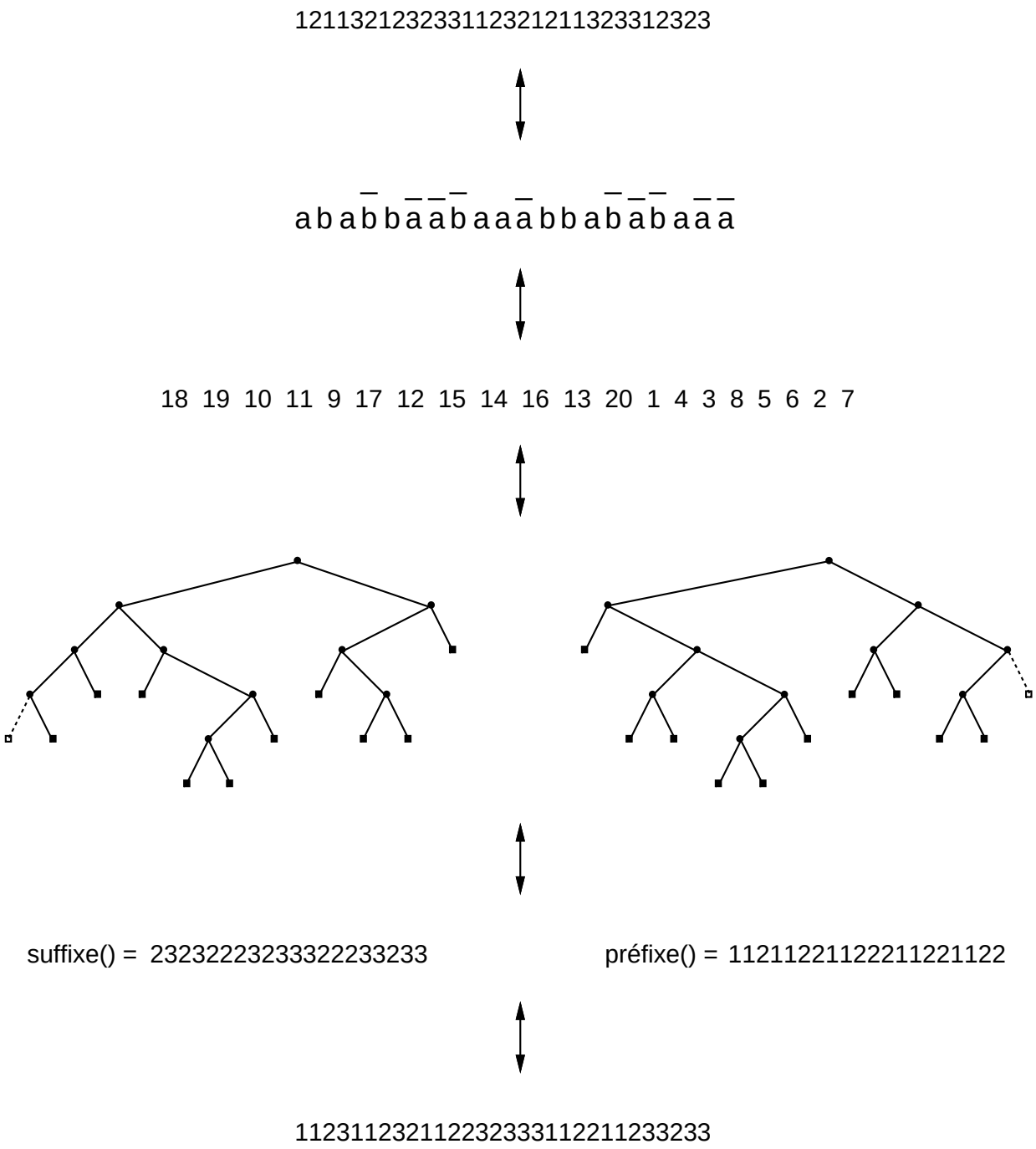


Figure 7.10 La correspondance entre un mot de piles sans facteur 22 et un mot de piles sans facteur 13.

de Narayana, la formule pour le langage C_n. La même distribution pour C'_n est une conséquence de la proposition 7.13. □

Exemple 7.27 *La figure 7.10 illustre la correspondance composant les bijections* Φ, Υ_1, Υ_2, Ω *et* Λ, *permettant ainsi de relier les langages* C_n *(ensemble des mots de piles sans facteur* 22*) et* C'_n *(ensemble des mots de piles sans facteur* 13*).*

7.2 Tableaux de Young standard rectangulaires de hauteur 3 n'ayant pas deux entiers consécutifs sur une même ligne

Nous établissons maintenant les résultats suivants qui précisent le théorème 7.3.

Théorème 7.28 *Les mots du langage* B_n *(ensemble des mots de piles sans facteur* $22, 11, 33$*) codant les tableaux de Young standard rectangulaires de hauteur* 3 *et de longueur* n *n'ayant pas deux entiers consécutifs sur une même ligne sont en correspondance avec les mots du langage* B'_n *(ensemble des mots de piles sans facteur* $13, 32, 21$*) codant les arbres 1-2 filiformes ayant* n *points simples et ne possédant aucun point double fils droit ni aucun point simple fils gauche. Ils sont dénombrés par*

$$|B_n| = |B'_n| = \sum_{m=0}^{n-1} \frac{\binom{n+1}{m}.\binom{n+1}{m+1}.\binom{n+1}{m+2}}{\binom{n+1}{1}.\binom{n+1}{2}}$$

Proposition 7.29 *Les tableaux de Young standard rectangulaires de hauteur* 3 *et de longueur* n *n'ayant pas deux entiers consécutifs sur une même ligne et possédant* m *couples d'entiers consécutifs situés sur les première et troisième lignes sont en correspondance avec les arbres 1-2 filiformes ayant* n *points simples et ne possédant aucun point double fils droit ni aucun point simple fils gauche et possédant* m *points doubles fils gauches. Ils sont dénombrés par*

$$|\{f \in B_n : |f|_{13} = m\}| = |\{f' \in B'_n : |f'|_{22} = m\}| = \frac{\binom{n+1}{m}.\binom{n+1}{m+1}.\binom{n+1}{m+2}}{\binom{n+1}{1}.\binom{n+1}{2}}$$

Pour établir ces résultats, nous caractérisons sur la correspondance entre mots de C_n et de C'_n, la restriction apportée à C_n pour obtenir B_n (interdiction des facteurs 11 et 33).

Lemme 7.30 *Le morphisme* Φ *(voir lemme 7.16) est une bijection entre* B_n *(ensemble des mots de piles sans facteur* $22, 11, 33$*) et* $\widetilde{M}_{2n} = \{\alpha \in M_{2n} : |\alpha|_{aa} = |\alpha|_{ba} = |\alpha|_{\overline{a}\overline{b}} = |\alpha|_{\overline{b}\overline{b}} = 0\}$ *(langage du produit de mélange de deux mots de parenthèses sans facteur* aa, ba, $\overline{a}\overline{b}$, $\overline{b}\overline{b}$*).*

Preuve C'est une conséquence directe de la définition du morphisme Φ. □

Définition 7.31 *Soit* $\widetilde{Baxter}_{2n}$ *l'ensemble des permutations de Baxter alternantes* π *telles que, pour tout* $p \in [2n-1]$, *si* $\pi = \pi'(p+1)\overset{\geq}{\pi}\overset{\leq}{\pi}p\pi''$ *alors* $\overset{\geq}{\pi} = \varepsilon \iff \overset{\leq}{\pi} = \varepsilon$ *(*$\overset{\geq}{\pi}$ *et* $\overset{\leq}{\pi}$ *sont soit tous deux vides, soit tous deux non vides).*

Lemme 7.32 *La bijection Υ (voir lemme 7.18) met en correspondance le langage $\widetilde{M}_{2n}$, l'ensemble des permutations $\widetilde{Baxter}_{2n}$ et l'ensemble des arbres binaires jumeaux J_n (voir définition 6.14) complétés.*

Preuve C'est une conséquence de la propriété 7.20 en appliquant les restrictions aux objets considérés. Clairement, $\widetilde{Baxter}_{2n}$ est bien l'ensemble recherché compte-tenu de sa définition.
La restriction sur le couple d'arbres binaires complets se traduit par la caractérisation des arbres binaires jumeaux de J_n complétés. En effet, pour tout $p \in [2n-1]$, une feuille gauche de l'arbre binaire complet croissant étiquetée $p+1$ est indicée $2i+1$ dans l'ordre infixe si et seulement si une feuille droite de l'arbre binaire complet décroissant étiquetée p a le même indice (exception faite des deux feuilles extrêmes, c'est à dire pour tout $i \in [n-1]$). Réciproquement, montrons que si une permutation π appartient à $\widehat{Baxter}_{2n} \backslash \widetilde{Baxter}_{2n}$, alors les arbres $abc(\pi)$ et $abd(\pi)$ effeuillés ne sont pas jumeaux. En effet, d'après la sixième [resp. septième] ligne du tableau (figure 7.8) de la propriété 7.20, les $(2j+1)^{ème}$ [resp. $(2i+1)^{ème}$] sommets de $abc(\pi)$ et $abd(\pi)$ sont tous deux des feuilles droites [resp. gauches]. □

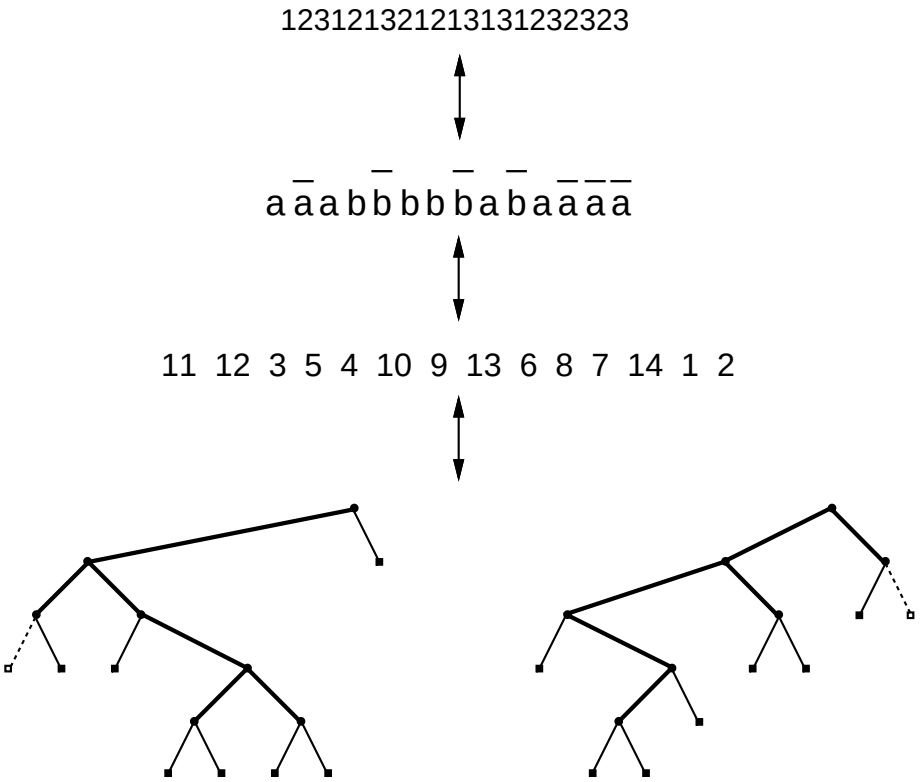


Figure 7.11 La correspondance entre mot de piles sans facteur $22, 11, 33$ et arbres binaires jumeaux complétés.

Preuve du théorème 7.28 et de la proposition 7.29. Les bijections Φ, Υ et Ψ (voir théorème 6.16) mettant en correspondance les mot de piles sans facteur $22, 11, 33$ et les permutations de Baxter, nous déduisons du corollaire 6.23 (et de la propriété 7.20) la formule dénombrant les mots du langage B_n selon

le nombre de facteurs 13. La proposition 7.13 nous permet de conclure pour les mots du langage B'_n selon le nombre de facteurs 22. □

Exemple 7.33 *La figure 7.11 présente les bijections* Φ *et* Υ *appliquées à un mot de piles sans facteur* $22, 11, 33$.

7.3 Tableaux de Young standard rectangulaires non séparables de hauteur 3

Nous établissons maintenant les résultats suivants qui précisent le théorème 7.4.

Théorème 7.34 *Les mots du langage* H_n *(ensemble des mots de piles sans facteur* $2g2$ *où* $g \in Y$*, c'est à dire vérifiant la condition "tour de Hanoï") codant les tableaux de Young standard rectangulaires non séparables de hauteur* 3 *et de longueur* n *sont en correspondance avec les mots du langage* H'_n *(ensemble des mots de piles sans facteur* $1g3$ *où* $g \in Y$*) codant les arbres 1-2 filiformes non séparables ayant* n *points simples, eux-mêmes en bijection avec les cartes planaires cubiques pointées non séparables ayant* $2n$ *sommets de* CNS_{2n}*. Ils sont dénombrés par*

$$|H_n| = |H'_n| = |CNS_{2n}| = \frac{2^n.(3n)!}{(2n+1)!(n+1)!}$$

De plus, cette correspondance met en bijection les mots f *de* H_n *tels que* $|f|_{11} = n_1, |f|_{33} = n_2, |f|_{13} = n_3$ *et les mots* f' *de* H'_n *tels que* $|f'|_{32} = n_1, |f'|_{21} = n_2, |f'|_{22} = n_3$.

Afin de prouver ce résultat, nous étudions la restriction apportée à C_n pour obtenir H_n (interdiction du facteur $2g2$ pour tout $g \in Y$) sur la composition des bijections Φ, Υ_1, Υ_2, Ω et Λ entre mots de C_n et de C'_n. Ensuite, nous montrons que les mots de H'_n sont les mots du langage de Lehman-Lenormand [70] codant les cartes planaires cubiques pointées non séparables ayant $2n$ sommets de CNS_{2n}.

Nous présentons au préablable deux conjectures.

Conjecture 7.35 *Les mots de* $H_{n,m} = \{f \in H_n : |f|_{11} + |f|_{33} = m\}$ *codant les tableaux de Young standard rectangulaires non séparables de hauteur* 3 *et de longueur* n *et possédant* m *couples d'entiers consécutifs sur les première et troisième lignes et les mots de* $H'_{n,m} = \{f' \in H'_n : |f'|_{32} + |f'|_{21} = m\}$ *codant les arbres 1-2 filiformes non séparables ayant* n *points simples et possédant* m *points doubles fils droits et points simples fils gauches sont au nombre de*

$$|H_{n,m}| = |H'_{n,m}| = \binom{n-1}{m} \frac{2.(3n)!}{(2n+1)!(n+1)!}$$

Notons une conséquence du théorème 7.34 et du corollaire 7.9 permettant d'affirmer que, pour tout $m \in [0, n-1]$, les langages $H_{n,m}$, $H'_{n,m}$, $H'_{n,n-1-m}$ et $H_{n,n-1-m}$ sont en bijection.

Conjecture 7.36 *Les mots de $H_n \setminus \{\mathcal{A}^*33\mathcal{A}^*\}$ codant les tableaux de Young standard rectangulaires non séparables de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur la troisième ligne et les mots de $H'_n \setminus \{\mathcal{A}^*21\mathcal{A}^*\}$ codant les arbres 1-2 filiformes non séparables ayant n points simples et ne possédant aucun point simple fils gauche sont au nombre de*

$$\frac{2.(4n+1)!}{(n+1)!(3n+2)!}$$

Rappelons que cette formule dénombre certaines cartes planaires considérées par W.T. Tutte, à savoir les cartes planaires cubiques pointées non séparables 3-connexes [103] ou encore les triangulations planaires [102].

7.3.1 Tableaux $3 \times n$ non séparables et arbres 1-2 filiformes non séparables

Lemme 7.37 *Le morphisme Φ (voir lemme 7.16) met en correspondance les mots du langage H_n (ensemble des mots de piles sans facteur $2g2$ où $g \in Y$) et les mots du langage $\overline{M}_{2n} = \{\alpha \in M_{2n} : \forall \alpha = \alpha' b\beta x\alpha''$ où $x \in \{\overline{a}, b\}, a\beta \notin M\}$ (langage des mots non séparables du produit de mélange de deux mots de parenthèses).*

Preuve Ce résultat est une conséquence directe de la définition du morphisme Φ. □

Lemme 7.38 *La bijection Υ_1 (voir lemme 7.18) met en correspondance les mots du langage $\overline{M}_{2n}$ et les permutations de $\widehat{NSép}_{2n} = \widehat{S}_{2n}(2413, 41\overline{3}52)$ (ensemble des permutations non séparables alternantes).*

Preuve Remarquons tout d'abord qu'exclure le motif 2413 revient à exclure simultanément les motifs $25\overline{3}14$ et 25314. Ainsi, l'ensemble des permutations non séparables alternantes est égal à l'ensemble des permutations de Baxter alternantes n'admettant pas de surcroit le motif 25314.
Soient π une permutation de $\widehat{Baxter}_{2n}$ et α un mot de M_{2n} en bijection par Υ_1. Il nous faut donc montrer que α contient un facteur $b\beta\overline{a}$ ou $b\beta b$ avec $a\beta$ appartenant à M si et seulement si π contient une sous-suite de type 25314.
Plus précisément, nous allons prouver que si α admet une telle factorisation ($\alpha \in M_{2n} \setminus \overline{M}_{2n}$), alors l'arbre binaire complet croissant obtenu par application des opérateurs du mot α (voir figure 7.6) est exactement de la forme présentée figure 7.12, ce qui conduit à l'obtention d'une sous-suite *rtpsq* de type 25314 pour la permutation π.
Réciproquement, nous montrons que si une permutation de Baxter alternante π admet une telle sous-suite ($\pi \in \widehat{Baxter}_{2n} \setminus \widehat{NSép}_{2n}$), alors son arbre binaire complet croissant ne peut être que de la forme indiquée par la figure 7.12. De ce fait, la suite des opérateurs appliquée ne peut correspondre qu'à un mot α se factorisant en $\alpha = \alpha' b\beta x\alpha''$ avec $x = \overline{a}$ ou $x = b$ et $a\beta$ appartenant à M.

- Considérons la factorisation $\alpha = \alpha' b\beta x\alpha''$ avec $a\beta$ appartenant à M et $x \in \{\overline{a}, b\}$. Notons $p = |\alpha' b|$ et $q = |\alpha' b\beta x|$.
  Après avoir appliqué successivement tous les opérateurs du facteur α', nous sommes dans la situation où la feuille située à l'extrémité de la branche gauche est libre (c'est le cas pour tout facteur gauche de α) et au moins deux feuilles droites sont libres (les opérateurs suivants sont ceux de

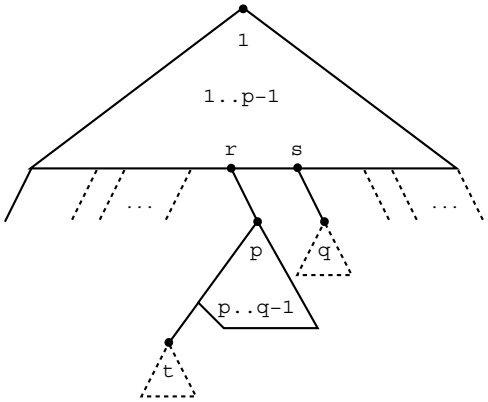


Figure 7.12 Arbre binaire complet croissant correspondant aux mots séparables du produit de mélange de deux mots de parenthèses et aux permutations de Baxter alternantes admettant une sous-suite de type $rtpsq$.

$b\beta$). Soient r et s les étiquettes des pères respectivement des première et deuxième feuilles libres à droite. Par construction, r est supérieur à s puisque Υ_1 étiquette la feuille libre à droite la plus à gauche.

Appliquons maintenant les opérateurs du facteur $b\beta$. Alors, le sous-arbre droit du sommet étiqueté r est un arbre dont tous les sommets, excepté la dernière feuille de la branche gauche qui est libre, sont étiquetés par les entiers de p à $q-1$.

Ensuite, l'opérateur $x \in \{\overline{a}, b\}$ étiquette q le fils droit du sommet d'étiquette s.

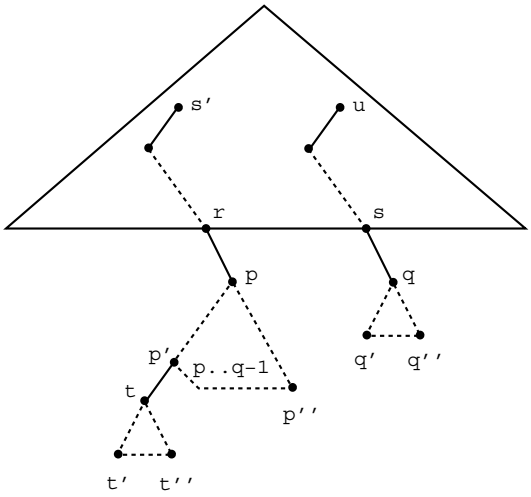
Enfin, α'' contient au moins une lettre a ou $\overline{b}$ ayant pour effet d'étiqueter t la feuille libre de la branche gauche du sous-arbre issu du sommet d'étiquette p obtenu après l'application des opérateurs du facteur $\alpha' b\beta$.

Ainsi, l'arbre binaire complet croissant obtenu est du type de celui présenté figure 7.12.

- Choisissons tout d'abord les éléments r et s de la sous-suite $rtpsq$ de type 25314.
  Soit $e_1e_2e_3e_4$ une sous-suite quelconque de π de type 2413.
  Il est possible de choisir r et t', en remplacement respectivement de e_1 et e_2, de sorte que r soit juste à gauche de t' et que la sous-suite $rt'e_3e_4$ soit également de type 2413. Pour cela, prenons r l'élément appartenant à $]e_3, e_4[$ situé entre e_1 (inclus) et e_2 (exclu) et le plus à droite. De même, prenons t' l'élément supérieur à e_4 situé entre r (exclu) et e_2 (inclus) et le plus à gauche. Ainsi, il ne peut pas y avoir d'élément e situé entre r (exclu) et t' (exclu) car comme cet élément devrait être inférieur à e_3, la sous-suite $ret'e_3$ serait de type 3142 mais ne ferait pas elle-même partie d'une sous-suite de type 41352.
  Il est également possible de choisir s et q', en remplacement respectivement de e_3 et e_4, de sorte que s soit juste à gauche de q' et que la sous-suite $rt'sq'$ soit également de type 2413. Pour cela, prenons s l'élément inférieur à r situé entre e_3 (inclus) et e_4 (exclu) et le plus à droite. De même, prenons q' l'élément appartenant à $]r, t'[$ situé entre s (exclu) et e_4 (inclus) et le plus à gauche. Ainsi, il ne peut pas y avoir d'élément e situé entre s (exclu) et q' (exclu) car comme cet élément devrait être supérieur à t', la sous-suite $t'seq'$ serait de type 3142 mais ne ferait pas elle-même partie d'une sous-suite de type 41352.
  Alors, parmi l'ensemble de telles sous-suites $rt'sq'$ de π, prenons-en une avec $\pi^{-1}(s) - \pi^{-1}(r)$

minimal.

Nous allons reconstituer l'arbre binaire complet croissant de π (partiellement étiqueté) illustré ci-dessous.



Recherchons maintenant les éléments q et p de la sous-suite $rtpsq$ de type 25314.

Dans l'arbre binaire complet croissant, r [resp. s] étiquette un sommet interne car il est inférieur à t' [resp. q'] situé à sa droite dans π.

Soit q l'étiquette du fils droit du sommet étiqueté s dans l'arbre binaire complet croissant. Alors, q appartient à $]r, q']$. En effet, l'étiquetage de l'arbre binaire complet étant croissant, q doit appartenir à $]s, q']$. De plus, q est supérieur à r car sinon la sous-suite $rsq'q$ serait de type 3142 mais ne ferait pas elle-même partie d'une sous-suite de type 41352.

Soit p le plus petit des éléments supérieurs à r situé entre t' (exclu) et s (exclu). Tout d'abord, p existe et est inférieur à q car la sous-suite $rt'sq$ de type 2413 doit faire elle-même partie d'une sous-suite de type 25314. Ensuite, tous les éléments e situés entre t' (exclu) et p (exclu) sont supérieurs à p car sinon la sous-suite $rt'ep$ serait de type 2413 mais ne ferait pas elle-même partie d'une sous-suite de type 25314. Enfin, t' n'est pas le fils droit de r dans l'arbre binaire complet croissant car sinon la sous-suite $rt's'q$ (où s' serait le successeur de t' dans π) serait de type 2413 (s' est un ancêtre de r) mais ne ferait pas elle-même partie d'une sous-suite de type 25314. Nous en déduisons que p est l'étiquette du fils droit de r dans l'arbre binaire complet croissant.

Montrons maintenant que tous les éléments appartenant à $[p, q[$ sont consécutifs et constituent à eux tous le sous-arbre de racine étiquetée p dans l'arbre binaire complet croissant.

Aucun élément e situé à gauche de r ne peut appartenir à $]p, q[$ car sinon la sous-suite $ert'p$ serait de type 3142 mais ne ferait pas elle-même partie d'une sous-suite de type 41352. De plus, tous les éléments e situés entre p et s sont inférieurs à q car sinon la sous-suite $pesq$ serait de type 2413 et donc plus minimale que la sous-suite $rt'sq$.

Soit t'' l'élément supérieur à q situé entre t' (inclus) et p (exclu) et le plus à droite. Alors, tous les éléments e situés entre r et t'' sont supérieurs à q car sinon la sous-suite $et''sq$ serait de type 2413 et donc plus minimale que la sous-suite $rt'sq$.

p étant inférieur à son successeur dans π, le sommet d'étiquette p est un sommet interne. Soit p'' l'étiquette du dernier sommet de la branche droite de racine le sommet d'étiquette p et soit s' (éventuellement s) le successeur de p'' dans π. s' est inférieur à r car c'est l'un de ses ancêtres dans

l'arbre binaire complet croissant. Alors, tous les éléments e situés entre s' et s sont inférieurs à r car sinon la sous-suite $rt's'e$ serait de type 2413 et donc plus minimale que la sous-suite $rt'sq$.
Soit q'' l'étiquette du dernier sommet de la branche droite de racine le sommet d'étiquette q ; tous les éléments situés entre q' (inclus) et q'' (inclus) valent au moins q. Soit u l'élément situé à droite de q'' dans π, u étant inférieur à s car c'est l'un de ses ancêtres; alors, aucun élément e situé à droite de u ne peut appartenir à $]p,q[$ car sinon la sous-suite $sq''ue$ serait de type 2413 mais ne ferait pas elle-même partie d'une sous-suite de type 25314.
Recherchons finalement l'élément t de la sous-suite $rtpsq$ de type 25314.
Soit t la plus petite des étiquettes supérieures à q appartenant à la branche gauche de racine le sommet d'étiquette p. Le père du sommet d'étiquette t est un sommet d'étiquette p' inférieure à q et p' est le successeur de t'' dans π puisque toutes les étiquettes des sommets appartenant au sous-arbre du sommet d'étiquette t doivent être au moins égales à t. En particulier, t'' est l'étiquette du dernier sommet de la branche droite du sommet d'étiquette t.
L'arbre binaire complet croissant de la permutation π ainsi reconstitué correspond effectivement à celui présenté figure 7.12.

□

Définition 7.39 *L'ensemble des* arbres binaires complets séparables D_n *(voir figure 7.13) est l'ensemble*

$$D_n = \{(a_1, a_2) : a_1, a_2 \in A_n \text{ et } \Delta_d(a_1) \cap \Delta_g(a_2) \neq \emptyset\}$$

où

- Δ_d *est l'ensemble des couples d'entiers* $(x-1, y-1)$ *tels que* x *et* y *sont les numéros d'ordre infixe respectivement d'un sommet interne* s *et d'une feuille* t *appartenant à la branche respectivement gauche et droite d'un même sommet interne droit de* a_1,
- Δ_g *est l'ensemble des couples d'entiers* (x, y) *tels que* x *et* y *sont les numéros d'ordre infixe respectivement d'une feuille* s *et d'un sommet interne* t *appartenant à la branche respectivement gauche et droite d'un même sommet interne gauche de* a_2.

Ainsi, deux arbres binaires complets séparables (a_1, a_2) *possèdent chacun un sous-arbre tronqué (la branche principale est rompue) qui coïncident relativement à la numérotation en ordre infixe des sommets (à une unité près). Plus précisément, le sous-arbre de* a_1 *[resp.* a_2*] est tronqué à gauche [resp. droite] et est issu d'une arête droite [resp. gauche].*

Exemple 7.40 *La figure 7.13 représente deux arbres binaires complets séparables* (a_1, a_2) *de* D_{10}. *En effet, l'intersection des ensembles* $\Delta_d(a_1) = \{(7,8),(3,12),(5,12),(9,12),(11,12),(17,18),(15,20),(19,20)\}$ *et* $\Delta_g(a_2) = \{(1,2),(5,6),(5,8),(5,12),(5,14),(9,10),(17,18)\}$ *est égale à* $\{(5,12),(17,18)\}$.

Remarque 7.41 *Soit* (a_1, a_2) *un couple d'arbres binaires complets séparables de* D_n. *Alors, les mots de parenthèses* $u = code(a_1)$ *de* $P_{x,\overline{x}}$ *et* $v = code(a_2)$ *de* $P_{y,\overline{y}}$ *sont tels qu'il existe au*

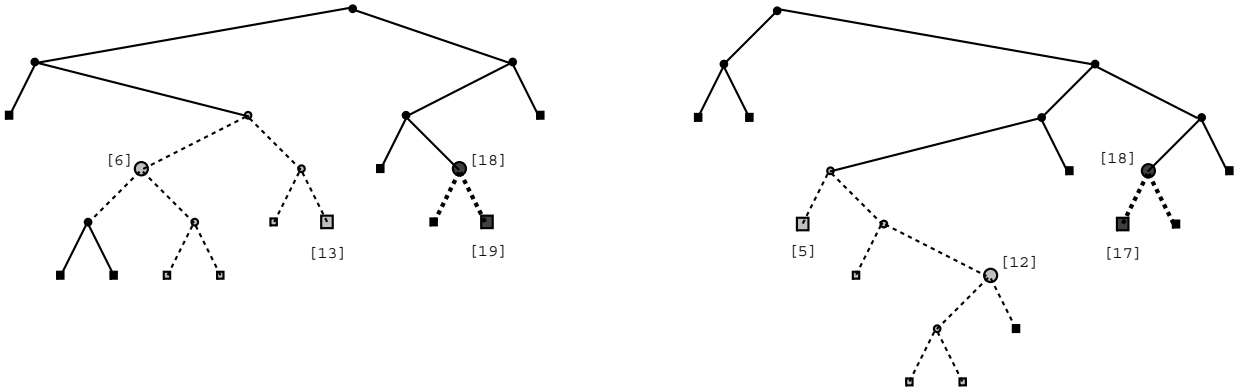


Figure 7.13 Deux arbres binaires complets séparables.

moins une factorisation $u = u_1\overline{x}x^k u_2\overline{x}u_3\overline{x}u_4$, $v = v_1 y v_2 v_3 \overline{y} v_4$ *avec* $u_2 \in P_{x,\overline{x}}$, $x^k\overline{x}u_3 \in P_{x,\overline{x}}$, $v_2 \in P_{y,\overline{y}}\backslash\{\varepsilon\}$, $v_3 \in P_{y,\overline{y}}$, $|u_1\overline{x}x^k u_2|_{\overline{x}} = |v_1 y|_{\overline{y}}$ *et* $|\overline{x}u_4|_{\overline{x}} = |v_3\overline{y}v_4|_{\overline{y}}$.

Lemme 7.42 *La bijection* Υ_2 *(voir lemme 7.18) met en correspondance les permutations de* $\widehat{NSép}_{2n}$ *et les arbres binaires complets non séparables ayant* n *sommets internes.*

Preuve Soient π une permutation de Baxter alternante et (a_1, a_2) un couple d'arbres binaires complets en bijection par Υ_2.
Nous allons prouver que π appartient à $\widehat{Baxter}_{2n}\backslash\widehat{NSép}_{2n}$ (c'est à dire que π possède une sous-suite de type 2413) si et seulement si (a_1, a_2) appartient à D_n. En fait, nous montrons plus précisément que $(\pi(2i+1)-p+1)(\pi(2i+2)-p+1)\ldots(\pi(2j)-p+1) \in \widehat{Baxter}_{q-p}$ avec $q-p = 2j-2i$ si et seulement si $(2i+1, 2j) \in \Delta_d(a_1) \cap \Delta_g(a_2)$.
La figure 7.14 présente deux arbres binaires complets partiellement étiquetés et indicés de façon à mettre en évidence les relations liant π et (a_1, a_2).

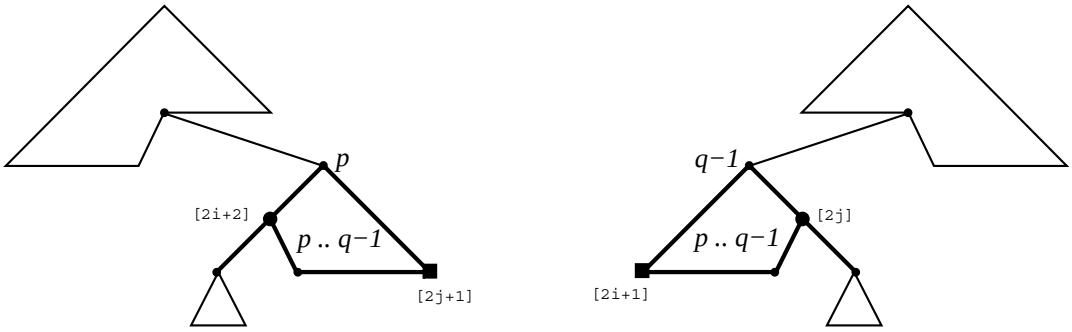


Figure 7.14 Arbres binaires complets séparables partiellement étiquetés représentant les arbres binaires complets croissant et décroissant d'une permutation de Baxter alternante admettant une sous-suite de type 2413.

- Soit f le mot en bijection avec π par Φ et Υ_1. Compte-tenu des lemmes 7.38 et 7.37, f appartient à $C_n\backslash H_n$ et se factorise en $f'x21f''32yf'''$ avec $x, y \in \{1,3\}$, $1f''3 \in Y$, $|f'x|_1 + |f'x|_3 = p-1$ et $|f'''|_1 + |f'''|_3 = 2n-q$; en particulier, le facteur 21 [resp. $2y$] de la factorisation de f code la $p^{ème}$

[resp. $q^{ème}$] lettre de $\Phi(f)$. Alors, il suffit d'appliquer les opérateurs de la figure 7.9 (voir remarque 7.21) permettant de construire les deux arbres binaires complets séparables a_1 et a_2 à partir du mot f.

- Soit p [resp. $q-1$] l'étiquette de la racine du sous-arbre contenant $\pi(2i+1)$ et $\pi(2j)$ dans l'arbre binaire complet croissant [resp. décroissant] de π. Nous avons $\pi(2j+1) < p-1$ [resp. $\pi(2i) > q$] car c'est un ancêtre de p [resp. $q-1$] autre que son père dans l'arbre binaire complet croissant [resp. décroissant]. Soit $\pi(2k+1)$ [resp. $\pi(2l)$] l'étiquette du père de p [resp. $q-1$] dans l'arbre binaire complet croissant [resp. décroissant] vérifiant $\pi(2j+1) < \pi(2k+1) < p$ [resp. $q-1 < \pi(2l) < \pi(2i)$] car c'est un descendant de $\pi(2j+1)$ [resp. $\pi(2i)$] et c'est le père de p dans l'arbre binaire complet croissant [resp. décroissant]. Finalement, comme $p < q-1$, nous avons que la sous-suite $\pi(2k+1)\pi(2i)\pi(2j+1)\pi(2l)$ est de type 2413.

□

Lemme 7.43 *La composition des bijections Λ (voir lemme 7.24) et Ω (voir définition 7.22) permet de mettre en correspondance les mots f de H'_n (ensemble des mots de piles sans facteur $1g3$ où $g \in Y$ codant les arbres 1-2 filiformes non séparables) et les couples (a_1, a_2) d'arbres binaires complets non séparables.*

Preuve

- Compte-tenu de la remarque 7.41 et des définitions des codages préfixe et suffixe d'un arbre binaire complet, la bijection Ω met en correspondance les couples (a_1, a_2) d'arbres binaires complets non séparables et les couples (a, b) de mots de parenthèses de $P_{2,3} \times P_{1,2}$ tels qu'il existe pas de factorisation $a = a'a''3a'''$, $b = b'1b''b'''$ avec $a'' \in P_{2,3}\backslash\{\varepsilon\}$, $b'' \in P_{1,2}\backslash\{\varepsilon\}$, $|a'|_2 = |b'|_2$ et $|a'''|_2 = |b'''|_2$.
- Soient $f = f'1f''3f''' \in C'_n\backslash H'_n$ et (a, b) le couple de mots de parenthèses de $P_{2,3} \times P_{1,2}$ mis en correspondance par Λ. La suppression des lettres 1 dans les mots f', f'', f''' conduit respectivement aux mots a', a'', a''' pour $a = a'a''3a'''$ et la suppression des lettres 3 dans les mots f', f'', f''' conduit respectivement aux mots b', b'', b''' pour $b = b'1b''b'''$, et qui vérifient $(a'', b'') \in P_{1,2}\backslash\{\varepsilon\} \times P_{2,3}\backslash\{\varepsilon\}$, $|a'|_2 = |b'|_2 = |f'|_2$ et $|a'''|_2 = |b'''|_2 = |f'''|_2$.
  Il est clair que réciproquement, pour un couple de mots de parenthèses $(a, b) = (a'a''3a''', b'1b''b''')$ avec $(a'', b'') \in P_{1,2}\backslash\{\varepsilon\} \times P_{2,3}\backslash\{\varepsilon\}$, $|a'|_2 = |b'|_2 = |f'|_2$ et $|a'''|_2 = |b'''|_2 = |f'''|_2$, le facteur f'' tel que $\Lambda(f'') = (a'', b'')$ appartient à $Y\backslash\{\varepsilon\}$ et est encadré par les lettres 1 et 3.

□

7.3.2 Arbres 1-2 filiformes non séparables et cartes planaires cubiques pointées non séparables

Rappelons tout d'abord que le parcours de l'arbre recouvrant d'une carte planaire pointée ayant n arêtes permet de la coder par un mot de longueur $2n$ d'un langage non algébrique L appelé langage de Lehman-Lenormand [70] sur l'alphabet $\{x, \overline{x}, y, \overline{y}\}$.
R. Cori [16] a montré que ce langage L est l'unique solution de l'équation $L = \varepsilon + yL\overline{y}L + xD(L)$ où l'opérateur D est défini de la manière suivante. Si $w = w_1w_2\ldots w_m$ avec $w_i \in P_{x,\overline{x}} \sqcup\!\sqcup \{y, \overline{y}\}^*$ pour tout $i \in [m]$ et m maximal, alors $D(w) = \sum_{i=0}^{m} d_i(w)$ où

$d_i(w) = w_1 w_2 \dots w_i \overline{x} w_{i+1} w_{i+2} \dots w_m$ pour tout $i \in [0, m]$.
Notons de plus qu'une carte planaire pointée non séparable est codée par un mot du langage $L \backslash \{w = w'w''w''' \in L : w'' \in L \text{ et } w'w''' \neq \varepsilon\}$.

Lemme 7.44 *Il existe une bijection Ξ (voir figure 7.15) entre arbres 1-2 filiformes non séparables ayant n points simples de $\overline{F}_n$ et cartes planaires cubiques pointées non séparables ayant $2n$ sommets de CNS_{2n}.*

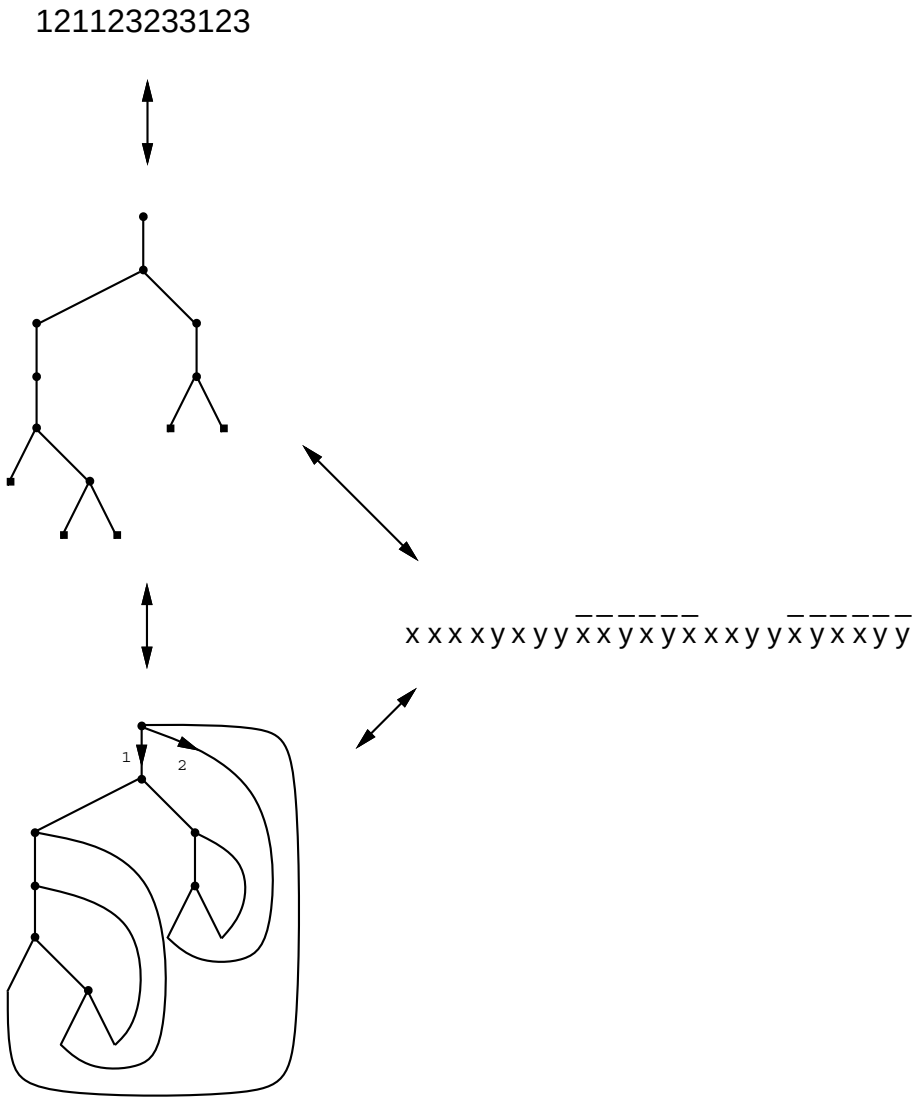


Figure 7.15 La bijection Ξ entre un arbre 1-2 filiforme non séparable et une carte planaire cubique pointée non séparable.

L'application Ξ et son inverse sont définies de la façon suivante.

- Ξ consiste, pour un arbre 1-2 filiforme non séparable, à prolonger à droite chacune des arêtes menant à la dernière, puis à l'avant-dernière, ..., et enfin à la première des feuilles jusqu'au prochain point simple (ou la racine) non saturé relativement à un parcours en profondeur. Le brin pointé de la carte planaire cubique pointée non séparable ainsi obtenue correspond à l'ancien arc partant de la racine de l'arbre.

- L'application inverse Ξ^{-1} consiste à ajouter à l'arbre recouvrant de la carte planaire cubique pointée non séparable une feuille pour toute arête de la carte n'appartenant pas à l'arbre recouvrant et rencontrée pour la première fois lors du parcours en profondeur.

Preuve Clairement, l'application Ξ est bien définie. En effet, l'opération de prolongement des feuilles est valide en raison des conditions **C1** et **C2** que vérifient les arbres 1-2 filiformes non séparables et la carte ainsi construite est planaire et cubique. De plus, le fait qu'elle soit non séparable résulte de la condition **C3'**.
L'application Ξ^{-1} est également bien définie et est clairement, par construction, l'application réciproque de Ξ. □

Notons que, partant d'un arbre 1-2 filiforme non séparable, le mot du langage de Lehman-Lenormand s'obtient en effectuant un parcours en profondeur de l'arbre et en codant une arête externe (dont l'extrémité est une feuille) par la lettre y, une arête interne gauche ou droite par la lettre x à l'aller et la lettre $\overline{x}$ au retour, une arête interne centrale par la lettre x à l'aller et le facteur $\overline{xy}$ au retour, et en ajoutant une lettre $\overline{y}$ à la fin du mot.

Preuve du théorème 7.34. La composition des bijections Φ, Υ, Ω, Λ et Ξ met en correspondance H_n, H'_n et CNS_{2n}. Or, W.T. Tutte [103] a établi la formule d'énumération de ces cartes. L'équidistribution des mots H_n et H'_n suivant le nombre de facteurs de longueur deux se déduit directement de la proposition 7.13. □

7.4 Tableaux de Young standard rectangulaires non séparables de hauteur 3 n'ayant pas deux entiers consécutifs sur une même ligne

Nous établissons maintenant les résultats suivants qui précisent le théorème 7.5.

Théorème 7.45 *Les mots du langage P_n (ensemble des mots de piles sans facteur $2g2, 11, 33$ où $g \in Y$) codant les tableaux de Young standard rectangulaires non séparables de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur une même ligne sont en correspondance avec les mots du langage P'_n (ensemble des mots de piles sans facteur $1g3, 32, 21$ où $g \in Y$) codant les arbres 1-2 filiformes non séparables ayant n points simples et ne possédant aucun point double fils droit ni aucun point simple fils gauche, et sont en bijection avec les cartes planaires pointées*

non séparables ayant $n+1$ arêtes de NS_{n+1}. Ils sont dénombrés par

$$|P_n| = |P'_n| = |NS_{n+1}| = \frac{2.(3n)!}{(2n+1)!(n+1)!}$$

Proposition 7.46 *Les tableaux de Young standard rectangulaires non séparables de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur une même ligne et possédant s couples d'entiers consécutifs situés sur les troisième et première lignes sont en correspondance avec les arbres 1-2 filiformes non séparables ayant n points simples ne possédant aucun point double fils droit ni aucun point simple fils gauche et ayant s points simples fils droits et sont en bijection avec les cartes planaires pointées non séparables ayant $n+1$ arêtes et s sommets. Ils sont dénombrés par*

$$|\{f \in P_n : |f|_{31} = s\}| = |\{f' \in P'_n : |f'|_{31} = s\}| = |\{c \in NS_{n+1} : c \text{ possède } s \text{ sommets}\}| =$$

$$\frac{(2n-s-1)!(n+s)!}{(2n-2s-1)!(n-s)!(2s+1)!(s+1)!}$$

Lemme 7.47 *La correspondance composant les bijections Φ (voir lemme 7.16), Υ (voir lemme 7.18) et Ψ (voir théorème 6.16) relie les mots de P_n (ensemble des mots de piles sans facteur $2g2, 11, 33$ où $g \in Y$) et les permutations non séparables de $S_n(2413, 41\overline{3}52)$.*
De plus, cette correspondance met en bijection les mots f de P_n tels que $|f|_{31} = s$ et les permutations π de $S_n(2413, 41\overline{3}52)$ telles que $desc(\pi) = s$.

Preuve Rappelons que les bijections Φ et Υ_1 mettent en correspondance, d'une part les mots de C_n et les permutations de Baxter alternantes de $\widehat{S}_{2n}(25\overline{3}14, 41\overline{3}52)$, et d'autre part les mots de H_n et les permutations non séparables alternantes de $\widehat{S}_{2n}(2413, 41\overline{3}52)$. De plus, les bijections Φ, Υ et Ψ mettent en correspondance les mots de B_n et les permutations de Baxter de $S_n(25\overline{3}14, 41\overline{3}52)$. Or, l'interdiction d'un facteur $2g2$ pour tout $g \in Y$ dans les mots de C_n pour n'autoriser que les mots de H_n équivaut à l'exclusion du motif 25314 dans les permutations de Baxter alternantes pour n'autoriser que les permutations non séparables alternantes. Nous appliquons ici cette même interdiction aux mots de B_n pour n'autoriser que les mots de P_n, ce qui revient à exclure le motif 25314 dans les permutations de Baxter pour n'autoriser que les permutations non séparables de $S_n(2413, 41\overline{3}52)$.
De plus, d'après la propriété 7.20, le nombre de facteurs 31 d'un mot de P_n correspond au nombre de descentes d'une permutation de $S_n(2413, 41\overline{3}52)$. □

Preuve du théorème 7.45. Les bijections Φ, Υ et Ψ (voir théorème 6.16) mettent en correspondance les mots de P_n et les permutations non séparables sur $[n]$. S. Dulucq, S. Gire et J. West [28, 45] relient ces permutations aux cartes planaires pointées non séparables ayant $n+1$ arêtes, cartes dénombrées par W.T. Tutte [105]. Le théorème 7.34 nous permet de conclure pour les mots de P'_n. □

Preuve de la proposition 7.46. Ce résultat se déduit du théorème 7.45 et de la formule dénombrant les cartes planaires pointées non séparables ayant $n+1$ arêtes et s sommets, formule due à W.G. Brown et W.T. Tutte [12]. □

Remarquons qu'une des façons de prouver la conjecture 7.35 consisterait à montrer que $(n-m).|H_{n,m-1}| = m.|H_{n,m}|$ ou bien que $(n-m).|H'_{n,m-1}| = m.|H'_{n,m}|$, et ce pour tout $n \geq 1$ et pour tout $m \in [\lfloor \frac{n-1}{2} \rfloor]$. En effet, d'après le théorème 7.45, $H_{n,0} = P_n$ et $H'_{n,0} = P'_n$ satisfont la conjecture.

7.5 D'autres restrictions sur les mots de piles

En considérant plusieurs autres restrictions sur les mots de piles, nous mettons en évidence différents langages en bijection avec l'ensemble des couples de chemins de Dyck ne se coupant pas, avec les arbres binaires, ou encore avec les arbres ternaires complets.

7.5.1 Mots de piles et couples de chemins de Dyck ne se coupant pas

Nous considérons maintenant certains chemins à rapprocher de ceux étudiés par M. Desainte Catherine et X. Viennot [22] et par S. Hee Choi et D. Gouyou-Beauchamps [56].

Le langage des facteurs gauches de mots de parenthèses sur $\{z, \overline{z}\}$, de longueur l et de hauteur finale p, est le langage $FGD_{l,p} = \{w \in \{z,\overline{z}\}^* : |w| = l; |w|_z - |w|_{\overline{z}} = p; \forall w = w'w'', |w'|_z \geq |w'|_{\overline{z}}\}$.

Nous notons $V_{l,p}$ (voir figure 7.16) l'ensemble des couples de facteurs gauches de mots de parenthèses, de même longueur l et de même hauteur finale p, dont les chemins correspondants ne se coupent pas : $V_{l,p} = \{(u,v) \in FGD_{l,p} \times FGD_{l,p}$ sur $\{x,\overline{x}\}^* \times \{y,\overline{y}\}^* : \forall u = u'u'', v = v'v'', |u'| = |v'| \Longrightarrow |u'|_x - |u'|_{\overline{x}} \geq |v'|_y - |v'|_{\overline{y}}\}$.

Ainsi, $V_{2n,0}$ (voir figure 7.17) désigne l'ensemble des couples de mots de parenthèses de même longueur $2n$ codant des chemins de Dyck ne se coupant pas.

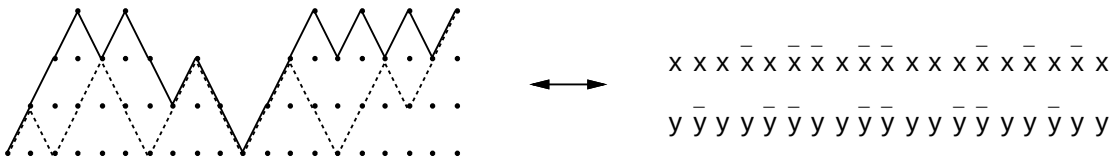


Figure 7.16 Un couple de facteurs gauches de mots de parenthèses dont les chemins correspondants ne se coupent pas de $V_{19,3}$.

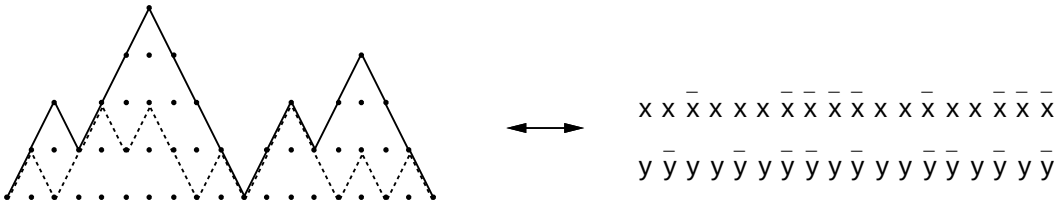


Figure 7.17 Un couple de mots de parenthèses codant des chemins de Dyck ne se coupant pas de $V_{18,0}$.

D. Gouyou-Beauchamps [48, 49] a établi combinatoirement les deux résultats suivants.

$$|V_{l,p}| = |\{\sigma \in I_l(54321) : \sigma \text{ a } p \text{ points fixes}\}| = \frac{(p+3)!l!(l+2)!}{p!\frac{l-p}{2}!(\frac{l-p}{2}+1)!(\frac{l+p}{2}+2)!(\frac{l+p}{2}+3)!}$$

$$\sum_{k=0}^{n-1} |V_{2n-1,2k+1}| = |I_{2n-1}(54321)| = c_n.c_n \quad \text{et} \quad \sum_{k=0}^{n} |V_{2n,2k}| = |I_{2n}(54321)| = c_{n+1}.c_n$$

Théorème 7.48 *Les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs sur la première ligne et les tableaux de Young standard rectangulaires de hauteur 3 et de longueur n n'ayant pas deux entiers consécutifs situés sur les deuxième et première lignes sont en bijection avec les couples de chemins de Dyck ne se coupant pas de longueur $2n$. Ils sont dénombrés par*

$$|\{f \in Y_n : |f|_{11} = 0\}| = |\{f \in Y_n : |f|_{21} = 0\}| = |V_{2n,0}| = \frac{3!(2n)!(2n+2)!}{n!(n+1)!(n+2)!(n+3)!}$$

En considérant les opérations miroir et complémentaire d'un mot, nous en déduisons que cette même formule dénombre l'ensemble des mots de piles sans facteur 32 et l'ensemble des mots de piles sans facteur 33.

Corollaire 7.49 *Les tableaux de Young standard n'ayant pas deux entiers consécutifs sur la première ligne, possédant $\frac{l+p}{2}$ entiers sur les deux premières lignes et $\frac{l-p}{2}$ entiers sur la troisième ligne, et les tableaux de Young standard n'ayant pas deux entiers consécutifs situés sur les deuxième et première lignes, possédant $\frac{l+p}{2}$ entiers sur les deux premières lignes et $\frac{l-p}{2}$ entiers sur la troisième ligne, sont en bijection avec les couples de facteurs gauches de mots de parenthèses de même longueur l et de même hauteur finale p dont les chemins correspondants ne se coupent pas. Ils sont dénombrés par*

$$\frac{(p+3)!l!(l+2)!}{p!\frac{l-p}{2}!(\frac{l-p}{2}+1)!(\frac{l+p}{2}+2)!(\frac{l+p}{2}+3)!}$$

Lemme 7.50 *Il existe une bijection Φ_{11} entre couples de chemins de Dyck ne se coupant pas et mots de piles sans facteur 11. Celle-ci est donnée par le morphisme*

$$\Phi_{11} : \begin{array}{ccc} V_{2n,0} & \longrightarrow & Y_n \backslash \{\mathcal{A}^* 11 \mathcal{A}^*\} \\ (u,v) & \longmapsto & f \end{array} \quad \textit{défini par} \begin{cases} \Phi_{11}(x,y) = 12 \\ \Phi_{11}(x,\overline{y}) = 13 \\ \Phi_{11}(\overline{x},y) = 2 \\ \Phi_{11}(\overline{x},\overline{y}) = 3 \end{cases}$$

Preuve Soient $(u,v) \in V_{2n,0}$ et $f = \Phi_{11}(u,v)$. Nous avons alors

- $|u|_x = |u|_{\overline{x}} = |v|_y = |v|_{\overline{y}} = n \Longrightarrow |f|_1 = |f|_2 = |f|_3 = n$,
- Soient $u = u'u'', v = v'v''$ tels que $|u'| = |v'|$ et $f' = \Phi_{11}(u',v')$; alors
  - $|u'|_x - |u'|_{\overline{x}} \geq |v'|_y - |v'|_{\overline{y}} \Longrightarrow |f'|_1 \geq |f'|_2$,
  - $|v'|_y \geq |v'|_{\overline{y}} \Longrightarrow |f'|_2 \geq |f'|_3$.

De plus, l'ensemble $\{12, 13, 2, 3\}$ constituant un code préfixe, l'application réciproque de Φ_{11} est clairement définie. □

Exemple 7.51 *Le mot* 12132121312332312123131232 3 *de* $Y_9\backslash\{\mathcal{A}^*11\mathcal{A}^*\}$ *est en correspondance par* Φ_{11} *avec le couple de mots de parenthèses ne se coupant pas de* $V_{18,0}$ *illustré par la figure 7.17.*

Lemme 7.52 *Il existe une bijection* Λ_{21} *entre mots de piles sans facteur* 21 *et couples de chemins de Dyck ne se coupant pas. Celle-ci est donnée par le morphisme*

$$\Lambda_{21} \quad : \quad \begin{array}{ccc} Y_n\backslash\{\mathcal{A}^*21\mathcal{A}^*\} & \longrightarrow & V_{2n,0} \\ f & \longmapsto & (u,v) \end{array} \quad \textit{défini par} \begin{cases} \Lambda_{21}(1) = (x, \varepsilon) \\ \Lambda_{21}(2) = (\varepsilon, y) \\ \Lambda_{21}(3) = (\overline{x}, \overline{y}) \end{cases}$$

L'application réciproque consiste à envoyer, avant chaque couple $(\overline{x}, \overline{y})$, d'abord toutes les lettres x sur 1 avant toutes les lettres y sur 2.

Preuve Soient $f \in Y_n\backslash\{\mathcal{A}^*21\mathcal{A}^*\}$ et $(u, v) = \Lambda_{21}(f)$. Nous avons alors

- $|f|_1 = |f|_2 = |f|_3 = n \Longrightarrow |u|_x = |u|_{\overline{x}} = |v|_y = |v|_{\overline{y}} = n$,
- pour toute factorisation $f = f'f''$ telle que $\Lambda_{21}(f') = (u', v')$, $|f'|_1 \geq |f'|_2 \Longrightarrow |u'|_x \geq |v'|_y$ et donc $|u'|_x - |u'|_{\overline{x}} \geq |v'|_y - |v'|_{\overline{y}}$ car $|u'|_{\overline{x}} = |v'|_{\overline{y}}$.

□

Exemple 7.53 *Le mot* 11231111223233231122311321323 *de* $Y_9\backslash\{\mathcal{A}^*21\mathcal{A}^*\}$ *est en correspondance par* Λ_{21} *avec le couple de mots de parenthèses ne se coupant pas de* $V_{18,0}$ *illustré par la figure 7.17.*

Preuve du théorème 7.48. Les bijections Φ_{11} et Λ_{21} mettent en correspondance les couples de chemins de Dyck ne se coupant pas avec les mots de piles respectivement sans facteur 11 et sans facteur 21. Nous obtenons alors le résultat d'énumération annoncé à partir de la formule de D. Gouyou-Beauchamps [48, 49] dénombrant $V_{l,p}$, en posant $l = 2n$ et $p = 0$. □

Preuve du corollaire 7.49. Il suffit d'étendre les morphismes Φ_{11} et Λ_{21} aux couples de facteurs gauches de mots de parenthèses ne se coupant pas de $V_{l,p}$. □

Exemple 7.54 *Les mots* 12131221332123312121331221321 2 *et* 11123122331223311122313122312 2 *sont en correspondance respectivement par* Φ_{11} *et* Λ_{21} *avec le couple de facteurs gauches de mots de parenthèses ne se coupant pas de* $V_{19,3}$ *illustré par la figure 7.16.*

7.5.2 Mots de piles et arbres binaires

Théorème 7.55 *Les tableaux de Young standard rectangulaires de hauteur* 3 *et de longueur* n *n'ayant pas deux entiers consécutifs sur la première ligne, n'ayant pas deux entiers consécutifs situés sur les première et troisième lignes et les tableaux de Young standard rectangulaires de*

hauteur 3 *et de longueur* n *n'ayant pas deux entiers consécutifs situés sur les deuxième et première lignes, n'ayant pas deux entiers consécutifs situés sur les troisième et première lignes sont en bijection avec les mots de parenthèses de longueur* $2n$*. Ils sont dénombrés par*

$$|\{f \in Y_n : |f|_{11} = |f|_{13} = 0\}| = |\{f \in Y_n : |f|_{21} = |f|_{31} = 0\}| = c_n$$

Plusieurs autres ensembles des mots de piles excluant deux facteurs de longueur deux sont également dénombrés par le $n^{ème}$ nombre de Catalan. Ils s'obtiennent bijectivement à partir de l'un des deux langages donnés par le théorème 7.55, en utilisant la bijection Λ (entre mots de piles sans facteur 13 ou 31 et arbres binaires), avec la correspondance composant les bijections Φ, Υ, Ω et Λ (entre mots de piles sans facteur 22 et mots de piles sans facteur 13) ou encore en appliquant les opérations miroir et complémentaire sur un mot.

Lemme 7.56 *Les mots de piles sans facteur* $11, 13$ *sont exactement les mots de parenthèses de* $P_{12,3}$*.*

Preuve Il suffit de remarquer que tout mot de piles f appartenant à Y_n tel que $|f|_{11} = |f|_{13} = 0$ vérifie $|f|_{12} = n$. □

Lemme 7.57 *Les mots de piles sans facteur* $21, 31$ *sont exactement les mots de* $\{1\}^* P_{2,3}$ *ayant autant de* 1 *que de* 2*.*

Preuve Il suffit de remarquer que tout mot de piles f appartenant à Y_n tel que $|f|_{21} = |f|_{31} = 0$ vérifie $|f|_{11} = n - 1$. □

Preuve du théorème 7.55. Ce résultat est une conséquence des deux lemmes précédents. □

7.5.3 Mots de piles et arbres ternaires complets

Définition 7.58 *Un* arbre ternaire complet *est un arbre dessiné et enraciné pour lequel chaque sommet interne possède exactement trois fils.*

De manière générale, le nombre d'arbres p-aires ayant $pn + 1$ sommets (n sommets internes et $(p-1)n + 1$ feuilles) est $\frac{(pn)!}{((p-1)n+1)!n!}$ [61].

Le système de réécriture $\begin{cases} (p) \\ (t) \quad \rightsquigarrow \quad (p), (p+1), \ldots, (p+t-1) \end{cases}$ caractérise un arbre de génération des arbres p-aires où l'étiquette (t) associée à un arbre indique que ses t feuilles les plus à gauche sont actives (c'est à dire qu'il est possible de les faire croître).

Théorème 7.59 *Les mots du langage* R_n *(ensemble des mots de piles sans facteur* $2g2, 1g3$ *où* $g \in Y$*) sont en bijection avec les arbres ternaires complets ayant* $3n + 1$ *sommets. Ils sont dénombrés par*

$$|R_n| = \frac{(3n)!}{(2n+1)!n!}$$

Preuve Cette bijection est obtenue par le codage suivant d'un arbre ternaire complet a.

$$tern(a) = \begin{cases} \varepsilon & \text{si } a \text{ est réduit à un sommet} \\ 1\ tern(gauche(a))\ 2\ tern(central(a))\ 3\ tern(droit(a)) & \text{sinon} \end{cases}$$

Clairement, ce codage constitue une bijection entre arbres ternaires complets ayant n sommets internes et mots du langage R_n. □

1121233211212331232331123123231123

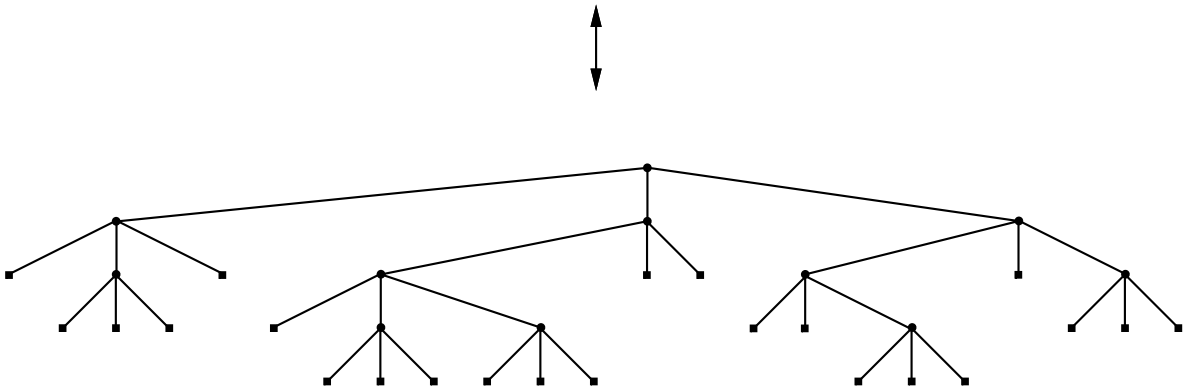


Figure 7.18 Codage d'un arbre ternaire complet ayant 11 sommets internes par un mot de R_{11}.

Exemple 7.60 *La figure 7.18 illustre cette bijection.*

Remarquons pour terminer que nous avons $2.|R_n| = (n+1).|P_n|$ où P_n est l'ensemble des mots de piles sans facteur $2g2, 11, 33$ avec $g \in Y$; il est également possible de considérer, au lieu de P_n, l'ensemble des mots de piles sans facteur $1g3, 32, 21$ avec $g \in Y$, ou l'ensemble des mots de piles sans facteur $1g3, 22, 31$ avec $g \in Y$ ou encore l'ensemble des mots de piles sans facteur $2g2, 13, 31$ avec $g \in Y$. Prouver combinatoirement cette formule fournirait, compte-tenu des résultats obtenus dans cette thèse, une preuve combinatoire de la formule dénombrant les cartes planaires pointées non séparables ayant $n+1$ arêtes, problème soulevé par R. Cori [17].

Perspectives

La méthode des arbres de génération se révèle être une technique pouvant avoir plusieurs applications et dont il est naturel d'aborder certaines questions qu'elle pose.

Comme nous l'avons montré, cette méthode peut être utilisée pour effectuer la génération aléatoire d'objets combinatoires.

Dans de nombreux cas, nous avons obtenu des systèmes de réécriture différents permettant d'engendrer des objets combinatoires ayant la même formule d'énumération. Par exemple, nous avons exhibé cinq systèmes de réécriture correspondant à des objets énumérés par les coefficients binomiaux centraux. Ainsi, il est naturel de se demander s'il existe des opérations qui transforment les règles (et les étiquettes) d'un système de réécriture pour en obtenir un autre, avec pour unique contrainte que les deux arbres de dérivation correspondants aient le même nombre de sommets par niveau? Ce principe est à rapprocher de la notion de réécriture de termes en programmation fonctionnelle. Un exemple simple utilisant cette technique nous a permis de relier combinatoirement involutions vexillaires (motif 2143 interdit) et involutions excluant le motif 1243.

M.P. Schützenberger [91, 93] a montré les liens qui pouvaient exister entre certains problèmes d'énumération et certaines classifications de langages. Qu'en est-il pour la méthode des arbres de génération? Nous pouvons actuellement affirmer que des objets combinatoires dont les séries génératrices sont rationnelles, algébriques ou différentiablement finies peuvent être engendrés par des arbres de génération se caractérisant ensuite par un système de réécriture.
Par exemple, les nombres de Fibonacci (série génératrice rationnelle) et les nombres de Catalan (série génératrice algébrique) sont obtenus avec les systèmes de réécriture caractérisant respectivement l'arbre de génération $T(123, 132, 213)$ et l'arbre de génération des mots de parenthèses. D'autre part, la série génératrice associée au système de réécriture obtenu en effectuant le produit cartésien de deux systèmes de réécriture caractérisant l'arbre de génération des mots de parenthèses est différentiablement finie. En effet, D. Gouyou-Beauchamps a montré lors de l'énumération des tableaux de Young standard de hauteur au plus 4 [48, 49] que la série génératrice correspondante est différentiablement finie puisque la suite des carrés des nombres de Catalan satisfait une P-récurrence, c'est à dire une récurrence linéaire homogène à coefficients polynomiaux [42].

Introduite par F.R.K. Chung, R.L. Graham, V.E. Hoggatt et M. Kleiman [15] pour dénombrer les permutations de Baxter [4], la méthode des arbres de génération a été utilisée dans de nombreux travaux portant sur l'énumération d'ensembles de permutations à motifs exclus [110, 53, 45, 114, 97, 112] et a permis à S. Dulucq, S. Gire, O. Guibert et J. West [28, 27] d'établir une correspondance entre permutations triables par deux passages consécutifs dans une pile et cartes planaires pointées non séparables, prouvant ainsi une conjecture de J. West [110, 113].

Ainsi, les correspondances obtenues par isomorphisme d'arbres de génération des permutations viennent en complément des trois bijections classiques miroir, complémentaire et inverse sur les permutations à motifs exclus [110]. Les résultats développés dans les chapitres 4 et 5, en partie devinés à l'aide du logiciel *forbid*, en sont une illustration.

Toutefois, certaines familles de permutations à motifs exclus, bien qu'ayant une même formule d'énumération (par exemple, 12 ensembles sont dénombrés par les coefficients binomiaux centraux), ne peuvent être reliées entre elles par la méthode des arbres de génération ou par ces trois bijections classiques.

Ainsi, apparait la nécessité d'obtenir des résultats revétant un caractère général sur l'énumération des permutations à motifs exclus, les seuls connus à ce jour étant ceux d'E. Babson et J. West [110, 2, 115].

Un vaste travail reste à entreprendre pour énumérer involutions et permutations alternantes à motifs exclus, à l'instar des résultats de R. Simion et F.W. Schmidt [95] et de J. West [110, 114] pour les permutations à motifs exclus.

Le dénombrement de tels ensembles pour des motifs simples constituerait une première étape de ce travail, d'autant que des formules classiques en Combinatoire apparaissent comme par exemple les nombres de Motzkin.

Une telle approche est également motivée par les travaux [79, 99, 49, 118, 6, 42] sur les tableaux de Young standard de hauteur bornée, tableaux en bijection [83, 89, 92] avec les involutions excluant le motif identité.

Les involutions excluant le motif 54321 et les permutations de Baxter alternantes ont même formule d'énumération (alternativement le carré des nombres de Catalan et le produit de deux nombres de Catalan successifs), formule prouvée combinatoirement respectivement par D. Gouyou-Beauchamps [49] et par R. Cori, S. Dulucq et X. Viennot [18], mais sans que n'ait été encore trouvé de bijection directe reliant ces deux ensembles.

Nous pensons avoir franchi une première étape vers un tel résultat. En effet, nous avons caractérisé un nouvel ensemble d'involutions à motifs exclus, directement en correspondance avec les involutions excluant le motif 54321, et possédant des distributions qui coïncident (les premières valeurs ont été vérifiées à l'aide du logiciel *forbid*) avec certaines distributions des permutations de Baxter alternantes. Ces distributions font apparaître les nombres de Delannoy et de Narayana.

Les permutations vexillaires, introduites par A. Lascoux et M.P. Schützenberger [69], sont telles que les partitions correspondant aux tables d'inversion de ces permutations et de leurs inverses sont conjuguées. Elles correspondent également aux permutations excluant le motif 2143 [72]. En combinant les travaux de J. West [110] et d'I.M. Gessel [42], nous obtenons une formule les dénombrant.

Nous nous sommes intéressés aux involutions vexillaires et avons conjecturé qu'elles sont énumérées par les nombres de Motzkin.

Pour l'instant, nous avons seulement montré que les involutions vexillaires sans point fixe sont en bijection avec les permutations vexillaires. Plus précisément, nous avons établi qu'une permutation π appartient à $S_n(2143)$ si et seulement si l'involution sans point fixe $(n+\pi^{-1}(1))(n+\pi^{-1}(2))\dots(n+\pi^{-1}(n))\pi(1)\pi(2)\dots\pi(n)$ appartient à $I_{2n}(2143)$.

En dépit de la correspondance de S. Dulucq, S. Gire, O. Guibert et J. West [28, 27] reliant cartes planaires pointées non séparables, permutations non séparables et permutations triables par deux passages consécutifs dans une pile, il n'existe pas de preuve combinatoire de la formule dénombrant ces objets, problème soulevé par R. Cori [17].

Une étude approfondie de la caractérisation des cartes planaires pointées non séparables en terme d'arbres bien étiquetés due à S. Dulucq et J-G. Penaud [31] ou du codage des permutations triables par deux passages consécutifs dans une pile par des chemins de Raney dû à I.P. Goulden et J. West [46] pourrait éventuellement permettre de résoudre ce problème.

Nous pouvons également considérer une autre approche qui fait directement suite à nos travaux. En effet, nous avons obtenu quatre langages de mots de piles en bijection avec les permutations non séparables. Or, un autre langage de mots de piles est en correspondance avec les arbres ternaires complets. Exprimer l'un des quatre langages en bijection avec les permutations non séparables en fonction de celui codant les arbres ternaires complets constituerait donc une solution à ce problème.

Plusieurs nouvelles questions portant sur l'énumération des mots de langages correspondant à des restrictions sur les mots de piles restent sans réponse. Toutefois, nous avons remarqué que dans plusieurs cas apparaissent des formules énumérant des familles de cartes planaires considérées par W.T. Tutte.

Par exemple, nous conjecturons qu'un de ces langages est dénombré par la formule donnant le nombre de triangulations planaires [102] ou de cartes planaires cubiques pointées non séparables 3-connexes [103]. De même, il serait fort utile d'établir une correspondance permettant d'expliquer le rapport du nombre de cartes planaires cubiques pointées non séparables [103] au nombre de cartes planaires pointées non séparables [105].

Plus généralement, nous avons souvent constaté que des permutations à motifs exclus et des cartes planaires ont même formule d'énumération. Ceci est source de nouvelles recherches laissant apparaître de nombreux problèmes.

Annexe A

Catalogue sur les permutations à motifs exclus

Ce catalogue, inspiré de celui de J. West [111], complété par les résultats obtenus dans cette thèse et quelques recherches bibliographiques, présente les résultats que nous connaissons à ce jour sur l'énumération des permutations à motifs exclus. Il ne prétend pas toutefois être exhaustif sur le sujet.

Propriétés et résultats généraux

Nous présentons tout d'abord des résultats très généraux, c'est à dire des propriétés pouvant s'appliquer à toute une classe de permutations à motifs exclus.

- $|S_n(12\ldots(l+1),(m+1)m\ldots1)| = 0, \forall n > l.m$ [34]

- $|S_n(12\ldots(k+1))| \sim \alpha_k.\frac{(k-1)^{2n}}{n^{(k^2-2k)/2}}$ où α_k est une constante [79]
  $|I_n(12\ldots k)| = |I_n(k(k-1)\ldots1)|$

- $T(12a_3a_4\ldots a_k) \cong T(21a_3a_4\ldots a_k)$ [110]
  $T(123a_4a_5\ldots a_k) \cong T(321a_4a_5\ldots a_k)$ [2]
  $T(12\ldots ra_{r+1}a_{r+2}\ldots a_k) \cong T(r(r-1)\ldots1a_{r+1}a_{r+2}\ldots a_k)$ [115]

- $\pi \in S_n(\tau_1,\tau_2,\ldots,\tau_p) \Longleftrightarrow \pi^* \in S_n({\tau_1}^*,{\tau_2}^*,\ldots,{\tau_p}^*) \Longleftrightarrow \pi^c \in S_n({\tau_1}^c,{\tau_2}^c,\ldots,{\tau_p}^c) \Longleftrightarrow \pi^{-1} \in S_n({\tau_1}^{-1},{\tau_2}^{-1},\ldots,{\tau_p}^{-1})$ [110]
  $\pi \in I_n(\tau_1,\tau_2,\ldots,\tau_p) \Longleftrightarrow \pi^{*c} \in I_n({\tau_1}^{*c},{\tau_2}^{*c},\ldots,{\tau_p}^{*c})$ [corollaire 1.13]
  $\pi \in I_n(\tau_1,\tau_2,\ldots,\tau_p) \Longleftrightarrow \pi \in I_n({\tau_1}^{-1},{\tau_2}^{-1},\ldots,{\tau_p}^{-1})$ [corollaire 1.13]
  $\pi \in \widehat{S}_{2k}(\tau_1,\tau_2,\ldots,\tau_p) \Longleftrightarrow \pi^{*c} \in \widehat{S}_{2k}({\tau_1}^{*c},{\tau_2}^{*c},\ldots,{\tau_p}^{*c})$ [corollaire 1.13]
  $\pi \in \widehat{S}_{2k+1}(\tau_1,\tau_2,\ldots,\tau_p) \Longleftrightarrow \pi^* \in \widehat{S}_{2k+1}({\tau_1}^*,{\tau_2}^*,\ldots,{\tau_p}^*)$ [corollaire 1.13]

Exclusion d'au moins une permutation d'ordre 3

Nous présentons les résultats d'énumération connus sur les permutations excluant des motifs dont le plus petit d'entre eux est une permutation d'ordre 3.

Les motifs exclus sont tous des permutations d'ordre 3

- $|S_n(\tau)| = \frac{(2n)!}{(n+1)!n!} = c_n$ le $n^{ème}$ nombre de Catalan, $\forall \tau \in S_3$ [62]
- $|S_n(123,132)| = |S_n(132,231)| = 2^{n-1}$ [95]
  $|S_n(132,213)| = 2^{n-1}$ [88]
- $|S_n(123,231)| = 1 + \binom{n}{2}$ [95]
- $|S_n(123,132,213)| = f_n$ le $n^{ème}$ nombre de Fibonacci défini par $f_n = f_{n-1} + f_{n-2}$ ($f_0 = f_1 = 1$) [95]
- $|S_n(123,132,231)| = |S_n(123,231,312)| = |S_n(132,213,231)| = n$ [95]

Deux motifs exclus : deux permutations d'ordre 3 et 4

- $|S_n(123,1432)| = |S_n(123,2143)| = |S_n(123,2413)| = |S_n(132,1234)| = |S_n(132,2134)| = |S_n(132,2314)| = |S_n(132,2341)| = |S_n(132,3241)| = |S_n(132,3412)| = f_{2n-2}$ le $(n-2)^{ème}$ nombre de Fibonacci défini par $f_n = f_{n-1} + f_{n-2}$ ($f_0 = f_1 = 1$) [114]
- $|S_n(123,2431)| = 3.2^{n-1} - \binom{n+1}{2} - 1$ [114]
- $|S_n(123,3412)| = 2^{n+1} - \binom{n+1}{3} - 2n - 1$ [9]
- $|S_n(123,3421)| = \binom{n}{4} + 2\binom{n}{3} + n$ [114]
- $|S_n(123,4231)| = \binom{n}{5} + 2\binom{n}{4} + \binom{n}{3} + \binom{n}{2} + 1$ [114]
- $|S_n(132,3214)|$ a pour fonction génératrice $\frac{(1-x)^3}{1-4x+5x^2-3x^3}$ [114]
- $|S_n(132,3421)| = 1 + (n-1)2^{n-2}$ [114]
  $|S_n(132,4231)| = 1 + (n-1)2^{n-2}$ [53]
- $|S_n(132,4321)| = \binom{n}{4} + \binom{n+1}{4} + \binom{n}{2} + 1$ [114]

Trois motifs exclus : une permutation d'ordre 3 et deux d'ordre 4

- $|S_n(123,2143,3214)| = |S_n(213,1234,1243)| = |S_n(132,2341,3241)| = \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n}{2k+1} 2^k = p_n$ le $n^{ème}$ nombre de Pell vérifiant $p_n = 2p_{n-1} + p_{n-2}$ ($p_1 = 1, p_2 = 2$) [section 4.1]
- $|S_n(123,1432,3214)|$ a pour fonction génératrice $\frac{1-x}{1-2x-x^3-x^4+x^5}$ [53]

Exclusion d'au moins une permutation d'ordre 4

Nous présentons les résultats d'énumération connus sur les permutations excluant des motifs dont le plus petit d'entre eux est une permutation d'ordre 4.

Un seul motif exclu d'ordre 4

- $|S_n(1234)| = 2\sum_{k=0}^{n} \binom{2k}{k}\binom{n}{k}^2 \frac{3k^2+2k+1-n-2kn}{(k+1)^2(k+2)(n-k+1)}$ [42]
  $T(1234) \cong T(1243) \cong T(2143)$ [110]
- $T(3142) \cong T(4132)$ [97]
- $|S_n(1423)| < |S_n(1234)| < |S_n(1324)|, \forall n > 7$ [10]

Deux motifs exclus d'ordre 4

- $|S_n(2413, 3142)| = \sum_{i=0}^{n-1} \binom{n-1+i}{n-1-i} c_i$ le $(n-1)^{ème}$ nombre de Schröder [110]
  $|S_n(3124, 3214)| = \sum_{i=0}^{n-1} \binom{n-1+i}{n-1-i} c_i$ [45]
  $|S_n(1234, 2134)| = |S_n(1324, 2134)| = |S_n(1324, 2314)| = |S_n(1342, 2341)| = |S_n(2134, 3124)| = |S_n(2314, 3124)| = |S_n(3142, 3241)| = |S_n(3412, 3421)| = \sum_{i=0}^{n-1} \binom{n-1+i}{n-1-i} c_i$ [section 4.4]
- $|S_n(3412, 4231)|$ a pour fonction génératrice $\frac{1-5x+4x^2-2x^3C(x)}{1-6x+8x^2-4x^3} = \frac{x}{1-\frac{2x}{2-C(x)}}$ [54]
  $|S_n(3124, 4213)|$ a pour fonction génératrice $\frac{1-5x+4x^2-2x^3C(x)}{1-6x+8x^2-4x^3} = \frac{x}{1-\frac{2x}{2-C(x)}}$ [97]
  où $C(x) = \frac{1-\sqrt{1-4x}}{2x}$ est la fonction génératrice des nombres de Catalan

Quatre motifs exclus d'ordre 4

- $|S_n(1234, 1243, 1423, 4123)| = |S_n(1324, 1342, 1432, 4132)| = |S_n(2134, 2143, 2413, 4213)| = |S_n(2314, 2413, 3142, 3241)| = |S_n(1234, 1324, 2134, 2314)| = |S_n(1234, 2134, 2314, 3124)| = |S_n(1324, 2134, 2314, 3124)| = |S_n(1324, 2134, 3124, 3214)| = |S_n(1324, 2314, 3124, 3214)| = |S_n(1342, 2341, 3142, 3241)| = |S_n(1324, 1342, 2314, 2341)| = |S_n(1342, 2341, 2431, 3241)| = \binom{2n-2}{n-1}$ [section 4.2]

Classes des symétries $(*, c, -1)$ complètes d'ordre 4

- $|S_n(1243, 2134, 3421, 4312)| = 14n, \forall n \geq 6$ [97]
- $|S_n(1324, 4231)| = 2 + 2^{n-5}(\frac{n^3-18n^2+59n-138}{3}) + 2^{n-\frac{5}{2}}((1+\frac{1}{\sqrt{2}})^{n+1} - (1-\frac{1}{\sqrt{2}})^{n+1})$ [53]
- $|S_n(1342, 1423, 2314, 2431, 3124, 3241, 4132, 4213)| = 2^n - 2, \forall n \geq 5$ [97]
- $|S_n(1432, 2341, 3214, 4123)| = 2.|S_n(123, 1432, 3214)|\ \forall n \geq 6$ [53]

- $S_n(2143, 3412)$: ensemble des permutations obtenues par mélange d'une sous-suite croissante et d'une sous-suite décroissante [97]

Autres résultats

Nous présentons finalement des résultats sur l'énumération de permutations excluant des motifs dont au moins l'un d'entre eux est une permutation barrée, et ceux portant sur l'énumération des involutions excluant le motif identité.

Permutations triables par deux passages consécutifs dans une pile

- $|S_n(2341, 3\overline{5}241)| = \frac{2.(3n)!}{(2n+1)!(n+1)!}$ [119]
- $T(2413, 41\overline{3}52)$ est isomorphe à l'arbre de génération [45, 28] des cartes planaires pointées non séparables ayant $n+1$ arêtes énumérées par $\frac{2.(3n)!}{(2n+1)!(n+1)!}$ [105]
- $|S_n(3241, \overline{2}4153)| = |S_n(2413, \overline{4}2315)| = |S_n(3142, 45\overline{3}12)| = \frac{2.(3n)!}{(2n+1)!(n+1)!}$ [45, 27, section 5.1]
- $|S_n(3412, \overline{2}4531)| = \frac{2.(3n)!}{(2n+1)!(n+1)!}$ [section 5.3]

Nombres de Motzkin

- $|I_n(1234)| = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$ [79]
- $|S_n(321, 3\overline{1}42)| = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$ [45]
- $|S_n(231, 4\overline{1}32)| = |I_n(3412)| = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$ [section 4.3]

Nombres de permutations de Baxter et produit de nombres de Catalan

- $|S_n(25\overline{3}14, 41\overline{3}52)| = |S_n(21\overline{3}54, 41\overline{3}52)| = \sum_{m=0}^{n-1} \frac{\binom{n+1}{m} \cdot \binom{n+1}{m+1} \cdot \binom{n+1}{m+2}}{\binom{n+1}{1} \cdot \binom{n+1}{2}}$ [15, 108, chapitre 6]
- $|\widehat{S}_n(25\overline{3}14, 41\overline{3}52)| = c_{\lceil \frac{n}{2} \rceil} . c_{\lfloor \frac{n}{2} \rfloor}$ [18, chapitre 6]

Involutions excluant le motif identité d'ordre au plus 6

- $|I_n(123)| = \binom{n}{\lfloor \frac{n}{2} \rfloor}$
- $|I_n(1234)| = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} c_i$ [79]
- $|I_n(12345)| = c_{\lceil \frac{n+1}{2} \rceil} . c_{\lfloor \frac{n+1}{2} \rfloor}$ [49]
- $|I_n(123456)| = 6 \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n!(2i+2)!}{(n-2i)!i!(i+1)!(i+2)!(i+3)!}$ [49]

Bibliographie

[1] **D. Arquès**, Une relation fonctionnelle nouvelle sur les cartes planaires pointées, *Journal of Combinatorial Theory (Series B)* **39** (1985) 27–42.

[2] **E. Babson** et **J. West**, The permutations $123p_4 \dots p_l$ and $321p_l \dots p_4$ are Wilf equivalent, soumis à *Society for Industrial and Applied Mathematics Journal of Discrete Mathematics*.

[3] **E. Barcucci**, **A. Del Lungo**, **E. Pergola** et **R. Pinzani**, Towards a methodology for tree enumeration, $7^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Marne-la-Vallée (1995) 53–65.

[4] **G. Baxter**, On fixed points of the composite of commuting functions, *Proceedings of the American Mathematical Society* **15** (1964) 851–855.

[5] **J.S. Beissinger**, Similar constructions for Young tableaux and involutions, and their application to shiftable tableaux, *Discrete Mathematics* **67** (1987) 149–163.

[6] **F. Bergeron**, **L. Favreau** et **D. Krob**, Some conjectures on the enumeration of tableaux of bounded height, pré-publication.

[7] **F. Bergeron**, **G. Labelle** et **P. Leroux**, Théorie des espèces et combinatoire des structures arborescentes, *Publication du Laboratoire de Combinatoire et d'Informatique Mathématique de l'Université du Québec à Montréal* **19** (1994).

[8] **J. Berstel**, Axel Thue's papers on repetitions in words : a translation, *Publication du Laboratoire de Combinatoire et d'Informatique Mathématique de l'Université du Québec à Montréal* **20** (1994).

[9] **S. Billey**, **W. Jockusch** et **R.P. Stanley**, Some combinatorial properties of Schubert polynomials, *Journal of Algebraic Combinatorics* **2** (1993) 345–374.

[10] **M. Bóna**, Permutations avoiding certain patterns : the case of length 4 and some generalizations, pré-publication.

[11] **W.G. Brown**, Enumeration of non-separable planar maps, *Canadian Journal of Mathematics* **15** (1963) 526–545.

[12] **W.G. Brown** et **W.T. Tutte**, On the enumeration of rooted non separable planar maps, *Canadian Journal of Mathematics* **16** (1964) 572–577.

[13] **E. Catalan**, Note sur une équation aux différences finies, *Journal de Mathématiques Pures et Appliquées* **3** (1838) 508–516.

[14] **S.J. Chang** et **K.Y. Lin**, Rigourous results for the number of convex polygons on the square and honeycomb lattices, *Journal of Physics A: Mathematical and General* **21** (1988) 2635–2642.

[15] **F.R.K. Chung**, **R.L. Graham**, **V.E. Hoggatt** et **M. Kleiman**, The number of Baxter permutations, *Journal of Combinatorial Theory (Series A)* **24** (1978) 382–394.

[16] **R. Cori**, Un code pour les graphes planaires et ses applications, *Astérisque, Société Mathématique de France* **27** (1975).

[17] **R. Cori**, Bijective census of rooted planar maps : a survey, $5^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Florence (1993) 131–141.

[18] **R. Cori**, **S. Dulucq** et **G. Viennot**, Shuffle of parenthesis systems and Baxter permutations, *Journal of Combinatorial Theory (Series A)* **43** (1986) 1–22.

[19] **R. Cori** et **J. Richard**, Enumération des graphes planaires à l'aide des séries formelles en variables non commutatives, *Discrete Mathematics* **2** (1972) 115–162.

[20] **M. Delest**, Langages algébriques : à la frontière entre la combinatoire et l'informatique, $6^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Dimacs (1994) 69–78.

[21] **M. Delest** et **X. Viennot**, Algebraic languages and polyominoes enumeration, *Theorical Computer Science* **34** (1984) 169–206.

[22] **M. Desainte Catherine** et **G. Viennot**, Enumeration of certain Young tableaux with bounded height, Combinatoire Enumérative, G. Labelle et P. Leroux édition, *Lecture Notes in Mathematics* **1234** (1986), Springer-Verlag, 58–67.

[23] **R. Donaghey**, Restricted plane tree representations of four Motzkin-Catalan equations, *Journal of Combinatorial Theory (Series B)* **22** (1977) 114–121.

[24] **R. Donaghey**, Automorphisms on Catalan trees and bracketings, *Journal of Combinatorial Theory (Series B)* **29** (1980) 75–90.

[25] **R. Donaghey** et **L.W. Shapiro**, Motzkin numbers, *Journal of Combinatorial Theory (Series A)* **23** (1977) 291–301.

[26] **S. Dulucq**, Equations avec opérateurs : un outil combinatoire, *Thèse de l'Université Bordeaux I* (1980).

[27] **S. Dulucq**, **S. Gire** et **O. Guibert**, A combinatorial proof of J. West's conjecture, soumis à *Discrete Mathematics*.

[28] **S. Dulucq**, **S. Gire** et **J. West**, Permutations à motifs exclus et cartes planaires non séparables, $5^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Florence (1993) 165–178, à paraître dans *Discrete Mathematics*.

[29] **S. Dulucq** et **O. Guibert**, Mots de piles, tableaux standards et permutations de Baxter, $6^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Dimacs (1994) 119–128, à paraître dans *Discrete Mathematics*.

[30] **S. Dulucq** et **O. Guibert**, Permutations de Baxter, $7^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Marne-la-Vallée (1995) 139–150, soumis à *Discrete Mathematics*.

[31] **S. Dulucq** et **J-G. Penaud**, communication personnelle.

[32] **I. Dutour**, Grammaires d'objets : énumérations, bijections et génération aléatoire, *Thèse de l'Université Bordeaux I* (1996).

[33] **I. Dutour** et **J-M. Fédou**, Grammaires d'objets, *Rapport interne du LaBRI de l'Université Bordeaux I* **963-94** (1994).

[34] **P. Erdös** et **G. Szekeres**, A combinatorial problem in geometry, *Compositio Mathematica* **2** (1935) 463–470.

[35] **K. Eriksson** et **S. Linusson**, Combinatorics of Fulton's ranked essential set, $7^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Marne-la-Vallée (1995) 195–202.

[36] **A. Errera**, Un problème d'énumération, *Mémoires publiées par l'Académie royale de Belgique*, Bruxelles, tome **11** (1931).

[37] **L. Favreau**, Combinatoire des tableaux oscillants et des polynômes de Bessel, *Thèse de l'Université Bordeaux I* (1991).

[38] **J.S. Frame**, **G. de B. Robinson** et **R.M. Trall**, The hook graphs of the symetric group, *Canadian Journal of Mathematics* **6** (1954) 316–324.

[39] **F. Françon** et **X. Viennot**, Permutations selon les pics, creux, doubles-montées, doubles-descentes, nombres d'Euler et de Genocchi, *Discrete Mathematics* **28** (1979) 21–35.

[40] **D.S. Franzblau** et **D. Zeilberger**, A bijective proof of the hook-length formula, *Journal of Algorithms* **3** (1982) 317–343.

[41] **W. Feller**, An introduction to probability theory and its applications, volume **I**, John Wiley & Sons, New York - London - Sydney (1968).

[42] **I.M. Gessel**, Symmetric functions and P-recursiveness, *Journal of Combinatorial Theory (Series A)* **53** (1990) 257–285.

[43] **I.M. Gessel** et **G. Viennot**, Binomial determinants, paths, and hook length formulae, *Advances in Mathematics* **58** (1985) 300–321.

[44] **I.M. Gessel** et **G. Viennot**, Determinants, paths, and plane partitions, pré-publication.

[45] **S. Gire**, Arbres, permutations à motifs mxclus et cartes planaires : quelques problèmes algorithmiques et combinatoires, *Thèse de l'Université Bordeaux I* (1993).

[46] **I.P. Goulden** et **J. West**, Raney paths and a combinatorial relationship between rooted nonseparable planar maps and two-stack-sortable permutations, pré-publication.

[47] **D. Gouyou-Beauchamps**, Codages par des mots et des chemins : problèmes combinatoires et algorithmiques, *Thèse d'Etat de l'Université Bordeaux I* (1985).

[48] **D. Gouyou-Beauchamps**, Chemins sous-diagonaux et tableaux de Young, Combinatoire Enumérative, G. Labelle et P. Leroux édition, *Lecture Notes in Mathematics* **1234** (1986), Springer-Verlag, 112–125.

[49] **D. Gouyou-Beauchamps**, Standard Young tableaux of height 4 and 5, *European Journal of Combinatorics* **10** (1989) 69–82.

[50] **D. Gouyou-Beauchamps** et **B. Vauquelin**, Deux propriétés combinatoires des nombres de Schröder, *Revue française d'Automatique, d'Informatique et de Recherche Opérationnelle Informatique Théorique et Applications* **22** (1988) 361–388.

[51] **C. Greene**, An extension of Schensted's theorem, *Advances in Mathematics* **14** (1974) 254–265.

[52] **C. Greene**, **A. Nijenhuis** et **H.S. Wilf**, A probabilistic proof of a formula for the number of Young tableaux of a given shape, *Advances in Mathematics* **31** (1979) 104–109.

[53] **O. Guibert**, Permutations sans sous-séquence interdite, *Mémoire de Diplôme d'Etudes Approfondies de l'Université Bordeaux I* (1992).

[54] **M.D. Haiman**, Noncommutative rational power series and algebraic generating functions, *European Journal of Combinatorics* **14** (1993) 335–339.

[55] **P. Hanlon**, Counting interval graphs, *Transactions of the American Mathematical Society* **272** (1982) 383–426.

[56] **S. Hee Choi** et **D. Gouyou-Beauchamps**, Enumération de tableaux de Young semi-standard, $3^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Bordeaux (1991) 229–243.

[57] **T. Hickey** et **J. Cohen**, Uniform random generation of strings in a context-free language, *Society for Industrial and Applied Mathematics Journal on Computing* **12** (1983) 645–655.

[58] **G. Higman**, Ordering by divisibility in abstract algebras, *Proceedings of the London Mathematical Society* **2** (1952) 326–336.

[59] **A.P. Hillman**, Elementary problems and solutions, *The Fibonacci Quarterly* **4** (1966) 373–378.

[60] **A. Joyal**, Une théorie combinatoire des séries formelles, *Advances in Mathematics* **42** (1981) 1–82.

[61] **D.A. Klarner**, Correspondences between plane trees and binary sequences, *Journal of Combinatorial Theory (Series A)* **9** (1970) 401–411.

[62] **D.E. Knuth**, The art of computer programming, volume **1**, Fundamental algorithms, Addison-Wesley, Reading, Massachusetts (1973).

[63] **D.E. Knuth**, The art of computer programming, volume **3**, Sorting and searching, Addison-Wesley, Reading, Massachusetts (1973).

[64] **C. Krattenthaler**, Bijective proofs of the hook formulas for the number of standard Young tableaux, ordinary and shifted, *The Electronic Journal of Combinatorics* **2** (1995) #R13.

[65] **G. Kreweras**, Sur les éventails de segments, *Cahiers du Bureau Universitaire de Recherche Opérationnelle* **15** (1970) 1–41.

[66] **G. Kreweras**, Sur les partitions non croisées d'un cycle, *Discrete Mathematics* **4** (1972) 333–350.

[67] **J-C. Lalanne**, Une involution sur les chemins de Dyck, $3^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Bordeaux (1991) 263–274.

[68] **S.K. Lando** et **A.K. Zvonkin**, Plane and projective meanders, $3^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Bordeaux (1991) 287–303.

[69] **A. Lascoux** et **M.P. Schützenberger**, Schubert polynomials and the Littlewood-Richardson rule, *Letters in Mathematical Physics* **10** (1985) 505–507.

[70] **A.B. Lehman**, A bijective census of rooted planar maps, Communication à Ontario Mathematical Conference (1970), non publié.

[71] **M. Lothaire**, Combinatorics on words, G.C. Rota édition, *Encyclopedia of Mathematics and its Applications* **17**, Addison-Wesley, Reading, MA (1983).

[72] **I.G. MacDonald**, Notes on Schubert polynomials, *Publication du Laboratoire de Combinatoire et d'Informatique Mathématique de l'Université du Québec à Montréal* **6** (1991).

[73] **P.A. MacMahon**, Combinatory analysis, Chelsea (1960), version originale publiée par Cambridge University Press, London (1915).

[74] **C.L. Mallows**, Baxter permutations rise again, *Journal of Combinatorial Theory (Series A)* **27** (1979) 394–396.

[75] **T. Motzkin**, Relations between hypersurface cross ratios, and a combinatorial formula for partitions of a polygon, for permanent preponderance, and for non-associative products, *Bulletin of the American Mathematical Society* **54** (1948) 352–360.

[76] **R.C. Mullin**, The enumeration of hamiltonian polygons in triangular maps, *Pacific Journal of Mathematics* **16** (1966) 139–145.

[77] **T.V. Narayana**, A partial order and its applications to probability theory, *Sankhya* **21** (1959) 91–98.

[78] **I.M. Pak** et **A.V. Stoyanovskii**, A bijective proof of the hook-length formula and its analogs, *Fonctional Analysis and its Applications* **26** (1992) 216–218.

[79] **A. Regev**, Asymptotic values for degrees associated with strips of Young diagrams, *Advances in Mathematics* **41** (1981) 115–136.

[80] **J.B. Remmel**, Bijective proofs of formulae for the number of standard Young tableaux, *Linear and Multilinear Algebre* **11** (1982).

[81] **J.B. Remmel** et **R. Whitney**, A bijective proof of the hook formula for the number of column strict tableaux with bounded entries, *European Journal of Combinatorics* **4** (1983) 45–63.

[82] **J. Riordan**, Enumeration of plane trees by branches and endpoints, *Journal of Combinatorial Theory (Series A)* **19** (1975) 214–222.

[83] **G. de B. Robinson**, On the representations of the symmetric group, *American Journal of Mathematics* **60** (1938) 745–760.

[84] **D.G. Rogers**, A Schröder triangle : three combinatorial problems, Combinatorial Mathematics V, C.H.C. Little édition, *Lecture Notes in Mathematics* **622** (1976), Springer-Verlag, 175–196.

[85] **D.G. Rogers**, Ascending sequences in permutations, *Discrete Mathematics* **22** (1978) 35–40.

[86] **D.G. Rogers** et **L.W. Shapiro**, Some correspondences involving the Schröder numbers and relations, Combinatorial Mathematics, D.A. Holton et J. Seberry édition, *Lecture Notes in Mathematics* **686** (1978), Springer-Verlag, 267–274.

[87] **D.G. Rogers** et **L.W. Shapiro**, Deques, trees and lattice paths, Combinatorial Mathematics VIII, K.L. MacAvaney édition, *Lecture Notes in Mathematics* **884** (1981), Springer-Verlag, 293–303.

[88] **D. Rotem**, Stack sortable permutations, *Discrete Mathematics* **33** (1981) 185–196.

[89] **C. Schensted**, Longest increasing and decreasing subsequences, *Canadian Journal of Mathematics* **13** (1961) 179–191.

[90] **E. Schröder**, Vier kombinatorische probleme, *Zeitschrift für Mathematik und Physik* **15** (1870) 361–376.

[91] **M.P. Schützenberger**, Certain elementary families of automata, *Proceedings of the Symposium on Mathematical Theory of Automata*, Polytechnic Institute of Brooklyn (1962) 139–153.

[92] **M.P. Schützenberger**, Quelques remarques sur une construction de Schensted, *Mathematica Scandinavica* **12** (1963) 117–128.

[93] **M.P. Schützenberger**, Context-free languages and pushdown automata, *Information and Control* **6** (1963) 246–264.

[94] **L.W. Shapiro** et **A.B. Stephens**, Bootstrap percolation, the Schröder numbers, and the N-kings problem, *Society for Industrial and Applied Mathematics Journal of Discrete Mathematics* **4** (1991) 275–280.

[95] **R. Simion** et **F.W. Schmidt**, Restricted permutations, *European Journal of Combinatorics* **6** (1985) 383–406.

[96] **N.J.A. Sloane**, A handbook of integer sequences, Academic Press (1973).

[97] **Z.E. Stankova**, Forbidden subsequences, *Discrete Mathematics* **132** (1994) 291–316.

[98] **R.P. Stanley**, Enumerative Combinatorics, volume **I**, Wadsworth & Brooks/Cole, Monterey, California (1986).

[99] **R.P. Stanley**, Differentiably finite power series, *European Journal of Combinatorics* **1** (1980) 175–188.

[100] **A. Thue**, Über unendliche zeichenreihen, *Norske Vidensk. Selsk. Skrifter. I. Mat. Naturv. Klasse, Christiania* **7** (1906) 1–22.

[101] **A. Thue**, Über die gegenseitige lage gleicher teile gewisser zeichenreihen, *Norske Vidensk. Selsk. Skrifter. I. Mat. Naturv. Klasse, Christiania* **10** (1912) 1–67.

[102] **W.T. Tutte**, A census of planar triangulations, *Canadian Journal of Mathematics* **14** (1962) 21–38.

[103] **W.T. Tutte**, A census of hamiltonian polygons, *Canadian Journal of Mathematics* **14** (1962) 402–417.

[104] **W.T. Tutte**, A census of slicings, *Canadian Journal of Mathematics* **14** (1962) 708–722.

[105] **W.T. Tutte**, A census of planar maps, *Canadian Journal of Mathematics* **15** (1963) 249–271.

[106] **B.L. Van der Waerden**, Beweis einer Baudet'schen vermutung, *Nieuw Arch. Wisk.* **15** (1927) 212–216.

[107] **G. Viennot**, Une forme géométrique de la correspondance de Robinson-Schensted, Combinatoire et Représentation du Groupe Symétrique, D. Foata édition, *Lecture Notes in Mathematics* **579** (1977), Springer-Verlag, 29–58.

[108] **G. Viennot**, A bijective proof for the number of Baxter permutations, $3^{ème}$ *Séminaire Lotharingien de Combinatoire*, Le Klebach (1981) 28–29, également paru sous la forme d'un résumé de l'American Mathematical Society suite à une session spéciale de Combinatoire, Minnéapolis, Novembre 1984.

[109] **X.G. Viennot**, A survey of polyominoes enumeration, $4^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Montréal (1992) 399–420.

[110] **J. West**, Permutations with forbidden subsequences and stack-sortable permutations, PHD-thesis, Massachusetts Institute of Technology, Cambridge (1990).

[111] **J. West**, A catalogue of forbidden subsequence results, pré-publication.

[112] **J. West**, Permutation trees and the Catalan and Schröder numbers, à paraître dans *Discrete Mathematics*.

[113] **J. West**, Sorting twice through a stack, $3^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Bordeaux (1991) 397–406, *Theorical Computer Science* **117** (1993) 303–313.

[114] **J. West**, Generating trees and forbidden subsequences, $6^{ème}$ *conférence Séries Formelles et Combinatoire Algébrique*, Dimacs (1994) 441–450.

[115] **J. West**, Wilf-equivalence for singleton classes, pré-publication.

[116] **A. Young**, The collected papers of Alfred Young, *Mathematical Expositions* **21**, University of Toronto Press.

[117] **D. Zeilberger**, A short hook-lengths bijection inspired by the Greene-Nijenhuis-Wilf proof, *Discrete Mathematics* **51** (1984) 101–108.

[118] **D. Zeilberger**, A holonomic systems approach to special functions identities, *Journal of Computational and Applied Mathematics* **32** (1990) 321–368.

[119] **D. Zeilberger**, A proof of Julian West's conjecture that the number of two-stack sortable permutations of length n is $2(3n)!/((n+1)!(2n+1)!)$, *Discrete Mathematics* **102** (1992) 85–93.

Résumé

Ces travaux portent sur la Combinatoire des permutations à motifs exclus et de certains mots codant les mouvements de deux piles.

Nous obtenons des formules d'énumération pour plusieurs ensembles de permutations à motifs exclus en utilisant la méthode des arbres de génération avec le concours du logiciel *forbid* que nous avons développé. C'est ainsi que la correspondance que nous donnons entre permutations triables par deux passages consécutifs dans une pile et permutations non séparables (elles-mêmes en bijection avec les cartes planaires pointées non séparables) aboutit sur l'obtention d'une preuve de la conjecture de J. West.

Ensuite, nous établissons une nouvelle bijection entre permutations de Baxter et certains triplets de chemins deux à deux disjoints. Cette correspondance, dans laquelle le caractère alternant des permutations s'interprète naturellement, unifie des travaux antérieurs sur le sujet.

Finalement, nous prouvons combinatoirement trois conjectures sur l'énumération de certains mots de piles en faisant intervenir tableaux de Young standard, permutations de Baxter, permutations non séparables et cartes planaires cubiques pointées non séparables.

Mots clefs

arbres	énumération
bijection	mots
cartes planaires	permutations à motifs exclus
combinatoire	tableaux de Young

N° d'ordre : 1898

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par Philippe DUCHON

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Q-grammaires: un outil pour l'énumération

Soutenue le : 5 juin 1998

Après avis de : MM. Jean-Marc Fédou Rapporteurs
Renzo Pinzani ...

Devant la Commission d'examen formée de :

MM.	Philippe Flajolet	Directeur de recherche à l'INRIA	Président
	Mireille Bousquet-Mélou	Chargée de recherche au CNRS .	Rapporteur
	Maylis Delest	Professeur	Examinateurs
	Jean-Marc Fédou	Professeur	
	Daniel Krob	Directeur de recherche au CNRS	
	Renzo Pinzani	Professeur	

1998

Les travaux de Philippe Flajolet ont souvent été pour moi une prćieuse source d'inspiration; c'est un grand honneur qu'il me fait en présidant ce jury, et je l'en remercie vivement.

Maylis Delest a su, dans un emploi du temps toujours chargé, trouver le temps de me prodiguer conseils et encouragements, tout en me laissant la liberté d'avancer à mon propre rythme. Je tiens à l'en remercier, et à lui dire tout le plaisir que j'ai eu à travailler avec elle.

Jean-Marc Fédou et Renzo Pinzani ont lu avec attention mon manuscrit, et m'ont offert conseils et suggestions. Je leur en suis reconnaissant.

Mireille Bousquet-Mélou et Daniel Krob ont également accepté de faire partie de ce jury, et je les en remercie.

Xavier Viennot, par son enthousiasme et la clarté de ses cours, m'a fait découvrir la combinatoire et m'a attiré à Bordeaux; je profite de cette occasion pour lui exprimer toute ma gratitude.

Mes divers camarades de bureau, et tout particulièrement Emmanuel Godard et Augustin Ido, ont supporté avec courage mes moments de découragement tout au long de la rédaction de cette thèse; je leur souhaite de se trouver, le moment venu, dans une aussi bonne ambiance.

Marie-Line sait déjà combien sa présence et son soutien m'ont apporté, mais je le lui redis tout de même.

Table des matières

Introduction

L'objet de la combinatoire énumérative peut être résumé ainsi : déterminer, de manière exacte ou approchée, le *nombre* d'objets vérifiant des propriétés données. Le plus souvent, on s'intéresse à une classe infinie $\mathcal{A}$ d'objets (figures planes, mots d'un langage, cartes, arbres...), sur lesquels on définit un certain nombre de *paramètres* : nombre de sommets, hauteur, largeur, nombre de sous-arbres vérifiant telle ou telle propriété ... Les problèmes peuvent provenir de l'informatique théorique (analyse en moyenne ou dans le cas le pire d'algorithmes et de structures de données), des mathématiques (repésentations du groupe symétrique, bases de fonctions symétriques ...), ou de la physique statistique (modèles discrets de percolation, fonctions de partitions ...). Bien que d'origines et souvent de natures très différentes, ces problèmes ont souvent la caractéristique commune de pouvoir être modélisés par des objets relativement simples, qui se prêtent bien à l'énumération par des techniques combinatoires.

Lorsqu'un paramètre p est suffisamment discriminant pour que, pour chaque valeur possible n de ce paramètre, l'ensemble $\mathcal{A}_n$ des objets de $\mathcal{A}$ pour lesquels le paramètre prend cette valeur soit *fini*, on peut alors se poser la question de savoir *combien* il y a de tels objets. Cette énumération peut être soit exacte, soit approchée. Dans ce dernier cas, on donne un développement asymptotique ou un simple équivalent de la suite $(|\mathcal{A}_n|)_{n\geq 0}$.

L'outil le plus fréquemment utilisé dans les problèmes d'énumération est la *série génératrice* : à chaque objet w d'une classe $\mathcal{A}$, on associe un *poids*, ou *valuation*, $v(w)$, pris dans un anneau de polynômes $\mathbb{K}[x_1, \ldots, x_k]$; la série génératrice de la classe $\mathcal{A}$ suivant la valuation v est alors, sous réserve de convergence, la somme formelle des poids des objets

$$A(x_1, \ldots, x_k) = \sum_{w \in \mathcal{A}} v(w).$$

Généralement, on associe à chacune des variables x_i, un paramètre p_i, défini sur $\mathcal{A}$ et ne prenant que des valeurs positives ou nulles. La valuation v est alors définie par

$$v(w) = \prod_{1 \leq i \leq k} x_i^{p_i(w)},$$

et la série génératrice qui en résulte est appelée *série génératrice ordinaire*[1] *de* $\mathcal{A}$ *suivant les paramètres* $p_1, \ldots, p_k$.

Savoir ce qui constitue une solution d'un problème d'énumération n'est pas forcément évident. La réponse dépend fortement de la complexité du problème d'énumération. Ce peut être une formule donnant la série génératrice, ou une autre donnant les coefficients de Taylor de cette série; ou même une simple équation portant sur la série génératrice, ou une relation de récurrence vérifiée par ses coefficients de Taylor. De manière approchée, on pourra se concentrer sur un équivalent asymptotique de la série ou de ses coefficients. Les travaux d'Odlyzko [64], Flajolet et Odlyzko [41], et Flajolet et Sedgewick [45, 46], montrent en quoi le comportement asymptotique des coefficients de Taylor d'une série est lié à son comportement au voisinage de ses singularités dominantes. Pour les problèmes d'origine physique, où les objets combinatoires sont souvent des approximations discrètes de modèles continus, le comportement asymptotique est généralement plus significatif que l'énumération exacte.

Lorsque les séries génératrices sont données par des équations, il est usuel de classifier les problèmes suivant le *type* de ces équations : algébriques, différentielles, fonctionnelles ... Un phénomène fréquemment rencontré pour les séries génératrices à plusieurs variables est qu'une des variables q apparaisse de telle sorte qu'à la limite $q \to 1$, l'équation se transforme en une équation beaucoup plus simple; ces équations sont appelées *q-équations*.

Le travail présenté dans cette thèse se situe dans le cadre de la combinatoire énumérative. Nous formalisons la notion de Q-grammaires, et montrons en quoi elles constituent un moyen d'approche de certains problèmes d'énumération suivant plusieurs paramètres.

Différentes méthodes d'énumération

Une première méthode d'approche possible consiste à calculer par des méthodes appropriées les premières termes de la série génératrice (Redelmeier [72]). Souvent, les objets de "petite taille" sont peu nombreux, et il est ainsi possible d'obtenir plusieurs termes de la suite des coefficients de Taylor. Dans certains cas, une simple comparaison avec un ensemble de suites connues [75, 76] permet d'identifier la suite. Des techniques plus évoluées d'approximation [17] recherchent, à partir des premiers termes, une équation algébrique ou différentielle susceptible d'être satisfaite par la série génératrice. Ces calculs sont automatisés par la bibliothèque Maple *gfun* [43].

1. *Une autre forme fréquemment employée de série génératrice est la série génératrice* exponentielle, *dans laquelle la valuation d'un objet de "taille" n est divisée par n!.*

Pour certains problèmes, il est plus facile de chercher à obtenir directement une équation fonctionnelle satisfaite par la série génératrice. Le principe de la combinatoire *bijective* est d'exhiber entre deux classes d'objets combinatoires une bijection qui conserve la valuation; on en déduit alors l'égalité des séries génératrices. Le plus souvent, on s'attache à "décomposer" les objets étudiés de manière à établir une bijection avec une classe d'objets dont l'énumération directe est possible. Lorsqu'une telle bijection est établie entre deux classes d'objets dont les séries génératrices sont déjà connues, elle fournit alors une *preuve bijective* de l'identité des séries, que cette identité ait été ou non préalablement démontrée par le calcul. L'intérêt d'une telle preuve bijective réside généralement dans le fait qu'elle permet en quelque sorte d'"expliquer" certaines propriétés des objets étudiés, en les reliant à des propriétés connues d'autres objets.

Les polyominos verticalement convexes peuvent naturellement se découper en "tranches" successives. En tenant compte de différents paramètres (aire, largeur, périmètre, et hauteur de la dernière tranche), il est possible de dire comment évoluent ces paramètres lorsqu'une nouvelle colonne est ajoutée. On en déduit une équation fonctionnelle vérifiée par la série génératrice ou, de manière équivalente, des relations de récurrence portant sur les coefficients. Cette méthode "à la Temperley" [77, 14] permet ainsi d'étudier différentes classes de polyominos verticalement convexes. La méthode ECO, qui combine ces idées avec l'utilisation d'arbres de génération, a également été appliquée à l'énumération de diverses classes de chemins colorés [6] et d'arbres planaires [7]

Dans la méthodologie DSV, développée initialement par Schützenberger dans [73, 74], on relie l'algébricité de certaines séries génératrices à la théorie des *langages algébriques* non ambigus [1, 5]. Le principe est d'établir une bijection (ou *codage*) entre les objets à étudier et les mots d'un langage *algébrique*, de telle sorte que le paramètre *taille* des objets corresponde à la *longueur* des mots; éventuellement, les différentes lettres de l'alphabet peuvent correspondre à différents paramètres intéressants. Une grammaire non ambiguë engendrant le langage, fournit alors, d'après un théorème de Chomsky et Schützenberger [18], un système d'équations algébriques dont la série génératrice cherchée est une des composantes d'une solution. Ce système peut alors être résolu explicitement, ou, s'il est trop complexe, des informations précises sur les coefficients de la série génératrice peuvent en être extraites. Dans les cas simples, la formule d'inversion de Lagrange [81] ou l'une de ses variantes [53, 50], permet d'obtenir des expressions exactes pour les coefficients; dans d'autres, où aucune formule acceptable n'est accessible, l'analyse des singularités peut donner des renseignements précis sur l'asymptotique des coefficients.

Enfin, les coefficients de Taylor de séries algébriques vérifiant des relations de récurrence

polynomiales (Comtet [21]), il est possible de calculer explicitement un grand nombre de coefficients.

L'énumération des polyominos convexes suivant le périmètre, par Delest et Viennot [31], constitue un exemple de résultat nouveau obtenu par l'application de cette méthode. On trouvera d'autres exemples dans [30, 25].

Les *grammaires à opérateurs* (Cori et Richard [24], Cori [23], Chottin [19]) constituent une variante de la méthodologie DSV, et permettent de traiter certains cas où l'on code les objets par les mots d'un langage qui n'est pas forcément algébrique, mais qui est solution d'un système d'équations avec opérateurs en variables non commutatives. Si les opérateurs employés ont une image en variables commutatives, cette équation se traduit directement sur la série génératrice. Cette méthode a notamment permis à Cori et Richard de retrouver certains résultats d'énumération de cartes planaires dûs à Tutte [78].

Il est possible de s'affranchir du passage par les mots en remarquant qu'à certaines opérations sur les objets, correspondent des opérations sur les séries génératrices. Ainsi, l'union disjointe de deux classes d'objets correspond à la somme des séries génératrices, et le produit cartésien, au produit des séries. Ces idées sont utilisées dans la théorie des *structures décomposables* [42, 43, 44, 47], ou, de manière plus visuelle, dans les *grammaires d'objets* [36].

Une même classe d'objets, énumérée suivant des paramètres différents, admet généralement des séries génératrices qui n'ont aucune ressemblance entre elles. Obtenir la série génératrice suivant *plusieurs* paramètres est souvent beaucoup plus compliqué; dès lors que la série génératrice suivant l'un de ces paramètres n'est pas algébrique, la série multivariée n'est pas algébrique, et la méthodologie DSV ne peut s'appliquer directement. Par ailleurs, il arrive que les différentes séries à une variable soient algébriques sans que la série multivariée le soit. Ainsi, les *polyominos verticalement convexes* ont une série génératrice suivant l'aire qui est *rationnelle* (Temperley [77], Klarner [58, 59]), et une série génératrice suivant le périmètre *algébrique* (Delest [25], Feretić [38, 39]), mais la série génératrice bivariée suivant ces deux paramètres est beaucoup plus compliquée, et n'est pas algébrique (Bousquet-Mélou [14]).

La notion de q-grammaire, introduite par Delest et Fédou dans [28], repose sur l'idée que le codage par les mots d'un langage algébrique fournit en fait une *structure* sur les objets codés. En adaptant la notion de *grammaire d'attributs* (Knuth [61]) utilisée en compilation, il est parfois possible d'écrire des équations *non algébriques* vérifiées par la série génératrice suivant un paramètre supplémentaire, compté par une nouvelle variable q. Ces équations

sont alors des q-analogues des équations algébriques de départ : une variable x est parfois remplacée par xq dans certaines séries inconnues. En posant $q = 1$ (ce qui revient à "oublier" le paramètre compté par q), on retrouve les équations algébriques. Les séries génératrices solutions de telles équations se présentent donc sous la forme de q-séries [4].

Un exemple d'utilisation de q-grammaires pour l'énumération des polyominos parallélogrammes (suivant l'aire et le périmètre) est donné par Delest et Fédou dans [29]. On trouvera d'autres exemples dans les travaux de Denise et Simion [32] (hauteurs de pyramides et paires extérieures dans les chemins de Dyck), et dans ceux de Delest, Dubernard et Dutour [27] (énumération des polyominos parallélogrammes suivant l'aire, la largeur et le nombre de coins).

La résolution de q-équations dans le cas général est un problème complètement ouvert. Il existe plusieurs q-analogues de la formule d'inversion de Lagrange, dus à Andrews [3], Gessel [49], Garsia [48], Gessel et Stanton [51] ou Krattenthaler [62]. Prellberg et Brak [69] et Bousquet-Mélou [14] ont décrit des méthodes permettant de résoudre des cas particuliers de q-équations. La résolution d'équations q-différentielles a également permis à Bousquet-Mélou et Fédou [16] d'obtenir une expression relativement simple pour la série génératrice des polyominos convexes suivant l'aire et le périmètre.

Notre principal objet d'étude, les Q-grammaires, constitue une généralisation de la notion de q-grammaire. Nous nous intéressons à tous les paramètres qu'il est possible d'"attraper" en s'autorisant à multiplier certaines variables d'énumération par d'autres. Ces paramètres, qui dépendent de la grammaire, sont appelés *Q-comptables*.

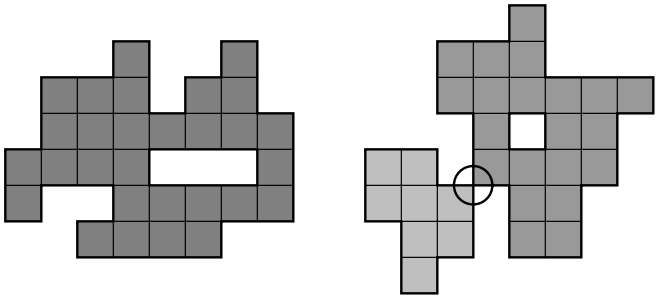
Polyominos

Nous donnons ici une brève définition des polyominos, car ceux-ci constituent le support de plusieurs exemples étudiés dans ce mémoire.

Les polyominos sont un sujet d'étude fréquent en combinatoire. Les premières recherches ont concerné le pavage de régions du plan au moyen de polyominos donnés (le terme de polyomino est généralement attribué à Golomb [52]). Les problèmes d'énumération ont rapidement été abordés (Temperley [77]; Read [71]; Klarner [58, 59]; Pólya [67]). Des résultats asymptotiques assez précis ont été obtenus pour des sous-classes de polyominos (Bender [9]; Klarner et Rivest [60]).

Une *cellule* est un carré unitaire du plan $\mathbb{R}^2$, dont les sommets ont des coordonnées entières. Un *polyomino* est une union finie de cellules, dont l'intérieur est connexe – voir

figure 1.



Un polyomino

Deux polyominos avec un point de contact

FIG. 1: *Exemples de polyominos*

Nous considérerons toujours les polyominos *à translation près*: deux polyominos qui sont images l'un de l'autre par une translation sont pour nous identiques. En revanche, deux polyominos distincts peuvent être images l'un de l'autre par une symétrie ou une rotation.

En physique statistique, où ils sont utilisés comme modèles discrets, les polyominos portent généralement le nom d'*animaux*, et les cellules sont souvent remplacées par leurs centres.

L'énumération des polyominos dans le cas général est un problème extrêmement difficile et complètement ouvert. Le nombre a_n de polyominos d'aire n n'est connu que jusqu'à $n = 24$ (Redelmeier [72]). Le comportement asymptotique de la suite (a_n) n'est connu que de manière partielle: $\lim(a_n)^{1/n} = \mu$, avec $3.72 < \mu < 4.64$.

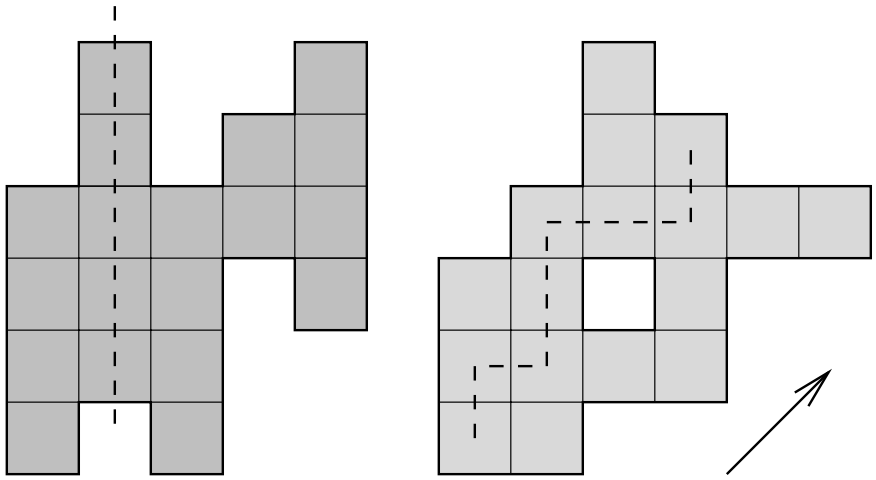
Une manière de rendre abordables les problèmes d'énumération de polyominos est de se restreindre à des sous-classes définies par des propriétés particulières. Les propriétés qui, jusqu'à présent, se sont révélées les plus fécondes (tout au moins du point de vue des résultats d'énumération) sont la *convexité* suivant une direction, et le fait d'être *dirigés* suivant une direction donnée. On trouvera des survols dans [55, 26, 79], et un autre plus récent dans [15], qui contient également une bibliographie détaillée sur le sujet.

Si $\mathcal{D}$ est une direction de droites dans le plan, un polyomino P est *convexe* suivant la direction $\mathcal{D}$ si son intersection avec toute droite de $\mathcal{D}$ qui passe par le centre d'une cellule est un segment. Le plus souvent, la direction $\mathcal{D}$ est soit verticale, soit horizontale[2];

2. Bien évidemment, une symétrie transforme bijectivement les polyominos verticalement convexes en polyominos horizontalement convexes.

les polyominos *diagonalement convexes* ont également été abordés. Un polyomino à la fois horizontalement et verticalement convexe est dit tout simplement *convexe*.

Un polyomino P est *dirigé* suivant la direction Sud-Ouest/Nord-Est s'il existe une cellule $c \subset P$, appelé *cellule source*, telle que chaque cellule de P peut être atteinte à partir de c en n'effectuant que des pas élémentaires Nord ou Est, sans passer par une cellule extérieure à P. Si l'on autorise également les pas Sud, le polyomino est dit *semi-dirigé* suivant la direction Ouest-Est.



verticalement convexe dirigé Sud-Ouest/Nord-Est

FIG. 2: *Exemples de polyominos avec contraintes*

En combinant les conditions de convexité et de directions privilégiées, il est possible de définir la plupart des familles de polyominos étudiées dans la littérature. Ainsi, les polyominos *parallélogrammes* sont exactement ceux qui sont verticalement et diagonalement convexes, et dirigés suivant les directions Sud-Ouest/Nord-Est et Nord-Ouest/Sud-Est; les polyominos *murs*, qui correspondent aux *compositions* d'entiers, sont exactement les polyominos verticalement convexes qui sont dirigés à la fois vers le Nord-Est *et* vers le Nord-Ouest.

Plan de la thèse

Le chapitre 1 introduit la plupart des définitions et notations classiques utilisées. La seule notion qui diffère sensiblement de sa définition usuelle est celle des arbres de dérivation d'une grammaire, qui seront pour nous étiquetés par les règles de réécriture.

Le chapitre 2 introduit les notions de *Q-grammaire* et de *paramètre Q-comptable*.

Le résultat principal est le théorème 2.36, qui établit l'équivalence entre Q-grammaires et certaines formes d'équations vérifiées par les séries génératrices. Les paramètres Q-comptables sont également interprétés en termes d'arbres de dérivation, et leurs ordres de grandeur maximaux sont déterminés.

Le chapitre 3 étudie dans quelles circonstances il est possible de remplacer une grammaire par une autre engendrant le même langage, sans perdre de paramètres Q-comptables. Au moyen de différents lemmes, nous montrons en particulier qu'il est possible de donner des formes normales aux Q-grammaires.

Le chapitre 4 est consacré aux calculs de statistiques sur les paramètres Q-comptables. Par différentiation, les séries de moments sont décrites (théorème 4.14). Les problèmes d'asymptotiques sont également abordés.

Le chapitre 5 donne, à titre d'exemples d'applications, des Q-grammaires pour différentes familles de polyominos verticalement convexes. Dans chaque cas, nous étudions un certain nombre de paramètres et déterminons ceux qui sont Q-comptables pour les grammaires données.

Chapitre 1

Définitions

Dans ce chapitre, nous nous contentons de définir les objets de base utilisés dans ce travail : langages, arbres, séries génératrices, grammaires, et chemins du plan discret. Seuls les arbres de dérivation diffèrent légèrement de leur définition la plus classique.

1.1 Mots et langages

Définition 1.1. Pour tout ensemble X, X^* désigne l'ensemble des suites finies (éventuellement vides) d'éléments de X. Une telle suite w est également appelée un *mot* de X^*, et sa longueur sera notée $|w|$. Chaque terme de la suite est appelé une *lettre*.

L'opération de concaténation fait de X^* un monoïde (monoïde libre engendré par X).

La suite (unique) de longueur nulle, aussi appelée *mot vide*, sera notée ϵ.

Définition 1.2. On appelle *langage* sur X une partie L de X^*; X est alors appelé l'*alphabet* de L.

Nous ne nous intéresserons qu'à des langages sur des alphabets finis. Il existe une classification extrêmement détaillée pour ces langages; voir à ce sujet [11]. Dans ce travail, nous nous intéresserons plus particulièrement aux langages *algébriques*, définis à la section 1.4.

Définition 1.3. Un mot w' est un *facteur* d'un mot w, s'il existe des mots w_1 et w_2 tels que $w = w_1 w' w_2$.

Le mot w' est un *facteur gauche* si $w_1 = \epsilon$, et un *facteur droit* si $w_2 = \epsilon$. Un facteur de w qui n'est ni w, ni le mot vide, est appelé *facteur propre*.

De manière plus générale, on peut parler de *factorisation* :

Définition 1.4. Une *factorisation* d'un mot w est une suite finie de mots $(w_1, \ldots, w_k)$, telle que $w = w_1 w_2 \ldots w_k$. L'entier k est la *longueur* de la factorisation.

Lorsqu'aucun des mots w_i n'est vide, la factorisation est dite *propre*.

Ainsi, un facteur propre de w est un facteur qui peut apparaître dans une factorisation propre non triviale (dont la longueur n'est pas 1).

Définition 1.5. Un *sous-mot* d'un mot $w = x_1 \ldots x_n$ $(x_i \in X)$ est une sous-suite $w' = x_{i_1} \ldots x_{i_k}$, avec $i_1 < i_2 < \ldots < i_k$.

Il est clair qu'un facteur d'un mot w, n'est rien d'autre qu'un sous-mot pour lesquels les indices $i_1, \ldots, i_k$ sont consécutifs.

Définition 1.6.
- Le *nombre d'occurrences* d'un *facteur* w' de w, est le nombre de factorisations distinctes $F = (w_1, w', w_2)$ de w.
- Le *nombre d'occurrences* d'un *sous-mot* w' de $w = x_1 x_2 \ldots x_n$, est le nombre de sous-suites $i_1 < \ldots < i_k$ de $1, \ldots, n$, telles que $w' = x_{i_1} \ldots x_{i_k}$.

Exemple 1.7. Dans le mot $w = aababbab$, il y a 3 occurrences du facteur ab, mais 12 occurrences du sous-mot ab.

Un exemple classique de nombre d'occurrences de sous-mots est celui du nombre d'inversions :

Définition 1.8. Supposons l'alphabet X totalement ordonné. Une *inversion* d'un mot $w = x_1 \ldots x_n$, est un couple (i, j) tel que $i < j$ et $x_i > x_j$.

Le nombre d'inversions d'un mot w est noté $\mathrm{inv}(w)$.

Le nombre d'inversions d'un mot est la somme des nombres d'occurrences de tous les sous-mots possibles de longueur 2 dont les lettres sont décroissantes. Ainsi, dans le cas d'un alphabet à deux lettres $\{a, b\}$ (avec $a < b$), il s'agit tout simplement du nombre d'occurrences du sous-mot ba.

Notation 1.9. Soient $w \in X^*$, $x \in X$, et $A \subset X$. Le nombre d'occurrences de x dans w est noté $|w|_x$, et également appelé *longueur de w en x*.

De même, le nombre de lettres de w qui appartiennent à A est appelé *longueur de w en A*, et noté $|w|_A$.

1.2 Arbres

Il existe différentes catégories d'arbres en informatique; dans cette thèse, le terme d'arbre désigne exclusivement un arbre planaire, généralement étiqueté.

1.2.1 Arbres planaires

Définition 1.10. Soit S un ensemble fini non vide. Un *arbre d'ensemble de sommets* S est défini de la manière suivante :

- Si S est un singleton, $\mathcal{A} = (s)$ est le seul arbre d'ensemble de sommets S.
- Sinon, $\mathcal{A} = (s, \mathcal{A}_1, \dots, \mathcal{A}_k)$, où $s \in S$, et pour chaque $i \leq k$, $\mathcal{A}_i$ est un arbre d'ensemble de sommets S_i, avec la condition que $\{S_1, \dots, S_k\}$ forme une partition de $S \backslash \{s\}$.

Dans les deux cas, s est appelé la *racine* de $\mathcal{A}$.

Avec les notations de la définition 1.10, les éléments de S sont appelés *sommets* ou *nœuds* de $\mathcal{A}$. La *taille* de $\mathcal{A}$ est son nombre de sommets.

Lorsque $\mathcal{A} = (s, \mathcal{A}_1, \dots, \mathcal{A}_k)$ est un arbre, les racines respectives $s_1, \dots, s_k$ des arbres $\mathcal{A}_1, \dots, \mathcal{A}_k$ sont appelés *fils* de s, et s est leur *père* commun. $\mathcal{A}_i$ est appelé le *i-ème sous-arbre* de $\mathcal{A}$, ou encore le *sous-arbre issu de s_i*; de manière générale, un *sous-arbre de* $\mathcal{A}$ est le sous-arbre issu d'un sommet quelconque de $\mathcal{A}$.

Traditionnellement, les arbres sont représentés par des graphes, chaque sommet étant relié à ses fils ordonnés de gauche à droite et placés en dessous de lui, de sorte que la racine est placée en haut de la figure (voir figure 1.1).

Un sommet d'un arbre qui n'a aucun fils est appelé une *feuille*. Les sommets qui ne sont pas des feuilles sont parfois appelés sommets ou nœuds *internes*.

Une *branche* d'un arbre est une suite $(s_1, \dots, s_k)$ de sommets telle que, pour $1 \leq i < k$, s_{i+1} soit un fils de s_i, s_1 étant la racine de l'arbre. L'entier k est la *longueur* de la branche. Pour chacun des sommets s d'un arbre, il existe une unique branche qui se termine en s. La *hauteur* de l'arbre $\mathcal{A}$ est la plus grande longueur de ses branches.

Si $(s_1, \dots, s_k)$ est une branche d'un arbre, k est la *profondeur* du sommet s_k dans cet arbre.

Une branche est *maximale* (pour l'inclusion) si son dernier sommet est une feuille de l'arbre.

Les *descendants* d'un sommet s sont tous les sommets du sous-arbre issu de s; les *ancêtres* de s sont tous les sommets qui composent la branche dont il est le dernier sommet. En ce sens, un sommet est son propre ancêtre et son propre descendant.

Si $\mathcal{A}$ et $\mathcal{B}$ sont deux arbres, et si s est un sommet de $\mathcal{A}$, on notera $\mathcal{A}(s, \mathcal{B})$ l'arbre obtenu en remplaçant, dans $\mathcal{A}$, le sous-arbre de racine s par $\mathcal{B}$ (voir figure 1.1).

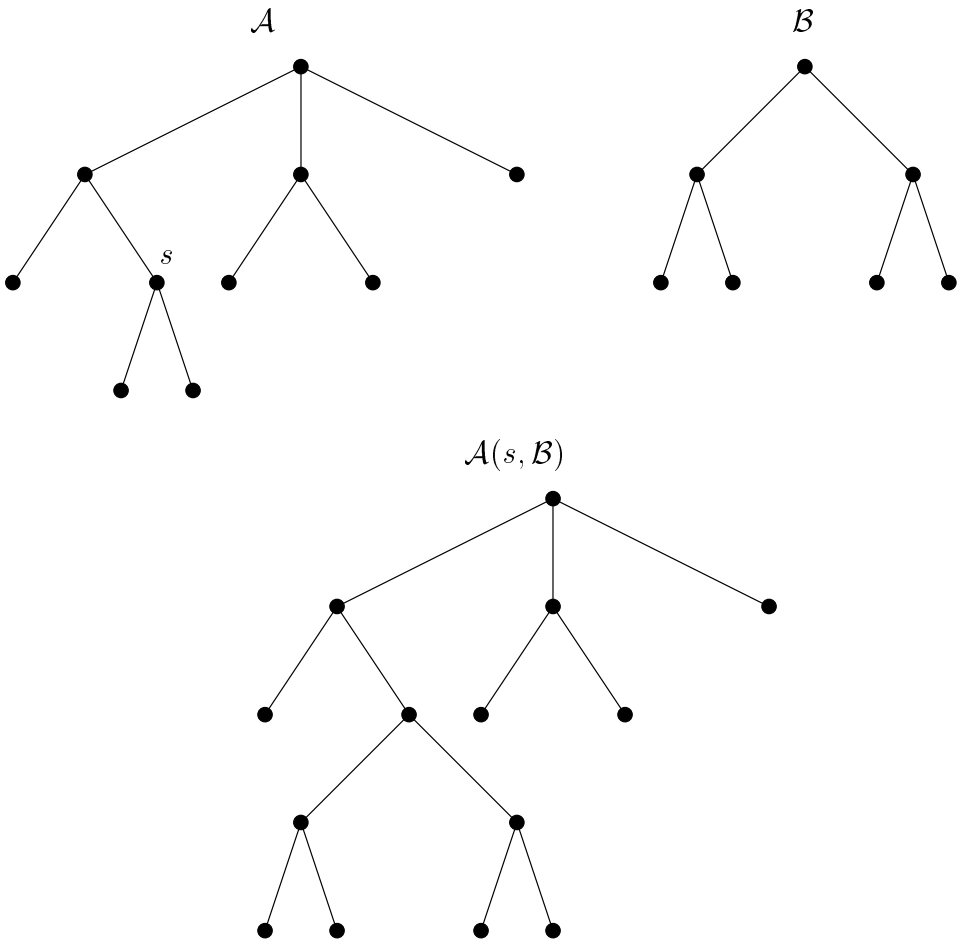


FIG. 1.1: *Substitution d'arbres*

Dans le cadre de cette thèse, les arbres manipulés sont exclusivement des *arbres de dérivation* dans une grammaire; voir section 1.4.

1.3 Séries formelles et séries génératrices

Soient $X = \{x_1, \ldots, x_k\}$ un alphabet fini, et A un anneau unitaire; $A \ll X \gg$ désigne l'algèbre des *séries formelles à variables non commutatives dans X et à coefficients dans*

A. L'algèbre des *séries formelles à variables commutatives dans X et à coefficients dans A* sera notée $A[[X]]$. Dans la pratique, A sera $\mathbb{Z}$, $\mathbb{Q}$, ou $\mathbb{C}$.

Pour chaque k-uplet $\mathbf{n} = (n_1, \dots, n_k) \in \mathbb{N}^k$, nous notons $\mathbf{x^n}$ le monôme $x_1^{n_1} \dots x_k^{n_k}$.

A toute série formelle à variables non commutatives correspond naturellement une série à variables commutatives, obtenue par le morphisme χ_0 qui "fait commuter" les lettres.

Un langage L peut être identifié à la série formelle en variables non commutatives $\sum_{w \in L} w$ (les coefficients sont 1 pour les mots de L et 0 pour les autres). La série $\chi_0(L)$ est alors appelée *série génératrice* du langage L; le coefficient de $\mathbf{x^n}$ est le nombre de mots de L qui ont, pour $1 \leq i \leq k$, exactement n_i occurrences de la lettre x_i. Nous utiliserons fréquemment la même lettre pour désigner un langage et sa série génératrice.

De manière plus générale, soit L un ensemble dénombrable d'"objets" combinatoires, et soient $p_1, \dots, p_k$ des *paramètres* positifs ou nuls définis sur L (chaque p_i est une application de L dans $\mathbb{N}$). La *série génératrice de L suivant les paramètres* $(p_i)_{1 \leq i \leq k}$ sera alors la série formelle à variables commutatives

$$L(x_1, \dots, x_k) = \sum_{w \in L} x_1^{p_1(w)} \cdots x_k^{p_k(w)}$$

Pour qu'une telle série soit définie, il faut que, pour chaque k-uplet $(v_1, \dots, v_k)$, l'ensemble des objets pour lesquels chaque paramètre p_i vaut v_i soit un ensemble fini.

Définition 1.11. Un paramètre λ, défini sur un ensemble d'objets A, est une *taille* si, pour chaque valeur de n, l'ensemble

$$A_n = \{a \in A, \lambda(a) = n\}$$

est fini.

L'existence d'une taille parmi les paramètres $p_1, \dots, p_k$, est une condition suffisante pour que la série génératrice suivant ces paramètres soit définie. Le paramètre "longueur totale" est toujours une taille pour les mots d'un langage, les alphabets étant toujours finis.

La série génératrice d'un langage suivant les paramètres "nombres d'occurrences des différentes lettres" est la série $\chi_0(L)$, que nous avons appelée plus haut sa série génératrice. Il est possible qu'aucun paramètre "nombre d'occurrences de la lettre x_i" ne soit une taille : il peut parfaitement y avoir, dans le langage considéré, une infinité de mots ne comportant pas la lettre x_i, et ce, pour chaque x_i. Cependant, la série génératrice est toujours bien définie, le nombre de mots du langage ayant une composition fixée étant fini.

Il arrivera fréquemment, dans le cours de cette thèse, que nous manipulions des combinaisons linéaires formelles de mots. Nous travaillons alors implicitement dans l'algèbre $\mathbb{Q} < X >$ des polynômes à variables non commutatives dans X et à coefficients dans $\mathbb{Q}$. Le plus souvent, une application $\varphi : X \to \mathbb{Q} < Y >$ sera définie, et implicitement étendue à $\mathbb{Q} < X >$ comme morphisme d'algèbres : d'abord à X^* par concaténation, puis à $\mathbb{Q} < X >$ par combinaisons linéaires.

Exemple 1.12. Soient $X = \{a, b\}$ et $Y = \{a, a', b\}$. Si φ est définie par

$$\begin{cases} \varphi(a) &= a + a' \\ \varphi(b) &= \epsilon + b, \end{cases}$$

alors on a, par exemple,

$$\begin{aligned} \varphi(aba + aa) &= \varphi(aba) + \varphi(aa) \\ &= \varphi(a)\varphi(b)\varphi(a) + \varphi(a)\varphi(a) \\ &= (a + a')(\epsilon + b)(a + a') + (a + a')(a + a') \\ &= 2aa + 2aa' + 2a'a + 2a'a' + aba + aba' + a'ba + a'ba'. \end{aligned}$$

1.4 Grammaires

Définition 1.13. Une *grammaire hors-contexte (context-free)* est un quadruplet $G = (X, N, \mathcal{R}, S)$, où :

- X est un ensemble fini appelé *alphabet*. Les éléments de X, que dans la mesure du possible nous noterons par des minuscules, sont appelés *lettres* (ou *symboles terminaux*).
- N est un ensemble fini disjoint de X, appelé *alphabet des symboles*; les éléments de N (normalement notés par des majuscules) sont appelés *symboles non terminaux*, ou *non terminaux*, ou *symboles*.
- $\mathcal{R}$ est une partie finie de $N \times (X \cup N)^*$, dont chaque élément est appelé *règle de dérivation* ou *transition*. Pour plus de clarté, les règles de dérivation seront présentées sous la forme $U \to W$, avec $U \in N$ et $W \in (X \cup N)^*$.
- S est un symbole non terminal appelé *axiome*.

Il pourra arriver que nous définissions une grammaire sans préciser son axiome, ou que nous changions cet axiome.

Pour chaque règle de dérivation $R = (U \to W)$, U est appelé *membre gauche* de R, et noté $g(R)$. Le *membre droit* de R, noté $d(R)$, est W. Le nombre de symboles non terminaux de W est appelé l'*arité* de la règle, et noté $\alpha(R)$. Une règle d'arité 0 est dite *terminale*. Enfin, $d(R, i)$ désigne le i-ème symbole du membre droit de R.

Nous écrirons fréquemment nos règles sous la forme $R : U \to w_0 U_1 w_1 \dots U_k w_k$; une telle notation suppose implicitement $w_i \in X^*$ et $U_i \in N$, de telle sorte que $d(R, i) = U_i$. Les symboles du membre droit seront systématiquement numérotés de gauche à droite.

Pour chaque symbole U, les règles de dérivation ayant U comme membre gauche sont appelées *U-dérivations*. L'ensemble des U-dérivations est noté $\mathcal{R}_U$.

Dans une grammaire, chaque règle de dérivation $R = (U \to V)$ définit sur $(X \cup N)^*$ une relation binaire $\overset{R}{\to}$ de la manière suivante : pour tous mots W_1 et W_2 de $(X \cup N)^*$, $W_1 U W_2 \overset{R}{\to} W_1 V W_2$; en d'autres termes, $W \overset{R}{\to} W'$ si l'on peut obtenir W' en remplaçant, dans W, une occurrence de U (le "membre gauche" de la règle de dérivation) par V (le "membre droit"). On dit alors que R *réécrit* ou *dérive* W en W'.

La réunion de toutes les relations $\overset{R}{\to}$ pour $R \in \mathcal{R}$ est notée $\overset{G}{\to}$, et la clôture transitive de $\overset{G}{\to}$ est notée $\overset{*}{\to}$. On a $W \overset{G}{\to} W'$ si et seulement si il existe une règle R qui réécrit W en W', et $W \overset{*}{\to} W'$ si et seulement si il existe une suite finie de règles qui, appliquées successivement, réécrivent W en W'.

Définition 1.14. Soit $G = (X, N, \mathcal{R}, S_0)$ une grammaire.

Le *langage engendré par G pour le symbole S* est l'ensemble $L_G(S)$ des mots $w \in X^*$ tels que $S \overset{*}{\to} w$.

Le *langage non terminal* engendré par G pour le symbole S est l'ensemble $\tilde{L}_G(S)$ des mots $w \in (X \cup V)^*$ tels que $S \overset{*}{\to} w$.

Lorsque le symbole n'est pas précisé, le langage engendré par une grammaire est celui qui est engendré pour l'axiome de la grammaire.

La notion de grammaire non ambiguë, ou grammaire algébrique, est fondamentale lorsqu'il s'agit d'obtenir des résultats d'énumération. Intuitivement, une grammaire est non ambiguë si elle n'engendre chaque mot que d'une seule manière, à certaines permutations inévitables près.

Définition 1.15. Une grammaire $G = (X, N, \mathcal{R}, S)$ est dite *non ambiguë* si, pour chaque symbole $U \in N$ et chaque mot w tel que $U \overset{G}{\to} w$, il existe une *unique* suite finie $(W_i)_{0 \le i \le n}$ de mots de $(X \cup N)^*$ telle que $U = W_0$, $W_n = w$, et, pour chaque $i \le n - 1$, $W_i \overset{G}{\to} W_{i+1}$,

avec la condition que W_{i+1} est obtenu en réécrivant dans W_i le *premier* symbole non terminal.

La non-ambiguïté d'une grammaire s'exprime beaucoup plus simplement par le fait que chaque mot d'un langage engendré par une grammaire non ambiguë possède un unique arbre de dérivation; cette propriété est valable aussi bien avec la définition "classique" des arbres de dérivation qu'avec la convention que nous adoptons dans cette thèse (voir le paragraphe 1.4.1 plus loin). Cette unicité doit être comprise dans le sens suivant : si un même mot appartient à plus d'un langage engendré par la grammaire, il possède un unique arbre de dérivation pour chaque langage auquel il appartient.

Définition 1.16. Un langage L est *algébrique* s'il existe une grammaire non ambiguë G qui engendre L.

Il existe des langages engendrés par des grammaires hors-contexte, mais qui ne peuvent l'être par une grammaire non ambiguë (Parikh [66]). De tels langages sont dits ambigus. L'ambiguïté d'un langage hors-contexte est un problème indécidable; on trouvera un éventail de méthodes permettant dans certains cas de démontrer cette ambiguïté dans Flajolet [40].

Définition 1.17. Soit G une grammaire, et soient S_1 et S_2 deux symboles non terminaux de G. On dira que S_2 est *accessible* à partir de S_1 s'il existe deux mots $W_1, W_2 \in (X \cup N)^*$ tels que $S_1 \stackrel{*}{\to} W_1 S_2 W_2$, avec $W_1 W_2 \neq \epsilon$.

On dira que S_1 et S_2 sont *simultanément accessibles* à partir de S s'il existe trois mots $W_1, W_2, W_3 \in (X \cup N)^*$ tels que $S \stackrel{*}{\to} W_1 S_1 W_2 S_2 W_3$ ou $S \stackrel{*}{\to} W_1 S_2 W_2 S_1 W_3$. Dans le cas où $S_1 = S_2$, il est indispensable que le symbole S_1 apparaisse deux fois.

La relation "accessible à partir de" est transitive. Les grammaires que nous manipulerons seront *propres*, ce qui signifie que tout symbole autre que l'axiome doit être accessible à partir de celui-ci (voir [1]). En effet, on ne change pas le langage engendré par une grammaire en éliminant les symboles non accessibles à partir de l'axiome, ainsi que toutes les règles de dérivation dont le membre gauche ou droit comporte l'un des symboles éliminés.

Etant donnée une grammaire, si l'on change l'axiome pour le remplacer par un autre des symboles non terminaux, on change généralement le langage engendré. Les langages $L_G(U)$, où U est un symbole autre que l'axiome, sont appelés *langages auxiliaires.*

Exemple 1.18 (Langage de Dyck). Un exemple fondamental de langage algébrique est le langage de Dyck, qui reviendra fréquemment dans nos exemples. Le langage de Dyck est

celui des systèmes correctement formés de parenthèses (Comtet [22]). En remplaçant par la lettre a la parenthèse ouvrante, et par b la parenthèse fermante, un mot $w \in \{a,b\}^*$ est un *mot de Dyck* s'il vérifie les conditions suivantes :

- $|w|_a = |w|_b$;
- pour tout facteur gauche w' de w, $|w'|_a \geq |w'|_b$.

Le langage de Dyck est engendré par la grammaire $G = (\{a,b\}, \{D\}, \mathcal{R}, D)$, où $\mathcal{R}$ contient les deux règles de dérivation

$$\begin{cases} R_1 : & D & \to & \epsilon, \\ R_2 : & D & \to & aDbD. \end{cases}$$

Le nombre de mots de Dyck de longueur $2n$ est le n-ième *nombre de Catalan* $C_n = \frac{1}{n+1}\binom{2n}{n}$. Ce résultat bien connu peut être montré de multiple façons : grâce à la formule d'inversion de Lagrange [81, 54] en utilisant l'équation sur la série génératrice $D(x)$ fournie par la grammaire ($D(x) = 1 + xD(x)^2$); en extrayant directement les coefficients de Taylor de la série génératrice obtenue explicitement en résolvant cette équation. Il en existe également de nombreuses preuves combinatoires; voir André [2], Bertrand [12] pour les plus anciennes. Parmi les méthodes permettant d'obtenir de telles preuves, le *principe de réflexion* d'André et le *principe de Raney* (Raney [70]), basé sur la conjugaison de mots, sont les plus universels.

1.4.1 Arbres de dérivation

Nous utiliserons fréquemment la notion d'*arbre de dérivation* d'un mot dans une grammaire. L'arbre de dérivation d'un mot est la représentation arborescente de la suite de transitions qui permettent de passer d'un symbole non terminal à un mot du langage engendré par ce symbole.

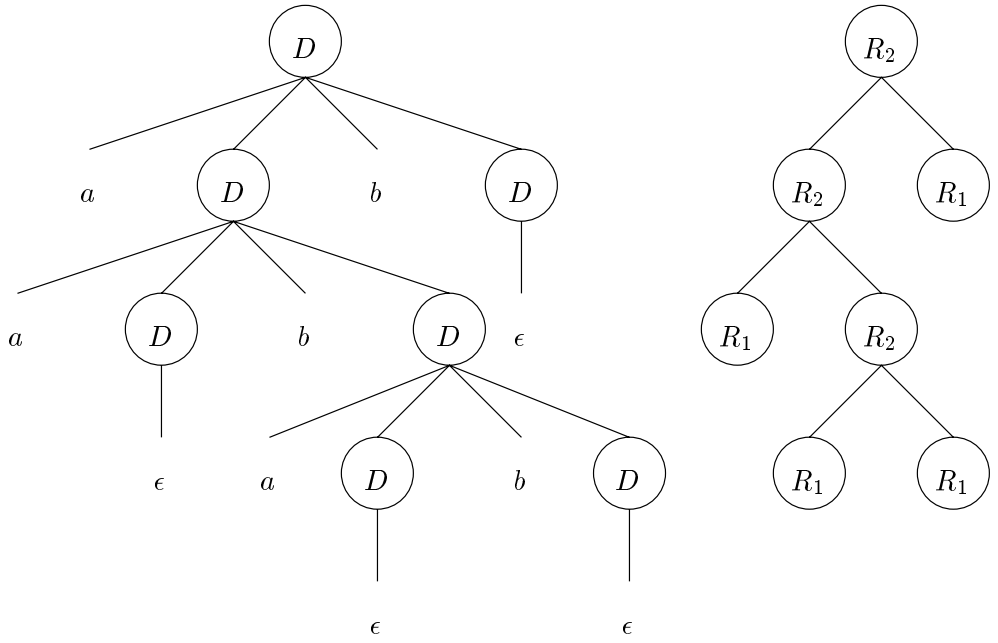
Dans le sens usuel, l'arbre de dérivation d'un mot comporte un nœud interne par transition. Les nœuds internes sont alors étiquetés par des symboles non terminaux, et les feuilles par des mots (éventuellement vides) formés de lettres terminales. Le mot représenté par un arbre de dérivation donné est alors obtenu en lisant les étiquettes des feuilles dans l'ordre symétrique.

Dans notre travail, nous avons besoin de repérer quelle règle de dérivation a été utilisée à chaque étape de la formation du mot. C'est pourquoi les nœuds de nos arbres de dérivation sont étiquetés par les "noms" des règles de dérivation utilisées (plutôt que par les symboles non terminaux qui en forment les membres gauches). La connaissance des règles

de dérivation rendant alors superflues les feuilles et leurs étiquettes, nous les omettons systématiquement. La taille d'un arbre de dérivation est alors exactement la longueur de la suite de transitions qui fait passer du symbole non terminal au mot produit.

La figure 1.2 montre les deux formes possibles d'arbres de dérivation pour le mot $w = aababb$ dans la grammaire classique engendrant le langage de Dyck :

$$\begin{aligned} R_1 : &\quad D \rightarrow \epsilon \\ R_2 : &\quad D \rightarrow aDbD. \end{aligned}$$



(a) Arbre de dérivation classique (b) Arbre de dérivation condensé

FIG. 1.2: *Exemples d'arbres de dérivation*

Formellement, nos arbres de dérivation sont donc définis récursivement de la manière suivante :

Définition 1.19. Soit U un symbole d'une grammaire G, et $w \in L_G(U)$ un mot. Les arbres de dérivation de w dans le langage $L_G(U)$ sont définis ainsi :

- Si $R : U \rightarrow w$ est une des règles de dérivation de la grammaire, l'arbre réduit à une racine étiquetée R est un arbre de dérivation de w;
- si $R : U \rightarrow u_0U_1u_1 \dots U_ku_k$ est une des règles de dérivation de la grammaire, si $w = u_0w_1u_1 \dots w_ku_k$, avec pour $1 \leq i \leq k$, $w_i \in L_G(U_i)$, et si pour $1 \leq i \leq k$, T_i est

un arbre de dérivation de w_i dans le langage $L_G(U_i)$, alors l'arbre $T = (R; T_1, \ldots, T_k)$ est un arbre de dérivation de w dans le langage $L_G(U)$.

La définition donnée plus haut d'une grammaire non ambiguë est équivalente à la suivante :

Définition 1.20. Une grammaire G est non ambiguë si et seulement si, pour chaque symbole U et chaque mot $w \in L_G(U)$, w a un unique arbre de dérivation dans le langage $L_G(U)$.

En termes d'arbres de dérivation, un symbole U_2 est accessible à partir d'un symbole U_1 s'il existe une arbre de dérivation de la grammaire, dont la racine est étiquetée par une U_1-dérivation, et qui contient un sommet étiqueté par une U_1-dérivation (avec la condition supplémentaire que ce sommet doit être distinct de la racine, si $U_2 = U_1$). De même, U_2 et U_1 sont simultanément accessibles à partir de U, s'il existe un arbre de dérivation dont la racine est étiquetée par une U-dérivation, et qui contient deux sommets, étiquetés respectivement par une U_1-dérivation et une U_2-dérivation, qui ne sont pas sur la même branche (l'un ne doit pas être un descendant de l'autre).

Lorsque $\mathcal{A}_1$ et $\mathcal{A}_2$ sont deux arbres de dérivation d'une même grammaire, et que s est un sommet de $\mathcal{A}_1$, l'arbre $\mathcal{A}_1(s, \mathcal{A}_2)$ est un arbre de dérivation à condition que les étiquettes de s et de la racine de $\mathcal{A}_2$ aient le même membre gauche; nous nous interdirons d'effectuer des substitutions dans les arbres de dérivation lorsque cette condition ne sera pas remplie.

1.4.2 Grammaires attribuées

Les grammaires attribuées sont classiquement utilisées pour la construction de compilateurs. Nous ne nous intéressons ici qu'à leur utilisation dans le domaine de la combinatoire, et nous nous limitons à des attributs *synthétisés* tels qu'ils sont définis dans l'article de Knuth [61].

Définition 1.21. Soit $G = (X, N, \mathcal{R}, S)$ une grammaire. Une *famille d'attributs* définie sur G est la donnée, pour chaque symbole $U \in N$, d'un ensemble fini T_U d'attributs ayant les caractéristiques suivantes :

- chaque attribut $\tau \in T_U$ possède un *domaine de valeurs* D_τ, qui est un ensemble (fini ou non); le produit cartésien $\prod_{\tau \in T_U} D_\tau$ est noté $\mathcal{D}_U$;

– pour chaque attribut $\tau \in T_U$ et chaque U-dérivation $R : U \to w_0 U_1 \dots U_k w_k$, on donne une *règle de calcul* $f_{\tau,R}$, qui est une fonction définie sur $\mathcal{D}_{U_1} \times \dots \times \mathcal{D}_{U_k}$ et à valeurs dans D_τ. Si la règle R est d'arité 0, $f_{\tau,R}$ est simplement un élément de D_τ.

Le couple $\left(G, (T_U)_{U \in N}\right)$ est alors appelé *grammaire attribuée.*

Une grammaire attribuée permet de calculer récursivement, pour chaque mot w de chaque langage $L_G(U)$ engendré par G, une valeur $\tau(w) \in D_\tau$, et ce, pour chaque attribut défini sur U. Ce calcul est plus simple à lire sur un arbre de dérivation $\mathcal{A}$: à chaque sommet s de l'arbre, étiqueté par une U-dérivation, on donne une étiquette supplémentaire $\tau(s)$ pour chaque $\tau \in T_U$, cette étiquette étant calculée en fonction des différentes étiquettes des fils de s : si le sommet s est étiqueté par la règle R, $\tau(s)$ est calculé en appliquant la "règle de calcul" $f_{\tau,R}$ à l'ensemble des valeurs calculées pour les fils du sommet s. De proche en proche, on obtient ainsi les attributs de la racine $(\tau(s_0))_{\tau \in T_U}$, qui sont les valeurs des attributs pour le *mot* w dont $\mathcal{A}$ est l'arbre de dérivation.

Exemple 1.22. Considérons la grammaire $G = (\{a, b\}, \{D\}, \mathcal{R}, D)$, dont les règles de dérivation sont :

$$\begin{cases} R : & D & \to & \epsilon, \\ R' : & D & \to & aDbD. \end{cases}$$

Les arbres de dérivation de G sont exactement les *arbres binaires complets* (les sommets internes, étiquetés R', ont 2 fils, et les feuilles sont étiquetées R).

Le *nombre de Strahler* $St(\mathcal{A})$ d'un arbre binaire complet $\mathcal{A}$ est défini récursivement de la manière suivante :

– $St(\mathcal{A}) = 0$ si $\mathcal{A}$ se réduit à sa racine;

– si $\mathcal{A}$ a pour sous-arbres gauche et droit $\mathcal{A}_g$ et $\mathcal{A}_d$ respectivement, $St(\mathcal{A})$ est le *maximum* de $St(\mathcal{A}_g)$ et $St(\mathcal{A}_d)$, auquel on ajoute 1 si $St(\mathcal{A}_g) = St(\mathcal{A}_d)$.

Le nombre de Strahler peut par conséquent être défini par un attribut St, avec les règles de calcul :

$$\begin{cases} f_{St,R} & = & 0 \\ f_{St,R'}(n,m) & = & \begin{cases} n+1 \text{ si } n = m \\ \max(n,m) \text{ si } n \neq m \end{cases} \end{cases}$$

Le calcul du nombre de Strahler est illustré figure 1.3.

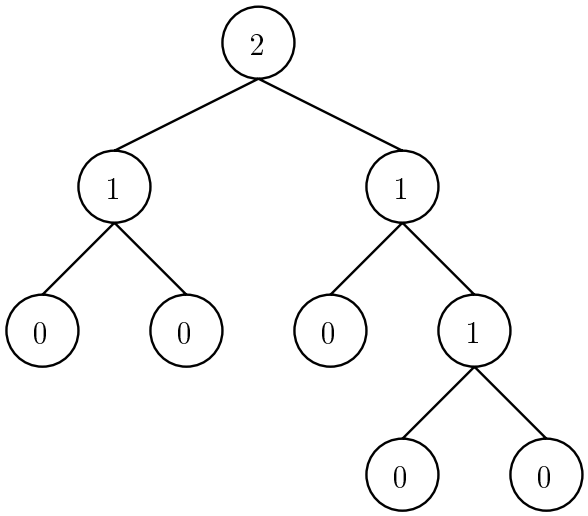


FIG. 1.3: *Exemple de calcul du nombre de Strahler*

Les q-grammaires, introduites par Fédou [37], sont un cas particulier de grammaires attribuées, où un attribut unique est défini sur chaque symbole, les règles de calcul étant astreintes à avoir une forme particulière. Nous en proposons une version plus générale, appelée Q-grammaires, au chapitre 2, et qui constituent l'objet principal d'étude de ce travail.

1.5 Chemins discrets

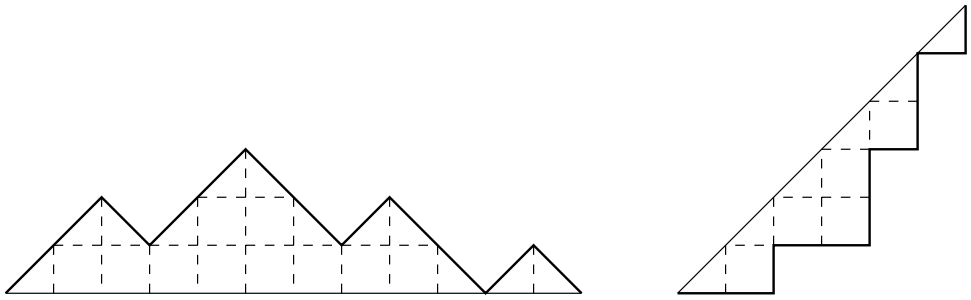
Dans notre travail, un *chemin* est une suite $(s_0, \ldots, s_n)$ de points du plan discret $\mathbb{Z} \times \mathbb{Z}$. Nous ne considérons les chemins qu'à translation près, aussi supposerons-nous le plus souvent que le premier sommet s_0 est l'origine $(0, 0)$.

Un *pas* d'un chemin est le vecteur reliant deux sommets consécutifs, de coordonnées $(x_{i+1} - x_i, y_{i+1} - y_i)$. Il est fréquent d'utiliser les points cardinaux pour désigner certains types de pas : un pas $(0, 1)$ est ainsi un pas Nord, un pas $(1, -1)$ est un pas Sud-Est, etc.

Il est commode de représenter certains mots par des chemins, le plus simple étant alors d'associer à chaque lettre de l'alphabet un type de pas, ceux-ci se succédant dans le même ordre que dans le mot.

Les représentations les plus courantes des mots de Dyck sont celles que nous appellerons représentations diagonale et horizontale.

La représentation *diagonale* consiste à traduire la lettre a par un pas Est et la lettre b par un pas Nord. Le chemin se termine alors sur la droite d'équation $y = x$, sans avoir de sommet au-dessus de cette droite. Dans la représentation *horizontale*, la lettre a se traduit



Représentation horizontale Représentation diagonale

FIG. 1.4: *Chemins de Dyck associés à* $w = aabaabbabbab$

par un pas Nord-Est, et la lettre b se traduit par un pas Sud-Est. Le chemin se termine alors sur la droite d'équation $y = 0$, sans avoir de sommet en-dessous de cette droite.

Dans les deux cas, le chemin obtenu est appelé chemin de Dyck. Ces deux représentations sont illustrées figure 1.4.

Le passage de l'une à l'autre de ces représentations se fait très simplement par un changement d'échelle suivi d'une symétrie.

1.6 Deux exemples classiques

Deux exemples classiques de grammaires et de paramètres reviendront fréquemment au cours de cette thèse, tous deux basés sur le langage des mots de Dyck. La terminologie d'origine géométrique que nous emploierons, est basée sur la représentation horizontale des mots de Dyck par des chemins discrets allant du point $(0, 0)$ à un point $(2n, 0)$, où $2n$ est la longueur du mot.

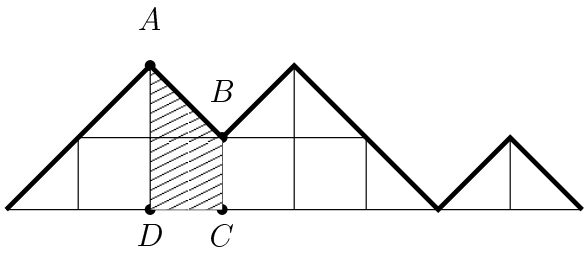
1.6.1 Aire des chemins de Dyck

Par définition, l'*aire* d'un chemin de Dyck est l'aire comprise entre ce chemin et l'axe d'équation $y = 0$. Par extension, nous parlerons également de l'aire d'un mot de Dyck. En découpant cette surface en tranches verticales délimitées par les pas du chemin, cette aire apparaît comme la somme d'aires de trapèzes : chaque pas AB du chemin, contribue à l'aire pour $(y_A + y_B)/2$.

Lorsque le pas AB est un pas Nord-Est, $y_B = y_A + 1$, et par conséquent, si $h = y_B$ est l'ordonnée du point B (également appelée *hauteur finale* du pas AB), la contribution à l'aire du pas AB est $h - 1/2$.

Inversement, si le pas AB est un pas Sud-Est, $y_B = y_A + 1$, et la contribution du pas AB est $h + 1/2$.

Lors de la sommation, les termes $+1/2$ et $-1/2$ associés aux pas Nord-Est et Sud-Est se compensent, puisqu'un chemin de Dyck comporte autant de pas des deux types; par conséquent *l'aire d'un chemin de Dyck* est aussi égale à *la somme des hauteurs finales des pas qui le constituent.*



$$1/2 + 3/2 + 3/2 + 3/2 + 3/2 + 1/2 + 1/2 + 1/2 \quad = \quad 8$$

$$1 \; + \; 2 \; + \; 1 \; + \; 2 \; + \; 1 \; + \; 0 \; + \; 1 \; + \; 0 \quad = \quad 8$$

$$2(\quad 1 \quad + \quad 1 \; + \; 0 \quad + \quad 0 \quad) + 4 = \; 8$$

FIG. 1.5: *Différents modes de calcul de l'aire d'un chemin de Dyck*

Par ailleurs, les pas Nord-Est et Sud-Est peuvent être naturellement appariés de manière à ce qu'à chaque pas Nord-Est de hauteur finale h, corresponde un pas Sud-Est de hauteur finale $h - 1$ (ce qui, en interprétant le mot de Dyck comme un mot de parenthèses ouvrantes et fermantes, correspond aux paires de parenthèses). Par conséquent, l'aire d'un chemin de Dyck est également obtenue en faisant la somme de sa *demi-longueur* et de *deux fois la somme des hauteurs finales de ses pas Sud-Est.*

Tous ces moyens de calcul de l'aire d'un chemin de Dyck sont illustrés figure 1.5.

Tous ces paramètres se "lisent" parfaitement sur les mots de Dyck correspondant aux chemins. En effet, si $w = w_1 w_2$ est un mot de Dyck, la hauteur finale du chemin correspondant au facteur gauche w_1 est $h = |w_1|_a - |w_1|_b$; il est donc simple de transformer les relations donnant l'aire d'un chemin de Dyck, en expressions portant sur les mots de Dyck associés.

Le paramètre *aire* des mots de Dyck est lié à un autre paramètre classique: le *nombre d'inversions* [32]. Nous considérerons l'ordre $a < b$, et par conséquent une inversion dans un mot de Dyck sera une occurrence du sous-mot ba. Le nombre d'inversions d'un tel mot

w peut donc être décrit de plusieurs façons :

- la somme, pour chaque occurrence de a dans w, du nombre d'occurrences de b qui se trouvent avant elle dans w;

- la somme, pour chaque occurrence de b dans w, du nombre d'occurrences de a qui se trouvent après elle dans w.

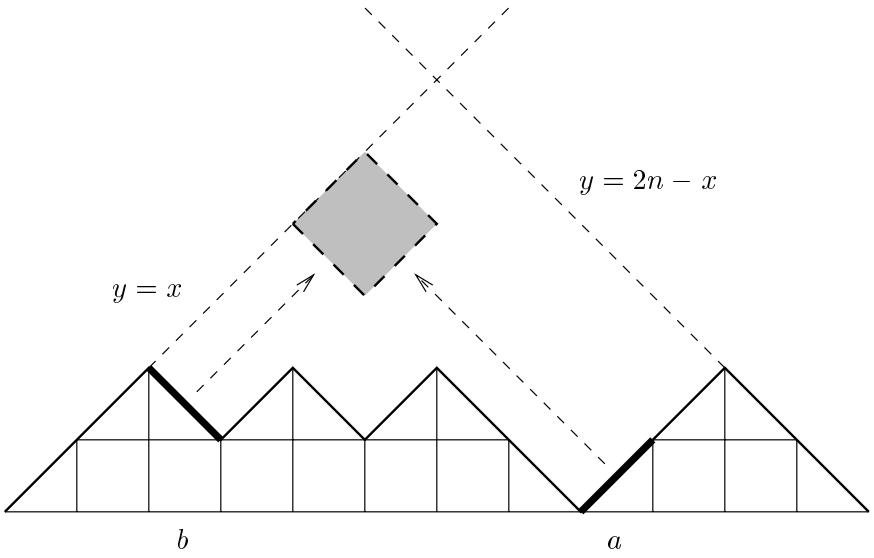


FIG. 1.6: *Interprétation géométrique du nombre d'inversions*

La figure 1.6 montre comment le nombre d'inversions d'un mot de Dyck peut être interprété géométriquement comme la moitié de l'aire comprise *au-dessus* du chemin de Dyck correspondant, et *en-dessous* des droites diagonales d'équations $y = x$ et $y = 2n - x$ (où $2n$ est la longueur du mot) : à chaque inversion du mot, correspond exactement un carré d'aire 2 situé dans cette zone.

Une autre définition, très proche, de l'aire associée à un mot de Dyck, est celle qui correspond aux q-analogues des nombres de Catalan étudiés par Carlitz. Dans ce cas, on considère des chemins de Dyck sous-diagonaux, et l'aire d'un mot est celle de la zone située entre le chemin sous-diagonal et la diagonale d'équation $y = x$. Cette aire n'étant pas entière pour les chemins de longueur $2n$ lorsque n est impair, on ne considère que la différence avec l'aire associée au mot $(ab)^n$, soit $n/2$.

Le passage d'une représentation géométrique à l'autre se fait par une symétrie suivie d'une homothétie de rapport $\sqrt{2}$, par conséquent, si l'aire d'un mot de Dyck w de longueur

$2n$ est notée $A(w)$, et l'aire de Carlitz, $A_c(w)$, on a

$$A(w) = 2A_c(w) + n.$$

L'aire de Carlitz est plus directement liée au nombre d'inversions; on a, pour un mot de longueur $2n$,

$$A_c(w) + \mathrm{inv}(w) = \binom{n}{2}.$$

Les deux définitions de l'aire sont illustrées figure 1.7.

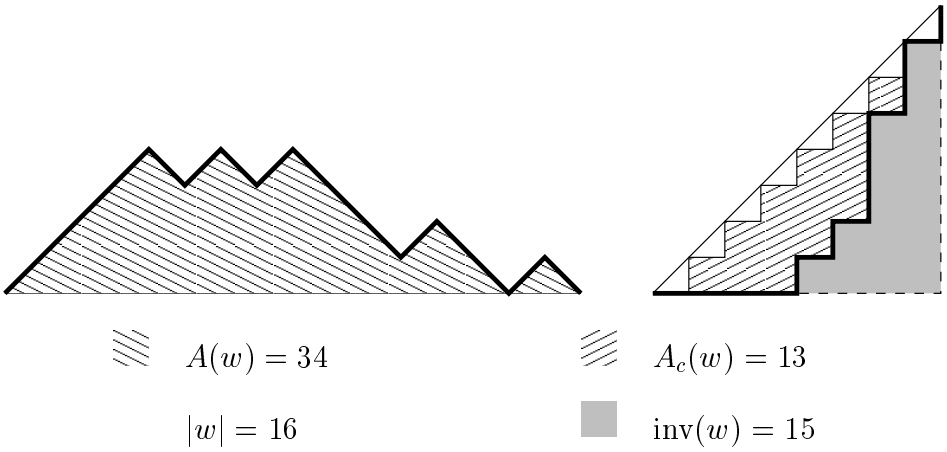


FIG. 1.7: *Les deux définitions de l'aire pour* $w = aaaababababbbabbab$

Les q-analogues des nombres de Catalan définis par Carlitz énumèrent les mots de Dyck suivant le nombre d'inversions ou l'aire de Carlitz :

$$\begin{aligned} C_n(q) &= \sum_{w \in D_n} q^{\mathrm{inv}(w)}, \\ \tilde{C}_n(q) &= \sum_{w \in D_n} q^{A_c(w)}. \end{aligned}$$

Ces polynômes vérifient les récurrences suivantes, qui sont deux q-analogues de la récurrence classique des nombres de Catalan :

$$\begin{aligned} \tilde{C}_0 &= 1, \\ \tilde{C}_{n+1} &= \sum_{0 \le k \le n} q^k \tilde{C}_k \tilde{C}_{n-k}; \\ C_0 &= 1, \\ C_{n+1} &= \sum_{0 \le k \le n} q^{(k+1)(n-k)} C_k C_{n-k}. \end{aligned}$$

Nous mentionnons pour mémoire deux résutats d'énumération concernant l'aire des chemins de Dyck :

- Soit, pour chaque mot de Dyck de longueur $2n$, $A'(w) = A(w) + 2n + 1$; $A'(w)$ s'interprète comme le nombre de points du plan discrets situés (au sens large) entre le chemin de Dyck et l'axe horizontal. Alors

$$\sum_w A'(w) = 4^n,$$

  où la sommation porte sur tous les mots de Dyck de longueur $2n$. On trouvera une preuve bijective de cette formule dans [20].

- La somme des aires de Carlitz des chemins de Dyck de longueur $2n$,

$$\sum_w A_c(w) = \frac{1}{2}\left(4^n - (3n+1)C_n\right),$$

  est également le nombre de cartes planaires pointées sans isthmes à 2 sommets et n arêtes (alors que le nombre de cartes planaires pointées à 1 sommet et n arêtes est le nombre de Catalan C_n). La suite de nombres $(1, 7, 37, 176\dots)$ apparaît dans [80]; un codage de ces cartes par des mots de Dyck marqués est donné dans [63].

1.6.2 Somme des hauteurs de pics de chemins de Dyck

Dans un chemin de Dyck, un *pic* est un sommet immédiatement précédé d'un pas Nord-Est, et immédiatement suivi d'un pas Sud-Est.

Inversement, un *creux* d'un chemin de Dyck est un sommet immédiatement précédé d'un pas Sud-Est, et immédiatement suivi d'un pas Nord-Est.

Dans le mot de Dyck correspondant, un pic correspond à un facteur ab, et un creux, à un facteur ba. La *hauteur* d'un pic ou d'un creux est l'ordonnée du sommet correspondant. Comme pour le calcul de l'aire, la hauteur d'un pic se calcule simplement sur le mot de Dyck correspondant. La hauteur du pic situé entre les facteurs w_1a et bw_2 dans $w = w_1abw_2$, est $h = 1 + |w_1|_a - |w_1|_b$.

Un mot de Dyck peut parfaitement être défini par la suite de ses hauteurs de pics et de creux; cette remarque est à la base d'un codage des polyominos parallélogrammes par les mots de Dyck [25], dans lequel la hauteur de chaque pic devient la hauteur d'une colonne du polyomino, et, par conséquent, la somme des hauteurs de pics devient l'aire du polyomino codé. La longueur du mot de Dyck devient, à un décalage près, le périmètre du polyomino codé.

Fédou a montré dans [37], que la somme des hauteurs des pics de mots de Dyck a la même distribution que le paramètre *somme des nombres de feuilles des sous-arbres gauches* défini sur les arbres binaires complets; nous verrons au chapitre 2 que ce genre d'interprétation peut être automatisé.

$w = aabaaabbababbbab$

Longueur 16

5 pics de hauteurs 2, 4, 3, 3, 1

Périmètre 18

5 colonnes

de hauteurs 2, 4, 3, 3, 1.

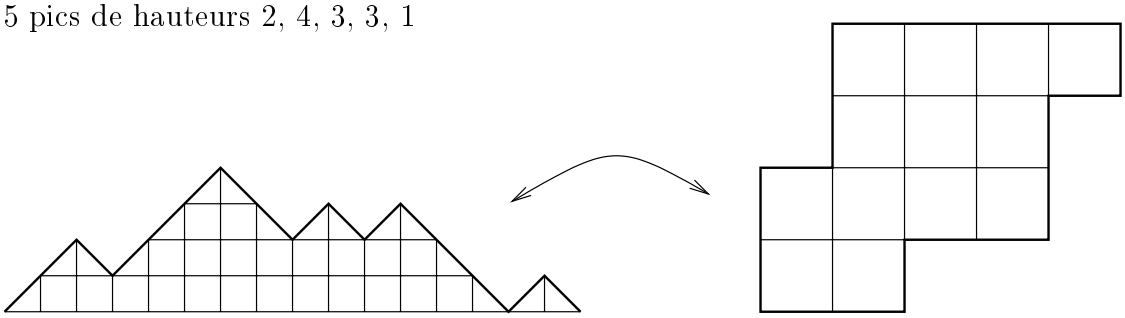


FIG. 1.8: *Somme des hauteurs de pics et polyominos parallélogrammes*

La figure (1.8) montre un exemple de mot de Dyck de somme des hauteurs de pics 13, et le polyomino parallélogramme correspondant.

Chapitre 2

Q-grammaires

Dans ce chapitre, nous définissons les objets principalement étudiés dans cette thèse : les Q-grammaires, qui sont des grammaires attribuées dont les attributs ont une forme particulière.

2.1 Notations

Dans ce chapitre, chaque fois qu'une grammaire $G = (X, N, \mathcal{R}, S)$ aura été définie et qu'il n'y aura pas d'ambiguïté possible, nous identifierons un symbole et le langage qu'il engendre, c'est-à-dire que nous écrirons U pour $L_G(U)$.

Une règle de dérivation générique d'une grammaire sera notée sous la forme

$$R : U \to w_0 U_1 w_1 \dots w_{n-1} U_n w_n,$$

avec $w_i \in X^*$ et $U_i \in N$.

Dans cette notation, R est le nom de la règle de dérivation, utilisé pour étiqueter les arbres de dérivation; n est l'arité de la règle R.

Un mot $u \in L_G(U)$ est obtenu par l'utilisation de la règle R lorsque la racine de son arbre de dérivation est étiquetée par R. Cela revient à dire qu'il existe des mots $u_i \in U_i$ pour $1 \leq i \leq n$, tels que

$$u = w_0 u_1 w_1 \dots w_{n-1} u_n w_n.$$

Les grammaires considérées étant non ambiguës, pour chaque symbole U et chaque mot $u \in U$, il n'y a qu'une seule règle dont est issu u; nous noterons alors sans plus de précision u_i le i-ème mot apparaissant dans la décomposition ci-dessus. Ce mot u_i est celui qu'engendre le i-ème sous-arbre de l'arbre de dérivation de u.

2.2 Paramètres Q-comptables

2.2.1 Termes de croissance d'un paramètre

Soit $G = (X, N, \mathcal{R}, S)$ une grammaire non ambiguë engendrant un langage L, et soit p un paramètre défini sur la réunion disjointe[1] des langages engendrés par G (même si p n'est défini a priori que sur le langage L, nous supposerons qu'un prolongement à cette réunion disjointe a été choisi).

Définition 2.1. Soit u un mot d'un langage engendré par G.

Avec les notations définies précédemment, on appelle *terme de croissance de p pour u*, la quantité

$$\theta_p(u) = p(u) - \sum_{i=1}^{n} p(u_i).$$

Lorsque le mot u se trouve être le membre droit d'une U-dérivation d'arité 0, le terme de croissance de p pour u (dans le langage U) est simplement $\theta_p(u) = p(u)$.

En termes d'arbres de dérivation, et si l'on considère p comme un attribut, $\theta_p(u)$ est la différence entre l'attribut calculé sur l'arbre tout entier, et la somme des valeurs attribuées aux sous-arbres issus des fils de la racine; c'est pourquoi nous l'appelons terme de croissance. Il est parfaitement possible, pour un attribut quelconque, qu'il n'y ait aucun lien entre $p(u)$ et la somme des $p(u_i)$. Dans ce cas, le terme de croissance n'a aucun sens particulier.

Exemple 2.2. Considérons le langage des mots de Dyck, engendré par la grammaire classique G_1 correspondant aux deux règles de dérivation

$$\begin{cases} R_1 : & D & \to & \epsilon \\ R_2 : & D & \to & aDbD. \end{cases}$$

Notons, pour tout mot $w \in \{a, b\}^*$, $h(w) = |w|_a - |w|_b$, et soit

$$p(w) = \max_{w=w_1w_2} h(w_1).$$

Le paramètre p représente l'ordonnée maximale atteinte par le chemin de Dyck associé au mot w; il est aussi appelé hauteur maximale de w. Il est clair que le paramètre p peut

1. Dans le cas où un même mot appartient à plus d'un des langages engendrés par une grammaire, nous considérerons qu'un même paramètre peut prendre, pour ce mot, une valeur différente dans chaque langage.

être calculé comme attribut synthétisé sur la grammaire G_1 : $p(\epsilon) = 0$, et, si $w = aw_1bw_2$ est obtenu par la règle R_2, on a

$$p(w) = \begin{cases} 1 + p(w_1) & \text{si } p(w_1) \geq p(w_2), \\ p(w_2) & \text{si } p(w_2) > p(w_1). \end{cases}$$

Dans ce cas, la somme $p(w_1) + p(w_2)$ n'apparaît pas dans le calcul de $p(w)$: le terme de croissance n'a donc ici pas de sens naturel.

Un exemple similaire est celui du *nombre de Strahler* vu au chapitre 1. De manière générale, on peut s'attendre à ce que le terme de croissance d'un paramètre défini comme un *maximum* n'ait pas de sens "naturel".

Notre travail a pour cadre le cas inverse, où chaque paramètre étudié peut être "simplement" défini au moyen de ses termes de croissance.

Exemple 2.3. Considérons de nouveau le langage de Dyck, engendré par les deux grammaires G_1 (voir exemple 2.2) et $G_2 = (\{a, b\}, \{D, E\}, \{R'_1, R'_2, R'_3, R'_4, R'_5, R'_6\}, D)$:

$$\begin{cases} R'_1 : & D & \to & \epsilon \\ R'_2 : & D & \to & E \\ R'_3 : & E & \to & ab \\ R'_4 : & E & \to & abE \\ R'_5 : & E & \to & aEb \\ R'_6 : & E & \to & aEbE. \end{cases}$$

Il est facile de voir que, dans la grammaire G_2, seules les règles R'_3 et R'_4 font apparaître des facteurs ab (les pics des chemins de Dyck correspondants); par conséquent, dans G_2, le terme de croissance du paramètre "nombre de pics" p' est 1 pour les règles R'_3 et R'_4, et 0 pour les autres règles.

En revanche, dans G_1, la situation n'est pas aussi simple. Le terme de croissance du même paramètre p' pour la règle R_2 est toujours positif ou nul; si le mot $w_1 \in D$ a k_1 facteurs ab et le mot $w_2 \in D$ en a k_2, le mot $aw_1bw_2 \in D$ en a au moins $k_1 + k_2$. En fait, on a $p'(aw_1bw_2) = p'(w_1) + p'(w_2)$ si $w_1 \neq \epsilon$, et $p'(aw_1bw_2) = p'(w_1) + p'(w_2) + 1 = p'(w_2) + 1$ si $w_1 = \epsilon$.

Nous parlerons également de terme de croissance d'un paramètre pour un arbre de dérivation, ou pour un nœud d'un arbre de dérivation.

Ainsi, pour un arbre de dérivation $\mathcal{A}$, le terme de croissance du paramètre p pour $\mathcal{A}$ sera par définition $\theta_p(\mathcal{A}) = \theta_p(u)$, où u est le mot dont $\mathcal{A}$ est l'arbre de dérivation.

De même, si s est un nœud d'un arbre de dérivation $\mathcal{A}$, le terme de croissance de p pour s est $\theta_p(s) = \theta_p(\mathcal{B})$, où $\mathcal{B}$ est le sous-arbre de $\mathcal{A}$ dont la racine est s.

2.2.2 Paramètres Q-comptables et Q-grammaires

Le principe de la formation d'une Q-grammaire consiste à définir tous les paramètres auxquels l'on s'intéresse par le moyen de leurs termes de croissance. Chaque terme de croissance ne doit dépendre que de la règle de dérivation utilisée et d'autres paramètres préalablement définis. Chaque paramètre est ainsi défini par un ensemble de "règles de calcul", une règle étant associée à chaque règle de dérivation de la grammaire. Les "bons" paramètres, que nous nommerons Q-comptables, sont ceux dont les termes de croissance sont eux-mêmes définis par d'autres paramètres Q-comptables :

Définition 2.4 (paramètre Q-comptable). Un paramètre p est dit :

- Q-comptable de rang 1, si, pour chaque règle de dérivation R de G, il existe une constante entière $c_R \geq 0$ telle que, pour tout mot u obtenu par l'application de la règle R, $\theta_p(u) = c_R$;
- Q-comptable de rang $k+1$, si, pour chaque règle de dérivation R de G, il existe des paramètres Q-comptables $p_1, \ldots p_n$, de rangs inférieurs ou égaux à k, et une constante $c_R \geq 0$, tels que, pour tout mot u obtenu par l'application de la règle R, on ait $\theta_p(u) = c_R + p_1(u_1) + \cdots + p_n(u_n)$.

Afin que notre définition soit consistante, nous exigeons que, pour un paramètre de rang 1, l'une au moins des constantes c_R soit non nulle, et que, pour un paramètre de rang $k+1$, l'un au moins des paramètres utilisés soit exactement de rang k.

Un "paramètre" constamment égal à un nombre entier positif est considéré comme Q-comptable de rang 0.

Un paramètre Q-comptable est donc un attribut synthétisé sur la grammaire G, avec des restrictions sur la forme des règles de calcul de cet attribut, qui doivent ne faire intervenir que des combinaisons affines d'autres paramètres Q-comptables.

Par la suite, nous ne nous intéresserons qu'aux paramètres Q-comptables d'une grammaire.

La définition ci-dessous généralise celle des q-grammaires définie dans [37, 28] :

Définition 2.5. On appelle Q-grammaire, une grammaire attribuée $(G; p_1, \ldots, p_n)$, où les attributs $p_1, \ldots, p_n$ sont des paramètres Q-comptables sur G et tels que, pour $1 \leq i \leq n$, les termes de croissance de p_i ne fassent intervenir que les paramètres $p_1, \ldots, p_{i-1}$.

La notion de paramètre Q-comptable est liée à la grammaire G plus qu'au langage qu'elle engendre (dans l'exemple 2.3, le paramètre "nombre de pics" est clairement Q-comptable dans la grammaire G_2, alors que nous verrons qu'il ne l'est pas dans la grammaire G_1). Certains paramètres sont Q-comptables pour toutes les grammaires.

Lorsqu'il y aura risque d'ambiguïté sur la grammaire, on précisera celle-ci en disant qu'un paramètre est G-Q-comptable.

Exemple 2.6. Dans toute grammaire, le paramètre "longueur" $|w|$ et, plus généralement, tout paramètre "longueur en X'" où $X' \subset X$, est Q-comptable de rang 1; en effet, le terme de croissance de chaque règle $R : U \to w_0 U_1 w_1 \dots U_n w_n$ est la constante $\theta_p = c_R = |w_0 w_1 \dots w_n|_{X'}$.

Exemple 2.7. Reprenons la grammaire G_1 de l'exemple 2.2 (grammaire classique engendrant le langage de Dyck).

Notons $A(w)$ l'aire d'un mot de Dyck quelconque w. Comme nous l'avons vu précédemment, $A(w)$ est aussi la somme des hauteurs de tous les sommets du chemin de Dyck correspondant à w.

L'aire des chemins de Dyck est Q-comptable de rang 2 pour cette grammaire, avec les termes de croissance ainsi définis :

- Pour la règle $D \to \epsilon$: $A(\epsilon) = 0$ (l'aire du chemin vide est nulle);
- Pour la règle $D \to aDbD$: $A(ad_1bd_2) = A(d_1) + A(d_2) + 1 + |d_1|$, donc le terme de croissance est $1 + |d_1|$. La figure 2.1 illustre cette règle de calcul.

Puisque la longueur est elle-même Q-comptable de rang 1, l'aire est alors Q-comptable de rang 2.

De même, l'aire de Carlitz est également Q-comptable de rang 2 dans la grammaire G_1. En effet, $A_c(ad_1bd_2) = A_c(d_1) + A_c(d_2) + |d_1|/2$, donc le terme de croissance est $\theta_{A_c}(ad_1bd_2) = |d_1|/2 = |d_1|_a$ qui est lui aussi un paramètre Q-comptable de rang 1.

Enfin, nous pouvons aller un cran plus loin et définir pour un mot de Dyck son *moment d'inertie* de la manière suivante : $M(w)$ est la somme des ordonnées des points à coordonnées entières positives situés (au sens large) sous le chemin codé par w. Ainsi, la contribution à $M(w)$ des points situés sous un sommet de hauteur h, est $1 + \dots + h = h(h+1)/2$.

Le moment d'inertie M est un paramètre Q-comptable de rang 3, et sa règle de calcul pour la règle $D \to aDbD$ peut être obtenue en examinant de nouveau la figure 2.1. Dans le chemin codé par $d = ad_1bd_2$, chacun des points à coordonnées entières situés sous le chemin

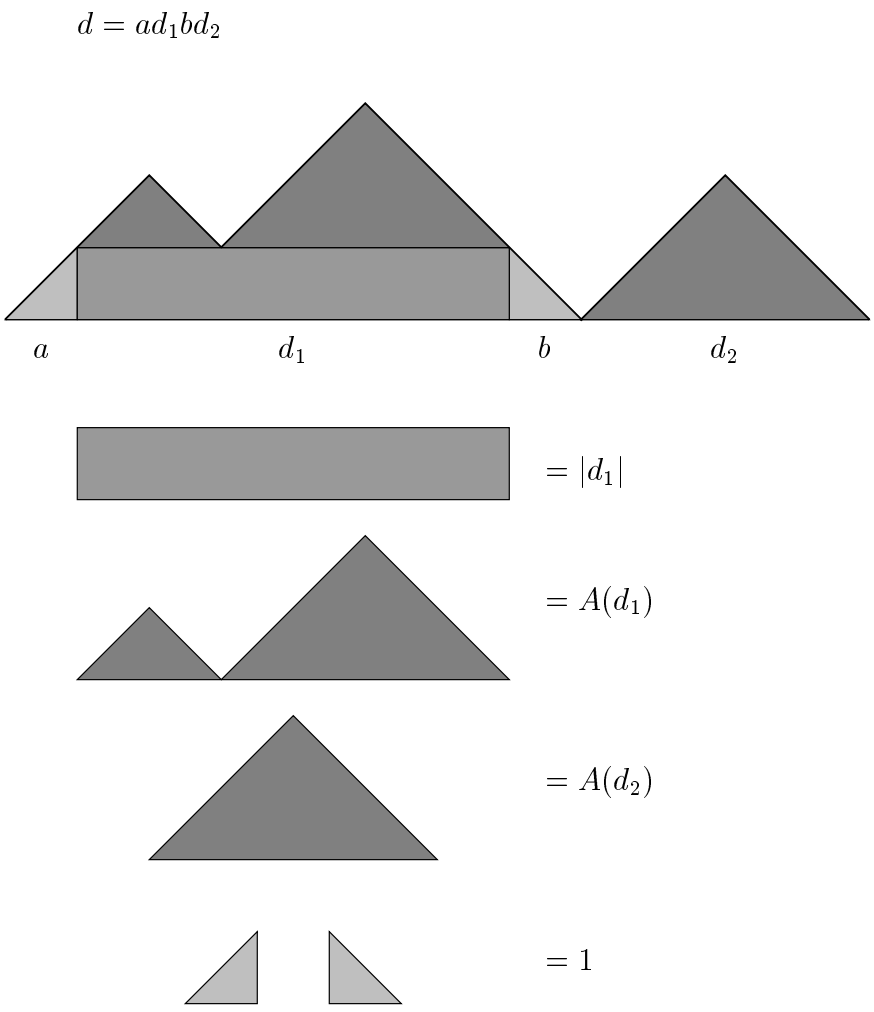


FIG. 2.1: *Règle de calcul de l'aire d'un chemin de Dyck*

codé par d_1 voit son ordonnée augmentée de 1; ces points sont au nombre de $A(d_1)+|d_1|+1$. Par conséquent, le moment d'inertie $M(d)$ est donné par

$$M(d) = (M(d_1) + A(d_1) + |d_1| + 1) + M(d_2).$$

Les règles de calcul du paramètre M sont donc :

- Pour la règle $D \to \epsilon$: $M(\epsilon) = 0$;
- Pour la règle $D \to aDbD$: $M(ad_1bd_2) = M(d_1) + M(d_2) + A(d_1) + |d_1| + 1$; le terme de croissance est $A(d_1) + |d_1| + 1$: l'aire étant un paramètre de rang 2 et la longueur de rang 1, le moment d'inertie est bien de rang 3.

2.2.3 Paramètres élémentaires

Certaines propriétés des paramètres Q-comptables découlent immédiatement de leur définition par règles de croissance :

Proposition 2.8. *Toute combinaison linéaire à coefficients entiers positifs de paramètres Q-comptables, est un paramètre Q-comptable dont le rang est le rang maximum des paramètres concernés.*

Preuve. Soient $p_1, \ldots, p_n$, n paramètres Q-comptables, et soit $p = \lambda_1 p_1 + \cdots + \lambda_n p_n$ (avec $\lambda_i \in \mathbb{N}$) une combinaison linéaire de ces paramètres. Notons, pour chaque paramètre p_i et pour chaque règle de dérivation R, $\theta_{R,i}$ le terme de croissance de p_i pour cette règle : il est clair que le terme de croissance de p pour R est $\lambda_1\theta_{R,1} + \cdots + \lambda_n\theta_{R,n}$; par conséquent, si chaque p_i est Q-comptable et si le maximum des rangs est k, p est bien Q-comptable de rang k. □

Dès lors, il est naturel de rechercher une description atomique des paramètres Q-comptables.

Définition 2.9. Si p est un paramètre Q-comptable de rang 1, p est *élémentaire* s'il existe une règle $R_0 \in \mathcal{R}$, telle que le terme de croissance de p soit 0 pour toute autre règle que R_0, et 1 pour R_0.

Récursivement, si un paramètre p est Q-comptable de rang $k+1$, p est *élémentaire* s'il existe une règle $R_0 \in \mathcal{R}$, un entier $i \leq \alpha(R_0)$, et un paramètre p', élémentaire de rang k, tels que le terme de croissance de p soit 0 pour toute autre règle que R_0, et $p'(u_i)$ pour la règle R_0.

Exemple 2.10. Reprenons l'exemple de l'aire de Carlitz telle que nous l'avons définie comme paramètre Q-comptable de rang 2 dans la grammaire G_1. Soit $p_1(w) = |w|_a$, et $p_2(w) = A_c(w)$. Les termes de croissance des paramètres p_1 et p_2 pour les règles de dérivation R_1 et R_2 sont, d'après les exemples 2.6 et 2.7 :

$$\begin{array}{lll} R_1 : & \theta_{p_1}(\epsilon) = 0; & \theta_{p_2}(\epsilon) = 0; \\ R_2 : & \theta_{p_1}(ad_1bd_2) = 1; & \theta_{p_2}(ad_1bd_2) = p_1(d_1). \end{array}$$

Par conséquent, p_1 est un paramètre élémentaire de rang 1, et p_2, un paramètre élémentaire de rang 2.

Afin de décrire tous les paramètres élémentaires existant dans une grammaire donnée, il convient de considérer l'ensemble $\mathcal{R}$ des règles de dérivation comme un nouvel alphabet, et de définir l'alphabet $\mathcal{R}_p$ des "dérivations pointées" :

Définition 2.11. Une dérivation pointée est une règle de dérivation R dont on a distingué l'un des symboles non terminaux du membre droit. Si l'on a distingué le k-ème symbole, on la notera $R^{(k)}$.

Exemple 2.12. Dans la grammaire G_1 de l'exemple 2.2, la règle R_1 (règle terminale) ne donne aucune dérivation pointée, et la règle R_2 (d'arité 2) en donne deux. On a donc $\mathcal{R}_p = \left\{R_2^{(1)}, R_2^{(2)}\right\}$.

Les alphabets $\mathcal{R}$ et $\mathcal{R}_p$ nous permettent de décrire tous les paramètres élémentaires en leur donnant des "noms". Les noms de paramètres élémentaires sont des mots de $\mathcal{R}_p^*\mathcal{R}$, c'est-à-dire des suites de noms de règles de dérivation, dont toutes sauf la dernière sont des dérivations pointées.

Soit en effet p un paramètre élémentaire. Le nom de p est défini récursivement de la manière suivante :

- Si p est de rang 1, le nom de p est R, où R est l'unique règle de dérivation telle que $c_R = 1$;
- si p est de rang $k+1$, il existe un paramètre élémentaire p', de rang k, une règle $R \in \mathcal{R}$, et un entier $i \leq \alpha(R)$, tel que l'unique terme de croissance non identiquement nul de p soit, pour la règle R, $p'(u_i)$. Alors, si le nom de p' est W', le nom de p est $W = R^{(i)}W'$.

Notons que la longueur du nom d'un paramètre élémentaire, est également le rang de ce paramètre.

Notation 2.13. Pour tout mot $W \in \mathcal{R}_p^*\mathcal{R}$, le paramètre élémentaire dont le nom est W est noté p_W.

Par extension, si $W = \lambda_1 W_1 + \cdots + \lambda_k W_k$ est une combinaison linéaire à coefficients entiers positifs de mots de $\mathcal{R}_p^*\mathcal{R}$, on notera p_W le paramètre Q-comptable

$$p_W = \sum_{i=1}^{k} \lambda_i p_{W_i}.$$

De telles combinaison linéaires de noms de paramètres élémentaires, seront appelées *noms de paramètres Q-comptables.*

Exemple 2.14. Dans la grammaire G_1 de l'exemple 2.2, l'aire de Carlitz est le paramètre $A_c = p_{R_2^{(1)}R_2}$. L'aire géométrique est $A = 2A_c + p_{R_2} = 2p_{R_2^{(1)}R_2} + p_{R_2}$. Le moment d'inertie (voir exemple 2.7) est $M = 2p_{R_2^{(1)}R_2^{(1)}R_2} + 3p_{R_2^{(1)}R_2} + p_{R_2}$.

La proposition suivante montre clairement que les paramètres élémentaires suffisent à décrire tous les paramètres Q-comptables :

Proposition 2.15. *Tout paramètre Q-comptable de rang k peut s'écrire comme combinaison linéaire à coefficients entiers positifs de paramètres élémentaires de rangs inférieurs ou égaux à k, l'un au moins étant de rang exactement k.*

Preuve. La preuve, par récurrence sur k, est sans difficulté. Si $k = 1$, le paramètre p est entièrement défini par les constantes $(c_R)_{R\in\mathcal{R}}$. Il est alors immédiat que l'on a la décomposition

$$p = \sum_{R\in\mathcal{R}} c_R p_R.$$

Supposons maintenant la propriété vraie pour k, et soit p un paramètre Q-comptable de rang $k+1$. Pour chaque règle $R \in \mathcal{R}$, le terme de croissance de p est de la forme

$$\theta_p(w_0u_1\dots u_{\alpha(R)}w_{\alpha(R)}) = c_R + \sum_{i=1}^{\alpha(R)} p_{R,i}(u_i),$$

où chaque $p_{R,i}$ est un paramètre Q-comptable de rang au plus k (éventuellement, identiquement nul).

D'après l'hypothèse de récurrence, chaque paramètre $p_{R,i}$ peut s'écrire $p_{R,i} = p_{W_{R,i}}$, où chaque $W_{R,i}$ est une combinaison linéaire de mots de $\mathcal{R}_p^*\mathcal{R}$. Il est alors immédiat que l'on peut décomposer p de la manière suivante :

$$p = \sum_{R\in\mathcal{R}} \left(c_R p_R + \sum_{i=1}^{\alpha(R)} p_{R^{(i)}W_{R,i}} \right).$$

Ceci termine la récurrence, et la preuve de la proposition. □

Nous verrons plus tard que cette association entre paramètres et noms de paramètres n'est pas forcément bijective : deux paramètres p_W et $p_{W'}$ (avec $W \neq W'$) peuvent être identiques sur tous les mots engendrés par la grammaire; toutefois, lorsqu'un paramètre est donné par des règles de croissance, ces règles ne l'associent qu'à une seule combinaison de mots de $\mathcal{R}_p^*\mathcal{R}$.

2.2.4 Interprétation des paramètres Q-comptables

Les paramètres naturellement étudiés sur les mots sont fréquemment décrits comme le "nombre d'occurrences de tel ou tel événement" : nombre d'inversions, nombre d'occurrences d'une lettre, d'un facteur ... Nous allons donner une interprétation formelle de tout paramètre élémentaire et donc, par suite, de tout paramètre Q-comptable.

Il est relativement aisé d'interpréter un paramètre élémentaire de rang 1 : puisque le terme de croissance vaut 1 à chaque utilisation d'une règle de dérivation particulière, la valeur du paramètre sur un mot donné sera le nombre d'utilisations de cette règle de dérivation qui sont nécessaires pour obtenir ce mot à partir de l'axiome. En d'autres termes, un paramètre élémentaire de rang 1 "compte" une certaine règle, ou, ce qui est équivalent, les nœuds de l'arbre de dérivation qui sont étiquetés par cette règle.

Nous avons vu précédemment qu'un paramètre "nombre d'occurrences de la lettre x" est toujours Q-comptable de rang 1. Réciproquement, pour tout paramètre p, Q-comptable de rang 1, il est possible d'ajouter au langage une lettre qui sert de "marqueur" du paramètre p, de telle sorte que le nombre d'occurrences de cette nouvelle lettre soit toujours la valeur du paramètre :

Proposition 2.16. *Soit L' un langage algébrique engendré par une grammaire $G' = (X, N, \mathcal{R}', S)$, et soit p un paramètre Q-comptable de rang 1 sur G'. Soit également m une lettre n'appartenant pas à l'alphabet X.*

Il existe un langage L, engendré par une grammaire $G = (X \cup \{m\}, N, \mathcal{R}, S)$, tel que

- *la projection $\varphi : (X \cup \{m\})^* \to X^*$ établit une bijection de L sur L';*
- *pour tout mot $w \in L$, $|w|_m = p(\varphi(w))$;*
- *pour tout mot $w \in L$, les arbres de dérivation de w (dans G) et de $\varphi(w)$ (dans G') ont même forme.*

Preuve. L'ensemble des règles de dérivation $\mathcal{R}$ se forme très simplement à partir de $\mathcal{R}'$: pour chaque règle $R \in \mathcal{R}'$, si R apparaît c_R fois dans la décomposition en paramètres élémentaires de p, la règle correspondante de $\mathcal{R}$ s'obtient en ajoutant c_R occurrences de m dans le membre droit de R. Ces occurrences de m peuvent être ajoutées en n'importe quelles positions; le simple fait que la grammaire G' soit non ambiguë suffit à assurer que G le sera. □

Pour interpréter les paramètres de rang supérieur à 1, nous avons besoin de la notion

de *chaîne* dans un arbre :

Définition 2.17. Soit $\mathcal{A}$ un arbre de dérivation de la grammaire G. Une *chaîne de longueur* k de $\mathcal{A}$ est un k-uplet $(s_1, \ldots, s_k)$ de nœuds de $\mathcal{A}$ tel que, pour tout $i < k$, s_{i+1} soit un descendant de s_i, distinct de s_i. Le *type* d'une chaîne de longueur k est le mot $R_1^{(d_1)} \ldots R_{k-1}^{(d_{k-1})} R_k \in \mathcal{R}_p^{k-1}\mathcal{R}$, où R_i est l'étiquette de s_i et où s_{i+1} appartient à l'arbre dont la racine est le d_i-ème fils de s_i.

Remarque. Les différents nœuds qui composent une chaîne doivent se trouver sur une même branche, mais peuvent parfaitement ne pas être consécutifs sur cette branche.

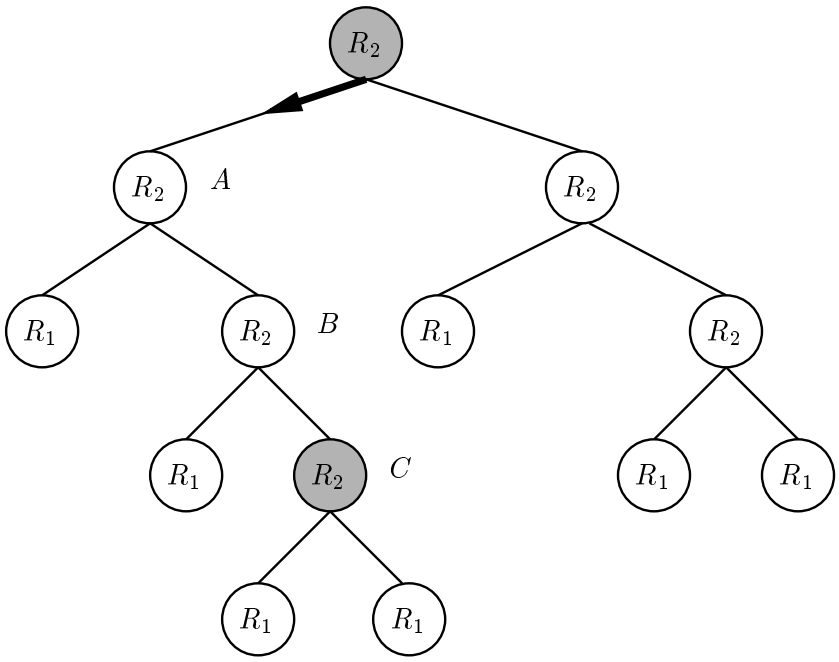


FIG. 2.2: *Une chaîne de type* $R_2^{(1)}R_2$

Exemple 2.18. Nous avons vu précédemment que, dans la grammaire G_1 engendrant les mots de Dyck, le paramètre "aire de Carlitz" est le paramètre élémentaire $p_{R_2^{(1)}R_2}$. Autrement dit, son terme de croissance, pour chaque sommet s étiqueté R_2 d'un arbre de dérivation, est égal au nombre de sommets étiquetés R_2 dans le sous-arbre gauche de s. Ceci revient à dire que le paramètre $p_{R_2^{(1)}R_2}$ compte le nombre de chaînes de type $R_2^{(1)}R_2$ dans les arbres de dérivation.

Ainsi, l'arbre de la figure 2.2 possède 3 chaînes de type $R_2^{(1)}R_2$ (chacune ayant la racine comme premier sommet, et l'un des 3 sommets internes du sous-arbre gauche, A, B, et C, comme second sommet). Cet arbre est, dans la grammaire G_1, l'arbre de dérivation du

mot $w = aababababbabab$. Le chemin de Dyck correspondant, représenté figure 2.3, a bien pour aire de Carlitz 3.

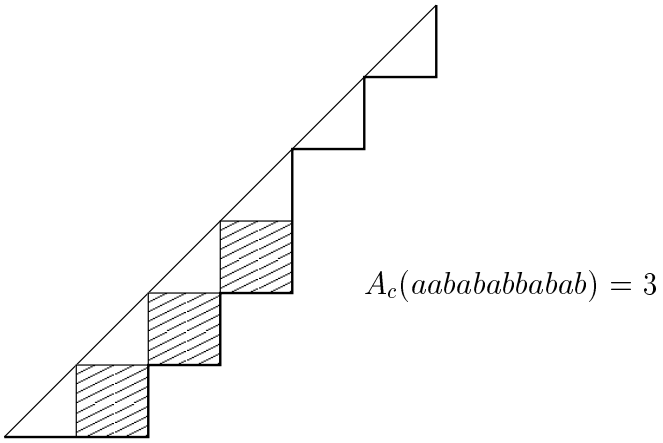


FIG. 2.3: *Chemin de Dyck associé à l'arbre de la figure 2.2*

Nous allons voir que ce type d'interprétation est généralisable à tous les paramètres élémentaires, quel que soit leur rang.

Lemme 2.19. *Soit w un mot engendré par la grammaire G, et p un paramètre Q-comptable.*

Soit $\mathcal{A}$ l'arbre de dérivation de w. Pour chaque nœud s de $\mathcal{A}$, soit $p'(s)$ le terme de croissance de p correspondant au nœud s.

Alors

$$p(w) = \sum_{s \in \mathcal{A}} p'(s) \tag{1}$$

Preuve. Ce lemme se prouve par récurrence sur la taille des arbres de dérivation. Il est en effet évident pour un arbre n'ayant qu'un seul nœud; supposons-le vrai pour les arbres ayant au plus n nœuds, et soit w un mot dont l'arbre de dérivation comporte $n+1$ nœuds.

Soit r la racine de cet arbre, $s_1, \ldots, s_k$ ses fils, et $w_1, \ldots, w_k$ les mots dont les arbres de dérivation sont les sous-arbres $\mathcal{A}_1, \ldots, \mathcal{A}_k$ de $\mathcal{A}$ de racines respectives $s_1, \ldots, s_k$. Chacun des sous-arbres de racine s_i a au plus n nœuds, donc le lemme s'applique à chaque w_i. Or, la différence entre $p(w)$ et la somme $\sum_{i=1}^{k} p(w_i)$ est, par définition, le terme de croissance

$p'(r)$. On a donc

$$\begin{aligned} p(w) &= p'(r) + \sum_{i=1}^{k} p(w_i) \\ &= p'(r) + \sum_{i=1}^{k} \sum_{s \in \mathcal{A}_i} p'(s) \\ &= \sum_{s \in \mathcal{A}} p'(s) \end{aligned}$$

ce qui prouve le lemme pour $n+1$ et permet de conclure par récurrence. □

Ce lemme exprime simplement le fait qu'un paramètre est la somme de tous ses termes de croissance; il nous permet de donner rapidement une interprétation des paramètres élémentaires.

Proposition 2.20. *Soit w un mot engendré par la grammaire G, et soit p_W un paramètre élémentaire ($W \in \mathcal{R}_p^*\mathcal{R}$). Alors $p_W(w)$ est le nombre de chaînes de type W de l'arbre de dérivation de w.*

Preuve. Dans le cas où le paramètre est de rang 1 (en ce cas, $W = R \in \mathcal{R}$), il s'agit d'une application immédiate du lemme 2.19. En effet, dans l'arbre de dérivation, le terme de croissance d'un paramètre élémentaire de rang 1 vaut 1 pour un nœud étiqueté par la règle R, et 0 dans les autres cas. Par conséquent, la somme de ces contributions sera le nombre de nœuds étiquetés R, qui forment les "chaînes de type R".

Supposons maintenant que p_W soit de rang $k > 1$; W est alors de longueur k, et $W = R^{(i)}W'$ où $R \in \mathcal{R}$ et $|W'| = k-1$. Toute chaîne de type W est composée d'un premier nœud s_1, étiqueté R, et d'une chaîne de type W' du sous-arbre dont la racine est le i-ème fils de s_1. Le nombre de chaînes de type W dont le premier nœud est s_1, est donc le terme de croissance $p'(s_1)$. Le lemme 2.19 permet alors d'affirmer que $p_W(w)$ est le nombre de chaînes de type W de l'arbre de dérivation de w. □

La proposition 2.20 permet, connaissant une décomposition d'un paramètre Q-comptable comme combinaison linéaire de paramètres élémentaires, d'en donner une interprétation comme comptant certains types de chaînes dans les arbres de dérivations. Ce comptage doit être pondéré par les coefficients de la décomposition. Par exemple, si un paramètre élémentaire est présent avec un coefficient 2, chaque chaîne correspondante doit être comptée 2 fois.

Dans [37], Fédou considère sur les arbres binaires complets une valuation qui fait intervenir un paramètre "somme des nombres de feuilles des sous-arbres gauches", noté $\mathcal{G}(\mathcal{A})$. Les arbres binaires complets sont équivalents aux arbres de dérivation de la grammaire classique engendrant le langage de Dyck (grammaire G_1 de l'exemple 2.3), puisque dans ces arbres de dérivation les feuilles sont toujours étiquetées R_1, et les nœuds internes, R_2. Dans ces conditions, compter la somme des nombres de feuilles des sous-arbres gauches revient exactement à compter les chaînes de type $R_2^{(1)}R_1$.

Notons à ce propos le lien que l'on peut établir entre ce paramètre $\mathcal{G}$ et l'aire de Carlitz : tout arbre binaire complet ayant une feuille de plus qu'il n'a de sommets internes, un arbre de dérivation a toujours autant de chaînes de type $R_2^{(1)}R_1$ que de chaînes de type $R_2^{(1)}R_2$ ou R_2. En d'autres termes, $p_{R_2^{(1)}R_1} = p_{R_2} + p_{R_2^{(1)}R_2}$, ce qui signifie que le paramètre $\mathcal{G}$ est identique au paramètre "aire de Carlitz plus demi-longueur".

2.2.5 Ordre de grandeur maximal de paramètres

La proposition 2.20 permet également de donner a priori une borne maximale à la valeur d'un paramètre Q-comptable sur un mot dont l'arbre de dérivation est de taille fixée ou, ce qui revient à peu près au même, sur un mot de longueur fixée.

En effet, si un arbre comporte n nœuds, il est évident qu'il ne peut avoir plus de n^k chaînes – tous types confondus – de longueur k. Par conséquent, un paramètre élémentaire de rang k est forcément inférieur ou égal à n^k sur un mot dont l'arbre de dérivation comporte n nœuds.

Dans une grammaire non ambiguë, il y a une relation linéaire entre la longueur d'un mot et la taille de son arbre de dérivation :

Proposition 2.21. *Soit G une grammaire non ambiguë, engendrant un langage L. Pour chaque mot non vide $w \in L$, notons $||w||$ la taille de l'arbre de dérivation de w.*

Alors il existe deux constantes strictement positives C_1 et C_2 telles que, pour tout mot non vide $w \in L$,

$$C_1||w|| \leq |w| \leq C_2||w||. \tag{2}$$

Preuve. La majoration de $|w|$ est facile, et ne dépend pas de la non-ambiguïté de la grammaire : on peut prendre pour C_2, le nombre maximum de lettres (avec multiplicités) écrites par une règle de dérivation, ce qui assurera $|w| \leq C_2||w||$ pour tout w engendré par la grammaire.

Pour la minoration, il est nécessaire que la grammaire soit non ambiguë.

Soit, pour chaque $U \in N$, $\mathcal{A}_U$ un arbre de dérivation dont l'étiquette de la racine soit une U-dérivation, et qui représente un mot dont la longueur soit minimale parmi les mots du langage $L_G(U)$. Soit $C' = \max_{U \in N} ||\mathcal{A}_U||$.

Soit maintenant k un entier tel que tout arbre de dérivation de profondeur k, vérifie $||\mathcal{A}|| > C'$ ($k = C' + 1$ convient parfaitement), et soit C la plus grande taille d'un arbre de dérivation de profondeur inférieure ou égale à k.

Nous allons montrer par récurrence sur n que, pour tout arbre de dérivation $\mathcal{A}$, si $|\mathcal{A}| \leq n$, alors $||\mathcal{A}|| \leq Cn + C'$.

La propriété est clairement vraie pour $n = 0$, par définition de C'. Supposons-la vraie pour n, et soit $\mathcal{A}$ un arbre de dérivation tel que $|\mathcal{A}| = n + 1$.

Si $\mathcal{A}$ est de profondeur inférieure à k, alors nécessairement $||\mathcal{A}|| \leq C \leq C(n+1) + C'$. Supposons donc que $\mathcal{A}$ soit de profondeur supérieure ou égale à k, et choisissons dans $\mathcal{A}$ un sommet s tel que le sous-arbre issu de s soit, lui, de profondeur k; soit U le membre gauche de l'étiquette de s, et soit $\mathcal{B} = \mathcal{A}(s, \mathcal{A}_U)$.

Nous avons

$$|\mathcal{B}| = |\mathcal{A}| - |s| + |\mathcal{A}_U|, \tag{3}$$

$$||\mathcal{B}|| = ||\mathcal{A}|| - ||s|| + ||\mathcal{A}_U||. \tag{4}$$

Nous avons forcément $|s| > C' \geq |\mathcal{A}_U|$, donc $|\mathcal{B}| \leq n$ et l'hypothèse de récurrence s'applique à $\mathcal{B}$: $||\mathcal{B}|| \leq Cn + C'$. En reprenant (4), nous obtenons alors

$$\begin{aligned} ||\mathcal{A}|| &= ||\mathcal{B}|| + ||s|| - ||\mathcal{A}_U|| \\ &\leq ||\mathcal{B}|| + ||s|| \\ &\leq Cn + C' + C, \end{aligned}$$

qui prouve bien que la propriété est vraie pour $n + 1$. □

L'intérêt de cette proposition est essentiellement de nous garantir que, pour des mots très longs, la taille d'un arbre de dérivation ou la longueur du mot correspondant ont toujours le même ordre de grandeur. Par conséquent, évaluer un ordre de grandeur en fonction de la taille des mots, ou en fonction de la taille des arbres de dérivation, est équivalent.

Nous commençons par donner une majoration élémentaire de l'ordre de grandeur d'un paramètre Q-comptable de rang connu.

Proposition 2.22. *Soit p un paramètre Q-comptable de rang k. Il existe une constante K telle que, pour tout mot non vide engendré par la grammaire,*

$$p(w) \leq K|w|^k. \tag{5}$$

Preuve. Dans le cas d'un paramètre élémentaire de rang k, cela découle immédiatement du fait que le nombre de chaînes est forcément inférieur au nombre de k-uplets de sommets de l'arbre; la décomposition en combinaison linéaire de paramètres élémentaires permet d'étendre la proposition à tout paramètre Q-comptable. □

Essentiellement, nous venons de montrer que l'ordre de grandeur maximal d'un paramètre Q-comptable par rapport à la taille des mots sur lesquels on le calcule, est limité par son rang. Nous allons maintenant donner une *minoration* de cet ordre de grandeur maximal qui permettra, dans certains cas, d'affirmer que cette limite est bien atteinte.

Définition 2.23. On appelle *rang minimal* d'un paramètre Q-comptable, la plus petite constante positive k telle qu'il existe une constante K vérifiant, pour tout mot w non vide engendré par la grammaire,

$$p(w) \leq K|w|^k. \tag{6}$$

Il est clair, d'après la proposition 2.22, que le rang minimal d'un paramètre Q-comptable existe[2] et est inférieur ou égal à son rang. Notons ici que ce rang minimal ne dépend pas de la grammaire, mais seulement des valeurs du paramètre, contrairement au rang. Il est moins évident que, comme nous allons le montrer, il s'agit toujours d'un entier.

Le cas des paramètres de rang 1 est un peu à part, et nous le traitons en premier.

Proposition 2.24. *Soit $G = (X, N, \mathcal{R}, S_0)$ une grammaire non ambiguë.*

Soit $p = p_R$ un paramètre élémentaire de rang 1, et soit S le symbole du membre gauche de R. Les conditions suivantes sont équivalentes :

1. *p est de rang minimal 1.*
2. *p n'est pas borné sur $L_G(S_0)$.*
3. *il existe un symbole S' tel que p ne soit pas borné sur $L_G(S')$.*
4. *S est accessible à partir d'un symbole figurant dans le membre droit de R, ou il existe un symbole S' tel que S et S' soient simultanément accessibles à partir de S'.*

2. Tel qu'il est défini ci-dessus, le rang minimal n'est qu'une borne inférieure et non un minimum; toutefois, dans toutes les démonstrations qui suivent pour les paramètres Q-comptables, il est aisé de vérifier qu'il s'agit effectivement d'un minimum.

La condition (1) est celle qui nous intéresse; les conditions (2) et (3) sont en apparence plus faibles, et la condition (4) est celle qui, techniquement, est la plus simple à tester sur une grammaire donnée lorsque le paramètre est seulement connu par les chaînes qu'il énumère.

En particulier, les conditions (1) et (2) impliquent qu'un paramètre élémentaire de rang 1, est forcément de rang minimal 0 ou 1; il n'est pas possible qu'un tel paramètre ait un ordre de grandeur maximal qui suive une loi de puissance non entière, ou logarithmique.

Preuve. (1) $\Rightarrow$ (2) $\Rightarrow$ (3) est évident. (3) $\Rightarrow$ (4) est plus simple à prouver par l'absurde : montrons donc que, si la condition (4) n'est pas vérifiée, le paramètre p est borné sur chaque langage $L_G(S')$, ce qui revient à dire qu'un arbre de dérivation de G ne peut contenir qu'un nombre borné de sommets étiquetés R.

Tout d'abord, dire que S n'est pas accessible à partir des symboles du membre droit de S, revient à dire qu'aucune branche d'un arbre de dérivation de G ne peut contenir plus d'un sommet étiqueté R. De plus, si une branche d'un tel arbre de dérivation contient deux sommets s_1 et s_2 ayant la même étiquette, aucun sommet compris entre s_1 et s_2 sur cette branche ne peut avoir R pour étiquette.

Supposons de plus que, pour tout symbole S', S et S' ne sont pas simultanément accessibles à partir de S'. Alors, si deux sommets s_1 et s_2 d'un arbre de dérivation ont même étiquette et sont sur une même branche (s_1 étant un ancêtre de s_2), tout sommet étiqueté R dans le sous-arbre issu de s_1, doit en fait se trouver dans le sous-arbre issu de s_2. Par conséquent, on peut, dans l'arbre de dérivation, remplacer le sous-arbre issu de s_1 par celui issu de s_2, sans changer le nombre de sommets étiquetés R. En procédant ainsi tant qu'il reste dans l'arbre des branches contenant des sommets de même étiquette, on se ramène à un arbre de profondeur bornée (par le nombre de règles de dérivation), sans modifier la valeur de p. Par conséquent, p, qui prend toutes ses valeurs sur un ensemble fini d'arbres de dérivation, est borné.

Supposons maintenant (4) vraie, et prouvons (1). Nous avons deux cas possibles : soit S est accessible à partir des symboles du membre droit de R, soit il existe un symbole S' tel que S et S' soient simultanément accessibles à partir de S'. Nous exhibons, dans chaque cas, une famille d'arbres de dérivation $(\mathcal{A}_n)_{n\geq 1}$, tels que $|\mathcal{A}| \leq Kn$ et $p(\mathcal{A}_n) \geq n$.

Considérons d'abord le cas où S est accessible à partir d'un symbole du membre droit de R : cela revient à dire que, dans un arbre de dérivation dont la racine est étiquetée R, il peut y avoir un autre sommet étiqueté R. Soit $\mathcal{A}$ un tel arbre, et soit s un sommet étiqueté R de $\mathcal{A}$. Soit alors $(\mathcal{A}_n)_{n\geq 1}$ la suite d'arbres définie récursivement par :

– $\mathcal{A}_1 = \mathcal{A}$;

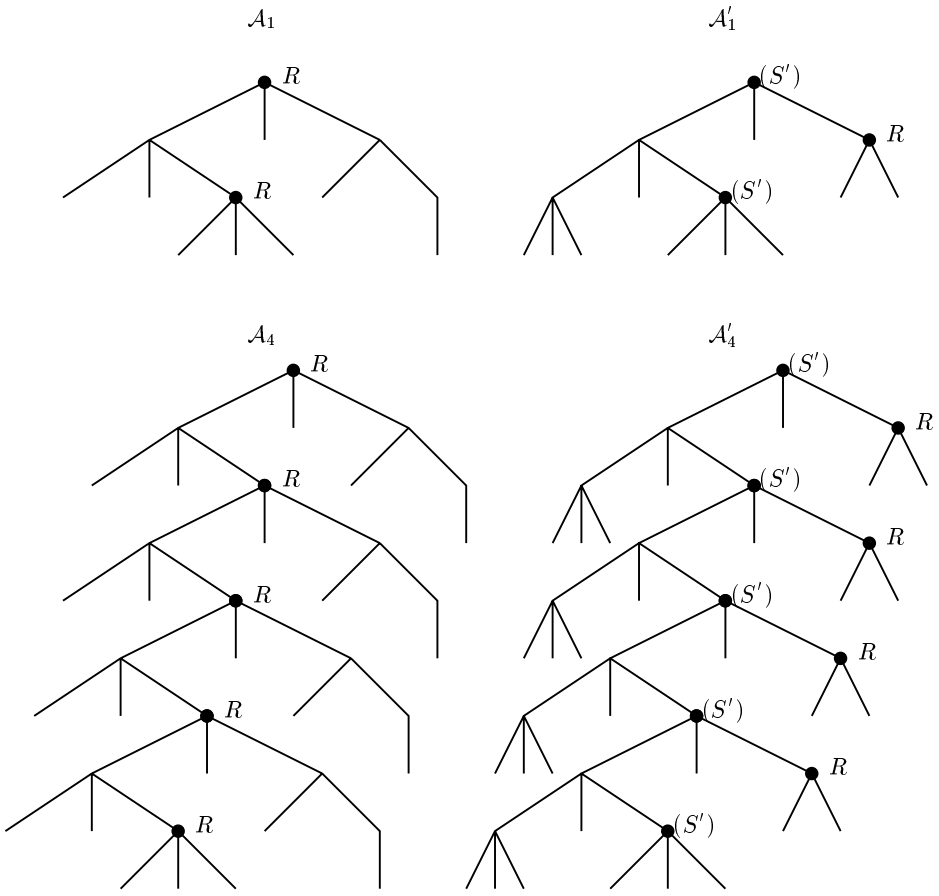


FIG. 2.4: *Construction des arbres* $\mathcal{A}_k$ *et* $\mathcal{A}'_k$

– $\mathcal{A}_{k+1} = \mathcal{A}(s, \mathcal{A}_k)$.

Un exemple de construction des arbres $\mathcal{A}_k$ est présenté figure 2.4.

Il est clair que $|\mathcal{A}_n| \leq n|\mathcal{A}|$ et que $p(\mathcal{A}_n) \geq n+1$.

Lorsque S et S' sont simultanément accessibles à partir de S', il suffit de prendre comme arbre $\mathcal{A}'$ un arbre dont la racine est étiquetée par une S'-dérivation, et comportant des sommets s et s' respectivement étiquetés par cette même S'-dérivation et par R, sans que ces deux sommets soient sur la même branche. La famille $(\mathcal{A}'_n)_{n\geq 1}$ est alors définie de la même manière : on a encore $|\mathcal{A}'_n| \leq n|\mathcal{A}'|$ et $p(\mathcal{A}'_n) \geq n+1$.

La construction des arbres $\mathcal{A}'_k$ est également présentée figure 2.4. □

Exemple 2.25. Reprenons la grammaire G_2 de l'exemple 2.3, qui engendre le langage de

Dyck. Les règles de cette grammaire d'axiome D sont :

$$\begin{aligned} R_1 : \quad & D \rightarrow \epsilon \\ R_2 : \quad & D \rightarrow E \\ R_3 : \quad & E \rightarrow ab \\ R_4 : \quad & E \rightarrow abE \\ R_5 : \quad & E \rightarrow aEb \\ R_6 : \quad & E \rightarrow aEbE. \end{aligned}$$

Dans cette grammaire, D n'est accessible à partir d'aucun symbole (il n'apparaît dans le membre droit d'aucune règle), tandis que E est accessible à partir de D comme de E. Pour finir, E et E sont simultanément accessibles à partir de D ou de E (règle R_6).

Les règles R_1 et R_2 étant des D-dérivations, les paramètres p_{R_1} et p_{R_2} sont de rang minimal 0 : ils sont tous deux bornés. Plus précisément, p_{R_1} vaut 1 pour le mot vide et 0 pour tout autre mot de Dyck, et p_{R_2} vaut 0 pour le mot vide et 1 pour tout autre mot de Dyck.

En revanche, les règles R_3 à R_6 étant des E-dérivations, les paramètres p_{R_3}, p_{R_4}, p_{R_5} et p_{R_6} sont de rang minimal 1. Le paramètre p_{R_3} représente, sur tous les arbres de dérivation de mots de Dyck non vides, le nombre de feuilles (la règle R_3 est la seule règle terminale apparaissant dans leurs arbres de dérivation); p_{R_6} compte, pour sa part, les sommets internes de degré 2 (R_6 est la seule règle d'arité 2), et p_{R_4} et p_{R_5} comptent chacun un type différent de sommets de degré 1.

A titre d'exemples de suites de mots illustrant le fait que ces paramètres sont bien de rang minimal 1, citons :

- pour p_{R_4} : $w_n = (ab)^n$ vérifie $p_{R_4}(w_n) = n - 1$, et $|w_n| = 2n$;
- pour p_{R_5} : $w_n = a^n b^n$ vérifie $p_{R_5}(w_n) = n - 1$, et $|w_n| = 2n$;
- pour p_{R_3} et p_{R_6} : $w_n = (aabb)^n ab$ vérifie $p_{R_3}(w_n) = n + 1$ et $p_{R_6}(w_n) = n$, et $|w_n| = 4n + 2$.

Le rang minimal d'un paramètre Q-comptable quelconque peut être déterminé assez aisément par la proposition suivante :

Proposition 2.26. *Soit $p = p_W$ un paramètre élémentaire de rang k, non identiquement nul, avec $W = R_1^{(d_1)} \dots R_{k-1}^{(d_{k-1})} R_k$. Notons, pour chaque $i \leq k$, S_i le symbole non terminal du membre gauche de R_i, et, pour $i < k$, S_i' le d_i-ème symbole du membre droit de R_i. S_0 désigne l'axiome de la grammaire.*

Le rang minimal de p est obtenu en comptant, parmi les indices $i < k$, ceux pour lesquels S_i est accessible depuis S'_i; à ce total, il convient d'ajouter 1 si et seulement si p_{R_k} est de rang minimal 1.

Preuve. Tout d'abord, remarquons que p_W n'est pas identiquement nul, si et seulement si, pour chaque $i < k$, S_{i+1} est accessible à partir de S'_i ou égal à S'_i (il faut que le symbole S_{i+1} apparaisse dans au moins un arbre de dérivation dont la racine est une S'_i-dérivation).

Notons k' le rang minimal donné par l'énoncé. Nous commençons par prouver que le paramètre p est au moins de rang minimal k', en exhibant une famille $(\mathcal{A}_n)_{n\geq 0}$ d'arbres de dérivation tels que $|\mathcal{A}_n| \leq Kn$ et $p(\mathcal{A}_n) \geq n$.

Pour chaque $i < k$, soit $\mathcal{B}_i$ un arbre de dérivation vérifiant les conditions suivantes :

- la racine de $\mathcal{B}_i$ est étiquetée par R_i;
- le d_i-ème sous-arbre de $\mathcal{B}_i$ comporte un sommet s_{i+1}, étiqueté par R_{i+1};
- si S_i est accessible à partir de S'_i (c'est-à-dire, si l'indice i fait partie de ceux qui contribuent au rang minimal annoncé), le d_i-ème sous-arbre de $\mathcal{B}_i$ comporte un sommet s'_i étiqueté R_i.

Pour chaque indice i tel que S_i soit accessible à partir de S'_i, soit $(\mathcal{B}_{i,n})_{n\geq 1}$ la suite d'arbres de dérivation définie par :

- $\mathcal{B}_{i,1} = \mathcal{B}_i$;
- $\mathcal{B}_{i,k+1} = \mathcal{B}_i(s'_i, \mathcal{B}_{i,k})$.

Cette définition assure que, dans chaque arbre $\mathcal{B}_{i,n}$, il existe une branche contenant n sommets étiquetés R_i, chacun étant un descendant du d_i-ème fils du précédent. Autrement dit, l'arbre $\mathcal{B}_{i,n}$ possède au moins une chaîne de type $\left(R_i^{(d_i)}\right)^{n-1} R_i$. La taille de cet arbre vérifie également $|\mathcal{B}_{i,n}| \leq n|\mathcal{B}_i|$.

Posons maintenant

$$\mathcal{B}'_i = \begin{cases} \mathcal{B}_{i,n} & \text{si } S_i \text{ est accessible à partir de } S'_i, \\ \mathcal{B}_i & \text{sinon.} \end{cases}$$

Pour $\mathcal{B}'_k$, nous prenons l'arbre $\mathcal{A}_n$ construit dans la preuve de la proposition 2.24 si p_{R_k} est de rang minimal 1 (c'est-à-dire si k contribue au rang minimal), et $\mathcal{B}_k$ sinon.

Soit maintenant $\mathcal{A}_n = \mathcal{B}'_1(s_2, \mathcal{B}'_2(s_3, \dots \mathcal{B}'_{k-1}(s_k, \mathcal{B}'_k)\cdots))$. Il est clair que la taille de $\mathcal{A}_n$ est $|\mathcal{A}_n| \leq n(|\mathcal{B}_1| + \cdots + |\mathcal{B}_k|$.

Par ailleurs, l'arbre $\mathcal{A}_n$ vérifie $p(\mathcal{A}_n) \geq n^{k'}$, ce qui permet d'affirmer que p est au moins de rang minimal k'.

Montrons maintenant, par récurrence sur k, que k' est bien le rang minimal de p; pour cela, nous devons montrer que tout arbre $\mathcal{A}$ tel que $p(\mathcal{A}) \geq Kn^{k'}$ vérifie $|\mathcal{A}| \geq K'n$. La propriété est vraie pour $k = 1$ (c'est la proposition 2.24), supposons donc qu'elle l'est pour k et choisissons un paramètre élémentaire $p = p_W$ de rang $k+1$. Posons $W = R_1^{(d_1)}W'$.

Deux cas se présentent : ou bien S_1 est accessible à partir de S'_1, auquel cas $p_{W'}$ est de rang minimal $k'-1$; ou bien il ne l'est pas, auquel cas $p_{W'}$ est de rang minimal k'.

Le premier cas ($p_{W'}$ est de rang minimal $k'-1$) est facile : en effet, un arbre de taille n a au plus $Kn^{k'-1}$ chaînes de type W', et au plus n sommets étiquetés R_1; par conséquent, il ne peut avoir plus de $n.K.n^{k'-1} = Kn^{k'}$ chaînes de type W.

Le deuxième cas ($p_{W'}$ est déjà de rang minimal k') est légèrement plus complexe. Chaque chaîne de type W' ne peut être complétée en chaîne de type W qu'au plus une fois, puisque la branche reliant la racine de l'arbre au sommet s_1 ne peut comporter qu'au plus une dérivation pointée de type $R_1^{d_1}$. Par conséquent $p_W(\mathcal{A}) \leq p_{W'}(\mathcal{A}) \leq Kn^{k'}$.

Dans tous les cas, le paramètre p_W est bien de rang minimal k'. □

Un cas particulier intéressant est le suivant :

Corollaire 2.27. *Si chaque symbole de la grammaire est accessible à partir de chaque symbole, tout paramètre Q-comptable a un rang minimal égal à son rang.*

Remarque. Si la grammaire ne comporte qu'un seul symbole, et engendre un langage infini, l'unique symbole est forcément accessible à partir de lui-même, et par conséquent chaque paramètre a un rang minimal égal à son rang.

Il est intéressant de noter que, dans le cas général, calculer le rang minimal d'un paramètre est assez simple, puisque tout se ramène à déterminer quels symboles sont accessibles à partir desquels, et quels couples de symboles sont accessibles à partir desquels. Dutour donne dans [36] un algorithme permettant de décider si un symbole est ou non accessible à partir d'un autre; quant à l'accessibilité simultanée, elle peut être calculée en adaptant aux couples de symboles l'algorithme donné dans [36].

La proposition 2.26 peut être utile dans la pratique lorsqu'il s'agit de déterminer si un paramètre donné p, défini indépendamment de toute grammaire, est Q-comptable dans une grammaire donnée. Si le rang minimal de p est connu, les paramètres élémentaires pouvant intervenir dans une décomposition de p sont en nombre fini. En calculant, pour un assez

grand nombre de mots, les valeurs de chacun de ces paramètres élémentaires $p_1, \ldots, p_N$, il est possible, soit de montrer que p ne peut pas s'écrire comme combinaison linéaire de tels paramètres (parce que, par exemple, pour M mots $w_1, \ldots, w_M$, le vecteur $(p(w_j))_{1\leq j\leq M}$ ne peut s'écrire comme combinaison linéaire des vecteurs $(p_i(w_j))_{1\leq j\leq M}$, pour $1 \leq i \leq N$), soit de restreindre les choix possibles de telles combinaisons linéaires jusqu'à ce qu'une preuve directe de la Q-comptabilité de p soit envisageable. Le principal obstacle à l'automatisation de ce genre de calculs réside dans le choix judicieux des mots pour lesquels effectuer les calculs; le nombre de mots d'un langage croît fréquemment de manière exponentielle avec leur longueur. Dans les cas simples, toutefois, il est fréquent que l'examen de quelques mots courts du langage engendré soit suffisant pour montrer qu'un paramètre donné n'est pas Q-comptable; la quasi-totalité de nos preuves de non-Q-comptabilité seront de ce type.

2.3 Séries génératrices

Nous nous intéressons maintenant aux liens entre Q-grammaires et séries génératrices, qui nous fournissent en fait la principale motivation pour l'étude des Q-grammaires et des paramètres Q-comptables.

2.3.1 Substitutions de variable

Nous considérons ici un ensemble ordonné $(x_1, \ldots, x_n)$ de variables formelles, et des séries formelles $F(x_1, \ldots, x_n)$, à coefficients entiers (le plus souvent positifs ou nuls).

Dans de nombreux problèmes d'énumération suivant plus d'un paramètre, on rencontre des équations portant sur des séries formelles et qui font intervenir, au lieu de la série à une variable $F(x)$, une série bivariée $F(x, q)$ et la série $F(xq, q)$, où la variable x a été multipliée par q; voir par exemple [37, 13, 28]. Nous allons considérer une généralisation de ce genre de transformation, au cas où les variables formelles sont plus nombreuses.

Définition 2.28. Soit, pour tout $i \leq n$, $A_i = x_i x_{i+1}^{\alpha_{i,i+1}} \ldots x_n^{\alpha_{i,n}}$ un monôme ne faisant intervenir que les variables formelles x_j telles que $i \leq j$ (et dans lequel x_i n'apparaît qu'au degré 1).

Nous noterons $\sigma_{(x_i \leftarrow A_i)}$ l'opérateur linéaire défini sur l'algèbre de séries formelles $\mathbb{Q}[[x_1, \ldots, x_n]]$ par

$$\sigma_{(x_i \leftarrow A_i)} F(x_1, \ldots, x_n) = F(A_1, \ldots, A_n). \tag{7}$$

Nous appellerons une telle transformation une substitution des variables $x_1, \ldots, x_n$.

Notons que si l'on pose $\alpha_{i,i} = 1$ et $\alpha_{i,j} = 0$ lorsque $i > j$, une substitution de variables σ est entièrement définie par la matrice à coefficients entiers positifs ou nuls $M_\sigma = (\alpha_{i,j})_{1\le i,j\le n}$. Nous appellerons cette matrice M_σ, la *matrice de la substitution*. Les matrices de substitutions de variables sont exactement les matrices triangulaires supérieures à coefficients dans $\mathbb{N}$ et dont les coefficients diagonaux sont tous égaux à 1.

Le rapport entre une substitution de variables et sa matrice est donné par le lemme suivant :

Lemme 2.29. *Soient σ et σ' deux substitutions de variables de matrices respectives $M_\sigma = (\alpha_{i,j})_{1\le i,j\le n}$ et $M_{\sigma'} = \left(\alpha'_{i,j}\right)_{1\le i,j\le n}$. Alors $\sigma'\circ\sigma$ est une substitution de variables, dont la matrice est $M_{\sigma'}M_\sigma$.*

Preuve. Soit, pour $1 \le i \le n$, $B_i = A_i A_{i+1}^{\alpha'_{i,i+1}} \dots A_n^{\alpha'_{i,n}}$. En composant σ et σ', il vient naturellement

$$\begin{aligned}\sigma'\circ\sigma F(x_1,\dots,x_n) &= \sigma' F(A_1,\dots,A_n)\\ &= F(B_1,\dots,B_n).\end{aligned}$$

Ceci montre que $\sigma'\circ\sigma$ est une substitution de variables, pourvu que chaque B_i soit bien de la forme requise.

Or, nous avons

$$\begin{aligned}B_i &= A_i A_{i+1}^{\alpha'_{i,i+1}} \dots A_n^{\alpha'_{i,n}}\\ &= \left(x_i x_{i+1}^{\alpha_{i,i+1}} \dots x_n^{\alpha_{i,n}}\right)\left(x_{i+1} x_{i+2}^{\alpha_{i+1,i+2}} \dots x_n^{\alpha_{i+1,n}}\right)^{\alpha'_{i,i+1}} \dots \left(x_{n-1} x_n^{\alpha_{n-1,n}}\right)^{\alpha'_{i,n-1}} x_n^{\alpha'_{i,n}}\\ &= x_i x_{i+1}^{\beta_{i,i+1}} \dots x_n^{\beta_{i,n}},\end{aligned}$$

où chaque coefficient $\beta_{i,j}$ (pour $i < j \le n$) est donné par

$$\beta_{i,j} = \sum_{k=1}^{n} \alpha_{k,j}\alpha'_{i,k}. \tag{8}$$

On reconnaît dans $\beta_{i,j}$ le coefficient de $M_{\sigma'}M_\sigma$, ce qui prouve bien que $\sigma'\circ\sigma$ est la substitution de variables dont la matrice est $M_{\sigma'}M_\sigma$. □

Dans la pratique, il arrivera fréquemment que des substitutions de variables ne modifient qu'une partie des variables formelles :

Notation 2.30. Lorsque certains des monômes A_i ne sont pas définis, $A_i = x_i$ est implicitement supposé; ainsi, dans $\mathbb{Q}[[x,y,z]]$, $\sigma_{x\leftarrow xz^2} = \sigma_{x\leftarrow xz^2, y\leftarrow y, z\leftarrow z}$ et l'on a

$$\sigma_{x\leftarrow xz^2} F(x,y,z) = F(xz^2,y,z).$$

Il est clair que $\sigma_{x_i \leftarrow x_i A}$ et $\sigma_{x_i \leftarrow x_i B}$ commutent, et que l'on a $\sigma_{x_i \leftarrow x_i A} \circ \sigma_{x_i \leftarrow x_i B} = \sigma_{x_i \leftarrow x_i AB}$; cette propriété ne s'étend pas aux substitutions dans deux variables différentes.

Exemple 2.31. Considérons des séries formelles à trois variables x, y, z, et posons $\sigma_1 = \sigma_{x \leftarrow xy}$ et $\sigma_2 = \sigma_{y \leftarrow yz}$. Alors

$$\begin{aligned} \sigma_1 \circ \sigma_2 F(x,y,z) &= \sigma_1 F(x, yz, z) \\ &= F(xy, yz, z), \\ \sigma_2 \circ \sigma_1 F(x,y,z) &= \sigma_2 F(xy, y, z) \\ &= F(xyz, yz, z). \end{aligned}$$

Ou, en termes de matrices :

$$M_{\sigma_1} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M_{\sigma_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

$$M_{\sigma_1 \circ \sigma_2} = M_{\sigma_2}.M_{\sigma_1} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad M_{\sigma_2 \circ \sigma_1} = M_{\sigma_1}.M_{\sigma_2} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Ce genre de substitutions de variables apparaît fréquemment dans les travaux sur les q-grammaires, où l'on ne rencontre généralement qu'une substitution $\sigma_{x \leftarrow xq}$. Ainsi que nous allons le voir, notre définition des paramètres Q-comptables correspond en fait à la généralisation à d'autres substitutions.

L'exemple (2.31) montre que, lorsque les substitutions portent sur des variables différentes, l'ordre dans lequel on les compose est important. Or, si l'on pense à σ_1 comme "multiplier x par y" et à σ_2 comme "multiplier y par z", le résultat attendu de la composition est $\sigma_1 \circ \sigma_2$ plutôt que $\sigma_2 \circ \sigma_1$. En d'autres termes, il convient de composer de telles substitutions dans l'ordre décroissant des variables substituées; ainsi, en utilisant la notation définie précédemment,

$$\sigma_{(x_i \leftarrow A_i)} = \sigma_{x_1 \leftarrow A_1} \circ \cdots \circ \sigma_{x_{n-1} \leftarrow A_{n-1}}. \tag{9}$$

Définition 2.32. Soient $x_1, \ldots, x_n$ n variables formelles, et soit $\sigma = \sigma_{(x_i \leftarrow A_i)}$ une substitution des variables $x_1, \ldots, x_n$, dont la matrice est M_σ. Nous appellerons restriction de σ à $x_1, \ldots, x_k$ $(k < n)$, la substitution de $x_1, \ldots, x_k$ dont la matrice est formée des k premières lignes et colonnes de M_σ.

En termes de variables et de séries formelles, restreindre une substitution de variables aux k premières variables revient à fixer toutes les autres variables formelles à la valeur 1.

2.3.2 Q-analogue d'un système d'équations

Dans la méthodologie "classique" de Schützenberger, une grammaire algébrique se traduit, en termes de séries génératrices, par un système d'équations algébriques dont les inconnues sont les séries génératrices des langages engendrés par la grammaire. Ces séries sont des séries formelles en des variables $x_1, \ldots, x_n$ correspondant aux lettres de l'alphabet, et les coefficients du système sont des polynômes en ces mêmes variables.

La façon la plus simple d'obtenir un tel système algébrique à partir de la grammaire est tout bonnement d'écrire la grammaire comme un système d'équations (en variables non commutatives) sur les langages, et de faire commuter lettres et langages, ces derniers étant remplacés par leurs séries génératrices qui font figure de séries formelles inconnues. La non ambiguïté des grammaires est ici cruciale.

De manière générale, on peut écrire le système d'équations sous la forme

$$U = \sum_{R \in \mathcal{R}(U)} v(R) \prod_{1 \leq j \leq \alpha(R)} d(R, j) \quad (U \in N) \tag{10}$$

où $\mathcal{R}(U)$ désigne l'ensemble des U-dérivations, $v(R)$ le produit commutatif des lettres produites par la règle R, et $d(R, j)$, le j-ème symbole (pris ici comme série inconnue) du membre droit de R.

La notion usuelle de q-analogue d'un tel système correspond à ajouter aux séries formelles une variable q, et à modifier le système de telle sorte que, lorsque $q = 1$, on retrouve le système d'origine; généralement, ces q-systèmes font intervenir des substitutions $\sigma_{x_i \leftarrow x_i q}$.

Exemple 2.33. La grammaire G_2 de l'exemple 2.3 se traduit automatiquement par le système d'équations

$$\begin{cases} D(a,b) &= 1 + E(a,b) \\ E(a,b) &= ab + 2abE(a,b) + abE(a,b)^2. \end{cases}$$

Un exemple de q-analogue de ce système est le suivant :

$$\begin{cases} D(a,b;q) &= 1 + E(a,b;q), \\ E(a,b;q) &= abq + abqE(a,b;q) + abqE(aq,bq;q) + abqE(a,b;q)E(aq,bq;q). \end{cases}$$

En posant $x = ab$ (les variables a et b apparaissent toujours avec le même degré), nous obtenons deux autres systèmes. Le second est toujours un q-analogue du premier :

$$\begin{cases} D(x) &= 1 + E(x), \\ E(x) &= x + 2xE(x) + xE(x)^2; \end{cases}$$

$$\begin{cases} D(x;q) &= 1 + E(x;q), \\ E(x;q) &= xq + xqE(x;q) + xqE(xq^2;q) + xqE(x;q)E(xq^2;q). \end{cases}$$

Définition 2.34. Soit (S) un système de m équations algébriques, où chaque équation est de la forme

$$U_i = P_i(x_1, \dots, x_n, U_1, \dots, U_m) \tag{11}$$

avec P_i un polynôme à coefficients dans $\mathbb{N}$.

Soit $Q = (q_1, \dots, q_k)$ un ensemble ordonné de k nouvelles variables formelles. Un Q-analogue de (S) est un système (S') de m équations, dont les inconnues sont des séries formelles des variables $x_1, \dots, x_n, q_1, \dots, q_k$, et qui s'écrit sous la forme

$$\tilde{U}_i = \tilde{P}_i\left(x_1, \dots, x_n, q_1, \dots, q_k, (\tilde{U}_i)_{1\le i\le m}, \left(\sigma_j(\tilde{U}_i)\right)_{1\le j\le s, 1\le i\le m}\right) \tag{12}$$

où chaque $\tilde{P}_i$ est un polynôme à coefficients dans $\mathbb{N}$ et σ_j (pour $1 \le j \le s$) une substitution des variables $x_1, \dots, x_n, q_1, \dots, q_k$, avec les conditions suivantes :

- la restriction de σ_j aux variables $x_1, \dots, x_n$ est l'identité;
- si, dans $\tilde{P}_i$, chaque variable q_ℓ $(1 \le \ell \le k)$ est remplacée par 1 et chaque $\sigma_j(\tilde{U}_i)$ est remplacé par U_i, on retrouve le polynôme P_i.

La façon la plus simple de comprendre comment former un Q-analogue d'un système d'équations est de considérer chaque polynôme P_i comme une somme de monômes unitaires, et de remplacer certains facteurs U_i dans ces monômes par $\sigma_j(U_i)$. De plus, chaque terme du polynôme est multiplié par un monôme unitaire ne faisant intervenir que les variables $q_1, \dots, q_k$.

Examinons sur un exemple le lien entre Q-grammaires et Q-analogue d'un système d'équations.

Exemple 2.35. Reprenons la grammaire G' de l'exemple (2.33) :

$$\begin{cases} R_1 : & D & \to & \epsilon & & R_2 : & D & \to & E \\ R_3 : & E & \to & ab & & R_4 : & E & \to & aEb \\ R_5 : & E & \to & abD & & R_6 : & E & \to & aEbE \end{cases}$$

Le système d'équations algébriques correspondant est le suivant :

$$\begin{cases} D(x) &= 1 + E(x), \\ E(x) &= x + 2xE(x) + xE(x)^2. \end{cases}$$

En le comparant à la grammaire G_2, il apparaît que le paramètre p_x compté par x, est égal à $p_{R'_3} + p_{R'_4} + p_{R'_5} + p_{R'_6}$.

En utilisant la substitution de variables $\sigma = \sigma_{x \leftarrow xq^2, r \leftarrow rs}$, nous pouvons former le (q, r, s)-analogue suivant :

$$\begin{cases} D(x,q,r,s) &= 1 + E(x,q,r,s), \\ E(x,q,r,s) &= xqrs + xqrsE(x,q,r,s) + xqE(xq^2,q,rs,s) \\ & \quad + xqE(xq^2,q,rs,s)E(x,q,r,s). \end{cases}$$

Chacune des variables compte un paramètre ayant un sens géométrique sur les chemins de Dyck : q compte *l'aire*, r compte les *pics*, et s compte la *somme des hauteurs des pics*.

La restriction aux variables x et q redonne le q-analogue présenté dans l'exemple (2.33).

La croissance du paramètre compté par q se fait de plusieurs manières :

- lors de l'utilisation des règles R'_3, R'_4, R'_5 et R'_6, il y a croissance de 1 (monômes $xqrs$ et xq);
- lors de l'utilisation des règles R'_5 et R'_6, il y a croissance d'un terme égal au double du degré de x dans une série $E(x,q,r,s)$ (substitution σ).

Par conséquent, q compte le paramètre Q-comptable de rang 2

$$p_q = p_{R'_3+R'_4+R'_5+R'_6} + 2.p_{(R'^{(1)}_5+R'^{(1)}_6)(R'_3+R'_4+R'_5+R'_6)}.$$

De même, r croît de 1 à chaque utilisation des règles R'_3 et R'_4, donc $p_r = p_{R'_3+R'_4}$, et, en analysant la croissance de p_s, on obtient

$$p_s = p_{R'_3+R'_4} + p_{(R'^{(1)}_5+R'^{(1)}_6)(R'_3+R'_4)}.$$

Ainsi, nous avons décomposé en somme de paramètres élémentaires chacun des quatre paramètres.

Le théorème suivant indique dans toute sa généralité le lien entre Q-grammaires et Q-analogues de systèmes d'équations :

Théorème 2.36. *Soit G une grammaire engendrant k langages $L_1, \ldots, L_k$, et soit (S) le système d'équations algébriques correspondant à G.*

- *Les solutions de tout Q-analogue du système S, sont les séries génératrices d'une Q-grammaire basée sur la grammaire G;*

- *réciproquement, les séries génératrices de toute Q-grammaire basée sur G sont les solutions d'un Q-analogue de (S).*

Preuve. Commençons par montrer de quelle manière, à partir d'une Q-grammaire, on peut former un système d'équations qui est un Q-analogue du système algébrique (S).

Notons $q_1, \ldots, q_N$ les lettres de la grammaire G, et $q_{N+1}, \ldots, q_{N+M}$ les M lettres supplémentaires. Soient $p_1, \ldots, p_{N+M}$ les paramètres suivant lesquels s'effectue l'énumération (les N premiers étant de rang 1), et notons, pour chaque règle R et chaque paramètre p_m, les règles de calcul

$$R: \quad p_m(w) = \sum_{i=1}^{a(R)} p_m(w_i) + C_{R,m} + \sum_{\substack{1 \le j < m \\ 1 \le i \le a(R)}} C_{R,m,i,j}.p_j(w_i).$$

Les N premiers paramètres étant de rang 1, pour $m \le N$ les coefficients $C_{R,m,i,j}$ sont tous nuls, et $C_{R,m} = |v_0 v_1 \ldots v_{a(R)}|_{q_m}$.

Notons encore, pour $w \in L_G(U)$,

$$v(w) = \prod_{1 \le m \le N+M} q_m^{p_m(w)}$$

la valuation donnée au mot w par ces paramètres. Si w appartient à plusieurs langages $L_G(U)$ différents, on distinguera le langage, en notant $v(w, U)$ et $p_m(w, U)$.

Les règles de calcul sommatoire pour les paramètres se transforment en produits; ainsi, si $w = v_0 w_1 \ldots w_k v_k$ pour une certaine règle $R : U \to v_0 U_1 \ldots U_k v_k$, avec $w_i \in L_G(U_i)$, on

a

$$
\begin{aligned}
v(w,U) &= \prod_{m=1}^{N+M} q_m^{p_m(w,U)} \\
&= \prod_{m=1}^{N+M} q_m^{\left(\sum_{i=1}^{k} p_m(w_i,U_i)+C_{R,m}+\sum_{\substack{1\le j\le m\\ 1\le i\le k}} C_{R,m,i,j}\cdot p_j(w_i,U_i)\right)} \\
&= \left(\prod_{i=1}^{k}\prod_{m=1}^{N+M} q_m^{p_m(w_i,U_i)}\right)\left(\prod_{m=1}^{N+M} q_m^{C_{R,m}}\right)\left(\prod_{\substack{1\le j<m\le N+M\\ 1\le i\le k}} q_m^{C_{R,m,i,j}\cdot p_j(w_i,U_i)}\right) \qquad (13)\\
&= \left(\prod_{1\le m\le N+M} q_m^{C_{R,m}}\right)\left(\prod_{i=1}^{k} v(w_i,U_i)\prod_{1\le j<m\le N+M} q_m^{C_{R,m,i,j}\cdot|v(w_i,U_i)|_{q_j}}\right). \qquad (14)
\end{aligned}
$$

Posons, pour chaque i $(1 \le i \le k)$,

$$
\sigma_{R,i} = \sigma_{\left(q_j \leftarrow q_j \prod_{m=j+1}^{N+M} q_m^{C_{R,m,i,j}}\right)}. \qquad (15)
$$

Les conditions sur les coefficients $C_{R,m,i,j}$ qui définissent les termes de croissance des paramètres, sont exactement celles qui font de chaque $\sigma_{R,i}$ une substitution de variables. L'équation (14) devient

$$
v(w,U) = \left(\prod_{m=1}^{N+M} q_m^{C_{R,m}}\right)\prod_{i=1}^{k}\sigma_{R,i}(v(w_i,U_i)). \qquad (16)
$$

Pour obtenir la série génératrice $U(q_1,\ldots,q_{N+M})$ d'un langage $L_G(U)$, il suffit alors de sommer suivant toutes les U-dérivations :

$$
\begin{aligned}
&U(q_1,\ldots,q_{N+M})\\
&\quad= \sum_{w\in L_G(U)} v(w,U)\\
&\quad= \sum_{R\in\mathcal{R}(U)}\left(\prod_{m=1}^{N+M} q_m^{C_{R,m}}\right)\sum_{w_i\in L_G(d(R,i))}\prod_{i=1}^{a(R)}\sigma_{R,i}(v(w_i,d(R,i))).
\end{aligned}
$$

Nous avons donc,

$$
U(q_1,\ldots,q_{N+M}) = \sum_{R\in\mathcal{R}(U)}\left(\prod_{m=1}^{N+M} q_m^{C_{R,m}}\right)\prod_{i=1}^{a(R)}\sigma_{R,i}\left(d(R,i)\right). \qquad (17)
$$

Le système formé des équations (17) (pour $U \in N$), est un Q-analogue du système donné par (10).

Inversement, étant donné un Q-analogue du système algébrique (10), il est toujours possible de l'écrire sous la forme (17). Notons à ce propos que, à chaque fois qu'une règle R présente dans son membre droit deux symboles identiques (c'est-à-dire $d(R,i) = d(R,j)$ avec $i \neq j$), les substitutions $(\sigma_{R,i})_{R\in\mathcal{R},1\leq i\leq a(R)}$ ne sont pas forcément déterminées de manière unique. Les substitutions $\sigma_{R,i}$ et $\sigma_{R,j}$ peuvent être échangées sans modifier le système d'équations; bien sûr, ceci peut avoir des conséquences sur l'interprétation qui sera donnée des paramètres d'énumération.

Une fois déterminées les substitutions $\sigma_{R,i}$, les coefficients $C_{R,m,i,j}$ sont obtenus en utilisant la formule (15). Ceci permet de reconstituer les règles de calcul des paramètres $p_{N+1}, \ldots, p_{N+M}$, et donc la Q-grammaire dont les séries génératrices vérifient le système d'équations donné. □

Remarque. Le théorème 2.36 donne en quelque sorte une justification *a posteriori* de la définition que nous avons adoptée des paramètres Q-comptables : il s'agit des paramètres qui peuvent être paramètres d'énumération d'un système qui soit un Q-analogue du système algébrique fourni par la grammaire elle-même. Ainsi, il aurait été possible de définir les paramètres Q-comptables en autorisant les combinaisons linéaires à coefficients entiers (et non seulement positifs), mais les équations obtenues auraient impliqué des exposants négatifs, et donc, parfois, la perte des séries génératrices comme séries formelles.

2.4 Grammaires linéaires et croissance polynômiale

2.4.1 Grammaires et langages linéaires

Un cas très particulier de grammaires algébriques est celui des grammaires *linéaires*. Une grammaire est dite linéaire, si le second membre de chaque règle de dérivation fait apparaître au plus un symbole non terminal.

Les langages engendrés par des grammaires linéaires sont appelés langages linéaires. Un exemple est $L = \{a^n b^n, n \geq 0\}$, engendré par la grammaire

$$\begin{cases} L & \to & \epsilon \\ L & \to & aLb. \end{cases}$$

Les langages linéaires, bien que ne se limitant pas aux langages rationnels comme le montre l'exemple ci-dessus, ont des séries génératrices rationnelles.

2.4.2 Paramètres à croissance polynômiale

Reprenons la définition d'un terme de croissance de paramètre telle qu'elle est donnée en 2.2.1. Pour définir les paramètres Q-comptables, nous avons exigé que tous les termes de croissance s'expriment de manière linéaire en fonction d'autres paramètres.

Définition 2.37. Soit G une grammaire d'alphabet $X = \{x_1, \ldots, x_k\}$.

Un paramètre p, défini sur G, est dit *à croissance polynômiale*, si, pour chaque règle de dérivation $R \in \mathcal{R}$, de la forme $R : U \to w_1 U' w_2$, il existe un polynôme P de k variables, à coefficients entiers positifs, tel que le terme de croissance de p pour le mot $w = w_1 w' w_2$, soit

$$\theta_p(w) = P\left(|w'|_{x_1}, \ldots, |w'|_{x_k}\right).$$

En d'autres termes, un paramètre à croissance polynômiale est un paramètre dont le terme de croissance, au lieu d'être défini par une combinaison linéaire de paramètres Q-comptables, est défini par un polynôme des nombres d'occurrences des différentes lettres. Les polynômes doivent avoir des coefficients positifs afin que les termes de croissance soient assurés d'être positifs. Cette condition, un peu restrictive, est également une condition suffisante pour que la preuve du lemme suivant ne fasse pas intervenir de coefficients négatifs.

Lemme 2.38. *Soit G une grammaire linéaire d'alphabet $X = \{x_1, \ldots, x_k\}$, et soit, pour chaque $i \leq k$, p_k le paramètre "nombre d'occurrences de la lettre x_k". Soit également $p = p_1^{\alpha_1} \ldots p_k^{\alpha_k}$ un monôme en les variables $p_1, \ldots, p_k$, de degré total $\alpha_1 + \ldots + \alpha_k$.*

Alors, p est un paramètre à croissance polynômiale, et les polynômes qui définissent ses termes de croissance sont tous de degré total strictement inférieur à $\alpha_1 + \ldots + \alpha_k$.

Preuve. Lors de l'application de la règle $R : U \to w_1 U' w_2$, la croissance de chaque paramètre p_i est une constante : $p_i(w) - p_i(w') = |w_1 w_2|_{x_i} = a_i$. Le terme de croissance de p pour R est donc

$$\begin{aligned} \theta_p &= p(w) - p(w') \\ &= (p_1(w') + a_1)^{\alpha_1} \ldots (p_k(w') - a_k)^{\alpha_k} - (p_1(w'))^{\alpha_1} \ldots (p_k(w'))^{\alpha_k} \end{aligned}$$

En développant cette dernière expression, on trouve bien un polynôme des variables $p_1(w'), \ldots, p_k(w')$, à coefficients entiers positifs; le seul terme de degré total $\alpha_1 + \ldots + \alpha_k$ ayant été annulé, ce polynôme est de degré total strictement inférieur. □

Une conséquence de ce lemme est la proposition suivante :

Proposition 2.39. *Tout paramètre à croissance polynômiale dans une grammaire linéaire, est un paramètre Q-comptable de cette grammaire; le rang de ce paramètre Q-comptable est au plus supérieur de 1 au degré total maximal des polynômes qui définissent ses termes de croissance.*

Preuve. Nous procédons par récurrence sur le degré total maximal des polynômes P qui définissent les termes de croissance.

Si tous ces polynômes sont de degré total au plus 1, alors le paramètre p est déjà défini comme un paramètre Q-comptable de rang 2 (à moins que tous les polynômes ne soient constants, auquel cas p est de rang 1).

Supposons maintenant que la propriété soit vraie lorsque tous les polynômes qui définissent les termes de croissance sont de degré total inférieur ou égal à n. Alors, d'après le lemme précédent, tous les polynômes qui définissent les termes de croissance du paramètre p, peuvent eux-mêmes se définir par des termes de croissance qui sont des polynômes de degré total strictement inférieur à n. D'après l'hypothèse de récurrence, chacun des monômes de degré inférieur à n peut s'écrire comme paramètre Q-comptable de rang au plus n dans G; par conséquent, le terme de croissance de p s'écrit bien comme combinaison linéaire de paramètres Q-comptables de rang au plus n, et p est lui-même Q-comptable de rang au plus $n+1$. □

La proposition 2.39 n'a pas d'équivalent pour les grammaires non linéaires. Pour s'en convaincre, il suffit d'examiner l'exemple suivant :

Exemple 2.40. Reprenons la grammaire classique des mots de Dyck :

$$\begin{cases} R' : & D & \to & \epsilon, \\ R : & D & \to & aDbD. \end{cases}$$

Les arbres de dérivation de cette grammaire sont équivalents aux arbres binaires complets non étiquetés (toutes les feuilles sont étiquetées R', et tous les nœuds internes sont étiquetés R); comme, dans tout arbre binaire complet, le nombre de feuilles est supérieur de 1 au nombre de nœuds internes, on a pour tout mot de Dyck w,

$$p_R(w) = 1 + p_{R'}(w).$$

Par conséquent, tout paramètre Q-comptable dans G peut s'exprimer en n'utilisant que les paramètres élémentaires ne faisant intervenir que la règle R : ainsi, $p_{R^{(1)}R^{(2)}R'} = p_{R^{(1)}R^{(2)}R} + p_{R^{(1)}R}$.

Notons $l(w) = p_R(w)$ (le paramètre l est la demi-longueur du mot w), et soit p le paramètre défini par les règles de croissances polynômiales suivantes :

$$\begin{aligned} p(\epsilon) &= 0 \\ p(aw_1bw_2) &= p(w_1) + p(w_2) + l(w_1).l(w_2) \end{aligned}$$

Le paramètre p est la somme, sur tous les nœuds internes de l'arbre de dérivation, du produit des tailles (nombres de nœuds internes) des sous-arbres gauche et droit. Afin de prouver qu'un tel paramètre n'est pas Q-comptable, nous nous contentons d'établir un tableau des premières valeurs des différents paramètres :

w	R	$R^{(1)}R$	$R^{(2)}R$	$R^{(1)}R^{(1)}R$	$R^{(1)}R^{(2)}R$	$R^{(2)}R^{(1)}R$	$R^{(2)}R^{(2)}R$	$p(w)$
ab	1	0	0	0	0	0	0	0
$aabb$	2	1	0	0	0	0	0	0
$abab$	2	0	1	0	0	0	0	0
$aaabbb$	3	3	0	1	0	0	0	0
$aababb$	3	2	1	0	1	0	0	0
$abaabb$	3	1	2	0	0	1	0	0
$ababab$	3	0	3	0	0	0	1	0
$aabbab$	3	1	1	0	0	0	0	1

Il apparaît, au vu de ce tableau, que la dernière colonne ne saurait être combinaison linéaire des précédentes. Ajouter d'autres colonnes, correspondant à des paramètres Q-comptables de rangs plus élevés, n'y changerait rien : les lignes correspondant aux mots déjà inscrits seraient toutes nulles, puisque des paramètres de rangs supérieurs à 3 compteraient des chaînes de longueur supérieure à 3 (et l'arbre de dérivation de *aabbab* n'est que de profondeur 3). Tout simplement, le paramètre p n'est pas Q-comptable dans la grammaire G.

2.5 Résolution de Q-équations

Il n'existe aucune méthode générale pour résoudre les équations données par une Q-grammaire dans le cas général. Des méthodes *ad hoc* pour résoudre une équation faisant intervenir la substitution $\sigma_{x \leftarrow xq}$ existent cependant dans la littérature; nous en donnons ici un exemple, que nous adaptons à des substitutions légèrement plus générales.

2.5.1 La méthode de Prellberg et Brak

Dans [69], Prellberg et Brak donnent une méthode pour résoudre, sous certaines conditions, une équation de la forme

$$F.\sigma(F) + aF + b\sigma(F) + c = 0, \tag{18}$$

où a, b et c appartiennent à $\mathbb{K}[[x,q]]$. En posant $G = F + b$, (18) devient

$$G.\sigma(G) + a'G + c' = 0, \tag{19}$$

avec $a' = a - \sigma(b)$ et $c' = c - a.b$. L'équation (19) peut à son tour être linéarisée en posant

$$G = \alpha\frac{\sigma(H)}{H}, \tag{20}$$

α (qui ne dépend plus de x) devant être choisi de manière à assurer la compatibilité pour $x = 0$ et $q = 0$, c'est-à-dire, α doit être solution de $\alpha^2 + \alpha.a'(0,0) + c'(0,0) = 0$. L'équation (19) devient alors,

$$\alpha^2\sigma^2(H) + \alpha a'\sigma(H) + c'H = 0. \tag{21}$$

Lorsque la substitution σ est $\sigma_{x\leftarrow xq}$, Prellberg et Brak obtiennent un développement de H suivant les puissances de x dans le cas où c' ne dépend pas de x, et où a' est de degré 1 en x. Dutour [36] donne une version plus générale, où c' peut également être de degré 1 en x.

L'idée est de réécrire (21) sous la forme plus générale

$$\sum_{k=0}^{N} (\alpha_k + x.\beta_k)\,\sigma^k(H) = 0, \tag{22}$$

où les coefficients α_k et β_k ne dépendent plus de x.

En développant la série inconnue H comme série en x : $H(x) = \sum_n x^n h_n(q)$, et en extrayant le coefficient de x^n dans l'équation (22), nous obtenons alors la récurrence suivante :

$$h_n.\sum_{k=0}^{N} \alpha_k q^{kn} + h_{n-1}.\sum_{k=0}^{N} \beta_k q^{k(n-1)} = 0. \tag{23}$$

Une fois posé

$$\begin{cases} \Lambda_\alpha(t) &= \sum_{k=0}^{N} \alpha_k t^k, \\ \Lambda_\beta(t) &= \sum_{k=0}^{N} \beta_k t^k, \end{cases}$$

l'équation (23) a pour solution (en prenant $h_0 = 1$)

$$h_n = (-1)^n \frac{\prod_{i=0}^{n-1} \Lambda_\beta(q^i)}{\prod_{i=0}^{n-1} \Lambda_\alpha(q^{i+1})}. \tag{24}$$

Notons ici que la condition de compatibilité sur α se traduit par $\Lambda_\alpha(1) = 0$, condition retrouvée en extrayant le terme constant (en x) de (22).

Par ailleurs, notons qu'il existe un résultat plus général permettant de résoudre des équations linéarisées similaires à (21), mais faisant également intervenir l'image de la série inconnue H par la spécialisation $x = 1$; voir à ce sujet [14] et une variante dans [7].

2.5.2 Une extension de la méthode

Il serait agréable de pouvoir étendre la méthode ci-dessus à des équations sur des séries à plus de deux variables et faisant intervenir des substitutions de variables quelconques, voire plusieurs substitutions différentes.

Attaquer le problème des substitutions quelconques semble largement hors de portée; toutefois, il est possible d'étendre la résolution de l'équation (22) au cas où la série H est une série à plusieurs variables $H \in \mathbb{K}[[x_1, \dots x_k; q_1, \dots q_\ell]]$, à condition que la substitution σ ne corresponde qu'au calcul de paramètres de rang 2, c'est-à-dire qu'elle soit de la forme

$$\sigma = \sigma_{(x_i \leftarrow x_i q_1^{a_{i,1}} \dots q_\ell^{a_{i,\ell}})}.$$

Cette substitution est entièrement déterminée par la matrice $\mathbf{A} = (a_{i,j})_{1 \le i \le k, 1 \le j \le \ell}$ (qui n'est qu'une sous-matrice particulière de la matrice M_σ définie au paragraphe 2.3.1).

Nous traitons complètement le cas $k = \ell = 2$; le cas le plus général n'est pas plus difficile, mais demande des notations plus lourdes.

L'équation à résoudre se présente alors sous la forme

$$\sum_{i=0}^{N} (\alpha_i + x_1 \beta_{1,i} + x_2 \beta_{2,i}) \sigma^i(H) = 0. \tag{25}$$

Nous notons $\mathbf{x}$ le vecteur-ligne (x_1, x_2), et, par convention, $\mathbf{x}^\mathbf{n} = x_1^{n_1} x_2^{n_2}$. Lorsque $\mathbf{x}$ est un tel vecteur-ligne, $\mathbf{x}'$ désigne le vecteur-colonne transposé.

En posant

$$\begin{cases} H(\mathbf{x}, \mathbf{q}) &= \sum_{n_1, n_2 \ge 0} \mathbf{x}^\mathbf{n} h_\mathbf{n}(\mathbf{q}), \\ \Lambda_\alpha(t) &= \sum_{0 \le i \le N} \alpha_i t^i, \\ \Lambda_{\beta_j}(t) &= \sum_{0 \le i \le N} \beta_{j,i} t^i, \end{cases}$$

nous avons

$$\sigma^i(H) = \sum_{n_1,n_2\geq 0} \mathbf{x}^{\mathbf{n}} \mathbf{q}^{i\mathbf{A}.\mathbf{n}'} h_{\mathbf{n}}(\mathbf{q}). \tag{26}$$

Ainsi, en extrayant le coefficient de $\mathbf{x}^{\mathbf{n}}$ de (25), nous obtenons la relation de récurrence

$$\sum_{i=0}^{N} \left(\alpha_i \mathbf{q}^{i\mathbf{A}.\mathbf{n}'} h_{\mathbf{n}} + \beta_{1,i} \mathbf{q}^{i\mathbf{A}.(\mathbf{n}-(1,0))'} h_{\mathbf{n}-(1,0)} + \beta_{2,i} \mathbf{q}^{i\mathbf{A}.(\mathbf{n}-(0,1))'} h_{\mathbf{n}-(0,1)} \right) = 0. \tag{27}$$

Dans la relation ci-dessus, nous avons par convention $h_{n_1,-1} = h_{-1,n_2} = 0$.

Nous pouvons donc écrire h_{n_1,n_2} en fonction de h_{n_1-1,n_2} et de h_{n_1,n_2-1} :

$$h_{n_1,n_2} = -\frac{\Lambda_{\beta_1}\left(\mathbf{q}^{\mathbf{A}.(n_1-1,n_2)'}\right) h_{n_1-1,n_2} + \Lambda_{\beta_2}\left(\mathbf{q}^{\mathbf{A}.(n_1,n_2-1)'}\right) h_{n_1,n_2-1}}{\Lambda_{\alpha}\left(\mathbf{q}^{\mathbf{A}.\mathbf{n}'}\right) h_{\mathbf{n}}}. \tag{28}$$

Nous pouvons ainsi interpréter les coefficients $h_{\mathbf{n}}$ en termes de *chemins dirigés* :

Définition 2.41. Soient A et B deux points du plan discret $\mathbb{N} \times \mathbb{N}$. Un *chemin dirigé* de A à B est un chemin discret $S = (s_i)_{0\leq i\leq n}$ du plan $\mathbb{N}^2$, avec $s_0 = A$, $s_n = B$, et ne faisant que des pas Nord ($s_{i+1} - s_i = (0,1)$) et Est ($s_{i+1} - s_i = (1,0)$).

Nous notons $P_{A,B}$ l'ensemble des chemins dirigés de A à B.

Pour chaque chemin dirigé, nous définissons une valuation en utilisant les notations introduites précédemment :

Définition 2.42. Soit $S = (s_0, \ldots, s_n)$ un chemin dirigé de $A = s_0$ à $B = s_n$. Notons, pour $0 \leq i \leq n$, (n_i, m_i) les coordonnées de s_i.

La *valuation* du pas $s_i s_{i+1}$ est la série

$$v(s_i, s_{i+1}) = -\frac{\Lambda_{\beta_j}\left(\mathbf{q}^{\mathbf{A}.(n_i,m_i)'}\right)}{\Lambda_{\alpha}\left(\mathbf{q}^{\mathbf{A}.(n_{i+1},m_{i+1})'}\right)}, \tag{29}$$

où $j = 1$ si le pas $s_i s_{i+1}$ est un pas Est, et $j = 2$ si $s_i s_{i+1}$ est un pas Nord.

La *valuation* du *chemin* S est

$$v(S) = \prod_{i=0}^{n-1} v(s_i, s_{i+1}). \tag{30}$$

La récurrence (28) se traduit alors immédiatement de la manière suivante :

Proposition 2.43. *Le coefficient h_{n_1,n_2} de la série H est*

$$h_{n_1,n_2} = \sum_{S\in P_{O,(n_1,n_2)}} v(S). \tag{31}$$

Remarque. En définissant une seconde valuation $v'(S) = v(S)\mathbf{x^n}$, où n_1 et n_2 sont les coordonnées du dernier point de S, c'est la série H toute entière qui s'exprime comme somme de valuations de chemins :

$$H(\mathbf{x}, \mathbf{q}) = \sum_{S \in P_O} v'(S), \tag{32}$$

où P_O désigne l'ensemble de tous les chemins dirigés d'origine O.

Il semble difficile d'étendre ce genre de méthodes à des équations faisant intervenir des paramètres de rang supérieur à 2. Ainsi, dans le cas des chemins de Dyck énumérés suivant les paramètres longueur (comptée par x), aire géométrique (comptée par q) et moment d'inertie (compté par r; voir exemple 2.7), l'équation obtenue est :

$$D(x, q, r) = 1 + x^2 qr D(x, q, r) D(xqr, qr, r).$$

La linéarisation de l'équation est toujours possible en posant

$$D(x, q, r) = F(xqr, qr, r)/F(x, q, r),$$

et l'équation obtenue est alors :

$$x^2 qr F(xq^2r^3, qr^2, r) - F(xqr, qr, r) + F(x, q, r) = 0.$$

Toutefois, bien que l'équation soit en quelque sorte linéaire, développer F suivant les puissances de x ne fait pas disparaître complètement les substitutions de variables. Si nous posons

$$F(x, q, r) = \sum_{n=0}^{\infty} x^{2n} f_n(q, r),$$

l'équation se traduit, pour $n \geq 1$, par une relation de récurrence faisant intervenir la substitution de variables $\sigma_{q \leftarrow qr}$:

$$q^{4n-3} r^{6n-5} f_{n-1}(qr^2, r) - q^{2n} r^{2n} f_n(qr, r) + f_n(q, r) = 0.$$

La condition initiale $f_0(q, r) = 1$ permet de résoudre pour $n = 1$, et l'on obtient alors

$$f_1(q, r) = -\sum_{k=0}^{\infty} q^{2k+1} r^{(k+1)^2};$$

toutefois, pousser le calcul ne serait-ce qu'un cran plus loin amène à développer suivant les puissances de q, et la récurrence double obtenue ne semble pas pouvoir se résoudre de manière générale.

Chapitre 3

Changements de grammaires

Dans ce chapitre, nous étudions dans quelle mesure les paramètres Q-comptables associés à un langage dépendent de la grammaire utilisée pour l'engendrer. Nous montrons également qu'il est possible de faire subir un certain nombre de transformations classiques à une grammaire, sans perte de paramètres Q-comptables.

Nous avons vu que certains paramètres, comme la longueur des mots et ses variantes (nombre d'occurrences d'une lettre donnée ou de certaines des lettres de l'alphabet) sont toujours des paramètres Q-comptables, indépendamment de la grammaire.

Nous montrons ici que cette situation n'est pas générale. Ainsi, un paramètre peut être Q-comptable dans une grammaire, mais pas dans une autre qui engendre le même langage. Nous commençons par un exemple de deux paramètres qui sont Q-comptables dans deux grammaires différentes, sans qu'il existe de grammaire dans laquelle ils soient tous deux Q-comptables.

3.1 Un exemple: nombre de passages au niveau final ou initial

Considérons le langage $\{a, b\}^*$, formé de tous les mots sur les deux lettres a et b.

Définition 3.1. Un mot $w \in \{a, b\}^*$ est dit *équilibré* si $|w|_a = |w|_b$.

Les mots équilibrés sont aussi appelés mots de Dyck bilatères. Soient p et p' les paramètres définis sur $\{a, b\}^*$ par : $p(w)$ est le nombre de facteurs gauches équilibrés de w, et $p'(w)$ est le nombre de ses facteurs droits équilibrés.

Par symétrie, il est clair que p et p' ont la même distribution; par ailleurs, si p est Q-comptable dans une grammaire G, p' est forcément Q-comptable dans la grammaire G'

obtenue en remplaçant par leur image miroir tous les membres droits de toutes les règles de G.

A chaque mot, associons un chemin du plan discret en transformant chaque a en un pas Nord-Est et chaque b en un pas Sud-Est (représentation horizontale classique d'un mot de Dyck par un chemin de Dyck). Le paramètre p devient le nombre de sommets du chemin dont l'ordonnée est nulle (passages au niveau initial), et p', le nombre de sommets dont l'ordonnée est celle du dernier sommet (passages au niveau final) – voir figure 3.1. Il apparaît intuitivement qu'il est possible de "marquer" les passages au niveau initial en lisant le mot "de gauche à droite", ou les passages au niveau final en le lisant "de droite à gauche"; nous allons montrer de manière précise qu'il est impossible de marquer les deux types de passages.

$w = aabaabbbbaabbbaaaa$
$p(w) = 5$
$p'(w) = 4$

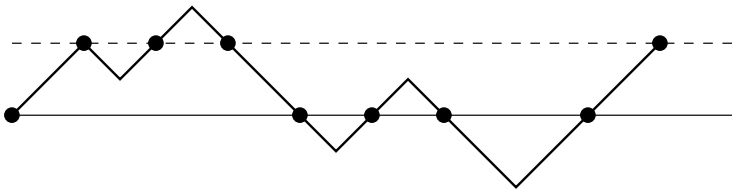


FIG. 3.1: *Passages aux niveaux initial et final*

Nous allons montrer que p peut être Q-comptable, mais que p et p' ne peuvent pas être Q-comptables dans la même grammaire. Puisque la somme de deux paramètres Q-comptable est Q-comptable, il nous suffit de montrer que $p + p'$ n'est Q-comptable dans aucune grammaire.

Soit $G = (\{a, b\}, \{T, P, N, F, G\}, \mathcal{R}, T)$ la grammaire définie par ses règles de dérivation :

$$\begin{aligned} T &\rightarrow \epsilon + aPbT + bNaT + F + G \\ P &\rightarrow \epsilon + aPbP \\ N &\rightarrow \epsilon + bNaN \\ F &\rightarrow aP + aPF \\ G &\rightarrow bN + bNG \end{aligned}$$

Cette grammaire engendre le langage $\{a, b\}^*$, et le paramètre p compte les T-dérivations; il est donc Q-comptable de rang 1 dans cette grammaire.

Supposons que, dans une grammaire G', le paramètre $p+p'$ soit Q-comptable. Il est alors de rang minimal 1. Nous montrerons plus tard, et nous l'admettrons pour le moment, que lorsqu'un paramètre est de rang minimal k, il existe toujours une grammaire engendrant le même langage et dans laquelle le même paramètre est de rang k. Dans le cas du paramètre $p+p'$, la proposition 2.16 implique alors que le paramètre $p+p'$ peut être représenté par une lettre c supplémentaire. Afin de montrer que $p+p'$ ne peut être Q-comptable, il nous suffit donc de montrer le lemme suivant :

Lemme 3.2. *Il n'existe pas de langage algébrique non ambigu $L \subset \{a,b,c\}^*$ tel que*

- *la projection φ de $\{a,b,c\}^*$ sur $\{a,b\}^*$ établit une bijection de L sur $\{a,b\}^*$;*
- *pour tout mot $w \in L$, $|w|_c = (p+p')(\varphi(w))$.*

Preuve. Supposons qu'un tel langage algébrique L existe. Pour chaque mot $w \in L$, nous noterons $w' = \varphi(w)$ le mot obtenu en effaçant les c de w. Lorsque w' est défini, w désigne alors son antécédent par φ dans L.

Nous utilisons le lemme d'Ogden sur les langages algébriques (voir [65, 56, 1, 11]). Soit donc $w'_1 = a^{2N}b^Na^{2N}$, avec N assez grand pour appliquer le lemme au mot w_1 correspondant. Nous avons alors une factorisation $w_1 = \alpha u\beta v\gamma$ telle que u' (ou v', ce qui est équivalent par image miroir) soit de la forme $u' = b^k$, et tel que, pour tout $n \geq 0$, $w_n = \alpha u^{n+1}\beta v^{n+1}\gamma \in L$.

Il est facile de voir que $(p+p')(w'_1) = 2$; par conséquent, $|w_n|_c = 2 + n|uv|_c$. Sachant que $|uv|_c$ vaut 0, 1 ou 2, la suite $(|w_n|_c)_{n\geq 0}$, soit est constante, soit tend vers $+\infty$. Nous allons montrer que $(p+p')(w'_n)$ ne peut avoir ce comportement.

Les différents cas possibles pour α', u', β', v' et γ', également représentés figure 3.2, sont les suivants :

a. $\alpha' = a^{2N}b^{k_1}$, $u' = b^{k_2}$, $\beta' = b^{k_3}$, $v' = b^{k_4}$, $\gamma' = b^{k_5}a^{2N}$ avec $k_1+k_2+k_3+k_4+k_5 = N$ et $k_2+k_4 > 0$. Dans ce cas, $w'_n = a^{2N}b^{N+n(k_2+k_4)}a^{2N}$ et donc, pour n assez grand, $(p+p')(w'_n) = 4$.

b. $\alpha' = a^{2N}b^{k_1}$, $u' = b^{k_2}$, $\beta' = b^{k_3}a^{k_4}$, $v' = a^{k_5}$, $\gamma' = a^{k_6}$ avec $k_1+k_2+k_3 = N$, $k_4+k_5+k_6 = 2N$, et $k_2 > 0$. Dans ce cas, $w'_n = a^{2N}b^{N+nk_2}a^{2N+nk_5}$. Pour n assez grand, $(p+p')(w'_n)$ vaut 4 (ou 6 si $k_2 = k_5$).

c. $\alpha' = a^{2N}b^{k_1}$, $u' = b^{k_2}$, $\beta' = b^{k_3}$, $v' = b^{k_4}a^{k_5}$, $\gamma' = a^{k_6}$ avec $k_1+k_2+k_3+k_4 = N$, $k_5+k_6 = 2N$, et $k_2+k_4 > 0$. Dans ce cas, $w'_n = a^{2N}b^{N+nk_2}(a^{k_5}b^{k_4})^{n-1}a^{2N}$. Si $k_5 \geq k_4$, $(p+p')(w'_n) = 4$ lorsque n est assez grand; si $k_4 > k_5$, $(p+p')(w'_n)$ est ultimement constante avec une valeur plus grande que 4 (voir figure 3.2).

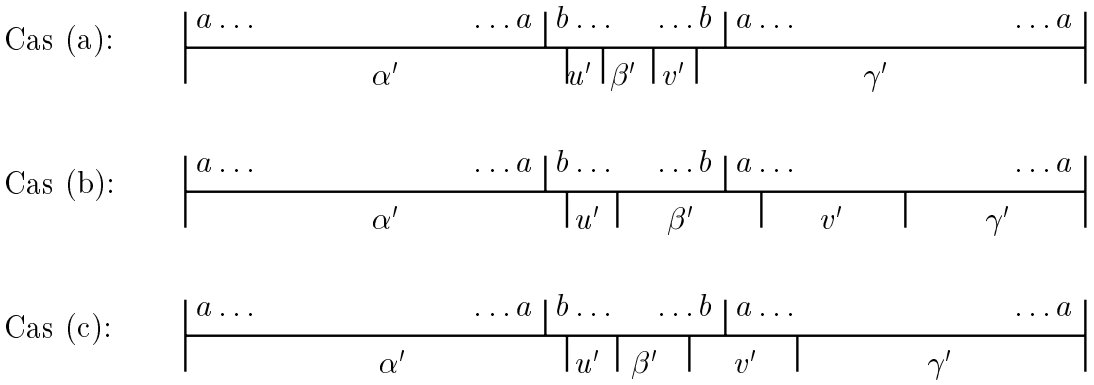


FIG. 3.2: *Différents cas*

Dans tous les cas, le comportement de $(p + p')(w'_n)$ est différent de celui de $|w_n|_c$, ce qui donne une contradiction et termine donc la preuve du lemme. □

Une conséquence de ce lemme (et de la propriété 2.16) est qu'il n'existe pas de grammaire non ambiguë engendrant $\{a, b\}^*$ dans laquelle $p+p'$ soit un paramètre Q-comptable. Par conséquent, puisque la somme de deux paramètres Q-comptables dans la même grammaire est Q-comptable dans cette grammaire, les paramètres p et p' ne peuvent être Q-comptables dans la même grammaire.

3.2 Grammaire plus fine qu'une autre

Comme le prouve l'exemple précédent, un paramètre peut être Q-comptable dans une grammaire, mais pas dans une autre qui engendre le même langage.

Définition 3.3. Soient G et G' deux grammaires engendrant le même langage L. G' est *plus fine* que G si, pour tout paramètre p, Q-comptable dans G, il existe un paramètre p', Q-comptable dans G', tel que, pour tout mot $w \in L$, $p(w) = p'(w)$.

Définition 3.4. Deux grammaires G et G' sont *Q-équivalentes* si G est plus fine que G' et réciproquement.

Remarque. Dire qu'une grammaire est plus fine qu'une autre revient à dire que tout paramètre Q-comptable de la seconde coïncide, sur le langage engendré, avec un paramètre Q-comptable de la première. Dire que tout paramètre Q-comptable de l'une est Q-comptable dans l'autre serait un abus de langage, les paramètres étant a priori définis sur tous les langages auxiliaires. Une telle définition serait trop restrictive et perdrait son intérêt

pratique, car cela exigerait que les langages auxiliaires des deux grammaires soient les mêmes.

Remarque. Dans le cadre des grammaires d'objets, Dutour [36] donne une définition de l'isomorphisme de grammaires d'objets qui est basée sur l'identité de systèmes caractéristiques de polynômes. Notre définition de la Q-équivalence de grammaires est plus restrictive, et les résultats sur les grammaires d'objets ne semblent pas pouvoir se transposer dans notre cadre. En effet, notre définition de la Q-équivalence de grammaires impose que les paramètres soient conservés pour chaque mot, et non pas pour le langage engendré pris dans son ensemble. Nous verrons sur un exemple (voir section 3.3) que la Q-équivalence de deux grammaires ne se réduit pas à l'isomorphisme de deux grammaires d'objets.

Il est clair qu'une condition nécessaire et suffisante pour que G' soit plus fine que G est que tout paramètre *élémentaire* de G, lorsqu'on ne le considère que sur les mots du langage engendré par G et G', soit Q-comptable dans G'.

L'exemple précédent nous montre qu'il est possible d'avoir deux grammaires engendrant le même langage, sans qu'il existe une grammaire plus fine que chacune d'elles.

Nous donnons maintenant quelques lemmes qui permettent, étant donnée une grammaire, d'en construire d'autres plus fines.

Etant donnée une grammaire G, nous allons montrer qu'il est possible de construire :

- une grammaire plus fine, de forme 1-2;
- une grammaire plus fine, où l'on a itéré l'une des règles de dérivation;
- une grammaire plus fine, où chaque sommet des arbres de dérivation est également marqué suivant la présence ou non, dans la branche qui le relie à la racine, de certaines étiquettes (marquage supérieur);
- une grammaire plus fine, où chaque sommet est marqué suivant l'existence, dans l'un de ses sous-arbres, de certaines étiquettes (marquage inférieur);
- une grammaire plus fine, où le rang formel d'un ou de chaque paramètre de G est égal à son rang minimal.

3.2.1 Passage en forme 1-2

Une grammaire est dite de forme 1-2 si toutes ses règles de dérivation sont d'arité 0, 1 ou 2. Le fait que tout langage algébrique non ambigu peut être engendré par une grammaire

de forme 1-2 est bien connu. Nous montrons essentiellement que le passage en forme 1-2 se fait en transformant la grammaire en une grammaire Q-équivalente.

L'outil de base est la transformation décrite dans la proposition suivante, qui diminue strictement le nombre de règles de dérivation d'arité supérieure à 2.

Proposition 3.5. *Soit G une grammaire engendrant un langage L, et soit $R : U \to u_0U_1u_1 \dots U_ku_k$ une règle de dérivation de G, d'arité $k \geq 3$.*

Soit G' la grammaire obtenue en ajoutant $k-2$ symboles $V_1, \dots, V_{k-2}$, et en remplaçant la règle R par

$$\begin{aligned} (R'_0): \quad U &\to u_0U_1u_1V_1, \\ (R'_i): \quad V_i &\to U_{i+1}u_{i+1}V_{i+1} \quad (1 \leq i < k-2), \\ (R'_{k-2}): \quad V_{k-2} &\to U_{k-1}u_{k-1}U_ku_k. \end{aligned}$$

La grammaire G' engendre également L, et G et G' sont Q-équivalentes.

Preuve. Le passage de G à G' est une opération classique utilisée pour donner une grammaire en forme 1-2. Tout arbre de dérivation de G peut facilement être transformé en l'arbre de dérivation correspondant de G' en remplaçant les sommets étiquetés R par des sous-arbres étiquetés R', tout en rattachant les sous-arbres issus de ces sommets (voir figure 3.3).

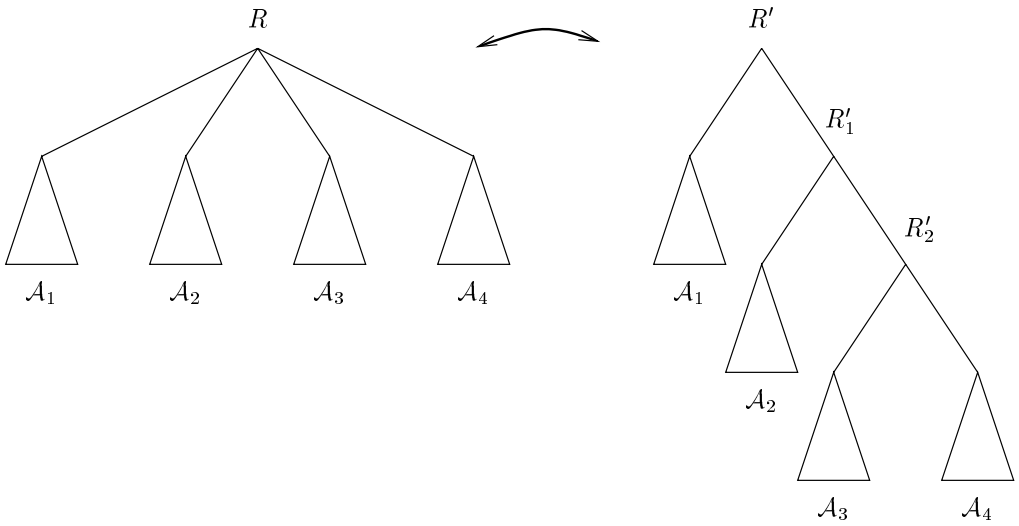


FIG. 3.3: *Passage en forme* 1-2

Cette transformation n'a aucune influence sur les sommets qui ne sont pas étiquetés R, donc les paramètres élémentaires de G qui ne font pas intervenir la règle R ne sont pas modifiés.

Afin de montrer que les grammaires G et G' sont bien Q-équivalentes, nous indiquons comment "traduire" dans G' le nom d'un paramètre élémentaire de G, et, inversement, comment "traduire" dans G le nom d'un paramètre élémentaire de G'.

Le passage de G à G' est extrêmement simple, puisqu'il suffit d'appliquer sur un nom de paramètre un morphisme alphabétique :

$$\begin{array}{rcll} \varphi(R) & = & R'_0, & \\ \varphi(R^{(i)}) & = & R'^{(1)}_{i-1} & (1 \leq i < k), \\ \varphi(R^{(k)}) & = & R'^{(2)}_{k-1}, & \\ \varphi(r) & = & r & (\text{pour tout autre } r \in \mathcal{R}_p \cup \mathcal{R}). \end{array}$$

Quel que soit le mot $W \in \mathcal{R}_p^*\mathcal{R}$, et quel que soit l'arbre de dérivation $\mathcal{A}$ de G, les chaînes de type W de $\mathcal{A}$ sont en bijection avec les chaînes de type $\varphi(W)$ de l'arbre de dérivation de G' correspondant à $\mathcal{A}$. Par conséquent, G' est plus fine que G.

Inversement, la transformation est légèrement plus compliquée, car un paramètre élémentaire de G' peut avoir dans G un équivalent qui n'est pas un paramètre élémentaire.

Nous noterons $\mathcal{R}'_0 = \varphi(\mathcal{R}_p \cup \mathcal{R})$, et $\mathcal{R}'_1 = (\mathcal{R}'_p \cup \mathcal{R}') - \mathcal{R}'_0$. L'ensemble $\mathcal{R}'_1$ est composé des dérivations pointées $R'^{(2)}_i$, pour $0 \leq i \leq k-1$, et des dérivations R'_i, pour $1 \leq i \leq k-2$.

Soit W' un nom de paramètre élémentaire dans G'. Si W' ne fait intervenir aucune des lettres de $\mathcal{R}'_1$, W' fait déjà partie des noms de paramètres qui correspondent à des paramètres élémentaires de G. Supposons donc que W' contient de telles lettres.

Afin de "traduire" un nom de paramètre élémentaire de G' vers G, nous définissons $\overline{\varphi}$ de la manière suivante :

- $\overline{\varphi}(r) = \varphi^{-1}(r)$ si $r \in \mathcal{R}'_0$;
- $\overline{\varphi}(R'_i) = R$;
- $\overline{\varphi}(R'^{(2)}_i) = R^{(i+2)} + \cdots + R^{(k)}$ (pour $0 \leq i < k-2$).

Soit, dans un arbre de dérivation $\mathcal{A}'$ de G', une chaîne S' de sommets, de type W'. Dans l'arbre de dérivation $\mathcal{A}$ de G qui correspond à $\mathcal{A}'$, cette chaîne devient une chaîne S, qui peut être plus courte que S', car plusieurs sommets de S' peuvent correspondre à un seul et même sommet de S – dans le second arbre de la figure 3.3, les sommets étiquetés R'_0, R'_1 et R'_2 proviennent du même sommet étiqueté R, et donc, si plusieurs d'entre eux font partie d'une chaîne de sommets de $\mathcal{A}'$, la chaîne de sommets correspondante dans $\mathcal{A}$ sera plus courte. Toutefois, si S est de même longueur que S', le type de S est l'un des mots de $\overline{\varphi}(W')$.

Afin de rendre compte de tous les cas possibles, nous dirons qu'une lettre de W' est *non maximale* si elle est de la forme $R_i'^{(2)}$ et est suivie, dans W', par une lettre de la forme $R_j'^{(1)}$, $R_j'^{(2)}$ ou R_j', avec $i < j$. Ainsi, une lettre d'un nom de paramètre W' est non maximale s'il est possible que, dans une chaîne de type W', les sommets correspondant à cette lettre et à la suivante proviennent d'un même sommet de l'arbre $\mathcal{A}$.

La figure 3.4 montre un exemple avec $k = 3$. L'arbre $\mathcal{A}'$ (représenté en haut) possède 7 chaînes de type $W' = R_0'^{(2)} R_1'^{(1)} r$, dont 3 seulement sont représentées. Dans le nom W', la lettre $R_0'^{(2)}$ est non maximale : cela se traduit par le fait que, dans l'arbre $\mathcal{A}$, les deux premiers sommets d'une chaîne de type W' peuvent n'en former qu'un. Dans ce cas, le type de la chaîne de $\mathcal{A}$ est $R^{(2)}r$. Sinon, le type est soit $R^{(2)}R^{(2)}r$, soit $R^{(3)}R^{(2)}r$.

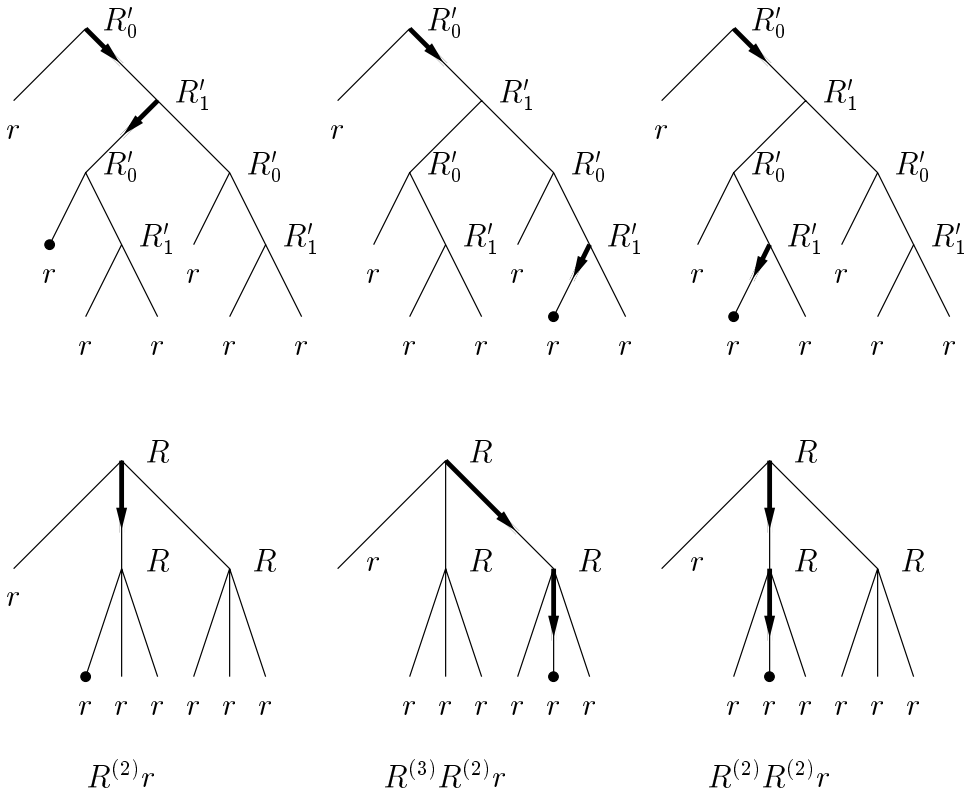


FIG. 3.4: *Transformation de chaînes de type* $R_0'^{(2)} R_1'^{(1)} r$

Au total, pour obtenir le nom du paramètre Q-comptable correspondant, dans G, au paramètre $p_{W'}$ de G', il faut donc traduire chaque lettre maximale r de W' par $\overline{\varphi}(r)$, et chaque lettre non maximale r par $\epsilon + \overline{\varphi}(r)$.

□

La traduction d'un nom de paramètre de G' en nom de paramètre de G peut également

être définie de la manière suivante: un ordre partiel est défini sur $\mathcal{R}'_p \cup \mathcal{R}'$ par

$$\begin{cases} R_i'^{(2)} < R'_j & \Longleftrightarrow \quad i \leq j, \\ R_i'^{(2)} < R_j'^{(k)} & \Longleftrightarrow \quad i < j \ et \ 1 \leq k \leq 2. \end{cases}$$

Notons que, dans un nom de paramètre W' de G', une lettre est non maximale au sens de la preuve précédente lorsqu'elle est inférieure (strictement) à celle qui la suit dans W.

La fonction τ qui à tout nom de paramètre W' de G' associe le nom $\tau(W')$ de ce même paramètre dans G, peut alors être définie récursivement ainsi:

$$\begin{cases} \tau(r') & = & \overline{\varphi}(r') & \text{si } r' \in \mathcal{R}', \\ \tau(r'W'_1W') & = & \left(\overline{\varphi}(r') + 1_{r'<W'_1}\right)\tau(W'_1W') & \text{si } r' \in \mathcal{R}'_p, W'_1 \in \mathcal{R}'_p \cup \mathcal{R}'. \end{cases}$$

3.2.2 Itération d'une règle

Une autre opération permettant de changer de grammaire sans changer le langage engendré, est l'*itération* qui consiste à remplacer, dans le membre droit d'une règle, l'un des symboles par tous les membres droits des règles qui réécrivent ce symbole.

Proposition 3.6. *Soit G une grammaire, R une de ses règles d'arité $n \geq 1$, et $k \leq n$. Soit G' la grammaire engendrant le même langage, obtenue en retirant la règle R et en ajoutant, pour chaque $d(R,k)$-dérivation R', la règle obtenue en remplaçant, dans le membre droit de R, le k-ème symbole par le membre droit de R'.*

Dans ces conditions, la grammaire G' est plus fine que G; plus précisément, tout paramètre Q-comptable dans G est également Q-comptable dans G'.

Remarque. Une preuve simple de la proposition 3.6 consiste à remarquer que n'importe quel Q-analogue du système d'équations de la grammaire G peut être réécrit sous forme de Q-analogue du système d'équations de la grammaire G', sans évidemment changer les solutions. Le théorème 2.36 permet alors de conclure. Toutefois, la preuve que nous donnons ci-dessous présente l'avantage de fournir explicitement le "dictionnaire" qui traduit les noms de paramètres de G vers G'.

Preuve. Soit $R : U \to u_0U_1 \dots U_nu_n$ la règle à itérer, et soient $R_1, \dots, R_m$ les U_k-dérivations. Les m règles qui, dans G', remplacent la règle R, sont

$$R'_i : U \to u_0U_1 \dots u_{k-1}.d(R_i).u_{k+1} \dots U_nu_n \quad (1 \leq i \leq m).$$

Nous commençons par le cas le plus simple: celui où le k-ème symbole du membre droit de R est distinct de U, ce qui revient à dire que chaque R_i est distinct de R.

Dans ce cas, le passage d'un arbre de dérivation $\mathcal{A}$ de G à l'arbre de dérivation correspondant $\mathcal{A}'$ de G' est simple : chaque sommet étiqueté R et son k-ème fils, étiqueté R_i, sont remplacés par un seul sommet étiqueté R'_i, les autres fils de ces deux sommets étant attachés dans l'ordre préfixe comme indiqué figure 3.5.

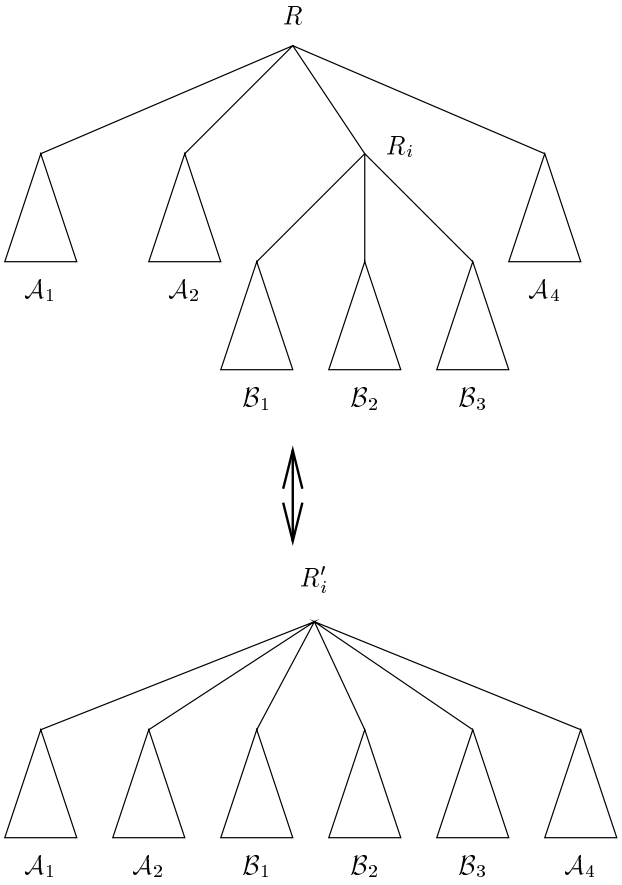


FIG. 3.5: *Itération de la règle R*

Si S est une chaîne de type W de l'arbre $\mathcal{A}$, elle devient dans $\mathcal{A}'$ une chaîne S'. Chaque sommet s d'étiquette R_i devient un sommet d'étiquette R'_i, et chaque sommet s d'étiquette R devient un sommet d'étiquette R'_i (l'indice i dépendant de l'étiquette du k-ème fils de s).

Chaque fois que deux lettres consécutives de W sont $R^{(k)}$ et $R_i^{(j)}$ (ou $R^{(k)}$ et R_i, en fin du mot W), il est possible que les deux sommets correspondants dans S deviennent, dans S', un seul et même sommet; dans ce cas, la chaîne S' est plus courte que S.

Comme dans la preuve de la proposition 3.5, nous dirons donc qu'une lettre de W est *non maximale* si cette lettre est $R^{(k)}$ et si la lettre suivante est une lettre R_i ou $R_i^{(j)}$.

La traduction d'un nom de paramètre élémentaire de G en son nom dans G' est alors

essentiellement assurée par φ, définie ainsi :

$$\begin{array}{rcll}
\varphi(R) & = & R'_1 + \cdots + R'_m; & \\
\varphi(R_i) & = & R'_i & (1 \leq i \leq m); \\
\varphi(R^{(k)}) & = & \sum_{i=1}^{m} \sum_{j=1}^{\alpha(R_i)} R_i^{\prime(k-1+j)}; & \\
\varphi(R_i^{(j)}) & = & R_i^{\prime(k-1+j)} & (1 \leq i \leq m, 1 \leq j \leq \alpha(R_i)); \\
\varphi(r) & = & r & (\text{pour tout autre } r \in \mathcal{R}_p \cup \mathcal{R}).
\end{array}$$

Pour obtenir le nom, dans G', du paramètre p_W, il suffit de traduire chaque lettre maximale r de W par $\varphi(r)$, et chaque lettre non maximale $R^{(k)}$ par $\epsilon + \varphi(R^{(k)})$.

La situation est plus complexe à décrire lorsque $R = R_{i_0}$, car il est possible d'avoir, dans une même branche d'un arbre de dérivation $\mathcal{A}$, une succession de sommets étiquetés R, chacun étant le k-ème fils du précédent. Dans l'arbre de dérivation $\mathcal{A}'$, les premier et deuxième sommets se contractent en un seul, ainsi que le troisième et le quatrième, etc. Autrement dit, en conservant la même définition d'une lettre non maximale du mot W, dans tout facteur de W formé de lettres non maximales suivies d'une lettre maximale (de la forme $R^{(k)} \ldots R^{(k)} R_i^{(j)}$), chaque lettre peut être effacée à condition que la suivante ne le soit pas, et soit remplacée par son image par φ. Notons qu'il convient de revoir la définition donnée de $\varphi(R)$ (qui cumule les définitions précédentes de $\varphi(R)$ et de $\varphi(R_{i_0})$) et $\varphi(R^{(k)})$ (qui cumule les définitions précédentes de $\varphi(R^{(k)})$ et $\varphi(R_{i_0}^{(k)})$), ce qui donne :

$$\begin{array}{rcl}
\varphi(R) & = & R'_{i_0} + \sum_{i=1}^{m} R'_i \\
\varphi(R^{(k)}) & = & R_{i_0}^{\prime(2k-1)} + \sum_{i=1}^{m} \sum_{j=1}^{\alpha(R_i)} R_i^{\prime(k-1+j)}
\end{array}$$

□

Similairement au passage en forme 1-2, nous pouvons définir un ordre partiel sur $\mathcal{R}_p \cup \mathcal{R}$, correspondant à la définition d'une lettre non maximale, par

$$R^{(k)} \leq \begin{cases} R_i & (1 \leq i \leq m) \\ R_i^{(j)} & (1 \leq i \leq m, 1 \leq j \leq \alpha(R_i)). \end{cases}$$

La fonction de traduction τ se définit alors récursivement :

$$\begin{cases}
\tau(r) = \varphi(r) & \text{si } r \in \mathcal{R}, \\
\tau(rW) = \varphi(r)\tau(W) & \text{si } r \in \mathcal{R}_p, r \neq R^{(k)}, \\
\tau(R^{(k)}rW) = \varphi(R^{(k)})\tau(rW) + 1_{R^{(k)}<r} \cdot \varphi(r)\tau(W) & \text{si } r \in \mathcal{R}_p \cup \mathcal{R}, W \in \mathcal{R}_p^* \mathcal{R}.
\end{cases}$$

Contrairement à ce qui se passe avec le passage d'une grammaire en forme 1-2, l'itération d'une règle ne donne pas une grammaire équivalente, mais une grammaire plus fine,

au moins dans le cas où le symbole itéré est le même que celui du membre gauche de la règle itérée. Nous allons le vérifier sur un exemple.

Exemple 3.7. Reprenons la grammaire G_1 engendrant les mots de Dyck. L'alphabet des dérivations pointées comportant deux lettres, il y a deux façons possibles d'itérer une règle dans cette grammaire. En choisissant d'itérer $R_2^{(1)}$, on obtient la grammaire G' définie par ses règles de dérivation :

$$\begin{aligned} R_1 : &\quad D \to \epsilon \\ R'_1 : &\quad D \to abD \\ R'_2 : &\quad D \to aaDbDbD \end{aligned}$$

La fonction φ est alors définie par :

$$\begin{aligned} \varphi(R_1) &= R_1 + R'_1 \\ \varphi(R_2) &= R'_1 + 2.R'_2 \\ \varphi(R_2^{(1)}) &= 2.R_2^{\prime(1)} + R_2^{\prime(2)} \\ \varphi(R_2^{(2)}) &= R_2^{\prime(2)} + R_2^{\prime(3)} \end{aligned}$$

En appliquant les règles de calcul de τ données dans la preuve de la proposition 3.6, nous obtenons pour le paramètre $p_{R_2^{(1)}R_2^{(1)}R_1}$:

$$\begin{aligned} \tau(R_2^{(1)}R_2^{(1)}R_1) &= \varphi(R_2^{(1)}R_2^{(1)}R_1) + \varphi(R_2^{(1)}R_1) + \varphi(R_2^{(1)})R'_1 \\ &= \left(2.R_2^{\prime(1)} + R_2^{\prime(2)}\right)\left(2.R_2^{\prime(1)} + R_2^{\prime(2)}\right)\left(R_1 + R'_1\right) \\ &\quad + \left(2.R_2^{\prime(1)} + R_2^{\prime(2)}\right)\left(R_1 + 2.R'_1\right) \end{aligned}$$

Le paramètre $p_{R_2^{(1)}R_2^{(1)}R_1}$ est compté par la variable s dans la solution du Q-système suivant, basé sur la grammaire G :

$$D(q,r,s) = q + D(qr, rs, s)D(q,r,s). \tag{1}$$

Dans ce système, q compte le paramètre p_{R_1}, et r, le paramètre $p_{R_2^{(1)}R_1}$. Nous utilisons l'équation 1 pour exprimer $D(qr, rs, s)$:

$$D(qr, rs, s) = qr + D(qr^2s, rs^2, s)D(qr, rs, s). \tag{2}$$

En reportant (2) dans (1), il vient alors :

$$D(q,r,s) = q + qrD(q,r,s) + D(qr^2s, rs^2, s)D(qr, rs, s)D(q,r,s).$$

Cette équation est une Q-équation basée sur la grammaire itérée G', et il est simple d'interpréter les paramètres comptés par q, r, et s :

– La variable q compte p_{W_1}, avec $W_1 = R_1 + R'_1$;

– la variable r compte p_{W_2}, avec $W_2 = R'_1 + (2.R_2'^{(1)} + R_2'^{(2)})W_1$;

– la variable s compte p_{W_3}, avec $W_3 = (2.R_2'^{(1)} + R_2'^{(2)})W_2$.

On retrouve ainsi le résultat donné en appliquant τ.

Afin de vérifier que la grammaire G' n'est pas Q-équivalente à G, il nous suffit d'observer la table donnant, pour les mots de longueurs 0, 2 et 4, les valeurs des paramètres élémentaires de rang 1 dans les deux grammaires.

	R_1	R_2	R_1	R'_1	R'_2
ϵ	1	0	1	0	0
ab	2	1	1	1	0
$aabb$	3	2	3	0	1
$abab$	3	2	1	2	0

Chacune des deux grammaires n'ayant qu'un seul symbole, chaque paramètre a un rang égal à son rang minimal. Par conséquent, si le paramètre $p_{R'_2}$ de G' était Q-comptable dans G, il devrait être combinaison linéaire (à coefficients entiers positifs) des paramètres p_{R_1} et p_{R_2} de G, ce qui n'est clairement pas le cas. Par conséquent, G n'est pas plus fine que G'.

Une autre façon de prouver cette propriété consiste à dire que les paramètres de rang 1 de G' sont capables de distinguer les mots $aabb$ et $abab$, alors que les paramètres de rang 1 de G ne le peuvent pas; par conséquent, il est impossible d'exprimer les paramètres de rang 1 de G' comme combinaisons linéaires de ceux de G.

Ce type d'arguments est naturel lorsqu'il s'agit de montrer qu'un paramètre n'est pas Q-comptable dans une grammaire donnée: il suffit généralement d'examiner les valeurs de ce paramètre pour les premiers mots engendrés par la grammaire, et de les comparer aux valeurs prises par les paramètres élémentaires dont le rang minimal ne dépasse pas celui du paramètre étudié.

Il existe une situation classique où l'itération d'une règle est une opération parfaitement naturelle, et que l'on risque d'effectuer sans même y penser. Si un symbole U ne peut être réécrit que d'une seule façon (lorsqu'il n'y a qu'une seule U-dérivation), il est naturel de faire disparaître ce symbole de la grammaire en remplaçant chaque apparition de U dans un membre droit de règle de dérivation, par le membre droit de la U-dérivation. Cela correspond en fait à itérer chacune des règles pointées qui font apparaître U. Cette opération est alors essentiellement l'inverse de celle décrite pour le passage en forme 1-2, et fournit une grammaire Q-équivalente.

3.2.3 Lemmes de marquage

Les lemmes suivants, que nous appelons lemmes de marquage, nous permettent de ne compter dans un paramètre que les chaînes vérifiant certaines conditions. Ils sont essentiels à la preuve du théorème de réduction du rang au rang minimal.

Lemme 3.8 (marquage supérieur). *Soient $G = (N, X, \mathcal{R}, S)$ une grammaire, et $\mathcal{R}_{p+}$ une partie de l'ensemble de ses règles de dérivation pointées $\mathcal{R}_p$.*

Pour chaque arbre de dérivation $\mathcal{A}$ de G, chaque règle de dérivation pointée $R \in \mathcal{R}_p$, et chaque nom de paramètre élémentaire $W \in \mathcal{R}_p^*\mathcal{R}$, soit $p_W^R(\mathcal{A})$ le nombre de chaînes de $\mathcal{A}$, de type W, qui sont des sous-chaînes droites d'au moins une chaîne de type RW.

Il existe une grammaire $G' = (N', X, \mathcal{R}', S)$, plus fine que G, qui vérifie les conditions suivantes :

1. *chaque paramètre G-Q-comptable élémentaire de rang k a, en tant que paramètre G'-Q-comptable, un rang au plus égal à k;*
2. *chaque paramètre p_W^R est G'-Q-comptable, de rang au plus $|W|$.*

Avant de donner la preuve de ce lemme, examinons un exemple.

Exemple 3.9. Considérons la grammaire $G = (\{D\}, \{a, b\}, \mathcal{R}, D)$, dont les règles de dérivation sont :

$$\begin{array}{llll} R_1 : & D \to ab & R_3 : & D \to abD \\ R_2 : & D \to aDb & R_4 : & D \to aDbD \end{array}$$

Cette grammaire, qui provient de la grammaire G_2 de l'exemple 2.3, engendre les mots de Dyck non vides.

Cette grammaire n'ayant qu'un seul symbole, tous les paramètres ont un rang minimal égal à leur rang. En particulier, le paramètre $p_{R_1+R_3}$ compte les "pics" (facteurs ab) des mots de Dyck engendrés. Mais que se passe-t-il si nous désirons compter les pics dont la hauteur n'est pas 1 ?

Un pic est représenté dans l'arbre de dérivation par un sommet s, étiqueté R_1 ou R_3. Sa hauteur est égale à 1, plus le nombre de dérivations pointées $R_2^{(1)}$ ou $R_4^{(1)}$ qui se trouvent sur la branche reliant ce sommet à la racine de l'arbre. Par conséquent, avec les notations du lemme, le nombre de pics de hauteur différente de 1 est le paramètre $p_{R_1}^{R_2^{(1)}} + p_{R_3}^{R_2^{(1)}} + p_{R_1}^{R_4^{(1)}} + p_{R_3}^{R_4^{(1)}}$: celui-ci compte en effet les sommets R_1 qui font partie d'une (ou plus) chaîne $R_2^{(1)}R_1$ ou $R_4^{(1)}R_1$, et les sommets R_3 qui font partie d'au moins une chaîne $R_2^{(1)}R_3$ ou $R_4^{(1)}R_3$.

Rien ne permet de penser *a priori* que ce paramètre est G-Q-comptable, les sommets étiquetés R_1 ou R_3 pouvant se trouver ou non "sous" les sommets R_2 ou R_4.

Nous voulons pouvoir, en comptant simplement les occurrences de certaines étiquettes dans l'arbre de dérivation d'un mot, dénombrer les sommets étiquetés R_1 ou R_3 qui font partie de chaînes convenables. Il faudrait pour cela que les étiquettes de ces sommets portent une information (une "marque") sur la présence ou l'absence, dans la branche qui les relie à la racine, de sommets étiquetés R_2 ou R_4 (avec, dans le cas de R_4, la bonne direction, ce qui revient à dire que la branche contient une dérivation pointée $R_2^{(1)}$ ou $R_4^{(1)}$). La solution est de réétiqueter les sommets pour inclure cette information : tout sommet, étiqueté R_i, qui se trouve dans un sous-arbre d'un sommet étiqueté R_2, ou dans un sous-arbre gauche d'un sommet étiqueté R_4, est réétiqueté R'_i.

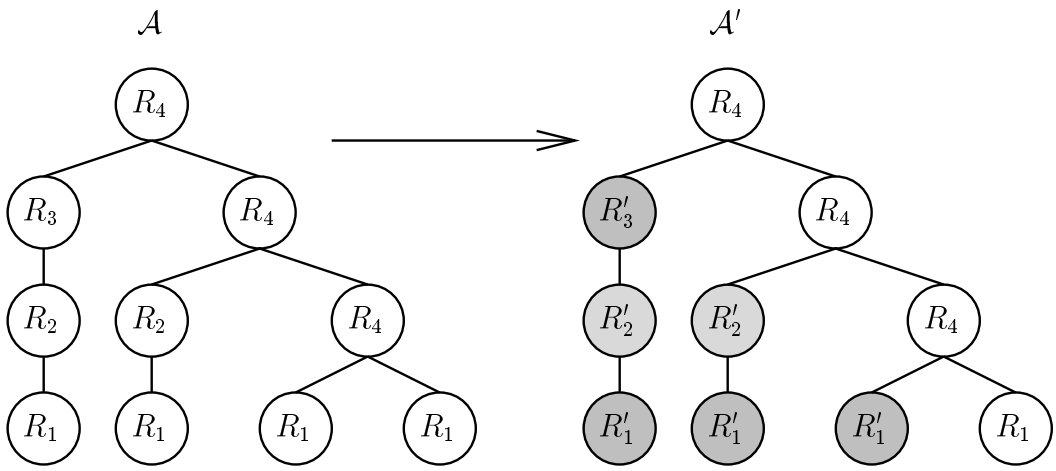


FIG. 3.6: *Marquage supérieur des règles* $R_2^{(1)}$ *et* $R_4^{(1)}$

La figure 3.6 montre l'arbre de dérivation du mot $w = aabaabbbaaabbbaabbab$, ainsi que la version réétiquetée. Le paramètre "nombre de pics de hauteur au moins 2" est égal au nombre de sommets étiquetés R'_1 ou R'_3 dans ce nouvel arbre. Par différence, le nombre de pics de hauteur 1 est le nombre de sommets étiquetés R_1 ou R_3 dans ce même arbre. Comme on peut le voir, le mot w possède 1 pic de hauteur 1, et 4 autres pics de hauteurs respectives 2, 3, 3, et 2.

Dans cet exemple, il n'a pas été nécessaire de marquer différemment les deux règles pointées $R_2^{(1)}$ et $R_4^{(1)}$, comme c'est le cas en général dans le lemme.

Preuve. Nous donnons tout de suite la grammaire G'. Les symboles de N' sont de la forme U_E, où U est l'un des symboles de N et E est une partie de l'ensemble $\mathcal{R}_{p+}$. Les règles de

dérivation de G' sont formées à partir de celles de G de la manière suivante :

- pour chaque règle $R \in \mathcal{R}$ et chaque ensemble $E \subset \mathcal{R}_{p+}$, on forme une règle R_E qui est identique à R, à ceci près que chaque symbole U (du membre gauche comme du membre droit) est remplacé par U_E;

- pour chaque règle pointée spéciale $R^{(d)} \in \mathcal{R}_{p+}$, et pour chaque ensemble $E \subset \mathcal{R}_{p+}$, le d-ème symbole du membre droit de R_E, U_E, est remplacé par $U_{E \cup \{R^{(d)}\}}$.

Les indices ajoutés aux symboles ne modifient en rien les mots engendrés, en ce sens que, pour tout symbole $U \in N$ et tout $E \subset \mathcal{R}_{p+}$, $L_G(U) = L_{G'}(U_E)$. En revanche, la façon dont sont formées les règles de dérivation de G' assure que, dans chaque arbre de dérivation de G', chaque sommet a une étiquette qui nous renseigne sur la branche reliant ce sommet à la racine de l'arbre.

Soit $\mathcal{A}$ l'arbre de dérivation, dans G, d'un mot w. Pour chaque sommet s de $\mathcal{A}$, notons $(s_1, \ldots, s)$ la chaîne de sommets de $\mathcal{A}$ formée de tous les sommets ancêtres de s, et $E(s)$ l'ensemble de toutes les règles pointées qui apparaîssent à la fois dans le type de $(s_1, \ldots, s)$ et dans $\mathcal{R}_{p+}$.

Soit $\mathcal{A}'$ l'arbre de dérivation du mot w dans la grammaire G'. Alors, $\mathcal{A}'$ est obtenu à partir de $\mathcal{A}$ en remplaçant chaque étiquette R de chaque sommet s, par $R_{E(s)}$. En effet, la définition que nous avons donnée des membres droits des règles R_E, assure que les ensembles indices $E(s)$ se propagent correctement.

La grammaire G' remplit les conditions indiquées. En effet, notons φ l'application "effacement des indices", définie sur N' par $\varphi(U_E) = U$ et sur $\mathcal{R}'$ par $\varphi(R_E) = R$. L'application φ est étendue naturellement aux dérivations pointées et, par morphisme, aux noms de paramètres de G'. Alors, il est clair que, dans un arbre de dérivation $\mathcal{A}$ de G, toute chaîne de type W devient, dans l'arbre $\mathcal{A}'$ correspondant de G', une chaîne dont le type appartient à $\varphi^{-1}(W)$. Réciproquement, lors du passage de $\mathcal{A}'$ à $\mathcal{A}$, une chaîne de type W' devient une chaîne de type $\varphi(W')$. Par conséquent, les paramètres p_W (dans G) et $p_{\varphi^{-1}(W)}$ (dans G') coïncident. Ainsi, G' est bien plus fine que G, et le rang des paramètres est conservé (condition 1).

Enfin, les chaînes comptées dans un arbre $\mathcal{A}$ par le paramètre $p_W^{R^{(d)}}$ sont exactement celles qui sont de type W et dont le premier sommet s_1 vérifie $R^{(d)} \in E(s_1)$. Or $E(s_1)$ est, par construction, l'indice de l'étiquette de s_1 dans l'arbre $\mathcal{A}'$. Il suffit donc de choisir pour W', les noms de $\varphi^{-1}(W)$ dont la première lettre est de la forme $R_E^{(d)}$, avec $R^{(d)} \in E$, pour assurer que $p_W^{R^{(d)}}$ (dans G) coïncide avec $p_{W'}$ (dans G'). De plus, le rang de $p_{W'}$ est bien la longueur du plus long mot de W', qui est $|W|$. □

Exemple 3.10. Dans l'exemple 3.9, les règles pointées dont nous voulions repérer la présence étaient $R_2^{(1)}$ et $R_4^{(1)}$, qui forment donc l'ensemble $\mathcal{R}_{p+}$. Toutefois, il n'était nécessaire que de repérer la présence d'au moins l'une de ces deux règles, sans faire de distinction entre elles. Nous avons donc utilisé, comme étiquettes, R_i pour $R_{i,\emptyset}$ et R'_i pour $R_{i,E}$ avec $\emptyset \subsetneq E \subset \mathcal{R}_{p+}$.

La grammaire G' obtenue est donc :

$$\begin{array}{llll} R_1 : & D & \to & ab \\ R_2 : & D & \to & aD'b \\ R_3 : & D & \to & abD \\ R_4 : & D & \to & aD'bD \end{array} \qquad \begin{array}{llll} R'_1 : & D' & \to & ab \\ R'_2 : & D' & \to & aD'b \\ R'_3 : & D' & \to & abD' \\ R'_4 : & D' & \to & aD'bD' \end{array}$$

L'arbre $\mathcal{A}'$ de la figure 3.6 est un arbre de dérivation de G'. Dans cette grammaire, le symbole D' n'est accessible à partir de D qu'au travers des dérivations pointées $R_2^{(1)}$ et $R_4^{(1)}$, et par conséquent, les pics de hauteur $h > 1$ sont ceux introduits par les dérivations R'_1 et R'_3, tandis que les pics de hauteurs 1 sont introduits par les dérivations R_1 et R_3.

Lemme de marquage inférieur

Le lemme de marquage supérieur nous permet de propager à chaque sommet d'un arbre de dérivation une information sur ce qui se trouve au-dessus de lui, c'est-à-dire dans la branche qui le relie à la racine. Le lemme ci-dessous permet un marquage similaire, mais portant sur ce qui se trouve en dessous de chaque sommet, c'est-à-dire dans les sous-arbres issus de ses fils.

Lemme 3.11 (marquage inférieur). *Soient $G = (N, X, \mathcal{R}, S)$ une grammaire, $\mathcal{R}_+ \subset \mathcal{R}$ une partie de ses règles de dérivation, et un entier $m > 0$.*

Pour chaque arbre de dérivation $\mathcal{A}$ de G, chaque nom de paramètre élémentaire $W \in \mathcal{R}_p^+$, chaque $R \in \mathcal{R}_+$, et chaque entier $n \leq m$, notons $p^R_{W,n}(\mathcal{A})$ le nombre de chaînes de $\mathcal{A}$, de type W, qui sont des sous-chaînes gauches d'exactement n (ou d'au moins m, si $n = m$) chaînes de type WR.

Il existe une grammaire $G' = (N', X, \mathcal{R}', S')$, plus fine que G, et qui vérifie les conditions suivantes :

1. *chaque paramètre G-Q-comptable de rang k a, en tant que paramètre G'-Q-comptable, un rang au plus égal à k;*

2. *chaque paramètre $p^R_{W,n}$ est G'-Q-comptable de rang au plus $|W|$.*

La principale différence de forme entre les deux lemmes de marquage réside dans le fait que le lemme de marquage inférieur, tel qu'il est énoncé ci-dessus, utilise un entier m supplémentaire là où le lemme de marquage supérieur se contentait, en fait, de 0 ou 1 (le lemme de marquage supérieur permet de compter les chaînes qui sont sous-chaînes droites d'*au moins* une chaîne d'un type donné, tandis que le lemme de marquage inférieur permet de distinguer celles qui sont sous-chaînes gauches d'*exactement* $0, 1, \ldots, m-1$ telles chaînes). Il est possible d'énoncer le lemme de marquage supérieur de façon similaire, mais un tel raffinement complique les notations, et la version que nous avons démontrée est suffisante pour démontrer le théorème de réduction du rang.

Preuve. Soit $\mathcal{A}$ un arbre de dérivation de G, et s un sommet de $\mathcal{A}$. Pour chaque $R \in \mathcal{R}_+$, notons $R(s)$ le nombre de sommets étiquetés R dans le sous-arbre issu de s, avec la convention que, si ce nombre est supérieur à m, $R(s) = m$. Si s' est le i-ème fils de s, notons $R(s,i) = R(s')$.

Comme dans le cas du lemme de marquage supérieur, le passage à la grammaire G' consiste essentiellement à d'ajouter à l'étiquette de chaque sommet d'arité k, la liste des valeurs de $R(s)$ et $R(s,i)$, pour $R \in \mathcal{R}_+$ et $1 \leq i \leq k$.

La grammaire G' est définie comme suit. Chaque symbole est formé d'un symbole de G et d'un entier $n(R)$ pour chaque règle $R \in \mathcal{R}_+$:

$$N' = \left\{ \left(U, (n(R))_{R \in \mathcal{R}_+} \right), U \in N, 0 \leq n(R) \leq m \right\}.$$

Chaque entier $n(R)$ indique la valeur du paramètre p_R sur les mots engendrés par le symbole $\left(U, (n(R))_{R \in \mathcal{R}_+} \right)$:

$$L_{G'} \left(U, (n(R))_{R \in \mathcal{R}_+} \right) = \{ w \in L_G(U) : \forall R \in \mathcal{R}_+, n(R) = \min(m, p_R(w)) \}. \tag{3}$$

Pour chaque règle $R_0 : U \to w_0 U_1 w_1 \ldots U_k w_k \in \mathcal{R}$, la grammaire G' comprendra $(m+1)^{k|\mathcal{R}_+|}$ règles, dont les membres droits sont obtenus en remplaçant successivement, de toutes les manières possibles, chaque symbole U_i par un $\left(U_i, (n_i(R))_{R \in \mathcal{R}_+} \right)$. Le membre gauche de chaque règle est alors $\left(U, (n(R))_{R \in \mathcal{R}_+} \right)$ avec

$$n(R) = \min \left(m, \delta_{R,R_0} + \sum_{i=1}^{k} n_i(R) \right),$$

où δ_{R,R_0} vaut 1 si $R = R_0$, 0 sinon. Par une récurrence évidente sur la taille des arbres de dérivation, ces règles de dérivation engendrent le langage prévu pour chaque symbole.

Il convient de donner un axiome à G'; soit donc S_0 un nouveau symbole, et pour chaque symbole $\left(S, (n(R))_{R \in \mathcal{R}_+} \right)$, ajoutons une règle $S_0 \to \left(S, (n(R))_{R \in \mathcal{R}_+} \right)$: cela assurera que

$L_{G'}(S_0) = L_G(S)$. Le symbole S_0 n'étant accessible à partir d'aucun autre, ses dérivations n'apparaîtront qu'à la racine des arbres de dérivation.

Notons φ l'application qui envoie chaque règle de $\mathcal{R}'$ ainsi définie (autre qu'une S_0-dérivation) sur la règle $R \in \mathcal{R}$ dont elle est issue, et étendons φ naturellement aux règles pointées de $\mathcal{R}'_p$, puis, par morphisme, aux noms de paramètres dans G'. Le passage de G à G' s'effectue simplement en réétiquetant les sommets de l'arbre de dérivation, et en ajoutant une nouvelle racine. Donc, pour tout mot $w \in L_{G'}\left(U, (n(R))_{R\in\mathcal{R}_+}\right)$ et tout nom de paramètre W dans G, $p_W(w) = p_{\varphi^{-1}(W)}(w)$. Ainsi, G' est bien plus fine que G, et le rang des paramètres G-Q-comptables n'augmente pas lors du passage à G'.

Quant au paramètre $p^R_{W,n}$, si la dernière lettre de W est $R_1^{(d)} \in \mathcal{R}_p$, il compte dans les arbres de dérivation de G' les chaînes de sommets dont le type est dans $\varphi^{-1}(W)$, et dont le d-ème symbole au membre droit de la dernière dérivation vérifie $n(R) = n$. Cela revient à ne prendre, dans $\varphi^{-1}(W)$, que les mots dont la dernière lettre appartient à une partie convenable de $\varphi^{-1}(R_1)$. Ceci assure que $p^R_{W,n}$ est bien G'-Q-comptable de rang $|W|$.

□

Remarque. La grammaire G' construite dans la preuve du lemme de marquage inférieure peut parfaitement ne pas être complète; il est possible en effet que certains des langages engendrés soient vides.

Exemple 3.12. Reprenons les grammaires G_1 et G_2, qui engendrent tous deux le langage de Dyck. Nous allons voir que la grammaire G_2 peut être obtenue en appliquant la construction du lemme de marquage inférieur à G_1.

Dans la grammaire G_1, la règle R_1 est la seule règle terminale; par conséquent $p_{R_1}(w) \geq 1$ pour tout mot de Dyck w, et $p_{R_1}(w) = 1$ seulement pour le mot vide. Prenons $\mathcal{R}_+ = \{R_1\}$, et $m = 2$.

La construction donnée de la grammaire G' prévoit 4 symboles, S_0, $(D,0)$, $(D,1)$ et $(D,2)$, et, pour chaque $i = 0, 1, 2$,

$$L_{G'}(D,i) = \{w \in D, i = \min(2, p_{R_1}(w))\}.$$

En particulier, $L_{G'}(D,0) = \emptyset$, et $L_{G'}(D,1) = \{\epsilon\}$, donc $L_{G'}(D,2)$ est l'ensemble des mots de Dyck non vides.

Les règles de dérivation de la grammaire G' sont :

$$\begin{array}{rcl@{\qquad}rcl}
S_0 & \to & (D,0) & S_0 & \to & (D,1) \\
S_0 & \to & (D,2) & (D,1) & \to & \epsilon \\
(D,0) & \to & a.(D,0).b.(D,0) & (D,1) & \to & a.(D,0).b.(D,1) \\
(D,1) & \to & a.(D,1).b.(D,0) & (D,2) & \to & a.(D,0).b.(D,2) \\
(D,2) & \to & a.(D,1).b.(D,1) & (D,2) & \to & a.(D,1).b.(D,2) \\
(D,2) & \to & a.(D,2).b.(D,0) & (D,2) & \to & a.(D,2).b.(D,1) \\
(D,2) & \to & a.(D,2).b.(D,2) & & &
\end{array}$$

Il apparaît immédiatement que le seul symbole accessible à partir de $(D,0)$ est $(D,0)$, tandis que la seule règle terminale a pour membre gauche $(D,1)$; par conséquent $L_{G'}(D,0)$ est vide, et toutes les règles de dérivation faisant intervenir $(D,0)$ peuvent être éliminées. La grammaire alors obtenue, une fois les symboles $S_0, (D,1), (D,2)$ renommés en D, F, E respectivement, est :

$$\left\{\begin{array}{rcl@{\qquad}rcl}
D & \to & F & D & \to & E \\
F & \to & \epsilon & E & \to & aFbF \\
E & \to & aFbE & E & \to & aEbF \\
E & \to & aEbE & & &
\end{array}\right.$$

La seule F-dérivation de cette grammaire est $F \to \epsilon$; en itérant chacune des dérivations pointées qui produisent le symbole F, on fait disparaître ce symbole de tous les membres droits, ce qui le rend inaccessible à partir de l'axiome D. La grammaire alors obtenue est

$$\left\{\begin{array}{rcl@{\qquad}rcl}
D & \to & \epsilon & D & \to & E \\
E & \to & ab & E & \to & abE \\
E & \to & aEb & E & \to & aEbE
\end{array}\right.$$

qui est exactement la grammaire G_2. Par conséquent, cette grammaire est plus fine que G_1.

3.2.4 Réduction du rang

Les lemmes de marquage ont pour principal but de permettre la preuve du théorème suivant :

Théorème 3.13. *Soit G une grammaire. Il existe une grammaire G', plus fine que G, dans laquelle tout paramètre G-Q-comptable non borné a un rang égal à son rang minimal.*

Preuve. Etant donnée la grammaire G, nous devons essentiellement trouver une grammaire G', plus fine que G, dans laquelle tout paramètre *élémentaire* de G, non borné, a dans G' un rang qui correspond à son rang minimal. Rappelons que, d'après la proposition (2.24), il est équivalent, pour un paramètre Q-comptable, d'être non borné et d'avoir rang minimal au moins 1. Il est donc normal que les paramètres bornés soient exclus du cadre de ce théorème.

La proposition (2.26) décrit exactement d'où peut provenir l'écart entre rang minimal et rang formel pour un paramètre élémentaire : il peut s'agir de dérivations pointées ne pouvant se retrouver à plus d'un exemplaire sur une même branche ($R^{(i)}$, lorsque $g(R)$ n'est pas accessible à partir de $d(R,i)$), ou de la dernière dérivation R du nom du paramètre, à condition que le paramètre (de rang formel 1) p_R soit borné, au moins sur $L_G(d(R_1,i))$, où $R_1^{(i)}$ est l'avant-dernière lettre du nom du paramètre.

Soit $\mathcal{R}_{p+}$ l'ensemble des règles pointées de la forme $R^{(k)}$, telles que $g(R)$ ne soit pas accessibles à partir de $d(R,k)$. Ces règles pointées sont exactement celles qui ne peuvent pas "boucler", en ce sens que le k-ème sous-arbre d'un sommet étiqueté R ne peut contenir de sommet étiqueté R. Soit G_1 la grammaire obtenue en appliquant le lemme de marquage supérieur à cet ensemble de règles $\mathcal{R}_{p+}$. Nous allons montrer que, dans G_1, l'écart entre rang formel et rang minimal pour les paramètres G-Q-comptables élémentaires est inférieur ou égal à 1.

Soit en effet $W \in \mathcal{R}_p^*\mathcal{R}$ un nom de paramètre G-Q-comptable élémentaire de rang minimal 1 ou plus. Posons $W = W_0 R_1^{(d_1)} W_1 \dots W_{k-1} R_k^{(d_k)} W_k R'$, où $R_i^{(d_i)} \in \mathcal{R}^+$ et $W_i \in (\mathcal{R}_p - \mathcal{R}_{p+})^*$. Notons encore $n_i = |W_i|$: le rang minimal de p_W est donc, d'après la proposition (2.26), $1 + n_1 + \dots + n_k$ ou $n_1 + \dots + n_k$, suivant que R' participe ou non au rang minimal.

Puisque p_W est supposé non borné, il n'est pas identiquement nul, et donc il existe un arbre de dérivation de G qui contient une chaîne de type W. Par construction de $\mathcal{R}_{p+}$, la règle $R_i^{(d_i)}$ ne peut apparaître qu'une fois dans le type d'une chaîne de sommets d'un arbre de dérivation, et par conséquent, si $R^{(d)}$ est une dérivation pointée qui apparaît dans W après $R_i^{(d_i)}$, aucune chaîne de type $R^{(d)} R_i$ ne peut exister. De même, si $R^{(d)}$ apparaît dans W avant $R_i^{(d_i)}$, aucune chaîne de type $R_i^{(d_i)} R$ ne peut exister. Donc, dans un arbre de dérivation $\mathcal{A}$ de G, les chaînes $(s_1, \dots, s_n)$ de type $W' = W_0 W_1 \dots W_k R'$ qui sont des sous-chaînes d'une chaîne de type W, sont exactement celles dont le dernier sommet s_n se trouve dans le d_i-ème sous-arbre d'un sommet étiqueté R_i, pour chaque $i \leq k$. Ces chaînes sont comptées, dans G_1, par un paramètre Q-comptable de rang $|W'| = 1 + \sum_i n_i$, ce qui assure que, dans G_1, l'écart entre rang minimal et rang formel des paramètres G-Q-comptables

non bornés est au plus de 1.

Appliquons maintenant à la grammaire G_1 le lemme de marquage inférieur, en prenant comme ensemble de règles $\mathcal{R}_+$,

$$\mathcal{R}_+ = \left\{R \in \mathcal{R}' : \exists U \in N', p_R \text{ est borné sur } L_{G_1}(U)\right\}$$

Nous prenons, comme entier m pour le lemme de marquage inférieur,

$$m = 1 + \max\left\{p_R(w) : w \in L_G(U), p_R \text{ est borné sur } L_G(U)\right\}.$$

Soit G' la grammaire ainsi obtenue.

Tout paramètre G_1-Q-comptable élémentaire de rang n dont la dernière dérivation ne compte pas dans le rang minimal (d'après la proposition (2.26)), a, en tant que paramètre G'-Q-comptable, un rang au plus $n-1$. Etant donné que, pour chaque nom de paramètre élémentaire de G' qui intervient dans la décomposition d'un paramètre G-Q-comptable, seule la dernière lettre peut être source d'écart entre rang formel et rang minimal, il est donc clair que tous ces paramètres ont, dans G', un rang formel égal à leur rang minimal.

La grammaire G' remplit donc les conditions exigées. □

Ce théorème permet de justifier la supposition faite dans l'exemple présenté section 3.1 : si un paramètre est de rang minimal 1, il est possible de trouver une grammaire plus fine dans laquelle il est de rang formel 1.

Remarque. Rien n'indique que *tous* les paramètres G'-Q-comptables aient un rang minimal égal à leur rang formel; seuls ceux qui apparaissent dans la décomposition des paramètres G-G-comptables sont tenus de vérifier une telle propriété.

Par ailleurs, remarquons que, pour chaque opération de changement de grammaire que nous avons étudiée, il est possible de définir, le plus souvent sous forme récursive, la façon dont un nom de paramètre de la grammaire de départ se transforme dans la nouvelle grammaire.

Enfin, notons que les constructions décrites dans les deux lemmes de marquage ont tendance à augmenter grandement la complexité des grammaires, le nombre de symboles et de règles de dérivation pouvant exploser rapidement.

L'intérêt du théorème 3.13 est principalement théorique. Il serait tentant de l'utiliser pour déterminer si un paramètre est Q-comptable dans une grammaire donnée (voir les commentaires sur la proposition 2.26), mais la taille des grammaires construites augmente trop vite pour que l'idée soit applicable en pratique.

Toutefois, ce théorème nous permet d'affirmer que les paramètres dont le rang minimal est strictement inférieur au rang n'ont pas un comportement fondamentalement différent de ceux pour lesquels le rang minimal est égal au rang. Cette remarque peut être précieuse pour démontrer des résultats négatifs, comme celui de la section 3.1.

3.3 Q-grammaires et grammaires d'objets

Dans [36], Dutour définit la notion de grammaires d'objets isomorphes. Une grammaire algébrique peut être considérée comme un cas particulier de grammaire d'objets, et il semble naturel de comparer les deux notions.

Dans le cas d'une grammaire n'ayant qu'un seul symbole non terminal, deux grammaires sont isomorphes si, pour chaque entier n, elles ont toutes deux le même nombre de règles d'arité n; il est alors possible de décrire, par un simple réétiquetage des arbres de dérivation, des bijections entre les familles d'objets engendrés. De plus, dans le cas de grammaires n'ayant qu'un seul symbole non terminal, Dutour montre que deux grammaires ayant chacune au moins une règle d'arité au moins égale à 2 ont des itérées isomorphes. La notion d'itération employée n'est pas très différente de celle que nous avons définie précédemment.

Dans le cas des Q-grammaires, toutefois, ces résultats n'ont pas d'équivalents simples. Considérons les deux grammaires G et G', engendrant toutes deux le langage de Dyck :

$$G : \begin{cases} R_1 : & D & \to & \epsilon \\ R_2 : & D & \to & aDbD \end{cases} \qquad G : \begin{cases} R'_1 : & D & \to & \epsilon \\ R'_2 : & D & \to & DaDb \end{cases}$$

Ces deux grammaires ont chacune une règle d'arité 0 et une règle d'arité 2, mais il est facile de montrer qu'elles ne sont pas Q-équivalentes. Le tableau ci-dessous indique, pour les mots de Dyck de longueur inférieure ou égale à 6, les valeurs des paramètres Q-comptables de rang au plus 2 dans chacune des deux grammaires. Les paramètres $R_2^{(d)R_1}$ et $R_2'^{(d)}R'_1$, pour $d = 1, 2$, ne sont pas mentionnés, car ils s'écrivent systématiquement comme combinaisons linéaires des autres paramètres (ainsi, $p_{R_2^{(1)}R_1} = p_{R_2^{(1)}R_2} + p_{R_2}$).

	G				G'			
w	R_1	R_2	$R_2^{(1)}R_2$	$R_2^{(2)}R_2$	$R-_1$	R'_2	$R_2^{\prime(1)}R'_2$	$R_2^{\prime(2)}R'_2$
ϵ	1	0	0	0	1	0	0	0
ab	2	1	0	0	2	1	0	0
$aabb$	3	2	1	0	3	2	0	1
$abab$	3	2	0	1	3	2	1	0
$aaabbb$	4	3	3	0	4	3	0	3
$aababb$	4	3	2	1	4	3	1	2
$aabbab$	4	3	1	1	4	3	2	1
$abaabb$	4	3	1	2	4	3	1	1
$ababab$	4	3	0	3	4	3	3	0

Dans ce tableau, il est impossible d'obtenir la colonne correspondant au paramètre $p_{R_2^{\prime(1)}R'_2}$ comme combinaison linéaire à coefficients entiers positifs des colonnes correspondant aux paramètres de la grammaire G; par conséquent, $p_{R_2^{\prime(1)}R'_2}$ n'est pas Q-comptable dans G (tous les paramètres élémentaires des deux grammaires ayant un rang minimal égal à leur rang, il est inutile de faire entrer les paramètres élémentaires de rang supérieur en ligne de compte).

Notons qu'il serait possible, en permutant les mots de même longueur entre eux, de faire coïncider les paramètres élémentaires des deux grammaires. Cette permutation (*aaabbb* devient *ababab*, *aababb* devient *aabbab*, *aabbab* devient *abaabb*, *abaabb* devient *aababb*, et *ababab* devient *aaabbb* pour les mots de longueur 6) correspond exactement à interpréter les arbres de dérivation de G comme arbres de dérivation de G' (après réétiquetage) : il n'est pas surprenant que cette transformation ne laisse pas inchangés les paramètres Q-comptables, qui sont définis mot par mot, et non pas globalement "à une bijection près".

En fait, il semble raisonnable de conjecturer que deux grammaires n'ayant chacune qu'un symbole non terminal, et ayant, pour chaque n, le même nombre de règles d'arité n, ne sont Q-équivalentes que si elles sont identiques (à un renommage des règles près).

3.4 Conclusion

Nous avons pu montrer dans ce chapitre que la plupart des transformations élémentaires sur les grammaires se comportent bien du point de vue des paramètres Q-comptables. Ainsi, une grammaire est Q-équivalente à sa forme 1-2, et, au pire, l'itération ne fait qu'ajouter des paramètres Q-comptables. Il est donc possible d'effectuer des modifications

mineures dans la forme d'une grammaire, sans perte de paramètres Q-comptables. D'un point de vue pratique, de telles propriétés de stabilité sont confortables, comme nous le verrons au chapitre 5.

D'un point de vue plus théorique, le théorème de réduction du rang peut s'avérer précieux. Nous en avons vu un exemple lorsqu'il s'est agi de prouver qu'un paramètre donné ne pouvait être Q-comptable, lorsque le simple examen du rang minimal ne permettait pas de conclure.

Il pourrait être intéressant de savoir si les transformations de grammaires que nous avons définies, suffisent pour obtenir toutes les grammaires plus fines que la grammaire de départ; il est probable que non.

Une autre direction possible de recherche serait de déterminer quels sont, pour un langage donné, les paramètres qui sont Q-comptables dans toute grammaire. Nous savons déjà que tout paramètre qui s'exprime comme combinaison linéaires des nombres d'occurrences des lettres, est Q-comptable dans toute grammaire. Dans le cas du langage de Dyck, il semble vraisemblable que le paramètre "aire de Carlitz", ou une de ses variantes, soit Q-comptable dans toute grammaire.

Dans le même registre, il serait intéressant de savoir s'il peut exister entre deux paramètres p et p' des relations du type "p' est Q-comptable dans toute grammaire où p est Q-comptable". Un exemple serait, pour le langage de Dyck, de prouver que le paramètre "somme des hauteurs des pics" est Q-comptable dans toute grammaire où le paramètre "nombre de pics" l'est.

Enfin, il pourrait être intéressant, étant données deux grammaires, de savoir déterminer s'il existe une grammaire plus fine que chacune d'elles. Dans ce domaine, les techniques qui s'appliquent aux grammaires d'objets ne semblent pas pouvoir être transposées. Une question proche serait: étant donnés deux paramètres qui sont Q-comptables dans des grammaires différentes, existe-t-il une grammaire dans laquelle ces deux paramètres soient Q-comptables ? Nous avons vu Section 3.1 que ce n'est pas vrai dans le cas général; toutefois, il semble que ce le soit pour toutes les grammaires issues d'une même grammaire initiale par les transformations décrites dans ce chapitre.

Chapitre 4

Statistiques et asymptotiques

Dans certains cas, bien qu'il soit possible d'obtenir des formules exactes d'énumération suivant certains paramètres pour une famille donnée d'objets combinatoires, une expression asymptotique apporte des informations beaucoup plus lisibles qu'une expression comportant des sommations multiples.

Nous nous concentrons dans ce chapitre sur le lien entre Q-comptabilité et deux aspects essentiels de la combinatoire énumérative : le calcul de valeurs moyennes de paramètres, et l'obtention d'expressions asymptotiques dans les problèmes d'énumération. En particulier, nous montrons que les séries de moments suivant un paramètre Q-comptable sont algébriques.

On trouvera une description généraliste de techniques d'énumération asymptotique dans les articles de Bender [9, 8], ainsi qu'une version multivariée dans l'article de Bender et Richmond [10]. L'étude asymptotique du nombre de mots de longueur donnée d'un langage algébrique peut être menée au moyen des techniques décrites par Flajolet et Sedgewick dans [45]. Drmota [34, 35] et Flajolet et Sedgewick [46] ont montré que, sous des hypothèses raisonnables, le nombre d'occurrences d'une lettre dans les mots de longueur n d'un langage algébrique admet une distribution limite gaussienne, d'espérance et de variance proportionnelles à n. De tels résultats s'appliquent directement aux paramètres Q-comptables de rang 1. La détermination de lois limites pour des paramètres Q-comptables de rang supérieur est beaucoup plus difficile. Prellberg [68] a montré que l'aire (paramètre de rang 2) des polyominos parallélogrammes de périmètre $2n$ admet une loi limite qui suit la *distribution d'Airy*.

4.1 Introduction et notations

Les questions d'ordre statistique relatives aux paramètres définis sur une famille A d'objets combinatoires, sont essentiellement : quelle est la valeur moyenne de tel paramètre $p(w)$, lorsque w est un objet pris au hasard dans A ? Comment se répartissent les valeurs de $p(w)$? Y a-t-il une corrélation entre les valeurs de $p_1(w)$ et celles de $p_2(w)$?

Avant tout, pour pouvoir parler d'objet pris "au hasard" dans A, il est indispensable de définir une loi de probabilité sur A. En général, les familles d'objets considérées sont infinies et dénombrables (mots d'un langage algébrique, différentes classes de polyominos ...); il n'existe pas sur de tels ensembles de lois de probabilités privilégiées, et en particulier pas de lois uniformes. Pour que chaque objet ait la même probabilité, il convient alors de se ramener à des ensembles finis. C'est ce que nous faisons en étudiant la distribution de paramètres Q-comptables pour les mots de *taille* donnée.

Dans ce chapitre, nous manipulerons des séries formelles à un grand nombre de variables. Si $F = F(q_1, \dots, q_k)$ est une série formelle des variables $q_1, \dots, q_k$, le coefficient de $q_1^{n_1} \dots q_k^{n_k}$ dans la série F est noté

$$[q_1^{n_1} \dots q_k^{n_k}]F.$$

Une opération très utile est la *projection* π de $\mathbb{Q}[[q_1, \dots, q_{k'}]]$ vers $\mathbb{Q}[[q_1, \dots, q_k]]$, avec $k' > k$. Elle correspond à donner la valeur 1 à chacune des variables $q_{k+1}, \dots, q_{k'}$. La série ainsi obtenue sera simplement notée $F(q_1, \dots, q_k) = \pi F(q_1, \dots, q_{k'})$.

Formellement, on a

$$[q_1^{n_1} \dots q_k^{n_k}]F(q_1, \dots, q_k) = \sum_{n_{k+1} \geq 0} \dots \sum_{n_{k'} \geq 0} [q_1^{n_1} \dots q_{k'}^{n_{k'}}]F(q_1, \dots, q_{k'}).$$

La projection d'une série n'est définie que si la somme ci-dessus est une somme finie; cette propriété sera toujours vraie dans les cas que nous étudierons.

4.2 Généralités

4.2.1 Distributions de paramètres

En général, il existe pour chaque classe d'objets au moins un paramètre "naturel", qui possède la propriété que, pour chaque valeur possible du paramètre, il n'existe qu'un nombre fini d'objets pour lesquels le paramètre prend cette valeur. Nous appelons un tel paramètre une *taille*.

La longueur des mots d'un langage sur un alphabet fini, le périmètre ou l'aire des polyominos, sont des exemples de tailles. Le fait que l'un des paramètres suivant lesquels on établit une série génératrice soit une taille, est une condition suffisante pour que cette série génératrice ait un sens en tant que série formelle.

Dans un langage, il existe une taille plus naturelle que les autres : il s'agit de la longueur des mots. Toutefois, dans le cadre de la combinatoire énumérative, les mots sont souvent un moyen de coder d'autres objets, et il peut exister différents paramètres pouvant faire office de taille pour une même classe d'objets combinatoires – les plus classiques, pour des objets comme les polyominos, étant l'*aire* et le *périmètre*.

Soit D un ensemble d'objets, et soit $D(q_1, \ldots, q_k)$ sa série génératrice suivant k paramètres $\lambda_1, \ldots, \lambda_k$. Le premier paramètre λ_1 est supposé être une taille.

Sur chaque partie finie de D, comme par exemple chaque ensemble

$$D_{\lambda_1=n} = \{w \in D, \lambda_1(w) = n\}$$

il existe une probabilité uniforme, et donc chaque paramètre ou famille de paramètres a sur chaque $D_{\lambda_1=n}$ une loi de distribution; par conséquent, parler de valeurs moyennes, de variance, de corrélation entre deux paramètres a un sens, dès lors que l'on s'est restreint à une partie finie de D.

Nous nous bornerons donc, dans le cas général, à parler de valeur moyenne, ou de "loi", d'un paramètre, *connaissant la valeur d'un autre paramètre faisant office de taille*. L'absence de loi de probabilité privilégiée sur D nous interdira toutefois de parler de probabilités conditionnelles, bien que nous en utilisions à l'occasion les notations. Ainsi, la notation

$$P(A|\lambda_1 = n)$$

désigne la probabilité que l'événement A se produise lorsqu'un objet est pris aléatoirement et de manière uniforme parmi ceux qui vérifient $\lambda_1(w) = n$.

De même, nous utiliserons la notation

$$E(\lambda|\lambda_1 = n)$$

pour désigner l'espérance de $\lambda(w)$, lorsque l'objet w est pris aléatoirement et de manière uniforme parmi ceux qui vérifient $\lambda_1(w) = n$.

Les coefficients de la série D et de ses projections obtenues en fixant une ou plusieurs de ses variables à 1, contiennent toute l'information souhaitable sur ces lois conjointes.

Plus précisément, la probabilité pour un objet w de D, de taille $\lambda_1(w) = n$, d'avoir pour ses paramètres $(\lambda_i)_{2\leq i\leq k}$ les valeurs $(n_i)_{2\leq i\leq k}$, est

$$P\left(\lambda_2 = n_2, \ldots, \lambda_k = n_k | \lambda_1 = n\right) = \frac{\left[q_1^n q_2^{n_2} \cdots q_k^{n_k}\right] D(q_1, \ldots, q_k)}{\left[q_1^n\right] D(q_1)}. \tag{1}$$

De manière plus générale, nous pouvons choisir une sous-famille $(\lambda_i)_{i\in I}$ de paramètres fixés (avec la condition que chaque partie $D_{(\lambda_i)_{i\in I}=(n_i)_{i\in I}}$ soit finie), et une sous-famille $(\lambda_j)_{j\in J}$ de paramètres "libres". Les ensembles I et J doivent être disjoints : si un paramètre est fixé, il est inutile d'étudier sa répartition. Nous pouvons alors étudier la répartition des paramètres $(\lambda_j)_{j\in J}$. En posant $(q_i)_{i\in I}^{(n_i)_{i\in I}} = \prod_{i\in I} q_i^{n_i}$, on a :

$$P\left(\lambda_j = n_j, j \in J | \lambda_i = n_i, i \in I\right) = \frac{\left[(q_j)_{j\in J}^{(n_j)_{j\in J}} (q_i)_{i\in I}^{(n_i)_{i\in I}}\right] D\left(q_i, q_j\right)}{\left[(q_i)_{i\in I}^{(n_i)_{i\in I}}\right] D\left(q_i\right)}. \tag{2}$$

Dans la plupart des cas, le calcul de coefficients de séries à un grand nombre de variables est trop complexe pour être mené à bien, et ne donne pas de résultats "lisibles". On peut alors se contenter d'obtenir, pour un paramètre donné, sa moyenne, voire sa variance, et pour deux paramètres, leur covariance.

Les opérations essentielles pour le calcul de paramètres moyens sont l'extraction de coefficients d'une série génératrice, et la différentiation suivant une variable.

4.2.2 Différentiation

Soit $F = F(q_1, \ldots, q_k)$ la série génératrice, suivant k paramètres $\lambda_1, \ldots, \lambda_k$, d'un ensemble d'objets noté également F.

En dérivant formellement, par rapport à la variable q_i, puis en multipliant par q_i, la définition de la série

$$F(q_1, \ldots, q_k) = \sum_{w\in F} q_1^{\lambda_1(w)} \cdots q_k^{\lambda_k(w)},$$

on obtient

$$q_i \frac{\partial}{\partial q_i} F(q_1, \ldots, q_k) = \sum_{w\in F} \lambda_i(w) q_1^{\lambda_1(w)} \cdots q_k^{\lambda_k(w)}. \tag{3}$$

Cette nouvelle série peut être interprétée comme la série génératrice de F avec comme valuation, pour chaque objet w, $v(w) = \lambda_i(w) v_0(w)$, si $v_0(w)$ est la valuation initiale. C'est également la série génératrice, suivant les mêmes paramètres, d'un nouvel ensemble d'objets F', obtenu en remplaçant chaque objet w par $\lambda_i(w)$ copies de lui-même. Si le paramètre

λ_i compte des éléments distincts de w, cela correspond, pour obtenir F', à distinguer de toutes les façons possibles l'un de ces éléments : ainsi, si par exemple F est l'ensemble des chemins de Dyck, et λ_i le paramètre "nombre de pics", F' est l'ensemble des chemins de Dyck (non vides) dont un pic a été distingué.

Définition 4.1. Soient $q_1, \ldots, q_n$ des variables formelles. L'opérateur Δ_{q_i} est défini sur l'algèbre de séries formelles $\mathbb{Q}[[q_1, \ldots, q_n]]$ par

$$\Delta_{q_i} F(q_1, \ldots, q_n) = q_i \frac{\partial F}{\partial q_i}(q_1, \ldots, q_n) \tag{4}$$

Notation 4.2. Lorsqu'il sera nécessaire de composer les opérateurs Δ, nous noterons de manière générale,

$$\Delta_{q_{i_1} \ldots q_{i_k}} = \Delta_{q_{i_1}} \circ \cdots \circ \Delta_{q_{i_k}}.$$

Les opérateurs Δ_{q_i} commutent entre eux, et sont des dérivations sur l'algèbre de séries formelles $\mathbb{Q}[[q_1, \ldots, q_n]]$. Leur interprétation sur les séries génératrices est donnée par la proposition suivante :

Proposition 4.3. *Lorsque $F(q_1, \ldots, q_k)$ est la série génératrice de F suivant la valuation $v_0(w) = q_1^{\lambda_1(w)} \ldots q_k^{\lambda_k(w)}$, la série $\Delta_{q_{i_1} \ldots q_{i_m}} F$ est la série génératrice de F avec valuation*

$$v(w) = \left(\prod_{j=1}^{m} \lambda_{i_j}(w) \right) v_0(w).$$

Preuve. La proposition est immédiatement prouvée par récurrence sur m, chaque application de Δ_{q_i} multipliant la valuation de chaque objet w par $\lambda_i(w)$. □

La proposition 4.3 permet de comprendre pourquoi il est avantageux de considérer comme opérateurs, $\Delta_{q_i} = q_i \frac{\partial}{\partial q_i}$ plutôt que $\frac{\partial}{\partial q_i}$: l'utilisation de ces derniers compliquerait l'interprétation des composées de tels opérateurs. Avec deux variables q_1 et q_2, comptant respectivement des paramètres λ_1 et λ_2,

$$\begin{aligned} [q_1^{n_1} q_2^{n_2}] \left(\Delta_{q_2}\right)^k F(q_1, q_2) &= n_2^k \, [q_1^{n_1} q_2^{n_2}] \, F(q_1, q_2), \\ [q_1^{n_1} q_2^{n_2}] \left(\frac{\partial}{\partial q_2}\right)^k F(q_1, q_2) &= (n_2 + 1) \ldots (n_2 + k) \left[q_1^{n_1} q_2^{n_2 + k} \right] F(q_1, q_2). \end{aligned}$$

Ainsi, la première forme, plus simple, est préférable.

Cette proposition se traduit, en termes de coefficients de séries, de la manière suivante :

Proposition 4.4. *Soit $F(q_1, \dots, q_n)$ la série génératrice de F suivant les paramètres $\lambda_1, \dots, \lambda_n$. Alors, pour tout paramètre $\lambda = \lambda_{i_1}^{\alpha_1} \dots \lambda_k^{\alpha_{i_k}}$, la moyenne de λ sur $F_{\lambda_1=m}$ est*

$$E(\lambda(w)|\lambda_1(w) = m) = \frac{[q_1^m]\Delta_{i_1}^{\alpha_1} \dots \Delta_{i_k}^{\alpha_k} F(q_1, 1, \dots, 1)}{[q_1^m]F(q_1, 1, \dots, 1)}. \tag{5}$$

Preuve. Il est clair, d'après la proposition (4.3), que l'on a

$$[q_1^m]\Delta_{i_1}^{\alpha_1} \dots \Delta_{i_k}^{\alpha_k} F(q_1, 1, \dots, 1) = \sum_{w \in F_{\lambda_1=m}} \lambda(w). \tag{6}$$

Puisque le coefficient $[q_1^m]F(q_1, 1, \dots, 1)$ est exactement le cardinal de $F_{\lambda_1=m}$, nous obtenons directement

$$\frac{[q_1^m]\Delta_{i_1}^{\alpha_1} \dots \Delta_{i_k}^{\alpha_k} F(q_1, 1, \dots, 1)}{[q_1^m]F(q_1, 1, \dots, 1)} = \sum_{w \in F_{\lambda_1=m}} \lambda(w) P(w|\lambda_1(w) = m),$$

qui donne bien l'espérance du paramètre λ. □

Dans la pratique, il est fréquent de fixer à 1 toutes les variables formelles d'énumération, sauf la première, qui compte la taille des objets. Nous noterons π la projection de l'algèbre de séries formelles $\mathbb{Q}[[q_1, \dots, q_k]]$ dans $\mathbb{Q}[[q_1]]$ [1]; pour chaque opérateur Δ obtenu en composant des opérateurs Δ_{q_i}, nous noterons $\Delta' = \pi \circ \Delta$. Dans le cadre des Q-grammaires, les relations que nous obtiendrons en associant opérateurs Δ et substitutions de variables se simplifieront nettement en remplaçant l'opérateur Δ par Δ'.

Définition 4.5. Lorsque $U(q_1, \dots, q_n)$ est la série génératrice d'un langage L suivant des parmètres $\lambda_1, \dots, \lambda_n$, la série $\Delta_i'(U)$ est appelée *série de moments* du langage L suivant le paramètre λ_i.

4.2.3 Calculs asymptotiques

Il est particulièrement instructif de s'intéresser au comportement de la valeur moyenne d'un paramètre pour les objets de taille n, lorsque n tend vers $+\infty$; c'est le principe de l'analyse asymptotique de paramètres.

En général, on cherche pour les coefficients a_n d'une série génératrice, un développement asymptotique dans l'échelle des fonctions

$$\mathcal{H}_{\mu,\theta}(n) = \mu^n n^{-\theta}. \tag{7}$$

1. π n'est pas définie pour *toutes* les séries formelles, mais seulement sur une sous-algèbre qui contient les séries génératrices pour lesquelles q_1 compte une taille.

Dans la pratique, on cherche un développement de la forme

$$a_n = \mu^n P(1/n) + O(\mu^n n^{-\theta}), \tag{8}$$

où $P(x) = C_1 x^{\theta_1} + \cdots + C_k x^{\theta_k}$, avec $\theta_1 < \ldots < \theta_k < \theta$ (P n'est pas forcément un polynôme, les exposants θ_i n'étant généralement pas entiers).

La règle de Cauchy-Hadamard permet d'affirmer que, lorsqu'un tel développement existe, $1/\mu$ est le rayon de convergence de la série génératrice. Cette série ayant des coefficients positifs, $1/\mu$ en est alors une singularité dominante. De plus, si la série est algébrique, μ est forcément un nombre algébrique, et les exposants θ_i sont rationnels [57, 40].

Une variante relativement fréquente de cette forme consiste à séparer les coefficients suivant différentes progressions arithmétiques. Ainsi, lorsque la série $f(x)$ peut s'écrire $f(x) = g(x^2)$ pour une autre série g, seuls les coefficients d'indice pair de f peuvent vérifier une relation de la forme (8); les coefficients d'indice impair sont évidemment nuls. Le plus souvent, toutefois, de telles propriétés sont détectées à l'avance, et la série génératrice considérée est $g(x)$ plutôt que $f(x)$. Ainsi, la série génératrice habituellement considérée pour les mots de Dyck est $f(x) = \sum_n C_n x^n$, où le n-ième nombre de Catalan C_n est le nombre de mots de Dyck de longueur $2n$.

En utilisant la proposition (4.3), le calcul asymptotique de la valeur moyenne d'un paramètre sur les objets de taille n, lorsque n tend vers $+\infty$, se ramène en fait au calcul de développements asymptotiques pour les coefficients de deux séries : la série génératrice $D(q_1)$ suivant la taille, et une série $(\Delta'_{q_i} D)(q_1)$. Le quotient de ces deux développements asymptotiques, donnera le comportement de la valeur moyenne lorsque n tend vers $+\infty$.

Un avantage de l'échelle de développements asymptotiques de la forme (8) est qu'un quotient de deux telles expressions se présente sous une forme semblable. Dans le cas de calcul de valeurs moyennes de paramètres, ceci permet d'obtenir des expressions asymptotiques simples pour ces valeurs moyennes.

Sous certaines conditions, le comportement asymptotique des coefficients d'une série à une seule variable, peut être décrit assez précisément en examinant les singularités de la fonction analytique définie par cette série (voir par exemple [45]).

La recherche d'expressions asymptotiques pour les coefficients d'une série génératrice peut généralement être résumée ainsi :

– Déterminer les singularités dominantes (de plus petit module) de la série. Lorsque les coefficients de la série sont positifs, l'une au moins de ces singularités est réelle positive; très souvent, on obtient une singularité dominante unique ρ.

– Déterminer un développement asymptotique de la série au voisinage de la singularité dominante; pour des séries algébriques, on obtient une expression de la forme $F(x) = F_0 + C.(\rho - x)^\alpha + o\left((\rho - x)^\alpha\right)$, où α est un nombre rationnel.

– A partir d'une telle expression, il est généralement possible d'écrire automatiquement le développement de a_n.

Ce processus est automatisé dans le logiciel $\Lambda_\Upsilon\Omega$ (`luo`) [42, 43], conçu pour l'analyse en moyenne de *structures décomposables*, dont les langages algébriques sont un cas particulier. Ce logiciel permet donc, dans le cadre des paramètres Q-comptables, de traiter les paramètres de rang 1.

Recherche de singularités : développements de Puiseux

Nous donnons ici un aperçu de la méthode du *polygone de Newton* utilisée pour obtenir le *développement de Puiseux* d'une série algébrique d'une variable. On trouvera une description plus complète dans l'ouvrage de Dieudonné [33].

Nous nous plaçons dans le cas où la série $F(x)$ dont nous recherchons les singularités est algébrique, et ne dépend que d'une seule variable x. Elle est donc solution d'une équation

$$P(x, F(x)) = 0 \tag{9}$$

où $P(x, y)$ est un polynôme.

Le théorème des fonctions implicites prévoit qu'au voisinage de tout point (x_0, y_0) tel que $P(x_0, y_0) = 0$, l'équation (9) admet une solution analytique dès que $\frac{\partial}{\partial y}P(x_0, y_0) \neq 0$. Par conséquent, pour que la série F ait une singularité en x_0, il faut que (x_0, y_0) soit solution de

$$\left\{ \begin{array}{rcl} 0 & = & P(x, y) \\ 0 & = & \frac{\partial P}{\partial y}(x, y) \end{array} \right. \tag{10}$$

Pour rechercher la (ou les) singularités dominantes, il faut chercher, parmi les solutions de (10), celles qui ont une coordonnée x de module minimal *et* qui correspondent à une singularité d'une branche analytique à l'origine de la solution de (9). Cette partie du problème ne peut être résolue de manière automatique dans le cas général; toutefois, dans le cas de séries génératrices (qui ne font intervenir que des coefficients positifs ou nuls), l'une au moins des singularités dominantes est assurée d'être un nombre réel positif, ce qui peut aider à éliminer les "fausses" solutions.

Par un changement de variables de la forme

$$\left\{ \begin{array}{rcl} F & = & y_0 - f, \\ x & = & x_0 - h, \end{array} \right.$$

on se ramène à chercher une solution au voisinage de $(0,0)$ de l'équation

$$P(x_0 - h, y_0 - f) = 0. \tag{11}$$

Le fait que la solution recherchée soit une série génératrice permet de ne prendre en compte que des solutions pour $h \geq 0$ et $f \geq 0$, car la fonction $F(x)$ est forcément croissante sur $[0, x_0]$.

L'équation (11) se développe sous la forme

$$0 = \sum_{i=1}^{k} c_i h^{\alpha_i} f^{\beta_i}, \tag{12}$$

où les coefficients c_i sont non nuls, et les exposants α_i et β_i, entiers positifs, ne peuvent être nuls simultanément. En factorisant cette équation par $h^{\alpha_0} f^{\beta_0}$, on peut également supposer que l'un au moins des exposants α_i est nul, ainsi que l'un des exposants β_i.

Plaçons dans le plan les points $(A_i)_{1 \leq i \leq k}$ de coordonnées respectives (α_i, β_i). Supposons que les indices sont ordonnés de telle sorte que A_1 est le point le plus bas sur l'axe $\alpha = 0$ ($\alpha_1 = 0$ et $\beta_1 \leq \beta_i$ pour tout i tel que $\alpha_i = 0$), A_p est le point le plus à gauche sur l'axe $\beta = 0$ ($\beta_p = 0$ et $\alpha_p \leq \alpha_i$ pour tout i tel que $\beta_i = 0$), et le parcours dans le sens trigonométrique de l'enveloppe convexe du nuage de points $(A_i)_{1 \leq i \leq k}$, entre A_1 et A_p, passe par les points $A_2, \ldots, A_{p-1}$. Cette ligne polygonale $(A_1, \ldots, A_p)$ est appelée *polygone de Newton* de l'équation (11).

Pour chaque segment de ce polygone de Newton, on obtient, en ne conservant de (12) que les termes qui correspondent à des points situés sur ce segment (le plus souvent, il n'y a que 2 tels points pour chaque segment), une équation approchée dont la résolution donne le premier terme d'un développement asymptotique d'une branche de la solution de (11), à condition que ce développement soit compatible avec les hypothèses sur le signe de f et h. La forme générale de ce développement asymptotique pour le segment $[A_i, A_j]$, est

$$f = C.h^{\rho} + o\left(h^{\rho}\right), \tag{13}$$

où ρ est le nombre rationnel solution de l'équation

$$\alpha_i + \beta_i \rho = \alpha_j + \beta_j.\rho.$$

En théorie, le polygone de Newton peut donner un assez grand nombre de branches. Dans la pratique, il est assez fréquent que l'équation (12) comporte un terme en $h^1 f^0$ et un terme en $h^0 f^2$, auquel cas $A_1 = (0,2)$, $A_2 = (1,0)$, et le polygone de Newton se réduit au segment $[A_1 A_2]$. On a alors une branche unique avec comportement $f = C.h^{1/2} + o(h^{1/2})$.

4.3 Un exemple de calcul de moyenne

Nous donnons ici un exemple de calcul de valeur moyenne de paramètre, en reprenant l'étude de l'aire des chemins de Dyck. Bien que la série génératrice des chemins de Dyck soit un exemple extrêmement classique de série qu'il est possible de calculer explicitement et dont les coefficients s'expriment simplement, tout le travail asymptotique peut être fait sans connaître d'expressions exactes.

Les calculs qui suivent ne présentent aucune difficulté technique. Toutefois, la forme des résultats peut être généralisée à n'importe quel système de Q-équations données par une Q-grammaire, comme nous le verrons par la suite.

La série génératrice des chemins de Dyck comptés suivant la demi-longueur (par x) et l'aire (par q) vérifie l'équation :

$$D(x,q) = 1 + xqD(xq^2, q)D(x,q) \tag{14}$$

En remplaçant q par 1 dans cette équation, on retrouve l'équation algébrique satisfaite par la série génératrice suivant la demie longueur seule :

$$D(x) = 1 + xD(x)^2. \tag{15}$$

Les coefficients de la série génératrice suivant la demi-longueur seule sont bien connus : il s'agit des nombres de Catalan $C_n = \frac{1}{n+1}\binom{2n}{n}$.

L'équation algébrique (15) s'écrit sous la forme implicite

$$P(x, D(x)) = 0,$$

avec $P(x,y) = 1 - y + xy^2$. Les singularités des fonctions solutions sont donc solutions du système

$$0 = P(x,y) = \frac{\partial P}{\partial y}(x,y).$$

Dans notre cas, ce système s'écrit

$$0 = 1 - y + xy^2 = -1 + 2xy,$$

et a pour solution unique, $(x_0, y_0) = (1/4, 2)$. La branche de la solution de (15) qui nous intéresse doit être analytique au voisinage de 0, et être croissante sur $[0, 1/4]$; par conséquent, nous pouvons écrire $D(1/4 - h) = 2 - d(h)$, avec $h \geq 0$ et $d(h) \geq 0$: le développement asymptotique de $D(x)$ au voisinage de $1/4$ nous sera donné par celui de $d(h)$ au voisinage de 0.

En écrivant $P(1/4 - h, 2 - d)$, nous obtenons à partir de (15),

$$0 = \frac{1}{4}d^2 - 4h + 4hd - hd^2. \tag{16}$$

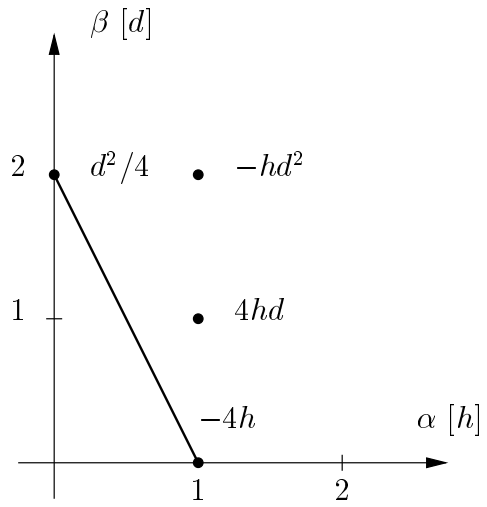


FIG. 4.1: *Polygone de Newton de l'équation* $d^2/4 - 4h + 4hd - hd^2$

Le polygone de Newton de l'équation (16) se réduit au segment $\alpha + \beta/2 = 1, \alpha \geq 0, \beta \geq 0$ (voir figure 4.1), ce qui nous donne immédiatement le développement asymptotique de $d(h)$:

$$d = 4h^{1/2} + o\left(h^{1/2}\right).$$

Le développement asymptotique de la série $D(x)$ au voisinage de sa singularité dominante[2] est par conséquent,

$$D(x) = 2 - 4\sqrt{\frac{1}{4} - x} + o\left(\sqrt{\frac{1}{4} - x}\right) = 2 - 2\sqrt{1 - 4x} + o\left(\sqrt{1 - 4x}\right).$$

2. Bien entendu, un tel développement eût été plus simple à obtenir à partir de la formule close $D(x) = \left(1 - \sqrt{1 - 4x}\right)/2x$.

De cette expression, on déduit immédiatement une forme asymptotique des coefficients de la série génératrice $D(x)$:

$$C_n \sim \frac{4^n n^{-3/2}}{\sqrt{\pi}}.$$

Pour obtenir l'aire moyenne des chemins de Dyck de longueur $2n$, il nous faut reprendre l'équation (14), et dériver par rapport à q. En notant $D_x(x) = \partial D/\partial x(x,q)$ et $D_q(x) = \partial D/\partial q(x,q)$ (la deuxième variable q est implicite), nous obtenons :

$$D_q(x) = xD(xq^2)D(x) + xqD(x)\left(D_q(xq^2) + 2xqD_x(xq^2)\right) + xqD(xq^2)D_q(x). \tag{17}$$

Lorsque $q = 1$, $D_x(x,q)$ devient la dérivée de la série génératrice à une seule variable $D(x) = D(x,1)$; en dérivant (15), il vient :

$$\begin{aligned} D'(x) &= D^2(x) + 2xD(x)D'(x) \\ &= \frac{D^2(x)}{1-2xD(x)}. \end{aligned}$$

Par conséquent, (17) devient pour $q = 1$,

$$\begin{aligned} D_q(x,1) &= xD^2(x,1) + xD(x,1)\left(D_q(x,1) + 2x\frac{D^2(x,1)}{1-2xD(x,1)}\right) \\ &\quad + xD(x,1)D_q(x,1) \\ &= \frac{xD^2(x,1)}{1-2xD(x,1)} + \frac{2x^2D^3(x,1)}{(1-2xD(x,1))^2} \\ &= \frac{xD^2(x,1)}{(1-2xD(x,1))^2}. \end{aligned}$$

Remarquons que $D_q(x,1)$, tout comme $D_x(x,1)$, peut s'écrire comme une fraction rationnelle en x et $D(x,1)$. Ce n'est pas le cas de la série à deux variables $D_q(x,q)$. En connaissant le développement asymptotique de $D(x,1)$ au voisinage de $x = 1/4$, on en déduit donc celui de $D_q(x,1)$:

$$D_q(x,1) \sim \frac{1}{1-4x}. \tag{18}$$

On en déduit immédiatement une expression asymptotique pour les coefficients de $D_q(x,1)$ (à savoir, $[x^n]D_q(x) \sim 4^n$), et surtout un équivalent de l'aire moyenne $\overline{A}_n$ des chemins de Dyck de longueur $2n$:

$$\overline{A}_n \sim \sqrt{\pi} n^{3/2}. \tag{19}$$

Bien entendu, en acceptant d'utiliser la forme exacte de $D(x,1)$, et en posant $J(x) = 1-2xD(x,1) = \sqrt{1-4x}$, on peut aisément exprimer $D_x(x,1)$ et $D_q(x,1)$ comme fractions rationnelles en J :

$$D_x(x,1) = \frac{1-J}{J(1+J)}, \tag{20}$$

$$D_q(x,1) = \frac{1-J}{J^2(1+J)}, \tag{21}$$

ce qui permet d'obtenir une expression exacte pour les coefficients de $D_q(x,1)$:

$$D_q(x,1) = \sum_{n\geq 0} \left(4^n - (2n+1)C_n\right) x^n.$$

On retrouve ici le résultat de Chottin et Cori [20] (ajouter $(2n+1)C_n$ pour obtenir 4^n, correspondrait à surélever de 1 chacun des $2n+1$ sommets des chemins, ce qui revient à compter les ordonnées à partir de 1 et non 0).

Nous pouvons pousser plus loin les calculs portant sur l'aire, afin d'obtenir également la *variance* de l'aire des chemins de Dyck de longueur $2n$. En utilisant les opérateurs Δ et Δ' définis en 4.2.2, le moment d'ordre 2 de l'aire est donné par les coefficients de $(\Delta'_{qq}D)(x)$. L'équation (17) peut se réécrire en :

$$(\Delta_q D)(x,q) = xq(q+1)D(x,q)D(xq^2,q) + xqD(x,q)\left((\Delta_q D)(xq^2,q) + (\Delta_x D)(xq^2,q)\right) \tag{22}$$

Appliquer Δ_q à cette equation fait apparaître les séries $\Delta_{xx}D(x)$, $\Delta_{xx}D(xq^2)$, $\Delta_{xq}D(x)$ et $\Delta_{xq}D(xq^2)$. En fixant $q=1$, nous n'avons plus que $\Delta'_{xx}D(x)$ et $\Delta'_{xq}D(x)$. Nous devons donc également calculer ces séries, de la même manière que fait précédemment pour $\Delta'_q D(x) = D_d(x)$.

Pour calculer $\Delta'_{xx}D(x)$ et $\Delta'_{xq}D(x)$, il suffit en fait de travailler avec des séries à une variable, en appliquant Δ_x aux équations donnant $\Delta'_x D(x)$ et $\Delta'_q D(x)$, respectivement. Pour obtenir $\Delta'_{qq}D(x)$, en revanche, il est indispensable de conserver la variable q et de ne la faire disparaître qu'à la dernière étape du calcul.

Une fois de plus, la deuxième variable q est implicite dans les expressions qui suivent. F désigne $F(x,q)$; $F(xq)$ désigne $F(xq,q)$, et $F(xq^2)$ représente $F(xq^2,q)$. Les équations

obtenues sont :

$$\begin{aligned}
\Delta'_{xx}D(x) &= xD^2(x) + 4xD(x)\Delta'_xD(x) + 2x\left(\Delta'_xD(x)\right)^2 + 2xD(x)\Delta'_{xx}D(x)\\
&= \frac{xD^2(x) + 4xD(x)\Delta'_xD(x)}{1-2xD(x)},\\
\Delta'_{xq}D(x) &= xD^2(x) + 4xD(x)\Delta'_xD(x) + 2xD(x)\Delta'_qD(x) + 2x\Delta'_xD(x)\Delta'_qD(x)\\
&\quad +2xD\Delta'_{xq}D(x) + 2x\left(\Delta'_xD(x)\right) + 2xD(x)\Delta'_{xx}D(x)\\
&= x\frac{D(x)\left(D(x) + 4\Delta'_xD(x) + 2\Delta'_{xx}D(x)\right) + 2\Delta'_xD(x)\left(\Delta'_xD(x) + \Delta'_qD(x)\right)}{1-2xD(x)},\\
\Delta_{qq}D &= xq\left(D.D(xq^2) + \Delta_qD.D(xq^2) + D.\Delta_qD(xq^2) + 2D.\Delta_xD(xq^2)\right.\\
&\quad +\Delta_qD.D(xq^2) + \Delta_{qq}D.D(xq^2) + \Delta_qD.\Delta_qD(xq^2) + 2\Delta_qD.\Delta_xD(xq^2)\\
&\quad +D.\Delta_qD(xq^2) + \Delta_qD.\Delta_qD(xq^2) + D.\Delta_{qq}D(xq^2) + 2D.\Delta_{xq}D(xq^2)\\
&\quad \left.+2D.\Delta_xD(xq^2) + 2\Delta_qD.\Delta_xD(xq^2) + 2D.\Delta_{xq}D(xq^2) + 4D.\Delta_{xx}D(xq^2)\right).
\end{aligned}$$

La dernière équation se simplifie quelque peu lorsque l'on fixe $q = 1$, et donne :

$$\Delta'_{qq}D = x\frac{D.\left(D + 4\Delta'_xD + 4\Delta'_qD + 4\Delta'_{xq}D + 4\Delta'_{xx}D\right) + 2\Delta'_qD.\left(\Delta'_qD + 2\Delta'_xD\right)}{1-2xD}.$$

Une fois de plus, en posant $J(x) = 1 - 2xD(x)$, les trois séries que nous venons de calculer s'expriment comme fractions rationnelles en J :

$$\Delta'_{xx}D = \tfrac{(1-J)(1+2J-J^2)}{2J^3(1+J)}, \tag{23}$$

$$\Delta'_{xq}D = \tfrac{(1-J)(1+J-J^2)}{J^4(1+J)}, \tag{24}$$

$$\Delta'_{qq}D = \tfrac{(1-J)(5+2J-5J^2)}{2J^5(1+J)}. \tag{25}$$

L'équation (25) nous permettrait d'obtenir une expression exacte pour les coefficients de $\Delta'_{qq}D(x)$, et, partant, pour le moment d'ordre 2 de l'aire des chemins de Dyck de longueur $2n$. Nous nous contenterons d'une version asymptotique :

$$\Delta_{qq}D(x) \sim \frac{5}{2}(1-4x)^{-5/2}, \tag{26}$$

$$[x^n]\,\Delta_{qq}D(x) \sim \frac{10}{3\sqrt{\pi}}4^n n^{3/2}. \tag{27}$$

Par conséquent, l'écart-type σ_n de l'aire des chemins de Dyck de longueur $2n$, est asymptotiquement,

$$\sigma_n \sim n^{3/2}\sqrt{\frac{10}{3} - \pi}. \tag{28}$$

L'écart-type du paramètre aire est donc du même ordre de grandeur que l'espérance du paramètre lui-même. En conséquence, ce paramètre ne peut avoir, après normalisation ($A^* = (A - \mu_n)/\sigma_n$), une loi limite gaussienne, puisque le support de A^* reste borné inférieurement.

4.4 Opérateurs Δ et substitutions de variables

La série génératrice d'un langage algébrique suivant les différents paramètres Q-comptables d'une Q-grammaire, ne nous est a priori connue que par l'intermédiaire d'un système de Q-équations faisant intervenir des substitutions de variables σ.

Lors du calcul précédent, nous avons pu voir que tout se résume à appliquer des opérateurs Δ ou Δ' à des équations faisant intervenir des substitutions de variables (dans le cas de l'aire des chemins de Dyck, la seule substitution est $\sigma_{x\leftarrow xq^2}$). L'ensemble des calculs est simplifié lorsque l'on remarque que, pour une série $F(x,q)$ quelconque,

$$\begin{aligned}\left(\Delta_x \circ \sigma_{x\leftarrow xq^2}\right) F &= \left(\sigma_{x\leftarrow xq^2} \circ \Delta_x\right) F \\ \left(\Delta_q \circ \sigma_{x\leftarrow xq^2}\right) F &= \left(\sigma_{x\leftarrow xq^2} \circ \Delta_q\right) F + 2\left(\sigma_{x\leftarrow xq^2} \circ \Delta_x\right) F.\end{aligned}$$

Il est donc naturel d'examiner en toute généralité quels sont les liens possibles entre les opérateurs σ et les opérateurs Δ, et les conséquences que nous pouvons en tirer sur les statistiques de paramètres Q-comptables.

Lemme 4.6. *Soient $q_1, \ldots, q_n$ des variables formelles. Notons $\sigma_{i,j} = \sigma_{q_i\leftarrow q_iq_j}$ (pour $1 \leq i < j \leq n$). On a alors :*

- *si $k \neq j$, Δ_{q_k} et $\sigma_{i,j}$ commutent;*
- $\Delta_j\sigma_{i,j} = \sigma_{i,j}\Delta_j + \sigma_{i,j}\Delta_{q_i}$

Preuve. Il est clair que, pour n'importe quelle série formelle $U = U(q_1, \ldots, q_n)$, lorsque k est distinct de i et de j, $\Delta_{q_k}\sigma_{i,j}U = \sigma_{i,j}\Delta_{q_k}U$. Il reste à examiner les cas $k = i$ et $k = j$.

Lorsque $k = i$, le calcul est très simple :

$$\begin{aligned}\frac{\partial}{\partial q_i}\sigma_{i,j}U &= \frac{\partial}{\partial q_i}U(q_1, \ldots, q_{i-1}, q_iq_j, q_{i+1}, \ldots, q_n) \\ &= q_j\frac{\partial U}{\partial q_i}(q_1, \ldots, q_{i-1}, q_iq_j, q_{i+1}, \ldots, q_n) \\ \Delta_{q_i}\sigma_{i,j}U &= q_i\frac{\partial}{\partial q_i}\sigma_{i,i}U \\ &= q_iq_j\frac{\partial U}{\partial q_i}(q_1, \ldots, q_{i-1}, q_iq_j, q_{i+1}, \ldots, q_n) \\ &= \sigma_{i,i}\Delta_{q_i}U\end{aligned}$$

Dans le cas où $k = j$, on obtient :

$$\begin{aligned}
\frac{\partial}{\partial q_j}\sigma_{i,j}U &= \frac{\partial}{\partial q_j}U(q_1,\dots,q_{i-1},q_iqj,q_{i+1},\dots,q_n) \\
&= \frac{\partial U}{\partial q_j}(q_1,\dots,q_{i-1},q_iq_j,q_{i+1},\dots,q_n) + q_i\frac{\partial U}{\partial q_i}(q_1,\dots,q_{i-1},q_iq_j,q_{i+1},\dots,q_n) \\
\Delta_{q_j}\sigma_{i,j}U &= q_j\frac{\partial U}{\partial q_j}(q_1,\dots,q_{i-1},q_iqj,q_{i+1},\dots,q_n) + q_iq_j\frac{\partial U}{\partial q_i}(q_1,\dots,q_iq_j,\dots,q_n) \\
&= \sigma_{i,j}\Delta_{q_i}U + \sigma_{i,j}\Delta_{q_j}U
\end{aligned}$$

□

Le résultat du lemme (4.6) peut également s'écrire sous forme matricielle, à condition de définir quelques notations.

Notons $\overrightarrow{\Delta}$ le vecteur-ligne d'opérateurs $(\Delta_{q_1},\dots,\Delta_{q_n})$. Le produit d'un vecteur-ligne avec une matrice de substitution de variables se fait de la manière usuelle, de telle sorte que

$$(\Delta_{q_1},\Delta_{q_2},\Delta_{q_3})\begin{pmatrix}1&1&1\\0&1&2\\0&0&1\end{pmatrix} = (\Delta_{q_1},\Delta_{q_1}+\Delta_{q_2},\Delta_{q_1}+2\Delta_{q_2}+\Delta_{q_3}).$$

Enfin, si F est une série formelle, σ une substitution de variables, $\overrightarrow{\Delta} = (\Delta_1,\dots,\Delta_n)$ un vecteur-ligne d'opérateurs, et $(U_1,\dots,U_n)$ un vecteur-ligne de séries formelles, les calculs suivants se distribuent composante par composante :

$$\begin{aligned}
\overrightarrow{\Delta}(U) &= (\Delta_1(U),\dots,\Delta_n(U)); \\
\sigma\,(U_1,\dots,U_n) &= (\sigma(U_1),\dots,\sigma(U_n))\,.
\end{aligned}$$

Alors, le lemme (4.6) peut s'exprimer de la manière suivante :

$$\overrightarrow{\Delta}\,(\sigma_{ij}(U)) = \left(\sigma_{ij}\circ\overrightarrow{\Delta}\right)(U).$$

Cette formulation est en fait valable quelle que soit la substitution de variables σ. En effet, toute matrice triangulaire supérieure, à coefficients entiers positifs, ne présentant que des 1 sur sa diagonale, peut s'écrire comme produit de matrices de la forme $M_{i,j}$; ou, ce qui revient au même, toute substitution de variables peut s'écrire comme composée de substitutions de la forme $\sigma_{i,j}$. Le lemme (4.6) s'étend alors, par récurrence, à n'importe quelle substitution de variables :

Proposition 4.7. *Soit σ une substitution de variables, représentée par la matrice M. Alors*

$$\overrightarrow{\Delta}.(\sigma U) = \sigma\left(\left(\overrightarrow{\Delta}.M\right)(U)\right) \tag{29}$$

Il est peut-être plus simple d'exprimer ce résultat en fonction des coefficients de la matrice : si $M = (a_{i,j})_{1 \leq i,j \leq n}$, l'équation (29) s'écrit

$$\Delta_{q_j}(\sigma U) = \sigma\left(\sum_{i=1}^{j} a_{i,j} \Delta_{q_i} U\right). \tag{30}$$

Pour obtenir des résultats statistiques, nous devons restreindre nos séries génératrices à une seule variable, ce qui se fait par l'intermédiaire de l'opérateur de projection π. Nous devons donc donner de la proposition (4.7) une version ne concernant que les opérateurs Δ'.

Remarquons que, pour toute substitution de variables σ, on a $\pi \circ \sigma = \pi$. Cette remarque nous donne l'équivalent de la proposition (4.7) pour les opérateurs Δ' :

Proposition 4.8. *Soit σ une substitution de variables, représentée par la matrice M. Notons, pour une série formelle $F = F(q_1, \ldots, q_n)$, $\overrightarrow{\Delta'}$ le vecteur-ligne $(\Delta'_{q_1}, \ldots, \Delta'_{q_n})$. Alors*

$$\overrightarrow{\Delta'}(\sigma(F)) = \left(\overrightarrow{\Delta'}.M\right)(F) \tag{31}$$

Preuve. L'équation (31) s'obtient directement à partir de (29) en appliquant π de part et d'autre. □

Exemple 4.9. Considérons une série formelle $F(x, y, z)$, et la substitution de variables $\sigma = \sigma_{x \leftarrow xy, y \leftarrow yz^2}$, représentée par la matrice

$$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

La proposition (4.8) implique :

$$\begin{aligned} \left(\Delta'_z \circ \sigma\right)(F) &= \Delta'_z(F) + 2\Delta'_y(F) \\ \left(\Delta'_y \circ \sigma\right)(F) &= \Delta'_y(F) + \Delta'_x(F) \\ \left(\Delta'_x \circ \sigma\right)(F) &= \Delta'_x(F) \\ &= x\frac{d}{dx}(\pi(F)) \end{aligned}$$

Remarque. La relation $\pi \circ \sigma = \pi$, qui permet d'obtenir la proposition (4.8), est également vraie si π n'est pas la projection dans $\mathbb{Q}[[q_1]]$, mais dans $\mathbb{Q}[[q_1, \ldots, q_{n'}]]$, à condition que

les substitutions σ considérées, lorsqu'elles sont réduites aux variables $q_1, \dots, q_{n'}$, soient l'identité. Dans le cadre des substitutions apparaissant dans le système de Q-équations d'une Q-grammaire, c'est le cas lorsque les variables formelles $q_1, \dots, q_{n'}$ comptent toutes des paramètres de rang 1.

Chaque fois qu'une telle relation est vraie, la proposition (4.8) l'est aussi.

La proposition (4.8) exprime $\Delta'_{x_j}\sigma$ comme combinaison linéaire des Δ'_{x_i} $(i \leq j)$, les coefficients étant ceux de la j-ème colonne de la matrice M_σ. Ainsi, le coefficient de Δ'_{x_j} est toujours 1.

Dans la pratique, la proposition (4.8), plus simple que (4.7), sera la plus utile; la proposition (4.7) est toutefois indispensable lorsqu'il s'agit de différencier plus d'une fois (pour un calcul de moment d'ordre 2 par exemple).

4.5 Moyennes de paramètres Q-comptables

Nous en venons maintenant à l'évaluation des valeurs moyennes de paramètres Q-comptables.

4.5.1 Cas général

Supposons donnée une Q-grammaire G, d'axiome D_1, à m symboles $D_1, \dots, D_m$. Cette Q-grammaire nous fournit un système de m Q-équations portant sur les séries génératrices $D_i(q_1, \dots, q_n)$ des différents langages engendrés par la grammaire, suivant les n paramètres Q-comptables $\lambda_1, \dots, \lambda_n$.

Les n' premiers paramètres $\lambda_1, \dots, \lambda_{n'}$ sont supposés être de rang 1, et π désignera ici la projection de $\mathbb{Q}[[q_1, \dots, q_n]]$ dans $\mathbb{Q}[[q_1, \dots, q_{n'}]]$. Ainsi, pour chaque symbole D_j, $\pi(D_j)$ est la série génératrice (algébrique) du langage $L_G(D_j)$ suivant les n' paramètres de rang 1.

Pour chaque mot $w \in D_j$, le n'-uplet $(\lambda_1(w), \dots, \lambda_{n'}(w))$ sera appelé *composition* de w; cette composition nous servira de taille. Il est en effet fréquent que chacun des paramètres de rang 1 compte une lettre de l'alphabet.

Pour obtenir la valeur moyenne du paramètre λ_i sur les mots de composition $L = (l_1, \dots, l_{n'})$, il nous faut comparer les coefficients de $q_1^{l_1} \dots q_{n'}^{l_{n'}}$ dans les séries $\pi(D_1)$ et $\Delta'_{q_i}(D_1)$. Nous allons donc nous intéresser de près à $\Delta'_{q_i}(D_j)$.

Les m Q-équations peuvent, d'après le théorème (2.36), être présentées sous la forme

$$0 = \tilde{P}\left((q_j)_{1\leq j\leq n}, (D_j)_{1\leq j\leq m}, (\sigma_k(D_j))_{1\leq k\leq p, 1\leq j\leq m}\right). \tag{32}$$

Rappelons que, si chaque $\sigma_k(D_j)$ est remplacé par D_j, et si chaque variable qui n'est pas une des lettres de l'alphabet prend la valeur 1, l'équation redonne l'équation algébrique fournie par la grammaire, que nous écrivons sous la forme

$$0 = P\left((q_j)_{1\leq j\leq n'}, (D_j)_{1\leq j\leq m}\right). \tag{33}$$

En appliquant $\Delta_{q_i} = q_i \frac{\partial}{\partial q_i}$ à l'équation (32), il vient

$$0 = q_i \frac{\partial \tilde{P}}{\partial q_i} + \sum_{j=1}^{m} \Delta_{q_i}(D_j).\frac{\partial \tilde{P}}{\partial D_j} + \sum_{j=1}^{m}\sum_{k=1}^{p} \Delta_{q_i}\left(\sigma_k(D_j)\right) \frac{\partial \tilde{P}}{\partial \sigma_k(D_j)}. \tag{34}$$

En appliquant maintenant la projection π, cette équation devient

$$0 = \pi\left(q_i \frac{\partial \tilde{P}}{\partial q_i}\right) + \sum_{j=1}^{m} \Delta'_{q_i}(D_j).\pi\left(\frac{\partial \tilde{P}}{\partial D_j}\right) + \sum_{j=1}^{m}\sum_{k=1}^{p} \Delta'_{q_i}\left(\sigma_k(D_j)\right) \pi\left(\frac{\partial \tilde{P}}{\partial \sigma_k(D_j)}\right). \tag{35}$$

Ici, $\pi(P)$ doit être interprété de la manière suivante : chaque variable q_j avec $j > n'$ est remplacée par 1, et chaque variable $\sigma_k(D_j)$ est remplacée par D_j – ce qui correspond bien à l'idée que chaque variable q_i (pour $i > n'$) vaut 1. Cette projection transforme les polynômes $\tilde{P}_i$ du Q-système, en les polynômes P_i du système algébrique de départ. On a donc

$$\pi\left(\frac{\partial \tilde{P}}{\partial D_j}\right) + \sum_{k=1}^{p} \pi\left(\frac{\partial \tilde{P}}{\partial \sigma_k(D_j)}\right) = \frac{\partial P}{\partial D_j} \tag{36}$$

où $P(q_1, \ldots, q_{n'}, D_1, \ldots, D_m)$ est le polynôme du système algébrique de départ, dont est issu (lors du passage aux Q-équations) le polynôme $\tilde{P}$.

Or, la proposition (4.8) exprime chaque terme $\Delta'_{q_i}\left(\sigma_k(D_j)\right)$ de (35) comme combinaison linéaire, à coefficients entiers positifs, de termes $\Delta'_{q_\ell}(D_j)$, avec $\ell \leq i$, le coefficient de $\Delta'_{q_i}(D_j)$ étant 1. Par conséquent, on peut réécrire (35) en regroupant d'une part les termes faisant intervenir Δ'_{q_i}, et d'autre part ceux faisant intervenir $\Delta'_{q_{i'}}$ avec $i' < i$:

$$0 = \pi\left(q_i \frac{\partial \tilde{P}}{\partial q_i}\right) + \sum_{j=1}^{m} \Delta'_{q_i}(D_j)\frac{\partial P}{\partial D_j} + \sum_{j=1}^{m}\sum_{i'=1}^{i-1} B_{j,i'}(q_1, \ldots, q_{n'}, D_1, \ldots, D_m)\Delta'_{q_{i'}}(D_j) \tag{37}$$

Chaque $B_{j,i}$ est un polynôme de ses $n' + m$ variables, à coefficients entiers, obtenu en sommant tous les termes faisant intervenir $\Delta'_{q_{i'}}(D_j)$.

Exemple 4.10. Reprenons l'équation

$$D(x, q) = 1 + xD(x, q)D(xq, q) \tag{38}$$

correspondant à l'énumération des mots de Dyck suivant la demi-longueur (x) et l'aire de Carlitz (q). La seule substitution de variables présente est $\sigma = \sigma_{x \leftarrow xq}$, et l'équation s'écrit

$$\tilde{P}\left(x, D, \sigma(D)\right) = 0,$$

avec $\tilde{P}(x, q, y, y') = 1 - y + xyy'$. En appliquant Δ'_q, l'équation (38) devient

$$0 = -\Delta'_q D + x\Delta'_q D.D + xD.(\Delta'_x D + \Delta'_q D),$$

si bien que nous avons, pour cette équation,

$$B_{2,1}(x, D) = x.D.$$

La série $\Delta'_q(D)$ peut alors être calculée de la même manière que pour l'aire au paragraphe 4.3.

Plus précisément, l'examen des équations (37) permet de montrer la proposition suivante :

Proposition 4.11. *Soit, pour $1 \leq j \leq m$, $D_j = D_j(q_1, \ldots, q_{n'})$ la série génératrice du langage D_j suivant les seuls paramètres de rang 1 (précédemment notée $\pi(D_j)$). Notons également $P_j = P_j(q_1, \ldots, q_{n'}, D_1, \ldots, D_m)$ le polynôme correspondant à la j-ème équation du système algébrique donné par la grammaire sur laquelle est basée la Q-grammaire G.*

Soit également

$$J = \det\left(\frac{\partial P_i}{\partial D_j}\right)_{1 \leq i \leq m, 1 \leq j \leq m} \tag{39}$$

le jacobien du système algébrique.

Alors, si $q = q_{i_0}$ est une variable formelle qui compte un paramètre de rang k, la série formelle $\Delta'_q(D_j)$ est de la forme

$$\Delta'_q(D_j) = \frac{N_{i_0,j}(q_1, \ldots, q_{n'}, D_1, \ldots, D_m)}{J^k} \tag{40}$$

où $N_{i_0,j}$ est un polynôme qui dépend de la Q-grammaire.

Preuve. Pour chaque entier ℓ, $1 \leq \ell \leq m$, la Q-grammaire nous donne une équation, caractérisée par un polynôme $\tilde{P}_\ell$.

Les m équations (37) forment un système d'équations affines portant sur les séries $\Delta_{q_i}(D_j)$, dont les coefficients sont des polynômes en les variables q_j (pour $1 \leq j \leq n'$), $\pi(D_j)$ (pour $1 \leq j \leq m$). Le second membre de ces équations affines regroupe alors tous

les termes faisant apparaître $\Delta'_{q_{i'}}(D_j)$ (pour $1 \leq j \leq m$, et $q_{i'}$ comptant un paramètre $\lambda_{i'}$ dont le rang est strictement inférieur à celui de λ_i).

Il est alors remarquable que le coefficient de $\Delta_{q_i}(D_j)$ dans chaque équation, $\partial P_\ell / \partial D_j$, ne dépende ni de i, ni du système de Q-équations ou des substitutions σ_k, mais seulement de j et du polynôme P_ℓ, qui est donné par la grammaire algébrique sous-jacente à la Q-grammaire G. En d'autres termes, lorsque i varie, seul le second membre du système affine change, mais pas son déterminant.

Lorsque $q = q_{i_0}$ compte un paramètre de rang 1 (c'est-à-dire, lorsque $i \leq n'$), les équations (37) s'écrivent :

$$\sum_{j=1}^{m} \frac{\partial P_i}{\partial D_j} \Delta'_q(D_j) = -q \frac{\partial P_i}{\partial q} \tag{41}$$

Le déterminant de ce système étant J, les formules de Cramer donnent directement la forme (40).

Procédons par récurrence sur le rang du paramètre compté par q : supposons (40) vraie pour tout paramètre de rang strictement inférieur à k, et soit $q = q_{i_0}$ une variable comptant un paramètre de rang k. Les m équations (37) forment encore un système dont le déterminant est J. Le second membre de ce système, fait intervenir des séries $\Delta_{q_l}(D_j)$, où q_l compte un paramètre de rang au plus $k-1$, et par conséquent, par hypothèse de récurrence, ce second membre peut se mettre sous la forme d'une fraction rationnelle des variables $q_1, \ldots, q_{n'}$ et $D_1, \ldots, D_m$, avec dénominateur J^{k-1}. Dès lors, il est clair que les formules de Cramer donnent (40). □

La proposition (4.11) a des conséquences importantes pour le calcul des singularités des séries génératrices et, partant, des valeurs moyennes de paramètres Q-comptables. Le comportement d'une série $\Delta'_q(D_j)$ au voisinage de ses singularités dominantes nous est en effet donné par la comparaison des termes dominants de J^k et de $N_{i_0,j}(D_1, \ldots, D_m)$.

Le numérateur dépend fortement du système de Q-équations considérées. Il est déterminé par les substitutions de variables employées dans le système, autant que par les polynômes $\tilde{P}_\ell$ eux-mêmes. En revanche, le dénominateur J^k ne dépend que très peu du paramètre Q-comptable considéré : J est entièrement déterminé par le système d'équations *algébriques* fournies par la grammaire sous-jacente à la Q-grammaire, et k est le rang du paramètre.

Par ailleurs, le jacobien J s'annule en chaque singularité des séries algébriques, et son développement asymptotique au voisinage de ces singularités (ou tout au moins le terme dominant) découle naturellement du développement de Puiseux cherché pour chaque

série au voisinage de la singularité dominante – pour peu que les contributions liées aux différents termes dominants ne se compensent pas, auquel cas il sera nécessaire d'obtenir un développement asymptotique plus fin des différentes séries génératrices.

Dans le cas le plus fréquent, le corollaire suivant donne le comportement asymptotique de paramètres Q-comptables :

Corollaire 4.12. *Supposons que la singularité dominante unique du système soit x_0, et que l'on ait pour J un équivalent de la forme $J \sim K(x_0 - x)^\alpha$.*

Alors, si le numérateur $N_{i,j}(D_1, \dots, D_m)$ a une valeur finie non nulle en x_0, la forme asymptotique des coefficients de $\Delta'_{q_i}(D_j)$ est :

$$[x^n]\Delta'_{q_i}(D_j) \sim K' x_0^{-n} n^{\alpha k - 1}. \tag{42}$$

Preuve. Sous les hypothèses indiquées, la série $\Delta'_{q_i}(D_j)$ a un équivalent de la forme $K.(x_0 - x)^{-k\alpha}$, et

$$[x^n]\,(x_0 - x)^{-\beta} \sim \frac{x_0^{-n} n^{\beta-1}}{\Gamma(\beta)}.$$

□

Dans le cas le plus fréquent, $\alpha = 1/2$, et les séries génératrices $D_j(x)$ ont également un développement asymptotique de la forme $D_j(x) = D_{j,0} + K_j(x_0 - x)^{1/2} + o(x_0 - x)^{1/2}$. Nous avons alors le corollaire suivant :

Corollaire 4.13. *Sous les hypothèses du corollaire précédent, et si de plus*

$$D_j(x) = D_{j,0} + K_j(x_0 - x)^{1/2} + o(x_0 - x)^{1/2},$$

alors la valeur moyenne A_n d'un paramètre de rang k, parmi les mots de taille n, vérifie

$$A_n \sim K'' n^{(k+1)/2}.$$

Ceci donne aux paramètres de rang 1 un comportement moyen linéaire, chaque rang au-dessus de 1 multipliant l'ordre de grandeur de la valeur moyenne par $\sqrt{n}$. Ce comportement moyen est à comparer à l'ordre de grandeur maximal, qui est donné par le rang minimal du paramètre : ainsi, il semblerait qu'un paramètre Q-comptable dont l'ordre de grandeur *maximal* est n^k pour les mots de longueur n, ait tendance à avoir un ordre de grandeur *moyen* de $n^{(k+1)/2}$.

4.5.2 Décomposition en paramètres élémentaires

Le calcul de statistiques sur les paramètres Q-comptables est compatible avec leur décomposition en combinaison linéaire de paramètres élémentaires. En effet, soit λ un paramètre, compté par une variable q, et qui s'écrit comme combinaison linéaire d'autres paramètres :

$$\lambda(w) = \alpha_1\lambda_1(w) + \cdots + \alpha_k\lambda_k(w), \quad \forall w \in D_1 \cup \ldots \cup D_m.$$

Si chaque paramètre λ_i est compté par la variable q_i, on a la même relation entre les séries :

$$\Delta'_q(D_j) = \alpha_1\Delta'_{q_1}(D_j) + \cdots + \alpha_k\Delta'_{q_k}(D_j) \tag{43}$$

Par conséquent, pour obtenir les séries $\Delta'_q(D_j)$, nous pouvons nous contenter de les calculer pour des variables qui comptent des paramètres élémentaires. Ainsi, on pourra limiter le calcul des numérateurs $N_{i_0,j}$ de la proposition 4.11 au cas où la variable q_{i_0} compte un paramètre élémentaire.

4.5.3 Série de moments suivant un paramètre élémentaire

Soit G une grammaire algébrique, et soit $p = p_{R_k^{(a_k)}R_{k-1}^{(a_{k-1})}\ldots R_1}$ un paramètre élémentaire[3]. Soit, pour $\ell = 1, \ldots, k$,

$$p_\ell = p_{R_\ell^{(a_\ell)}\ldots R_1}$$

(les paramètres p_ℓ sont tous les paramètres élémentaires qu'il est indispensable d'incorporer à une Q-grammaire pour pouvoir obtenir $p = p_k$).

Soient $x_1, \ldots, x_n$ des variables comptant des paramètres de rang 1, et $q_1, \ldots, q_k$ des variables supplémentaires, q_ℓ comptant le paramètre p_ℓ (techniquement, q_1 compte également un paramètre de rang 1).

La projection π est ici la projection de $\mathbb{Q}[[x_1, \ldots, x_n, q_1, \ldots, q_k]]$ dans $\mathbb{Q}[[x_1, \ldots, x_n]]$, et les opérateurs Δ' sont définis en conséquence.

Nous nous intéressons au calcul de $\Delta'_{q_k}(D_j)$, pour chaque série génératrice D_j d'un langage engendré par la grammaire G.

Afin de pouvoir exprimer facilement cette série, nous avons besoin de quelques notations :

– i_ℓ désigne l'indice du symbole gauche de la règle R_ℓ;

3. Pour des raisons de notations, l'ordre des indices est ici inversé par rapport à notre habitude.

- pour $\ell > 1$, j_ℓ désigne l'indice du a_ℓ-ème symbole droit de la règle R_ℓ (j_1 n'est pas défini);
- pour chaque règle $R \in \mathcal{R}$, T_R désigne le terme (monôme) introduit dans l'une des équations de la grammaire G, par la règle R (il s'agit du produit commutatif des symboles et lettres présents au second membre de R);
- pour chaque règle pointée $R^{(i)} \in \mathcal{R}_p$, si D est le i-ème symbole du membre droit de R, $T'_{R,i} = T_R/D$;
- M est la matrice jacobienne $(\partial P_i/\partial D_j)$ du système d'équations de la grammaire;
- $M_{i,j}$ est la matrice M, privée de sa i-ème ligne et de sa j-ème colonne;
- $J = \det M$, $J_{i,j} = (-1)^{i+j} \det M_{i,j}$.

Notons que J et les mineurs $J_{i,j}$ ne dépendent pas de la Q-grammaire, mais seulement de la grammaire algébrique sous-jacente, de même que les monômes T_R et $T'_{R,i}$. Toutes ces expressions s'expriment comme polynômes en les variables $x_1, \ldots, x_n$ et en les séries génératrices algébriques $D_1, \ldots, D_m$.

Théorème 4.14. *La série $\Delta'_{q_k}(D_j)$ est donnée par:*

$$\Delta'_{q_k}(D_j) = \frac{(-1)^k}{J^k} T_{R_1} T'_{R_2,a_2} \ldots T'_{R_k,a_k} J_{i_1,j_2} J_{i_2,j_3} \ldots J_{i_{k-1},j_k} J_{i_k,j}. \tag{44}$$

Preuve. La preuve est par récurrence sur k.

Lorsque $k = 1$, la variable q_1 n'apparaît dans le système d'équations de la Q-grammaire que dans le monôme T_{R_1}, avec degré 1; par conséquent, le système d'équations (37) s'écrit :

$$\begin{cases} 0 &= \sum_{j=1}^m \frac{\partial P_i}{\partial D_j} \Delta'_{q_1}(D_j) \qquad (i \neq i_1); \\ -T_{R_1} &= \sum_{j=1}^m \frac{\partial P_{i_1}}{\partial D_j} \Delta'_{q_1}(D_j). \end{cases}$$

La résolution de ce système donne immédiatement

$$\Delta'_{q_1}(D_j) = \frac{-T_{R_1} J_{i_1,j}}{J}.$$

Supposons maintenant la formule vraie pour k, et montrons qu'elle est également vraie pour $k+1$. La variable q_{k+1} n'apparaît dans le système d'équations de la Q-grammaire que grâce à la substitution de variables $\sigma_{q_k \leftarrow q_k q_{k+1}}$, et ce, dans un seul terme (qui donne $T_{R_{k+1}}$

dans le système algébrique sous-jacent). Comme précédemment, le système (37) s'écrit

$$\begin{aligned} 0 &= \sum_{j=1}^{m} \frac{\partial P_i}{\partial D_j} \Delta'_{q_{k+1}}(D_j) \qquad (i \neq i_{k+1}) \\ -T'_{R_{k+1},a_{k+1}} \Delta'_{q_k}(D_{j_{k+1}}) &= \sum_{j=1}^{m} \frac{\partial P_{i_{k+1}}}{\partial D_j} \Delta'_{q_{k+1}}(D_j). \end{aligned}$$

En résolvant, nous obtenons en utilisant l'hypothèse de récurrence

$$\begin{aligned} \Delta'_{q_{k+1}}(D_j) &= \frac{-T'_{R_{k+1},a_{k+1}} \Delta'_{q_k}(D_j)}{J} J_{i_{k+1},j} \\ &= \frac{(-1)^{k+1}}{J^{k+1}} T_{R_1} . \left(\prod_{l=2}^{k+1} T'_{R_l,a_l} J_{i_{l-1},j_l} \right) . J_{i_{k+1},j} \end{aligned}$$

qui termine la récurrence. □

Remarque sur la forme de l'équation (44) **:** La forme donnée précédemment pour l'équation (44) montre que toutes les séries de moments suivant des paramètres Q-comptables s'expriment au moyen de m^2 mineurs et des termes T_R et $T'_{R,a}$. En revanche, elle n'est pas très "parlante" lorsqu'il s'agit de l'appliquer à une série et un paramètre donnés. Nous pouvons la réécrire en utilisant comme indices les symboles non terminaux :

- si la i-ème ligne de la matrice M correspond à l'équation définissant la série U, et si la j-ème colonne correspond à la différentiation suivant la série V, le mineur $J_{i,j}$ peut être noté $J_{U,V}$;
- si U est le i-ème symbole du membre droit de la règle R, le terme $T'_{R,i}$ peut être noté $T'_{R/U}$.

Avec ces notations, et en prenant comme paramètre élémentaire[4] $p = p_{R_1^{a_1} \ldots R_{k-1}^{a_{k-1}} R_k}$, notons V_i le symbole du membre gauche de la règle R_i (pour $1 \leq i \leq k$), et U_i le a_i-ème symbole du membre droit de R_i (pour $1 \leq i \leq k-1$). Le théorème (4.14) s'exprime alors sous la forme suivante :

$$\Delta'_q(U_0) = (-1)^k \left(\prod_{i=1}^{k-1} \frac{T'_{R_i/U_i} J_{V_i,U_{i-1}}}{J} \right) \frac{T_{R_k} J_{V_k,U_{k-1}}}{J}.$$

Le théorème (4.14) répond de manière raisonnablement satisfaisante au problème de l'énumération suivant la somme des valeurs d'un paramètre Q-comptable : connaissant la

4. L'ordre des indices est ici conforme à notre notation usuelle.

décomposition de ce paramètre en combinaison linéaire de paramètres élémentaires, il est relativement simple d'exprimer la série génératrice comme fraction rationnelle des séries génératrices initiales, la fraction rationnelle faisant intervenir le jacobien J et des mineurs de la matrice jacobienne du système.

Une autre conséquence intéressante de ce théorème se situe au niveau des asymptotiques. Chaque paramètre élémentaire ne prenant que des valeurs positives, et les paramètres Q-comptables étant formés par combinaisons linéaires à coefficients positifs de ces paramètres élémentaires, il n'y a pas à craindre que des contributions provenant de paramètres élémentaires se compensent. Dès lors, pour estimer la valeur moyenne d'un paramètre Q-comptable, il nous suffit de faire le même travail pour chaque paramètre élémentaire apparaissant dans sa décomposition.

Dans la pratique, il devient possible de calculer le développement asymptotique de n'importe quelle série $\Delta'_{q_i}(D_j)$, quel que soit le rang du paramètre compté par la variable q_i, une fois calculés ceux des m séries algébriques D_i, celui du jacobien J, et ceux des m^2 mineurs $J_{i,j}$.

Exemple 4.15 (somme des hauteurs de pics des chemins de Dyck). Nous reprenons la grammaire G_2 de l'exemple (2.3). Dans cette grammaire, considérons les trois paramètres Q-comptables p_1, p_2 et p_3, comptant respectivement la demi-longueur des chemins, leur nombre de pics, et la somme des hauteurs de pics. Leurs décompositions en paramètres élémentaires sont :

Paramètre	Nom
p_1 (demi-longueur)	$R_3 + R_4 + R_5 + R_6$
p_2 (nombre de pics)	$R_3 + R_4$
p_3 (somme des hauteurs de pics)	$(\epsilon + R_5^{(1)} + R_6^{(1)})(R_3 + R_4)$

Le paramètre p_3 est donc une somme de 6 paramètres élémentaires, dont 2 sont de rang 1 et 4 de rang 2. Grâce au théorème (4.14), lorsque la variable q compte l'un des ces paramètres élémentaires, la série $\Delta'_q(D)$ peut être obtenue sans écrire une seule Q-équation.

En effet, du système algébrique

$$\begin{cases} D &= 1 + E \\ E &= x + 2xE + xE^2 \end{cases}$$

nous tirons la matrice jacobienne M :

$$M = \begin{pmatrix} -1 & 1 \\ 0 & -1 + 2x + 2xE \end{pmatrix}.$$

Nous avons donc $J = 1 - 2x(1 + E)$; nous retrouvons le même jacobien que lors de l'énumération suivant l'aire. Les mineurs, quant à eux, sont (en indiçant lignes et colonnes par les symboles D et E plutôt que par 1 et 2) :

$$\begin{aligned} J_{D,D} &= -J, & J_{D,E} &= 0, \\ J_{E,D} &= -1, & J_{E,E} &= -1. \end{aligned}$$

Tous les paramètres élémentaires apparaissant dans la décomposition des paramètres qui nous intéressent, n'utilisent que des E-dérivations, et donc seuls les mineurs $J_{E,D}$ et $J_{E,E}$ (tous deux égaux à -1) apparaîtront dans les calculs.

Commençons par appliquer le théorème au calcul de $\Delta'_x(D)$. La variable x compte la demi-longueur, soit $p_{R_3+R_4+R_5+R_6}$; nous avons donc

$$\begin{aligned} \Delta'_x(D) &= -\frac{(T_{R_3} + T_{R_4} + T_{R_5} + T_{R_6})\, J_{E,D}}{J} \\ &= \frac{x + 2xE + xE^2}{J} = \frac{E}{J}. \end{aligned}$$

De même, si la variable q compte le nombre de pics, le théorème donne pour $\Delta'_q(D)$:

$$\begin{aligned} \Delta'_q(D) &= -\frac{T_{R_3} J_{E,D}}{J} - \frac{T_{R_4} J_{E,D}}{J} \\ &= \frac{x(E+1)}{J}. \end{aligned}$$

Si, maintenant, la variable r compte la somme des hauteurs de pics, il faut prendre en compte les contributions dues aux paramètres élémentaires $p_{R_5^{(1)} R_3}$, $p_{R_5^{(1)} R_4}$, $p_{R_6^{(1)} R_3}$ et $p_{R_6^{(1)} R_4}$. Pour les deux règles R_5 et R_6, le premier symbole du membre droit est toujours E, donc, pour chacun de ces 4 paramètres élémentaires, $j_2 = E$ (l'indice j_ℓ indique, pour chaque lettre $R_\ell^{(d_\ell)}$ du nom de paramètre, quel est le d_ℓ-ème symbole du membre droit de R_ℓ). Le théorème donne donc :

$$\begin{aligned} \Delta'_r(D) &= \Delta'_q(D) + \frac{\left(T'_{R_5,1}.J_{E,E} + T'_{R_6,1}.J_{E,E}\right)\left(T_{R_3} J_{E,D} + T_{R_4}.J_{E,D}\right)}{J^2} \\ &= \frac{x(E+1)}{J} + \frac{x^2(E+1)^2}{J^2}. \end{aligned}$$

En récrivant les résultats comme fractions rationnelles en J (on a $x(E+1) = (1-J)/2$ et $E = (1-J)/(1+J)$), nous obtenons les expressions suivantes :

$$\Delta'_x(D) = \frac{1-J}{J(1+J)}, \tag{45}$$

$$\Delta'_q(D) = \frac{1-J}{2J}, \tag{46}$$

$$\Delta'_r(D) = \frac{1-J^2}{4J^2}. \tag{47}$$

L'expression (45) est bien celle qui a été trouvée au paragraphe 4.3 : il s'agit de la même série.

Les coefficients des séries $\Delta'_q(D)$ et $\Delta'_r(D)$ peuvent être calculés explicitement. Pour $\Delta'_q(D)$, nous avons

$$\begin{aligned}\Delta'_q(D) &= \frac{1-J}{1+J}\frac{1+J}{2J} \\ &= E\left(\frac{1}{2}+\frac{1}{2J}\right) \\ &= \frac{1}{2}\left(-1+D+\Delta'_x(D)\right).\end{aligned}$$

Par conséquent, puisque $D=\sum_{n\geq 0} C_n x^n$ et $\Delta'_x(D)=\sum_{n\geq 0} nC_n x^n$, nous avons

$$\Delta'_q(D)=\sum_{n\geq 0}\frac{n+1}{2}C_n x^n,$$

et le nombre moyen de pics des chemins de Dyck de longueur $2n$ est exactement $(n+1)/2$.

En reprenant l'expression $J=\sqrt{1-4x}$ calculée au paragraphe 4.3, on trouve pour $\Delta'_r(D)$ l'expression extrêmement simple

$$\Delta'_r(D)=\frac{x}{1-4x}. \tag{48}$$

On retrouve aisément un résultat bien connu : la somme des hauteurs de pics des chemins de Dyck de longueur $2n$ est 4^{n-1}.

Enfin, l'expression (47) peut être comparée à (21), qui correspond à l'énumération suivant l'aire. Le rapport entre les deux séries est de x, ce qui, pour une singularité en $x=1/4$, donne un rapport proche de 4 entre les moyennes des aires et de la somme des hauteurs de pics.

L'utilité du théorème (4.14) comme moyen pratique de calcul peut être mesurée en comparant les calculs effectués ci-dessus, à ceux effectués section 4.3. Il est à noter, toutefois, que nous n'avons pas de ce théorème une version portant sur les séries non projetées. Par conséquent, il ne nous permet pas de calculer les séries de moments d'ordre 2 ou plus.

Chapitre 5

Application à l'énumération de polyominos

Dans ce chapitre, nous mettons en œuvre la théorie des Q-grammaires sur différentes familles de polyominos codés par des langages algébriques. Dans la pratique, nous nous concentrons sur différentes classes de polyominos *verticalement convexes*, dont la frontière forme un chemin qu'il est relativement aisé de coder par les mots d'un langage algébrique.

Pour chaque famille de polyominos, nous donnons un codage par les mots d'un langage algébrique, et nous indiquons, parmi les paramètres classiques d'étude, lesquels sont Q-comptables dans la grammaire donnée.

5.1 Paramètres étudiés

Soit P un polyomino. Nous notons ∂P sa frontière, qui est formée d'un certain nombre de segments reliant des points adjacents du plan $\mathbb{Z}^2$; ces segments séparent chacun deux cellules dont l'une appartient à P et l'autre non.

Nous pouvons définir un certain nombre de paramètres :

- Le *périmètre* de P est la longueur de ∂P. Ce périmètre peut être décomposé en la somme du *périmètre vertical* (le nombre de segments verticaux qui forment ∂P) et du *périmètre horizontal* (le nombre de segments horizontaux qui forment ∂P). Chacun de ces périmètres étant pair, nous notons $pe(P)$ (respectivement, $ph(P)$, $pv(P)$) le demi-périmètre (respectivement, le demi-périmètre horizontal, le demi-périmètre vertical) de P.

- L'*aire* de P est le nombre de cellules qui le composent; nous la notons $a(P)$.

- La *largeur* (respectivement *hauteur*) de P est la largeur (respectivement hauteur) du plus petit rectangle contenant P. Ces deux paramètres sont respectivement notés $\ell(P)$ et $h(P)$. Notons que les polyominos verticalement convexes sont caractérisés par $\ell(P) = ph(P)$, et les polyominos convexes, par $h(P) = pv(P)$ et $\ell(P) = ph(P)$.
- Le *périmètre de sites* est le nombre total de cellules n'appartenant pas à P, mais qui sont adjacentes (le long d'une arête) à au moins une cellule de P. Ce périmètre de sites, noté $ps(P)$, est toujours inférieur ou égal au périmètre $2pe(P)$.
- Le *nombre d'angles rentrants* de P, est le nombre de sommets de ∂P pour lesquels, parmi les 4 cellules adjacentes qui partagent ce sommet, trois appartiennent à P et une ne lui appartient pas. Il est possible de distinguer, parmi les angles rentrants, des angles *Nord-Est, Sud-Est, Sud-Ouest, et Nord-Ouest*, suivant la position de la cellule extérieure par rapport aux trois autres. Pour les polyominos convexes, le nombre d'angles rentrants est égal à la différence entre le périmètre et le périmètre de sites.

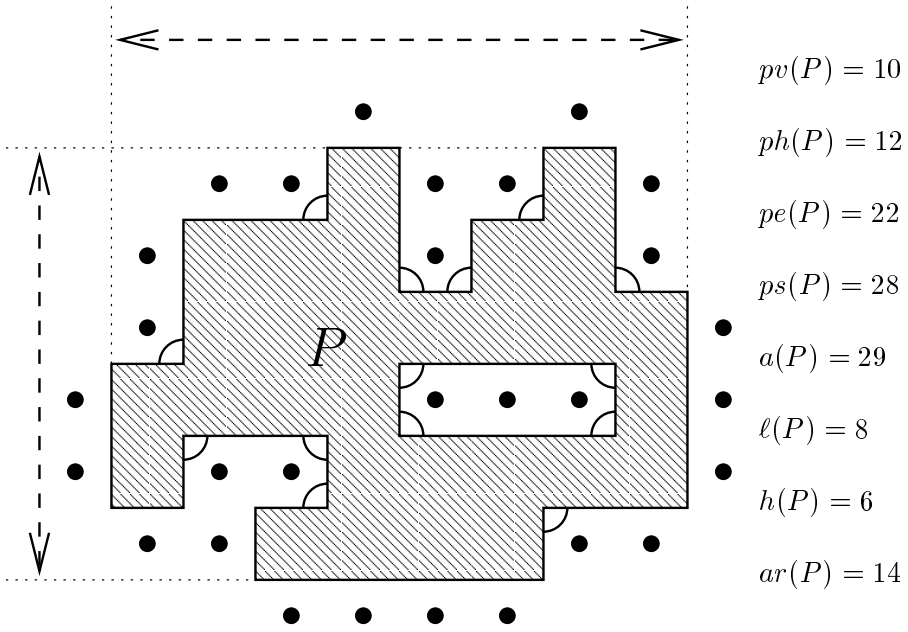


FIG. 5.1: *Un exemple de polyomino*

Pour chaque classe de polyominos, nous pouvons également définir un autre paramètre, le *périmètre de croissance* $pc(P)$, qui est le nombre de cellules c n'appartenant pas à P, mais telles que $P \cup \{c\}$ soit toujours un polyomino de la même classe. Pour les polyominos généraux, dont l'étude sort du cadre de ce travail, ce périmètre de croissance coïncide avec le périmètre de site, mais, en règle générale, $pc(P) \leq ps(P)$.

La figure 5.1 montre un exemple de polyomino général, et les valeurs de ces différents paramètres.

5.2 Polyominos parallélogrammes

Les polyominos parallélogrammes étant convexes, leur demi-périmètre vertical est égal à leur hauteur, et leur demi-périmètre horizontal, à leur largeur.

5.2.1 Codage

Le codage des polyominos parallélogrammes par des mots de Dyck est classique, et n'est rappelé ici que pour mémoire.

Définition 5.1. Soit P un polyomino parallélogramme. Nous notons k son nombre de colonnes, a_i $(1 \leq i \leq k)$ la hauteur de sa i-ème colonne, et b_i $(1 \leq i \leq n-1)$ le nombre de cellules suivant lesquelles les i-ème et $(i+1)$-ème colonnes sont accolées.

Le mot de Dyck codant P est le mot de Dyck $w = \psi_1(P)$, ayant k pics et $k-1$ creux, le i-ème pic étant de hauteur a_i et le i-ème creux, de hauteur $b_i - 1$.

Ce codage vérifie les propriétés suivantes :

Proposition 5.2. *Le codage ψ_1 établit une bijection entre l'ensemble $\mathcal{P}$ des polyominos parallélogrammes et le langage D des mots de Dyck; de plus,*

- *le nombre de colonnes de P est le nombre de pics de $\psi_1(P)$;*
- *si $p(P)$ est le périmètre de P, et si P n'est pas le polyomino vide, $p(P) = |\psi_1(P)| + 2$;*
- *si $\mathcal{A}(P)$ est l'aire de P et $S(w)$ la somme des hauteurs des pics de w, $\mathcal{A}(P) = S(\psi_1(P))$.*

Dans ce qui suit, nous acceptons comme polyomino parallélogramme, le polyomino vide, codé par le mot vide ϵ. L'exclure reviendrait à coder les polyominos parallélogrammes par les mots de Dyck non vides; dans les deux cas, la série génératrice suivant le demi-périmètre n'est pas exactement celle des mots de Dyck suivant la longueur, en raison du décalage de 1 entre longueur des mots non vides et demi-périmètre.

5.2.2 Grammaire

Pour que la grammaire puisse fournir d'autres types de paramètres que le simple périmètre, il faut que le nombre de pics des mots de Dyck soit Q-comptable. Nous avons vu dans les chapitres précédents qu'une grammaire convenable est la grammaire G_2, d'axiome D :

$$\begin{cases} R_1 : & D & \to & \epsilon \\ R_2 : & D & \to & E \\ R_3 : & E & \to & ab \\ R_4 : & E & \to & abE \\ R_5 : & E & \to & aEb \\ R_6 : & E & \to & aEbE \end{cases}$$

5.2.3 Paramètres Q-comptables

Dans la grammaire G_2, les paramètres suivants sont facilement décomposables en paramètres élémentaires :

- Hauteur : $h = p_{R_2} + p_{R_5} + p_{R_6}$.
- Largeur : $\ell = p_{R_3} + p_{R_4}$.
- Aire : $\mathcal{A} = p_{R_5^{(1)} R_3} + p_{R_5^{(1)} R_4} + p_{R_6^{(1)} R_3} + p_{R_6^{(1)} R_4}$.

Il est facile de voir que les paramètres "nombre d'angles rentrants" et "nombre d'angles Nord-Ouest rentrants" ne sont pas Q-comptables dans la grammaire G_2. En effet, étant tous deux inférieurs au périmètre, ils devraient être de rang 1 (dans la grammaire G_2, tous les paramètres Q-comptables non identiquement nuls et dont le nom ne commence pas par $R_2^{(1)}$ ont un rang minimal égal à leur rang formel, et si leur nom commence par $R_2^{(1)}$, cette lettre peut être retirée du nom sans changer la valeur du paramètre). Or, on vérifie immédiatement, en examinant les deux polyominos de la figure 5.2, qu'aucun paramètre de rang 1 ne convient : les arbres de dérivation correspondants utilisent les mêmes règles, mais les polyominos n'ont pas le même nombre d'angles Nord-Ouest rentrants. Le même argument prouve que le périmètre de sites n'est pas non plus Q-comptable, puisque, pour chaque polyomino parallélogramme P, $ps(P) = 2pe(P) - ar(P)$.

En revanche, le nombre d'angles rentrants Sud-Est est p_{R_6}. Les angles rentrants correspondent, sur le mot de Dyck, aux creux qui sont précédés immédiatement par au moins deux occurrences de b; ces creux apparaissent lors de l'utilisation de la règle R_6, et jamais autrement.

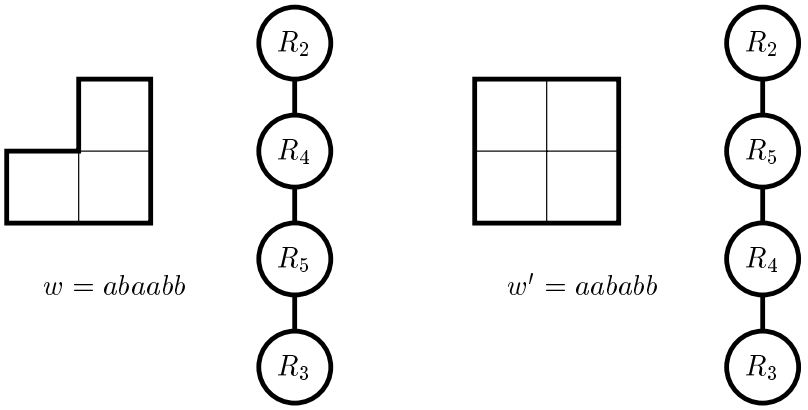


FIG. 5.2: *Deux polyominos parallélogrammes et leurs arbres de codage*

Compte tenu de la simplicité de la grammaire, nous pouvons calculer aisément la série génératrice suivant la hauteur (comptée par x), la largeur (comptée par p), et le nombre d'angles rentrants Sud-Est (comptés par r) :

$$\begin{cases} D(x,p,r) &= 1 + xE(x,p,r) \\ E(x,p,r) &= p + pE(x,p,r) + xE(x,p,r) + xrE(x,p,r)^2 \end{cases}$$

qui donne, après résolution d'une équation du second degré,

$$D(x,p,r) = \frac{1-p-x+2r-\sqrt{(1-p-x)^2-4xpr}}{2r}.$$

Dans cette situation, le jacobien (en ne gardant que les variables x et r) vaut $J = 1-p-x-2xE = 3-p-x-2D$.

En prenant comme *taille* le couple formé de la hauteur et de la largeur (puisque aucun de ces deux paramètres ne constitue à lui seul une taille), le calcul de la série de moments est également très simple : l'application du théorème 4.14 donne

$$\begin{aligned} \Delta'_q(D) &= -\tfrac{J_{E,D}T_{R_6}}{J} \\ &= \tfrac{x^2E^2}{1-p-x-2xE} \\ &= \tfrac{(D-1)^2}{3-p-x-2D}. \end{aligned}$$

Le nombre d'angles rentrants Sud-Est a la même distribution (par rapport à la hauteur et à la largeur) que le nombre d'angles rentrants Nord-Ouest : une rotation d'angle π échange ces deux paramètres sans modifier les différents périmètres ni l'aire. Par conséquent, la série génératrice et la série de moments correspondant à ce paramètre sont les mêmes que pour le nombre d'angles rentrants Sud-Est. Si la variable s compte le nombre

total d'angles rentrants (qui n'est pas non plus Q-comptable), la série de moments $\Delta'_s(D)$ est le double de $\Delta'_r(D)$:

$$\Delta'_s(D) = 2\Delta'_r(D) = \frac{2(D-1)^2}{3-p-x-2D}.$$

Toutefois, le nombre d'angles rentrants n'étant pas Q-comptable, nous ne pouvons donner aussi simplement la série génératrice bivariée suivant le périmètre et le nombre d'angles rentrants.

De même que le nombre d'angles rentrants, la hauteur de la première ou de la dernière colonne du polyomino ne forment pas des paramètres Q-comptables. L'examen des mêmes polyominos permet de s'en convaincre rapidement, tout comme le simple fait que, lors de l'application de la règle R_4, la hauteur de la première colonne peut décroître : la première colonne du polyomino codé par $w = abw'$ est de hauteur 1, alors même que le mot w' peut coder un polyomino dont la première colonne est plus haute.

Toutefois, nous pouvons utiliser le lemme de marquage supérieur sur la règle pointée $R_6^{(1)}$ afin d'obtenir la grammaire G', plus fine que G_2 :

$$\begin{cases} R_1 : & D & \to & \epsilon & R_2 : & D & \to & E \\ R_3 : & E & \to & ab & R'_3 : & F & \to & ab \\ R_4 : & E & \to & abE & R'_4 : & F & \to & abF \\ R_5 : & E & \to & aEb & R'_5 : & F & \to & aFb \\ R_6 : & E & \to & aFbE & R'_6 : & F & \to & aFbF \end{cases}$$

Dans cette nouvelle grammaire, le paramètre "hauteur du dernier pic" est Q-comptable, et vaut $p_{R_3} + p_{R_5}$. Si la variable t compte ce paramètre, et toujours en utilisant comme taille le couple formé de la largeur et de la hauteur, le théorème 4.14 donne, après calculs,

$$\Delta'_t(D)(x,p) = \frac{xE(x,p)}{1-x\,(1+E(x,p))}.$$

Pour obtenir la hauteur du premier pic comme paramètre Q-comptable, il faut appliquer le lemme de marquage supérieur aux règles $R_4^{(1)}$ et $R_6^{(2)}$; on obtient alors une autre expression pour la série de moments (qui est la même série, puisque le premier pic devient, par image miroir, le dernier ; cela correspond, sur les polyominos parallélogrammes, à effectuer une symétrie centrale) :

$$\Delta'_t(D)(x,p) = \frac{xp + x^2E(x,p)}{1-p-x\,(1+F(x,p))}.$$

Dans les deux cas, en fixant $p = x$ (énumération suivant le périmètre seul), on obtient en développant en série entière :

$$\Delta'_t(D)(x, x) = \sum_{n \geq 1} (C_n - C_{n-1})\, x^n.$$

Ainsi, la hauteur moyenne de la première colonne des polyominos parallélogrammes de périmètre $2n$, est exactement $C_n/C_{n-1} - 1$, qui tend vers 3 lorsque n tend vers $+\infty$.

5.3 Polyominos verticalement convexes

Dans [25], Delest décrit un codage des polyominos verticalement convexes par des mots de Dyck colorés formant un langage algébrique, et donne une grammaire engendrant leur langage, ainsi qu'une expression pour la série génératrice. Ce codage est inspiré de celui des polyominos parallélogrammes, et la série génératrice obtenue énumère les polyominos verticalement convexes suivant le paramètre *périmètre*.

Dans [14], Bousquet-Mélou obtient, par une méthode ne faisant pas appel au codage par des mots, une expression pour la série génératrice de ces polyominos suivant les paramètres *largeur*, *périmètre vertical*, *aire*, et *hauteurs des première et dernière colonnes*. Les calculs font intervenir des q-équations d'une forme proche de celles des équations fournies par les Q-grammaires, mais où interviennent directement des projections de séries.

Plus récemment, Feretić [38, 39] a donné, pour la série génératrice des polyominos verticalement convexes suivant la largeur et le périmètre vertical, une expression nettement plus simple que celle donnée dans [25], et dont la preuve ne fait intervenir que des équations du second degré. La méthode passe par l'énumération d'objets formés à partir des polyominos verticalement convexes, et par une bijection avec les *polyominos murs*, qui codent les *compositions* d'un entier – nous les étudierons au paragraphe 5.4.

5.3.1 Codage

Nous reprenons, en le modifiant très légèrement, le codage des polyominos verticalement convexes par des mots de Dyck colorés tel qu'il est décrit dans [25].

Nous codons la frontière des polyominos verticalement convexes par des mots utilisant les lettres a, b, a', b' et p. La première condition que doivent remplir nos mots est qu'en leur appliquant le morphisme φ défini par

$$\begin{cases} \varphi(a) = \varphi(a') = a \\ \varphi(b) = \varphi(b') = b \\ \varphi(p) = ab \end{cases}$$

on doit obtenir des mots de Dyck non vides. De plus, ces mots ne doivent contenir aucun facteur ab, $a'b$, ab' ou $a'b'$: tous les pics du mot de Dyck $\varphi(w)$ doivent provenir des occurrences de la lettre[1] p.

Enfin, les mots de L doivent être de la forme forme

$$w = w_0 p w_1 p \dots w_{k-1} p w_k$$

où les mots w_i vérifient les conditions suivantes :

- $w_0 \in a^*$;
- $w_k \in b^*$;
- pour $1 \leq i \leq k-1$, $w_i \in b^*a^* \cup b'^*a'^* \cup b^*b'^* \cup a'^*a^*$.

Le langage L peut également être décrit comme le langage des mots de Dyck, où tous les pics sont remplacés par la lettre p, et où les facteurs $b^i a^j$ (les "vallées") qui séparent deux pics consécutifs sont colorés de différentes manières suivant les lettres qu'ils contiennent :

- si $i > 0$ et $j > 0$ (la vallée contient au moins un pas descendant et au moins un pas montant), toutes les lettres sont de la même "couleur" : $b^i a^j$ ou $b'^i a'^j$;
- si $i = 0$ (la vallée ne contient en fait que des pas montants), les lettres a' précèdent les lettres a : $a'^{j_1} a^{i_2}$;
- si $j = 0$ (la vallée ne contient que des pas descendants), les lettres b' sont placées après les lettres b : $b^{i_1} b'^{i_2}$.

Par analogie avec les mots et chemins de Dyck, nous appelons *pic* d'un mot de L, chaque occurrence de la lettre p dans ce mot. Si $w = w_1 p w_2$, la *hauteur* de ce pic est $|w_1|_{a,a'} - |w_1|_{b,b'} + 1$.

Soit P un polyomino verticalement convexe. Nous appelons *coin Sud-Ouest* de P, le coin inférieur gauche A de la plus basse cellule de la première colonne de P, et *coin Nord-Est* de P, le coin supérieur droit C de la plus haute cellule de la dernière colonne de P. Soit B le point situé immédiatement au-dessus de A, et D le point situé immédiatement au-dessous de C (voir figure 5.3). Le polyomino P est parfaitement décrit par un chemin allant de A à D (chemin inférieur) et un chemin allant de B à C (chemin supérieur). Ces chemins sont semi-dirigés (ils ne font que des pas Nord, Sud et Est), et ne se rencontrent pas; le chemin

1. Dans [25], le codage est fait avec 4 lettres x, y, $\overline{x}$, $\overline{y}$, les pics étant laissés sous forme $x\overline{x}$; il semble toutefois plus naturel de les distinguer comme nous le faisons ici.

AD commence forcément par un pas Est, et le chemin BC se termine forcément par un pas Est.

Le codage de P par un mot de L se fait de la manière suivante : chaque colonne est codée par un pic (lettre p) de hauteur égale à celle de la colonne. Entre deux pics successifs, on code les pas des chemins inférieur et supérieur qui se situent à la jonction entre les deux colonnes codées, dans l'ordre suivant :

- les éventuels pas Nord du chemin inférieur, par autant de b;
- les éventuels pas Sud du chemin supérieur, par autant de b';
- les éventuels pas Sud du chemin inférieur, par autant de a';
- les éventuels pas Nord du chemin supérieur, par autant de a.

Ainsi, les lettres a' et b' codent des pas qui ne peuvent apparaître dans un polyomino parallélogramme : si P est un polyomino parallélogramme, le mot $\psi_2(P)$ ne diffère du mot $\psi_1(P)$ que par le fait que les facteurs ab ont été remplacés par p. En ce sens, le codage ψ_2 peut être considéré comme une extension aux polyominos verticalement convexes du codage ψ_1 des polyominos parallélogrammes. Comme pour le codage des polyominos pa-

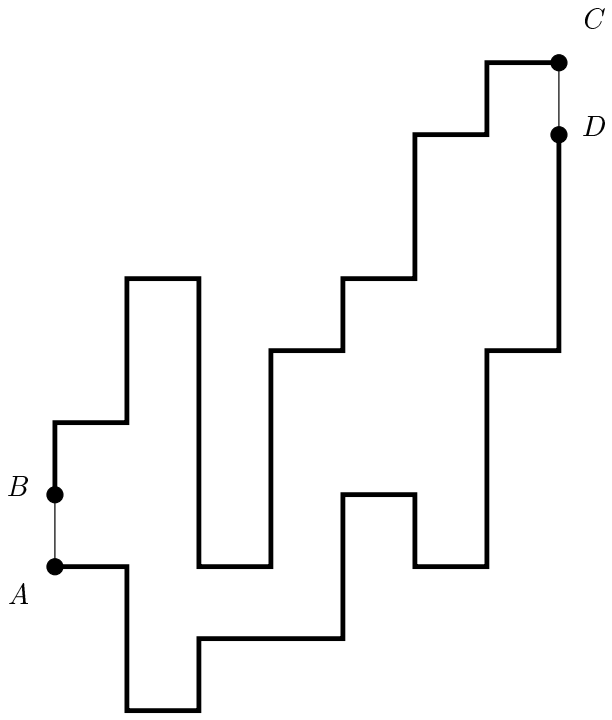


FIG. 5.3: *Le polyomino codé par* $w = apaaa'a'pb'b'b'b'bpaaapbbapa'aapbbbapbbb$

rallélogrammes, remplacer le mot w par son image miroir (lue de droite à gauche, en remplaçant a par b, b par a, a' par b', et b' par a') donne un mot w' qui code un nouveau polyomino verticalement convexe; ces deux polyominos verticalement convexes sont images l'un de l'autre par une symétrie centrale.

D'après la définition que nous avons donnée de $\psi_2(P)$, il est clair qu'un polyomino de largeur k et de demi-périmètre vertical n, est codé par un mot $\psi_2(P)$ vérifiant $|\psi_2(P)|_p = k$ et $|\psi_2(P)|_{a,a'} = |\psi_2(P)|_{b,b'} = n-1$. Enfin, l'*aire* de P devient, dans $\psi_2(P)$, la *somme des hauteurs des pics*.

5.3.2 Grammaire

Une grammaire algébrique non ambiguë qui engendre L, est donnée, de manière incomplète, dans [25]; une fois écrite *in extenso*, nous obtenons les équations (non commutatives) :

$$\begin{cases}
OO &= p + pGO + aOOb(\epsilon + OO) + aODb'p(\epsilon + GO) + a(\epsilon + OD)pb'\overline{G}O \\
&\quad + aO\overline{D}b'\overline{G}O \\
OD &= p + pGD + aOOb(\epsilon + OD) + aODb' + aODb'p(\epsilon + GD) \\
&\quad + a(\epsilon + OD)pb'\overline{G}D + aO\overline{D}b'\overline{G}D \\
O\overline{D} &= pG\overline{D} + aOObO\overline{D} + a(\epsilon + OD)pb' + aO\overline{D}b' + aODb'pG\overline{D} \\
&\quad + a(\epsilon + OD)pb'\overline{GD} + aO\overline{D}b'\overline{GD} \\
\overline{G}O &= a'pb(\epsilon + OO) + a'pGOb(\epsilon + OO) + a'\overline{G}Ob(\epsilon + OO) + a'\overline{G}Db'pGO \\
&\quad + a'p(\epsilon + GD)b'pGO + a'(p + pp + pG\overline{D} + \overline{G}Dp + pGDp + \overline{GD})b'\overline{G}O \\
\overline{G}D &= a'p(\epsilon + GO)b(\epsilon + OD) + a'\overline{G}Ob(\epsilon + OD) + a'p(\epsilon + GD)b' + a'\overline{G}Db' \\
&\quad + a'p(\epsilon + GD)b'p(\epsilon + GD) + a'\overline{G}Db'p(\epsilon + GD) \\
&\quad + a'(p + pp + pG\overline{D} + \overline{G}Dp + pGDp + \overline{GD})b'\overline{G}D \\
\overline{GD} &= a'p(\epsilon + GO)bO\overline{D} + a'\overline{G}ObO\overline{D} \\
&\quad + a'(p + pp + pG\overline{D} + \overline{G}Dp + pGDp + \overline{GD})b'(\epsilon + \overline{GD}) \\
&\quad + a'p(\epsilon + GD)b'pG\overline{D} + a'\overline{G}Db'pG\overline{D} \\
GO &= OO + a'GOb(\epsilon + OO) + a'GDb'p(\epsilon + GO) + a'(\epsilon + GD)pb'\overline{G}O \\
&\quad + a'G\overline{D}b'\overline{G}O \\
GD &= OD + a'GOb(\epsilon + OD) + a'GDb' + a'GDb'p(\epsilon + GD) + a'(\epsilon + GD)pb'\overline{G}D \\
&\quad + a'G\overline{D}b'\overline{G}D \\
G\overline{D} &= O\overline{D} + a'GObO\overline{D} + a'(\epsilon + GD)pb' + a'G\overline{D}b' + a'GDb'pG\overline{D} \\
&\quad + a'(\epsilon + GD)pb'\overline{GD} + a'G\overline{D}b'\overline{GD}
\end{cases}$$

L'axiome de la grammaire est le symbole OO; dans chaque symbole, la première lettre (O, G ou $\overline{G}$) indique quelles lettres précèdent la première occurrence de p, et la deuxième

lettre (O, D ou $\overline{D}$) indique ce qui suit la dernière occurrence de p:

- la lettre O indique une montée (ou descente) composée uniquement de a (ou de b), éventuellement vide;

- la lettre G (respectivement D) indique une montée (éventuellement vide) composée de a', puis de a (respectivement, de b, puis de b');

- la lettre $\overline{G}$ (respectivement $\overline{D}$) indique une montée (respectivement descente) non vide, composée uniquement de a' (respectivement b').

Dans le codage, la largeur (nombre de colonnes) des polyominos verticalement convexes devient le nombre d'occurrences de p, et le demi-périmètre vertical devient le nombre total d'occurrences de a et a' (auquel il faut ajouter 1). Par conséquent, les paramètres largeur, périmètre vertical, et périmètre total, sont tous Q-comptables de rang 1. Par ailleurs, l'aire du polyomino correspond à la somme des hauteurs des pics du mot. C'est donc un paramètre Q-comptable de rang 2, pour lequel la seule substitution de variables utilisée sera $\sigma_{p\leftarrow pq}$. Le système de q-équations est obtenu en remplaçant, dans le système algébrique, chaque symbole U (respectivement, chaque p) apparaissant entre deux lettres a (ou a') et b (ou b'), par $\sigma_{p\leftarrow pq}(U)$ (respectivement, par pq).

Cette grammaire comporte 9 symboles et 108 règles de dérivation, et n'est pas une grammaire propre; itérer les règles qui n'écrivent aucune lettre pour la rendre propre ferait encore augmenter le nombre de nombre de règles. Tel quel, le jacobien du système algébrique est trop gros pour être calculé par Maple. Il est donc hors de question, pour des raisons pratiques, d'exploiter directement ce système.

Toutefois, le calcul sur les séries génératrices (à variables commutatives) permet de ramener ce système à des proportions plus raisonnables. La première simplification consiste à remarquer que les langages GO et OD (respectivement, $\overline{G}O$ et $O\overline{D}$; $\overline{G}D$ et $G\overline{D}$) sont images miroir l'un de l'autre. Nous en déduisons l'identité suivante :

$$OD(a, a', b, b', p) = GO(b, b', a, a', p),$$

ainsi que des identités similaires pour les deux autres couples de langages. Si l'on renonce à différencier les lettres a et a' d'une part, et b et b' d'autre part (ce qui implique de ne pas distinguer, dans le périmètre vertical, la contribution du chemin inférieur, codé par les lettres a' et b, de celle du chemin supérieur, codé par a et b'), on obtient, en posant

$a = a' = b = b' = x^{1/2}$,

$$\begin{cases} OD(x,p) &= GO(x,p) \\ O\overline{D}(x,p) &= \overline{G}O(x,p) \\ G\overline{D}(x,p) &= \overline{G}D(x,p) \end{cases}$$

qui permet d'éliminer 3 des 9 équations du système de départ.

La seconde étape de simplification revient à remarquer qu'il existe une bijection naturelle entre OO et $\overline{G}O + p + p.GO$, laquelle consiste à remplacer tous les a initiaux d'un mot $w \in OO$, par des a'. La même transformation établit également une bijection entre OD et $\overline{G}D + p + p.GD$. Les identités qui en découlent sur les séries génératrices sont :

$$\begin{cases} \overline{G}O(x,p) &= OO(x,p) - p - p.GO(x,p) \\ \overline{G}D(x,p) &= OD(x,p) - p - p.GD(x,p). \end{cases}$$

Le système peut alors s'écrire en n'utilisant que les séries OO, OD et GD (la série $\overline{GD}$, n'apparaissant plus dans les 3 équations, est abandonnée) :

$$\begin{aligned}
(1)\quad OO &= p + pOD + x\underline{OO}(1+OO) + xp\underline{OD}(1+OD) \\
&\quad + x\underline{p}(1+\underline{OD})(OO - p - pOD) + x(\underline{OO} - \underline{p} - \underline{pOD})(OO - p - pOD) \\
(2)\quad OD &= p + pGD + x\underline{OO}(1+OD) + x\underline{OD} + xp\underline{OD}(1+GD) \\
&\quad + x\underline{p}(1+\underline{OD})(OD - p - pGD) + x(\underline{OO} - \underline{p} - \underline{pOD})(OD - p - pGD) \\
(3)\quad GD &= OD + x\underline{OD}(1+OD) + x\underline{GD} + xp\underline{GD}(1+GD) \\
&\quad + x\underline{p}(1+\underline{GD})(OD - p - pGD) + x(\underline{OD} - \underline{p} - \underline{pGD})(OD - p - pGD)
\end{aligned}$$

Dans le système ci-dessus, les termes soulignés sont ceux qui, dans le q-système correspondant à l'énumération suivant l'aire, doivent être remplacés par leur image par $\sigma_{p \leftarrow pq}$.

5.3.3 Paramètres Q-comptables

Nous avons déjà vu que les paramètres *largeur* (ou *demi-périmètre horizontal*) et *demi-périmètre vertical* sont Q-comptables de rang 1 dans la grammaire présentée ci-dessus (il s'agit respectivement, sur les mot, de $|w|_p$ et de $|w|_{a,a'}$), et que le paramètre *aire* est Q-comptable de rang 2.

Comme dans le cas des polyominos parallélogrammes, les paramètres *nombre d'angles rentrants*, *périmètre de sites* et *hauteur de la première colonne* ne sont pas Q-comptables; il suffit d'ailleurs, pour le prouver, d'écrire les arbres de dérivation des mots codants les deux polyominos de la figure 5.2, qui utilisent toujours les mêmes règles.

Ici encore, il serait possible d'utiliser le lemme de marquage supérieur pour obtenir une grammaire plus fine dans laquelle les hauteurs des première et dernière colonnes soient Q-comptables; malheureusement, la taille de la grammaire initiale rendrait cette transformation totalement inexploitable. Toutefois, il est possible d'accéder aux séries de moments de la hauteur de la dernière colonne, ou du produit des hauteurs des première et dernière colonnes : il s'agit des séries génératrices des langages OD et GD.

En effet, si un mot $w \in OO$ s'écrit $w = a^i p w' p b^j$ $(i \geq 0,\ j \geq 0)$, il code un polyomino dont la première colonne est de hauteur $i+1$, et la dernière, de hauteur $j+1$. Or, nous pouvons associer à w, $j+1$ mots de OD :

$$w_{j'} = a^i p w' p b^{j-j'} b'^{j'} b^{j-j'} \quad (0 \leq j' \leq j)$$

et $(i+1)(j+1)$ mots de GD :

$$w'_{i',j} = a'^{i'} a^{i-i'} p w' p b^{j-j'} b'^{j'} \quad (0 \leq i' \leq i, 0 \leq j' \leq j).$$

Tous les mots $w_{j'}$ et $w_{i',j'}$ vérifient

$$\left\{ \begin{array}{rcccl} |w_{j'}|_p & = & |w_{i',j'}|_p & = & |w|_p \\ |w_{j'}|_{a,a'} & = & |w_{i',j'}|_{a,a'} & = & |w|_{a,a'} \end{array} \right.$$

et apportent donc la même contribution à leurs séries génératrices respectives. Par conséquent, la série $OD(x,p)$ est la série de moments de la hauteur de la dernière colonne, et $GD(x,p)$, la série de moments du produit des hauteurs de la première et de la dernière colonne.

5.3.4 Séries génératrices

Il est possible de résoudre explicitement le système d'équations (1) − (3) pour obtenir la série génératrice, mais l'expression obtenue est plus complexe que celle, remarquablement simple, obtenue par Feretić dans [38, 39] (après les changements de variables appropriés) :

$$OO(x,p) = \frac{1-x}{x}\left(1 - \frac{4}{6 - \sqrt{2(1+p) + 2\sqrt{(1-p)^2 - 16\frac{xp}{(1-x)^2}}}}\right).$$

Le système d'équations algébriques fourni par la grammaire ne permet pas d'obtenir une expression aussi élégante, mais il rend possible la comparaison à la série $OD(x,p)$. La méthode n'est ici qu'une version à deux variables de celle utilisée dans [25]; les expressions données par Maple sont également plus simples que celles fournies par Macsyma.

Après élimination de GD (qui n'apparaît qu'au degré 1 dans l'équation 2), on obtient un système algébrique portant sur OO et OD. Pour chaque inconnue, on se ramène alors à une équation de degré 6, à chaque fois factorisable en un produit de deux équations de degrés 2 et 4. Parmi les solutions explicitement fournies par Maple, une seule est alors analytique à l'origine, et les premiers termes de son développement correspondent bien à ceux obtenus en comptant directement les polyominos verticalement convexes de petit périmètre.

Les séries obtenues sont alors :

$$\begin{aligned} OO(x,p) &= \frac{T_1+T_2+T_3}{4xA}, \\ OD(x,p) &= \frac{T_1'+T_2'+T_3'}{4xA}, \end{aligned}$$

avec

$$\begin{aligned} A &= 18(1-x)^2 - p(2-5x+2x^2) \\ B &= 1-2x-2p-12xp+x^2+p^2-2xp^2-2x^2p+x^2p^2 \\ T_1 &= (1-x)\left\{21(1-x)^2 - p(5-14x+5x^2)\right\} \\ T_1' &= -(1-x)^2(17+38x) + p(1-2x)(1+6x-3x^2) \\ T_2 &= -3(1-x)^2\sqrt{B} \\ T_2' &= -(1-x)(1-2x)\sqrt{B} \\ T_3 &= -(1-x)^2\sqrt{2\left(P+Q\sqrt{B}\right)} \\ T_3' &= (1-x)^3\sqrt{\frac{2}{p}\left(P'+Q'\sqrt{B}\right)} \\ P &= (1-x)^2\left(81(1-x)^2 + p(46-232x+26x^2) + p^2(1+x)^2\right) \\ Q &= (1-x)\left(81(1-x)^2 - p(1+x)^2\right) \\ P' &= 144(1-x)^3 - 3p(1-x)(5-14x)(1+10x) \\ &\quad -2p^2(1-2x)(1-3x+14x^2) + p^3(1-x)(1-2x)^2 \\ Q' &= -144(1-x)^2 + 3p(1-2x)(11-14x) - p^2(1-2x)^2 \end{aligned}$$

5.4 Polyominos murs

Les polyominos murs ont été étudiés par Feretić dans [38]; leur série génératrice suivant les périmètres vertical et horizontal sert d'intermédiaire de calcul pour obtenir celle des polyominos verticalement convexes. Prellberg et Brak [69] ont également étudié leur série

génératrice suivant l'aire et les périmètres vertical et horizontal, au moyen de q-équations similaires à celles que nous donnons plus loin.

Les polyominos murs sont les polyominos verticalement convexes dont toutes les colonnes ont leur plus basse cellule à la même hauteur; ce sont également les polyominos verticalement convexes qui sont à la fois dirigés suivant les directions Sud-Ouest/Nord-Est *et* Sud-Est/Nord-Ouest. La figure 5.4 montre un exemple de polyomino mur.

De manière classique, un polyomino mur d'aire n est codé par une *composition* de n, c'est-à-dire une suite (ordonnée) d'entiers strictement positifs dont la somme est n. La bijection est immédiate, chaque colonne du polyomino étant codée par sa hauteur. Toutefois, afin d'avoir un codage par un langage, et de pouvoir étudier simultanément l'aire et les périmètres, nous codons leur frontière par des mots et obtenons l'aire comme paramètre Q-comptable de rang 2.

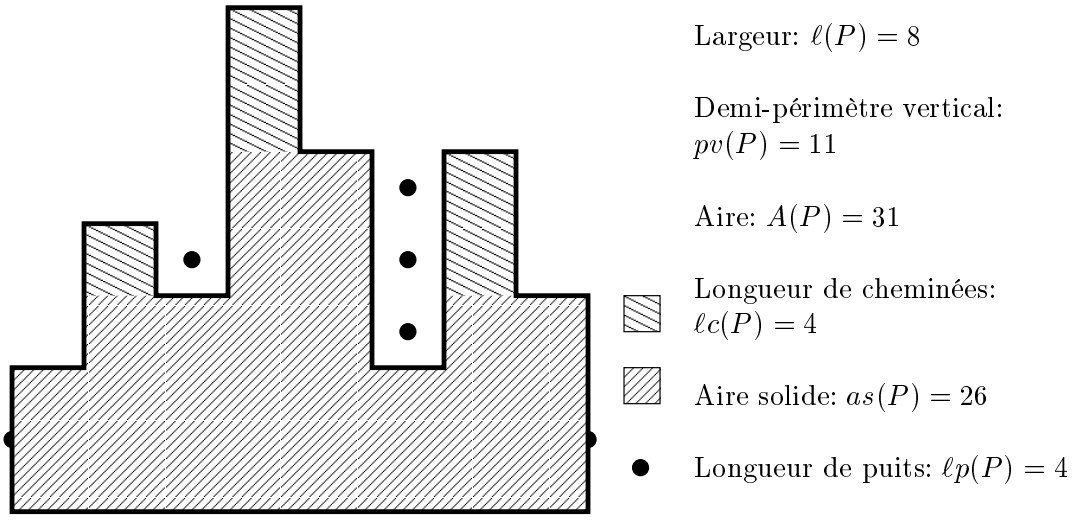


FIG. 5.4: *Un polyomino mur*

5.4.1 Codage et grammaire

Comme sous-classe de polyominos verticalement convexes, les polyominos murs peuvent être codés de la même manière que les polyominos verticalement convexes; un mot du langage OO code un polyomino mur s'il ne comporte pas la lettre a' (puisque cette lettre code les pas verticaux descendants du chemin inférieur), ni la lettre b, sauf éventuellement après la dernière occurrence de p (puisque la lettre b code les pas verticaux montants du chemin inférieur).

Le plus simple, pour coder un polyomino mur, est en fait d'utiliser un mot de OD ne comportant ni a' ni b, ce qui correspond à changer en b' tous les b finaux du mot de OO correspondant. Le mot w codant un polyomino mur P correspond alors à une simple lecture du chemin reliant le coin Nord-Ouest de P à son coin Nord-Est : chaque pas Nord est codé par la lettre a, chaque pas Sud, par la lettre b', et chaque pas Est, par la lettre p.

Nous obtenons donc une grammaire engendrant un codage des polyominos murs, en reprenant la grammaire pour les polyominos verticalement convexes, et en effaçant toutes les règles faisant apparaître l'une des lettres a' et b; l'axiome est, bien entendu, OD. La grammaire obtenue est :

$$\begin{cases} OO &= p + pGO + aODb'p(\epsilon + GO) + a(\epsilon + OD)pb'\overline{G}O \\ & \quad +aO\overline{D}b'\overline{G}O \\ OD &= p + pGD + aODb' + aODb'p(\epsilon + GD) \\ & \quad +a(\epsilon + OD)pb'\overline{G}D + aO\overline{D}b'\overline{G}D \\ O\overline{D} &= pG\overline{D} + a(\epsilon + OD)pb' + aO\overline{D}b' + aODb'pG\overline{D} \\ & \quad +a(\epsilon + OD)pb'\overline{GD} + aO\overline{D}b'\overline{GD} \\ \overline{G}O &= \emptyset \\ \overline{G}D &= \emptyset \\ \overline{GD} &= \emptyset \\ GO &= OO \\ GD &= OD \\ G\overline{D} &= O\overline{D} \end{cases}$$

En retirant les règles faisant apparaître les symboles $\overline{G}O$, $\overline{G}D$ et $\overline{GD}$, et en remplaçant les symboles GO, GD et $G\overline{D}$ par, respectivement, OO, OD et $O\overline{D}$, nous obtenons une grammaire où seul le symbole OD est accessible à partir de lui-même. Par conséquent, le langage codant les polyominos murs est engendré par la grammaire suivante :

$$\begin{cases} R_1 : & OD \rightarrow p \\ R_2 : & OD \rightarrow pOD \\ R_3 : & OD \rightarrow aODb' \\ R_4 : & OD \rightarrow aODb'p \\ R_5 : & OD \rightarrow aODb'pOD \end{cases}$$

Cette grammaire correspond exactement à la décomposition donnée dans [38], où elle est présentée sous la forme d'une grammaire d'objets.

5.4.2 Paramètres Q-comptables

La grammaire utilisée pour coder les polyominos murs étant, en quelque sorte, une restriction de celle utilisée pour les polyominos verticalement convexes, il est clair que la largeur et le demi-périmètre vertical sont toujours Q-comptables de rang 1, et que l'aire est Q-comptable de rang 2.

Les hauteurs des première et dernière colonnes ne sont pas Q-comptables, mais, comme dans le cas des polyominos parallélogrammes, le lemme de marquage supérieur peut être utilisé pour obtenir une grammaire plus fine dans laquelle ils sont Q-comptables de rang 1.

Les angles rentrants Nord-Est du polyomino correspondent exactement, sur le mot qui le code, aux facteurs $b'p$. Il est facile de voir que de tels facteurs sont engendrés par les règles R_4 et R_5, et par elles seules; le nombre d'angles rentrants Nord-Est est donc Q-comptable. En revanche, le nombre d'angles rentrants Nord-Ouest n'est pas Q-comptable. Il est toutefois facile d'écrire une grammaire plus fine dans laquelle ce paramètre est Q-comptable, tout en restant à un niveau de complexité raisonnable :

$$\begin{cases} R'_1 : U \to P & R'_2 : U \to P \\ R'_3 : P \to p & R'_4 : P \to pA \\ R'_5 : P \to pP & \\ R'_6 : A \to aUb' & R'_7 : A \to aUb'p \\ R'_8 : A \to aUb'pA & R'_9 : A \to aUb'pbP \end{cases}$$

Dans cette grammaire d'axiome U, le langage A contient tous les mots de U dont la première lettre est a, et le langage P, ceux dont la première lettre est p. Ainsi, le nombre d'angles rentrants Nord-Ouest (ou nombre d'occurrences du facteur pa) est $p_{R'_4} + p_{R'_8} + p_{R'_9}$, et le nombre d'angles rentrants Nord-Est (ou de facteurs $b'p$) est $p_{R'_7} + p_{R'_8} + p_{R'_9}$.

Dans le travail de Feretić, seuls les polyominos murs ayant un nombre *impair* de colonnes sont réellement utilisés, et, pour une énumération suivant l'aire, seule la contribution à l'aire des colonnes de rang impair doit être considérée. Il n'est pas possible, dans les grammaires présentées ci-dessus, de déterminer si une colonne donnée (codée par une occurrence de p) est de rang pair ou impair. Il n'est donc pas étonnant que le paramètre *somme des hauteurs des colonnes de rang impair* ne soit pas Q-comptable. Nous pouvons toutefois modifier légèrement le codage, de telle sorte que les colonnes de rang impair soient codées par la lettre p, et celles de rang pair, par une nouvelle lettre p'. En distinguant quatre langages auxiliaires suivant que les mots qui les composent ont un nombre pair (P et P') ou impair (I et I') de lettres p ou p', et suivant que la première de celles-ci est un p (I et P)

ou un p' (I' et P'), nous adaptons très simplement la grammaire initiale pour en produire une nouvelle, dans laquelle les paramètre *somme des hauteurs des colonnes de rang impair* est Q-comptable[2] de rang 2 :

$$\begin{cases} I &= p + pP' + aIb' + aPb'p + aIb'p'I + aPb'pP' \\ I' &= p' + p'P + aI'b' + aP'b'p' + aI'b'pI' + aP'b'p'P \\ P &= pI' + aPb' + aIb'p' + aIb'p'P + aPb'pI' \\ P' &= p'I + aP'b' + aI'b'p + aI'b'pP' + aP'b'p'I \end{cases}$$

Un exemple de paramètre intéressant, mais qui semble difficile à obtenir comme paramètre Q-comptable, est le périmètre de sites. La différence entre le périmètre et le périmètre de sites se compose de la somme de deux autres paramètres, dont l'un est le nombre d'angles rentrants (que nous avons pu "attraper" au prix d'une modification raisonnable de la grammaire la plus "simple"), et l'autre est ce que nous pouvons appeler la *longueur totale de puits* : le nombre de cellules extérieures au polyomino, mais dont les deux cellules situées immédiatement à droite et à gauche appartiennent au polyomino – voir figure 5.4. Sur notre codage des polyominos murs, ce paramètre peut être compté en sommant, pour chaque facteur $b'^i p a^j$ maximal, le plus petit parmi i et j. Lorsqu'un paramètre est défini par un *minimum* (ou par un maximum), il y a peu d'espoir de le rendre Q-comptable; il est donc probable que, s'il est possible de rendre ce paramètre Q-comptable, ce sera au moyen d'une grammaire, voire d'un codage, radicalement différents.

Un autre paramètre, en apparence proche de la longueur totale de puits, est la *longueur totale de cheminées* : le nombre de cellules appartenant au polyomino, mais dont aucun des voisins Est et Ouest n'appartient au polyomino. Un tel paramètre est, sur notre langage, très similaire au *poids de pyramides* étudié sur le langage de Dyck par Denise et Simion [32]. Nous pouvons aisément raffiner notre grammaire pour en obtenir une dans laquelle la longueur totale de cheminées est un paramètre Q-comptable de rang 1 :

$$\begin{cases} R_1 : & M \to P & R_2 : & M \to N \\ R_3 : & P \to p & R_4 : & P \to aPb' \\ R_5 : & N \to pP & R_6 : & N \to pN \\ R_7 : & N \to aPb'p & R_8 : & N \to aNb'p \\ R_9 : & N \to aPb'pP & R_{10} : & N \to aPb'pN \\ R_{11} & N \to aNb'pP & R_{12} : & N \to aNb'pN \end{cases}$$

2. Cette nouvelle grammaire n'est pas à proprement parler plus fine que l'originale, car le langage engendré n'est pas le même; en remplaçant p' par p, on obtient une grammaire plus fine que l'originale.

Dans cette nouvelle grammaire (P est le langage des mots n'ayant qu'une occurrence de la lettre p, et N est celui des mots qui en ont au moins 2; cette grammaire est obtenue en effectuant un marquage inférieur des règles qui écrivent la lettre p), la longueur totale de cheminée correspond au paramètre $p_{R_1} + p_{R_4} + p_{R_7} + p_{R_9} + p_{R_{10}}$.

Dans cette même grammaire, l'aire des polyominos peut, sans difficulté, s'exprimer comme paramètre Q-comptable de rang 2, avec comme nom de paramètre $R_1 + \alpha\beta$, où:

$$\begin{cases} \alpha &= 1 + R_4^{(1)} + R_7^{(1)} + R_8^{(1)} + R_9^{(1)} + R_{10}^{(1)} + R_{11}^{(1)} + R_{12}^{(1)}, \\ \beta &= R_3 + R_4 + R_5 + R_6 + R_7 + R_8 + R_9 + R_{10} + R_{11} + R_{12}. \end{cases}$$

Sous cette forme, il n'apparaît pas immédiatement que la différence entre l'aire du polyomino et sa longueur totale de cheminées (qui est le nombre de cellules du polyomino ayant au moins un voisin Est ou Ouest dans le polyomino), est également Q-comptable : si l'on fait formellement la différence entre les deux noms de paramètres, nous obtenons une décomposition faisant intervenir des coefficients négatifs. Toutefois, il convient de remarquer que le symbole N n'est pas accessible à partir de P dans cette grammaire. Par conséquent, les paramètres $p_{R_i^{(1)} R_j}$, pour $i = 4, 7, 9, 10$ et $5 \leq j \leq 12$, sont identiquement nuls, et $p_{R_i^{(1)} R_3} = p_{R_i}$ pour $i = 4, 7, 9, 10$. Nous obtenons alors une décomposition de l'aire en somme de deux paramètres Q-comptables, la longueur de cheminées ℓc et la partie "solide" as (les cellules ayant au moins un voisin latéral dans le polyomino) :

$$\begin{cases} aire &= \ell c + as \\ \ell c &= p_{R_1} + p_{R_4} + p_{R_7} + p_{R_9} + p_{R_{10}} \\ as &= p_{\left(R_8^{(1)} + R_{11}^{(1)} + R_{12}^{(1)}\right).(R_3 + R_5 + R_6 + R_7 + R_8 + R_9 + R_{10} + R_{11} + R_{12})} \\ & \quad + p_{R_3 + R_5 + R_6 + R_7 + R_8 + R_9 + R_{10} + R_{11} + R_{12}} \end{cases}$$

Les équations fournies par cette grammaire étant de faible degré, il n'est pas difficile de les résoudre et d'obtenir les séries génératrices suivant largeur et périmètre vertical.

Bibliographie

[1] A.V. Aho and J.D. Ullman. *The Theory of Parsing, Translation and Compiling. Vol. 1: Parsing.* Prentice-Hall, 1972.

[2] D. André. Solution d'un problème posé par M. Bertrand. *C.R. Acad. Sc.*, pages 436–437, 1887.

[3] G.E. Andrews. Identities in combinatorics II: A q-analog of the Lagrange inversion theorem. *Proc. Amer. Math. Soc*, 53:240–245, 1975.

[4] G.E. Andrews. *q-series: their Development and application in Analysis, Number Theory, Combinatorics, Physics, and Computer Algebra.* American Mathematical Society, 1986.

[5] J.-M. Autebert. *Langages algébriques.* Masson, 1987.

[6] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. A construction for enumerating k-coloured Motzkin paths. In *Computing and Combinatorics, First annual conference, COCOON 95, Proceedings*, number 959 in Lecture Notes in Computer Science, pages 254–263. Springer, 1995.

[7] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. A methodology for plane tree enumeration. *Discrete Math.*, 180(1-3):45–64, 1998.

[8] E.A. Bender. Central and local limit theorems applied to asymptotic enumeration. *J. Combin. Theory Ser. A*, 15:91–111, 1973.

[9] E.A. Bender. Convex n-ominoes. *Discrete Math.*, 8:219–226, 1974.

[10] E.A. Bender and L.B. Richmond. Central and local limit theorems applied to asymptotic enumeration II: Multivariate generating functions. *J. of Comb. Th. A*, 34:255–265, 1983.

[11] J. Berstel. *Transductions and Context-Free Languages*. Teubner Studienbucher, Stuttgart, 1979.

[12] J. Bertrand. Solution d'un problème. *C.R. Acad. Sc.*, page 369, 1887.

[13] M. Bousquet-Mélou. q-énumération de polyominos convexes. Thèse de doctorat, Université Bordeaux 1, 1991.

[14] M. Bousquet-Mélou. A method for the enumeration of various classes of column-convex polygons. *Discrete Math.*, 154:1–25, 1996.

[15] M. Bousquet-Mélou. Rapport d'habilitation. Technical report, LaBRI, Université Bordeaux 1, 1996.

[16] M. Bousquet-Mélou and J.-M. Fédou. The generating function of convex polyominoes: The resolution of a q-differential system. *Discrete Math.*, 137(1-3):53–75, 1995.

[17] R. Brak and A.J. Guttmann. Algebraic approximants: A new method of series analysis. *J. Phys. A: Math. Gen.*, 23(24):1331–1337, 1990.

[18] N. Chomsky and M.-P. Schützenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, pages 118–161. North-Holland, 1963.

[19] L. Chottin. Etude syntaxique de certains langages solutions d'équations avec opérateurs. *Theor. Comp. Sci.*, 5:51–84, 1977.

[20] L. Chottin and R. Cori. Une preuve combinatoire de la rationalité d'une série génératrice associée aux arbres. *RAIRO Informatique théorique et applications*, 16(2):113–128, 1982.

[21] L. Comtet. Calcul pratique des coefficients de Taylor d'une fonction algébrique. *Enseign. Math.*, 10:267–270, 1964.

[22] L. Comtet. *Advanced Combinatorics*. Reidel, 1974.

[23] R. Cori. Un code pour les graphes planaires et ses applications. *Astérisque*, 27, 1975.

[24] R. Cori and J. Richard. Enumération des graphes planaires à l'aide des séries formelles en variables non commutatives. *Discrete Math.*, 2:115–162, 1972.

[25] M. Delest. Generating functions for column-convex polyominoes. *J. of Comb. Th. A*, 48(1):12–31, 1988.

[26] M. Delest. Polyominoes and animals: some recent results. *J. of Math. Chem.*, 8:3–18, 1991.

[27] M. Delest, J.-P. Dubernard, and I. Dutour. Parallelogram polyominoes and corners. *J. Symbolic Computation*, 20(5–6):503–515, 1995.

[28] M. Delest and J.-M. Fédou. Attribute grammars are useful for combinatorics. *J. Theor. Comput. Sci.*, 98:65–76, 1992.

[29] M. Delest and J.-M. Fédou. Enumeration of skew Ferrers diagrams. *Discrete Math.*, 112:65–79, 1993.

[30] M. Delest, D. Gouyou-Beauchamps, and B. Vauquelin. Enumeration of parallelogram polyominoes with given bond and site perimeter. *Graphs and Combinatorics*, 3(4):325–339, 1987.

[31] M. Delest and X. Viennot. Algebraic languages and polyominoes enumeration. *Theor. Comp. Sci.*, 34:169–206, 1984.

[32] A. Denise and R. Simion. Two combinatorial statistics on Dyck paths. *Discrete Math.*, 137:155–176, 1995.

[33] J. Dieudonné. *Calcul Infinitésimal.* Hermann, Paris, 1968.

[34] M. Drmota. Asymptotic distributions and a multivariate Darboux method in enumeration problems. *J. Combin. Theory Ser. A*, 67:169–184, 1994.

[35] M. Drmota. Systems of functional equations. *Random Structures and Algorithms*, 10:103–124, 1997.

[36] I. Dutour. Grammaires d'objets: énumération, bijections et génération aléatoire. Thèse de doctorat, Université Bordeaux 1, 1996.

[37] J.-M. Fédou. Grammaires et q-énumérations de polyominos. Thèse de doctorat, Université Bordeaux 1, 1989.

[38] S. Feretić. The column-convex polyominoes perimeter generating function for everybody. *Croatica Chemica Acta*, 69:741–756, 1996.

[39] S. Feretić. A new way of counting the column-convex polyominoes by perimeter. *Discrete Math.*, 180:173–184, 1998.

[40] P. Flajolet. Analytic models and ambiguity of context-free languages. *Theor. Comp. Sci.*, 49:283–309, 1987.

[41] P. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM J. Discrete Math.*, 3:216–240, 1990.

[42] P. Flajolet, B. Salvy, and P. Zimmermann. Lambda-Upsilon-Omega: An assistant algorithms analyzer. In T. Mora, editor, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 357 of *Lect. Notes Comput. Sci.*, pages 201–212. Springer Verlag, 1989.

[43] P. Flajolet, B. Salvy, and P. Zimmermann. Lambda-Upsilon-Omega: The 1989 cookbook. Rapport technique 1073, Institut National de Recherche en Informatique et en Automatique, 1989.

[44] P. Flajolet, B. Salvy, and P. Zimmermann. Automatic average-case analysis of algorithms. *Theor. Comp. Sci.*, 79(1):37–109, 1991.

[45] P. Flajolet and R. Sedgewick. The average case analysis of algorithms: Complex asymptotics and generating functions. Rapport de recherche 2026, Institut National de Recherche en Informatique et en Automatique, 1993.

[46] P. Flajolet and R. Sedgewick. The average case analysis of algorithms: Multivariate asymptotics and limit distributions. Rapport de recherche 3162, Institut National de Recherche en Informatique et en Automatique, 1997.

[47] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for the random generation of combinatorial structures. *Theor. Comp. Sci.*, 132:1–35, 1994.

[48] A. Garsia. A q-analogue of the Lagrange inversion formula. *Houston Journal of Mathematics*, 7:205–237, 1981.

[49] I. Gessel. A noncommutative generalization and q-analog of the Lagrange inversion formula. *Trans. Amer. Math. Soc*, 257:455–482, 1980.

[50] I. Gessel. A combinatorial proof of the multivariable Lagrange inversion formula. *J. of Comb. Th. A*, 45:178–195, 1987.

[51] I. Gessel and D. Stanton. Applications of q-Lagrange inversion to basic hypergeometric series. *Trans. Amer. Math. Soc*, 277(1):173–201, 1983.

[52] S.W. Golomb. Checker boards and polyominoes. *Amer. Math. Monthly*, 61(10):675–682, 1954.

[53] I.J. Good. Generalizations to several variables of Lagrange's expansion, with applications to stochastic processes. *J. Proc. Camb. Philos. Soc.*, pages 367–380, 1960.

[54] I.P. Goulden and D.M. Jackson. *Combinatorial Enumeration*. John Wiley and Sons, 1983.

[55] A.J. Guttmann. Planar polygons: Regular, convex, almost convex, staircase and row convex. In *Proceedings of the 1991 International Symposium in Statistical Physics*, volume 248 of *Published Conference Proceedings*, pages 12–33. America Institute of Physics, 1991.

[56] M.A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, Reading, Mass., 1978.

[57] K. Kendig. *Elementary Algebraic Geometry*. Springer, New York, 1977.

[58] D.A. Klarner. Some results concerning polyominoes. *Fibonacci Quart.*, 3:9–20, 1965.

[59] D.A. Klarner. Cell growth problems. *Canad. J. Math.*, 19:851–863, 1967.

[60] D.A. Klarner and R.L. Rivest. Asymptotic bounds for the number of convex n-ominoes. *Discrete Math.*, 8:31–40, 1974.

[61] D.E. Knuth. Semantics of context-free languages. *Math. Systems Theory*, 2:127–145, 1968.

[62] C. Krattenthaler. A new q-Lagrange formula and some applications. *Proc. Amer. Math. Soc.*, 90:338–344, 1984.

[63] M. Marcus. Cartes, hypercartes et diagrammes de cordes. Thèse de doctorat, Université de Bordeaux 1, 1997.

[64] A. Odlyzko. Periodic oscillations of coefficients of power series that satisfy functional equations. *Adv. in Math.*, 44:180–205, 1982.

[65] W. Ogden. A helpful result for proving inherent ambiguity. *Math. Syst. Theory*, 2:191–194, 1968.

[66] R.J. Parikh. On context-free languages. *J. Assoc. Comput. Mach*, 13:570–580, 1966.

[67] G. Pólya. On the number of certain lattice polygons. *J. Combinatorial Theory*, 6:102–105, 1969.

[68] T. Prellberg. Uniform q-series asymptotics for staircase polygons. *J. Phys. A: Math. Gen.*, 28:1289–1304, 1995.

[69] T. Prellberg and R. Brak. Critical exponents from non-linear functional equations for partially directed cluster models. *J. Statist. Phys.*, 78:701–730, 1995.

[70] G. N. Raney. Functional composition patterns and power series reversion. *Trans. Amer. Math. Soc.*, 94:441–451, 1960.

[71] R.C. Read. Contributions to the cell growth problem. *Canad. J. Math.*, 14:1–20, 1962.

[72] D.H. Redelmeier. Counting polyominoes: Yet another attack. *Discrete Math.*, 36:191–203, 1981.

[73] M.-P. Schützenberger. Certain elementary families of automata. In *Proc. Symp. on Mathematical Theory of Automata*, pages 139–153. Polytechnic Institute of Brooklyn, 1962.

[74] M.-P. Schützenberger. On context-free languages and push-down automata. *Information and Control*, 6:246–264, 1963.

[75] N.J.A. Sloane. *A Handbook of integer sequences.* Academic Press, 1973.

[76] N.J.A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences.* Academic Press, 1995.

[77] H.N.V. Temperley. Combinatorial problems suggested by the statistical mechanics of domains and of rubber-like molecules. *Phys. Rev.*, 103:1–16, 1956.

[78] W.T. Tutte. A census of planar maps. *Canad. J. Math*, 15:249–271, 1963.

[79] X. Viennot. A survey of polyomino enumeration. In P. Leroux and C. Reutenauer, editors, *Actes du 4e colloque Séries formelles et combinatoire algébrique*, volume 11 of *Publications du LaCIM*, pages 399–420, Montréal, 1992.

[80] T. R. S. Walsh and A. B. Lehman. Counting rooted maps by genus. III: Nonseparable maps. *J. Comb. Theory, Ser. B*, 18:222–259, 1975.

[81] E.T. Whittaker and G.N. Watson. *A Course of Modern Analysis.* Cambridge University Press, 4 edition, 1935.

Q-grammaires: un outil pour l'énumération

Résumé : Cette thèse se situe dans le domaine de la combinatoire énumérative. Les Q-grammaires constituent une extension de la méthode DSV: les objets à énumérer sont codés par les mots d'un langage algébrique. Nous nous concentrons sur l'énumération suivant plusieurs paramètres d'objets codés par des mots. Le formalisme introduit permet d'écrire, pour les séries génératrices suivant plusieurs paramètres, des équations fonctionnelles non nécessairement algébriques, et dont les solutions ne sont généralement pas des séries algébriques.

Les paramètres d'énumération, appelés Q-comptables, sont des paramètres cumulatifs définis au moyen d'attributs synthétisés, et peuvent être interprétés comme comptant certaines familles de sommets dans les arbres de dérivation. Essentiellement, chaque utilisation d'une règle de dérivation fait croître un paramètre Q-comptable d'une quantité qui doit être un paramètre Q-comptable de rang inférieur. Les paramètres Q-comptables comprennent le nombre d'occurrences d'une lettre donnée, et, dans le cas de mots codant des chemins discrets, peuvent inclure des statistiques telles que l'aire de la surface délimitée par le chemin, ou son moment d'inertie.

Nous étudions dans quelles conditions la grammaire utilisée pour engendrer un langage peut être modifiée sans que certains paramètres ne perdent leur caractère Q-comptable. Nous montrons qu'il est possible de mettre les grammaires sous des formes normales sans perte de paramètres Q-comptables.

Sont également abordés le calcul de séries de moments destinées à l'évaluation de valeurs moyennes de paramètres Q-comptables. En particulier, les séries de moments "projetées" sont algébriques, et s'écrivent explicitement en fonction des séries génératrices algébriques des langages engendrés par la grammaire.

Mots-clés : Combinatoire, langages algébriques, q-analogues, énumération, grammaires attribuées, séries génératrices.

Q-grammars: a tool for enumeration

Abstract : Our field of interest is that of enumerative combinatorics. We define Q-grammars, which are an extension of the DSV methodology : the objects to be enumerated are coded by the words in an unambiguous context-free language. Our main concern is with enumeration according to several parameters. Our method allows us to write functional equations for the generating functions which are not algebraic, but rather "multiple-q"-analogs of algebraic equations. The generating functions themselves are usually not algebraic.

Our enumeration parameters, called *Q-countable parameters*, are cumulative parameters which we define through synthesized attributes, and count certain families of nodes in the corresponding derivation trees. Each occurrence of a given derivation rule "increases" a Q-countable parameter by an amount which must be a Q-countable parameter of a lower rank. Q-countable parameters include the number of occurrencs of any given letter; when the words code lattice paths, the area delimited by the path, as well as some moments of inertia, can be other examples of Q-countable parameters.

We describe conditions under which the grammar generating a given language can be changed with no loss of Q-countable parameters. Among other things, we prove that the grammar can be written in a variety of normal forms without such loss.

The problem of computing moment series for Q-countable parameters, as a way of obtaining mean values for these parameters, is also covered. We prove that "projected" moment series are algebraic, and can be written as explicit functions of the algebraic generating functions for the language generated by the context-free grammar.

Keywords : Combinatorics, algebraic languages, q-analogs, enumeration, attribute grammars, generating functions.

The Impact of Technology

on the

Doing of Mathematics

Jonathan Borwein, FRSC

CECM

Centre for Experimental & Constructive Mathematics

Simon Fraser University, Burnaby, BC Canada

Revised April 2000

Joint work in part with T. Stanway

www.cecm.sfu.ca/personal/jborwein/talks.html

MY INTENTIONS

- Part I: TALK a bit
- Part II: SHOW some things
- Part III: and TELL some more

ABSTRACT

Technology has repeatedly promised to transform mathematics pedagogically. More recently it has made similar promises to the research community. That said, mathematics in 1999 looked a lot more like mathematics in 1939 than was the case with any of its sister sciences.

That this is changing is inarguable. The confluence of ubiquitous compute power with new networking and collaborative environments will push the teaching and discovering of mathematics in conflicting directions often beyond our control. The burgeoning role of corporate edu-packages is hardly likely to diminish. Nor are battles over curriculum and its delivery about to stop.

PART I:

I intend to survey and illustrate some of the ways in which twenty-first century mathematics will be changed by these new technologies. I will try to distinguish issues of ownership of technology from those of control over content. I also intend to discuss how as mathematical educators we might best prepare for the coming storms. Finally, as a partner in a small educational technology firm, I will offer some modest prescriptions for living on both sides of the fence.

- Intellectual issues
- Technological issues
- Commercial issues

all bang up against each other.

A CHANGING WORLD

"The world will change. It will probably change for the better. It won't seem better to me."

• J.B. Priestley

.....

"It's generally the way with progress that it looks much greater than it really is."

• From *The Wittgenstein Controversy*, by Evelyn Toynton in the *Atlantic Monthly*, June 1997, pp. 28-41.

◇ The epigraph that Ludwig Wittgenstein (1889-1951) ("whereof one cannot speak, thereof one must be silent") had wished for a never realized joint publication of *Tractatus Logico-Philosophicus* (1922) and *Philosophical Investigations* (1953): suggesting the two volumes are not irreconcilable.

INNOVATION

• Academics mean *new ideas*. Decision makers usually don't:

"Innovation. The process of bringing new goods and services to market. or the result of that process." ('Hard Economic Definition')
◇ *Public Investments in University Research: Reaping the Benefits* (Govt of Canada, 1999)

- 'Sustaining' vs 'disruptive' technologies: e.g.,
  - Hard drives (technology's fruit fly)
  - The backhoe
  - Health Management Organizations
  - The Internet??
- Clayton Christensen, *When New Technologies Cause Great Firms To Fail*, 1997.

PI

• Modern Computer Algebra Systems *know*

$$\boxed{\pi \neq \frac{22}{7}}$$

...

Indeed

$$\int_0^1 \frac{(1-x)^4 x^4}{1+x^2}\,dx = \frac{22}{7} - \pi.$$

and the integrand is positive on $(0, 1)$.

◇ Who knows why Maple (open) or Mathematica (closed) knows what they know?

• Is symbolic computation a sustaining or disruptive technology in the classroom?

THE KEPT UNIVERSITY

" Thorstein Veblen [...] comment[ed] acerbically in 1908 that "business principles" were transforming higher education into "a merchantable commodity, to be produced on a piece-rate plan, rated, bought, and sold by standard units, measured, counted and reduced to staple equivalence by impersonal, mechanical tests."

.....

"New products and new processes do not appear full-grown," Vannevar Bush, President Franklin Roosevelt's chief science adviser, declared in 1944. "They are founded on new principles and new conceptions, which in turn are painstakingly developed by research in the purest realms of science."

• Eyal Press and Jennifer Washburn in *The Kept University*, Atlantic Monthly, March 2000
`www.theatlantic.com/issues/2000/03/press.htm`

◇ Which quote more accurately reflects 2001?

INTELLECTUAL PROMISES ···

- Lively and realistic examples: learning by doing (Papert)
  - 'we are all constructivists now'
- Math goes into colour: sliding down surfaces/virtual reality
- Background pattern-checkers and *inverse calculation*
- Speed & space ≡ insight (demands rapid reinforcement via *micro-parallelism*)
- Individually tailored learning: varied pathways for quick/slow and for distinct modes of thinking
  - *algebraic, analytic, topological*

··· INTELLECTUAL PROMISES

- Promises students richer means to represent and present the fruits of their mathematical imagination
- Increased need to teach how to judge the results of computation (visual candy everywhere)
- Unifying research and teaching, theory and practice (jobs)
- Serious curricular insights from neurobiology ("Sources of Mathematical Thinking: Behavioral and Brain Imaging Evidence," S. Dehaene et al, in *Science*, May 7, **284** (1999)).

INTELLECTUAL PITFALLS

- Wasted or wonderful add-ons ("Newton & Euclid meet Java". The "Idiot pivoter")
- Loss of focus
- Loss of control: student centred learning of hierarchical subjects
- Degradation of long-lived robust mathematical knowledge (unique to our discipline)
- Growing reliance on effectively closed architecture software ('total solutions')
- 'Haves and havenots': class, race, gender
- Degeneration to machine-based rote learning ('buzzword compliant shovelware')

IN THE LONG TERM ···

"Keynes distrusted intellectual rigour of the Ricardian type as likely to get in the way of original thinking and saw that it was not uncommon to hit on a valid conclusion before finding a logical path to it.

.....

'I don't really start', he said, 'until I get my proofs back from the printer. Then I can begin serious writing.' "

• From *Keynes the man* written on the 50th Anniversary of Keynes' death. (Sir Alec Cairncross, in the *Economist*, April 20, 1996)

TECHNICAL PROMISES

- Teachers abilities vs students demands
- Access to global data bases (*free access to information* not *access to free information*)
- Doing what is easy: machines don't think like us.
  - cognitive vs descriptive models
- What we learned earlier is not always easier
- Expert systems & belief revision
- Seamless work-spaces: marriage of text and computation

TECHNICAL PITFALLS

- Legacy software
- Legacy hardware
- The weakest link determines the value
- Over promising payoffs and underestimating effort (reform calculus)
- Infinite time-sinks – especially in higher level courses
- Growing (unavoidable) reliance on commercial software

PART II: SOME DEMONSTRATIONS

- MathSciNet: `e-math.ams.org/mathscinet/`
- Sloane's Encyclopedia of Integer Sequences: `www.research.att.com/~njas/sequences/`
- Let's Do Math (Math Resources): `www.mathresources.ca`
- Math On the Web (Tele-Learning): `www.cecm.sfu.ca/TLRN/`
- Cinderella (Geometry): `www.cinderella.de` (not 'net' (music) or 'com' (porn))
- JavaView: `www-sfb288.math.tu-berlin.de/vgp/javaview/demo/PaPlatonic.html`

PART III: INFORMATION RULES

- Economic laws have not been suspended

◇ Carl Shapiro & Hal Varian, *Information Rules*, 1999.

- Some of the topics they discuss and terms worth reflecting on:
  - branding
  - value networks
  - switching costs
  - lock in
  - vicious and virtuous cycles
  - tipping

THE INFORMATION REVOLUTION

"What the new industries and institutions will be, no one can say yet. No one in the 1520s anticipated secular literature, let alone the secular theater. No one in the 1820s anticipated the electric telegraph, or public health, or photography.

"The one thing (to say it again) that is highly probable, if not nearly certain, is that the next twenty years will see the emergence of a number of new industries. At the same time, it is nearly certain that few of them will come out of information technology, the computer, data processing, or the Internet."

• Peter Drucker, *Beyond the Information Revolution*, Atlantic Monthly, Oct 1999.
`www.theatlantic.com/issues/99oct/9910drucker.htm`

INTELLECTUAL PROPERTY ISSUES

- Different stake-holders often have wildly different views
  - Supervisors and teachers
  - Students (and parents)
  - Professional societies (big and small)
  - Publishing houses (big and small)
  - Software companies (big and small)
- As job security disappears more students see *IP* as their future: (Ma vs Phong &Stein, non-disclosure, insider-trading, interleukin).
- The researcher as CEO: conflicts of interest are inevitable. They must be declared. They are rarely resolved.

OPEN PUBLISHING

- So many issues: access, cost, reliability, inter-operability, charging mechanisms, etc.
- Every day another initiative:
  - Los Alamos server and ArXiv (Math)
    `http://xxx.lanl.gov/archive/math`
  - Santa Fe Initiative (metadata, MathML)
  - International Math Union's *Math-Net*
    `www.ceic.math.ca`
  - National Institutes of Health (grey literature)
  - DOE, AAAS and Fathom Web Sites (validation?)

COMMERCIAL ISSUES

- Can't make what you can't sell
- Can't sell what you can't make (market discipline?)
- Conservatism in the edu-software business: no R&D model
- Commoditization (*macro-media everywhere*)
- Machine closets versus kitchen cabinets
- Weaning from software: overloading the senses (HCI issues)
- Corporate asset stripping: 'dot-com fever'

RIGO(U)R

"I have no satisfaction in formulas unless I feel their numerical magnitude."

• The scientist and entrepreneur, Lord Kelvin (William Thomson, 1824-1907)

.....

"The object of mathematical rigor is to sanction and legitimize the conquests of intuition, and there was never any other object for it."

• J. Hadamard, in E. Borel, Lecons sur la theorie des fonctions, 3rd ed. 1928, quoted in G. Polya, *Mathematical discovery: On understanding, learning, and teaching problem solving (Combined Edition)*, Wiley, (1981).

REALITY

"If you have a great idea, solid science, and earth shaking discoveries, you are still only 10% of the way there."

• David Tomei, LXR Biotechnology Inc, on the vicissitudes of startup companies.

◇ Quoted in *Science* page 1039, Nov. 7, 1997.

.....

"A truly popular lecture cannot teach, and a lecture that truly teaches cannot be popular."

• Michael Faraday: 'When Gladstone was British Prime Minister he visited Faraday's laboratory and asked if some esoteric substance called 'Electricity' would ever have practical significance. *"One day, sir, you will tax it."* was the answer.' (Science, 1994)

SUGGESTIONS AND ···

- Clearly identify expectations of technology
- Be realistic about the learning curve for advanced software (such as *Mathematica* or *Maple*)
- Commit to use of open architecture software (Linux) and open publishing
- Form (not for profit and 'pre-competitive') consortia
  - to share expertise
  - access to markets
  - ability to compete with the big guys

··· CONCLUSIONS

- Opportunity to recapture computing from our sister sciences
- Realistic now to benefit from:
  - advances in cognitive neuroscience
  - advances in software design, and testing, interfaces, expert systems
- Good technology will never be cheap (*Malthusian principle* that 'expectations outstrip performance')

FREEDOM AND DISCIPLINE

"... so long as we conceive intellectual education as merely consisting in the acquirement of mechanical mental aptitudes, and of formulated statements of useful truths, there can be no progress; although there will be much activity, amid aimless rearrangement of syllabuses, in the fruitless endeavour to dodge the inevitable lack of time. "

• A.N. Whitehead, "The Rythmic Claims of Freedom and Discipline" in *The Aims of Education and Other Essays* (1929).

BASIC ANALYTIC COMBINATORICS OF DIRECTED LATTICE PATHS

CYRIL BANDERIER AND PHILIPPE FLAJOLET

Abstract. This paper develops a unified enumerative and asymptotic theory of *directed 2-dimensional lattice paths* in half-planes and quarter-planes. The lattice paths are specified by a finite set of rules that are both time and space homogeneous, and have a privileged direction of increase. (They are then essentially 1-dimensional objects.) The theory relies on a specific "kernel method" that provides an important decomposition of the algebraic generating functions involved, as well as on a generic study of singularities of an associated algebraic curve. Consequences are precise computable estimates for the number of lattice paths of a given length under various constraints (bridges, excursions, meanders) as well as a characterization of the limit laws associated to several basic parameters of paths.

To Maurice Nivat, with many thanks for so many things!

Introduction

By a *lattice path* is meant in all generality a polygonal line of the discrete Cartesian plane $\mathbb{Z} \times \mathbb{Z}$. The lattice paths to be considered here are specified by a finite set of simple rules: typically, from each point, there is a finite set of allowable moves that are both "time independent" and "space independent". Throughout this study, we also assume the existence of some privileged *direction of increase* (the horizontal axis, say), so that paths become essentially similar to one-dimensional objects, namely, walks on the line. Such *directed* lattice paths intervene in many areas of mathematics and computer science. They play a rôle, for instance, in probability theory (sums of discrete random variables), statistics (non-parametric tests), formal language theory, random generation of planar diagrams (animals and polyominoes), the analysis of dynamic data structures, and queueing theory models.

In probability theory, lattice paths describe the evolution of sums of independent discrete random variables, for instance, the succession of your gains if a die is repeatedly cast and your capital is increased by j when face number j shows up. A typical question in this context is the following: *Determine the probability of a "lucky game" in the sense that, at* any *time t, the partial gain is at least as large as the "mean gain", $\frac{7}{2}t$.* Such questions are indeed addressed by classical probability theory, with Brownian motion entering the game. However, by design, stochastic processes only provide a first-order asymptotic theory, while some purely discrete phenomena remain out of reach of this theory.

Date: August 26, 2001.

Key words and phrases. Lattice path, analytic combinatorics, generating function, kernel method, algebraic function, singularity analysis, generalized ballot problem, Catalan numbers.

Statistics, though not our primary motivation in this paper, is historically an other important source of problems regarding lattice paths. We may mention the Kolmogorov-Smirnov test in non-parametric statistics that aims at discerning whether two random variates have the same distribution (see, e.g., [47]). As a matter of fact, the early books on lattice path combinatorics and lattice path statistics by Mohanty and Narayana [57, 59] specifically draw some of their motivations from such questions.

In discrete mathematics, all sorts of constrained lattice paths serve to describe apparently complex objects. Two-sorted permutations are for instance equivalent to paths made of horizontal and vertical steps that connect the origin to a point lying on the main diagonal—such facts are directly relevant to the analysis of the mergesort and shellsort algorithms [48, 69, 74]. Dyck paths that are closely related to diagonal paths describe traversal sequences of general and binary trees; they belong to what Riordan has named the "Catalan domain", that is, the orbit of structures counted by the Catalan numbers, $\frac{1}{n+1}\binom{2n}{n}$. The wealth of properties surrounding Dyck paths can be perceived when examining either Gould's monograph [41] that lists 243 references or from Exercise 6.19 in Stanley's book [72] whose statement alone spans more than ten full pages. More generally, trees constrained by degrees—e.g., term trees in free magmas, of interest in formal semantics [60]—are known to be bijectively equivalent to Łukasiewicz words, themselves isomorphic to lattice paths of a special form; Lothaire's book offers a good description within the framework of combinatorics on words [52, Chap. 11].

Lattice paths also intervene in the analysis of dynamically evolving structures, and, as such, they surface in the continuous as well as discrete parts of the theory. On the discrete side, we have Flajolet's combinatorial theory of continued fractions [29] motivated by Françon's theory of "histories" of dynamic data structures [32, 36] or Knuth's dynamic storage allocation model (see [46, 2.2.2–13] for the statement of the problem and [30, 75] for solutions). As regards continuous aspects, the Karlin-McGregor theory of birth-death processes (of which [33, 58] offer lattice-path perspectives), itself closely related to various queueing theory models, involves lattice paths that describe an interesting collection of events (the embedded Markov chain). The recent book by Fayolle *et al.* on random walks in the quarter-plane [26] is historically motivated by such queueing theory questions [25].

Word representations of lattice paths also provide many examples of context-free languages. This side of the coin is closely related to encodings of trees by words, so that Dyck paths (that are associated to general trees and binary trees) and Motzkin paths (that encode unary-binary trees) play an especially important rôle. The theory of context-free languages and pushdown automata then combines nicely with the Chomsky-Schützenberger theorems [10, 73], to the effect that many types of paths can be *a priori* recognized as admitting generating functions that are algebraic. Examples are provided by Labelle and Yeh [49, 50], Merlini *et al.* [56], and Duchon [22]. (In return, enumerative studies related to context-free languages can sometimes provide structural information on generation mechanisms and formal languages as is evidenced by the analytic theory of inherent ambiguity of [31].)

Finally, because of the rich combinatorics surrounding them, lattice paths intervene at many places in the random generation of structured objects. The problem there is to draw a combinatorial object from some class $\mathcal{C}$, and do so uniformly at random amongst all objects of size n in $\mathcal{C}$. Strong decomposability properties of paths usually make random generation possible in low polynomial time (usually

with a complexity between $O(n)$ and $O(n^2)$). Consequently, any easily computable bijection between a class $\mathcal{C}$ and a class of simple enough lattice paths induces a random generation algorithm for $\mathcal{C}$. Known examples include the random generation of two dimensional diagrams like polyominoes and animals. For instance, the Delest-Viennot methodology of [18] allows us to generate parallelogram polyominoes in linear time; the rejection methods of the "Florence School" [8] make it possible to generate various types of directed lattice animals in a surprisingly efficient manner. The design of such algorithms is clearly dependent on the basic combinatorics of lattice paths while the corresponding performance analyses rely on fine probabilistic estimates of characteristic properties of paths; see Louchard's contribution [53] for a neat example and the paper [4] for algebraic techniques related to the present paper.

In this introduction, we cannot do more than scratch the surface of such rich combinatorial, probabilistic, and algorithmic aspects of lattice paths. Accordingly we cut short our discussion of motivations at this point.

Scope of the paper. This paper assembles combinatorics of words and paths, some algebra of formal power series, and complex analysis. Under this angle, we believe the enterprise to be original. Quite a lot is otherwise known regarding probabilistic properties of paths, as these represent sums of random variables. Accordingly, our treatment can be, to some extent, regarded as a parallel of probabilistic-analytic methods in the realm of enumerative combinatorics.

In Section 2, we show that the counting generating functions of paths of various sorts are invariably *algebraic functions*. This algebraic character is predictable since the word encodings of the object considered are clearly recognizable by deterministic pushdown automata, hence are deterministic context-free languages. However, for directed lattice paths, we demonstrate that a strong algebraic *decomposability* prevails that is obtained by a specific technique, the *"kernel method"* (historical remarks are given at the end of Section 2.2) and is not clearly visible on combinatorial and grammatical descriptions. Our purpose in this paper is to arrive eventually at a complete characterization of the singular structure of intervening generation functions (Section 3)—by virtue of the method of *singularity analysis*, this leads to very precise asymptotic information on the counting quantities involved. At this level also, the decomposability granted by the kernel method is central as it enables us to determine the location and nature of dominant singularities. Then, once the singular structure of counting generating functions has been extracted, tight estimates on probability distributions of parameters follow easily: see Section 4 for a sample of what can be done. Section 5 sketches extensions to the enumeration of certain types of planar objects provided they satisfy a strong directedness condition.

1. Lattice paths and generating functions

This section presents the varieties of lattice paths to be studied as well as their companion generating functions.

Definition 1. *Fix a finite set of vectors of* $\mathbb{Z} \times \mathbb{Z}$, $\mathcal{S} = \{(a_1, b_1), \dots, (a_m, b_m)\}$. *A* lattice path *or* walk *relative to* $\mathcal{S}$ *is a sequence* $v = (v_1, \dots, v_n)$ *such that each* v_j *is in* $\mathcal{S}$. *The geometric realization of a lattice path* $v = (v_1, \dots, v_n)$ *is the sequence of points* $(P_0, P_1, \dots, P_n)$ *such that* $P_0 = (0, 0)$ *and* $\overrightarrow{P_{j-1}P_j} = v_j$. *The quantity* n *is referred to as the* size *of the path.*

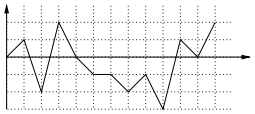
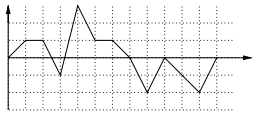
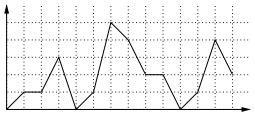
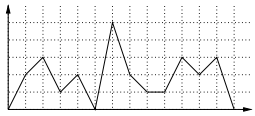
	ending anywhere	ending at 0
unconstrained (on $\mathbb{Z}$)	 walk/path ($\mathcal{W}$) $W(z) = \dfrac{1}{1 - zP(1)}$	 bridge ($\mathcal{B}$) $B(z) = z\sum_{i=1}^{c} \dfrac{u_i'(z)}{u_i(z)}$
constrained (on $\mathbb{Z}_{\geq 0}$)	 meander ($\mathcal{M}$) $M(z) = \dfrac{1}{1 - zP(1)} \prod_{i=1}^{c} (1 - u_i(z))$	 excursion ($\mathcal{E}$) $E(z) = \dfrac{(-1)^{c-1}}{p_{-c} z} \prod_{i=1}^{c} u_i(z)$

FIGURE 1. The four types of paths: walks, bridges, meanders, and excursions and the corresponding generating functions.

In the sequel, we shall identify a lattice path with the polygonal line admitting $P_0, \ldots, P_n$ as vertices. The elements of $\mathcal{S}$ are called *steps* or *jumps*, and we also refer to the vectors $\overrightarrow{P_{j-1}P_j} = v_j$ as the steps of a particular path.

Various constraints will be imposed on paths. In particular we restrict attention throughout this paper to *directed paths* defined by the fact that if (a, b) lies in $\mathcal{S}$, then necessarily one should have $a > 0$. In other words, a step always entails progress along the horizontal axis and the geometric realization of the path naturally lives in the half plane $\mathbb{Z}_{\geq 0} \times \mathbb{Z}$. (This constraint rules out paths like the ones occurring in Pólya's "drunkard problem" as described in the attractive booklet of Doyle and Snell [19]; it also implies that the paths studied can be treated essentially as 1-dimensional objects.) The following conditionings are to be considered (Figure 1).

Definition 2. *A* bridge *is a path whose end-point P_n lies on the x-axis. A* meander *is a path that lies in the quarter plane $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$. An* excursion *is a path that is at the same time a meander and a bridge; it thus connects the origin to a point lying on the x-axis and involves no point with negative y–coordinate.*

A family of paths is said to be simple *if each allowed step in $\mathcal{S}$ (Definition 1) is of the form $(1, b)$ with $b \in \mathbb{Z}$. In this case, we also abbreviate $\mathcal{S}$ as $\mathcal{S} = \{b_1, \ldots, b_m\}$.*

In the simple case the size of a path coincides with its span along the horizontal direction, that is, its *length*. The terminology of bridges, meanders, and excursions is chosen to be consistent with the standard one adopted in Brownian motion theory; see, e.g., [62].

The main objective of this paper is to enumerate exactly as well as asymptotically paths, bridges, and meanders, this with special attention to simple families. Once the set of steps is fixed, we let $\mathcal{W}$ and $\mathcal{B}$ denote the set of paths and bridges

respectively ($\mathcal{W}$ being reminiscent of "walk"); we denote by $\mathcal{M}$ and $\mathcal{E}$ the set of meanders and excursions.

Given a class $\mathcal{C}$ of paths, we let $\mathcal{C}_n$ denote the subclass of paths that have size n, and, whenever appropriate, $\mathcal{C}_{n,k} \subset \mathcal{C}_n$ those that have final vertical abscissa (also known as "final altitude") equal to k. With the convention of using standard fonts to denote cardinalities of the corresponding sets (themselves in calligraphic style), $C_n = \mathrm{card}(\mathcal{C}_n)$ and $C_n = \mathrm{card}(\mathcal{C}_{n,k})$, the corresponding (ordinary) *generating functions* (GF's) are then

$$C(z) := \sum_n C_n z^n, \qquad C(z,u) = \sum_{n,k} C_{n,k}\, u^k z^n.$$

This paper is entirely devoted to characterizing these generating functions: they are either rational functions (W) or algebraic functions (B, E, M). As we shall see, a strong algebraic decomposition prevails which, as opposed to other approaches, renders the calculation of the GF's effective. Even more importantly, the decomposability of GF's makes it possible to extract their singular structure, and in turn solve the corresponding asymptotic enumeration problems in a wholly satisfactory fashion.

Weighted paths. For several applications, it is useful to associate *weights* to single steps. In this case, the set of steps $\mathcal{S}$ is coupled with a system of weights $\Pi = \{w_1, \dots, w_m\}$, with $w_j > 0$ the weight associated to $(a_j, b_j) \in \mathcal{S}$; the weight of a path is then defined as the *product* of the weights of its individual steps. Then the quantity C_n, still referred to as *number of paths* (of size n), represents the total weight of all paths of size n. Such weighted paths cover several situations of interest: (i) combinatorial paths in the standard sense above when each $w_j = 1$; (ii) paths with coloured steps, e.g., $w_j = 2$ means that the corresponding step (a_j, b_j) has two possible coloured incarnations (say blue and yellow); (iii) $\sum w_j = 1$ corresponds to a probabilistic model of paths where, at each stage, step (a_j, b_j) is chosen with probability w_j.

2. Algebraic structures and the kernel method

In this section, we characterize the generating functions of the four types of directed paths (unconstrained, bridges, meanders, and excursions). For ease of exposition, we restrict attention to simple families of paths till Section 5, where we briefly discuss the more general directed models. It will be seen that a specific algebraic curve, the "characteristic curve" plays a central rôle. In this section, a modicum of analysis is introduced for convenience, but it is limited to the vicinity of $z = 0$, and consequently, it is largely equivalent to formal series manipulations[1].

Definition 3. *Let $\mathcal{S} = \{b_1, \dots b_m\}$ be a simple set of jumps, with $\Pi = \{w_1, \dots, w_j\}$ the corresponding system of weights ($w_j \equiv 1$ in the unweighted case). The* characteristic polynomial *of $\mathcal{S}$ is defined as the polynomial in u, u^{-1} (a Laurent polynomial)*

$$P(u) := \sum_{j=1}^{m} w_j\, u^{b_j}.$$

[1] Following a remark by a referee, we note that analyticity considerations in this section could be logically dispensed with; see Gessel's paper [38] for a proper framework. However, the authors' feeling is that purely algebraic proofs, though feasible, tend to be less transparent. More importantly, analyticity considerations developed here serve as a useful preparation for our "nonlocal" treatment of singularities in the next section.

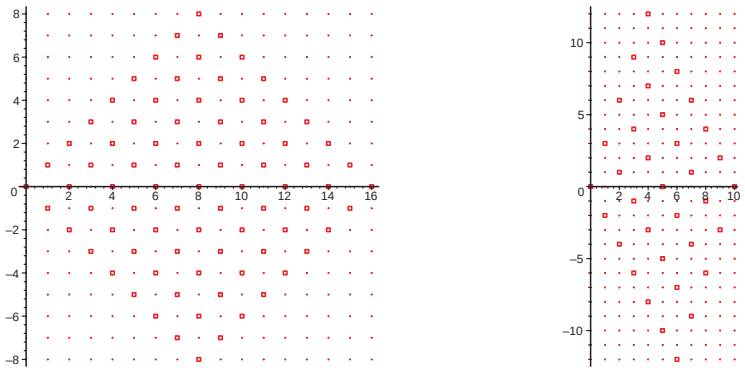


FIGURE 2. Fragments of the sublattices accessible from the origin by the Dyck walk ($\mathcal{S} = \{-1, +1\}$) and Duchon's clubs ($\mathcal{S} = -2, +3\}$). The periods are 2 and 5 respectively.

Let $c = -\min_j b_j$ and $d = \max_j b_j$ be the two extreme vertical amplitudes of any jump, and assume throughout $c, d > 0$. The characteristic curve *of the lattice paths determined by $\mathcal{S}$ is the plane algebraic curve defined by the equation*

$$1 - zP(u) = 0, \qquad \textit{or equivalently} \quad u^c - z(u^c P(u)) = 0. \tag{1}$$

The quantity $K(z, u) := u^c - zu^c P(u)$ is also referred to as the kernel *and Equation* (1) *as the* kernel equation.*.*

As we shall see the characteristic equation plays a central rôle, the second form being the entire version (that is, a form without negative powers).

We also need to introduce technical conditions on periodicities. In a coin-tossing game ($\mathcal{S} = \{-1, +1\}$) for instance, a bridge or an excursion only exists for even lengths; consequently, what is observed of a random path at time n depends on the residue class of n modulo 2 (Figure 2).

Definition 4. *A Laurent series $h(z) = \sum_{n \geq -a} h_n z^n$ is said to admit* period *p if there exists a Laurent series H and an integer b such that*

$$h(z) = z^b H(z^p); \tag{2}$$

the largest p such that a decomposition (2) holds is called the period *of h and is denoted by* $\mathrm{per}(h)$*. The series h is called* aperiodic *if* $\mathrm{per}(h) = 1$*.*

A simple walk defined by the set of jumps $\mathcal{S}$ is said have period *p if the characteristic polynomial $P(u)$ has period p.*

A simple walk is said to be reduced *if the gcd of the jumps is equal to 1.*

In what follows, we systematically restrict attention to *reduced walks* since, up to a linear change of abscissa, any walk can be reduced. For instance, the walks corresponding to $\mathcal{U} = \{-3, +3\}$ are transformed (upon shrinking the vertical axis by a factor of $\frac{1}{3}$) into the reduced form $\mathcal{S} = \{-1, +1\}$. (Aperiodic walks are from their definition automatically reduced.) *Periodic walks* live on sublattices: the walks associated to $\mathcal{S} = \{-1, +1\}$ (Dyck walks) and $\mathcal{T} = \{-1, 0, +1\}$ (Motzkin walks) are naturally reduced, but Dyck walks are periodic with $p = 2$ (since $uP(u) = 1 + u^2$),

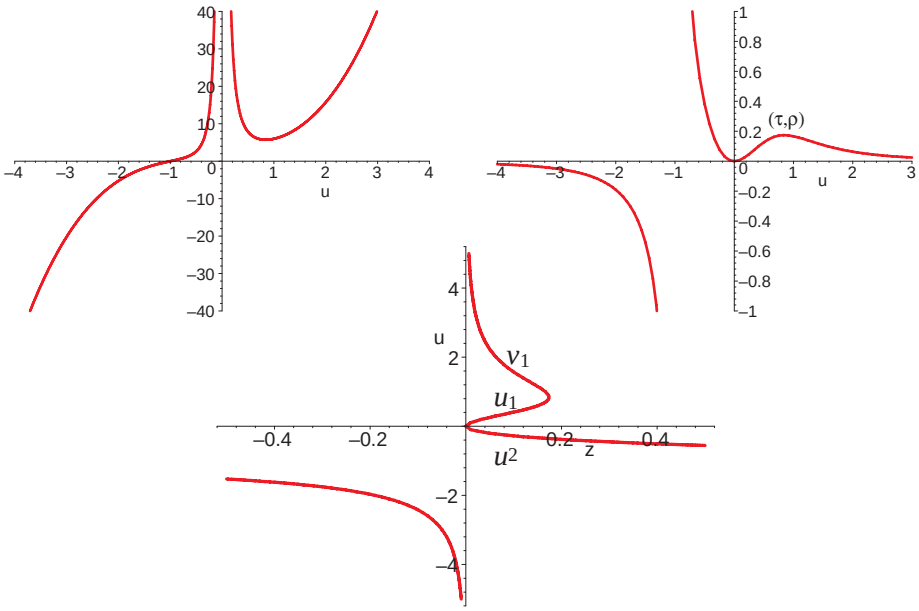


FIGURE 3. Graphs associated to the set of jumps $\mathcal{S} = \{-2,-1,0,1,2,3\}$, with characteristic polynomial $P(u) = u^{-2} + u^{-1} + 1 + u + u^2 + u^3$. Top: the graphs of $P(u)$ and $1/P(u)$ for real u. Bottom: the three real branches of the characteristic curve, one large of order $z^{-1/3}$, and two small of order $\pm z^{1/2}$ (two complex branches of order $e^{\pm 2i\pi/3} z^{-1/3}$ are not shown).

while Motzkin walks are aperiodic; "Duchon's clubs" studied below and defined by $\mathcal{S} = \{-2, +3\}$ have period $p = 5$ (since $u^2 P(u) = 1 + u^5$), etc.

Notice that, if we write

$$P(u) = \sum_{j=1}^{m} w_j u^{b_j}, \qquad w_j \neq 0,\ b_j \in \mathbb{Z}, \tag{3}$$

the period of P (and of the set of jumps $\mathcal{S}$) is

$$p = \mathrm{per}(P) = \gcd\left(b_2 - b_1, \ldots, b_m - b_1\right).$$

Also, by the strong form of the triangle inequality, for an aperiodic $P(u)$, the *strict* inequality holds in

$$|P(u)| < P(|u|) \quad \text{for all } u \in \mathbb{C} \setminus \mathbb{R}_{>0}. \tag{4}$$

It proves convenient to rewrite

$$P(u) = \sum_{k=-c}^{d} p_k u^k.$$

Examination of the asymptotic regimes consistent with the characteristic equation near $z = 0$ shows that the equation can only be satisfied if one of the two relations,

$$p_d z u^d \sim 1 \qquad \text{or} \qquad p_{-c} z u^{-c} \sim 1 \qquad (z \to 0), \tag{5}$$

is satisfied. The characteristic equation being of degree $c + d$ in u is known to have generically $c + d$ roots; these constitute the *branches* of a single algebraic curve defined by (1) and called the *characteristic curve*. Then, as suggested by (5), one expects, in the complex domain (for z near 0), c "small branches" that we write as $u_1, \dots, u_c$ and d "large branches" $v_1 \equiv u_{c+1}, \dots, v_d \equiv u_{c+d}$ satisfying (Figure 3)

$$u_j(z) \sim e^{2i(j-1)\pi/c} (p_{-c})^{1/c} z^{1/c}, \qquad v_k(z) \sim e^{2i(1-k)\pi/d} (p_d)^{-1/d} z^{-1/d}. \tag{6}$$

For determinacy, *one restricts attention to the complex plane slit along the negative real axis*, which allows us to talk freely of the individual branches in the sequel.

The informal discussion summarized by (6) is vindicated by the classical theory of Newton-Puiseux expansions—the fundamental result in the elementary theory of algebraic curves that determines constructively all the possible behaviours of solutions of polynomial equations. For an exposition, we refer to one of the many excellent books on the basic theory of algebraic curves, e.g., [1, 45]. Precisely, the general theory teaches us that the small branches are conjugate of each other at 0, and similarly for the large branches at ∞. This means that there exist functions A and B analytic at 0 and nonzero there, such that, in a neighbourhood of 0, one has

$$\begin{array}{rclcrcl} u_j(z) & = & \omega^{j-1} z^{1/c} A(\omega^{j-1} z^{1/c}) & = & u_1(e^{2i(j-1)\pi} z), & \omega = e^{2i\pi/c} \\ v_k(z) & = & \varpi^{1-k} z^{-1/d} B(\varpi^{k-1} z^{1/d}) & = & v_1(e^{2i(k-1)\pi} z), & \varpi = e^{2i\pi/d}. \end{array} \tag{7}$$

In summary, the u_j and v_ℓ organize themselves into two "cycles" of c and d elements respectively; for analytic details, we refer to Hille's crisp presentation based on monodromy and analytic continuation in [44].

The branch u_1 defined near 0 by (6) is real positive and is called the *principal* (small) branch. The graph of branches is obtained by interchanging the axes in the graph of $1/P(u)$, with u_1 appearing as the real positive branch near the origin; see Figure 3 for an example. We shall prove in Section 3 that in a proper sense u_1 "dominates" all the other small branches.

2.1. **Walks and bridges.** We start with the easy case of unconstrained walks and bridges. This already makes use of the characteristic curve and some of its branches.

Theorem 1. *The bivariate generating function (BGF) of paths (with z marking size and u marking final altitude) relative to a simple set of steps $\mathcal{S}$ with characteristic polynomial $P(u)$ is a rational function. It is given by*

$$W(z, u) = \frac{1}{1 - zP(u)}. \tag{8}$$

The GF of bridges is an algebraic function given by

$$B(z) = z \sum_{j=1}^{c} \frac{u_j'(z)}{u_j(z)} = z \frac{d}{dz} \log\left(u_1(z) \cdots u_c(z)\right), \tag{9}$$

where the expressions involve all the small branches $u_1, \dots, u_c$ of the characteristic curve (1). Generally, the GF W_k of paths terminating at altitude k is, for $-\infty <$

$k < c$,

$$W_k(z) = z \sum_{j=1}^{c} \frac{u_j'(z)}{u_j(z)^{k+1}} = -\frac{z}{k}\frac{d}{dz}\left(\sum_{j=1}^{c} u_j(z)^{-k}\right), \tag{10}$$

and for $-d < k < +\infty$,

$$W_k(z) = -z \sum_{j=1}^{d} \frac{v_j'(z)}{v_j(z)^{k+1}} = \frac{z}{k}\frac{d}{dz}\left(\sum_{j=1}^{d} v_j(z)^{-k}\right), \tag{11}$$

where $v_1, \ldots, v_d$ *are the large branches.*

(For W_0, the second form is to be taken in the limit sense $k \to 0$.)

Proof. Set $w_n(u) = [z^n]W(z,u)$, the Laurent polynomial that describes the possible altitudes and the number of ways to reach them in n steps. We have $w_0(z) = 1$, $w_1(z) = P(u)$, and $w_{n+1}(z) = P(u)w_n(z)$, so that $w_n(z) = P(u)^n$ for all n. The determination of $W(z,u)$ in (8) follows from

$$\sum_{n\geq 0} P(u)^n z^n = \frac{1}{1 - zP(u)},$$

where the sum converges and represents an analytic function of both arguments for $|z| < 1/P(|u|)$. Observe that the resulting series is entire in z but of the Laurent type in u (it involves arbitrary negative powers of u).

For positive u, the radius of convergence of $W(z,u)$ viewed as a function of z is exactly $1/P(u)$. Also, by dominance of coefficients (one has $B_n \leq P(1)^n$), the radius of convergence of $B(z)$ as a function of z is at least $1/P(1)$. Consider now $|z| < r$, where $r := \frac{1}{2}P(1)^{-1}$. Then, since $1/P(u)$ is continuous and unimodal for $u \in (0, +\infty)$ (where $P''(u) > 0$, so that P is convex) and $1/P(0) = 1/P(\infty) = 0$, there exists an interval (α, β) such that for $\alpha \leq u \leq \beta$, one has $1/P(u) > r$. More generally, by positivity of the coefficients, the function $W(z,u)$ is seen to be analytic in the product domain

$$(z,u) \in \{z \mid |z| < r\} \times \{u \mid \alpha < |u| < \beta\}.$$

Thus, by Cauchy's formula applied to the function $W(z,u)$ (viewed now as a function of u analytic in a crown), one has[2]

$$B(z) = [u^0]W(z,u) = \frac{1}{2i\pi}\int_{|u|=(\alpha+\beta)/2} W(z,u)\,\frac{du}{u}.$$

Take z small enough, so that all the large branches that escape to infinity lie outside of $|u| \leq (\alpha+\beta)/2$ *and* the small branches are all distinct. Then, only the small branches remain inside, and, since there are only simple poles, one has

$$\operatorname*{Res}_{u=u_j}\left(\frac{1}{u(1-zP(u))}\right) = -\frac{1}{zu_jP'(u_j)}. \tag{12}$$

The integration contour is shrunk to 0, which is legitimate since $W(z,u)$ remains $O(1)$, and residues are taken into account. The residue theorem then gives $B(z)$ as a sum of residues of the form (12) over all small branches. The formula simplifies

[2]We make use of the conventional notation for coefficients of entire and Laurent series: $[z^n]\sum_n f_n z^n := f_n$.

to (9) since differentiation of the characteristic equation shows that $P'(u)^{-1} = -z^2u'$ for any branch u.

The same procedure is applicable to

$$W_k(z) \equiv [u^k]W(z,u) = \frac{1}{2i\pi}\int_{|u|=(\alpha+\beta)/2} W(z,u)\,\frac{du}{u^{k+1}}.$$

The integration contour can be shrunk to zero provided the integrand (which is of order u^{c-k-1}) remains bounded as $u \to 0$, which necessitates $k \le (c-1)$. The result of (10) follows again from a residue calculation involving small branches. (The proof shows the formulæ to be valid in a small enough neighbourhood of the origin. The identities are then *a posteriori* valid as identities between formal (fractional) power series.)

When $k > -d$, which covers the case (11) of an arbitrary positive k, the residue calculation is completed by extending the contour to a large circle at ∞; in this case, the large branches contribute.

The algebraic character of $B(z)$ and the $W_k(z)$ finally results from the well-known fact that algebraic functions are closed under sums, products, and multiplicative inverses. □

The quantity $B(z) \equiv W_0(z)$ is equivalently given as the diagonal of a bivariate rational function,

$$B(z) = \sum_n \left([z^n u^{cn}] \frac{1}{1 - zu^c P(u)} \right) z^n,$$

and as such it must be algebraic: see Pólya's paper [63] of 1921 and [37] for developments regarding diagonals of rational functions.

Example 1. *Central binomial and trinomial numbers.* These are perhaps the most famous examples, associated to the sets $\mathcal{S} = \{-1, +1\}$ and $\mathcal{T} = \{-1, 0, +1\}$. The corresponding polynomials are $P^{\mathcal{S}}(u) = u^{-1} + u$ and $P^{\mathcal{T}}(u) = u^{-1} + 1 + u$. In this case, the characteristic curve is of degree 2 and there is only one small branch, namely

$$u_1^{\mathcal{S}}(z) = \frac{1 - \sqrt{1-4z^2}}{2z}, \quad u_1^{\mathcal{T}}(z) = \frac{1 - z - \sqrt{1-2z-3z^2}}{2z}.$$

The algebraic generating functions of bridges are then

$$\begin{aligned} B^{\mathcal{S}}(z) &= \frac{1}{\sqrt{1-4z^2}} = 1 + 2\,z^2 + 6\,z^4 + 20\,z^6 + 70\,z^8 + 252\,z^{10} + \cdots \\ B^{\mathcal{T}}(z) &= \frac{1}{\sqrt{1-2z-3z^2}} = 1 + z + 3\,z^2 + 7\,z^3 + 19\,z^4 + 51\,z^5 + \cdots, \end{aligned}$$

the coefficients being[3] *EIS* **A000984** and *EIS* **A002426**

$$[z^n]B^{\mathcal{S}}(z) = [t^n](1+t^2)^n \equiv \binom{2n}{n}, \qquad [z^n]B^{\mathcal{T}}(z) = [t^n](1+t+t^2)^n.$$

The names of central binomial and trinomial numbers are suggested by the usual expansions of $(1+t^2)^n$ and $(1+t+t^2)^n$:

[3] References to EIS point to Sloane's *Encyclopedia of Integer Sequences* [70], of which a version also exists in print [71].

$$\begin{array}{ccccccccc}
 & & & & \mathbf{1} & & & & \\
 & & & 1 & + & t^2 & & & \\
 & & 1 & + & \mathbf{2t^2} & + & t^4 & & \\
 & 1 & + & 3t^2 & + & 3t^4 & + & t^6 & \\
1 & + & 4t^2 & + & \mathbf{6t^4} & + & 4t^6 & + & t^8
\end{array}
\qquad
\begin{array}{ccccccccc}
 & & & & \mathbf{1} & & & & \\
 & & & 1 & +\mathbf{1t} & +t^2 & & & \\
 & & 1 & +2t & +\mathbf{3t^2} & +2t^3 & +t^4 & & \\
 & 1 & +3t & +6t^2 & +\mathbf{7t^3} & +6t^4 & +3t^5 & +t^6 & \\
1 & +4t & +10t^2 & +16t^3 & +\mathbf{19t^4} & +16t^5 & 10t^6 & +4t^7 & +t^8
\end{array}$$

It is notable that these cases were already considered by Euler [24] who also gave linear recurrences (with polynomial coefficients) satisfied by $B_n^{\mathcal{T}}$. □

2.2. **Meanders and excursions.** In this section, we consider meanders, that is paths that never go below the horizontal axis. The meanders whose final altitude is 0 are called excursions, in accordance with Definition 2, and they turn out to be the objects with the richest combinatorial properties.

We continue with a simple system of paths defined by the set of jumps $\mathcal{S}$, possibly endowed with weights. The new generating functions will again involve the characteristic curve together with its small and large branches. Let now $F_{n,k}$ be the number of meanders of size (i.e., length) n that end at altitude k. The corresponding BGF is

$$F(z,u) := \sum_{n,k} F_{n,k} u^k z^n,$$

which is now an entire series in both z and u. By the combinatorial origin of the problem, $F(z,u)$ is bivariate analytic for $|u| \leq 1$ and $|z| < 1/P(1)$. We also make use of the polynomials $f_n(u)$ that describe the possible positions after n steps and write

$$F(z,u) = \sum_{n\geq 0} f_n(u) z^n = \sum_{k\geq 0} F_k(z) u^k. \tag{13}$$

Combinatorially, the natural decomposition is the one based on the last step added. For the $f_n(u)$, "adding a slice" is translated by the recurrence,

$$f_0(u) = 1, \qquad f_{n+1}(u) = P(u) f_n(u) - \{u^{<0}\} P(u) f_n(u). \tag{14}$$

There, the notation $\{u^{<r}\}g(u)$ means the sum of all the monomials with exponent less than r that appear in the Laurent series $g(u)$:

$$\{u^{<r}\} \left(\sum_{j=-a}^{+\infty} g_j u^j \right) := \sum_{j=-a}^{r-1} g_j u^j. \tag{15}$$

Then, multiplying the terms of the recurrence by z^n and summing yields

$$F(z,u) = 1 + zP(u)F(z,u) - z\{u^{<0}\}\left(P(u)F(z,u)\right), \tag{16}$$

where $\{u^{<0}\}$ is to be understood as applied to the u-expansion of $F(z,u)$ in (13). The relation (16) is the *fundamental functional equation* defining meanders. It reads as follows: "A path is either the empty path or it consists of a step ($zP(u)$ describes the possibilities) added to a path except that the steps that would take the walk below level 0 (the operator $\{u^{<0}\}$) are to be taken out". Now, P involves only a finite number of negative powers, so that

$$F(z,u)(1 - zP(u)) = 1 - z \sum_{k=0}^{c-1} r_k(u) F_k(z), \tag{17}$$

for some Laurent polynomials $r_k(u)$ that are immediately computable from P via (16):

$$r_k(u) := \{u^{<0}\}\left(P(u)u^k\right) \equiv \sum_{j=-c}^{-k-1} p_j u^{j+k}. \tag{18}$$

Theorem 2. *For a simple set of steps, the BGF of meanders (with z marking size and u marking final altitude) relative to a simple set of path $\mathcal{S}$ is algebraic. It is given in terms of the small and large branches of the characteristic curve of $\mathcal{S}$ by*

$$F(z,u) = \frac{\prod_{j=1}^{c}(u-u_j(z))}{u^c(1-zP(u))} = -\frac{1}{p_d z}\prod_{\ell=1}^{d}\frac{1}{(u-v_\ell(z))}. \tag{19}$$

In particular the GF of excursions, $E(z) = F(z,0)$, satisfies

$$E(z) = \frac{(-1)^{c-1}}{p_{-c}z}\prod_{j=1}^{c} u_j(z) = \frac{(-1)^{d-1}}{p_d z}\prod_{\ell=1}^{d}\frac{1}{v_\ell(z)}. \tag{20}$$

Proof. The point is that the fundamental equation in its form (17) looks grossly underdetermined as it involves $(c+1)$ unknown functions; to wit, the bivariate $F(z,u)$ and the univariate $\{F_k(z)\}_{k=0}^{c-1}$. The main idea of a method known as the "*kernel method*" (see also historical notes below) consists in binding z and u in such a way that the left hand side vanishes.

Indeed, substitute in (17) any small branch of the characteristic equation. Take $|z| < \frac{1}{P(1)}$ and restrict z to a small neighbourhood of the origin in such a way that: (i) all the small branches are distinct; (ii) all the small branches satisfy $|u_j(z)| < 1$. Then the substitution is analytically legitimate and, taking all small branches into account, it provides a system of c equations in the unknown functions $F_0, \ldots, F_{c-1}$:

$$\left\{\begin{array}{rcl} u_1^c - z\displaystyle\sum_{k=0}^{c-1} u_1^c r_k(u_1)F_k & = & 0 \\ \vdots & & \\ u_c^c - z\displaystyle\sum_{k=0}^{c-1} u_c^c r_k(u_c)F_k & = & 0. \end{array}\right. \tag{21}$$

This system is nonsingular for the reason that its determinant is a variant of the Vandermonde determinant and the small branches are clearly all distinct. This observation is enough to justify that each of the F_k is an algebraic function expressible rationally in terms of the algebraic branches u_j.

Instead of pursuing in the direction of determinantal calculations, we make use here of a cute observation of Mireille Bousquet-Mélou (introduced in [13] and employed in the parallel paper [4]). The quantity

$$N(z,u) := u^c - z\sum_{k=0}^{c-1} u^c r_k(u)F_k \tag{22}$$

is by (21) a polynomial in u whose roots are precisely all the u_j. The leading monomial of this polynomial is u^c, so that the polynomial factorizes as

$$N(z,u) = \prod_{j=1}^{c}(u-u_j(z)). \tag{23}$$

Then, the constant term is at the same time the product $(-1)^c u_1 \cdots u_c$ and the quantity $-zp_{-c}F_0$, as is apparent from the definition (22) and the form (18) of the coefficients. The form of F_0 follows.

Finally, the result for the BGF $F(z,u)$ derives from (17) made entire,

$$F(z,u) = \frac{N(z,u)}{u^c(1-zP(u))},$$

and from the factorization (23). □

An immediate corollary of Theorems 1 and 2 is the generating function of all paths and meanders irrespective of their final altitude.

Corollary 1. *The generating functions of all paths and all meanders are*

$$\begin{aligned} W(z) \equiv W(z,1) &= \tfrac{1}{1-zP(1)}, \\ M(z) \equiv F(z,1) &= \tfrac{1}{1-zP(1)} \prod_{j=1}^{c}(1-u_j(z)) = -\tfrac{1}{p_d z} \prod_{\ell=1}^{d} \tfrac{1}{1-v_\ell(z)}. \end{aligned}$$

A somewhat deeper consequence is a direct relation between the GF's of excursions and bridges that obtains by comparing Equations (9) and (20).

Corollary 2. *The generating functions of bridges (B) and excursions (E) are related by*

$$\begin{aligned} B(z) &= 1 + z\frac{d}{dz}(\log E(z)) = 1 + z\frac{E'(z)}{E(z)}, \\ E(z) &= \exp\left(\int_0^z (B(t)-1)\,\frac{dt}{t}\right). \end{aligned}$$

In the same vein, consider paths whose intermediate steps may be negative, but with a final altitude that is ≥ 0. Their BGF is

$$W^+(z,u) := \sum_{k=0}^{\infty} W_k(z) u^k.$$

Then, comparison of the forms involving large branches for W_k and $F(z,u)$ and a trite calculation shows that

$$\begin{aligned} W^+(z,u) &= 1 + z\frac{d}{dz}\left(\log F(z,u)\right), \\ F(z,u) &= \exp\left(\int_0^z (W^+(t,u)-1)\,\frac{dt}{t}\right). \end{aligned}$$

Finally, with $F_k(z)$ being the generating function of meanders that end at altitude k, one has $F_k(z) = [u^k]F(z,u)$. Since $F(z,u)$ is a rational function of u with a simple product expression in terms of the large branches, its expansion with respect to u is easily accessible via a partial fraction decomposition, and one finds:

Corollary 3. *The generating function of meanders terminating at altitude k is*

$$F_k(z) = \frac{1}{p_d z} \sum_{\ell=1}^{d} \xi_\ell v_\ell^{-k-1}, \qquad \xi_\ell := \prod_{j\neq\ell} \frac{1}{v_j - v_\ell}.$$

Some of these relations admit of combinatorial interpretations succinctly discussed in Section 4.1.

EXAMPLE 2. *Ballot problem, Dyck paths, and Motzkin paths.* These are the most famous problems in the area, and they are closely related to Example 1. The ballot problem asks for the probability, in a two candidate election between A and B that eventually results in a tie, of A dominating B throughout the poll. Recording the difference between the scores of A and B as time evolves, we model the problem as the counting of excursions associated with $\mathcal{S} = \{-1, +1\}$. The characteristic curve is the one examined in Example 1 in connection with central binomial coefficients and the GF of excursions is

$$E^{\mathcal{S}}(z) = \frac{1 - \sqrt{1 - 4z^2}}{2z^2} = \sum_{n \geq 0} \frac{1}{n+1} \binom{2n}{n} z^{2n},$$

where the coefficients $\frac{1}{n+1}\binom{2n}{n}$ are the Catalan numbers (*EIS* **A000108**). For $\mathcal{T} = \{-1, 0, +1\}$, one finds similarly

$$E^{\mathcal{T}}(z) = \frac{1 - z - \sqrt{1 - 2z - 3z^2}}{2z^2} = \sum_{n \geq 0} E_n^{\mathcal{T}} z^n,$$

where the coefficients are the Motzkin numbers (*EIS* **A001006**). □

EXAMPLE 3. *Łukasiewicz paths and tree codes.* Consider generally a finite set Ω that contains -1 as single negative value. The corresponding paths are known as Łukasiewicz paths. Set $\phi(u) := uP(u)$, which is a polynomial. There is only one small branch satisfying

$$u_1(z) = z\phi(u_1(z)), \tag{24}$$

and the GF of excursions is $\frac{1}{zp_{-1}} u_1(z)$. Łukasiewicz paths of type Ω encode trees whose node degrees are constrained to lie in $1 + \Omega$, this by virtue of a well-known correspondence [52, Chap. 11]. (Traverse the tree in preorder and output a step of $d - 1$ when a node of outdegree d is encountered.) In this way, it is seen that Equation (24) gives the GF of trees counted according to the number of their nodes, an otherwise classical result [55]. By Lagrange inversion, the number of trees comprised of n nodes is

$$T_n = \frac{1}{n} [w^{n-1}] \phi(w)^n,$$

where ϕ can be directly interpreted as the characteristic polynomial of the allowed node (out)degrees. □

EXAMPLE 4. *Walks with steps in* $\{-2, -1, 0, +1, +2\}$. This is our first example involving inherently more than one branch. The characteristic equation is

$$u^2 - z(1 + u + u^2 + u^3 + u^4) = 0.$$

The two small branches are conjugate and given by

$$\begin{aligned} u_1(z) &= +z^{1/2} + \tfrac{1}{2}z + \tfrac{5}{8}z^{3/2} + z^2 + \tfrac{231}{128}z^{5/2} + 3z^6 + \cdots \\ u_2(z) &= -z^{1/2} + \tfrac{1}{2}z - \tfrac{5}{8}z^{3/2} + z^2 - \tfrac{231}{128}z^{5/2} + 3z^6 + \cdots . \end{aligned}$$

Then, by (20), the first few terms of $E(z)$ are easily determined as

$$E(z) = -\frac{u_1(z)\, u_2(z)}{z} = 1 + z + 3z^2 + 9z^3 + 32z^4 + 120z^5 + 473z^6 + 1925z^7 + \cdots .$$

Similarly, for meanders, one has

$$M(z) = \frac{(1-u_1(z))(1-u_2(z))}{1-5z} = 1 + 3z + 12z^2 + 51z^3 + 226z^4 + 1025z^5 + \cdots .$$

It is then a natural question to ask for an equation satisfied directly by $E(z)$ or $F(z,1)$. Regarding excursions, an equation may be obtained by elimination of u_1, u_2 from the system

$$zE + u_1u_2 = 0, \quad u_1^2 - z(1 + u_1^2 + u_1^3 + u_1^4) = 0, \quad u_2^2 - z(1 + u_2u_2^2 + u_2^3 + u_2^4) = 0.$$

Either resultants or Gröbner bases do the job. For instance, resultants give a polynomial equation of degree 12 satisfied by $E(z)$. The polynomial factorizes (this is expected as we did not impose conditions like $u_1 \neq u_2$ in the process). Eventually, it is found that $E(z)$ satisfies a polynomial equation of degree 4:

$$z^4y^4 - z^2(1+z)y^3 + z(2+z)y^2 - (1+z)y + 1 = 0. \tag{25}$$

We shall examine shortly a much better way to perform such computations. □

EXAMPLE 5. *Duchon's clubs and underdiagonal paths.* The following problem[4] was considered by Duchon [22] (under a different formulation): *A club opens in the evening and closes in the morning. People arrive by pairs and leave in threesomes. What is the possible number of scenarios from dusk to dawn as seen from the club's entry?* For instance, an event may be $+2$ (two enter), $+2$ (two more enter), -3 (three leave), $+2$ (two, again arrive), -3 (and the club closes). Naturally the population inside the club is never negative and a business night starts with the empty club and ends with the empty club. The generalized problem then calls for the number of excursions with step set $\{-c, d\}$ (where Duchon's case is $\widehat{\mathcal{S}} = \{-3, +2\}$ or, equivalently by time reversal, $\mathcal{S} = \{-2, +3\}$). We assume here without loss of generality that c and d are coprime integers, so that the system of paths is reduced.

The characteristic polynomial is $P(u) = u^{-c} + u^d$ and the kernel equation is equivalent to

$$u^c = z(1 + u^e) \quad \text{with} \quad e = c + d.$$

Thus, the period is $e = c + d$ and the horizontal axis is only touched at places that are a multiple of e. Set $z = t^c$, where t is a local uniformizing parameter at 0. Then, the quantity $y(t) := u_1(t^c)$ satisfies the equation $y = t(1 + y^e)^{1/c}$, which is Lagrangean. By Lagrange inversion [42], one finds

$$y(t) = \sum_{n\geq 1} \frac{1}{n} \binom{n/c}{(n-1)/e} t^n. \tag{26}$$

(By convention, $\binom{a}{b} = 0$ if b is nonintegral.) Let ω be a primitive cth root of unity; then all the branches admit an expansion similar to $y(z)$. Indeed, by conjugacy, one has

$$u_{j+1}(t^c) = y(\omega^j t) = \sum_{n\geq 1} y_n \omega^{nj} t^n,$$

[4] After this paper had been submitted, Christian Krattenthaler pointed us to Ref. [68] by Masako Sato, dating from 1989. In that paper, Sato derives directly our equation (27) by matrix generating function methods and provides valuable additional results regarding underdiagonal paths in a strip.

where $y_n = [t^n]y(t)$ is given by (26). Then, the number of excursions is a convolution:

$$(-1)^{c-1}E_n = \sum_{n_1+\cdots+n_c=c(n+1)} y_{n_1}y_{n_2}\cdots y_{n_c}\omega^{0n_1+1n_2+\cdots+(c-1)n_c}.$$

It can be checked that E_n is automatically zero unless $n \equiv 0 \pmod{e}$ (see also the discussion on periodicities in Section 3.3 below). In summary, taking ω any primitive cth root of unity, and setting $n_j = 1 + e\nu_j$, $n = e\nu$, we find

(27)
$$E_{e\nu} = \sum_{\nu_1+\cdots\nu_c=c\nu} \frac{1}{1+\nu_1 e}\binom{(1+\nu_1 e)/c}{\nu_1}\cdots\frac{1}{1+\nu_c e}\binom{(1+\nu_c e)/c}{\nu_c}\omega^{0\nu_1+1\nu_2+(c-1)\nu_c}.$$

In particular, for $c = 1$, no summation is needed and

$$\frac{1}{1+ne}\binom{1+ne}{n}$$

gives the number of excursions of length n and type $\{-1, e-1\}$, which is also the number of e-ary trees having n internal nodes (Example 3). If $c = 2$ the formula (27) yields a single convolution. For $\mathcal{S} = \{-2, 3\}$, the result is

$$E_{5n} = \sum_{\nu=0}^{2n} \frac{(-1)^\nu}{1+5\nu}\binom{(1+5\nu)/2}{\nu}\frac{1}{1+5(2n-\nu)}\binom{(1+5(2n-\nu))/2}{2n-\nu},$$

to be compared to

$$E_{5n} = \sum_{i=0}^{n} \frac{1}{5n+i+1}\binom{5n+1}{n-i}\binom{5n+2i}{i}, \tag{28}$$

which Duchon obtained from quite specific series manipulations. In general if the jump in the negative direction is $-c$, formula (27) is a $(c-1)$-fold convolution of binomial coefficients.

Duchon's clubs can also be interpreted as *underdiagonal paths*. Consider paths in the $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ lattice whose allowed steps are of type either *East* (horizontal) or *North* (vertical), with a straight line barrier Δ. It is assumed that Δ passes through the origin and has a rational slope, $\frac{p}{q} \leq 1$. The number of ways $N_{m,n}$ of reaching point (m, n) by North and East steps then satisfies a recurrence of the same type as Pascal's triangle but with boundary conditions. For instance, the case of slope 1 gives rise to the original formulation [54] of the *ballot problem* (Example 2).

If one measures at each step of a path the vertical distance to Δ, then, this distance can only evolve by $+\frac{p}{q}$ for a horizontal step and -1 for a vertical step. Thus, up to rescaling, such an underdiagonal path is equivalent to a Duchon path of type $\{-q, +p\}$. The numbers $N_{m,n}$ are then amenable to the analysis of the paper since their determination is equivalent to counting meanders and excursions. For instance, here is a table of values for slope $\frac{2}{3}$:

									377	1144
								136	377	767
						23	66	136	241	390
					9	23	43	70	105	149
			2	5	9	14	20	27	35	44
		1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1	1

The sequence of numbers in this array that correspond to the number of ways of touching the boundary line is (*EIS* **A060941**)

$$1,\ 2,\ 23,\ 377,\ 7229,\ 151491,\ 3361598,\ 77635093,\ 1846620581,\ \cdots$$

which precisely coincides with the sequence of Duchon numbers, $\{E_{5n}\}_{n\geq 0}$, in (28).

Related enumerative results have been obtained by Durand [23] in the context of the "klam" recurrence that arises in complexity theory. Mohanty [57, p. 22] even quotes results of Takács relative to underdiagonal paths under a line of arbitrary slope. □

As the last example shows, the decomposability afforded by the kernel method provides a grasp on the structural complexity of summatory formulæ expressing the number of walks, excursions, etc. Following Comtet [15, p. 216], we observe that the "rank" (defined as the minimal number of summations) of the excursion formula in the general case is at most $c(q-1)-1$ if $P(u)$ comprises q terms. For instance, Catalan numbers ($(c,q)=(1,2)$) are of rank 0, Motzkin numbers ($(c,q)=(1,3)$) and the Duchon numbers E_n of (28) (having $(c,q)=(2,2)$) are of rank 1, etc.

Some origins of the kernel method. What we named here the "kernel method" has been part of the folklore of combinatorialists for some time. Earlier references usually deal with the case of a functional equation of the form

$$K(z,u)F(z,u) = A(z,u) + B(z,u)G(z)$$

(with F, G the unknown functions), when there is only one small branch, u_1, such that $K(z,u_1(z))=0$. In that case, a single substitution does the job, and $G(z) = -A(z,u_1)/B(z,u_1)$. One clear source of this is the exercise section of the first edition (in 1968) of Knuth's book [46]: the detailed solution to Exercise 2.2.1–4 (see [46, p.536–537] and also Ex. 2.2.1.11) presents a "new method for solving the ballot problem", for which the characteristic equation is quadratic. See also Odlyzko's splendid survey [61, Sec. 15.4] for a discussion of a pebbling game and Prodinger's recent note [64] for an original application to a quadratic problem arising from queueing theory.

The kernel method in its more general version was used recently in a few unpublished works by the authors, including a systematization to directed lattice paths by Banderier in his memoir [2]. Independent combinatorial developments at the end of the last century are due to Bousquet-Mélou and Petkošek whose recent paper offers a penetrating perspective on the subject of multidimensional walks, recurrences, and kernels [13]. In fact, as indicated earlier, a remark of Bousquet-Mélou has been used to simplify our proof of Theorem 2 (see also [4] for another application).

That probabilists had known a lot since the early 1950's regarding related questions is manifest upon reading Chapter XII of Fellers' book [28]. It appears that our presentation parallels in some ways what is obtained by the famous Wiener-Hopf

approach: refer in particular to the example on bounded arithmetic distributions in [28, p. 407–408]. Such techniques prove in turn valuable in the theory of queueing systems: see, e.g., Robert's book [66] for an account. The synthesis by Fayolle, Iasnogorodski, and Malyshev [26] exposes the deep ramifications of the theory in the harder case of walks in a quarter plane *not* satisfying directedness restrictiction (thus, a "pure" 2-dimensional problem), but their methods only apply to nearest-neighbour moves. The book [26] itself draws some of its inspiration from the early paper [25] where a sophisticated use of the kernel method already plays a central rôle (amongst other techniques like conjugacy and Riemann–Hilbert problems); see also the references to Flatto and Malyshev's works in [61, p. 1208] and the historical comments in [26, p. VII-XI].

2.3. **Computational aspects.** We discuss now a way to determine directly the equations satisfied by the algebraic functions encountered so far. Because of Corollary 2, we know that bridges and excursions are tightly coupled, and the case of excursions will be detailed here.

It is assumed that the characteristic polynomial $P(u)$ is fixed. Then, what is needed in view of Theorem 2 is the equation satisfied by the product $Y = u_1 \cdots u_c$ of c distinct roots of a polynomial of degree $c+d$. As roots are in general "indistinguishable", we expect a polynomial of degree $\binom{c+d}{c}$ to cancel Y.

Take a polynomial $Q(u)$ of degree e in $\mathbb{C}(z)[u]$ normalized by $Q(0) = 1$ and assume it has distinct roots $u_1, \ldots, u_e$. For us, $e = c + d$, and

$$Q(u) = -\frac{1}{z p_{-c}} \left(u^c - z u^c P(u)\right),$$

yet another reformulation of the kernel. We first develop the computational process when $c = 2$, so that the equation for $Y = u_1 u_2$ with u_1, u_2 two distinct roots of Q is sought. Write α, α' for generic roots of Q. Then, since $Q(0) = 1$, one has

$$Q(u) = \prod_{\alpha} \left(1 - \frac{u}{\alpha}\right),$$

while what we need to determine is

$$R(u) = \prod_{\{\alpha,\alpha'\}} \left(1 - \frac{u}{\alpha\alpha'}\right).$$

(A sum or product over $\{\alpha, \alpha'\}$ means a sum or product over all unordered pairs of *distinct* elements.) Now, take logarithms. One has

$$\begin{aligned} \log\left(\frac{1}{Q(u)}\right) &= \sum_{n\geq 1} S_n \frac{u^n}{n} \quad \text{with} \quad S_n := \sum_{\alpha} \frac{1}{\alpha^n} \\ \log\left(\frac{1}{R(u)}\right) &= \sum_{n\geq 1} S_n^{(2)} \frac{u^n}{n} \quad \text{with} \quad S_n^{(2)} := \sum_{\{\alpha,\alpha'\}} \frac{1}{\alpha^n \alpha'^n}. \end{aligned}$$

Then, a simple combinatorial reasoning shows that

$$\sum_{\{\alpha,\alpha'\}} \frac{1}{\alpha^n \alpha'^n} = \frac{1}{2} \sum_{(\alpha,\alpha')} \frac{1}{\alpha^n \alpha'^n} - \frac{1}{2} \sum_{\alpha} \frac{1}{\alpha^{2n}},$$

so that

$$S_n^{(2)} = \frac{1}{2} S_n^2 - \frac{1}{2} S_{2n}. \tag{29}$$

The degree of R is $\delta := \binom{e}{2}$ *a priori*, and R can be recovered from the formula ("I am always the exponential of my logarithm!")

$$R(u) := \{u^{\leq \delta}\} \left[\exp\left(-\sum_{n=1}^{\delta} \frac{1}{2}(S_n^2 - S_{2n}) \frac{u^n}{n} \right) \right], \tag{30}$$

where $\{u^{\leq \delta}\} f$ means the truncation of the series expansion of f with all terms of degree $\leq \delta$ included (see the analogous notation (15)).

The general formulæ for $c > 2$ are easily found from the usual relations between elementary and power sum symmetric functions. Set $x_j = \alpha_j^{-n}$. What is sought is plainly a formula expressing the sum Φ_c of all products $x_{j_1} \cdots x_{j_c}$ taken over all distinct *subsets* $\{j_1, \ldots, j_c\}$ when the power sums $s_k := \sum_j x_j^k$ are known. Then, one has (by exponentials of logarithms again)

$$\Phi_c = [t^c] \prod_j (1 + t x_j) = [t^c] \exp\left(\sum_{k \geq 1} (-1)^{k-1} s_k \frac{t^k}{k} \right). \tag{31}$$

Thus, Φ_c is a computable polynomial in $s_1, \ldots, s_c$, obtained from extracting the coefficient $[t^c]$ in the exponential form of (31) that we write as $\Phi_c(s_1, \ldots, s_c)$. Define finally

$$S_n^{(c)} := \sum_{\{j_1, \cdots, j_c\}} u_{j_1}^{-n} \cdots u_{j_c}^{-n},$$

the sum being on all subsets of c elements. Then we have

$$S_n^{(c)} = \Phi_c(S_n, S_{2n}, \cdots, S_{cn}).$$

For instance, the formulæ analogous to (29) for $c = 3, 4$ are found to be

$$\begin{aligned} S_n^{(3)} &= \tfrac{1}{6} S_n^3 - \tfrac{1}{2} S_n S_{2n} + \tfrac{1}{3} S_{3n} \\ S_n^{(4)} &= \tfrac{1}{24} S_n^4 - \tfrac{1}{4} S_n^2 S_{2n} + \tfrac{1}{3} S_n S_{3n} + \tfrac{1}{8} S_{2n}^2 - \tfrac{1}{4} S_{4n}. \end{aligned} \tag{32}$$

These considerations give rise to a simple algorithm for computing the polynomial cancelled by the product of all small branches.

Platypus Algorithm. *Computes the polynomial $R(u) \in \mathbb{C}(z)[u]$ of degree $\delta = \binom{e}{c}$ such that $R(Y) = 0$, where $Y = u_1 \cdots u_c = (-1)^{c-1} z p_c E(z)$ is the product of all small branches of the characteristic curve.* The input is the characteristic polynomial of steps, $P(u)$.

1. Set up the symbolic formulæ of type (29) and (32) appropriate for the given value of c. To this effect, perform the symbolic expansion of (31) with $\Phi_c(s_1, \ldots, s_c)$ denoting the coefficient of t^c in the exponential form.
2. Take the normalized kernel $Q(u) = (-z p_c)^{-1}(u^c - z u^c P(u))$. Set $\delta = \binom{e}{c}$ and determine the expansion
$$\log\left(\frac{1}{Q(u)} \right) = \sum_{n=1}^{c\delta} S_n \frac{u^n}{n} + O(u^{c\delta+1}).$$
3. Recover $R(u)$ from the truncated series
$$R(u) := \{u^{\leq \delta}\} \left[\exp\left(-\sum_{n=1}^{\delta} \Phi_c(S_n, S_{2n}, \ldots, S_{cn}) \frac{u^n}{n} \right) \right].$$

Half a dozen instructions in a symbolic manipulation language are sufficient to translate the algorithm. In contrast to Gröbner basis or resultant calculations, the process is efficient, whenever the degree of the result remains reasonable. For instance, we could successfully determine polynomials R of degree $45 = \binom{10}{2}$ in a matter of seconds on a machine with a 500MHz clock.

On coefficients of algebraic functions. As it is well known [14], any algebraic function $f(z)$ satisfies a linear differential equation $L(f) = 0$ with coefficients that are rational functions of the variable. This in turn translates into a linear recurrence with polynomial coefficients in n for the quantities $[z^n]f$. Thus, the coefficient of index n of any algebraic function is computable in a number of operations that is linear in n. (The procedure is implemented in Salvy and Zimmermann's `Gfun` package [67].) This remark applies to all the generating functions considered in this paper. For instance, the excursion generating function $E(z)$ corresponding to the set of jumps $\{-2, -1, 0, +1, +2\}$ (Example 4) satisfies an inhomogeneous differential equation of order 3

$$z^3(5z+4)(5z+1)(z-1)^2(5z-1)^2\frac{d^3E}{dz^3} + \cdots + (-100z^2+56z-4) = 0, \tag{33}$$

and its coefficients can be obtained from a recurrence of order 6,

$$2(n+7)(n+8)(2n+13)E_{n+6} + \cdots + 625(n+1)(n+2)(n+3)E_n = 0. \tag{34}$$

3. Singular structures

We now examine paths, bridges, meanders and excursions under the angle of asymptotics. As is well known, the asymptotic behaviour of counts is closely related to the singular structure of the corresponding generating functions [34, 61]. Thanks to the factorizations afforded by the kernel method, the singular forms of intervening generating functions become manageable. This part of the analysis makes use of global properties of branches followed by local analysis in the vicinity of a quantity called the "structural radius" ρ.

Lemma 1. *Let $P(u)$ be the polynomial associated to the steps of a simple walk. Then, there exists a unique number τ, called the* structural constant*, such that*

$$P'(\tau) = 0, \quad \tau > 0.$$

The structural radius *is by definition the quantity*

$$\rho := \frac{1}{P(\tau)}.$$

Proof. Differentiating twice P as given in (3), we see that $P''(x) > 0$ for all $x > 0$. Thus, the real function $x \mapsto P(x)$ is strictly convex. Since it satisfies $P(0) = P(+\infty) = +\infty$, it must have a unique positive minimum attained at some τ, and $P'(\tau) = 0$. □

Structural constants *a priori* live in a field of degree $e := c + d$ over the base field of weights. However, for symmetric walks ($P(u) = P(u^{-1})$), they automatically reduce to the value $\tau = 1$ and ρ becomes automatically a member of the field of coefficients of P.

In Section 2, we have defined the principal branch $u_1(z)$ near the origin by means of its expansion at 0. We show here that this branch satisfies a useful domination property for $0 \le z \le \rho$. Cf. Figure 4 for an illustration.

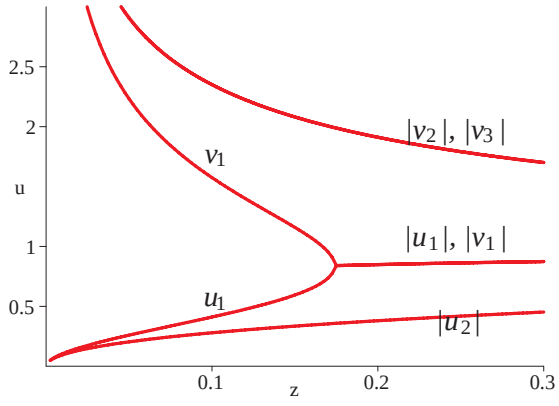


FIGURE 4. A rendering of the modulus of the five branches of the characteristic curve in the example of Figure 3 illustrates the domination properties of the principal small and large branches.

Lemma 2. *For an aperiodic walk, the principal small branch $u_1(z)$ is analytic on the open interval $z \in (0, \rho)$. It dominates strictly in modulus all the other small branches, $u_2(z), \dots, u_c(z)$, throughout the half-closed interval $z \in (0, \rho]$.*

Proof. By the discussion of Lemma 1, the function $1/P(z)$ is continuously increasing for $z \in [0, \tau]$. Hence the equation (in u) $z = 1/P(u)$ admits a unique positive solution, say $u^+(z)$, that is less than τ when $z \in [0, \rho]$. This positive solution $u^+(z)$ must coincide with the branch u_1 at 0^+ (since the expansions at 0^+ are the same). Also, the analytic version of the implicit function theorem guarantees that the positive solution $u^+(z)$ remains analytic all along $z \in (0, \rho)$, so that the principal small branch u_1 and the positive solution u^+ must coincide throughout this interval. Consequently, u_1 (originally only defined near 0^+) increases from 0 to τ as ρ increases from 0 to ρ.

Next, a general fact about polynomials with positive coefficients enters the game: if $P(u)$ is aperiodic, then one has for positive r

$$|P(re^{i\theta})| < P(r) \quad \text{for all } \theta \not\equiv 0 \pmod{2\pi}, \tag{35}$$

as seen from the strong form of the triangle inequality. Fix $z = x$, with x real positive and $x < \rho$, and let w be an arbitrary solution of the kernel equation $1 - xP(w) = 0$ that is at most τ in modulus and *not* equal to $u_1(x)$ (i.e., not real and positive). Then, one has by the strict inequality in (35)

$$x = \frac{1}{P(u_1(x))} = \frac{1}{P(w)} > \frac{1}{P(|w|)},$$

which implies $|w| < u_1(x)$ since $1/P$ is increasing in the region considered, $[0, \tau]$. Thus, near 0^+ and since the nonprincipal small branches $u_2, \dots, u_c$ are majorized by τ in modulus (they tend to 0), they must satisfy $|u_j(x)| < u_1(x)$. Additionally, the domination property cannot cease to hold on $(0, \rho)$: by continuity of the modulus of any branch, this would imply that $u_1(x)$ itself reaches the value τ for some $x < \rho$, yielding a clear contradiction. Domination must finally continue to hold at ρ, since otherwise, there would be a contradiction with the strong triangle inequality (35). □

Stronger domination properties are in fact derivable from similar uses of the strong triangle inequality, under the aperiodicity condition (see also [3] for details). For $|z| \leq \rho$, one has: $|u_j(z)| < u_1(|z|)$ for $j = 2, \ldots, c$; also, $|u_1(z)| < |v_1(z)|$ safe at $z = \rho$. Simply put, the principal small branch u_1 is the "largest" of all the small branches.

In Section 4, it will also prove handy to have available the corresponding properties of large branches. For instance, the principal large branch, v_1, is in a similar sense the smallest of all large branches. Generally, the domination properties of large branches are counterparts of those of small branches, as can be seen by mimicking the arguments. Alternatively, one can introduce duality: If $P(u)$ is a Laurent polynomial, then $\widetilde{P}(u) = P(u^{-1})$ is called its dual. It is then easy to see that the small and large branches, $\widetilde{u}_j$ and $\widetilde{v}_\ell$ of the dual are respectively the inverses of the large and small branches of the primal: $\widetilde{u}_j v_j = 1$ and $\widetilde{v}_\ell u_\ell = 1$. Duality thus exchanges small and large branches. (Combinatorially, duality may be realized either as a symmetry along the horizontal axis applied to steps, or by the time-reversal transformation that changes a path into another path obtained by reading steps backwards.)

3.1. **Bridges and excursions.** We first address the important problem of estimating the numbers of bridges and excursions. The discussion makes use of the assumption that the walk is reduced and aperiodic.

Theorem 3. *Consider a simple system of walks that is aperiodic. Let τ be the structural constant determined by $P'(\tau) = 0$, $\tau > 0$. The number of bridges of size n admits a complete asymptotic expansion*

$$B_n \sim \beta_0 \frac{P(\tau)^n}{\sqrt{2\pi n}} \left(1 + \frac{a_1}{n} + \frac{a_2}{n^2} + \cdots\right), \quad \beta_0 = \frac{1}{\tau}\sqrt{\frac{P(\tau)}{P''(\tau)}}. \tag{36}$$

The number of excursions of size n satisfies

$$E_n \sim \epsilon_0 \frac{P(\tau)^n}{2\sqrt{\pi n^3}} \left(1 + \frac{b_1}{n} + \frac{b_2}{n^2} + \cdots\right), \tag{37}$$

where (the u_j are the small branches, with u_1 the principal branch)

$$\epsilon_0 = \frac{(-1)^{c-1}}{p_{-c}} \sqrt{\frac{2P(\tau)^3}{P''(\tau)}} Y_1(\rho), \quad Y_1(z) := \prod_{j=2}^{c} u_j(z), \quad \rho := \frac{1}{P(\tau)}. \tag{38}$$

By Lemma 2, the constant $Y_1(\rho)$ is equivalently characterized as

$$Y_1(\rho) = \prod_{|\upsilon|<\tau,\ P(\upsilon)=\rho^{-1}} \upsilon.$$

Proof. The result for bridges is known as it is equivalent to the local limit theorem for sums of discrete random variables [40, Chapter 9], of which the first proof goes back to Laplace[5] in [51]. For completeness, we briefly sketch the argument here.

[5]Quite remarkably, in his *Théorie analytique des probabilités*, in 1812. Laplace expresses the problem as a Cauchy coefficient formula presented by its Fourier series counterpart (analytic functions are not yet invented by Cauchy!) and proceeds with a saddle point argument expressed as an application of the "Laplace method" that was specifically developed for that occasion (saddle point integrals will only emerge half-a-century later!).

Start from the fact that the number of bridges of length n is $[u^0]P(u)^n$. By Cauchy's coefficient formula, one has

$$B_n = \frac{1}{2i\pi}\int_\gamma P(u)^n \frac{du}{u},$$

where the contour γ is any positively oriented loop about the origin. The positive real point τ is a simple saddle point of $P(u)$ (hence of $P(u)^n$), so that the choice of the circle $|u| = \tau$ as integration contour suggests itself by the saddle-point method [16]. By the aperiodicity condition, $P(u)$ is uniquely maximal in modulus along the contour at $u = \tau$; see (4). Therefore, the following saddle-point approximations are justified:

$$\begin{aligned} B_n &= \frac{1}{2i\pi}\int_{|u|=\tau} P(u)^n \frac{du}{u} \\ &\sim \frac{1}{2i\pi}\int_{\tau e^{-i\epsilon}}^{\tau e^{+i\epsilon}} \exp\left(n\left(\log P(\tau) + \frac{1}{2}\frac{P''(\tau)}{P(\tau)}(u-\tau)^2 + O((u-\tau)^3)\right)\right)\frac{du}{u} \\ &\sim \frac{P(\tau)^n}{2\pi\tau}\int_{-\infty}^{+\infty} e^{-nht^2/2}\,dt \;=\; \frac{P(\tau)^n}{\tau\sqrt{2\pi n h}}, \qquad h = \frac{P''(\tau)}{P(\tau)}. \end{aligned}$$

By the usual process, the contribution is first localized near τ, taking for instance $\epsilon = (\log n)/\sqrt{n}$, and local expansions are applied; then the contour is extended back to yield a complete Gaussian integral. This streamlined version of the method is then extended to a full asymptotic expansion in the usual way [43, p. 419], so that (36) results.

The saddle point method thus provides an easy access to the enumeration of bridges. This gives indirectly valuable information on the small branches that can be translated into the singular structure of the GF $B(z)$. First, the relation that determines the branches of the characteristic curve can be put under the form

$$z = \frac{1}{P(u)}. \tag{39}$$

This shows that a branch can become infinite only at $z = 0$; in fact the corresponding solutions give rise precisely to the large branches $v_1, \ldots, v_d$. By general principles (the inverse of an analytic function at a point where the derivative is nonzero is analytic), the relation (39) is invertible analytically in a the neighbourhood of any point v such that $P'(v) \neq 0$. Accordingly, a singularity (in the sense of analytic functions) *must* occur at any value ζ such that $P'(\zeta) = 0$.

At $u = \tau$, with τ the structural constant, one has $P'(\tau) = 0$ by construction, while $P''(\tau) > 0$. Then, the local form of (39), reads

$$z = \rho - \frac{1}{2}P''(\tau)(u-\tau)^2 + O((u-\tau)^3). \quad \rho := \frac{1}{P(\tau)}. \tag{40}$$

This is readily inverted, yielding two local solutions

$$u(z) = \tau \pm \sqrt{2\frac{P(\tau)}{P''(\tau)}}\sqrt{1 - z/\rho} + \cdots \qquad (z \to \rho^-). \tag{41}$$

In particular, the principal branch $u_1(z)$ has a square root singularity; it takes as value the structural constant τ at the place

$$\rho = \frac{1}{P(\tau)}.$$

and the $-\sqrt{}$ determination must be adopted in (41) since $u_1(z)$ increases as $z \to \rho^-$:

$$u_1(z) = \tau - \sqrt{2\frac{P(\tau)}{P''(\tau)}}\sqrt{1 - z/\rho} + \cdots \qquad (z \to \rho^-). \tag{42}$$

Next, for $z \neq 0$, all singularities of the solutions of (39), since they correspond to finite values of u, can only be finite branch points ζ with a local expansion of the form $a_0 + b_0(z - \zeta)^{1/r}$ for some ramification index $r > 1$. (This is easily seen directly by a suitable generalization of (40) and(41) upon taking into account the first nonzero derivative of $1/P$).

We can now confront the result of (42) with the the saddle point estimation (36), remembering that one has by (9)

$$B(z) = z\frac{d}{dz}\log Y(z), \qquad Y(z) := (u_1(z) \cdots u_c(z)).$$

First, $Y(z)$ that is analytic near 0 must remain analytic throughout the disk $|z| < \rho$, since otherwise $B(z)$ would be singular for some value inside the disk and this would contradict the asymptotic growth (36) that is of type $P(\tau)^n$ for B_n. Next, $Y(z)$ cannot have any (algebraic) singularity other than $z = \rho$ on the circle $|z| = \rho$, since, by singularity analysis[6], this would entail the presence of oscillating terms in the asymptotic expansion of B_n, again contradicting (36). Also, $Y(z)$ can only have a branch point of ramification index $r = 2$ at $z = \rho$, since otherwise some term of the form $n^{-1+1/r}$ would have been present in the expansion of B_n. Finally, the deflated product $Y_1(z) = u_2(z) \cdots u_c(z)$ must be analytic at ρ since otherwise, being capable only of having a branch point with ramification index 2, one would reach a contradiction regarding the leading coefficient of B_n (as checked from comparing (36) against the consequences of (42) on coefficients).

In other words, this sequence of indirect arguments shows the following[7]: *The product of all the nonprincipal small branches*

$$Y_1(z) = u_2(z) \cdots u_c(z) \tag{43}$$

is analytic at all points of the closed disk $|z| \leq \rho$.

It is now an easy matter to complete the estimate of the number of excursions by singularity analysis applied to (20) in Theorem 2. The unique dominant singularity of $E(z)$ must be at $z = \rho$ where the local expansion (42) gives

$$E(z) \sim E(\rho) - \epsilon_0\sqrt{1 - z/\rho}, \qquad \epsilon_0 = \frac{(-1)^{c-1}}{p_{-c}\rho}Y_1(\rho)\sqrt{2\frac{P(\tau)}{P''(\tau)}},$$

with Y_1 given by (43). A full expansion of $u_1(z)$ in powers of $(1 - z/\rho)^{1/2}$ being available, and $Y_1(z)$ being analytic on the whole of $|z| \leq \rho$, the proof of (37) is at last completed. □

EXAMPLE 6. *Asymptotics of tree codes.* The case of walks with only one type of descending step equal to -1 corresponds to tree codes, as discussed in Example 3.

[6]Singularity analysis [34, 61] allows us to transfer a singular element of the form $(1 - z/\alpha)^\kappa$ in the expansion of a function $f(z)$ at a singularity α into a corresponding asymptotic element of the form $\alpha^{-n}n^{-\kappa-1}/\Gamma(-\kappa)$ in the expansion of the coefficient $[z^n]f(z)$ at infinity. It is applicable unconditionally to algebraic functions.

[7]An alternative argument based on the refinement of domination relations evoked after the proof of Lemma 2 is possible; see Banderier's thesis [3] for details.

In this very special case, there is only one small branch, and the GF of excursions is $E(z) = u_1(z)/(p_{1-}z)$. For aperiodic walks, the result (37) of Theorem 3, or plainly the estimate (41), gives us

$$\begin{aligned} &\tau \ : \quad P'(\tau) = 0 \\ &E_n \sim \frac{1}{p_{-1}} \frac{1}{\sqrt{2\pi n^3}} \sqrt{\frac{P(\tau)^3}{P''(\tau)}}\, P(\tau)^n. \end{aligned} \tag{44}$$

In terms of trees, the principal branch $u_1(z)$ is precisely the GF of trees corresponding to the degree set $1 + \mathcal{S}$ with generating polynomial $\phi(u) := uP(u)$ and one has $T(z) = p_{-1} z E(z) = u_1(z)$. The estimate (44) then coincides with the well-known asymptotic estimate of the number T_n of trees of size n,

$$\begin{aligned} &\tau \ : \quad \phi(\tau) - \tau\phi'(\tau) = 0 \\ &T_n \sim \frac{1}{\sqrt{2\pi n^3}} \sqrt{\frac{\phi(\tau)}{\phi''(\tau)}} \left(\frac{\phi(\tau)}{\tau}\right)^n, \end{aligned} \tag{45}$$

which was first discovered by Meir and Moon [55]. □

As soon as $c > 1$, there are several small branches, and, in this case, the algebraic constant $Y_1(\rho)$ intervenes. Numerically, this constant can be determined easily as it only involves the product of the small solutions to the kernel equation taken at $z = \rho$. Algebraically, since $Y_1(\rho)$ is the product of $c-1$ solutions to an algebraic equation of degree $c + d$, it is an algebraic number of degree at most $\binom{c+d}{c-1}$ over $\mathbb{Q}(\rho) \equiv \mathbb{Q}(\tau)$ that is computable by the techniques of Section 2.3 (upon changing c to $c - 1$ in Platypus Algorithm). However, since τ is a double root of the kernel equation instantiated at $z = \rho$, further simplifications accrue. This explains that constants involving radicals are often to be observed when analysing problems of relatively low "complexity". The next example is typical of this state of affairs.

EXAMPLE 7. *Asymptotics of the* $\{-2, -1, 0, 1, 2\}$*-excursions.* The walk introduced in Example 4 is symmetric, and like for any symmetric walk system, the structural constant is equal to 1 while the structural radius is the rational number, $\rho = 1/P(1) = \frac{1}{5}$. The product of the nonprincipal small branches at ρ reduces to $u_2(\rho)$. This quantity is *a priori* one of the roots of an equation of degree 4 (Equation (25) instantiated at $z = \rho$), but since this equation has already $\tau = 1$ as a double root, the equation satisfied by $u_2(\rho)$ is in fact of degree 2 (it is $u^2 + 3u + 1 = 0$) so that

$$u_2(\rho) = -\frac{3}{2} + \frac{1}{2}\sqrt{5},$$

and this quantity is precisely $Y_1(\rho)$ of (38). Thus, we can conclude and get easily

$$E_n = \frac{5}{4}(3 - \sqrt{5}) \frac{5^n}{\sqrt{\pi n^3}} \left(1 + O(\frac{1}{n})\right).$$

The quality of the asymptotic approximation provided by the first term is 11% when $n = 10$ and 1.2% when $n = 100$, where the E_n are conveniently determined by (34). The estimate is also consistent with the nature of the singularity at $\rho = \frac{1}{5}$ of the differential equation (33). □

3.2. **Paths and meanders.** Now that the bulk of the work is done, asymptotic estimates of the basic counts of paths and meanders fall as a ripe fruit. The result for unconstrained paths is trivial, since the number of possibilities for size n is $P(1)^n$, a fact consistent with the simple pole of $W(z,1) = (1 - zP(1))^{-1}$. For meanders, three cases are to be distinguished depending upon the value of a quantity called the drift.

Definition 5. *Given a simple walk with characteristic polynomial $P(u)$, the* drift *is by definition the quantity*

$$\delta = P'(1).$$

In the unweighted case, the drift is thus the sum of all the possible values of the jumps, which constitutes an indicator of the "tendency" for the walk to go up or down. In the probabilistic case ($P(1) = 1$), the drift represents exactly the expected movement in the y-direction of any single step. For a symmetric walk, the drift is $\delta = 0$, while $\tau = 1$.

Theorem 4. *Consider a simple aperiodic walk. The number of paths of length n, $[z^n]W(z,1)$, is $P(1)^n$ exactly. Set*

$$\overline{Y}_1(z) := \prod_{j=2}^{c} (1 - u_j(z)).$$

The asymptotic number of meanders depends on the sign of the drift $\delta = P'(1)$ as follows:

$$\begin{aligned}
\delta = 0: \quad M_n &\sim \mu_0 \frac{P(1)^n}{\sqrt{\pi n}} \left(1 + \frac{c_1}{n} + \frac{c_2}{n^2} + \cdots\right) \\
&\mu_0 := \sqrt{2\frac{P(1)}{P''(1)}}\overline{Y}_1(\rho), \quad \rho = P(\tau)^{-1} = P(1)^{-1};
\end{aligned}$$

$$\begin{aligned}
\delta < 0: \quad M_n &\sim \mu_0^- \frac{P(\tau)^n}{2\sqrt{\pi n^3}} \left(1 + \frac{c_1^-}{n} + \frac{c_2^-}{n^2} + \cdots\right) \\
&\mu_0^- := -\sqrt{2\frac{P(\tau)^3}{P''(\tau)}} \frac{\overline{Y}_1(\rho)}{P(\tau) - P(1)}, \quad \rho = P(\tau)^{-1};
\end{aligned}$$

$$\begin{aligned}
\delta > 0: \quad M_n &\sim \mu_0^+ P(1)^n + \mu_0^- \frac{P(\tau)^n}{2\sqrt{\pi n^3}} \left(1 + \frac{c_1^+}{n} + \frac{c_2^+}{n^2} + \cdots\right) \\
&\mu_0^+ := (1 - u_1(\rho_1))\overline{Y}_1(\rho_1), \quad \rho_1 := P(1)^{-1}.
\end{aligned}$$

The formulæ have an intuitive meaning. In the case of a positive drift, a fraction close to μ_0^+ of all the (unconstrained) walks is a meander, in accordance for the walks to have a natural tendency to go up. For negative drift, most paths tend to go down and the proportion of meanders is exponentially small, roughly like $(P(\tau)/P(1))^n$. For zero drift, the proportion becomes as large as $1/\sqrt{n}$, while the walks tend to oscillate not too far from the horizontal axis.

Proof. The discussion is based on the formula of Corollary 1 rewritten as

$$M(z) = F(z,1) = \frac{1 - u_1(z)}{1 - zP(1)}\overline{Y}_1(z), \qquad \overline{Y}_1(z) := \prod_{j=2}^{c} (1 - u_j(z)).$$

It suffices to examine the position of the zeros and the dominant singularity of the numerator in relation to $1/P(1)$ that is always a zero of the denominator. By proof arguments similar to Lemma 2, the quantity $\overline{Y}_1(z)$, being a symmetric function of small branches each of which is dominated by u_1, must remain analytic throughout $|z| \leq \rho$.

In the case $\delta = 0$, one has $P'(1) = 0$, $\tau = 1$, and $\rho = 1/P(\tau) = 1/P(1)$. Thus, $(1 - u_1)$ contributes a term of the form $(1 - z/\rho)^{1/2}$ at $z = \rho$ while the denominator $(1 - zP(1))$ has a simple zero there. Globally, the singularity of $F(z, 1)$ is thus of type $1/\sqrt{\ }$, and the result follows.

For a negative drift, meaning $P'(1) < 0$, one must have $\tau > 1$, since $P'(u)$ increases from $-\infty$ to $+\infty$ when u ranges from 0^+ to $+\infty$. With $\rho = 1/P(\tau)$ (the structural radius) and $\rho_1 := 1/P(1)$, one then has $\rho_1 < \rho$. In this case, the prefactor $(1 - zP(1))^{-1}$ has a pole at ρ_1; this pole is however cancelled by a zero in the numerator induced by the numerator $(1 - u_1(z))$ (since $u_1(\rho_1) = 1$), so that ρ_1 is a removable singularity of $F(z, 1)$. Consequently, the dominant singularity of $F(z, 1)$ is at ρ, where $F(z, 1)$ is of the square-root type.

For a positive drift, one must have $\tau < 1$, so that the prefactor induces a pole at $\rho_1 := 1/P(1)$ before $\overline{Y}_1$ or $1 - u_1$ become singular. The argument concludes by "subtracting singularities", since the function,

$$F(z, 1) - \frac{\overline{Y}_1(\rho_1)(1 - u_1(\rho_1))}{1 - zP(1)}, \quad \rho_1 := \frac{1}{P(1)},$$

now has a dominant singularity of the square-root type at ρ. □

The earlier discussion about the algebraic character of asymptotic constants applies: quantities like $\overline{Y}_1(\rho_1)$ and $\overline{Y}_1(\rho)$ can be determined by adapting Platypus Algorithm of Section 2.3. Should the degrees of the algebraic numbers involved become fairly large, one can always resort to numerical analysis as the next example illustrates.

EXAMPLE 8. *Lucky periods in die casting.* In [63, p. 45], Pólya introduces the following problem: *"En jetant $2n$ dés à la fois, on peut obtenir différentes sommes de points de $2n$ à $12n$. Le cas le plus probable est celui de $7n$ points. Désignons par A_n le nombre de combinaisons où se produit cet événement."* Imagine that at each of n rounds two dice are cast and the score of the round is the sum of the two dice's values. Pólya thus considers the number of ways A_n (and probability $A_n/36^n$) of reaching the balanced score $7n$ at the end of a game of dice consisting of n rounds. Pólya proceeds by an integral representation (precisely of the type used in the proof of Theorem 1) from which he concludes that the GF $A(z)$ has the character of an algebraic function, but does not make the calculation explicit.

By centring around the mean score of a round, which equals 7, it is easily realized that the problem is equivalent to a walk whose characteristic polynomial is

$$P(u) = u^{-5}\left(1 + u + u^2 + u^3 + u^4 + u^5\right)^2.$$

Let B_n be the number of bridges. (The quantity B_n is exactly Pólya's A_n.) Here, $c = -5$, $d = +5$; also $\tau = 1$ as the walk is symmetric, and $\rho = 1/36$. The asymptotic number of bridges is simply

$$B_n \sim \frac{6 \cdot 36^n}{\sqrt{2^2 \cdot 3 \cdot 5 \cdot 7\, \pi n}},$$

which is nothing but an avatar of the local limit gaussian law.

Consider next the modification of Pólya's problem where we ask for the number of "lucky" games, in the sense that at any time t the score is at least $7t$. This is equivalent to finding the number of meanders. Excursions surface if we further impose the final score to be $7n$ exactly. We have $\tau = 1$ and $\rho = \frac{1}{36}$. One should then examine the kernel equation at $z = \rho$,

$$u^5 - \frac{1}{36}u^5 P(u) = 0,$$

as this gives all the values of the small branches there. We find that there are 10 roots, amongst which $\tau = 1$ is a double root. The eight other go by pairs of complex conjugates, with

$$\zeta \doteq -0.36381 + 0.22924\,i, \quad \zeta' \doteq 0.06208 + 0.47622\,i,$$
$$\zeta'' \doteq -1.96746 + 1.23976\,i, \quad \zeta''' \doteq 0.26919 + 2.06476\,i.$$

Then, the quantity $Y_1(\rho)$ is determined numerically as the product of the roots of modulus less than $\tau = 1$, namely, $\zeta\,\overline{\zeta}\,\zeta'\,\overline{\zeta'}$. We find $Y_1(\rho) \doteq 0.42648$, so that the constant in the asymptotic formula for excursions can be determined to great accuracy:

$$E_n \sim C \cdot \frac{36^n}{\sqrt{n^3}}, \quad C \doteq 0.35865\,42111\,34518\,86172. \tag{46}$$

In the same vein, we determine $\overline{Y}_1(\rho) = (1-\zeta)(1-\overline{\zeta})(1-\zeta')(1-\overline{\zeta'})$ to be $\overline{Y}_1(\rho) \doteq 2.11615$, and

$$\frac{1}{36^n}[z^n]F(z,1) \sim \frac{C'}{\sqrt{n}}, \quad C' \doteq 0.93071\,59694\,87799\,20216$$

gives the probability of a lucky game (a meander). □

Pólya's example is interesting structurally. For instance, the excursion constant C in (46) involves $Y_1(\rho)$ that is a root of a self-reciprocal polynomial $\Xi(y)$ of degree 16 (found by Platypus Algorithm and factorization), itself equivalent to a resolvent of degree 8 that turns out to be irreducible,

$$\Xi(y) = y^8 \widehat{\Xi}(y + y^{-1})$$
$$\widehat{\Xi}(v) = v^8 - 17v^7 - 152v^6 + 34v^5 - 551v^4 - 12053v^3 + 8038v^2 + 38692v + 12664,$$

but algebra stops there. In contrast, analysis based on the decomposability devolving from the kernel method provides fully satisfactory numerical answers.

3.3. **Periodicities.** The discussion above has been conducted under the assumption of aperiodicity. As we explain now, similar results hold for *periodic* walks provided suitable congruence conditions are imposed on the indices of coefficients of generating function. For reasons explained after Definition 4, we freely assume the set of jumps to be at least reduced, as this implies no loss in generality.

Take a set $\mathcal{S}$ corresponding to period p. We sketch the discussion in the case of excursions, with $E(z)$ the corresponding GF. Then, $E(z)$ is periodic with period p, meaning that it is of the form $E(z) = \widehat{E}(z^p)$ for some $\widehat{E}(z)$ that is analytic at 0. The foregoing discussion of small branches continues to apply as long as $|z|$ stays inside the disk $|z| < \rho$, and the local analysis (42) of u_1 continues to hold as $z \to \rho$.

However, it appears now that there are p conjugate dominant singularities at the points

$$\rho_j := \rho\eta^j, \qquad \eta = e^{2i\pi/p}.$$

Indeed, $E(z)$ satisfies $E(z) = E(\eta z)$, while Equation (42) describes the behaviour of $u_1(z)$ at ρ_j upon changing z into z/η^j. Then, each of the p singular elements cumulate and contribute jointly to $[z^n]E(z)$ provided $n \equiv 0 \mod p$. One finds in this way

$$E_n \sim p\,\epsilon_0 \frac{P(\tau)^n}{2\sqrt{\pi n^3}}, \qquad n = p\nu,\ \nu \in \mathbb{Z}_{\geq 0}$$

where ϵ_0 is (still) given by (38).

The analysis easily adapts to the other types of paths considered, and is summarized by a simple rule: *For a system of jumps of period p, the asymptotic form of the count of index n must be restricted to a suitable congruence class of $n \mod p$ in order for objects to exists; then the corresponding asymptotic formula is obtained from the estimate of the aperiodic case through multiplication by a factor of p.*

Example 9. *Asymptotics of generalized Duchon's clubs.* We return to Example 5. The kernel equation is $1 - z(u^{-c} + u^d) = 0$, which gives the structural constant

$$\tau = \left(\frac{c}{d}\right)^{1/e}, \quad e = c + d.$$

The period is equal to e. The number of excursions of length n is nonzero only if $n \equiv 0 \pmod e$ and it satisfies (with $r = \rho^e$)

$$E_{e\nu} \sim D_{c,d}\, r_{c,d}^{-\nu}\, \nu^{-3/2}, \qquad r_{c,d} = \frac{c^c d^d}{e^e},$$

for some computable constant $D_{c,d}$. This generalizes the estimate of Duchon [22] who determined $D_{2,3}$ by a particular grammar construction followed by a specific algebraic elimination. □

4. Basic parameters and limit laws

The singular structure of basic generating functions of paths, bridges, meanders, and excursions is well established by Section 3. On the other hand, many parameters "decompose" combinatorially, so that their GF's are expressible in terms of the basic generating functions, or equivalently, they lie in $\mathbb{Q}(z, X; u_1, \ldots, u_c)$ for some set X of markers. In this paper, we only exhibit few sample cases of application of this methodology. As pointed by Philippe Robert (private communication), the whole combinatorial-analytic apparatus largely parallels what probabilists do by means of Wiener-Hopf decompositions (this is analogous to the separation between small and large branches) and Tauberian theorems (instead of singularity analysis that affords greater asymptotic accuracy through complete asymptotic expansions).

4.1. **Arches and contacts.** Define an arch as an excursion of size > 0 whose only contact with the horizontal axis is at its end points and let $\mathcal{A}$ be the set of arches. The set $\mathcal{E}$ of excursions satisfies the combinatorial equation

$$\mathcal{E} \cong \mathfrak{S}\{\mathcal{A}\},$$

where $\mathfrak{S}$ denotes the combinatorial construction that freely forms sequences. By well known mechanisms this translates directly into the GF equation

$$E(z) = \frac{1}{1 - A(z)}, \quad \text{or, equivalently,} \quad A(z) = 1 - \frac{1}{E(z)}. \tag{47}$$

The singular form of $A(z)$ then reads immediately:

$$E(z) \sim E(\rho) - \epsilon_0 \sqrt{1 - z/\rho}, \quad \text{implying} \quad A(z) \sim \left(1 - \frac{1}{E(\rho)}\right) - \frac{\epsilon_0}{E(\rho)^2}\sqrt{1 - z/\rho}.$$

Thus, the number of arches A_n is asymptotically proportional to $\rho^{-n} n^{-3/2}$, hence also to the number of excursions E_n.

Define a vertex of an excursion not equal to one of the end points to be a *contact* if its altitude is 0. Then, $A(z)^{k+1}$ is the GF of excursions having k contacts. For any fixed k, the function A^{k+1} has again a singularity of the square root type that is amenable to singularity analysis. An easy calculation then gives:

Theorem 5. *The probability that a random excursion of size n has k contacts is for any fixed k of the form*

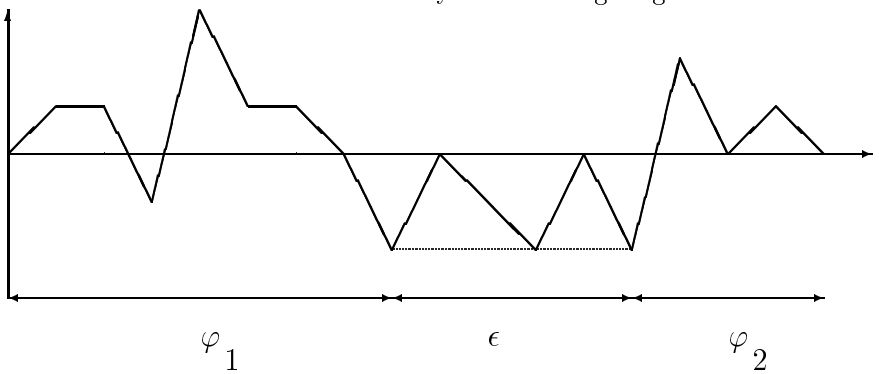
$$\frac{1}{E(\rho)^2}(k+1)\left(1 - \frac{1}{E(\rho)}\right)^k + O\left(\frac{1}{n}\right).$$

The number of contacts is thus asymptotically distributed like the sum of two independent geometric random variables with parameter $1 - E(\rho)^{-1}$. In particular,

$$A_n \sim \frac{1}{E(\rho)^2} E_n.$$

The constant $E(\rho)$ is expressible in terms of the quantity $Y_1(\rho)$ and is thus a close relative of β_0 introduced in Theorem 3.

On the relation between bridges and excursions. We briefly discuss here a construction that relates excursions to arches. Consider a bridge and let m (with $m \leq 0$) be the minimal altitude of any vertex. Any nonempty bridge β decomposes uniquely into a walk φ_1 of size ≥ 1 from 0 to m that only reaches level m at its right end, followed by an excursion ε (this is the part where one wanders around but above level m), followed by a path φ_2 of size ≥ 0 from m to 0 that only touches level m at its beginning. By rearrangement, one can write $\beta = \varepsilon \cdot (\varphi_2 | \varphi_1)$, where the glueing of $\varphi_2\varphi_1$ is an arch and the bar keeps track of where the splitting should occur. This construction is illustrated by the following diagram:



In other words, the set of nonempty bridges is combinatorially isomorphic to the product of the set of excursions by the set of arches with a split step that is

distinguished. This construction is then nothing but the combinatorial reflex of the identity

$$\overbrace{B(z)-1}^{\text{bridges}} = \overbrace{E(z)}^{\text{excursions}} \cdot \overbrace{\left(z\,\frac{d}{dz}A(z)\right)}^{\text{split arches}}, \tag{48}$$

which, in view of (47) is equivalent to

$$B(z)-1 = E(z)\cdot z\frac{d}{dz}\left(1-\frac{1}{E(z)}\right) = z\frac{E'(z)}{E(z)}.$$

(Thus, combinatorics of arches gives back Corollary 2.) Such relations are ubiquitous in the theory of paths, the most famous ones being known by the names of Spitzer and Sparre Andersen: see Kittel's appendix to [35] and Lothaire's book [52, Sec. 5.3] for a summary. Raney's classic [65] and Gessel's papers [38, 39] make use of similar ideas (*inter alia*, the "cycle lemma") in combinatorial proofs of the Lagrange inversion formula. One of the many consequences of this orbit of ideas, is for instance the possibility of analysing the number of times a bridge attains its minimum value by adapting the decomposition (48) and closely mimicking the proof of Theorem 5. Louchard's analyses in [53] provide many striking illustrations of such an interplay between probabilistic and combinatorial properties.

4.2. **Final altitude of a meander.** The *final altitude* of a path is the abscissa of its end point. For unconstrained paths, the usual local and central limit theorems for discrete random variables apply [40, Chapter 9], so that the limit law, after normalization, is Gaussian, the underlying technology being plainly the saddle point method. We consider now meanders. The random variable associated to finite altitude when taken over the set of all meanders of length n is denoted by X_n, and it satisfies

$$\Pr(X_n = k) = \frac{[z^n u^k]F(z,u)}{[z^n]F(z,1)}.$$

We state:

Theorem 6. *The final altitude of a random meander of size n admits a limit distribution, with the limit law being dictated by the value of the drift δ.*

(i) For a negative drift, $\delta < 0$, the limit distribution is a discrete one characterized in terms of the large branches:

$$\lim_{n\to\infty}\Pr(X_n = k) = [u^k]\,\varpi(u), \quad \textit{where} \quad \varpi(u) = \frac{(1-\tau)^2}{(u-\tau)^2}\prod_{\ell\geq 2}\frac{1-v_\ell(\rho)}{u-v_\ell(\rho)}.$$

(ii) In the case of zero drift, $\delta = 0$, the normalized random variable

$$\frac{X_n}{\vartheta\sqrt{n}}, \qquad \vartheta = \sqrt{\frac{P''(1)}{P(1)}},$$

converges in law to a Rayleigh distribution defined by the density $xe^{-x^2/2}$:

$$\lim_{n\to\infty}\Pr\left(\frac{X_n}{\vartheta\sqrt{n}}\leq x\right) = 1-e^{-x^2/2}.$$

(*iii*) *In the case of a positive drift, $\delta > 0$, the standardized version of X_n,*

$$\frac{X_n - \mu n}{\sigma\sqrt{n}}, \quad \mu = \frac{P'(1)}{P(1)}, \quad \sigma^2 = \left(\frac{P''(1)}{P(1)} + \frac{P'(1)}{P(1)} - \left(\frac{P'(1)}{P(1)} \right)^2 \right),$$

converges in law to a Gaussian variable $\mathcal{N}(0,1)$:

$$\lim_{n\to\infty} \Pr\left(\frac{X_n - \mu n}{\sigma\sqrt{n}} \le x \right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2}\, dy.$$

In the case of a negative drift, the limiting distribution admits an explicit form

$$[u^k]\, \varpi(u) = \tau^{-k}(c_0 + c_1 k) + \sum_{\ell \ge 2} c_\ell v_\ell(\rho)^{-k},$$

for a set of constants c_j that can be made explicit by a partial fraction expansion of $\varpi(u)$.

Proof. (*i*) For a negative drift, one directly shows that the probability generating function of X_n at u converges pointwise to a limit that precisely equals $\varpi(u)$, the convergence holding for $u \in (0,1)$. By the fundamental continuity theorem [27, p. 280] for probability generating functions (PGF's), this entails convergence in law of the corresponding discrete distributions.

We now fix a value of u taken arbitrarily in $(0,1)$ and treated as a parameter. The PGF of X_n is

$$\frac{[z^n]F(z,u)}{[z^n]F(z,1)},$$

where $F(z,u)$ is given by Theorem 2. In the case of a negative drift we know from the proof of Theorem 4 that $\tau = v_1(\rho)$ satisfies $\tau > 1$ while the radius of convergence of $F(z,1)$ coincides with the structural radius ρ. Then, the quantity

$$\overline{Y_1}(z,u) = \prod_{\ell \ge 2}^{d} \frac{1}{u - v_\ell(z)}$$

is analytic in the closed disk $|z| \le \rho$: being a symmetric function of the nonprincipal large branches, it has no algebraic singularity there; given the already known domination relations between the large branches (Lemma 2), the denominators cannot vanish.

It then suffices to analyse the factor containing the principal large branch v_1. This factor has a branch point at ρ, where

$$\frac{1}{u - v_1(z)} \sim \frac{1}{u-\tau} + \frac{1}{(u-\tau)^2}\sqrt{2\frac{P(\tau)}{P''(\tau)}}\sqrt{1 - z/\rho},$$

as follows directly from (42) and the fact that v_1 is conjugate to u_1 at $z = \rho$. Singularity analysis then gives instantly the fact that, for some nonzero constant C,

$$[z^n]F(z,u) \sim C\rho^{-n} n^{-3/2} \Omega(u), \quad \text{where} \quad \Omega(u) = \frac{1}{(u-\tau)^2}\overline{Y_1}(\rho, u),$$

and the result follows after normalization by $[z^n]F(z,1)$.

For the remaining two cases, it will prove convenient first to estimate the mean value (expectation $\mathrm{E}(\cdot)$) of X_n,

$$\mathrm{E}(X_n) = \frac{[z^n]F'_u(z,1)}{[z^n]F(z,1)}, \tag{49}$$

where F'_u indicates differentiation with respect to u. Logarithmic differentiation gives

$$F'_u(z,1) = F(z,1)\sum_{\ell=1}^{d}\frac{1}{1-v_\ell(z)} \tag{50}$$

from which one attains singularities easily.

(*ii*) In the case of a zero drift, the value of the structural constant is $\tau{=}1$ and the radius of convergence of $F(z,1)$ is $\rho = 1/P(\tau) = 1/P(1)$. Then, the singularity at ρ of $F'_u(z,1)$ combines a factor $1/\sqrt{1-z/\rho}$ that arises from $F(z,1)$ and another similar factor that arises from the term $(1-v_1(z))^{-1}$. This singularity is thus, to first order asymptotics, similar to a simple pole. A computation based again on (42) reveals that the mean value of X_n is of the order of $\sqrt{n}$. Precisely, one finds

$$E(X_n) \sim \vartheta\sqrt{\frac{\pi n}{2}}, \qquad \vartheta = \sqrt{\frac{P''(1)}{P(1)}}.$$

(Note that $\sqrt{\pi/2}$ is the mean of the standard Rayleigh distribution.)

The formula of Corollary 3 then suggests that $F_k(z)$ should behave very much like v_1^k, implying that the coefficients should resemble, up to scaling, the coefficients in the large power $[z^n](1-\sqrt{1-z})^k$. Such a situation is known to be conducive to Rayleigh laws: it is covered extensively in Drmota and Soria's study [21] and revisited in the paper [5]; see also [20]. In particular Theorem 1 of [21] gives us the convergence in distribution to the Rayleigh law, while a simple adaptation of the results of Appendix B in [5] provides corresponding density estimates (a "local" limit law). We omit the tedious but routine details.

(*iii*) For a positive drift, probabilistic intuition indicates that there are relatively few chances for a walk to ever come under the negative axis, and when this happens, it only tends to do so early in the history of the walk. Consequently, the final altitude should be only marginally affected by the meander conditioning.

In this case, one has $\tau < 1$ and the radius of convergence of $F(z,1)$ is $\rho_1 = 1/P(1)$ while the structural radius satisfies $\rho > \rho_1$. By definition, one has $v_1(\rho_1) = 1$. Consequently, the function $F'_u(z,1)$ in (50) admits a double pole at ρ_1, with

$$F'_u(z,1) \sim F(z,1)\frac{1}{v_1'(\rho_1)(z-\rho_1)}.$$

so that (one has $v_1'(\rho_1) = -(\rho_1^2 P'(1))^{-1}$),

$$\mathrm{E}(X_n) = \frac{[z^n]F'_u(z,1)}{[z^n]F(z,1)} = n\frac{P'(1)}{P(1)} + O(1).$$

In the probabilistic case, the coefficient of n in the estimate reduces to the drift, and this estimate does agree with the probabilistic argument sketched above. Similarly, the variance is found to satisfy

$$\operatorname{Var} X_n = \left(\frac{P''(1)}{P(1)} + \frac{P'(1)}{P(1)} - \left(\frac{P'(1)}{P(1)}\right)^2\right) n + O(1).$$

Finally, the Gaussian law is established from the power-sum form of Corollary 3 upon applying Cauchy's coefficient formula. One has

$$[z^n]F_k(z) = \frac{1}{2i\pi}\int_{|z|=\rho_1} \xi_1(z)v_1(z)^{-k-1}\,\frac{dz}{z^{n+1}} + R_{n,k},$$

The error term $R_{n,k}$ that arises from all the nonprincipal branches is exponentially smaller than ρ_1^{-n} because of the domination properties of $1/v_1(z)$ (see the proof of Lemma 2, once more). The main integral is then treated by the saddle point method in the range considered, $k = \mu n + O(\sqrt{n})$ with $\mu := P'(1)/P(1)$. The saddle point of the integrand is at ρ_1, very nearly. The Gaussian density then comes out from a standard saddle point perturbation analysis. □

5. Directed two-dimensional models

The kernel method is generally well suited to problems where all the jumps are of the form (a_j, b_j) with $a_j \geq 0$. In this case, each choice of a step implies progression along the horizontal axis. One considers the trivariate GF

$$F(z; x, y) := \sum_{n,p,q} F_{n,p,q}\, z^n x^p y^q,$$

where $F_{n,p,q}$ is the number of meander paths in $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ with size (number of steps) equal to n that connect the origin to the point of coordinates (p, q). The walk is thus directed in the sense of Section 1. As we now explain, such enumeration problems, though formulated in two-dimensional space, are in fact fake 1-dimensional problems amenable to the kernel method.

In the directed case, the method of "adding a slice" encountered in Equations (14) and (16) gives rise to the fundamental equation

$$F(z; x, y)(1 - zP(x, y)) = 1 - z\{y^{<0}\}\left(P(x, y)F(z; x, y)\right), \tag{51}$$

where the characteristic polynomial is now

$$P(x, y) := \sum_j x^{a_j} y^{b_j},$$

which is entire in x but of Laurent type with respect to y. The parameters of size (marked by z) and horizontal displacement (marked by x) are bound by linear inequalities, and one of them can be treated as the basic variable, the other as an auxiliary parameter or even the constant 1. Then, the adaptation of the kernel method consists in *binding* the Laurent variable, here y, to the basic variable chosen (x or z) by

$$1 - zP(x, y) = 0. \tag{52}$$

Newton's polygon then shows that, for the bound equation, the number of "small" roots of the kernel equation coincides with the maximum negative vertical span, namely, $c := |\min_j b_j|$, and this number is precisely the number of unknown functions in the right side of (51). We let u_j represent these small branches. The treatment of walks and bridges adapts easily from what has been done earlier. Regarding excursions and meanders, substitution of the u_j then shows the following: *The GF of excursions (defined by final altitude 0) and the BGF of meanders (defined by final altitude ≥ 0) depend rationally on the variables z, x and the set of small branches $\{u_j\}$ of the associated "kernel equation"* (52).

EXAMPLE 10. *Chess moves of Labelle and Yeh.* In two papers [49, 50], Labelle and Yeh develop an interesting set of decompositions for generalized knight moves on a chessboard. The standard version of the problem is: *Consider the* $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ *chessboard. How many sequences of Eastbound knight moves* ($\mathcal{S} = \{(1,2),(1,-2),(2,1),(2,-1)\}$) *are there from* $(0,0)$ *to* $(n,0)$*?* By definition, the moves are not allowed to involve points with negative coordinates.

As size is not needed, we take x as the independent variable and set $z = 1$. The kernel equation is then

$$1 - (xy^2 + xy^{-2} + x^2y + x^2y^{-1}) = 0.$$

so that the characteristic curve is a quartic. The vertical symmetry of the moves implies that the kernel equation can be rewritten as a combination of two quadratic equations,

$$1 - x(W^2 + xW - 2) = 0, \qquad W := y + \frac{1}{y}.$$

There results that the four branches of the characteristic equation are given by

$$y_{\pm}(W) = \frac{1}{2}\left(W \pm \sqrt{W^2 - 4}\right), \quad W_{\pm}(x) = \frac{1}{2x}\left(-x^2 \pm \sqrt{x^4 + 8x^2 + 4x}\right).$$

It appears that the two small branches u_1, u_2 correspond to taking opposite signs in the determinations of $y(W)$ and $W(x)$, and one finds for the GF of excursions (i.e., paths terminating at altitude 0), in complete analogy to the simple walk,

$$\begin{aligned} E(x) &= -\frac{1}{x}(u_1(x)u_2(x)) = -\frac{1}{x}\, y_-(W_+(x)) \cdot y_+(W_-(x)) \\ &= 1 + x^2 + 3\,x^4 + 2\,x^5 + 12\,x^6 + 14\,x^7 + 54\,x^8 + 86\,x^9 + \cdots . \end{aligned}$$

This is the sequence (a_n) of [49] and also *EIS* **A005220**. Decomposability renders especially easy the asymptotic analysis of the number of excursions and of corresponding parameters. More general knight moves can be treated similarly by the kernel method. In particular, the equation satisfied by the excursion generating functions tends to be of a degree exponential in c; see [49, 50]. Here, the kernel method yields a reduction to an equation of degree $2c$, which even reduces to a resolvent of degree c when symmetry is taken into account via the W-parameterization. This illustrates a sharp contrast between the exponential blow-up in combinatorial complexity and the linear character of the analytic complexity. □

6. CONCLUSION

In this paper, we have aimed at illustrating the analytic tractability of many 1-dimensional path problems, a boon of the kernel method. The reduction in the asymptotic-analytic complexity of the problem is often spectacular, as exemplified by Duchon's clubs or the Labelle-Yeh knight moves. Parameters that are easily readable on paths lead to generating functions whose singularities arise simply from the branches of a characteristic curve of low degree. The method applies to all 1-dimensional problems as well as to 2-dimensional problems provided they remain directed. For a thorough discussion of the algebraic power of the kernel method, we refer once more to the study by Bousquet-Mélou and Petkovšek [13]. (The kernel technique is also reminiscent of Tutte's quadratic method much of use in the enumerative theory of planar maps [42]; see Bousquet-Mélou's paper [11] for a perspective.)

The case of undirected 2-dimensional problems, where one can go back and forth in all four cardinal directions, is appreciably harder. Even in the case of movement of amplitude ≤ 1, Fayolle *et al.* show in [26] that *stationary* solutions involve elliptic functions and integrals. Some directed path problems in dimension higher than 2 can however still be successfully treated by specific combinatorial decompositions; see [12] for an example.

A tribute to Maurice Nivat. As is apparent from the bibliography of this paper, many papers directly relevant to our study have been published in the journal *Theoretical Computer Science* along the years. We owe much for this to the Editor-in-Chief, Maurice Nivat. His openness of mind has been a constant help in the emergence and shaping up of sub-communities within theoretical computer science. Examples are the GASCOM (Generation of Random Combinatorial Objects) and AofA (Analysis of Algorithms) communities which have greatly benefitted from special issues of TCS, this at the invariably encouraging initiative of Maurice. In view of this and of Maurice's long-standing interest in similar discrete geometrical objects (see, e.g., [6, 7, 9, 17]), we kindly dedicate this study to him.

Acknowledgements. This work was supported in part by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT). The authors are grateful to Philippe Robert for many insightful discussions on probabilistic aspects of random walks as well as to Mireille Bousquet-Mélou and an anonymous referee for a careful scrutiny of the paper that greatly helped us improve our presentation. Thanks finally to Christian Krattenthaler for valuable bibliographical remarks.

References

1. S.-S. Abhyankar, *Algebraic geometry for scientists and engineers*, American Mathematical Society, 1990.
2. Cyril Banderier, *Combinatoire analytique: application aux marches aléatoires*, D.E.A. memoir, Université Paris VI, July 1998.
3. ———, *Combinatoire analytique des chemins et des cartes*, Ph.D. thesis, Université Paris VI, June 2001.
4. Cyril Banderier, Mireille Bousquet-Mélou, Alain Denise, Philippe Flajolet, Danièle Gardy, and Dominique Gouyou-Beauchamps, *Generating functions of generating trees*, Technical Report ALCOM FT-TR-01-17, Alcom-FT Project, February 2001, 26 pages. Accepted for publication in *Discrete Mathematics*.
5. Cyril Banderier, Philippe Flajolet, Gilles Schaeffer, and Michèle Soria, *Random maps, coalescing saddles, singularity analysis, and Airy phenomena*, Preprint, March 2001, 47 pages. Accepted for publication in *Random Structures & Algorithms*.
6. Elena Barcucci, Sara Brunetti, Alberto Del Lungo, and Maurice Nivat, *Reconstruction of discrete sets from three or more X-rays*, Algorithms and complexity (Rome, 2000), Springer, Berlin, 2000, pp. 199–210.
7. Elena Barcucci, Alberto Del Lungo, Maurice Nivat, and Renzo Pinzani, *Reconstructing convex polyominoes from horizontal and vertical projections*, Theoretical Computer Science **155** (1996), no. 2, 321–347.
8. Elena Barcucci, Renzo Pinzani, and Renzo Sprugnoli, *The random generation of directed animals*, Theoretical Computer Science **127** (1994), no. 2, 333–350.
9. Danièle Beauquier, Maurice Nivat, Éric Rémila, and Mike Robson, *Tiling figures of the plane with two bars*, Computational Geometry, Theory and Applications **5** (1995), no. 1, 1–25.
10. Jean Berstel (ed.), *Séries formelles*, LITP, University of Paris, 1978, (Proceedings of a School, Vieux–Boucau, France, 1977).
11. Mireille Bousquet-Mélou, *On (some) functional equations arising in enumerative combinatorics*, Preprint, 2001.

12. Mireille Bousquet-Mélou and Anthony J. Guttmann, *Three-dimensional self-avoiding convex polygons*, Physical Review E. Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics. Third Series **55** (1997), no. 6, part A, R6323–R6326.
13. Mireille Bousquet-Mélou and Marko Petkovšek, *Linear recurrences with constant coefficients: the multivariate case*, Discrete Mathematics **225** (2000), no. 1-3, 51–75.
14. Louis Comtet, *Calcul pratique des coefficients de Taylor d'une fonction algébrique*, Enseignement Mathématique. **10** (1964), 267–270.
15. ———, *Advanced combinatorics*, Reidel, Dordrecht, 1974.
16. N. G. de Bruijn, *Asymptotic methods in analysis*, Dover, 1981, A reprint of the third North Holland edition, 1970 (first edition, 1958).
17. Alberto Del Lungo, Maurice Nivat, and Renzo Pinzani, *The number of convex polyominoes reconstructible from their orthogonal projections*, Proceedings of the 6th Conference on Formal Power Series and Algebraic Combinatorics (New Brunswick, NJ, 1994), vol. 157, 1996, pp. 65–78.
18. Marie-Pierre Delest and Gérard Viennot, *Algebraic languages and polyominoes enumeration*, Theoretical Computer Science **34** (1984), 169–206.
19. Peter G. Doyle and J. Laurie Snell, *Random walks and electric networks*, Mathematical Association of America, Washington, DC, 1984.
20. Michael Drmota, *Asymptotic distributions and a multivariate Darboux method in enumeration problems*, Journal of Combinatorial Theory, Series A **67** (1994), 169–184.
21. Michael Drmota and Michèle Soria, *Images and preimages in random mappings*, SIAM Journal on Discrete Mathematics **10** (1997), no. 2, 246–269.
22. Philippe Duchon, *On the enumeration and generation of generalized Dyck words*, Discrete Math. **225** (2000), no. 1-3, 121–135, Formal power series and algebraic combinatorics (Toronto, ON, 1998).
23. Marianne Durand, *Asymptotics of the "klam" recurrence*, Preprint, 2001.
24. L. Euler, *Observationes analyticae*, Novi Commentarii Acad. Sci. Imper. Petropolitanae **11** (1765), 124–143.
25. Guy Fayolle and Roudolf Iasnogorodski, *Two coupled processors: the reduction to a Riemann-Hilbert problem*, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete **47** (1979), no. 3, 325–351.
26. Guy Fayolle, Roudolf Iasnogorodski, and Vadim Malyshev, *Random walks in the quarter-plane*, Springer-Verlag, Berlin, 1999.
27. W. Feller, *An introduction to probability theory and its applications*, third ed., vol. 1, John Wiley, 1968.
28. ———, *An introduction to probability theory and its applications*, vol. 2, John Wiley, 1971.
29. Philippe Flajolet, *Combinatorial aspects of continued fractions*, Discrete Mathematics **32** (1980), 125–161.
30. ———, *The evolution of two stacks in bounded space and random walks in a triangle*, Mathematical Foundations of Computer Science (J. Gruska, B. Rovan, and J. Wiedermann, eds.), Lecture Notes in Computer Science, vol. 233, Springer Verlag, 1986, Proceedings of the 12th MFCS Symposium, Bratislava, August 1986, pp. 325–340.
31. ———, *Analytic models and ambiguity of context–free languages*, Theoretical Computer Science **49** (1987), 283–309.
32. Philippe Flajolet, Jean Françon, and Jean Vuillemin, *Sequence of operations analysis for dynamic data structures*, Journal of Algorithms **1** (1980), 111–141.
33. Philippe Flajolet and Fabrice Guillemin, *The formal theory of birth-and-death processes, lattice path combinatorics, and continued fractions*, Advances in Applied Probability **32** (2000), 750–778.
34. Philippe Flajolet and Andrew M. Odlyzko, *Singularity analysis of generating functions*, SIAM Journal on Algebraic and Discrete Methods **3** (1990), no. 2, 216–240.
35. D. Foata, *La série génératrice exponentielle dans les problèmes d'énumération*, S.M.S, Montreal University Press, 1974.
36. Jean Françon, *Histoires de fichiers*, RAIRO Informat. Théor. **12** (1978), no. 1, 49–62.
37. Harry Furstenberg, *Algebraic functions over finite fields*, Journal of Algebra **7** (1967), 271–277.
38. Ira M. Gessel, *A factorization for formal Laurent series and lattice path enumeration*, J. Combin. Theory Ser. A **28** (1980), no. 3, 321–337.

39. ———, *A noncommutative generalization and q–analog of the Lagrange inversion formula*, Transactions of the American Mathematical Society **257** (1980), no. 2, 455–482.
40. B. V. Gnedenko and A. N. Kolmogorov, *Limit distributions for sums of independent random variables*, Addison-Wesley, 1968.
41. H. W. Gould, *Research bibliography on two number sequences*, In *Mathematica Monongaliae*, 1971, (A comprehensive bibliography on Bell and Catalan numbers).
42. Ian P. Goulden and David M. Jackson, *Combinatorial enumeration*, John Wiley, New York, 1983.
43. Peter Henrici, *Applied and computational complex analysis*, vol. 2, John Wiley, New York, 1974.
44. E. Hille, *Analytic function theory*, Blaisdell Publishing Company, Waltham, 1962, 2 Volumes.
45. Frances Kirwan, *Complex algebraic curves*, London Mathematical Society Student Texts, no. 23, Cambridge University Press, 1992.
46. Donald E. Knuth, *The art of computer programming*, 3rd ed., vol. 1: Fundamental Algorithms, Addison-Wesley, 1997.
47. ———, *The art of computer programming*, 3rd ed., vol. 2: Seminumerical Algorithms, Addison-Wesley, 1998.
48. ———, *The art of computer programming*, 2nd ed., vol. 3: Sorting and Searching, Addison-Wesley, 1998.
49. Jacques Labelle and Yeong Nan Yeh, *Dyck paths of knight moves*, Discrete Appl. Math. **24** (1989), no. 1-3, 213–221, First Montreal Conference on Combinatorics and Computer Science, 1987.
50. ———, *Generalized Dyck paths*, Discrete Mathematics **82** (1990), 1–6.
51. Pierre-Simon Laplace, *Théorie analytique des probabilités. Vol. I, II*, Éditions Jacques Gabay, Paris, 1995, Reprint of the 1819 and 1820 editions.
52. M. Lothaire, *Combinatorics on words*, Encyclopedia of Mathematics and its Applications, vol. 17, Addison–Wesley, 1983.
53. Guy Louchard, *Asymptotic properties of some underdiagonal walks generation algorithms*, Theoretical Computer Science **218** (1999), no. 2, 249–262.
54. E. Lucas, *Théorie des Nombres*, Gauthier-Villard, Paris, 1891, Reprinted by A. Blanchard, Paris 1961.
55. A. Meir and J. W. Moon, *On the altitude of nodes in random trees*, Canadian Journal of Mathematics **30** (1978), 997–1015.
56. Donatella Merlini, D. G. Rogers, Renzo Sprugnoli, and M. Cecilia Verri, *Underdiagonal lattice paths with unrestricted steps*, Discrete Applied Mathematics. Combinatorial Algorithms, Optimization and Computer Science **91** (1999), no. 1-3, 197–213.
57. Sri Gopal Mohanty, *Lattice path counting and applications*, Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1979, Probability and Mathematical Statistics.
58. ———, *Combinatorial aspects of some random walks*, Random walks (Budapest, 1998), János Bolyai Math. Soc., Budapest, 1999, pp. 259–273.
59. T. V. Narayana, *Lattice path combinatorics with statistical applications*, University of Toronto Press, Toronto, Ont., 1979.
60. Maurice Nivat, *Langages algébriques sur le magma libre et sémantique des schémas de programme*, Automata, languages and programming (Proc. Sympos., Rocquencourt, 1972), North Holland, Amsterdam, 1973, pp. 293–308.
61. A. M. Odlyzko, *Asymptotic enumeration methods*, Handbook of Combinatorics (R. Graham, M. Grötschel, and L. Lovász, eds.), vol. II, Elsevier, Amsterdam, 1995, pp. 1063–1229.
62. Jim Pitman, *Brownian motion, bridge, excursion, and meander characterized by sampling at independent uniform times*, Electron. J. Probab. **4** (1999), no. 11, 33 pp. (electronic).
63. G. Pólya, *Sur les séries entières dont la somme est une fonction algébrique*, Enseignement mathématique **1–2** (1921–1922), 38–47.
64. Helmut Prodinger, *On a functional-difference equation of Runyon, Morrison, Carlitz, and Riordan*, Séminaire Lotharingien de Combinatoire **46** (2001), paper B46a, 4 pages (electronic).
65. G. N. Raney, *Functional composition patterns and power series reversion*, Transactions of the American Mathematical Society **94** (1960), 441–451.
66. Philippe Robert, *Réseaux et files d'attente: méthodes probabilistes*, Mathématiques & Applications, vol. 35, Springer, Paris, 2000.

67. Bruno Salvy and Paul Zimmermann, *GFUN: a Maple package for the manipulation of generating and holonomic functions in one variable*, ACM Transactions on Mathematical Software **20** (1994), no. 2, 163–167.
68. Masako Sato, *Generating funtions for the number of lattice paths between two parallel lines with a rational incline*, Mathematica Japonica **34** (1989), no. 1, 123–137.
69. Robert Sedgewick and Philippe Flajolet, *An introduction to the analysis of algorithms*, Addison-Wesley Publishing Company, 1996.
70. N. J. A. Sloane, *The on-line encyclopedia of integer sequences*, 2000, Published electronically at `http://www.research.att.com/~njas/sequences/`.
71. N. J. A. Sloane and Simon Plouffe, *The encyclopedia of integer sequences*, Academic Press, 1995.
72. R. P. Stanley, *Enumerative combinatorics*, vol. II, Cambridge University Press, 1998.
73. J. van Leeuwen (ed.), *Handbook of theoretical computer science*, vol. B: Formal Models and Semantics, North Holland, 1990.
74. Andrew Chi Chih Yao, *An analysis of* $(h,\ k,\ 1)$*-Shellsort*, J. Algorithms **1** (1980), no. 1, 14–50.
75. ———, *An analysis of a memory allocation scheme for implementing stacks*, SIAM Journal on Computing **10** (1981), no. 2, 398–403.

Cyril Banderier, Algorithms Project, INRIA, Rocquencourt, 78150 Le Chesnay (France).
E-mail address: `Cyril.Banderier@inria.fr`, `http://algo.inria.fr/banderier`

Philippe Flajolet, Algorithms Project, INRIA, Rocquencourt, 78150 Le Chesnay (France).
E-mail address: `Philippe.Flajolet@inria.fr`, `http://algo.inria.fr/flajolet`

Stories about groups and sequences

Peter J. Cameron

School of Mathematical Sciences
Queen Mary and Westfield College
Mile End Road
London E1 4NS
U.K.

Beyond Ghor there was a city. All its inhabitants were blind. A king with his entourage arrived near by. He brought his army and camped in the desert. He had a mighty elephant, which he used in attack and to increase the people's awe.

The populace became anxious to see the elephant, and some sightless ones from among this blind community ran to find it. As they did not even know the form or shape of the elephant they groped sightlessly, gathering information by touching some part of it. Each thought he knew something, because he could feel a part.

When they returned to their fellow-citizens, eager groups clustered around them. Each of these was anxious to learn the truth from those who were themselves astray. They asked about the form, the shape of the elephant, and they listened to all they were told.

The man whose hand had reached an ear was asked about the elephant's nature. He said: "It is a large, rough thing, wide and broad, like a rug."

And the one who had felt the trunk said: "I have the real facts about it. It is like a straight and hollow pipe, awful and destructive."

The man who had felt its feet and legs said: "It is mighty and firm, like a pillar."

Mualana Jalaluddin Rumi (13th century) (from [34])

Abstract

The main theme of this article is that counting orbits of an infinite permutation group on finite subsets or tuples is very closely related to combinatorial enumeration; this point of view ties together various disparate "stories".

1 Two-graphs and even graphs

The first story originated with Neil Sloane, when he was compiling the first edition of his dictionary of integer sequences [35]. He observed that certain counting sequences appeared to agree.

The first sequence enumerates *even graphs*, those in which any vertex has even valency (so that the graph is a disjoint union of Eulerian graphs). These graphs were enumerated by Robinson [29] and Liskovec [18].

The second sequence counts switching classes of graphs. If Γ is a graph on the vertex set X, and Y is a subset of X, the result of *switching* Γ with respect to Y is obtained by deleting all edges between Y and its complement, putting in all edges between Y and its complement which didn't exist before, and leaving the rest unaltered. Switching is an equivalence relation on the graphs with vertex set X; the equivalence classes are called *switching classes*. This concept was introduced by Seidel [30] for studying strongly regular graphs.

The final sequence counts two-graphs. A *two-graph* on a set X consists of a set $\mathcal{T}$ of *triples* or 3-element subsets of X with the property that any 4-element subset of $\mathcal{T}$ contains an even number of elements of $\mathcal{T}$. Two-graphs were introduced by G. Higman in a construction of Conway's third sporadic group. The theory has been developed in many directions: Seidel has written several surveys [31], [33], [32]. They also link several themes in combinatorics, including equiangular lines in Euclidean space, and double covers of complete graphs.

It was already known that switching classes and two-graphs are equinumerous. There is a map from graphs on the set X to two-graphs on X, as follows: the triples of the two-graph are all 3-sets which contain an odd number of edges of the graph. Every two-graph is obtained in this way, and graphs Γ_1 and Γ_2 give the same two-graph if and only if they lie in the same switching class. So there is a natural bijection from switching classes to two-graphs.

It was also known that switching classes and even graphs on an odd number of vertices are equinumerous. (Any switching class on an odd number of vertices contains a unique even graph, obtained by taking any graph in the class and switching with respect to the set of vertices of odd degree.) But no such correspondence exists if the number of vertices is even. Mallows and Sloane [21] proved that the numbers were equal by deriving a formula for the number of switching classes and observing that it coincides with the Robinson–Liskovec formula for the number of even graphs.

The "right" explanation [6] actually shows that the classes are dual. Let X be a set of n points, and V the set of all graphs on the vertex set X. Each graph can be represented by a binary vector of length $n(n-1)/2$ whose ones give the positions of the edges. So V is a vector space over GF(2) of dimension $n(n-1)/2$. The addition in V corresponds to taking the symmetric difference of the edge sets of the two graphs. We consider two subsets of V:

- U, the set of complete bipartite graphs;
- W, the set of even graphs.

It is easy to see that U is a subspace of V, spanned by the stars. Now a graph is even if and only if it is orthogonal to all stars; so $W = U^\perp$, and W is also a subspace.

The cosets of U in V are precisely the switching classes of graphs. So V/U is the set of switching classes. Since $W = U^\perp$, this quotient V/U is isomorphic to the dual space W^* of W, not just as vector space, but as module for the symmetric group on X. Now a group acting on a finite vector space has equally many orbits on the space and on its dual, by Brauer's lemma [4]; and the orbits of the symmetric group are the isomorphism classes. So the numbers of switching classes and even graphs are equal.

Recently, I noticed another feature, which may be related in some way to this duality. As noted above, an even graph is the disjoint union of Eulerian graphs. A similar-looking decomposition holds for two-graphs. We define a relation $\sim$ on the point set of a two-graph by the rule that $x \sim y$ if and only if either $x = y$ or no triple contains x and y. From the definition of a two-graph, it is easy to see that this is an equivalence relation, and is even a congruence, that is, membership of a triple in $\mathcal{T}$ is unaffected if we replace some of its points by equivalent ones. Thus, a two-graph is described by a partition of X, with no structure on the parts of the partition, and the structure of a *reduced* two-graph (one in which all $\sim$-classes are singletons) on the set of parts. (By contrast, for even graphs, we have an Eulerian graph on each part of the partition, and no structure on the set of parts; this is, in some vague sense, "dual" to the preceding.)

The numbers of Eulerian graphs and of reduced two-graphs on n points agree for $n \leq 4$ but differ for $n = 5$.

2 Groups and counting

Let G be a permutation group on a set Ω. Usually Ω will be infinite. The group G is said to be *oligomorphic* if the number of orbits of G on the set of n-subsets of Ω is finite for every positive integer n. (More about the derivation of this term below.) So every finite permutation group is oligomorphic. If G is oligomorphic, we let $f_n(G)$ (or just f_n, if the group is clear) denote the number of orbits of G on n-sets.

Design theorists will recognise this set-up. Suppose that we want to construct a t-design on Ω with block size k admitting the group G. Let $T_1, \ldots, T_a$ be the orbits on t-sets, and $K_1, \ldots, K_b$ the orbits on k-sets, where $a = f_t$, $b = f_k$. Now we build a collapsed incidence matrix $M = (m_{ij})$ of size $a \times b$, where m_{ij} is the number of k-sets in the jth orbit which contain a fixed t-set from the ith orbit. Now the game is to select a subset of the columns of M such that the

submatrix has constant row sums; then the union of the corresponding orbits is the block set of the design.

This doesn't work if Ω is infinite, since the numbers m_{ij} may be infinite. However, collapsing the matrix the other way does make sense: let $P = (p_{ij})$, where p_{ij} is the number of t-sets in the ith orbit which are contained in a fixed k-set from the jth orbit. We will return to this later; but, unfortunately, I have nothing more to say about constructing designs!

The concept which links this kind of orbit counting to combinatorial enumeration is that of a homogeneous relational structure. A *relational structure* X on Ω consists of a number of relations on X of various arities. Thus, many of our favourite structures (graphs, digraphs, tournaments, total or partial orders, two-graphs) are relational. An *induced substructure* of a relational structure on a subset of Ω is obtained by simply taking the restrictions of all the relations to this subset. Now X is *homogeneous* if every isomorphism between finite substructures of X can be extended to an automorphism of X.

The classical example of a homogeneous structure is the rational numbers $\mathbb{Q}$ as ordered set. Given any two n-sets of rationals, arranged in increasing order as $a_1 < a_2 < \cdots < a_n$ and $b_1 < b_2 < \cdots < b_n$, there is a unique isomorphism between the substructures, taking a_i to b_i for $i = 1, \ldots, n$. This can be extended to an order-preserving map on all the rationals by "filling in" the intervals (a_i, a_{i+1}) with linear maps, and translating the two ends suitably.

Based on this example, Fraïssé [13] gave a necessary and sufficient condition for a class $\mathcal{C}$ of finite structures to be all the finite substructures of a countable homogeneous structure. I will give only a brief description of Fraïssé's condition here (it is discussed in detail in [7]). It is required that $\mathcal{C}$ is closed under isomorphism; closed under taking induced substructures; contains only countably many structures up to isomorphism; and has the *amalgamation property* (which asserts that, given two structures $B_1, B_2 \in \mathcal{C}$ with a common substructure A, there is a structure $C \in \mathcal{C}$ in which B_1 and B_2 can both be embedded, so that their intersection is at least A). The first three conditions are usually obvious, but the amalgamation property may require more effort to verify. Many familiar classes of finite structures (graphs, tournaments, posets, triangle-free graphs, two-graphs, ...) satisfy the condition, and many others (bipartite graphs, trees, ...) can be made to satisfy it after small modification. For example, graphs with a fixed bipartition satisfy Fraïssé's conditions.

Now let X be a homogeneous structure, and $\mathcal{C}$ the class of its finite substructures. If G is the automorphism group of X, then G-orbits on n-sets correspond to isomorphism classes of n-element structures in $\mathcal{C}$ (unlabelled substructures of X). Moreover, given any permutation group on a countable set, it is possible to construct a structure on which the group acts "homogeneously". So the problem of calculating the numbers $f_n(G)$ for oligomorphic groups G is identical to that of enumerating unlabelled structures in a class satisfying Fraïssé's condition (a *Fraïssé class*, I will say for short).

The term "oligomorphic" is derived from "few shapes", and is chosen to

express this relationship between the group orbits and the isomorphism classes of structures ("shapes") in a class with only finitely many of any given finite size ("few").

3 An inequality and a Ramsey problem

Because of the connection described in the last section, any general result on orbit numbers for oligomorphic groups is a metatheorem about enumerating structures in Fraïssé classes. The most basic result of this kind is that the numbers f_n are non-decreasing: $f_n \leq f_{n+1}$.

This was proved for finite permutation groups by Livingstone and Wagner [19], using character theory of the symmetric group. This result can be translated into a proof using Block's lemma together with the fact that the reduced incidence matrices defined in the last section have full rank provided that $|\Omega| \geq t + k$. As mentioned there, the matrix P is meaningful even when Ω is infinite, and can be shown to have full rank, from which the inequality can be deduced (taking $t = n$, $k = n + 1$).

A second, completely different proof was found by Pouzet [25], based on Ramsey's Theorem. The essential ingredient can be stated as a Ramsey theorem as follows:

Theorem 3.1 *Suppose that $t \leq k$, and let the t-subsets of the infinite set Ω be partitioned into finitely many classes T_i ($1 \leq i \leq a$), all non-empty. For any k-set U, let $p_i(U)$ denote the number of t-subsets of U in the class T_i. Let $P = (p_{ij})$ be the matrix whose columns are the distinct vectors $(p_1(U), \ldots, p_a(U))^\top$ which occur. Then, after re-ordering rows and columns if necessary, the matrix P is upper triangular with non-zero diagonal (that is, $p_{ij} = 0$ for $i > j$, while $p_{ii} \neq 0$).*

Like all good Ramsey theorems, this one has a finite version as well: it holds if Ω is sufficiently large in terms of t, k, a. Here the proof gives "sufficiently large" as a vast, iterated Ramsey number; yet there is some evidence that the result holds for sets of quite modest size. Nobody knows the true value of this Ramsey function.

Note that the fact that the rows of P are linearly independent is a simple consequence of the Ramsey theorem, and the inequality follows directly. (We take the classes of t-sets to be the orbits of G. Now two k-sets giving rise to different columns lie in different orbits, so f_k is at least equal to the number of distinct columns, which is at least the number f_t of rows.)

Macpherson, in [20] and other papers, has proved some powerful results about the rate of growth of the sequence $(f_n(G))$. For example, if G is primitive (that is, preserves no non-trivial equivalence relation), then either $f_n(G) = 1$ for all n, or the sequence grows at least exponentially.

4 Direct and wreath products

Next we turn to two methods of constructing new groups from old. If our groups are automorphism groups of homogeneous structures, then these two constructions translate into operations on the finite substructures, and hence on the sequences enumerating them. These operations are quite general, and do not depend on having a group around. (This point is the heart of the philosophy of these notes. In fact, a combinatorial setting more general than group orbits has been developed by A. Joyal [16] and his school, under the name *species*. This is very close in spirit to what I am doing here.)

The operations on sequences can often be expressed concisely in terms of their generating functions. Accordingly, if G is oligomorphic, we let

$$f_G(t) = \sum_{n=0}^{\infty} f_n(G)t^n.$$

(Note that $f_0(G) = 1$, since there is a unique empty set.)

First, let's have a couple of groups to feed into the constructions. Let S denote the symmetric group on an infinite set, and A the group of order-preserving permutations of the rational numbers. Then $f_n(S) = f_n(A) = 1$ for all n. (This is clear for S, and follows for A from our proof of the homogeneity of $\mathbb{Q}$.) Hence $f_S(t) = f_A(t) = 1/(1-t)$. The Fraïssé class corresponding to S consists of finite sets without any additional structure; that for A consists of finite totally ordered sets. In each case, there is just one object of each size n.

Let H be a permutation group on a set Γ, and K a permutation group on Δ. The *direct product* $H \times K$ (the set of all ordered pairs (h,k) with $h \in H$ and $k \in K$, with pointwise operations) acts on the disjoint union of the sets Γ and Δ, where the first component of a pair acts on Γ and the second component acts on Δ. Now a finite subset of $\Gamma \cup \Delta$ has the form $\Gamma_0 \cup \Delta_0$, where Γ_0 and Δ_0 are finite subsets of Γ and Δ respectively; two such sets lie in the same orbit of $H \times K$ if and only if their intersections with Γ lie in the same H-orbit, and similarly for Δ and K. So the sequence $(f_n(H \times K))$ is the *convolution* of the sequences $(f_n(H))$ and $(f_n(K))$:

$$f_n(H \times K) = \sum_{i=0}^{n} f_i(H) f_{n-i}(K),$$

and the generating functions simply multiply: $f_{H\times K} = f_H f_K$. Note that the terms of the sequence $(f_n(H \times S))$ are the partial sums of the sequence $(f_n(H))$.

More importantly, we see that a structure in the Fraïssé class for $H \times K$ is just the disjoint union of structures for H and K. So the direct product of permutation groups corresponds to the disjoint union of combinatorial structures. For example, the objects in the Fraïssé class for $S \times S$ can be taken to be finite sets whose elements are coloured red and blue; and $f_n(S \times S) = n+1$, since an n-set can contain $0, 1, 2, \ldots, n$ blue elements.

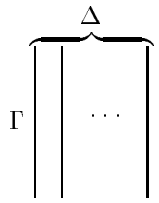


Figure 1: $\Gamma \times \Delta$ as a covering of Δ

There is another well-known permutation action of the direct product, on the Cartesian product of the sets Γ and Δ: the pair (h, k) maps (γ, δ) to $(\gamma h, \delta k)$. (This is the *product action* of $H \times K$.) If H and K are oligomorphic, then so is $H \times K$ in this action. However, the number of orbits on n-sets is not uniquely determined by the corresponding numbers for H and K. (*Exercise:* check that, in the product action, $f_2(S \times S) = 3$, while $f_2(A \times A) = 4$.) There are some very interesting questions here, but I won't say any more about this.

The other construction is the *wreath product* of permutation groups. It is convenient to build up the action first. The group $G = H \operatorname{Wr} K$ acts on the set $\Gamma \times \Delta$; but the factors should not be regarded as having the same status. Rather, think of $\Gamma \times \Delta$ as the disjoint union of $|\Delta|$ copies of Γ, each copy indexed by a point of Δ, as in Figure 1. (Formally, the copy Γ_δ of Γ indexed by δ is $\{(\gamma, \delta) : \gamma \in \Gamma\}$.) In topological terms, we regard $\Gamma \times \Delta$ as a covering of Δ whose *fibres* are the sets Γ_δ, each isomorphic to Γ.

The *base group* B of the wreath product consists of all permutations built from $|\Delta|$ independently chosen elements of H, each acting on the corresponding fibre. It is a cartesian product of $|\Delta|$ copies of H. The *top group* T is the group K, permuting the fibres by acting on their indices according to its given action on Δ. The wreath product is now the product BT. (In group-theoretic terms, B is normalised by T and $B \cap T = 1$, so the wreath product is the semi-direct product of B by T.)

What do the orbits of $H \operatorname{Wr} K$ on n-sets look like? Each n-set is partitioned by its intersections with the fibres; these intersections can be independently permuted to any other sets in the same fibre by the base group. However, the way in which the set of parts of the partition is permuted by the top group is less easy to describe.

Suppose that H and K are automorphism groups of homogeneous structures. Then an n-element structure in the Fraïssé class for $H \operatorname{Wr} K$ consists of a partition of the point set, together with independently chosen structures from the Fraïssé class for H on each part of the partition, and a structure from the Fraïssé class for K on the set of parts.

This combinatorial "composition", as with the disjoint union for the direct product, is meaningful even if there are no groups around. Consider the example in the first section. The class of even graphs is the composition of the class of

Eulerian graphs with the Fraïssé class for S; while the class of two-graphs is the composition of the Fraïssé class for S with the class of reduced two-graphs. (If there were homogeneous structures for the relevant classes, with automorphism groups $Even$, $Eulerian$, $TwoGr$ and $RedTwoGr$, then we would have

$$Even \sim Eulerian \operatorname{Wr} S, \qquad TwoGr \sim S \operatorname{Wr} RedTwoGr,$$

where $\sim$ means that the orbit counting sequences (f_n) are the same. (Unfortunately, the homogeneous structure exists only in the case of two-graphs.) These relations express formally the puzzle at the end of the first section.

It turns out that the sequence $(f_n(H \operatorname{Wr} K))$ is not determined by the corresponding sequences for H and K. We need the sequence $(f_n(H))$ and more detailed information about K. Later, I will describe what information we actually need. Here, I will describe the situation in two particularly important examples. We have

$$f_{H \operatorname{Wr} S}(t) = \prod_{i=1}^{\infty} (1-t^i)^{-f_i(H)} = \exp\left(\sum_{j=1}^{\infty} \frac{f_H(t^j)-1}{j}\right),$$

while

$$f_{H \operatorname{Wr} A}(t) = \frac{1}{2 - f_H(t)}.$$

These relations also describe the counting functions for the compositions of classes of structures with S or A.

I will take the viewpoint that, with any oligomorphic group K, there is associated an operator (which I also denote by K) on integer sequences, so that

$$(f_n(H \operatorname{Wr} K)) = K(f_n(H)).$$

If convenient, the operator can be taken to act on generating functions. So, for example, if the sequence f counts connected graphs of some type (e.g. Eulerian graphs), then Sf counts disjoint unions of such graphs (e.g. even graphs), while Af also describes disjoint unions but where there is a total order on the set of components. Bernstein and Sloane [3] refer to the operators S and A as EULER and INVERT respectively.

There is also a *product action* of the wreath product, on the set of functions from Δ to Γ. It is not oligomorphic unless H is oligomorphic and K is a finite permutation group (that is, Δ is finite). As in the case of the direct product, I will not consider this action.

5 N-free graphs and posets

In an experiment involving a number of nuisance factors with discrete levels, the statistician needs to allow for the fact that each nuisance factor may contribute

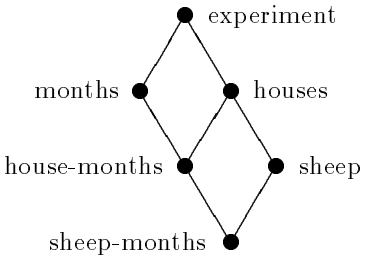


Figure 2: An experiment

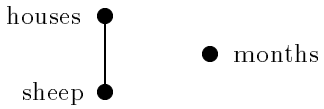


Figure 3: A poset

to the variance of responses. The relationship among these factors therefore needs to be clarified before the experiment can be designed (that is, before the assignment of treatments to experimental units can be decided). Here is an example. Suppose that we are testing various treatments on sheep. The sheep are kept in a number of houses for a number of months (a month being the period of one treatment). A single experimental unit is a sheep for a month, or a sheep-month. The relevant nuisance factors (apart from trivial ones) are houses, sheep, house-months, and months, which are partially ordered as shown in Figure 2.

This poset is a distributive lattice, and hence is representable as the lattice of ancestral sets (up-sets) in a simpler poset, formed by sheep, houses, and months, as in Figure 3.

In statistical terminology, sheep are *nested* (!) in houses, since there is no relation between the fifth sheep (say) in different houses. On the other hand, houses and months are *crossed*, since both "same house" and "same month" are potentially significant. In general, *crossing* two posets consists of taking their disjoint union, and *nesting* them to taking their ordered sum (where one is above the other). Statisticians had worked out rules for dealing with nesting and crossing and their iterates [23], but it turns out that a similar analysis can

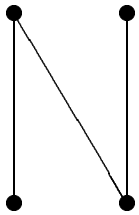


Figure 4: N

be developed for nuisance factors based on any poset (a *poset block structure*, see Speed and Bailey [36]).

Poset block structures give a large class of imprimitive association schemes whose P and Q matrices can be calculated exactly. Moreover, they are homogeneous (assuming the poset is finite; the association scheme may be finite or infinite). But my concern here is the question, posed by Bailey [1]: How typical are structures obtained by nesting and crossing? In particular, how many posets are obtained in this way, and how does this number compare to the total number of posets?

The symbol N will denote the graph or the poset which is shown in Figure 4. A graph or poset is called *N-free* if it doesn't contain N as an induced substructure. The class of N-free graphs has been studied in many contexts, under many different names. I summarise the main facts.

- The complement of an N-free graph is N-free.
- An N-free graph with more than one vertex is connected if and only if its complement is disconnected.
- The class of N-free graphs is the smallest class containing the one-vertex graph and closed under complementation and disjoint union.
- The edges of an N-free graph can be oriented to form an N-free poset.
- A poset is N-free if and only if it can be built from the one-element poset by nesting and crossing.

We see that, for $n > 1$, the numbers of connected and disconnected N-free graphs on n vertices are equal. Let a be the sequence enumerating connected N-free graphs. Then we have

$$a_1 = 1, \qquad (Sa)_n = 2a_n \quad \text{for } n > 1.$$

This gives a recurrence relation for a_n, since $(Sa)_n$ is equal to a_n plus terms involving a_i for $i < n$; so the numbers are easily calculated. It is not an easy recurrence to solve, but it can be shown that the sequence grows exponentially. The number a_n is a lower bound for the number of N-free posets.

We "bracket" the number of N-free posets as follows. An *N-free biposet* is a set supporting two posets, which are complementary (in the sense that any two distinct points are comparable in exactly one of the posets) and both N-free. Any N-free graph and its complement can be oriented to form an N-free biposet. (*Exercise:* show that, if we set $x < y$ when this relation holds in either poset of an N-free biposet, the result is a total order.) Given the order $1 < \cdots < m$ and biposets $B_1, \ldots, B_m$, we can combine them to get a new biposet B whose diconnected poset is the disjoint union of the connected posets of the B_i and whose connected poset is the ordered sum of the disconnected posets of the B_i. Hence, if b is the sequence enumerating N-free biposets for which the first poset is connected, then the total number of N-free biposets is $2b_n$ for $n > 1$, and we have

$$b_1 = 1, \qquad (Ab)_n = 2b_n \quad \text{for } n > 1.$$

This also gives a recurrence which implies that b_n grows exponentially. This recurrence can be solved explicitly: if $b(t)$ is the generating function, and $u(t) = b(t) - 1$ (so that $u(0) = 0$), we have

$$1/(1-u) = 1 + 2u - t,$$

giving $u = \frac{1}{4}(1 + t - \sqrt{1 - 6t + t^2})$. The Binomial Theorem now gives a formula for the coefficients. The function u has a singularity at $t = 3 - 2\sqrt{2}$, so this is its radius of convergence, and the exponential constant is $3 + 2\sqrt{2}$.

Now let c and d be the sequences enumerating connected and disconnected N-free posets, where we use the strange convention that $c_1 = d_1 = 1$. This case is a curious mixture of the two preceding. Since any disconnected N-free poset is a disjoint union of connected ones, and any connected N-free poset (on more than one element) an ordered sum of disconnected ones, we get the mutual recurrence

$$c_1 = d_1 = 1, \qquad (Sc)_n = (Ad)_n = c_n + d_n \quad \text{for } n > 1.$$

This enables the sequences to be calculated. They grow exponentially, with exponential constant approximately 4.62 (see Cameron [10] for more precise asymptotics). If $c(t)$ and $d(t)$ are the generating functions of the sequences, then

$$c(t) + d(t) - t - 1 = \frac{1}{2 - d(t)} = \prod_{i=1}^{\infty} (1 - t^i)^{-c_i}.$$

In any case, we have more than enough information to answer the motivating question. Since there are roughly $2^{n^2/4}$ posets altogether (indeed, this many two-level posets), only a vanishingly small proportion of them are obtained by nesting and crossing.

6 Algebraic interlude

There is a graded algebra which can be constructed from a permutation group, such that the dimensions of its homogeneous components are the numbers of orbits of the group on n-sets. Its algebraic structure can give a bit more insight into the combinatorics of the orbits.

For any infinite set Ω, let V_n denote the set of all functions from $\binom{\Omega}{n}$ (the set of n-element subsets of Ω) to your favourite field of characteristic zero (which I will take to be the rational numbers here). Each V_n is a rational vector space, and V_0 has dimension 1 (there is only one empty set). Now let

$$\mathcal{A} = \bigoplus_{n=0}^{\infty} V_n$$

be the direct sum of these spaces. We define a multiplication on $\mathcal{A}$ by the rule that, for any $f \in V_k$, $g \in V_l$, the product fg is the function in V_{k+l} defined by

$$fg(M) = \sum_{K \in \binom{M}{k}} f(K)g(M \setminus K)$$

for any $(k+l)$-set M. This makes $\mathcal{A}$ a commutative, associative, graded algebra over $\mathbb{Q}$. (It is in fact the reduced incidence algebra of the poset of finite subsets of Ω, but this fact plays no role here. I also remark that Glynn [14] has made use of a similar algebra, where the supports of the k-set and l-set to which f and g are applied in defining the product are not required to be disjoint. This algebra has very different properties. Glynn uses it to study reconstruction problems.)

An element of V_n is called a *homogeneous element of degree n* in the algebra $\mathcal{A}$. (This has no connection with our earlier usage of the word "homogeneous".) A particular homogeneous element of degree 1 is the constant function e with value 1. Multiplication by e induces a linear map from V_n to V_{n+1} for each n; this map is represented by the matrix P of Section 2, and Theorem 3.1 implies that it is a non-zero-divisor.

Now let G be a permutation group on Ω. Then G acts on each space V_n, by permuting the arguments of the functions. Let V_n^G be the space of functions in V_n fixed by G. Since a function is fixed by G if and only if it is constant on the orbits of G, we have

$$\dim(V_n^G) = f_n(G)$$

if G is oligomorphic. Furthermore, we define

$$\mathcal{A}^G = \sum_{n=0}^{\infty} V_n^G$$

to be the set of fixed points of G in $\mathcal{A}$. If G fixes two functions, it fixes their product; so $\mathcal{A}^G$ is a subalgebra of $\mathcal{A}$. For oligomorphic groups G, we see that

the generating function $f_G(t)$ is the Poincaré series of $\mathcal{A}^G$. In particular, if S is the symmetric group on Ω, then $\mathcal{A}^S$ is the polynomial algebra in one variable over $\mathbb{Q}$, the generator being the element e defined above.

If G is oligomorphic, then V_n^G is spanned by the characteristic functions of the G-orbits on n-sets; each orbit corresponds to an isomorphism type of n-element structures in the Fraïssé class of G. According to our philosophy, it is possible to define an analogous algebra for more general classes of finite structures. I leave it as an exercise to write out the precise definition of this algebra.

We now consider the structure of $\mathcal{A}^G$ when G is a direct or wreath product. The direct product is straightforward: we have

$$\mathcal{A}^{H\times K} = \mathcal{A}^H \otimes_{\mathbb{Q}} \mathcal{A}^K.$$

Wreath products are more difficult, but there are results in some special cases. First, let $G = S \operatorname{Wr} K$. If K is a finite permutation group on a set of size n, then it can be represented as a group of $n \times n$ matrices (using permutation matrices corresponding to the elements of K). Such a linear group K has a ring $I(K)$ of invariants, the polynomial functions on $\mathbb{Q}^n$ fixed by K. It turns out that $\mathcal{A}^{S \operatorname{Wr} K}$ is isomorphic to $I(K)$. In particular, the generating function $f_{S \operatorname{Wr} K}(t)$ is the *Molien series* [22] of the linear group K. If K is the symmetric group S_n then, by Newton's Theorem, $I(K)$ is a polynomial ring generated by the elementary symmetric functions, which have degrees $1, 2, \ldots, n$; and we have

$$f_{S \operatorname{Wr} S_n}(t) = \prod_{i=1}^{n} (1-t^i)^{-1}.$$

There is a completely different situation in which we can guarantee that $\mathcal{A}^G$ is a polynomial ring generated by homogeneous elements. Suppose that G is the automorphism group of a homogeneous structure, whose Fraïssé class has a "good notion of connectedness". (I will not define this precisely. It holds for graphs, etc. In general, what is required is that every structure can be uniquely expressed as the disjoint union of connected structures, and that given an arbitrary structure and a partition of its points, the structure "contains" (as a substructure) the disjoint union of the induced substructures on its parts.) Then it can be shown that $\mathcal{A}^G$ is a polynomial algebra. Its generators are in one-to-one correspondence with the connected structures.

Now another interpretation of the S-transform is that, if a sequence f enumerates the number of polynomial generators of given degree in a polynomial algebra, then the nth term of Sf is the degree of the nth homogeneous component of the algebra. So the relation between connected and arbitrary structures is exactly mirrored in the algebra.

A special case occurs for the group $H \operatorname{Wr} S$. Recall that a structure in the Fraïssé class of this group consists of a set with a partition, having a structure

in the Fraïssé class of H on each part of the partition. Taking the connected structures as those with just one part, we have a "good notion of connectedness"; so $\mathcal{A}^{H \,\mathrm{Wr}\, S}$ is a polynomial algebra with $f_n(H)$ generators of degree n for each n. Note that the structure of $\mathcal{A}^{H \,\mathrm{Wr}\, S}$ does not depend on the detailed structure of $\mathcal{A}^H$, only on its Poincaré series.

I end this section with a puzzle. There is a countable homogeneous two-graph, since finite two-graphs form a Fraïssé class. Let G be its automorphism group, and consider $\mathcal{A}^G$. Is it a polynomial algebra? The answer is not known. If it is, then the number of polynomial generators of degree n is equal to the number of Eulerian graphs on n vertices. Also, how do reduced two-graphs fit into the picture?

The general pattern of this puzzle is a group G for which the sequence $(a_n) = S^{-1}(f_n(G))$ has a natural combinatorial interpretation; we want to know whether $\mathcal{A}^G$ is a polynomial algebra with generators enumerated by (a_n).

Here is an example where this approach succeeded, and connected the theory here with a very different part of mathematics. Let q be a positive integer. It is known that there is a partition of the set of rational numbers into q disjoint dense subsets $S_1, \ldots, S_q$, and that any two such partitions are related by an order-preserving permutation. Let $G(q)$ be the group of permutations of $\mathbb{Q}$ which preserve the order and the subsets $S_1, \ldots, S_q$. An orbit of $G(q)$ on n-sets is specified by the word $x_1 \ldots x_n$ in the alphabet $A = \{1, \ldots, q\}$, where x_i is the index of the set containing the i^{th} point of the n-set (in the order induced by $\mathbb{Q}$). Every word of length n is realised; so $f_n(G(q)) = q^n$.

Now $\mathcal{A}^{G(q)}$ is the algebra spanned by the set A^* of all words in the alphabet A; multiplication of two words is given by the sum of all words obtained by "shuffling" them together. For example, using $\{a, b\}$ instead of $\{1, 2\}$ for the alphabet, we have

$$(ab) \cdot (aab) = abaab + 3aabab + 6aaabb.$$

This is the *shuffle algebra*, which arises in the theory of free Lie algebras (see Reutenauer [28]). It was proved by Radford [26] that the shuffle algebra on a given alphabet is a polynomial algebra generated by the *Lyndon words*. In order to explain these, we assume that the alphabet A is totally ordered, and take the lexicographic order on the words. Now a *Lyndon word* is a word which is smaller (in this order) than any proper cyclic shift of itself; that is, w is a Lyndon word if, whenever $w = xy$ is a proper factorisation, we have $w < yx$. Now the combinatorial assertions required for Radford's theorem are the following:

(a) any word has a unique expression as a concatenation $w_1 w_2 \ldots w_n$, where $w_1, \ldots, w_n$ are Lyndon words and $w_1 \geq w_2 \geq \ldots \geq w_n$;

(b) of all the words which can be obtained by shuffling Lyndon words $w_1, \ldots, w_n$ together, the lexicographically greatest is the concatenation in non-increasing order.

Now we take the "connected" words to be the Lyndon words, and the relation of "involvement" to be lexicographic order reversed; and this result fits into the previous formalism.

Note that the number of Lyndon words of length n is $\frac{1}{n}\sum_{d|n}\mu(d)q^{n/d}$, where μ is the Möbius function. This is a well-known expression, which also counts (among other things) the number of monic irreducible polynomials of degree n over the finite field of order q, if q is a prime power. But that is another story (see Bailey *et al.* [2]).

7 Reconstruction

The algebraic considerations of the last section are also related to the vertex reconstruction conjecture for graphs. Viewed in this way, we have a reconstruction problem for the age of any oligomorphic group. The details differ greatly from one class to another.

Let G be the automorphism group of the random graph, so that the Fraïssé class of G is the class of all finite graphs. We can regard the vector space V_n as having a basis which consists of the isomorphism types of n-vertex graphs. Let $T_{n,n-1}$ be the linear map from V_n to V_{n-1} which takes each n-vertex graph to the sum of its $(n-1)$-vertex induced subgraphs. Then $T_{n,n-1}$ is the map represented by the matrix M of Section 2; its dual is the map $T_{n-1,n}$ from V_{n-1} to V_n induced by multiplication by the element e of the preceding section, with matrix P as in Section 2.

Now two n-vertex graphs are *hypomorphic* if they have the same deck of vertex-deleted subgraphs; that is, if their images under $T_{n,n-1}$ are equal. So if X and Y are hypomorphic, then $X - Y \in \ker(T_{n,n-1})$. Moreover, for any X and Y, if $aX + bY \in \ker(T_{n,n-1})$, with $ab \neq 0$, then $b = -a$, and X and Y are hypomorphic.

So the *vertex reconstruction conjecture* for graphs can be stated in the form: *For $n > 2$, the kernel of $T_{n,n-1}$ has minimum weight greater than 2.* (The *minimum weight* of a subspace, as in coding theory, is the smallest number of non-zero coordinates of a non-zero vector in that subspace.)

We could thus ask the question: *What is the minimum weight of* $\ker(T_{n,n-1})$*?* For example, a trivial upper bound for the minimum weight is $1 + n/2$ if n is even. For, if $X_{n,k}$ is the graph with n vertices and k disjoint edges, then

$$\langle X_{n,0}, X_{n,1}, \ldots, X_{n,n/2}\rangle T_{n,n-1} \subseteq \langle X_{n-1,0}, X_{n-1,1}, \ldots, X_{n-1,n/2-1}\rangle.$$

So some non-zero element in $\langle X_{n,0}, \ldots, X_{n,n/2}\rangle$ belongs to the kernel of $T_{n,n-1}$. This can surely be improved; but is the minimum weight bounded by an absolute constant?

We can generalise further, and ask: *What is the minimum weight of* $\ker(T_{n,m})$ *for $m < n$?* (We define $T_{n,m}$ to be the linear map taking an n-vertex graph to

the sum of its m-vertex subgraphs.) Since

$$T_{n,l}T_{l,m} = \binom{n-m}{l-m} T_{n,m}$$

for $m < l < n$, the minimum weight of $\ker(T_{n,m})$ decreases as m decreases. *Is there an absolute constant k such that* $\ker(T_{n,n-k})$ *has minimum weight* 2 *for all n?*

Two further generalisations suggest themselves. First, what happens if we work instead over a field of non-zero characteristic p (such as the integers mod p)? If p divides n, then $\ker(T_{n,n-1})$ has minimum weight 1: any graph with all its vertex-deleted subgraphs isomorphic belongs to the kernel (for example, any vertex-transitive graph).

Second, these questions can be posed for other Fraïssé (or more general) classes of structures. As an example, consider strings of length n over a binary alphabet $\{a, b\}$. As earlier, we consider these as sets with a total order whose elements are partitioned into two distinguished subsets. So a substructure is a (not necessarily consecutive) substring. The class of such strings is the Fraïssé class of the group $G(2)$ of order-preserving permutations of $\mathbb{Q}$ which fix two complementary dense subsets.

Now $T_{n,m}$ maps a string to the sum of its m-element substrings, counted with multiplicities. Call two strings u and v *m-equivalent* if they have the same image; that is, if each string of length m has the same multiplicity in u and v. (This can be extended to strings of length less than m by defining such a string to be m-equivalent only to itself.) For example, the strings $X = abbbaab$ and $Y = baabbba$ of length 7 are 3-equivalent, since $T_{7,3}$ maps both X and Y to

$$aaa + 3aab + 6aba + 6abb + 3baa + 6bab + 6bba + 4bbb.$$

Now the obvious question is: *What is the smallest n, as a function of m, for which there are two m-equivalent binary strings of length n?* The answer is not known, and the known upper and lower bounds are very far apart. John Dixon [11] proved a result characterising m-equivalence in purely algebraic terms. He showed that two strings are m-equivalent if and only if, when regarded as words in the generators of the free nilpotent group of class m, they are equal.

The *edge reconstruction conjecture* for graphs can be fitted into this formalism to some extent as well. Let G be the symmetric group on an infinite set (say $\mathbb{N}$), in its induced action on the set $\Omega = \binom{\mathbb{N}}{2}$ of 2-element subsets of $\mathbb{N}$. Now an n-element member of the Fraïssé class of G consists of a graph with n edges (in other words, an n-vertex graph which is a line graph, in a specified way: so the triangle counts twice, according as it is the line graph of a triangle or of a star). The edge-reconstruction conjecture asserts that $\ker(T_{n,n-1})$ has minimum weight greater than 2 in this class, provided that $n > 3$. Questions like those posed earlier for vertex-reconstruction can now be asked.

There are further links between edge-reconstruction and finite permutation groups; but that is another story.

8 Cycle index

Now we come to the rule for calculating the sequence operator corresponding to any oligomorphic group. We will also see how to count orbits on ordered n-tuples of distinct elements (which amounts to the same thing as enumerating labelled structures in the Fraïssé class of the group).

We begin with a little Pólya theory. Let Ω be a finite set of size n. For any permutation g of Ω, we define the *cycle index* $z(g)$ of g to be $s_1^{c_1(g)}s_2^{c_2(g)}\cdots s_n^{c_n(g)}$, where $s_1, s_2, \ldots, s_n$ are independent indeterminates, and $c_i(g)$ is the number of cycles of length i in the cycle decomposition of g. If G is a permutation group on Ω, the *cycle index* of G is the average of the cycle indices of its elements:

$$Z(G) = \frac{1}{|G|}\sum_{g\in G} z(g).$$

The role of the cycle index in enumeration problems is well-known.

Clearly it is impossible to define the cycle index of an infinite group by anything like this formula; so we adopt a different approach. Let G be oligomorphic. Choose representatives for the orbits of G on finite subsets of Ω. For each such representative Δ, let $H(\Delta)$ be the group induced on Δ by its setwise stabiliser in G. Now define the *modified cycle index* $\tilde{Z}(G)$ of G to be

$$\tilde{Z}(G) = \sum_{\Delta} Z(H(\Delta)),$$

where the sum is over the orbit representatives. This is meaningful, since by assumption there are only finitely many orbits of size n, and hence a monomial of weight n occurs only finitely many times in the sum (where the weight of $s_1^{c_1}s_2^{c_2}\cdots s_n^{c_n}$ is defined to be $c_1 + 2c_2 + \cdots + nc_n$).

This procedure is meaningful for finite groups G, but it gives nothing new: in fact, for a finite group G, $\tilde{Z}(G)$ is obtained from $Z(G)$ by the substitution replacing s_i by $s_i + 1$ for all i. (For experts in Pólya theory, this is an exercise.)

I now list three pairs of facts about the modified cycle index: first, its values for the groups S and A; second, its behaviour under taking direct and wreath products; and third, a couple of interesting specialisations of it. First, another definition. If G is oligomorphic on Ω, we let $F_n(G)$ be the number of G-orbits on n-tuples of distinct elements of Ω. The finiteness of this number for all n is equivalent to the oligomorphy of G; indeed, we have

$$f_n \le F_n \le n!f_n$$

for all n. If G is the automorphism group of a homogeneous relational structure X, then $F_n(G)$ is the number of labelled n-element structures in the Fraïssé class (that is, the number of structures on the set $\{1, 2, \ldots, n\}$ which are embeddable in X). As standard in enumeration theory, we describe the sequence (F_n) by an *exponential generating function* given by

$$F_G(t) = \sum_{n=0}^{\infty} \frac{F_n(G)t^n}{n!}.$$

- $\tilde{Z}(S) = \exp\left(\sum_{j=1}^{\infty} \frac{s_j}{j}\right)$.
- $\tilde{Z}(A) = \frac{1}{1-s_1}$.
- $\tilde{Z}(H \times K) = \tilde{Z}(H)\tilde{Z}(K)$.
- $\tilde{Z}(H \operatorname{Wr} K)$ is obtained from $\tilde{Z}(K)$ by substituting $\tilde{Z}(H)(s_i, s_{2i}, \ldots) - 1$ for s_i, for $i = 1, 2, \ldots$.
- $f_G(t)$ is obtained from $\tilde{Z}(G)$ by substituting t^i for s_i for $i = 1, 2, \ldots$.
- $F_G(t)$ is obtained from $\tilde{Z}(G)$ by substituting t for s_1 and 0 for s_i for $i = 2, 3, \ldots$.

It follows from the direct product rule and the two specialisations that, as well as $f_{H\times K}(t) = f_H(t)f_K(t)$, we also have $F_{H\times K}(t) = F_H(t)F_K(t)$. But, because these are exponential generating functions, the convolution rule for sequences is a little different, namely

$$F_n(H \times K) = \sum_{k=0}^{n} \binom{n}{k} F_k(H)F_{n-k}(K).$$

This is the so-called *exponential convolution*.

The fifth of the six points gives us the rule for calculating the sequence $(f_n(H \operatorname{Wr} K))$ from $(f_n(H))$: $f_{H \operatorname{Wr} K}(t)$ is obtained from $\tilde{Z}(K)$ by substituting $f_H(t^i) - 1$ for s_i, for $i = 1, 2, \ldots$. We see that the information about K we require is its modified cycle index. Accordingly, for any oligomorphic group K, we can define an operator K on sequences by using this rule, so that

$$K(f_n(H)) = (f_n(H \operatorname{Wr} K)).$$

In a similar way, wreath products define operators on the sequences $(F_n(H))$. These operators are much easier to work with, since they are just given by

substitution in the exponential generating functions, after first removing the constant term:

$$F_{H \operatorname{Wr} K}(t) = F_K(F_H(t) - 1).$$

The most famous case of this occurs when H is the symmetric group S. We have $F_S(t) = \exp(t)$, and $F_{S \operatorname{Wr} K}(t) = F_K(\exp(t) - 1)$. In particular, $F_{S \operatorname{Wr} S}(t) = \exp(\exp(t) - 1)$, the exponential generating function for the sequence of *Bell numbers*. (The nth Bell number counts partitions of an n-set, that is, Fraïssé structures for the group $S \operatorname{Wr} S$.) This operation has another interpretation. If $F_n^*(G)$ denotes the number of orbits of G on all n-tuples (of not necessarily distinct elements), then we have

$$F_n^*(G) = F_n(S \operatorname{Wr} G),$$

as can be seen by replacing identical points of Δ in an n-tuple (where G acts on Δ) by distinct points of the fibre over that point. Furthermore, this relation is equivalent to

$$F_n^*(G) = \sum_{k=1}^{n} S(n,k) F_k(G),$$

where $S(n,k)$ is the *Stirling number of the second kind*, the number of partitions of an n-set into k parts. The operator on sequences given by the above formula is called STIRLING by Bernstein and Sloane [3].

"Dual" to this operator, in some sense, is the operator which maps $(F_n(G))$ to $(F_n(G \operatorname{Wr} S))$, given by $F_{G \operatorname{Wr} S}(t) = \exp(F_G(t) - 1)$. This operator, referred to as EXP in [3], maps the sequence enumerating labelled connected structures in some class to arbitrary labelled structures in the class; the same job that S (or EULER) does for the unlabelled structures. Explicitly, it is given by the recurrence

$$A_n = \sum_{k=1}^{n} \binom{n-1}{k-1} C_k A_{n-k},$$

where $(C_n) = (F_n(G))$ counts connected objects and $(A_n) = (F_n(G \operatorname{Wr} S))$ counts arbitrary ones.

9 A product identity

This section contains a proof of the identity

$$\mathrm{e}^{t/(1-t)} = \prod_{n=1}^{\infty} (1 - t^n)^{-\phi(n)/n},$$

where ϕ is Euler's totient function. We need another example of an oligomorphic group.

Let C be the group of all permutations preserving the cyclic order on the complex roots of unity. (The cyclic order is a ternary relation R which holds for (x,y,z) when the points are visited in this order starting at x and proceeding in an anticlockwise sense around the circle; so, if $R(x,y,z)$ holds, then $R(y,z,x)$ holds but $R(x,z,y)$ doesn't.) The group C is transitive, and the stabiliser of a point preserves a linear order on the remaining points; so the stabiliser is isomorphic to A. Using this fact, or by showing that the relational structure is homogeneous (much as we did for A earlier), we see that C has just one orbit on n-sets for every $n > 0$, and the stabiliser of an n-set induces on it the cyclic group C_n of order n.

Now C_n contains $\phi(d)$ elements of order d for each divisor d of n; and each of these elements has n/d cycles of length d. So we have

$$\begin{aligned}
\tilde{Z}(C) &= 1 + \sum_{n=1}^{\infty} \frac{1}{n} \sum_{d|n} \phi(d) s_d^{n/d} \\
&= 1 + \sum_{d=1}^{\infty} \frac{\phi(d)}{d} \sum_{m=1}^{\infty} \frac{s_d^m}{m} \\
&= 1 - \sum_{d=1}^{\infty} \frac{\phi(d)}{d} \log(1 - s_d).
\end{aligned}$$

Since $f_n(C) = 1$ for all n, we have $f_C(t) = 1/(1-t) = 1 + t/(1-t)$. Hence

$$1 + \frac{t}{1-t} = 1 - \sum_{d=1}^{\infty} (\phi(d)/d) \log(1 - t^d).$$

Now subtracting 1 from each side, taking the exponential, and replacing the dummy variable d by n gives the result.

Note that, having worked out $\tilde{Z}(C)$, we can write down the sequence operator corresponding to C, in terms of its action on generating functions:

$$(Cf)(t) = 1 - \sum_{n=1}^{\infty} \frac{\phi(n)}{n} \log(2 - f(t^n)).$$

Having added C to our repertoire, it is interesting to consider the group $C \operatorname{Wr} S$. A member of the Fraïssé class for it consists of a set carrying a partition with a circular order on each part. This is precisely the specification of a permutation, decomposed into disjoint cycles. So the group $C \operatorname{Wr} S$ "represents" permutations.

The numbers of permutations and of total orders on an n-set are both equal to $n!$. So there should be some relation between $C \operatorname{Wr} S$ and A. However, the bijection between linear orders and permutations is not a "natural" one:

we must first choose a distinguished order λ, and then any other order is a permutation of λ.

We know already that $\tilde{Z}(A) = 1/(1-s_1)$. A straightforward calculation, using the value of $\tilde{Z}(C)$ found above, shows that $\tilde{Z}(C \operatorname{Wr} S) = \prod_{n\geq 1}(1-s_n)^{-1}$. These two expressions are different; but, to compute the e.g.f. for the number of labelled structures, we substitute t for s_1 and 0 for s_n $(n > 1)$; the results are the same, as they should be:

$$F_A(t) = F_{C \operatorname{Wr} S}(t) = (1-t)^{-1}.$$

10 Stirling numbers

We already saw that Stirling numbers are involved with the formalism of wreath products. It is possible to define and generalise them using this philosophy.

I begin with a brief course on Stirling numbers. The *Stirling number of the first kind*, $S(n,k)$, is the number of partitions of an n-set into k parts. We see immediately that the sum $\sum_{k=1}^{n} S(n,k) = B(n)$ (the *Bell number*) is the total number of partitions of an n-set, which we recognise as $F_n(S \operatorname{Wr} S)$.

The *unsigned Stirling number of the second kind*, $s(n,k)$, is the number of permutations of an n-set with k disjoint cycles. Thus we have $\sum_{k=1}^{n} s(n,k) = n! = F_n(A)$. It is more useful to re-interpret this in the light of the remarks in the last section. A permutation with k cycles is given by a partition into k parts with a cyclic order on each part; and we have $\sum_{k=1}^{n} s(n,k) = F_n(C \operatorname{Wr} S)$.

This immediately suggests a generalisation. Let G be any oligomorphic permutation group. We define the *generalised Stirling number* $S[G](n,k)$ to be the number of partitions of an n-set into k parts, with a member of the Fraïssé class for G on each part. Thus we have $\sum_{k=1}^{n} S[G](n,k) = F_n(G \operatorname{Wr} S)$. In this notation, the "classical" Stirling numbers are $S(n,k) = S[S](n,k)$ and $s(n,k) = S[C](n,k)$.

It is clear that the generalised Stirling numbers $S[G](n,k)$ are determined by the numbers $F_n(G)$. This can be expressed most concisely in terms of the exponential generating functions:

$$\sum_{n=k}^{\infty} S[G](n,k)t^n/n! = (F_G(t)-1)^k/k!.$$

From this, the equation $F_{G \operatorname{Wr} S}(t) = \exp(F_G(t)-1)$ is obtained by summing over k.

The generalised Stirling numbers have a composition property:

$$\sum_{l=k}^{n} S[G](n,l)S[H](l,k) = S[G \operatorname{Wr} H](n,k).$$

For consider $S[G](n,l)S[H](l,k)$. This counts pairs consisting of a partition of $\{1,\ldots,n\}$ into l parts with a G-structure on each part, and a partition of the set of parts into k parts with an H-structure on each part. (Here "G-structure" is short for "member of the Fraïssé class of G".) Viewed otherwise, we have a partition of $\{1,\ldots,n\}$ into k parts, each part carrying a partition into "subparts" with a G-structure on each subpart and an H-structure on the set of subparts (in other words, a $G\,\mathrm{Wr}\,H$-structure), subject to the condition that there are l subparts altogether. Summing over l removes the final condition and yields $S[G\,\mathrm{Wr}\,H](n,k)$.

This result can be expressed more compactly in matrix form. Let $T[G]$ be the triangular array of generalised Stirling numbers associated with G, the infinite lower triangular mtrix with (n,k) entry $S[G](n,k)$. Then we have

$$T[G]T[H] = T[G\,\mathrm{Wr}\,H].$$

For example, $T[S]$ and $T[C]$ are the arrays of classical Stirling numbers; and we have

$$T[C]T[S] = T[C\,\mathrm{Wr}\,S] = T[A].$$

The numbers $S[A](n,k)$ are the *Lah numbers* $L(n,k)$, sometimes called "Stirling numbers of the third kind": see Lah [17], Bridgeman [5]. Unlike the classical Stirling numbers, there is a closed formula for the Lah numbers:

$$L(n,k) = \frac{(n-1)!}{(k-1)!}\binom{n}{k} = \frac{n!}{k!}\binom{n-1}{k-1}.$$

This can be shown by using the formula

$$\sum_{n\geq k} L(n,k)t^n/n! = \left(\frac{t}{1-t}\right)^k/k!$$

and computing the coefficient of t^n on the right-hand side.

In a similar manner, it can be shown that

$$\sum_{k=1}^{n} S[G](n,k)F_k(H) = F_n(G\,\mathrm{Wr}\,H).$$

This property generalises the STIRLING transform we met earlier.

There is another remarkable property of classical Stirling and Lah numbers. Let $S^*[G](n,k) = (-1)^{n-k}S[G](n,k)$ be the *signed* generalised Stirling numbers, and let $T^*[G]$ be the corresponding triangular array. Then

$$\sum_{l=k}^{n} S(n,l)(-1)^{l-k}s(l,k) = \delta_{nk},$$

or in other words

$$T[S]T^*[C] = I.$$

It follows that also $T[C]T^*[S] = I$ and $T[A]T^*[A] = I$. I do not know whether this inversion relation has analogues for other groups.

11 Stabilisers and derivatives

We've seen that the group-theoretic operations of direct and wreath product "correspond" to multiplication and composition of formal power series. It is possible to interpret differentiation in similar terms. In this section, I assume that the permutation group G is transitive on Ω, though it is possible to formulate the results more generally.

The *stabiliser* G_α of the point $\alpha \in \Omega$ is the subgroup of G consisting of the permutations which fix α. We consider it as a permutation group on $\Omega \setminus \{\alpha\}$. Now we have

$$\tilde{Z}(G_\alpha) = \frac{\partial}{\partial s_1}\tilde{Z}(G).$$

It follows that

$$F_{G_\alpha}(t) = \frac{\mathrm{d}}{\mathrm{d}t}F_G(t).$$

(In fact, it is easy to see this directly. Differentiating an exponential generating function corresponds to shifting the terms of the sequence one place to the left, so the preceding equation says

$$F_n(G_\alpha) = F_{n+1}(G).$$

The correspondence between orbits of G_α on n-tuples and of G on $(n+1)$-tuples can be described thus: take an orbit of G on $(n+1)$-tuples, select all the tuples which begin with α, and delete α from them.)

On the other hand, the sequence $(f_n(G_\alpha))$ is not determined by $(f_n(G))$.

The Fraïssé class for G_α is obtained from that for G by distinguishing a point x in each finite substructure and deleting x. (This is not the same as just deleting a point, since it leaves a shadow, the extra structure obtained when x was distinguished. For example, if the objects in the Fraïssé class are graphs, then by distinguishing and deleting x we specify a subset of the remaining vertices, those which were joined to x.) In view of the effect on the generating function, I will denote this operation on Fraïssé classes by ∂.

Two-graphs provide an example (see Seidel [31]). If x is a point of the two-graph (X, T), there is a unique graph in the corresponding switching class with the property that x is an isolated vertex. Thus, if Gr and $TwoGr$ denote the classes of graphs and two-graphs, we have

$$Gr = \partial TwoGr.$$

In combinatorial terms, it is more natural to leave the point x in, obtaining a "rooted" structure. This is easily handled: adding the fixed point back in corresponds to taking the direct product of G_α with the trivial group acting on a single point, whose modified cycle index is $1 + s_1$.

Having defined derivatives, we can consider differential equations. For example, is there a group G for which $G_\alpha \cong G \times G$? For such a group, the function $F = F_G$ satisfies $F' = F^2$, $F(0) = 1$, with solution $F(t) = (1-t)^{-1}$. Thus $F_n(G) = n!$. This sequence is the same as the one realised by the group A. Indeed, the stabiliser of 0 in A has two orbits, the positive and the negative rationals; each orbit, as ordered set, is isomorphic to $\mathbb{Q}$, and A_0 induces all order-preserving permutations on each. So indeed $G = A$ satisfies the original equation. (The fact that $\partial A = A \times A$, where A is the class of finite total orders, can be regarded as the basis for the recursive QUICKSORT algorithm [15] for sorting a list: select an element 0, partition the list into elements before and after 0, and sort these two sublists.)

The group $G = C \operatorname{Wr} S$ also satisfies $F_n(G) = n!$, corresponding combinatorially to the fact that any permutation can be decomposed into a disjoint union of cycles. This group, like A itself, satisfies the related equation $G_\alpha \cong A \times G$.

What about the differential equation $G_\alpha = G \operatorname{Wr} G$? It can be shown that no such group exists. Nevertheless, we obtain an interesting integer sequence $(F_n(G))$ for such a non-existent group. With $f(t) = F_G(t) - 1$, we have

$$f'(t) = 1 + f(f(t)), \qquad f(0) = 0,$$

somewhat reminiscent of the Feigenbaum–Cvitanović equation

$$g(t) = -\alpha g(g(t/\alpha))$$

(Feigenbaum [12]). The unique power series solution does not converge in any neighbourhood of 0. Is the a combinatorial interpretation of the coefficients (a class of structures enumerated by them)? The first few terms of the sequence are 1, 2, 7, 37, 269, 2535, 29738, 421790, 7076459, ...

12 The probability of connectedness

According to Cayley's Theorem, the number of labelled trees on n points is n^{n-2}. It is a surprising fact, proved by Rényi [27] in 1959, that the number of labelled forests on n points is asymptotic to cn^{n-2}, where $c = \sqrt{\mathrm{e}}$; that is, the probability that a random forest on $\{1, 2, \ldots, n\}$ is connected tends to $1/\sqrt{\mathrm{e}}$ as $n \to \infty$. (I am grateful to Dominic Welsh for this reference.) Moreover, for labelled forests of rooted trees, the limiting probability of connectedness is $1/\mathrm{e}$.

In terms of our earlier notation, if $C_n = n^{n-2}$ and (A_n) is the sequence obtained by applying the operator EXP to (C_n), then $\lim_{n\to\infty} A_n/C_n = \sqrt{\mathrm{e}}$. And, if we put $C_n = n^{n-1}$ instead, the limit is e.

One could ask more generally: for which classes of structures (with a notion of connectedness) is it true that the probability of connectedness for a labelled or unlabelled structure tends to a limit strictly between zero and one? A class of examples is provided by the N-free graphs. As we saw, exactly half of the N-free graphs on n points are connected if $n > 1$, and this is true for labelled or unlabelled structures, since complementation gives a bijection between connected and disconnected structures. Furthermore, it can be shown that the probability that a (labelled or unlabelled) N-free poset is connected tends to the golden ratio as the number of points tends to infinity (see [10]).

In the unlabelled case, it is easy to handle rooted trees, since the number of forests of rooted trees on n vertices is equal to the number of rooted trees on $n+1$ vertices. (Take a new root, and join it to all the old roots.) Since these numbers grow exponentially with constant 2.95576... [24], the limiting probability of connectedness is the reciprocal of this number, namely 0.33832.... It appears that exponential growth for the number of n-element unlabelled structures is necessary for the probability of connectedness to be strictly between 0 and 1, though I cannot prove such a precise result.

In terms of groups, the question becomes: for which oligomorphic groups G is it true that either $\lim_{n\to\infty} F_n(G \operatorname{Wr} S)/F_n(G)$, or $\lim_{n\to\infty} f_n(G \operatorname{Wr} S)/f_n(G)$, exists and is finite and greater than 1? Having formulated the question in this way, it immediately generalises. We can replace the group S by any oligomorphic group, take the wreath product in either order, or use direct product instead of wreath product. For more on this, see [10].

13 Two-graphs revisited

The last story, like the first, is about two-graphs, and is taken from Cameron [9], which contains all references for this section (and is available electronically).

There is a simple construction for two-graphs from trees, as follows. Let T be a tree with edge set Ω. Now let $\mathcal{T}$ consist of all triples of edges which do not lie on a path in the tree (those for which the paths connecting them in the tree form a subtree containing a trivalent vertex). It is easily verified that $(\Omega, \mathcal{T})$ is a two-graph (by considering the four possible configurations of four edges). These two-graphs arose in the work of Tsaranov [37] on a class of groups related to Coxeter groups. Which two-graphs are produced by the construction?

The *pentagon* and *hexagon* two-graphs refer to the two-graphs associated, as in the first section, with the switching classes of the pentagon and hexagon graphs respectively. In [8], I proved that a two-graph arises from a tree by the construction described if and only if it doesn't contain either the pentagon or the hexagon two-graph as an induced substructure. Moreover, non-isomorphic trees give rise to non-isomorphic two-graphs. This solves the counting problem for unlabelled pentagon- and hexagon-free two-graphs: the number on n points is equal to the number of trees with n edges, calculated by Otter [24].

However, there is a further difficulty associated with counting the labelled pentagon- and hexagon-free two-graphs. For example, a path with n edges can have its edges labelled in $n!/2$ different ways, but all of these give rise to the null two-graph (the two-graph with no triples).

The solution to the problem comes by showing that the two-graph obtained from a tree T is reduced (in the sense of the first section) if and only if the tree is *series-reduced*, that is, has no vertices of valency 2. So we should first count the series-reduced edge-labelled trees. The number of these with n edges turns out to be

$$x_n = \frac{1}{n}\sum_{j=0}^{n-1}(-1)^j\binom{n+1}{j}\binom{n-1}{j}j!(n+1-j)^{n-1-j}$$

for $n \geq 2$, with $x_1 = 1$. Then the number of labelled pentagon- and hexagon-free two-graphs is given by the STIRLING transform

$$\sum_{k=1}^{n} S(n,k)x_k.$$

We have a language to describe this behaviour. We can associate a sequence operator with a class of objects even if it is not the Fraïssé class associated with some group: define the "modified cycle index" to be the sum of the cycle indices of the automorphism groups of the unlabelled structures in the class, and then use the same formalism as described earlier. Now series-reduced trees (counted by edges) and reduced pentagon- and hexagon-free two-graphs have the same modified cycle index, because of the correspondence, and hence define the same sequence operator. If we denote this class by SRT, then the class of all pentagon- and hexagon-free two-graphs corresponds to $S\,\mathrm{Wr}\,SRT$, and the class of all trees to $A\,\mathrm{Wr}\,SRT$ apart from a slight mismatch for paths. (The edges on a path have two possible orders which cannot be distinguished, but which are counted twice by $A\,\mathrm{Wr}\,SRT$.)

The class of pentagon-free two-graphs (those containing no induced pentagon) is also interesting. It is closely connected with the class of N-free graphs; in fact, the operator ∂, applied to the class of pentagon-free two-graphs, gives the class of N-free graphs (like the relation between two-graphs and graphs). Its members can also be represented by trees (in a different way); and it can be enumerated by techniques similar to those described. This is also found in [8], [9].

End note

Jalaluddin Rumi was one of the leading Sufi poets. The story of the blind people and the elephant is common to several other religious traditions, including Quakers and Buddhists.

References

[1] R. A. Bailey, Designs: mappings between structured sets, pp. 22–51 in *Surveys in Combinatorics, 1989* (ed. J. Siemons), Cambridge Univ. Press, Cambridge, 1989.

[2] R. A. Bailey, P. J. Cameron and D. G. Fon-Der-Flaass, in preparation.

[3] M. Bernstein and N. J. A. Sloane, Some canonical sequences of integers, *Linear Algebra Appl.*, to appear.

[4] R. Brauer, On the connection between the ordinary and the modular characters of groups of finite order, *Ann. Math.* **42** (1941), 926–935.

[5] T. Bridgeman, Lah's triangle — Stirling numbers of the third kind, preprint, July 1995.

[6] P. J. Cameron, Cohomological aspects of two-graphs, *Math. Z.* **157** (1977), 101–119.

[7] P. J. Cameron, *Oligomorphic Permutation Groups*, London Math. Soc. Lecture Notes **152**, Cambridge University Press, Cambridge, 1990.

[8] P. J. Cameron, Two-graphs and trees, *Discrete Math.* **127** (1994), 63–74.

[9] P. J. Cameron, Counting two-graphs related to trees, *Electronic J. Combinatorics* **2** (1995), #R4.

[10] P. J. Cameron, On the probability of connectedness, in preparation.

[11] J. D. Dixon, personal communication (1985).

[12] M. J. Feigenbaum, Quantitative universality for a class of nonlinear transformations, *J. Statist. Phys.* **19**, 25–52.

[13] R. Fraïssé, Sur certains relations qui généralisent l'ordre des nombres rationnels, *C. R. Acad. Sci. Paris* **237** (1953), 540–542.

[14] D. Glynn, Rings of geometries, I, *J. Combinatorial Theory* (A) **44** (1987), 34–48; II, *ibid.* (A) **49** (1988), 26–66.

[15] C. A. R. Hoare, Quicksort, *Computer Journal* **5** (1962), 10–15.

[16] A. Joyal, Une théorie combinatoire des séries formelles, *Advances Math.* **42** (1981), 1–82.

[17] I. Lah, Eine neue Art von Zahlen, ihre Eigenschaften und Anweldung in der mathematischen Statistik, *Mitt. Math. Statistik* **7** (1955), 203–212.

[18] V. A. Liskovec, Enumeration of Euler graphs, *Vescī Akad. Navuk BSSR Ser. Fīz–Mat. Navuk* (1970), 38–46.

[19] D. Livingstone and A. Wagner, Transitivity of finite permutation groups on unordered sets, *Math. Z.* **90** (1965), 393–403.

[20] H. D. Macpherson, The action of an infinite permutation group on the unordered subsets of a set, *Proc. London Math. Soc.* (3) **51** (1983), 471–486.

[21] C. L. Mallows and N. J. A. Sloane, Two-graphs, switching classes, and Euler graphs are equal in number, *SIAM J. Appl. Math.* **28** (1975), 876–880.

[22] T. Molien, Über die Invarianten der lineare Substitutionsgruppe, *Sitzungsber. König. Preuss. Akad. Wiss.* (1897), 1152–1156.

[23] J. A. Nelder, The analysis of randomized experiments with orthogonal block structure, *Proceedings of the Royal Society, Series A*, **283** (1965), 147–178.

[24] R. Otter, The number of trees, *Ann. Math.* (2) **49** (1948), 583–599.

[25] M. Pouzet, Application d'une propriété combinatoire des parties d'un ensemble aux groupes et aux relations, *Math. Z.* **150** (1976), 117–134.

[26] D. E. Radford, A natural ring basis for the shuffle algebra and an application to group schemes, *J. Algebra* **58** (1979), 432–454.

[27] A. Rényi, Some remarks on the theory of trees, *Publ. Math. Inst. Hungar. Acad. Sci.* **4** (1959), 73–85.

[28] C. Reutenauer, *Free Lie Algebras*, London Math. Soc. Monographs (New Series) **7**, Oxford University Press, 1993.

[29] R. W. Robinson, Enumeration of Euler graphs, *Proof Techniques in Graph Theory* (Proc. Second Ann Arbor Graph Theory Conf., Ann Arbor 1968), 147–153, Academic Press, New York 1969.

[30] J. J. Seidel, Strongly regular graphs of L_2-type and of triangular type, *Proc. Kon, Nederl. Akad. Wetensch.* (A) **70** (1967), 188–196.

[31] J. J. Seidel, A survey of two-graphs, pp. 481–511 in *Proc. Int. Colloq. Teorie Combinatorie*, Accad. Naz. Lincei, Roma, 1977.

[32] J. J. Seidel, More about two-graphs, in *Combinatorics, Graphs and Complexity* (Proc. 4th Czech Symp., Prachatice 1990), 297–308, *Ann. Discrete Math.* **51** (1992).

[33] J. J. Seidel and D. E. Taylor, Two-graphs: A second survey, in *Algebraic Methods in Graph Theory*, Szeged, 1978.

[34] Idries Shah (ed.), *World Tales*, Harcourt Brace Jovanovich, New York, 1979.

[35] N. J. A. Sloane, *A Handbook of Integer Sequences*, Academic Press, New York, 1973.

[36] T. P. Speed and R. A. Bailey, Factorial dispersion models, *Internat. Statist. Review* **55** (1987), 261–277.

[37] S. Tsaranov, On a generalization of Coxeter groups, *Algebra Groups Geom.* **6** (1989), 281–318.

SEQUENCES FROM SQUARES OF INTEGERS

T. Aaron Gulliver
Department of Electrical and Computer Engineering
University of Victoria, P.O.Box 3055, STN CSC
Victoria, BC, Canada V8W 3P6
agullive@ece.uvic.ca

Abstract

This paper presents a number of sequences based on integers arranged in arrays. This approach provides a simple derivation of some well known sequences. In addition, a number of new integer sequences are obtained.

Keywords–integer arrays, integer sequences.
AMS Subject Classification–11Y55

1. Introduction

This paper begins with a well known combinatorial expression for the sum of the first n natural numbers

$$1 + 2 + 3 + 4 + 5 + \ldots + n = \frac{n(n+1)}{2}. \tag{1}$$

known as the triangular numbers. Arranging the values for this sum in a sequence starting from $n = 1$ gives

$$1, 3, 6, 10, 15, \ldots$$

This is sequence A000217 in the Encyclopedia of Integer Sequences maintained by Sloane [1]. One could view the components of the sums that make up this sequence as lines or 1-dimensional arrays

$$1, \quad 1\ 2, \quad 1\ 2\ 3, \quad 1\ 2\ 3\ 4, \quad 1\ 2\ 3\ 4\ 5, \quad \ldots$$

The question then arises, what sequences occur when one considers m-dimensional arrays of integers? For $m = 0$, the result is the trivial sequence

$$1, 1, 1, 1, 1, \ldots$$

More interesting is the case $m = 2$, which gives rise to two-dimensional arrays of integers. This provides connections between seemingly unrelated sequences. The case of sequences from squares is considered in the next section, followed by an investigation of triangles and hexagons.

2. Squares

A square array of integers has the following structure

$$\begin{array}{cccc} 1 & 2 & 3 & \cdots & n \\ n+1 & n+2 & n+3 & \cdots & 2n \\ \vdots & \vdots & \vdots & & \vdots \\ n^2-n+1 & n^2-n+2 & n^2-n+3 & \cdots & n^2 \end{array} \tag{2}$$

For $n = 1$ to 5, the matrices are

$$\begin{array}{c} 1 \end{array}, \quad \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array}, \quad \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array},$$

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}, \quad \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{array}.$$

One can easily see that the 0-dimensional sequence is located in the upper left hand corner, and the 1-dimensional sequence is given by the first rows. The sequence formed from the sum of the elements in the squares given by

$$s_n = 1 + 2 + 3 + 4 + 5 + \ldots + n^2 = \sum_{i=1}^{n^2} i = \frac{n^2(n^2+1)}{2}, \tag{3}$$

is

$$1, 10, 45, 136, 325, \ldots$$

The following simple sequences

$$\begin{array}{ccccccc} 1, & 2, & 3, & 4, & 5, & \ldots \\ 1, & 4, & 9, & 16, & 25, & \ldots \\ 1, & 3, & 7, & 13, & 21, & \ldots \end{array}$$

are formed from the elements in the upper right, lower right and lower left corners, respectively. The first of these is just the sequence of natural numbers (A000027)

$$1, 2, 3, 4, 5, \ldots, n, \ldots$$

the second is the sequence of squares of the natural numbers (A000290)

$$1, 4, 9, 16, 25, \ldots, n^2, \ldots$$

while the third is the sequence of central polygonal numbers (A002061)

$$1, 3, 7, 13, 21, \ldots, n^2 - n + 1, \ldots$$

all of which are well known sequences.

By considering shapes in the squares of numbers, many other sequences can be obtained. For example, the sequence of sums of the first column of numbers in each matrix is

$$1, 4, 12, 28, 55, 96, \ldots, \frac{n(n^2 - n + 2)}{2}, \ldots$$

This is sequence A006000 in [1]. which has generating function

$$\frac{(1 + 2x^2)}{(1 - x)^4}$$

The sum of the diagonal elements (upper left to lower right) gives

$$1, 5, 15, 34, 65, \ldots, \frac{n(n^2 + 1)}{2}, \ldots \tag{4}$$

which is sequence A006003, the row sums of an $n \times n$ magic square. Summing these elements with all those above the diagonal gives

$$1, 7, 26, 70, 155, \ldots, \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} in + j + 1 = \frac{n(n+1)(n^2 + n + 1)}{6}, \ldots \tag{5}$$

which is sequence A006325.

Thus far, only one new sequence has been obtained, namely (3), but considering the three other triangular shapes in the matrix results in the following new sequences. The sum of the diagonal elements and those below it gives

$$1, 8, 34, 100, 235, \ldots, \sum_{i=0}^{n-1} \sum_{j=0}^{i} in + j + 1 = \frac{n(n+1)(2n^2 - n + 2)}{6}, \ldots \tag{6}$$

The sum of the anti-diagonal elements and those below it gives

$$1, 9, 38, 110, 255, \ldots, \sum_{i=0}^{n-1} \sum_{j=0}^{i} (i+1)n - j = \frac{n(n+1)(2n^2 + 1)}{6}, \ldots \tag{7}$$

Finally, the sum of the anti-diagonal elements and those above it gives

$$1, 6, 22, 60, 135, \ldots, \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} (i+1)n - j = \frac{n(n+1)(n^2+2)}{6}, \ldots \tag{8}$$

The sum of any two of these sequences yields another sequence, in particular adding (7) and (8) gives

$$1, 15, 60, 170, 390, \ldots, \frac{n(n+1)(n^2+1)}{2}, \ldots \tag{9}$$

This sequence is equivalent to (3) + (4), as can be seen by simplifying

$$\frac{n^2(n^2+1)}{2} + \frac{n(n^2+1)}{2} = \frac{n(n+1)(n^2+1)}{2}$$

Summing just those elements that lie above the diagonal gives

$$0, 2, 11, 36, 90, \ldots, \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} in + j + 1 = \frac{n(n-1)(n^2+2)}{6}, \ldots \tag{10}$$

The sum of the elements below the diagonal results in the sequence

$$0, 3, 19, 66, 170, \ldots, \sum_{i=0}^{n-1} \sum_{j=0}^{i-1} in + j + 1 = \frac{n(n-1)(2n^2+1)}{6}, \ldots \tag{11}$$

The sum of those elements below the anti-diagonal is

$$0, 4, 23, 76, 190, \ldots, \sum_{i=0}^{n-1} \sum_{j=0}^{i-1} (i+1)n - j = \frac{n(n-1)(2n^2+n+2)}{6}, \ldots \tag{12}$$

Finally, the sum of the elements above the anti-diagonal is

$$0, 1, 7, 26, 70, \ldots, \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} (i+1)n - j = \frac{n(n-1)(n^2-n+1)}{6}, \ldots \tag{13}$$

As before, the sum of two of these sequences provides a new sequence, in particular adding (12) and (13) gives

$$0, 5, 30, 102, 260, \ldots, \frac{n(n-1)(n^2+1)}{2}, \ldots \tag{14}$$

This sequence is equivalent to (3) - (4), as can be seen by simplifying

$$\frac{n^2(n^2+1)}{2} - \frac{n(n^2+1)}{2} = \frac{n(n-1)(n^2+1)}{2}.$$

Therefore adding (9) to (14) gives

$$s_n = \frac{n(n-1)(n^2+1)}{2} + \frac{n(n+1)(n^2+1)}{2} = n^2(n^2+1) = 2\sum_{i=1}^{n^2} i.$$

Another interesting combination is (8) + (10) which has elements

$$s_n = \frac{n^2(n^2+2)}{3}, \tag{15}$$

and corresponds to sequence A014820

$$1, 8, 33, 96, \ldots$$

The expression given in [1] for these sequence elements is

$$s_n = \frac{(n^2+2n+3)(n+1)^2}{3},$$

but substituting $n = n-1$ gives the simpler expression (15). A related sequence is (7) + (11) which has elements

$$s_n = \frac{n^2(2n^2+1)}{3}$$

giving

$$1, 12, 57, 176, 425, \ldots$$

For n odd, the elements at the centre of the squares form the sequence

$$1, 5, 13, 25, \ldots, 2n(n-1)+1, \ldots \tag{16}$$

which is sequence A001844 (appropriately named the centered square numbers). If the elements that lie on the diagonal and anti-diagonal of the squares are summed, the sequence is

$$1, 10, 25, 68, 117, \ldots$$

This sequence is equal to twice (4) when n is even, but equal to twice (4) minus the centre element (given by (16)) when n is odd. Thus the n-th sequence element is

$$s_n = \begin{cases} n(n^2+1), & n \text{ even;} \\ (n-\frac{1}{2})(n^2+1), & n \text{ odd.} \end{cases}$$

The final construction for $m = 2$ begins with the four corner elements, each having a sequence which was given previously. Adding these elements together gives

$$4, 10, 20, 34, 52, \ldots, 2(n^2+1), \ldots \tag{17}$$

which is sequence A005893 (without the first element). The sum of the perimeter elements is equal to the sum of the four lines which make up the matrix edges, minus (17). The sum of the two horizontal lines is

$$n(n+1)+n^2(n-1),$$

and the sum of the two vertical lines is

$$n(n^2-n+2)+n(n-1).$$

Therefore we have the elements of the perimeter sequence

$$\begin{aligned} s_n &= n(n+1)+n^2(n-1)+n(n^2-n+2)+n(n-1)-2(n^2+1) \\ &= 2(n-1)(n^2+1). \end{aligned}$$

Returning to the sum of all array elements

$$s_n=\sum_{i=1}^{n^2} i=\frac{n^4+n^2}{2},$$

similar expressions can be obtained for larger m. For $m=3$ the sequence elements are

$$s_n=\sum_{i=1}^{n^3} i=\frac{n^3(n+1)(n^2-n+1)}{2}=\frac{n^6+n^3}{2}.$$

and for $m=4$

$$s_n=\sum_{i=1}^{n^4} i=\frac{n^4(n^4+1)}{2}=\frac{n^8+n^4}{2}.$$

For $m=5$

$$s_n=\sum_{i=1}^{n^5} i=\frac{n^5(n+1)(n^4-n^3+n^2-n+1)}{2},$$

for $m=6$

$$s_n=\sum_{i=1}^{n^6} i=\frac{n^6(n^2+1)(n^4-n^2+1)}{2},$$

and for arbitrary m

$$s_n=\sum_{i=1}^{n^m} i=\frac{n^m(n^m+1)}{2}.$$

3. Triangles

In this section, integers arranged in a triangle are considered, From the triangular numbers (1), there are obviously $\frac{n(n+1)}{2}$ elements in each triangle. These elements can be arranged in the following form

$$\begin{array}{ccccc} 1 & 2 & 3 & \cdots & n \\ & n+1 & n+2 & \cdots & 2n-1 \\ & & 2n & \cdots & 3n-3 \\ & & & & \vdots \\ & & & & \frac{n(n+1)}{2} \end{array} \tag{18}$$

which for $n = 1$ to 5, gives

$$1\ , \qquad \begin{array}{cc} 1 & 2 \\ & 3 \end{array}\ , \qquad \begin{array}{ccc} 1 & 2 & 3 \\ & 4 & 5 \\ & & 6 \end{array}\ ,$$

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ & 5 & 6 & 7 \\ & & 8 & 9 \\ & & & 10 \end{array}\ , \qquad \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ & 6 & 7 & 8 & 9 \\ & & 10 & 11 & 12 \\ & & & 13 & 14 \\ & & & & 15 \end{array}\ .$$

Alternatively, the form can be

$$\begin{array}{cccccc} 1 & & & & & \\ 2 & & 3 & & & \\ 4 & & 5 & 6 & & \\ \vdots & & \vdots & \vdots & & \\ \frac{n^2-n+2}{2} & \cdots & \cdots & \cdots & \frac{n(n+1)}{2} \end{array} \tag{19}$$

which for $n = 1$ to 5, gives

$$1\ , \qquad \begin{array}{cc} 1 & \\ 2 & 3 \end{array}\ , \qquad \begin{array}{ccc} 1 & & \\ 2 & 3 & \\ 4 & 5 & 6 \end{array}\ ,$$

$$\begin{array}{cccc} 1 & & & \\ 2 & 3 & & \\ 4 & 5 & 6 & \\ 7 & 8 & 9 & 10 \end{array}\ , \qquad \begin{array}{ccccc} 1 & & & & \\ 2 & 3 & & & \\ 4 & 5 & 6 & & \\ 7 & 8 & 9 & 10 & \\ 11 & 12 & 13 & 14 & 15 \end{array}\ .$$

The elements of the sequence formed from the sums of the triangle elements are

$$s_n = 1+2+3+4+5+\ldots+\frac{n(n+1)}{2} = \sum_{i=1}^{\frac{n(n+1)}{2}} i = \frac{n(n+1)(n^2+n+2)}{8}, \quad (20)$$

giving

$$1, 6, 21, 55, 120, \ldots$$

These are the doubly triangle numbers (A002817). The expression in [1] given for the sequence elements is

$$\frac{(n+1)(n+2)(n^2+3n+4)}{8},$$

but substituting $n = n-1$ gives (20).

The sequence formed from the lower left corner of (19) is

$$1, 2, 4, 7, 11, \ldots, \frac{n^2-n+2}{2}, \ldots$$

which is (A000124) and is closely related to the central polygonal numbers (A002061). Note that the sequence elements are given by $n(n+1)/2+1$ in [1], but substituting $n = n-1$ gives the above result.

The sequence formed from the sum of the right column of (18) is

$$1, 5, 14, 30, 55, \ldots, \frac{n(n+1)(2n+1)}{6}, \ldots$$

which are the square pyramidal numbers (A000330). The sequence formed from the sum of the diagonal elements of (18) is

$$1, 4, 11, 24, 45, \ldots, \frac{n(n^2+2)}{3}, \ldots$$

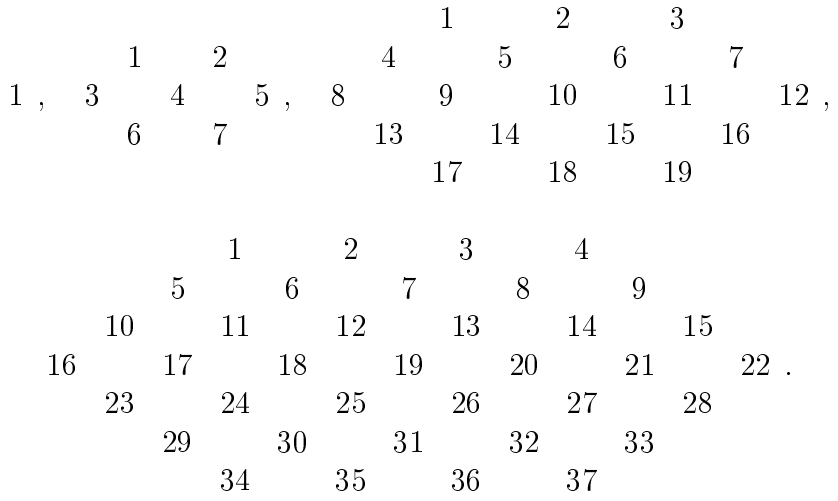
which is A006527. The sequence formed from the sum of the left column of (19) is just the sequence of lower left elements in the squares given in the previous section. The sequence formed from the sum of the diagonal elements of (19) is

$$1, 4, 10, 20, 45, \ldots, \frac{n(n+2)(n+1)}{6}, \ldots$$

which are the tetrahedral numbers (A000292).

4. Hexagons

For $n = 1$ to 4, the hexagons are



In lexicographic order, the sequences formed from the 6 corner elements are

$$1,\ 1,\ 1,\ 1,\ 1,\quad \ldots, 1, \ldots$$

$$1,\ 2,\ 3,\ 4,\ 5,\quad \ldots, n, \ldots$$

$$1,\ 3,\ 8,\ 16,\ 27,\quad \ldots, \tfrac{3n^2-5n+4}{2}, \ldots$$

$$1,\ 5,\ 12,\ 22,\ 35,\quad \ldots, \tfrac{3n^2-n}{2}, \ldots$$

$$1,\ 6,\ 17,\ 34,\ 57,\quad \ldots, 3n^2-4n+2, \ldots$$

$$1,\ 7,\ 19,\ 37,\ 61,\quad \ldots, 3n^2-3n+1, \ldots$$

The first and second sequences are trivial. The fourth sequence corresponds to the pentagonal numbers (A000326) while the last sequence corresponds to the hex numbers (A003215). It is interesting to note that

$$3n^2 - 3n + 1 = (n+1)^3 - n^3,$$

so that

$$\sum_{n=1}^{m} (3n^2 - 3n + 1) = m^3.$$

The sum of the elements on each edge of the hexagon also provide se-

quences. In lexicographic order, the sequences formed from the 6 edges are

$$1,\quad 3,\quad 6,\quad 10,\quad 15,\qquad \ldots, \tfrac{n(n+1)}{2}, \ldots$$

$$1,\quad 4,\quad 13,\quad 32,\quad 65,\quad \ldots, \tfrac{n(2n^2-3n+4)}{3}, \ldots$$

$$1,\quad 7,\quad 22,\quad 50,\quad 95,\quad \ldots, \tfrac{n(n+1)(4n-1)}{6}, \ldots$$

$$1,\quad 9,\quad 38,\quad 102,\quad 215,\quad \ldots, \tfrac{n(14n^2-21n+13)}{6}, \ldots$$

$$1,\quad 12,\quad 47,\quad 120,\quad 245,\quad \ldots, \tfrac{n(7n^2-6n+2)}{3}, \ldots$$

$$1,\quad 13,\quad 54,\quad 142,\quad 295,\quad \ldots, \tfrac{n(6n^2-7n+3)}{2}, \ldots$$

For the final sequence, consider the horizontal and two diagonal lines in the hexagons (which all have the same sum). The sum of the elements on these lines is given by

$$s_n = \frac{(2n-1)(3n^2-3n+2)}{2}$$

Acknowledgement

The author would like to thank Torrie Moore for his help in simplifying the above expressions.

References

[1] N.J.A. Sloane, On-Line Encyclopedia of Integer Sequences, http://www.research.att.com/~njas/sequences/index.html.

Unscrambling Address Lines

Andrei Broder* Michael Mitzenmacher* Laurent Moll*

Abstract

A writer leaves a message in a write-once memory accessible via address lines. Before the intended recipient has a chance to get the message, the address lines are permuted by an adversary. We provide a simple, nearly optimal algorithm for the reader and writer to communicate over such a channel.

This problem arose in the context of FPGA hardware design. Our algorithm has been implemented and is part of the design tool suite in use within Compaq.

1 Introduction

Consider the following problem regarding the transmission of a message between a writer and a reader facing an adversary. The writer stores logical zeroes and ones in a table of size 2^n stored in consecutive locations in a write-once memory. The memory is accessed through n one bit address lines. After the writing is complete, an adversary permutes the address lines. For example, for $n = 4$ there are sixteen memory locations: if the address lines are set to 0010, before the adversary acts, the memory returns the value stored in location 2. If the adversary permutes the second and third address line, the memory sees a request for location 0100 and returns the value stored in location 4.

The reader does not know the permutation used by the adversary, but can read all the memory locations. The reader's goal is to discover how the address lines were permuted, and, in addition, to obtain a message from the writer. Assuming the reader and writer establish a protocol ahead of time, how many bits can they communicate? More practically, what is a good protocol?

This problem arose in the context of Field-Programmable Gate Arrays (FPGAs) hardware design. An FPGA is a simple reconfigurable hardware device. The first commercial FPGA was introduced in 1986 [1]. For a large part of today's FPGAs, their basic logical element is equivalent to a look-up table [4]. The usual tools for FPGA design lay out a circuit on these logical elements, routing the wiring as appropriate. In particular, one tool currently in use permutes the address lines as appropriate to improve the wiring layout. This process is perfectly reasonable if the FPGA programmer want to use the design as a "black box." However, if the FPGA programmer wants to patch the design, an effective means of determining this permutation is necessary. The number of memory locations in the table dedicated to this end should be as low as possible, so that the rest of the table can be used for other purposes. (Because of the layered structure of the complex software used for wiring layout, keeping track of the permutation through the layers is not feasible.)

We describe a brute-force approach to the problem, as well as a simple algorithmic solution.

2 Brute force: table look-up

For any specific n, the problem can be solved by brute force. We divide all possible settings of table-content bits into equivalence classes; two settings are equivalent if and only if the first yields the same memory output as the second via some address lines permutation. We then count the number of equivalence classes with $n!$ distinct members. If C_n is the number of such classes, then the writer can effectively transmit any value in the range $[0 \ldots C_n - 1]$ in such a way that the reader can determine the value plus the permutation used by the adversary. This is accomplished by establishing one representative member from each of the C_n equivalence classes, and sending one of these C_n representatives. The value from $[0 \ldots C_n - 1]$ is determined by the reader from the class of the read memory bits; the permutation is similarly determined by which of the $n!$ permutations of the representative appears in the memory. Essentially, then, one can reduce the problem to a large table look-up.

In practice, however, this approach appears infeasible for all but the smallest values of n, as there are 2^{2^n} possible ways to set the memory. Using a brute force table-look up approach rapidly becomes infeasible in terms of memory utilization and preprocessing. The first few values of C_n are $2, 4, 16, 1792, 34339072, \ldots$. We have not determined a closed form for C_n; this remains an open problem.

In a similar vein, we might ask how many values D_n can be passed if we do not care whether the reader learns the adversarial permutation. In this case, all the

*Compaq Systems Research Center, Palo Alto, California. E-mail: {broder,michaelm,moll}@pa.dec.com

equivalence classes (and not just those with $n!$ members) count, as each class determines a possible value from $[0 \ldots D_n - 1]$. The first few values in this case are $2, 12, 80, 3984, 37333248, \ldots$. A closed form for D_n also remains an open problem. We note that neither C_n or D_n appear as sequences in the famous Sloane's list [2, 3].

3 An algorithmic solution

We have devised a simple algorithmic solution which requires at most $n \log_2 n$ memory probes to determine the permutation, and uses only $n \log_2 n$ of the 2^n bits of the memory. These are both within a $1 + o(1)$ factor of optimal, since on average (a) it takes at least $\log_2(n!)$ memory probes to determine the permutation; and (b) the writer cannot transmit more than $2^n - \log_2(n!)$ bits of information if the writer has to specify a permutation as well. (Note that if the reader does not need to determine the permutation, then our algorithm still works, but we can no longer claim that it is within an $1 + o(1)$ factor of optimal. Finding non-trivial bounds for this case remains open.)

We establish the appropriate notation. Initially, we assume that the number of address lines is $n = 2^r$ for some r. We label the memory locations by n-dimensional $\{0,1\}$ vectors. Originally the writer assigns bit values $f(x) \in \{0,1\}$ to the vectors (locations) $x \in \{0,1\}^n$. We denote the permutation chosen by the adversary as π and view it as a permutation of the numbers 0 to $n-1$. We use $\hat{\pi}$ to represent the action of π on vectors in the natural way: for example, if there are 4 address lines, and $\pi(0) = 0, \pi(1) = 2, \pi(2) = 1$, and $\pi(3) = 3$, then $\hat{\pi}(x) = \hat{\pi}(x_3x_2x_1x_0) = x_3x_1x_2x_0$. The values returned by the memory, after the adversary's evil deed, are denoted by $g(x)$, where $g(x) = f(\hat{\pi}(x))$.

The reader learns the permutation π after r rounds. For each round the reader reads the value of $g(x)$ in n distinct locations. These locations are independent of π and different from round to round. As we explain, before the permutation, the writer sets only the locations that eventually will be read. Hence $n \log_2 n$ values in the table are stored and read by our algorithm and the other locations are available for message transmission. We maintain the following invariant: after round k, for each line i, we know $\pi(i)$ modulo 2^k. Note that this invariant is trivially true before round 1. We call this the *bit-by-bit* approach. To simplify exposition, we describe the writing and the reading round by round, although in fact the writer does all the writing before the reading begins.

For the first round (round 1), the writer sets $f(x)$ to be 1 for all unit vectors $x = e_i$ for odd i, and 0 for all unit vectors $x = e_i$ for even i. The reader sets exactly one line j to 1 and all the others to 0. The memory returns 1 if and only if $\pi(j) = 1$, that is, j is mapped to an odd-numbered line.

Similarly, for round k, let the values of z range over $[0 \ldots 2^k - 1]$. The writer sets $f(x)$ for all x with a 1 in *all* positions x_i with $i = z - 1 \bmod 2^k$, exactly one 1 in one of the $n/2^k$ positions x_i with $i = z \bmod 2^k$ (call this position j), and 0's elsewhere. Note that there are n possibilities for x corresponding to the n possible values for j. The writer sets $f(x)$ to 1 if $(j - z)/2^k$ is odd and to 0 otherwise.

The reader, given the information gathered in prior rounds, can determine the permuted position of each line modulo 2^k. Hence it can compute all x such that $\hat{\pi}(x)$ has $\hat{\pi}(x)_i = 1$ in *all* positions with $i = z - 1 \bmod 2^k$, $\hat{\pi}(x)$ has exactly one 1 in one of the $n/2^k$ positions $\hat{\pi}(x)_i$ with $i = z \bmod 2^k$. Let j be the index of this particular position within x. That is, the reader can determine how to set the address bits to read values $g(x) = f(\hat{\pi}(x))$ precisely for the x's that the writer has defined for this round. Again, these reads determine for each j whether the $(k+1)$'st bit from the right of $\pi(j)$ is 0 or 1. Our invariant is maintained, and hence only $n \cdot r = n \log_2 n$ values are set and read in the memory.

Minor improvements can be made. For example, the reader need not read n values each round, but only $n-1$ values, since the nth value to be read is determined by the other $n - 1$.

When $n = 2^r + a$, where $0 < a < 2^r$, we use an $(r+1)$'st round for locations which are not determined by the first r bits from the right. The same argument shows that the total number of memory locations that need to be set and read is at most $n \cdot r + 2a = n\lfloor \log_2 n \rfloor + 2a$.

4 Acknowledgement

We wish to thank Mike Burrows, who computed the computable terms of the C_n and D_n sequences.

References

[1] W. S. Carter & Al., *A user programmable reconfigurable logic array*, in Proceedings of the IEEE 1986 Custom Integrated Circuits Conference., May 1986, pp. 233–235.

[2] N. J. A. Sloane, *Sloane's on-line encyclopedia of integer sequences.* Available on-line via `http://www.research.att.com/~njas/sequences/` .

[3] ———, *A Handbook of Integer Sequences*, Academic Press, 1973.

[4] *The programmable logic data book 1998.* Xilinx Inc., San Jose, CA, 1998. Available on line via `http://www.xilinx.com/partinfo/databook.htm` .

Number theoretic aspects of a combinatorial function

Lorenz Halbeisen[1] and Norbert Hungerbühler

Abstract

We investigate number theoretic aspects of the integer sequence $\mathrm{seq}^{1\text{-}1}(n)$ with identification number A000522 in Sloane's On-Line Encyclopedia of Integer Sequences: $\mathrm{seq}^{1\text{-}1}(n)$ counts the number of sequences without repetition one can build with n distinct objects. By introducing the the notion of the "shadow" of an integer function, we examine divisibility properties of the combinatorial function $\mathrm{seq}^{1\text{-}1}(n)$: We show that $\mathrm{seq}^{1\text{-}1}(n)$ has the reduction property and its shadow d therefore is multiplicative. As a consequence, the shadow d of $\mathrm{seq}^{1\text{-}1}(n)$ is determined by its values at powers of primes. It turns out that there is a simple characterization of regular prime numbers, i.e. prime numbers p for which the shadow d of $\mathrm{seq}^{1\text{-}1}$ has the socket property $d(p^k) = d(p)$ for all integers k. Although a stochastic argument supports the conjecture that infinitely many irregular primes exist, there density is so thin that there is only one irregular prime number less than $2.5 \cdot 10^6$, namely 383.

1 Introduction

The sequence we are interested in has the ID number A000522 in Sloane's On-Line Encyclopedia of Integer Sequences (http://www.research.att.com/~njas/sequences). Former identification numbers of this sequence were M1497 in [SP] and N0589 in [Sl].

The sequence A000522 has many faces (see, e.g., [Ga], [Si] or [Ri]). The most accessible one is its combinatorial interpretation:

Definition 1 *For $n \in \mathbb{N} = \{0, 1, 2, \dots\}$ let $\mathrm{seq}^{1\text{-}1}(n)$ denote the number of one-to-one sequences – these are sequences without repetitions – we can build with n distinct objects.*

[1]The author would like to thank the *Swiss National Science Foundation* for supporting him.

2000 Mathematics Subject Classification: **11A51** 11B50 11B75 11A41

Notice that for $l \leqslant n$, each one-to-one function from $\{0, \ldots, l-1\}$ to $\{0, \ldots, n-1\}$ corresponds in a unique way to a sequence without repetitions of $\{0, \ldots, n-1\}$ of length l. For example, for two objects, say a_1 and a_2, we can build the following sequences:

$$\langle\ \rangle\ (= \text{the empty sequence}), \langle a_1\rangle, \langle a_2\rangle, \langle a_1, a_2\rangle, \langle a_2, a_1\rangle.$$

Hence, $\mathrm{seq}^{1\text{-}1}(2) = 5$. Of course, it is easy to find a general expression for $\mathrm{seq}^{1\text{-}1}(n)$. Since there are $\binom{k}{n}$ possible ways to choose k objects from a set of n (distinct) objects, and since k (distinct) objects give rise to $k!$ permutations, we get the following

Lemma 2 $\mathrm{seq}^{1\text{-}1}(n) = \displaystyle\sum_{k=0}^{n} \binom{n}{k} k! = \sum_{j=0}^{n} \frac{n!}{j!}$. ■

Also the next representation for $\mathrm{seq}^{1\text{-}1}(n)$ is elementary.

Lemma 3 *For all positive $n \in \mathbb{N}$ we have*

$$\mathrm{seq}^{1\text{-}1}(n) = \lfloor e\, n! \rfloor.$$

Remark: For $n = 0$ the formula does not hold, since $\mathrm{seq}^{1\text{-}1}(0) = 1 < 2 = \lfloor e\, 0! \rfloor$.

Proof of Lemma 3. According to Lemma 2 we have

$$\begin{aligned} en! &= \mathrm{seq}^{1\text{-}1}(n) + \sum_{j=n+1}^{\infty} \frac{n!}{j!} \\ &= \mathrm{seq}^{1\text{-}1}(n) + \\ &\quad \underbrace{+ \frac{1}{n+1}\Big(1 + \frac{1}{n+2} + \frac{1}{(n+2)(n+3)} + \frac{1}{(n+2)(n+3)(n+4)} + \ldots\Big)}_{\leqslant \frac{1}{n+1}(e-1) < 1 \text{ for } n \geqslant 1}. \end{aligned}$$

■

The following recursive relation for $\mathrm{seq}^{1\text{-}1}(n)$ is an immediate consequence of the second formula in Lemma 2.

Lemma 4 *For all positive $n \in \mathbb{N}$ we have* $\mathrm{seq}^{1\text{-}1}(n) = n\,\mathrm{seq}^{1\text{-}1}(n-1) + 1$. ■

Using this formula, we finally get the following integral representation of $\mathrm{seq}^{1\text{-}1}(n)$.

Lemma 5 *For all $n \in \mathbb{N}$ we have*

$$\mathrm{seq}^{1\text{-}1}(n) = e \int_1^\infty t^n e^{-t} dt\,.$$

Proof. The formula is correct for $n = 0$. Moreover, by integration by parts, we have inductively

$$\begin{aligned}\mathrm{seq}^{1\text{-}1}(n) = e \int_1^\infty \underbrace{t^n}_{\downarrow} \underbrace{e^{-t}}_{\uparrow} dt &= e\left(-t^n e^{-t}\right)\Big|_1^\infty + e \int_1^\infty n t^{n-1} e^{-t} dt \\ &= 1 + n\,\mathrm{seq}^{1\text{-}1}(n-1)\end{aligned}$$

■

Just for the sake of completeness we like to mention that the exponential generating function $g(z)$ of $\mathrm{seq}^{1\text{-}1}(n)$ is given by $g(z) = \frac{e^z}{1-z}$. This is easily checked directly, or deduced, e.g. by Oberschelp's technique (see [Ob]).

In the sequel, to keep the formulas short, let $n^\star := \mathrm{seq}^{1\text{-}1}(n)$.

Notation: Throughout this text we adopt the standard notation $a|b$ to express that a divides b for $a, b \in \mathbb{N}$. Moreover, if $b \geqslant 1$ then $\mathrm{Mod}(a,b) := a - b\lfloor \frac{a}{b} \rfloor$ denotes the reminder of the division of a by b; and (a,b) denotes the greatest common divisor of a and b.

2 The divisibility of $n^\star$

We start our investigation on divisibility properties of $n^\star$ with a simple fact which has first been proved in [HS].

Lemma 6 *For natural numbers $n, k \in \mathbb{N}$, the following implication holds: If $2^k | n^\star$, then $2^k | (n+2^k)^\star$ and $2^k \nmid (n+t)^\star$ for any t with $0 < t < 2^k$.*

Proof. The implication $2^k | n^\star \implies 2^k | (n+2^k)^\star$ follows easily from the reduction property of the sequence $\mathrm{seq}^{1\text{-}1}(n)$ (see Lemma 9 below). So, we only have to prove here that if $2^k | n^\star$, then $2^k \nmid (n+t)^\star$ for any t with $0 < t < 2^k$.

For $k \leqslant 4$, an easy calculation modulo 2^k shows that for each n we have: If $2^k \mid n^\star$, then $2^k \nmid (n+t)^\star$ for $0 < t < 2^k$ (cf. also Lemma 9).

Assume there is a smallest k ($k \geqslant 4$) such that $2^{k+1} | n^\star$ and $2^{k+1} \mid (n+t)^\star$ for some t with $0 < t < 2^{k+1}$. Then, because $2^k | 2^{k+1}$, we have $2^k | n^\star$ and $2^k | (n+t)^\star$. Since k

is by definition the smallest such number, we know that t must be 2^k.

$$
\begin{array}{rcccccccccccl}
(n+2^k)^\star = \sum_{i=0}^{n+2^k} \frac{(n+2^k)!}{i!} = & & 1\cdot & 2\cdot & \ldots & \cdot 2^k\cdot & (2^k+1)\cdot & \ldots & \cdot(2^k+n) & & & & (1)\\
& + & & 2\cdot & \ldots & \cdot 2^k\cdot & & \ldots & \cdot(2^k+n) & & & & (2)\\
& & & & \ddots & & & & & & & & \vdots\\
& + & & & & 2^k\cdot & & \ldots & \cdot(2^k+n) & & & & (2^k)\\
& & & & & & & \ddots & & & & & \vdots\\
& + & & & & & & & (2^k+n) & & & & (2^k+n)\\
& + & & & & & & & & & 1 & & (2^k+n+1)
\end{array}
$$

It is easy to see that 2^{k+1} divides lines (1) – (2^k) since $k \geqslant 2$ and $n \geqslant 2$.

If we expand the products in the lines (2^k+1) – (2^k+n+1), we can collect all terms which are obviously divisible by 2^{k+1}. So, for a suitable natural number m we get

$$(n+2^k)^\star = 2^k \cdot \left(\sum_{j=0}^{n-1} \sum_{i>j}^{n} \frac{n!}{i\cdot j!} \right) + n^\star + 2^{k+1}\cdot m\,. \tag{1}$$

Remember that we have assumed $2^{k+1} | n^\star$, where $n \geqslant 3$ and $k \geqslant 4$. Thus, $n^\star$ is even and hence n has to be odd. If j is $n-1$, $n-2$ or $n-3$, then $\sum_{i>j}^{n} \frac{n!}{i\cdot j!}$ is odd. Moreover, if $0 \leqslant j \leqslant (n-4)$, then $\sum_{i>j}^{n} \frac{n!}{i\cdot j!}$ is even and therefore, $\sum_{j=0}^{n-1} \sum_{i>j}^{n} \frac{n!}{i\cdot j!}$ is odd. Hence, by (1) and $2^{k+1} | n^\star$ we get $2^{k+1} \nmid (n+2^k)^\star$, which is a contradiction. ■

Remark. The Lemma 6 is the crucial point in the proof – which does not make use of the axiom of choice – of the following fact (cf. [HS, Theorem 4]): For any infinite set M, there exists no bijection between the power-set of M and the set of all finite one-to-one sequences of M.

A natural question that arises in connection with Lemma 6 is whether for every $k \in \mathbb{N}$ there exists an $n \in \mathbb{N}$ such that $2^k \mid n^\star$. To answer this and related questions involving divisibility properties of integer sequences in general and of the sequence $\mathrm{seq}^{1\text{-}1}(n)$ in particular, we introduce the notion of the "shadow" of a sequence.

Definition 7 *If $\{f(n)\}_{n\in\mathbb{N}}$ is a sequence of natural numbers, we define its* **shadow** *to be the sequence $\{d(h)\}_{h\in\mathbb{N}}$ given by*

$$d(h) := |D(h)|,$$

where $D(h) := \big\{n \in \mathbb{N} : (n < h) \wedge (h \mid f(n))\big\}$ are the **shadow sets** *of the sequence f.*

The shadow $d(h)$ counts the sequence entries $f(0), f(1), \ldots, f(h-1)$ which are divisible by h. So, the shadow measures (to a certain extent) how "divisible" the entries of the sequence $f(n)$ are: For example, if only prime numbers occur in the sequence, then its shadow will reflect this fact by being small. If the entries of $f(n)$ have many divisors, the shadow will typically be large.

Remark. Lemma 6 implies that the shadow of $f(n) = \mathrm{seq}^{1\text{-}1}(n)$ has the following property: For all $k \in \mathbb{N}$, there holds $d(2^k) \leqslant 1$. Actually, as a consequence of Lemma 15, it will turn out that $d(2^k) = 1$ for all k.

Examples. If $f(n) = c \in \mathbb{N}$ is a constant function, then the shadow of f is

$$d(h) = \begin{cases} h & \text{if } h|c \text{ and } h > 1, \\ 0 & \text{otherwise.} \end{cases}$$

If $f(n)$ is an arithmetic sequence of first order, then its shadow is periodic, and for the shadow of Euler's φ-function we have $d(h) = 1$ for all $h \geqslant 1$. ◯

The shadow gives a certain amount of information on the divisibility of the entries of a sequence. Nevertheless, two different sequences can "cast" the same shadow as the following example shows.

Example. If for a function f there exists an $n_0 \in \mathbb{N}$ such that for all $h \geqslant n_0$ we have $d(h) = 0$, then for all $h \geqslant n_0$ we have $f(h) \leqslant h$. Vice versa, if $f(h) \leqslant h$ for all $h \in \mathbb{N}$, then $d(h)$ equals the number of zeros in $(f(0), f(1), \ldots, f(h-1))$. Hence, it is easy to construct different functions which have the same shadow:

n	0	1	2	3	4	5	6	7	...
$f_1(n)$	0	1	2	3	4	5	6	7	...
$f_2(n)$	0	1	1	2	3	4	5	6	...
$f_3(n)$	0	1	1	1	2	3	4	5	...
shadow	0	1	1	1	1	1	1	1	...

◯

Now, we want to investigate the shadow of $\mathrm{seq}^{1\text{-}1}(n)$. First, we show that this particular shadow is multiplicative and it turns out that the reason for this is the fact that $\mathrm{seq}^{1\text{-}1}$ has the reduction property:

Definition 8 *A sequence $\{f(n)\}_{n\in\mathbb{N}}$ is said to have the reduction property, if for all $n, q \in \mathbb{N}$, $q \geqslant 1$, we have*

$$\mathrm{Mod}(f(n), q) = \mathrm{Mod}(f(\mathrm{Mod}(n, q)), q).$$

Lemma 9 *The sequence $\{\mathrm{seq}^{1\text{-}1}(n)\}_{n\in\mathbb{N}}$ has the reduction property.*

Proof. For $q = 1$ or $q > n$, the statement is trivial. So, we may assume $1 < q \leqslant n$.

First we consider the case when $\mathrm{Mod}(n, q) = 0$. By Lemma 4 we have $\mathrm{seq}^{1\text{-}1}(n) = n \cdot \mathrm{seq}^{1\text{-}1}(n-1) + 1$ and hence by $\mathrm{Mod}(n, q) = 0$ we get $\mathrm{seq}^{1\text{-}1}(n) \equiv 1 \mod q$, which implies $\mathrm{Mod}(\mathrm{seq}^{1\text{-}1}(n), q) = \mathrm{Mod}(\mathrm{seq}^{1\text{-}1}(\mathrm{Mod}(n, q)), q)$, because $\mathrm{seq}^{1\text{-}1}(0) = 1$.

Now assume that $\mathrm{Mod}(n+1, q) \neq 0$ and that the statement holds for n. Again by Lemma 4 we have $\mathrm{seq}^{1\text{-}1}(n+1) = (n+1) \cdot \mathrm{seq}^{1\text{-}1}(n) + 1$ and by the assumption we get

$$\begin{aligned} \mathrm{seq}^{1\text{-}1}(n+1) &\equiv \mathrm{Mod}((n+1), q) \cdot \mathrm{seq}^{1\text{-}1}(\mathrm{Mod}(n, q)) + 1 \mod q \\ &\equiv \mathrm{seq}^{1\text{-}1}(\mathrm{Mod}(n+1, q)) \mod q. \end{aligned}$$

Therefore, $\mathrm{Mod}(\mathrm{seq}^{1\text{-}1}(n+1), q) = \mathrm{Mod}(\mathrm{seq}^{1\text{-}1}(\mathrm{Mod}(n+1, q)), q)$ is validated. ■

Lemma 10 *The shadow d of a sequence $f(n)$ which has the reduction property is multiplicative, i.e. if $(a, b) = 1$, then $d(ab) = d(a)d(b)$.*

Proof. Suppose $(a, b) = 1$, then we have by the reduction property

$$\begin{aligned} D(ab) &= \{n \in \mathbb{N} : n < ab \wedge ab | f(n)\} \\ &= \{n \in \mathbb{N} : n < ab \wedge a | f(n) \wedge b | f(n)\} \\ &= \{n \in \mathbb{N} : n < ab \wedge a | f(\mathrm{Mod}(n, a)) \wedge b | f(\mathrm{Mod}(n, b))\}. \end{aligned}$$

This means that a natural number n is an element of the shadow set $D(ab)$ if and only if it lies in the intersection of the two sets

$$A := \{i + ax : i \in D(a) \wedge x \in \{0, 1, \ldots, b-1\}\}$$

and

$$B := \{j + by : j \in D(b) \wedge y \in \{0, 1, \ldots, a-1\}\}.$$

In other words $D(ab) = A \cap B$.

Observe that since $(a, b) = 1$, we have that for all $\langle i, j \rangle \in \{0, 1, \ldots, a-1\} \times \{0, 1, \ldots, b-1\}$ there exists a unique $\langle x, y \rangle \in \{0, 1, \ldots, b-1\} \times \{0, 1, \ldots, a-1\}$ such that $i + ax = j + by$. This implies that $|A \cap B| = |D(a)|\,|D(b)|$ and hence,

$$d(ab) = |D(ab)| = |A \cap B| = |D(a)|\,|D(b)| = d(a)\,d(b).$$ ■

As an immediate consequence we get the following

Corollary 11 *If d is the shadow of* $\mathrm{seq}^{1\text{-}1}$ *and if $n = \prod_{i=1}^{k} p_i^{k_i}$ is the prime decomposition of n, then*

$$d(n) = \prod_{i=1}^{k} d(p_i^{k_i}).$$ ■

Therefore, the shadow d of $\mathrm{seq}^{1\text{-}1}$ is fully determined by its values on the powers of prime numbers. But what can we say about $d(p^k)$ for p prime? Let us start our discussion of this question by the following observation.

By the reduction property, all elements $m \in D(p^{k+1})$ must be of the form $m = n + l\,p^k$ for some $n \in D(p^k)$ and some $l \in \{0, 1, \ldots, p-1\}$. Hence, we get inductively that if $d(p) = 0$, then $d(p^k) = 0$ for all positive $k \in \mathbb{N}$.

Definition 12 *A prime number p with $d(p) = 0$ is called* **annihilating**.

Example. The sequence of annihilating primes is 3, 7, 11, 17, 47, 53, 61, 67, 73, 79, 89, 101, 139, 151, 157, 191, 199, ... ◯

From the observation above and the multiplicativity property, we have

Proposition 13 *If $n \in \mathbb{N}$ is divisible by an annihilating prime, then $d(n) = 0$.* ■

What can we say about primes that are not annihilating? For positive numbers $p, k, l, n \in \mathbb{N}$ we have the following:

$$\begin{aligned}
(n + l\,p^k)^\star &= \sum_{j=0}^{l\,p^k+n} \frac{(l\,p^k+n)!}{j!} \\
&= \frac{(l\,p^k+n)!}{0!} + \ldots + \frac{(l\,p^k+n)!}{(l\,p^k-1)!} + \frac{(l\,p^k+n)!}{(l\,p^k)!} + \ldots + \frac{(l\,p^k+n)!}{(l\,p^k+n)!} \\
&= \frac{(l\,p^k+n))!}{(l\,p^k-1)!}\,(l\,p^k-1)^\star + \sum_{j=l\,p^k}^{l\,p^k+n} \frac{(l\,p^k+n)!}{j!} \\
&= \Big((l\,p^k)\,(l\,p^k+1)\,\ldots\,(l\,p^k+n)\Big)\,(l\,p^k-1)^\star + \sum_{j=l\,p^k}^{l\,p^k+n} \frac{(l\,p^k+n)!}{j!} \\
&\equiv l\,p^k\,n!\,(l\,p^k-1)^\star + \sum_{j=l\,p^k}^{l\,p^k+n} \frac{(l\,p^k+n)!}{j!} \mod p^{k+1} \\
&\equiv l\,p^k\,n!\,(l\,p^k-1)^\star + l\,p^k \sum_{j=0}^{n-1}\sum_{i>j}^{n} \frac{n!}{j!\,i} + n^\star \mod p^{k+1} \\
&\equiv l\,p^k \left(n!\,(l\,p^k-1)^\star + \sum_{i=1}^{n}\sum_{j=0}^{i-1} \frac{n!}{j!\,i}\right) + n^\star \mod p^{k+1} \\
&\equiv l\,p^k \left(n!\,(l\,p^k-1)^\star + \sum_{j=0}^{n-1} \frac{n!}{(j+1)!}\,j^\star\right) + n^\star \mod p^{k+1} \\
&\equiv n^\star + l\,p^k \underbrace{\left(n!\,(p-1)^\star + \sum_{j=0}^{n-1} \frac{n!}{(j+1)!}\,j^\star\right)}_{=:s_{p,n}} \mod p^{k+1} \qquad (2)
\end{aligned}$$

From this calculation it is clear that the numbers $s_{p,n}$ defined in the previous line are crucial for a further investigation of the shadow of seq$^{1\text{-}1}$.

Definition 14 *The number*

$$X(p) := \prod_{n \in D(p)} \mathrm{Mod}(s_{p,n}, p)$$

is called the **excess** *of the prime p. A prime number p with $X(p) \neq 0$ is called* **regular** *and otherwise* **irregular**.

Example. Since the empty product is by definition equal to 1, all annihilating primes are regular. The smallest irregular prime number is 383, all other primes less than $2.5 \cdot 10^6$ are regular.

Lemma 15 *If p is a regular prime number, then the shadow d of $\mathrm{seq}^{1\text{-}1}$ has the socket property at powers of p, i.e. $d(p^k) = d(p)$ holds for all positive $k \in \mathbb{N}$.*

Before we prove Lemma 15, we state the following consequence.

Proposition 16 *If d is the shadow of $\mathrm{seq}^{1\text{-}1}$ and if $n = \prod_{i=1}^{k} p_i^{k_i}$ is the prime decomposition of n, then*

$$d(n) = \prod_{i=1}^{k} d(p_i)$$

provided each prime p_i is regular or one of the primes is annihilating. ■

To prepare the proof of Lemma 15, we need a property of $s_{p,n}$, which is given in the following

Lemma 17 *If p and n are natural numbers, then*

$$s_{p,n} \equiv s_{p,n+p} \mod p.$$

Proof. Let $r := \mathrm{Mod}(n,p)$, then $n = ap + r$ for some $a \in \mathbb{N}$. We first consider the case $n \geqslant p$, thus $a \neq 0$. Because $n \geqslant p$ we have $n! \equiv 0 \mod p$ and therefore $s_{p,n} \equiv \sum_{j=0}^{n-1} \frac{n!}{(j+1)!}\, j^\star \mod p$. Further we get

$$\begin{aligned}
\sum_{j=0}^{n-1} \frac{n!}{(j+1)!}\, j^\star &= \sum_{j=0}^{ap-2} \frac{n!}{(j+1)!}\, j^\star + \sum_{j=ap-1}^{n-1} \frac{n!}{(j+1)!}\, j^\star \\
&\equiv \sum_{j=ap-1}^{n-1} \frac{n!}{(j+1)!}\, j^\star \mod p \\
&\equiv \sum_{j=-1}^{r-1} \frac{r!}{(j+1)!}\, (p+j)^\star \mod p \\
&\equiv r!(p-1)^\star + \sum_{j=0}^{r-1} \frac{r!}{(j+1)!}\, j^\star \mod p\,.
\end{aligned}$$

If $n < p$, then $\mathrm{Mod}(n,p) = n$ and we get $r = n$. Hence, we have for all $p, n \in \mathbb{N}$ that

$$s_{p,n} \equiv r!(p-1)^\star + \sum_{j=0}^{r-1} \frac{r!}{(j+1)!}\, j^\star \mod p\,,$$

where $r := \mathrm{Mod}(n,p)$. ■

Proof of Lemma 15. Let p be a regular prime number. We proceed inductively: For $k = 1$ there is nothing to show. For exponents larger than 1 we recall that all elements $m \in D(p^{k+1})$ must be of the form $m = n + l\,p^k$ for some $n \in D(p^k)$ and some $l \in \{0, 1, \ldots, p-1\}$. By the calculation (2) above, we have

$$(n + l\,p^k)^\star \equiv n^\star + l\,p^k s_{p,n} \mod p^{k+1}.$$

Hence, it suffices to show, that

$$n \in D(p^k) \implies s_{p,n} \not\equiv 0 \mod p \tag{3}$$

In fact, since p is prime, if the conclusion of (3) holds, the congruence $n^\star + l p^k s_{p,n} \equiv 0 \mod p^{k+1}$ has a unique solution $l \in \{0, 1, \ldots, p-1\}$ and therefore, the sets $D(p^k)$ and $D(p^{k+1})$ have the same cardinality, which implies $d(p^k) = d(p^{k+1})$.

On the other hand, by Lemma 17, (3) holds for all k if it is true for $k = 1$. But this, by definition, is exactly the case for regular primes p. ■

3 How peculiar are irregular primes?

In this section we investigate the value of $d(p^k)$ for irregular primes p and $k \geqslant 1$, but first we recall some facts concerning regular primes.

For a regular prime p we have $d(p^k) = d(p)$ for any positive $k \in \mathbb{N}$. Further, by definition, a prime number p is annihilating if and only if $d(p) = 0$. Remember that all annihilating prime numbers are regular. Now, fix an irregular prime number p. What can we say for $k \geqslant 1$ about $d(p^k)$?

Example. If we consider the smallest irregular prime number $p = 383$, it turns out that $d(383) = 3$, but $d(383^k) = 2$ for all $k \geqslant 2$. The reason for this shall be explained below. ◯

First note that – because p is not annihilating – $d(p) > 0$. Because p is assumed to be irregular, there exists at least one $n \in D(p)$ such that $\mathrm{Mod}(s_{p,n}, p) = 0$ and therefore, by Lemma 17, we have $\mathrm{Mod}(s_{p,n+lp}, p) = 0$ for all $l \in \mathbb{N}$.

For $k \geqslant 1$ and any $n \in D(p^k)$ with $\mathrm{Mod}(s_{p,n}, p) = 0$ we have either the case $p^{k+1} \nmid n^\star$ or the case $p^{k+1} \mid n^\star$.

If $n \in D(p^k)$ with $\mathrm{Mod}(s_{p,n}, p) = 0$ – depending in which case we are – we have either $p^{k+1} \nmid (n + l\,p)^\star$ (for all $l \in \mathbb{N}$) or $p^{k+1} \mid (n + l\,p)^\star$ (for all $l \in \mathbb{N}$). To see this, remember that by (2), for any $n, l \in \mathbb{N}$ we have

$$(n + l\,p^k)^\star \equiv n^\star + l\,p^k \cdot s_{p,n} \mod p^{k+1}\,.$$

Therefore, if $p^{k+1} \mid n^\star$ (or $p^{k+1} \nmid n^\star$) and $p \mid s_{p,n}$, then we get $p^{k+1} \mid (n + l\,p^k)^\star$ (or $p^{k+1} \nmid (n + l\,p^k)^\star$, respectively) for any $l \in \mathbb{N}$.

Now let

$$\delta(p) := |\{n \in D(p) : \mathrm{Mod}(s_{p,n}, p) \neq 0\}|\,,$$

and for $k \geqslant 2$ let

$$\varepsilon(p^k) := |\{n \in D(p^{k-1}) : \mathrm{Mod}(s_{p,n}, p) = 0 \wedge p^k \mid n^\star\}|\,.$$

Notice that if $\varepsilon(p^{k_0}) = 0$ for some $k_0 \geqslant 2$, then $\varepsilon(p^k) = 0$ for any $k \geqslant k_0$. By the facts given above, it is not hard to verify that for $k \geqslant 2$ we have

$$d(p^k) = \delta(p) + p \cdot \varepsilon(p^k)\,.$$

Example. If we consider again the smallest irregular prime number $p = 383$, where $D(383) = \{296, 340, 353\}$ and therefore $d(383) = 3$, it turns out that $\delta(383) = 2$ and $\varepsilon(383^2) = 0$. This we get because $\mathrm{Mod}(s_{383,296}, 383) = 0$ and $383^2 \nmid 296^\star$. Thus, $d(383^k) = \delta(383) = 2$ for all $k \geqslant 2$. ◯

4 How rare are irregular primes?

We recall that a prime number p is irregular, if there exists an $n \in D(p)$ with $\mathrm{Mod}(s_{p,n}, p) = 0$. The function $n \mapsto \mathrm{Mod}(s_{p,n}, p)$ shows (for different primes p) a rather random-like behavior. The idea is now, to replace $n \mapsto \mathrm{Mod}(s_{p,n}, p)$ by equidistributed independent random variables $X_{p,n}$ which take values in $\{0, 1, \ldots, p-1\}$, i.e. the probability that $X_{p,n} = i$ is $\frac{1}{p}$ for each $i \in \{0, 1, \ldots, p-1\}$. From $X_{p,n}$ we construct a new random variable Y_p which takes, for each prime number p, the value 1 if $X_{p,n} = 0$ for some $n \in D(p)$ and zero otherwise. In other words, instead of looking whether $\mathrm{Mod}(s_{p,n}, p) = 0$ for $n \in D(p)$, we throw a dice with p faces $\{0, 1, \ldots, p-1\}$ for each $n \in D(p)$. Therefore, the values p for which $Y_p = 1$ are now called randomly irregular primes. The idea is, that randomly irregular primes

should have approximately the same distribution as the ordinary irregular prime numbers. The probability that p is randomly regular is

$$P(p \text{ is randomly regular}) = \left(1 - \frac{1}{p}\right)^{d(p)}.$$

Thus, we have

$$\begin{aligned} P(p_1, p_2, \ldots, p_k \text{ are all randomly regular}) &= \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right)^{d(p_i)} \\ &= \exp \sum_{i=1}^{k} d(p_i) \log\left(1 - \frac{1}{p_i}\right). \end{aligned}$$

Observe, that $\log(1-x) \leqslant -x$ for $x \geqslant 0$ (and $|\log(1-x) + x| = O(x^2)$ for $x \to 0$). Thus, we can estimate

$$P(p_1, p_2, \ldots, p_k \text{ are all randomly regular}) \lesssim \exp\left(-\sum_{i=1}^{k} \frac{d(p_i)}{p_i}\right).$$

If we suppose for the moment – and experiments support this to some extent – that in average $d(p) \approx c > 0$ is approximately constant (with a numerical value of $c \approx 0.9$), then we have

$$P(p_1, p_2, \ldots, p_k \text{ are all randomly regular}) \lesssim \exp\left(-c \sum_{i=1}^{k} \frac{1}{p_i}\right). \tag{4}$$

Now, the sum of inverse primes is divergent, and hence,

$$P(p_1, p_2, \ldots, p_k \text{ are all randomly regular}) \to 0 \quad \text{for } k \to \infty.$$

In other words, the probability that after a certain prime number no other randomly irregular prime number occurs is – under the made hypothesis on $d(p)$ – zero. So, we should expect that infinitely many irregular prime numbers exist.

On the other hand, what can we say about the frequency of occurrence of (randomly) irregular primes? In order to answer this question, we close this discussion by calculating the distribution function of randomly irregular prime numbers. In other words we ask: How many randomly irregular primes may we expect in the set $\{p_1, p_2, \ldots, p_k\}$. This is simply

$$E\left[\sum_{i=1}^{k} \tilde{Y}_{p_i}\right] = \sum_{i=1}^{k} E[\tilde{Y}_{p_i}] = \sum_{i=1}^{k} \frac{d(p_i)}{p_i}.$$

Example. The expected number of randomly irregular prime numbers in the range $\{2, \ldots, 10^3\}$ is $1.99703\ldots$ (the actual number of irregular primes in this interval is 1). Further, the expected number of randomly irregular primes in the interval $\{2, \ldots, 10^6\}$ is about 2.67758, so still far below 3, and the expected number of randomaly irregular primes in the interval $\{385, \ldots, 2.5 \cdot 10^6\}$ is about 0.874123 (the actual number of irregular primes in this interval is 0). ◯

Again, under the assumption that $d(p)$ is in average a positive constant c, we can now state the following conjecture:

Conjecture 18 *There exist infinitely many irregular primes. Furthermore the distribution function of the irregular primes is asymptotically*

$$\big|\{p \leqslant n : p \text{ is an irregular prime number}\}\big| \sim c \sum_{\substack{p \leqslant n \\ p \text{ prime}}} \frac{1}{p}$$

for a positive constant c.

Remark. If we consider the random variable Z which takes the value p where p is the smallest randomly irregular prime, then a similar calculation as above shows that the expected value of Z is $E[Z] = \infty$.

As a final remark we should mention that similar arguments as above support the conjecture that there are infinitely many prime numbers p, such that

$$2^{p-1} \equiv 1 \mod p^2 \tag{5}$$

This conjecture is related to generalized Carmichael numbers (see [HH]). The prime numbers satisfying (5) seem to have a similar distribution as irregular primes, which makes them equally hard to find. In fact, at the moment, the only known prime numbers which satisfy (5) are 1093 and 3511.

Acknowledgment. We wish to thank Stephanie Halbeisen for writing all the C-programs, which built the touchstones for our conjectures.

References

[Ga] J. M. Gandhi: On logarithmic numbers. *The Mathematics Student* **31** (1963), 73–83.

[HS] L. Halbeisen and S. Shelah: Consequences of arithmetic for set theory. *Journal of Symbolic Logic* **59** (1994), 30–40.

[HH] L. Halbeisen and N. Hungerbühler: On generalized Carmichael numbers. *Hardy-Ramanujan Journal* **22** (1999), 8–22.

[Ob] W. Oberschelp: Solving linear recurrences from differential equations in the exponential manner and vice versa, *in* "Applications of Fibonacci numbers, Vol. 6," (G. E. Bergum, A. N. Philippou and A. F. Horadam, Ed.), 365–380, Kluwer Acad. Publ., (Dordrecht), 1996.

[Ri] J. Riordan: "An Introduction to Combinatorial Analysis." Princeton University Press, Princeton, New Jersey (1980).

[Si] D. Singh: The numbers $L(m, n)$ and their relations with prepared Bernoulli and Eulerian numbers. *The Mathematics Student* **20** (1952), 66–70.

[Sl] N. J. A. Sloane: "A Handbook of Integer Sequences." Academic Press, New York (1973).

[SP] N. J. A. Sloane and S. Plouffe: "The Encyclopedia of Integer Sequences." Academic Press, San Diego (1995).

Lorenz Halbeisen
Dept. of Mathematics
U.C. Berkeley
Evans Hall 938
Berkeley, CA 94720
USA
halbeis@math.berkeley.edu

Norbert Hungerbühler
Dept. of Mathematics
U.A. Birmingham
452 Campbell Hall
Birmingham, AL 35294-1170
USA
buhler@math.uab.edu

TRIVIA HUNT ANSWERS **CS304, January 1989**

1. Who were the winners of the first Computer Science Trivia Hunt at Stanford? *5 points each*
What did they win? *10 points*

Tomás Feder, Barry Hayes, Tom Henzinger, and Alex Wang. (Reference: CS1154, Appendix A.) They received certificates (printed with POX, a historic computer typesetting system); they were also treated to dinner at Late for the Train restaurant by Don and Jill Knuth on 10 March 1988. (Source: The team members.)

2. What computer scientist was born on 23 June 1912? *15 points*

Alan Mathison Turing. (Ref: Hodges, *Alan Turing: The Enigma*, p. 5.)

3. In what house did Bill Walsh live when he was a Stanford coach? Who lives there now? *15 points each*

He was coach in 1977–1978. According to the Stanford Faculty/Staff Directory, 1978, he lived at 903 Cottrell Way, Stanford CA 94305; this is confirmed by the present owner, Prof. Thomas J. Hughes (chair of Mechanical Engineering). [A plausible, but false, answer was also submitted: Inquirers at the Athletic Department were told that Walsh lived in Menlo Park; and there is a Wm. D Walsh living in Menlo Park, listed continuously in local phone books since 1977. However, *that* Bill Walsh was a high school football coach, not college or pro; the "real" Bill Walsh lives on Valparaiso Avenue and has an unlisted phone number. Incidentally, Walsh's announcement of his retirement was front page news on Trivia Hunt day.]

4. What Stanford mathematics professor wrote one of the first papers ever published about the Tower of Hanoi? What were the dates of his birth and death? What is his relationship to Professor Floyd of our department? *15 points each*

Robert Edgar Allardice was co-author of "La Tour d'Hanoï," *Proceedings of the Edinburgh Mathematical Society* **2** (1884), 50–53; he was born 2 March 1862, came to Stanford in 1892, became emeritus in 1927, and died on 6 May 1928. (Reference: Poggendorf's *Handwörterbuch*; *Proceedings of the Royal Society of Edinburgh* **48** (1927–1928), 209–210.) Floyd lives at 895 Allardice Way.

5. What Stanford computer has its name displayed in stained glass? *15 points*

The SUMEX-AIM computer in Stanford Medical School. [People also found 'Solomon', 'charity', 'thing', 'sheep', 'how', and 'why' on the windows in Stanford Memorial Church; these are all names of computers at Stanford, according to `/etc/hosts`.]

6. What are the common names of *Formica rufa* Linnæus? *10 points each*

The fallow ant, according to Wheeler, *Ants*, p. 8, or McCook, *The Agricultural Ant of Texas*, p. 152; also called hill ant, wood ant, horse ant, and Waldameise (German), according to Donisthorpe, *British Ants*, p. 248; also red ant, Grizmek's *Animal Life*, vol. 2.

7. Problem 4 in this year's CS304 is based on an article by Leslie Valiant. Find all published papers that refer to his article and give a full citation for every such paper in the following style: L. G. Valiant, "Short monotone formulae for the majority function," *Journal of Algorithms* **5** (1984), 363–366. *10 points each*

The following can be found via *Science Citation Index*: Joel Friedman, "Constructing $O(n \log n)$ size monotone formulae for the kth threshold function of n boolean variables," *SIAM Journal on Computing* **15** (1986), 641–654. David S. Johnson, "The NP-completeness column: An ongoing guide," *Journal of Algorithms* **7** (1986), 289–305. Ravi B. Boppana, "Threshold

functions and bounded depth monotone circuits," *Journal of Computer and System Sciences* **32** (1986), 222–229. S. A. Lozkin and A. A. Semenov, "On construction of a complete system of compression functions and on complexity of monotone realization of threshold boolean functions," *Lecture Notes in Computer Science* **278** [*Fundamentals of Computation Theory*, proceedings of FCT87 in Kazan, USSR] (1987), 297–300. And, there are two other references in publications that (unfortunately) are not yet covered by Science Citation Index: Ravi B. Boppana, "Amplification of probabilistic boolean formulas," *Proceedings of the 26th Annual Symposium on Foundations of Computer Science* (1985), 20–29. (This one, unknown to Knuth before the Trivia Hunt, is quite relevant to Problem 4.) M. Karchmer and A. Wigderson, "Monotone circuits for connectivity require super-logarithmic depth," *Proceedings of the 20th Annual Symposium on Theory of Computing* (1988), 539–550.

8. What identification numbers and dates are stamped on the following Bench Marks of the U.S. Coast and Geodetic Survey on Stanford's campus? (1) near a monumental horse; (2) near a mosaic; (3) near a potted umbrella tree; (4) near the 9th fairway. *25 points each*

Bench Marks are shown on the Palo Alto quadrangle of the U.S. Geological Survey maps in Branner Library. (1) B151, 1933, at the base of the statue of Sherwood, near the Old Red Barn on Fremont Road. (2) R875, 1954, embedded in the NE corner of the Stanford Art Museum building. (3) A151, 1933, in concrete steps by the main entrance to the Carnegie Institution of Washington Plant Biology building. (4) C151, 1933, on top of a granite rock outcropping between the fairway and San Francisquito Creek, not far from the 9th tee of Stanford Golf Course. Another one (D151, 1933) appears near the 7th fairway. Still another (U110, 1932) is embedded in sandstone in the main quad, on a corner of building 310 facing the rear of Memorial Church. Several of us searched fruitlessly for yet another near the Children's Hospital. According to the Geological Survey in Denver, the Army Corps of Engineers came to Stanford in 1938 to determine the horizontal locations of the bench marks whose vertical elevations had been previously determined.

9. What artist made a painting of Jane Stanford's jewel collection, before she sold it to help pay faculty salaries? What were the dates of his birth and death? *10 points each*

Astley David Montague Cooper's painting entitled Mrs. Stanford's Jewel Collection hangs in the Stanford Museum, and it says he lived 1856–1924. Further research via the Master Index of biographical reference books leads to *Artists of the American West*, where his death date is given as 10 September 1924 in San Jose. The *San Jose Mercury Herald* for 11 September 1924, p. 11, gives his birthdate as 23 December 1856. According to A. Nagel, *Iron Will: The life and letters of Jane Stanford*, Mrs. Stanford used money from the sale of the jewels for an endowment whose income was "to be used exclusively for the purchase of books and other publications"; hence, the use of jewel money to pay faculty salaries is apparently a myth, although there was definitely a period when she contributed her own funds to help the faculty while her husband's estate was tied up in court.

10. What three faculty members of Stanford's Computer Science Department were born on the same day of the month (but not necessarily in the same month)? *30 points*

The `lookup` program on polya or the `find` program on SAIL gives Charles Bigelow on July 29, David Cheriton on March 29, and Gene Golub on February 29; also Consulting Professor Joe Halpern on May 29, and Visiting Professor John Sowa on March 29. If we exclude professors of the latter type, there are no two with the same birthday, although the "birthday paradox" says that there probably should be. Another answer, using a different database: John Hennessy, 22 Sep 1952; Yoav Shoham, 22 Jan 1956; Jeffrey Ullman, 22 Nov 1942.

11. What were the date and place of the first battle in the war between Mexico and the United States? *10 points each*

8 May 1846 at Palo Alto battlefield, Cameron County, Texas. (First blood was drawn on April 24 when an American reconnoitering party was attacked and captured; but the Palo Alto battle involved thousands of troops.)

12. Identify the author and source of the following quotations: *10 points for each author*
15 points for each source

a. He teaches him to hick and to hack, which they'll do fast enough of themselves . . .—fie upon you.

Shakespeare, *Merry Wives of Windsor*; Act IV, Scene 1, line 60 (or other line numbers in other sources). The NeXt computer has this online.

b. As a slow-witted human being I have a very small head and I had better learn to live with it and to respect my limitations and give them full credit, rather than try to ignore them, for the latter vain effort will be punished by failure.

Dijkstra, in *Structured Programming*, Academic Press, 1972, p 3.

c. My thesis is that high-performance systolic arrays can be used effectively by providing to the user a simple machine abstraction supported by optimizing compilation techniques. The user sees the systolic array as an array of sequential processors communicating asynchronously.

Monica Sin-Ling Lam, *A Systolic Array Optimizing Compiler* (thesis), CMU-CS-87-187, p. 2.

13. Obtain xerographic copies of the title pages of the journal articles in which (1) Binet published "Binet's formula" for Fibonacci numbers; (2) Chebyshev published "Chebyshev's inequality"; (3) Vandermonde published "Vandermonde's convolution". *15 points each*

(1) J. Binet, "Mémoire sur l'intégration des équations linéaires aux différences finies, d'un ordre quelconque, à coefficients variables," *Comptes Rendus hebdomadaires des séances de l'Académie des Sciences* (Paris) **17** (1843), 559–567. (2) P.-L. Tchébyshef, "Des valeurs moyennes," *Journal de Mathmatiques pures et appliquées*, series 2, **12** (1867), 177–184; that's a translation of the Russian original, which was "O srednikh velichinakh," *Matematicheskiĭ Sbornik'* **2** (1867), 1–9. Stanford's library doesn't own that journal, but copies exist at Berkeley, Brown, Columbia, Duke, Illinois, Penn, and Yale, as well as the Library of Congress, according to the National Union Catalog. With a friend at one of those places it would have been possible to fax the page (but nobody did). Karl Pearson, in *Biometrika* **12**, p. 285, said that he couldn't trace the Russian original "at all." The French version was reprinted in Chebyshev's *Œuvres*, volume 1, 685–694; the Russian original was reprinted in his *Polnoe Sobranie Sochineniĭ*, volume 2, 431–437 (and Stanford does own that). (3) A. Vandermonde, "Mémoire sur des irrationnelles de différens ordres avec une application au cercle," *Histoire de l'Académie Royale des Sciences* (1772), part 1, 71–72; *Mémoires de Mathématique et de Physique, Tirés des Registres de l'Académie Royale des Sciences* (1772), 489–498.

14. What are the next two numbers in the sequence 1, 1, 2, 5, 12, 35, 108, 369, ...? Who first computed them? Who first computed the values 108 and 369? *10 points each*

Sloane's *Handbook of Integer Sequences* identifies this as sequence #561, the number P_n of polyominoes made from n squares (possibly enclosing one or more blank squares). Sloane refers to a paper by W. F. Lunnon, "Counting polyominoes," *Computers in Number Theory* (Academic Press, 1971), 347–372; Lunnon discusses the history on pp. 356–357. Chasing down his references, we find that R. Read computed $P_9 = 1285$ in "Contributions to the cell growth problem," *Canadian Journal of Mathematics* **14** (1962), 1–20, where an incorrect value $P_{10} = 4466$ is stated; the correct value $P_{10} = 4655$ must therefore have been computed first by T. R. Parkin, L. J. Lander, and D. R. Parkin in unpublished work announced at the SIAM fall meeting in 1967 (according to Lunnon). Going back from Read, we find an article by Frank Harary, "Unsolved problems in the enumeration of graphs," *Magyar Tudományos Akadémia, Matematikai Kutató Intézetének, Közleményei* **5** (1960), 63–95, where he states that Golomb's incorrect claim $P_7 = 109$ was corrected by Stein, Walden, and Williamson, who also computed P_8. They did their calculations on the MANIAC II at Los Alamos, according to Read. Incidentally, the calculation of P_n seems to be fraught with difficulty, since Lunnon claims that Parkin et al. had P_{15} wrong.

15. Who coined the term 'Artificial Intelligence'? What was research in that field called previously? *15 points each*

John McCarthy chose it late in 1955, and used it in his grant application to the Rockefeller Foundation for the 1956 Dartmouth Summer Research Project on Artificial Intelligence. Minsky drafted his essay "Steps toward artificial intelligence" after that key conference. Previously the subject had been called 'automata studies'; see the book *Automata Studies*, edited by McCarthy and Shannon, in which W. Ross Ashby writes about 'machines with "synthetic" intellectual powers'. Another term, proposed by Newell and Simon, was 'complex information processing' (RAND report P-850); see their book *Human Problem Solving*, 883–884. McCarthy's recollections are documented in *Machines who think* by Pamela McCorduck, p. 96.

16. Who wrote the report STAN-CS-88-1233? What is that author's favorite color? *10 points each*

Ken Ross, our friendly TA, likes sky blue best (finger `kar @ polya`).

17. Suppose the words of English were alphabetized from right to left instead of from left to right, so that all words ending in `a` would come first, then all words ending in `b`, etc. What would be the last word in the dictionary? What words would immediately precede and follow `trivia`? Note: Abbreviations, proper nouns, and hyphenated words do not count. If your words are not commonly known, you must state their meaning and give the name of a standard English dictionary that lists them. *15 points each*

According to the 'Normal and reversed word list...' in the Math/CS library (PE1680 N6), which is based on Webster's Second Unabridged and other dictionaries, the last word is `bruzz`, a wheelwright's corner chisel. That dictionary contains the sequence `parathyroprivia`, `trivia`, `Opiconsivia`, `plenalvia`, `salvia`. The proper name `Opiconsivia` doesn't count; according to Webster's Second, `parathyroprivia` is a disease, a deficiency of hormones from the parathyroid glands; according to Chambers's Technical Dictionary, `plenalvia` is "impaction of the rumen of cattle"; and `salvia` is a genus of herbs that includes sage. Of these words, only `salvia` can be found in Webster's Third Unabridged. But there are better answers: The Oxford English Dictionary contains `vuzz`, a southern variant of furze (an evergreen shrub); the Official Scrabble Players' Dictionary mentions `lixivia`, the plural of lixivium—solutions obtained by lixiviation (also in OED).

18. Identify the computer language in which each of the following program fragments is written: *10 points each*

a. `+/0=100|VV>0`

APL (from Gilman and Rose, *APL*, exercise 8H).

b. **procedure** Innerproduct(a,b) Order:(k,p) Result:(y); **value** k; **integer** k,p; **real** y,a,b;
begin real s; $s := 0$; **for** $p := 1$ **step** 1 **until** k **do** $s := s + a \times b$; $y := s$ **end** Innerproduct

Algol 60 (from the original report, *CACM* **3** (1960), 311); reprinted in Horowitz, *Programming languages: A grand tour*.

c.
```
stacks←(Array new:3)collect:[:each|OrderedCollection new].
(height to: 1 by: -1)do:[:each|(stacks at: 1)addFirst:
  (Character value:($A asciiValue) + each - 1)].
```

Smalltalk (from Kaehler and Patterson, *A Taste of Smalltalk*, p. 45).

d. *linkage* **class** *link*;
begin procedure *out*;
if *suc* =/= *none* **then begin** *suc.pred* :− *pred*; *pred.suc* :− *suc*; *suc* :− *pred* :− *none* **end** ...**end**

SIMULA 67 (from Helmut Rolfing, *SIMULA*, p. 165).

e.
```
10100800
00E88C03
00000000
00000004
```

The ant language of Problem 5. (It also disassembles into valid but uninspiring 68000 code, but it is definitely not VAX code.)

f.
```
IF DAY EXCEEDS 31 THEN SUBTRACT 31 FROM DAY;
MOVE "APRIL" TO MONTH; OTHERWISE MOVE "MARCH" TO MONTH.
```

COBOL (from *CACM* **5** (1962), 210).

g.
```
Procedure Mguvar (x,y)
      Begin Includes(x,y) ==> Return(False),
            Return([x/y])
      End
```

Demonstration language in Genesereth and Nilsson, *Logical Foundations of Artificial Intelligence*, p. 68.

h. $top\ y_2 = top\ y3 = .45\ bot\ y_0;\ z_2 = whatever[z_1, z_{4r}];$

METAFONT (from Knuth's *METAFONTbook*, p. 164)

i.
```
R2      J60
      70  J8
      40  H0
      40  H0
          R2
      12  H0
          J65  J68
```

IPL-V (from Sammet, *Programming Languages*, p. 392).

j.
```
: SQUARE DUP *;
: CUBE DUP SQUARE *;
: FOURTH DUP CUBE *;
```

FORTH (from Churlian, *Beginning FORTH*, p37); note also `: BETTERFOURTH SQUARE SQUARE;`

k.
```
Für j=1(1)n :
hj-1+(aijbjk)⇒hj
Ende Index j
```

From Heinz Rutishauser, *Automatische Rechenplanfertigung...* (1952), p. 26.

l.
```
picnic(Day) :- holiday(Day,july_4), !.
picnic(Day) :- weather(Day,fair), weekend(Day).
```

Prolog (from Jean Rogers, *A Prolog Primer*, p. 118).

m. *Node* = **pointer to** *Object*;
Object = **record** *key*, *x*, *y*: **integer**; *left*, *right*: *Node* **end**;
Rectangle = **pointer to** *RectObject*;
RectObject = **record**(*Object*) *w*, *h*: **real end**;
... **if** *p* **is** *Rectangle* **then** *area* := *p*(*Rectangle*).*w* ∗ *p*(*Rectangle*).*h*; ...

Oberon (see N. Wirth, "From Modulo to Oberon," *Software—Practice & Experience* **18** (1988), 66–77). But in Oberon one must type the reserved words all in uppercase letters.

n.
```
/increase-x{xpos radius add /xpos exch def}def
/doCircle{xpos ypos radius 0 360 circ stroke}def
{xpos pagewidth le {doCircle increase-x}{exit}ifelse}loop
```

PostScript (from Adobe Systems, *PostScript Language Tutorial and Cookbook*, pp. 69–70).

o. $testr[x, p, f, u] \leftarrow$ **if** $p[x]$ **then** $f[x]$ **else**
if $atom[x]$ **then** $u[\,]$ **else**
$testr[cdr[x], p, f, \lambda{:}testr[car[x], p, f, u]]$.

McCarthy's publication language for LISP (from Wexelblatt, *History of Programming Languages*, p. 180); it is properly called M-language (see p. 177 of that book).

Scores:

Problem	Rajeev Alur Tom Henzinger* Sherry Listgarden Alex Wang*	Adam G Urs H Sanjoy M Daniel S	Eddie C Dinesh K Patrick L Michael Y	Arul M Steven P Alon L Robert K Roland C
1	30	30	30	18
2	15	15	15	15
3	30	10	30	30
4	60	20	20	60
5	35	15	20	10
6	30	10	50	10
7	43	30	10	10
8	80	120	25	75
9	35	26	35	16
10	50	30	40	30
11	25	25	25	25
12	75	50	0	0
13	40	40	30	20
14	30	35	10	20
15	30	15	30	20
16	20	1	20	20
17	30	40	40	45
18	62	72	22	51
Totals	720*	584†	452	475

*Successfully defending their championship performance of 1987
†The winning score from this year's CS304 students

Self-describing sequences and the Catalan family tree

Zoran Šunić
Department of Mathematics and Statistics
810 Oldfather Hall, University of Nebraska
Lincoln, NE 68588-0323, USA
zsunik@math.unl.edu

Submitted: March 19, 2002; Accepted: ?,? .
MR Subject Classifications: 05A15, 05C05, 11Y55

Abstract

We introduce a transformation of finite integer sequences, show that every sequence eventually stabilizes under this transformation and that the number of fixed points is counted by the Catalan numbers. The sequences that are fixed are precisely those that describe themselves — every term t is equal to the number of previous terms that are smaller than t. In addition, we provide an easy way to enumerate all these self-describing sequences by organizing them in a Catalan tree with a specific labelling system.

Prefix ordered sequences and rooted labelled trees

The following connection between prefix ordered sequences and rooted labelled trees is well known and we briefly mention only the instance which is useful for our considerations.

Let $\mathcal{A}$ be the set of finite integer sequences $a = (a_0, a_1, \dots)$ with the property that $0 \leq a_i \leq i$, for all indices. We order the sequences in $\mathcal{A}$ by the *prefix* relation, i.e.,

$$(a_0, a_1, \dots, a_n) \preceq (b_0, b_1, \dots, b_m)$$

if $n \leq m$ and $a_i = b_i$, for $i = 0, \dots, n$. The sequences in $\mathcal{A}$ can be organized in a rooted labelled tree $\mathcal{T}$ which reflects the prefix order relation. The root of the tree $\mathcal{T}$ is labelled by 0. Every vertex that is at distance n from the root has $n + 2$ children labelled by $0, 1, \dots, n, n+1$ (see Figure 1). The vertices whose distance to the root is n form the n-th *level* of the tree $\mathcal{T}$, which is also called the n-th *generation*. For every vertex v at the level n in the tree $\mathcal{T}$ there exist a unique path of length n from the root to v. The labels of the vertices on this path form a unique sequence $(a_0, a_1, \dots, a_n)$ in $\mathcal{A}$ that corresponds to the vertex v and this sequence is called the *full name* of v. The correspondence

$$v \leftrightarrow \text{the full name of } v$$

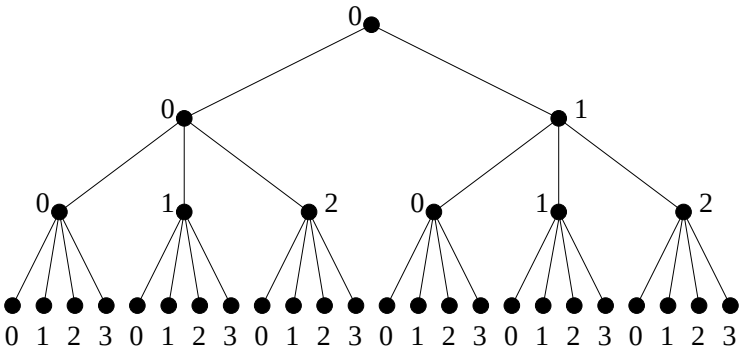


Figure 1: The rooted labelled tree $\mathcal{T}$ up to the third generation

provides a bijection between the vertices in $\mathcal{T}$ and the sequences in $\mathcal{A}$. Under this bijection, the vertices from the n-th generation in $\mathcal{T}$ correspond to the sequences of length $n+1$ in $\mathcal{A}$. The set of vertices in the n-th generation is denoted by $\mathcal{T}_n$ and the corresponding set of sequences by $\mathcal{A}_n$.

The sequence $a = (a_0, a_1, \ldots, a_n)$ is a prefix of the sequence $b = (b_0, b_1, \ldots, b_m)$ if and only if the vertex v_a with full name a is on the unique path between the root and the vertex v_b with full name b, i.e., if and only if the vertex v_a is an ancestor of the vertex v_b. Consider a graph endomorphism α of $\mathcal{T}$ that fixes the root (and therefore also preserves the levels). Such an endomorphism corresponds to a transformation of sequences $\alpha : \mathcal{A} \to \mathcal{A}$ that preserves the length of the sequences and also their prefix order, i.e.,

$$a \preceq b \qquad \text{implies} \qquad \alpha a \preceq \alpha b,$$

for all sequences a and b in $\mathcal{A}$.

In the sequel, we often deliberately blur the distinction between the vertices in $\mathcal{T}$ and the corresponding sequences in $\mathcal{A}$. Similarly, we do not distinguish tree endomorphisms of $\mathcal{T}$ fixing the root from sequence transformations that preserve the length and the prefix order. This mistake actually improves our presentation.

Let α be an endomorphism of $\mathcal{T}$. Since every generation in $\mathcal{T}$ is finite, the α *orbit*

$$\alpha^* u = \{\alpha^i u \mid i \geq 0\}$$

of every vertex u of $\mathcal{T}$ is finite. Thus, starting from any vertex, repeated applications of α produce *periodic points*, i.e., points a for which $\alpha^k a = a$ for some $k > 0$. The *period* of the periodic point a is the smallest k for which $\alpha^k a = a$. The points of period 1 are *fixed points* and the points of period dividing 2 are *double points*. Obviously, if u and v are periodic points of α and u is a prefix of v then the period of u divides the period of v.

Sometimes it is easy to estimate how long does it take before a periodic point is reached. We make use of the *lexicographical ordering* $\leq$ of the sequences in $\mathcal{A}_n$ (note the difference with the prefix ordering $\preceq$). Namely, for $a = (a_0, a_1, \ldots, a_n)$ and $b = (b_0, b_1, \ldots, b_n)$, set $a < b$ if $a_i < b_i$ at the first index where a and b differ.

Theorem 1. *Let α be an endomorphism of the tree $\mathcal{T}$ and assume that, for some $n \geq 1$, there exists $k \geq 1$ such that, for every vertex u in generation n, either*

$$u \leq \alpha^k u \leq \alpha^{2k} u \leq \dots$$

or

$$u \geq \alpha^k u \geq \alpha^{2k} u \geq \dots .$$

Then, starting from any point in generation n, repeated applications of α lead to a periodic point of period dividing k is reached in $O(n^2)$ steps.

Proof. We show that $\beta = \alpha^k$ reaches a fixed point in no more than

$$1 + 2 + \cdots + n = n(n+1)/2$$

steps.

Start with any vertex u in generation n. Without loss of generality we may assume

$$u \leq \beta u \leq \beta^2 u \leq \dots .$$

After the first application of β the initial segment up to index 1 of βu is fixed under β. After the next two steps the entry at index 2 will be fixed. Proceeding in the same fashion we see that the initial segment of $\beta^{1+2+\cdots+k} u$ up to index k is fixed under β. Indeed, once the initial segment up to index $k-1$ is fixed the entry at index k can go up no more than k times (from 0 to k) before it stabilizes. Thus, $\beta^{1+2+\cdots+n} u$ is fixed under β. □

Self-describing sequences

We define an endomorphism $\delta : \mathcal{A} \to \mathcal{A}$ transforming sequences in $\mathcal{A}$ by

$$(\delta a)_i = \#\{j \mid j < i,\ a_j < a_i\}.$$

Thus, for each term t in the sequence a, $(\delta a)_i$ counts the number of previous terms that are smaller than t. The transformation δ makes perfect sense even for sequences out of $\mathcal{A}$, but the image is in $\mathcal{A}$ and it stays there under further iterations. A sequence that is fixed under δ is called a *self-describing sequence*. Therefore, the sequence $a = (a_0, a_1, \dots)$ is self-describing if

$$\#\{j \mid j < i,\ a_j < a_i\} = a_i,$$

for all indices, i.e., every term t is equal to the number of previous terms that are smaller than t.

The Catalan family tree

We describe now a rooted labelled subtree of $\mathcal{T}$, denoted by $\mathcal{C}$ and called *the Catalan family tree* or just the *Catalan family*. The root vertex 0 belongs to $\mathcal{C}$. It has two children named 0 and 1 and we consider 0 the older sibling. The oldest sibling in this family always

has 2 children, the second oldest 3, the third oldest 4, and so on. The oldest child of a member of the family x gets named after the oldest sibling of x, the second oldest child after the second oldest sibling, and so on, until x uses its own name for its second to last child and n for the youngest one, where n is the generation number of the children (the level in the tree). The diagram in Figure 2 depicts the family members of $\mathcal{C}$ up to the third generation.

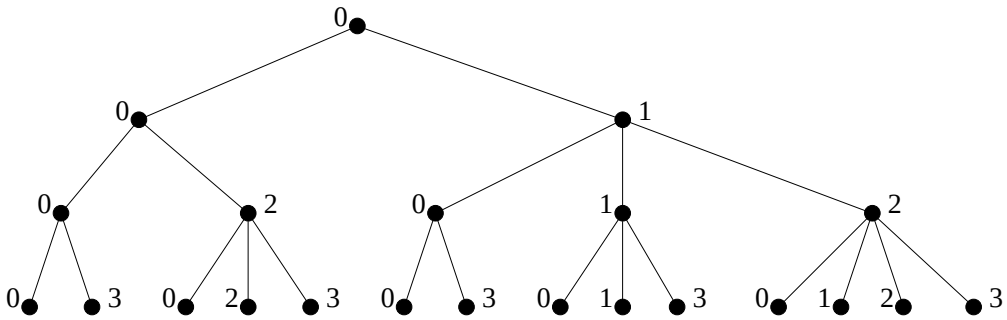


Figure 2: The Catalan family tree $\mathcal{C}$ up to the third generation

The connection

We establish now a connection between the self-describing sequences and the Catalan family tree.

Theorem 2. *The full names of the members of the Catalan family are precisely the self-describing sequences. In other words, they are the fixed points of the endomorphism δ.*

Moreover, repeated applications of δ to any sequence in $\mathcal{A}$ eventually produce a member of the Catalan family, i.e. a fixed point of δ. The number of applications needed to reach such a point is $O(n^2)$.

All statements of the theorem are implied by Theorem 1 and the following lemma.

Lemma 1. *If a is a member of the Catalan family then $a = \delta a$. Otherwise, $a < \delta a$.*

Proof. The proof is by induction on the generation number n. The statement is true for $n = 0$ and $n = 1$. Assume that the statement is true for all vertices up to the n-th generation.

Let

$$a = (a_0, a_1, \ldots, a_n, x)$$

be a $(n+1)$-st generation member of the Catalan family. We consider two cases.

If $x = n + 1$ then

$$\#\{j \mid j < n+1,\ a_j < x\} = \#\{j \mid j < n+1,\ a_j < n+1\} = n + 1 = x,$$

and a is a fixed point of δ.

If $x \neq n+1$, then $a_n \geq x$ and there exists an n-th generation member of the Catalan family whose full name is

$$a' = (a_0, a_1, \ldots, a_{n-1}, x),$$

namely the one after whom a was named. We have

$$\#\{j \mid j < n+1,\ a_j < x\} = \#\{j \mid j < n,\ a_j < x\} = x,$$

where the first equality comes from the fact that $a_n \geq x$ and the second from the inductive hypothesis, since $\delta a' = a'$.

Thus all members of the Catalan family are fixed under δ.

Now, let

$$a = (a_0, a_1, \ldots, a_n, x)$$

be a full name of a vertex in $\mathcal{T}$ in the n-th generation that is not a member of the Catalan family $\mathcal{C}$. If any proper prefix of a is not in $\mathcal{C}$ we obtain the claim directly from the inductive hypothesis. Thus we may assume that

$$a'' = (a_0, a_1, \ldots, a_n)$$

is a member of the Catalan family. Since a is not in $\mathcal{C}$ we have $a_n \neq x$ and $n+1 \neq x$. We consider two cases.

If $a_n > x$ then $a' = (a_0, a_1, \ldots, a_{n-1}, x)$ is not in $\mathcal{C}$ and

$$\#\{j \mid j < n+1,\ a_j < x\} = \#\{j \mid j < n,\ a_j < x\} > x,$$

where the equality comes from the fact that $a_n > x$ and the inequality from the inductive hypothesis.

If $a_n < x < n+1$ then

$$\#\{j \mid j < n+1,\ a_j < x\} = \#\{j \mid j < n,\ a_j < x\} + 1 \geq x+1,$$

where the equality comes from the fact that $a_n < x$ and the inequality from the inductive hypothesis. The equality in the last case is possible only when $a' = (a_0, a_1, \ldots, a_{n-1}, x)$ is in $\mathcal{C}$. □

We proceed by counting the self-describing sequences with fixed length. In addition, we obtain a result on the distribution of names in $\mathcal{C}$. Recall that the n-th Catalan number is equal to

$$c_n = \frac{1}{n+1}\binom{2n}{n}.$$

A recursive definition of the Catalan numbers is given by

$$c_0 = 1,$$
$$c_{n+1} = c_0 c_n + c_1 c_{n-1} + \cdots + c_n c_0.$$

Theorem 3. *The number of self-describing sequences in $\mathcal{A}_n$, i.e., the number of n-th generation members of the Catalan family is the $(n+1)-th$ Catalan number c_{n+1}.*

Moreover, for $r = 0, \ldots, n$, the number of n-th generation members of the Catalan family whose name is r is equal to $c_r c_{n-r}$.

Proof. Denote by z_n the number of n-th generation members of the Catalan family whose name is 0. More generally, for $r = 0, \ldots, n$ denote by $f_{n,r}$ the number of n-th generation members of the Catalan family whose name is r. Finally, denote by g_n the number of n-th generation members of the Catalan family.

Since the oldest child of every member of the Catalan family is named 0, we have, for all n,

$$z_{n+1} = g_n.$$

Since the youngest sibling in the r-th generation is always named r and the oldest 0 we also have, for all r,

$$f_{r,r} = f_{r,0} = z_r.$$

For some fixed r, consider the set of $f_{r,r}$ r-th generation members named r together with all their descendants in $\mathcal{C}$ whose names are greater or equal to r. This forest of $f_{r,r}$ identical subtrees of $\mathcal{C}$ contains all members of $\mathcal{C}$ whose name is r. Moreover, each tree in this forest looks exactly like the Catalan family tree, except that all labels are increased by r. Indeed, each r-th generation member od $\mathcal{C}$ named r has two children, named r and $r+1$, the oldest sibling always has two children, the second oldest three, etc. Thus, for any n and $r = 0, \ldots, n$, the number $f_{n,r}$ of n-th generation members of $\mathcal{C}$ named r is $f_{r,r}$ times larger than the number of $(n-r)$-th generation members of $\mathcal{C}$ named 0, i.e.,

$$f_{n,r} = f_{r,r} f_{n-r,0} = z_r z_{n-r}.$$

Since $z_0 = 1$ and

$$\begin{aligned} z_{n+1} = g_n &= f_{n,0} + f_{n,1} + \cdots + f_{n,n} \\ &= z_0 z_n + z_1 z_{n-1} + \cdots + z_n z_0 \end{aligned}$$

we conclude that, for all n, z_n is the $n-th$ Catalan number. The statements of the theorem follow now easily from the relations $g_n = z_{n+1}$ and $f_{n,r} = z_r z_{n-r}$. □

Connection to other Catalan trees and objects

It is well known that the Catalan numbers appear naturally under many circumstances. The exercises on Catalan numbers in [Sta99] provide a trove of examples, along with references, in which Catalan numbers count the number of objects of particular type and size. The self-describing sequences provide yet another example that we now relate to some other objects counted by the Catalan numbers.

Consider the sequences in $\mathcal{A}$ with the property that $a_{i+1} \leq a_i + 1$, for all indices (see the Exercise 6.19.u in [Sta99]). Such sequences are called *sequences with unit increase.*

The rooted labelled tree that corresponds to the set of sequences with unit increase looks the same as the Catalan family tree, just with a different labelling and we obtain an easy bijective correspondence between the self-describing sequences and the sequences with unit increase. We could use this bijective connection to show that the Catalan numbers count the number of self-describing sequences. Instead, we provided a direct proof of Theorem 3 and the reason is that there is an important difference in the distribution of labels in the Catalan family tree and the tree of the sequences with unit increase.

Theorem 4. *For $r = 0, \ldots, n$, the number of n-th generation vertices in the tree of sequences with unit increase labelled by r is*

$$\frac{r+1}{n+1}\binom{2n-r}{n}.$$

Proof. Let $a = (a_0, a_1, \ldots, a_n)$ be a sequence with unit increase. Following Exercise 6.19.u in [Sta99], we define, for $i = 0, \ldots, n-1$,

$$b_i = a_i - a_{i+1} + 1.$$

Construct a sequence of n 1's and $n - a_n$ negative 1's by replacing each b_i, $i = 0, \ldots, n-1$ by one 1 followed by b_i negative 1's. The newly obtained sequence has non-negative partial sums. The correspondence between the sequences in $\mathcal{A}_n$ with unit increase that end by r and the sequences of n 1's and $n - r$ negative 1's with non-negative partial sums is bijective. It is shown in [Bai96] that the number of sequences with non-negative partial sums that consist of n 1's and k negative 1's is equal to

$$\frac{n+1-k}{n+1}\binom{n+k}{n}$$

and this implies our claim. □

In passing, we make a slightly more general remark. Namely, for a fixed positive integer m, consider the sequences with the property that $a_0 = 0$ and $0 \leq a_{i+1} \leq a_i + m$, for all indices. Such sequences are called *sequences with m-increase*. We can easily construct the rooted labelled tree that corresponds to such sequences. For a sequence $(a_0, a_1, \ldots, a_n)$ with m-increase, define, for $i = 0, \ldots, n-1$,

$$b_i = a_i - a_{i+1} + m.$$

Following the same approach as before, construct a sequence of n m's and $n - a_n$ negative 1's by replacing each b_i, $i = 0, \ldots, n-1$ by one m followed by b_i negative 1's. The newly obtained sequence has non-negative partial sums and the correspondence between the sequences $(a_0, a_1, \ldots, a_n)$ with m-increase that end by r and the sequences of n 1's and $mn - r$ negative 1's with non-negative partial sums is bijective. Such sequences are discussed in [FS01], where simple recursive formulae for their number is provided.

Unfortunately, closed formulae are not provided yet, but we note that the number of n-th generation sequences with m-increase is given by $c_m(n+1)$ where

$$c_m(n) = \frac{1}{mn+1}\binom{(m+1)n}{n}.$$

The last displayed number is the generalization of the Catalan numbers which counts, for example, the number of rooted $(m+1)$-ary trees with n interior vertices.

It is worth nothing that Julian West [Wes95] recursively constructs a rooted labelled tree whose root is labelled by 2 and each vertex labelled by x has x children labelled by $2, 3, \ldots, x+1$. This tree, which West calls a Catalan tree, looks again exactly like the Catalan family tree, but with different labels. In fact, the tree of the sequences with unit increase can be obtained from the Catalan tree constructed by Julian West by decreasing all labels by 2.

Similarly, in the spirit of the Julian West construction, for any positive integer m, construct a rooted labelled tree whose root is labelled by $m+1$ and each vertex labelled by x has x children labelled by $m+1, m+2, \ldots, m+x$. The tree of sequences with m-increase can be obtained from this tree by decreasing all labels by $m+1$.

Mirror symmetry and mutually describing sequences

We introduce another endomorphism $\gamma : \mathcal{A} \to \mathcal{A}$ transforming sequences in $\mathcal{A}$ by

$$(\gamma a)_i = \#\{j \mid j < i,\ a_j \geq a_i\}.$$

Clearly $\gamma = \mu\delta$ where μ is the *mirror involution* of $\mathcal{A}$ given by

$$(\mu a)_i = i - a_i.$$

We call μ the mirror involution of $\mathcal{A}$ since μ mirrors the tree $\mathcal{T}$ through its vertical axis of symmetry.

The endomorphism γ is studied in [Šun02]. Clearly, γ has no fixed points other than the sequence (0). However, γ has a lot of double points. If a is a double point of γ then so is $b = \gamma a$. Moreover, then $\gamma b = a$ and the sequences a and b mutually describe each other.

Theorem 5 ([Šun02]). *Repeated applications of γ to any sequence in $\mathcal{A}$ eventually produce a double point of γ. The number of application needed to reach a double point in $\mathcal{A}_n$ is $O(n^2)$ and there are more than 2^n such points.*

The sequence that counts the number of double points of γ in the n-th generation starts as follows

$$1, 2, 4, 10, 26, 70, 216, \ldots$$

This sequence does not appear in the Encyclopedia of Integer Sequences [SP95] nor in the online version [Slo] as of January 2002. It is interesting that we have such a good

understanding of the fixed points of δ, via the Catalan family tree, but we were still not able to count the number of double points of the mirror related endomorphism $\gamma = \mu\delta$.

Some other endomorphisms leading to fixed or double points are studied in [Šun02]. For one of them, the set of double points of length n is in bijective correspondence with the Young tableaux of size n.

Acknowledgements

Thanks to Richard Stanley and Louis Shapiro for their interest and input.

References

[Bai96] D. F. Bailey, *Counting arrangements of* 1*'s and* −1*'s*, Math. Mag. **69** (1996), no. 2, 128–131.

[FS01] Darrin D. Frey and James A. Sellers, *Generalizing Bailey's generalization of the Catalan numbers*, Fibonacci Quart. **39** (2001), no. 2, 142–148.

[Slo] N. J. A. Sloane, `http://www.research.att.com/~njas/sequences/`.

[SP95] N. J. A. Sloane and Simon Plouffe, *The encyclopedia of integer sequences*, Academic Press Inc., San Diego, CA, 1995.

[Sta99] Richard P. Stanley, *Enumerative combinatorics. Vol. 2*, Cambridge University Press, Cambridge, 1999, With a foreword by Gian-Carlo Rota and appendix 1 by Sergey Fomin.

[Šun02] Zoran Šunić, *Younf tableaux and other mutually describing sequences*, preprint, 2002.

[Wes95] Julian West, *Generating trees and the Catalan and Schröder numbers*, Discrete Math. **146** (1995), no. 1-3, 247–262.

Automatic Asymptotics and Generating Functions

Bruno Salvy
INRIA Rocquencourt

September 16, 1992

[summary by Bruno Salvy]

Abstract

Computer algebra systems can be of help in the asymptotic analysis of combinatorial sequences. Several algorithms are presented, most of which have been implemented in Maple.

Introduction

We assume a sequence is given, either by its first terms or by a combinatorial description of a class of objects it enumerates. The main tool we use is the *generating function* of the sequence. The idea is to consider this formal power series as an analytic function. When the series has a non-zero radius of convergence, Cauchy's theory makes it possible to find an asymptotic estimate of the sequence we started with.

1. From the sequence to the series

The preferred method naturally depends on the available information concerning the sequence.

Empirical method. When only the first few terms of the sequence are known, there are *a priori* an infinite number of possible sequences, and there seems to be little sense in looking for an asymptotic behaviour. However, there is quite often a "simple" sequence defined by these first terms. This approach was initiated by F. Bergeron and S. Plouffe [2], who looked for Padé approximants of the generating series. When the number of non-zero coefficients of the Padé approximant is "significantly" smaller than the number of given terms of the sequence, it is natural to conjecture that the generating series is rational and that a closed-form was found. This method can be extended by applying it to the logarithmic derivative or to the functional inverse of the given power series, which yields nice generating functions.

With P. Zimmermann, we applied this idea of looking for a "simple" generating function given its first coefficients to the quest of "holonomic" sequences, i.e. sequences satisfying a linear recurrence with polynomial coefficients. Rather than looking for a Padé approximant, this recurrence is sought by an undeterminate coefficients method. When the number of non-zero coefficients of the recurrence is "sufficiently" smaller than the number of given terms, the recurrence is conjectured as being satisfied by the whole sequence. This is implemented in the Gfun package [12].

Both these methods are very efficient in practice. Among the approximately 6000 sequences of the next edition of Sloane's book [14], roughly 25% of the sequences are thus conjectured rational, and an extra 5% are conjectured holonomic non-rational [9].

Combinatorial method. A large number of sequences f_n enumerate the number of objects of size n in some *decomposable* combinatorial data-structure. This means that the structure can be expressed in terms of a small combinatorial toolbox comprising cartesian product, disjoint union, list, set, cycle and basic atoms. Thus the structure "functional graph" (the graph of an application of a set of n elements into itself) is

expressed as a set of connected components, these components being cycles of trees, these trees themselves being recursively defined as the cartesian product of a node (the root of the tree) by a set of trees.

The $\Lambda\Upsilon^{\Omega}$ system, developed jointly with P. Zimmermann and Ph. Flajolet [3, 4] implements a translation of these combinatorial specifications into equations relating the corresponding generating functions. In the example of functional graphs, the first part of the system will produce the following equations:

$$\mathrm{FuncGraph}(z) = \exp(\mathrm{comp}(z)), \quad \mathrm{comp}(z) = \log[1/(1-\mathrm{tree}(z))], \quad \mathrm{tree}(z) = z\,\exp(\mathrm{tree}(z)).$$

A second part of the system then attempts to find an explicit form of the generating function from this system. For, in its current state, the asymptotic part of the $\Lambda\Upsilon^{\Omega}$ system can only handle explicit generating functions. In this example, thanks to Maple's W function, the following "explicit" form is obtained:

$$\frac{1}{1+W(-z)}.$$

Conclusion. Two very different methods have been described to obtain the generating function of a sequence. The first one finds *holonomic* generating functions, i.e. solutions of linear differential equations with polynomial coefficients. The second one is more combinatorial and finds generating functions that obey functional equations expressed in terms of some "elementary" functions. In some cases, these equations can be solved.

Known algorithms to get "explicit" forms from these equations can be summarised as follows.

- Liouvillian solutions of linear differential equations can be obtained by Kovacic's algorithm for the case of order 2. This algorithm is (at least partially) implemented in most computer algebra systems. An algorithm due to M. Singer treats the general case, but is not practical. The third order has been made practical by F. Ulmer, but there is no generally available implementation;
- Hypergeometric solutions of linear differential equations can be found by an algorithm due principally to M. Petkovšek, without any limitation on the order of the equation [8];
- Elementary functional equations can only be solved in some special cases.

2. From generating functions to asymptotics

When the generating series defines an analytic function, Cauchy's formula yields the nth Taylor coefficient as

$$[z^n]f(z) = \frac{1}{2i\pi}\oint \frac{f(z)}{z^{n+1}}dz.$$

The path of integration is a closed contour containing the origin and no other singularity.

We are looking for an asymptotic estimate as n tends to infinity. First of all, Hadamard's rule implies that the coefficients grow roughly as $1/R^n$, where R is the radius of convergence. This relates the exponential growth of the Taylor coefficients of a generating function to the location of its singularities. Besides, simple functions whose coefficients are known, such as $1/(1-z)^{\alpha}$, give the intuition that sub-exponential growth of the coefficients is related to the local growth of the generating function in the neighbourhood of its singularity of smallest modulus. This can be made precise.

2.1. Singularity analysis. In 1878, G. Darboux treated the case of algebraic singularities. This result was extended by R. Jungen in 1934 to handle singularities in $(1-z)^{\alpha}\log^k(1-z)$, where k is a non-negative integer. Finally, Ph. Flajolet and A. Odlyzko [5] described the more general case where the exponents of $(1-z)$ and of the logarithm are complex numbers. These methods yield a full asymptotic expansion of the Taylor coefficients.

This leads to the following algorithm to find the asymptotic expansion of coefficients of a generating function.

(1) Locate the singularities of smallest modulus;
(2) Compute the expansion of the function in the neighbourhood of these singularities;
(3) Translate this expansion into the expansion of the coefficients.

The last step above is easy. We now insist on how the first two steps can be automated. This depends on the type of equation defining the generating function.

When the generating function is given as a solution to a linear differential equation, its singularities are found among the poles of the coefficients of the equation and the roots of its leading coefficient. Since the coefficients are polynomials, singularities in this case are therefore algebraic numbers. When the generating function is given explicitly in terms of elementary functions, it is easy to find a set of points containing the singularities by a recursive algorithm.

Then one has to compare the moduli of the singularities. Algebraic numbers can be compared by purely algebraic methods using resultants and Sturm sequences. It is also possible to make use of guaranteed numerical estimates, see [6]. In the more general case of elementary constants one is confined to heuristics, the problem being related to difficult questions of transcendency.

Once the dominant singularities have been located, one looks for the local behaviour of the generating function in the neighbourhood of these singularities. When the function is given explicitly as an exp-log function (functions built up from $\mathbb{Q}$ and x by field operation, exp and $x \mapsto \log|x|$), a recent algorithm due to J. Shackell [13] makes it possible to compute the local expansion. When the generating function is holonomic, the possible behaviours have been given by E. Fabry in 1885, and have the form

$$\exp[P(1/(1-(z/\rho)^{1/d}))](1-z/\rho)^{\alpha}\sum_{k=0}^{K}\phi_k(z)\log^k(1-z/\rho),$$

where ϕ_k are formal power series in $1-z/\rho$. Such local solutions can be determined automatically [15]. Once a basis of local solutions has been found, one has to find the right linear combination in terms of the first elements of the sequence. While these elements are given by the Taylor expansion of the function at the origin, we have a basis of local solutions at the singularity. Besides, the formal power series ϕ_k are generally divergent. One must then resort to the theory of resummation [1].

2.2. Saddle-point method. When the function is entire or has a singularity of a more "violent" type than a mere algebraico-logarithmic type, it is often possible to use a saddle-point method. Setting $h(z) = \log(f(z)) - (n+1)\log z$, the contour of Cauchy's integral is deformed to pass through a point (*the saddle-point*) where $h'(z) = 0$. With a few extra hypotheses, Cauchy's integral is then concentrated in the neighbourhood of the saddle-point and the integral can be approximated by a Gaussian. If we denote the saddle-point by R, the nth coefficient is then estimated as

$$[z^n]f(z) \approx \frac{f(R)}{R^{n+1}\sqrt{2\pi h''(R)}}.$$

To automate this method and the approximations it requires, one uses a theorem due to W. K. Hayman [7], which makes it possible to decide sufficient conditions under which the method applies. A last technical problem is that the saddle-point is often only available as an asymptotic expansion deduced from the equation $h'(R) = 0$. An algorithm to compute this expansion under very general conditions has been developed in [11].

Bibliography

[1] Balser (W.), Braaksma (B. L. J.), Ramis (J.-P.), and Sibuya (Y.). – Multisummability of formal power series solutions of linear ordinary differential equations. *Asymptotic Analysis*, vol. 5, 1991, pp. 27–45.

[2] Bergeron (F.) and Plouffe (S.). – Computing the generating function of a series given its first terms. *Journal of experimental mathematics*, 1993.

[3] Flajolet (P.), Salvy (B.), and Zimmermann (P.). – *Lambda-Upsilon-Omega: The 1989 CookBook*. – Research Report n° 1073, Institut National de Recherche en Informatique et en Automatique, August 1989. 116 pages.

[4] Flajolet (P.), Salvy (B.), and Zimmermann (P.). – Automatic average-case analysis of algorithms. *Theoretical Computer Science, Series A*, vol. 79, n° 1, February 1991, pp. 37–109.

[5] Flajolet (Philippe) and Odlyzko (Andrew M.). – Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics*, vol. 3, n° 2, 1990, pp. 216–240.

[6] Gourdon (Xavier) and Salvy (Bruno). – Asymptotics of linear recurrences with rational coefficients. In Barlotti (A.), Delest (M.), and Pinzani (R.) (editors), *Formal Power Series and Algebraic Combinatorics*, pp. 253–266. – 1993. Proceedings of FPACS'5, Florence (Italy).

[7] Hayman (W. K.). – A generalization of Stirling's formula. *Journal für die reine und angewandte Mathematik*, vol. 196, 1956, pp. 67–95.

[8] Petkovšek (Marko) and Salvy (Bruno). – Finding all hypergeometric solutions of linear differential equations. In Bronstein (Manuel) (editor), *ISSAC'93*. pp. 27–33. – ACM Press, July 1993.

[9] Plouffe (S.). – *Approximations de séries génératrices et quelques conjectures.* – Master's thesis, Université du Québec à Montréal, September 1992. Also available as Research Report 92-61, Laboratoire Bordelais de Recherche en Informatique, Bordeaux, France.

[10] Salvy (Bruno). – *Asymptotique automatique et fonctions génératrices.* – PhD thesis, École Polytechnique, 1991.

[11] Salvy (Bruno) and Shackell (John). – Asymptotic expansions of functional inverses. In Wang (Paul S.) (editor), *Symbolic and Algebraic Computation.* pp. 130–137. – ACM Press, 1992. Proceedings of ISSAC'92, Berkeley.

[12] Salvy (Bruno) and Zimmermann (Paul). – *Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable.* – Technical Report n° 143, Institut National de Recherche en Informatique et en Automatique, 1992. To appear in *ACM Transactions on Mathematical Software*.

[13] Shackell (John). – Growth estimates for exp-log functions. *Journal of Symbolic Computation*, vol. 10, December 1990, pp. 611–632.

[14] Sloane (N. J. A.). – *A Handbook of Integer Sequences.* – Academic Press, 1973.

[15] Tournier (Évelyne). – *Solutions formelles d'équations différentielles.* – Doctorat d'État, Université scientifique, technologique et médicale de Grenoble, 1987.

An algebraic characterization of the set of succession rules

Luca Ferrari * Elisa Pergola † Renzo Pinzani †
Simone Rinaldi †

"Qui dedit beneficium taceat; narret qui accepit" (Seneca)
Merci Maurice

Abstract

In this paper we will give a formal description of succession rules in terms of linear operators satisfying certain conditions. This representation allows us to introduce a system of *well-defined operations* into the set of *succession rules* and then to tackle problems of combinatorial enumeration simply by using operators instead of generating functions. Finally we will suggest several open problems whose solution should lead to an algebraic characterization of the set of succession rules.

1 Introduction

A *succession rule* Ω is a system consisting of an *axiom* (b), $b \in \mathbb{N}^+$, and a set of *productions*:

$$\{(k_t) \rightsquigarrow (e_1(k_t))(e_2(k_t)) \dots (e_{k_t}(k_t)) : \ t \in \mathbb{N}\},$$

where $e_i : \mathbb{N}^+ \to \mathbb{N}^+$, which explains how to derive the *successors* $(e_1(k))$, $(e_2(k))$, $\dots$ $(e_k(k))$ of any given label (k), $k \in \mathbb{N}^+$. In general for a succession rule Ω, we use the more compact notation

*Dipartimento di Matematica, Viale Morgagni, 67/A, Firenze. ferrari@math.unifi.it
†Dipartimento di Sistemi e Informatica, Via Lombroso 6/17, Firenze. {elisa, pinzani, rinaldi}@dsi.unifi.it

$$\begin{cases} (b) \\ (k) \rightsquigarrow (e_1(k))(e_2(k))\dots(e_k(k)), \end{cases} \tag{1}$$

to mean that there can be infinitely many productions in the system, but at most one for each integer $k \in \mathbb{N}^+$.

The positive integers (b), (k), $(e_i(k))$, are called *labels* of Ω. The rule Ω can be represented by means of a *generating tree*, that is a rooted tree whose vertices are the labels of Ω; (b) is the label of the root and each node labeled (k) has k sons labeled by $e_1(k), \dots, e_k(k)$ respectively, according to the production of (k) in (1). A succession rule Ω defines a sequence of positive integers $\{f_n\}_{n\geq 0}$, f_n being the number of the nodes at level n in the generating tree defined by Ω. By convention the root is at level 0, so $f_0 = 1$. The function $f_\Omega(x) = \sum_{n\geq 0} f_n x^n$ is the *generating function* determined by Ω.

One of the most common succession rules is that defining Schröder numbers [4], $1, 2, 6, 22, 90, 394$, M2898 in [12]:

$$\begin{cases} (2) \\ (2) \rightsquigarrow (3)(3) \\ (k) \rightsquigarrow (3)\dots(k+1), \qquad k \geq 3. \end{cases} \tag{2}$$

In Fig. 1 the first levels of the generating tree of (2) are shown. We refer to [3] for further details and examples.

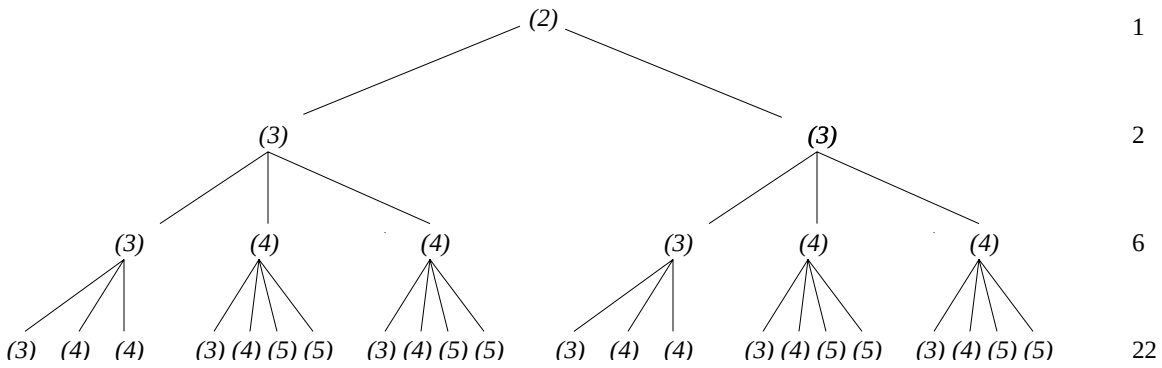


Figure 1: The first levels of the generating tree of (2), and its number sequence.

The concept of succession rule was first introduced in [6] by Chung et al. to study reduced Baxter permutations, and was later applied to the enumeration of permutations with forbidden subsequences [8, 13]. Moreover,

they represent an excellent tool for ECO method [3], which is a general method for the enumeration of combinatorial objects. The basic idea of this method is the following: given a class $\mathcal{O}$ of combinatorial objects and a parameter p of $\mathcal{O}$, let us consider the set $\mathcal{O}_n = \{x \in \mathcal{O} : p(x) = n\}$. If we are able to define an operator ϑ which satisfies the following conditions:

1. for each $Q \in \mathcal{O}_{n+1}$ there exists $P \in \mathcal{O}_n$ such that $Q \in \vartheta(P)$,
2. for each $P_1, P_2 \in \mathcal{O}_n$ such that $P_1 \neq P_2$, then $\vartheta(P_1) \cap \vartheta(P_2) = \emptyset$,

then $\mathcal{F}_{n+1} = \{\vartheta(P) : \forall\, P \in \mathcal{O}_n\}$ is a partition of $\mathcal{O}_{n+1}$. Therefore, we have a recursive construction of the elements of $\mathcal{O}$. A generating tree is then associated to the operator ϑ, in such a way that the number of nodes appearing in the tree at level n gives the number of n-sized objects in the class, and the sons of each object are the objects it produces through ϑ. Such a generating tree can be formally represented by means of a succession rule of the form (1), meaning that the root object has b sons, and the k objects $O'_1, \ldots, O'_k$, produced by an object O through ϑ are such that $|\vartheta(O'_i)| = e_i(k)$, $1 \leq i \leq k$.

A succession rule is called rational, algebraic or trascendental if its generating function is rational, algebraic or trascendental, respectively. The relationship between the structural properties of the rules and their rationality, algebraicity or trascendence is studied in [1].
However, the complete analytic characterization of the set of algebraic succession rules and of the set of algebraic generating functions remains an open problem.
In literature, succession rules can have several different forms. However, this paper will focus only on the rules having the form (1), where each label (k) produces exactly k sons, also named *ECO-systems*.

Two rules Ω_1 and Ω_2 are said to be *equivalent*, $\Omega_1 \cong \Omega_2$, if they define the same number sequence, that is $f_{\Omega_1}(x) = f_{\Omega_2}(x)$. For example, the following rules are equivalent to (2), and define the Schröder numbers [4, 5]:

$$\begin{cases} (2) \\ (2k) \rightsquigarrow (2)(4)^2 \dots (2k)^2(2k+2) \end{cases}$$

$$\begin{cases} (2) \\ (2) \rightsquigarrow (3)(3) \\ (2k-1) \rightsquigarrow (3)^2(5)^2 \dots (2k-1)^2(2k+1) \end{cases}$$

$$\begin{cases} (2) \\ (2^k) \rightsquigarrow (2)^{2^{k-1}}(4)^{2^{k-2}}(8)^{2^{k-3}} \dots (2^{k-1})^2(2^k)(2^{k+1}) \end{cases}$$

where the power notation is used to express repetitions, that is $(h)^i$ stands for $\underbrace{(h)\dots(h)}_{i \text{ times}}$.

Next we slightly extend the definition of succession rule given at the beginning, and introduce *colored rules* as follows: a rule Ω is colored when there are at least two labels (k) and $(\bar{k})$ having the same value but different productions. For example, it is easily proved, that the sequence $1, 2, 3, 5, 9, 17, 33, \dots, 2^{n-1}+1$, having

$$\frac{1-x-x^2}{1-3x+2x^2}$$

as generating function, can only be described by means of colored rules, such as:

$$\begin{cases} (2) \\ (1) \rightsquigarrow (\bar{2}) \\ (2) \rightsquigarrow (1)(2) \\ (\bar{2}) \rightsquigarrow (\bar{2})(\bar{2}). \end{cases} \tag{3}$$

In this paper we first solve two open problems on the set of finite succession rules. In Section 3, we introduce the concept of *rule operator* associated with a succession rule, that is, the algebraic counterpart of the combinatorial concept of succession rule: it is a linear operator on $\mathbb{R}[x]$, considered as an $\mathbb{R}$-vector space, and it gives us a formal tool to deal with ECO-systems from an algebraic view-point. Indeed it allows us to define some operations in the set of rule operators, reflecting some well-known operations on the number sequences associated with them.

2 Finite succession rules

A succession rule Ω is *finite* if it has a finite number of different labels. For example, for any positive integer, the number sequences $\{a_{n,k}\}_n$, defined by the recurrences:

$$\sum_{j=0}^{k}(-1)^j\binom{k}{j}a_{n-j,k}=0 \qquad k\in\mathbb{N},$$

having $\frac{1}{(1-x)^k}$ as generating function, have finite succession rules:

$$\Omega(k):\ \begin{cases}(k)\\ (1)\rightsquigarrow(1)\\ (2)\rightsquigarrow(1)(2)\\ (3)\rightsquigarrow(1)(2)(3)\\ \dots \qquad \dots\\ (k)\rightsquigarrow(1)(2)(3)\dots(k-1)(k).\end{cases}$$

Moreover, let $\{a_n\}_n$ be the sequence of integers satisfying the recurrence:

$$a_n=ka_{n-1}+ha_{n-2}, \qquad k\in\mathbb{N}^+, h\in\mathbb{Z},$$

subject to the initial conditions $a_0=1$, $a_1=b\in\mathbb{N}^+$; thus every term of the sequence is a positive number if $k+h>0$. In this case, the sequence $\{a_n\}_n$ is defined by the finite succession rule:

$$\Omega_{\mathcal{F}^b_{k,h}}:\ \begin{cases}(b)\\ (b)\rightsquigarrow(k)^{b-1}(k+h)\\ (k)\rightsquigarrow(k)^{k-1}(k+h)\\ (k+h)\rightsquigarrow(k)^{k+h-1}(k+h).\end{cases} \tag{4}$$

Finite succession rules play an important role in enumerative combinatorics, because of their strong relations with rational functions and regular languages; in particular they allow the enumeration of some restricted classes of combinatorial objects [9]. Let us first recall some basics about *PD0L systems* [11]. A PD0L system is a triple:

$$G=(\Sigma,h,w_0),$$

where Σ is an alphabet, h is an endomorphism defined on Σ^+ and w_0, named the *axiom*, is an element of Σ^+. The *language* of G is defined by:

$$L(G)=\{h^i(w_0):i\geq 0\}.$$

The function $f_G(n) = |h^n(w_0)|$, $n \geq 0$ is the *growth function* of G, and the sequence $|h^n(w_0)|$, $n \geq 0$ is termed *growth sequence*.

It is important to point out that we can regard any finite succession rule Ω as a particular PD0L system using the set of labels of Ω as the alphabet Σ, where h is defined by productions of Ω, and $w_0 \in \Sigma$. These remarks together with Theorem III.8.1 [11] lead us to the solution of the *equivalence problem* for finite succession rules.

Equivalence. *Let Ω_1 and Ω_2 be two finite succession rules having h_1 and h_2 labels respectively, then $\Omega_1 \cong \Omega_2$, if and only if the first $h_1 + h_2$ terms of the two sequences defined by Ω_1 and Ω_2 coincide.*

For example, let us consider the number sequences defined by (3) and by (4) with $b = 2, k = 1, h = 1$ (which is the rule for Fibonacci numbers). The sequences determined by (3) and (4) coincide for the first four terms, but not for the fifth.

Let $\mathcal{N}$ be the set of rational generating functions of positive sequences, $\mathcal{R}$ the set of generating functions of regular languages and $\mathcal{S}$ the set of generating functions of finite succession rules. The set of $\mathbb{N}$-rational functions $f(x)$, for which $f(0)$ equals 0 or 1, coincides with $\mathcal{R}$ [11]. Moreover, the analytic characterization of $\mathbb{N}$-rational functions is also given in [11]. With reference to [2], or by the methods of [11, 10], given a rational function $f(x)$, it is possible to establish whether $f(x) \in \mathcal{R}$. Furthermore, there are some examples of rational generating functions of positive sequences, which are not the generating functions of any regular language (see Section 5, [2]). Below, we state a result obtained through Theorem III.4.11 in [10], which gives an analytic characterization of the set of generating functions of PDOL growth sequences:

Generating functions. *The function $f(x)$ is the generating function of a finite succession rule if and only if:*

1. $f(x) = \frac{P(x)}{Q(x)}$, with $P(x), Q(x) \in \mathbb{Z}[x]$, and $Q(0) = P(0) = 1$;
2. $\frac{1}{x}(f(x) - 1) - f(x)$ is $\mathbb{N}$-rational.

This proves that each generating function of a finite succession rule is the generating function of a regular language, whereas the converse does not hold. For example, let $g(x) = \frac{1}{1-10x}$ and $h(x) = \frac{1-3x+36x^2}{(1-9x)(1+2x+81x^2)}$; $h(x)$ is a rational function having all positive coefficients (see [2] for the proof) but it is not $\mathbb{N}$-rational, since the poles of minimal modulus are complex numbers. Let

$$f(x) = g(x^2) + x[g(x^2) + h(x^2)] = k_1(x^2) + xk_2(x^2); \qquad (5)$$

$f(x)$ is $\mathbb{N}$-rational, since it is the merge in the sense of [10] of the two functions $k_1(x)$ and $k_2(x)$, each of them having a real positive dominating root, $x = 10$. This proves the existence of a regular language having $f(x)$ as its generating function. Moreover, it is clear that $f(x)$ defines a strictly increasing sequence of positive numbers. Neverthless $\frac{1}{x}(f(x) - 1) - f(x)$ is not $\mathbb{N}$-rational, since it is a merge of $g(x)$ and $h(x)$, and $h(x)$ is not $\mathbb{N}$-rational. Thus there are no finite succession rules having $f(x)$ as its generating function. We conclude that

$$\mathcal{S} \subset \mathcal{R} \subset \mathcal{N}.$$

The equivalence and the generating functions problems remain still open in the case of not finite succession rules.

3 Rule operators

In this section we introduce the concept of *rule operator*, which represents a simple algebraic tool to handle succession rules. This notion is not completely new in combinatorics, indeed it has been widely applied without a suitable algebraic formalization, especially when computing generating functions of succession rules [1, 3, 4].

Let us consider a succession rule having the form (1). We define the rule operator L_Ω associated with Ω as follows:

$$L_\Omega : \mathbb{R}[x] \to \mathbb{R}[x]$$

$$L_\Omega(1) = x^b;$$

$$L_\Omega(x^k) = x^{e_1(k)} + \ldots + x^{e_k(k)};$$

$$L_\Omega(k) = kx^k, \qquad \text{if the label } (k) \text{ is not in the generating tree of } \Omega\ ,$$

and then extending by linearity on $\mathbb{R}[x]$ (considered as a $\mathbb{R}$-vector space). In general, we use the power notation to express the iterated application of L_Ω: $L_\Omega^{n+1}(1) = L_\Omega(L_\Omega^n(1))$. In the sequel we will always write L in place of L_Ω, if not required by the context.

The following proposition characterizes the set of rule operators associated to ECO-systems:

Proposition 3.1 Let L be a linear operator on $\mathbb{R}[x]$. It is the rule operator associated with a ECO-system if and only if:

1) $L(x^k) \in \mathbb{N}[x]$, for all $k \in \mathbb{N}$;

2) $L(1) = x^b$, for some $b \in \mathbb{N}^+$;

3) $[L(x^k)]_{x=0} = 0$, $k \in \mathbb{N}$;

4) $[L(x^k)]_{x=1} = k$, $k \in \mathbb{N}$.

The linear operator L clearly retains the properties of the succession rule Ω; in particular, the sequence of positive integers $\{f_n\}$ defined by Ω can be easily obtained from L. We have the following proposition, which can be easily proved by induction on $n \in \mathbb{N}$:

Proposition 3.2 For any $n \in \mathbb{N}$ we have:

1) $f_n = [L^{n+1}(1)]_{x=1}$;

2) $f_n = [DL^n(1)]_{x=1}$;

where D is the derivative operator in the variable x.

We remark that condition 4) of Proposition 3.1 implies $[L^{n+1}(1)]_{x=1} = [DL^n(1)]_{x=1}$, as stated in Proposition 3.2.

Example 3.1 We present a small catalogue of ECO-systems and the corresponding rule operators associated with sequences of combinatorial interest. The identification numbers refer to [12].

Number sequence	ECO-system	rule operator
Fibonacci (**M0692**)	$\begin{cases} (2) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (1)(2); \end{cases}$	$L(1) = x^2, L(x) = x^2,$ $L(x^2) = x + x^2$
Factorial (**M1675**)	$\begin{cases} (2) \\ (k) \rightsquigarrow (k+1)^k \end{cases}$	$L(1) = x^2,$ $L(x^k) = kx^{k+1} = x^2 D(x^k)$;
Arrangements (**M1497**)	$\begin{cases} (2) \\ (k) \rightsquigarrow (k)(k+1)^{k-1} \end{cases}$	$L(1) = x^2,$ $L(x^k) = x^k + (k-1)x^{k+1}$;
Involutions (**M1221**)	$\begin{cases} (2) \\ (k) \rightsquigarrow (k-1)^{k-1}(k+1) \end{cases}$	$L(1) = x^2,$ $L(x^k) = (k-1)x^{k-1} + x^{k+1}$;
Bell (**M1484**)	$\begin{cases} (2) \\ (k) \rightsquigarrow (k)^{k-1}(k+1) \end{cases}$	$L(1) = x^2,$ $L(x^k) = (k-1)x^k + x^{k+1}$;
Catalan (**M1459**)	$\begin{cases} (2) \\ (k) \rightsquigarrow (2)(3)\ldots(k)(k+1) \end{cases}$	$L(1) = x^2,$ $L(x^k) = x^2 + \ldots + x^{k+1}$;
Motzkin (**M1184**)	$\begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (k) \rightsquigarrow (1)(2)\ldots(k-1)(k+1) \end{cases}$	$L(1) = x, L(x) = x^2,$ $L(x^k) = x + \ldots + x^{k-1} + x^{k+1}$.

Now we aim at extending the concept of rule operator also to the set of colored succession rules. Consider a 2-colored succession rule Ω written as follows:

$$\begin{cases} (a) \\ (h) \rightsquigarrow (e_1(h))(e_2(h))\dots(e_\alpha(h))(\overline{e_{\alpha+1}(h)})\dots(\overline{e_h(h)}) \\ (\overline{k}) \rightsquigarrow (c_1(k))(c_2(k))\dots(c_\beta(k))(\overline{c_{\beta+1}(k)})\dots(\overline{c_k(k)}). \end{cases} \tag{6}$$

The 2-colored operator L_Ω associated with (6) is then:

$$L_\Omega : \mathbb{R}[x] \oplus y\mathbb{R}[y] \to \mathbb{R}[x] \oplus y\mathbb{R}[y]$$

$$L_\Omega(1) = x^a;$$

$$L_\Omega(x^h) = x^{e_1(h)} + \dots + x^{e_\alpha(h)} + y^{e_{\alpha+1}(h)} + \dots + y^{e_h(h)};$$

$$L_\Omega(y^k) = x^{c_1(k)} + \dots + x^{c_\beta(k)} + y^{c_{\beta+1}(k)} + \dots + y^{c_k(k)}.$$

extended by linearity on the vector space $\mathbb{R}[x] \oplus y\mathbb{R}[y]$. Of course, this definition generalizes to n-colored rules. Operators for 2-colored rules possess analogous properties to those already stated for rule operators in the first part of this section.

Proposition 3.3 The linear operator L on $\mathbb{R}[x] \oplus y\mathbb{R}[y]$ is the rule operator of a 2-colored ECO-system if and only if the following conditions are satisfied:

1) $L(x^k), L(y^k) \in \mathbb{N}[x]$, for all $k \in \mathbb{N}$;

2) $[L(x^k)]_{x=y=0} = [L(y^k)]_{x=y=0} = 0$ for all $k \in \mathbb{N}$;

3) $[L(x^k)]_{x=y=1} = [L(y^k)]_{x=y=1} = k$ for all $k \in \mathbb{N}$.

Proposition 3.4 Let Ω be a 2-colored ECO-system, L the associated 2-colored rule operator, and $\{f_n\}$ the sequence defined by Ω. We have:

$$f_n = \left[L^{n+1}(1)\right]_{x=y=1} = \left[(D_x + D_y)L^n(1)\right]_{x=y=1},$$

for $n \in \mathbb{N}$, where D_x and D_y denote the partial derivative operators with respect to x and y, respectively.

4 Operations on succession rules

Now we aim at defining some operations, to be carried out on the set of rule operators, which reflect some well-known operations on the related number sequences. Let L_Ω and $L_{\Omega'}$ be two rule operators, associated to the succession rules Ω and Ω', defining the sequences $\{f_n\}_n$ and $\{g_n\}_n$, and having $f(x)$ and $g(x)$ as generating functions, respectively. Below we will deal with L_Ω and L'_Ω having the following general forms:

$$\begin{cases} L_\Omega(1) = x^a \\ \\ L_\Omega(x^h) = x^{e_1(h)} + x^{e_2(h)} + \ldots + x^{e_h(h)}, \end{cases}$$

$$\begin{cases} L_{\Omega'}(1) = x^b \\ \\ L_{\Omega'}(x^k) = x^{c_1(k)} + x^{c_2(k)} + \ldots + x^{c_k(k)}. \end{cases}$$

4.1 Sum of rule operators

Given two rule operators L_Ω and $L_{\Omega'}$, their *sum*, $L_\Omega \oplus L_{\Omega'}$, is the rule operator defining the sequence $\{h_n\}_n$ such that $h_0 = 1$ and $h_n = f_n + g_n$, when $n > 0$, and having $f(x) + g(x) - 1$ as generating function. We define:

$$L_\Omega \oplus L_{\Omega'} : \mathbb{R}[x] \oplus y\mathbb{R}[y] \oplus z\mathbb{R}[z] \to \mathbb{R}[x] \oplus y\mathbb{R}[y] \oplus z\mathbb{R}[z]$$

$$L_\Omega \oplus L_{\Omega'}(1) = z^{a+b},$$

$$L_\Omega \oplus L_{\Omega'}(z^{a+b}) = L_\Omega(x^a) + L_{\Omega'}(y^b),$$

$$L_\Omega \oplus L_{\Omega'}(x^h) = L_\Omega(x^h),$$

$$L_\Omega \oplus L_{\Omega'}(y^k) = L_{\Omega'}(y^k).$$

If we define $L_\Omega \oplus L_{\Omega'}$ as the identity on the remaining powers of x, y, z, and then we extend it by linearity, we obtain the desired rule operator which defines the sequence $\{h_n\}_n$.

4.2 Product of succession rules

Given two rule operators L_Ω and $L_{\Omega'}$, their *product* , $L_\Omega \otimes L_{\Omega'}$, is the rule operator defining the sequence $\left\{\sum_{k\leq n} f_{n-k}g_k\right\}_n$, and having $f(x) \cdot g(x)$ as generating function. We define:

$$L_\Omega \otimes L_{\Omega'} : \mathbb{R}[x] \oplus y\mathbb{R}[y] \to \mathbb{R}[x] \oplus y\mathbb{R}[y]$$

$$L_\Omega \otimes L_{\Omega'}(1) = x^{a+b},$$

$$L_\Omega \otimes L_{\Omega'}(x^{h+b}) = x^b L_\Omega(x^h) + L_{\Omega'}(y^b),$$

$$L_\Omega \times L_{\Omega'}(y^k) = L_{\Omega'}(y^k).$$

We will prove that:

$$\left[\left(L_\Omega \otimes L_{\Omega'}\right)^{n+1}(1)\right]_{x=y=1} = \sum_{k\leq n} f_{n-k}g_k.$$

Since $\left(L_\Omega \otimes L_{\Omega'}\right)(x^b p(x)) = \left(L_\Omega \otimes L_{\Omega'}\right)\left(\sum_k p_{n,k}x^{k+b}\right) =$
$\sum_k p_{n,k}(x^b L_\Omega(x^k) + L_{\Omega'}(y^b)) = x^b L_\Omega(p(x)) + p(1)L_{\Omega'}(y^b)$,
Lemma 4.1 follows:

Lemma 4.1 For each polynomial $p(x) = \sum_{k=1}^m p_{n,k}x^k$, we have:

$$\left(L_\Omega \otimes L_{\Omega'}\right)(x^b p(x)) = x^b L_\Omega(p(x)) + p(1)L_{\Omega'}(y^b).$$

Proposition 4.1 For each $n \in \mathbb{N}$, we have

$$\left(L_\Omega \otimes L_{\Omega'}\right)^n(1) = x^b L_\Omega^n(1) + \sum_{k=1}^{n-1}\left[L_\Omega^k(1)\right]_{x=1} \cdot L_{\Omega'}^{n+1-k}(1).$$

Proof. We work by induction on $n \in \mathbb{N}$. It is easy to show that the statement holds for $n = 1, 2, 3$. Supposing it holds for a fixed n, then we have:

$$\left(L_\Omega \otimes L_{\Omega'}\right)^{n+1}(1) = \left(L_\Omega \otimes L_{\Omega'}\right)\left(L_\Omega \otimes L_{\Omega'}\right)^n(1) = \left(L_\Omega \otimes L_{\Omega'}\right)(x^b L_\Omega^n(1) +$$

$$\sum_{k=1}^{n-1}[L_\Omega^k(1)]_{x=1} \cdot L_{\Omega'}^{n+1-k}(1)) = x^b L_\Omega^{n+1}(1) + [L_\Omega^n(1)]_{x=1}L_{\Omega'}^2(1) +$$

$$\sum_{k=1}^{n-1}[L_\Omega^k(1)]_{x=1} \cdot L_{\Omega'}^{n+2-k}(1)) = x^b L_\Omega^{n+1}(1) + \sum_{k=1}^{n}[L_\Omega^k(1)]_{x=1} \cdot L_{\Omega'}^{n+2-k}(1).\square$$

Corollary 4.1 For each $n \in \mathbb{N}$, we have

$$\left[\left(L_\Omega \otimes L_{\Omega'}\right)^{n+1}(1)\right]_{x=y=1} = \sum_{k\leq n} f_{n-k}g_k.$$

In a completely similar way it can also be proved that

$$[(D_x + D_y)(L_\Omega \otimes L_{\Omega'})^n(1)]_{x=y=1} = \sum_{k \leq n} f_{n-k} g_k.$$

Example 4.1 i) *Product of Catalan and Fibonacci numbers.* The rule operator obtained by applying the previously defined operation $\otimes$ to the rule operators for Catalan and Fibonacci numbers (see Example 3.1) is:

$$L_{\mathcal{C}} \otimes L_{\mathcal{F}}(1) = x^4$$

$$L_{\mathcal{C}} \otimes L_{\mathcal{F}}(x^{k+2}) = x + x^2 + x^4 + x^5 + \ldots + x^k + x^{k+1}$$

$$L_{\mathcal{C}} \otimes L_{\mathcal{F}}(x) = x^2$$

$$L_{\mathcal{C}} \otimes L_{\mathcal{F}}(x^2) = x + x^2.$$

and it defines the number sequence $1, 4, 12, 35, 103, 312....$ The reader can check that in this case the product can be expressed with no need of other variables.

ii) *The rule operator for the n-th power Catalan numbers.* We want to prove that the rule operator $L_{\mathcal{C}}^n$ for the sequence defined by $C(x)^n$ is the following:

$$\begin{aligned} &L_{\mathcal{C}}^n(1) = x^n \\ &L_{\mathcal{C}}^n(x^k) = L_{\mathcal{C}}(x^k) = x^2 + x^3 + x^4 + \ldots + x^k + x^{k+1}. \end{aligned} \tag{7}$$

We can prove this statement inductively, supposing it holds for $n \in \mathbb{N}$, and therefore verifying it for $n+1$. Since $L_{\mathcal{C}}^{n+1} = L_{\mathcal{C}} \otimes L_{\mathcal{C}}^n$, we have $L_{\mathcal{C}}^{n+1}(1) = x^{n+1}$. Moreover we have:
$L_{\mathcal{C}}^{n+1}\left(x^{k+1}\right) = L_{\mathcal{C}} \otimes L_{\mathcal{C}}^n\left(x^{k+1}\right) = xL_{\mathcal{C}}^n(x^k) + L_{\mathcal{C}}(x) = x^2 + x^3 + x^4 + \ldots$
$+x^h + x^{h+1} + x^{h+2} = L_{\mathcal{C}}^{n+1}(x^{k+1})$.

4.3 The Star of a rule operator

The *star* of the rule operator L_Ω is denoted as L_Ω^*, briefly L^*, and it is the operator defining the number sequence having

$$g(x) = \frac{1}{1 - f_0(x)} = 1 + f_0(x) + f_0^2(x) + \ldots + f^n(x) + \ldots = \sum_{n \geq 0} f_0^n(x)$$

as its generating function, where $f_0(x) = f(x) - 1$. Set $L(1) = x^a$, the operator L^* is defined as:

$$\begin{aligned} L^*(1) &= x^a = L(1) \\ L^*(x^a) &= x^a L(x^a) = L(1)L^2(1) \\ L^*(x^{a+h}) &= x^a(L(x^a) + L(x^h)). \end{aligned} \tag{8}$$

We then prove that, for every $n \in \mathbb{N}$:

$$\left[(L^*)^{n+1}(1)\right]_{x=1} = [x^n]g(x),$$

where $[x^n]g(x)$ indicates, as usual, the coefficient of x^n in $g(x)$.

Lemma 4.2 For every polynomial $p(x) \in \mathbb{R}[x]$ such that $\deg p(x) \geq 1$, we have:

$$L^*\left(x^a p(x)\right) = x^a \left(L^2(1)p(1) + L(p(x))\right).$$

Proof. Let $p(x) = \sum_{k=1}^n p_{nk} x^k$. Therefore we have:

$$L^*\left(x^a p(x)\right) = L^*\left(\sum_k p_{nk} x^{a+k}\right) = \sum_k p_{nk}\left(x^a\left(L(x^a) + L(x^k)\right)\right) =$$

$$x^a\left(L^2(1)p(x) + L(p(x)\right). \qquad \square$$

Recall that the coefficients g_n of the generating function $g(x) = \sum_n g_n x^n$ satisfy the recurrence relation:

$$\begin{aligned} &g_0 = 1 \\ &g_n = f_0 g_{n-1} + f_1 g_{n-2} + \ldots + f_{n-1} g_1 = \textstyle\sum_{k=1}^{n-1} f_k g_{n-k}, \qquad n \geq 1. \end{aligned} \tag{9}$$

From Lemma 4.2 and (9) we have:

Proposition 4.2 For any $n \in \mathbb{N}$, the following identity holds:

$$(L^*)^n(1) = x^a \sum_{k=2}^{n} \left(L^k(1)\left[(L^*)^{n+1-k}\right]_{x=1}\right). \tag{10}$$

Proof. For $n = 2, 3$ the identity (10) clearly holds. Now, if we suppose it holds for $n \in \mathbb{N}$, we immediately have:

$$\begin{aligned}
(L^*)^{n+1}(1) &= L^*((L^*)^n(1)) = L^*\left(x_a \cdot \frac{(L^*)^n(1)}{x^a}\right) \\
&= x^a\left(L^2(1)\left[L^{*n}(1)\right]_{x=1} + L\left(\frac{L^{*n}(1)}{x^a}\right)\right) \\
&= x^a\left(L^2(1)\left[L^{*n}(1)\right]_{x=1} + \sum_{k=2}^{n} L^{k+1}(1)\left[L^{*n+1-k}(1)\right]_{x=1}\right) \\
&= x^a \sum_{k=2}^{n+1}\left(L^k(1)\left[(L^*)^{n+2-k}\right]_{x=1}\right). \qquad \square
\end{aligned}$$

Corollary 4.2 For any $n \in \mathbb{N}$ we have $\left[L^{*n+1}(1)\right]_{x=1} = g_n$.

Example 4.2 i) *The star of Catalan numbers.* The rule operator $L^*_{\mathcal{C}}$ is:

$$L^*_{\mathcal{C}}(1) = x^2$$

$$L^*_{\mathcal{C}}(x^2) = x^4 + x^5$$

$$L^*_{\mathcal{C}}(x^{k+2}) = 2x^4 + 2x^5 + x^6 + x^7 + \ldots + x^k + x^{k+1} + x^{k+2} + x^{k+3}.$$

and it defines the sequence $1, 2, 9, 42, 199, \ldots$, having

$$\frac{1}{1 - \left(\frac{1-2x-\sqrt{1-4x}}{2x^2} - 1\right)}$$

as its generating function.

ii) *The star of Schröder numbers.* Consider the rule operator $L_{\mathcal{S}}$:

$$L_{\mathcal{S}}(1) = x^2$$

$$L_{\mathcal{S}}(x^{2k}) = x^2 + 2x^4 + 2x^6 + \ldots + 2x^{2k} + x^{2k+2}.$$

We get:

$$L^*_{\mathcal{S}}(1) = x^2$$

$$L^*_{\mathcal{S}}(x^2) = x^4 + x^6$$

$$L^*_{\mathcal{S}}(x^{2k+2}) = 2x^4 + 3x^6 + \ldots + 2x^{2k+2} + x^{2k+4}.$$

4.4 Partial sum of a succession rule

Let L be a rule operator and $\{f_n\}_n$ its associated sequence, having $f(x)$ as generating function. The *partial sum* ΣL, is the rule operator leading to the sequence $\{F_n\}_n = \left\{\sum_{j\leq n} f_j\right\}_n$. We can obtain ΣL by means of the product operation, since $F(x) = \sum_n F_n x^n = \frac{1}{1-x} \cdot f(x)$. Thus:

$$\Sigma L = L_1 \otimes L,$$

where L_1 is the rule operator for the sequence $f_n = 1$, for all n, that is:

$$\begin{cases} L(1) = x \\ L(x^k) = kx^k. \end{cases}$$

By applying the product operation we have:

$$\Sigma L(1) = x^{a+1}$$

$$\Sigma L(x) = x$$

$$\Sigma L(x^{h+1}) = x(1 + x^{e_1(h)} + \ldots + x^{e_n(h)})) = x(1 + L(x^h)).$$

This result can also be obtained by proving explicitely the following proposition:

Proposition 4.3 For any $n \in \mathbb{N}$ we have:

$$(\Sigma L)^n (1) = x \left(\sum_{i=1}^{n-1} \left[L^i(1)\right]_{x=1} + L^n(1) \right).$$

For example, the rule operator $L_{\mathcal{C}}$ for Catalan numbers leads to the operator:

$$\Sigma L_{\mathcal{C}}(1) = x^3,$$

$$\Sigma L_{\mathcal{C}}(x) = x,$$

$$\Sigma L_{\mathcal{C}}(x^{h+1}) = x + x^3 + x^4 + \ldots + x^{h+1} + x^{h+2},$$

giving the sequence $1, 3, 8, 22, 64, \ldots$
Moreover, it is easy to prove the following property.

Proposition 4.4 Let L be a rule operator defining the sequence $\{f_n\}_n$. Then a rule operator L' defining a sequence $\{g_n\}_n$, such that $f_n = g_n - rg_{n-1}$, for $n > 1$, exists:

$$\begin{cases} L'(1) = x^{a+r} = x^r L(1) \\ \\ L'(x^r) = rx^r \\ \\ L'(x^{h+r}) = rx^r + x^r L(x^h). \end{cases}$$

Proof. We first prove that for any $n \in \mathbb{N}$,

$$\left(L'\right)^n (1) = x^r \left(\sum_{i=1}^{n-1} r^{n-1} \left[L^i(1)\right]_{x=1} + L^n(1) \right). \tag{11}$$

From (11) we immediately obtain:

$$\left(L'\right)^{n+1} (1) - r(L')^n(1) = x^r \left(r \left[L^n(1)\right]_{x=1} + L^{n+1}(1) - rL^n(1) \right),$$

and then:

$$\left[\left(L'\right)^{n+1} (1) - rL'^n(1) \right]_{x=1} = \left[L^{n+1}(1)\right]_{x=1} = f_n. \qquad \square$$

Proposition 4.5 Let L be a succession rule, defining the sequence $\{f_n\}_n$ and let $L^2(1) - L(1) \in \mathbb{N}[x]$. Then there is a rule operator L' defining the sequence $\{g_n\}_n$ such that $g_0 = 1$, and $g_n = f_n - f_{n-1}$, for $n \geq 1$.

Sketch of proof. Let us consider the following rule operator:

$$L' : \begin{cases} L'(1) = \frac{L(1)}{y} \\ \\ L'\left(\frac{L(1)}{y}\right) = L^2(1) - L(1) \\ \\ L'(x^k) = L(x^k) \end{cases}$$

and let g_n be the sequence described by L'. By applying the sum operation, we easily conclude that:

$$L = L' \oplus xL.$$

Finally, L' defines a sequence for $g_n = \begin{cases} 1 & \text{if } n = 0; \\ f_n - h_n = f_n - f_{n-1} & \text{otherwise.} \end{cases}$ $\square$

Example 4.3 Let $L_{\mathcal{S}}$ be the rule for Schröder numbers:

$$L_{\mathcal{S}}: \quad \begin{cases} L_{\mathcal{S}}(1) = x^2 \\ \\ L_{\mathcal{S}}(x^{2h}) = x^2 + 2x^4 + \ldots + 2x^{2h} + x^{2h+2}. \end{cases}$$

The rule operator L_1,

$$L_1(1) = x^3,$$

$$L_1(x) = x,$$

$$L_1(x^{2h+1}) = x + x^3 + 2x^5 + \ldots 2x^{2h+1}x^{2h+3},$$

defines such a sequence $\{g_n\}_n = \{1, 3, 9, 31, 121, 515, \ldots\}$, that $f_n = g_n - g_{n-1}$, where f_n denotes the nth Schröder number. Moreover, since the rule operator $L_{\mathcal{S}}$ satisfies the hypotheses of Proposition 4.5, there is a rule operator L' defining the sequence k_n such that $k_0 = 1$, and $k_n = f_n - f_{n-1}$ for $n > 0$, that is the sequence $\{1, 1, 4, 16, 68,$
$304, 1412, \ldots\}$ (sequence M3521 in [12]):

$$\begin{cases} L'(1) = x, \\ \\ L'(x) = x^4 \\ \\ L'(x^{2h}) = x^2 + 2x^4 + \ldots + 2x^{2h} + x^{2h+2}. \end{cases}$$

5 Open problems

There are several open problems related to the definition of an algebra of succession rules which, in turn, lead to problems concerning the set of rule operators. Below an overview of the most interesting problems is given:

- **Other operations.**

  Subtraction Let us consider two rule operators L_Ω and $L_{\Omega'}$, defining the sequences $\{f_n\}$ and $\{g_n\}$ respectively. Moreover, let $L_\Omega \ominus L_{\Omega'}$ be the rule operator defining the sequence $\{h_n\}_n$ such that $h_n = \begin{cases} 1 & \text{if } n = 0 \\ |f_n - g_n| & \text{otherwise.} \end{cases}$

The construction of the operator $L_\Omega \ominus L_{\Omega'}$ presents an open problem.

Hadamard product Let L_Ω and $L_{\Omega'}$ be rule operators and, as usual, $\{f_n\}_n$ and $\{g_n\}_n$ be their sequences, with their respective generating functions $f(x)$ and $g(x)$. The *Hadamard product* of L_Ω and $L_{\Omega'}$, denoted as $L_\Omega \odot L_{\Omega'}$, is the rule defining the sequence $\{f_n g_n\}_n$. It is generally quite difficult to determine the generating function $f(x) \odot g(x)$, although the Hadamard product of two $\mathbb{N}$-rational series has been proved to be $\mathbb{N}$-rational [11]. The problem lies in the construction of the rule operator $L_\Omega \odot L_{\Omega'}$. However, we can prove that, in the case of finite rules, it is possible to determine a rule defining the Hadamard product. More precisely we can state that *the Hadamard product of two finite rules is a finite rule.*

Here is an example of our technique: let Ω be the rule for Pell numbers, $\{1, 2, 5, 12, 29, \dots\}$, and $L_{\Omega'}$ be the rule for the Fibonacci numbers having an odd index, $\{1, 2, 5, 13, 34, \dots\}$,

$$\Omega: \quad \begin{cases} (2) \\ (2) \rightsquigarrow (2)(3) \\ (3) \rightsquigarrow (2)(2)(3), \end{cases} \qquad \Omega': \quad \begin{cases} (\bar{2}) \\ (\bar{2}) \rightsquigarrow (\bar{2})(\bar{3}) \\ (\bar{3}) \rightsquigarrow (\bar{2})(\bar{3})(\bar{3}). \end{cases}$$

For each label (h) of Ω and $(\bar{k})$ of Ω', $(h \cdot k)$ is a label of the rule $\Omega \odot \Omega'$, and it is colored only if there is already another label having the same value. The axiom is $(a \cdot b)$, where (a) and (b) are the axioms of the rules. If the productions of (h) and $(\bar{k})$ are:

$$(h) \rightsquigarrow (c_1) \dots (c_h)$$

$$(\bar{k}) \rightsquigarrow \overline{(e_1)} \dots \overline{(e_k)},$$

then the production of $(h \cdot k)$ is:

$$(h \cdot k) \rightsquigarrow (c_1 \cdot e_1) \dots (c_1 \cdot e_k) \dots (c_h \cdot e_1) \dots (c_h \cdot e_k).$$

Referring to our example, the labels of $\Omega \odot \Omega'$ are $(2 \cdot \bar{2}) = (4)$, $(2 \cdot \bar{3}) = (6)$, $(3 \cdot \bar{2}) = (\bar{6})$, $(3 \cdot \bar{3}) = (9)$. For instance, the production for the label (4) is:

$$(4) = (2 \cdot \bar{2}) \rightsquigarrow (2 \cdot \bar{2})(2 \cdot \bar{3})(3 \cdot \bar{2})(3 \cdot \bar{3}) = (4)(6)(\bar{6})(9).$$

In the same way we obtain:

$$\Omega \odot \Omega' : \qquad \begin{cases} (4) \\ (4) \rightsquigarrow (4)(6)(\bar{6})(9) \\ (6) \rightsquigarrow (4)(6)(6)(\bar{6})(9)(9) \\ (\bar{6}) \rightsquigarrow (4)(4)(6)(6)(\bar{6})(9) \\ (9) \rightsquigarrow (4)(4)(6)(6)(6)(6)(\bar{6})(9)(9). \end{cases}$$

The rule $\Omega \odot \Omega'$ has ij labels, i and j being the number of labels of Ω and Ω' respectively.

- **Equivalence.** Is there a criterion whereby we can establish whether two given succession rules are equivalent simply by working on their labels, that is, with no need to determine the corresponding generating functions? Furthermore, given a succession rule, is there a method to obtain some equivalent rules?

- **Inversion.** Let $\{f_n\}_n$ be a non-decreasing sequence of positive integers. Is there a method allowing us to decide whether a succession rule defining the sequence $\{f_n\}_n$ exists and, if it does, to find it? Note that this problem can be solved for finite rules.

- **Colored rules.** Let $\{f_n\}_n$ be a non-decreasing sequence of positive integers defined by a colored succession rule Ω. Is there a criterion to establish whether a non-colored succession rule defining $\{f_n\}_n$ exists? This problem is still open also for finite rules. Regarding the matter, the following facts should be mentioned:

  1. if the sequence $\{f_n\}_n$ has repetitions, that is there exists j such that $f_j = f_{j+1}$, then it is easy to check whether the rule for $\{f_n\}_n$ needs to be colored;
  2. therefore, we can focus exclusively on the case of a strictly increasing $\{f_n\}_n$. The only thing that can be surely stated is that if the sequence $\{f_{n+1} - f_n\}$ is strictly increasing too, then a non-colored succession rule defining $\{f_n\}_n$ must exist, although sometimes it may have a very complicated form:

$$\begin{cases} (f_1) \\ (1) \rightsquigarrow (1) \\ (f_k) \rightsquigarrow (1)^{k-1}(f_{k+1} - f_k + 1). \end{cases}$$

5.1 A Conjecture

Conjecture: *if a succession rule has a rational generating function, then it is equivalent to a finite succession rule.* It is sufficient to prove that each rational generating function of a succession rule satisfies the same properties shared by the generating functions of finite rules, as described in Section 1. If the conjecture proves true, rational functions such as (5) cannot be the generating functions of any succession rule. For example, let Ω be the rule, studied in [1], whose set of labels is the whole set of prime numbers:

$$\Omega: \qquad \begin{cases} (2) \\ (p_n) \rightsquigarrow (p_{n+1})(q_n)(r_n)(2)^{p_n-3}, \end{cases}$$

where p_n denotes the nth prime number, and q_n and r_n are two primes such that $2p_n - p_{n+1} + 3 = q_n + r_n$ (via Goldbach conjecture). According to our conjecture, as its generating function is rational, $f(x) = \frac{1-2x}{1-4x+3x^2}$, it is possible to find a finite succession rule Ω' equivalent to Ω:

$$\Omega': \qquad \begin{cases} (2) \\ (2) \rightsquigarrow (2)(3) \\ (3) \rightsquigarrow (2)(3)(4) \\ (4) \rightsquigarrow (2)(3)(4)(4). \end{cases}$$

It should be noticed that the rule Ω' was further exploited in [9], being the 4-approximating rule for Catalan numbers. Furthermore, such a rule describes a recursive construction for Dyck paths whose maximal ordinate is 4.

References

[1] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy and D. Gouyou-Beauchamps, On generating functions of generating trees, *Proceedings of 11th FPSAC, Barcelona* (1999) 40-52.

[2] E. Barcucci, A. Del Lungo, A. Frosini, S. Rinaldi, From rational functions to regular languages, in *Formal Power Series and Algebraic Combinatorics,* D. Krob, A. A. Mikhalev, A. V. Mikhalev Eds., Springer-Verlag, Berlin (2000) 633-644.

[3] E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, ECO: a methodology for the Enumeration of Combinatorial Objects, *Journal of Difference Equations and Applications,* Vol.5 (1999) 435-490.

[4] E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, Some combinatorial interpretations of q-analogs of Schröder numbers, *Annals of Combinatorics,* 3 (1999) 173-192.

[5] E. Barcucci, E. Pergola, R. Pinzani, S. Rinaldi, Eco method and hill-free paths, *Seminaire Lotharingien de Combinatoire,* B46b (2001), 14pp.

[6] F. R. K. Chung, R. L. Graham, V. E. Hoggatt, M. Kleimann, The number of Baxter permutations, *J. Combin. Theory Ser. A,* 24 (1978) 382-394.

[7] S. Corteel, Problèmes énumératifs issus de l'Informatique, de la Physique et de la Combinatorie, *PhD Thesis, Université de Paris-Sud,* 6082 (2000).

[8] O. Guibert, Combinatoire des permutations à motifs exclus en liason avec mots, cartes planaries et tableaux de Young, *Thèse de l'Université de Bordeaux I,* (1996).

[9] E. Pergola, R. Pinzani, S. Rinaldi, ECO-Approximation of algebraic functions, in *Formal Power Series and Algebraic Combinatorics,* D. Krob, A. A. Mikhalev, A. V. Mikhalev Eds., Springer-Verlag, Berlin (2000) 719-730.

[10] G. Rozenberg, A. Salomaa, *The mathematical theory of L systems*, Academic Press, London (1980).

[11] A. Salomaa and M. Soittola, *Automata-theoretic aspects of formal power series*, Springer-Verlag, New York (1978).

[12] N. J. A. Sloane and S. Plouffe, *The encyclopedia of integer sequences* Academic press, New York (1996).

[13] J. West, Generating trees and the Catalan and Schröder numbers, *Discrete Mathematics,* 146 (1995) 247-262.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Solution to the SIAM "Hundred-dollar, Hundred-digit Challenge"

Michel Kern

N° 4472

Mai 2002

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4472--FR+ENG

INRIA
ROCQUENCOURT

Solution to the SIAM "Hundred-dollar, Hundred-digit Challenge"

Michel Kern

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet estime

Rapport de recherche n° 4472 — Mai 2002 — 18 pages

Abstract: In February 2002, L. N. Trefethen proposed a lis of 10 short problems, each with a numerical answer. The challenge was to compute each of the numbers to 10 digits accuracy. This report gives a solution to each of the 10 problems. The problems range to the computation of an integral, to optimizing oscillatory functions, through partial differnetial equations and probability theory. We detail the methods used in each case, and comment on how we obtained the requested accuracy.

Key-words: Numerical computation, Maple, numerical quadrature, linear algebra, global optimization, random walk, heat equation, Fourier series.

This is a slightly edited version of the report I submitted for the SIAM Challenge, that earned me Second Prize with 99 correct digits out of 100. I have corrected a simple mistake that made me report a wrong digit in problem 2. In this report, all given digits are correct. I have also provided a short introduction to the problems, as well as the statement of the problem themselves, in an appendix.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Une solution au Concours SIAM «100 dollars, 100 chiffres»

Résumé : En février 2002, L. N. Trefethen a soumis une liste de 10 problèmes à la communauté. Chaque problème a une solution numérique, que l'on doit calculer avec 10 chiffres significatifs. Ce rapport présente une solution à chacun de ces problèmes. Les sujets vont du calcul d'intégrale à la minimisation de fonctions oscillantes, en passant par les équations aux dérivées partielles et les probabilités. Dans chaque cas nous détaillons la méthode utilisée, et comment nous avons pu obtenir la précision désirée.

Mots-clés : Calcul numérique, Maple, algèbre linéaire, quadrature numérique, optimisation globale, marche aléatoire, équation de la chaleur, série de Fourier.

Introduction

In February 2002, L. N. Trefethen proposed a set of 10 problems to the applied mathematics community. Each problem is stated in a few sentences, and each has a solution that is a single number. The challenge was to compute the 10 "magic numbers" to 10 digits accuracy. The problems covered over a wide range of different areas: one needs to compute (oscillatory, and singular) integrals, sum series, compute the minimum of a wildly oscillatory function, invert a sparse matrix, and solve problems from random walk and Brownian motion.

For completeness, the problems are recalled in Appendix A.

1 Summary of the results

Problem	Solution
1	0.3233674317
2	0.9952629194
3	$0.1274224153\ 10^{1}$
4	$-0.3306868648\ 10^{1}$
5	0.2143352346
6	$0.6191395447\ 10^{-1}$
7	0.7250783463
8	0.4240113870
9	0.7859336744
10	$0.3837587979\ 10^{-6}$

Table 1: Summary of the 10 magic numbers

2 General remarks

As the hint advises: these problems are hard! In a few cases, how to compute *some* approximation to the answer is clear, in most of the others, it is not even clear how to get any answer, let alone an accurate one. Also, for someone accustomed to discretizing continuous problems, a source of psychological difficulty is the mere fact of trying to compute solutions to 10 digits accuracy!

For all problems, the main decision was find the right tool for the job. Most of the time, the task was to compute an integral, or a series, or solve an equation to 10 digits accuracy. Programming languages, like Fortran or C, are limited to 15 digits accuracy. On the other hand, modern computer algebra systems, like Maple or Mathematica, provide a fairly complete numerical library, including all the tasks listed above, in arbitrary precision arithmetic. In most cases, this was the deciding factor. Of course in a few cases, Maple was not the right tool for the jobs: two of the problems reduced to linear algebra problems, and Matlab is much better suited to "pure" linear algebraic problems. In one other case, I found a Matlab toolbox, and in the last one a special purpose Fortran code (though Maple finished the job).

Of course Maple is not the final answer to all these problems. In each case, one has to find some specific way of assessing what accuracy to expect. I try to address this point below.

3 The problems

3.1 A singular integral

This problem becomes much simpler when introducing the Lambert W function, defined as the inverse function to $x \to x \exp x$. This function has been extensively studied in [4], and is known to Maple.

The change of variable $x = \exp(-W(z))$ (so that $z = -\dfrac{\ln x}{x}$ and $\dfrac{dx}{x} = W'(z)dz$) transforms the integral to

$$I_\epsilon = \int_0^{1/\epsilon \ln(1/\epsilon)} \cos z \, W'(z) \, dz. \tag{3.1.1}$$

As $1/\epsilon \ln(1/\epsilon)$ goes to ∞ as ϵ goes to zero, the limit is the same as the limit for $A \to \infty$ of

$$I_A = \int_0^{A} \cos z \, W'(z) \, dz. \tag{3.1.2}$$

It is still not obvious that the limit is finite. To see this, we recall that W is asymptotic to $\ln z$ for large z, and so $W^{(k)}$ will be asymptotic to z^{-k}, for $k > 0$. By integrating by parts and differentiating W (although my first thought was to *integrate* W), we can make the integral absolutely convergent. In order to obtain an easily computable integral, we integrate by parts several times: doing it three times seems like an acceptable compromise, giving z^{-4} decay. We obtain :

$$\lim_{\epsilon \to 0} I_\epsilon = -W''(0) + \int_0^{\infty} \sin z \, W^{(4)}(z) \, dz, \tag{3.1.3}$$

with $W''(0) = -2$.

It is "clear" (though I can't prove it) that $W^{(4)}$ is negative and increasing on $\mathbf{R}^+$. By integrating between the zeros of the sine function, the integral can be seen as an alternating series, so that the remainder is bounded by the first neglected term. To keep matters simple, I argue as follows: I need an integer k such that $\int_{2k\pi}^{(2k+1)\pi} \sin z \, |W^{(4)}(z)| \, dz < 10^{-10}$. By the mean value theorem, there is a $\xi_k \in [2k\pi, (2k+1)\pi]$ such that

$$\int_{2k\pi}^{(2k+1)\pi} \sin z \, |W^{(4)}(z)| \, dz = \pi |W^{(4)}(\xi_k)| \leq \pi |W^{(4)}(2k\pi)|. \tag{3.1.4}$$

A little experimentation shows that taking $k = 300$ should give 11 digits accuracy. Indeed, using Maple with 15 digits accuracy, I computed the integral for several values of k, and the difference between the values obtained for $k = 240$ and $k = 300$ is in the 11th digit. Eventually

$$\int_0^{300\pi} \sin z \, W^{(4)}(z) \, dz \approx -1.6766325683 \tag{3.1.5}$$

so the requested value is $.3233674317$, with all digits shown believed to be correct.

3.2 Photon scattering

This problem is simple in principle, but was much harder to solve than I expected. The basic idea is straightforward : after each collision, find which mirror will be hit next, compute the intersection point, find the next direction for the photon.

Each of the three steps is itself simple:

find next mirror: the only way I could think of was a systematic search. The search can be restricted to one quarter of the plane by considering the half-line on which the photon moves, and can be speeded-up by searching along diagonals, or anti-diagonals (depending on the quarter-plane).

compute intersection: This is completely trivial in principle, but most likely quite tricky numerically. One has to solve a second degree equation, and some of them seem to be quite ill-conditioned. I have resorted to using high precision computations in Maple. Details are given below.

Find next direction: Again, this is quite simple: compute the symmetric of the incoming line with respect to the diameter through the intersection point.

The last point is not reached. One has to backtrack along the last segment, by an amount proportional to the amount by which the length exceeds 10.

It is quite clear that a geometric engine will be a very handy tool, if not an essential one, so one does not have to take care of the low-level tasks such as : computing the intersection of a line with a circle, taking the symmetric of a line with respect to another line,... My first attempt was to (mis)use the 2D mesh generator `emc2` [10] by Fréderic Hecht (available at `http://www-rocq.inria.fr/gamma/cdrom/www/emc2/fra.htm`. It includes a geometric engine, able to compute the intersection of a line with a circle, and to draw the symmetric of a line with respect to another line. This is all that is needed for this problem. The main drawback is that this software is written partly in single precision, so I could certainly not get 10 digits accuracy for this problem (this should not be taken as a criticism of the software, after it a mesh generator, where 10 digits accuracy is usually not needed). As I found out later, things are much worse: single precision will give a completely wrong result... For later comparison, I show the results obtained with `emc2` on figure 1.

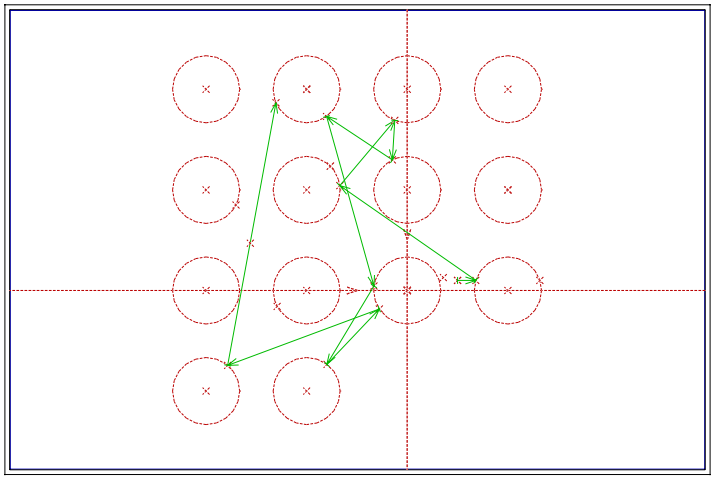


Figure 1: Photon scattering by circular mirrors, results with `emc2`

I then switched to Maple. Maple has a nice `geometry` package, that lets you manipulate directly geometric entities such as points, lines, segments, circles. A line can be defined by two points or by its Cartesian equation, and a circle can be defined by its center and radius, or by its Cartesian equation. Basic geometric operations such as reflection and intersection between objects are supported. And of course, the package benefits from the usual Maple features. In that case the important points will be the ability to use arbitrarily high precision arithmetic.

I wrote a short Maple program solving the problem, and ran it first in 10 digits accuracy (the default). To my surprise, the result was quite different from the one I had gotten with `emc2` (and not just a refinement). I then went to 20 digits, and obtained yet another result. Pictures of the trajectories obtained with Maple are shown in figure 2. They are qualitatively different (look at the last three reflections), and also different from the one obtained with `emc2` (compare figure 1 above). One sees

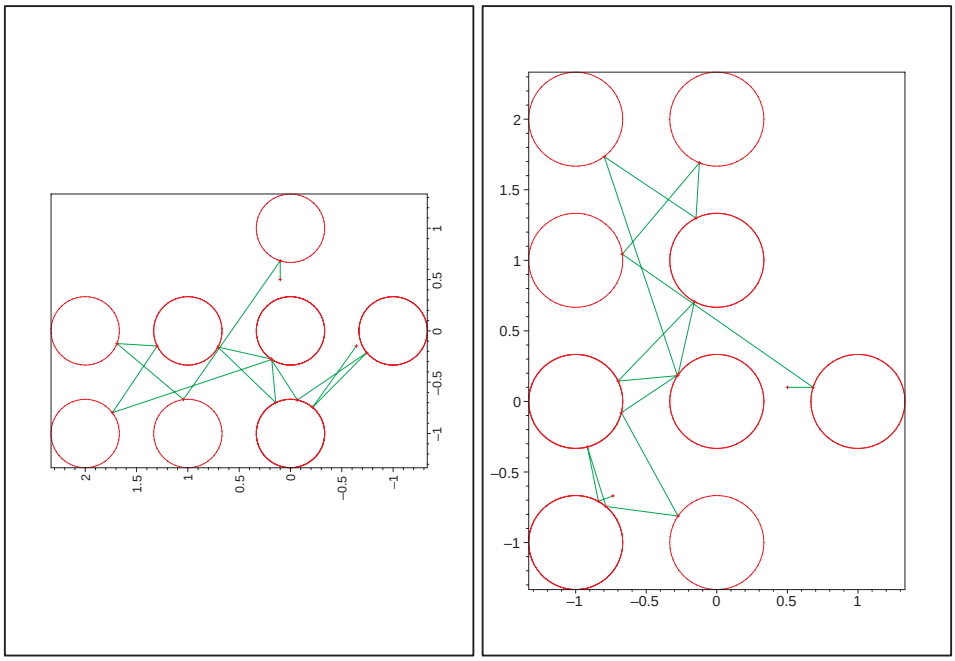


Figure 2: Trajectories obtained using Maple. Left: 10 digits accuracy right: 20 digits accuracy.

that in order to obtain *qualitatively* correct results, at least 20 digits are needed.

Now, I need to obtain (and justify) 10 digits accuracy. To do this, the easiest way seems to simply increase the number of digits used by Maple (though it is known that this is not foolproof). Using 40, then 80 digits (computer time is cheap), confirmed the result with 20 digits. The computation seems to be *very* sensitive to the precision used. I lose between one and two digit at each new collision. This is most likely due to "sensitive dependence on the initial conditions", and is related to the chaotic nature of this "billiard".

Figure 2 shows some of the results obtained with Maple (compare figure 1 above), while table 2 shows the difference between 20 and 40 digits (digits that differ are italicized). The results for 80

		First point	Last point
x	20 digits	-.66949971878499157*67*	.69338200*642544047977*
	40 digits	-.66949971878499157*83*	.69338200*475953252931*
y	20 digits	1.0433667525635879516	-.13075365*053191627015*
	40 digits	1.0433667525635879516	-.13075364*662535335423*

Table 2: Results with varying number of digits. First line: first collision point, second line: last collision point

digits agree with those for 40 digits, to 20 digits. The results given below are taken from a 40 digits computation, truncated to 10 digits.

The time just before the last (extraneous) collision is 11.5278649041657541328 and the time along the last segment is 1.637361901. Eventually, at $t = 10$, the distance of the photon to the origin is $.99526291944335416 09$, and the coordinates of the corresponding point are $-.73629269861$ and $-.6696426964$.

3.3 Norm of infinite matrix

The matrix is bounded on l^2 because it is actually a Hilbert-Schmidt operator, because obviously:

$$\sum_{i,j} a_{ij}^2 = \sum_{n=1}^{\infty} \frac{1}{n^2} < \infty.$$

Thus $\|A\| \leq \pi/\sqrt{6} \approx 1.2825$.

I cannot see a way of simplifying this problem, so I used brute force. That is, I replaced the infinite matrix by a finite section, as large as I could handle, and let Scilab compute the norm Scilab is a free, interactive, scientific software package for numerical computations, with a syntax close to that of Matlab. It is available at `http://www-rocq.inria.fr/scilab/`, and is described in the book [3].

The first task is to find a convenient way to generate the matrix entries. A little experimentation (helped by Sloane's "Online Encyclopedia" [5], at `http://www.research.att.com/~njas/sequences/`) reveals that:

$$a_{ij} = \frac{1}{(i+j-1)(i+j-2)/2+j}. \tag{3.3.1}$$

Next I computed the norm of this matrix for several sizes. Results are show in table 3:

n	norm(A)
500	1.27422411595291
1000	1.27422414812948
1500	1.27422415142163
2000	1.27422415222862
2500	1.27422415251711
3000	1.27422415264495
3500	1.27422415271009
4000	1.27422415274670

Table 3: Norm of truncated matrix

As expected, these values are less then $\pi/\sqrt{6}$. Without further understanding of the limit process as the size of the matrix goes to infinity, it is difficult to assess whether or not convergence has actually taken place.

A plausible value, probably accurate to 6 to 8 digits is 1.274224153.

3.4 Global minimum of a noisy function

I proceed in three steps:

1. Bound the region where the minimum lies;
2. Using a global minimization algorithm, locate an approximation to the minimum, so that the function is convex in a neighborhood of this approximation;
3. Refine this approximation with Newton's method.

For the first part, a little experimentation shows that for $|x| \geq 3,\ |y| \geq 3$, we have

$$J_4(x,y) \geq \frac{18}{4} + e^{-1} - 4 \geq J(0,0) = 1 + \sin(60),$$

so the minimum lies in the rectangle $[-3,3] \times [-3,3]$.

For the global optimization, I used the DIRECT algorithm. Kelley [11] gives it as one that addresses the problem of locating the global minimum of a (possibly noisy) function. He mentions an implementation by Gablonsky [7]. Fortunately, this implementation is available on the Web, at `http://www4.ncsu.edu/eos/users/c/ctkelley/www/optimization_codes.html`.

All one has to do to use the code is write a function (in Fortran) implementing the cost function, provide bounds for the variables, and set a few parameters (tolerance, number of function evaluations). It was not difficult to obtain a value for the minimum of -3.3068686, reached at the point $(-0.0244033, 0.2106123)$. This actually used 73929 function evaluations, and 91 seconds on a Pentium II 366 PC.

Now one can use a local minimization method, after checking that the function is actually convex over the rectangle $-0.04 \leq x \leq 0.01,\ 0.18 \leq y \leq 0.24$. Letting Maple solve for zeros of the gradient of our cost function improves the minimum to -3.30686864747527, reached at the point $x = -.0244030796943752, y = .210612427155356$. The computation was carried out with 15 digits, so x and y should be accurate to 10 digits, and the value of the minimum should be more accurate (see the discussion on this issue in problem 9).

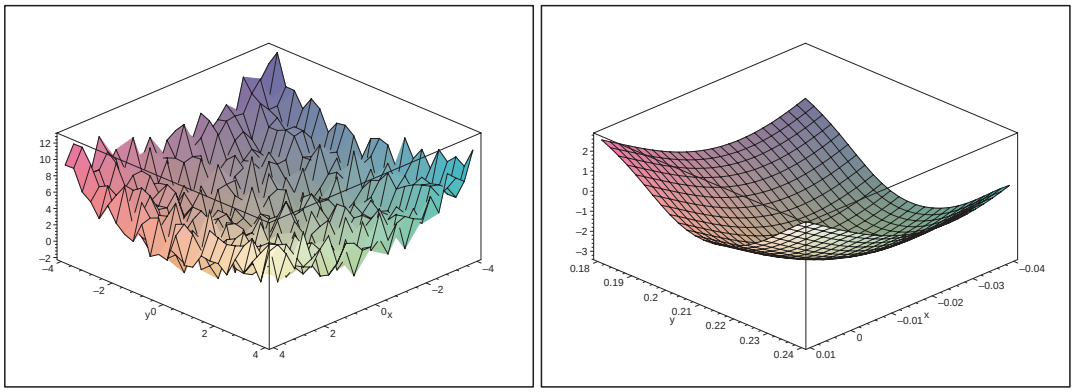


Figure 3: Cost function and a zoom near the global minimum

3.5 Complex best approximation

For this problem, I have had the good luck to find all the software I needed on the web, without necessarily understanding the theory behind it. I first downloaded the `coca` toolbox by B. Fischer and J. Modersitzki, at `http://www.math.mu-luebeck.de/workers/modersitzki/COCA/coca5.html`. This is a Matlab toolbox designed for solving exactly the problem at hand, to wit finding best approximations in the complex plane. It assumes the function to be approximated is analytic in the region of interest (which is the case here), so that the minimum is attained on the boundary.

That left me with the task of computing accurate approximation to the complex Γ function. Here I used a Matlab file from P. Godfrey, at `http://winnie.fit.edu/~gabdo/gamma.m` (described in[8]), based on an approximation due to Lanczos, that claims 15 digits accuracy. This is somewhat difficult to check, as I have no other means of computing the (complex) gamma function.

I show below some of the Matlab commands I used for solving this problem.

```
FUN.name='igamma';
BASIS.name      = 'monom';      % standard basis
BASIS.choice=[0:3];             % cubic polynomial
BASIS.C_dim     = length(BASIS.choice);
BOUNDARY.name   = 'gcircle';    % name of boundary
real_coef       = 1;
PARA.relative_error_bound = 1e-10;
PARA.stepsize             = 500;    % number of steps in COMPNORM.M
PARA.max_iterations       = 100;
[PARA] = coca(PARA);

PARA.error_norm
ans = 0.21433523459984
```

I also show, on figure 4, the error on the boundary of the unit circle.

Accuracy is hard to estimate here mainly because of my lack of understanding of the method used.

3.6 Biased random walk on a lattice

We follow the exposition in Chapter 2 of Barber and Ninham [1].

Let $P^n_{k_1,k_2}$ be the probability that the flea is at point (k_1, k_2) after n steps, let $P_{k_1,k_2}(z) = \sum_{n=0}^{\infty} P^n_{k_1,k_2} z^n$ be its generating function, and let $G(\phi, z)$ be its Fourier transform (with $\phi = (\phi_1, \phi_2)$:

$$G(z\phi, z) = \sum_{k_1,k_2} e^{ik.\phi} P_{k_1,k_2}(z).$$

The probability that the flea returns to the origin at step n is P^n_{00}. According to the recurrence theorem of Feller [6], the probability that the flea ever returns to the origin is given by $1 - 1/u$, where $u = \sum_{n=0}^{\infty} P^n_{0,0} = P_{0,0}(1)$. Inverting the Fourier series, we end up with the expression

$$u = \frac{1}{4\pi^2} \int_{[-\pi,\pi]^2} G(\phi, 1)\, d\phi.$$

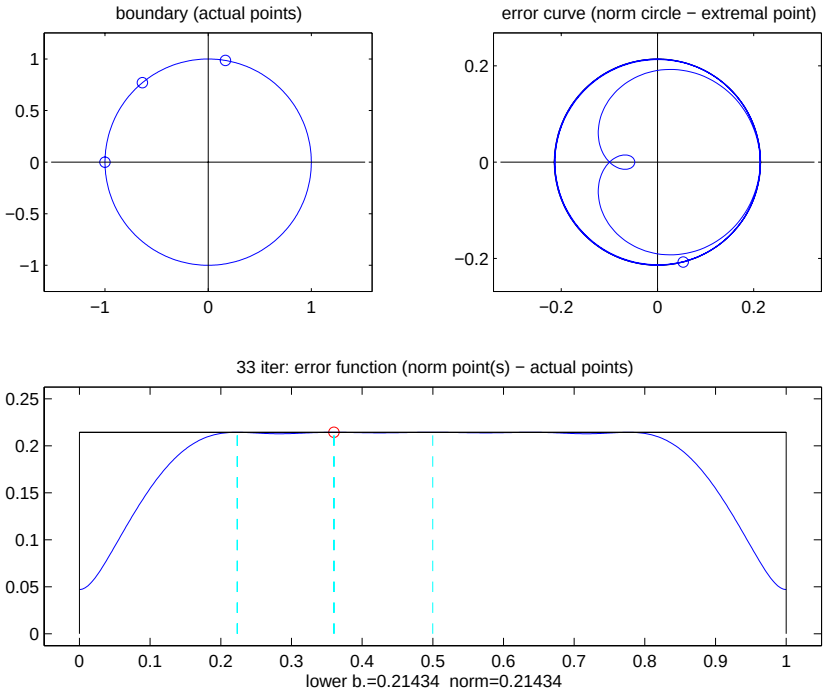


Figure 4: Error for the complex approximation problem

In order to compute G, we start by using the definition of the walk to obtain a recurrence relation among the probabilities $P^n_{k_1 k_2}$:

$$P^{n+1}_{k_1,k_2} = \frac{1}{4}P^n_{k_1,k_2-1} + \frac{1}{4}P^n_{k_1,k_2+1} + (\frac{1}{4}+\epsilon)P^n_{k_1+1,k_2} + (\frac{1}{4}-\epsilon)P^n_{k_1-1,k_2}, \tag{3.6.1}$$

then among the generating functions :

$$P_{k_1,k_2}(z) - z\left(\frac{1}{4}P_{k_1,k_2-1}(z) + \frac{1}{4}P_{k_1,k_2+1}(z) + (\frac{1}{4}+\epsilon)P_{k_1+1,k_2}(z) + (\frac{1}{4}-\epsilon)P_{k_1-1,k_2}(z)\right) = \delta_{0,0}, \tag{3.6.2}$$

using the fact that the flea starts at $(0,0)$.

Now multiply the last equation by $e^{ik.\phi}$ and sum over all lattice points, giving the following expression for $G(\phi, z)$

$$G(\phi,z) = \frac{1}{1 - z\left(1/4e^{i\phi_2} + 1/4e^{-i\phi_2} + (1/4+\epsilon)e^{-i\phi_1} + (1/4-\epsilon)e^{i\phi_1}\right)} \tag{3.6.3}$$

Eventually, we need to solve the equation

$$\frac{1}{4\pi^2}\int_{[-\pi,\pi]^2} \frac{1}{1-(1/2\cos\phi_2 + (1/4+\epsilon)e^{-i\phi_1} + (1/4-\epsilon)e^{i\phi_1})}\, d\phi_1\, d\phi_2 = 2$$

The ϕ_1 integral can be computed by the residue theorem, letting $z = e^{j\phi_1}$. Actually, Maple can perform the whole computation (though I do not know how to prove that the first root is the one inside the unit circle).

```
assume(epsilon>0,epsilon<1/4, phi_1,real);
g:=1/(1-1/2*cos(phi_2)-(1/4-epsilon)*exp(I*phi_2)
                      -(1/4+epsilon)*exp(-I*phi_2)):
g:=normal(1/(I*z)*subs({exp(I*phi_2)=z, exp(-I*phi_2)=1/z},g)):
racs:={solve(denom(g),z)}:
inint:=2*I*Pi*residue(g, z=racs[1]):
```

This gives

$$u = 1/\pi \int_{-\pi}^{\pi} \frac{d\phi_1}{\sqrt{3 - 4\cos\phi_1 + \cos^2\phi_1 + 16\epsilon^2}}$$

It turns out that this integral is a combination of elliptic integrals, that Maple can again evaluate exactly. I cannot resist showing the exact value :

$$\begin{aligned} u = 8/\pi \frac{1}{\sqrt{2 + 2\sqrt{1 - 16\,\epsilon^2} + 16\,\epsilon^2}} \Bigg[& 2K\left(2\sqrt{\frac{\sqrt{1 - 16\,\epsilon^2}}{\left(3 - \sqrt{1 - 16\,\epsilon^2}\right)\left(1 + \sqrt{1 - 16\,\epsilon^2}\right)}} \right) \\ & + F\left(\frac{\sqrt{2}}{2}\sqrt{\frac{3 - \sqrt{1 - 16\,\epsilon^2}}{2 - \sqrt{1 - 16\,\epsilon^2}}}, 2\sqrt{\frac{\sqrt{1 - 16\,\epsilon^2}}{\left(3 - \sqrt{1 - 16\,\epsilon^2}\right)\left(1 + \sqrt{1 - 16\,\epsilon^2}\right)}} \right) \\ & + F\left(\frac{\sqrt{2}}{2}\sqrt{\frac{1 + \sqrt{1 - 16\,\epsilon^2}}{2 + \sqrt{1 - 16\,\epsilon^2}}}, 2\sqrt{\frac{\sqrt{1 - 16\,\epsilon^2}}{\left(3 - \sqrt{1 - 16\,\epsilon^2}\right)\left(1 + \sqrt{1 - 16\,\epsilon^2}\right)}} \right) \Bigg] \end{aligned} \quad (3.6.4)$$

where K is the complete elliptic integral of the first kind, and F is the incomplete elliptic integral of the first kind.

Figure 5 shows the variation of u as a function of ϵ. As expected, u goes (slowly) to infinity when ϵ goes to 0 (this is the case of a symmetric walk, and the return probability is 1). The fact that the limit for $\epsilon = 1/4$ is non-zero is somewhat surprising... Maple solved the equation $u(\epsilon) = 2$ (it is clear that this function is monotone decreasing, so that there is only one solution), giving the solution 0.061913954473991. All digits are believed to be exact (based on a 40 digits computation).

3.7 Matrix inversion

I do not see any other way than numerically solving the system

$$Ax = e_1$$

and taking the first component of x. The matrix A is 20000×20000 with a bandwidth of 16384 and only 31 nonzero elements per line. It is very sparse, but the large bandwidth makes it impossible to use a sparse direct solver. I tried using UMFPACK, and even though reordering using reverse Cuthill-McKee gives the best results, factorization fails for lack of space.

I have resorted to using the conjugate gradient method, with incomplete Cholesky preconditioning, even though I do not know whether or not the matrix is positive definite (I actually do not think it is, but see below).

Here are the Matlab commands I used (the file `primes` contains a list of the first 20000 primes generated by Maple):

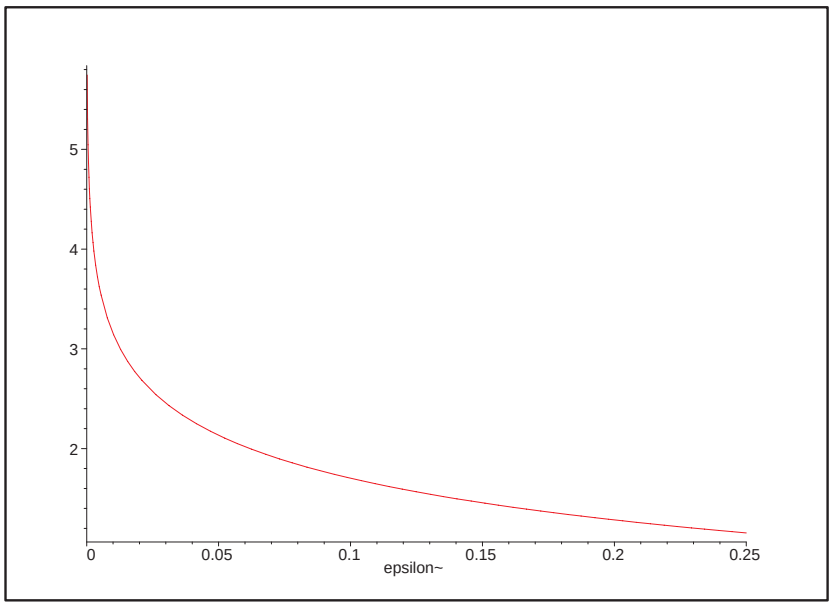


Figure 5: u as a function of ϵ, random walk problem

```
fid=fopen('primes'); primes=fscanf(fid, '%d,');
diags=2.^(0:14); diags=[-diags(15:-1:1),0, diags];
B=ones(20000,31); B(:,16)=primes;
A=spdiags(B, diags, 20000, 20000);
b=zeros(20000,1); b(1)=1;
R=cholinc(A,'0');

tol=1e-10; maxit=100;
[x,flag,relres,iter,resvec] = pcg(A,b,tol,maxit,R',R);

x(1)
ans =
  0.72507834626840
```

The conjugate gradient after 7 iterations, so the matrix must be "positive definite enough". It is difficult to estimate the accuracy of this solution, as I can see no way to estimate the condition number of this matrix. A good sign is the quick convergence of the conjugate gradient, and the fact that x_1 seems to be the largest component of x.

3.8 Heat equation

For definiteness, I will take the side $\{x = L\}$ $(L = 1)$ as the one with the temperature maintained at $d = 5$. I also denote the square by Ω, the side $\{x = L\}$ by Γ_d and the 3 other sides by Γ_0.

This problem cries out for a Fourier series solution. However, since the computations are somewhat involved, I started by using the PDE toolbox in Matlab to get a crude approximation to the answer. This way, I found that the time for the temperature at the center of the plate to reach 1 is between 0.42 and 0.43. This will provide a useful check on the semi-analytical solution.

The first step is to get rid of the inhomogeneous Dirichlet boundary condition on Γ_d. This can be done by a harmonic lifting : I solve for the function $u_0(x,y)$ solution of :

$$\begin{cases} -\Delta u_0 = 0 & \text{in } \Omega \\ u_0 = d & \text{on } \Gamma_d \\ u_0 = 0 & \text{on } \Gamma_0. \end{cases} \tag{3.8.1}$$

This can also be done by Fourier series: take $u_0(x,y) = \sum_{n=1}^{\infty} u_n(x) \sin n\pi \dfrac{(y+L)}{2L}$. One easily finds

$$u_n(x) = \mu_n \frac{\sinh n\pi \dfrac{(x+L)}{2L}}{\sinh n\pi}, \qquad \text{with } \mu_n = \begin{cases} \dfrac{4d}{n\pi} & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even.} \end{cases}$$

For later use, note that

$$u_0(0,0) = \sum_{n=0}^{\infty} (-1)^n \frac{4d}{(2n+1)\pi} \frac{\sinh(2n+1)\pi/2}{\sinh(2n+1)\pi}. \tag{3.8.2}$$

Now, let $u = v + u_0$. Then v solves the problem:

$$\begin{cases} \dfrac{\partial v}{\partial t} - \Delta v = 0 & \text{in } \Omega \\ v = 0 & \text{on } \partial\Omega \\ v(x,y,0) = -u_0(x,y) & \text{in } \Omega. \end{cases} \tag{3.8.3}$$

We search for v in the form $v(x,y,t) = \sum_{p,q} v_{pq}(t) \sin p\pi \dfrac{x+L}{2L} \sin q\pi \dfrac{y+L}{2L}$. v_{pq} satisfies the ODE :

$$v'_{pq}(t) + (p^2+q^2)\pi^2/(4L^2) v_{pq}(t) = 0$$

with initial condition $v_{pq}(0) = -b_{pq}$, where b_{pq} is the Fourier coefficient of u_0 :

$$b_{pq} = \frac{2}{\pi}\mu_q \int_0^{\pi} \frac{\sinh q\pi t}{\sinh q\pi} \sin pt \; dt = -\frac{2}{\pi}\mu_q \frac{(-1)^p p}{p^2+q^2}.$$

Eventually, we have

$$v(x,y,t) = \sum_{p,q} \frac{2}{\pi}\mu_q \frac{(-1)^p p}{p^2+q^2} e^{-(p^2+q^2)\pi^2/(4L^2)t}.$$

Putting everything together, we get the expression for the temperature at the center of the plate, as a function of time :

$$u(0,0,t) = \frac{20}{\pi} \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \frac{\sinh(2n+1)\pi/2}{\sinh(2n+1)\pi} - \frac{40}{\pi^2} \sum_{q=0}^{\infty} \sum_{p=0}^{\infty} \frac{(-1)^{p+q} (2p+1) e^{-1/4 \left((2p+1)^2+(2q+1)^2\right)\pi^2 t}}{(2q+1)\left((2p+1)^2+(2q+1)^2\right)} \tag{3.8.4}$$

I show, on figure 6, the evolution of the temperature at the center of the plate.

Except for small values of t, the series above is rapidly convergent. It is easy to see that the solution is indeed between 0.42 and 0.43. Then, I used Maple to solve the equation $u(0,0,t) = 1$ in this interval. The solution is $t = 0.424011387033688$, truncated from a 20 digits approximation. This should give at least ten correct digits.

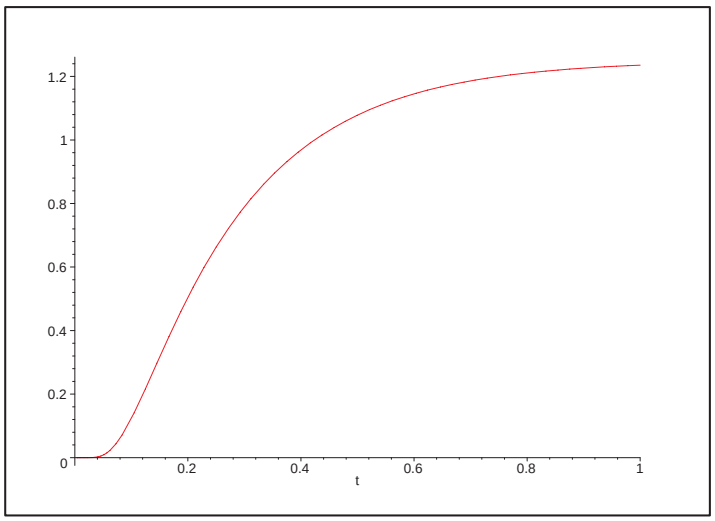


Figure 6: Evolution of the temperature at the center of the plate

3.9 A parametric integral

As $\alpha > 0$, the given integral exists. The given function is highly oscillatory, so I must again proceed in several stages.

I start with a visual examination of the function. Maple can fairly easily compute single precision accuracy approximations to the integral (more on that issue below), and I obtained the graph shown on the left part of figure 7. It is easier to compute the integral if one first takes the factor $2 + \sin(10\alpha)$ out of the integral. Maple seems to be able to handle the singularity at $x = 2$ without my needing to transform to a unbounded interval. It is visually apparent that the maximum is somewhat less than one. The right part of figure 7 shows a zoom of the graph on the interval $[1/2, 1]$, and it seems that the maximum is just less than 0.8. More importantly, on this interval the given function is unimodal, and thus amenable to a local maximization algorithm.

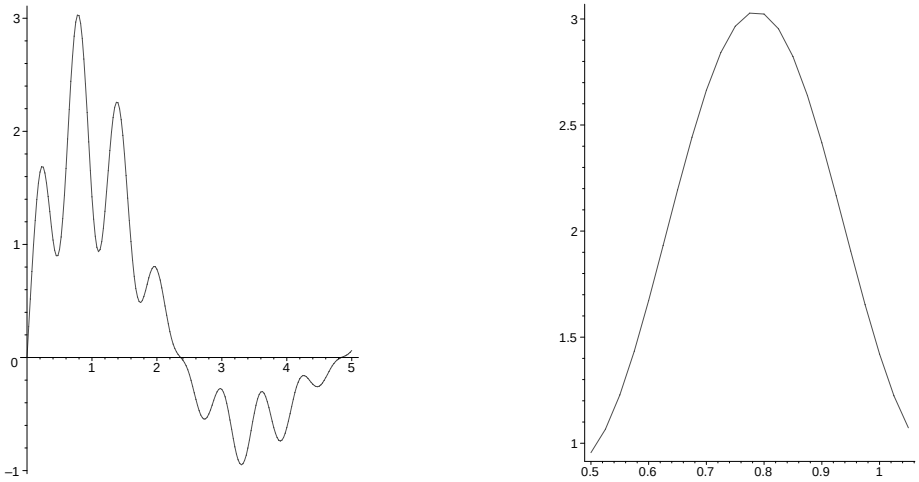


Figure 7: Graph of given integral as a function of α, left: on $[0, 5]$, right: zoom between 0.5 and 1

As it is more difficult (not to say more expensive) to compute the derivative of this function, I chose to use a minimization method that does not require derivatives. The most famous one is probably Brent's `localmin`, described in the book [2]. It is based on inverse parabolic interpolation, safeguarded by a golden search section. I have implemented the code as described in the Algol routine given on page 79 of [2].

Before giving the result, it is worth commenting on the attainable accuracy with this (or any other) method. The detailed discussion in [2] shows that we cannot hope to resolve the location of the minimum of a function f computed in finite precision to more than $2|f_0|\epsilon/(x_0^2 f_0'')^{1/2}$, where ϵ is the machine precision, x_0 is the location of the minimum, $f_0 = f(x_0)$, and $f_0'' = f''(x_0)$. If we want the minimum to 10 digits, we need to compute the function to (at least) 20 digits (a crude estimate of the second derivative is -106). This forces the use of Maple, which (in principle) can compute an integral to any given accuracy.

In practice, I found that Maple 6 can not handle the very stringent accuracy required, and that fortunately Maple 7 can. I used Brent's method on the interval $[0.78, 0.79]$ (I have checked by hand that the maximum is between these two values), and also on smaller intervals obtained by lower accuracy searches. The solution obtained (after 25 iterations, with a tolerance of 10^{-12}) is $\alpha_0 = .7859336744$, the last digit being not quite stable.

3.10 Brownian motion

For definiteness, let $L = 10$ be the length of the sides, and $l = 1$ be the length of the ends (I found it convenient to let the ration l/L vary, so as to check the computation). We denote the rectangle by R.

According to [9, thm 13.7 (5)], the probability that the particle will exit through one of the ends rather than one of the sides is given by the value at the center of the rectangle of the solution to the partial differential equation :

$$\begin{cases} -\Delta p = 0 & \text{in } R \\ p = 1 & \text{on } \{x = 0\} \cup \{x = L\} \\ p = 0 & \text{on } \{y = 0\} \cup \{y = l\} \end{cases} \tag{3.10.1}$$

This problem can easily be solved by Fourier series, as in question 3.8.
Let $p(x, y) = \sum_{n=0}^{\infty} p_n(x) \sin\left(\frac{n\pi y}{l}\right)$, then p_n satisfies the ordinary differential equation

$$p_n''(x) - \frac{n^2\pi^2}{l^2} p_n(x) = 0,$$

so p_n has the form

$$p_n(x) = A_n \sinh\left(\frac{n\pi x}{l}\right) + B_n \cosh\left(\frac{n\pi x}{l}\right),$$

and we can determine A_n and B_n by using the boundary conditions on the ends:

- On the left end $x = 0$: $\sum_{n=0}^{\infty} B_n \sin\left(\frac{n\pi y}{l}\right) = 1$,
- On the right end $x = L$: $\sum_{n=0}^{\infty} \left(A_n \sinh\left(\frac{n\pi L}{l}\right) + B_n \cosh\left(\frac{n\pi L}{l}\right)\right) \sin\left(\frac{n\pi y}{l}\right) = 1$.

By expanding the right hand side (a constant) in a series of sines, we find the coefficients A_n and B_n :

- If n is even, $A_n = B_n = 0$,

- if n is odd, $A_n = 4/n\pi \left(\dfrac{1 - \cosh n\pi L/l}{\sinh n\pi L/l} \right), \quad B_n = 4/n\pi.$

Eventually, the probability we seek is given by the sum of the series :

$$p(0,0) = \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^n}{2n+1} \frac{1}{\cosh(2n+1)\dfrac{\pi L}{2l}}. \tag{3.10.2}$$

I have plotted the value of $p(0,0)$ as a function of the ratio $r = L/l$ in figure 8. As expected, the probability is 1 if r goes to zero, and goes to 0 if r goes to infinity.

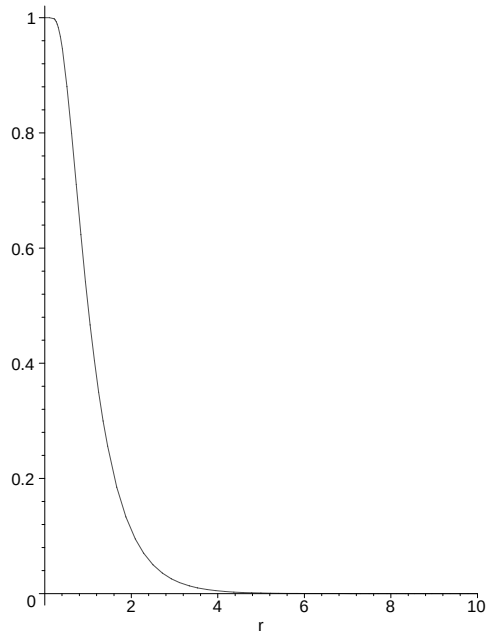


Figure 8: Probability of exit along the ends as a function of aspect ratio

For $r = 10$, as required, Maple gives the value of the sum as $0.3837587979\,10^{-6}$, with all digits thought to be correct.

Acknowledegements

I wish to thanks all the authors of the packages I have used. This work would not have been possible without them. I have also benefitted from conversations with P. Joly, V. Martin, B. Salvy and S. Tordeux.

A Statement of the problems

from *SIAM News*, Volume 35, Number 1

A Hundred-dollar, Hundred-digit Challenge

Each October, a few new graduate students arrive in Oxford to begin research for a doctorate in numerical analysis. In their first term, working in pairs, they take an informal course called the "Problem Solving Squad." Each week for six weeks, I give them a problem, stated in a sentence or two, whose answer is a single real number. Their mission is to compute that number to as many digits of precision as they can.

Ten of these problems appear below. I would like to offer them as a challenge to the SIAM community. Can you solve them?

I will give $100 to the individual or team that delivers to me the most accurate set of numerical answers to these problems before May 20, 2002. With your solutions, send in a few sentences or programs or plots so I can tell how you got them. Scoring will be simple: You get a point for each correct digit, up to ten for each problem, so the maximum score is 100 points.

Fine print? You are free to get ideas and advice from friends and literature far and wide, but any team that enters the contest should have no more than half a dozen core members. Contestants must assure me that they have received no help from students at Oxford or anyone else who has already seen these problems.

Hint: They're hard! If anyone gets 50 digits in total, I will be impressed. The ten magic numbers will be published in the July/August issue of *SIAM News*, together with the names of winners and strong runners-up.—*Nick Trefethen, Oxford University.*

The Hundred-dollar, Hundred-digit Challenge Problems

1. What is $\lim_{\varepsilon \to 0} \int_{\varepsilon}^{1} x^{-1} \cos(x^{-1} \log x)\, dx$?

2. A photon moving at speed 1 in the x–y plane starts at $t = 0$ at $(x,y) = (0.5, 0.1)$ heading due east. Around every integer lattice point (i,j) in the plane, a circular mirror of radius 1/3 has been erected. How far from the origin is the photon at $t = 10$?

3. The infinite matrix A with entries $a_{11} = 1$, $a_{12} = 1/2$, $a_{21} = 1/3$, $a_{13} = 1/4$, $a_{22} = 1/5$, $a_{31} = 1/6$, etc., is a bounded operator on ℓ^2. What is $\|A\|$?

4. What is the global minimum of the function

$$\exp(\sin(50x)) + \sin(60e^{y}) + \sin(70\sin(x)) + \sin(\sin(80y)) - \sin(10(x+y)) + \tfrac{1}{4}(x^2+y^2)\,?$$

5. Let $f(z) = 1/\Gamma(z)$, where $\Gamma(z)$ is the gamma function, and let $p(z)$ be the cubic polynomial that best approximates $f(z)$ on the unit disk in the supremum norm $\|\cdot\|_{\infty}$. What is $\|f - p\|_{\infty}$?

6. A flea starts at (0, 0) on the infinite 2D integer lattice and executes a biased random walk: At each step it hops north or south with probability 1/4, east with probability 1/4 + ε, and west with probability 1/4 − ε. The probability that the flea returns to (0, 0) sometime during its wanderings is 1/2. What is ε?

7. Let A be the 20,000 × 20,000 matrix whose entries are zero everywhere except for the primes 2, 3, 5, 7, . . . , 224737 along the main diagonal and the number 1 in all the positions a_{ij} with $|i-j| = 1,2,4,8,\ldots,16384$. What is the (1, 1) entry of A^{-1}?

8. A square plate [–1, 1] × [–1, 1] is at temperature $u = 0$. At time $t = 0$ the temperature is increased to $u = 5$ along one of the four sides while being held at $u = 0$ along the other three sides, and heat then flows into the plate according to $u_t = \Delta u$. When does the temperature reach $u = 1$ at the center of the plate?

9. The integral $I(\alpha) = \int_0^2 [2 + \sin(10\alpha)] x^{\alpha} \sin(\alpha/(2-x))\, dx$ depends on the parameter α. What is the value $\alpha \in [0, 5]$ at which $I(\alpha)$ achieves its maximum?

10. A particle at the center of a 10 × 1 rectangle undergoes Brownian motion (i.e., 2D random walk with infinitesimal step lengths) till it hits the boundary. What is the probability that it hits at one of the ends rather than at one of the sides?

Solutions should be sent to Nick Trefethen at Oxford University (LNT@comlab.ox.ac.uk), no later than May 20, 2002.

References

[1] Michael N. Barber and B. W. Ninham. *Random and Restricted Walks – Theory and Applications*. Gordon and Breach, 1970.

[2] Richard P. Brent. *Algorithms for Minimization Without Derivatives*. Prentice Hall Series in Automatic Computation. Prentice Hall, 1973.

[3] Carey Bunks, Jean-Philippe Chancelier, François Delebecque, and Claude Gomez, editors. *Engineering and Scientific Computing with Scilab*. Birkhauser, 1999.

[4] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, and D.E. Knuth. On the Lambert W function. *Adv. Comp. Math.*, 1996.

[5] N. J. A. Sloane (Editor). The On-Line Encyclopedia of Integer Sequences. published electronically at `http://www.research.att.com/~njas/sequences/`, 2002.

[6] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley and Sons, 2nd edition, 1957.

[7] J. M. Gablonsky. An implementation of the DIRECT algorithm. Technical Report CRSC-TR98-29, Center for Research in Scientific Computation, North Carolina State University, August 1998.

[8] Paul Godfrey. A note on the computation of the convergent Lanczos complex gamma approximation. published electronically at `http://winnie.fit.edu/~gabdo/gamma.txt`, 2001.

[9] G. R. Grimmet and D. R. Stirzaker. *Probability and Random Processes*. Oxford Science Publications, 2nd edition, 1992.

[10] Fréderic Hecht and Eric Saltel. Emc2 un logiciel d'edition de maillages et de contours bidimensionnels. Technical Report RT-118, INRIA, 1990.

[11] C. T. Kelley. *Iterative methods for optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On the Analysis of Linear Probing Hashing

Philippe Flajolet, Patricio Poblete, Alfredo Viola

N 3265
Septembre 1997

THÈME 2

ISSN 0249-6399

Thème 2 — Génie logiciel et calcul symbolique
Projet Algorithmes

Rapport de recherche— Septembre 1997 — 24 pages

On the Analysis of Linear Probing Hashing

Philippe Flajolet, Patricio Poblete, Alfredo Viola

Abstract: This paper presents moment analyses and characterizations of limit distributions for the construction cost of hash table under the linear probing strategy. Two models are considered, that of full tables and that of sparse tables with a filling ratio strictly smaller than 1. For full tables, the construction cost has expectation $O(n^{3/2})$, the standard deviation is of the same order, and a limit law of the Airy type holds. (The Airy distribution is a semi-classical distribution that is defined in terms of the usual Airy functions or equivalently in terms of Bessel functions of indices $-\frac{1}{3}, \frac{2}{3}$.) For sparse tables, the construction cost has expectation $O(n)$, standard deviation $O(\sqrt{n})$, and a limit law of the Gaussian type. Combinatorial relations with other problems leading to Airy phenomena (like graph connectivity, tree inversions, tree path length, or area under excursions) are also briefly discussed.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33) 01 39 63 55 11 – Télécopie : (33) 01 39 63 53 30

Sur l'analyse du hachage avec essais linéaires

Résumé : Cet article présente analyses de moments et caractérisations de lois limites pour le coût de construction de tables de hachage selon la stratégie dite d'adressage ouvert et essais linéaires. Deux modèles sont considérés, celui des tables pleines et celui des tables "éparses" dont le taux de remplisaage est strictement inférieur à 1. En ce qui concerne les tables pleines, le coût de construction présente une espérance en $O(n^{3/2})$, l'écart type est du même ordre de grandeur et une loi limite de type Airy prévaut. (La distribution d'Airy est une distribution semi-classique qui se caractérise en termes de fonctions d'Airy, c'est-à-dire en terme de fonctions de Bessel d'indices $-\frac{1}{3}, \frac{2}{3}$.) Quant aux tables éparses, leur coût de construction est $O(n)$ en moyenne, avec un écart type en $O(\sqrt{n})$, et il existe une loi limite de type Gaussien. Les relations avec d'autres problèmes combinatoires conduisant à des lois d'Airy (connectivité dans les graphes, longueur de cheminement dans les arbres, aire sous les excursions) sont aussi brièvement examinées dans cet article.

ON THE ANALYSIS OF LINEAR PROBING HASHING

PHILIPPE FLAJOLET, PATRICIO V. POBLETE, AND ALFREDO VIOLA

Dedicated to Don Knuth on the occasion of the 35th anniversary of his first analysis of an algorithm in 1962–1963.

ABSTRACT. This paper presents moment analyses and characterizations of limit distributions for the construction cost of hash table under the linear probing strategy. Two models are considered, that of full tables and that of sparse tables with a filling ratio strictly smaller than 1. For full tables, the construction cost has expectation $O(n^{3/2})$, the standard deviation is of the same order, and a limit law of the Airy type holds. (The Airy distribution is a semi-classical distribution that is defined in terms of the usual Airy functions or equivalently in terms of Bessel functions of indices $-\frac{1}{3}, \frac{2}{3}$.) For sparse tables, the construction cost has expectation $O(n)$, standard deviation $O(\sqrt{n})$, and a limit law of the Gaussian type. Combinatorial relations with other problems leading to Airy phenomena (like graph connectivity, tree inversions, tree path length, or area under excursions) are also briefly discussed.

INTRODUCTION

Linear probing hashing, defined below, is certainly the simplest "in place" hashing algorithm [11, 20, 35].

> *A table of length m, $T[1\,..\,m]$ is set up, as well as a hash function h that maps keys from some domain to the interval $[1\,..\,m]$ of table addresses. A collection of n elements with $n \leq m$ are entered sequentially into the table according to the following rule: Each element x is placed at the first unoccupied location starting from $h(x)$ in cyclic order, namely the first of $h(x), h(x)+1, \ldots, h(1), h(2), \ldots, h(x)-1$.*

For each element x that gets placed at some location y, the circular distance between y and $h(x)$ (that is, $y - h(x)$ if $h(x) \leq y$, and $m + h(x) - y$ otherwise) is called its *displacement*. Displacement is both a measure of the cost of inserting x and of the cost of searching x in the table. *Total displacement* corresponding to a sequence of hashed values is the sum of the invidual displacements of elements. As it determines the *construction cost* of the table, we use both terms interchangeably.

We analyse here the total displacement $d_{m,n}$ of a table of length m (the number of table locations) and size n (the number of keys), under the assumption that all m^n hash sequences are equally likely. The problem has an equivalent formulation in terms of the classical *parking problem*, where the total displacement of cars from their intended base has exactly the same distribution as the construction cost of linear probing hashed tables as seen by a "cycle lemma" originally due to Knuth and presented in [20]. The basic version of linear probing hashing, as described above, is based on a first-come-first-serve (FCFS) policy; alternative priority rules

Date: September 28, 1997.

The work of Philippe Flajolet was supported in part by the Long Term Research Project *Alcom-IT* (# 20244) of the European Union. The work of Patricio Poblete was supported in part by FONDECYT(Chile) under grant 1960881. The work of Alfredo Viola was supported in part by proyecto BID-CONICYT 140/94 and proyecto CONICYT fondo Clemente Estable 2078/96.

exist (like last-come-first-serve or "Robin Hood"), but total displacement remains unchanged. Thus, our analysis also applies directly to such variants of the basic algorithm.

Linear probing hashing has been the object of intense study; see the table on results and the bibliography in [11, pp. 51-54]. The simplicity of the algorithms goes well with efficiency, at least when tables are not too much filled. However, despite the simplicity of the algorithm, some of the probabilistic phenomena involved are not quite easy to capture. In addition, there is also special value for these problems since the first analysis of algorithms ever performed by Knuth [17] in 1962–1963 was that of linear probing hashing. As Knuth indicates in many of his writings, the problem has had a strong influence on his scientific carreer[1].

Sparse tables, by which we mean tables with a filling ratio $\alpha = n/m$ strictly less than 1, tend to behave reasonably well. This has been known, in the average case at least, since Knuth's first analysis. We establish here that the construction cost of a sparse table has an average that is $O(n)$, a standard deviation that is $O(\sqrt{n})$, and we provide very precise estimates for these quantities. The expectation estimate agrees naturally with the known fact that a random search or insertion in a sparse table has expected cost $O(1)$. In addition, we precisely characterize the distribution of construction costs and prove that it is Gaussian in the asymptotic limit. Thus, for sparse tables, observed values of costs are highly likely to be extremely close to what the average case analysis predicts.

In contrast, *full* ($m = n$) or *almost full* ($m = n - 1$) tables are much less well-behaved. The construction cost is $O(n^{3/2})$ on average, a fact also consistent with Knuth's early analyses demonstrating that late insertions in a table that fills up tend to have a superlinear cost. We provide here precise estimates for the standard deviation which turns out to be of the same order as the mean, namely $O(n^{3/2})$, an indication of the fairly high dispersion of costs. In fact, the construction cost admits a limit distribution that is of the "Airy type", involving Airy functions, or equivalently Bessel functions of orders that are multiple of $\frac{1}{3}$.

The analysis starts with almost full tables (Section 2) that are the basic combinatorial objects. The combinatorial principle on which this paper rests is a binary tree-like decomposition of full tables. From this, a difference-differential equation is derived that is the key to the analysis (Lemma 2). Moments, in either exact or asymptotic form, are obtained by a "pumping" process akin to the analysis of other cumulative parameters of combinatorial structures. For instance, similar methods have been used in the investigation of limit distributions for path length in trees (Takacs [42, 41]), the comparison cost of quicksort (Hennequin [14]), the area under random walks (Louchard [26, 27]), as well as in moment analysis of other combinatorial structures [16].

Sparse tables (Section 3) are then treated as labelled products of (almost) full tables, so that the corresponding generating functions involve large powers. For moments, especially for the mean and variance, the analysis results rather directly from that of full tables. However, for the limiting distribution, a somewhat delicate perturbative analysis of saddle point integrals is needed in order to derive a Gaussian law by means of characteristic function estimates.

Globally, these results reinforce our confidence that linear probing represents an excellent tradeoff between algorithmic simplicity and efficiency, as long as the filling ratio is not too large, say less than 2/3 or 3/4. These conclusions also apply to linear probing sort [11, pp. 168–170] whose analysis is almost isomorphic to that of linear probing hashing.

From the methodological standpoint, linear probing connects to a wealth of interesting combinatorial and analytic problems. A primary rôle is played by the tree function first studied by Eisenstein and by the Ramanujan-Knuth Q-function whose major properties we briefly recall in Section 1. Regarding limit laws, the Airy distribution that surfaces in the case of full tables is also present in random trees (inversions and path length), in random graphs (the complexity

[1]See the footnote in [20, p. 529].

or excess parameter), and in random walks (area); we discuss briefly in Section 4 some of the "reasons" for this fact. The Gaussian law of sparse tables is an instance of a general combinatorial scheme of some generality: our methods actually demonstrate that it should be expected in most cases where one deals with an additive parameter on a random assembly of a large number of random components.

1. The tree function and the Q-functions

The main character in this paper is the tree function that is defined implicitly by $T(z) = ze^{T(z)}$ and appears originally in problems related with the counting of rooted labeled trees [8, 12, 29, 36, 44]. The Lagrange inversion theorem provides a number of related series expansion like

$$T(z) = \sum_{n\geq 1} \frac{n^{n-1}}{n!} z^n, \qquad T(z)^m = m \sum_{n\geq m} \frac{n^{n-m-1}}{n!} n^{\underline{m}} z^n, \tag{1}$$

where $a^{\underline{k}} = a(a-1)\cdots(a-k+1)$. Most generating functions in this paper involve rational fractions in $T(z)$ with denominators that are powers of $(1-T)^{-1}$. Lagrange inversion also provides

$$\frac{1}{1-T(z)} = 1 + \sum_{n=1}^{\infty} n^n \frac{z^n}{n!}. \tag{2}$$

The asymptotic form of coefficients of any rational function of T is also directly recovered by singularity analysis [7, 30]. Application of the method requires the singular expansion of $T(z)$, itself obtained from the implicit function theorem.

Lemma 1. *The function $T(z)$ has a dominant singularity at $z = 1/e$, and its singular expansion there is*

$$T(z) = 1 - \delta(z) + \frac{1}{3}\delta(z)^2 - \frac{11}{72}\delta(z)^3 + \frac{43}{540}\delta(z)^4 + O\left(\delta(z)^5\right). \tag{3}$$

where $\delta(z) = \sqrt{2}\sqrt{1-ez}$.

The Q-functions. In close association with the tree function is what Knuth has popularized under the name of the "Ramanujan Q-function". This function [1, 18, 19, 20, 36] and its close relatives play a central rôle in the analysis of many algorithms and data structures —hashing with linear probing [17, 20], union-find algorithms [24], interleaved memory [23], optimal caching [21], and random mappings [2, 6, 19], most notably. The Q-function is defined by

$$Q(n) = 1 + \frac{n-1}{n} + \frac{(n-1)(n-2)}{n^2} + \cdots,$$

or, in a way that is equivalent thanks to (1),

$$\log \frac{1}{1-T(z)} = \sum_{n\geq 0} Q(n) n^{n-1} \frac{z^n}{n!}. \tag{4}$$

Singularity analysis of the generating function yields immediately

$$Q(n) \sim \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}} - \frac{4}{135n} + \cdots. \tag{5}$$

An asymptotic series for $Q(n)$ was first derived by Ramanujan [1], and tight estimates are obtained in [4].

For the purpose of expressing the average-case analysis of sparse tables, Knuth has extended the Ramanujan Q-function as

$$Q_0(m,n) = \sum_{i\geq 0} \frac{n^{\underline{i}}}{m^i}, \tag{6}$$

so that $Q(n) = Q_0(n, n-1)$. From the definition, one has

$$\sum_{n=0}^{\infty} Q_0(m,n) m^n \frac{t^n}{n!} = \frac{e^{mt}}{1-t}. \tag{7}$$

Basic asymptotic approximations entail

$$\begin{aligned} Q_0(m, \alpha m - 1) &= \frac{1}{1-\alpha} - \frac{1}{(1-\alpha)^3} m^{-1} + \frac{2+\alpha}{(1-\alpha)^5} m^{-2} - \frac{\alpha^2+8\alpha+6}{(1-\alpha)^7} m^{-3} \\ &\quad + \frac{\alpha^3+22\alpha^2+58\alpha+24}{(1-\alpha)^9} m^{-4} + O(m^{-5}). \end{aligned} \tag{8}$$

See [32] for a general framework.

2. Full tables

Throughout this paper, we consider tables that have m locations (m is called the "length" of the table) and we let n denote the number of keys (the "size"). Clearly, the number of tables (the number of hash sequences) with length m and size n is m^n, and such a table has $m-n$ empty locations. By circular symmetry [20], for tables such that $m > n$, we may freely assume that one of the empty locations is the rightmost one,. *This assumption of a last empty location is made from now onwards.* When $n = m-1$, we say that such a table table is *almost full*. Since there are $m-n$ empty locations, then the probability of the rightmost cell being empty is $(m-n)/m$, and therefore there are $m^{n-1}(m-n)$ ways of creating such tables. In particular, the number of almost full tables is $m^{m-2} = (n+1)^{n-1}$.

Inserting the last element into an almost full table yields a *completely full* table. Since this last element may hash to any of the m locations of the table, there are $m^{m-1} = n^{n-1}$ ways of creating a full table in this way. Thus, by our convention almost full and completely full tables don't "wrap around.". Clearly, the distributions of total displacements $d_{n,n-1}$ and $d_{n,n}$ are not affected by such a restriction.

Notations. The analysis is carried out by means of bivariate generating functions and moments are then recovered via a family of operators defined as follows. For any function $G(z,q)$,

$$\begin{cases} \mathrm{U}\, G(z,q) = G(z,1) & \partial_q G(z,q) = \dfrac{\partial G(z,q)}{\partial q} \\ \mathrm{Z} G(z,q) = zG(z,q) & \partial_z G(z,q) = \dfrac{\partial G(z,q)}{\partial z} \\ \mathrm{H}\, G(z,q) = \dfrac{G(z,q) - qG(qz,q)}{1-q}. \end{cases} \tag{9}$$

These operators act in the usual way on formal power series $G(z,q) = \sum_n g_n(q) \frac{z^n}{n!}$, with each $g_n(q)$ a polynomial; in particular,

$$\mathrm{H}\, G(z,q) = \sum_n g_n(q)(1+q+q^2+\cdots+q^n)\frac{z^n}{n!}.$$

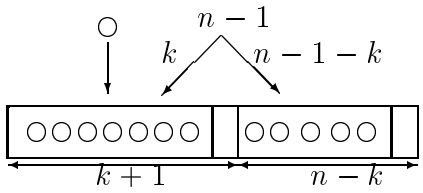


FIGURE 1. The binary tree decomposition of almost full tables.

Mike Paterson has designed an ingenious operator framework for the "local" analysis of displacements; see the account of the "cookie monster" in [13]. The problem of total displacement being fully history–dependent is however not clearly amenable to Paterson's techniques.

2.1. **Combinatorial analysis.** We define $F_{n,k}$ as the number of ways of creating an almost full table with n elements and total displacement k. The corresponding bivariate generating function is then

$$F(z,q) = \sum_{n,\, k\geq 0} F_{n,k}\ q^k \frac{z^n}{n!}.$$

and it starts like

$$F(z,q) = 1 + \frac{z}{1!} + (2+q)\frac{z^2}{2!} + (6+6q+3q^2+q^3)\frac{z^3}{3!} + \cdots,$$

Consider an almost full table of size n (and length $n+1$). Right before the last element is inserted, the table has two empty cells: one at some position $k+1$, the other at position $n+1$ (see Figure 1). Then, the element that is last to be inserted has an address that is any number of the interval $[1\,..\,k+1]$, which corresponds to a displacement that assumes any value in $[0\,..\,k]$. The counting of possibilities gives rise to a recurrence on the $F_n(q) = n![z^n]F(z,q)$:

$$F_n(q) = \sum_{k=0}^{n-1} \binom{n-1}{k} F_k(q)(1+q+\cdots+q^k)F_{n-1-k}(q). \tag{10}$$

This fundamental recurrence reflects a recursive binary decomposition of full tables. We recognize here a product of exponential generating functions modified by the occurrence of the H-operator defined in (9).

Lemma 2 (Basic functional equation)**.**

$$\frac{\partial}{\partial z}F(z,q) = F(z,q)\cdot\frac{F(z,q)-qF(qz,q)}{1-q}. \tag{11}$$

In operator notation, this reads simply as $\partial_z F = F\cdot \mathrm{H}F$.

Let similarly $C_{n,k}$ be the number of completely full tables of size n, with $C(z,q) = \sum_{n,k} C_{n,k}q^k z^n/n!$ the corresponding bivariate generating function. Since a completely full table of size $n+1$ is created by inserting the last element in an almost full table of size n, we have from the definition of the H-operator ,

$$\partial_z C(z,q) = \mathrm{H}F(z,q).$$

Note that the basic functional equation together with this last relation implies the additional relations

$$F(z,q) = e^{C(z,q)} \quad \text{or} \quad C(z,q) = \log F(z,q). \tag{12}$$

Not surprisingly, the analyses of total displacement in full and in almost full tables are closely related.

2.2. **Moments.** For total displacement in almost full tables, what we call the generating function of rth factorial moments is by definition,

$$f_r(z) := \mathrm{U}\,\partial_q^r F(z,q) = \frac{\partial^r}{\partial q^r} F(z,q)\Big|_{q=1}. \tag{13}$$

This name is justified by the fact that the rth factorial moment of total displacement is given by

$$\mathbf{E}[d\,(d-1)\cdots(d-r+1)] = \frac{[z^n]f_r(z)}{[z^n]f_0(z)} \qquad \text{where} \quad d \equiv d_{n,n-1}.$$

The basic functional equation (11) implicitly contains all the information about moments. We now develop properties of the family of operators introduced in (9) designed to extract such moments explicitly.

First, let us rederive the enumeration of full tables. What is needed is $f_0(z) := \mathrm{U}\,F(z,q)$, where F is determined by (11). Now, from the action of H on power series, one has

$$\mathrm{U}\,\mathrm{H} = \partial_z\,\mathrm{Z}\,\mathrm{U} \quad \text{ or equivalently } \quad \mathrm{U}\,\mathrm{H}\,F(z,q) = \frac{\partial}{\partial z}(zG(z,1)).$$

Thus, $f_0(z)$ satisfies the nonlinear differential equation obtained by applying U to (11):

$$Y'(z) = Y(z)(zY(z))'.$$

This equation is equivalent to $(\log Y(z))' = (zY(z))'$, and so $Y(z) = e^{zY(z)}$. In other words,

$$f_0(z) \equiv F(z,1) = \frac{1}{z}T(z) = e^{T(z)},$$

where $T(z)$ is the classical tree function. Therefore, by (1), the number of almost full tables is $(n+1)^{n-1}$. Similarly, by (12), $\mathrm{U}\,C(z,q) = \log(f_0(z)) = T(z)$ so that the number of completely full tables is n^{n-1}. These values in accordance with what we know already from direct combinatorial reasoning.

A similar device produces moments upon applying $\mathrm{U}\,\partial_q^r$ to the fundamental equation (11). What is needed is a "commutation rule" for $\mathrm{U}\,\partial_q^r$ and H. This is readily found for $r = 1$ since

$$\mathrm{U}\,\partial_q\,\mathrm{H}(z^n q^k) = \mathrm{U}\,\partial_q((1+q+\cdots+q^n)z^n q^k) = ((1+2+\cdots+n)+(n+1)k)z^n.$$

Thus, symbolically

$$\mathrm{U}\,\partial_q\,\mathrm{H} = \frac{1}{2}\mathrm{Z}\partial_z^2 Z + \mathrm{U}\,\partial_z\,\mathrm{Z}\partial_q, \tag{14}$$

and by similar devices,

$$\mathrm{U}\,\partial_q^2\,\mathrm{H} = \partial_z\,\mathrm{Z}\,\mathrm{U}\,\partial_q^2 + \mathrm{Z}\partial_z^2\,\mathrm{Z}\,\mathrm{U}\,\partial_q + \frac{1}{3}\mathrm{Z}^2\partial_z^3\,\mathrm{Z}\,\mathrm{U}. \tag{15}$$

As a consequence, $f_1(z)$ and $f_2(z)$ satisfy the following linear ordinary differential equations,

$$\mathcal{L}Y = \frac{1}{2}zf_0(zf_0)'' \tag{16}$$

$$\mathcal{L}Y = zf_1(zf_0)'' + 2f_1(zf_1)' + \frac{1}{3}z^2 f_0(zf_0)''' + zf_0(zf_1)'', \tag{17}$$

where $\mathcal{L}$ is the differential operator

$$\begin{aligned} \mathcal{L}Y &= Y' \cdot (1 - z f_0) - Y \cdot ((z f_0)' + f_0) \\ &= Y' \cdot (1 - T) - Y \cdot \frac{T(2-T)}{z(1-T)}. \end{aligned} \tag{18}$$

The corresponding homogeneous ordinary differential equation,

$$\mathcal{L}Y = 0$$

admits the solution,

$$Y(z) = \frac{e^{T(z)}}{1 - T(z)}. \tag{19}$$

The variation–of–constant method then applies to the inhomogeneous differential equations (16) and (17) that are of the both form

$$\mathcal{L}Y(z) = R(z),$$

and yields the solution

$$Y(z) = \frac{e^{T(z)}}{1 - T(z)} \int_0^z R(u) e^{-T(u)}\, du. \tag{20}$$

The quantities appearing in these differential equations can be expressed as functions of $T(z)$ alone since $z = Te^{-T}$ and $dz = (1-T)e^{-T}dT$. Thus the integrations needed in the variation-of-constant method all eventually reduce to integration of elementary functions for which decision procedures exist. We then obtain mechanically the generating functions of the first two moments for an almost full table. (This is for instance well within the capabilities of the computer algebra system Maple.)

Lemma 3 (Almost full tables, generating functions for the moments)**.**

$$\begin{aligned} z f_1(z) &= \frac{1}{2} \frac{T^3(z)}{(1 - T(z))^2}, \\ z f_2(z) &= \frac{1}{12} \frac{T(z)^4 (24 - 11T(z) + 2T(z)^2)}{(1 - T(z))^5} \end{aligned}$$

For a completely full table, the corresponding generating functions result from Lemma 3 and equation (12):

$$\mathrm{U}\, \partial_q C(z,q) = \frac{f_1}{f_0} = \frac{1}{2} \frac{T^2}{(1-T)^2}, \tag{21}$$

$$\mathrm{U}\, \partial_q^2 C(z,q) = \frac{f_2 f_0 - f_1^2}{f_0^2} = \frac{1}{12} \frac{T^3 (24 - 14T + 5T^2)}{(1-T)^5}. \tag{22}$$

Explicit expressions for the coefficients of functions appearing in (21) and (22) are then obtained from the expansions (2) and (4). Since $T(z)$ satisfies the differential relation

$$(\mathrm{Z}\partial_z) T(z) = \frac{T(z)}{1 - T(z)},$$

the class of functions

$$\left\{ (\mathrm{Z}\partial_z)^r \frac{1}{1-T} \right\}_{r=0}^{\infty}, \qquad \left\{ (\mathrm{Z}\partial_z)^r \log \frac{1}{1-T} \right\}_{r=1}^{\infty},$$

spans a linear space that contains all the rational functions of the form $A(T)/(1-T)^r$, with A a polynomial of degree $< r$. As a consequence, for any such rational function of T, there exists an expansion

$$[z^n]\frac{A(T(z))}{(1-T(z))^r} = \frac{n^{n-1}}{n!}\left(U(n)+V(n)\,Q(n)\right), \tag{23}$$

for some polynomials U and V that can be mechanically determined.

Theorem 1 (Full tables, exact form of moments)**.**

$$\begin{aligned}\mathbf{E}[d_{n,n}] &= \frac{n}{2}(Q(n)-1)\\ \mathbf{E}[d_{n,n}^2] &= \frac{n}{12}(5n^2+4n-1-8n\;Q(n)).\end{aligned}$$

Thanks to (3), singularity analysis applies directly to the solutions (21) and (22). (Alternatively, the explicit forms of Theorem 1 can be used in conjunction with (5).)

Theorem 2 (Full tables, asymptotic form of moments)**.**

$$\begin{aligned}\mathbf{E}[d_{n,n}] &= \frac{\sqrt{2\pi}}{4}n^{3/2}-\frac{2}{3}n+\frac{\sqrt{2\pi}}{48}n^{1/2}-\frac{2}{135}+O(n^{-1}),\\ \mathbf{Var}[d_{n,n}] &= \frac{10-3\pi}{24}n^3+\frac{16-3\pi}{144}n^2+\frac{\sqrt{2\pi}}{135}n^{3/2}-\frac{\pi+48}{576}n+O(n^{1/2}).\end{aligned}$$

2.3. **Limit law.** Our goal in this subsection is to establish the existence of a limit distribution for the construction cost of almost full tables. As this limit distribution turns out not to be part of the set of classical continuous distributions, we first precisely specify it.

Definition 1. *The* Airy distribution *is the probability distribution of a random variable X with support on $[0,+\infty)$ that is uniquely determined by its moments,*

$$\mathbf{E}[X^r] = -\frac{\Gamma(-\frac{1}{2})}{\Gamma(\frac{3r-1}{2})}\,\Omega_r,$$

where the basic constants Ω_r are defined by the formal series expansion

$$\sum_{r\geq 0}\Omega_r\frac{w^r}{r!} = -\frac{\Phi_{2/3}(w)}{\Phi_{-1/3}(w)},$$

with

$$\begin{aligned}\Phi_\nu(w) &= 1-(4\nu^2-1)\left(\frac{w}{24}\right)+\frac{(4\nu^2-1)(4\nu^2-9)}{2!}\left(\frac{w}{24}\right)^2\\ &\quad-\frac{(4\nu^2-1)(4\nu^2-9)(4\nu^2-25)}{3!}\left(\frac{w}{24}\right)^3+\cdots.\end{aligned}$$

Under various guises, the Airy distribution arises as a limit distribution in quite diverse contexts. Examples include the area under nonnegative random walks (Louchard [26, 27]) or path length in random trees (Takacs [42, 41]); this limit law also relates to asymptotic estimates of connectivity in random graphs (Wright [46], Janson *et al.* [15]). At the end of this paper, we comment briefly on the combinatorics that underlies some of these connections. Our derivation here follows in spirit the approach of Louchard and Takacs [26, 27, 42, 41] who also justified that the Airy distribution as defined here is indeed uniquely characterized by its moments.

We now examine in detail the process that yields the moments asymptotically and show how the Airy distribution arises from a recurrent determination of moments.

First, a process similar to the one employed for the first two moments yields a general commutation rule for the H and ∂_q operators

Lemma 4.

$$\mathrm{U}\,\partial_q^j\,\mathrm{H} = \sum_{s=0}^{j} \binom{j}{s} \frac{1}{s+1} \mathrm{Z}^s \partial_z^{s+1} \mathrm{Z}\,\mathrm{U}\,\partial_q^{j-s}. \tag{24}$$

Proof. The left hand side applied to $z^n q^k$ gives

$$\mathrm{U}\,\partial_q^j\,\mathrm{H}(z^n q^k) = z^n\,\mathrm{U}\,\partial_q^j \left(\sum_{i=0}^{n} q^i q^k \right).$$

Then, the Leibniz rule applied to the differentiation of products $q^i q^k$ yields

$$\begin{aligned} z^n\,\mathrm{U}\,\partial_q^j \left(\sum_{i=0}^{n} q^i q^k \right) &= z^n \sum_{i=0}^{n} \sum_{s=0}^{j} \binom{j}{s} i^{\underline{s}}\, k^{\underline{j-s}} = z^n \sum_{s=0}^{j} \binom{j}{s} \frac{1}{s+1}\, (n+1)^{\underline{s+1}}\, k^{\underline{j-s}} \\ &= \sum_{s=0}^{j} \binom{j}{s} \frac{1}{s+1} \mathrm{Z}^s \partial_z^{s+1} \mathrm{Z}\,\mathrm{U}\,\partial_q^{j-s}(z^n q^k). \end{aligned}$$

□

Then, a differential equation for the rth factorial moment generating function f_r is directly obtained by a combination of the Leibniz's rule and of the commutation relation (24) applied to the fundamental equation (11):

$$\partial_z f_r(z) = \sum_{j=0}^{r} \sum_{s=0}^{j} \binom{r}{j} \binom{j}{s} \frac{1}{s+1} f_{r-j}(z) \cdot \left(\mathrm{Z}^s \partial_z^{s+1} \mathrm{Z} f_{j-s}(z) \right). \tag{25}$$

The differential equation (25) that gives access to the rth moment is of the form

$$\mathcal{L} Y(z) = R_r(z),$$

where $\mathcal{L}$ is the linear differential operator of (18). There, $R_r(z)$ is exactly the right hand side of (25) stripped of its terms than contain $f_r(z), f_r'(z)$, namely the terms corresponding to $(j,s) = (0,0)$ and $(j,s) = (r,0)$. By the variation-of-constant method, moments can then be pumped *ad libidinem*, and we have from (20)

$$f_r(z) = \frac{e^{T(z)}}{1 - T(z)} \int_0^z R_r(u) e^{-T(u)}\, du. \tag{26}$$

For instance, we obtain automatically

$$\begin{aligned} z f_1 &= \frac{T^3}{2} \frac{1}{(1-T)^2}, \qquad z f_2 = \frac{T^4}{12} \frac{24 - 11\,T + 2\,T^2}{(1-T)^5} \\ z f_3 &= \frac{T^4}{8} \frac{8 + 144\,T - 110\,T^2 + 63\,T^3 - 17\,T^4 + 2\,T^5}{(1-T)^8} \\ z f_4 &= \frac{T^5}{240} \frac{10800 + 64560\,T - 60072\,T^2 + 53760\,T^3 - 26865\,T^4 + 9140\,T^5 - 1750\,T^6 + 152\,T^7}{(1-T)^{11}} \end{aligned} \tag{27}$$

The success of the pumping method is obvious as regards asymptotic forms at least since conditions of singularity analysis are preserved under multiplication by rational functions of T and under integration. (In fact, there always exist exact rational forms in T, as shown by a more sophisticated argument, but this is immaterial here.) An asymptotic pattern clearly emerges,

$$z f_1 \sim \frac{1}{2} \frac{1}{(1-T)^2},\ z f_2 \sim \frac{5}{4} \frac{1}{(1-T)^5},\ z f_3 \sim \frac{45}{4} \frac{1}{(1-T)^8},\ z f_4 \sim \frac{3315}{16} \frac{1}{(1-T)^{11}},$$

where the approximations hold when $z \to e^{-1}$, that is to say $T \to 1$. The following lemma precisely characterizes the dominant terms of $f_r(z)$.

Lemma 5. *The factorial moment generating functions satisfy, for $r \geq 1$,*

$$zf_r(z) \sim \frac{C_r}{(1-T(z))^{3r-1}} \sim \frac{C_r}{(2(1-ez))^{3r/2-1/2}} \qquad (z \to e^{-1}), \tag{28}$$

where the constants C_r are determined by the nonlinear recurrence

$$(3r-4)rC_{r-1} + \sum_{j=0}^{r} \binom{r}{j} C_j C_{r-j} - \delta_{r,0} = 0, \qquad C_0 = -1. \tag{29}$$

Proof. The property holds for $r = 1, 2$ by Lemma 3. For general r, the variation-of-constant formula (26) entails (by induction) that the singular behaviour f_r is of the form $zf_r \sim C_r(1-T)^{-3r+1}$ as $z \to e^{-1}$. In other words, the ∂_z operator shifts a singular expansion by a factor of $(1-T)^{-2}$ while the ∂_q operator shifts such an expansion by a factor of $(1-T)^{-3}$.

Thus, the dominant contribution from (25) arises from the terms corresponding to $s = 0$ and $(j, s) = (r, 1)$. As a consequence we have, as $z \to e^{-1}$,

$$\partial_z f_r(1-zf_0) - f_r \partial_z(zf_0) = \frac{r}{2}\partial_z^2(zf_{r-1}) + \sum_{j=1}^{r-1} \binom{r}{j} f_{r-j}\partial_z(zf_j) + O((1-T)^{-3r+1}). \tag{30}$$

Integration and multiplication of both sides by $2z$ yields the asymptotic relation

$$2zf_r(1-zf_0) = rz\partial_z(zf_{r-1}) + \sum_{j=1}^{r-1} \binom{r}{j} (zf_{r-j})(zf_j) + O((1-T)^{-3r+3}). \tag{31}$$

The coefficients of the dominant terms involving $(1-T)^{-3r+2}$ can then be identified, and this provides a recursive determination of the coefficients C_r:

$$2C_r = (3r-4)rC_{r-1} + \sum_{j=1}^{r-1} \binom{r}{j} C_j C_{r-j} \quad r \geq 1.$$

There, by a natural convention, we take $C_0 = -1$ since $f_0 = 1 - (1-T)$ and it is singular components that count. This recurrence is equivalent to the one stated in (29). □

The constants C_r determine the dominant asymptotic form of the *moments* of the law of total displacement. Clearly, factorial moments and power moments are asymptotically equivalent, and, by singularity analysis, one has

$$\mu_n^{(r)} \equiv \mathbf{E}[(d_{n,n-1})^r] = \frac{2\sqrt{\pi}}{\Gamma(\frac{3r-1}{2})} C_r \left(\frac{n}{2}\right)^{3r/2} \left(1 + O(n^{-1/2})\right). \tag{32}$$

In order to establish the Airy limit distribution property, it is then necessary to identify the coefficients in (32). We are going to show that in fact $C_r = \Omega_r$, with the Ω_r the fundamental constants of Definition 1.

From (29), the quantities $\gamma_r := C_r/r!$ satisfy a nonlinear recurrence

$$(3r-4)\gamma_{r-1} + \sum_{j=0}^{r} \gamma_j \gamma_{r-j} - \delta_{r,0} = 0, \tag{33}$$

so that the exponential generating function of the C_r, $\gamma(z) := \sum_{r\geq 0} C_r z^r/r!$, itself satisfies a nonlinear first order ODE of the Riccati type.

$$3z^2\gamma'(z) - z\gamma(z) + \gamma(z)^2 - 1 = 0. \tag{34}$$

r	0	1	2	3	4	5	6	7	8
Ω_r	-1	$\frac{1}{2}$	$\frac{5}{4}$	$\frac{45}{4}$	$\frac{3315}{16}$	$\frac{25425}{4}$	$\frac{18635625}{64}$	$\frac{18592875}{1}$	$\frac{1282031525}{32768}$
ω_r	-1	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{15}{8}$	$\frac{1105}{128}$	$\frac{1695}{32}$	$\frac{414125}{1024}$	$\frac{59025}{16}$	$\frac{1282031525}{32768}$
$\omega_r^\star$	$-\frac{1}{2}$	1	5	60	1105	27120	828250	30220800	1282031525
$\mu^{(r)}$	1	$\sqrt{\pi}$	$\frac{10}{3}$	$\frac{15}{4}\sqrt{\pi}$	$\frac{884}{63}$	$\frac{565}{32}\sqrt{\pi}$	$\frac{662600}{9009}$	$\frac{19675}{192}\sqrt{\pi}$	$\frac{4102500880}{8729721}$

TABLE 1. The Airy constants Ω_r and their various normalizations: $\omega_r = \Omega_r/r!$, $\omega_r^\star = 2^{2r-1}\Omega_r/r!$, $\mu^{(r)} = -\Omega_r\Gamma(-1/2)/\Gamma((3r-1)/2)$. (The $\mu^{(r)}$ are the moments of the Airy distribution.)

In a way, this basic equation is a "reduced image" of the fundamental difference-differential equation when only dominant singular parts are retained. Now, it is known that Riccati equation are reducible to linear second order ODE's: set $\gamma(z) = 3z^2 g'(z)/g(z)$, so that

$$9z^4 g''(z) + 15z^3 g'(z) - g(z) = 0. \tag{35}$$

From there, the connection with Bessel functions is easy to establish and a computer algebra like Maple provides valuable hints. Some care is however needed due to the multivalued character of Bessel functions of nonintegral order, so that we provide some detail.

The "modified" Bessel functions are defined by

$$\left\{ \begin{array}{rcl} I_\nu(z) & = & \left(\frac{z}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{(z^2/4)^k}{k!\Gamma(\nu+k+1)} \\ K_\nu(z) & = & \frac{\pi}{2\sin\nu\pi}\left(I_{-\nu}(z) - I_\nu(z)\right), \end{array} \right. \tag{36}$$

and for nonintegral ν, they form of basis of solutions to the Bessel equation

$$z^2\frac{d^2w}{dz^2} + z\frac{dw}{dz} - (z^2+\nu^2)w = 0. \tag{37}$$

One can then simply match Equations (35) with (37) and verify that the general solution to (35) is

$$g(z) = z^{-1/3}\left(\lambda_1 K_{-1/3}(\frac{1}{3z}) + \lambda_2 I_{-1/3}(\frac{1}{3z})\right).$$

A simple computation then shows that the general solution of the original Riccati equation is

$$\gamma^{\langle\lambda\rangle}(z) = -\frac{I_{2/3}(\frac{1}{3z}) - \lambda K_{2/3}(\frac{1}{3z})}{I_{-1/3}(\frac{1}{3z}) + \lambda K_{-1/3}(\frac{1}{3z})} \tag{38}$$

For determinacy, we restrict (38) to the complex z-plane slit along $(-\infty, 0)$.

We note at this stage that Bessel functions of order a multiple of $1/3$ are related to the classical Airy functions that are defined as solutions to the linear differential equation $w'' - zw = 0$. In particular, one has

$$\begin{array}{rcl} \mathrm{Ai}(z) & = & \frac{1}{\pi}\int_0^\infty \cos\left(\frac{1}{3}t^3 + zt\right) dt \\ & = & \frac{1}{\pi}\left(\frac{z}{3}\right)^{1/2} K_{1/3}\left(\frac{2z^{3/2}}{3}\right). \end{array}$$

This (and other connections) justify our choice of naming the distribution of Definition 1 the Airy distribution.

Obviously, the $\gamma^{\langle\lambda\rangle}(z)$ as obtained in (38) are nonanalytic at 0. Then, Eq. (38) is to be taken in the sense that the divergent (formal) series $\gamma^{\langle\lambda\rangle}(z)$ represents asymptotically the right hand side as $z \to 0^+$. But, asymptotic expansions of Bessel functions are well-known: with $\mu = 4\nu^2$, we have as the variable z tends to $+\infty$,

$$I_\nu(z) \sim \frac{e^z}{\sqrt{2\pi z}} \left(1 - \frac{\mu-1}{8z} + \frac{(\mu-1)(\mu-9)}{2!(8z)^2} - \frac{(\mu-1)(\mu-9)(\mu-25)}{3!(8z)^3} + \cdots\right), \tag{39}$$

while each $K_\nu(z) = O(e^{-z})$ is exponentially small. Thus, the asymptotic expansions of all the $\gamma^{\langle\lambda\rangle}(z)$ in the scale $\{z^{-m}\}$ coincide, and we may as well take $\gamma(z) = \gamma^{\langle 0\rangle}(z)$. In other words, the C_r are generated as coefficients in the asymptotic expansion,

$$-\frac{I_{2/3}(\frac{1}{3z})}{I_{-1/3}(\frac{1}{3z})} \sim \sum_{r=0}^{\infty} C_r \frac{z^r}{r!} \qquad (z \to 0^+). \tag{40}$$

From (39) and (40), we thus obtain a purely algebraic and explicit specification of $\gamma(z)$ as a quotient of two divergent hypergeometric series (of the ${}_2F_0$ type) that matches exactly the definition of the Airy distribution, with $C_r = \Omega_r$. This characterizes the distribution of construction cost in almost full hash tables.

Theorem 3 (Limit law for full tables)**.** *For almost full tables, the distribution of the random variable* $\frac{d_{n,n-1}}{(n/2)^{3/2}}$ *converges to the Airy distribution, in the sense that, pointwise for each* x*,*

$$\Pr\{\frac{d_{n,n-1}}{(n/2)^{3/2}} \le x\} \to \Pr\{X \le x\} \qquad (n \to \infty),$$

where X *is Airy distributed in the sense of Definition 1. The same property holds for completely full tables and the random variable* $\frac{d_{n,n}}{(n/2)^{3/2}}$*.*

(The property for full tables results from the fact that $d_{n,n}$ has the same distribution as $d_{n,n-1} + \mathcal{U}_n$, where $\mathcal{U}_n$ is uniform over $[0\,..\,n-1]$.)

Initial values of the Airy constants are given in Table 1. The normalized constants $\omega_r^\star = 2^{2r-1}\Omega_r/r!$ turn out to be integers for $r \ge 1$. This interesting sequence starts like

> 1, 5, 60, 1105, 27120, 828250, 30220800, 1282031525, 61999046400, 3366961243750,
> 202903221120000, 13437880555850250, 970217083619328000, 75849500508999712500,
> 6383483988812390400000, 575440151532675686278125, 55318762960656722780160000,

and we propose to call it the Wright–Louchard-Takacs sequence (see the remarks above and our conclusion). It is however *not* to be found in Sloane and Plouffe's *Encyclopedia of Integer Sequences* [38]. The variance of the Airy distribution is

$$\frac{10}{3} - \pi = 0.19174\,06797\,43540\ldots$$

and the appearance of this magic value in a variance expression may be taken as a good indication of the possible occurrence of the Airy distribution.

3. Sparse Tables

In this section, we analyse sparse tables, where the filling ratio defined as $\alpha = n/m$ is bounded away from 1. The behaviour of such tables turns out to be much more tame than that of full tables discussed in the previous section.

3.1. **Combinatorial analysis.** As seen at the beginning of Section 2, a simple circular symmetry argument enables us to restrict attention to tables whose last location is empty. Such a table then decomposes as a labelled product of $m-n$ clusters (sometimes also figuratively called "islands") that are, up to relabelling, almost full tables. Note that a cluster may well have size 0, in which case it comprises only one unoccupied cell. For instance, the table

	3	9	4	7			5	2			8		1	6	

is, up to relabelling, a sequence of six almost full tables of respective sizes 0,4,0,2,1,2.

Define the generating function $H_{m,n}(q)$ that counts the number of ways of creating a non-full table of lenght m and size n (the rightmost location is empty) with q marking the total displacement. The construction cost (or total displacement) of partial tables is inherited additively from component clusters. Therefore, the total displacement in partial tables of parameter (m,n) has generating function

$$H_{m,n}(q) = n![z^n]F(z,q)^{m-n}.$$

The number of tables of length m, size n, with the last location empty is then

$$H_{m,n}(1) = n![z^n]f_0(z)^{m-n} = n![z^m]T(z)^{m-n},$$

a quantity that, by virtue of (1), equals $(m-n)m^{n-1}$, in agreement with the circular symmetry argument. The probability generating function of the total displacement $d_{m,n}$ is then

$$\frac{H_{m,n}(q)}{H_{m,n}(1)} = \frac{n!}{(m-n)m^{n-1}}[z^n]F(z,q)^{m-n}.$$

3.2. **Moments.** The generating functions for sparse tables admit power forms that lend themselves nicely to differentiation. In this way, moment generating functions are obtained immediately from the corresponding computation for full tables.

The analysis still relies on the functions $f_r = \mathrm{U}\,\partial_q{}^r F$ introduced in (13). We have

$$\begin{aligned}
\mathrm{U}\,\partial_q F(z,q)^{m-n} &= (m-n)f_0^{m-n-1}f_1\\
\mathrm{U}\,\partial_q{}^2 F(z,q)^{m-n} &= (m-n)(m-n-1)f_0^{m-n-2}f_1^2 + (m-n)f_0^{m-n-1}f_2.
\end{aligned}$$

The values of zf_0, zf_1, zf_2 are known from Section 2, and are expressible in terms of $T = T(z)$ alone; this gives for instance,

$$\frac{n!}{(m-n)m^{n-1}}[z^n]\,\mathrm{U}\,\partial_q F(z,q)^{m-n} = \frac{n!}{m^{n-1}}[z^m]\frac{1}{2}\frac{T(z)^{m-n+2}}{(1-T(z))^2},$$

What is required at this point in order to obtain explicit forms is a method for coefficient extraction,

$$[z^m]\,T(z)^{m-n}\,\frac{A(T(z))}{(1-T(z))^r}, \tag{41}$$

where A is a polynomial of degree $< r$. For computational purposes, it is convenient to introduce the change of variables in Cauchy coefficient integrals that underlies Lagrange inversion.

$$\begin{aligned}
[z^m]\lambda(T(z)) &= \frac{1}{2i\pi}\int \lambda(T(z))\frac{dz}{z^{m+1}}\\
&= \frac{1}{2i\pi}\int \lambda(T)(1-T)e^{-T}\,\frac{dT}{(Te^{-T})^{m+1}}\\
&= [t^n]e^{mt}(1-t)\lambda(t). \qquad (42)
\end{aligned}$$

(Small contours around 0 are understood in this derivation, and this shortcut is of course logically equivalent to Lagrange-Bürmann inversion.)

Then, the application of (42) to (41) yields

$$[z^m]T^{m-n}\,\frac{A(T)}{(1-T)^r} = [t^n]e^{mt}\,\frac{A(t)}{(1-t)^{r-1}}.$$

This is close to the form (7) of the generating function of $Q_0(m,n)$. Now, an argument similar to the one used in (23) for full tables applies. The linear space spanned by

$$\left\{\left(t\frac{d}{dt}\right)^r \frac{e^{mt}}{1-t}\right\}_{r=0}^{\infty} \cup \left\{e^{mt}\right\}$$

contains all the rational functions of the form $e^{mt}A(t)/(1-t)^{r-1}$. Thus, there exist polynomials U and V such that

$$[z^m]T^{m-n}\,\frac{A(T)}{(1-T)^r} = \frac{m^n}{n!}\left(U(m,n) + V(m,n)\,Q_0(m,n)\right).$$

The computation is again purely mechanical. It can be recast in terms of $Q_0(m,n-1)$ since $Q_0(m,n) = 1 + \frac{m}{n}Q_0(m,n-1)$.

Theorem 4 (Sparse tables, exact form of moments)**.**

$$\begin{aligned}
\mathbf{E}[d_{m,n}] &= \frac{n}{2}(Q_0(m,n-1)-1),\\
\mathbf{E}[d^2_{m,n}] &= \frac{n}{12}\left((m-n)^3+(n+3)(m-n)^2+(8n+1)(m-n)+5n^2+4n-1\right.\\
&\qquad \left.-((m-n)^3+4(m-n)^2+(6n+3)(m-n)+8n)Q_0(m,n-1)\right).
\end{aligned}$$

The approximation formula (8) then produces the asymptotic form of the first moments and the variance of an α-sparse table.

Theorem 5 (Sparse tables, asymptotic form of moments)**.**

$$\begin{aligned}
\mathbf{E}[d_{m,n}] &= \frac{\alpha}{2(1-\alpha)}n - \frac{\alpha}{2(1-\alpha)^3} + O(n^{-1}),\\
\mathbf{Var}[d_{m,n}] &= \frac{6\alpha-6\alpha^2+4\alpha^3-\alpha^4}{12(1-\alpha)^4}n - \frac{6\alpha^3+24\alpha^2+6\alpha}{12(1-\alpha)^6} + O(n^{-1}).
\end{aligned}$$

3.3. **Limit law.** In this subsection, we estimate the distribution of total displacement in sparse tables, when m,n tend to infinity in such a way that the filling ration $\alpha = n/m$ remains constant. We thus fix throughout α and assume $0<\alpha<1$ The mean μ_n and the variance σ_n^2 of the distribution are in this case both $O(n)$ and their precise form has been given by the last two theorems.

The limit law is approached here by characteristic functions rather than by moments as was done in the case of full tables. Indeed, cancellations already present in the variance preclude a moment approach. On the other hand, the power form of the involved generating functions suggests an appeal to the saddle point method applied to Cauchy coefficient integrals, this in order to estimate characteristic functions. Some care is however needed since $F(z,q)$ is sharply nonanalytic at $q=1$. The analysis proceeds by a (delicate) perturbation of the (easy) saddle point estimates of the univariate problem of counting sparse tables, namely $[z^n]F(z,1)^{m-n}$.

Theorem 6 (Limit law for sparse tables)**.** *The limit law of total displacement $d_{m,n}$ in tables with filling ratio $\alpha = \frac{n}{m}$ that satisfies $\alpha<1$ is asymptotically Gaussian, as $n\to\infty$,*

$$\Pr\left\{\frac{d_{m,n}-\mu_n}{\sigma_n} \le x\right\} \to \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-s^2/2}\,ds,$$

where $\mu_n = \mathbf{E}[d_{m,n}]$ is the mean of the distribution and σ_n defined by $\sigma_n^2 = \mathbf{Var}[d_{m,n}]$ is the standard deviation, as given by Theorems 4 and 5.

Proof. By Lévy's continuity theorem, it is sufficient to consider the characteristic function of the standardized distribution (centred around it mean and scaled by its standard deviation), that is,

$$\phi_n^\star(t) = \frac{1}{[z^n]F(z,1)^{m-n}} \left(e^{it\mu_n/\sigma_n} [z^n] F(z, e^{it/\sigma_n})^{m-n} \right), \tag{43}$$

and prove that it converges pointwise for any fixed t to the characteristic function of a standard normal variate,

$$\phi_n^\star(t) \to e^{-t^2/2}. \tag{44}$$

Since $\sigma_n = O(\sqrt{n})$, we analyse instead the closely related quantity

$$h_n(t) := [z^n] F(z, e^{it/\sqrt{n}})^{m-n} \quad \text{so that} \quad \frac{h_n(t)}{h_n(0)} = e^{-it\mu_n/\sqrt{n}} \phi^\star \left(\frac{\sigma_n t}{\sqrt{n}} \right). \tag{45}$$

The occurrence of large coefficients of large powers is known in the univariate case to be amenable to the saddle point method [3], and we start by briefly reviewing the case $t = 0$ that corresponds to the "unperturbed" integral,

$$[z^n] F(z,1) = \frac{1}{2i\pi} \int F(z,1)^{m-n} \frac{dz}{z^{n+1}}. \tag{46}$$

By a standard argument, such an integral (46) involving large powers is precisely of the type amenable to saddle point analysis. Here, we have $F(z,1) = f_0(z) = T(z)/z$, and the saddle point equation is

$$\frac{d}{dz} \left((m-n) f_0(z) - n \log z \right) = 0,$$

which has a unique positive root between 0 and e^{-1} at $\zeta = \alpha e^{-\alpha}$. At that point, one has additionally $T(\zeta) = \alpha$ and $f_0(\zeta) = e^\alpha$.

The classical saddle point analysis is based on integration on the circle $|z| = \zeta$ together with the fact that only a small sector of amplitude δ around ζ dictates the asymptotic contribution of the integral in (46). One should take $n\delta^2 \to \infty$ and $n\delta^3 \to 0$, for instance $\delta = n^{-0.4}$ is suitable, a choice that we fix here. Then, a local expansion reduces asymptotically and up to normalization the integral to be evaluated to a complete integral of $e^{-w^2/2}$.

Now, the strategy for evaluating the integral in (45) consists in adopting the same integration contour $|z| = \zeta$ as in the unperturbed case (46). The perturbation introduced in (45) by $q = e^{it/\sqrt{n}}$ must then be quantified precisely. It turns out that concentration in a sector of amplitude $\delta = n^{-0.4}$ still holds as the maximum of the integrand's modulus on the contour only gets displaced by a much smaller amount, namely $O(n^{-0.5})$. Local expansions near the real axis then provide the asymptotic form of $h_n(t)$, from which the Gaussian law eventually results.

First, we establish globally that the geometry of $F(z,q)$ on $|z| = \zeta$ does not differ much from that of $F(z,1)$ when $q = e^{i\theta}$ and θ lies in a suitably restricted interval around 0. The derivatives

$$f_r(z) = \frac{\partial^r}{\partial q^r} F(z,q) \Big|_{q=1},$$

exist as formal power series in z that are furthermore analytic in $|z| < e^{-1}$. Also, since the total displacement parameter on an object of size n is always at most n^2, we have

$$\left(q\frac{\partial}{\partial q}\right)^r F(z,q)\Big|_{q=1} \ll \left(z\frac{\partial}{\partial z}\right)^{2r} F(z,1),$$

where $\ll$ indicates here coefficientwise dominance between power series with nonnegative coefficients. There results that $F(z,e^{i\theta})$ is in fact an infinitely differentiable function of θ for all fixed z inside the disc $|z| < e^{-1}$. (Construct formal derivatives whose analytic existence is guaranteed by the domination property and then recover $F(z,e^{i\theta})$ by repeated integration.) In particular, Taylor's formula with remainder, when aplied to $F(z,e^{i\theta})$, with z treated as a parameter, yields

$$F(z,e^{i\theta}) = f_0(z) + i\theta f_1(z) - \frac{\theta^2}{2}(f_2(z)+f_1(z)) + \frac{1}{3!}\int_0^\theta (\theta-u)^2 \frac{\partial^3}{\partial u^3} F(z,e^{iu})\,du. \tag{47}$$

The last term is $O(\theta^3)$ and this estimate holds uniformly with respect to z, for z in any subdisc of $|z| < e^{-1}$, since, by coefficient dominance again, the third partial derivative is dominated coefficientwise by $(z\frac{d}{dz})^6 f_0(z)$. The *uniform* estimate (47) precisely quantifies the way $F(z,e^{i\theta})$ approximates $F(z,1)$.

Next, along the circle $|z| = \zeta$, the quantity $|F(z,1)|$ has a unique maximum on the real axis at the saddle point $z = \zeta$. Also, $|F(\zeta e^{i\phi},1)|$ is an upward concave function of the argument ϕ in a fixed neighbourhood of $\phi = 0$. By the uniform approximation property (47) and the continuity that it implies, upward ϕ-concavity, that is expressed by a sign condition on second derivatives, must persist for $F(\zeta e^{i\phi},e^{i\theta})$ provided θ stays in a sufficiently small neighbourhood of 0. Also, for values of ϕ outside of the guaranteed concavity interval and again θ suitably small, the approximation relation (47) entails that $|F(\zeta e^{i\phi},e^{i\theta})| < F(\zeta,1) - \epsilon$, for some fixed $\epsilon > 0$.

The preceding discussion thus provides a clear picture of $|F(z,e^{i\theta})|$ on the circle $|z| = \zeta$. When θ, now a parameter, is such that $|\theta|$ remains less than a small fixed nonzero threshold θ_0, the quantity $|F(\zeta e^{i\phi},e^{i\theta})|$ is upward concave near $\phi = 0$ (that is for z near the real axis) while its values at least remain boundedly smaller than the absolute maximum $f_0(\zeta)$, outside the concavity interval.

Take now $\theta = t/\sqrt{n}$, which is needed for estimating $h_n(t)$. The value of t is fixed and n is assumed to be large enough so that the local concavity and majorization properties hold. A local expansion shows that the maximum of $|F(\zeta e^{i\phi},e^{it/\sqrt{n}})|$ occurs at $\phi = \phi_0(n)$, where

$$\phi_0(n) = -c_1\frac{t}{\sqrt{n}}(1+O(n^{-1})), \qquad c_1 = \frac{f_1(\zeta)}{\zeta f_0'(\zeta)}.$$

This is well within the range of the unperturbed saddle point integral which is given by the boundary points $\zeta e^{\pm i\delta}$, where $\delta = n^{-0.4}$. Therefore, we can conclude in the usual way that

$$h_n(t) = \frac{1}{2\pi}\int_{-\delta}^{+\delta} F(\zeta e^{i\phi},e^{it/\sqrt{n}})^{m-n} e^{-ni\phi}\,d\phi\,(1+O(n^{-1/2})), \tag{48}$$

where the error term is in fact exponentially small.

Now, the analysis can be performed in the small interval $[-\delta,+\delta]$ by means of local expansions of the integrand, themselves attainable from the main approximation (47). For estimates up to relative $O(n^{-1/2})$ error terms, it suffices to use the quadratic approximation part of (47), so

that

$$h_n(t) = \frac{1}{2\pi}\int_{-\delta}^{+\delta} f_0(\zeta e^{i\phi})^{m-n} A(\zeta e^{i\phi})^{m-n} e^{-ni\phi}\, d\phi\, (1+O(n^{-1/2})), \tag{49}$$

where

$$A(z) = 1 + i\frac{t}{\sqrt{n}}\frac{f_1(z)}{f_0(z)} - \frac{t^2}{2}\frac{f_2(z)+f_1(z)}{f_0(z)}.$$

From this point on, the computations are routine but particularly tedious, so that we only sketch them. It suffices to expand $(m-n)\log A(\zeta e^{i\phi})$ with respect to ϕ up to quadratic terms again, then set $\phi = w/\sqrt{n}$, and extend the integration bounds to $(-\infty, +\infty)$. The integral is thereby reduced asymptotically to a form

$$\int_{-\infty}^{+\infty} \exp(a_0 + ia_1 w - a_2 w^2/2)\, dw = \sqrt{\frac{2\pi}{a_2}} \exp\left(a_0 - \frac{a_1^2}{2a_2}\right), \tag{50}$$

that is evaluated by completing the square. Once more the support of a computer algebra system like Maple is especially welcome, and one finds (some details omitted),

$$\begin{aligned}
a_0 &= \beta \log\frac{f_0(\zeta)}{\zeta} n + \frac{i\beta t f_1(\zeta)}{f_0(\zeta)} n^{1/2} - \frac{\beta t^2}{2}\left(\frac{f_2(\zeta)+f_1(\zeta)}{f_0(\zeta)} - \frac{f_1(\zeta)^2}{f_0(\zeta)^2}\right) + O(n^{-1/2}) \\
a_1 &= \left(\frac{\beta f_0'(\zeta)}{f_0(\zeta)} - 1\right) n + \frac{it\beta\zeta}{f_0(\zeta)^2}(f_0(\zeta)f_1'(\zeta) - f_0'(\zeta)f_1(\zeta)) n^{1/2} + O(1) \\
a_2 &= \frac{\beta\zeta}{2}\,\frac{\zeta f_0'(\zeta)^2 - \zeta f_0''(\zeta) f_0(\zeta) - f_0(\zeta) f_0'(\zeta)}{f_0(\zeta)^2} n + O(n^{1/2})
\end{aligned} \tag{51}$$

with $\beta = \alpha^{-1} - 1$.

All reductions done (!), we obtain from Equations (50) and (51) the asymptotic estimate

$$\frac{h_n(t)}{h_n(0)} = \exp\left(i\mu_n \frac{t\sigma_n}{\sqrt{n}} - \frac{t^2\sigma_n^2}{2n}\right)(1+O(n^{-1/2})), \tag{52}$$

where use is made of the asymptotic forms of μ_n and σ_n.

We observe in passing (see also the comments below) that the asymptotic form of moments derives systematically from the basic saddle point method and that the expressions can be all obtained directly in terms of f_r and their derivatives evaluated at ζ. For instance,

$$\mu_n = (m-n)\frac{\int f_0(z)^{m-n-1} f_1(z) z^{-n-1}\, dz}{\int f_0(z)^{m-n} z^{-n-1}\, dz} \sim \beta n \frac{f_1(\zeta)}{f_0(\zeta)},$$

and so on.

The final estimate (52) after renormalization according to (45) then yields the convergence of characteristic functions (44). This completes the proof of the Gaussian limit law. □

The saddle point method has been used in a technically different context by Pittel [31] who showed that the size of the largest cluster (hence, also the maximum displacement) in a sparse linear probing table only grows logarithmically, on average and in probability. The process used in the proof of the last theorem is in fact very general and we encapsulate it into a statement.

Corollary 1. *A Gaussian limit law holds for the coefficients of any "large power",*

$$[z^n] G(z,q)^{\beta n}, \quad \beta > 0,$$

provided the following conditions hold:

(C_1) $G(z,q) = \sum_n g_n(q) z^n$ *has nonnegative coefficients and* $\deg g_n(q) = O(n^\kappa)$ *for some integer* κ*;*

(C_2) $G(z,1)$ *is analytic in* $|z| < r$, $G(0,1) \neq 0$, $G_z'(0,1) \neq 0$*;*

(C_3) $G_z'(r,1)/G(r,1) = +\infty$

Proof. (Sketch) Condition (C_1) ensures analyticity of partial derivatives and smooth perturbation; (C_3) ensures existence of the basic saddle point; (C_2) ensures unicity of this saddle point. It can be recognized that these are the only conditions used in the proof of Theorem 6, when one defines abstractly the functions f_r by $\mathrm{U}\,\partial_q^r G$ and the saddle point ζ by the equation $\beta\zeta f_0'(\zeta) - f_0(\zeta) = 0$. □

Given its mild analytic conditions, Corollary 1 applies to a diversity of situations where large random assemblies of labelled or unlabelled combinatorial objects are involved. In the case of linear probing hashing, it immediately implies that the number of clusters of some fixed size p has a distribution that is asymptotically Gaussian with mean and variance that are both $O(n)$.

4. Conclusion

The analysis of sparse tables (Section 3) is a by-product of the treatment of full tables (Section 2) that do constitute the primary combinatorial objects, so that we discuss them in more depth here. The Airy distribution and its companion moment formulæ turn out to be part of a ring of problems treated often independently by a variety of methods and authors. A brief census of "Airy phenomena" in combinatorial applications then reveals five main ranges of problems that we now list.

- (P_1) *Construction cost in linear probing hashing.* This is the context of Section 2 and the analysis applies almost *verbatim* to total displacement in parking sequences.
- (P_2) *Number of inversions in trees.* An inversion in a rooted labelled tree is a pair (i, j) such that i is on the path from the root to j and $i > j$. Exact generating functions have been first found by Mallows and Riordan [28] in the case of "Cayley" trees and other families of trees are considered in [9].
- (Q) *Connectivity in graphs.* A major problem in graphical enumeration and random graph theory [5, 15] is the determination of the number $\gamma(n, k)$ of *connected* graphs with n vertices and k edges. The basic problem was first solved by Wright in a famous series of papers [46, 47, 48]. Wright's solution involves a quadratic recurrent sequence that, after normalization, is the same as that of Section 2, so that the Airy constants make an appearance.
- (R_1) *Area of excursions.* By an excursion is meant a random walk that is never negative, and has initial and final altitudes both equal to 0; area is defined as the sum of altitudes of all nodes. The simplest type is the Bernoulli excursion defined by ± 1 steps (also called gambler's ruin sequence); Louchard [26, 27] established that the area of the Bernoulli excursion is asymptotically Airy distributed. Louchard's results are also related to other contemporary works from the early 1980's dealing with Brownian motion [34, 37] where Airy functions also crop up explicitly.
- (R_2) *Path length in trees.* The path length of a tree is the sum of the distances of all nodes to the root of the tree. In a series of papers, Takács [41, 43] has established limit Airy distribution results for various families of trees including Cayley trees and Catalan trees as special cases, while rederiving independently in [40, 42] some of the results of Louchard.

Regarding methods, our Theorem 3 establishes directly the Airy law for (P_1) by a recursive determination of moments. A similar process has been employed by Louchard and Takacs for (R_1) and (R_2). The underlying combinatorial decompositions are however quite different. For (P_1), one may regard the Airy law —via the quadratic recurrence or the Riccati equation— as a reflection of the basic binary tree-like decomposition of linear probing tables, while other decompositions prevail for (R_1) and (R_2). Problem (P_2), as far as we know, has not been

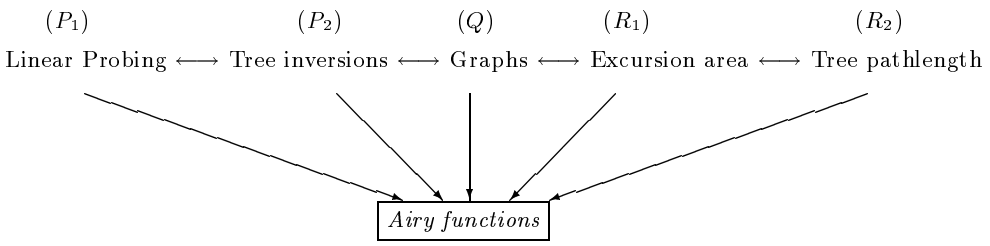


FIGURE 2. Five problems resorting to the "Airy phenomenon": (P_1, P_2, R_1, R_2) lead to the Airy distribution while (Q) involves the Airy coefficients.

previously considered under the angle of asymptotics and limit distributions, but it appears eventually to be an essential element in the combinatorial picture (see below).

The enumeration of connected graphs (Q) has a different status, since combinatorial enumerations rather than limit distributions are involved. Wright's major result states that

$$\gamma(n, n+k) \sim \frac{\sqrt{\pi} 2^{(1-3k)/2} \sigma_k}{\Gamma((3k/2)+1)}\, n^{n+(3k-1)/2},$$

for a family of constants σ_k. A direct comparison between our basic recurrence of (29) and [46] shows that

$$\sigma_k = \frac{1}{2(k+1)!}\, \Omega_{k+1},$$

where the Ω_r are the Airy coefficients of Section 2.

Thus, three limit distribution problems, namely (P_1, R_1, R_2), have been found previously to lead to Airy laws, while structural constants of the Airy type are seen to appear in the graph connectivity problem (Q). This suggests a closer look at combinatorial relations between these problems.

- $(P_1 - P_2)$**:** The fundamental recurrence of Section 2 in relation to (P_1) is indeed *identical* to the recurrence of [28] that models (P_2). Thus, the two problems must be combinatorially isomorphic. This fact has been noted by Knuth [22], following Kreweras [25]. Knuth also evokes in [22] the alternative of a direct and exact combinatorial correspondence based on [20, Ex. 6.4–31]. An immediate consequence of the present work in conjunction with the observations of Kreweras and Knuth is then: *the number of inversions in a random Cayley trees is asymptotically Airy distributed.*
- $(P_2 - Q)$**:** An *exact* correspondence between inversions in trees and connectivity in graphs seems to have been first detected around the turn of the 1980's by several authors. Gessel and Wang [10] have an especially elegant formulation in terms of *depth* first search, where this mode of graph traversal leads precisely to a tree augmented by return edges that form inversions.
- $(Q - R_1)$**:** This thread is due to Spencer [39] who noted that *breadth*–first search of a random connected graph has a "trace" that is a random excursion of the so-called Poisson type. (Depth–first search in the style of [10] is an alternative for this correspondence.) Under Spencer's correspondence, area under the Poisson excursion relates in *exact* terms to "excess" of the original graph.

($R_1 - R_2$)**:** There are many known similarities between random walks and random trees. One of the most classical combinatorial correspondences relates bijectively Bernoulli excursions and general Catalan trees (see, *e.g.*, [18, 36]). Under this correspondence, area of an excursion transforms into path length of the associated tree.

These various relations[2] in a way "explain" the common occurrence of the Airy distribution in (P_1, P_2, R_1, R_2) as well as the rôle of the Airy coefficients in problem (Q). They also point to an alternative and more combinatorial deduction of the Airy law based on the following steps: (i) the exact equivalence between linear probing to inversions in trees by $(P_1 - P_2)$; (ii) the exact equivalence between tree inversions and graph connectivity $(P_2 - Q)$ by depth first search; (iii) the exact reduction to Poisson walks $(Q - R_1)$ by Spencer's principle; (iv) the reduction to Louchard's derivation of the Airy law through an appeal to universality of Brownian motion. Our derivation of Section 2 offers instead a self-contained approach to the problem.

Postscript. Apart from this conclusion section, the technical developments of our paper are otherwise independent of the recent preprint of Knuth [22]. Knuth has been kind enough to share numerous informations regarding [22], and this sheds additional light on the structure of the generating functions that appear in Sections 2 and 3. A major consequence of [22] is that the fundamental difference-differential equation admits of a closed form solution. This can be checked by direct comparison with [28] or [22]. (Alternatively, the combinatorial correspondences mentioned above could be used.) As a result, the bivariate generating function $F(z, q)$ happens to have an explicit expression,

$$
\begin{aligned}
F(z,q) &= (q-1)\frac{\partial}{\partial z}\log\left(1+\sum_{n=1}^{\infty} q^{n(n-1)/2}\frac{z^n(q-1)^{-n}}{n!}\right)\\
&= \frac{\displaystyle\sum_{n=0}^{\infty} q^{n(n+1)/2}\frac{z^n(q-1)^{-n}}{n!}}{\displaystyle\sum_{n=0}^{\infty} q^{n(n-1)/2}\frac{z^n(q-1)^{-n}}{n!}}
\end{aligned}
\tag{53}
$$

These forms are recognizable variants of the bivariate generating functions of graphs,

$$F(z, q+1) = \sum_{n,t} \gamma(n, n+t-1) q^t \frac{z^{n-1}}{(n-1)!},$$

and Knuth's analysis precisely starts from this relation.

Notice that (53) does not trivialize our moment computations, since it is already far from clear, given (53), that the tree functions should be involved. The manipulation of q-series expansions like (53) is in fact particularly delicate, as attested by the "Giant component" paper [15] on which [22] relies.

The exact correspondence with graph connectivity is at any rate neatly exposed by the equation (53). Unpublished work by Flajolet and Salvy (1995) inspired by Prellberg [33] indicates that moments can in fact be extracted from such a q-series expansion by a method of coalescent saddle points [45] that is well known to lead to Airy functions. We thus find yet another "reason" for the Airy phenomena observed here: a law of the Airy type may be expected whenever there occurs a coalescence of two neighbouring saddle points (a so-called "monkey saddle") in a Cauchy coefficient integral.

[2]Exchanges with Joel Spencer in 1995, while [39] was being developed, have been at the origin of our interest in Airy phenomena. Private communications with Knuth have then greatly helped us to complete the picture offered in this Section. See [22] for an insightful perspective.

Acknowledgements. The authors are thankful to Don Knuth, to whom this paper is dedicated, for his constant support and his openness in sharing his thoughts on the subject.

References

[1] Berndt, B. C. *Ramanujan's Notebooks, Part II*. Springer Verlag, 1989.

[2] Broder, A. Two counting problems solved via string encodings. In *Combinatorial Algorithms on Words*, A. Apostolico and Z. Galil, Eds., vol. 12 of NATO Advance Science Institute Series. Series F: Computer and System Sciences. Springer Verlag, 1985, pp. 229–240.

[3] De Bruijn, N. G. *Asymptotic Methods in Analysis*. Dover, 1981. A reprint of the third North Holland edition, 1970 (first edition, 1958).

[4] Flajolet, P., Grabner, P., Kirschenhofer, P., and Prodinger, H. On Ramanujan's Q-function. *Journal of Computational and Applied Mathematics 58*, 1 (1995), 103–116.

[5] Flajolet, P., Knuth, D. E., and Pittel, B. The first cycles in an evolving graph. *Discrete Mathematics 75* (1989), 167–215.

[6] Flajolet, P., and Odlyzko, A. M. Random mapping statistics. In *Advances in Cryptology* (1990), J.-J. Quisquater and J. Vandewalle, Eds., vol. 434 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 329–354. Proceedings of Eurocrypt'89, Houtalen, Belgium, April 1989.

[7] Flajolet, P., and Odlyzko, A. M. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics 3*, 2 (1990), 216–240.

[8] Flajolet, P., and Sedgewick, R. Analytic combinatorics. Book in preparation, 1998. (Individual chapters are available as INRIA Research Reports 1888, 2026, 2376, 2956, 3162.).

[9] Gessel, I., Sagan, B. E., and Yeh, Y.-N. Enumeration of trees by inversions. *Journal of Graph Theory 19*, 4 (1995), 435–459.

[10] Gessel, I., and Wang, D. L. Depth-first search as a combinatorial correspondence. *Journal of Combinatorial Theory*, Series A *26*, 3 (1979), 308–313.

[11] Gonnet, G. H., and Baeza-Yates, R. *Handbook of Algorithms and Data Structures: in Pascal and C*, second ed. Addison–Wesley, 1991.

[12] Goulden, I. P., and Jackson, D. M. *Combinatorial Enumeration*. John Wiley, New York, 1983.

[13] Greene, D. H., and Knuth, D. E. *Mathematics for the analysis of algorithms*, second ed. Birkhauser, Boston, 1982.

[14] Hennequin, P. Combinatorial analysis of quicksort algorithm. *RAIRO Theoretical Informatics and Applications 23*, 3 (1989), 317–333.

[15] Janson, S., Knuth, D. E., Łuczak, T., and Pittel, B. The birth of the giant component. *Random Structures and Algorithms 4*, 3 (1993), 233–358.

[16] Kirschenhofer, P., Prodinger, H., and Tichy, R. Über einige Funktionaldifferentialgleichingen aus des Analyse von Algorithmen. In *Zahlentheoretische Analysis II* (1987), E. Hlawka, Ed., no. 1262 in Lecture Notes in Mathematics, pp. 111–123.

[17] Knuth, D. E. Notes on "open" addressing. Unpublished memorandum, 1963. (Memo dated July 22, 1963. With annotation "*My first analysis of an algorithm, originally done during Summer 1962 in Madison*". Also conjectures the asymptotics of the Q-function, with annotation "*Proved May 24, 1965*".).

[18] Knuth, D. E. *The Art of Computer Programming*, vol. 1 Fundamental Algorithms. Addison-Wesley Publishing Company, 1968.

[19] Knuth, D. E. *The Art of Computer Programming*, vol. 2 Seminumerical Algorithms. Addison-Wesley Publishing Company, 1969.

[20] Knuth, D. E. *The Art of Computer Programming*, vol. 3 Sorting and Searching. Addison-Wesley Publishing Company, 1973.

[21] Knuth, D. E. Analysis of optimum caching. *Journal of Algorithms 6* (1985), 181–199.

[22] Knuth, D. E. Linear probing and graphs. Preprint, July 1997.

[23] Knuth, D. E., and Rao, G. S. Activity in an interleaved memory. *IEEE Transactions on Computers C-24* (1975), 943–944.

[24] Knuth, D. E., and Schönhage, A. The expected linearity of a simple equivalence algorithm. *Theoretical Computer Science 6* (1978), 281–315.

[25] Kreweras, G. Une famille de polynômes ayant plusieurs propriétés énumératives. *Periodica Mathematica Hungarica 11* (1980), 309–320.

[26] Louchard, G. The Brownian excursion: a numerical analysis. *Computers and Mathematics with Applications 10*, 6 (1984), 413–417.

[27] Louchard, G. Kac's formula, Lévy's local time and Brownian excursion. *Journal of Applied Probability 21* (1984), 479–499.
[28] Mallows, C. L., and Riordan, J. The inversion enumerator for labeled trees. *Bulletin of the American Mathematical Society 1968* (74), 92–94.
[29] Moon, J. W. Counting labelled trees. In *Canadian Mathematical Monographs*, vol. 1. Canadian Mathematical Congress, 1970.
[30] Odlyzko, A. M. Asymptotic enumeration methods. In *Handbook of Combinatorics*, M. G. R. Graham and L. Lovász, Eds., vol. II. Elsevier, Amsterdam, 1995, pp. 1063–1229.
[31] Pittel, B. Linear probing: The probable largest search time grows logarithmically with the number of records. *Journal of Algorithms 8* (1987), 236–249.
[32] Poblete, P. Approximating functions by their Poisson transform. *Information Processing Letters 23* (1986), 127–130.
[33] Prellberg, T. Uniform q-series asymptotics for staircase polygons. *Journal of Physics A: Math. Gen. 28* (1995), 1289–1304.
[34] Rice, S. O. The integral of the absolute value of the pinned Wiener process—calculation of its probability density by numerical integration. *Annals of Probability 10* (1982), 240–243.
[35] Sedgewick, R. *Algorithms*, second ed. Addison–Wesley, Reading, Mass., 1988.
[36] Sedgewick, R., and Flajolet, P. *An Introduction to the Analysis of Algorithms*. Addison-Wesley Publishing Company, 1996.
[37] Shepp, L. A. On the integral of the absolute value of the pinned Wiener process. *Annals of Probability 10* (1982), 234–239.
[38] Sloane, N. J. A., and Plouffe, S. *The Encyclopedia of Integer Sequences*. Academic Press, 1995.
[39] Spencer, J. Enumerating graphs and Brownian motion. *Communications on Pure and Applied Math. 50* (1997), 293–296.
[40] Takács, L. A Bernoulli excursion and its various applications. *Advances in Applied Probability 23* (1991), 557–585.
[41] Takács, L. Conditional limit theorems for branching processes. *Journal of Applied Mathematics and Stochastic Analysis 4*, 4 (1991), 263–292.
[42] Takács, L. On a probability problem connected with railway traffic. *Journal of Applied Mathematics and Stochastic Analysis 4*, 1 (1991), 1–27.
[43] Takács, L. The asymptotic distribution of the total heights of random rooted trees. *Acta Scientifica Mathematica (Szeged) 57* (1993), 613–625.
[44] Wilf, H. S. *generatingfunctionology*. Academic Press, 1994.
[45] Wong, R. *Asymptotic Approximations of Integrals*. Academic Press, 1989.
[46] Wright, E. M. The number of connected sparsely edged graphs. *Journal of Graph Theory 1* (1977), 317–330.
[47] Wright, E. M. The number of connected sparsely edged graphs. II. Smooth graphs. *Journal of Graph Theory 2* (1978), 299–305.
[48] Wright, E. M. The number of connected sparsely edged graphs. III. Asymptotic results. *Journal of Graph Theory 4* (1980), 393–407.

P.F.: Algorithms Project, INRIA, Rocquencourt, 78150 Le Chesnay (France).
E-mail address: Philippe.Flajolet@inria.fr.

P.P.: Department of Computer Science, University of Chile, Casilla 2777, Santiago, Chile.
E-mail address: ppoblete@dcc.uchile.cl.

A.V.: Pedeciba Informatica, Casilla de Correo 16120, Distrito 6, Montevideo, Uruguay.
E-mail address: viola@fing.edu.uy

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105,
78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS
Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
(France)
http://www.inria.fr
ISSN 0249-6399

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Analytic Combinatorics of Non-crossing Configurations

Philippe Flajolet, Marc Noy

N 3196
Juin 1997

THÈME 2

ISSN 0249-6399

Analytic Combinatorics of Non-crossing Configurations

Philippe Flajolet, Marc Noy

Thème 2 — Génie logiciel
et calcul symbolique
Projet Algo

Rapport de recherche— Juin 1997 — 24 pages

Abstract: This paper describes a systematic approach to the enumeration of "non-crossing" geometric configurations built on vertices of a convex n-gon in the plane. It relies on generating functions, symbolic methods, singularity analysis, and singularity perturbation. A consequence is exact and asymptotic counting results for trees, forests, graphs, connected graphs, dissections, and partitions. Limit laws of the Gaussian type are also established in this framework; they concern a variety of parameters like number of leaves in trees, number of components or edges in graphs, etc.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33) 01 39 63 55 11 – Télécopie : (33) 01 39 63 53 30

Combinatoire analytique des configurations sans croisement

Résumé : Cet article décrit une approche systématique au dénombrement de configurations géométriques "sans croisements" construites sur les sommets d'un n-gone convexe plan. L'approche repose sur les fonctions génératrices, les méthodes symboliques, l'analyse de singularités et la perturbation de singularités. On en déduit des résultats tant exacts qu'asymptotiques pour arbres, forêts, graphes connexes et généraux, dissections et partitions. Des lois limites de formes gaussienne résultent également de cette méthode; elles concernent le nombre de feuilles dans les arbres, le nombre de composantes ou d'arêtes dans les graphes, etc.

Analytic Combinatorics of Non-crossing Configurations

Philippe Flajolet
Algorithms Project, INRIA Rocquencourt, F-78153 Le Chesnay (France)

Marc Noy
Dept. Applied Mathematics, Universitat Politècnica de Catalunya,
Pau Gargallo, 5, 08028 Barcelona (Spain)

June 19, 1997

Abstract

This paper describes a systematic approach to the enumeration of "non-crossing" geometric configurations built on vertices of a convex n-gon in the plane. It relies on generating functions, symbolic methods, singularity analysis, and singularity perturbation. A consequence is exact and asymptotic counting results for trees, forests, graphs, connected graphs, dissections, and partitions. Limit laws of the Gaussian type are also established in this framework; they concern a variety of parameters like number of leaves in trees, number of components or edges in graphs, etc.

Introduction

The enumeration of planar configurations defined on vertices of a convex n-gon has a long and dignified history. In 1753, Euler and Segner counted triangulations —the well-known answer involves the Catalan numbers— and on this occasion Euler invented combinatorial generating functions. Since then, many other configurations have been enumerated: see for instance Comtet's book [6], for an account of known results. The interest for such configurations comes first and foremost from the combinatorics of classical structures [6], but also from computational geometry, and even the interpretation of perturbative expansions in statistical physics [7].

The purpose of this paper is to re-examine these problems in the light of recent general methods of *analytic combinatorics* [14, 28]. First thanks to symbolic methods developed by various schools [4, 14, 15, 18, 28, 29, 32], there is a systematic and purely formal correspondence between combinatorial constructions and *generating functions*. In this way, specifications of combinatorial structures can be translated automatically into generating function equations. This approach is, as we propose to show, especially effective here, since planarity entails neat decompositions for the planar configurations to be enumerated. Second, analytic methods based on the analysis of singularities [13] give a transparent access to asymptotic counts that plainly appear as morphic images of the local expansions of generating functions near a singularity.

This programme is carried out here on six of the most basic planar "non-crossing" configurations: trees and forests, graphs and connected graphs, dissections and partitions. The generating functions involved are all *algebraic functions*, a property to be somewhat expected

given the context-free character of these objects. However, their forms are sometimes more complicated than what is encountered in the Catalan domain comprehensively reviewed by Gould in [17]. *Singularity analysis* then makes it possible to derive precise estimates; see especially our Theorem 4. In addition, a general approach of "singularity perturbation asymptotics" [12] permits us to refine the counting estimates and derive *limit laws* for many parameters of interest.

Given the vast literature on the subject, we cannot expect to derive only new results; our hope is that the unified treatment presented here could be of methodological interest and that the present paper could also serve as a partial survey of the the enumerative, asymptotic, and probabilistic aspects of non-crossing configurations. The analytic approach followed here, when contrasted to more classical combinatorial bijective proofs, proves especially effective when exact formulæ either become too intricate or fade away.

In the first sections of this paper, numbered 1,2,3, we make explicit the basic decompositions of the six fundamental types of planar configurations considered. We characterize in each case the counting generating functions by the minimal polynomial equation they satisfy, which serves two goals: in some cases, this leads to explicit counting results; in all cases, the equations can be fed into the asymptotic machinery of Section 4, leading eventually to the precise asymptotic estimates of Theorem 4. In addition, many parameters of interest are easily taken into account by bivariate generating functions, the corresponding equations serving as input to the bivariate asymptotic process of Section 5. A consequence, stated in Theorem 5, is that all the parameters discussed, e.g., the number of edges or components in non-crossing graphs of a fixed size, have distributions that are Gaussian in the asymptotic limit.

Combinatorial preliminaries. Let $P_n = \{v_1, v_2, \ldots, v_n\}$ be a fixed set of points, conventionally ordered counter-clockwise, that are vertices of a convex polygon, for instance, the vertices of a regular n-gon. Define a *non-crossing graph* as a graph with vertex set P_n whose edges are straight line segments that do not cross. Several classical combinatorial objects can be viewed as non-crossing graphs (we omit the qualifier non-crossing from now on). For instance, triangulations of a convex polygon are graphs with the maximum number of edges; dissections of a convex polygon are graphs containing the edges $v_1v_2, v_2v_3, \ldots, v_nv_1$; non-crossing partitions are graphs whose components are points, edges or cycles.

We recall that a graph is connected if any two vertices can be joined by a path. A tree is a connected acyclic graph and the number of edges in a tree is one less than the number of vertices. A forest is an acyclic graph, or a graph whose components are trees.

Let $\mathcal{A}$ be a class of combinatorial objects and let $|a|$ be the size of an object $a \in \mathcal{A}$. If $\mathcal{A}_n$ denotes the objects in $\mathcal{A}$ of size n and $a_n = |\mathcal{A}_n|$, then the (ordinary) *generating function*, GF for short, of the class $\mathcal{A}$ is

$$A(z) = \sum_{a \in \mathcal{A}} z^{|a|} = \sum_{n \geq 0} a_n z^n.$$

Here, the size of a graph is its number of vertices and we consider various classes of non-crossing graphs.

There is a direct correspondence between set-theoretic operations (or "constructions") on combinatorial classes and algebraic operations on GF. For an exposition of the symbolic enumeration method, see for instance [14, 28]. Table 1 summarizes this correspondence for the operations that are used in the paper. There "union" means union of disjoint copies, "product" is the usual cartesian product, "sequence" forms sequences, and "substitution" $\mathcal{A} = \mathcal{B} \circ \mathcal{C}$ corresponds to grafting objects of $\mathcal{C}$ on nodes of $\mathcal{B}$.

Enumerations according to size and an auxiliary parameter χ are described by bivariate

Construction		*Operation on GF*
Union	$\mathcal{A} = \mathcal{B} \cup \mathcal{C}$	$A(z) = B(z) + C(z)$
Product	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$A(z) = B(z)C(z)$
Sequence	$\mathcal{A} = \mathtt{Seq}(\mathcal{B})$	$A(z) = 1/(1 - B(z))$
Substitution	$\mathcal{A} = \mathcal{B} \circ \mathcal{C}$	$A(z) = B(C(z))$

Table 1: The basic combinatorial constructions and their translation into generating functions.

generating functions, or BGFs,

$$A(z, w) = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|} w^{\chi[\alpha]} = \sum_{n,k \geq 0} A_{n,k} z^n w^k,$$

with $A_{n,k}$ the number of objects of size n with χ-parameter equal to k. Throughout the paper the variable z is reserved for marking vertices of the different kinds of graphs, and the variable w for marking a secondary parameter, like leaves in trees or edges in graphs. Classes and their GFs are consistently denoted by the same letters.

We will need repeatedly the Lagrange-Bürmann inversion theorem in order to extract coefficients of GF that satisfy functional equations of the implicit type [6, 18, 28, 32]:

> **Lagrange inversion.** *Let $\phi(u)$ be a formal power series with $\phi_0 \neq 0$, and let $Y(z)$ be the unique formal power series solution of the equation $Y = z\phi(Y)$. Then the coefficient of $\psi(Y)$, for an arbitrary series ψ, is given by*
>
> $$[z^n]\psi(Y(z)) = \frac{1}{n}[u^{n-1}]\phi(u)^n \psi'(u).$$
>
> *In particular, for every $k > 0$ we have*
>
> $$[z^n]Y(z)^k = \frac{k}{n}[u^{n-k}]\phi(u)^n.$$

Lagrange inversion obviously applies to bivariate generating functions upon treating the auxiliary variable as a parameter.

1 Trees and forests

In this section a tree means a non-crossing tree, and a forest is a non-crossing forest. Basic decompositions reflect the geometric structure of trees and forests (Fig. 1 and 2), which leads to algebraic generating functions that prove to be amenable to Lagrange expansion.

Theorem 1 (*i*) *The number of non-crossing trees with n vertices equals*

$$T_n = \frac{1}{2n-1}\binom{3n-3}{n-1},$$

and the number of non-crossing trees with n vertices and k leaves is equal to

$$T_{n,k} = \frac{1}{n-1}\binom{n-1}{k}\sum_{j=0}^{k-1}\binom{n-1}{j}\binom{n-k-1}{k-1-j}2^{n-2k+j}.$$

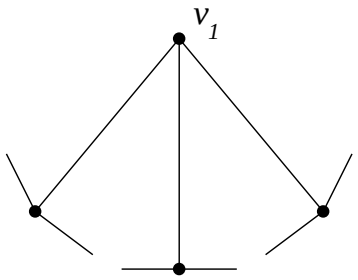


Figure 1: Butterflies pending from vertex v_1.

(ii) The number of trees with degree partition $(n_0, n_1, \ldots, n_r)$*, where* $\sum n_i = n$ *and* $\sum i n_i = n-1$*, is equal to*

$$\frac{1}{n(n-1)}\binom{n}{n_0, n_1 \ldots, n_r} 1^{n_0} 2^{n_1} \cdots (r+1)^{n_r} \sum_{i=1}^{r} \frac{i}{i+1} n_i.$$

(iii) The number of forests of size n *is*

$$F_n = \sum_{j=1}^{n} \frac{1}{2n-j}\binom{n}{j-1}\binom{3n-2j-1}{2n-j-1}, \tag{1}$$

and the number of forests with n *nodes and* k *components is*

$$F_{n,k} = \frac{1}{2n-k}\binom{n}{k-1}\binom{3n-2k-1}{2n-k-1}. \tag{2}$$

(iv) The GF of forests, the BGF of trees and leaves, and the BGF of forests and components, are algebraic functions given by (10), (6) and (11).

Trees were first enumerated by Dulucq and Penaud [9], and their result is summarized in part (i) of the theorem; the enumeration of forests by GF in (10) below is due to Noy [25]. We recover both results, as well as several new ones in the form of multivariate extensions. In particular, the counting of trees according to the number of leaves as stated in (i) solves a problem that was left open in [25]. The explicit forms for the number of forests in part (iii), formulæ (1) and (2), provide explicit expansions for the GF computations of [25].

Trees. We use the following basic decomposition for counting trees. Let d be the degree of v_1 in a tree τ. Then τ can be viewed as a sequence attached to v_1 of d ordered pairs of trees sharing a common vertex. This motivates the following definition: a *butterfly* is an ordered pair of trees with a common vertex. The name aims to convey the idea that the pair of trees looks like the two wings of a butterfly. If v_1 has degree d, then τ can be identified with a sequence of d butterflies pending from v_1 (see Figure 1).

Hence we have the following equations, where $T(z)$ is the GF for trees and $B(z)$ is the GF for butterflies:

$$\begin{aligned} T &= \frac{z}{1-B}; \\ B &= T^2/z. \end{aligned} \tag{3}$$

The division by z in the second equation is because we identify two root vertices to form a butterfly. From this it follows that T satisfies

$$T^3 - zT + z^2 = 0. \tag{4}$$

If we set $z = \zeta^2, T = \zeta U$, the equation becomes $U - U^3 = \zeta$, a direct case of application of the Lagrange inversion theorem. As a consequence, we get the first assertion in part (i) of the theorem. An alternative transformation that is useful for the sequel is as follows. Set $T = z + zy$, and "solve" for z in terms of y; this gives

$$z = \frac{y}{(1+y)^3}, \qquad y = z(1+y)^3, \tag{5}$$

which is amenable to Lagrange inversion. These derivations also show that T_{n+1} is the number of ternary trees drawn in the plane (without reference to a fixed convex polygon) that have n internal nodes.

In this paper we consider non-crossing trees as being rooted at vertex v_1. The degree of a vertex in a tree is then its out-degree, and leaves are vertices of degree zero. Let $T(z, w)$ be a bivariate generating function, where z marks vertices as before, and w marks leaves. Then we have

$$\begin{aligned} T(z,w) &= \frac{z}{1-B}; \\ B(z,w) &= T^2/z - z + zw. \end{aligned}$$

The first equation is the same as (3), since the number of leaves in τ is just the sum of the number of leaves in the sequence of butterflies defining τ. The second equation is because when the two wings of a butterfly are empty we have a leave. Hence the term z in $B(z, w)$ has to be replaced with zw. Eliminating B we obtain

$$T^3 + (z^2w - z^2 - z)T + z^2 = 0. \tag{6}$$

Expansion of $T(z, w)$ can be carried out by the same process as in (5). Set $T = z + zy$, and solve for z, which gives

$$z = \frac{y}{(y+1)(y^2+2y+w)}, \qquad y = z(y+1)(y^2+2y+w).$$

Then, by Lagrange, one has

$$[z^n]T(z,w) = [z^{n-1}]y = \frac{1}{n-1}[u^{n-2}]\left((u+1)(u^2+2u+w)\right)^{n-1},$$

and upon extracting $[w^k]$,

$$\begin{aligned} [z^n w^k]T(z,w) &= \frac{1}{n-1}[u^{n-2}w^k]\left((u+1)(u^2+2u+w)\right)^{n-1} \\ &= \frac{1}{n-1}\binom{n-1}{k}[u^{n-2}](u+1)^{n-1}(u^2+2u)^{n-1-k}. \end{aligned}$$

This last form yields directly the expression of $T_{n,k}$ stated in part (i) of the theorem.

Given a tree τ of size n and maximum degree r, the (degree) *partition* $p(\tau)$ is the sequence $(n_0, n_1, \ldots, n_r)$, where n_i is the number of vertices of degree i in τ, for $i = 0, \ldots, r$. Clearly

$\sum n_i = n$ and, since the number of edges is $n-1$, $\sum i n_i = n-1$. Given a sequence of non-negative integers $(n_0, n_1, \ldots, n_r)$ with $\sum n_i = n$ and $\sum i n_i = n-1$, we consider the problem of determining the number of trees of size n having partition $(n_0, n_1, \ldots, n_r)$.

To solve this problem we have to look again at butterflies. A butterfly β has a left and a right tree with a common vertex v. If d is the degree of v, then β can be seen in turn as a sequence of d butterflies attached to v_1. There are $d+1$ ways of distributing them among the left and right trees, hence we have $B = z(1+2B+3B^2+\cdots)$. Let now $u_0, u_1, \ldots$ be a sequence of variables, where u_i marks a vertex of degree i, either in trees or in butterflies. Then the equation becomes

$$B = z(u_0 + 2u_1 B + 3u_2 B^2 + \cdots + (r+1)u_r B^r + \cdots), \tag{7}$$

where $B = B(z, u_0, u_1, \ldots)$ is a GF in an infinite number of variables. On the other hand, the basic equation (3) becomes

$$T = z(u_0 + u_1 B + u_2 B^2 + \cdots + u_r B^r + \cdots). \tag{8}$$

Using Lagrange inversion in (7) we find that

$$[u_0^{n_0} u_1^{n_1} \cdots u_r^{n_r} z^n](z u_k B^k) = \frac{k}{n-1} \binom{n-1}{n_0, \ldots, n_k - 1, \ldots, n_k} 1^{n_0} 2^{n_1} \cdots (k+1)^{n_k - 1} \cdots (r+1)^{n_r}.$$

Now we use (8) to express the coefficient of $[u_0^{n_0} u_1^{n_1} \cdots u_r^{n_r} z^n]$ in T as the sum of the above expression for $k = 1, \ldots, r$. A straightforward manipulation gives the final compact solution stated in part (ii) of the theorem.

Forests. A forest is an acyclic graph, i.e., a graph whose connected components are trees. Let ϕ be a forest and let r be the number of vertices in the component τ containing v_1 (see Figure 2). Then ϕ has to be completed with r additional forests (some of them possibly empty), one to the right of every vertex of τ. Thus the class of forests is obtained from the class of trees by substituting a vertex by a pair (vertex, forest). Let F be the GF of forests, then

$$F = 1 + T(zF), \tag{9}$$

where T is the GF of trees as before, and 1 is the GF of the empty forest of size 0. Since T satisfies (4) one can eliminate T and recover a result from [25] (the equation here is marginally different since we are taking the constant term of F to be 1):

$$F^3 + (z^2 - z - 3)F^2 + (z+3)F - 1 = 0. \tag{10}$$

In order to expand, we set $F = 1 + y$, then "solve" for z, which yields,

$$y = z(1+y)\left(\frac{1 - \sqrt{1-4y}}{2y}\right),$$

an equation of the Lagrange type that also suggests a Catalan tree decomposition for non-crossing forests. Formula (1) then results from the Lagrange expansion of powers of the Catalan GF.

Let now $F(z, w)$ be the bivariate GF for forests, where w marks components. We only have to add a factor w in (9) to take into account the component of v_1 that was singled out, to obtain $F(z, w) = 1 + wT(zF)$. Eliminating T as before we get

$$F^3 + (w^3 z^2 - w^2 z - 3)F^2 + (w^2 z + 3)F - 1 = 0. \tag{11}$$

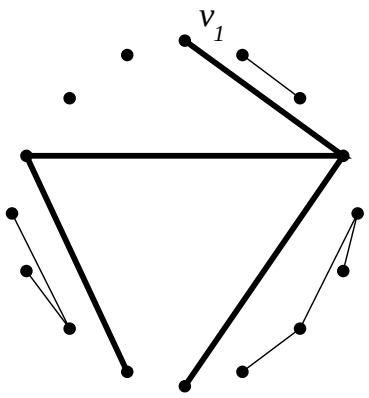


Figure 2: A forest.

This equation also admits a Lagrange form, upon setting $F = 1 + wy$,

$$y = z(1+wy)\left(\frac{1-\sqrt{1-4y}}{2y}\right),$$

hence again the explicit formula for $F_{n,k}$ in part (*iii*). We remark that counting counting edges instead of components is an equivalent problem, since the number of edges in a forest is equal to the number of vertices minus the number of components.

2 Connected graphs and general graphs

Like before, a graph means a non-crossing graph. Planarity once more entails strong decomposition properties (Fig. 3) reflected by algebraic generating functions and Lagrange expansions.

Theorem 2 (*i*) *The number of connected graphs of size n is given by*

$$C_n = \frac{1}{n-1}\sum_{j=0}^{n-2}\binom{n+j}{n}\binom{2n-4-j}{n-2}2^{n-2-j}.$$

The number of connected graphs of size n with k edges is given by

$$C_{n,k} = \frac{1}{n-1}\sum_{j=0}^{n-2}\binom{n+j}{n}\binom{2n-4-j}{n-2}\binom{n-2-j}{k-n+1}.$$

(*ii*) *The number of graphs of size $n \geq 3$ is expressible in terms of Schröder numbers,*

$$G_n = 2^n c_{n-1}, \qquad c_n := \sum_{0\leq\nu\leq(n/2)}(-1)^\nu \frac{1\cdot 3\cdots(2n-2\nu-3)}{\nu!\,(n-2\nu)!}3^{n-2\nu}2^{-\nu-2}, \tag{12}$$

the number of graphs of size n with k edges is

$$G_{n,k} = \frac{1}{n-1}\sum_{j=0}^{n-2}\binom{n-1}{k-j}\binom{n-1}{j+1}\binom{n-2+j}{n-2}, \tag{13}$$

and the number of graphs of size n with k connected components is

$$\widehat{G}_{n,k} = \frac{1}{n}\binom{n}{k-1}\sum_{j=0}^{n-k}\binom{n+j-1}{j}\binom{2n-2k-j}{n-k}\frac{j2^{n-k-j}}{2n-2k-j}. \tag{14}$$

(*iii*) *The BGFs of connected graphs and the BGF of graphs counted according to edges are algebraic functions given by (18) and (22). The BGF of graphs and number of connected components is an algebraic function given by (23).*

The univariate generating functions of connected graphs and general graphs were obtained by Domb and Barrett [7] after considerable effort. In both cases, these authors also obtained the bivariate GF according to the number of edges, building upon the work of the Rev. T. P. Kirkman in 1857; see [7] for a thorough historical discussion. We recover all the results of [7] plus two new ones, namely the enumeration of graphs according to the number of components by GF (part (*iii*)) and an explicit formula for the number of connected graphs (part (*i*)). The result concerning $G_{n,k}$ is roughly equivalent to Kirkman's results in view of Eq. (9–10) of [7], while the one concerning $\widehat{G}_{n,k}$ seems to be new. Our approach in this problem is a direct adaptation of the scheme we used for counting trees and forests, and as such it is purely "algebraic"; in contrast, in [7], recourse had to be made to a combination of algebraic and differential arguments. The Schröder numbers[1] c_n count generalized bracketings (equivalently, plane trees with n leaves and internal nodes of degree ≥ 2), and they are defined in [6, p. 57].

Connected graphs. We use a decomposition technique analogous to that for counting trees. Let d be the degree of vertex v_1 in a connected graph Γ, and let v_i and v_j be two consecutive neighbours of v_1 in Γ. Then the subgraph induced on the vertex set $\{v_i, v_{i+1}, \ldots, v_j\}$ is either a connected graph (not reduced to a point), or two disjoint connected graphs containing v_i and v_j, respectively. The two possibilities are exemplified in Figure 3. If we let C be the GF for connected graphs, the first possibility is counted by $C - z$, and the second one by C^2. If v_i is the first neighbour of v_1 then one has a connected graph on $\{v_2, \ldots, v_i\}$, whereas if v_j is the last neighbour one has a connected graph on $\{v_{j+1}, \ldots, v_n\}$. Taking into account that the d neighbours of v_1 are counted twice, we obtain

$$\begin{aligned} C &= z + z\frac{C^2}{z} + z\frac{C^2(C-z+C^2)}{z^2} + \cdots + z\frac{C^2(C-z+C^2)^{d-1}}{z^d} + \cdots \\ &= z\left(1 + \frac{C^2}{z-(C-z+C^2)}\right). \end{aligned}$$

Simplification gives

$$C^3 + C^2 - 3zC + 2z^2 = 0. \tag{15}$$

It is perhaps not immediately clear how to derive a simple expression for the coefficients of (15). However, the equation involves monomials of only two different total degrees, 2 and 3, and as such it can be parametrized rationally. The cubic has a double point at the origin, so that we set $C = tz$ and adopt the slope t as the basic parameter. Then, one has

$$z = -\frac{(t-1)(t-2)}{t^3}, \qquad C = -\frac{(t-1)(t-2)}{t^2}. \tag{16}$$

[1] Stanley observes in a vivid account [30] that the 10th Schröder number 103,049 was already known to Hipparchus in the second century B.C. and to Plutarch in the first century A.D.

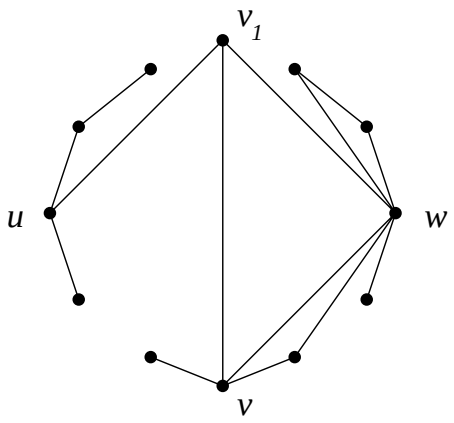


Figure 3: The basic decomposition of graphs with two disjoint graphs between u and v and only one graph between v and w.

The parametrization becomes a polynomial one, upon setting $t = 1/v$, and since the branch of interest $C = z + z^2 + 4z^3 + \cdots$ has slope 1 at the origin (where the expansion of C is sought), it is convenient to set $v = 1 - x$, with x near 0. Then, the parametrization becomes

$$z = x(1-x)(1-2x), \qquad C = \frac{z}{1-x}. \tag{17}$$

The first relation in (17) defines implicitly x as a function of z, while the second one expresses C as a function of $x(z)$. The expansions can then be obtained by the Lagrange inversion theorem, and one finds

$$[z^n]C = [z^{n-1}]\frac{1}{1-x} = \frac{1}{n-1}[z^{n-2}]\frac{x'}{(1-x)^2},$$

which entails

$$[z^n]C = \frac{1}{n-1}[x^{n-2}]\frac{1}{(1-x)^{n+1}(1-2x)^{n-1}}.$$

This last form, suggestive of interesting bijective combinatorics, is equivalent to the one stated in part (i); it does not appear in [7] since the Cardano solution of the cubic equation for $C(z)$ found there does not allow for a simple expansion.

Let now $C(z, w)$ be the GF for connected graphs, where w marks edges. If v_1 has degree d we have to introduce a factor w^d in the corresponding summand before (15), and a simple computation gives

$$wC^3 + wC^2 - z(1+2w)C + z^2(1+w) = 0. \tag{18}$$

This is equation (47) of [7]. To obtain the numbers $C_{n,k}$ we can do the same parametrization as in the univariate case. Put $C = tz$ to obtain

$$zw = x(1-x)(1-x(1+w)),$$

and from here

$$[z^n]C(z,w) = \frac{1}{n-1}[x^{n-2}]\frac{w^{n-1}}{(1-x)^{n+1}(1-x(1+w))^{n-1}}.$$

This expression is equivalent to the formula stated in part (i).

Graphs. Let Γ be a graph and let r be the number of vertices in the component Γ_1 containing v_1. Then Γ has to be completed with r additional graphs (some of them possibly empty), one to the right of every vertex of Γ_1. Let G be the GF of graphs and C the GF of connected graphs as above, then

$$G = 1 + C(zG). \tag{19}$$

Taking into account that C satisfies (15), we can eliminate C and obtain (after cancelling a factor G),

$$G^2 + (2z^2 - 3z - 2)G + 3z + 1 = 0. \tag{20}$$

This equation appears in [7], but in a slightly different form since we are taking the constant term of G to be 1. Solving the quadratic yields

$$G(z) = 1 - \frac{3}{2}z - z^2 - \frac{z}{2}\sqrt{1 - 12z + 4z^2}, \tag{21}$$

which is a recognizable variant of the GF of Schröder numbers [6].

Using this scheme we can easily enumerate graphs according to the number of edges. Let w marks edges, and let $C(z, w)$ be the bivariate GF for connected graphs. Then (19) becomes $G(z, w) = 1 + C(zG(z, w), w)$, because the number of edges in a graph is simply the sum of the number of edges in its components. Eliminating C in (18) we arrive at

$$wG^2 + ((1 + w)z^2 - (1 + 2w)z - 2w)G + w + z(1 + 2w) = 0. \tag{22}$$

This equation becomes amenable to Lagrange inversion, upon the change of variables $G = 1 + z + zy$ that transforms it into

$$y = z(1 + w)\left(\frac{1 + y}{1 - wy}\right).$$

Similarly, let now w mark components. Then (19) becomes $G(z, w) = 1 + wC(zG(z, w))$, where $C(z)$ is the univariate GF for connected graphs, and the factor w takes into account the component containing v_1. Eliminating C in (15) we arrive at

$$G^3 + (2w^3z^2 - 3w^2z + w - 3)G^2 + (3w^2z - 2w + 3)G + w - 1 = 0. \tag{23}$$

The explicit expansion obeys principles similar to what has been done before. Set $G = 1 + wy$, solve for z, and obtain the Lagrange form,

$$y = 4z(1 + yw)\left(3 + \sqrt{1 - 8y}\right)^{-1}.$$

What is required now is an expansion of the negative powers of $q(u) = 3 + \sqrt{1 - 8u}$. A change of variables similar to the one that underlies Lagrange inversion in Cauchy coefficient integrals, namely $q(u) = 4 - 4t$, then shows that

$$[u^a]4^b q(u)^{-b} = [t^a](1 - t)^{-b}(1 - 2t)^{-a-1}(1 - 4t).$$

The rest of the computation is routine.

3 Dissections and Partitions

A *dissection* of a convex polygon $P_n = \{v_1, v_2, \ldots, v_n\}$ is a partition of the polygon into polygonal regions by means of non-crossing diagonals; that is, a non-crossing graph containing the edges $v_1v_2, v_2v_3, \ldots, v_nv_1$. A non-crossing partition of size n is a partition of $[n] = \{1, 2, \ldots, n\}$ such that if $a < b < c < d$ and a block contains a and c, then no block contains b and d. One can draw such a partition on a circle by representing each block as a convex polygon on the points belonging to the block. Then non-crossing partitions are the same as non-crossing graphs whose connected components are points, edges and cycles. (We do not consider here triangulations as they have been investigated so extensively since Euler's time; see [17].)

Theorem 3 (*i*) *The number of dissections of size n is a Schröder number defined in (12) that admits the alternative form,*

$$D_n = \frac{1}{n}\sum_{i=0}^{n-1}(-1)^i\binom{n}{i}\binom{2n-2-i}{n-1-i}2^{n-1-i}.$$

The number of dissections of size n with k regions satisfies

$$D_{n,k} = \frac{1}{k}\binom{n-3}{k-1}\binom{n+k-2}{k-1}.$$

(*ii*) *The number of (noncrossing) partitions of size n is a Catalan number,*

$$P_n = \frac{1}{n+1}\binom{2n}{n},$$

and the number of partitions of size n with k parts is a Narayana number

$$P_{n,k} = \frac{1}{n}\binom{n}{k}\binom{n}{k-1}.$$

The results in the above theorem are all classical and can be found in many references. We include them in order to show how the general methodology allows easy derivations. Kreweras first discussed noncrossing partitions in [23] while results for dissections are summarized in [6, p. 74].

Dissections of a convex polygon. Let δ be a dissection of P_n and let ρ be the region containing the edge v_1v_2. If ρ has $r+1$ sides, then δ is identified with a sequence of r dissections (some of them possibly reduced to a single edge). If we mark with z^2 the dissection consisting of a single edge, then

$$D = z^2 + \frac{D^2}{z} + \cdots + \frac{D^r}{z^{r-1}} + \cdots \tag{24}$$

where the denominator z^{r-1} means that $r-1$ pairs of vertices have been identified. Summation and simplification gives

$$2D^2 - z(1+z)D + z^3 = 0.$$

Solving the quadratic equation yields again a GF that is a variant of the GF of Schröder numbers.

There is an alternative way to expand the GF, not to be found in Comtet's book [6]. Set $D = zy$. Then y satisfies an equation similar to (24),

$$y = z + \frac{y^2}{1-y} \qquad \text{or} \qquad z = y\frac{1-2y}{1-y}.$$

This equation is of the Lagrange type and it can be subjected to inversion,

$$[z^n]y(z) = \frac{1}{n}[u^{n-1}]\left(\frac{1-u}{1-2u}\right)^n,$$

which gives the first relation of part (i). This relation also reveals a combinatorial curiosity: the quantity $nc_n = n[z^n]y(z)$ equals the number of n-tuples of integer compositions with grand total sum equal to $n-1$.

Let now z mark vertices and w mark regions. Then (24) becomes

$$D = z^2 + w(D^2/z + D^3/z^2 + \cdots),$$

where the factor w marks the region containing v_1v_2. This is equivalent to

$$(1+w)D^2 - z(1+z)D + z^3 = 0.$$

Like before, we set $y(z,w) = D(z,w)/z$ and get

$$y = z + w\frac{y^2}{1-y}, \qquad y = z\left(1 - w\frac{y}{1-y}\right)^{-1}.$$

This is again an equation of the Lagrange type and inversion gives

$$[z^n]y(z,w) = \frac{1}{n}[u^{n-1}]\left(1 - w\frac{u}{1-u}\right)^{-n}.$$

From there, the explicit form stated in part (i) results by extracting the coefficient of w^k. We remark that $D_{n,k}$ is also the number of plane trees of the Schröder type, built on $n-1$ external nodes that have k internal nodes, each of degree ≥ 2.

Let us also remark that once we know how to enumerate dissections we can enumerate general graphs. Indeed, a graph is the set of internal diagonals of a dissection plus any set of boundary edges. As a consequence, the number of graphs of size n is $G_n = 2^n D_n$. If the graph has k edges, j of them are internal diagonals and $k-j$ are boundary edges. Hence we obtain $G_{n,k} = \sum_{j=0}^{k}\binom{n}{j}D_{n,j+1}$ as an alternative to the formula stated in Theorem 2.

Non-crossing partitions. Let π be any non-crossing partition and let r be the size of the part π_1 containing vertex v_1. Then π can be encoded as a sequence of r partitions (some of them possibly empty), one for every point in π_1. If P is the GF of non-crossing partitions and 1 denotes the empty partition, then

$$P = \frac{1}{1-zP},$$

and of course we recover the GF for the Catalan numbers (see [17]),

$$zP^2 - P + 1 = 0,$$

with the corresponding Lagrange form for $y = zP$ that reads $y = z(1-y)^{-1}$.

If z marks vertices and w marks parts, then

$$P = 1 + wzP + wz^2P^2 + \cdots = 1 + \frac{wzP}{1-zP},$$

and we get

$$zP^2 + (wz - z - 1)P + 1 = 0.$$

With $y = zP$, this can be written as $y = z(1 + wy/(1-y))$, and Lagrange inversion gives the classical Narayana numbers,

$$[z^n w^k]P(z,w) = \frac{1}{n}\binom{n}{k}\binom{n}{k-1},$$

that also enumerate general plane trees of size $n+1$ that have k leaves.

4 Asymptotic counting

In this section, we prove that each class of non-crossing configurations leads to an asymptotic estimate of the form

$$f_n \sim \gamma \frac{\omega^n}{\sqrt{\pi} n^{3/2}}, \tag{25}$$

where f_n is the number of objects of size n, and γ, ω are context-dependent algebraic numbers. Such estimates are for instance familiar in the theory of tree enumerations [8, 19, 24, 26].

Roughly, each of the six counting generating functions is an algebraic function, as seen in Sections 1,2,3. It is known that the singularities of GFs determine the asymptotics of their coefficients. Here, we *a priori* expect local singular expansions in the form of Puiseux expansions, that is to say expansions involving fractional exponents. Generically, singularities of the square-root type are expected, like in many implicitly defined functions [8, 19]. All our GFs appear to be of this type, with a local expansion near the dominant singularity ρ being

$$f(z) \sim c_0 + c_1\sqrt{1 - z/\rho}. \tag{26}$$

Then singularity analysis [13] is used to achieve the transfer of (26) to coefficients leading to estimates of the form (25).

Rather than examining each case separately, we develop here a common strategy that is adequate for treating all classes discussed in previous sections (in one case, the argument needs to be mildly amended) and is systematic to be amenable to treatment by a computer algebra system, while paving the way for the distributional analyses of the next section.

Theorem 4 *Consider the configurations of trees, forests, connected graphs, graphs, dissections, and partitions. The corresponding counts each satisfy an asymptotic estimate of the form*

$$f_n = \gamma \frac{\omega^n}{\sqrt{\pi} n^{3/2}} \left(1 + \mathcal{O}(\frac{1}{n})\right),$$

where γ, ω are algebraic numbers given in Table 2.

The asymptotic counting of graphs was obtained by Domb and Barrett [7] using Darboux's method; the asymptotic form of Schröder numbers is certainly known to many and is close to the framework of simple families of trees introduced by Meir and Moon [24]. The asymptotics of trees and partitions can be directly obtained from explicit formulæ and Stirling's approximation. The present approach is introduced because it has the merit of providing a global approach while lending itself naturally to a perturbation analysis that leads to Gaussian laws.

Proof. The generating functions considered so far are *algebraic* functions, meaning that they satisfy a system of polynomial equations. From classical elimination theory, any system can be reduced to a single polynomial equation,

$$P(z,y) = 0, \qquad P \in \mathbb{Q}[z,y], \tag{27}$$

	Class	ω	*Num. value*	γ
(T)	Trees	$\frac{27}{4}$	6.75000	$\frac{\sqrt{3}}{27}$
(F)	Forests	$\frac{1}{\xi}$	8.22469	0.07465
(C)	Connected graphs	$6\sqrt{3}$	10.39230	$\frac{\sqrt{6}}{9} - \frac{\sqrt{2}}{6}$
(G)	Graphs	$6 + 4\sqrt{2}$	11.65685	$\frac{1}{4}\sqrt{-140 + 99\sqrt{2}}$
(D)	Dissections	$3 + 2\sqrt{2}$	5.82842	$\frac{1}{4}\sqrt{-140 + 99\sqrt{2}}$
(P)	Partitions	4	4.00000	1

Table 2: The constants appearing in the statement of Theorem 4. There, ξ denotes the root of the polynomial $4 - 32x - 8x^2 + 5x^3$ that is near 0.121, and 0.07465 represents the explicit algebraic number of degree 6 equal to $\beta/2$, with β given in the text.

and reduction to such a form may be achieved systematically by either resultant or Groebner basis elimination [16]. Here, our combinatorial specifications being simple enough, elimination is immediate, so that the form (27) is directly available from previous sections.

Consider a polynomial equation

$$P(z, y) \equiv \sum_{j=0}^{d} a_j(z) y^j = 0. \tag{28}$$

It has in general (that is, except for a finite set of exceptional values) d distinct solutions that are then analytic branches of a complex algebraic curve; see for instance the discussion of the Weierstrass Preparation Theorem in [1] or [20].

A finite set Ω of candidate singularities can be determined systematically by a general process explained below. The problem is then to determine which of the elements of Ω are dominant singularities (that is, singularities of smallest modulus) of the branch that coincides with the counting generating function under study and is thereby identified by its expansion at 0. In all generality, such a determination implies solving a so-called connection problem between branches [5]. However, the problems under consideration are once more simple enough, so that Ω can be "filtered" and reduced, in each case, to a single element by means of elementary arguments. We find that each generating function $f(z)$ has a unique dominant and positive real singularity at some $\rho > 0$ near which it satisfies an expansion of the square-root type,

$$f(z) = c_0 + c_1(1 - z/\rho)^{1/2} + c_2(1 - z/\rho) + \mathcal{O}((1 - z/\rho)^{3/2}). \tag{29}$$

Then, by Darboux's method [6, 19] or singularity analysis [13], *transfer* from the singular expansion (29) to coefficients is permissible and

$$[z^n]f(z) = \frac{c_1}{\Gamma(-1/2)} \frac{\rho^{-n}}{\sqrt{n^3}} \left(1 + \mathcal{O}(\frac{1}{n})\right), \tag{30}$$

a form that matches (25) with $\omega = \rho^{-1}$ and $\gamma = -c_1/2$.

The last phase of asymptotic transfer is a standard one. We thus concentrate on the problem of singularity localization and singular expansion and refer to the papers by Klarner and Woodworth [22] as well as by Canfield [5] for background.

A partial algorithm. The polynomial equation $P(z,y) = 0$ has in general d roots or branches for a fixed value of z. When the leading coefficient $a_d(z)$ vanishes, some of these branches escape to infinity and are thus potential singularities. Singularities may otherwise only arise at points z such that the two equations

$$P(z,y) = 0, \qquad \frac{\partial}{\partial y} P(z,y) = 0$$

have a common root y. In this case, two branches meet and there is possibly a branch point. Such places where branches meet are thus zeros of the resultant polynomial,

$$R(z) := \mathrm{Result}_y \left(P(z,y), \frac{\partial}{\partial y} P(z,y) \right). \tag{31}$$

At all other points, there are d distinct branches that are each analytic by Weierstrass preparation. Then, a superset of the set of singularities is

$$\Omega = \left\{ z \;\; \middle| \;\; R(z) \cdot a_d(z) = 0 \right\}. \tag{32}$$

The generating functions of noncrossing configurations all have a radius of convergence in the interval $[0,1]$ since their coefficients satisfy combinatorial bounds of the form $A^n < f_n < B^n$, for some A, B with $1 < A < B < \infty$. Thus, one needs only consider

$$\Omega_1 = \Omega \cap \left\{ z \;\; \middle| \;\; |z| < 1 \right\},$$

which must contain at least one positive element ρ. (Pringsheim's theorem asserts that a function with nonnegative coefficients is singular at its radius of convergence [31].) If Ω_1 has cardinality 1, a unique dominant singularity has been found[2] We thus assume the uniqueness condition to be satisfied.

In all cases under consideration, the function $f(z)$ remains finite at its singularity since $a_d(\rho) \neq 0$. We set

$$\tau := \lim_{z \to \rho^-} f(z),$$

so that τ also equals the quantity c_0 in (26). The quantity τ is a double root of $P(\rho, y) = 0$ and it has to be positive. It is thus a root of the resultant polynomial

$$S(y) := \mathrm{Result}_z \left(P(z,y), \frac{\partial}{\partial y} P(z,y) \right). \tag{33}$$

(If these conditions are not sufficient, at least τ could be isolated by carefully controlled numerical analysis of $f(z)$ for $z \in (0, \rho)$.)

By the general theory of algebraic functions [20], a Puiseux expansion —an expansion into fractional powers, that is, into powers of $(1 - z/\rho)^{1/r}$— holds locally at $z = \rho$, for some integer $r > 1$. Such an expansion derives explicitly from the bivariate expansion of $P(z,y)$ at (ρ, τ),

$$P(z,y) = p_{00} + p_{10} Z + p_{01} Y + p_{20} Z^2 + p_{11} ZY + p_{02} Y^2 + \cdots, \tag{34}$$

[2] This situation covers five out of our six cases. The exception is the case of connected graphs where $\Omega_1 = \left\{ -1/(6\sqrt{3}),\ 1/(6\sqrt{3}) \right\}$, but for which the parametrization (16,17) permits us to eliminate the negative value from the set of candidate singularities by simply following the branch at the origin that corresponds to the combinatorial GF. (Domb and Barrett [7] do not adress this issue explicitly.) Alternatively, one could appeal to the powerful theorems of Drmota [8].

$$p_{ij} := \frac{1}{i!\,j!}\,\frac{\partial^{i+j}}{\partial z^i \partial y^j} P(z,y)\Big|_{(\rho,\tau)}, \qquad Z = z-\rho,\ Y = y-\tau.$$

By assumption, $p_{00} = p_{01} = 0$. Provided the condition,

$$p_{02} \neq 0, \tag{35}$$

holds, then the dependency between Y and Z is locally quadratic, and as $z \to \rho$,

$$f(z) = c_0 + c_1(1-z/\rho)^{1/2} + \mathcal{O}((1-z/\rho)), \qquad c_0 = \tau, \quad c_1 = -\left(\frac{2\rho\, p_{10}}{p_{02}}\right)^{1/2} \tag{36}$$

(The minus sign in c_1 must be adopted here since the generating function increases with its argument.)

In summary, if the condition (35) is satisfied, then the singular expansion (29) holds, and the asymptotic forms of coefficients (25,30) have been established. Condition (35) is itself satisfied generically and is easily checked numerically in each individual case. The coefficients in the expansions are then all explicitly computable algebraic numbers. □

The above programme has been carried out for all non-crossing configurations defined in previous sections. Computations have been performed under the Maple system for symbolic manipulations, together with the `Gfun` extension due to Salvy and Zimmermann [27]. In particular, the `Gfun` package provides automatically Puiseux expansions of algebraic functions, a great help here.

Here is an outline of the computation for the case of forests, where $y(z) = T(z)$ is defined by (10). There, some care is needed in selecting correct algebraic conjugates amongst various possibilities. The basic GF equation is (10). The resultant polynomial $R(z)$ defined in (31) is found mechanically to be

$$R(z) = -z^3(4 - 32z - 8z^2 + 5z^3),$$

whose roots are the four algebraic numbers,

$$\Omega = \{0,\ -1.93028,\ 0.12158,\ 3.40869\}$$

(approximately). Therefore, a unique dominant singularity of $F(z)$ has been isolated,

$$\Omega_1 = \{\xi \doteq 0.12158,\ 4 - 32\xi - 8\xi^2 + 5\xi^3 = 0\}.$$

The three branches of the cubic give rise at $z = \rho$ to one branch that is analytic when $z = \xi$, with value numerically close to 0.67816, and two conjugate branches with value 1.21429 at $z = \rho$. The expansion of the two conjugate branches starts as

$$\alpha \pm \beta\sqrt{1 - z/\xi} + \cdots,$$

where

$$\alpha = \frac{43}{37} + \frac{18}{37}\xi - \frac{35}{74}\xi^2 \doteq 1.21429, \qquad \beta = \frac{1}{37}\sqrt{228 - 981\xi - 5290\xi^2} \doteq 0.14931\,,$$

and the determination with the minus sign must be taken for the combinatorial GF. The computation can be conveniently based upon `Gfun`'s ability to determine Puiseux expansions. The data for our six families are summarized in Table 2.

5 Limit laws

The six basic combinatorial types of Sections 1–3 give rise to seven basic parameters for which BGFs $f(z,w)$ have been found to satisfy polynomial equations of the form

$$P(z,w,f(z,w)=0.$$

These equations, together with a few initial conditions provided by the combinatorics of the problems, fully determine the BGFs. The problem of estimating the coefficients

$$f_{n,k}=[z^n u^k]f(z,w)$$

is then a bivariate asymptotic problem.

The quantities

$$\pi_{n,k}=\frac{f_{n,k}}{f_n},$$

represent discrete probability distributions. Let μ_n and σ_n^2 be the mean and variance of such a distribution $\pi_{n,k}$. Classically, the distribution $\pi_{n,k}$ is said to be *asymptotically normal* (or Gaussian) if, pointwise for each $x\in\mathbb{R}$,

$$\lim_{n\to\infty}\sum_{k\le\mu_n+x\sigma_n}\pi_{n,k}=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}e^{-t^2/2}\,dt. \tag{37}$$

In other words, the distribution of the random variable X_n representing parameter χ taken on non-crossing configurations of size n, has a distribution function that, after normalization, tends to the Gaussian distribution function. We establish now that our seven reference parameters all have laws that are asymptotically normal. For background information on these analytic techniques, we refer globally to [2, 3, 8, 21] and the exposition in [11] or [14, Ch. 9].

Theorem 5 *Consider the following parameters: number of leaves in trees, components in forests, edges in connected graphs, components in graphs, edges in graphs, regions in dissections, parts in partitions. The corresponding distributions over objects of size n each have mean μ_n and variance σ_n^2 that satisfy*

$$\mu_n\sim\kappa n,\qquad \sigma_n^2\sim\lambda n,$$

where κ,λ are algebraic numbers given in Table 3. The laws are in each case asymptotically normal.

Proof. As seen in the proof of Theorem 4, each of the counting GF $f(z)$ has a unique dominant singularity ρ that is of the square-root type, see (26,29). This in turn entails, by singularity analysis, that the various types of non-crossing configurations all obey an asymptotic formula of the form (29).

Consider a parameter χ like the number of leaves, edges, components, etc, and let $f(z,w)$ be the corresponding bivariate GF. Our goal is to establish a lifted form of the singular expansion (26),

$$f(z,w)=c_0(w)+c_1(w)\sqrt{1-z/\rho(w)}+\mathcal{O}(1-z/\rho(w)), \tag{38}$$

Class, Parameter	κ (mean)		λ (variance)	
Trees, leaves	$\frac{4}{9}$	0.444	$\frac{28}{243}$	0.115
Forests, components	$\frac{8}{37} - \frac{13}{37}\xi + \frac{15}{74}\xi^2$	0.176	$\frac{192}{1369} + \frac{5}{2738}\xi - \frac{47}{2738}\xi^2$	0.140
Connected graphs, edges	$\frac{1}{2} + \frac{\sqrt{3}}{2}$	1.366	$\frac{1}{4}$	0.250
Graphs, edges	$\frac{1}{2} + \frac{\sqrt{2}}{2}$	1.207	$\frac{1}{4} + \frac{\sqrt{2}}{8}$	0.426
Graphs. components	$\frac{5}{7} - \frac{3}{7}\sqrt{2}$	0.108	$\frac{50}{2401} + \frac{255}{4802}\sqrt{2}$	0.095
Dissections, parts	$\frac{\sqrt{2}}{2}$	0.707	$\frac{\sqrt{2}}{8}$	0.176
Partitions, parts	$\frac{1}{2}$	0.500	$\frac{1}{8}$	0.125

Table 3: The constants appearing in the statement of Theorem 5. There, ξ denotes the root near 0.121 of the polynomial $4 - 32z - 8z^2 + 5z^3$.

uniformly with respect to w for w in a small neighbourhood of 1, and with $\rho(w), c_0(w), c_1(w)$ analytic at $w = 1$. There, $\rho(w)$ is the dominant singularity (assumed to be unique) of $f(z, w)$, where w is treated as a parameter. If (38) is granted, then, by singularity analysis,

$$f_n(w) := [z^n]f(z, w) = \gamma(w) \left(\frac{1}{\rho(w)}\right)^n \left(1 + \mathcal{O}(\frac{1}{n^{1/2}})\right), \tag{39}$$

for some analytic function $\gamma(w)$, with an error term that is uniform with respect to w. Uniformity is crucial and is granted in all generality by the constructive character of the singularity analysis method. (See the discussion in [13].)

The probability generating function of χ satisfies

$$q_n(w) := \frac{f_n(w)}{f_n} = \frac{\gamma(w)}{\gamma(1)} \left(\frac{\rho(1)}{\rho(w)}\right)^n \left(1 + \mathcal{O}(\frac{1}{\sqrt{n}})\right). \tag{40}$$

This means that $q_n(w)$ is a so-called "quasi-power". In particular, the mean $\mu_n = q_n'(1)$ and the variance $\sigma_n^2 = q_n''(1) + q_n'(1) - q_n'(1)^2$ result by differentiation of (40), so that

$$\kappa = -\frac{\rho'(1)}{\rho(1)}, \qquad \lambda = -\frac{\rho''(1)}{\rho(1)} - \frac{\rho'(1)}{\rho(1)} + \left(\frac{\rho'(1)}{\rho(1)}\right)^2. \tag{41}$$

Then, by extensions due to Bender, Richmond and Hwang of the central limit theorem, a limiting Gaussian law for the distribution of χ results from (40). Basically, from the quasi-powers form, the normalized characteristic functions $\phi_n(t) = e^{-it\mu_n/\sigma_n} q_n(e^{it/\sigma_n})$ converge to the characteristic function of a standard normal, namely $e^{-t^2/2}$. The limit law then derives as a consequence of the continuity theorem for characteristic functions.

At this stage, the proof of the theorem is completed as soon as one can establish the lifted expansion (38) for each of the seven parameters under consideration. The proof relies on the

permanence of analytic relations under "perturbation" by an auxiliary parameter, a property that is technically granted by the Weierstrass preparation theorem.

Consider the lifted version of the resultant of (31),

$$R(z,w) = \mathrm{Result}_y \left(P(z,y,w), \frac{\partial}{\partial y} P(z,y,w) \right). \tag{42}$$

This is a polynomial whose restriction $P(z,1)$ has, by the developments of the proof of Theorem 4 and the companion computations, a unique isolated root at $z=\rho$. By the implicit function theorem and the Weierstrass preparation theorem [1, 20], this root lifts to a unique root near ρ that is an analytic branch $\rho(w)$ of an algebraic function, for w in a small neighbourhood of 1:

$$R(\rho(w), w) = 0, \qquad \rho(1) = 1. \tag{43}$$

Then, by Weierstrass preparation again, the analytic factorization

$$P(z,y) = (y^2 + m_1(z)y + m_2(z)) \cdot H(z,y),$$

with $H(\rho,\tau) \neq 0$, that corresponds to a square root singularity, lifts to

$$P(z,y,w) = (y^2 + m_1(z,w)y + m_2(z,w)) \cdot H(z,y,w),$$

with $H(\rho,\tau,1) \neq 0$. Then, the quadratic formula yields

$$f(z,w) = \frac{1}{2}\left(-m_1(z,w) - \sqrt{m_1(z,w)^2 - 4m_2(z,w)} \right).$$

It then suffices to expand $f(z,w)$ near $(\rho(w),w)$ in order to get the uniform family of singular expansions (38), hence eventually, the Gaussian limit law[3]. □

Globally, the process discussed here is one of "singularity perturbation" where one has to establish that the singular expansion of a BGF has a smooth analytic behaviour when the auxiliary parameter w varies in a small neighbourhood of 1. Computationally, the process is simple. The algebraic function $\rho(w)$ is determined by Eq. (43). The regular expansion of the branch that coincides with ρ at $w=1$ provides the first two moments.

For instance, for edges in connected graphs, the algebraic equation is (18). The resultant polynomial is found to be

$$R(z,w) = w^2 z^2 \left(27w(w+1)^2 z^2 + 2(w-1)(2w+1)(w+2)z - w \right).$$

The expansion of $\rho(w)$ at $w=1$ is determined by the implicit function theorem, and its coefficients are simply rational functions of ρ as $\rho(w)$ is analytic. The computation is again conveniently handled by the `Gfun` package of Maple,

$$\rho(w) = \frac{1}{18}\sqrt{3} - \left(\frac{1}{12} + \frac{1}{36}\sqrt{3} \right)(w-1) + \left(\frac{1}{12} + \frac{5}{144}\sqrt{3} \right)(w-1)^2 + \mathcal{O}((w-1)^3).$$

The result found is then best expressed under logarithmic-exponential form, where the mean and variance coefficients of (41) read directly:

$$\log\left(\frac{\rho(1)}{\rho(e^s)} \right) = \kappa s + \frac{1}{2}\lambda s^2 + \mathcal{O}(s^3) = \left(\frac{1}{2} + \frac{\sqrt{3}}{2} \right) s + \frac{1}{8}s^2 + \mathcal{O}(s^3).$$

This gives $\kappa = \frac{1}{2} + \frac{\sqrt{3}}{2}$ and $\lambda = \frac{1}{4}$. The data for the seven parameters under consideration are all obtained in this way and summarized in Table 3.

[3] In addition, by the Berry-Esseen inequalities [10], the speed of convergence to the Gaussian limit is $\mathcal{O}(n^{-1/2})$ uniformly.

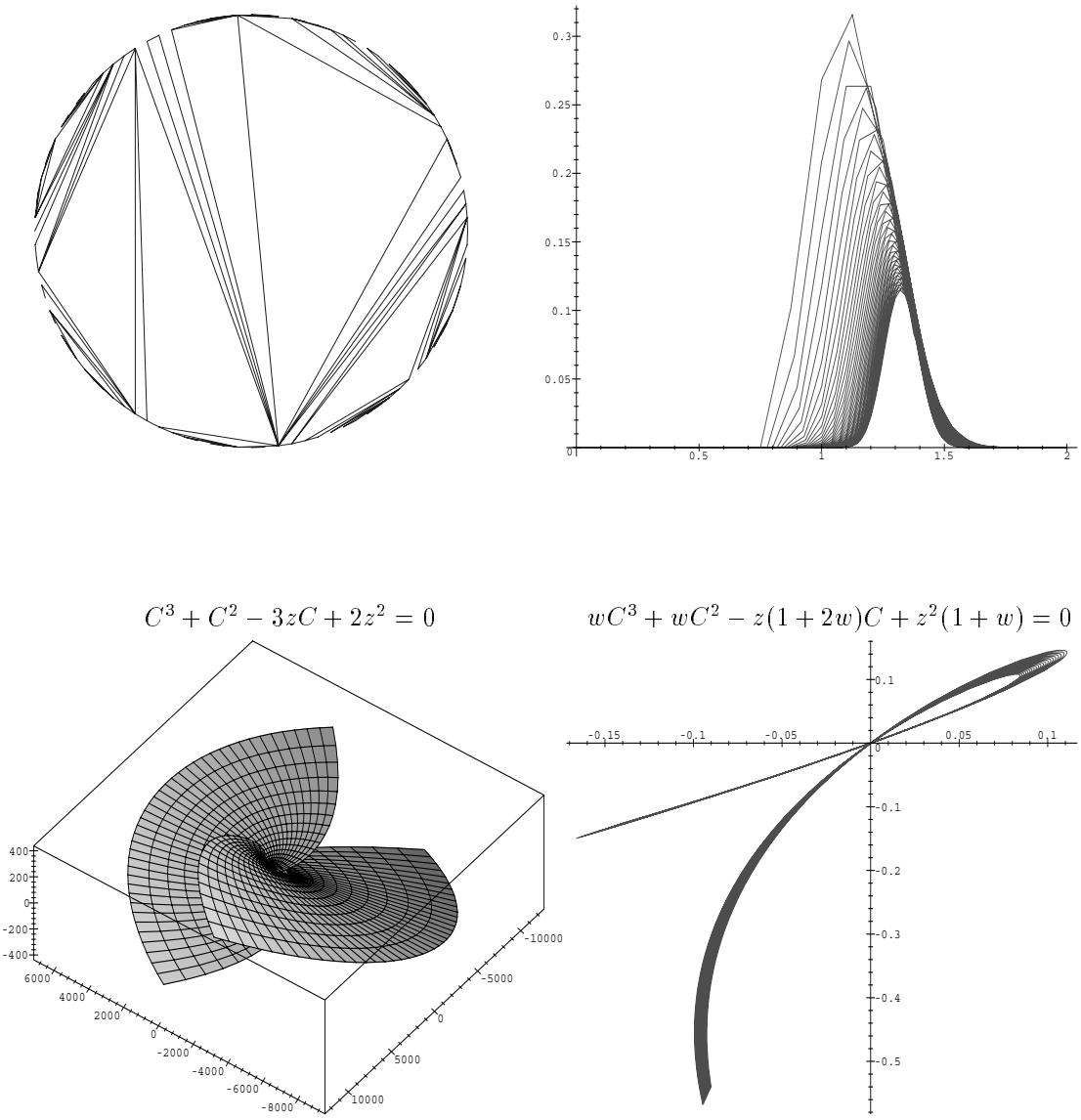


Figure 4: Noncrossing connected graphs (top, left: a random instance of size 100) have a combinatorial decomposition of the "cubic" type, reflected by cubic generating functions. The counting generating function (bottom left: a 3-dimensional plot of $\Im C(z)$ for complex z) has an algebraic branch point of the square-root type that induces an asymptotic count of type $\omega^n/n^{-3/2}$. The family of generating functions $\{C(z,w)\}_w$ where w records the number of edges (bottom right: plot of $C(z,w)$ for real z, when w varies in between 0.9 and 1.1) exhibit a common square-root singularity that moves analytically with w, a fact that induces a limit law of the Gaussian type for the number of edges (top right: histograms of the distribution for $n = 8\,..\,50$, with x-axis scaled to n).

6 Conclusion

Symbolic methods in combinatorial enumerations lead in many cases to easy derivations of generating function equations. This observation applies with special strength here since planarity constraints and the distinguishable character of vertices entail strong decomposition properties. As a result, the generating functions are all algebraic. Singularity analysis and singularity perturbation methods then allow for a transparent treatment that is also computationally effective. A graphical illustration of the chain is presented in Fig. 4.

It is clear that a large number of similar problems are amenable to this chain. Instances are leaves in forests and isolated points or vertices in graphs for which Gaussian laws can be proved to hold by the methods employed here. Trees whoses degrees are bounded by some fixed integer b can be enumerated for each fixed b, their generating functions remain algebraic, and similarly for 1-regular and 2-regular graphs. In all these cases, symbolic methods in conjunction with complex asymptotics allow for a concise and unified characterization of properties of random structures, a distinctive feature of analytic combinatorics.

Acknowledgements. This work was supported in part by the Long Term Research Project *Alcom-IT* (# 20244) of the European Union. The authors are grateful to Frédéric Cazals for his contribution in establishing the data in Tables 2, 3 and his help with random generation and plots.

References

[1] Abhyankar, S.-S. *Algebraic geometry for scientists and engineers.* American Mathematical Society, 1990.

[2] Bender, E. A. Central and local limit theorems applied to asymptotic enumeration. *Journal of Combinatorial Theory 15* (1973), 91–111.

[3] Bender, E. A., and Richmond, L. B. Central and local limit theorems applied to asymptotic enumeration II: Multivariate generating functions. *Journal of Combinatorial Theory,* Series A *34* (1983), 255–265.

[4] Bergeron, F., Labelle, G., and Leroux, P. *Théorie des espèces et combinatoire des structures arborescentes.* No. 19 in Publications du LACIM. Université du Québec à Montréal, 1994.

[5] Canfield, E. R. Remarks on an asymptotic method in combinatorics. *Journal of Combinatorial Theory,* Series A *37* (1984), 348–352.

[6] Comtet, L. *Advanced Combinatorics.* Reidel, Dordrecht, 1974.

[7] Domb, C., and Barrett, A. Enumeration of ladder graphs. *Discrete Mathematics 9* (1974), 341–358.

[8] Drmota, M. Systems of functional equations. *Random Structures and Algorithms 10*, 1–2 (1997), 103–124.

[9] Dulucq, S., and Penaud, J.-G. Cordes, arbres et permutations. *Discrete Mathematics 117* (1993), 89–105.

[10] Feller, W. *An Introduction to Probability Theory and Its Applications*, vol. 2. John Wiley, 1971.

[11] Flajolet, P., Hwang, H.-K., and Soria, M. The ubiquitous Gaussian law in analytic combinatorics, 1997. In preparation.

[12] Flajolet, P., and Lafforgue, T. Search costs in quadtrees and singularity perturbation asymptotics. *Discrete and Computational Geometry 12*, 4 (1994), 151–175.

[13] Flajolet, P., and Odlyzko, A. M. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics 3*, 2 (1990), 216–240.

[14] Flajolet, P., and Sedgewick, R. Analytic combinatorics. Book in preparation, 1998. (Individual chapters are available as INRIA Research Reports 1888, 2026, 2376, 2956, 3162.).

[15] Foata, D. *La série génératrice exponentielle dans les problèmes d'énumération.* S.M.S. Montreal University Press, 1974.

[16] Geddes, K. O., Czapor, S. R., and Labahn, G. *Algorithms for Computer Algebra.* Kluwer Academic Publishers, Boston, 1992.

[17] Gould, H. W. Research bibliography on two number sequences. In *Mathematica Monongaliae*, 1971. (A comprehensive bibliography on Bell and Catalan numbers).

[18] Goulden, I. P., and Jackson, D. M. *Combinatorial Enumeration.* John Wiley, New York, 1983.

[19] Harary, F., and Palmer, E. M. *Graphical Enumeration.* Academic Press, 1973.

[20] Hille, E. *Analytic Function Theory.* Blaisdell Publishing Company, Waltham, 1962. 2 Volumes.

[21] Hwang, H.-K. *Théorèmes limites pour les structures combinatoires et les fonctions arithmetiques.* PhD thesis, École Polytechnique, Dec. 1994.

[22] Klarner, D. A., and Woodworth, P. Asymptotics for coefficients of algebraic functions. *Aequationes Mathematicae 23* (1981), 236–241.

[23] Kreweras, G. Sur les partitions non croisées d'un cycle. *Discrete Mathematics 1* (1972), 333–350.

[24] Meir, A., and Moon, J. W. On the altitude of nodes in random trees. *Canadian Journal of Mathematics 30* (1978), 997–1015.

[25] Noy, M. Enumeration of non-crossing trees. In *Proceedings of the 7th Conference on Formal Power Series and Algebraic Combinatorics* (1995), pp. 465–472. (Full version to appear in *Discrete Mathematics.*).

[26] Pólya, G., and Read, R. C. *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds.* Springer Verlag, New York, 1987.

[27] Salvy, B., and Zimmermann, P. GFUN: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Trans. Math. Softw. 20*, 2 (1994), 163–167.

[28] Sedgewick, R., and Flajolet, P. *An Introduction to the Analysis of Algorithms.* Addison-Wesley Publishing Company, 1996.

[29] Stanley, R. P. Generating functions. In *Studies in Combinatorics,* M.A.A. Studies in Mathematics, Vol. 17. (1978), G.-C. Rota, Ed., The Mathematical Association of America, pp. 100–141.

[30] Stanley, R. P. Hipparchus, Plutarch, Schröder and Hough. *American Mathematical Monthly 104* (1997), 344–350.

[31] Titchmarsh, E. C. *The Theory of Functions*, second ed. Oxford University Press, 1939.

[32] Wilf, H. S. *Generatingfunctionology.* Academic Press, 1990.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105,
78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS
Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
(France)
http://www.inria.fr
ISSN 0249-6399

IDENTITIES FOR THE TOTAL NUMBER OF PARTS IN PARTITIONS OF INTEGERS

ARNOLD KNOPFMACHER † AND NEVILLE ROBBINS

ABSTRACT. We consider the total number of parts in partitions of the natural number n, and derive identities relating this function to the number of partitions and to other familiar number theoretic functions. The total number of parts in partitions with distinct parts and partitions with other restrictions is also considered.

1. INTRODUCTION

Let $a_1 + a_2 + \cdots + a_k = n$, with $a_{i+1} \geq a_i$ for $i \geq 1$, $a_1 \geq 1$ be a partitition of n. It is well known that the ordinary generating function for partitions is

$$P(x) := \sum_{n\geq 0} p(n)x^n = \prod_{i\geq 1} \frac{1}{1-x^i}.$$

If we are interested to count the number of parts (or summands) in the partitions we can introduce the formal bivariate generating function

$$P(x,u) := \sum_{n\geq 0}\sum_{k\geq 1} p(n,k)x^n u^k = \prod_{i\geq 1} \frac{1}{1-ux^i},$$

where $p(n,k)$ counts partitions with exactly k parts.

Let $s(n)$ be the total number of parts in all partitions of the natural number n. Then the formula $s(n) = \sum_{k=1}^{n} kp(n,k)$ implies that

$$S(x) := \sum_{n\geq 0} s(n)x^n = \frac{\partial}{\partial u}P(x,u)|_{u=1} = P(x)\sum_{k\geq 1}\frac{z^k}{1-z^k} = P(x)\sum_{m\geq 1} d(m)x^m,$$

where $d(m)$ is the number of divisors of m.

By equating coefficients above we obtain the formula

$$s(n) = \sum_{i=1}^{n} d(i)p(n-i).$$

The $s(n)$ sequence begins as follows for $n \geq 0$,

$$0, 1, 3, 6, 12, 20, 35, 54, 86, 128, 192, 275, 399, 556, 780, 1068, 1463, 1965, 2644, 3498, \cdots.$$

In Sloane's online encyclopaedia of integer sequences [3] this is **A006128**.

Date: February 25, 2003.

†This material is based upon work supported by the National Research Foundation under grant number 2053740.

In addition, by transposing the Ferrers graph of each partition of n, it follows that $s(n)$ is also equal to the sum of the largest parts of all partitions of n. From this interpretation we can derive a further form for the generating function of $s(n)$:

Let $r_k(x)$ be the generating function for partitions with largest part equal to k. Then

$$r_k(x) = x^k \prod_{i=1}^{k} \frac{1}{1-x^i}.$$

Therefore

$$S(x) = \sum_{k\geq 1} k r_k(x) = \sum_{k\geq 1} k x^k \prod_{i=1}^{k} \frac{1}{1-x^i}.$$

The results given so far are known and appear without proof in Sloane's online encyclopaedia of integer sequences [3] for example. The earliest study of $s(n)$ is due to Erdös and Lehner [2].

Our aim in this section is to find some new identities involving $s(n)$. To do this we make use of a number well known partition identities which can be found for example in Andrews [1].

Theorem 1. *For $n \geq 1$,*

$$s(n) = \sum_{i\geq 1} (-1)^{i-1} (s(n-\omega(i)) + s(n-\omega(-i)) + d(n),$$

where $\omega(i) = \frac{i(3i-1)}{2}$, $i \geq 1$.

Proof. Firstly

$$S(x) \prod_{i\geq 1} (1-x^i) = \sum_{n\geq 1} d(m) x^m. \tag{1}$$

Using Euler's pentagonal number theorem

$$\prod_{i\geq 1} (1-x^i) = \left(1 + \sum_{i\geq 1} (-1)^i (x^{\omega(i)} + x^{\omega(-i)})\right),$$

we obtain

$$S(x) \left(1 + \sum_{i\geq 1} (-1)^i (x^{\omega(i)} + x^{\omega(-i)})\right) = \sum_{n\geq 1} d(m) x^m.$$

Equating coefficients of x^n on each side and rearranging gives the result. □

Recall that the ordinary generating function for partitions with distinct parts is

$$Q(x) := \sum_{n\geq 0} q(n) x^n = \prod_{i\geq 1} (1+x^i). \tag{2}$$

Theorem 2. *For $n \geq 1$,*

$$s(n) + \sum_{i\geq 1}(-1)^i(s(n-2\omega(i)) + s(n-2\omega(-i)) = \sum_{i=1}^{n} d(i)q(n-i).$$

Proof. Multiplying (1) by (2) we obtain

$$S(x)\prod_{i\geq 1}(1-x^{2i}) = Q(x)\sum_{n\geq 1} d(m)x^m.$$

Equating coefficients of x^n on each side gives the result. □

More generally, let $b_r(n)$ denote the number of r-regular partitions of n, that is the number of partitions of n such that no part is divisible by r, or equivalently, the number of partitions of n such that no part occurs r or more times. It is known that the generating function of $b_r(n)$ is given by:

$$\sum_{n\geq 0} b_r(n)x^n = \prod_{n\geq 1}\frac{1-x^{rn}}{1-x^n}. \tag{3}$$

Then we have:

Theorem 3. *Let $m \geq 1$. Then*

$$s(n) + \sum_{j\geq 1}(-1)^j(s(n-2^m\omega(j)) + s(n-2^m\omega(-j))) = \sum_{k=1}^{n} b_{2^m}(n-k)d(k).$$

Proof. Identities (1) and (3) imply that

$$S(x)\prod_{n\geq 1}(1-x^{2^m}n) = (\sum_{n\geq 1} d(n)x^n)(\prod_{n\geq 1} b_{2^m}(n)x^n).$$

The conclusion now follows by matching coefficients of like powers of x.

□

Theorem 4. *For $n \geq 1$,*

$$\sum_{r\geq 0} s(n - \frac{r(r+1)}{2}) = \sum_{i+j+k=n} d(i)q(j)q(k)$$

Remarks: In the right member of the formula, we have $i, j, k \geq 0$.

Proof. Replacing x by x^2 in (1), we obtain:

$$S(x^2)\prod_{n\geq 1}(1-x^{2n}) = \sum_{n\geq 1} d(n)x^{2n},$$

hence

$$S(x^2)\prod_{n\geq 1}(1-x^{2n})(1+x^{2n-1}z)(1+x^{2n-1}z^{-1}) = (\sum_{n\geq 1} d(n)x^{2n})\prod_{n\geq 1}(1+x^{2n-1}z)(1+x^{2n-1}z^{-1}).$$

By the Jacobi triple product identity, the left member of the preceding equation may be simplified, so that we get

$$S(x^2)\sum_{n=-\infty}^{\infty} x^{n^2}z^n = (\sum_{n\geq 1} d(n)x^{2n})\prod_{n\geq 1}(1+x^{2n-1}z)(1+x^{2n-1}z^{-1}).$$

Letting $z = x$, we obtain:

$$S(x^2)\sum_{n=-\infty}^{\infty} x^{n^2+n} = (\sum_{n\geq 1} d(n)x^{2n})\prod_{n\geq 1}(1+x^{2n})\prod_{n\geq 1}(1+x^{2n-2}),$$

that is,

$$S(x^2)\sum_{n=-\infty}^{\infty} x^{n^2+n} = 2(\sum_{n\geq 1} d(n)x^{2n})\prod_{n\geq 1}(1+x^{2n})^2.$$

Replacing x^2 by x, we have:

$$S(x)\sum_{n=-\infty}^{\infty} x^{\frac{n^2+n}{2}} = 2(\sum_{n\geq 1} d(n)x^{n})\prod_{n\geq 1}(1+x^{n})^2.$$

By symmetry, we have:

$$S(x)\sum_{n=0}^{\infty} x^{\frac{n^2+n}{2}} = (\sum_{n\geq 1} d(n)x^{n})(\sum_{n\geq 0} q(n)x^{n})^2,$$

that is,

$$(\sum_{n\geq 0} s(n)x^{n})(\sum_{n=0}^{\infty} x^{\frac{n^2+n}{2}}) = (\sum_{n\geq 1} d(n)x^{n})(\sum_{n\geq 0} q(n)x^{n})^2.$$

The conclusion now follows by matching coefficients of like powers of x. □

Theorem 5. *Let* $\prod_{n\geq 1}(1-x^n) = \sum_{n\geq 0} E(n)x^n$, *so that*

$$E(n) = \begin{cases} (-1)^r & \text{if } n = \omega(\pm r) \\ 0 & \text{otherwise} \end{cases}.$$

then

$$s(n) + \sum_{j\geq 1}(-1)^j(2j+1)s(n-\frac{j(j+1)}{2}) = \sum_{i+j+k=n} d(i)E(j)E(k).$$

Proof. Identity (1) implies

$$S(x)\prod_{n\geq 1}(1-x^n)^3 = (\sum_{n\geq 1} d(n)x^n)\prod_{n\geq 1}(1-x^n)^2.$$

A well-known identity of Jacobi states that

$$\prod_{n\geq 1}(1-x^n)^3 = \sum_{j\geq 0}(-1)^j(2j+1)x^{\frac{j(j+1)}{2}}.$$

The conclusion now follows by matching coefficients of like powers of x. □

2. Partitions with distinct parts

Let $a_1 + a_2 + \cdots + a_k = n$, with $a_{i+1} > a_i$ for $i \geq 1$, $a_1 \geq 1$ be a partitition of n with distinct parts, called *distinct partitions*. As mentioned above, the ordinary generating function for distinct partitions is

$$Q(x) := \sum_{n\geq 0} q(n)x^n = \prod_{i\geq 1}(1+x^i).$$

To count the number of parts in distinct partitions we can introduce the bivariate generating function

$$Q(x,u) := \sum_{n\geq 0}\sum_{k\geq 1} q(n,k)x^n u^k = \prod_{i\geq 1}(1+ux^i),$$

where $q(n,k)$ counts distinct partitions with exactly k parts.

Let $s_d(n)$ be the total number of parts in all distinct partitions of the natural number n. Then the formula $s_d(n) = \sum_{k=1}^{n} kq(n,k)$ implies that

$$S_d(x) := \sum_{n\geq 0} s_d(n)x^n = \frac{\partial}{\partial u}Q(x,u)|_{u=1} = Q(x)\sum_{k\geq 1}\frac{z^k}{1+z^k} = Q(x)\sum_{m\geq 1} e(m)x^m, \quad (4)$$

where $e(m)$ is the number of odd divisors of m minus the number of even divisors of m.

Note that if $n = 2^k m$, where $k \geq 0$, $2 \not| m$, then $e(n) = -(k-1)d(m)$.

By equating coefficients above we obtain the formula

$$s_d(n) = \sum_{i=1}^{n} e(i)q(n-i).$$

The $s_d(n)$ sequence begins as follows for $n \geq 0$,

$$0, 1, 1, 3, 3, 5, 8, 10, 13, 18, 25, 30, 40, 49, 63, 80, 98, 119, 179, 218, \cdots.$$

In Sloane's online encyclopaedia of integer sequences [3] this is **A015723**.

Theorem 6. *For $n \geq 1$,*

$$s_d(n)+\sum_{i\geq 1}(-1)^i(s_d(n-\omega(i))+s_d(n-\omega(-i)) = e(n)+\sum_{i\geq 1}(-1)^{i-1}(e(n-2\omega(i))+e(n-2\omega(-i)).$$

Proof. We have from (4)

$$S_d(x)\prod_{i\geq 1}(1-x^i) = \prod_{i\geq 1}(1-x^{2i})\sum_{m\geq 1}e(m)x^m.$$

Using the pentagonal number theorem

$$S_d(x)\left(1+\sum_{i\geq 1}(-1)^i(x^{\omega(i)}+x^{\omega(-i)})\right) = \left(1+\sum_{i\geq 1}(-1)^i(x^{2\omega(i)}+x^{2\omega(-i)})\right)\sum_{m\geq 1}e(m)x^m.$$

Equating coefficients of x^n on each side gives the result. □

Theorem 7. *For $n \geq 1$,*

$$\sum_{j=1}^{n/2} s_d(n-2j)p(j) = \sum_{i=1}^{n} e(i)p(n-i).$$

Proof. Again from (4),

$$S_d(x)P(x^2) = P(x)\sum_{m\geq 1}e(m)x^m.$$

Equating coefficients of x^n on each side gives the result. □

Theorem 8. *For $n \geq 1$,*

$$\sum_{i=1}^{n}(-1)^i s_d(i)q(n-i) = \sum_{j=1}^{n/2} e(n-2j)q(j).$$

Proof. If we replace x by $-x$ in (4),

$$S_d(-x) = \sum_{m\geq 1}(-1)^m e(m)x^m\prod_{i\geq 1}(1+x^{2i})\prod_{i\geq 1}(1-x^{2i-1}) \tag{5}$$

or

$$S_d(-x)Q(x) = \sum_{m\geq 1}(-1)^m e(m)x^m Q(x^2).$$

Equating coefficients of x^n on each side gives the result. □

Theorem 9. *For $n \geq 1$,*

$$\sum_{i\geq 0}(-1)^{\frac{i(i+1)}{2}} s_d(n-\frac{i(i+1)}{2}) = e(n) + \sum_{j\geq 1}(-1)^j (e(n-4\omega(i)) + e(n-4\omega(-i))).$$

Proof. In (5), multiply both sides to get

$$S_d(-x)\frac{\prod_{i\geq 1}(1-x^{2i})}{\prod_{i\geq 1}(1-x^{2i-1})} = \sum_{m\geq 1}(-1)^m e(m)x^m \prod_{i\geq 1}(1+x^{2i})\prod_{i\geq 1}(1-x^{2i}).$$

Then

$$S_d(-x)\sum_{i\geq 0} x^{i(i+1)/2} = \sum_{m\geq 1}(-1)^m e(m)x^m \prod_{i\geq 1}(1-x^{4i}).$$

Replace x by $-x$,

$$S_d(x)\sum_{i\geq 0}(-x)^{i(i+1)/2} = \sum_{m\geq 1} e(m)x^m \prod_{i\geq 1}(1-x^{4i}).$$

Equating coefficients of x^n on each side gives the result. □

3. Partitions into distinct odd parts

Let $s_o(n)$ denote the total number of parts in all partitions of the natural number n into distinct odd parts. Let the corresponding generating function be given by:

$$S_o(x) := \sum_{n\geq 0} s_o(n)x^n.$$

Let the generating function for partitions into distinct odd parts be given by:

$$Q_0(x) = \sum_{n\geq 0} q_0(n)x^n = \prod_{i\geq 1}(1+x^{2i-1}).$$

Then, reasoning as in prior sections, we obtain:

$$S_o(x) = Q_0(x)\sum_{k\geq 1}\frac{x^{2k-1}}{1+x^{2k-1}}.$$

Now let

$$T(x) = \sum_{k\geq 1}\frac{x^{2k-1}}{1+x^{2k-1}} = \sum_{k\geq 1} x^{2k-1}\sum_{j\geq 0}(-1)^j (x^{2k-1})^j =$$

$$\sum_{k\geq 1}\sum_{j\geq 0}(-1)^j(x^{2k-1})^{j+1} = \sum_{k\geq 1}\sum_{i\geq 1}(-1)^{i-1}(x^{2k-1})^i.$$

Let $n = (2k-1)i$. Then, reversing the order of summation in the double sum, we have:

$$T(x) = \sum_{r\geq 1}(\sum_{n/i\, odd}(-1)^i)x^n.$$

Let $f(n) = \sum_{n/i\, odd}(-1)^i$. If n is odd, then $f(n) = d(n)$; If $n = 2^k t$ where $k \geq 1$ and t is odd, then $f(n) = -d(t)$. Therefore

$$f(n) = (-1)^{n-1}d(t).$$

Now we have

$$S_o(x) = Q_0(x)T(x), \quad T(x) = \sum_{n\geq 1} f(n)x^n.$$

Matching coefficients of like powers of x, we obtain the identity:

$$s_o(n) = \sum_{j=1}^{n} q_0(n-j)f(j).$$

The first few terms of the $s_o(n)$ sequence for $n \geq 1$ are given below:
$1, 0, 1, 2, 1, 2, 1, 4, 4, 4, 4, 6, 7, 6, 10, 12, 13, 12, 16, 18, 22, 22, 25, 32, 36, 36, 42, 50, 53, 58, \cdots$.
This sequence is not presently in Sloane [3].

Theorem 10. *If $n = 2^k t$ where $k \geq 0$ and t is odd, then*

$$\sum_{j=1}^{n}(-1)^{j-1}q(n-j)s_o(j) = d(t).$$

Proof. Recall that if $n = 2^k t$ where $k \geq 0$ and t is odd, then

$$S_o(x) = Q_0(x)\sum_{n\geq 1} f(n)x^n,$$

where $f(n) = (-1)^{n-1}d(t)$. Therefore

$$S_o(-x) = Q_0(-x)\sum_{n\geq 1}(-1)^n f(n)x^n,$$

that is,

$$S_o(-x) = \prod_{i\geq 1}(1-x^{2i-1})\sum_{n\geq 1} -d(t)x^n.$$

This implies

$$S_o(-x)\prod_{i\geq 1}(1+x^i) = -\sum_{n\geq 1} d(t)x^n,$$

that is

$$\sum_{n\geq 1}(-1)^n s_o(n)x^n \sum_{n\geq 0} q(n)x^n = -\sum_{n\geq 1} d(t)x^n.$$

Multiplying this last equation by -1 and then matching coefficients of like powers of x, we arrive at our conclusion. □

A well known bijection shows that there are the same number of self conjugate partitions of n as there are partitions of n into distinct odd parts. However, the number of parts in a self conjugate partition and its corresponding partitions into distinct odd parts need not be the same. In fact the bijection shows that if n_1 is the largest part in the distinct odd partition then its corresponding self conjugate partition has number of parts equal to $\frac{n_1+1}{2}$.

Thus if $s_c(n)$ denotes the sum of the number of parts in all self conjugate partitions of n, then

$$s_c(n) = \sum_{\lambda\vdash n} \frac{n_1+1}{2},$$

where the sum is over all partitions λ of n into distinct odd parts, and n_1 is the largest part of λ.

We use this to show

Theorem 11.

$$S_c(x) := \sum_{n\geq 1} s_c(n)x^n = \sum_{k\geq 1} kx^{2k-1}\prod_{i=1}^{k-1}(1+x^{2i-1}).$$

Proof. Let $t_{2k-1}(x)$ be the generating function for partitions λ of n into distinct odd parts with largest part $2k-1$. Then

$$t_{2k-1}(x) = x^{2k-1}\prod_{i=1}^{k-1}(1+x^{2i-1}).$$

Thus the generating function for the sum of the largest parts in partitions of n into distinct odd parts is

$$l(x) := \sum_{n\geq 1} l(n)x^n = (\sum_{k\geq 1}(2k-1)t_{2k-1}(x) = \sum_{k\geq 1}(2k-1)x^{2k-1}\prod_{i=1}^{k-1}(1+x^{2i-1}).$$

Now

$$\begin{aligned} s_c(n) = \sum_{\lambda\vdash n}\frac{n_1+1}{2} &= \frac{1}{2}(\sum_{\lambda\vdash n} n_1 + \sum_{\lambda\vdash n} 1) \\ &= \frac{l(n)+q_0(n)}{2}. \end{aligned}$$

Using the fact that $Q_0(x) = \sum_{k\geq 1} t_{2k-1}(x)$ we find that

$$S_c(x) := \frac{1}{2}(l(x) + Q_0(x)) = \sum_{k\geq 1} t_{2k-1}(x) = \sum_{k\geq 1} kx^{2k-1} \prod_{i=1}^{k-1}(1 + x^{2i-1}),$$

as claimed.

□

The $s_c(n)$ sequence begins as follows for $n \geq 0$,

$$0, 1, 0, 2, 2, 3, 3, 4, 7, 8, 9, 10, 15, 16, 18, 23, 30, 32, 35, 42, 51, 59, 63, \cdots .$$

In Sloane's online encyclopaedia of integer sequences [3] this is **A067619**.

By analogy with partitions into distinct odd parts, it is natural to try express

$$S_c(x) = Q_0(x)R(x), \quad R(x) := \sum_{n\geq 1} r(n)x^n.$$

However the integer coefficients $r(n)$ that appear do not seem to follow any discernable pattern. The sequence of $r(n)$ values for $n \geq 1$ begins as follows:

$$1, -1, 3, -2, 5, -5, 8, -7, 13, -13, 18, -19, 26, -29, 39, -40, 52, -60, 72, -81, 101, \cdots .$$

References

[1] G. E. Andrews, *The theory of partitions*, Encyclopaedia of mathematics and its applications, **2**, Addison-Wesley, 1976.

[2] P. Erdös and J. Lehner, The distribution of the number of summands in the partitions of a positive integer, *Duke Mathematical Journal*, **8**, 335–345, 1941.

[3] N.J.A. Sloane and S. Plouffe, *The encyclopaedia of integer sequences*, Academic Press, 1995. Online edition available at *http://www.research.att.com/∼njas/sequences/*.

Arnold Knopfmacher, The John Knopfmacher Centre for Applicable Analysis and Number Theory, University of the Witwatersrand, P. O. Wits, 2050 Johannesburg, South Africa

E-mail address: arnoldk@cam.wits.ac.za

URL: http://www.wits.ac.za/science/number_theory/arnold.htm

Neville Robbins, Mathematics Department, San Francisco State University, CA 94132, U.S.A.

E-mail address: robbins@math.sfsu.edu

The Ramanujan Journal, , 1–15 ()

A Class of Series Acceleration Formulae for Catalan's Constant

DAVID M. BRADLEY bradley@gauss.umemat.maine.edu
University of Maine, Department of Mathematics & Statistics, 5752 Neville Hall, Orono, ME 04469-5752

Editor: Jonathan M. Borwein

Abstract. In this note, we develop transformation formulae and expansions for the log tangent integral, which are then used to derive series acceleration formulae for certain values of Dirichlet L-functions, such as Catalan's constant. The formulae are characterized by the presence of an infinite series whose general term consists of a linear recurrence damped by the central binomial coefficient and a certain quadratic polynomial. Typically, the series can be expressed in closed form as a rational linear combination of Catalan's constant and π times the logarithm of an algebraic unit.

Keywords: log tangent integral, central binomial coefficient, algebraic unit, Catalan's constant.

1991 Mathematics Subject Classification: Primary 11Y60, Secondary 11M06

1. Introduction

Catalan's constant may be defined by means of [1]

$$G := \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)^2} = L(2, \chi_4), \tag{1}$$

where χ_4 is the non-principal Dirichlet character modulo 4. It is currently unknown whether or not G is rational.

The purpose of this note is to develop and classify acceleration formulae for slowly convergent series such as (1), based on transformations of the log tangent integral. The simplest acceleration formula of its type that we wish to consider is

$$G = \frac{\pi}{8}\log(2+\sqrt{3}) + \frac{3}{8}\sum_{k=0}^{\infty} \frac{1}{(2k+1)^2\binom{2k}{k}}, \tag{2}$$

due to Ramanujan [4, 14]. We shall see that Ramanujan's formula (2) is the first of an infinite family of series acceleration formulae for G, each of which is characterized by the presence of an infinite series whose general term consists of a linear recurrence damped by the summand in (2). In each case, the series evaluates to a rational linear combination of G and π times the logarithm of an algebraic unit (i.e. an invertible algebraic integer). Perhaps the most striking example of this phenomenon is

$$G = \frac{\pi}{8}\log\left(\frac{10+\sqrt{50-22\sqrt{5}}}{10-\sqrt{50-22\sqrt{5}}}\right) + \frac{5}{8}\sum_{k=0}^{\infty} \frac{L(2k+1)}{(2k+1)^2\binom{2k}{k}}, \tag{3}$$

where $L(1) = 1$, $L(2) = 3$, and $L(n) = L(n-1) + L(n-2)$ for $n > 2$ are the Lucas numbers [11], (M2341 in [15]).

We shall see that series acceleration results such as (2) and (3) have natural explanations when viewed as consequences of transformation formulae for the log tangent integral, although we should remark that Ramanujan apparently derived his result (2) by quite different methods. The connection with log tangent integrals is best explained by the equation

$$G = -\int_0^{\pi/4} \log(\tan\theta)\, d\theta, \tag{4}$$

obtained by expanding the integrand into its Fourier cosine series and integrating term by term. It will be shown that Ramanujan's result (2) arises from the transformation

$$2\int_0^{\pi/4} \log(\tan\theta)\, d\theta = 3\int_0^{\pi/12} \log(\tan\theta)\, d\theta. \tag{5}$$

The roccoco formula (3) arises in a similar manner from the transformation

$$2\int_0^{\pi/4} \log(\tan\theta)\, d\theta = 5\int_0^{3\pi/20} \log(\tan\theta)\, d\theta - 5\int_0^{\pi/20} \log(\tan\theta)\, d\theta. \tag{6}$$

Heuristically, one expects such transformations to succeed because the reduced range of integration on the right hand side, when re-expanded into a series, provides a continuous analog of bunching together many terms of the original series.

2. The Log Tangent Integral

There is a limitless supply of transformation formulae for the log tangent integral. In subsection 2.2, an infinite family of linear relations, of which both (5) and (6) are members, will be derived. These relations will be used in conjunction with the series expansions given in subsection 2.1 to develop a corresponding infinite family of series acceleration formulae which includes both (2) and (3) as special cases.

2.1. Series Expansions

We shall be concerned with only two series expansions for the log tangent integral. These are given in Theorems 1 and 2 below.

THEOREM 1. *For* $0 \le x \le \frac{1}{2}\pi$,

$$\int_0^x \log(\tan\theta)\, d\theta = -\sum_{k=0}^{\infty} \frac{\sin((4k+2)x)}{(2k+1)^2}.$$

Proof: Expand the integrand into its Fourier cosine series. Integrating term by term is justified by the fact that the Fourier series is boundedly convergent on compact subintervals of $(0, \frac{1}{2}\pi]$. ∎

For us, the significance of Theorem 1 derives mostly from the specialization $x = \frac{1}{4}\pi$, which yields the relationship (4) between Catalan's constant and the log tangent integral. On the other hand, the expansion in powers of sines provided by Theorem 2 below is more widely applicable.

THEOREM 2. *For* $0 \le x \le \frac{1}{4}\pi$,

$$\int_0^x \log(\tan\theta)\,d\theta = x\log(\tan x) - \frac{1}{4}\sum_{k=0}^{\infty} \frac{(2\sin 2x)^{2k+1}}{(2k+1)^2\binom{2k}{k}}.$$

Proof: First integrate by parts, rescale, and use the double angle formula for sine:

$$\begin{aligned}
\int_0^x \log(\tan\theta)\,d\theta - x\log(\tan x) &= -\int_0^x \frac{\theta\sec^2\theta}{\tan\theta}\,d\theta \\
&= -\int_0^{2x} \frac{\theta\,d\theta}{4\tan(\frac{1}{2}\theta)\cos^2(\frac{1}{2}\theta)} \\
&= -\int_0^{2x} \frac{\theta\,d\theta}{2\sin\theta} \\
&= -\int_0^{\sin(2x)} \frac{2t\sin^{-1}t}{\sqrt{1-t^2}}\cdot\frac{dt}{4t^2}.
\end{aligned}$$

Now employ the power series expansion [6]

$$\frac{2t\sin^{-1}t}{\sqrt{1-t^2}} = \sum_{k=1}^{\infty}\frac{(2t)^{2k}}{k\binom{2k}{k}}, \qquad |t|<1,$$

and integrate term by term. The result follows. ∎

In addition to Theorem 1, the following representations were also more or less known to Ramanujan, and can be easily verified by differentiation:

$$\begin{aligned}
\int_0^x \log(\tan\theta)\,d\theta &= x\log(\tan x) + \sum_{k=0}^{\infty}\frac{(-1)^{k+1}(\tan x)^{2k+1}}{(2k+1)^2}, \quad 0\le x\le\tfrac{1}{4}\pi, \\
\int_0^x \log(\tan\theta)\,d\theta &= (\tfrac{1}{2}\pi - x)\log(\cos x) - \sum_{k=1}^{\infty}\frac{(\cos x)^k(\sin kx)}{k^2}, \quad 0\le x\le\tfrac{1}{2}\pi, \\
\int_0^x \log(\tan\theta)\,d\theta &= x\log(\tan x) + \tfrac{1}{2}\pi\log(2\cos x) \\
&\quad - \sum_{k=0}^{\infty}\binom{2k}{k}\frac{(\cos x)^{2k+1}+(\sin x)^{2k+1}}{4^k(2k+1)^2}, \quad 0\le x\le\tfrac{1}{2}\pi.
\end{aligned}$$

2.2. Transformation Formulae

It will be convenient to define

$$T(r) := \int_0^{r\pi} \log(\tan\theta)\, d\theta, \qquad 0 \le r \le \tfrac{1}{2}. \tag{7}$$

Our development will provide two distinct transformation formulae for the T-function: the multiplication formula, which expresses T at odd multiples of the argument in terms of a multitude of other T-values; and the reflection formula, which makes it possible to restrict the domain to the interval $0 \le r \le \frac{1}{4}$, and which will effect a number of simplifications in our intermediate calculations, as we shall see.

THEOREM 3. *For all $0 \le r \le \frac{1}{2}$, the reflection formula*

$$T(r) = T(\tfrac{1}{2} - r)$$

holds.

Proof: First, note that $T(\frac{1}{2}) = 0$. This can be seen either by putting $x = \frac{1}{2}\pi$ in Theorem 1, or by observing that

$$T(\tfrac{1}{2}) = \int_0^{\pi/2} \log(\tan\theta)\, d\theta = \int_0^{\pi/2} \log(\sin\theta)\, d\theta - \int_0^{\pi/2} \log\sin(\tfrac{1}{2}\pi - \theta)\, d\theta = 0.$$

It follows that

$$\begin{aligned} T(r) &= \int_0^{r\pi} \log(\tan\theta)\, d\theta = \int_0^{\pi/2} \log(\tan\theta)\, d\theta - \int_{r\pi}^{\pi/2} \log(\tan\theta)\, d\theta \\ &= \int_{\pi/2 - r\pi}^{0} \log(\tan(\tfrac{1}{2}\pi - \theta))\, d\theta \\ &= \int_{\pi/2 - r\pi}^{0} \log(\cot\theta)\, d\theta \\ &= \int_0^{(1/2 - r)\pi} \log(\tan\theta)\, d\theta \\ &= T(\tfrac{1}{2} - r), \end{aligned}$$

as stated. ∎

To prove the multiplication formula, we require the following product expansion for the tangent function.

LEMMA 1. *Let $m = 2n + 1$ be an odd positive integer, and let $x \in \mathbf{R}$. Then*

$$\frac{\tan(mx)}{\tan(x)} = \prod_{j=1}^{n} \tan\left(\frac{j\pi}{m} + x\right) \tan\left(\frac{j\pi}{m} - x\right).$$

Proof: Let $w = e^{ix}$. Then

$$\begin{aligned}
\frac{\tan(mx)}{\tan(x)} &= \left(\frac{w^{2m}-1}{w^{2m}+1}\right)\left(\frac{w^2+1}{w^2-1}\right) \\
&= \prod_{k=1}^{n}\left(\frac{w^2-e^{2k\pi i/m}}{w^2-e^{(2k-1)\pi i/m}}\right)\left(\frac{w^2-e^{-2k\pi i/m}}{w^2-e^{-(2k-1)\pi i/m}}\right) \\
&= \prod_{k=1}^{n}\left(\frac{we^{-k\pi i/m}-w^{-1}e^{k\pi i/m}}{we^{-(2k-1)\pi i/2m}-w^{-1}e^{(2k-1)\pi i/2m}}\right) \\
&\qquad \times\left(\frac{we^{k\pi i/m}-w^{-1}e^{-k\pi i/m}}{we^{(2k-1)\pi i/2m}-w^{-1}e^{-(2k-1)\pi i/2m}}\right) \\
&= \prod_{k=1}^{n}\frac{\sin(k\pi/m-x)\sin(k\pi/m+x)}{\sin((2k-1)\pi/2m-x)\sin((2k-1)\pi/2m+x)}.
\end{aligned}$$

After expressing the sines in the denominator in terms of cosines and letting $j = n-k+1$, we have

$$\begin{aligned}
\frac{\tan(mx)}{\tan(x)} &= \prod_{j=1}^{n}\frac{\sin(j\pi/m-x)\sin(j\pi/m+x)}{\cos(j\pi/m-x)\cos(j\pi/m+x)} \\
&= \prod_{j=1}^{n}\tan\left(\frac{j\pi}{m}+x\right)\tan\left(\frac{j\pi}{m}-x\right)
\end{aligned}$$

as required. ■

THEOREM 4. *Let $m = 2n+1$ be an odd positive integer, and let $0 \le r \le 1/(2m)$. Then the multiplication formula*

$$T(mr) = m\sum_{j=0}^{n}T\left(\frac{j}{m}+r\right) - m\sum_{j=1}^{n}T\left(\frac{j}{m}-r\right)$$

holds.

Proof: By Lemma 1,

$$\begin{aligned}
T(mr) &= \int_0^{mr\pi}\log(\tan\theta)\,d\theta = m\int_0^{r\pi}\log(\tan(mx))\,dx \\
&= m\int_0^{r\pi}\log(\tan x)\,dx + m\sum_{j=1}^{n}\int_0^{r\pi}\log\tan\left(\frac{j\pi}{m}+x\right)\,dx \\
&\quad + m\sum_{j=1}^{n}\int_0^{r\pi}\log\tan\left(\frac{j\pi}{m}-x\right)\,dx \\
&= mT(r) + m\sum_{j=1}^{n}\left\{T\left(\frac{j}{m}+r\right) - T\left(\frac{j}{m}\right)\right\}
\end{aligned}$$

$$\begin{aligned} & -m\sum_{j=1}^{n}\left\{T\left(\frac{j}{m}-r\right)-T\left(\frac{j}{m}\right)\right\} \\ = \; & m\sum_{j=0}^{n}T\left(\frac{j}{m}+r\right)-m\sum_{j=1}^{n}T\left(\frac{j}{m}-r\right), \end{aligned}$$

as stated. ■

To obtain transformations such as (5) and (6), we apply the reflection formula (Theorem 3) and the multiplication formula (Theorem 4) with r chosen so as to express $T(\frac{1}{4})$ in terms of the T-function at values of the argument less than $\frac{1}{4}$. The resulting transformations are distinguished according to the parity of n in the multiplier $m = 2n + 1$.

THEOREM 5. *Let n be an odd positive integer. Then*

$$G = -T\left(\frac{1}{4}\right) = \frac{2n+1}{n+1}\sum_{j=1}^{n}(-1)^j T\left(\frac{2j-1}{8n+4}\right).$$

Proof: Let p be a nonnegative integer. In the multiplication formula, let $n = 2p + 1$, so that $m = 4p + 3$, and put $r = 1/(4m)$. Then

$$\begin{aligned} T\left(\frac{1}{4}\right) \; = \; & m\sum_{j=0}^{p}\left\{T\left(\frac{4j+1}{4m}\right)+T\left(\frac{4(n-j)+1}{4m}\right)\right\} \\ & -m\sum_{j=1}^{p}\left\{T\left(\frac{4j-1}{4m}\right)+T\left(\frac{4(n-j+1)-1}{4m}\right)\right\} \\ & -mT\left(\frac{4(p+1)-1}{4m}\right). \end{aligned}$$

Applying the reflection formula (Theorem 3) to each term in the preceding sums yields the simplification

$$T\left(\frac{1}{4}\right) = 2m\sum_{j=0}^{p}T\left(\frac{4j+1}{4m}\right) - 2m\sum_{j=1}^{p}T\left(\frac{4j-1}{4m}\right) - mT\left(\frac{1}{4}\right).$$

The preceding expression can be simplified further by combining the two sums into a single alternating sum. Thus,

$$-T\left(\frac{1}{4}\right) = \frac{2m}{m+1}\sum_{j=1}^{2p+1}(-1)^j T\left(\frac{2j-1}{4m}\right).$$

Writing p and m in terms of n completes the proof. ■

Theorem 6 below addresses the alternative case in which the multiplier is congruent to 1 modulo 4.

THEOREM 6. *Let n be an even positive integer. Then*

$$G = -T\left(\frac{1}{4}\right) = \frac{2n+1}{n}\sum_{j=1}^{n}(-1)^{j+1}T\left(\frac{2j-1}{8n+4}\right).$$

Proof: Let p be a nonnegative integer. In the multiplication formula let $n = 2p$, so that $m = 4p+1$, and again put $r = 1/(4m)$. Then

$$\begin{aligned} T\left(\frac{1}{4}\right) \;=\; & m\sum_{j=0}^{p-1}\left\{T\left(\frac{4j+1}{4m}\right)+T\left(\frac{4(n-j)+1}{4m}\right)\right\} \\ & -m\sum_{j=1}^{p}\left\{T\left(\frac{4j-1}{4m}\right)+T\left(\frac{4(n-j+1)-1}{4m}\right)\right\} \\ & +mT\left(\frac{4p+1}{4m}\right). \end{aligned}$$

Applying the reflection formula (Theorem 3) to each term in the preceding sums yields the simplification

$$T\left(\frac{1}{4}\right) = 2m\sum_{j=0}^{p-1}T\left(\frac{4j+1}{4m}\right) - 2m\sum_{j=1}^{p}T\left(\frac{4j-1}{4m}\right) + mT\left(\frac{1}{4}\right).$$

The preceding expression can be simplified further by combining the two sums into a single alternating sum. Thus,

$$-T\left(\frac{1}{4}\right) = \frac{2m}{m-1}\sum_{j=1}^{2p}(-1)^{j+1}T\left(\frac{2j-1}{4m}\right),$$

Writing p and m in terms of n completes the proof. ■

EXAMPLE: Putting $n = 1$ in Theorem 5 yields the transformation $2T(\frac{1}{4}) = 3T(\frac{1}{12})$, which is a restatement of (5). Putting $n = 2$ in Theorem 6 yields the transformation $2T(\frac{1}{4}) = 5T(\frac{3}{20}) - 5T(\frac{1}{20})$, which is (6). □

3. Applications to Series Acceleration

3.1. Catalan's Constant

THEOREM 7. *Let n be an odd positive integer. For nonnegative integers k, define a sequence*

$$F_n(k) := \sum_{j=1}^{n}\left((-1)^{n-j+1}2\cos\left(\frac{j\pi}{2n+1}\right)\right)^k,$$

and let

$$u_n := \prod_{j=1}^{n} \left(\tan\left(\frac{2j-1}{8n+4} \right) \pi \right)^{(2j-1)(-1)^j}.$$

Then u_n is a unit algebraic integer, and Catalan's constant has the series acceleration formula

$$G = \left(\frac{\pi}{4n+4} \right) \log u_n - \left(\frac{2n+1}{4n+4} \right) \sum_{k=0}^{\infty} \frac{F_n(2k+1)}{(2k+1)^2 \binom{2k}{k}}.$$

Proof: Apply Theorems 1 and 2 to the right hand side of Theorem 5. Thus,

$$\begin{aligned} G &= \left(\frac{2n+1}{n+1} \right) \sum_{j=1}^{n} (-1)^j \left(\frac{2j-1}{8n+4} \right) \pi \log \left(\tan\left(\frac{2j-1}{8n+4} \right) \pi \right) \\ &\quad - \left(\frac{2n+1}{4n+4} \right) \sum_{j=1}^{n} (-1)^j \sum_{k=0}^{\infty} \frac{\left(2 \sin((2j-1)\pi/(4n+2)) \right)^{2k+1}}{(2k+1)^2 \binom{2k}{k}} \\ &= \frac{\pi}{4} \sum_{j=1}^{n} (-1)^j \left(\frac{2j-1}{n+1} \right) \log \left(\tan\left(\frac{2j-1}{8n+4} \right) \pi \right) \\ &\quad - \left(\frac{2n+1}{4n+4} \right) \sum_{k=0}^{\infty} \frac{1}{(2k+1)^2 \binom{2k}{k}} \sum_{j=1}^{n} (-1)^j \left(2 \sin\left(\frac{2j-1}{4n+2} \right) \pi \right)^{2k+1}. \end{aligned} \tag{8}$$

The inner sum in (8) simplifies somewhat if the sines are expressed in terms of cosines. Thus,

$$\begin{aligned} &\sum_{j=1}^{n} (-1)^j \left(2 \sin\left(\frac{2j-1}{4n+2} \right) \pi \right)^{2k+1} \\ &= \sum_{j=1}^{n} (-1)^j \left(2 \cos\left(\frac{2n+1-(2j-1)}{4n+2} \right) \pi \right)^{2k+1} \\ &= \sum_{j=1}^{n} (-1)^j \left(2 \cos\left(\frac{n-j+1}{2n+1} \right) \pi \right)^{2k+1} \\ &= \sum_{j=1}^{n} (-1)^{n-j+1} \left(2 \cos\left(\frac{j\pi}{2n+1} \right) \right)^{2k+1}. \end{aligned} \tag{9}$$

Substituting (9) into (8) completes the derivation of the stated formula.

It now remains to show that u_n is indeed an algebraic unit (i.e. an invertible algebraic integer) as claimed. Let $x = (2j-1)\pi/(8n+4)$. Since the units in any ring form a multiplicative group, it suffices to show that the numbers $t := \tan x$ are all algebraic units, or equivalently, that the numbers t satisfy monic polynomials with integer coefficients and constant term ± 1.

From the addition formula for the tangent function, one sees that $\tan(kx)$ is a rational function of t for each nonnegative integer k. Indeed, if polynomials $p_k, q_k \in \mathbf{Z}[t]$ are defined by the recursion

$$\begin{pmatrix} p_{k+1} \\ q_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & t \\ -t & 1 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix}, \quad \text{for} \quad k \geq 0, \quad \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{10}$$

then for all nonnegative integers k,

$$\tan((k+1)x) = \frac{\tan x + \tan(kx)}{1 - \tan(x)\tan(kx)} = \frac{tq_k + p_k}{q_k - tp_k} = \frac{p_{k+1}}{q_{k+1}}.$$

Since $\tan((2n+1)x) = \tan((2j-1)\pi/4) = (-1)^{j+1}$, it follows that $t = \tan((2j-1)\pi/(8n+4))$ satisfies the polynomial equation

$$p_{2n+1}(t) \pm q_{2n+1}(t) = 0.$$

It remains to show that $p_{2n+1} \pm q_{2n+1}$ has both highest degree coefficient and constant coefficient equal to ± 1.

Let k be an odd positive integer. From the recursion (10), it follows that

$$p_{k+2} + q_{k+2} = (1 - 2t - t^2)p_k + (1 + 2t - t^2)q_k, \tag{11}$$

$$p_{k+2} - q_{k+2} = (1 + 2t - t^2)p_k - (1 - 2t - t^2)q_k. \tag{12}$$

An easy induction shows that the respective degrees of p_k and q_k are k and $k-1$, for all odd positive integers k. This fact, combined with a second induction, shows that the highest degree coefficient of $p_k \pm q_k$ is equal to ± 1 for all odd positive integers k. Finally, (11) and (12) show that

$$p_{k+2}(0) \pm q_{k+2}(0) = p_k(0) \pm q_k(0)$$

and so a final induction proves that $p_k \pm q_k$ has constant coefficient equal to ± 1 for all odd positive integers k. ■

Remark. Suppose n is fixed, and we partition the algebraic numbers

$$(-1)^{n-j+1} 2\cos\left(\frac{j\pi}{2n+1}\right)$$

into disjoint sets of mutual conjugates. Then the product of the minimum polynomials for each set of conjugates is precisely the characteristic polynomial of the linear recurrence satisfied by the sequence $\{F_n(k)\}_{k=0}^{\infty}$.

EXAMPLE: Putting $n = 1$ in Theorem 7 gives

$$G = -\frac{\pi}{8}\log\left(\tan\left(\frac{\pi}{12}\right)\right) + \frac{3}{8}\sum_{k=0}^{\infty} \frac{(2\cos(\pi/3))^{2k+1}}{(2k+1)^2\binom{2k}{k}},$$

which is Ramanujan's formula (2). □

Theorem 7 has its even counterpart in Theorem 8 below.

THEOREM 8. *Let n be an even positive integer. For nonnegative integers k, define a sequence*

$$F_n(k) := \sum_{j=1}^{n} \left((-1)^j 2 \cos\left(\frac{j\pi}{2n+1} \right) \right)^k,$$

and let

$$u_n := \prod_{j=1}^{n} \left(\tan\left(\frac{2j-1}{8n+4} \right) \pi \right)^{(2j-1)(-1)^{j+1}}.$$

Then u_n is a unit algebraic integer, and Catalan's constant has the series acceleration formula

$$G = \left(\frac{\pi}{4n} \right) \log u_n - \left(\frac{2n+1}{4n} \right) \sum_{k=0}^{\infty} \frac{F_n(2k+1)}{(2k+1)^2 \binom{2k}{k}}.$$

We omit the proof of Theorem 8, as it closely mimicks the proof of Theorem 7. Instead, we derive the formula (3) which relates Catalan's constant and the Lucas sequence.

COROLLARY. *Let $L(1) = 1$, $L(2) = 3$, and $L(n) = L(n-1) + L(n-2)$ for $n > 2$ be the Lucas numbers. Then Catalan's constant has the series acceleration formula*

$$G = \frac{\pi}{8} \log \left(\frac{10 + \sqrt{50 - 22\sqrt{5}}}{10 - \sqrt{50 - 22\sqrt{5}}} \right) + \frac{5}{8} \sum_{k=0}^{\infty} \frac{L(2k+1)}{(2k+1)^2 \binom{2k}{k}}.$$

Proof: Put $n = 2$ in Theorem 8. Letting $\phi := 2\cos(2\pi/5) = \frac{1}{2}(\sqrt{5} - 1)$ and $\tau := 2\cos(\pi/5) = \frac{1}{2}(\sqrt{5} + 1)$, we have

$$G = \frac{\pi}{8} \log \left(\frac{\tan(\pi/20)}{\tan^3(3\pi/20)} \right) - \frac{5}{8} \sum_{k=0}^{\infty} \frac{\phi^{2k+1} - \tau^{2k+1}}{(2k+1)^2 \binom{2k}{k}}.$$

Now recall [11] that

$$L(k) = \left(\frac{1+\sqrt{5}}{2} \right)^k + \left(\frac{1-\sqrt{5}}{2} \right)^k$$

for all nonnegative integers k. It follows that

$$G = \frac{\pi}{8} \log \left(\frac{\tan(\pi/20)}{\tan^3(3\pi/20)} \right) + \frac{5}{8} \sum_{k=0}^{\infty} \frac{L(2k+1)}{(2k+1)^2 \binom{2k}{k}},$$

and so it remains only to verify the non-trivial denesting relationship

$$\frac{\tan(\pi/20)}{\tan^3(3\pi/20)} = \frac{10+\sqrt{50-22\sqrt{5}}}{10-\sqrt{50-22\sqrt{5}}}. \tag{13}$$

To express the tangent values in (13) in terms of radicals, we follow [13], p. 50. Let $t := \tan(\pi/20)$. Then

$$\tan\frac{3\pi}{20} = \frac{3t-t^3}{1-3t^2} = \tan\left(\frac{\pi}{4}-\frac{2\pi}{20}\right) = \frac{1-2t/(1-t^2)}{1+2t/(1-t^2)}.$$

Equating the previous rational expressions in t gives the quintic equation

$$(t-1)^5 = 20t^2(t-1), \quad \text{or} \quad (t-1)^2 = 2t\sqrt{5},$$

since $t \neq 1$. Putting $t = (1-\varepsilon)/(1+\varepsilon)$, it follows that $\varepsilon\sqrt{5+2\sqrt{5}} = \sqrt{5}$, and

$$\tan\frac{\pi}{20} = \frac{\sqrt{5+2\sqrt{5}}-\sqrt{5}}{\sqrt{5+2\sqrt{5}}+\sqrt{5}}, \quad \tan\frac{3\pi}{20} = \frac{\sqrt{5+2\sqrt{5}}-1}{\sqrt{5+2\sqrt{5}}+1}.$$

Therefore, we may write

$$\frac{\tan(\pi/20)}{\tan^3(3\pi/20)} = \left(\frac{\sqrt{5+2\sqrt{5}}+1}{\sqrt{5+2\sqrt{5}}-1}\right)^3 \frac{\sqrt{5+2\sqrt{5}}-\sqrt{5}}{\sqrt{5+2\sqrt{5}}+\sqrt{5}} = \frac{a+b}{a-b}, \tag{14}$$

where a and b are to be determined. Cross multiplying and expanding both sides, we have

$$5b(3+\sqrt{5}) = a(3-\sqrt{5})\sqrt{5+2\sqrt{5}}. \tag{15}$$

Since $(3-\sqrt{5})/(3+\sqrt{5}) = \frac{1}{2}(7-3\sqrt{5})$, we may write (15) in the form

$$10b = a\sqrt{(7-3\sqrt{5})^2(5+2\sqrt{5})} = a\sqrt{50-22\sqrt{5}}.$$

Therefore, if in (14), we take $a = 10$ and $b = \sqrt{50-22\sqrt{5}}$, then (13) holds, and the proof is complete. ■

Remark. It is unlikely that (13) will simplify any further. Zippel [16] gives two formulae (caution: there are misprints) for denesting expressions involving square roots. Borodin et. al. [5] show that these are the only two ways that such expressions can be denested over the rational number field. In particular, $\sqrt{50-22\sqrt{5}}$ cannot be denested, because $50^2-5\times22^2 = 80$ and $22^2\times5^2-50^2\times5 = -400$ are not squares of rational numbers.

3.2. Some Additional Examples

One can derive additional acceleration formulae by specializing the value of x in Theorems 1 and 2 and equating the two results. In general, convergence improves as the value of x decreases. The following selections provide a representative sample of perhaps the most interesting results that can obtained using this approach.

EXAMPLE: Putting $x = \frac{1}{4}\pi$ gives (cf. (1))

$$G = L(2, \chi_4) = \frac{1}{2}\sum_{k=0}^{\infty} \frac{4^k}{(2k+1)^2\binom{2k}{k}}, \tag{16}$$

which Ramanujan [4] derived previously by other methods. We remark that (16) is actually a series *deceleration* result. The reason for the poor convergence is we have used the trivial transformation $T(\frac{1}{4}) = T(\frac{1}{4})$ which fails to exploit the reduced range of integration present in the other transformations. □

EXAMPLE: Putting $x = \frac{1}{6}\pi$ gives

$$L(2, \chi_6) = \frac{\pi\sqrt{3}}{18}\log 3 + \frac{1}{2}\sum_{k=0}^{\infty} \frac{3^k}{(2k+1)^2\binom{2k}{k}},$$

where χ_6 is the non-principal Dirichlet character modulo 6 (i.e. $\chi_6(5) = -1$). □

EXAMPLE: Putting $x = \frac{1}{8}\pi$ gives

$$L(2, \chi_8) = \frac{\pi\sqrt{2}}{8}\log(1+\sqrt{2}) + \frac{1}{2}\sum_{k=0}^{\infty} \frac{2^k}{(2k+1)^2\binom{2k}{k}},$$

where χ_8 is the Dirichlet character modulo 8 given by $\chi_8(1) = \chi_8(3) = 1$, and $\chi_8(5) = \chi_8(7) = -1$. □

Acknowledgments

I'm grateful to Chris Hill, Jonathan Borwein, Petr Lisoněk, and John Zucker for their helpful observations.

Appendix

Here, we outline the role that inverse symbolic computation – in particular, Maple's integer relations algorithms – played in the discovery process.

A vector $\vec{v} = (v_1, v_2, \ldots, v_n)$ of real numbers is said to possess an integer relation if there exists a vector $\vec{a} = (a_1, a_2, \ldots, a_n)$ of integers not all zero such that the scalar product vanishes, i.e. $a_1v_1 + a_2v_2 + \cdots + a_nv_n = 0$. In the past two decades, several algorithms which recover $\vec{a}$ given $\vec{v}$ have been discovered [2, 3, 9, 10, 12]. One of these, "LLL" [12], has been implemented in Maple V, and with its help, the authors of [7] and [8] discovered new formulae for values of the Riemann Zeta function. The obstacle which initially confounded efforts to extend the classical results

$$\zeta(2) = 3\sum_{k=1}^{\infty} \frac{1}{k^2\binom{2k}{k}}, \quad \zeta(3) = \frac{5}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^3\binom{2k}{k}}, \quad \zeta(4) = \frac{36}{17}\sum_{k=1}^{\infty} \frac{1}{k^4\binom{2k}{k}}$$

to higher zeta values was circumvented by the introduction of harmonic sums into the search space. Thus, for example, by searching for an identity of the form

$$\zeta(7) = r_1 \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^7 \binom{2k}{k}} + r_2 \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^5 \binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^2} + r_3 \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3 \binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^4},$$

we [7] found

$$\zeta(7) = \frac{5}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^7 \binom{2k}{k}} + \frac{25}{2} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3 \binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^4},$$

and infinitely many more, as well as some lovely integral and hypergeometric series evaluations, besides.

We suspected that a similar reverse-engineered approach might work for certain Dirichlet L-series values, such as Catalan's constant, but searching for similar variations on Ramanujan's example (2) failed. In view of the ornate complexity of (3) and its relatives (Theorem 7 and Theorem 8), we can now understand the reason for this failure. For a direct attack, one would have had to introduce, among other things, logarithms of algebraic units into the model, so that in effect, one would have needed to know beforehand the formula one was searching for in order to find it. Models based on the inverse tangent integral [13, 14] suffer the same drawbacks. On the other hand, the model based on the log tangent integral is suited perfectly.

The author arrived at the log tangent integral model while attempting to give an alternative proof of Ramanujan's acceleration formula (2). It was found that the proof reduced to that of proving the integral transformation (5). Isolating the T-function of section 3 for study was then a natural choice. After directing Maple's integer relations finding algorithms to hunt for linear relations amongst various T-values, the following list was produced:

$$\begin{aligned}
T(1/2) &= 0, && \text{(A.1)}\\
T(1/3) &= T(1/6), && \text{(A.2)}\\
T(1/8) &= T(3/8), && \text{(A.3)}\\
3T(4/9) &= T(1/3) + T(2/9) - 3T(1/9), && \text{(A.4)}\\
T(2/10) &= T(3/10), && \text{(A.5)}\\
T(1/10) &= T(2/5), && \text{(A.6)}\\
T(1/12) &= T(5/12), && \text{(A.7)}\\
2T(1/4) &= 3T(1/12), && \text{(A.8)}\\
T(3/14) &= T(4/14), && \text{(A.9)}\\
T(5/14) &= T(1/7), && \text{(A.10)}\\
T(1/14) &= T(3/7), && \text{(A.11)}\\
3T(2/5) &= -3T(1/15) + T(1/5) + 3T(4/15), && \text{(A.12)}\\
3T(7/15) &= -3T(2/15) + 3T(1/5) + T(2/5), && \text{(A.13)}\\
15T(1/15) &= 15T(2/15) - 5T(1/5) + 9T(1/3) - 10T(2/5), && \text{(A.14)}
\end{aligned}$$

$$T(3/16) = T(5/16), \tag{A.15}$$
$$T(1/16) = T(7/16), \tag{A.16}$$
$$T(2/9) = T(5/18), \tag{A.17}$$
$$T(1/9) = T(7/18), \tag{A.18}$$
$$T(1/18) = T(4/9), \tag{A.19}$$
$$3T(1/18) = 3T(5/18) + T(1/3) - 3T(7/18), \tag{A.20}$$
$$T(3/20) = T(7/20), \tag{A.21}$$
$$T(1/20) = T(9/20), \tag{A.22}$$
$$5T(3/20) = 5T(1/20) + 2T(1/4) = 5T(7/20). \tag{A.23}$$

Aside from trivial substitutions arising from the reflection formula (Theorem 3), the list evidently exhausts all linear relations amongst T-values with rational arguments having denominator no greater than 20. In fact, each list entry is a consequence of the reflection formula and the multiplication formula (Theorem 4). For example, (A.4) follows from the multiplication formula with $m = 3$ and $r = 1/9$. The slightly trickier (A.14) follows from three applications of the multiplication formula. One takes $m = 3$ with $r = 1/15$ and $r = 2/15$, and then one takes $m = 5$ with $r = 1/15$. This gives three equations. Multiplying the first through by $5/2$, the second through by $-5/2$, and the third through by $3/2$ and adding the three resulting equations gives (A.14).

From the list, it was easy to deduce and subsequently prove the reflection formula. At the same time, Chris Hill of the University of Illinois used the $m = 3$ case of Lemma 1 to prove (5) i.e. (A.8). This broke the dam, leading to the proof of Lemma 1, the multiplication formula (Theorem 4), and the remaining results of sections 2 and 3.

References

1. M. Abramowitz & I. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1972, p. 807.
2. D. H. Bailey & H. R. P. Ferguson, "Numerical Results on Relations Between Numerical Constants Using a New Algorithm," *Mathematics of Computation*, Vol. 53 (October 1989), pp. 649–656.
3. D. H. Bailey & H. R. P. Ferguson, "A Polynomial Time, Numerically Stable Integer Relation Algorithm," *RNR Technical Report*, RNR-91-032.
4. B. C. Berndt, *Ramanujan's Notebooks: Part I*, Springer-Verlag, 1985, p. 289.
5. A. Borodin, R. Fagin, J. E. Hopcroft, & M. Tompa, "Decreasing the nesting depth of expressions involving square roots," *J. Symbolic Comp.* **1** (1985), pp. 169–188.
6. J. M. Borwein & P. B. Borwein, *Pi and the AGM*, Wiley-Interscience, John Wiley & Sons, Toronto, 1987, p. 384.
7. J. M. Borwein & D. M. Bradley, "Empirically Determined Apéry-Like Formulae for $\zeta(4n+3)$," *Experimental Mathematics*, Vol. 6, Issue 3, October 1997, pp. 181–194.
8. J. M. Borwein & D. M. Bradley, "Searching Symbolically for Apéry-Like Formulae for Values of the Riemann Zeta Function," *SIGSAM Bulletin of Symbolic and Algebraic Manipulation*, Vol. 30, No. 2, Issue 116, (June 1996), pp. 2–7.
9. H. R. P. Ferguson & R. W. Forcade, "Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher Than Two," *Bulletin of the American Mathematical Society*, Vol. 1 (1979), pp. 912–914.

10. J. Hastad, B. Just, J. C. Lagarias, & C. P. Schnorr, "Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers," *SIAM Jouranl on Computing*, Vol. 18 (1988), pp. 859–881.
11. G. H. Hardy & E. M. Wright, *An Introduction to the Theory of Numbers*, (5th ed.) Oxford University Press, New York, 1979, p. 148.
12. A. K. Lenstra, H. W. Lenstra, & L. Lovasz, "Factoring Polynomials with Rational Coefficients," *Math. Annalen*, Vol. 261 (1982), pp. 515–534.
13. L. Lewin, *Polylogarithms and Associated Functions*, Elsevier North Holland, New York, 1981.
14. S. Ramanujan, "On the integral $\displaystyle\int_0^x \frac{\tan^{-1} t}{t}\, dt$," *Journal of the Indian Mathematical Society*, VII (1915), pp. 93–96.
15. N. J. A. Sloane & S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, 1995.
16. R. Zippel, "Simplifications of expressions involving radicals," *J. Symbolic Comp.* **1** (1985), pp. 189–210.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF NIJMEGEN
The Netherlands

Signed bits and fast exponentiation

Wieb Bosma

Report No. 9935 (July 1999)

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF NIJMEGEN
Toernooiveld
6525 ED Nijmegen
The Netherlands

Signed bits and fast exponentiation

Wieb Bosma

Abstract.

An exact analysis is given of the benefits of using the non-adjacent form representation for integers when computing powers of elements in a group in which inverting is easy. By counting the number of multiplications for a random exponent requiring a given number of bits in its binary representation, we arrive at a precise version of the known asymptotic result that on average one in three signed bits in the non-adjacent form is non-zero. This shows that the use of signed bits can reduce the cost of exponentiation by one ninth.

1. Introduction

To raise elements in a monoid into the power $e > 1$, the method of repeated squaring and multiplication is often employed. To calculate x^e, where $e = \sum_{i=0}^{n} b_i 2^i$, with $b_i \in \{0, 1\}$ and $b_n = 1$, the powers

$$y_0 = x^1, y_1 = x^2, y_2 = x^4, \ldots, y_n = x^{2^n}$$

are computed by repeated squaring, and x^e is found by taking the product of the y_i for which $b_i = 1$. It is clear that computing x^e this way takes $l(e) - 1$ squarings and $w(e) - 1$ multiplications, where the (binary) *length* $l(e) = n + 1$ and the *Hamming weight* $w(e)$ are the total number of bits and the number of non-zero bits b_i used to express the exponent e.

If the monoid is a group in which inverses can be computed efficiently, it may be advantageous to use a different representation of the exponent. Writing $e = \sum_{i=0}^{m} s_i 2^i$, where $s_i \in \{-1, 0, 1\}$, we have obtained a *signed bit* representation for e [2]. To determine x^e, again compute

$$y_0 = x^1, y_1 = x^2, y_2 = x^4, \ldots, y_m = x^{2^m}$$

via repeated squaring, and accumulate the product $y_i^{s_i}$ (for the non-zero s_i), which involves an inversion if $s_i = -1$.

The advantage of signed bit representations is that the signed bit weight $w_s(e)$ may be smaller than $w(e)$. Taking $e = 15$ for example, the binary representation consists of four bits equal to 1: in binary $e = 1111$. But $15 = -1 + 2^4$, so $e = 1000\bar{1}$, a signed bit representation of weight 2 and length 5. At the cost of one inversion and an extra squaring we have done away with two multiplications.

For certain exponents e there exist better ways to compute x^e, using arbitrary addition chains or addition-subtraction chains. We briefly discuss them in Section 3.

A complication in considering signed bits may seem that signed bit representations of integers are by no means unique. Indeed, using that the integer 1 has a representation $1 = 2^k + \sum_{i=0}^{k-1} -1 \cdot 2^i$, for any $k > 1$, it is seen that every integer admits infinitely many signed bit representations. In Section 2 we describe the *non-adjacent*

form, which selects a unique signed bit representation for any non-negative integer e. We indicate how it, and a modified version of it, can be determined efficiently, and we show that these special representations have certain optimal properties.

In Sections 4 and 5 we will analyze exactly the weight of non-adjacent forms for integers e. It is shown (in a precise sense) that on average this weight is a third of the length of e (as opposed to a half for the binary form). In general the gain that can be achieved from this in exponentiation will depend on the relative costs of inverting, multiplying, and squaring in the group. The standard application for signed bit exponentiation is to the arithmetic of elliptic curves, [7], [9]. The group of points on an elliptic curve over a field in Weierstrass form has the desired property that inverting is almost free. In such situations the results of Section 5 show that a reduction in cost of a ninth on average is obtained by using the non-adjacent form rather than the binary form. This makes precise a result that so far only seems to be known heuristically or asymptotically [1], [7], [9].

2. Signed Bits

To fix the notation, let a *signed-bit representation of length* $l(e)$ for a positive integer e be a sequence $s_{l(e)-1}, s_{l(e)-2}, \ldots, s_0$ such that $e = \sum_{i=0}^{l(e)-1} s_i 2^i$, with $s_i \in \{-1, 0, 1\}$ and $s_{l(e)-1} = 1$. Sometimes we will write $m = l(e) - 1$; the sequence of signed bits s_i is usually written without comma's with most-significant digit $s_{l(e-1}$ first.

As we have seen already, e will in general have signed-bit representations of various lengths; indeed, since we may replace the leading 2^m by $2^{m+1} - 2^m$, a process which can be repeated, we find infinitely many representations for any e, of arbitrary (large enough) length. With our application of minimizing costs of exponentiation in mind, we are particularly interested in *short* representations of *low weight*.

We will call a signed bit representation for e *optimal* if it has least possible weight and among all representations of minimal weight it has minimal length — clearly the length of the binary expansion is a lower bound for the length of a signed-bit representation. But note that optimality does not determine a unique representation in general, as the example $11 = 2^3 + 2 + 1 = 2^3 + 2^2 - 1$ shows.

Let us first worry about uniqueness. The *non-adjacent form* representation is the signed bit representation for e characterized by the property:

$$s_i \neq 0 \quad \Rightarrow \quad s_{i-1} = 0, \quad \text{for } i \geq 1.$$

Proposition 1. *Positive integers have unique non-adjacent form representations.*

Proof. Suppose that there exist positive integers e with two different non-adjacent forms. Among all such e select e_0 having a non-adjacent form of minimal length. The minimality condition requires that the least significant bit in the minimal representation of e_0 differs from that in any other. The only admissible pairs for the two least-significant bits in non-adjacent forms are 00, 01, 0−1, 10, −10; only −10 and 10 determine the same value modulo 4, but their least-significant bits are equal.

This ends the proof.

It is easy to obtain the non-adjacent form from the ordinary binary expansion: apply the following rule repeatedly, working from right to left (least-significant first):

$$\textit{replace any sequence } 01\cdots1 \textit{ by } 10\cdots0{-}1$$

where the number of consecutive 0's in the latter is one less than the number of consecutive 1's in the former.

Since $\sum_{i=0}^{k} = 2^{k+1} - 1$, it is clear that the result will always be a non-adjacent form representation for the given integer determined by the binary expansion. It will also be clear that the length of the non-adjacent form is either equal to or one larger than that of the binary expansion.

Example. Starting with the binary expansion for $3190 = 2^{11}+2^{10}+2^6+2^5+2^4+2^2+2$, the rule produces:

$$\begin{array}{cccccccccccc}
 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & -1 & 0 \\
 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\
1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 & 0
\end{array}$$

for $3190 = 2^{12} - 2^{10} + 2^7 - 2^3 - 2$.

In fact the above procedure can be generalized to transform any given signed bit representation into the non-adjacent form; first apply the following rule repeatedly working from left to right:

$$(*) \qquad \begin{array}{l} \textit{replace } -11 \textit{ by } 0{-}1, \textit{ and} \\ \textit{replace } 1{-}1 \textit{ by } 01, \end{array}$$

and then apply the following repeatedly (working from right to left).

$$(**) \qquad \begin{array}{lcll} \textit{replace} & 0\overbrace{1\cdots1}^{k>1} & \textit{by} & 1\overbrace{0\cdots0}^{k-1}{-}1, \quad \textit{and} \\ \textit{replace} & 0\underbrace{{-}1\cdots{-}1}_{k>1} & \textit{by} & -1\underbrace{0\cdots0}_{k-1}1. \end{array}$$

Proposition 2. *For any integer the non-adjacent form has minimal weight.*

Proof. Apply the above two rule-transformation to any signed bit represenatation of minimal weight; the result is the non-adjacent form. The transformation does not increase the weight.

Corollary 3. *For every integer there is a unique signed bit representation satisfying:*

$$s_k \neq 0 \quad \Rightarrow \quad s_{k-1} = 0, \quad \text{or} \quad k = m \quad \text{and} \quad s_{m-1} = 1 = s_m;$$

moreover this expansion is optimal.

Proof. Let $t_m t_{m-1} \cdots t_1 t_0$ be the non-adjacent form for e. If the three most significant bits $t_m t_{m-1} t_{m-2}$ are $10{-}1$, then let $n = m - 1$ and define

$$s_i = \begin{cases} 1 & \text{for } i = n, n-1 \\ t_i & \text{for } 0 \leq i \leq n-2. \end{cases}$$

In all other cases let $n = m$ and $s_i = t_i$ for $0 \leq i \leq n$. This way s is equal to the non-adjacent form except when the leading digits for the non-adjacent form are $10{-}10$, in which case we replace them by the shorter expansion with leading digits 110. Clearly s satisfies the non-adjacency conditions of the statement; we will show that it is optimal too.

In the exceptional case the weights of s and t are equal, but the length of s equals that of the binary expansion. Hence s is optimal in that case. We will prove that in all other cases the non-adjacent form t itself is optimal.

Suppose that e is an integer with non-adjacent form $t_m t_{m-1} \cdots t_1 t_0$ of minimal length that is not optimal. Since the non-adjacent weight *is* always minimal, this can only occur if the length of the non-adjacent form of e exceeds that of its binary expansion by 1. This only happens if in the final transformation step a sequence of $k \geq 2$ adjacent 1's is replaced by $10 \cdots 0{-}1$, where the number of 0's is $k-1$. If $k = 2$ we are in the exceptional case, so we will assume that $k > 2$. The binary expansion $u_{m-1} u_{m-2} \cdots u_0$ has $u_{m-1} = u_{m-2} = u_{m-3} = 1$, while $u_{m-4} = 0$ or 1.

Since the non-adjacent weight *is* minimal, there must exist a signed bit representation $v_{m-1} v_{m-2} \cdots v_0$ of length m, and it necessarily has $v_{m-1} = v_{m-2} = v_{m-3} = 1$, and $v_{m-4} = u_{m-4} \in \{0, 1\}$ since u and v represent the same number e. If $v_{m-4} = 1$, an extra reduction step reduces length plus weight, which contradicts optimality of v. So $v_{m-4} = 0$; but then $v \neq u$ contradicts minimality of m since $v_{m-5} v_{m-6} \cdots$ represents the same number as $u_{m-5} u_{m-6}$ with lower weight.

That ends the proof.

We will refer to the optimal representation of Corollary 3 as the *modified non-adjacent form*. It is the same as the non-adjacent form, except that non-adjacency is allowed in the most significant two bits, that is 110 is not transformed to $10{-}10$, because such transformation increases the length without decreasing the weight.

Note that this does *not* mean that the modified version is different for precisely those integers for which the leading bits in the binary expansion are 110 because of the propagation of carries in the transformations: non-adjacent and modified non-adjacent forms for $27 = 11011 = 100{-}10{-}1$ are the same, but for $25 = 11001$ they are different, namely $10{-}1001$ and 11001.

It is not so difficult to obtain the (modified) non-adjacent form directly from e, without computing the binary (or another signed-bit) expansion first. The method resembles the method for finding the binary expansion producing the least significant bit first: starting with $k = e$ repeat:

if k even: produce 0 and divide k by 2;

if k odd: produce 1, subtract 1 from k and divide k by 2;

until k is 0.

For the non-adjacent form one proceeds as follows. Starting with $k = e > 0$ again, one repeats:

$k \bmod 4 \equiv s \in \{-1, 1\}$: produce signed bits s and 0, and replace k by $(k - s)/4$;

$k \bmod 4 \equiv s \in \{0, 2\}$: produce 0 and replace k by $k/2$.

e	binary	NAF	modified NAF
1	1	1	1
2	1 0	1 0	1 0
3	1 1	1 0−1	1 1
4	1 0 0	1 0 0	1 0 0
5	1 0 1	1 0 1	1 0 1
6	1 1 0	1 0−1 0	1 1 0
7	1 1 1	1 0 0−1	1 0 0−1
8	1 0 0 0	1 0 0 0	1 0 0 0
9	1 0 0 1	1 0 0 1	1 0 0 1
10	1 0 1 0	1 0 1 0	1 0 1 0
11	1 0 1 1	1 0−1 0−1	1 1 0−1
12	1 1 0 0	1 0−1 0 0	1 1 0 0
13	1 1 0 1	1 0−1 0 1	1 1 0 1
14	1 1 1 0	1 0 0−1 0	1 0 0−1 0
15	1 1 1 1	1 0 0 0−1	1 0 0 0−1
16	1 0 0 0 0	1 0 0 0 0	1 0 0 0 0
17	1 0 0 0 1	1 0 0 0 1	1 0 0 0 1
18	1 0 0 1 0	1 0 0 1 0	1 0 0 1 0
19	1 0 0 1 1	1 0 1 0−1	1 0 1 0−1
20	1 0 1 0 0	1 0 1 0 0	1 0 1 0 0
21	1 0 1 0 1	1 0 1 0 1	1 0 1 0 1
22	1 0 1 1 0	1 0−1 0−1 0	1 1 0−1 0
23	1 0 1 1 1	1 0−1 0 0−1	1 1 0 0−1
24	1 1 0 0 0	1 0−1 0 0 0	1 1 0 0 0
25	1 1 0 0 1	1 0−1 0 0 1	1 1 0 0 1
26	1 1 0 1 0	1 0−1 0 1 0	1 1 0 1 0
27	1 1 0 1 1	1 0 0−1 0−1	1 0 0−1 0−1
28	1 1 1 0 0	1 0 0−1 0 0	1 0 0−1 0 0
29	1 1 1 0 1	1 0 0−1 0 1	1 0 0−1 0 1
20	1 1 1 1 0	1 0 0 0−1 0	1 0 0 0−1 0
31	1 1 1 1 1	1 0 0 0 0−1	1 0 0 0 0−1
32	1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0
33	1 0 0 0 0 1	1 0 0 0 0 1	1 0 0 0 0 1
34	1 0 0 0 1 0	1 0 0 0 1 0	1 0 0 0 1 0
35	1 0 0 0 1 1	1 0 0 1 0−1	1 0 0 1 0−1
36	1 0 0 1 0 0	1 0 0 1 0 0	1 0 0 1 0 0
37	1 0 0 1 0 1	1 0 0 1 0 1	1 0 0 1 0 1
38	1 0 0 1 1 0	1 0 1 0−1 0	1 0 1 0−1 0

until k is less than or equal to 3, after which

if $k = 0$: produce nothing;

if $k = 1$: produce 1;

if $k = 2$: produce 0 and 1;

if $k = 3$: produce -1 and 0 and 1;

and terminate.

For the modified version the only change necessary is to produce 11 in the case that $k = 3$.

Note the similarities with the continued fraction algorithm, where division by 2 is replaced by inverting, and truncation replaces extracting bits. The algorithm to obtain the non-adjacent form is similar to the nearest integer continued fraction algorithm.

The table shows binary expansion, non-adjacent form, and modified non-adjacent form for the first few positive integers.

3. Addition-subtraction chains

The method of repeated squaring and multiplication does not necessarily give the fastest way to evaluate powers. It is well-known [6] that for certain exponents there are ways to find x^e, using fewer multiplications.

An *addition chain* for a positive integer e is a sequence $1 = e_0, e_1, \ldots, e_k = e$ with the property that for $1 \leq i \leq k$ it holds that $e_i = e_u + e_v$ with $0 \leq u, v < i$. Each term is thus the sum of two (possibly the same) previous terms. One usually arranges the e_i in ascending order. The length of the addition chain is the integer k. It will be clear that an addition chain for e can be used to compute x^e: for any i the power x^{e_i} can be computed from $x^{e_0}, \ldots, x^{e_{i-1}}$ by a single multiplication.

The binary expansion $e = \sum_{i=0}^{n} b_i 2^i$ of any e of length $n+1$ defines an addition chain of length $n + w(e) - 1$ for e, corresponding to repeated squaring and multiplication as described in Section 1, as follows. Write down the powers $p_i = 2^i, i = 0, \ldots, n$ of 2 less than or equal to e. Next take $r_0 = 0$ and let r_j be $r_{j-1} + p_{i_j}$, where $i_1, \ldots, i_k$ are those i from 0 to n for which $b_i \neq 0$. The addition chain for e then consists of the the p_i (with $1 \leq i \leq n$) and r_j (with $j \geq 1$) in ascending order.

There is an alternative addition chain associated with the binary expansion, obtained by reading the bits from left to right (most significant first). Starting with $e_0 = 1$ one repeats for $i = 1, \ldots, n$:

if $b_{n-i} = 1$*:* append $2e_j$ and $2e_j + 1$ to the existing sequence $e_0, \ldots e_j$;
otherwise: append $2e_j$ to the existing sequence $e_0, \ldots e_j$.

There are two problems with addition chains. In the first place is it hard to find a shortest chain for given e [6]. Secondly, general addition chains make it necessary to remember entries $x^{e_0}, \ldots, x^{e_{i-1}}$ along the way to compute x_i^e. Note that this is not true for the left-to-right binary addition chain, as e_i is either $2e_{i-1}$ or $e_{i-1} + 1$, that is, every step is either a squaring or a multiplication by x ([4], see also [8] for the special case of integer exponentiation).

Taking the possibility of subtracting into account as well, we arrive at *addition-subtraction chains* [11]. In general we cannot insist on ascending entries anymore. Again, it will be clear that any signed-bit representation of e will give rise to two addition-subtraction chains, by reading the signed bits either way. It is also obvious that, since the weight of a signed bit representation can be smaller than that of the binary expansion, that the corresponding chain may be shorter.

Examples. Let $e = 43$; reading its bits 101011 right-to-left to obtain the sequence of

p_i's $1, 2, 4, 8, 16, 32$ and of r_j's $3, 11, 43$, we obtain an addition chain by merging and ordering: $1, 2, 3, 4, 8, 11, 16, 32, 43$ of length 8.

Reading the binary expansion 101011 left-to-right produces $e_0 = 1$, then $e_1 = 2$, and $e_2 = 4, e_3 = 5$, then $e_4 = 10$, and $e_5 = 20, e_6 = 21$, and finally $e_7 = 42, e_8 = 43$. Indeed, length 8 for 5 doublings and 3 multiplications.

Reading the modified non-adjacent form $43 = 110\bar{1}0\bar{1}$ left-to-right yields the addition-subtraction chain $1, 2, 3, 6, 12, 11, 22, 44, 43$, reading it right-to-left the chain $-1, 2, 4, -5, 8, 16, 11, 32, 43$. Both have length 8. The non-adjacent form produces chains of length 9.

There exists an addition chain of length 7 for 43: $1, 2, 4, 8, 9, 17, 34, 43$.

The addition-subtraction chain $1, 2, 4, 8, 16, 15$ associated with $15 = 2^4 - 2^0$, is shorter than the chain $1, 2, 3, 6, 7, 14, 15$ arising from the binary expansion $15 = 2^3 + 2^2 + 2^1 + 2^0$. In this case there is an addition chain of length 5 as well, however: $1, 2, 3, 5, 10, 15$ for example.

In general, for $e = 2^k - 1$ the binary expansion gives rise to an addition chain of length $2k - 2$ while the non-adjacent form leads to an addition-subtraction chain of length $k + 1$.

Outside numbers of this form, $e = 23$ is the first example where the modified non-adjacent form for e leads to an addition-subtraction chain ($1, 2, 3, 6, 12, 24, 23$ of length 6) that is strictly shorter than the binary addition chains ($1, 2, 4, 5, 10, 11, 22, 23$ and $1, 2, 3, 4, 7, 8, 16, 23$ of length 7). Again there exist addition chains of length 6, like $1, 2, 3, 5, 10, 13, 23$.

For $e = 27$ there are addition chains (such as $1, 2, 3, 6, 9, 18, 27$) that are shorter than both the chains obtained from the binary expansion ($1, 2, 3, 6, 12, 13, 26, 27$) and the addition-subtraction chain gotten from the (modified) non-adjacent form ($1, 2, 4, 8, 7, 14, 28, 27$).

For $e = 47$ the length of the chain given by the modified non-adjacent form ($1, 2, 3, 6, 12, 24, 48, 47$) is shorter than any addition chain (the shortest of which have length 8: $1, 2, 3, 4, 7, 10, 20, 27, 47$ for example, while the binary gives length 9: $1, 2, 4, 5, 10, 11, 22, 23, 46, 47$); in this case there is no shorter addition-subtraction chain either.

4. Analysis

To analyze the benefits of using the signed bit representations, we first prove some results on (average) length of non-adjacent and modified non-adjacent forms. Let c_n denote the number of positive integers requiring *exactly* n bits in their binary representation, and let c'_n and c''_n be the number of positive integers requiring *exactly* n signed bits in the non-adjacent form and in the modified non-adjacent form representation, respectively. Also, let C_n, C'_n and C''_n similarly define the number of positive integers requiring *at most* n bits in the three representations.

Proposition 4. *The number of positive integers with expansions of length n is given by $c_1 = c'_1 = c''_1 = 1$, and for $n \geq 2$:*

$$c_n = 2^{n-1}, \qquad c'_n = \frac{2}{3}2^{n-1} - \frac{(-1)^n}{3}, \qquad c''_n = \frac{5}{6}2^{n-1} + \frac{(-1)^n}{3}.$$

Hence, for $n \geq 0$*:*

$$C_n = 2^n, \qquad C'_n = \frac{2}{3}2^n + \frac{1}{2} - \frac{(-1)^n}{6}, \qquad C''_n = \frac{5}{6}2^n + \frac{1}{2} + \frac{(-1)^n}{6}.$$

Proof. Only 1 requires one bit in any expansion. It is also clear that there are exactly 2^{n-1} integers with most significant bit $b_{n-1} = 1$ (of length n), so $c_n = 2^{n-1}$ and $C_n = \sum_{k=0}^{n} c_k = 2^n$.

The easiest way to count integers with n signed bits in their non-adjacent form is to observe that the following recursion holds:

$$c'_{n+2} = c'_{n+1} + 2c'_n, \qquad \text{for} \quad n \geq 1.$$

Namely, the c'_n positive integers of length n (all having $s_{n-1} = 1$), when 'prepended' with $s_n = 0$ and $s_{n+1} = 1$ all contribute. We get another contribution of size c'_n by flipping the n-th bit b_{n-1} to -1. This accounts for all positive integers requiring $n+2$ bits for which $b_{n-1} \neq 0$. We obtain those with $b_{n-1} = 0$ by taking the c'_{n+1} representations of length $n+1$ and replacing the leading digit $b_n = 1$ by $b_n = 0$ and putting $b_{n+1} = 1$. This way the validity of the recursion can be seen to hold. With starting values $c'_1 = c'_2 = 1$ the closed form for c'_n in the statement of the proposition is then easily proved, for example by induction. The formula for C'_n is simply obtained by summation: $\sum_{k=0}^{n} c'_k$.

One way to count integers with modified non-adjacent form of length n is to use that their number also satisfies the recursion:

$$c''_{n+2} = c''_{n+1} + 2c''_n, \qquad \text{for} \quad n \geq 2.$$

This time one takes the representations of length n, and obtains from each two valid representations of length $n+2$ by shifting over 2 places and inserting $b_1 = 0$ and $b_0 = \pm 1$. From the length $n+1$ representations one gets length $n+2$ representations by shifting one place and taking $b_0 = 0$. This clearly leads to $2c''_n + c''_{n+1}$ valid representations of length $n+2$ (taking care that $n > 1$ to prevent the illegal representation $10{-}1$ for 3) that are all distinct (look at b_0); it is not terribly hard to see that we obtain *all* valid modified signed bit representations this way. The starting values for the recursion are $c''_2 = 2$ and $c''_3 = 3$. Again, C''_n can be derived by summation.

Here are the first few values for each of the functions:

n	= 1	2	3	4	5	6	7	8	9	10	11 ...
c_n	= 1	2	4	8	16	32	64	128	256	512	1024 ...
c'_n	= 1	1	3	5	11	21	43	85	171	341	683 ...
c''_n	= 1	2	3	7	13	27	53	107	213	427	853 ...
C_n	= 2	4	8	16	32	64	128	256	512	1024	...
C'_n	= 2	3	6	11	22	43	86	171	342	683	...
C''_n	= 2	4	7	14	27	54	107	214	427	854	...

Remarks. Note that c_n *also* satisfies the recursion that c'_n and c''_n satisfy. The sequence c'_n has been called the Jacobsthal sequence (A001045 in [10], [5]).

The six sequences satisfy many other intriguing relations, of which we just mention a few (see also [5]). For $n \geq 1$

$$\begin{aligned} c_{n+1} + c_n &= 3 \cdot 2^{n-2}, \\ c'_{n+1} + c'_n &= 2 \cdot 2^{n-2} = 2^{n-1}, \\ c''_{n+1} + c''_n &= \frac{5}{2} \cdot 2^{n-2} = 5 \cdot 2^{n-3}. \end{aligned}$$

Related to this, are

$$\begin{aligned} c_n &= (-1)^n(-1 + \sum_{k=0}^{n-2}(-2)^k) = 1 + \sum_{k=0}^{n-2} 2^k, \\ c'_n &= (-1)^{n-1} \sum_{k=0}^{n-1}(-2)^k, \\ c''_n &= (-1)^n(\frac{-1}{2} + \sum_{k=0}^{n-2} \frac{5}{2} \cdot (-2)^k), \end{aligned}$$

While $c_{n+1} = 2c_n$ and $C_{n+1} = 2C_n$ for all n, we have

$$\begin{aligned} c'_{n+1} = 2c'_n + (-1)^{n-1} \quad \text{and} \quad C'_{n+1} &= 2C'_n + \frac{1}{2} - \frac{(-1)^n}{2}, \\ c''_{n+1} = 2c''_n + (-1)^n \quad \text{and} \quad C''_{n+1} &= 2C''_n + \frac{1}{2} - \frac{(-1)^{n-1}}{2}. \end{aligned}$$

The various sequences are interrelated via, for example,

$$\begin{array}{rlll} & c_n = c'_n + c'_{n-1} & \text{and} & C_n = C'_n + C'_{n-1} - 1, \\ (**) & c'_n = c''_{n-1} + c_{n-2} & \text{and} & C'_n = C''_{n-1} + C_{n-2}, \\ & c''_n = c_{n-1} + c'_{n-1} & \text{and} & C''_n = C_{n-1} + C'_{n-1}. \end{array}$$

Next we count the total weight of all representations of fixed length. Define s_n to be the total number of ones in all different n-bit integers; we use s'_n and s''_n for the total number of non-zero signed bits in all different non-adjacent forms and modified non-adjacent forms of length n. Similarly, by S_n, S'_n and S''_n we denote the total number of non-zeroes in in all binary, non-adjacent and modified non-adjacent representations of length *at most* n.

Proposition 5. *For $n \geq 2$:*

$$\begin{aligned} s_n &= \frac{n+1}{2} \cdot 2^{n-1}, \\ s'_n &= \frac{6n+10}{27} \cdot 2^{n-1} + (-1)^{n-1}\frac{6n+5}{27}, \\ s''_n &= \frac{15n+34}{54} \cdot 2^{n-1} - (-1)^{n-1}\frac{6n+5}{27}. \end{aligned}$$

Also,

$$S_n = \frac{n}{2} \cdot 2^n,$$
$$S'_n = \frac{6n+4}{27} \cdot 2^n + (-1)^{n-1}\frac{3n+4}{27},$$
$$S''_n = (\frac{5n}{18} + \frac{19}{54}) \cdot 2^n - (-1)^{n-1}\frac{3n+4}{27}.$$

Proof. To count the total number of non-zero bits in n-bit words, note that $n+1$ bit words can be formed out of n-bit words by shifting and 'appending' a single bit (0 or 1). Since there are c_n such n-bit integers, having s_n non-zero bits, we find

$$s_{n+1} = s_n + (s_n + c_n).$$

From $s_1 = 1$ and $s_2 = 3$ we get the result by induction.

To prove the formula for s'_n, note that

$$s'_{n+2} = 2(s'_n + c'_n) + s'_{n+1}.$$

This follows immediately from the proof of the previous Proposition. Then use verification of $s'_1 = s'_2 = 1$ and induction.

For s''_n one derives similarly that

$$s''_{n+2} = s'_n + c'_n + 2 \cdot s'_{n+1} + c'_{n+1}.$$

For S_n and S'_n we sum $\sum_{k=0}^{n} s_k$ and $\sum_{k=0}^{n} s'_k$, only using that

$$\sum_{k=0}^{n} k2^k = (n-1)2^{n+1} + 2$$

.

Here are the first few values for each of the functions again:

n	$=1$	2	3	4	5	6	7	8	9	10	11	...
s_n	$=1$	3	8	20	48	112	256	576	1280	2816	6144	...
s'_n	$=1$	1	5	9	25	53	125	273	609	1325	2885	...
s''_n	$=1$	3	5	15	31	75	163	367	799	1747	3771	...
S_n	$=1$	4	12	32	80	192	448	1024	2304	5120		...
S'_n	$=1$	2	7	16	41	94	219	492	1101	2426		...
S''_n	$=1$	4	9	24	55	130	293	660	1459	3206		...

As a consequence we can determine how many non-zero (signed) bits there are on average in all integers requiring exactly or at most n bits in the various expansions; we denote these by g_n, g'_n, g''_n and t_n, t'_n, t''_n.

Corollary 6. *For all $n \geq 2$:*

$$g_n = \frac{s_n}{nc_n} = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{n},$$
$$g'_n = \frac{s'_n}{nc'_n} = \frac{1}{3} + \frac{5}{9} \cdot \frac{1}{n} - (-1)^n \frac{1}{3 \cdot (2^n - (-1)^n)},$$
$$g''_n = \frac{s''_n}{nc''_n} = \frac{1}{3} + \frac{34}{45} \cdot \frac{1}{n} + (-1)^n \frac{(1 - \frac{3}{5n})}{3 \cdot (5 \cdot 2^{n-2} + (-1)^n)},$$

and

$$G_n = \frac{S_n}{nC_n} = \frac{1}{2},$$
$$G'_n = \frac{S'_n}{nC'_n} = \frac{1}{3} + \frac{2}{9} \cdot \frac{1}{n} - \frac{3 + (-1)^n + (1 + (-1)^n)\frac{2}{n}}{3 \cdot (2^{n+2} + 3 + (-1)^n)},$$
$$G''_n = \frac{S''_n}{nC''_n} = \frac{1}{3} + \frac{19}{45} \cdot \frac{1}{n} - \frac{3 - (-1)^n + (19 - (-1)^n \cdot 7)\frac{1}{5n}}{3 \cdot (5 \cdot 2^n + 3 + (-1)^n)}$$

This Corollary, the proof of which is an easy computation, tells us that on average half the bits in a binary expansion are non-zero (as expected), one in three signed bits in the non-adjacent form are non-zero (compare [1, 3, 9]). For the modified non-adjacent form also a third of the bits are non-zero asymptotically, but the convergence is slightly slower because there are fewer zeroes in the exceptional case.

To give a fair comparison, we need to count the number of bits used for integer with binary expansion of length n. An *n-bit integer* is a non-negative integer for which the ordinary binary representation has length n exactly.

5. Analysis for integers of given length

First we count the total length and the total weight of n-bit integers in the various representations. As usual we denote by l, l', l'' and L, L', L'' the values for ordinary binary, non-adjacent form and modified non-adjacent form representation.

Proposition 7. *The total length of all numbers that take exactly n bits in binary:*

$$l_n = n2^{n-1},$$
$$l'_n = (n + \frac{2}{3})2^{n-1} - \frac{1}{2} - (-1)^{n-1}\frac{1}{6},$$
$$l''_n = (n + \frac{1}{3})2^{n-1} - \frac{1}{2} + (-1)^{n-1}\frac{1}{6}.$$

The total length of all numbers that take at most n bits (in the ordinary representation):

$$L_n = (n-1)2^n + 1,$$
$$L'_n = (n - \frac{1}{3})2^n - \frac{n}{2} + \frac{1}{4} + \frac{(-1)^n}{12},$$
$$L''_n = (n - \frac{2}{3})2^n - \frac{n}{2} + \frac{3}{4} - \frac{(-1)^n}{12}.$$

Proof. Obviously the c_n length n integers give

$$l_n = nc_n.$$

One way to count l'_n is to determine which length n integers contribute to length n non-adjacent forms. These are the binary expansions of length n for which $b_{n-2} = 0$ and for which the non-adjacent form of $b_{n-3}b_{n-4}\cdots b_0$ has length $n-2$. Of those there are exactly C'_{n-2}. The others, $c_n - C'_{n-2} = C_{n-1} - C'_{n-2} = C'_{n-1} - 1$ in number (compare $(**)$), contribute length $n+1$ each, so

$$l'_n = nC'_{n-2} + (n+1)(C'_{n-1} - 1) = (n+1)c_n - C'_{n-2}.$$

Using Proposition 4 immediately gives the desired result.

Similarly it can be proven that

$$l''_n = nC'_{n-1} + (n+1)(C'_{n-2} - 1),$$

For L_n we merely sum:

$$L_n = \sum_{k=0}^{n} l_k,$$

and likewise for L'_n and L''_n.

The first few values for these functions are:

n	$=$	1	2	3	4	5	6	7	8	9	10	11	...
l_n	$=$	1	4	12	32	80	192	448	1024	2304	5120	11264	...
l'_n	$=$	1	5	14	37	90	213	490	1109	2474	5461	11946	...
l''_n	$=$	1	4	13	34	85	202	469	1066	2389	5290	11605	...
L_n	$=$	1	5	17	49	129	321	769	1793	4097	9217		...
L'_n	$=$	1	6	20	57	147	360	850	1959	4433	9894		...
L''_n	$=$	1	5	18	52	137	339	808	1874	4263	9553		...

Let w_n, w'_n, w''_n denote the total weight of all non-negative integers requiring exactly n bits in binary representation, and W_n, W'_n, W''_n the same for integers of at most n bits.

Proposition 8.

$$w_n = (n+1)2^{n-2}, \qquad w'_n = w''_n = (\frac{n}{3} + \frac{7}{9})2^{n-1} + (-1)^n\frac{1}{9}$$

$$W_n = n2^{n-1}, \qquad W'_n = W''_n = (\frac{n}{3} + \frac{4}{9})2^n - \frac{1}{2} + (-1)^n\frac{1}{18}$$

Proof. Obviously again,

$$w_n = s_n.$$

The weight of non-adjacent and modified non-adjacent forms are the same, so $w'_n = w''_n$ and $W'_n = W''_n$. The first integer that requires n binary bits is $f_n = 2^{n-1}$. For every integer h larger than f_n for which the length of its non-adjacent form is n, there is an integer g smaller than f_n that has non-adjacent form of length $n-1$ and the same weight as h: simply reverse all bits of h except for the most significant one. Thus the integers with non-adjacent forms of length n other than f_n (which has weight 1) contribute exactly half their total weight, that is $(s'_n - 1)/2$, to w'_n. On the other hand, for the same reason exactly half the total weight of the length $n+1$ non-adjacent forms contribute to the binary length n count, which implies that

$$w'_n = \frac{s'_n - 1}{2} + \frac{s'_{n+1} - 1}{2} + 1,$$

the $+1$ being the contribution of f_n itself. Substitution then gives the result.

A small table again:

n	= 1	2	3	4	5	6	7	8	9	10	11 ...
w_n	= 1	3	8	20	48	112	256	576	1280	2816	6144 ...
$w'_n = w''_n$	= 1	3	7	17	39	89	199	441	967	2105	4551 ...
W_n	= 1	4	12	32	80	192	448	1024	2304	5120	...
$W'_n = W''_n$	= 1	4	11	28	67	156	355	796	1763	3868	...

Corollary 9. *The number of multiplications necessary to compute x^e for a random integer e of exactly n bits using the binary expansion, the non-adjacent form and the modified non-adjacent form for e is:*

$$m_n = \frac{l_n + w_n}{c_n} - 2 = \frac{3}{2}(n-1),$$
$$m'_n = \frac{l'_n + w'_n}{c_n} - 2 = \frac{4}{3}(n-1) + \frac{7}{9} - (\frac{1}{2} + (-1)^{n-1}\frac{1}{18}) \cdot \frac{1}{2^{n-1}},$$
$$m''_n = \frac{l''_n + w''_n}{c_n} - 2 = \frac{4}{3}(n-1) + \frac{4}{9} - (\frac{1}{2} - (-1)^{n-1}\frac{5}{18}) \cdot \frac{1}{2^{n-1}}.$$

If e is random of at most n digits, the cost functions are:

$$M_n = \frac{L_n + W_n}{C_n} - 2 = \frac{3}{2}(n-2) + \frac{1}{2^n},$$
$$M'_n = \frac{L'_n + W'_n}{C_n} - 2 = \frac{4}{3}(n-2) + \frac{7}{9} + (-\frac{n}{2} - \frac{1}{4} + (-1)^n\frac{5}{36}) \cdot \frac{1}{2^n},$$
$$M''_n = \frac{L''_n + W''_n}{C_n} - 2 = \frac{4}{3}(n-2) + \frac{4}{9} + (-\frac{n}{2} + \frac{1}{4} - (-1)^n\frac{1}{36}) \cdot \frac{1}{2^n}.$$

As expected we see that, for e of binary length n, it takes $n-1$ multiplications (all squarings) and on average $(n-1)/2$ multiplications using the binary expansion for e;

using the non-adjacent form the number of multiplications can be reduced to $(n-1)/3$, where on average we save $1/3$ multiplication using the modified form.

References

[1] Steven Arno, Ferrell S. Wheeler, *Signed digit representations of minimal Hamming weight*, IEEE Transactions on Computers **42** (1993), 1007–1010.

[2] Andrew D. Booth, *A signed binary multiplication technique*, Quart. Journ. Mech. and Applied Math. **4** (1951), 236–240.

[3] Daniel M. Gordon, *A survey of fast exponentiation methods*, Journal of Algorithms **27** (1998), 129–146.

[4] R. L. Graham, A. C.-C. Yao, F.-F. Yao, *Addition chains with multiplicative cost*, Discrete Math. **23** (1978), 115–119.

[5] A. F. Horadam, *Jacobsthal representation numbers*, Fibonacci Quart. **34** (1996), 40–54.

[6] D. E. Knuth, *The Art of Computer Programming* **2**: *Seminumerical Algorithms* (third edition), Reading: Addison Wesley, 1998.

[7] Neal Koblitz, *CM-curves with good cryptographic properties*, in: Feigenbaum (ed), *Advances in Cryptology — Proceedings of Crypto '91*, Lecture Notes in Computer Science **576**, (1991), 279–296.

[8] D. P. McCarthy, *Effect of improved multiplication efficiency on exponentiation algorithms derived from addition chains*, Math. Comp. **46** (1976), 603–608.

[9] F. Morain, J. Olivos, *Speeding up the computations on an eliptic curve using addition-subtraction chains*, RAIRO Inform. Theory **24** (1990), 531–543.

[10] N. J. A. Sloane, S. Plouffe, *The encyclopedia of integer sequences*, San Diego: Academic Press, 1995. `http://www.research.att.com/ njas/sequences/`

[11] Hugo Volger, *Some results on addition/subtraction chains*, Information Processing Letters **20** (1985), 155–160.

The Random Planar Graph

Alain Denise* Marcio Vasconcellos Dominic J.A. Welsh
LRI LaBRI Mathematical Institute
Université Paris-Sud Université Bordeaux I University of Oxford

Abstract

We construct a Markov chain whose stationary distribution is uniform over all planar subgraphs of a graph. In the case of the complete graph our experiments suggest that the random simple planar graph on n vertices is connected but not 2-connected and has approximately $2n$ edges. We present a first attack on the problem of describing what the random planar graph looks like.

1 Introduction

The basic questions which we will be considering are the following.

Problem 1. How does one generate a random simple planar graph uniformly at random from the set of simple planar graphs on n vertices?

Problem 2. What does this random planar graph look like?

First we clarify the issue. While there is a vast literature and long history of methods of generating random plane configurations such as Voronoi polygons, Delauney triangulations and the like, these are *not* random in the sense of being *uniformly* at random over the set of all planar graphs and are just ad hoc, fast, appealing methods of generating random plane configurations.

There is an intimate relationship between problems of counting and uniform generation and there is considerable literature on the problems of counting plane graphs and maps with a prescribed number of edges (see for example Tutte [13], Cori [2], Liskovets [7, 8], Cori and Vauquelin [3], Wormald [14, 15]). On the other hand, a few works are devoted to the random generation of certain types of planar maps and graphs (see [4, pp 74–83], [11]). However as far as we can see there is very little known about the two fundamental questions raised above.

*e-mail: denise@lri.fr, vasconce@labri.u-bordeaux.fr, dwelsh@maths.ox.ac.uk.

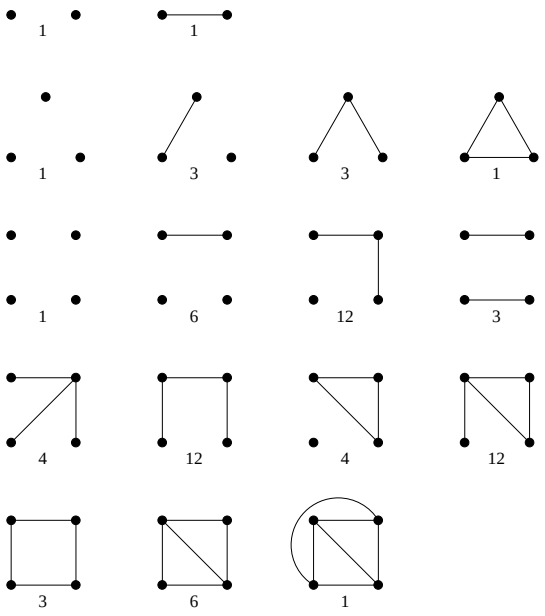


Figure 1: The (planar) graphs with 2, 3 and 4 vertices.

Consider first the collection of unlabelled simple planar graphs on n vertices. Call this set $\mathcal{U}(n)$ and denote its cardinality by $u(n)$. For example $u(2) = 2$, $u(3) = 4$ and $u(4) = 11$. Similarly let $\mathcal{L}(n)$ and $l(n)$ denote the set (respectively number) of simple planar *labelled* graphs on n vertices. Clearly $l(n) \geq u(n)$, for example the members of $\mathcal{U}(3)$ shown above give rise to respectively $1, 3, 3, 1$ distinct member of $\mathcal{L}(3)$ so that $l(3) = 8$. Figure 1 shows the set of unlabelled planar graphs with 2, 3 and 4 vertices. The number below each graph counts the associated labelled graphs.

In general, it is the labelled structures which are easier to deal with and it is these on which we shall be concentrating here.

In the recent book [10] the first terms of the sequence $(u(n))_{n\geq 1}$ are given: 1, 2, 4, 11, 33, 142, 822, 6910. It is easy to find the very first terms of $(l(n))_{n\geq 1}$: 1, 2, 8, 64, 1023.

The precise formulation of the question raised in problem 1 and to which we shall devote most of our attention is the following.

Problem 1(a). Does there exist an algorithm A which outputs a random planar subgraph of K_n and runs in time bounded by some polynomial function of n (written in unary)?

We are doubtful whether such an algorithm exists. First consider the exhaustive algorithm of listing all planar subgraphs of K_n and choosing one at random. There are $2^{\binom{n}{2}}$ subgraphs of K_n and $l(n)$, the number of these which are planar, is exponential (see section 6 below) so this approach cannot provide an answer. We next consider a randomised, slightly speeded up version of the above which can be applied to any input graph G and actually works well provided G is "close to planar".

(1) Generate a random subgraph of G by deleting each edge independently with probability $\frac{1}{2}$. Call the resulting graph R.

(2) If R is planar then $R_p = R$ else repeat.

(3) Output R_p.

When $G = K_n$ this randomised algorithm certainly gives a random planar graph. However it is extremely slow.

A more general version of problem 1 is the following:

Problem 1(b). Does there exist a polynomial time algorithm which for any input graph G will output a planar subgraph $R = R(G)$ chosen uniformly at random from all planar subgraphs of G?

We conjecture not.

2 A Markov chain algorithm

Let $G = (V, E)$ be any simple graph. We define a Markov chain $M(G)$ with state space all planar subgraphs of G and with transitions defined as follows.

A *position* of G consists of an unordered pair of distinct vertices of G. If X_t denotes the state $M(G)$, at time t, then X_{t+1} is chosen as follows. A position f of G is chosen uniformly at random.

(a) If the position f contains an edge e of X_t then $X_{t+1} = X_t \backslash e$.

(b) If the position $f = (i, j)$ does not contain an edge in X_t then X_{t+1} is formed from X_t by adding an edge (i, j) provided this addition preserves planarity,

(c) otherwise $X_{t+1} = X_t$.

It is easy to verify that (X_t) is an irreducible aperiodic Markov chain whose transition matrix is symmetric. Thus, X_t has a limiting stationary distribution which is uniform over the set of planar subgraphs of G. In principle therefore it gives an easily implemented algorithm for generating a planar subgraph of G which will be approximately uniformly at random. The closeness of the approximation will be governed by the mixing rate of the chain, and this will depend on the graph G. In particular, when $G = K_n$ it gives what appears to be a fairly effective way of generating a random planar graph.

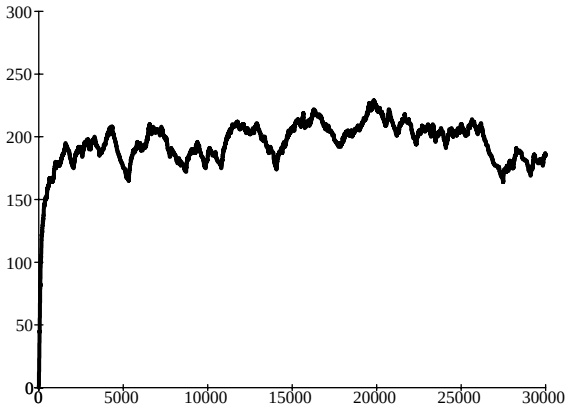


Figure 2: A typical simulation. Number of edges versus time-steps.

3 Experimental results

We present here the results of our experiments with this Markov Chain. The program was written in C++ using the LEDA library [9].

For practical reasons, we have usually chosen the empty graph as the initial state of the simulation. In each simulation, given n the number of vertices, we arbitrarily fix the number of time-steps to $3n^2$, which from our earlier pilot studies seems sufficiently large for the chain to settle down to what we believe is its equilibrium state.

Figure 2 shows the evolution of the number of edges during one execution of the program on a graph with 100 vertices. The curve increases rapidly then oscillates around a value near to 200. This can be seen more precisely in Figure 3 which presents average values of 50 simulations. The same experiment has been repeated on graphs with a number of vertices varying between 1 and 100. Figure 4 clearly suggests a linear relation between the number of vertices and the number of edges of a random planar graph. This result was obtained by computing the average number of edges of 10 graphs for each value of n varying from 1 to 100.

The next questions which we consider are the probabilities of a random planar graph being connected or biconnected. Figure 5 suggests that almost all planar graphs are connected: the probability of being connected seems to tend to 1 or to a value very near to 1 when n goes to infinity. On the contrary, the proportion of biconnected graphs decreases rapidly, as shown in Figure 6. These two experiments were done on 100 graphs for each value of n.

Finally we present in Figure 7 the distribution of the degrees of the vertices of 50 random planar graphs with 100 vertices. More precisely, in the random

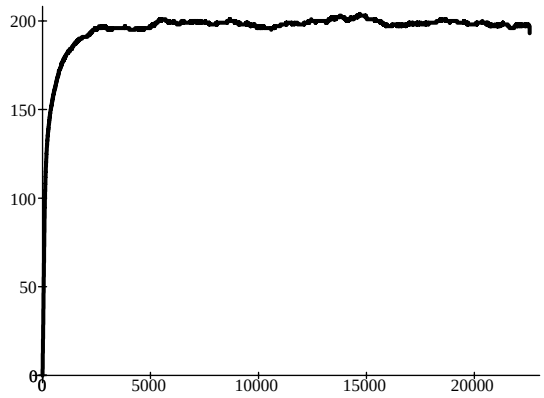


Figure 3: Number of edges versus time-steps: average values of 50 simulations.

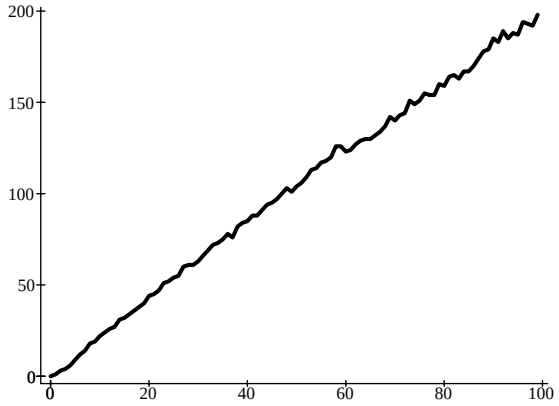


Figure 4: Average number of edges versus number of vertices.

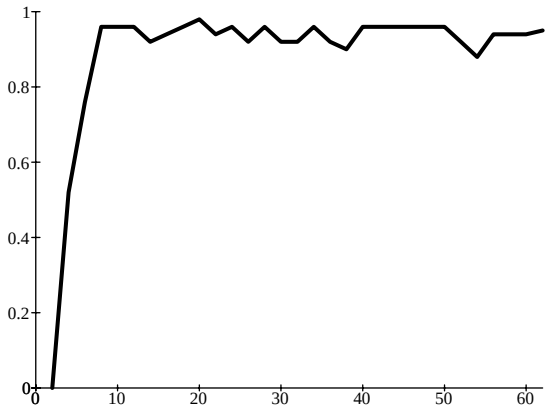


Figure 5: Experimental probability of being connected versus number of vertices.

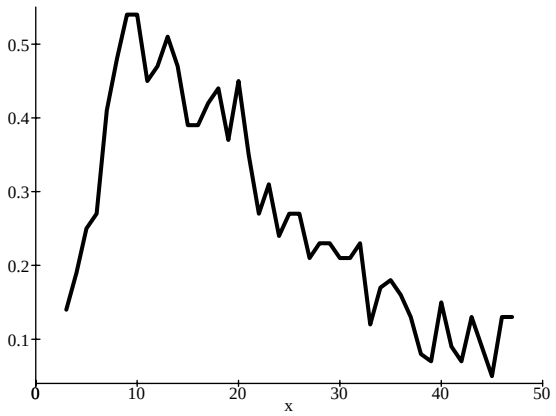


Figure 6: Experimental probability of being biconnected versus number of vertices.

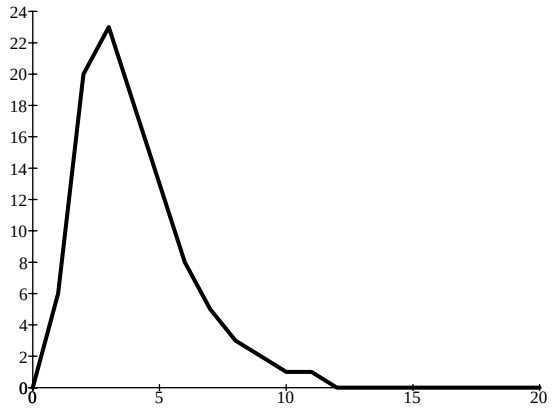


Figure 7: Distribution of the degrees of vertices.

planar graph on 100 vertices we would expect the values below:

degree	0	1	2	3	4	5	6	7	8	9	10	11	> 11
#vertices	0	6	20	23	18	13	8	5	3	2	1	1	0

In order to verify that our results do not depend on the initial state, other experiments were done with maximal planar graphs as initial states. Results seem to be equivalent. For instance, Figure 8 presents such a simulation for a graph with 100 vertices.

4 Properties of the random planar graph

We will denote by $R(G)$ the random planar subgraph of G, and when $G = K_n$, will abbreviate $R(K_n)$ to R_n.

If $e(R_n)$ denotes the expected number of edges in R_n, then our experimental evidence suggests that

$$\lim_{n\to\infty} n^{-1} e(R_n) = C$$

exists and that C is a constant fairly close to 2.

There is also a heuristic but wrong argument in support of the constant C being exactly 2. It runs as follows: the expected number of vertices of a planar map (as defined in [13]) with k edges is $\frac{k}{2} + 1$ (by duality and Euler's formula). Also, the expected number of vertices of a 3-connected unlabelled planar graph with k edges is $\frac{k}{2} + 1$ (because 3-connected planar graphs are in 1-to-1 correspondence with 3-connected planar maps, and the set of 3-connected planar maps is closed under duality).

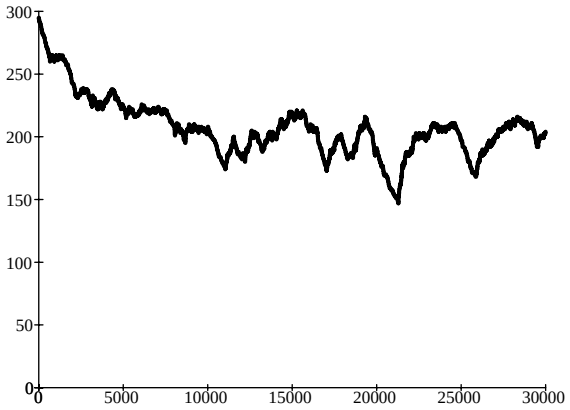


Figure 8: A typical simulation with a maximal planar graph as initial state.

It is obvious that $e(R_n) \leq 3n-6$ but better upper bounds seem difficult to find. What we can prove is:

Theorem 1 *The expected number of edges in R_n is at least $(3n-6)/2$.*

The proof is an almost immediate consequence of a more general result.

Theorem 2 *Let E be a finite set, and $\mathcal{D}$ a family of subsets of E such that*

1. *$X \in \mathcal{D}$, $Y \subset X \;\Rightarrow\; Y \in \mathcal{D}$,*
2. *all maximal members of $\mathcal{D}$ have same cardinality m.*

Then the expected number of elements of a random member of $\mathcal{D}$ is at least $m/2$.

Proof of Theorem 2. Let $\mathcal{A} = (A_1, ..., A_k)$ be the collection of maximal members of $\mathcal{D}$. We will prove the theorem by induction on k. Clearly, since each A_i has cardinality m, the theorem is true (with equality) when $k = 1$. Now assume it is true for families with k or fewer maximal members and consider the family $\mathcal{D}$ having $A_1, ..., A_{k+1}$ as maximal members (all of cardinality m).

Let $\mathcal{D}'$ be the family defined by

$$X \in \mathcal{D}' \Leftrightarrow \{X \subset A_i : 1 \leq i \leq k\}.$$

Then if $R(\mathcal{D})$ denotes a random member of $\mathcal{D}$, the expected size of R, written $\langle R(\mathcal{D})\rangle$, is given by

$$\begin{aligned} \langle R(\mathcal{D})\rangle &= \sum_{X\in\mathcal{D}} |X|/|\mathcal{D}| \\ &= \frac{\sum_{X\in\mathcal{D}'} |X| + \sum_{X\in\mathcal{D}\setminus\mathcal{D}'} |X|}{|\mathcal{D}'| + |\mathcal{D}\setminus\mathcal{D}'|}. \end{aligned}$$

We say that a collection of subsets $\mathcal{U}$ is *closed above* or *monotone increasing* if $X \in \mathcal{U},\ Y \supseteq X \ \Rightarrow\ Y \in \mathcal{U}$. We now use the following easy application of the FKG inequality [6].

Lemma 3 *Let E be any finite set, $\mathcal{U}$ any collection of subsets of E closed above, then*

$$\left(\sum_{X\in\mathcal{U}} |X|\right) / |\mathcal{U}| \geq \frac{1}{2}|E|.$$

Proof. Define f, g on 2^E by

$$\begin{aligned} f(Y) &= |Y| \quad Y \subseteq E, \\ g(Y) &= \begin{cases} 1 & Y \in \mathcal{U} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Then the FKG inequality gives for any positively correlated measure $\mu : 2^E \to \mathbf{R}^+$

$$\Sigma\mu(Y)\Sigma f(Y)g(Y)\mu(Y) \geq \Sigma f(Y)\mu(Y)\Sigma g(Y)\mu(Y)$$

where in all cases the sum is over all subsets of E. Taking $\mu(Y) = 1$ forall Y gives

$$2^{|E|} \sum_{Y\in\mathcal{U}} |Y| \geq \left(\sum_{Y\subset E} |Y|\right) |\mathcal{U}|$$

as required. □

Applying the lemma with $E = A_{k+1}$ gives

$$\frac{\sum_{X\in\mathcal{D}\setminus\mathcal{D}'} |X|}{|\mathcal{D}\setminus\mathcal{D}'|} \geq \frac{|A_{k+1}|}{2} = \frac{m}{2}.$$

Now, by the induction hypothesis,

$$\sum_{X\in\mathcal{D}'} \frac{|X|}{|\mathcal{D}'|} = \frac{c}{d} \geq \frac{m}{2}.$$

But if $c/d \geq m/2$ and $u/v \geq m/2$ then

$$\frac{c+u}{d+v} \geq \frac{m}{2}$$

which completes the proof of Theorem 2. □

Proof of Theorem 1 Take E to be the edge set of K_n in Theorem 2 and let a subset $X \in \mathcal{U}$ iff X is the edge set of a planar subgraph of K_n. □

We now turn to the relationship between R_n, the random planar graph, and the well understood random graph $G(n,p)$, (see Bollobás [1]).

First an elementary result which may be intuitively obvious but which we feel is worth spelling out. If π is any property of graphs, then we write $G \in \pi$ to signify that G has π. In other words we are identifying a property π with a class of graphs closed under isomorphism.

Lemma 4 *For any graph property π,*

$$Pr\{R_n \in \pi\} = Pr\{G(n,\frac{1}{2}) \in \pi \mid G(n,\frac{1}{2}) \text{ is planar}\}$$

Proof. Let us call $\pi(n)$ the set of graphs with n vertices having property π. Then

$$\begin{aligned} &Pr\{G(n,\frac{1}{2}) \in \pi \mid G(n,\frac{1}{2}) \text{ is planar}\} \\ &\quad = \frac{Pr\{G(n,\frac{1}{2}) \in \pi \text{ and } G(n,\frac{1}{2}) \text{ is planar}\}}{Pr\{G(n,\frac{1}{2}) \text{ is planar}\}} \\ &\quad = \frac{|\pi(n) \cap \mathcal{L}(n)|}{2^{\binom{n}{2}}} \cdot \frac{2^{\binom{n}{2}}}{l(n)} \\ &\quad = Pr\{R_n \in \pi\} \end{aligned}$$

□

An immediate consequence of this is the following.

We say that a property π is *monotone increasing* (respectively *decreasing*) if for any graph $G \in \pi$ any supergraph (respectively subgraph) of G having the same set of vertices also has π. Then using the FKG inequality we get:

Proposition 5 *Let π be any monotone property of graphs then*

(a) $Pr\{R_n \in \pi\} \geq Pr\{G(n,\frac{1}{2}) \in \pi\}$ *if π is decreasing*

(b) $Pr\{R_n \in \pi\} \leq Pr\{G(n,\frac{1}{2}) \in \pi\}$ *if π is increasing.*

For example, taking π to be the property of being connected, all it tells us is the intuitively obvious result that

$$Pr\{R_n \text{ is connected}\} \leq Pr\{G(n,\frac{1}{2}) \text{ is connected}\}$$

and it is well known that the right hand side tends to 1 as $n \to \infty$.

A more interesting comparison is between the behaviour of R_n, which we believe typically has about $2n$ edges, and the random graph $G(n,p(n))$, where $p(n) \sim 4/n$ is chosen so that the number of edges agree.

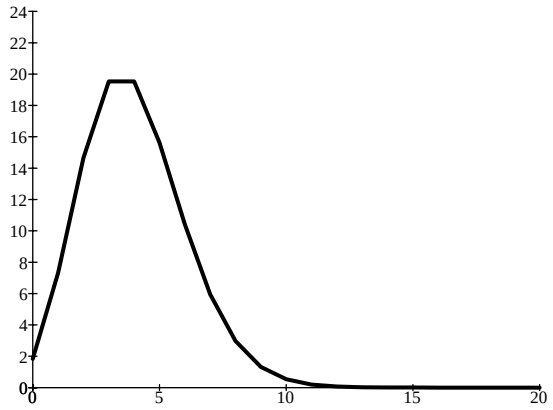


Figure 9: Distribution of the degrees of vertices in $G(n, 4/n)$.

Elementary results from random graph theory, see Bollobas [1, p.57], show that the number of vertices of degree k in $G(n,p)$ is asymptotically Poisson with parameter λ_k, and that

$$\lambda_k = n \left(\begin{array}{c} n-1 \\ k \end{array} \right) p^k (1-p)^{n-1-k}.$$

Thus the expected number of vertices of degree k in $G(n, 4/n)$ is

$$D_k(n) \sim n \frac{4^k}{k!} e^{-4} \text{ as } n \to \infty.$$

It is interesting to compare these, as shown in Figure 9, with our experimental results on degrees of R_n (Figure 7).

A more striking difference between R_n and $G(n, \frac{4}{n})$ is that almost certainly $G(n, \frac{4}{n})$ is disconnected for large n, whereas our simulations suggest R_n is connected. Intuitively this can be explained as follows, with about $2n$ edges to distribute, they have to be far more "spread out" in R_n than in $G(n, \frac{4}{n})$. This helps connectivity.

5 Connectivity properties

First we consider the probability $i(n)$ that a specific vertex of R_n, say vertex 1, is isolated. This is given by

$$i(n) = \frac{l(n-1)}{l(n)}.$$

We believe that $i(n)$ is monotone decreasing but note that showing this is equivalent to showing that

$$l(n)^2 \leq l(n+1)l(n-1),$$

in other words that the sequence $l(n)$ is log concave. Proving such inequalities tends to be extremely difficult.

Elementary computations for small n indicate also that

$$p_I(n) = Pr\{R_n \text{ has an isolated vertex}\}$$

decreases fairly rapidly. For example we have

n	1	2	3	4	5
$p_I(n)$	1	1/2	1/2	23/64	256/1023

We believe that $lim_{n\to\infty} p_I(n) = 0$ but have only been able to show:

Theorem 6 *The probability that R_n has an isolated vertex is $\Omega(n^{-10})$.*

Proof Suppose that (X_t) the Markov chain on n-vertex planar graphs is in equilibrium. Let (Z_t) be the Markov process defined by

$$Z_t = \begin{cases} 1 & \text{if } X_t \text{ has an isolated vertex} \\ 0 & \text{otherwise.} \end{cases}$$

There must be at least one vertex of degree ≤ 5 in X_t. Let it be v and let $i_1, ..., i_k$ be the neighbours of v. Then $Z_{t+5} = 1$ if the random mechanism governing X_t chooses, in some order, the positions $(v, i_1)...(v, i_k)$ in its next k transitions and avoids them in the remaining $5 - k$ transitions. The probability of this is at least Cn^{-10}. □

We now consider the probability that R_n, the random planar graph, is connected. If we denote this probability by $p_c(n)$ then clearly

$$p_c(n) = l_c(n)/l(n),$$

where $l_c(n)$ denotes the number of connected members of $\mathcal{L}(n)$, and for small n we get

n	2	3	4	5
$p_c(n)$	1/2	1/2	19/32	727/1023

From Theorem 6 $p_c(n) \leq 1 - Cn^{-10}$ as $n \to \infty$ but we believe that, as with the general random graph,

$$\lim_{n\to\infty} p_c(n) = 1.$$

We are unable to prove this but the following result indicates a certain drift towards there being only one connected component in R_n:

Proposition 7 *Let (Z_t) be the Markov process which counts the number of connected components in the graph X_t (having n vertices). Let us denote, for any $t \geq 0$ and for any $1 \leq i, j \leq n-1$, $P_t\{i \to j\} = Pr\{Z_{t+1} = j | Z_t = i\}$. Then, for $t \geq 0$ and $1 \leq k \leq n-1$,*

$$P_t\{k \to k+1\} \leq \frac{n-k}{\binom{n}{2}},$$

$$P_t\{k+1 \to k\} \geq \frac{k(n-k) + \frac{k(k-1)}{2}}{\binom{n}{2}}$$

and, for $1 \leq i, j \leq n$ and $|i-j| > 1$,

$$P_t\{i \to j\} = 0.$$

Proof. Suppose that X_t has k connected components and let i be the number of isthmi in X_t. Then obviously $P_t\{k \to k+1\} = i/\binom{n}{2}$. Now the number of isthmi in a graph with n vertices and k connected components is at most $n-k$ (this value can be reached only if the graph is a forest of trees). This gives the first inequality of the proposition.

On the other hand, if X_t has $k+1$ connected components with respective cardinalities $c_1, c_2, \ldots, c_{k+1}$, then the number of ways to add an edge in order that X_{t+1} has k connected components is

$$\sum_{\substack{i=1\ldots k+1 \\ j<i}} c_i c_j$$

The minimum of this expression is reached when $c_1 = c_2 = \ldots = c_k = 1$ and $c_{k+1} = n-k$ subject to the obvious constraints that $1 \leq c_i$ and $\Sigma c_i = n$ or some permutation of these values. To see this write

$$2\sum_{i<j} c_i c_j = (\Sigma c_i)^2 - \Sigma c_1^2 = n^2 - \Sigma c_i^2.$$

Hence the problem reduces to maximising Σc_i^2 subject to the same constraints. The result follows by standard dynamic programming arguments. Then

$$\sum_{\substack{i=1\ldots k+1 \\ j<i}} c_i c_j \geq k(n-k) + \frac{k(k-1)}{2}$$

and this gives the second inequality. □

We deduce immediately the corollary:

Corollary 8

$$\frac{P_t\{k \to k+1\}}{P_{t'}\{k+1 \to k\}} \leq \frac{1}{k} \quad \forall 1 \leq k \leq n-1, \quad t, t' \geq 0.$$

Proposition 9 *Let (Y_t) be an ergodic Markov chain having state space $\{1, \ldots, n\}$ and transition probabilities as in Proposition 7. Then, at equilibrium,*

$$Pr\{Y_t = 1\} \geq \frac{1}{e}.$$

Proof. Let $\pi = (\pi_1, \pi_2, \ldots \pi_n)$ denote the stationary distribution of (Y_t), and $M = (a_{ij})$ its transition matrix. Then π satisfies the following system of equations:

$$\begin{cases} \pi_1 &= a_{11}\pi_1 + a_{21}\pi_2 \\ \pi_2 &= a_{12}\pi_1 + a_{22}\pi_2 + a_{32}\pi_3 \\ \ldots & \\ \pi_k &= a_{k-1,k}\pi_{k-1} + a_{k,k}\pi_k + a_{k+1,k}\pi_{k+1} \quad (2 \leq k \leq n-1) \\ \ldots & \\ \pi_n &= a_{n-1,n}\pi_{n-1} + a_{n,n}\pi_n \end{cases}$$

Moreover, we know that M is stochastic, that is $a_{i,i-1} + a_{i,i} + a_{i,i+1} = 1 \ \forall i$. Then by induction we can prove that

$$\pi_{k+1} = \frac{a_{k,k+1}}{a_{k+1,k}}\pi_k \quad 1 \leq k \leq n-1,$$

and we deduce from Corollary 8 that

$$\pi_{k+1} \leq \frac{\pi_k}{k} \quad 1 \leq k \leq n-1.$$

Since $\sum_{i=1}^{n} \pi_i = 1$, we get, as required,

$$\begin{aligned} 1 &\leq \pi_1 \sum_{i=1}^{n} \frac{1}{(i-1)!} \\ &\leq \pi_1 e. \end{aligned}$$

□

6 Associated counting problems

In order to be more precise in our estimates above we need to understand better the behaviour of quantities such as $l_c(n)$ and $l(n)$. Crude counting arguments show that $\log l(n) = \theta(n \log n)$ as $n \to \infty$ and similarly for $l_c(n)$. Greater precision seems difficult. What we can prove is:

Lemma 10 $l_c(n) \geq (6n-16)l_c(n-1)$.

Proof. Let G be a graph of $\mathcal{L}_c(n-1)$. First, suppose that G is maximal planar. Let us count the number of ways to create a graph of $\mathcal{L}_c(n)$ by adding the vertex n. We can

- attach n to one vertex of G : there are $n-1$ possibilities;
- or attach n to the two extremities of one edge of G : there are $3n-9$ possibilities;
- or attach n to the three vertices of one face (in the unique planar representation of G): there are $2n-6$ possibilities if $n > 4$.

There are no more possibilities. This gives $6n-16$ ways of constructing a graph of $\mathcal{L}_c(n)$ from a maximal member of $\mathcal{L}_c(n-1)$.

If G is not maximal, then add some "virtual" edges to obtain a maximal graph G' which contains G, and then apply the previous constructions.

So the formula is true for $n > 4$. It is true too for $n \leq 4$ since $l_c(1) = 1$, $l_c(2) = 1$, $l_c(3) = 4$ and $l_c(4) = 38$. □

Corollary 11 $l(n) \geq (6n-15)l(n-1)$.

Proof. It suffices to add the case where n is connected to none of the other vertices. □

More generally, we can prove

Lemma 12 *If n is large enough that s exists satisfying the equation*

$$\left(\frac{n-1}{s-1}\right)^{\left(\frac{n-1}{s-1}-2\right)} \geq 6s-16.$$

then $l_c(n) \geq (6s-16)l(n-1)$.

For example this gives $l_c(n) \geq 20l(n-1)$ provided $n \geq 26$.

Proof. Let $G \in \mathcal{L}(n-1)$. First, fix $s > 1$ and suppose that there exists in G a connected component with at least s vertices. We can construct a graph of $\mathcal{L}_c(n)$ by attaching the vertex n to each connected component of G, using the method given in the proof of lemma 10. This gives at least $6s-16$ ways to construct the new graph.

Now suppose that all connected components of G have less than s vertices. We construct a graph of $\mathcal{L}_c(n)$ as follows: attach n to one vertex of each component of G, then attach together the neighbours of n in a way such that the subgraph containing only the neighbours of n is a tree. Let k be the number

of connected components of G: $k \geq \frac{n-1}{s-1}$. The number of labelled trees (Cayley trees) with k nodes is equal to k^{k-2}.

So, we have $l_c(n) \geq (6s-16)l(n-1)$, provided that $(\frac{n-1}{s-1})^{(\frac{n-1}{s-1}-2)} \geq 6s-16$. □

We believe that

$$\lim_{n\to\infty} \frac{l_c(n)}{l(n)} = 1$$

but cannot prove it. However, it would not be surprising if, for large n, the random planar graph had very few automorphisms, see for example the remark [13, page 138]. If this is the case it would be useful to have better understanding of the unlabelled counting problem. For this we obtain the following results:

Theorem 13 *There exists $\theta > 0$ such that*

$$\lim_{n\to\infty} (u_c(n))^{\frac{1}{n}} = \theta$$

and

$$\frac{256}{27} \leq \theta \leq 8\frac{256}{27}.$$

Proof. Let $\mathcal{M}$ be the set of *maximal* planar (unlabelled) graphs. Given an integer n, any graph of $\mathcal{U}(n)$ can be constructed from some graph of $\mathcal{M}(n)$ by deleting some edges. Thus, since there are $3n-6$ edges in a maximal planar graph, $u(n) \leq 2^{3n-6}m(n)$.

Tutte [12] has given the number of planar triangulations with n vertices. Let us consider the triangulations whose external face has degree 3. Their number is (using Tutte's notation)

$$\psi_{n-3,0} = \frac{2(4n-11)!}{(n-2)!(3n-7)!} \sim \frac{729\sqrt{2}\sqrt{3}}{2097152\sqrt{\pi}n^{5/2}} \left(\frac{256}{27}\right)^n.$$

Such a triangulation can be considered as a maximal planar graph in which one face[1] is distinguished (the external one), and the three vertices of this face are labelled a, b and c (see [12]). Then

$$u_c(n) \leq u(n) \leq 2^{3n-6}m(n) \leq 2^{3n-6}\psi_{n-3,0}$$

and finally we get

$$u_c(n) = O\left(n^{-\frac{5}{2}}\left(8\frac{256}{27}\right)^n\right).$$

Now let $\mathcal{B}$ Be the set of connected birooted graphs constructed as follows: take a graph which is not maximal planar in $\mathcal{U}_c$, choose two non adjacent vertices

[1] The faces are well defined here because there exists only one planar representation on the sphere of any maximal planar graph.

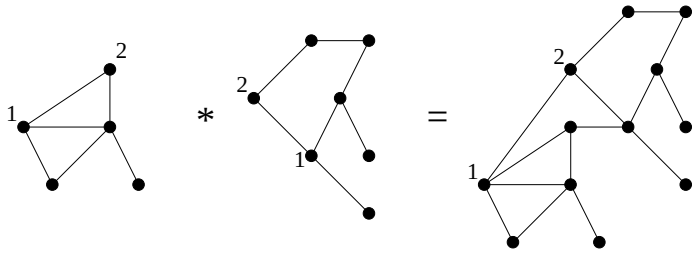


Figure 10: The operation $*$.

which lie in the same face in a planar representation of the graph and distinguish them as the first root r_1 and the second root r_2. Then create an edge between r_1 and r_2. It is easy to see that

$$b(n) = O\left(\left(8\frac{256}{27}\right)^n\right). \tag{1}$$

Now we define a binary operation in $\mathcal{B}$, as illustrated in Figure 10. Let G_1 and G_2 be in $\mathcal{B}$. The graph $G = G_1 * G_2$ is defined as follows: create an edge between the first root of G_1 and the second root of G_2 and an edge between the first root of G_2 and the second root of G_1. The first root of G_1 becomes the first root of G while the second root of G_2 becomes the second root of G.

We easily see that G belongs to $\mathcal{B}$: indeed, if we remove the edge between the two roots and forget the rooting, the resulting graph belongs to $\mathcal{U}_c$. On the other hand, $G_1 * G_2 = G_1' * G_2' \Rightarrow G_1 = G_1'$ and $G_2 = G_2'$. To see this, observe that, given $G = G_1 * G_2$, we find G_1 and G_2 by deleting the edge between the two roots of G and then, in the resulting graph, deleting the unique isthmus crossed by any path between the two roots of G. The second root of G_1 and the first root of G_2 are the extremities of this isthmus. Now notice that if G_1 and G_2 belong respectively to $\mathcal{B}(n_1)$ and $\mathcal{B}(n_2)$, then $G_1 * G_2$ belongs to $\mathcal{B}(n_1 + n_2)$. Thus

$$b(n_1 + n_2) \geq b(n_1)b(n_2) \quad \forall n_1, n_2. \tag{2}$$

From expressions (1) and (2) and the fundamental theorem of supermultiplicative functions we deduce that there exists θ such that $\lim_{n\to\infty} (b(n))^{1/n} = \theta$. Since the number of maximal planar graphs is such that $\lim_{n\to\infty} (m(n))^{1/n} = \frac{256}{27}$ we get

$$\lim_{n\to\infty} (u_c(n))^{1/n} = \theta.$$

□

We deduce the following corollary by standard asymptotic considerations (see for example [5]):

Corollary 14

$$\lim_{n\to\infty} u(n)^{1/n} = \theta.$$

This lends greater credence to our belief that as $n \to \infty$ the probability that R_n is connected tends to 1.

7 Conclusion

Of the many problems left open in the above the most pressing is deciding whether or not the Markov chain we propose is indeed rapidly mixing. Settling this would be greatly helped by a better knowledge of the random planar graph. However this seems a difficult combinatorial problem, and even a good upper bound on its number of edges is elusive.

Acknowledgement

We are very grateful for many interesting conversations with J.G. Penaud, helpful correspondence with N. Wormald and profitable discussions with D. Gardy.

References

[1] B. Bollobás. *Random graphs.* Academic Press Inc., 1985.

[2] R. Cori. *Un code pour les graphes planaires et ses applications.* Société Mathématique de France, 1975. Astérisque 27.

[3] R. Cori and B. Vauquelin. Planar maps are well labeled trees. *Canadian Journal of Mathematics*, 33(5):1023–1042, 1981.

[4] A. Denise. *Méthodes de génération aléatoire d'objets combinatoires de grande taille et problèmes d'énumération.* PhD thesis, Université Bordeaux I, 1994.

[5] P. Flajolet. Mathematical methods in the analysis of algorithms and data structures. In B. Egon, editor, *Trends in Theoretical Computer Science*, chapter 6. Computer Science Press, 1988.

[6] C. M. Fortuin, J. Ginibre, and P. N. Kasteleyn. Correlation inequalities on some partially ordered sets. *Communications in Mathematical Physics*, 22:89–103, 1971.

[7] V. A. Liskovets. Enumeration of nonisomorphic planar maps. *Selecta Math. Soviet.*, 4:304–323, 1985.

[8] V. A. Liskovets. Counting non-isomorphic planar maps: a general approach via rooted quotient maps. In B. Leclerc and J. Y. Thibon, editors, *Proceedings 7th FPSAC*, pages 363–370. Université de Marne-la-Vallée, 1995.

[9] S. Näher. LEDA manual. Technical Report MPI-I-93-109, Max-Planck-Institut für Informatik, 1993.

[10] N. J. Sloane and S. Plouffe. *The encyclopedia of integer sequences*. Academic Press Inc., New York, 1995.

[11] G. Tinhofer. Generating graphs uniformly at random. *Computing Supplementum*, 7:235–255, 1990.

[12] W. T. Tutte. A census of planar triangulations. *Canadian Journal of Mathematics*, 14:21–38, 1962.

[13] W. T. Tutte. A census of planar maps. *Canadian Journal of Mathematics*, 15:249–271, 1963.

[14] N. C. Wormald. Counting unrooted planar maps. *Discrete Mathematics*, 36:205–225, 1981.

[15] N. C. Wormald. On the number of planar maps. *Canadian Journal of Mathematics*, 33:1–11, 1981.

Problems and Remarks:

Each session of the Symposium was concluded by a period devoted to remarks and open problems. These are given in this section, in the chronological order in which they were presented.

1. Remark. For every $n \in \mathbb{N}$ let k_n be an integer with $0 \leq k_n \leq n$. For an arbitrary real number $\lambda_1 \in [0,1[$ define $\lambda_n := \frac{1}{n!}\left(\lambda_1 + \sum_{\nu=1}^{n-1} \nu! k_\nu\right)$ for all $n \in \mathbb{N}$. It is well known that then

$$f\left(\frac{m}{n!}\right) = e^{2\pi m \lambda_n i} \qquad (m \in \mathbb{Z}, n \in \mathbb{N})$$

defines a homomorphism f from $(Q,+)$ into the torus group $(T,\cdot)$ and that conversely every $f \in \mathrm{Hom}(Q,T)$ is obtained in this way.

Theorem. *The function f is continuous ifand only if*

i) *$k_n = 0$ for almost all $n \in \mathbb{N}$, or*

ii) *$k_n = n$ for almost all $n \in \mathbb{N}$.*

If it is continuous f has the form $f(x) = e^{2\pi c x i}$ $(x \in \mathbb{Q})$, where $c \leq 0$ in case i) and $c < 0$ in case ii).

References

[1] Hewitt, E. and Ross, K. A., *Abstract Harmonic Analysis I*, Springer, Berlin–Göttingen–Heidelberg, 1963, pp. 367–368 & 404–405.

[2] Maak, W., *Fastperiodische Funktionen*, Springer, Berlin–Göttingen–Heidelberg, 1950, pp. 89–90.

[3] Vietoris, L., *Zur Kennzeichnung des Sinus und verwandter Funkcionen durch Funktionalgleichungen*, J. Reine Angew. Math. **186** (1944), p. 4.

J. RÄTZ

2. Remark and problem. Using a recently developed method for solving certain types of inhomogeneous difference equations, we needed the following system of functional equations for $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$:

$$d(x+y,y) = d(x,y); \qquad d(x,y) = d(y,x). \tag{1}$$

L. Paganoni has proved that (1) has solutions different from identically constant functions, which we describe below.

Let H be a Hamel basis for the reals over the rationals $\mathbb{Q}$ and let H_0 be an arbitrary subset of H. Further, Iet $S_0 = V(H_0, \mathbb{Q}, +, \cdot)$ be the subspace of reals generated by H_0. We define the function $h : \mathbb{R} \to \mathbb{R}$ by:

$$h(x) = \begin{cases} 1 & \text{if } x \in S_0 \\ 0 & \text{if } x \notin S_0. \end{cases}$$

Then the function

$$d(x,y) = 1 - h(x)h(y)$$

fulfils conditions (1) and is obviously not constant.

Quite different ís the situation if we suppose continuity of d. Under this assumption all solutions of (1) are identically constant functions. This can be proved in a quite elementary way.

Problem. *Is it true that under the supposition of measurability the general solution of* (1) *is given by a.e. constant functions?*

I. Fenyő

3. Remark. In [1], Lorentz transformations in $\mathbb{R}^n$ (where $n \geq 3$) were characterized in a way for which there is no analogue in $\mathbb{R}^2$. For the indefinite metric

$$d((x_1, y_1), (x_2, y_2)) := (x_1 - x_2)^2 - (y_1 - y_2)^2$$

on $\mathbb{R}^2$, the bijective mappings $T : \mathbb{R}^2 \to \mathbb{R}^2$ with $T(0,0) = (0,0)$ satisfying

$$d((x_1, y_1), (x_2, y_2)) = 0 \qquad \text{iff} \qquad d(T(x_1, y_1), T(x_2, y_2)) = 0$$

are precisely those for which there exist $\delta \in \{-1, +1\}$ and $\phi, \psi : \mathbb{R} \to \mathbb{R}$ bijective such that $\phi(0) = 0 = \psi(0)$ and

$$T(x, y) = \left(\phi\left(\frac{x+y}{2}\right) + \psi\left(\frac{x-y}{2}\right), \delta\phi\left(\frac{x+y}{2}\right) - \delta\psi\left(\frac{x-y}{2}\right) \right)$$

for all $x, y \in \mathbb{R}$ ([2]). For these mappings, the condition

$$T(x_1, y_1) - T(x_2, y_2) = T(x_1, y_2) - T(x_2, y_2) \tag{E}$$

(where $x_1, x_2, y_1, y_2 \in \mathbb{R}$) is necessary and sufficient for T to be additive, while the condition that there exists $\sigma \in \{-1, +1\}$ such that whenever $x_1, x_2, y_1, y_2 \in \mathbb{R}$

$$d((x_1, y_1), (x_2, y_2)) > 0 \qquad \text{implies} \qquad \sigma d(T(x_1, y_1), T(x_2, y_2)) > 0 \tag{M}$$

is necessary and sufficient for T to be continuous.

References

[1] Borchers, H. J. and Hegerfeldt, G. C., *The structure of space-time transformations*, Comm. Math. Phys. **29** (1972), 259–266.

[2] A part of this result is due to R. Stettler (oral cammunication).

J. Rätz

4. Remark and Problem. Linearizing coordinate transformations for graph papers.

Semi-log and log-log graph papers provide a means of plotting exponential and monomial functions, respectively, as straight lines. This fact yields a convenient method for determining if empirical data are associated with one of these two types of functions.

The author [1] has developed analogous kinds of graph papers for functions satisfying the logistics equation:

$$\dot{x} = x(a - bx)$$

and the Gompertz equation:

$$\dot{x} = x(a - b \ln x).$$

Appropriately normalized solutions of these equations plot as straight lines on the graph papers. (Normalization is necessary since the general solutions of these

equations involve four arbitrary parameters, while straight line are determined by two.)

The form of all four kinds of graph paper was determined from the explicit form of the functions in question, rather than from the form of the corresponding functional or differential equation. (In the case of semi-log and log-log papers, of course, the "corresponding equations" are the appropriate multiplicative forms of Cauchy's equation.) This leads to the following open problem: how can the suitable coordinate spacing for the axes of the linearizing graph paper be obtained directly from the functional or differential equation without finding the explicit form of its solution?

To solve this problem we may require information about f^{-1} (whose functional equation is often obtainable from the functional equation for f, assuming f is invertible), and we may also require some means of numerically approximating the solution of the functional equation directly from the equation (see [2]).

References

[1] Snow, D. R., *Logistics and Gompertz graph papers*, Amer. Math. Soc. Abstracts **1** (1980), 468.

[2] Snow, D. R., *Remark: On numerical approximation methods for functional equations*, Aequationes Math. **15** (1977), 293–294.

D. R. Snow

5. Remark. This is a result by C. Wagner (lnstitute of Advanced Studies in the Behavioural Sciences, Stanford, CA. and the University of Tennessee, Knoxville), C. T. Ng, Pl. Kannappan, and myself. Let $f : [0,s]^n \to \mathbb{R}_+$ $(= \{x : x \geq 0\})$ be such that $f(0,0,\dots,0) = 0$ and

$$\sum_{i=1}^{m} x_{ij} = s \qquad (j = 1,2,\dots,n) \qquad \text{implies} \qquad \sum_{i=1}^{m} f(x_{i1}, x_{i2}, \dots, x_{in}) = s$$

(where $m > 2, n, s$ fixed). Then there exist $w_j \geq 0\,(j = 1,2,\dots,n)$, $\sum_{j=1}^{n} w_j = 1$ such that

$$f(x_1, x_2, \dots, x_n) = \sum_{j=1}^{n} w_j x_j \qquad \text{for all} \quad (x_1, x_2, \dots, x_n) \in [0,s]^n.$$

One of the possible interpretations is the following. A (say, grant) amount s should be allocated to m applicants. The decision maker (committee chairman) asks n advisors (committee members). The j-th advisor recommends that the i-th applicant obtain the amount x_{ij}. The decision maker allocates $f(x_{i1}, x_{i2}, \dots, x_{in})$ to the i-th applicant. The only conditions are that each advisor and also the decision maker allocate non-negative amounts to each applicant and the entire amount s is allocated by them to all appticants taken together, and the decision maker has to respect unanimous rejection (0 allocation) by all advisors. (Notice that the result compels the decision maker to respect also all other unanimous advice. The w_j in the result will be the "weight" of the j-th advisor and the final allocation will be a weighted arithmetic mean of the individual recommendations.) This is a characterization of the weighted arithmetic mean.

The cases $m \leq 2$ are also completely settled (then there are other solutions too).

The above results are stronger (the conditions weaker) than those reported at the 1979 meeting.

J. Aczél

6. Remark. Concerning Professor Fenyő's remark (Remark 2, these Proceedings) about non-constant and regular solutions $d : \mathbb{R}^2 \to \mathbb{R}^2$ of the system

$$d(x+y,y) = d(x,y), \qquad d(x,y) = d(y,x).$$

Consider Paganoni's solution $d := 1 - \chi_{V\times V}$ (with χ denoting characteristic functian), where V is an arbitrary subgroup of the additive group of all reals. If V is countable, then d is Borel measurable and locally integrable.

K. Baron

7. Remark. M. Laczkovich (University of Budapest) has solved Kemperman's problem (Aequationes Math. **4** (1970), 248–249) by proving that every solution of

$$2f(x) \le f(x+h) + f(x+2h)$$

(for all real x and all positive h) is nondcreasing.

J. Aczél

8. Problem. In connection with the construction of a collective preference from any n given individual preferences, the following problem arises:

Let $n, m \in \mathbb{N}; x^1, x^2, \dots, x^n \in S \subseteq \mathbb{R}^m$. Find all (continuous or even differentiable) vector-valued solutions $f^n : S^n \to S$ of the system of functional equations:

(1) $f^n(x^{\pi(1)}, \dots, x^{\pi(n)}) = f^n(x^1, \dots, x^n)$, for all permutations π and for all $x^1, \dots, x^n \in S$,

(2) $f^n(x, x, \dots, x) = x$ for all $x \in S$.

(3) $f^n(f^k(x^1, \dots, x^k), \dots, f^k(x^1, \dots, x^k), x^{k+1}, \dots, x^n) = f^n(x^1, \dots, x^k, x^{k+1}, \dots, x^n)$ for all natural numbers $k \le n$ and for all $x^1, \dots, x^n \in S$,

where additionally the i-th component of f^n (i.e. f_i^n) is a strictly monotonically increasing function of the i-th components of the vectors $x^1, \dots, x^n$ (i.e. of the variables $x_i^1, x_i^2, \dots, x_i^n$).

Remark. It is known that the functions f^n defined by

$$f_i^n(x^1, \dots, x^n) = g^{-1}\left(\frac{1}{n}\sum_{l=1}^{n} g(x_i^l)\right) \qquad (i = 1, 2, \dots, m)$$

with an arbitrary strictly monotonic (continuous or even differentiable) function g, defined on a proper subset $G \subset \mathbb{R}$, are solutions for any $n \in \mathbb{N}$.

F. Stehling

9. Remark. Let us consider the following functional equation:

$$f(x+y)[f(x) + f(y) - 1] = f(x)f(y) \qquad x, y \in S \tag{1}$$

where S is a given subset of the reals. l. Fenyő and L. Paganoni have proved the following theorem (see C. R. Math. Rep. Acad. Sci. Canada **2** (1980), 113–117).

Theorem 1. *The most general solution $f : S \to \mathbb{R}(S \subset \mathbb{R})$ of equation (1) is the following:*

$$f(x) = \begin{cases} 0 & \text{if } x \in S_0 \\ 1 & \text{if } x \notin S_1 \\ \frac{1}{1-g(x)} & \text{if } x \in S_2 \end{cases} \tag{2}$$

where S_0, S_1, S_2 are disjoint half-groupoids (some of which may be empty), whose union is the set S and which have the following properties:

$$S \cap (S_0 + S_2) \subset S_0, \tag{3a}$$

$$S \cap (S_1 + S_2) \subset S_1, \tag{3b}$$

and g is an arbitrary solution of the Cauchy functional equation which does not take the values 0 and 1.

Corollary. *If the domain of f contains the origin, then the most general solution of (1) is the characteristic function of a half-groupoid contained in S.*

The following problem suggested by J. Aczél arises: given an arbitrary subset S of the set of nonzero real numbers, is it in any case possible to cut it into three disjoint nonempty halfgroupoids so that conditions (3a) and (3b) are fulfilled? A partial answer to this problem is contained in the following theorem.

Theorem 2. *Let S be a subset of the nonzero reals; and let $V(S)$ be the rational subspace of $\mathbb{R}$ generated by S. If $\dim V(S) > 2$, then it is possible to find three disjoint nonempty halfgroupoids S_i $(i = 0, 1, 2)$ for which the conditions (3a) and (3b) are fulfilled.*

In a more general way we can state that the answer to the question above is surely affirmative if a maximal hyperplane H exists with $S \cap H \neq \emptyset$ and which divides all other elements of S into two disjoint parts.

I. Fenyő

10. Remark (concerning the talk of Professor J. Baker). Recently P. Cholewa (Silesian University, Katowice) has proved a generalization of Professor Baker's first result on a problem of E. Lukacs concerning the stability of the functional equation

$$f(x + y) = f(x)f(y).$$

In particular, if a nonempty set S, a positive real number δ, and a metric space (X, ρ) are given, then any function $f : S \to X$ fulfilling the condition

$$\rho(f(G(x, y)), H(f(x), f(y))) < \delta, \qquad x, y \in S,$$

has to be either (metrically) bounded or to satisfy the funetional equation

$$f(G(x, y)) = H(f(x), f(y)), \qquad x, y \in S,$$

where $G : S \times S \to S$ and $H : X \times X \to X$ are given functions subjected to some rather natural and fairly general assumptions.

R. Ger

11. Problem. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function with the properties that $f(0) = \frac{\partial f}{\partial x_i}(0) = 0$ $(i = 1, 2, \ldots, n)$, and that the rank of the matrix $\left|\frac{\partial^2 f}{\partial x_i \partial x_j}\right|$ is r at each point of $\mathbb{R}^n$.

Does there exist a linear coordinate transformation such that f can be expressed as a function of just r variables?

The answer is known to be affirmative in the case $r = 2$ and is negative on certain proper subsets of $\mathbb{R}^n$.

M. A. McKiernan

12. Remark. The functional equation

$$f(xy) + f(x + y) = f(xy + x) + f(y) \tag{1}$$

where $f : R \to G$, and R is a ring, G is a group, was introduced at the 17^{th} International Symposium on Functional Equations at Oberwolfach. At the present Symposium, R. Ger has announced some results on this equation, so it may be of interest to show (below) that if f satisfies (1), then the function taking x to $f(-x)$ satisfies Hosszú's functional equation:

$$f(xy) + f(x + y - xy) = f(x) + f(y). \tag{H}$$

So assume f satisfies (1). Let $y = -1$ in (1). Then we deduce

$$f(x - 1) = f(0) + f(-1) - f(-x). \tag{2}$$

Again in (1), let $x = u + 1$, $v = y - 1$, and use (2) to show

$$-f(u - v - uv) + f(u + v) = f(uv + v) - f(-v). \tag{3}$$

A final use of (1), with $x = v, y = u$ allows one to replace $f(uv + v)$ in (3) by $f(uv) + f(u + v) - f(u)$; and so (3) becomes:

$$f(uv) + f(u - v - uv) = f(u) + f(-v). \tag{4}$$

Replacing u by $-u$ we deduce

$$f(-(u + v - uv)) + f(-uv) = f(-u) + f(-v). \tag{5}$$

Hence, if we let $g(x) := f(-x)$, then g satisfies Hosszú's functional equation (H).

If R is a division ring with at least 5 elements, then solutions of Hosszú's equation satisfy

$$f(x + v) + f(0) = f(x) + f(y). \tag{6}$$

For such division rings R, therefore, the solutions of (1) are precisely tlte solutions of (6).

T. Davison

13. Remark. The characterization of the inner product in $\mathbb{R}^3$ given by J. Aczél ([1], p. 310, Satz l; [2], pp. 27–28) may be generalized as follows:

If $(X : \langle \cdot, \cdot \rangle)$ is a reat inner product space, let $\mathrm{SO}(X, 2)$ denote the set of all linear isometries $T : X \to X$ with a 2-dimensional invariant subspace M such that the restriction $T_M : M \to M$ of T is an orientation-preserving rotation of M (i.e. $T_M \in \mathrm{SO}(M : \langle \cdot, \cdot \rangle)$) and $Tx = x$ for every x in the orthogonal complement of M. Suppose that the mapping $g : X \times X \to \mathbb{R}$ has the properties

1) $g(Tx, Ty) = g(x, y)$ for all $x, y \in X$ and every $T \in \mathrm{SO}(X, 2)$.

2) $g(x_1 + x_2, y) = g(x_1, y) + g(x_2, y)$ for all $x_1, x_2, y \in X$,

3) $g(x, \lambda y) = \lambda g(x, y) = g(\lambda x, y)$ for all $x, y \in X$ and all $\lambda \in \mathbb{R}$.

Then the following statements can be proved:

a) If $\dim X \neq 2$, then $\langle x, y \rangle = 0$ implies $g(x, y) = 0$.

b) If $e, e' \in X$, with $\|e\| = \|e'\| = 1$, then $g(e, e) = g(e', e')$.

c) g is additive in its second variable, i.e. g is bilinear.

d) If $\dim X \neq 2$, g is symmetric.

e) If $\dim X \neq 2$, there exists $\alpha \in \mathbb{R}$ such that $g(x, y) = \alpha \langle x, y \rangle$ for all $x, y \in X$.

f) For the case $\dim X = 2$, the conclusions in a), d), and e) do not hold.

References

[1] Aczél, J., *Bemerkungen über die Multiplikation von Vektoren und Quaternionen*, Acta. Math. Acad. Sci. Hungar. **3** (1952), 309–316.

[2] Aczél, J., *Lectures on Functional Equations and their Applications*, Academic Press, New York–San Francisco–London, 1966.

J. RÄTZ

14. Remark. Some results of D. Zupnik on congruences and endomorphisms.

Let S be a set and n a positive integer. An n-ary operation on S is a function G from S^n into S. An equivalence relation $-$ on S is a congruence on S with respect to G if $x_i \sim y_i$ for $i = 1, 2, \ldots, n$ implies $G(x_1, \ldots, x_n) = G(y_1, \ldots, y_n)$. At the I976 Symposium at Lecce and Castro Marina, congruences were characterized in terms of functional equations (see Aequationes Math. **15** (1977), p. 284). Recently, D. Zupnik has developed this characterization and used it to obtain related results. Among these are the ones which follow.

Definition 1. A function f is an n-congruence on an n-ary operation G on S if $\operatorname{Dom} f = S$, f is idempotent, and

$$f(G(x_1, \ldots, x_n)) = f(G(f(x_1), \ldots, f(x_n))) \tag{1}$$

for all $x_1, \ldots, x_n$ in S.

Theorem 1. *An equivalence relation $-$ on S is a congruence on S with respect to the n-ary operation G on S if and only if there exists an n-congruence f on G such that $x - y$ iff $f(x) = f(y)$.*

An n-congruence f on G is always an endomorphism of the n-ary operation $f \circ G$, but need not be an endomorphism of G itself.

Theorem 2. *Let f be an n-congruence on the n-ary operation G. Let G_0 be the restriction of G to $(\operatorname{Ran} f)^n$. Then f is an endomorphism of G if and only if G_0 is an n-ary operation on $\operatorname{Ran} f$, or equivalently, if and only if*

$$f(G(x_1, \ldots, x_n)) = G(x_1, \ldots, x_n) \tag{2}$$

for all $x_1, \ldots, x_n$ in $\operatorname{Ran} f$.

Definition 2. An n-congruence f on an n-ary operation G admits an endomorphism of G if there exists an invertible function f_1 such that $\operatorname{Dom} f_1 = \operatorname{Ran} f$ and $f_1 \circ f$ is an endomorphism of G.

It is easily seen that if f is an endomorphism of G; then f admits an endomorphism of G. Furthermore, we have:

Theorem 3. *Let f be an n-congruence on an n-ary operation G. Then f admits an endomorphism of G if and only if there exists a subset S_1 of S such that*

a) $\operatorname{Card} S_1 = \operatorname{Card}(\operatorname{Ran} f)$,

b) *if G_1 denotes the restricrion of G to S_1^n, then G_1 is an n-ary operation on S_1,*

c) *the n-ary operation G_1 is isomorphic to the n-ary operation $f_2 \circ G_0$, where G_0 is as in the preceding theorem.*

A. Sklar

15. Remark. G. Fredricks (Texas Tech University) has proved the following result.

Let U be open in $\mathbb{R}^k$, A a smooth map of U into the gróup of symmetric $n \times n$ matrices, p and q nonnegative integers with $p + q \leq n$. Then there exists a smooth map $G : U \to GL(n)$ satisfying

$$G(\overline{x})A(\overline{x})G^T(\overline{x}) = \operatorname{diag}(1, \ldots, 1, -1, \ldots, -1, 0, \ldots, 0)$$

for all $\overline{x} \in U$ (with p 1's and q -1's in the diagonal matrix on the right) if A has p positive and q negative eigenvalúes at each $\overline{x} \in U$ and U is smoothly contractible.

B. Ebanks

16. Remark. The solution of a problem of Alsina, and its generalization.

Let F and G be functions from the unit square onto the unit interval that are associative, continuous, and non-decreasing in each place, and having no interior idempotents.

ln problem **P193** (Aequationes Math. **20** (1980), p. 308), C. Alsina proposed the equation

$$F(x, y) \cdot G(x, y) = xy.$$

Its only solutions consist of the one-parameter family

$$F_\alpha(x, y) = (x^{-\alpha} + y^{-\alpha} - 1)^{-\frac{1}{\alpha}}, \quad G_\alpha(x, y) = (x^\alpha + y^\alpha - x^\alpha y^\alpha)^{\frac{1}{\alpha}}, \qquad 0 < \alpha < \infty.$$

(Note the limiting case $F_\infty = \min$. $G_\infty = \max$.)

The related equation

$$F(x, y) + G(x, y) = x + y$$

is solved in my paper (Aequationes Math. **19** (1979),194–226). Extensions of this result to functions defined on unbounded intervals yield the solutions of the rnore general equation

$$H(F(x, y), G(x, y)) = H(x, y)$$

for any H which can be written $H(x, y) = k(h(x) + h(y))$, with continuous and monotonic h and k. In particular. when $h(0) = -\infty$ and $h(1) = 0$, the functions

$f_\alpha(x) = 1 - \exp[-\alpha h(x)]$, $0 < \alpha < \infty$, generate the family of solutions F_α.

M. J. Frank

17. Problems Let

$$D = \{(x, y) : x, y \in [0, 1[, x + y \leq 1\}$$

and let

$$D_0 = \{(x, y) : x, y, x + y \in]0, 1[\}$$

be its interior.

(1) Determine the general real-valued solutions f of

$$f(x, u) + (1 - x)f\left(\frac{y}{1 - x}, \frac{v}{1 - u}\right) = f(y, v) + (1 - y)f\left(\frac{x}{1 - y}, \frac{u}{1 - v}\right) \qquad (1)$$

on $D_0 \times D_0$.

(2) Determine the general (real-valued) solutions F, G, H, K (all four functions unknown) of

$$F(x) + (1 - x)^\alpha G\left(\frac{y}{1 - x}\right) = H(y) + (1 - y)^\alpha K\left(\frac{x}{1 - y}\right) \qquad (2)$$

on D_0, (α a fixed constant).

The second problem may lead to the solution of the first, but there may be a simpler way. Equation (1) has been solved on $D \times D$ and on $D \times D_0$ (the solutions are essentially diffierent): equation (2) has been solved on D.

(3) Determine the general solutions of (2) on D_0 when t^α is replaced on both sides by $m(t)$, $m :]0, 1[\to \mathbb{R}$ being an arbitrary multiplicative function ($m(tu) = m(t)m(u)$, $t, u \in]0, 1[$). Again, similar equations (but not this one) have been solved by Kannappan and Ng.

The general solution, on $D_0 \times D_0$, of equations similar to (1), but with $(1 - x)$ replaced by $(1 - x)^\alpha(1 - u)^\beta$ [and $(1 - y)$ by $(1 - y)^\alpha(1 - v)^\beta$] ($\alpha, \beta$ arbitrary constants but $(\alpha, \beta) \neq (0, 1), (1, 0)$), and of similar n-dimensional equations, have been determined by Ng.

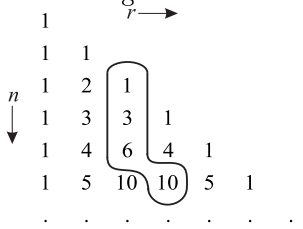
J. Aczél

18. Remark. A relationship of Catalan Numbers to Pascal's Triangle. We will call the identity

$$\binom{n+1}{r} = \sum_{k=0}^{r} \binom{n-r+k}{k}$$

the "stocking theorem" for Pascal's triangle, for the reason suggested by the figure below.

Figure 1:



(where in this case the overlay pattern illustrates the special.case $10 = 1 \cdot 6 + 1 \cdot 3 + 1 \cdot 1$ of the "stocking theorem").

The author has obtained generalizations of Pascal's triangle through the use of functional equations, and for each of these, there is a stocking theorem, analogous to the one above, which expresses each element of the generalized triangle as a certain linear combination of "higher" elements of the triangle. The coefficients in this linear combination are the first r elements of the stocking sequence associated with the triangle. (In the case of Pascal's triangle, the stocking sequence is simply $1, 1, 1, \ldots$.)

The generalized Pascal triangle T01 gives the number of ways of choosing n objects r at a time where, if an element is used at all, it must be used twice. The recurrence relation for this triangle is

$$C(n+1, r) = C(n, r) + C(n, r-2),$$

and the associated stocking sequence is

$$1, 0, 1, 0, 2, 0, 5, 0, 14, 0, 42, 0, 132, 0, \ldots .$$

which turns out to be the sequence of Catalan numbers

$$T_i = \frac{1}{i+1}\binom{2i}{i},$$

with zeros interspersed (see [1]).

For T01, it can be easily shown that $C(n, r) = 0$ for odd r. If we remove these zero columns from T01, we get Pascal's triangle T1, which means that the stocking theorem for T01 can be reinterpreted as the following statement relating the binomial coefftcients to the Catalan numbers T_i (defined above):

$$\binom{n+1}{r} = \sum_{i=0}^{r-1} T_i \binom{n-2i}{r-i-1},$$

where, for negative m, $\binom{m}{k}$ is the (unique) number determined by the Pascal recurrence relation

$$\binom{m+1}{k} = \binom{m}{k} + \binom{m}{k-1}$$

and by $\binom{m}{0} = 0$ for all integers m.

References

[1] Sloane, N. J. A., *Handbook of Integer Sequences*, Academic Press, New York, 1973.

D. R. Snow

19. Problem. Assume that

$$\sum_{i=1}^{k} \mu_i f(x + \phi_i(t)) = f(x) \tag{1}$$

for all $x \in \mathbb{R}^n, t \in \Delta \subset \mathbb{R}$: where $\sum_{i=i}^{k} \mu_i = 1, \mu_i > 0$ for $i = 1, \ldots, k$, and there exists an $\alpha \in \Delta$ such that $\phi_i(\alpha) = 0$ for $i = 1, \ldots, k$.

If the set of $\phi_i'(\alpha)$ (for $i = 1, \ldots, k$) spans $\mathbb{R}^N$, then every locally integrable solution f of (1) is a C^∞ function (see [1]).

Question. *Are all the locally integrable solutions of (1) C^∞ functions if $\{(\phi_i'(\alpha) : i = 1, \ldots, k\}$ does not span $\mathbb{R}^N$, but $\{(\phi_i''(\alpha) : i = 1, \ldots, k\}$ does?*

References

[1] Światak, H., *Criteria for the regularity of continuous and locally integrable solutions of a class of linear functional equarions*, .Aequationes Math. **6** (1971), 170–187.

H. Światak

20. Problem. Find all functions $F :]0, \infty[\to \mathbb{R}$ satisfying:

$$F(xy) = F(x)F(y) \qquad \text{and} \qquad F(x + y) \leq F(x) + F(y)$$

for all $x > 0$ and $v > 0$.

This problem arises in the calculation of entropy functions of degree $\alpha < 1$. Discontinuous solutions of the system are known to exist.

Gy. Maksa

21. Remark. It has been pointed out by V. I. Arnold and A. A. Kirilov that the function $\mathrm{Min}(x, y)$ admits no representation of the form

$$\mathrm{Min}(x, y) = f(g(x) + g(y)),$$

where f and g are continuous. A stronger result is easily established:

Theorem. *Let $A = [a, b]$ be a subinterval of the extended real line, and let $T : A \times A \to A$ define a semigroup on A such that for some $a < \overline{x} < b$,*

$$T(a, a) = a, \qquad T(\overline{x}, \overline{x}) = \overline{x}, \qquad T(b, b) = b.$$

Then there are no continuous functions f, g such that T can be represented in the form $T(x, y) = f(g(x) + g(y))$.

G. Krause

22. Remark. The following problem of Colin Rogers arises in gas dynamics in connection with the theory of Bäcklund transformations. Given real constants α, a, b, c, d, find smooth solutions $\phi :]0, \infty[\to \mathbb{R}$ such that

$$\phi(x) = \alpha(x + c)^2 \left[\phi\left(a + \frac{b}{x + c} \right) + d \right], \qquad x > 0. \tag{1}$$

We assume a, b, and c are such that $a + \frac{b}{x+c}$ is defined and positive whenever $x > 0$. In the homogeneous case ($d = 0$) the real analytic solutions of (1) can be found explicitly (they are rational functions in nontrivial cases) with the aid of the following theorem.

Theorem 1. *Let D be an open connected subset of $\mathbb{C}$ (the complex numbers) and let $g : D \to D$ be analytic and have a fixed point z_0 such thnt $0 < |g'(z_0)| < 1$ and $g^k(z) \longrightarrow z_0$ as $k \longrightarrow +\infty$ for every $z \in D$. Also let $f : D \to \mathbb{C}$ be anatytic with $f(z_0) = 1$, let $\lambda \in \mathbb{C}$ and suppose that $\phi : D \to \mathbb{C}$ is analytic and such that*

$$\lambda\phi(z) = f(z)\phi(g(z)), \qquad z \in D. \tag{2}$$

Then there exist analytic functions $F, G : D \to \mathbb{C}$ such that

(i) *if $\lambda \neq (g'(z_0))^k$ for all $k = 0, 1, 2, \ldots$, then $\phi \equiv 0$ and*

(ii) *if $\lambda = (g'(z_0))^k$ for some $k = 0, 1, 2, \ldots$ then there exists $\gamma \in \mathbb{C}$such that $\phi(z) = \gamma F(z)[G(z)(z - z_0)]^k$, $z \in D$.*

If we let $\Phi_k(z) = F(z)[G(z)(z - z_0)]^k$, for $z \in D, k = 0, 1, 2, \ldots$, then we can prove:

Theorem 2. *Given $h : D \to \mathbb{R}$ analytic, there exist $\delta > 0$ and a complex sequence $\{c_k\}_{k=0}^{+\infty}$ such that*

$$h(z) = \sum_{k=0}^{+\infty} c_k \Phi_k(z)$$

for $|z - z_0| < \delta$. Moreover the convergence is almost uniform on $\{z \in D : |z - z_0| < \delta\}$.

Using Theorem 2, one can determine the real analytic solutions of (1) in the nonhomogeneous case.

J. A. Baker

23. Remark. A function f, holomorphic in $D = \{z : |z| < 1\}$, is said to be annular in case there is a sequence $\{J_n\} \subset D$ of Jordan curves about 0 such that

$$\lim_{n\to\infty} \min\{|f(z)| : z \in J_n\} = \infty.$$

One can base a prooi for the annuiarim of

$$f(z) = \sum_{n=0}^{\infty} a^{cn} z^{a^n} \tag{1}$$

(where $c > 0, a = a(c)$, a sufiiciently large integer), on known methods and the fact that f satisfies the functional equation

$$f(z) - a^c f(az) = z.$$

Hardy and Littlewood in 1916 related (1) via a functinonal equation to

$$F(\zeta) = \sum_{n=1}^{\infty} n^{\delta-} e^{\beta n \log n} \zeta^n \tag{2}$$

($\delta > 0, \beta > 0$ certain constants), and thereby one can show that (2) is also annular. Fatou showed that for certain rational functions, for example

$$R(z) = \frac{z(z-s)}{1-sz}$$

c complex. $0 < |s| < 1$, the nontrivial analytic solutions of the Schröder equation

$$f(R(z)) = -sf(z)$$

are annular.

I would appreciate hearing of othter connections between functional equations and annular functions.

F. Carroll

24. Problem. Let $(F, +, \cdot)$ be a system with the following properties:

I. $(F, +)$ is a toop (with identity 0).

II. $(F - \{0\}, \cdot)$ is a group.

III. $(a+b) \cdot c = a \cdot c + b \cdot c$ and $c \cdot 0 = 0$, for all $a, b, c \in F$.

IV. (Limited associativity) $(x+a)+b$ is equal to $x+(a+b)$ if $b+a=0$, and is equal to $x(b+a)^{-1}(a+b)+(a+b)$ otherwíse.

Question. *Do the conditions I-IV imply that $(F, +)$ is an abelian group?*

The answer is known to be affirmative in case F has finite cardinality, or under some other additional assumptions, such as $a(1+1) = a+a$; or $1+1+1=0$.

W. Leissner

25. Remark. Solution of Problem 17 (2) (of these Proceedings).

In answer to a problem of J. Aczél, we have proved the following:

Theorem. *Ler $\alpha \in \mathbb{R}$ be fixed. $D_0 = \{(x,y) \in \mathbb{R}^2 : x, y, x+y \in]0,1[\}$. The functions $F, G, H, K :]0,1[\to \mathbb{R}$ satisfy*

$$F(x) + (1-x)^\alpha G\left(\frac{y}{1-x}\right) = H(y) + (1-y)^\alpha K\left(\frac{x}{1-y}\right)$$

for all $(x,y) \in D_0$ if and only if, for all $x \in]0,1[$,

$$F(x) = \begin{cases} \phi(x) + \phi(1-x) + a_1 x + a_2(1-x) + a_3 & \text{if } \alpha = 1 \\ l_1(1-x) + l_2(x) + a_1 & \text{if } \alpha = 0 \\ d(x) + a_1 x^2 + a_2(1-x)^2 + a_3 & \text{if } \alpha = 2 \\ a_1 x^\alpha + a_2(1-x)^\alpha + a_3 & \text{otherwise} \end{cases}$$

$$G(x) = \begin{cases} \phi(x) + \phi(1-x) + a_1' x \\ \quad +(a_1 - b_1 + a_3 - b_3 - b_1' + a_1' + b_2')(1-x) \\ \quad +b_1 - a_2 - a_3 + b_3 - a_1' & \text{if } \alpha = 1 \\ l_1(1-x) + l_3(x) - l_3(1-x) + b_1 - a_1 + b_1' & \text{if } \alpha = 0 \\ \quad -d(x) + b_1 x^2 + a_2'(1-x)^2 - a_2 & \text{if } \alpha = 2 \\ b_1 x^\alpha + a_2'(1-x)^\alpha - a_2 & \text{otherwise} \end{cases}$$

$$H(x) = \begin{cases} \phi(x) + \phi(1-x) + b_1 x + b_2(1-x) + b_3 & \text{if } \alpha = 1 \\ l_1(1-x) + l_2(1-x) + l_3(x) - l_3(1-x) + b_1 - a_1 + b_1' & \text{if } \alpha = 0 \\ -d(x) + b_1 x^2 + b_2(1-x)^2 + a_3 & \text{if } \alpha = 2 \\ b_1 x^\alpha + b_2(1-x)^\alpha + a_3 & \text{otherwise} \end{cases}$$

$$K(x) = \begin{cases} \phi(x) + \phi(1-x) + b_1' x + b_2'(1-x) & \\ \quad + a_1 + a_3 - b_2 - b_3 - b_1' & \text{if } \alpha = 1 \\ l_1(1-x) + l_2(x) - l_3(1-x) + b_1' & \text{if } \alpha = 0 \\ d(x) + a_1 x^2 + a_2'(1-x)^2 - b_2 & \text{if } \alpha = 2 \\ a_1 x^\alpha + a_2'(1-x)^\alpha - b_2 & \text{otherwise} \end{cases}$$

where $\phi :]0, \infty[\to \mathbb{R}$ *satisfies*

$$\phi(xy) = x\phi(y) + y\phi(x),$$

for all $x, y \in]0, \infty[$, $l_j :]0, \infty[\to \mathbb{R}$ *satisfies*

$$l_i(xy) = l_i(x) + l_i(y)$$

for all $x, y \in]0, \infty[$ *and* $i = 1, 2, 3$, *the function* d *is a real derivation and* a_i, b_i, a_k', b_k' $(i = 1, 2, 3; k = 1, 2)$ *are arbitrary real constants.*

Gy. Maksa

26. Remark Solution of Problem 17 (1) (of these Proceedings).

ln view of Gy. Maksa's solution (see Remark 25 above) to Problem 17 (2), the equation

$$f(x,u) + (1-x) f\left(\frac{y}{1-x}, \frac{v}{1-u}\right) = f(y,v) + (1-y) f\left(\frac{x}{1-y}, \frac{u}{1-v}\right) \tag{1}$$

for all $(x, y) \in D_0, (u, v) \in D_0$, where

$$D_0 = \{(s,t) : s, t, s+t \in]0,1[\}.$$

can be solved as follows.

Keeping u, v constant, (1) goes over into

$$F(x) + (1-x)^\alpha G\left(\frac{y}{1-x}\right) = H(y) + (1-y)^\alpha K\left(\frac{x}{1-y}\right)$$

for all $(x, y) \in D_0$.

From Maksa's solution of this equation ($\alpha = 1$).

$$f(s,u) = F(s) = \phi(s) + \phi(1-s) + a_1 s + b_1,$$
$$f(s,y) = H(s) = \phi(s) + \phi(1-s) + a_2 s + b_2,$$

that is, letting u vary again,

$$f(x,u) = \phi(x) + \phi(1-x) + A(u)x + B(u). \tag{2}$$

Here

$$\phi(xy) = x\phi(y) + y\phi(x) \tag{3}$$

(for $x, y \in]0,1[$) and in consequence,

$$\phi\left(\frac{s}{t}\right) = \frac{t\phi(s) - s\phi(t)}{t^2}$$

(where $s, t, \frac{s}{t} \in]0,1[$).

By substituting (2) into (1), we get

$$\begin{aligned}
&\phi(x) + \phi(1-x) + A(u)x + B(u) + \phi(y) - \phi(1-x) + \phi(1-x-y) \\
&\quad + A\left(\frac{v}{1-u}\right) y + B\left(\frac{v}{1-u}\right)(1-x) \\
&= \phi(y) + \phi(1-y) + A(v)y + B(v) + \phi(x) - \phi(1-y) + \phi(1-x-y) \\
&\quad + A\left(\frac{u}{1-v}\right) x + B\left(\frac{u}{1-v}\right)(1-y).
\end{aligned}$$

After cancellations and comparing the coefficients of x and the terms independent of x and y on both sides we get

$$A(u) = A\left(\frac{u}{1-v}\right) + B\left(\frac{v}{1-u}\right)$$

and

$$B(u) - B\left(\frac{v}{1-u}\right) = B(v) + B\left(\frac{u}{1-v}\right)$$

for all $(u,v) \in D_0$. By adding these two equations and writing $C = A + B, p = \frac{u}{1-v}, q = \frac{v}{1-u}$, $(p, q \in]0,1[$, but otherwise arbitrary), we get

$$C(pq) = C(p) + B(1-q) \qquad (p, q \in]0,1[).$$

This is a Pexider type equation with the general solution (cf. [1]) $B(1-q) = l(q), C(u) = l(u) + c$. So

$$B(u) = l(1-u), \qquad A(u) = l(u) - l(1-u) + c$$

where l is an arbitrary solution of

$$l(uv) = l(u) + l(v) \qquad (u, v \in]0,1[), \tag{4}$$

(cf [2],[3]). Since the converse part is obvious, we have proved the following.

Theorem. *The general solution of* (1) *is given by*

$$f(x,u) = \phi(x) + \phi(1-x) + xl(u) + (1-x)l(1-u) + cx,$$

where c is an arbitrary constant and ϕ and l are arbitrary solutions of (3) *and* (4) *respectively.*

Note. *By interchanging (x,y) and (u,v), we can also use Maksa's $\alpha = 0$ result for the same purpose.*

References

[1] Aczél, J., *On a generalization of the functional equation of Pexider*, Publ. Inst. Math. (Beograd) **4** (18) (1964), 77–80.

[2] Aczél, J. and Kannappan, PL., *General two-place information functions*, Submitted to Proc. Roy. Soc. Edinburgh Sect. A.

[3] Aczél, J. and Ng, C. T., *On general information functions*, Submitted to Utilitas Math.

J. Aczél

On the number of standard and of effective multiple alignments

A. Dress, B. Morgenstern, J. Stoye

March 30, 1998

Abstract

We study the number of all possible alignments of N sequences, $N \geq 2$, for two distinct alignment concepts proposed in the literature – standard alignments and effective alignments (consistent equivalence relations). Recursion formulae are developed to calculate these numbers. For standard alignments and for effective alignment of just two sequences an explicit formula is also presented. The number of all effective alignments of a given site space is shown to be related to Stirling numbers of second kind.

1 Introduction

Sequence alignment is one of the most important tools for data analysis in molelcular biology. There are different notions of what an alignment is: By standard theory, an alignment of N sequences $s_1, \ldots, s_N$ of length $L_1, \ldots, L_N$ is defined to be an $N \times L$ matrix A with $\max(L_1, \ldots, L_N) \leq L \leq \sum_{1 \leq i \leq N} L_i$ whose rows are obtained from the original sequences by insertion of so-called 'blanks' or 'gap characters' – with the additional requirement that no column of the matrix A consists exclusively of blanks (cf. [1]; p. 186).

Recently, Morgenstern *et al.* [2] have proposed a different way of defining alignments (see also [3] and [1], p. 188, for the case of two sequences and [4] for a thorough discussion of this concept for any number of sequences). In their definition, an alignment of the sequences $s_1, \ldots, s_N$ is a *consistent equivalence relation* defined on the so-called *site space* $\mathcal{S} := \{[i|j] \mid 1 \leq i \leq N, 1 \leq j \leq L_i\}$. This definition avoids a certain redundancy inherent in the standard definition and allows to apply the mathematical theory of sets and relations to investigate the *state space* associated with an alignment problem. To distinguish these alignments from standard alignments, we will refer to them as *effective alignments*.

No matter which definition is preferred, in either case the alignment problem is the problem of finding an *optimal alignment* – according to some well-defined criterion – and the search space for this optimization problem is the set of all possible alignments of a given set of sequences.

Therefore, it seems to be worthwile to study the structure of this space in more detail. In this paper, we show how to calculate the number of all possible alignments of N sequences. We generalize the results of Laquer [5] and Waterman [1] who solved this problem for the special case of $N = 2$ sequences. We derive recursive functions to calculate both, the number of standard alignments and the number of effective alignments. We also present explicit formulae for the number (i) of standard alignments and (ii) of effective alignments of just two sequences.

Although these numerical values themselves are of minor interest to biologists, our study might still be of some use as it sheds light on the structure of the *state space* associated with the alignment problem.

2 The number of standard alignments

Assume that we are given N sequences $s_1, s_2, \ldots, s_N$ of length $L_1, L_2, \ldots, L_N$. Then, clearly, there exist, for any given $L \geq \max(L_1, \ldots, L_N)$, exactly

$$f^+(L) = f^+(L_1, \ldots, L_N; L) := \prod_{i=1}^{N} \binom{L}{L_i}$$

standard alignments of total length L provided we allow columns consisting of blanks, only.

More precisely, given a subset X of $\{1, \ldots, L\}$ of cardinality

$$x \leq L - \max(L_1, \ldots, L_N),$$

there exist

$$f^+(X, L) = f^+(L_1, \ldots, L_N; X, L) := \prod_{i=1}^{N} \binom{L-x}{L_i}$$

such alignments with at least all those columns consisting of blanks only which are indexed by elements $j \in X$.

Consequently, by Möbius inversion [6], the sum

$$\sum_{0 \leq x \leq L - \max(L_1, \ldots, L_N)} (-1)^x \binom{L}{x} \prod_{i=1}^{N} \binom{L-x}{L_i}$$

coincides with the number $F(L_1, \ldots, L_N; L)$ of all standard alignments of total length L without any column consisting of blanks only.

Remark: The standard proof for this fact runs as follows: for $X \subseteq \{1, \ldots, L\}$ as above, let $f(X, L)$ denote the number of alignments of total length L with exactly those columns consisting of blanks only which are indexed by elements $j \in X$; then, if $x := \#X$, we have

$$f^+(X, L) = \prod_{i=1}^{N} \binom{L-x}{L_i} = \sum_{X \subseteq Y \subseteq \{1, \ldots, L\}} f(Y, L)$$

and hence

$$\sum_{x\geq 0}(-1)^x\binom{L}{x}\prod_{i=1}^{N}\binom{L-x}{L_i} = \sum_{X\subseteq\{1,\ldots,L\}}(-1)^{\#X}f^+(X,L) =$$
$$= \sum_{X\subseteq\{1,\ldots,L\}}(-1)^{\#X}\sum_{X\subseteq Y\subseteq\{1,\ldots,L\}}f(Y,L) =$$
$$= \sum_{Y\subseteq\{1,\ldots,L\}}f(Y,L)\sum_{X\subseteq Y}(-1)^{\#X} = f(\emptyset,L) = F(L_1,\ldots,L_N;L).$$

Clearly, this implies that the number $F(L_1,\ldots,L_N)$ of all standard alignments without any column consisting of blanks only coincides with the double sum

$$\sum_{L\geq 0}\sum_{x\geq 0}(-1)^x\binom{L}{x}\prod_{i=1}^{N}\binom{L-x}{L_i}$$

where the sum could be taken over all L and x, yet non-zero terms will arise only for $\max(L_1,\ldots,L_N)+x\leq L\leq L_1+\ldots+L_N$.

As any such alignment has a first column involving a well-defined non-empty subset V of $\{1,\ldots,N\}$ of rows without blanks, it is clear that for $L,L_1,\ldots,L_N>0$, we also have the Pascal-triangle type recursion formulae

$$F(L_1,\ldots,L_N;L) = \sum_{\emptyset\subsetneq V\subseteq\{1,\ldots,N\}}F(L_1-\chi_V(1),\ldots,L_N-\chi_V(N);L-1)$$

and

$$F(L_1,\ldots,L_N) = \sum_{\emptyset\subsetneq V\subseteq\{1,\ldots,N\}}F(L_1-\chi_V(1),\ldots,L_N-\chi_V(N))$$

with

$$\chi_V:\{1,\ldots,N\}\to\{0,1\}:i\mapsto\begin{cases}1 & \text{if } i\in V,\\ 0 & \text{else}\end{cases}$$

the characteristic function of $V\subseteq\{1,\ldots,N\}$, as usual. Together with

$$F(1;1)=F(1):=1,$$
$$F(1;L):=0 \text{ for } L>1,$$

and

$$F(L_1,\ldots,L_N;L):=F(L_1,\ldots,L_{i-1},L_{i+1},\ldots,L_N;L)$$

as well as

$$F(L_1,\ldots,L_N):=F(L_1,\ldots,L_{i-1},L_{i+1},\ldots,L_N)$$

whenever $L_i:=0$ for some $i\in\{1,\ldots,N\}$, this recursion formula can of course also be used to compute the values of $F(L_1,\ldots,L_N;L)$ and $F(L_1,\ldots,L_N)$ in an efficient way.

Remark: Note that a similar argument establishes the recursion formula

$$f^+(L_1,\ldots,L_N;L) = \sum_{V\subseteq\{1,\ldots,N\}}f^+(L_1-\chi_V(1),\ldots,L_N-\chi_V(N);L-1).$$

3 The number of effective alignments

Let us now denote by $G(L_1, \ldots, L_N)$ the number of *effective* alignments of the given sequences, that is, of equivalence relations A defined on the set $\mathcal{S} := \{[i|j] \mid 1 \leq i \leq N, 1 \leq j \leq L_i\}$ with the property that there exists a partial order $\preceq$ defined on the set $\mathcal{S}/A$ of A-equivalence classes $A(x), A(y), \ldots (x, y \in \mathcal{S})$ satisfying the consistency condition

$$(*) \qquad A([i|j]) \preceq A([i|k]) \iff j \leq k$$

for all $i \in \{1, \ldots, N\}$ and $j, k \in \{1, \ldots, L_i\}$. Note that, if any such partial order exists, there exists a unique smallest one which can be defined as the transitive closure of the relation defined by $(*)$ and which will be denoted by "$\preceq_A$".

In case $N = 1$, we clearly have $G(L_1) = 1$; and – just as above – we have

$$G(L_1, \ldots, L_N) = G(L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_N)$$

in case $L_i = 0$ for some $i \in \{1, \ldots, N\}$. It is also easy to see (cf. [1], p. 188) that, in case $N = 2$, we have

$$G(L_1, L_2) = \binom{L_1 + L_2}{L_1} = \binom{L_1 + L_2}{L_2}$$

because – in view of the identity

$$\sum_{l \geq 0} \binom{L_1 + L_2}{l} x^l = (1+x)^{L_1 + L_2} = (1+x)^{L_1} (1+x)^{L_2} =$$

$$= \left(\sum_{l_1 \geq 0} \binom{L_1}{l_1} x^{l_1} \right) \left(\sum_{l_2 \geq 0} \binom{L_2}{l_2} x^{l_2} \right) =$$

$$= \sum_{l \geq 0} \left(\sum_{l_1 + l_2 = l} \binom{L_1}{l_1} \binom{L_2}{l_2} \right) x^l$$

– this number is well known to coincide with

$$\sum_{l_1 + l_2 = L_1} \binom{L_1}{l_1} \binom{L_2}{l_2} = \sum_{l_1 + l_2 = L_1} \binom{L_1}{L_1 - l_1} \binom{L_2}{l_2} = \sum_{k \geq 0} \binom{L_1}{k} \binom{L_2}{k}$$

and because any effective alignment A of two sequences is uniquely determined by the two subsets $K_1 \subseteq \{1, \ldots, L_1\}$ and $K_2 \subseteq \{1, \ldots, L_2\}$ which are defined by

$$K_1 := \{j_1 \in \{1, \ldots, L_1\} \mid \text{ there exists } j_2 \in \{1, \ldots, L_2\} \text{ with } [1|j_1] \overset{A}{\sim} [2|j_2]\}$$

and

$$K_2 := \{j_2 \in \{1, \ldots, L_2\} \mid \text{ there exists } j_1 \in \{1, \ldots, L_1\} \text{ with } [2|j_2] \overset{A}{\sim} [1|j_1]\}$$

which can be chosen freely in $\{1,\ldots,L_1\}$ and $\{1,\ldots,L_2\}$ subject only to the condition that they have to have the same cardinality.

In the general case $N \geq 1$, we can at least derive a Pascal-triangle type recursion formula for $G(L_1,\ldots,L_N)$. To this end, consider a *partial partition* $\mathcal{V} = \{V_1,\ldots,V_k\}$ of $\{1,\ldots,N\}$, that is a non-empty set of non-empty and pairwise disjoint subsets $V_1,\ldots,V_k$ of $\{1,\ldots,N\}$ and define $G(L_1,\ldots,L_N;\mathcal{V})$ to denote the number of effective alignments A for which $\mathcal{V}$ coincides with the set

$$\mathcal{V}(A) := \{V \subseteq \{1,\ldots,N\} \;\big|\; \{[i|1] \;\big|\; i \in V\} \in \mathcal{S}/A\}.$$

Clearly, $\mathcal{V}(A)$ is non-empty because every A-equivalence class contained in $\mathcal{S}$ which is minimal with respect to the partial order $\preceq_A$ defined by A is necessarily of the form $\{[i|1] \;\big|\; i \in V\}$ for some non-empty subset $V \subseteq \{1,\ldots,N\}$.

So, we have

$$G(L_1,\ldots,L_N) = \sum_{\mathcal{V}} G(L_1,\ldots,L_N;\mathcal{V}),$$

where the sum is taken over all (non-empty) partial partitions $\mathcal{V}$ of $\{1,\ldots,N\}$.

Moreover, if we denote for every such $\mathcal{V}$ by $G^+(L_1,\ldots,L_N;\mathcal{V})$ the number of all effective alignments A with $\mathcal{V} \subseteq \mathcal{V}(A)$, we surely have

$$\sum_{\mathcal{V}\subseteq\mathcal{W}} G(L_1,\ldots,L_N;\mathcal{W}) = G^+(L_1,\ldots,L_N;\mathcal{V}) = G(L_1-\chi_{\underline{\mathcal{V}}}(1),\ldots,L_N-\chi_{\underline{\mathcal{V}}}(N))$$

where $\chi_{\underline{\mathcal{V}}}$ denotes the characteristic function of $\underline{\mathcal{V}} := \bigcup_{V\in\mathcal{V}} V$, because that last number just counts the number of effective alignments of the N suffix sequences resulting from our original sequences by eliminating the first entry in each of the sequences s_i with $i \in \underline{\mathcal{V}}$ which is exactly the number of those alignments A of the original sequences with $\mathcal{V} \subseteq \mathcal{V}(A)$.

Consequently, Möbius inversion yields the following recursion formula

$$\begin{aligned}
G(L_1,\ldots,L_N;\mathcal{V}) &= \sum_{\mathcal{V}\subseteq\mathcal{W}'} G(L_1,\ldots,L_N;\mathcal{W}') \sum_{\mathcal{V}\subseteq\mathcal{W}\subseteq\mathcal{W}'} (-1)^{\#(\mathcal{W}-\mathcal{V})} = \\
&= \sum_{\mathcal{V}\subseteq\mathcal{W}} (-1)^{\#(\mathcal{W}-\mathcal{V})} \sum_{\mathcal{W}\subseteq\mathcal{W}'} G(L_1,\ldots,L_N;\mathcal{W}') = \\
&= \sum_{\mathcal{V}\subseteq\mathcal{W}} (-1)^{\#(\mathcal{W}-\mathcal{V})} G^+(L_1,\ldots,L_N;\mathcal{W}) = \\
&= \sum_{\mathcal{V}\subseteq\mathcal{W}} (-1)^{\#(\mathcal{W}-\mathcal{V})} G(L_1-\chi_{\underline{\mathcal{W}}}(1),\ldots,L_N-\chi_{\underline{\mathcal{W}}}(N))
\end{aligned}$$

which obviously implies the recursion formula

$$G(L_1,\ldots,L_N) = \sum_{\emptyset\neq\mathcal{V}}\left(\sum_{\mathcal{V}\subseteq\mathcal{W}}(-1)^{\#(\mathcal{W}-\mathcal{V})}G(L_1-\chi_{\underline{\mathcal{W}}}(1),\ldots,L_N-\chi_{\underline{\mathcal{W}}}(N))\right) =$$
$$= \sum_{\emptyset\neq\mathcal{W}}\left(\sum_{\emptyset\neq\mathcal{V}\subseteq\mathcal{W}}(-1)^{\#(\mathcal{W}-\mathcal{V})}\right)G(L_1-\chi_{\underline{\mathcal{W}}}(1),\ldots,L_N-\chi_{\underline{\mathcal{W}}}(N)) =$$
$$= \sum_{\emptyset\neq\mathcal{W}}(-1)^{1+\#\mathcal{W}}G(L_1-\chi_{\underline{\mathcal{W}}}(1),\ldots,L_N-\chi_{\underline{\mathcal{W}}}(N))$$

in view of

$$\sum_{\emptyset\neq\mathcal{V}\subseteq\mathcal{W}}(-1)^{\#(\mathcal{W}-\mathcal{V})}+(-1)^{\#\mathcal{W}} = \sum_{\mathcal{V}\subseteq\mathcal{W}}(-1)^{\#(\mathcal{W}-\mathcal{V})} = 0.$$

Moreover, we can rewrite these formulae by introducing the numbers

$$a(k) := \sum_{\sim}(-1)^{\#(\{1,\ldots,k\}/\sim)}$$

where, for any given $k\in\mathbb{N}_0$, we sum over all equivalence "$\sim$" relations defined on $\{1,\ldots,k\}$. Clearly, we have $a(0)=1, a(1)=-1, a(2)=0, a(3)=1, a(4)=1, a(5)=-2, a(6)=-9, a(7)=-9, a(8)=50$ and so on, as can be read off from the obvious recursion formula

$$a(k+1) \quad = \quad -\sum_{p=0}^{k}\binom{k}{p}a(k-p).$$

Remark: The series $a(k)$ also describes the expansion of $\exp(1-e^x)$ and is closely related to the Stirling numbers of second kind σ_k^j [7, 8]: With σ_k^j being the number of equivalence classes with exactly j classes on a set of k distinct elements, we have

$$a(k) = \sum_{j=1}^{k}(-1)^j\sigma_k^j.$$

Using these numbers while sorting the above formulae for multiply occuring equal terms, we get

$$G(L_1,\ldots,L_N;\mathcal{V}) = \sum_{\mathcal{V}\subseteq\mathcal{W}}(-1)^{\#(\mathcal{W}-\mathcal{V})}G(L_1-\chi_{\underline{\mathcal{W}}}(1),\ldots,L_N-\chi_{\underline{\mathcal{W}}}(N)) =$$
$$= \sum_{\underline{\mathcal{V}}\subseteq W}\left(\sum_{\mathcal{V}\subseteq\mathcal{W},\underline{\mathcal{W}}=W}(-1)^{\#(\mathcal{W}-\mathcal{V})}G(L_1-\chi_W(1),\ldots,L_N-\chi_W(N))\right) =$$
$$= \sum_{\underline{\mathcal{V}}\subseteq W\subseteq\{1,\ldots,N\}}\left(\sum_{\sim}(-1)^{\#((W-\underline{\mathcal{V}})/\sim)}\right)G(L_1-\chi_W(1),\ldots,L_N-\chi_W(N)) =$$
$$= \sum_{\underline{\mathcal{V}}\subseteq W\subseteq\{1,\ldots,N\}}a(\#(W-\underline{\mathcal{V}}))G(L_1-\chi_W(1),\ldots,L_N-\chi_W(N))$$

as well as

$$G(L_1,\ldots,L_N) = \sum_{\emptyset\neq W\subseteq\{1,\ldots,N\}} -a(\#W)G(L_1-\chi_W(1),\ldots,L_N-\chi_W(N)).$$

In case $N := 2$, this implies

$$\begin{aligned} G(L_1,L_2;\{\{1\}\}) &= G(L_1-1,L_2)-G(L_1-1,L_2-1),\\ G(L_1,L_2;\{\{2\}\}) &= G(L_1,L_2-1)-G(L_1-1,L_2-1),\\ G(L_1,L_2;\{\{1\},\{2\}\}) &= G(L_1,L_2;\{\{1,2\}\}) = G(L_1-1,L_2-1) \end{aligned}$$

as well as

$$G(L_1,L_2) = G(L_1-1,L_2)+G(L_1,L_2-1)$$

corroborating the result

$$G(L_1,L_2) = \binom{L_1+L_2}{L_1}$$

in view of

$$\binom{L_1+L_2}{L_1} = \binom{L_1+L_2-1}{L_1-1}+\binom{L_1+L_2-1}{L_1} = \binom{L_1+L_2-1}{L_1-1}+\binom{L_1+L_2-1}{L_2-1}.$$

In case $N := 3$, we get

$$\begin{aligned} G(L_1,L_2,L_3;\{\{1\}\}) &= G(L_1-1,L_2,L_3)-G(L_1-1,L_2-1,L_3)-G(L_1-1,L_2,L_3-1)\\ G(L_1,L_2,L_3;\{\{1,2\}\}) &= G(L_1,L_2,L_3;\{\{1\},\{2\}\})\\ &= G(L_1-1,L_2-1,L_3)-G(L_1-1,L_2-1,L_3-1),\\ G(L_1,L_2,L_3;\{\{1,2,3\}\}) &= G(L_1,L_2,L_3;\{\{1,2\},\{3\}\})\\ &= G(L_1,L_2,L_3;\{\{1\},\{2\},\{3\}\}) = G(L_1-1,L_2-1,L_3-1) \end{aligned}$$

as well as

$$\begin{aligned} G(L_1,L_2,L_3) &= G(L_1-1,L_2,L_3)+G(L_1,L_2-1,L_3)+\\ &\quad +G(L_1,L_2,L_3-1)-G(L_1-1,L_2-1,L_3-1). \end{aligned}$$

Acknowledgment

We are grateful to Mike Steel for some useful comments regarding this topic, and we also want to acknowledge that using the World-Wide Web page of [7], `http://www.research.att.com/~njas/sequences/index.html`, proved to be very helpful.

References

[1] M.S. Waterman, *Introduction to Computational Biology. Maps, Sequences and Genomes*, Chapman & Hall, London (1995).

[2] B. Morgenstern, A.W.M. Dress, and T. Werner, Multiple DNA and protein sequence alignment based on segment-to-segment comparison, *Proc. Natl. Acad. Sci. USA* **93** (22) 12098-12103 (1996).

[3] J.B. Kruskal, An overview of sequence comparison, In *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, (Edited by D. Sankoff and J.B. Kruskal), pp. 1-44, Addison-Wesley, Reading (1983).

[4] B. Morgenstern, J. Stoye, and A. Dress, Some theoretical aspects of pairwise and multiple sequence alignment, In preparation.

[5] H.T. Laquer, Asymptotic limits for a two-dimensional recursion, *Stud. Appl. Math.* **64** 271-277 (1981).

[6] G.-C. Rota, On the foundations of combinatorial theory I. Theory of Möbius functions, *Z. Wahrscheinlichkeitstheorie* **2** 340-368 (1964).

[7] N.J.A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego (1995).

[8] V.R.R. Uppuluri and J.A. Carpenter, Numbers generated by the function $\exp(1 - e^x)$, *The Fibonacci Quarterly* **7** 437-448 (1969).

PARTIALLY ORDERED GENERALIZED PATTERNS

SERGEY KITAEV

Abstract. We introduce partially ordered generalized patterns (POGPs), which further generalize the generalized permutation patterns (GPs) introduced by Babson and Steingrímsson [BabStein]. A POGP p is a GP some of whose letters are incomparable. Thus, in an occurrence of p in a permutation π, two letters that are incomparable in p pose no restrictions on the corresponding letters in π. We describe many relations between POGPs and GPs and give general theorems about the number of permutations avoiding certain classes of POGPs. These theorems have several known results as corollaries but also give many new results. We also give the generating function for the entire distribution of the maximum number of non-overlapping occurrences of a pattern p with no dashes, provided we know the e.g.f. for the number of permutations that avoid p.

1. Introduction and Background

All permutations in this paper are written as words $\pi = a_1 a_2 \cdots a_n$, where the a_i consist of all the integers $1, 2, \ldots, n$.

We will be concerned with *patterns* in permutations. A pattern is a word on some alphabet of letters, where some of the letters may be separated by dashes. In our notation, the classical permutation patterns, first studied systematically by Simion and Schmidt [SchSim], are of the form $p = 1-3-2$, the dashes indicating that the letters in a permutation corresponding to an occurrence of p don't have to be adjacent. In the classical case, an occurrence of a pattern p in a permutation π is a subsequence in π (of the same length as the length of p) whose letters are in the same relative order as those in p. For example, the permutation 41352 has only one occurrence of the pattern $1-2-3$, namely the subword 135.

Note that a classical pattern should, in our notation, have dashes at the beginning and end. Since all patterns considered in this paper satisfy this, we suppress these dashes from the notation. Thus, a pattern with no dashes corresponds to a contiguous subword anywhere in a permutation.

In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns (GPs)* where two adjacent letters in a pattern may be required to be adjacent in the permutation. Such an adjacency requirement is indicated by the absence of a dash between the corresponding letters in the pattern. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, but the pattern 2-3-1 occurs also in the subwords 562 and 563. The motivation for introducing these patterns in [BabStein] was the study of Mahonian statistics.

A number of interesting results on GPs were obtained by Claesson in [Claes]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were shown there. In [Kit] the present author investigated simultaneous avoidance of two or more 3-letter GPs with no dashes. This work is of particular interest here since avoidance of the patterns considered in this paper has a close connection to simultaneous avoidance of two or more GPs with no dashes. Also important here is the work of Elizalde and Noy [ElizNoy] where they find the distribution of several patterns with no dashes.

In this paper we introduce a further generalization of GPs — namely *partially ordered generalized patterns (POGP)*. A POGP is a GP some of whose letters are incomparable. For instance, if we write $p = 1 - 1'2'$ then we mean that in an occurrence of p in a permutation π the letter corresponding to the 1 in p can be either larger or smaller than the letters corresponding to $1'2'$. Thus, the permutation 13425 has four occurrences of p, namely 134, 125, 325 and 425.

We consider two particular classes of POGPs — *shuffle patterns* and *multi-patterns*. A multi-pattern is of the form $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ and a shuffle pattern is of the form $p = \sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - a_k - \sigma_k$, where for any i and j, the letter a_i is greater than any letter of σ_j and for

any $i \neq j$ each letter of σ_i is incomparable with any letter of σ_j These patterns are investigated in Sections 4 and 5. A corollary to one of our theorems (Theorem 13) about the shuffle patterns is the result of Claesson [Claes, Proposition 2] that the number of n-permutations that avoid the pattern $12-3$ is the n-th Bell number.

Let $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ be an arbitrary multi-pattern and let $A_i(x)$ be the exponential generating function (e.g.f.) for the number of permutations that avoid σ_i for each i. In Theorem 28 we find the e.g.f., in terms of the $A_i(x)$, for the number of permutations that avoid p. In particular, this allows us to find the e.g.f. for the entire *distribution* of the maximum number of non-overlapping occurrences of a pattern p with no dashes, if we only know the e.g.f. for the number of permutations that *avoid* p. In many cases, this gives nice generating functions.

We also give alternative proofs, using inclusion-exclusion, of some of the results of Elizalde and Noy [ElizNoy]. Our proofs result in explicit formulas for the e.g.f. in terms of infinite series whereas Elizalde and Noy obtained differential equations for the same e.g.f..

2. Definitions and Preliminaries

A *partially ordered generalized pattern (POGP)* is a GP where some of the letters can be incomparable.

Example 1. The simplest non-trivial example of a POGP that differs from the ordinary GPs is $p = 1' - 2 - 1''$, where the second letter is the greatest one and the first and the last letters are incomparable to each other. The permutation 3142 has two occurrences of p, namely, the subwords 342 and 142.

It is easy to see that the number of permutations that avoid p in Example 1 is equal to 2^{n-1}. Indeed, if $\pi = a_1 \dots a_n$ and a_i is the leftmost letter in π that is smaller than its successor, then all letters to the right of a_i must be in increasing order. So any permutation π avoiding p can be written as $\pi_1 1 \pi_2$, where π_1 is decreasing and π_2 is increasing and there are 2^{n-1} ways to pick the permutation π_1, which determines π.

Definition 2. If the number of permutations in S_n, for each n, that avoid a POGP p is equal to the number of permutations that avoid a POGP q, then p and q are said to be *equivalent* and we write $p \equiv q$ in this case.

If A_n is the number of n-permutations that avoid a pattern p, then the *exponential generating function*, or *e.g.f.*, of the class of such permutations is

$$A(x) = \sum_{n \geq 0} A_n \frac{x^n}{n!}.$$

We will talk about *bivariate generating functions*, or *b.g.f.*, exclusively as generating functions of the form

$$A(u,x) = \sum_{\pi} u^{p(\pi)} \frac{x^{|\pi|}}{|\pi|!} = \sum_{n,k \geq 0} A_{n,k} u^k \frac{x^n}{n!},$$

were $A_{n,k}$ is the number of n-permutations with k occurrences of the pattern p.

The *reverse* $R(\pi)$ of a permutation $\pi = a_1 a_2 \dots a_n$ is the permutation $a_n a_{n-1} \dots a_1$. The *complement* $C(\pi)$ is the permutation $b_1 b_2 \dots b_n$ where $b_i = n + 1 - a_i$. Also, $R \circ C$ is the composition of R and C. For example, $R(13254) = 45231$, $C(13254) = 53412$ and $R \circ C(13254) = 21435$. We call these bijections of $\mathcal{S}_n$ to itself *trivial*, and it is easy to see that any pattern p is equivalent to the patterns $R(p)$, $C(p)$ and $R \circ C(p)$. For example, the number of permutations that avoid the pattern 132 is the same as the number of permutations that avoid the patterns 231, 312 and 213, respectively.

It is convenient to introduce the following definition.

Definition 3. Let p be a GP without internal dashes. A permutation π *quasi-avoids* p if π has exactly one occurrence of p and this occurrence consists of the $|p|$ rightmost letters of π.

For example, the permutation 51342 quasi-avoids the pattern $p = 231$, whereas the permutations 54312 and 45231 do not. Indeed, 54312 ends with 312, which is not an occurrence of the pattern p, and 45231 has an occurrence of p, namely 452, in a forbidden place.

Proposition 4. *Let p be a non-empty GP with no dashes. Let $A(x)$ (resp. $A^\star(x)$) be the e.g.f. for the number of permutations that avoid (resp. quasi-avoid) p. Then*

$$A^\star(x) = (x-1)A(x) + 1.$$

Proof. We first show that

$$A^\star_n = nA_{n-1} - A_n. \tag{1}$$

If we consider all $(n-1)$-permutations that avoid p and all possible extending of these permutations to the n-permutations by writing one more letter to the right, then the number of obtained permutations will be nA_{n-1}. Obviously, the set of these permutations is a disjoint union of the set of all n-permutations that avoid p and the set of all n-permutations that quasi-avoid p. Thus we get (1). Multiplying both sides of (1) with $x^n/n!$ and summing over all natural numbers n, observing that $A^\star_0 = 0$, we get the desired result. □

Definition 5. Suppose $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes and $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ where each letter of σ_i is incomparable with any letter of σ_j whenever $i \neq j$. We call such POGPs *multi-patterns*.

Definition 6. Suppose $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes and $a_1 a_2 \ldots a_k$ is a permutation of k letters. We define a *shuffle* pattern to be a pattern of the form

$$\sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - \sigma_{k-1} - a_k - \sigma_k,$$

where for any i and j, the letter a_i is greater than any letter of σ_j and for any $i \neq j$ each letter of σ_i is incomparable with any letter of σ_j. We also allow σ_0 and σ_k, but not the other σ_i, to be empty patterns.

The pattern from Example 1 is an example of a shuffle pattern. It follows from the definitions that we can get a multi-pattern from a shuffle pattern by removing all the a_i.

Let $\mathcal{S}_\infty$ denote the disjoint union of the $\mathcal{S}_n$ for all $n \in \mathbb{N}$. The POGPs (which include the GPs, as well as the classical patterns), can be considered as functions from $\mathcal{S}_\infty$ to $\mathbb{N}$ that count the number of occurrences of the pattern in a permutation in $\mathcal{S}_\infty$. This allows us to write a POGP (as a function) as a linear combination of GPs. For example,

$$1' - 2 - 1'' \;=\; (1-3-2) \;+\; (2-3-1),$$

from which, in particular, we see that to avoid $1' - 2 - 1''$ is the same as to avoid simultaneously the patterns $1-3-2$ and $2-3-1$. A straightforward argument leads to the following proposition.

Proposition 7. *For any POGP p there exists a set S of GPs such that a permutation π avoids p if and only if π avoids all the patterns in S.*

The following theorem can be easily proved by induction on k:

Theorem 8. *Let $p_1 = \sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - \sigma_{k-1} - a_k - \sigma_k$ (resp. $p_2 = \sigma_0 - \sigma_1 - \cdots - \sigma_k$) be an arbitrary shuffle pattern (resp. multi-pattern) with $|\sigma_i| = \ell_i$ for all $i = 0, \ldots, k$. Then to avoid the pattern p_1 (resp. p_2) is the same as to avoid*

$$\prod_{i=1}^{k} \binom{\ell_0 + \ell_1 + \cdots + \ell_i}{\ell_i} = \binom{\ell_0 + \ell_1}{\ell_1}\binom{\ell_0 + \ell_1 + \ell_2}{\ell_2} \cdots \binom{\ell_0 + \ell_1 + \cdots + \ell_k}{\ell_k}$$

ordinary GPs.

Example 9. Let $p = 1'2' - 3 - 1''$. That is $\sigma = 12$ and $\tau = 1$. By Theorem 8, to avoid p is the same as to avoid $\binom{3}{2} = 3$ GPs simultaneously, namely $12-4-3$, $13-4-2$ and $23-4-1$.

There is a number of results on the distribution of several classes of patterns with no dashes. These results can be used as building blocks for some of the results in the present paper. The most important of these is the following result by Elizalde and Noy [ElizNoy]:

Theorem 10. [ElizNoy, Theorem 3.4] *Let m and a be positive integers with $a \leq m$, let $\sigma = 12\dots a\tau(a+1) \in \mathcal{S}_{m+2}$, where τ is any permutation of $\{a+2, a+3, \dots, m+2\}$, and let $P(u,z)$ be the b.g.f. for permutations where u marks the number of occurrences of σ. Then $P(u,z) = 1/w(u,z)$, where w is the solution of*

$$w^{a+1} + (1-u)\frac{z^{m-a+1}}{(m-a+1)!}w' = 0$$

with $w(0) = 1$, $w'(0) = -1$ and $w^{(k)}(0) = 0$ for $2 \leq k \leq a$. In particular, the distribution does not depend on τ.

3. GPs with no dashes

In order to apply our results in what follows we need to know how many patterns avoid a given ordinary GP with no dashes. We are also interested in different approaches to studying these patterns. The theorems in this section can be proved using an inclusion-exclusion argument similar to the one given in the proof of Theorem 30 and we omit these proofs. This allows us to get explicit formulas for the e.g.f. in terms of infinite series instead of having to solve differential equations as done by Elizalde and Noy [ElizNoy] for the same e.g.f.. However, in particular cases, we use certain differential equations to simplify our series.

Theorem 11. [GoulJack] *Let $A_k(x)$ be the e.g.f. for the number of permutations avoiding the pattern $p = 123\dots k$. Then*

$$A_k(x) = 1/F_k(x),$$

where $F_k(x) = \sum_{i\geq 0} \frac{x^{ki}}{(ki)!} - \sum_{i\geq 0} \frac{x^{ki+1}}{(ki+1)!}$.

For some k it is possible to simplify the function $F_k(x)$ in the theorems above. Indeed, $F_k(x)$ satisfies the differential equation $F_k^{(k)}(x) = F_k(x)$ with the k initial conditions $F_k(0) = 1$, $F_k'(0) = -1$ and $F_k^{(i)}(0) = 0$ for all $i = 2, 3, \dots, k-1$. For instance, if $k = 4$ then

$$F_4(x) = \frac{1}{2}(\cos x - \sin x + e^{-x}).$$

Theorem 12. *Let k and a be positive integers with $a < k$, let $p = 12\dots a\tau(a+1) \in \mathcal{S}_{k+1}$, where τ is any permutation of the elements $\{a+2, a+3, \dots, k+1\}$, and let $A_{k,a}(x)$ be the e.g.f. for the number of permutations that avoid p. Let*

$$F_{k,a}(x) = \sum_{i\geq 1} \frac{(-1)^{i+1}x^{ki+1}}{(ki+1)!} \prod_{j=2}^{i} \binom{jk-a}{k-a}.$$

Then

$$A_{k,a}(x) = 1/(1 - x + F_{k,a}(x)).$$

If $k = 2$ and $a = 1$ in the previous theorem, corresponding to the pattern $p = 132$, then from Theorem 12 the function $F_{2,1}(x)$, which is the same for the patterns p, 231, 312 and 213 because of the trivial bijections, can be written as:

$$F_{2,1}(x) = \sum_{i\geq 1} \frac{(-1)^{i+1}x^{ki+1}}{i!(k!)^i(ki+1)} = x - \int_0^x e^{-t^2/2}\, dt.$$

That is

$$A_{2,1} = \frac{1}{1 - \int_0^x e^{-t^2/2}\, dt},$$

which is a special case of Theorem 4.1 in [ElizNoy].

4. The Shuffle Patterns

We recall that according to Definition 6, a shuffle pattern is a pattern of the form $\sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - \sigma_{k-1} - a_k - \sigma_k$, where $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes, $a_1 a_2 \ldots a_k$ is a permutation of k letters, for any i and j the letter a_i is greater than any letter of σ_j and for any $i \neq j$ each letter of σ_i is incomparable with any letter of σ_j.

Let us consider a shuffle pattern that in fact is an ordinary generalized pattern. This pattern is $p = \sigma - k$, where σ is an arbitrary pattern with no dashes that is built on elements $1, 2, \ldots, k-1$. So the last element of p is greater than any other element.

Theorem 13. *Let $p = \sigma - k$ and let $A(x)$ (resp. $B(x)$) be the e.g.f. for the number of permutations that avoid σ (resp. p). Then $B(x) = e^{F(x, A(y))}$, where*

$$F(x, A(y)) = \int_0^x A(y)\, dy.$$

Proof. Suppose that $\pi \in \mathcal{S}_{n+1}$ and that π avoids p. Suppose the letter $(n+1)$ is in the i-th position and $\pi = \pi_1(n+1)\pi_2$, where π_1 and π_2 might be empty.

Since π is p-avoiding, π_1 must be σ-avoiding, because otherwise an occurrence of σ in π_1 together with the letter $(n+1)$ gives an occurrence of p in π. But if π_1 is σ-avoiding then there is no interaction between π_1 and π_2, that is, if π_2 is p-avoiding and π_1 is σ-avoiding then π is p-avoiding. To see this it is enough to see that if an occurrence of σ in π contains the letter $(n+1)$, then this occurrence of σ can not lead to an occurrence of $p = \sigma - k$ containing the letter $(n+1)$.

From the above, considering all possible positions of $(n+1)$, we get the recurrence relation

$$B_{n+1} = \sum_i \binom{n}{i} A_i B_{n-i},$$

where B_j (resp. A_j) is the number of j-permutations that avoid p (resp. σ), because we can choose the elements of π_1 in $\binom{n}{i}$ ways.

Multiplying both sides of the equality by $x^n/n!$ we get

$$\frac{B_{n+1}}{n!} x^n = \sum_i \frac{A_i}{i!} x^i \frac{B_{n-i}}{(n-i)!} x^{n-i}.$$

Taking the sum over all natural numbers n leads us to

$$B'(x) = A(x)B(x)$$

where the derivative of B is with respect to x. Since $B(0) = 1$, the solution of the differential equation is $B(x) = e^{F(x, A(y))}$. □

Example 14. Let $p = 1 - 2$. Here $\sigma = 1$, so $A(x) = 1$ since $A_n = 0$ for all $n \geq 1$ and $A_0 = 1$. So

$$B(x) = e^{F(x,1)} = e^x.$$

This corresponds to the fact that for each $n \geq 1$ there is exactly one permutation that avoids the pattern p, namely $\pi = n(n-1)\ldots 1$.

Example 15. Suppose $p = 12-3$. Here $\sigma = 12$, so $A(x) = e^x$, since there is exactly one permutation that avoids the pattern σ. So

$$B(x) = \sum_{n \geq 0} \frac{B_n}{n!} x^n = e^{F(x, e^y)} = e^{e^x - 1}.$$

According to [Claes, Proposition 2], for all $n \geq 1$, B_n is the n-th Bell number and the e.g.f. for the Bell numbers is $e^{e^x - 1}$.

The table below gives the initial values of B_n for several patterns $p = \sigma - k$. These numbers were obtained by expanding the corresponding $B(x)$. The functions $A(x)$ are taken from the previous section.

pattern	initial values for B_n
132-4	1, 2, 6, 23, 107, 585, 3671, 25986, 204738
123-4	1, 2, 6, 23, 108, 598, 3815, 27532, 221708
1234-5	1, 2, 6, 24, 119, 705, 4853, 38142, 336291
12345-6	1, 2, 6, 24, 120, 719, 5022, 40064, 359400

Theorem 16. *Let p be the shuffle pattern $\sigma - k - \tau$. So k is the greatest letter of the pattern, and each letter of σ is incomparable with any letter of τ. Let $A(x)$, $B(x)$ and $C(x)$ be the e.g.f. for the number of permutations that avoid σ, τ and p respectively. Then $C(x)$ is the solution of the differential equation*

$$C'(x) = (A(x) + B(x))C(x) - A(x)B(x),$$

with $C(0) = 1$.

Proof. As before, we consider the symmetric group $\mathcal{S}_{n+1}$ and a permutation $\pi \in \mathcal{S}_{n+1}$ that avoids p. Suppose the letter $(n+1)$ is in the i-th position and $\pi = \pi_1(n+1)\pi_2$, where π_1 and π_2 might be empty.

There are exactly four mutually exclusive possibilities:

1) π_1 does not avoid σ, π_2 does not avoid τ.
2) π_1 avoids σ, π_2 does not avoid τ;
3) π_1 does not avoid σ, π_2 avoids τ;
4) π_1 avoids σ, π_2 avoids τ;

Obviously, the situation 1) is impossible, since an occurrence of σ in π_1 with $(n+1)$ and with an occurrence of τ in π_2 gives us an occurrence of p in π. On the other hand, if p occurs in π then it is easy to see that the letter $(n+1)$ cannot be one of the letters in the occurrences of σ or τ, so all p-avoiding permutations are described by the possibilities 2)–4). We count these permutations in the following way.

In $\binom{n}{i}$ ways we choose first i elements from the letters $1, 2 \dots n$, that is, the elements of π_1. Let A_i, B_i and C_i be the number of i-permutations that avoid σ, τ and p respectively.

If π_1 is σ-avoiding, we let π_2 be any p-avoiding permutation of the remaining $(n-i+1)$ letters. This accounts for all "good" permutations from the possibilities 2) and 4). There are $\binom{n}{i}A_iC_{n-i}$ such permutations.

If π_2 is τ-avoiding, we let π_1 be any p-avoiding permutation of chosen i letters. This covers all "good" permutations from 3) and 4). There are $\binom{n}{i}B_iC_{n-i}$ such permutations.

But we have counted p-avoiding permutations that correspond to 4) twice, so we must subtract $\binom{n}{i}A_iB_{n-i}$, which is the number of such permutations.

So we have

$$C_{n+1} = \sum_i \binom{n}{i}(A_iC_{n-i} + B_iC_{n-i} - A_iB_{n-i}).$$

Multiplying both sides of the equality by $x^n/n!$ we get

$$\frac{C_{n+1}}{n!}x^n = \sum_i \left(\frac{A_i + B_i}{i!}x^i \frac{C_{n-i}}{(n-i)!}x^{n-i} - \frac{A_i}{i!}x^i \frac{B_{n-i}}{(n-i)!}x^{n-i} \right),$$

so

$$C'(x) = (A(x) + B(x))C(x) - A(x)B(x).$$

□

Example 17. Let $p = 1' - 2 - 1''$. That is, $\sigma = 1$ and $\tau = 1$. So $A(x) = B(x) = 1$ and we need to solve the equation

$$C'(x) = 2C(x) - 1$$

with $C(0) = 1$. The solution of this equation is $C(x) = \frac{1}{2}(e^{2x} + 1)$, so for all $n \geq 1$ we have $C_n = 2^{n-1}$, as in Example 1.

In the table below we record the initial values of C_n for several patterns $p = \sigma - k - \tau$.

σ	τ	**initial values for C_n**
1	12	1, 2, 6, 21, 82, 354, 1671, 8536, 46814
1	132	1, 2, 6, 24, 116, 652, 4178, 30070, 240164
1	123	1, 2, 6, 24, 116, 657, 4260, 31144, 253400
1	1234	1, 2, 6, 24, 120, 715, 4946, 38963, 344350
12	12	1, 2, 6, 24, 114, 608, 3554, 22480, 152546
12	132	1, 2, 6, 24, 120, 710, 4800, 36298, 302780
12	123	1, 2, 6, 24, 120, 710, 4815, 36650, 308778
12	1234	1, 2, 6, 24, 120, 720, 5025, 39926, 355538
123	123	1, 2, 6, 24, 120, 720, 5020, 39790, 352470
123	132	1, 2, 6, 24, 120, 720, 5020, 39755, 351518
132	132	1, 2, 6, 24, 120, 720, 5020, 39720, 350496

Remark 18. The pattern $p = \sigma - k$ from Theorem 13 is a particular case of the pattern $p = \sigma - k - \tau$ from Theorem 16 when τ is the empty word. The e.g.f. for the number of permutations that avoid the empty word is zero, because no permutation avoids the empty word. So if τ is empty, we can use Theorem 16 to get Theorem 13. Indeed, $B(x) = 0$, and after renaming C with B we get in Theorem 16 exactly the same differential equation as we have in Theorem 13.

We now give two corollaries to Theorem 16.

Corollary 19. *Suppose we have the shuffle pattern $p = \sigma - k - \tau$. We consider the pattern $\varphi(p) = \varphi_1(\sigma) - k - \varphi_2(\tau)$, where φ_1 and φ_2 are any trivial bijections. Then $p \equiv \varphi(p)$.*

Proof. We just observe that if $A(x)$ (resp. $B(x)$) is the e.g.f. for the number of permutations that avoid σ (resp. τ) then $A(x)$ (resp. $B(x)$) is the e.g.f. for the number of permutations that avoid $\varphi_1(\sigma)$ (resp. $\varphi_2(\tau)$). □

Corollary 20. *We have $\sigma - k - \tau \equiv \tau - k - \sigma$.*

Proof. This follows directly from the differential equation of Theorem 16 ($A(x)$ and $B(x)$ are symmetric in that equation), but we can also obtain this as a corollary to Corollary 19. By Corollary 19, the pattern $\sigma - k - \tau$ is equivalent to the pattern $\sigma - k - R(\tau)$. Reversing the pattern $\sigma - k - R(\tau)$, we obtain the pattern

$$R(\sigma - k - R(\tau)) = R(R(\tau)) - k - R(\sigma) = \tau - k - R(\sigma),$$

which thus is equivalent to $\sigma - k - \tau$. Finally, we use Corollary 19 one more time to get

$$\tau - k - R(\sigma) \equiv \tau - k - R(R(\sigma)) = \tau - k - \sigma.$$

□

5. The Multi-Patterns

We recall that according to Definition 5, a multi-pattern is a pattern $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$, where $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes and each letter of σ_i is incomparable with any letter of σ_j whenever $i \neq j$.

We first discuss patterns of the type $p = \sigma - \tau$ which are a particular case of the multi-patterns to be treated in this section.

If σ or τ is the empty word then we are dealing with ordinary GPs with no dashes, some of which were investigated in [ElizNoy] and Section 3. The analysis of the case when σ or τ is equal to 1 can also be reduced to the analysis of ordinary GPs. For example, suppose that $\sigma = 1$, that is, $p = 1 - \tau$, and we want to count the number of permutations in $\mathcal{S}_n$ that avoid p. We can choose the leftmost letter of a permutation avoiding p in n ways, then the remainder of the permutation must avoid τ, so we multiply n by the number of permutations in $\mathcal{S}_{n-1}$ that avoid τ. For instance, if $p = 1 - 1'2'$ then the number of permutations in $\mathcal{S}_n$ avoiding p is exactly n.

Theorem 21. *Let $p = \sigma - \tau$ and $q = \varphi_1(\sigma) - \varphi_2(\tau)$, where φ_1 and φ_2 are any of the trivial bijections. Then p and q are equivalent.*

Proof. The theorem is equivalent to the following statement:

Let $p = \sigma - \tau$ and $q = \sigma - \varphi(\tau)$, where φ is a trivial bijection. Then p and q are equivalent.

It is obvious that the statement follows from Theorem 21. Conversely, suppose we have $p = \sigma - \tau$. We observe that any two trivial bijections commute, that is for any trivial bijection ψ, we have $\psi(R(x)) = R(\psi(x))$. This observation, the statement and the fact that $x \equiv R(x)$ give

$$p = \sigma - \tau \equiv \sigma - \varphi_2(\tau) \equiv R(\varphi_2(\tau)) - R(\sigma) \equiv R(\varphi_2(\tau)) - \varphi_1(R(\sigma)) \equiv$$
$$R(\varphi_2(\tau)) - R(\varphi_1(\sigma)) \equiv \varphi_1(\sigma) - \varphi_2(\tau).$$

So to prove the theorem we now prove the statement.

Let $p = \sigma - \tau$ and $q = \sigma - \varphi(\tau)$, where φ is a trivial bijection. Let A_n (resp. B_n) be the number of n-permutations that avoid p (resp. q). We are going to prove that $A_n = B_n$.

Suppose π avoids p and $\pi = \pi_1\sigma'\pi_2$, where $\pi_1\sigma'$ has exactly one occurrence of the pattern σ, namely σ'. Then π_2 must avoid τ, $\varphi(\pi_2)$ must avoid $\varphi(\tau)$ and $\pi_\varphi = \pi_1\sigma'\varphi(\pi_2)$ avoids q. The converse is also true, that is, if π_φ has no occurrences of q then π has no occurrences of p. If π has no occurrences of σ then π has no occurrences of p as well as no occurrences of q. Since any permutation either avoids σ or can be factored as above, we have a bijection between the class of permutations that avoid p and the class of permutations that avoid q. Thus $A_n = B_n$. □

We get the following corollary to Theorem 21:

Corollary 22. *The pattern $\sigma - \tau$ is equivalent to the pattern $\tau - \sigma$.*

Proof. We proceed as in the proof of Corollary 20. From Theorem 21 we have:

$$\sigma - \tau \equiv \sigma - R(\tau) \equiv R(R(\tau)) - R(\sigma) \equiv \tau - R(R(\sigma)) \equiv \tau - \sigma.$$

□

We observe that the presence of the dash in the patterns in Theorem 21 is essential. That is, generally speaking, the pattern $\sigma\tau$ is not equivalent to the pattern $\varphi_1(\sigma)\varphi_2(\tau)$ for any trivial bijections φ_1 and φ_2. For example, there are 66 permutations in $\mathcal{S}_5$ that avoid the pattern $122'1'$ but only 61 that avoid $121'2'$. In Section 6 we investigate the pattern $122'1'$.

Theorem 23 and Corollary 24 generalise Theorem 21 and Corollary 22:

Theorem 23. *Suppose we have multi-patterns $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ and $q = \tau_1 - \tau_2 - \cdots - \tau_k$, where $\tau_1\tau_2\dots\tau_k$ is a permutation of $\sigma_1\sigma_2\dots\sigma_k$. Then p and q are equivalent.*

Proof. We proceed by induction on k. If $k = 2$ then the statement is true by Corollary 22. Suppose the statement is true for all $k' < k$. Suppose p has exactly k blocks. If a permutation π avoiding p has no occurrences of σ_1 then it obviously avoids both p and q. Otherwise we factor π as $\pi = \pi_1\sigma_1'\pi_2$ where $\pi_1{\sigma_1}'$ has exactly one occurrence of the pattern σ_1, namely ${\sigma_1}'$. Then π_2 must avoid $\sigma_2 - \cdots - \sigma_k$. Moreover it is irrelevant from which letters $\pi_1{\sigma_1}'$ is built and therefore we can apply the inductive hypothesis. We can rearrange $\sigma_2'\dots\sigma_k'$ of $\sigma_2\dots\sigma_k$ in such a way that the blocks in $\tau_1\tau_2\dots\tau_k$ corresponding to $\sigma_2,\dots,\sigma_k$ are arranged in the same order as the τ's. Now we consider separately two cases: $\tau_k \neq \sigma_1$ and $\tau_k = \sigma_1$. In the first case we use the following equivalences:

$$p = \sigma_1 - \sigma_2 - \cdots - \sigma_k \equiv \sigma_1 - {\sigma_2}' - \cdots - \sigma_k' \equiv R(\sigma_k') - \cdots - R(\sigma_2') - R(\sigma_1).$$

For the pattern $R(\sigma_k') - \cdots - R(\sigma_2') - R(\sigma_1)$ we use the factorisation of a permutation π avoiding this pattern, where the role of σ_1 is played by $R(\sigma_k')$. So by the inductive hypothesis we put the pattern $R(\sigma_1)$ in the right place somewhere to the left of $R(\sigma_2')$ and apply R to get that $p \equiv q$.

In the second case we have:

$$p \equiv R(\sigma_k') - \cdots - R(\sigma_2') - R(\sigma_1) \equiv R(\sigma_k') - \cdots - R(\sigma_1) - R(\sigma_2') \equiv$$
$$\sigma_2' - \sigma_1 - \cdots - \sigma_k' \equiv \sigma_2' - \cdots - \sigma_k' - \sigma_1 = q$$

The first equivalence here is taken from the considerations above; the second one uses the inductive hypothesis; then we use the fact that $R(R(x)) = x$ and apply the inductive hypothesis again. □

Corollary 24. *Suppose we have multi-patterns $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ and $q = \varphi_1(\sigma_1) - \varphi_2(\sigma_2) - \cdots - \varphi_k(\sigma_k)$, where each φ_i is an arbitrary trivial bijection. Then p and q are equivalent.*

Proof. We use induction on k, Theorem 23 and the factorisation of permutations, which is discussed in the proof of Theorem 23. If $k = 2$ then the statement is true by Theorem 21. Suppose the statement is true for all $k' < k$. Then

$$p = \sigma_1 - \sigma_2 - \cdots - \sigma_k \equiv \sigma_1 - \varphi_2(\sigma_2) - \cdots - \varphi_k(\sigma_k) \equiv$$
$$\varphi_2(\sigma_2) - \sigma_1 - \cdots - \varphi_k(\sigma_k) \equiv \varphi_2(\sigma_2) - \varphi_1(\sigma_1) - \cdots - \varphi_k(\sigma_k) \equiv$$
$$\varphi_1(\sigma_1) - \varphi_2(\sigma_2) - \cdots - \varphi_k(\sigma_k) = q,$$

where first we apply the inductive hypothesis then Theorem 23 then the inductive hypothesis and finally Theorem 23 again. □

Theorem 25. *Suppose $p = \sigma - p'$, where p' is an arbitrary POGP, and the letters of σ are incomparable to the letters of p'. Let $C(x)$ (resp. $A(x)$, $B(x)$) be the e.g.f. for the number of permutations that avoid p (resp. σ, p'). Moreover let $A^\star(x)$ be the e.g.f. for the number of permutations that quasi-avoid σ. Then*

$$C(x) = A(x) + B(x)A^\star(x).$$

Proof. Let A_n, B_n, C_n be the number of n-permutations that avoid the patterns σ, p' and p respectively. Also $A_n^\star$ is the number of n-permutations that quasi-avoid σ. If a permutation π avoids σ then it avoids p. Otherwise we find the leftmost occurrence of σ in π. We assume that this occurrence consists of the $|\sigma|$ rightmost letters among the i leftmost letters of π. So the subword of π beginning at the $(i+1)$st letter must avoid p'. From this we conclude

$$C_n = A_n + \sum_{i=|\sigma|}^{n} \binom{n}{i} A_i^\star B_{n-i}.$$

We observe that we can change the lower bound in the sum above to 0, because $A_i^\star = 0$ for $i = 0, 1, \ldots, |\sigma| - 1$. Multiplying both sides by $x^n/n!$ and taking the sum over all n we get the desired result. □

Corollary 26. *Suppose $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ is a multi-pattern where $|\sigma_i| = 2$ for all i, so each σ_i is equal to either 12 or 21. If $B(x)$ is the e.g.f. for the number of permutations that avoid p then*

$$B(x) = \frac{1 - (1 + (x-1)e^x)^k}{1 - x}.$$

Proof. We use Theorem 25, induction on k and the fact that $A(x) = e^x$ and $A^\star(x) = 1 + (x-1)e^x$. □

The following corollary to Corollary 26 can be proved combinatorially.

Theorem 27. *There are $(n-2)2^{n-1} + 2$ permutations in $\mathcal{S}_n$ that avoid the pattern $p = 12 - 1'2'$ or, according to Theorem 21, the pattern $p = 12 - 2'1'$.*

One more corollary to Theorem 25 is the following theorem that is the basis for calculating the number of permutations that avoid a multi-pattern, and therefore is the main result for multi-patterns in this paper.

Theorem 28. *Let $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ be a multi-pattern and let $A_i(x)$ be the e.g.f. for the number of permutations that avoid σ_i. Then the e.g.f. $B(x)$ for the number of permutations that avoid p is*

$$B(x) = \sum_{i=1}^{k} A_i(x) \prod_{j=1}^{i-1} ((x-1)A_j(x) + 1).$$

Proof. We use Theorem 25 and prove by induction on k that

$$B(x) = \sum_{i=1}^{k} A_i(x) \prod_{j=1}^{i-1} A_j^\star(x).$$

Then we use Proposition 4 to get the desired result. □

Remark 29. One can consider the function $B(x)$ from Theorem 28 as a function in k variables $B(x) = B(A_1(x), A_2(x), \ldots, A_k(x))$. Then, by Theorem 23, this function is symmetric in the variables $A_1(x), A_2(x), \ldots, A_k(x)$. That means that we can rename the variables, which may simplify the calculation of $B(x)$.

6. Patterns of the Form $\sigma\tau$

Theorem 30. *Let $B(x)$ be the e.g.f. for the number of permutations that avoid the pattern $p = 122'1'$. Then*

$$B(x) = \frac{1}{2} + \frac{1}{4}\tan x(1 + e^{2x} + 2e^x \sin x) + \frac{1}{2}e^x \cos x.$$

Proof. Let B_n be the number of n-permutations that avoid p and A_n be the number of n-permutations that avoid p and begin with the pattern 12. Let also $A(x)$ be the e.g.f. for the numbers A_n. We set $B_0 = A_0 = A_1 = 1$. Suppose π is a $(n+1)$-permutation that avoids p. There are three mutually exclusive possibilities:

1) $\pi = (n+1)\pi_2$;
2) $\pi = \pi_1(n+1)$;
3) $\pi = \pi_1(n+1)\pi_2$ and $\pi_1, \pi_2 \neq \varepsilon$.

Obviously, in 1) and 2) the letter $(n+1)$ does not affect the rest of the permutation π, and therefore in each of these cases we have B_n permutations that avoid p. In 3), it is easy to see that if π_1 has more than one letter then π_1 must end with a 21 pattern whereas if π_2 has more than one letter then π_2 must begin with a 12 pattern. The key observation is that the number of n-permutations that avoid p and end with a 21 pattern is the same as the number of n-permutations that avoid p and begin with a 12 pattern. To see this it is enough to apply the reverse function to any n-permutation π that begins with 12-pattern and avoids p and observe that $R(p) = p$, that is, $R(\pi)$ avoids p and ends with a 21 pattern. Obviously this is a bijection. So if $|\pi_1| = i$ then we can choose the letters of π_1 in $\binom{n}{i}$ ways and then choose a permutation π_1 in A_i ways and a permutation π_2 in A_{n-i} ways, since the letters of π_1 and π_2 do not affect each other. From all this we get

$$B_{n+1} = 2B_n + \sum_{i=1}^{n-1} \binom{n}{i} A_i A_{n-i} = 2B_n + \sum_{i=0}^{n} \binom{n}{i} A_i A_{n-i} - 2A_n.$$

We multiply both sides of the last equality by $x^n/n!$ to get

$$B_{n+1}\frac{x^n}{n!} = 2B_n\frac{x^n}{n!} + \sum_{i=0}^{n} \frac{A_i}{i!}x^i \frac{A_{n-i}}{(n-i)!}x^{n-i} - 2A_n\frac{x^n}{n!}.$$

Summing both sides over all natural numbers n we get:

$$B'(x) = 2B(x) + A^2(x) - 2A(x). \tag{2}$$

To solve this differential equation with the initial condition $B(0) = 1$, we need to determine $A(x)$. One can observe that if a permutation π avoids p and begins with the pattern 12 then π has the structure $\pi = a_1 b_1 a_2 b_2 a_3 b_3 \cdots$, where $a_i < b_i$ for all i. Moreover, if $b_1 < a_2$ then we must have $a_1 < b_1 < a_2 < b_2 < a_3 < \cdots$ since otherwise we obviously have an occurrence of the pattern p. A first approximation is that $A_n = \binom{n}{2}A_{n-2}$, because we can choose $a_1 b_1$ in π in $\binom{n}{2}$ ways and then pick an arbitrary $(n-2)$-permutation that avoids p and begins with the pattern 12, to be $a_2 b_2 a_3 b_3 \ldots$, in A_{n-2} ways. But it is possible that $b_1 < a_2$ in which case $b_1 a_2 b_2 a_3$ can be an occurrence of p in π, and it is an occurrence of p unless $a_2 < b_2 < a_3 < \cdots$. So in order to avoid this we must subtract the number of permutations of the form $abcd\pi'$, where $a < b < c < d$ and π' is any $(n-4)$-permutation that avoids p, from the first approximation of A_n. Thus the second approximation is that $A_n = \binom{n}{2}A_{n-2} - \binom{n}{4}A_{n-4}$. We observe that in the second approximation we do not count the increasing permutation $123\ldots n$. Moreover, among the permutations counted by $\binom{n}{4}A_{n-4}$, there are the permutations that begin with 6 increasing letters. Except for the increasing permutation, such permutations are not counted by $\binom{n}{2}A_{n-2}$. We must therefore add the number

of such permutations. So the third approximation is that $A_n = \binom{n}{2}A_{n-2} - \binom{n}{4}A_{n-4} + \binom{n}{6}A_{n-6}$ and so on. That is,

$$A_n = \binom{n}{2}A_{n-2} - \binom{n}{4}A_{n-4} + \binom{n}{6}A_{n-6} - \binom{n}{8}A_{n-8} + \cdots = \sum_{i\geq 1}(-1)^{i+1}\binom{n}{2i}A_{n-2i}. \tag{3}$$

We observe that if $n = 4k$ or $n = 4k+1$ then we do not count the increasing permutation in our sum. This, together with Equation 3, gives us

$$\sum_{i\geq 0}(-1)^i\binom{n}{2i}A_{n-2i} = \begin{cases} 1, & \text{if } n = 4k \text{ or } n = 4k+1, \\ 0, & \text{if } n = 4k+2 \text{ or } n = 4k+3. \end{cases}$$

Multiplying both sides of the equality with $x^n/n!$ and summing over all natural numbers n we get

$$(A_0 + A_1x + \frac{A_2}{2!}x^2 + \cdots)(1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots) = \sum_{k=0}^{\infty}\left(\frac{x^{4k}}{(4k)!} + \frac{x^{4k+1}}{(4k+1)!}\right).$$

The left hand side of this equality is equal to $A(x)\cos x$. Let $F(x)$ be the function in the right hand side of the equality. Then it is easy to see that $F(x)$ is the solution to the differential equation $F^{(4)}(x) = F(x)$ with the initial conditions $F(0) = F'(0) = 1$, $F^{(2)}(0) = F^{(3)}(0) = 0$. So $F(x) = \frac{1}{2}(\cos x + \sin x + e^x)$ and

$$A(x) = \frac{1}{2}\left(1 + \tan x + \frac{e^x}{\cos x}\right).$$

Now we solve the differential equation (2) and get

$$B(x) = \frac{1}{2} + \frac{1}{4}\tan x(1 + e^{2x} + 2e^x\sin x) + \frac{1}{2}e^x\cos x.$$

□

Remark 31. The series expansion of $B(x)$ in Theorem 30 begins with

$$B(x) = 1 + x + x^2 + x^3 + \frac{3}{4}x^4 + \frac{11}{20}x^5 + \frac{7}{20}x^6 + \frac{7}{30}x^7 + \frac{103}{720}x^8 + \cdots.$$

That is, the initial values for B_n are 1, 2, 6, 18, 66, 252, 1176, 5768.

7. The Distribution of Non-Overlapping GPs

A descent in a permutation $\pi = a_1a_2\ldots a_n$ is an i such that $a_i > a_{i+1}$. The number of descents in a permutation π is denoted des π (and is equivalent to the generalized pattern 21). Any statistic with the same distribution as des is said to be *Eulerian*. The *Eulerian numbers* $A(n,k)$ count permutations in the symmetric group $\mathcal{S}_n$ with k descents and they are the coefficients of the *Eulerian polynomials* $A_n(t)$ defined by $A_n(t) = \sum_{\pi\in\mathcal{S}_n} t^{1+\text{des }\pi}$. The Eulerian polynomials satisfy the identity

$$\sum_{k\geq 0}k^nt^k = \frac{A_n(t)}{(1-t)^{n+1}}.$$

Two descents i and j *overlap* if $j = i+1$. We define a new statistic, namely the *maximum number of non-overlapping descents*, or MND, in a permutation. For instance, MND(321) = 1 whereas MND(41532) = 2. One can find the distribution of this new statistic by using Corollary 26. This distribution is given in Example 33. However, we prove a more general theorem:

Theorem 32. *Let p be a GP with no dashes. Let $A(x)$ be the e.g.f. for the number of permutations that avoid p. Let $D(x,y) = \sum_{\pi} y^{N(\pi)}\frac{x^{|\pi|}}{|\pi|!}$ where $N(\pi)$ is the maximum number of non-overlapping occurrences of p in π. Then*

$$D(x,y) = \frac{A(x)}{1 - y((x-1)A(x)+1)}.$$

Proof. We fix the natural number k and consider an auxiliary multi-pattern $P_k = p - p - \cdots - p$ with k copies of p. If a permutation avoids P_k then it has at most $k-1$ non-overlapping occurrences of p. From Theorem 28, the e.g.f. $B_k(x)$ for the number of permutations avoiding P_k is equal to $\sum_{i=1}^{k} A(x) \prod_{j=1}^{i-1}((x-1)A(x)+1)$. If we subtract $B_k(x)$ from the e.g.f. $B_{k+1}(x) = \sum_{i=1}^{k+1} A(x) \prod_{j=1}^{i-1}((x-1)A(x)+1)$ for the number of permutations avoiding P_{k+1}, which is obtained by applying Theorem 28 to the pattern P_{k+1}, then we get the e.g.f. $D_k(x)$ for the number of permutations that have exactly k non-overlapping occurrences of the pattern p. So

$$D_k(x) = \sum_n D_{n,k} \frac{x^n}{n!} = B_{k+1}(x) - B_k(x) = A(x)((x-1)A(x)+1)^k.$$

Now

$$D(x,y) = \sum_{n,k\geq 0} D_{n,k} y^k \frac{x^n}{n!} = \sum_k D_k(x) y^k = \frac{A(x)}{1 - y((x-1)A(x)+1)}.$$

□

All of the following examples are corollaries to Theorem 32.

Example 33. If we consider descents then $A(x) = e^x$, hence the distribution of MND is given by the formula:

$$D(x,y) = \frac{e^x}{1 - y(1 + (x-1)e^x)}.$$

Example 34. Theorems 11 and 32 give the distribution of the maximum number of non-overlapping occurrences of the increasing subword of length k (the pattern $123\ldots k$), which is equal to

$$D(x,y) = \frac{1}{(1-x)y + (1-y)F_k(x)},$$

were $F_k(x) = \sum_{i\geq 0} \frac{x^{ki}}{(ki)!} - \sum_{i\geq 0} \frac{x^{ki+1}}{(ki+1)!}$.

Example 35. If we consider the maximum number of non-overlapping occurrences of the pattern 132 then the distribution of these numbers is given by the formula

$$D(x,y) = \frac{1}{1 - yx + (y-1)\int_0^x e^{-t^2/2}\, dt}.$$

Example 36. The distribution of the maximum number of non-overlapping occurrences of the pattern from Theorem 12 is given by the formula:

$$D(x,y) = \frac{1}{1 - x + (1-y)F_{k,a}(x)},$$

where $F_{k,a}(x) = \sum_{i\geq 1} \frac{(-1)^{i+1} x^{ki+1}}{(ki+1)!} \prod_{j=2}^{i} \binom{jk-a}{k-a}$.

References

[BabStein] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, Séminaire Lotharingien de Combinatoire, B44b:18pp, 2000.

[Bon1] M. Bóna: Exact enumeration of 1342-avoiding permutations; A close link with labeled trees and planar maps, Journal of Combinatorial Theory, Series A, **80** (1997) 257-272.

[Bon2] M. Bóna: Permutations avoiding certain patterns; The case of length 4 and generalisations, Discrete Mathematics **175** (1997), 55-67.

[Bon3] M. Bóna: Permutations with one or two 132-subsequences, Discrete Mathematics **181** (1998), 267-274.

[Claes] A. Claesson: Generalised Pattern Avoidance, European Journal of Combinatorics **22** (2001), 961-971.

[ClaesMans] A. Claesson and T. Mansour: Permutations avoiding a pair of generalized patterns of length three with exactly one dash, preprint CO/0107044.

[ElizNoy] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, Proceedings of FPSAC 2001.

[GoulJack] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, A Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons Inc., New York, (1983).
[Kit] S. Kitaev: Multi-avoidance of generalised patterns, preprint. `http://www.math.chalmers.se/~kitaev/papers.html`
[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).
[Mans] T. Mansour: Restricted 1-3-2 permutations and generalized patterns, preprint CO/0110039.
[SloPlo] N. J. A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, (1995). http://www.research.att.com/∼njas/sequences/.
[Stan] R. Stanley: *Enumerative Combinatorics*, Volume **1**, Cambridge University Press, (1997).
[SchSim] R. Simion, F. Schmidt: Restricted permutations, European J. Combin. **6** (1985), no. 4, 383–406.

E-mail address: `kitaev@math.chalmers.se`

Matematik, Chalmers tekniska högskola och Göteborgs universitet, S-412 96 Göteborg, Sweden

Chern Classes of Tautological Sheaves on Hilbert Schemes of Points on Surfaces

Manfred Lehn

Introduction

Hilbert schemes $X^{[n]}$ of n-tuples of points on a complex projective manifold X are natural compactifications of the configuration spaces of unordered distinct n-tuples of points on X. Their geometry is determined by the geometry of X itself and the geometry of the 'punctual' Hilbert schemes of all zero-dimensional subschemes in affine space that are supported at the origin. Thus one is naturally led to the following problem:

Determine explicitly the geometric or topological invariants of the Hilbert schemes $X^{[n]}$ such as the Betti numbers, the Hodge numbers, the Chern numbers, the cohomology ring, from the corresponding data of the manifold X itself.

This problem is most attractive when X is a surface, since then the Hilbert schemes are themselves irreducible projective manifolds, by a result of Fogarty [9], whereas for higher dimensional varieties the Hilbert schemes are in general neither irreducible nor smooth nor pure of expected dimension.

In the surface case, the answer to the problem above for the Betti numbers was first given by Göttsche in [11]. The answer turns out to be particularly beautiful (cf. Theorem 2.1 below). The problem for the Hodge numbers was solved by Sörgel and Göttsche [12]. For a different approach to both results see [3]. The answer for the Chern classes will be implicitly given in a forthcoming paper by Ellingsrud, Göttsche and the author [4].

The question for the ring structure of the cohomology is more difficult. In general, $X^{[2]}$ is the quotient of the blow-up of $X \times X$ along the diagonal by the canonical involution that exchanges the factors. Thus the case of interest is $H^*(X^{[n]})$, $n \geq 3$. The ring structure was found for $(\mathbb{P}^2)^{[3]}$ by Ellingsrud and Strømme [5], and for $X^{[3]}$, X smooth projective of arbitrary dimension, by Fantechi and Göttsche [8]. In another direction, Ellingsrud and Strømme [6] gave generators for $H^*((\mathbb{P}^2)^{[n]}, \mathbb{Z})$, n arbitrary, and an implicit description of the relations.

Vafa and Witten [27] remarked that Göttsche's Formula for the Betti numbers is identical with the Poincaré series of a Fock space modelled on the cohomology of X. Nakajima [21] succeeded in giving a geometric construction of such a Fock space structure on the cohomology of the Hilbert schemes, leading to a natural 'explanation' of Göttsche's result. Similar results have been announced by Grojnowski [13].

Following the presentation of Grojnowski, this can be made more precise as follows: sending a pair (ξ', ξ'') of subschemes of length n' and n'', respectively, and of disjoint support to their union $\xi' \cup \xi''$ defines a rational map

$$m : X^{[n']} \times X^{[n'']} - - \to X^{[n'+n'']}.$$

This map induces linear maps on the rational cohomology

$$m_{n',n''} : H^*(X^{[n']}; \mathbb{Q}) \otimes H^*(X^{[n'']}; \mathbb{Q}) \longrightarrow H^*(X^{[n'+n'']}; \mathbb{Q})$$

and

$$m^{n',n''} : H^*(X^{[n'+n'']}; \mathbb{Q}) \longrightarrow H^*(X^{[n']}; \mathbb{Q}) \otimes H^*(X^{[n'']}; \mathbb{Q}).$$

If we let $\mathbb{H} := \oplus_n H^*(X^{[n]}; \mathbb{Q})$, then these maps define a multiplication and a comultiplication

$$m_* : \mathbb{H} \otimes \mathbb{H} \longrightarrow \mathbb{H}, \qquad m^* : \mathbb{H} \longrightarrow \mathbb{H} \otimes \mathbb{H},$$

which make $\mathbb{H}$ a commutative and cocommutative bigraded Hopf algebra. The result of Nakajima and Grojnowski says that this Hopf algebra is isomorphic to the graded symmetric algebra of the vector space $H^*(X; \mathbb{Q}) \otimes t\mathbb{Q}[t]$.

More explicitly, Nakajima constructed linear maps[1]

$$\mathfrak{q}_n : H^*(X; \mathbb{Q}) \longrightarrow \mathrm{End}_{\mathbb{Q}}(\mathbb{H}), \quad n \in \mathbb{Z},$$

and proved that they satisfy the 'oscillator' or 'Heisenberg' relations

$$[\mathfrak{q}_n(\alpha), \mathfrak{q}_m(\beta)] = (-1)^n \cdot n \cdot \delta_{n+m} \cdot \int_X \alpha\beta \cdot \mathrm{id}_{\mathbb{H}}.$$

Here the commutator is to be taken in a graded sense.

The multiplication and the comultiplication of $\mathbb{H}$ are not obviously related to the quite different ring structure of $\mathbb{H}$, which is given by the usual cup product on each direct summand $H^*(X^{[n]}; \mathbb{Q})$. (Strictly speaking, $\mathbb{H}$ contains a countable number of idempotents $1_{X^{[n]}} \in H^0(X^{[n]}; \mathbb{Q})$ but not a unit unless we pass to some completion).

This paper attempts to relate the Hopf algebra structure and the cup product structure. More precisely:

Let F be locally free sheaf of rank r on X. Attaching to a point $\xi \in X^{[n]}$, i.e. a zero-dimensional subscheme $\xi \subset X$, the $\mathbb{C}$-vector space $F \otimes \mathcal{O}_\xi$ defines a locally free sheaf $F^{[n]}$ of rank rn on $X^{[n]}$. The Chern classes of all sheaves on $X^{[n]}$ of this type generate a subalgebra $\mathcal{A} \subset \mathbb{H}$. We will describe a purely algebraic algorithm to determine the action of $\mathcal{A}$ on $\mathbb{H}$ in terms of the $\mathbb{Q}$-basis of $\mathbb{H}$ provided by Nakajima's results. We collect the Chern classes of all sheaves $F^{[n]}$ for a given sheaf F into operators

$$\mathfrak{ch}(F) : \mathbb{H} \to \mathbb{H}, \qquad \mathfrak{c}(F) : \mathbb{H} \to \mathbb{H}$$

and geometrically compute the commutators of these operators with the 'standard operators' defined by Nakajima.

[1] Our presentation differs in notations and conventions slightly from Nakajima's.

A central rôle is played by the operator $\mathfrak{d} := \mathfrak{q}_1(\mathcal{O}_X)$, which — up to a factor $(-1/2)$ — can also be interpreted as the intersection with the 'boundaries' of the Hilbert schemes, i.e. the divisors $\partial X^{[n]} \subset X^{[n]}$ of all tuples ξ which have a multiple point somewhere. The derivative of any operator $\mathfrak{f} \in \mathrm{End}(\mathbb{H})$ is defined by $\mathfrak{f}' := [\mathfrak{d}, \mathfrak{f}]$. Our main technical result then says that

$$\mathfrak{q}_n'(\alpha) = \frac{n}{2} \sum_\nu \mathfrak{q}_\nu \mathfrak{q}_{n-\nu} \delta(\alpha) + n \frac{|n|-1}{2} \mathfrak{q}_n(K\alpha), \tag{1}$$

where $\delta : H^*(X;\mathbb{Q}) \to H^*(X;\mathbb{Q}) \otimes H^*(X;\mathbb{Q})$ is the map induced by the diagonal embedding and K is the canonical class of X. An immediate algebraic consequence of this relation is

$$[\mathfrak{q}_n'(\alpha), \mathfrak{q}_m(\beta)] = -nm \cdot \mathfrak{q}_{n+m}(\alpha\beta) \tag{2}$$

for $n, m > 0$. By induction one concludes that the operators $\mathfrak{q}_1$ and $\mathfrak{d}$ suffice to generate all $\mathfrak{q}_n$, $n \geq 1$.

The commutator of the Chern character operator $\mathfrak{ch}(F)$ with the standard operator $\mathfrak{q}_1$ can be expressed in terms of higher derivatives of $\mathfrak{q}_1$:

$$[\mathfrak{ch}(F), \mathfrak{q}_1(\alpha)] = \sum_{n \geq 0} \frac{1}{n!} \mathfrak{q}_1^{(n)}(ch(F)\alpha). \tag{3}$$

Equations (1), (2) and (3) together give a complete description of the action of $\mathcal{A}$ on $\mathbb{H}$. Here are two applications:

1. We give a general algebraic solution to Donaldson's question for the integral N_n of the top Segre class of the bundles $L^{[n]}$ associated to a line bundle L for any n and explicitly compute N_n for $n \leq 7$.

2. We prove the following formula conjectured by Göttsche: If L is a line bundle on X then

$$\sum_{n \geq 0} c(L^{[n]}) z^n = \exp\left(\sum_{m \geq 1} \frac{(-1)^{m-1}}{m} \mathfrak{q}_m(c(L)) z^m \right).$$

This paper is organised as follows: In Section 1 we recall the basic geometric notions used in the later parts. Section 2 provides an introduction to Nakajima's results. Section 3 contains the core of this paper: we first define Virasoro operators $\mathfrak{L}_n$ in analogy to the standard construction and show how these arise geometrically. We then introduce the operator $\mathfrak{d}$ and compute the derivative of $\mathfrak{q}_n$. Finally, in Section 4 we apply these results to compute the action of the Chern classes of tautological bundles.

Discussions with A. King were important to me in clarifying and understanding the picture that Nakajima draws in his very inspiring article. I am very grateful to G. Ellingsrud for all the things I learned from his talks and conversations with him

about Hilbert schemes. To some extend the results in this article are a reflection on an induction method entirely due to him. Most of the research for this paper was carried out during my stay at the SFB 343 of the University of Bielefeld. On various occasions I was allowed to lecture on Hilbert schemes and their cohomology in the seminar of the algebraic geometry group in Bielefeld: it is a pleasure to thank S. Bauer, R. Brussee and T. Zink for their willingness to listen attentively and critically even to not yet fully correct preliminary results. I owe special thanks to S. Bauer for his continuous encouragement, interest and support.

Bielefeld, 8 October, 1997. Manfred Lehn

Contents

1 Preliminaries

In this section we introduce the basic notations that will be used throughout the paper and collect a number of results from the literature, mostly without proof.

1.1 Symmetric products

Let Y be a quasi-projective scheme over $\mathbb{C}$. The symmetric group $\mathfrak{S}_n$ acts on Y^n by permutation of the factors, and there exists a geometric quotient $\pi : Y^n \to S^nY$ for this action. S^nY is again quasi-projective, and if Y is irreducible (reduced, integral or normal) then the same is true for S^nY. Moreover, this construction is functorial: any morphism $f : Y' \to Y$ induces a morphism $S^nf : S^nY' \to S^nY$.

It follows from the theorem on elementary symmetric functions that $S^n\mathbb{A}^1 \cong \mathbb{A}^n$. Consequently, the symmetric products of smooth curves are again smooth. On the other hand, if Y is a smooth variety of dimension greater than one, then S^nY is singular for $n > 1$.

By a result of Grothendieck [15], the natural map

$$\pi^* : H^*(S^nY; \mathbb{Q}) \longrightarrow H^*(Y^n; \mathbb{Q}) \cong H^*(Y; \mathbb{Q})^{\otimes n}$$

is an isomorphism onto the subring of invariant elements under the action of $\mathfrak{S}_n$. From this Macdonald computed the following formula for the Betti numbers of S^nY by a purely algebraic argument:

Theorem 1.1 (Macdonald [20]) — *The Betti numbers of the symmetric products are given by the formula*

$$\sum_{n\geq 0}\sum_{i\geq 0} b_i(S^nY)t^iq^n = \prod_{i=0}^{2dim(Y)} (1-(-1)^it^iq)^{-(-1)^ib_i(Y)}.$$

□

There is another property of the symmetric product, which is important for the definition of the Hilbert-Chow morphism. Consider the following set-valued contravariant functor $\mathcal{M}_n(Y)$ on the category of locally Noetherian $\mathbb{C}$-schemes:

Let S be a $\mathbb{C}$-scheme, and let $p : S \times Y \to S$ be the projection. Then $\mathcal{M}_n(Y)(S)$ is the set of all isomorphism classes of coherent sheaves F on $S \times Y$, where F is S-flat, $p : \mathrm{Supp}(F) \to S$ is a finite map, and p_*F is locally free of rank n. If $f : S' \to S$ is a $\mathbb{C}$-morphism, then $\mathcal{M}_n(Y)(f) : \mathcal{M}(Y)(S) \to \mathcal{M}_n(Y)(S')$ maps the class of F to f_Y^*F. Here and in the following we write f_Y instead of $f \times \mathrm{id}_Y$.

Grothendieck [14] asserted that there is a natural transformation $\mathcal{M}_n(Y) \to S^nY$ sending any zero-dimensional sheaf F to its weighted support. This means that for any $[F] \in \mathcal{M}_n(Y)(S)$ there is a classifying morphism $\Phi_F : S \to S^nY$ such that $\Phi_F(s) = \sum_{y\in Y} \ell(F_{s,y}) \cdot y$ for all $s \in S$, where for any coherent sheaf $\mathcal{G}$ we let $\ell(\mathcal{G}_y)$

denote the length of the stalk $\mathcal{G}_y$ as a module over $\mathcal{O}_{Y,y}$. Moreover, $\Phi_{f_Y^* F} = \Phi_F \circ f$ for any $f : S' \to S$.

This was first proved by Iversen [18] using the technique of linear determinants. In fact, if Y is normal then $S^n Y$ *corepresents* the functor $\mathcal{M}_n(Y)$ (cf. [16, Ex. 4.3.6]).

1.2 Hilbert schemes and Hilbert-Chow morphism

Throughout this paper, the term 'Hilbert scheme' will always refer to Hilbert schemes of zero-dimensional subschemes.

Let Y be a quasi-projective scheme over $\mathbb{C}$. The Hilbert functor is the following set-valued functor on the category of locally Noetherian $\mathbb{C}$-schemes:

Let $\mathrm{Hilb}(Y,n)(S)$ be the set of all closed subschemes $Z \subset S \times Y$ such that the projection $p : Z \to S$ is flat and finite of degree n. If $f : S' \to S$ is a $\mathbb{C}$-morphism, the induced map is given by pull-back: $Z \mapsto f_Y^{-1}(Z) = S' \times_S Z$.

Grothendieck [14] showed that $\mathrm{Hilb}(Y,n)$ is represented by a quasi-projective scheme $Y^{[n]}$. If Y is projective, $Y^{[n]}$ is projective as well.

'Functoriality' in Y is limited to a few cases: if $f : Y' \to Y$ is an (open, closed) immersion, then there is a natural (open, closed) immersion

$$f^{[n]} : (Y')^{[n]} \to Y^{[n]},$$

defined by taking the image of subschemes under f. Moreover, suppose that $f : Y' \to Y$ is an étale (surjective) morphism. Let $U \subset (Y')^{[n]}$ denote the open subset of all subschemes $\xi \subset Y'$ such that the set-theoretic support of ξ is mapped injectively to Y. Then taking images under f defines an étale (surjective) morphism $U \to Y^{[n]}$.

For small values of n there are explicit descriptions of $Y^{[n]}$: Clearly, $Y^{[0]}$ is a reduced point, $Y^{[1]} \cong Y$, and $Y^{[2]}$ is the quotient for the $\mathfrak{S}_2$-action on the blow-up of $Y \times Y$ along the diagonal. Proceeding by induction, it is not difficult to see that all Hilbert schemes $Y^{[n]}$ are connected if Y is connected.

Observe that there is a natural transformation of functors

$$\mathrm{Hilb}(Y,n) \longrightarrow \mathcal{M}_n(Y)$$

which sends a subscheme $Z \subset S \times Y$ to its structure sheaf $\mathcal{O}_Z \in \mathrm{Coh}(S \times Y)$. As $\mathrm{Hilb}(Y,n)$ is respresented by $Y^{[n]}$, this transformation induces a morphism of schemes

$$\rho : Y^{[n]} \to S^n Y,$$

the *Hilbert-Chow* morphism. On a point $[\upsilon] \in Y^{[n]}$, i.e. a subscheme $\upsilon \subset Y$, this morphism is given by

$$\rho([\upsilon]) = \sum_{y \in Y} \ell(\mathcal{O}_{\upsilon,y}) \cdot y.$$

For example, if C is a smooth curve, then $\rho : C^{[n]} \to S^n C$ is an isomorphism.

1.3 Hilbert schemes of smooth surfaces

From now on, let X denote a smooth irreducible projective surface. The basic geometry of the Hilbert schemes of points on surfaces is governed by two theorems due to Fogarty [9] and Briançon [1].

Theorem 1.2 (Fogarty) — *$X^{[n]}$ is a $2n$-dimensional smooth irreducible projective variety.*

Here is a short sketch of the proof: projectivity is due to Grothendieck. He also showed that the Zariski tangent space of $X^{[n]}$ at a point ξ is canonically isomorphic to $\mathrm{Hom}(\mathcal{I}_\xi, \mathcal{O}_\xi)$. Since we already know that $X^{[n]}$ is connected, it therefore suffices to show that $\hom(\mathcal{I}_\xi, \mathcal{O}_\xi) = 2n$ for all $\xi \in X^{[n]}$. This can be done using Serre duality and the Hirzebruch-Riemann-Roch Theorem applied to the groups $\mathrm{Ext}^\bullet(\mathcal{O}_\xi, \mathcal{O}_\xi)$. □

Remark 1.3 — We already mentioned that $C^{[n]}$ is smooth for smooth curves. Computing the dimension of the tangent space one can show that $Y^{[3]}$ is smooth for a smooth variety Y of any dimension. On the other hand, $Y^{[n]}$ is singular if $\dim(Y) > 2$ and $n > 3$.

Fix a point $p \in X$ and let $X^{[n]}_p \subset X^{[n]}$ denote the closed subset of all subschemes $\xi \subset X$ with $\mathrm{Supp}(\xi) = \{p\}$ (with the reduced induced subscheme structure). This is indeed a closed subset, as it is the fibre $\rho^{-1}(np)$ of the Hilbert-Chow morphism over the point $np \in S^n X$.

Let $(\mathcal{O}, \mathfrak{m})$ denote the local ring of X at p. Since any point $\xi \in X^{[n]}_p$ may be considered as a subscheme of $\mathrm{Spec}(\mathcal{O}/\mathfrak{m}^n)$, and since $\mathcal{O}/\mathfrak{m}^n \cong \mathbb{C}[x,y]/(x,y)^n$, all schemes $X^{[n]}_p$ — for varying X and p — are (non-canonically) isomorphic. Clearly, $X^{[1]}_p = \{p\}$ and $X^{[2]}_p = \mathbb{P}(T_p X^\vee)$, moreover it is not too difficult to see that $X^{[3]}_p$ is isomorphic to the projective cone over the twisted cubic $C_3 \subset \mathbb{P}^3$, the vertex of the cone corresponding to the subscheme $\mathrm{Spec}(\mathcal{O}/\mathfrak{m}^2)$. It is not accidental that in these examples the dimension of $X^{[n]}_p$ increases by one in each step:

Theorem 1.4 (Briançon) — *For all $n \geq 1$, $X^{[n]}_p$ is an irreducible variety of dimension $n-1$.* □

For a proof see [1]. A new proof with a more geometric and conceptual argument was recently given by Ellingsrud and Strømme [7].

Briançon's Theorem emphasises the importance of curvilinear schemes: recall that a zero-dimensional subscheme $\xi \subset X$ is called *curvilinear* at $x \in X$, if ξ_x is contained in some smooth curve $C \subset X$. Equivalently, one might say that $\mathcal{O}_{\xi,x}$ is isomorphic to the $\mathbb{C}$-algebra $\mathbb{C}[z]/(z^\ell)$, where $\ell = \ell(\xi_x)$. Hence ξ is curvilinear at x if ξ_x is either empty, a reduced point, or if $\dim T_x \xi = 1$. From this criterion it is clear, that in any flat family of zero-dimensional subschemes the points in the base space which correspond to curvilinear subschemes form an open subset.

In particular, we may consider the open subset $X^{[n]}_{p,curv} \subset X^{[n]}_p$. This set has a very nice structure:

Lemma 1.5 — *If $n \geq 2$, then the morphism*

$$t : X^{[n]}_{p,curv} \longrightarrow \mathbb{P}(T_p X^\vee), [\xi] \mapsto [T_p \xi]$$

is a bundle morphism with affine fibres $\mathbb{A}^{n-2}$. In particular, $X^{[n]}_{p,curv}$ is an irreducible smooth variety of dimension $n-1$.

Proof. Let $x, y \in \mathcal{O}_{X,p}$ be local coordinates and consider the open subset $U = \{\langle y + \alpha_1 x\rangle | \alpha_1 \in \mathbb{C}\} \subset \mathbb{P}(T_p X^\vee)$. Then there is an isomorphism $\mathbb{A}^{n-1} \to t^{-1}(U)$ sending the $(n-1)$-tuple $(\alpha_1, \ldots, \alpha_{n-1})$ to the subsheaf corresponding to the ideal $(y + \alpha_1 x + \ldots + \alpha_{n-1} x^{n-1}) + \mathcal{I}^n_p$. □

As a consequence of this lemma we see that Briançon's Theorem is equivalent to saying that $X^{[n]}_{p,curv}$ is dense in $X^{[n]}_p$. This is a very important information: curvilinear subschemes are far easier to handle than any of the others. They contain only one subscheme for any given smaller length, any small deformation of a curvilinear subscheme is again locally curvilinear etc.

Generalising the definition of $X^{[n]}_p$ slightly, let $\Delta \subset S^n X$ denote the diagonal, and let $X^{[n]}_0 := \rho^{-1}(\Delta)$, endowed with the reduced induced subscheme structure. Thus $X^{[n]}_0$ consists of all subschemes $\xi \subset X$ of length n which are supported at *some* point in X. The fibres of the surjective morphism $\rho : X^{[n]} \to X$ are the schemes $X^{[n]}_p$ considered above. In fact, a choice of regular parameters near a point p leads to a trivialisation of the morphism $\rho : X^{[n]} \to X$ near p, i.e. ρ is a fibre bundle for the Zariski topology.

As an immediate consequence of Briançon's Theorem we get

Corollary 1.6 — *$X^{[n]}_0$ is an irreducible variety of dimension $n+1$.* □

Note that $X^{[n]}_p$ and $X^{[n]}_0$ have complementary dimensions as subvarieties in $X^{[n]}$. Their homological intersection is therefore zero-dimensional. However, the inclusion $X^{[n]}_p \subset X^{[n]}_0$ complicates the computation of the intersection product. The following result was obtained by Ellingsrud and Strømme [7] by an inductive geometric argument:

Theorem 1.7 (Ellingsrud, Strømme) — $\deg([X^{[n]}_p] \cdot [X^{[n]}_0) = (-1)^{n-1} \cdot n$. □

1.4 Incidence schemes

Since $X^{[n]}$ represents the functor $\mathrm{Hilb}^n(X)$, there is a *universal family* of subschemes

$$\Xi_n \subset X^{[n]} \times X.$$

Again, for small values of n there are explicit descriptions: Ξ_0 is empty, Ξ_1 is the diagonal in $X \times X$, and Ξ_2 is the blow-up $\mathrm{Bl}_\Delta(X \times X)$ of the diagonal in $X \times X$. The identification is given by the quotient map $\mathrm{Bl}_\Delta(X \times X) \to X^{[2]} = \mathrm{Bl}_\Delta(X \times X)/\mathfrak{S}_2$ and any of the two projections $\mathrm{Bl}_\Delta(X \times X) \to X$.

Assume that $n' > n > 0$. Then there is a uniquely determined closed subscheme $X^{[n',n]} \subset X^{[n']} \times X^{[n]}$ with the property that any morphism

$$f = (f_1, f_2) : T \to X^{[n']} \times X^{[n]}$$

factors through $X^{[n',n]}$ if and only if $f_{2,X}^{-1}(\Xi_n) \subset f_{1,X}^{-1}(\Xi_{n'})$. Closed points in $X^{[n',n]}$ correspond to pairs (ξ', ξ) of subschemes with $\xi \subset \xi'$. Let

$$X^{[n']} \xleftarrow{p_1} X^{[n',n]} \xrightarrow{p_2} X^{[n]}$$

denote the two projections. Then $X^{[n',n]}$ parametrises two flat families

$$p_{2,X}^{-1}(\Xi_n) \subset p_{1,X}^{-1}(\Xi_{n'}).$$

Consider the corresponding exact sequence

$$0 \to \mathcal{I}_{n',n} \to p_{1,X}^* \mathcal{O}_{\Xi_{n'}} \to p_{2,X}^* \mathcal{O}_{\Xi_n} \to 0. \tag{4}$$

The ideal sheaf $\mathcal{I}_{n',n}$ is a coherent sheaf on $X^{[n',n]} \times X$ which is flat over $X^{[n',n]}$ and fibrewise zero-dimensional of length $n'-n$. It therefore induces a classifying morphism to the symmetric product, analogously to the Hilbert-Chow morphism, which we will also denote by

$$\rho : X^{[n',n]} \to S^{n'-n}X.$$

As before let $X_0^{[n',n]} := \rho^{-1}(\Delta)$, where $\Delta \subset S^{n'-n}X$ is the small diagonal. A point in $X_0^{[n',n]}$ is a triple (ξ', x, ξ) with $\xi \subset \xi'$ and $\mathrm{Supp}(\mathcal{I}_{\xi/\xi'}) = \{x\}$.

We may decompose $X_0^{[n',n]}$ into locally closed subsets Z_ℓ, $\ell \geq 0$, with

$$Z_\ell := \{(\xi', x, \xi) | \ell(\xi_x) = \ell\}.$$

Lemma 1.8 — *Z_0 and Z_1 are irreducible of dimension $n+n'+1$ and $n+n'$, respectively, and $\dim(Z_\ell) < n+n'$ for all $\ell > 1$. Moreover, Z_1 is contained in the closure of Z_0.*

Proof. If $\ell = 0$ or 1, the map $(\xi', x, \xi) \mapsto (\xi - \xi_x, \xi'_x)$ is an open immersion

$$Z_\ell \longrightarrow X^{[n-\ell]} \times X_0^{[n'-n+\ell]}.$$

It follows from Briançon's Theorem that Z_ℓ is irreducible and

$$\dim(Z_\ell) = 2(n-\ell) + (n'-n+\ell+1) = n+n'+1-\ell.$$

For $\ell \geq 2$ consider the embedding

$$Z_\ell \longrightarrow X^{[n-\ell]} \times (X_0^{[\ell]} \times_X X_0^{[n'-n+\ell]}), \quad (\xi', x, \xi) \mapsto (\xi - \xi_x, \xi_x, \xi'_x).$$

In fact, the image of Z_ℓ is contained in a *proper* closed subset of the target variety: For *either* ξ'_x is curvilinear, in which case there is only a unique subscheme $\xi \subset \xi'_x$ of length ℓ, *or* ξ'_x is not curvilinear and therefore contained in a proper closed subset of $X_0^{[n'-n+\ell]}$. Now, the variety on the right hand side has dimension

$$2(n-\ell) + (\ell+1) + (n'-n+\ell+1) - 2 = n + n'.$$

Finally, a general point in Z_1 is of the form $(\zeta \cup \eta, x, \zeta \cup \{x\})$ where η is a curvilinear subscheme supported at x and disjoint from ζ. Now it is easy to deform η to a subscheme $\{x\} \cup \eta'$ with η' supported at a point $x' \neq x$. Hence a general point of Z_1 deforms into Z_0. □

Definition 1.9 — For any pair of nonnegative integers define subvarieties

$$E^{[n',n]}, Q^{[n',n]} \subset X^{[n']} \times X \times X^{[n]}$$

as follows: if $n' > n > 0$ let $Q^{[n',n]}$ and $E^{[n',n]}$ be the closure of Z_0 and Z_1, respectively. Moreover, $Q^{[n',0]} := X_0^{[n']}$, $E^{[n',0]} := \emptyset$ and $Q^{[n,n]} := \emptyset$, whereas $E^{[n,n]} := \{(\xi, x, \xi) | x \in \xi\} \cong \Xi_n$. On the other hand, if $n \geq n'$, let $Q^{[n',n]} = T(Q^{[n,n']})$ and $E^{[n',n]} = T(E^{[n,n']})$ under the twist

$$T : X^{[n]} \times X \times X^{[n']} \to X^{[n']} \times X \times X^{[n]}.$$

By construction $Q^{[n,n']}$ and $E^{[n,n']}$ are empty or irreducible varieties of dimension $n + n' + 1$ and $n + n'$, respectively.

Let us return to the particular case $n' - n = 1$, the most basic of all incidence situations: consider the projectivisation $\sigma : \mathbb{P}(\mathcal{I}_{\Xi_n}) \to X^{[n]} \times X$. It is an easy exercise to see that there is a natural isomorphism $\mathbb{P}(\mathcal{I}_{\Xi_n}) \cong X^{[n+1,n]}$ such that the diagram

$$\begin{array}{ccc} \mathbb{P}(\mathcal{I}_{\Xi_n}) & \xrightarrow{\cong} & X^{[n+1,n]} \\ & {\scriptstyle\sigma}\searrow \quad \swarrow{\scriptstyle(p_2,\rho)} & \\ & X^{[n]} \times X & \end{array}$$

commutes.

Theorem 1.10 (Ellingsrud, Strømme [7]) — *The incidence scheme $X^{[n+1,n]}$ is an irreducible variety.*

An immediate corollary is the following: there is a natural closed immersion $\mathrm{Bl}_{\Xi_n}(X^{[n]} \times X) \to \mathbb{P}(\mathcal{I}_{\Xi_n})$; since both are irreducible varieties, this must be an isomorphism. The exceptional divisor E is precisely the variety $E^{[n+1,n]}$ defined above. Hence in this situation we may write the sequence (4) as

$$0 \to (\mathrm{id}, \rho)_* \mathcal{O}_{X^{[n+1,n]}}(-E) \to p_{1,X}^* \mathcal{O}_{\Xi_{n+1}} \to p_{2,X}^* \mathcal{O}_{\Xi_n} \to 0. \tag{5}$$

In fact, the incidence scheme is smooth. This has independently be proved by Ellingsrud, Tikhomirov and Cheah. The proofs are unpublished.

2 The structure of the cohomology

As before, let X be a smooth irreducible projective surface. By Fogarty's Theorem the Hilbert schemes $X^{[n]}$ are projective manifolds of real dimension $4n$. The motivating problem in this study is to understand the cohomology rings $H^*(X^{[n]})$ in terms of the cohomology ring $H^*(X)$.

As far as the vector space structure of the cohomology is concerned, i.e. if we only ask for the dimensions of the graded pieces of the cohomology, this problem was solved by Göttsche [11]. The answer is given by the following beautiful formula for the Betti numbers.

Theorem 2.1 (Göttsche) — *The Betti numbers $b_i(X^{[n]})$ are determined by the Betti numbers $b_j(X)$. More precisely, the following formula holds:*

$$\sum_{n\geq 0}\sum_{i\geq 0} b_i(X^{[n]})t^i q^n = \prod_{m>0}\prod_{j\geq 0}(1-(-1)^j t^{2m-2+j}q^m)^{-(-1)^j b_j(X)}$$

Göttsches original proof uses the Weil Conjectures [11]. For a different approach see [3].

Among other things one learns from this formula that it is a good idea to consider all Hilbert schemes simultaneously. This will become even more striking through Nakajima's method which we will review in the next sections. As a preparation we collect a few definitions:

Definition 2.2 — Let $\mathbb{H} := \bigoplus_{n,i\geq 0}\mathbb{H}^{n,i}$ denote the double graded vector space with components $\mathbb{H}^{n,i} = H^i(X^{[n]};\mathbb{Q})$. Since $X^{[0]}$ is a point, $\mathbb{H}^{0,0} = \mathbb{Q}$. The unit in $H^0(X^{[0]};\mathbb{Q})$ is called the 'vacuum vector' and denoted by $\mathbf{1}$.

A linear map $\mathfrak{f}:\mathbb{H}\to\mathbb{H}$ is homogeneous of bidegree (ν,ι) if $\mathfrak{f}(\mathbb{H}^{n,i})\subset\mathbb{H}^{n+\nu,i+\iota}$ for all n and i. If $\mathfrak{f},\mathfrak{f}'\in\mathrm{End}(\mathbb{H})$ are homogeneous linear maps of bidegree (ν,ι) and (ν',ι'), respectively, their commutator is defined by

$$[\mathfrak{f},\mathfrak{f}'] = \mathfrak{f}\circ\mathfrak{f}' - (-1)^{\iota\cdot\iota'}\mathfrak{f}'\circ\mathfrak{f}.$$

We use the notation $|\alpha|$, $|\mathfrak{f}|$ etc. to denote the cohomological degree of homogeneous cohomology classes, homogeneous linear maps etc.

Setting

$$(\alpha,\beta) := \int_{X^{[n]}}\alpha\beta$$

for any $\alpha,\beta\in H^*(X^{[n]};\mathbb{Q})$ defines a non-degenerate (anti)symmetric bilinear form on $H^*(X^{[n]};\mathbb{Q})$ and hence on $\mathbb{H}$. For any homogeneous linear map $\mathfrak{f}:\mathbb{H}\to\mathbb{H}$ its adjoint $\mathfrak{f}^\dagger$ is characterised by the relation

$$(\mathfrak{f}(\alpha),\beta) = (-1)^{|\mathfrak{f}|\cdot|\alpha|}(\alpha,\mathfrak{f}^\dagger(\beta)).$$

Clearly, $(\mathfrak{f}\circ\mathfrak{g})^\dagger = \mathfrak{g}^\dagger\circ\mathfrak{f}^\dagger$.

2.1 Correspondences

Let Y_1 and Y_2 be smooth projective varieties, and let u be a class in the Chow group $A_n(Y_1 \times Y_2)$. (We tacitly assume rational coefficients. This will not always be necessary. On the other hand, we are not interested in integrality questions for the moment, and hence will not pay attention to this problem). The image of u in $H_{2n}(Y_1 \times Y_2)$ will be denoted by the same symbol. u induces a homogeneous linear map

$$u_* : H^i(Y_2) \to H^{i+2(\dim Y_1 - n)}(Y_1), \quad y \mapsto PD^{-1} p_{1*}(u \cap p_2^* y),$$

where $PD : H^*(Y_1) \to H_*(Y_1)$ is the Poincaré duality map.

Assume that Y_3 is another smooth projective variety, and $v \in A_m(Y_2 \times Y_3)$. Let p_{ij} be the projection from $Y_1 \times Y_2 \times Y_3$ to the factors $Y_i \times Y_j$, and consider the element

$$w := p_{13*}(p_{12}^* u \cdot p_{23}^* v) \in A_{n+m-\dim Y_2}(Y_1 \times Y_3).$$

Then

$$w_* = u_* \circ v_*.$$

See [10, Ch. 16] for details.

Suppose $U \subset Y_1 \times Y_2$ and $V \subset Y_2 \times Y_3$ are closed subschemes such that $u \in A_*(U)$ and $v \in A_*(V)$. Let

$$W := p_{13}(p_{12}^{-1}(U) \cap p_{23}^{-1}(V))$$

Then the class w defined above is already defined in $A_*(W)$.

The following type of arguments will often show up in the sequel: one shows that the dimension of W is smaller than the degree of w, which forces w to be zero; or that there is at most one irreducible component W_0 of W of maximal dimension with 'correct' dimension $\dim(W_0) = \deg(w)$. In this case one must have $w = \mu \cdot [W_0]$ and it suffices to determine the multiplicity μ.

Let $T : Y_1 \times Y_2 \to Y_2 \times Y_1$ exchange the factors. Then a Chow cycle u induces two maps

$$u_* : H^*(Y_2) \to H^*(Y_1) \quad \text{and} \quad (Tu)_* : H^*(Y_1) \to H^*(Y_2)$$

which are related by the formula

$$\int_{Y_1} u_*(\alpha) \cdot \beta = \int_{Y_2} \alpha \cdot (Tu)_*(\beta).$$

This follows directly from the projection formula. Thus $(Tu)_* = u_*^\dagger$.

The following operators were introduced by Nakajima [21]. The study of their properties is the major theme of this article. We take the liberty to change the notations and sign conventions.

Recall that we defined (1.9) subvarities

$$Q^{[n_1,n_2]} \subset X^{[n_1]} \times X \times X^{[n_2]}$$

of dimension $n_1 + n_2 + 1$. Their fundamental classes are cycles

$$[Q^{[n_1,n_2]}] \in A_{n_1+n_2+1}(X^{[n_1]} \times X \times X^{[n_2]}).$$

Let the projections to the factors be denoted by p_1, ρ and p_2.

Definition 2.3 (Nakajima) — Define linear maps

$$\mathfrak{q}_\ell : H^*(X;\mathbb{Q}) \longrightarrow \mathrm{End}(\mathbb{H}), \qquad \ell \in \mathbb{Z},$$

as follows: assume first that $\ell \geq 0$. For $\alpha \in H^*(X;\mathbb{Q})$ and $y \in H^*(X^{[n]};\mathbb{Q})$ let

$$\mathfrak{q}_\ell(\alpha)(y) := [Q^{[n+\ell,n]}]_*(\alpha \otimes y) = PD^{-1}p_{1*}([Q^{[n+\ell,n]}] \cap (\rho^*\alpha \cdot p_2^* y)).$$

The operators for negative indices then are determined by the relation

$$\mathfrak{q}_{-\ell}(\alpha) := (-1)^\ell \mathfrak{q}_\ell(\alpha)^\dagger.$$

By definition, $\mathfrak{q}_\ell(\alpha)$ is a homogeneous linear map of bidegree $(\ell, 2\ell - 2 + |\alpha|)$. Moreover, $\mathfrak{q}_0 = 0$, and if $\ell > 0$, the operator $\mathfrak{q}_\ell(\alpha)^\dagger$ is induced by the subvarieties $Q^{[n,n+\ell]}$, $n \geq 0$.

2.2 Nakajima's Main Theorem

In this section we review the main result of [21] and some of the immediate consequences. Similar results have been announced by Grojnowski [13].

Theorem 2.4 (Nakajima) — *For any integers n and m and cohomology classes α and β, the operators $\mathfrak{q}_n(\alpha)$ and $\mathfrak{q}_m(\beta)$ satisfy the following 'oscillator relations':*

$$[\mathfrak{q}_n(\alpha), \mathfrak{q}_m(\beta)] = n \cdot \delta_{n+m} \cdot \int_X \alpha\beta \cdot \mathrm{id}_{\mathbb{H}}.$$

□

Here and in the following we adopt the convention that δ_ν equals 1 if $\nu = 0$ and is zero else, and that any integral $\int_Z \alpha$ is zero if $\deg(\alpha) \neq \dim_{\mathbb{R}}(Z)$.

In [21] Nakajima only showed that the commutator relation hold with some universal nonzero constant instead of the coefficient n. The correct value was first computed directly by Ellingsrud and Strømme [7]: up to a sign factor, which depends on our convention, this number is the intersection number of Theorem 1.7. Briefly afterwards, Nakajima gave a different proof using 'vertex operators' [22].

Consider the vector spaces

$$W_+ := H^*(X;\mathbb{Q}) \otimes t\mathbb{Q}[t] \quad \text{and} \quad W_- := H^*(X;\mathbb{Q}) \otimes t^{-1}\mathbb{Q}[t^{-1}].$$

Define a non-degenerate skew-symmetric pairing on the vector space $W := W_- \oplus W_+$ by

$$\{\alpha \otimes t^n, \beta \otimes t^m\} := n \cdot \delta_{n+m} \cdot \int_X \alpha\beta.$$

Note that we are taking the expression 'skew-symmetric' in a graded sense:

$$\{\alpha \otimes t^n, \beta \otimes t^m\} = -(-1)^{|\alpha|\cdot|\beta|}\{\beta \otimes t^m, \alpha \otimes t^n\}.$$

The *Heisenberg algebra* is the quotient of the tensor algebra $\mathcal{T}W$ by the two-sided ideal I generated by the expressions $[v, w] - \{v, w\} \cdot 1$ with $v, w \in W$:

$$\mathcal{H} := \mathcal{T}W/I.$$

$\mathcal{H}$ is the (restricted) tensor product of countably many copies of Clifford algebras arising from $H^{odd}(X; \mathbb{Q})$ and countably many copies of Weyl algebras arising from $H^{even}(X; \mathbb{Q})$. As W_+ is isotropic with respect to the skew-form $\{\ ,\ \}$, the subalgebra in $\mathcal{H}$ generated by W_+ is the symmetric algebra S^*W_+ (taken again in a graded sense). This becomes a double graded vector space if we define the bidegree of $\alpha \otimes t^n$ as $(n, 2n - 2 + |\alpha|)$.

Using these notations, Nakajima's Theorem can be rephrased by saying:
Sending $\alpha \otimes t^n \in W$ to $\mathfrak{q}_n(\alpha) \in \mathrm{End}(\mathbb{H})$ defines a representation of $\mathcal{H}$ on $\mathbb{H}$.

The subspace W_- of monomials of negative degree annihilates the vacuum vector $\mathbf{1} \in \mathbb{H}$ for obvious degree reasons. Hence there is an embedding

$$S^*W_+ \cong \mathcal{H}/\mathcal{H} \cdot W_- \xrightarrow{\cdot \mathbf{1}} \mathcal{H} \cdot \mathbf{1} \subset \mathbb{H}.$$

It is not difficult to check that the Poincaré series of S^*W_+ equals the right hand side of Göttsche's formula. This implies:

Corollary 2.5 (Nakajima) — *The action of $\mathcal{H}$ on $\mathbb{H}$ induces a module isomorphism $S^*W_+ \to \mathbb{H}$. In particular, $\mathbb{H}$ is irreducible and generated by the vacuum vector.* □

In fact, this can be strengthened as follows:

Consider the rational map $a : X^{[n]} \times X^{[m]} - - \to X^{[n+m]}$ which is defined on the open subset of all pairs (ξ, ξ') with disjoint support by $a(\xi, \xi') := \xi \cup \xi'$. This rational map induces homomorphisms

$$a_* : H^*(X^{[n]}; \mathbb{Q}) \otimes H^*(X^{[m]}; \mathbb{Q}) \longrightarrow H^*(X^{[n+m]}; \mathbb{Q})$$

and

$$a^* : H^*(X^{[n+m]}; \mathbb{Q}) \longrightarrow H^*(X^{[n]}; \mathbb{Q}) \otimes H^*(X^{[m]}; \mathbb{Q})$$

and hence

$$a_* : \mathbb{H} \otimes \mathbb{H} \longrightarrow \mathbb{H} \quad \text{and} \quad a^* : \mathbb{H} \to \mathbb{H} \otimes \mathbb{H}.$$

Corollary 2.6 (Nakajima, Grojnowski) — *The homomorphism a^* and a_* endow $\mathbb{H}$ with the structure of a Hopf algebra. If S^*W_+ is given the canonical Hopf algebra structure of the symmetric product, then $S^*W_+ \to \mathbb{H}$ is an isomorphism of Hopf algebras.* □

3 The boundary operator

This section contains the main technical results of the paper. The key to our solution of the Chern class problem is the introduction of the boundary operator $\mathfrak{d} \in \mathrm{End}(\mathbb{H})$. This is done in 3.2. We begin with the discussion of related topics and ingredients for later proofs.

3.1 Virasoro generators

Starting from the basic generators $\mathfrak{q}_n$ and the fundamental oscillator relations we will define the corresponding Virasoro generators $\mathfrak{L}_n$ in analogy to the procedure in conformal field theory. We will then give concrete geometric interpretations for these generators.

Let $\delta : H^*(X) \to H^*(X \times X) = H^*(X) \otimes H^*(X)$ be the push-forward map associated to the diagonal embedding. Equivalently, this is the linear map adjoint to the cup-product map. If $\delta(\alpha) = \sum_i \alpha_i' \otimes \alpha_i''$, we will write $\mathfrak{q}_n\mathfrak{q}_m\delta(\alpha)$ for $\sum_i \mathfrak{q}_n(\alpha_i')\mathfrak{q}_m(\alpha_i'')$.

Definition 3.1 — Define operators $\mathfrak{L}_n : H^*(X;\mathbb{Q}) \to \mathrm{End}(\mathbb{H})$, $n \in \mathbb{Z}$, as follows:

$$\mathfrak{L}_n := \frac{1}{2}\sum_{\nu\in\mathbb{Z}} \mathfrak{q}_\nu\mathfrak{q}_{n-\nu}\delta, \qquad \text{if } n \neq 0$$

and

$$\mathfrak{L}_0 := \sum_{\nu>0} \mathfrak{q}_\nu\mathfrak{q}_{-\nu}\delta.$$

Remark 3.2 — i) The sums that appear in the definition are formally infinite. However, as operators on any fixed vector in $\mathbb{H}$, only finitely many of them are nonzero. Hence the sums are locally finite and the operators $\mathfrak{L}_n$ are well-defined. $\mathfrak{L}_n(\alpha)$ is homogeneous of bidegree $(n, 2n + |\alpha|)$

ii) Using the physicists' normal order convention

$$:\mathfrak{q}_n\mathfrak{q}_m: := \begin{cases} \mathfrak{q}_n\mathfrak{q}_m & \text{if } n \geq m, \\ \mathfrak{q}_m\mathfrak{q}_n & \text{if } n \leq m, \end{cases}$$

the operators $\mathfrak{L}_n$ can be uniformly expressed as

$$\mathfrak{L}_n = \frac{1}{2}\sum_{\nu\in\mathbb{Z}} :\mathfrak{q}_\nu\mathfrak{q}_{n-\nu}:\delta.$$

Theorem 3.3 — *The operators $\mathfrak{L}_n$ and $\mathfrak{q}_m$ satisfy the following commutator relations:*

1. $[\mathfrak{L}_n(\alpha), \mathfrak{q}_m(\beta)] = -m \cdot \mathfrak{q}_{n+m}(\alpha\beta)$.
2. $[\mathfrak{L}_n(\alpha), \mathfrak{L}_m(\beta)] = (n-m)\cdot\mathfrak{L}_{n+m}(\alpha\beta) - \frac{n^3-n}{12}\delta_{n+m}\cdot\int_X \alpha\beta\cdot \mathrm{id}_{\mathbb{H}}$.

Proof. Assume first that $n \neq 0$. For any classes α and β with

$$\delta(\alpha) = \sum_i \alpha_i' \otimes \alpha_i''$$

we have

$$\begin{aligned}
[\mathfrak{q}_\nu(\alpha_i')\mathfrak{q}_{n-\nu}(\alpha_i''), \mathfrak{q}_m(\beta)] &= \mathfrak{q}_\nu(\alpha_i')[\mathfrak{q}_{n-\nu}(\alpha_i''), \mathfrak{q}_m(\beta)] \\
&\quad +(-1)^{|\beta|\cdot|\alpha_i''|}[\mathfrak{q}_\nu(\alpha_i'), \mathfrak{q}_m(\beta)]\mathfrak{q}_{n-\nu}(\alpha_i'') \\
&= (-m)\delta_{n+m-\nu} \cdot \mathfrak{q}_{n+m}(\alpha_i') \cdot \int_X \alpha_i''\beta \\
&\quad +(-1)^{|\beta|\cdot|\alpha|}(-m)\delta_{\nu+m} \cdot \int_X \beta\alpha_i' \cdot \mathfrak{q}_{n+m}(\alpha_i'').
\end{aligned}$$

If we sum up over all ν and i, we get

$$2[\mathfrak{L}_n(\alpha), \mathfrak{q}_m(\beta)] = \sum_\nu [\mathfrak{q}_\nu\mathfrak{q}_{n-\nu}\delta(\alpha), \mathfrak{q}_m(\beta)] = (-m) \cdot \mathfrak{q}_{n+m}(\gamma)$$

with

$$\gamma = pr_{1*}(\delta(\alpha) \cdot pr_2^*(\beta)) + (-1)^{|\beta|\cdot|\alpha|} \cdot pr_{2*}(pr_1^*(\beta) \cdot \delta(\alpha)) = 2 \cdot \alpha\beta.$$

Similarly, for $\nu > 0$,

$$[\mathfrak{q}_\nu\mathfrak{q}_{-\nu}\delta(\alpha), \mathfrak{q}_m(\beta)] = -m \cdot \mathfrak{q}_m(\alpha\beta) \cdot (\delta_{m-\nu} + \delta_{m+\nu}).$$

Thus summing up over all $\nu > 0$ we find again

$$[\mathfrak{L}_0(\alpha), \mathfrak{q}_m(\beta)] = -m \cdot \mathfrak{q}_m(\alpha\beta).$$

This proves the first part of the theorem.

As for the second part, assume first that $n \geq 0$. In order to avoid case considerations let us agree that $\mathfrak{q}_{\frac{N}{2}}$ is zero if N is odd. Then we may write:

$$\mathfrak{L}_m = \frac{1}{2}\mathfrak{q}_{\frac{m}{2}}^2\delta + \sum_{\mu > \frac{m}{2}} \mathfrak{q}_\mu\mathfrak{q}_{m-\mu}\delta.$$

By the first part of the theorem we have

$$[\mathfrak{L}_n(\alpha), \mathfrak{q}_\mu\mathfrak{q}_{m-\mu}\delta(\beta)] = \Big(-\mu\mathfrak{q}_{n+\mu}\mathfrak{q}_{m-\mu} + (\mu - m)\mathfrak{q}_\mu\mathfrak{q}_{n+m-\mu}\Big)\delta(\alpha\beta).$$

In the following calculation we suppress α, β and δ up to the very end. Summing up over all $\mu \geq 0$, we get:

$$\begin{aligned}
[\mathfrak{L}_n, \mathfrak{L}_m] &= -\frac{m}{4}(\mathfrak{q}_{n+\frac{m}{2}}\mathfrak{q}_{\frac{m}{2}} + \mathfrak{q}_{\frac{m}{2}}\mathfrak{q}_{n+\frac{m}{2}}) \\
&\quad + \sum_{\mu > \frac{m}{2}} (\mu - m)\mathfrak{q}_\mu\mathfrak{q}_{n+m-\mu} + \sum_{\mu > \frac{m}{2}} (-\mu)\mathfrak{q}_{n+\mu}\mathfrak{q}_{m-\mu} \\
&= -\frac{m}{4}(\mathfrak{q}_{n+\frac{m}{2}}\mathfrak{q}_{\frac{m}{2}} + \mathfrak{q}_{\frac{m}{2}}\mathfrak{q}_{n+\frac{m}{2}}) \\
&\quad + \sum_{\mu > \frac{m}{2}} (\mu - m)\mathfrak{q}_\mu\mathfrak{q}_{n+m-\mu} + \sum_{\mu > n+\frac{m}{2}} (n - \mu)\mathfrak{q}_\mu\mathfrak{q}_{n+m-\mu}
\end{aligned}$$

Hence

$$\begin{aligned}[\mathfrak{L}_n, \mathfrak{L}_m] - (n-m)\sum_{\mu > \frac{n+m}{2}} \mathfrak{q}_\mu \mathfrak{q}_{n+m-\mu} \quad &= \quad -\frac{m}{4}(\mathfrak{q}_{n+\frac{m}{2}}\mathfrak{q}_{\frac{m}{2}} + \mathfrak{q}_{\frac{m}{2}}\mathfrak{q}_{n+\frac{m}{2}}) \\ &\quad + \sum_{\frac{m}{2} < \mu \leq \frac{m+n}{2}} (\mu - m)\mathfrak{q}_\mu \mathfrak{q}_{m+n-\mu} \\ &\quad - \sum_{\frac{n+m}{2} < \mu \leq n+\frac{m}{2}} (n-\mu)\mathfrak{q}_\mu \mathfrak{q}_{n+m-\mu}\end{aligned}$$

Now split off the summands corresponding to the indices $\mu = \frac{m+n}{2}$ and $\mu = n + \frac{m}{2}$ from the sums. Substituting $n + m - \mu$ for μ in the second sum on the right hand side, we are left with the expression:

$$[\mathfrak{L}_n, \mathfrak{L}_m] - (n-m)\mathfrak{L}_{n+m} = -\frac{m}{4}[\mathfrak{q}_{\frac{m}{2}}, \mathfrak{q}_{n+\frac{m}{2}}] + \sum_{\frac{m}{2} < \mu < \frac{n+m}{2}} (\mu - m)[\mathfrak{q}_\mu, \mathfrak{q}_{n+m-\mu}]$$

The right hand side is zero unless $n + m = 0$. Hence we see that

$$[\mathfrak{L}_n(\alpha), \mathfrak{L}_m(\beta)] = (n-m)\mathfrak{L}_{n+m}(\alpha\beta) + \delta_{n+m} \cdot \int_X \alpha\beta \cdot N,$$

where N is the number

$$N = \sum_{0 < \nu < \frac{n}{2}} \nu(\nu - n) \qquad \text{if } n \text{ is odd,}$$

and

$$N = \sum_{0 < \nu < \frac{n}{2}} \nu(\nu - n) - \frac{n^2}{8} \quad \text{if } n \text{ is even.}$$

An easy computation shows that in both cases N equals $(n - n^3)/12$. □

Recall the definition of the varieties $E^{[n,n']} \subset X^{[n]} \times X \times X^{[n']}$ in (1.9).

Definition 3.4 — Let ℓ be a nonnegative integer and let

$$\mathfrak{e}_\ell : H^*(X) \to \mathrm{End}(\mathbb{H})$$

be the linear map

$$\mathfrak{e}_\ell(\alpha)(y) = [E^{[n+\ell,n]}]_*(\alpha \otimes y) = PD^{-1}p_{1*}([E^{[n+\ell]}] \cap (\rho^*\alpha \cdot p_2^* y))$$

for $\alpha \in H^*(X; \mathbb{Q})$ and $y \in H^*(X^{[n]}; \mathbb{Q})$.

The following theorem gives a 'finite' geometric interpretation of the infinite sums which define the Virasoro operators.

Theorem 3.5 — *Let n be a nonnegative integer.*

1.
$$[\mathfrak{e}_n(\alpha), \mathfrak{q}_m(\beta)] = \begin{cases} m \cdot \mathfrak{q}_{n+m}(\alpha\beta) & \text{if } m > 0 \text{ or } m < -n. \\ 0 & \text{else.} \end{cases}$$

2.
$$\mathfrak{e}_n + \mathfrak{L}_n = \frac{1}{2} \sum_{0<\nu<n} \mathfrak{q}_\nu \mathfrak{q}_{n-\nu} \delta.$$

Proof. Ad 1: Assume first that $m \geq 1$. To simplify the notations we introduce the short-hand
$$X^{[n_1],[n_2],\ldots,[n_k]} := X^{[n_1]} \times X^{[n_2]} \times \ldots \times X^{[n_k]}$$
Suppose $\ell \geq 0$, and consider the following diagram
$$\begin{array}{ccccc} X^{[\ell+n+m],[1],[\ell+m]} & \xleftarrow{p_{123}} & X^{[\ell+n+m],[1],[\ell+m],[1],[\ell]} & \xrightarrow{p_{345}} & X^{[\ell+m],[1],[\ell]} \\ & & \downarrow{\scriptstyle p_{1245}} & & \\ & & X^{[\ell+n+m],[1],[1],[\ell]} & & \end{array}$$
The product operator $\mathfrak{e}_n\mathfrak{q}_m$ is induced by the class
$$z := p_{1245*}(p_{123}^*[E^{[\ell+m+n,\ell+m]}] \cdot p_{345}^*[Q^{[\ell+m,\ell]}]) \in A_{2\ell+n+m+1}(Z')$$
where
$$\begin{array}{rcl} Z' & := & p_{1245}(p_{123}^{-1}(E^{[\ell+m+n,\ell+m]}) \cap p_{345}^{-1}(Q^{[\ell+m,\ell]})) \\ & \subset & Z := \{(\xi', x, y, \xi) | \exists \eta : \xi' - \eta = nx, \eta - \xi = my, x \in \eta\} \end{array}$$
Here the notation $\eta - \xi = my$ should comprise the conditions: ξ is a subscheme of η, and the ideal sheaf of ξ in η is of length m and is supported at y etc.

Similarly, the operator $\mathfrak{q}_n\mathfrak{e}_m$ is induced by a class $v \in A_{2\ell+m+n+1}(V')$ with
$$V' \subset V := \{(\xi', x, y, \xi) | \exists \eta' : \xi' - \eta' = mx, \eta' - \xi = ny, y \in \xi\}.$$
Moreover, if $T : X^{[\ell+m+n],[1],[1],[\ell]} \longrightarrow X^{[\ell+m+n],[1],[1],[\ell]}$ exchanges the two copies of X in the middle, then the commutator $[\mathfrak{e}_n, \mathfrak{q}_m]$ is induced by $z - T(v)$.

Now observe that off the diagonal $\{x = y\} \subset X^{[\ell+m+n],[1],[1],[\ell]}$ the subsets Z and $T(V)$ are equal. Moreover, there is only one component of (maximal possible) dimension $2\ell + n + m + 1$. It is easy to see that this component has multiplicity 1 both in z and $T(v)$: the intersection
$$p_{123}^{-1}(E^{[\ell+m+n,\ell+m]}) \cap p_{345}^{-1}(Q^{[\ell+m,\ell]})$$
is transversal over a general point in this component of Z, and maps injectively into Z. Thus the only contributions to $z - T(v)$ may arise from the diagonal part. Now
$$V \cap \{x = y\} = \{(\xi', x, x, \xi) | \xi' - \xi = (n + m)x, x \in \xi\}.$$

We have seen earlier (1.8) that this set has dimension $\leq 2\ell+n+m$ and hence may be disregarded. On the other hand

$$Z \cap \{x=y\} = \{(\xi', x, x, \xi) | \xi' - \xi = (n+m)x\}.$$

Again using 1.8 we see that this set has only one component D of (maximal) dimension $2\ell+n+m+1$. Moreover, this component is the image of the embedding

$$\iota : Q^{[\ell+n+m,\ell]} \to X^{[\ell+n+m],[1],[1],[\ell]}, (\xi', x, \xi) \mapsto (\xi', x, x, \xi).$$

Let $\alpha, \beta \in H^*(X;\mathbb{Q})$ and $y \in H^*(X^{[\ell]};\mathbb{Q})$. Then we have

$$\begin{aligned} & p_{1*}([D] \cap p_{23}^*(\alpha\otimes\beta)\cdot p_4^* y) \\ &\quad = \; p_{1*}(\iota_*[Q^{[\ell+n+m,\ell]}] \cap p_{23}^*(\alpha\otimes\beta)\cdot p_4^* y) \\ &\quad = \; p_{1*}([Q^{[\ell+n+m,\ell]}] \cap \iota^*(p_{23}^*(\alpha\otimes\beta)\cdot p_4^* y)) \\ &\quad = \; p_{1*}([Q^{[\ell+n+m,\ell]}] \cap p_2^*(\alpha\beta)\cdot p_3^* y) \end{aligned}$$

This shows that

$$[\mathfrak{e}_n(\alpha), \mathfrak{q}_m(\beta)] = \mu \cdot \mathfrak{q}_{n+m}(\alpha\beta)$$

for some integer μ. Hence it remains to compute the multiplicity μ of $[D]$ in z. To this end we pick a general point $d \in D$ and inspect the intersection of $p_{123}^{-1}(E^{[\ell+n+m,\ell]})$ and $p_{345}^{-1}(Q^{[\ell+m,\ell]})$ along the fibre $p_{12345}^{-1}(d)$.

A general point in D is of the form

$$d = (\xi', x, x, \xi) \quad \text{with} \quad \xi' = \xi \cup \zeta,$$

where ζ is a curvilinear subscheme of X of length $n+m$, supported in a single point x which is disjoint from ξ. Since ζ is curvilinear, there is a unique subscheme $\eta \subset \zeta$ of length m, and hence $p_{12345}^{-1}(d)$ consists of the single point

$$d' = (\xi \cup \zeta, x, \xi \cup \eta, x, \xi)$$

Near d' the varieties $X^{[\ell+m+n],[1],[\ell+m],[1],[\ell]}$ and $X^{[\ell],[\ell],[\ell]} \times X^{[m+n],[1],[m],[1]}$ are locally isomorphic; and similarly $E^{[\ell+m+n,\ell+m]}$ to $X^{[\ell]} \times E^{[m+n,m]}$ and $Q^{[\ell+m,\ell]}$ to $X^{[\ell]} \times X_0^{[m]}$. Thus we may split off the factors $X^{[\ell]}$ from the geometric picture. In the end this amounts to saying that we may assume without loss of generality that $\ell = 0$.

Moreover, the calculation is local in X, so that we may assume that $X = \mathbb{A}^2 = \mathrm{Spec}\mathbb{C}[z,w]$ and $\mathcal{I}_\zeta = (w, z^{n+m})$, $\mathcal{I}_\eta = (w, z^m)$ and $\mathcal{I}_x = (w, z)$. Then d' has an affine neighbourhood $\cong \mathbb{A}^{4m+2n+4}$ in $X^{[n+m],[1],[m],[1]}$ with coordinate functions

$$a_0, \ldots, a_{n+m-1}, b_0, \ldots, b_{n+m-1}, w_1, z_1, c_0, \ldots, c_{m-1}, d_0, \ldots, d_{m-1}, w_2, z_2,$$

which parametrises quadrupels (ζ, x, η, y) of subschemes in X given by the ideals

$$(w - g_1(z), f_1(z)), \quad (w - w_1, z - z_1), \quad (w - g_2(z), f_2(z)), \quad (w - w_2, z - z_2),$$

where

$$f_1(z) = a_0 + a_1 z + \ldots + z^{n+m}, \quad g_1(z) = b_0 + b_1 z + \ldots + b_{n+m-1} z^{n+m-1}$$

and

$$f_2(z) = c_0 + c_1 z + \ldots + z^m, \quad g_2(z) = d_0 + d_1 z + \ldots + z^m.$$

Now (η, y) belongs to $X_0^{[m]}$, i.e. $\mathrm{Supp}(\eta) = \{y\}$, if and only if

$$f_2(z) = (z - z_2)^m \text{ and } w_2 = g_2(z_2). \tag{6}$$

And (ζ, x, η) belongs to $Q^{[n+m,m]}$ if and only if the following three conditions are satisfied: $\eta \subset \zeta$, i.e.

$$g_1(z) = g_2(z) + f_2(z) \cdot h(z) \text{ and } f_1(z) = f_2(z) \cdot k(z) \tag{7}$$

with polynomials h and k of degree $n-1$ and n, respectively; the ideal sheaf $\mathcal{I}_{\eta/\zeta}$ is supported at x, i.e.

$$k(z) = (z - z_1)^m \text{ and } w_1 = g_1(z_1) \tag{8}$$

and finally, x must be contained in η, which imposes the condition

$$f_2(z_1) = 0 \tag{9}$$

One easily checks that the equations (6) - (8) cut out a smooth subvariety which projects isomorphically to the affine space $\mathrm{Spec}\,\mathbb{C}[z_1, z_2, b_0, \ldots, b_{n+m-1}]$. Moreover, in these coordinates the last condition (9) simply reads $(z_1 - z_2)^m = 0$. Hence the multiplicity μ equals the exponent m.

Next, we consider the case $[\mathfrak{e}_n, \mathfrak{q}_{-m}]$ with $0 \leq m \leq n$. There is nothing to prove if $m = 0$. Hence assume that $m > 0$. Dimension arguments similar to the ones above show that the cycle v which induces the commutator $[\mathfrak{q}_{-m}, \mathfrak{e}_n]$ must be supported on the closed subsets

$$V := \{(\xi, x, x, \zeta) | \xi \supset \zeta \ni x, \xi - \zeta = (n+m)x\} \subset X^{[\ell+n-m],[1],[1],[\ell]}, \quad \ell \geq 0.$$

The cycle v has degree $2\ell + n - m + 1$, so that it suffices to show that $\dim(V) \leq 2\ell + n - m$. This follows from Lemma 1.8.

It remains to consider the case $[\mathfrak{e}_n, \mathfrak{q}_m]$ with $m < -n$. A dimension check of the set-theoretic support of the intersection cycle shows that we must have

$$[\mathfrak{e}_n(\alpha), \mathfrak{q}_{-m}(\beta)] = \mu \cdot \mathfrak{q}_{n-m}(\alpha\beta)$$

for some integer μ, independently of α and β. To determine μ, we proceed algebraically and take the commutator with $\mathfrak{q}_{m-n}(1)$:

$$[\,[\mathfrak{e}_n(\alpha), \mathfrak{q}_{-m}(\beta)], \mathfrak{q}_{m-n}(1)] = \mu \cdot [\mathfrak{q}_{n-m}(\alpha\beta), \mathfrak{q}_{m-n}(1)] = \mu(n-m) \int_X \alpha\beta \cdot \mathrm{id}_{\mathbb{H}}.$$

On the other hand, combining the Jacobi identity, the oscillator relations and the first part of the proof yields

$$\begin{aligned}[\,[\mathfrak{e}_n(\alpha),\mathfrak{q}_{-m}(\beta)],\mathfrak{q}_{m-n}(1)] &= [\,[\mathfrak{e}_n(\alpha),\mathfrak{q}_{m-n}(1)],\mathfrak{q}_{-m}(\alpha)] \\ &= (m-n)[\mathfrak{q}_m(\alpha),\mathfrak{q}_{-m}(\beta)] \\ &= m(m-n)\int_X \alpha\beta\cdot \mathrm{id}_{\mathbb{H}}.\end{aligned}$$

It follows that $\mu = -m$.

Ad 2: Consider the difference $\mathfrak{y} := \mathfrak{e}_n(\alpha) + \mathfrak{L}_n(\alpha) - \frac{1}{2}\sum_{\nu=1}^{n-1}\mathfrak{q}_\nu\mathfrak{q}_{n-\nu}\delta(\alpha)$. Comparing the expressions in 3.3 and part 1 of the theorem we see that $\mathfrak{y}$ commutes with all operators $\mathfrak{q}_m$, $m \in \mathbb{Z}$. Since $\mathbb{H}$ is a simple $\mathcal{N}$-module, $\mathfrak{y}$ must be a scalar (in some algebraic extension of $\mathbb{Q}$), which is impossible: if $n > 0$, then $\mathfrak{y}$ has non-trivial bidegree $(n, 2n+|\alpha|)$, and if $n = 0$, it is easy to see directly that $\mathfrak{y}\cdot\mathbf{1} = 0$. □

Remark 3.6 — In particular, the operator $\mathfrak{L}_0(\alpha)$ has the following geometric interpretation: the universal family $\Xi_n \subset X^{[n]}\times X$ induces a homomorphism

$$[\Xi_n]_* : H^*(X;\mathbb{Q}) \longrightarrow H^*(X^{[n]};\mathbb{Q}),$$

and

$$\mathfrak{L}_0(\alpha)(y) = [\Xi_n]_*(\alpha)\cdot y \quad \text{for all} \quad y \in H^*(X^{[n]},\mathbb{Q}).$$

In particular, if we insert $\alpha = 1_X$, we get

$$\mathfrak{L}_0(1_X)(y) = n\cdot y \text{ for all } y \in H^*(X^{[n]};\mathbb{Q}).$$

Thus $\mathfrak{L}_0(1_X)$ is the 'energy' or 'counting' operator, that measures with which 'energy level', i.e. how many points we are dealing. This can, of course, also be deduced directly from the definition of $\mathfrak{L}_0$.

3.2 The boundary of the Hilbert scheme

For any partition $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_s > 0)$ of n the tuples $\sum_{1\leq i\leq s}\lambda_i x_i$, $x_i \in X$, form a locally closed subset $S^n_\lambda X$ in $S^n X$. Let $X^{[n]}_\lambda = \rho^{-1}(S^n_\lambda X)$. It follows from Briançon's Theorem that $X^{[n]}_\lambda$ is irreducible and

$$\dim(X^{[n]}_\lambda) = \sum_{1\leq i\leq s}(\lambda_i+1) = n+s.$$

The generic open stratum is $X^{[n]}_{(1,1,\ldots,1)}$. It corresponds to the configuration space of unordered n-tuples of pairwise distinct points. Furthermore, there is precisely one stratum of codimension 1, namely $X^{[n]}_{(2,1,\ldots,1)}$.

If $\lambda = (\lambda_1, \dots, \lambda_s)$ and $\mu = (\mu_1, \dots, \mu_{s'})$ are partitions of n, then $X^{[n]}_\mu$ is contained in the closure of $X^{[n]}_\lambda$ if and only if there is a surjection

$$\varphi : \{1, \dots, s\} \to \{1, \dots, s'\}$$

such that $\mu_j = \sum_{i \in \varphi^{-1}(j)} \lambda_i$ for all j. It follows that

$$\partial X^{[n]} := \bigcup_{\lambda \neq (1,\dots,1)} X^{[n]}_\lambda = \overline{X^{[n]}_{(2,1,\dots,1)}}$$

is an irreducible divisor in $X^{[n]}$. As it is the complement of the configuration space in $X^{[n]}$ we might and will call it the *boundary* of $X^{[n]}$.

Lemma 3.7 — *Let $E \subset X^{[n+1,n]}$ be the exceptional divisor. Then*

$$p_1^* \partial X^{[n+1]} - p_2^* \partial X^{[n]} = 2 \cdot E.$$

Proof. Points in $X^{[n+1,n]}$ are triples (ξ', x, ξ) with $\xi \subset \xi'$ and $\mathcal{I}_{\xi/\xi'} \cong k(x)$, and $p_1^{-1}(\partial X^{[n+1]})$ consists of those triples such that there is a point $y \in X$ with $\ell(\xi'_y) \geq 2$. Now either $y = x$, in which case $\ell(\xi_x) = \ell(\xi'_x) - 1 \geq 1$ and therefore $(\xi', x, \xi) \in E$, or $y \neq x$, in which case $\ell(\xi_y) = \ell(\xi'_y) \geq 2$ so that $(\xi', x, \xi) \in p_2^{-1}(\partial X^{[n]})$. Hence set-theoretically, we have $p_1^{-1}(\partial X^{[n+1]}) = p_2^{-1}(\partial X^{[n]}) \cup E$. We must check the multiplicities.

Off the exceptional divisor E we have $X^{[n+1,n]} \setminus E = X^{[n]} \times X \setminus \Xi_n$, which embeds as an open subset into $X^{[n+1]}$. Clearly, $(X^{[n]} \times X \setminus \Xi_n) \cap \partial X^{[n+1]} = pr_1^* \partial X^{[n]}$. Thus $p_2^* \partial X^{[n+1]} - p_1^* \partial X^{[n]} = \mu \cdot E$. In order to compute the multiplicity μ we pick a general point in E which is of the form $(\eta \cup \zeta, x, \eta \cup \{x\})$, where $x \notin \eta$ and ζ has length 2 and is supported at x. Without loss of generality we may assume that η is empty, i.e., that $n = 1$. But then $X^{[2,1]}$ is the blow-up of $X \times X$ along the diagonal, E is the exceptional divisor, and $p_2 : X^{[2,1]} \to X^{[2]}$ is the quotient map for the action of $\mathfrak{S}_2$ on the blow-up. In this picture E is the ramification divisor, $\partial X^{[2]}$ is the branching divisor, and the ramification order is 2. Hence indeed, $\mu = 2$. □

We will need a different description of the divisor $\partial X^{[n]}$ in sheaf theoretic terms. Let $p : \Xi_n \to X^{[n]}$ be the projection, and define sheaves

$$\mathcal{O}_X^{[n]} := p_*(\mathcal{O}_{\Xi_n}) \in \mathrm{Coh}(X^{[n]}).$$

As p is flat and finite of degree n, $\mathcal{O}_X^{[n]}$ is locally free of rank n. The fibre at a point $\xi \in X^{[n]}$ is the $\mathbb{C}$-vector space underlying the algebra $\mathcal{O}_\xi$.

Lemma 3.8 — $c_1(\mathcal{O}_X^{[n]}) = -\frac{1}{2}\left[\partial X^{[n]}\right]$.

Proof. Consider the following incidence scheme with the natural projections:

$$X^{[n+1]} \longleftarrow X^{[n+1,n]} \longrightarrow X^{[n]}.$$

We have seen earlier in 1.4 that $\mathcal{I}_{n+1,n} = (id,\rho)_*\mathcal{O}_{X^{[n+1,n]}}(-E)$ and hence that $p_*\mathcal{I}_{n+1,n} = \mathcal{O}_X^{[n+1,n]}(-E)$. This shows

$$p_1^* c_1(\mathcal{O}_X^{[n+1]}) - p_2^* c_1(\mathcal{O}_X^{[n]}) = -E.$$

On the other hand, by Lemma 3.7,

$$p_1^*\partial X^{[n+1]} - p_2^*\partial X^{[n]} = 2\cdot E.$$

Therefore, if we put $\gamma_n := c_1(\mathcal{O}_X^{[n]}) + \frac{1}{2}\partial X^{[n]}$, we get $p_2^*\gamma_n = p_1^*\gamma_{n+1}$. Now $\gamma_0 = \gamma_1 = 0$, since $\mathcal{O}_X^{[1]} \cong \mathcal{O}_{X^{[1]}}$ and $\partial X = \emptyset$. Assume by induction that $\gamma_n = 0$. It follows that $p_1^*\gamma_{n+1} = 0$, and since $p_1 : X^{[n+1,n]} \to X^{[n+1]}$ is generically finite and surjective, we must have $\gamma_{n+1} = 0$ as well. □

Definition 3.9 — Let $\mathfrak{d} : \mathbb{H} \to \mathbb{H}$ be the homogeneous linear map of bidegree $(0,2)$ given by

$$\mathfrak{d}(x) := c_1(\mathcal{O}_X^{[n]})\cdot x = -\frac{1}{2}\left[\partial X^{[n]}\right]\cdot x \quad \text{for all } x \in H^*(X^{[n]}).$$

For any endomorphism $\mathfrak{f} \in \mathrm{End}(\mathbb{H})$ its derivative is $\mathfrak{f}' := [\mathfrak{d},\mathfrak{f}]$. As usual, we write $\mathfrak{f}^{(n)} := (\operatorname{ad}\mathfrak{d})^n(\mathfrak{f})$ for the higher derivatives.

It follows directly from the definition of the commutator that $\mathfrak{f} \mapsto \mathfrak{f}'$ is a derivation, i.e. for any two operators $\mathfrak{a},\mathfrak{b} \in \mathrm{End}(\mathbb{H})$ the 'Leibniz rule' holds:

$$(\mathfrak{a}\mathfrak{b})' = \mathfrak{a}'\mathfrak{b} + \mathfrak{a}\mathfrak{b}' \quad \text{and} \quad [\mathfrak{a},\mathfrak{b}]' = [\mathfrak{a}',\mathfrak{b}] + [\mathfrak{a},\mathfrak{b}'].$$

Moreover, if $\mathfrak{f} : H^*(X^{[\ell]}) \to H^*(X^{[n]})$ is a homogeneous linear map, then $|\mathfrak{f}'| = |\mathfrak{f}| + 2$, so that $\mathfrak{f}$ and $\mathfrak{f}'$ have the same parity. Furthermore,

$$(\mathfrak{f}')^\dagger = -(\mathfrak{f}^\dagger)'.$$

Indeed,

$$\begin{aligned}\int_{X^{[n]}} \mathfrak{f}'(y)\cdot z &= \int_{X^{[n]}} \mathfrak{f}(y)\cdot\mathfrak{d}(z) - \mathfrak{f}(\mathfrak{d}(y))\cdot z \\ &= (-1)^{|y|\cdot|\mathfrak{f}|}\int_{X^{[\ell]}} y\cdot\mathfrak{f}^\dagger(\mathfrak{d}(z)) - y\cdot\mathfrak{d}\mathfrak{f}^\dagger(z) \\ &= -(-1)^{|y|\cdot|\mathfrak{f}|}\int_{X^{[\ell]}} y\cdot(\mathfrak{f}^\dagger)'(z).\end{aligned}$$

Let $n' > n$ be nonnegative integers, and consider the incidence variety $X^{[n',n]} \subset X^{[n']}\times X^{[n]}$. Recall the definition of the ideal sheaf $\mathcal{I}_{n',n}$ and the exact sequence

$$0 \to \mathcal{I}_{n',n} \to p_{1,X}^*\mathcal{O}_{\Xi_{n'}} \to p_{2,X}^*\mathcal{O}_{\Xi_n} \to 0.$$

Then $p_*(\mathcal{I}_{n',n})$ is a locally free sheaf of rank $n'-n$ on $X^{[n',n]}$.

In a certain sense, the following lemma simply is a reformulation of the definition of the derivative.

Lemma 3.10 — *Let $u_* : H^*(X^{[n]};\mathbb{Q}) \to H^*(X^{[n']};\mathbb{Q})$ be the induced linear map associated to a class $u \in A_*(X^{[n',n]})$. Then*

$$(u_*)' = (c_1(p_*(\mathcal{I}_{n',n})) \cdot u)_*.$$

Proof. Let $y \in H^*(X^{[n]};\mathbb{Q})$. Then

$$\begin{aligned}(u_*)'(y) &= \mathfrak{d}(u_*(y)) - u_*(\mathfrak{d}(y)) \\ &= c_1(p_*\mathcal{O}_{\Xi_{n'}}) \cdot PD^{-1}p_{1*}(u \cdot p_2^*y) \\ &\quad -PD^{-1}p_{1*}(u \cdot p_2^*(c_1(p_*\mathcal{O}_{\Xi_n}) \cdot y)) \\ &= PD^{-1}p_{1*}((p_1^*c_1(p_*\mathcal{O}_{\Xi_{n'}}) - p_2^*c_1(p_*\mathcal{O}_{\Xi_n})) \cdot u \cdot p_2^*y) \\ &= v_*(y)\end{aligned}$$

with $v = (p_1^*c_1(p_*\mathcal{O}_{\Xi_{n'}}) - p_2^*c_1(p_*\mathcal{O}_{\Xi_n})) \cdot u$, and

$$\begin{aligned}p_1^*c_1(p_*\mathcal{O}_{\Xi_{n'}}) - p_2^*c_1(p_*\mathcal{O}_{\Xi_n}) &= c_1(p_*p_{1,X}^*\mathcal{O}_{\Xi_{x'}}) - c_1(p_*p_{2,X}^*\mathcal{O}_{X_n}) \\ &= c_1(p_*\mathcal{I}_{n',n}).\end{aligned}$$

□

3.3 The derivative of $\mathfrak{q}_n$

In order to understand the intersection behaviour of the boundary $\partial X^{[n]}$ we need to know how the operator $\mathfrak{d}$ commutes with the basic operators $\mathfrak{q}_n$, in other words: we need to compute the derivative of $\mathfrak{q}_n$.

The following theorem is the main technical theorem of this paper. It describes the derivative of the operator $\mathfrak{q}_n$ in two ways: By its action on any of the other basic operators, and as a polynomial expression in the basic operators.

Let K denote the canonical class of the surface X.

Theorem 3.11 — *For all $n, m \in \mathbb{Z}$ and $\alpha, \beta \in H^*(X;\mathbb{Q})$ the following holds:*

1. $[\mathfrak{q}_n'(\alpha), \mathfrak{q}_m(\beta)] = -nm \cdot \left\{ \mathfrak{q}_{n+m}(\alpha\beta) + \frac{|n|-1}{2}\delta_{n+m} \cdot \int_X K\alpha\beta \cdot \mathrm{id}_{\mathbb{H}} \right\}.$
2. $\mathfrak{q}_n'(\alpha) = n \cdot \mathfrak{L}_n(\alpha) + \frac{n(|n|-1)}{2}\mathfrak{q}_n(K\alpha).$

Corollary 3.12 — *The operators $\mathfrak{d}$ and $\mathfrak{q}_1(\alpha)$, $\alpha \in H^*(X)$, suffice to generate $\mathbb{H}$ from the vacuum $\mathbf{1}$.* □

Proof of the theorem. The second assertion is an immediate consequence of the first: by Nakajima's relations 2.4 and the relations 3.3 we see that

$$\begin{aligned}[n \cdot \mathfrak{L}_n(\alpha) + \frac{n(|n|-1)}{2}\mathfrak{q}_n(K\alpha), \mathfrak{q}_m(\beta)] = \\ -nm \cdot \mathfrak{q}_{n+m}(\alpha\beta) + \delta_{n+m}\frac{n^2(|n|-1)}{2}\int_X K\alpha\beta \cdot \mathrm{id}_{\mathbb{H}}.\end{aligned}$$

Hence the difference of $\mathfrak{q}'_n$ and the expression on the right hand side in the theorem commutes with all operators $\mathfrak{q}_m$, $m \in \mathbb{Z}$. Since $\mathbb{H}$ is an irreducible $\mathcal{N}$-module, it follows from Schur's Lemma that this difference is given by multiplication with a scalar (say, after passage to some algebraic closure of $\mathbb{Q}$). But this is impossible for degree reasons: the bidegree of $\mathfrak{q}'_n(\alpha)$ is $(n, 2n + |\alpha|)$. (The case $n = 0$ being trivial anyhow.)

The proof of the first assertion has two parts of quite different nature: We need to distinguish the cases $n + m \neq 0$ and $n + m = 0$ and deal with them separately.

Proposition 3.13 — $[\mathfrak{q}'_n(\alpha), \mathfrak{q}_m(\beta)] = -nm \cdot \mathfrak{q}_{n+m}(\alpha\beta)$ *for any two integers* n, m *with* $n + m \neq 0$ *and cohomology classes* $\alpha, \beta \in H^*(X)$.

Proof. Step 1: Assume that n and m are positive. We proceed as in the proof of Theorem 3.5. Let ℓ be nonnegative, and consider the diagram

$$\begin{array}{ccccc} X^{[\ell+n+m],[1],[\ell+m]} & \xleftarrow{p_{123}} & X^{[\ell+n+m],[1],[\ell+m],[1],[\ell]} & \xrightarrow{p_{345}} & X^{[\ell+m],[1],[\ell]} \\ & & \downarrow{\scriptstyle p_{1245}} & & \\ & & X^{[\ell+n+m],[1],[1],[\ell]}. & & \end{array}$$

Let

$$v := p_{123}^*[Q^{[\ell+m+n,\ell+m]}] \cdot p_{345}^*[Q^{[\ell+m,\ell]}] \in A_{2\ell+m+n+2}(V),$$
$$V := p_{123}^{-1}(Q^{[\ell+m+n,\ell+m]}) \cap p_{345}^{-1}(Q^{[\ell+m,\ell]}).$$

According to Lemma 3.10, the operator $\mathfrak{q}'_n\mathfrak{q}_m$ is induced by the class

$$w = p_{1245*}(p_{123}^* c_1(\mathcal{I}_{\ell+m+n,\ell+m}) \cdot v) \in A_{2\ell+m+n+1}(W), W := p_{1245}(V).$$

Let $V' \subset V$ and $W' \subset W$ denote the open subsets of those tuples $(\xi, x, \sigma, y, \zeta)$ and (ξ, x, y, ζ), respectively, where either $x \neq y$ or $x = y$ but ξ_x is curvilinear. Certainly, $V' = p_{1245}^{-1}(W')$, but in fact we even have that $p_{1245} : V' \to W'$ is an isomorphism: for the conditions imposed on V' imply that σ is already determined by the remaining data (ξ, x, y, ζ).

Claim: V' is irreducible of dimension $2\ell + n + m + 2$.

For it follows from Briançon's Theorem that the open part $V' \setminus \{x = y\}$ is irreducible of dimension $2\ell + (n + 1) + (m + 1)$, and tuples of the second kind, i.e. (ξ, x, x, ζ) with ξ_x curvilinear, are easily seen to deform into this open subset.

Claim: $\dim(W \setminus W') < 2\ell + m + n + 1$. *In particular, the complement of W' in W cannot support any contribution to w.*

Indeed, the set $T = \{(\xi, x, x, \zeta) | \xi - \zeta = (n + m)x\}$ has a stratification $T = \coprod_{i \geq 0} T_i$, where the stratum T_i is the locally closed set of all tuples with $\mathrm{length}(\zeta_x) = i$. Let $T'_0 \subset T_0$ be the closed subset that consists of tuples where ξ_x is not curvilinear. Then $W \setminus W' \subset T'_0 \cup T_1 \cup T_2 \ldots$. Now T_0 is irreducible of dimension $2\ell + (n + m + 1)$, and T'_0 is a proper closed subset and therefore has strictly smaller dimension. The assertion now follows from Lemma 1.8.

Claim: The intersection of $p_{123}^*[Q^{[\ell+m+n]}]$ and $p_{345}^*[Q^{[\ell+m,m]}]$ is transversal at general points of V'.

In fact, the intersection is transversal at all points with $x \neq y$ and ξ curvilinear.

We conclude, that the intersection cycle v equals $\overline{[V']} + r$, where r is a cycle supported on $p_{1245}^{-1}(W \setminus W')$ and therefore irrelevant for our further computations for dimension reasons. Let us return to the definition of the cycle w.

Identifying V' and W' we see that the variety W' parametrises three families

$$Z \subset \Sigma \subset \Xi \subset W' \times X$$

of subschemes in X. In terms of these we can summarise the discussion above by stating that $\mathfrak{q}_n' \mathfrak{q}_m$ is induced by the cycle

$$c_1(p_* \mathcal{I}_{\Sigma/\Xi}) \cdot [W'] \in A_*(W').$$

Having reached this point we pause to reflect what changes in this picture if we exchange the order of the operators $\mathfrak{q}_n$ and $\mathfrak{q}_m$. Up to the usual twist T that flips the factors X in $X^{[\ell+m+n],[1],[1],[\ell]}$, not a iota is changed in W'. Indeed, W' parametrises not only three but rather four families of subschemes

$$\begin{array}{ccccc} & & \Sigma' & & \\ & \nearrow & & \searrow & \\ Z & & & & \Xi \\ & \searrow & & \nearrow & \\ & & \Sigma'' & & \end{array}$$

where Σ' and Σ'' are characterised by the property that at a point $s = (\Xi_s, x, y, Z_s) \in W'$ the subschemes $\Sigma_s', \Sigma_s'' \subset \Xi_s$ are the unique ones with

$$\Sigma_s' - Z_s = mx\,, \quad \Xi_s - \Sigma_s' = ny$$

and

$$\Sigma_s'' - Z_s = ny\,, \quad \Xi_s - \Sigma_s'' = mx.$$

This means: the commutator $[\mathfrak{q}_n', \mathfrak{q}_m]$ is induced by the cycle

$$\Big(c_1(p_* \mathcal{I}_{\Sigma'/\Xi}) - c_1(p_* \mathcal{I}_{Z/\Sigma''})\Big) \cdot [W'] \in A_{2\ell+n+m+1}(X^{[\ell+n+m],[1],[1],[\ell]}).$$

The ideal sheaves corresponding to the various inclusions between the families Z, Σ', Σ'' and Ξ are related by the following commutative diagram of short exact sequences

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \mathcal{I}_{\Sigma'/\Xi} & \longrightarrow & \mathcal{I}_{Z/\Xi} & \longrightarrow & \mathcal{I}_{Z/\Sigma'} & \longrightarrow & 0 \\ & & \varphi \downarrow & & \| & & \uparrow & & \\ 0 & \longleftarrow & \mathcal{I}_{Z/\Sigma''} & \longleftarrow & \mathcal{I}_{Z/\Xi} & \longleftarrow & \mathcal{I}_{\Sigma''/\Xi} & \longleftarrow & 0 \end{array} .$$

The homomorphism

$$p_*\varphi : p_*\mathcal{I}_{\Sigma'/\Xi} \to p_*\mathcal{I}_{Z/\Sigma''}$$

is an isomorphism off the diagonal $\{x = y\} \subset W'$. On the other hand the closure of $W' \cap \{x = y\}$ equals the image of the 'diagonal' embedding $Q^{[\ell+m+n,\ell]} \to X^{[\ell+m+n],[1],[1],[\ell]}$. It follows that

$$\Big(c_1(p_*\mathcal{I}_{\Sigma'/\Xi}) - c_1(p_*\mathcal{I}_{Z/\Sigma''})\Big) \cdot [W'] = -\mu \cdot [Q^{[\ell+m+n,\ell]}]$$

where μ is the length of coker$(p_*\varphi)$ at the generic point of the variety $Q^{[\ell+m+n,\ell]}$. This proves

$$[\mathfrak{q}'_n(\alpha), \mathfrak{q}_m(\beta)] = -\mu \cdot \mathfrak{q}_{n+m}(\alpha\beta),$$

and it remains to show that

$$\mu = nm.$$

A general point $d = (\xi, x, y, \zeta)$ of $Q^{[\ell+m+n,\ell]}$ is of the form $(\zeta \cup \eta, x, x, \zeta)$ where $\eta \cap \zeta = \emptyset$ and η is a curvilinear subscheme supported at x. As the computation is local in X we may apply the same reduction process as in the proof of Theorem 3.5: we may assume that $\ell = 0$, that $X = \mathbb{A}^2 = \mathrm{Spec}\mathbb{C}[z, w]$, $x = (0, 0)$ and $I_\zeta = (w, z^n)$. Then there is an open neighbourhood of this point d in W' which isomorphic to $\mathbb{A}^{n+m+2} = \mathrm{Spec}\mathbb{C}[a_0, \ldots, a_{n+m-1}, s, t]$ such that the families Ξ, Σ' and Σ'' are given by the ideals

$$I_\Xi = (w - f(z), (z-t)^n(z-s)^m), \quad I_{\Sigma'} = (w - f(z), (z-s)^m)$$

and

$$I_{\Sigma''} = (w - f(z), (z-t)^n),$$

where $f(z) = a_0 + a_1 z + \ldots + a_{n+m-1}z^{n+m-1}$. We find

$$p_*\mathcal{O}_{\Sigma''} = \mathbb{C}[\underline{a}, s, t][z]/(z-t)^n$$

and

$$p_*\mathcal{I}_{\Sigma'/\Xi} = (z-s)^m \cdot \mathbb{C}[\underline{a}, s, t][z]/(z-s)^m(z-t)^n.$$

The cokernel of

$$p_*\varphi : (z-s)^m \cdot \mathbb{C}[\underline{a}, s, t][z]/(z-s)^m(z-t)^n \longrightarrow \mathbb{C}[\underline{a}, s, t][z]/(z-t)^n$$

is isomorphic to the $\mathbb{C}[\underline{a}, s, t]$-module

$$\mathbb{C}[\underline{a}, s, t][z]/((z-s)^m, (z-t)^n) \cong \mathbb{C}[\underline{a}, s+t][z-s, z-t]/((z-s)^m, (z-t)^n).$$

This module is supported along the diagonal $\{s = t\}$ (as we expected), and its stalk at the generic point of the diagonal has length nm (as we had to prove).

Step 2: Assume that m is positive and $-m < n < 0$. First one shows as above that the commutator $[\mathfrak{q}'_n, \mathfrak{q}_m]$ is induced by cycles in $A_{2\ell+n+m+1}(X^{[\ell+m+n],[1],[1],[\ell]})$

for each $\ell \geq 0$, which are supported on the diagonally embedded varieties $Q^{[\ell+m+n,\ell]}$, so that

$$[\mathfrak{q}_n'(\alpha), \mathfrak{q}_m(\beta)] = -c_{n,m} \cdot \mathfrak{q}_{n+m}(\alpha\beta)$$

for certain constants $c_{n,m}$. In order to determine these constants we apply the commutator $[\,.\,, \mathfrak{q}_{-n-m}(1)]$. Then the oscillator relations yield for the right hand side

$$-c_{n,m}(n+m) \int_X \alpha\beta \cdot \mathrm{id}_{\mathbb{H}}.$$

On the other hand

$$[\,[\mathfrak{q}_n'(\alpha), \mathfrak{q}_m(\beta)], \mathfrak{q}_{-n-m}(1)] \;\; = \;\; [\,[\mathfrak{q}_n'(\alpha), \mathfrak{q}_{-n-m}(1)], \mathfrak{q}_m(\beta)]$$

Now

$$\begin{aligned} [\mathfrak{q}_n'(\alpha), \mathfrak{q}_{-n-m}(1)] &= (-1)^m [(\mathfrak{q}_{-n}^\dagger)'(\alpha), \mathfrak{q}_{n+m}^\dagger(1)] \\ &= -(-1)^m [\mathfrak{q}_{n+m}(1), \mathfrak{q}'_{-n}(\alpha)]^\dagger, \end{aligned}$$

which by Step 1 equals $(-1)^m n(n+m)\mathfrak{q}_m(\alpha)^\dagger = n(n+m)\mathfrak{q}_{-m}(\alpha)$. Hence

$$\begin{aligned} [\,[\mathfrak{q}_n'(\alpha), \mathfrak{q}_m(\beta)], \mathfrak{q}_{-n-m}(1)] &= n(n+m)[\mathfrak{q}_{-m}(\alpha), \mathfrak{q}_m(\beta)] \\ &= n(n+m)(-m) \int_X \alpha\beta \cdot \mathrm{id}_{\mathbb{H}}. \end{aligned}$$

Choose classes α, β with $\int_X \alpha\beta \neq 0$. It follows that $c_{n,m} = nm$.

Step 3: The general case can now be reduced formally to the cases already treated. The assertion is certainly trivial if either $n = 0$ or $m = 0$. If the assertion is known to be true for some pair (n, m), we may apply the operation $\dagger$ to both sides and find:

$$\begin{aligned} [\mathfrak{q}'_{-n}(\alpha), \mathfrak{q}_{-m}(\beta)] &= (-1)^{n+m} [(\mathfrak{q}_n^\dagger)'(\alpha), \mathfrak{q}_m^\dagger(\beta)] \\ &= -(-1)^{n+m} [(\mathfrak{q}_n')^\dagger(\alpha), \mathfrak{q}_m^\dagger(\beta)] \\ &= (-1)^{n+m} [\mathfrak{q}_n'(\alpha), \mathfrak{q}_m(\beta)]^\dagger = -nm \cdot (-1)^{n+m} \mathfrak{q}_{n+m}^\dagger(\alpha\beta) \\ &= (-n)(-m) \cdot \mathfrak{q}_{-n-m}(\alpha\beta). \end{aligned}$$

This and the identity

$$[\mathfrak{q}_n'(\alpha), \mathfrak{q}_m(\beta)] = (-1)^{|\alpha|\cdot|\beta|} [\mathfrak{q}_m'(\beta), \mathfrak{q}_n(\alpha)]$$

allow us to reduce anything to cases checked in Step 1 and Step 2. □

In order to prove part 1 of Theorem 3.11, it remains to treat the case $n + m = 0$. This will be done in two steps. First, we prove a qualitative statement about the structure of the 'correction term', and afterwards we determine the precise value of the 'coefficient' K_n:

Proposition 3.14 — *There exist rational divisors $K_n \in Pic(X) \otimes \mathbb{Q}$, $n \in \mathbb{Z}$, with $K_0 = 0$ and $K_{-n} = K_n$ and such that*

$$[\mathfrak{q}'_n(\alpha), \mathfrak{q}_{-n}(\beta)] = n^2 \cdot \int_X K_n \alpha\beta \cdot \mathrm{id}_{\mathbb{H}} \tag{10}$$

for all $\alpha, \beta \in H^*(X)$.

Proof. There is nothing to prove for $n = 0$. Moreover,

$$[\mathfrak{q}'_n(\alpha), \mathfrak{q}_{-n}(\beta)] = (-1)^{|\alpha|\cdot|\beta|} \cdot [\mathfrak{q}'_{-n}(\beta), \mathfrak{q}_n(\alpha)].$$

It follows that if there is a divisor K_n so that (10) holds for n, then (10) also holds for $-n$ with the choice $K_{-n} = K_n$. Hence it suffices to prove the proposition for positive integers n.

Let ℓ be a nonnegative integer and consider the diagram

$$\begin{array}{ccccc} X^{[\ell],[1],[\ell+n]} & \xleftarrow{p_{123}} & X^{[\ell],[1],[\ell+n],[1],[\ell]} & \xrightarrow{p_{345}} & X^{[\ell+n],[1],[\ell]} \\ & & \downarrow{\scriptstyle p_{1245}} & & \\ & & X^{[\ell],[1],[1],[\ell]}. & & \end{array}$$

Let

$$v := p_{123}^*[Q^{[\ell,\ell+n]}] \cdot p_{345}^*[Q^{[\ell+n,\ell]}] \in A_{2\ell+2}(V),$$
$$V := p_{123}^{-1}(Q^{[\ell,\ell+n]}) \cap p_{345}^{-1}(Q^{[\ell+n,\ell]}).$$

According to Lemma 3.10, the operator $\mathfrak{q}'_{-n}\mathfrak{q}_n$ is induced by the class

$$w = (-1)^n p_{1245*}(p_{123}^* c_1(\mathcal{I}_{\ell,\ell+n}) \cdot v) \in A_{2\ell+1}(W), W := p_{1245}(V).$$

Consider the diagonal part $W \cap \{x = y\}$ first. It is contained in $\bigcup_{i\geq 0} T_i$, where $T_i = \{(\xi, x, x, \zeta) | \ell(\xi_x) = \ell(\zeta_x) = i$. The closure of T_0 is the diagonal $\Delta \cong X^{[\ell]} \times X \subset X^{[\ell],[1],[1],[\ell]}$ and is therefore irreducible of dimension $2\ell + 2$. Whereas for $i \geq 1$, the set T_i embeds into the irreducible variety $X^{[\ell-i]} \times (X_0^{[i]} \times_X X_0^{[i]})$ of dimension $2(\ell - i) + (i + 1) + (i + 1) - 2 = 2\ell$.

The off-diagonal part $W \cap \{x \neq y\}$ is empty if $\ell < n$. If $\ell \geq n$ it has precisely one irreducible component W' of maximal dimension $2\ell + 2$: it contains as a dense subset the image of the embedding

$$\{(\eta, \xi', \zeta') \in X^{[\ell-n]} \times X_0^{[n]} \times X_0^{[n]} | \eta, \xi' \text{ and } \zeta' \text{ are pairwise disjoint}\} \longrightarrow W,$$

$$(\sigma, \xi', \zeta') \mapsto (\sigma \cup \xi, \rho(\xi'), \rho(\zeta'), \sigma \cup \zeta').$$

Since the function $(\xi, x, y, \zeta) \mapsto \ell(\xi_x)$ is semicontinuous and is at least n on W', it follows that $\overline{W'} \cap \Delta$ is contained in $\bigcup_{\nu\geq n} T_n$. In particular, this intersection has

dimension $\leq 2\ell$. As we want to compute a cycle of degree $2\ell+1$, we may restrict our attention to the open part W' and may disregard the complement of W' in its closure.

$p_{1245} : p_{1245}^{-1}(W') \to W'$ is an isomorphism, which we use to identify W' and the off-diagonal part of V. Now W' parametrises four flat families of subschemes on X: besides the families Ξ and Z of fibrewise length ℓ, these are the families $\Xi \cap Z$ and $\Xi \cup Z$ of fibrewise length $\ell - n$ and $\ell + n$. The contribution of W' to w is the class

$$(-1)^n c_1(p_* \mathcal{I}_{\Xi/\Xi\cup Z}) \cdot [W'] \in A_{2\ell+1}(W').$$

Reversing the order of the operators $\mathfrak{q}'_{-n}$ and $\mathfrak{q}_n$ shows that the part of the cycle u inducing the commutator $[\mathfrak{q}'_{-n}, \mathfrak{q}_n]$, that is supported on W', is the class

$$(-1)^n \Big(c_1(p_* \mathcal{I}_{\Xi/\Xi\cup Z}) - c_1(p_* \mathcal{I}_{\Xi\cap Z/\Xi}) \Big) \cdot [W'].$$

Since the ideal sheaves $\mathcal{I}_{\Xi/\Xi\cup Z}$ and $\mathcal{I}_{\Xi/\Xi\cap Z}$ are isomorphic, this class is zero.

Thus we may fully concentrate on the contribution of the diagonal part Δ. (Also note that for the reversed order $\mathfrak{q}_n \mathfrak{q}'_{-n}$ any diagonal parts must be contained in $\bigcup_{\nu \geq n} T_\nu$ and are therefore too small and irrelevant.)

The complement of the open subset $T_0 \cong X^{[\ell]} \times X \setminus \Xi_\ell$ in Δ_0 has codimension ≥ 2. Locally near $p_{1245}^{-1}(T_0)$ there are isomorphisms between $X^{[\ell+n,\ell]}$ and $X^{[\ell]} \times X^{[n]}$, and similarly between $Q^{[\ell+n,\ell]}$ and $X^{[\ell]} \times X_0^{[n]}$. Hence if $\bar{w} \in A_1(X)$ is the intersection cycle for the special case $\ell = 0$, then the general cycle is simply given by $w = [X^{[\ell]}] \times \bar{w} \in A_{2\ell+1}(X^{[\ell]} \times X)$. But that was all we had to prove: a cycle of this form induces the linear map

$$\alpha \otimes \beta \otimes y \mapsto \int_{\bar{w}} \alpha\beta \cdot y, \qquad \alpha, \beta \in H^*(X; \mathbb{Q}), y \in \mathbb{H}.$$

□

Corollary 3.15 — *For all positive integers n one has*

$$\mathfrak{q}'_n(\alpha) = n\mathfrak{L}_n(\alpha) + n\mathfrak{q}_n(K_n\alpha).$$

Proof. Use the same argument as in the first paragraph of the proof of the main theorem after Corollary 3.12. □

To finish the proof of Theorem 3.11 it remains to show:

Proposition 3.16 — *For all positive integers n the rational divisor defined by Proposition 3.14 is given by*

$$K_n = \frac{n-1}{2} K,$$

where K is the canonical class of the surface X.

This will be done in the next section.

3.4 The vertex operator, completion of the proof

Definition 3.17 — Let $\gamma \in H^*(X)$ be an element which is of even degree though not necessarily homogeneous, and let t be a formal parameter. Define operators $S_m(\gamma)$, $m \geq 0$, by

$$S(\gamma, t) := \sum_{m \geq 0} S_m(\gamma) t^m := \exp\left(\sum_{n>0} \frac{(-1)^{n-1}}{n} \mathfrak{q}_n(\gamma) \cdot t^n \right).$$

Since γ is of even degree by assumption, any two operators $\mathfrak{q}_n(\gamma)$ and $\mathfrak{q}_{n'}(\gamma)$ commute in the ordinary, i.e. 'ungraded' sense. In particular, there is no ambiguity in the meaning of the expression on the right hand side in the definition.

The geometric meaning of the operators S_m is explained by the following theorem: let C be a smooth curve in X. There is an induced closed embedding $\mathcal{S}^n C = C^{[n]} \to X^{[n]}$. Let $[C] \in H^*(X)$ and $[C^{[n]}] \in H^*(X^{[n]})$ be the corresponding cohomology classes, i.e., the Poincaré dual classes of the fundamental classes of these varieties.

Theorem 3.18 (Nakajima, Grojnowski) — *The following relation holds for all non-negative integers n:*

$$[C^{[n]}] = S_n([C]) \cdot \mathbf{1}.$$

For proofs see [22] and [13]. □

Lemma 3.19 — *Let $\gamma \in H^*(X)$ be an element of even degree. Then*

$$S'(\gamma, t) = S(\gamma, t) \cdot \sum_{n>0} (-1)^{n-1} t^n \left\{ \mathfrak{L}_n(\gamma) + \mathfrak{q}_n\Big(\gamma K_n + \gamma^2 \frac{n-1}{2}\Big) \right\}.$$

Proof. Assume first that $\mathfrak{a}$ is an operator of even degree, and that $[\mathfrak{d}, \mathfrak{a}]$ commutes with $\mathfrak{a}$. Then

$$\begin{aligned}
\left(\sum_{n=0}^{\infty} \frac{\mathfrak{a}^n}{n!} \right)' &= \sum_{n=1}^{\infty} \frac{1}{n!} \sum_{i=1}^{n} \mathfrak{a}^{i-1} \cdot \mathfrak{a}' \cdot \mathfrak{a}^{n-i} \\
&= \sum_{n=1}^{\infty} \frac{1}{n!} \cdot \left\{ n \mathfrak{a}^{n-1} \mathfrak{a}' + \sum_{i=1}^{n} \mathfrak{a}^{n-2} \cdot (n-i) \cdot [\mathfrak{a}', \mathfrak{a}] \right\} \\
&= \sum_{n=0}^{\infty} \frac{\mathfrak{a}^n}{n!} \cdot \mathfrak{a}' + \sum_{n=1}^{\infty} \frac{\mathfrak{a}^{n-2}}{n!} \binom{n}{2} [\mathfrak{a}', \mathfrak{a}] \\
&= \exp(\mathfrak{a}) \cdot \left\{ \mathfrak{a}' + \frac{1}{2} [\mathfrak{a}', \mathfrak{a}] \right\}.
\end{aligned}$$

Next, let $\mathfrak{a}_\nu$ be a family of commuting operators of even degree such that any $[\mathfrak{a}'_\nu, \mathfrak{a}_\mu]$ commutes with every $\mathfrak{a}_\xi$. Then it follows from Step 1 and

$$[\mathfrak{a}'_\mu, \exp(\mathfrak{a}_\nu)] = \exp(\mathfrak{a}_\nu) \cdot [\mathfrak{a}'_\mu, \mathfrak{a}_\nu]$$

that

$$\left(\exp\Big(\sum_\nu \mathfrak{a}_\nu\Big)\right)' = \exp\Big(\sum_\nu \mathfrak{a}_\nu\Big)\cdot\left\{\sum_\nu \mathfrak{a}'_\nu + \frac{1}{2}\sum_{\nu,\mu}[\mathfrak{a}'_\nu,\mathfrak{a}_\mu]\right\}.$$

Now apply this formula to the family $\mathfrak{a}_\nu = \frac{(-1)^{\nu-1}}{\nu}\mathfrak{q}_\nu(\gamma)t^\nu$ and use our previous results $\mathfrak{a}'_\nu = (-1)^{\nu-1}t^\nu(\mathfrak{L}_n(\gamma)+\mathfrak{q}_\nu(K_\nu\gamma))$ and $[\mathfrak{a}'_\nu,\mathfrak{a}_\mu] = -(-t)^{\nu+\mu}\mathfrak{q}_{\nu+\mu}(\gamma^2)$. One gets $S'(\gamma,t) = S(\gamma,t)\cdot(*)$ with

$$\begin{aligned}(*) &= \sum_{n>0}(-1)^{n-1}t^n\big(\mathfrak{L}_n(\gamma)+\mathfrak{q}_n(K_n\gamma)\big) - \frac{1}{2}\sum_{\nu,\mu>0}(-t)^{\nu+\mu}\mathfrak{q}_{\nu+\mu}(\gamma^2)\\ &= \sum_{n>0}(-1)^{n-1}t^n\cdot\left\{\mathfrak{L}_n(\gamma)+\mathfrak{q}_n(K_n\gamma+\frac{1}{2}N_n\gamma^2)\right\}\end{aligned}$$

where N_n is the number of pairs of positive integers ν and μ that add up to n, i.e., $N_n = n-1$. □

Let $C \subset X$ be a smooth projective curve. The boundary $\partial X^{[n]}$ intersects $C^{[n]}$ generically transversely in the boundary $\partial C^{[n]}$ of $C^{[n]}$, i.e. in the set of all tuples with multiple points. The subvarieties $X_0^{[n]}$ and $\partial C^{[n]}$ have complementary dimensions $n+1$ and $n-1$ in $X^{[n]}$ and we may compute the intersection number

$$I := \int_{X^{[n]}}[X_0^{[n]}]\cup[\partial C^{[n]}].$$

We will do this first using our algorithmic language, and afterwards using a geometric argument. The comparison of the two results will lead to the identification of the divisors K_n.

Lemma 3.20 — $[X_0^{[n]}] = \mathfrak{q}_n(1_X)\cdot\mathbf{1}$ *and* $[\partial C^{[n]}] = -2\cdot S'_n([C])\cdot\mathbf{1}$.

Proof. The first assertion follows from the definition of the operators $\mathfrak{q}_n$. By Nakajima's Theorem, $S_n([C])\cdot\mathbf{1}$ is the class of the submanifold $C^{[n]} \subset X^{[n]}$, and hence according to Lemma 3.8:

$$S'_n([C])\cdot\mathbf{1} = \mathfrak{d}\cdot S_n([C])\cdot\mathbf{1} = -\frac{1}{2}[\partial X^{[n]}]\cdot[C^{[n]}] = -\frac{1}{2}[\partial C^{[n]}].$$

□

Lemma 3.21 —

$$\int_{X^{[n]}}(\mathfrak{q}_n(1_X)\cdot\mathbf{1})\cdot(S'_n([C])\cdot\mathbf{1}) = \int_X\left\{nK_nC + \binom{n}{2}C^2\right\}.$$

Proof. Indeed,

$$\begin{aligned}\int_{X^{[n]}} (\mathfrak{q}_n(1_X)\cdot\mathbf{1})\cdot(S'_n([C])\cdot\mathbf{1}) &= (-1)^n\int_{X^{[0]}}\mathfrak{q}_{-n}(1_X)S'_n([C])\cdot\mathbf{1}\\ &= (-1)^n\int_{X^{[0]}}[\,\mathfrak{q}_{-n}(1_X),S'_n([C])\,]\cdot\mathbf{1},\end{aligned}$$

since $\mathfrak{q}_{-n}(1_X)\cdot\mathbf{1}=0$. Now $\mathfrak{q}_{-n}$ commutes with any product $\mathfrak{q}_{i_1}\cdot\ldots\cdot\mathfrak{q}_{i_s}$ if $s\geq 2$, $i_j>0$ and $\sum_j i_j = n$. Thus the only summand in S'_n that contributes to the commutator with $\mathfrak{q}_{-n}$ is $(-1)^{n-1}\mathfrak{q}_n(C(K_n+C(n-1)/2))$. Hence

$$[\mathfrak{q}_{-n}(1_X),S'_n([C])] = (-1)^n n\int_X C\left(K_n+\frac{n-1}{2}C\right)\cdot\mathrm{id}_{\mathbb{H}}$$

This proves the lemma. □

Next, we give the geometric computation of I:

Lemma 3.22 —

$$\int_{X^{[n]}}[X_0^{[n]}]\cdot[\partial C^{[n]}] = -n(n-1)\cdot C(C+K).$$

Proof. We have $[X_0^{[n]}]\cdot[\partial C^{[n]}] = [\partial X^{[n]}]\cdot([X_0^{[n]}]\cdot[C^{[n]}])$. The intersection of $X^{[n]}$ and $C^{[n]}$ is transversal and is equal to the image of the closed immersion $\Delta: C\to C^{[n]}$ sending a point c to the unique subscheme of C of length n that is supported in c. Thus

$$I = \deg(\mathcal{O}_{X^{[n]}}(\partial X^{[n]})|_{\Delta(C)} = \deg(\mathcal{O}_{C^{[n]}}(\partial C^{[n]})|_{\Delta(C)}.$$

The embedding Δ factors through the diagonal embedding $C\to C^n$ and the quotient map $\pi: C^n\to C^{[n]}$. Moreover, if $\mathrm{pr}_{ij}: C^n\to C^2$ denotes the projection to the product of the i-th and j-th factor,

$$\pi^*(\mathcal{O}_{C^{[n]}}(\partial C^{[n]}))\cong\left(\bigotimes_{i<j}^{n} pr_{ij}^*\mathcal{O}_{C\times C}(\Delta_C)\right)^{\otimes 2}.$$

From this we conclude:

$$\begin{aligned}I=\deg(\Delta^*\mathcal{O}_{C^{[n]}}(\partial C^{[n]})) &= 2\cdot\binom{n}{2}\deg(\mathcal{O}_{C\times C}(\Delta_C)|_{\Delta_C}))\\ &= -n(n-1)\cdot C(C+K).\end{aligned}$$

□

Proof of Proposition 3.16. From Lemma 3.20 and Lemma 3.21 we conclude

$$I = (-2)\cdot C(nK_n+\binom{n}{2}C).$$

Comparison with Lemma 3.22 shows that $K_n=\frac{n-1}{2}K$. □

This finishes the proof of Theorem 3.11.

4 Towards the ring structure of $\mathbb{H}$

4.1 Tautological sheaves

There is a natural way to associate to a given vector bundle on X a series of tautological' vector bundles on the Hilbert schemes $X^{[n]}$, $n \geq 0$. The Chern classes of the tautological bundles may be grouped together to form operators on $\mathbb{H}$.

Consider the standard diagram

$$\begin{array}{ccccc} \Xi_n & \subset & X^{[n]} \times X & \stackrel{q}{\longrightarrow} & X \\ & & \downarrow{\scriptstyle p} & & \\ & & X^{[n]} & & \end{array}$$

Let F be a locally free sheaf on X. For each $n \geq 0$ the associated *tautological bundle* on $X^{[n]}$ is definded as

$$F^{[n]} := p_*(\mathcal{O}_{\Xi_n} \otimes q^* F).$$

Since p is a flat finite morphism of degree n, $F^{[n]}$ is locally free with

$$\mathrm{rk}(F^{[n]}) = n \cdot \mathrm{rk}(F).$$

Note that $F^{[0]} = 0$ and $F^{[1]} = F$.

Furthermore, if $0 \to F_1 \to F \to F_2 \to 0$ is a short exact sequence of locally free sheaves on X, the corresponding sequence $0 \to F_1^{[n]} \to F^{[n]} \to F_2^{[n]} \to 0$ is again exact. Hence sending the class $[F]$ of a locally free sheaf F to $[F^{[n]}]$ gives a group homomorphism

$$-^{[n]} : K(X) \longrightarrow K(X^{[n]}).$$

Definition 4.1 — Let u be a class in $K(X)$. Define operators

$$\mathfrak{c}(u) \in \mathrm{End}(\mathbb{H}) \quad \text{and} \quad \mathfrak{ch}(u) \in \mathrm{End}(\mathbb{H})$$

as follows: For each $n \geq 0$, the action on $H^*(X^{[n]}; \mathbb{Q})$ is given by multiplication with the total Chern class $c(u^{[n]})$ and the Chern character $ch(u^{[n]})$, respectively.

Let

$$\mathfrak{c}(u) = \sum_{k \geq 0} \mathfrak{c}_k(u) \quad \text{and} \quad \mathfrak{ch}(u) = \sum_{k \geq 0} \mathfrak{ch}_k(u)$$

be the decompositions into homogeneous components of bidegree $(0, 2k)$. Since all of these operators are of even degree and only act 'vertically' on $\mathbb{H}$ by multiplication, they commute with each other and in particular with the previously defined boundary operator $\mathfrak{d} = \mathfrak{c}_1(\mathcal{O}_X)$.

Moreover, we have

$$\mathfrak{c}(u+v) = \mathfrak{c}(u) \cdot \mathfrak{c}(v) \quad \text{and} \quad \mathfrak{ch}(u+v) = \mathfrak{ch}(u) + \mathfrak{ch}(v)$$

for all $u, v \in K(X)$.

Theorem 4.2 — *Let u be a class in $K(X)$ of rank r and let $\alpha \in H^*(X)$. Then*

$$[\mathfrak{ch}(u), \mathfrak{q}_1(\alpha)] = \exp(\operatorname{ad}\mathfrak{d})(\mathfrak{q}_1(ch(u)\alpha)),$$

or, more explicitely,

$$[\mathfrak{ch}_n(u), \mathfrak{q}_1(\alpha)] = \sum_{\nu=0}^{n} \frac{1}{\nu!}\mathfrak{q}_1^{(\nu)}(ch_{n-\nu}(u)\alpha).$$

Similarly,

$$\mathfrak{c}(u) \cdot \mathfrak{q}_1(\alpha) \cdot \mathfrak{c}(u)^{-1} = \sum_{\nu,k\geq 0} \binom{r-k}{\nu} \mathfrak{q}_1^{(\nu)}(c_k(u)\alpha).$$

Proof. We may assume that u is the class of a locally free sheaf F. Recall the standard diagram for the incidence variety $X^{[\ell,\ell+1]}$:

$$\begin{array}{ccccc} X & \xleftarrow{\rho} & X^{[\ell,\ell+1]} & \xrightarrow{\psi} & X^{[\ell+1]} \\ & & \downarrow{\scriptstyle \varphi} & & \\ & & X^{[\ell]} & & \end{array}$$

The variety $X^{[\ell,\ell+1]}$ parametrises two families of subschemes of X. Their structure sheaves fit into an exact sequence

$$0 \to \rho_X^*\mathcal{O}_{\Delta_X} \otimes p^*\mathcal{O}_{X^{[\ell,\ell+1]}}(-E) \to \psi_X^*(\mathcal{O}_{\Xi_{\ell+1}}) \to \varphi_X^*(\mathcal{O}_{\Xi_\ell}) \to 0,$$

where $p : X^{[\ell,\ell+1]} \times X \to X^{[\ell,\ell+1]}$ is the projection and E is the exceptional divisor. Applying the functor $p_*(\,\cdot\, \otimes q^*F)$ to this exact sequence yields

$$0 \to \rho^*F \otimes \mathcal{O}_{X^{[\ell,\ell+1]}}(-E) \to \psi^*F^{[\ell+1]} \to \varphi^*F^{[\ell]} \to 0. \tag{11}$$

Let $\lambda = c_1(\mathcal{O}_{X^{[\ell,\ell+1]}}(-E))$. Then

$$\psi^*ch(F^{[\ell+1]}) = \varphi^*ch(F^{[\ell]}) + \rho^*ch(F) \cdot \exp(\lambda)$$

and

$$\psi^*c(F^{[\ell+1]}) = \varphi^*c(F) \cdot \sum_{\nu,k\geq 0} \binom{r-k}{\nu} \lambda^\nu \rho^*c_k(F).$$

It follows for any $x \in H^*(X^{[\ell]}; \mathbb{Q})$:

$$\begin{aligned} \mathfrak{ch}(F)\mathfrak{q}_1(\alpha)(x) &= ch(F^{[\ell+1]}) \cdot PD^{-1}\psi_*([X^{[\ell,\ell+1]}] \cap \rho^*(\alpha)\varphi^*(x)) \\ &= PD^{-1}\psi_*([X^{[\ell,\ell+1]}] \cap \psi^*(ch(F^{[\ell+1]}))\rho^*(\alpha)\varphi^*(x)) \\ &= PD^{-1}\psi_*([X^{[\ell,\ell+1]}] \cap \rho^*(\alpha)\varphi^*(ch(F^{[\ell]})x)) \\ &\quad + \sum_{\nu\geq 0} \frac{1}{\nu!} PD^{-1}\psi_*(\lambda^\nu \cdot [X^{[\ell,\ell+1]}] \cap \rho^*(ch(F)\alpha)\varphi^*(x)) \\ &= \mathfrak{q}_1(\alpha)(\mathfrak{ch}(F)x) + \sum_{\nu\geq 0} \frac{1}{\nu!}\mathfrak{q}^{(\nu)}\mathfrak{q}_1(ch(F)\alpha)(x). \end{aligned}$$

Here we used Lemma 3.10 which says that the cycle $\chi^\nu \cdot [X^{[\ell,\ell+1]}]$ induces the operator $\mathfrak{q}_1^{(\nu)}$. This is the equation for the Chern character. The equation for the total Chern class is proved analogously. □

Remark 4.3 — The sequence (11) was used by Ellingsrud in a recursive method to determine Chern classes and Segre classes of tautological bundles (unpublished, but see [25],[4]). He expresses the classes $(\varphi, \rho)_* c(E)$ in terms of the Segre classes of the universal family $\Xi_{[n]} \subset X \times X^{[n]}$. Thus one needs to control the behaviour of these Segre classes under the induction procedure. This method yields qualitative results on the *structure* of certain classes and integrals, but all attempts to get numbers have ended so far in unsurmountable combinatorical difficulties. □

Remark 4.4 — The results of the present and the previous section provide an algorithmic description of the multiplicative action of the subalgebra $\mathcal{A} \subset \mathbb{H}$ which is generated by the Chern classes of all tautological bundles: The elements $\mathfrak{q}_{i_1}(\alpha_1) \cdot \ldots \mathfrak{q}_{i_s}(\alpha_s) \cdot \mathbf{1}$ generate $\mathbb{H}$ as a $\mathbb{Q}$-vector space. By Corollary 3.12, each such element can be written as a linear combination of expression $w \cdot \mathbf{1}$, where w is a word in an alphabet consisting of $\mathfrak{d}$ and operators $\mathfrak{q}_1(\alpha)$, $\alpha \in H^*(X;\mathbb{Q})$. By Theorem 4.2 the commutator of $\mathfrak{ch}(F)$ with any of these is again a word in this alphabet. And finally, Theorem 3.11 shows how such a word can be expressed in terms of the basic operators $\mathfrak{q}_n$. Admittedly, without a further understanding of the algebraic structure this description is useful for computations in $H^*(X^{[\ell]};\mathbb{Q})$ only for small values of ℓ or if one implements it in some computer algebra system.

4.2 The line bundle case

The results of the previous section suffice to compute the Chern classes of the tautological bundles $L^{[n]}$ associated to a line bundle L in terms of the basic operators.

Theorem 4.5 — *Let L be a line bundle on X. Then*

$$\sum_{n\geq 0} c(L^{[n]}) = \exp\left(\sum_{m\geq 1} \frac{(-1)^{m-1}}{m} \mathfrak{q}_m(c(L))\right) \cdot \mathbf{1}.$$

Remark 4.6 — Expanding the term on the right hand side, one realises that the cohomological degree of any summand contained in $H^*(X^{[n]};\mathbb{Q})$ is $\leq 2n$, and, moreover, the maximal degree $2n$ can only be attained if the arguments of all operators $\mathfrak{q}_\nu$ involved have degree 2. In other words, considering elements of top degree only, the equation of the theorem specialises to

$$\sum_{n\geq 0} c_n(L^{[n]}) = \exp\left(\sum_{m\geq 1} \frac{(-1)^{m-1}}{m} \mathfrak{q}_m(c_1(L))\right) \cdot \mathbf{1}. \qquad (12)$$

This is Nakajima's result 3.18: for suppose $C \subset X$ is a smooth curve and $L = \mathcal{O}_X(C)$. If $\xi \in X^{[n]}$, the natural homomorphism $\mathcal{O}_X \to \mathcal{O}_\xi(C)$ vanishes if and only if $\xi \subset C$. Hence the vanishing locus of the global vector bundle homomorphism

$$\mathcal{O}_{X^{[n]}} \longrightarrow (\mathcal{O}_X(C))^{[n]} = L^{[n]}$$

is the subvariety $C^{[n]}$. Therefore $[C^{[n]}] = c_n(L^{[n]})$. Inserting this into (12), we recover Nakajima's formula 3.18

$$\sum_{n\geq 0}[C^{[n]}] = \exp\left(\sum_{m\geq 1}\frac{(-1)^{m-1}}{m}\mathfrak{q}_m([C])\right)\cdot \mathbf{1}$$

Based on this observation, the theorem was conjectured by L. Göttsche in a letter to G. Ellingsrud and the author.

Proof of the theorem. We shall give two variants of the proof which differ slightly in flavour. Observe that the left hand side in the theorem equals

$$\sum_{n\geq 0} c(L^{[n]}) = \mathfrak{c}(L)\cdot\sum_{n\geq 0} 1_{X^{[n]}} = \mathfrak{c}(L)\cdot\exp(\mathfrak{q}_1(1_X))\cdot\mathbf{1}.$$

Variant 1. Applying Theorem 4.2 with $F = L$ and $r = 1$ we get

$$\mathfrak{c}(L)\cdot\mathfrak{q}_1(1_X)\cdot\mathfrak{c}(L)^{-1} = \{\mathfrak{q}_1(1_X + c_1(L)) + \mathfrak{q}_1'(1_X)\}.$$

Hence

$$\begin{aligned}\mathfrak{c}(L)\cdot\exp(\mathfrak{q}_1(1_X))\cdot\mathbf{1} &= \mathfrak{c}(L)\cdot\exp(\mathfrak{q}_1(1_X))\cdot\mathfrak{c}(L)^{-1}\cdot\mathbf{1}\\ &= \exp(\mathfrak{c}(L)\cdot\mathfrak{q}_1(1_X)\cdot\mathfrak{c}(L)^{-1})\cdot\mathbf{1}\\ &= \exp(\mathfrak{q}_1(c(L)) + \mathfrak{q}_1'(1_X))\cdot\mathbf{1}\\ &= \sum_{n\geq 0}\frac{1}{n!}(\mathfrak{q}_1(c(L)) + \mathfrak{q}_1'(1_X))^n\cdot\mathbf{1}.\end{aligned}$$

Expanding the right hand side yields summands which are words in the two symbols $\mathfrak{q}_1(c(L))$ and $\mathfrak{q}_1'(1_X)$. Moving all factors $\mathfrak{q}_1'(1_X)$ within a given word as far to the right as possible using the commutation relations of the main theorem we can write

$$\sum_{n\geq 0}\frac{1}{n!}(\mathfrak{q}_1(c(L)) + \mathfrak{q}_1'(1_X))^n = \mathfrak{A}\cdot\mathbf{1} + \mathfrak{B}\cdot\mathfrak{q}_1'(1_X)\cdot\mathbf{1} = \mathfrak{A}\cdot\mathbf{1},$$

where $\mathfrak{A}$ is a sum of expressions of the form

$$\nu_1!\cdot\ldots\cdot\nu_s!\cdot\frac{(-1)^{\nu_1-1}\mathfrak{q}_{\nu_1}(c(L))}{\nu_1}\cdots\frac{(-1)^{\nu_s-1}\mathfrak{q}_{\nu_s}(c(L))}{\nu_s}.$$

Let $\alpha = (1^{\alpha_1}2^{\alpha_2}3^{\alpha_3}\dots)$ denote a partition and let $|\alpha| := \sum_{i\geq 1} i\alpha_i$, and $\alpha! := \prod_i (i!)^{\alpha_i}$. We get

$$\sum_{n\geq 0}\frac{1}{n!}(\mathfrak{q}_1(c(L)) + \mathfrak{q}_1'(1_X))^n \cdot \mathbf{1} = \sum_{\alpha} N_\alpha \frac{\alpha!}{|\alpha|!}\prod_{i\geq 1}\left(\frac{(-1)^{i-1}\mathfrak{q}_i(c(L))}{i}\right)^{\alpha_i}\cdot \mathbf{1}, \quad (13)$$

where the natural number N_α counts how often the operator

$$\alpha!\prod_{i\geq 1}\left(\frac{(-1)^{i-1}\mathfrak{q}_i(c(L))}{i}\right)^{\alpha_i}$$

arises from a word in $\mathfrak{q}_1'(1_X)$ and $\mathfrak{q}_1(c(L))$ of length $|\alpha|$. It is not difficult to see that N_α equals the number of possibilities to partition a set of $|\alpha|$ elements into subsets in such a way that there are α_i subsets of cardinality i. Hence

$$N_\alpha := \frac{1}{\alpha_1!\alpha_2!\cdots}\cdot\frac{|\alpha|!}{\alpha!}.$$

Inserting this into equation (13) above one gets

$$\begin{aligned}
\sum_{n\geq 0}\frac{1}{n!}(\mathfrak{q}_1(c(L)) + \mathfrak{q}_1'(1_X))^n \cdot \mathbf{1} &= \sum_{\alpha}\prod_{i\geq 1}\frac{1}{\alpha_i!}\left(\frac{(-1)^{i-1}\mathfrak{q}_i(c(L))}{i}\right)^{\alpha_i}\cdot\mathbf{1} \\
&= \prod_{i\geq 1}\sum_{\alpha_i\geq 0}\frac{1}{\alpha_i!}\left(\frac{(-1)^{i-1}\mathfrak{q}_i(c(L))}{i}\right)^{\alpha_i}\cdot\mathbf{1} \\
&= \prod_{i\geq 1}\exp\left(\frac{(-1)^{i-1}\mathfrak{q}_i(c(L))}{i}\right)\cdot\mathbf{1} \\
&= \exp\left(\sum_{i\geq 1}\frac{(-1)^{i-1}}{i}\mathfrak{q}_i(c(L))\right)\cdot\mathbf{1}.
\end{aligned}$$

Variant 2. Starting again from the sequence

$$\mathfrak{c}(L)\cdot\mathfrak{q}_1(1_X) = \{\mathfrak{q}_1(1_X + c_1(L)) + \mathfrak{q}_1'(1_X)\}\cdot\mathfrak{c}(L),$$

we multiply by $\frac{1}{n!}\mathfrak{q}_1(1_X)^n t^n$ from the right and sum up over all $n \geq 0$:

$$\begin{aligned}
\frac{d}{dt}\left(\mathfrak{c}(L)\cdot\sum_{n\geq 0}\frac{1}{n!}\mathfrak{q}_1(1_X)^n t^n\right)\cdot\mathbf{1} &= \mathfrak{c}(L)\cdot\sum_{n\geq 0}\frac{1}{n!}\mathfrak{q}_1(1_X)^{n+1}t^n\cdot\mathbf{1} \\
&= \{\mathfrak{q}_1(1_X + c_1(L)) + \mathfrak{q}_1'(1_X)\}\cdot\left(\mathfrak{c}(L)\cdot\sum_{n\geq 0}\frac{1}{n!}\mathfrak{q}_1(1_X)^n t^n\right)\cdot\mathbf{1}.
\end{aligned}$$

This means that the series

$$\sum_{n\geq 0} c(L^{[n]})t^n = \mathfrak{c}(L)\cdot\exp(\mathfrak{q}_1(1_X)t)\cdot\mathbf{1}$$

satisfies the linear differential equation

$$\frac{d}{dt}\mathfrak{X} = \{\mathfrak{q}_1(1_X + c_1(L)) + \mathfrak{q}_1'(1_X)\} \cdot \mathfrak{X} \tag{14}$$

with initial condition

$$\mathfrak{X}(0) = \mathbf{1}. \tag{15}$$

On the other hand, consider the operator

$$S(c(L), t) = \exp\left(\sum_{m\geq 1} \frac{(-1)^{m-1}}{m} \mathfrak{q}_m(c(L))t^m\right).$$

We find

$$\frac{d}{dt}S(c(L), t) = S(c(L), t) \cdot \left(\sum_{m\geq 0} (-1)^m \mathfrak{q}_{m+1} t^m\right),$$

and

$$\begin{aligned}
&\Big[\{\mathfrak{q}_1(1_X + c_1(L)) + \mathfrak{q}_1'(1_X)\}, S(c(L), t)\Big] \\
&\qquad = \quad S(c(L), t) \cdot \left(\sum_{m\geq 1} \frac{(-1)^{m-1}}{m} \Big[\mathfrak{q}_1'(1_X), \mathfrak{q}_m(c(L))\Big] t^m\right) \\
&\qquad = \quad S(c(L), t) \cdot \left(\sum_{m\geq 1} (-1)^m \mathfrak{q}_{m+1}(c(L))t^m\right).
\end{aligned}$$

This shows

$$\begin{aligned}
&\{\mathfrak{q}_1(1_X + c_1(L)) + \mathfrak{q}_1'(1_X)\} \cdot S(c(L), t) \cdot \mathbf{1} \\
&\qquad = \quad S(c(L), t) \cdot \left(\sum_{m\geq 1} (-1)^m \mathfrak{q}_{m+1}(c(L))t^m\right) \cdot \mathbf{1} \\
&\qquad\quad + S(c(L), t) \cdot \mathfrak{q}_1(c(L)) \cdot \mathbf{1} \\
&\qquad = \quad S(c(L), t) \cdot \left(\sum_{m\geq 0} (-1)^m \mathfrak{q}_{m+1}(c(L))t^m\right) \cdot \mathbf{1}
\end{aligned}$$

Hence $S(c(L), t) \cdot \mathbf{1}$ satisfies the system (14) and (15) as well and therefore equals $\mathfrak{c}(L) \cdot \exp(\mathfrak{q}_1(1_X)t) \cdot \mathbf{1}$. This proves the theorem. □

4.3 Top Segre classes

The following problem was posed by Donaldson in connection with the computation of instanton invariants: let n be an integer ≥ 1, and consider a linear system $|H|$ of dimension $3n-2$ inducing a map $X - - \to \mathbb{P}^{3n-2}$. A zero-dimensional subscheme $\xi \in X^{[n]}$ does not impose independent conditions on the linear system $|H|$ if the natural homomorphism

$$H^0(\mathbb{P}^{3n-2}, \mathcal{O}_{\mathbb{P}}(1)) \longrightarrow H^0(\xi, \mathcal{O}_\xi(H))$$

fails to be surjective. The subscheme of all such $\xi \in X^{[n]}$ has virtual dimension zero, and its class is given by $c_{2n}(W^\vee)$, where W is the virtual vector bundle

$$H^0(\mathbb{P}^{3n-2}, \mathcal{O}_{\mathbb{P}}(H)) \otimes \mathcal{O}_{X^{[n]}} - \mathcal{O}(H)^{[n]}.$$

Thus the number of those ξ that impose dependent conditions is given by

$$N_n := \int_{X^{[n]}} c_{2n}(-\mathcal{O}(H)^{[n]}) = \int_{X^{[n]}} \mathfrak{c}(-\mathcal{O}(H)) \cdot \frac{\mathfrak{q}_1(1_X)^n}{n!} \cdot \mathbf{1}.$$

More explicitly, N_1 is the degree of the linear system, N_2 is the number of double points, N_3 is the number of trisecants to a surface in $\mathbb{P}^7$ and N_4 is the number of quadrupels of points on a surface in $\mathbb{P}^{10}$ that span a plane.

Problem: *Express N_n in terms of intrinsic invariants of X such as the degree $d := H.H$, the intersection $\kappa := H.K$ and $\kappa := K.K$ and the topological Euler characteristic χ.*

Note that even the fact that such an expression in terms of the given invariants exists is not evident *a priori*. This has been proved by Tikhomirov [25]. It also follows immediately from our approach.

Using our algorithm, we can attack this problem as follows. Theorem 4.2 yields for $F = -\mathcal{O}(H)$ and $r = -1$ the formula:

$$\begin{aligned}
\mathfrak{c}(-\mathcal{O}(H)) \cdot \mathfrak{q}_1(1_X) \cdot \mathfrak{c}(-\mathcal{O}(H))^{-1} &= \sum_{\nu,k\geq 0} \binom{-1-k}{\nu} \mathfrak{q}_1^{(\nu)}(c_k(-H)) \\
&= \sum_{\nu\geq 0} (-1)^\nu \mathfrak{q}_1^{(\nu)} \left(\sum_{k=0}^{2} \binom{\nu+k}{k} (-H)^k \right) \\
&= \sum_{\nu\geq 0} (-1)^\nu \mathfrak{q}_1^{(\nu)}((1-H+H^2)^{\nu+1}).
\end{aligned}$$

Denote the operator sum on the right hand side by $\mathfrak{N}$. It follows as in the proof of Theorem 4.5 that $\mathfrak{c}(-\mathcal{O}(H)) \cdot \exp(\mathfrak{q}_1(1_X)t) \cdot \mathbf{1}$ satisfies the following differential equation and initial value condition:

$$\frac{d}{dt}\mathfrak{X} = \mathfrak{N}\mathfrak{X} \quad \text{and} \quad \mathfrak{X}(0) = \mathbf{1}.$$

As long as no explicit generating function is available we must be content with the following semi-explicit solution to Donaldson's problem:

$$N_n = \frac{1}{n!}\int_{X^{[n]}} \mathfrak{N}^n \cdot \mathbf{1}.$$

Note that the right hand side is more than a mere reformulation of the definition of N_n: the expression on the right hand side is a linear combination of words in the operators $\mathfrak{q}_1$ and $\mathfrak{d}$ and can be explicitly evaluated by applying the rules of Theorem 3.11.

Example 4.7 — As a special case, let us compute N_2. This is the number of secant lines to an embedded surface in $\mathbb{P}^5$ that pass through a fixed but general point $x \in \mathbb{P}^5$. Hence we should find Severi's double point formula [23] (see also [2]). We have

$$2 \cdot N_2 = \int_{X^{[2]}} \left(\sum_{n\geq 0} (-1)^n \mathfrak{q}_1^{(n)}(1-(n+1)H + \binom{n+2}{2} H^2) \right)^2 \cdot \mathbf{1}.$$

Since $\mathfrak{q}_1^{(n)}(\alpha)\cdot \mathbf{1} = 0$ for all $n > 0$ and for all α, and for degree reasons the integral reduces to

$$2 \cdot N_2 = \int_{X^{[2]}} I$$

with

$$\begin{aligned} I &= \Big(\mathfrak{q}_1^2(H^2 \otimes H^2) + \mathfrak{q}_1'\mathfrak{q}_1(2H^2 \otimes H + 3H \otimes H^2) \\ &\quad + \mathfrak{q}_1''\mathfrak{q}_1(6H^2 \otimes 1 + 3H \otimes H + 1 \otimes H^2) \\ &\quad + \mathfrak{q}_1'''\mathfrak{q}_1(4H \otimes 1 + 1 \otimes H) + \mathfrak{q}_1''''\mathfrak{q}_1(1 \otimes 1)\Big) \cdot \mathbf{1}. \end{aligned}$$

Since $\mathfrak{q}_1'(\alpha)\cdot \mathbf{1} = 0$, one easily sees that

$$\mathfrak{q}_1^{(n)}\mathfrak{q}_1(\alpha \otimes \beta)\cdot \mathbf{1} = -\mathfrak{q}_2^{(n-1)}(\alpha\beta)\cdot \mathbf{1}.$$

This yields

$$I = \big(\mathfrak{q}_1^2(H^2 \otimes H^2) - \mathfrak{q}_2'(10H^2) - \mathfrak{q}_2''(5H) - \mathfrak{q}_2'''(1)\big)\cdot \mathbf{1}.$$

The term $\mathfrak{q}_2$ vanishes for degree reasons. Moreover, for $n \geq 2$ we have

$$\begin{aligned} \mathfrak{q}_2^{(n)}(\alpha)\cdot \mathbf{1} &= \big(\mathfrak{q}_1^2(\delta_*(\alpha)) + \mathfrak{q}_2(K\alpha)\big)^{(n-1)}\cdot \mathbf{1} \\ &= \big(-\mathfrak{q}_2^{(n-2)}(c_2(X)\alpha) + \mathfrak{q}_2^{(n-1)}(K\alpha)\big)\cdot \mathbf{1}. \end{aligned}$$

(Note that the composite map $H^*(X) \xrightarrow{\delta} H^*(X)\otimes H^*(X) \xrightarrow{\cup} H^*(X)$ is the multiplication with the self intersection of the diagonal, i.e. the second Chern class $c_2(X)$ of X.) Applying this to I, we find

$$\begin{aligned} I - \mathfrak{q}_1^2(H^2\otimes H^2)\cdot \mathbf{1} &= -\Big(10\mathfrak{q}_2'(H^2) + 5\mathfrak{q}_2''(H) + \mathfrak{q}_2'''(1)\Big)\cdot \mathbf{1} \\ &= -\Big(\mathfrak{q}_2(10H^2 - c_2(X)) + \mathfrak{q}_2''(5H+K)\Big)\cdot \mathbf{1} \\ &= -\mathfrak{q}_2'(10H^2 - c_2(X) + 5HK + K^2)\cdot \mathbf{1}. \end{aligned}$$

This yields:

$$I = \mathfrak{q}_1^2(H^2 \otimes H^2) + \delta_*(-10H^2 + c_2(X) - 5HK - K^2)) \cdot \mathbf{1},$$

and therefore

$$2 \cdot N_2 = \int_{X^{[2]}} I = d^2 - 10d - 5\pi - \kappa + \chi.$$

□

Obviously, for higher n, the practical calculation of N_n quickly becomes rather difficult. Already the case of N_3 surpassed my personal calculation skills. Using MAPLE, I computed the following expressions:

$$\begin{aligned} 3! \cdot N_3 &= d^3 - 30d^2 + 224d - 3d(5\pi + \kappa - \chi) \\ &\quad +192\pi + 56\kappa - 40\chi, \end{aligned}$$

$$\begin{aligned} 4! \cdot N_4 &= d^4 - 60d^3 + d^2(1196 - 30\pi + 6\chi - 6\kappa) \\ &\quad -d(7920 - 1068\pi + 220\chi - 284\kappa) + 3\chi^2 + 1944\chi - 6\chi\kappa \\ &\quad -30\chi\pi + 75\pi^2 + 3\kappa^2 + 30\kappa\pi - 9042\pi - 3300\kappa, \end{aligned}$$

$$\begin{aligned} 5! \cdot N_5 &= d^5 - 100d^4 + d^3(3740 + 10\chi - 50\pi - 10\kappa) \\ &\quad -d^2(62000 - 3420\pi + 700\chi - 860\kappa) + d(384384 + 15\chi^2 \\ &\quad +15960\chi - 30\chi\kappa - 150\pi\chi + 15\kappa^2 + 150\kappa\pi - 75610\pi \\ &\quad -24340\kappa + 375\pi^2) - 400\chi^2 - 117120\chi + 3920\pi\chi + 960\kappa\chi \\ &\quad +226560\kappa - 4720\kappa\pi - 560\kappa^2 + 530880\pi - 9600\pi^2. \end{aligned}$$

$$\begin{aligned} 6! \cdot N_6 &= d^6 - 150d^5 + d^4(8980 - 15\kappa + 15\chi - 75\pi) \\ &\quad -d^3(268200 - 2020\kappa + 1700\chi - 8340\pi) \\ &\quad +d^2(3996064 + 45\chi^2 + 71100\chi - 90\chi\kappa - 450\chi\pi \\ &\quad +450\kappa\pi + 1125\pi^2 - 101040\kappa + 45\kappa^2 - 340530\pi) \\ &\quad -d(23761920 + 2850\chi^2 + 1292320\chi - 28020\chi\pi \\ &\quad -6660\chi\kappa + 3810\kappa^2 + 32820\kappa\pi - 5995740\pi \\ &\quad -2224040\kappa + 68850\pi^2) + 15\chi^3 + \chi^2(45160 - 45\kappa - 225\pi) \\ &\quad +\chi(8517120 + 1125\pi^2 + 450\kappa\pi - 435030\pi - 123460\kappa + 45\kappa^2) \\ &\quad -18151200\kappa + 598170\kappa\pi - 1875\pi^3 - 37768560\pi - 1125\kappa\pi^2 \\ &\quad -15\kappa^3 + 1046790\pi^2 - 225\kappa^2\pi + 80860\kappa^2 \end{aligned}$$

These calculations verify LeBarz' trisecant formula for N_3 [19, Théorème 8] and the computation of N_4 by Tikhomirov and Troshina [26]. The formulae for N_5 and

N_6 seem to be new. I omit the presentation of N_7: the information is contained in the following analysis of these numerical data.

It is always possible to organise these data into the following form:

$$\sum_{n\geq 0} N_n(-z)^n = \exp\left(-\sum_{m>0}\frac{z^m}{m}d_m\right).$$

What is suprising is that the polynomials d_m in the variables d, π, κ, and χ should depend *linearly* on the *three* expressions d, $\pi_0 := \pi - 2\kappa$, and $\chi_0 := \chi - 11\kappa$. This holds for $m \leq 7$ according to the computations above, which imply that

$$\begin{aligned} d_1 &= d \\ d_2 &= 10d + 5\pi_0 - \chi_0 \\ d_3 &= 112d + 96\pi_0 - 20\chi_0 \\ d_4 &= 1320d + 1507\pi_0 - 324\chi_0 \\ d_5 &= 16016d + 22120\pi_0 - 4880\chi_0 \\ d_6 &= 198016d + 314738\pi_0 - 70976\chi_0 \\ d_7 &= 2480640d + 4402720\pi_0 - 1012032\chi_0, \end{aligned}$$

and it is only natural to conjecture that this holds in general. Observe also that the sequence of coefficients of χ_0 seems to be the square of the sequence of coefficients of d. More precisely:

Conjecture 4.8 — *Let $f(z) := \sum_{m>0} 2^{m-2}\binom{3m-1}{m}z^m$. Then there is a power series $g(z) := \sum_{m>1}(3m-1)\beta_m z^m$ with positive integral coefficients such that*

$$-z\frac{d}{dz}\log(\sum_{n\geq 0} N_n(-z)^n) = f(z)d + g(z)\pi_0 - f(z)^2\chi_0. \tag{16}$$

□

The fact that the right hand side in (16) depends linearly on χ_0 can be proved by the methods in the forthcoming paper [4].

We thank Don Zagier for pointing out to us the existence of Sloane's 'Encyclopedia of Integer Sequences' [24]. We had had reasons to believe that the sequence of coefficients of d be divisible by the binomial coefficients $\binom{3m-1}{2}$. After dividing by these, we are left with the sequence $1, 1, 4, 24, 176, 1456$. A search for this reduced sequence in the encyclopedia was successful and led to the above given (conjectural) identification of the coefficients of f. Unfortunately, the corresponding 'reduced' sequence of coefficients β_m of π_0 remains mysterious:

$$0, 1, 12, 137, 1580, 18514, 220136\ldots$$

References

[1] J. Briançon, *Description de $Hilb^n\mathbb{C}\{x,y\}$*. Inventiones math. 41, 45-89 (1977).

[2] F. Catanese, *On Severi's proof of the double point formula*. Comm. in Algebra 7 (1979), 763-773.

[3] J. Cheah, *On the Cohomology of Hilbert Schemes of Points*. J. Alg. Geom. 5 (1996), 479-511.

[4] G. Ellingsrud, L. Göttsche, M. Lehn, *On the cobordism class of Hilbert schemes of Points on Surfaces*. In preparation.

[5] G. Ellingsrud, A. Strømme, *On the homology of the Hilbert schemes on points in the plane*. Inv. Math. 87 (1987), 343-352.

[6] G. Ellingsrud, A. Strømme, *Towards the Chow ring of the Hilbert scheme of $\mathbb{P}^2$*. J. reine angew. Math. 441 (1993), 33-44.

[7] G. Ellingsrud, A. Strømme, *An intersection number for the punctual Hilbert scheme of a surface*. Trans. Amer. Math. Soc. to appear.

[8] B. Fantechi, L. Göttsche, *The cohomology ring of the Hilbert scheme of 3 points on a smooth projective variety*. J. reine angew. Math. 439 (1993), 147-158,

[9] J. Fogarty, *Algebraic Families on an Algebraic Surface*. Am. J. Math. 10, 511-521 (1968).

[10] W. Fulton, *Intersection Theory*. Erg. Math. (3. Folge) Band 2, Springer Verlag 1984.

[11] L. Göttsche, *The Betti numbers of the Hilbert scheme of points on a smooth projective surface*. Math. Ann. 286 (1990), 193-207.

[12] L. Göttsche, W. Sörgel, *Perverse sheaves and the cohomology of Hilbert schemes of smooth algebraic surfaces*. Math. Ann. 296 (1993), 235-245.

[13] I. Grojnowski, *Instantons and Affine Algebras I: The Hilbert Scheme and Vertex Operators*. Math. Res. Letters 3 (1996), 275-291.

[14] A. Grothendieck, *Techniques de construction et théorèmes d'existence en géométrie algébrique IV: Les schémas de Hilbert*. Séminaire Bourbaki, 1960/61, no. 221.

[15] A. Grothendieck, *Sur quelques points d'algèbre homologique*. Tôhoku Math. J. 9 (1957), 119-221.

[16] D. Huybrechts, M. Lehn, *The geometry of moduli spaces of coherent sheaves.* Aspects of Mathematics, Vol. E 31. Vieweg Verlag, 1997.

[17] A. Iarrobino, *Punctual Hilbert Schemes*. Memoirs of the AMS, Volume 10, Number 188, 1977.

[18] B. Iversen, *Linear determinants with applications to the Picard scheme of a family of algebraic curves*. Lect. Notes Math. 174, Springer Verlag, Berlin (1970).

[19] P. LeBarz, *Formules pour les trisécantes des surfaces algébriques*. L'Ens. Math. 33 (1987), 1-66.

[20] I. G. Macdonald, *The Poincaré Polynomial of a Symmetric Product*. Proc. Cambridge Phil. Soc. 58 (1962), 563-568.

[21] H. Nakajima, *Heisenberg algebra and Hilbert schemes of points on projective surfaces*. Ann. Math. 145 (1997), 379-388.

[22] H. Nakajima, *Lectures on Hilbert schemes of points on surfaces*. Preprint, University of Tokyo, 1996.

[23] F. Severi, *Sulle intersezioni delle varietá algebriche e sopra i loro caratterie singolaritá proiettive*. Mem. Accad. Scienze di Torino, S. II 52 (1902), 61-118. Also in: Memorie Scelte, I, Zuffi (1950), Bologna.

[24] N. J. A. Sloane, S. Plouffe, *The Encyclopedia of Integer Sequences*. Academic Press, San Diego, 1995. On-line version: `http://www.research.att.com/_njas/sequences.`

[25] A. S. Tikhomirov, *Standard bundles on a Hilbert scheme of points on a surface*. In: Algebraic geometry and its applications, Yaroslavl',1992. Aspects of Mathematics, Vol. E25. Vieweg Verlag, 1994.

[26] A. S. Tikhomirov, T. L. Troshina, *Top Segre class of a standard vector bundle $\mathcal{E}_D^4$ on the Hilbert scheme* $\mathrm{Hilb}^4(S)$ *of a surface* S. In: Algebraic geometry and its applications, Yaroslavl',1992. Aspects of Mathematics, Vol. E25. Vieweg Verlag, 1994.

[27] C. Vafa, E. Witten, *A strong coupling test of S-duality*. Nucl. Phys. 431 (1994), 3-77.

Manfred Lehn
Mathematisches Institut der Georg-August-Universtität
Bunsenstraße 3-5, D-37073 Göttingen, Germany
e-mail: `lehn@uni-math.gwdg.de`

ENUMERATING PERMUTATIONS AVOIDING A PAIR OF BABSON-STEINGRÍMSSON PATTERNS

ANDERS CLAESSON AND TOUFIK MANSOUR

ABSTRACT. Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. Subsequently, Claesson presented a complete solution for the number of permutations avoiding any single pattern of type $(1,2)$ or $(2,1)$. For eight of these twelve patterns the answer is given by the Bell numbers. For the remaining four the answer is given by the Catalan numbers.

In the present paper we give a complete solution for the number of permutations avoiding a pair of patterns of type $(1,2)$ or $(2,1)$. We also conjecture the number of permutations avoiding the patterns in any set of three or more such patterns.

1. INTRODUCTION

Classically, a pattern is a permutation $\sigma \in \mathcal{S}_k$, and a permutation $\pi \in \mathcal{S}_n$ avoids σ if there is no subword of π that is order equivalent to σ. For example, $\pi \in \mathcal{S}_n$ avoids 132 if there is no $1 \le i < j < k \le n$ such that $\pi(i) < \pi(k) < \pi(j)$. We denote by $\mathcal{S}_n(\sigma)$ the set permutations in $\mathcal{S}_n$ that avoids σ.

The first case to be examined was the case of permutations avoiding one pattern of length 3. Knuth [6] found that, for any $\tau \in \mathcal{S}_3$, $|\mathcal{S}_n(\tau)| = C_n$, where $C_n = \frac{1}{n+1}\binom{2n}{n}$ is the nth Catalan number. Later Simion and Schmidt [7] found the cardinality of $\mathcal{S}_n(P)$ for all $P \subseteq \mathcal{S}_3$.

In [1] Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. The motivation for Babson and Steingrímsson in introducing these patterns was the study of Mahonian statistics. Two examples of such patterns are 1-32 and 13-2 (1-32 and 13-2 are of type $(1,2)$ and $(2,1)$ respectively). A permutation $\pi = a_1 a_2 \cdots a_n$ avoids 1-32 if there are no subwords $a_i a_j a_{j+1}$ of π such that $a_i < a_{j+1} < a_j$. Similarly π avoids 13-2 if there are no subwords $a_i a_{i+1} a_j$ of π such that $a_i < a_j < a_{i+1}$.

Claesson [2] presented a complete solution for the number of permutations avoiding any single pattern of type $(1,2)$ or $(2,1)$ as follows.

Proposition 1 (Claesson [2]). *Let $n \in \mathbb{N}$. We have*

$$|\mathcal{S}_n(p)| = \begin{cases} B_n & \text{if } p \in \{1\text{-}23, 3\text{-}21, 12\text{-}3, 32\text{-}1, 1\text{-}32, 3\text{-}12, 21\text{-}3, 23\text{-}1\}, \\ C_n & \text{if } p \in \{2\text{-}13, 2\text{-}31, 13\text{-}2, 31\text{-}2\}, \end{cases}$$

where B_n and C_n are the nth Bell and Catalan numbers, respectively.

In addition, Claesson gave some results for the number of permutations avoiding a pair of patterns.

Proposition 2 (Claesson [2]). *Let $n \in \mathbb{N}$. We have*

$$\mathcal{S}_n(1\text{-}23,\ 12\text{-}3) = B_n^*, \quad \mathcal{S}_n(1\text{-}23,\ 1\text{-}32) = I_n, \quad \text{and} \quad \mathcal{S}_n(1\text{-}23,\ 13\text{-}2) = M_n,$$

Date: July 23, 2002.

Key words and phrases. permutation, pattern avoidance.

where B_n^* is the nth Bessel number (# non-overlapping partitions of $[n]$ (see [4])), I_n is the number of involutions in $\mathcal{S}_n$, and M_n is the nth Motzkin number.

This paper is organized as follows. In Section 2 we define the notion of a pattern and some other useful concepts. For a proof of Proposition 1 we could refer the reader to [2]. We will however prove Proposition 1 in Section 3 in the context of binary trees. The idea being that this will be a useful aid to understanding of the proofs of Section 4. In Section 4 we give a solution for the number of permutations avoiding any given pair of patterns of type $(1,2)$ or $(2,1)$. These results are summarized in the following table.

# pairs	$\lvert\mathcal{S}_n(p,q)\rvert$
2	$0, n > 5$
2	$2(n-1)$
4	$\binom{n}{2}+1$
34	2^{n-1}
8	M_n
2	a_n
4	b_n
4	I_n
4	C_n
2	B_n^*

Here

$$\sum_{n\geq 0} a_n x^n = \frac{1}{1-x-x^2\sum_{n\geq 0} B_n^* x^n}$$

and

$$b_{n+2} = b_{n+1} + \sum_{k=0}^{n} \binom{n}{k} b_k.$$

Finally, in Section 5 we conjecture the sequences $\lvert\mathcal{S}_n(P)\rvert$ for sets P of three or more patterns of type $(1,2)$ or $(2,1)$.

2. Preliminaries

By an *alphabet* X we mean a non-empty set. An element of X is called a *letter*. A *word* over X is a finite sequence of letters from X. We consider also the *empty word*, that is, the word with no letters; it is denoted by ϵ. Let $w = x_1 x_2 \cdots x_n$ be a word over X. We call $|w| := n$ the *length* of w. A *subword* of w is a word $v = x_{i_1} x_{i_2} \cdots x_{i_k}$, where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$.

Let $[n] := \{1, 2, \ldots, n\}$ (so $[0] = \emptyset$). A *permutation* of $[n]$ is bijection from $[n]$ to $[n]$. Let $\mathcal{S}_n$ be the set of permutations of $[n]$, and $\mathcal{S} = \cup_{n\geq 0}\mathcal{S}_n$. We shall usually think of a permutation π as the word $\pi(1)\pi(2)\cdots\pi(n)$ over the alphabet $[n]$.

Define the *reverse* of π by $\pi^r(i) = \pi(n+1-i)$, and define the *complement* of π by $\pi^c(i) = n+1-\pi(i)$, where $i \in [n]$.

For each word $w = x_1 x_2 \cdots x_n$ over the alphabet $\{1, 2, 3, 4, \ldots\}$ without repeated letters, we define the *projection* of w onto $\mathcal{S}_n$, which we denote $\mathrm{proj}(w)$, by

$$\mathrm{proj}(w) = a_1 a_2 \cdots a_n, \quad \text{where } a_i = |\{j \in [n] : x_j \leq x_i\}|.$$

Equivalently, $\mathrm{proj}(w)$ is the permutation in $\mathcal{S}_n$ which is order equivalent to w. For example, $\mathrm{proj}(2659) = 1324$.

We may regard a *pattern* as a function from $\mathcal{S}_n$ to the set $\mathbb{N}$ of natural numbers. The patterns of main interest to us are defined as follows. Let $xyz \in \mathcal{S}_3$ and $\pi = a_1 a_2 \cdots a_n \in \mathcal{S}_n$, then

$$(x\text{-}yz)\,\pi = |\{a_i a_j a_{j+1} : \mathrm{proj}(a_i a_j a_{j+1}) = xyz, 1 \leq i < j < n\}|$$

and similarly $(xy\text{-}z)\,\pi = (z\text{-}yx)\,\pi^r$. For instance

$$(1\text{-}23)\,491273865 = |\{127, 138, 238\}| = 3.$$

A pattern $p = p_1\text{-}p_2\text{-}\cdots\text{-}p_k$ containing exactly $k-1$ dashes is said to be of type $(|p_1|, |p_2|, \ldots, |p_k|)$. For example, the pattern $142\text{-}5\text{-}367$ is of type $(3,1,3)$, and any classical pattern of length k is of type $(\underbrace{1,1,\ldots,1}_{k})$.

We say that a permutation π *avoids* a pattern p if $p\,\pi = 0$. The set of all permutations in $\mathcal{S}_n$ that avoids p is denoted $\mathcal{S}_n(p)$ and, more generally, $\mathcal{S}_n(P) = \bigcap_{p\in P} \mathcal{S}_n(p)$ and $\mathcal{S}(P) = \bigcup_{n\geq 0} \mathcal{S}_n(P)$.

We extend the definition of reverse and complement to patterns the following way. Let us call π the *underlying permutation* of the pattern p if π is obtained from p by deleting all the dashes in p. If p is a pattern with underlying permutation π, then p^c is the pattern with underlying permutation π^c and with dashes at precisely the same positions as there are dashes in p. We define p^r as the pattern we get from regarding p as a word and reading it backwards. For example, $(1\text{-}23)^c = 3\text{-}21$ and $(1\text{-}23)^r = 32\text{-}1$. Observe that

$$\sigma \in \mathcal{S}_n(p) \iff \sigma^r \in \mathcal{S}_n(p^r)$$
$$\sigma \in \mathcal{S}_n(p) \iff \sigma^c \in \mathcal{S}_n(p^c).$$

These observations of course generalize to $\mathcal{S}_n(P)$ for any set of patterns P.

The operations reverse and complement generates the dihedral group D_2 (the symmetry group of a rectangle). The orbits of D_2 in the set of patterns of type $(1,2)$ or $(2,1)$ will be called *symmetry classes*. For instance, the symmetry class of $1\text{-}23$ is

$$\{1\text{-}23, 3\text{-}21, 12\text{-}3, 32\text{-}1\}.$$

We also talk about symmetry classes of sets of patterns (defined in the obvious way). For example, the symmetry class of $\{1\text{-}23, 3\text{-}21\}$ is $\{\{1\text{-}23, 3\text{-}21\}, \{32\text{-}1, 12\text{-}3\}\}$.

A set of patterns P such that if $p, p' \in P$ then, for each n, $|\mathcal{S}_n(p)| = |\mathcal{S}_n(p')|$ is called a *Wilf-class*. For instance, by Proposition 1, the Wilf-class of $1\text{-}23$ is

$$\{1\text{-}23, 3\text{-}21, 12\text{-}3, 32\text{-}1, 1\text{-}32, 3\text{-}12, 21\text{-}3, 23\text{-}1\}.$$

We also talk about Wilf-classes of sets of patterns (defined in the obvious way). It is clear that symmetry classes are Wilf-classes, but as we have seen the converse does not hold in general.

In what follows we will frequently use the following well known bijection between increasing binary trees and permutations (e.g. see [8, p. 24]). Let π be any word on the alphabet $\{1, 2, 3, 4, \ldots\}$ with no repeated letters. If $\pi \neq \epsilon$ then we can factor π as $\pi = \sigma\,\hat{0}\,\tau$, where $\hat{0}$ is the minimal element of π. Define $T(\epsilon) = \bullet$ (a leaf) and

$$T(\pi) = \begin{array}{ccc} & \hat{0} & \\ T(\sigma) & & T(\tau) \end{array}$$

In addition, we define $U(t)$ as the unlabelled counterpart of the labelled tree t. For instance

$$T(316452) = \begin{array}{ccccc} & & 1 & & \\ & 3 & & 2 & \\ & & & & \\ & & 4 & & \\ & & & & \\ & 6 & & 5 & \end{array} \qquad U \circ T(316452) = \begin{array}{ccccc} & & \circ & & \\ & \circ & & \circ & \\ & & & & \\ & & \circ & & \\ & & & & \\ & \circ & & \circ & \end{array}$$

Note that we, for ease of presentation, do not display the leafs ($\bullet$).

3. Single patterns

There are 3 symmetry classes and 2 Wilf-classes of single patterns. The details are as follows.

Proposition 3 (Claesson [2])**.** *Let $n \in \mathbb{N}$. We have*

$$|\mathcal{S}(p)| = \begin{cases} B_n & \text{if } p \in \{1\text{-}23, 3\text{-}21, 12\text{-}3, 32\text{-}1\}, \\ B_n & \text{if } p \in \{1\text{-}32, 3\text{-}12, 21\text{-}3, 23\text{-}1\}, \\ C_n & \text{if } p \in \{2\text{-}13, 2\text{-}31, 13\text{-}2, 31\text{-}2\}, \end{cases}$$

where B_n and C_n are the nth Bell and Catalan numbers, respectively.

Proof of the first case. Note that

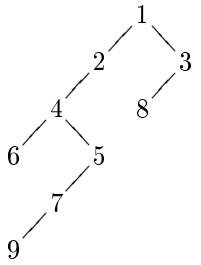
$$\sigma 1 \tau \in \mathcal{S}(1\text{-}23) \iff \begin{cases} \mathrm{proj}(\sigma) \in \mathcal{S}(1\text{-}23) \\ \mathrm{proj}(\tau) \in \mathcal{S}(12) \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

where of course $\mathcal{S}(12) = \{\epsilon, 1, 21, 321, 4321, \ldots\}$. This enable us to give a bijection Φ between $\mathcal{S}_n(1\text{-}23)$ and the set of partitions of $[n]$, by induction. Let the elements of 1τ form the first block of $\Phi(\sigma 1 \tau)$ and let the rest of the blocks be as in $\Phi(\sigma)$. □

The most transparent way to see the above correspondence is perhaps to view the permutation as an increasing binary tree.

Example 4. The tree

$T(649752183) =$



corresponds to the partition $\{\{1, 3, 8\}, \{2\}, \{4, 5, 7, 9\}, \{6\}\}$.

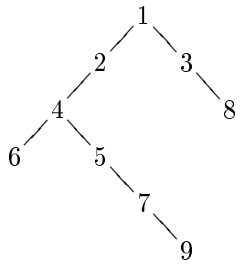
Proof of the second case. This case is analogous to the previous one. We have

$$\sigma 1 \tau \in \mathcal{S}(1\text{-}32) \iff \begin{cases} \mathrm{proj}(\sigma) \in \mathcal{S}(1\text{-}32) \\ \mathrm{proj}(\tau) \in \mathcal{S}(21) \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

We give a bijection Φ between $\mathcal{S}_n(1\text{-}23)$ and the set of partitions of $[n]$, by induction. Let the elements of 1τ form the first block of $\Phi(\sigma 1 \tau)$ and let the rest of the blocks be as in $\Phi(\sigma)$. □

Example 5. The tree

$T(645792138) =$



corresponds to the partition $\{\{1, 3, 8\}, \{2\}, \{4, 5, 7, 9\}, \{6\}\}$.

Now that we have seen the structure of $\mathcal{S}(1\text{-}23)$ and $\mathcal{S}(1\text{-}32)$, it is trivial to give a bijection between the two sets. Indeed, if $\Theta : \mathcal{S}(1\text{-}23) \to \mathcal{S}(1\text{-}32)$ is given by $\Theta(\epsilon) = \epsilon$ and $\Theta(\sigma 1 \tau) = \Theta(\sigma)\, 1\, \tau^r$ then Θ is such a bijection. Actually Θ is its own inverse.

Proof of the third case. It is plain that a permutation avoids 2-13 if and only if it avoids 2-1-3 (see [2]). Note that

$$\sigma 1 \tau \in \mathcal{S}(2\text{-}1\text{-}3) \iff \begin{cases} \mathrm{proj}(\sigma), \mathrm{proj}(\tau) \in \mathcal{S}(2\text{-}1\text{-}3) \\ \tau > \sigma \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

where $\tau > \sigma$ means that any letter of τ is greater than any letter of σ. Hence we get a unique labelling of the binary tree corresponding to $\sigma 1 \tau$, that is, if $\pi_1, \pi_2 \in \mathcal{S}(2\text{-}1\text{-}3)$ and $U \circ T(\pi_1) = U \circ T(\pi_2)$ then $\pi_1 = \pi_2$. It is well known that there are exactly C_n (unlabelled) binary trees with n (internal) nodes. The validity of the last statement is for example seen from the following simple bijection between Dyck words and binary trees. Fixing notation, we let the set of Dyck words be the smallest set of words over $\{u, d\}$ that contains the empty word and is closed under $(\alpha, \beta) \mapsto u\alpha d\beta$. Now the promised bijection is given by $\Psi(\bullet) = \epsilon$ and

$$\Psi\left(\begin{array}{ccc} & \circ & \\ L & & R \end{array} \right) = u\Psi(L)d\Psi(R).$$

□

4. Pairs of patterns

There are $\binom{12}{2} = 66$ pairs of patterns altogether. It turns out that there are 21 symmetry classes and 10 Wilf-classes. The details are as follows.

4.1. The Wilf-class corresponding to $\{0\}_n$.

Proposition 6. *Let $n \in \mathbb{N}$ with $n > 5$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 32\text{-}1\}, \{3\text{-}21, 12\text{-}3\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 0$.

Proof. We have

$$\sigma 1 \tau \in \mathcal{S}(1\text{-}23, 32\text{-}1) \iff \begin{cases} \mathrm{proj}(\sigma) \in \mathcal{S}(21, 1\text{-}23) \\ \mathrm{proj}(\tau) \in \mathcal{S}(12, 32\text{-}1) \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

The result now follows from $\mathcal{S}(21, 1\text{-}23) = \{\epsilon, 1, 12\}$ and $\mathcal{S}(12, 32\text{-}1) = \{\epsilon, 1, 21\}$. □

4.2. The Wilf-class corresponding to $\{2(n-1)\}_n$.

Proposition 7. *Let $n \in \mathbb{N}$ with $n > 1$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 3\text{-}21\}, \{32\text{-}1, 12\text{-}3\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 2(n-1)$.

Proof. Since 3-21 is the complement of 1-23, the cardinality of $\mathcal{S}_n(1\text{-}23, 3\text{-}21)$ is twice the number of permutations in $\mathcal{S}_n(1\text{-}23, 3\text{-}21)$ in which 1 precedes n. In addition, 1 and n must be adjacent letters in a permutation avoiding 1-23 and 3-21. Let $\sigma 1 n \tau$ be such a permutation. Note that τ must be both increasing and decreasing, that is, $\tau \in \{\epsilon, 2, 3, 4, \ldots, n-1\}$, so there are $n-1$ choices for τ. Furthermore, there is exactly one permutation in $\mathcal{S}_n(1\text{-}23, 3\text{-}21)$ of the form $\sigma 1 n$, namely $(\lceil \frac{n+1}{2} \rceil, \ldots, n-2, 3, n-1, 2, n, 1)$, and similarly there is exactly one of the form $\sigma 1 n k$ for each $k \in \{2, 3, \ldots, n-1\}$. This completes our argument. □

4.3. **The Wilf-class corresponding to $\{\binom{n}{2}+1\}_n$.**

Proposition 8. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 2\text{-}31\}, \{3\text{-}21, 2\text{-}13\}, \{12\text{-}3, 31\text{-}2\}, \{32\text{-}1, 13\text{-}2\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = \binom{n}{2} + 1$.

Proof. Note that

$$\sigma 1 \tau \in \mathcal{S}(1\text{-}23, 2\text{-}31) \iff \begin{cases} \mathrm{proj}(\sigma), \mathrm{proj}(\tau) \in \mathcal{S}(12) \\ \sigma 1 \tau \in \mathcal{S}(2\text{-}31) \end{cases}$$

It is now rather easy to see that $\pi \in \mathcal{S}_n(1\text{-}23, 2\text{-}31)$ if and only if $\pi = n \cdots 21$ or π is constructed the following way. Choose i and j such that $1 \le j < i \le n$. Let $\pi(i-1) = 1$, $\pi(i) = n + 1 - j$ and arrange the rest of the elements so that $\pi(1) > \pi(2) > \cdots > \pi(i-1)$ and $\pi(i) > \pi(i+1) > \cdots > \pi(n)$ (this arrangement is unique). Since there are $\binom{n}{2}$ ways of choosing i and j we get the desired result. □

4.4. **The Wilf-class corresponding to $\{2^{n-1}\}_n$.**

Proposition 9. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 2\text{-}13\}, \{3\text{-}21, 2\text{-}31\}, \{12\text{-}3, 13\text{-}2\}, \{32\text{-}1, 31\text{-}2\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 2^{n-1}$.

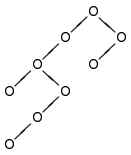
Proof. We have

$$\sigma 1 \tau \in \mathcal{S}(1\text{-}23, 2\text{-}13) \iff \begin{cases} \mathrm{proj}(\sigma) \in \mathcal{S}(1\text{-}23, 2\text{-}13) \\ \mathrm{proj}(\tau) \in \mathcal{S}(12) \\ \sigma > \tau \\ \sigma 1 \tau \in \mathcal{S}, \end{cases}$$

where $\sigma > \tau$ means that any letter of τ is greater than any letter of σ. This enable us to give a bijection between $\mathcal{S}_n(1\text{-}23, 2\text{-}13)$ and the set of compositions (ordered formal sums) of n. Indeed, such a bijection Ψ is given by $\Psi(\epsilon) = \epsilon$ and $\Psi(\sigma 1 \tau) = \Psi(\sigma) + |1\tau|$. □

Example 10. The tree

$$U \circ T(958764132) \;=$$



corresponds to the composition $1 + 3 + 1 + 4$ of 9.

Proposition 11. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 23\text{-}1\}, \{3\text{-}21, 21\text{-}3\}, \{12\text{-}3, 3\text{-}12\}, \{32\text{-}1, 1\text{-}32\} \,\}$$

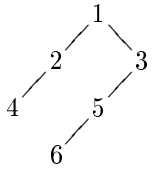
we have $|\mathcal{S}_n(p, q)| = 2^{n-1}$.

Proof. We have

$$\sigma 1 \tau \in \mathcal{S}(1\text{-}23, 23\text{-}1) \iff \begin{cases} \mathrm{proj}(\sigma), \mathrm{proj}(\tau) \in \mathcal{S}(12) \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

Hence a permutation in $\mathcal{S}(1\text{-}23, 23\text{-}1)$ is given by the following procedure. Choose a subset $S \subseteq \{2, 3, 4, \ldots, n\}$, let σ be the word obtained by writing the elements of S in decreasing order, and let τ be the word obtained by writing the elements of $\{2, 3, 4, \ldots, n\} \setminus S$ in decreasing order. □

Example 12. The tree

$T(421653) \;=$ 

corresponds to the subset $\{2,4\}$ of $\{2,3,4,5,6\}$.

Proposition 13. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p,q\}$ in the set*

$$\big\{\, \{1\text{-}23, 31\text{-}2\}, \{3\text{-}21, 13\text{-}2\}, \{12\text{-}3, 2\text{-}31\}, \{32\text{-}1, 2\text{-}13\} \,\big\}$$

we have $|\mathcal{S}_n(p,q)| = 2^{n-1}$.

Proof. This case is essentially identical to the case dealt with in Proposition 9. □

Proposition 14. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p,q\}$ in the set*

$$\big\{\, \{1\text{-}32, 2\text{-}13\}, \{3\text{-}12, 2\text{-}31\}, \{13\text{-}2, 21\text{-}3\}, \{23\text{-}1, 31\text{-}2\} \,\big\}$$

we have $|\mathcal{S}_n(p,q)| = 2^{n-1}$.

Proof. The bijection Θ between $\mathcal{S}(1\text{-}23)$ and $\mathcal{S}(1\text{-}32)$ (see page 3) provides a one-to-one correspondence between $\mathcal{S}_n(1\text{-}32, 2\text{-}13)$ and $\mathcal{S}_n(1\text{-}23, 2\text{-}13)$. Consequently the result follows from Proposition 9. □

Proposition 15. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p,q\}$ in the set*

$$\big\{\, \{1\text{-}32, 2\text{-}31\}, \{3\text{-}12, 2\text{-}13\}, \{31\text{-}2, 21\text{-}3\}, \{23\text{-}1, 13\text{-}2\} \,\big\}$$

we have $|\mathcal{S}_n(p,q)| = 2^{n-1}$.

Proof. We have

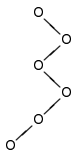
$$\sigma 1 \tau \in \mathcal{S}(3\text{-}12, 2\text{-}13) \iff \begin{cases} \mathrm{proj}(\sigma), \mathrm{proj}(\tau) \in \mathcal{S}(3\text{-}12, 2\text{-}13) \\ \sigma = \epsilon \;\text{ or }\; \tau = \epsilon \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

Thus a bijection between $\mathcal{S}_n(3\text{-}12, 2\text{-}13)$ and $\{0,1\}^{n-1}$ is given by $\Psi(\epsilon) = \epsilon$ and

$$\Psi(\sigma 1 \tau) = x\Psi(\sigma\tau) \text{ where } x = \begin{cases} 1 & \text{if } \sigma \neq \epsilon, \\ 0 & \text{if } \tau \neq \epsilon, \\ \epsilon & \text{otherwise.} \end{cases}$$

□

Example 16. The tree

$U \circ T(136542) \;=$ 

corresponds to $01011 \in \{0,1\}^5$.

Proposition 17. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p,q\}$ in the set*

$$\big\{\, \{1\text{-}32, 3\text{-}12\}, \{23\text{-}1, 21\text{-}3\} \,\big\}$$

we have $|\mathcal{S}_n(p,q)| = 2^{n-1}$.

Proof. Since 3-12 is the complement of 1-32, the cardinality of $\mathcal{S}_n(1\text{-}32, 3\text{-}12)$ is twice the number of permutations in $\mathcal{S}_n(1\text{-}32, 3\text{-}12)$ in which 1 precedes n. In addition, n must be the last letter in such a permutation or else a hit of 1-32 would be formed. We have

$$\sigma 1 \tau n \in \mathcal{S}(1\text{-}32, 3\text{-}12) \iff \begin{cases} \mathrm{proj}(\sigma 1 \tau) \in \mathcal{S}(1\text{-}32, 3\text{-}12) \\ \mathrm{proj}(\tau) \in \mathcal{S}(21) \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

$$\iff \begin{cases} \mathrm{proj}(\sigma) \in \mathcal{S}(1\text{-}32, 3\text{-}12) \\ \mathrm{proj}(\tau) \in \mathcal{S}(21) \\ \sigma < \tau \\ \sigma 1 \tau \in \mathcal{S} \end{cases}$$

The rest of the proof follows the same lines as the proof of Proposition 9. □

Proposition 18. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}32, 23\text{-}1\}, \{3\text{-}12, 21\text{-}3\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 2^{n-1}$.

Proof. We can copy almost verbatim the proof of Proposition 15, indeed, it is easy to see that $\mathcal{S}_n(1\text{-}32, 23\text{-}1) = \mathcal{S}_n(1\text{-}32, 2\text{-}31)$. □

Proposition 19. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}32, 31\text{-}2\}, \{3\text{-}12, 13\text{-}2\}, \{21\text{-}3, 2\text{-}31\}, \{23\text{-}1, 2\text{-}13\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 2^{n-1}$.

Proof. We can copy almost verbatim the proof of Proposition 17, indeed, it is easy to see that $\mathcal{S}_n(1\text{-}32, 31\text{-}2) = \mathcal{S}_n(1\text{-}32, 3\text{-}12)$. □

Proposition 20. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p, q\}$ in the set*

$$\{\, \{2\text{-}13, 2\text{-}31\}, \{31\text{-}2, 13\text{-}2\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 2^{n-1}$.

Proof. $|\mathcal{S}_n(2\text{-}13, 2\text{-}31)| = |\mathcal{S}_n(2\text{-}1\text{-}3, 2\text{-}3\text{-}1)| = 2^{n-1}$ by [7, Lemma 5(d)]. □

Proposition 21. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p, q\}$ in the set*

$$\{\, \{2\text{-}13, 13\text{-}2\}, \{2\text{-}31, 31\text{-}2\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 2^{n-1}$.

Proof. $|\mathcal{S}_n(2\text{-}13, 13\text{-}2)| = |\mathcal{S}_n(1\text{-}3\text{-}2, 2\text{-}1\text{-}3)| = 2^{n-1}$ by [7, Lemma 5(b)]. □

Proposition 22. *Let $n \in \mathbb{N}$ with $n > 0$. For any pair $\{p, q\}$ in the set*

$$\{\, \{2\text{-}13, 31\text{-}2\}, \{2\text{-}31, 13\text{-}2\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = 2^{n-1}$.

Proof. $|\mathcal{S}_n(2\text{-}13, 31\text{-}2)| = |\mathcal{S}_n(2\text{-}1\text{-}3, 3\text{-}1\text{-}2)| = 2^{n-1}$ by [7, Lemma 5(c)]. □

4.5. **The Wilf-class corresponding to $\{M_n\}_n$.**

Proposition 23. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 13\text{-}2\}, \{3\text{-}21, 31\text{-}2\}, \{12\text{-}3, 2\text{-}13\}, \{32\text{-}1, 2\text{-}31\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = M_n$, where M_n is the nth Motzkin number.

Proof. See Proposition 2. □

Proposition 24. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 21\text{-}3\}, \{3\text{-}21, 23\text{-}1\}, \{12\text{-}3, 1\text{-}32\}, \{32\text{-}1, 3\text{-}12\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = M_n$, where M_n is the nth Motzkin number.

Proof. We give a bijection $\Lambda : \mathcal{S}_n(1\text{-}23, 21\text{-}3) \to \mathcal{S}_n(1\text{-}23, 13\text{-}2)$ by means of induction. Let $\pi \in \mathcal{S}_n(1\text{-}23, 21\text{-}3)$. Define $\Lambda(\pi) = \pi$ for $n \leq 1$. Assume $n \geq 2$ and $\pi = a_1 a_2 \cdots a_n$. It is plain that either $a_1 = n$ or $a_2 = n$, so we can define

$$\Lambda(\pi) = \begin{cases} (a'_1 + 1, \ldots, a'_{n-1} + 1, a'_{n-2} + 1, 1) & \text{if } \begin{cases} a_1 = n \quad \text{and} \\ a'_1 \cdots a'_{n-1} = \Lambda(a_2 a_3 a_4 \cdots a_n), \end{cases} \\ (a'_1 + 1, \ldots, a'_{n-1} + 1, 1, a'_{n-2} + 1) & \text{if } \begin{cases} a_2 = n \quad \text{and} \\ a'_1 \cdots a'_{n-1} = \Lambda(a_1 a_3 a_4 \cdots a_n). \end{cases} \end{cases}$$

Observing that if $\sigma \in \mathcal{S}_n(1\text{-}23, 13\text{-}2)$ then $\sigma(n-1) = 1$ or $\sigma(n) = 1$, it easy to find the inverse of Λ. □

4.6. **The Wilf-class corresponding to $\{1, 1, 2, 4, 9, 22, 58, 164, 496, 1601, \ldots\}$.** In [2] Claesson introduced the notion of a monotone partition. A partition is *monotone* if its non-singleton blocks can be written in increasing order of their least element and increasing order of their greatest element, simultaneously. He then proved that monotone partitions and non-overlapping partitions are in ono-to-one correspondence. Non-overlapping partitions where first studied by Flajolet and Schot in [4]. A partition π is *non-overlapping* if for no two blocks A and B of π we have $\min A < \min B < \max A < \max B$. Let B_n^* be the number of non-overlapping partitions of $[n]$; this number is called the nth *Bessel number*. Proposition 2 tells us that there is a bijection between non-overlapping partitions and permutations avoiding 1-23 and 12-3. Below we define a new class of partitions called strongly monotone partitions and then show that there is a bijection between strongly monotone partitions and permutations avoiding 1-32 and 21-3.

Definition 25. Let π be an arbitrary partition whose blocks $\{A_1, \ldots, A_k\}$ are ordered so that for all $i \in [k-1]$, $\min A_i > \min A_{i+1}$. If $\max A_i > \max A_{i+1}$ for all $i \in [k-1]$, then we call π a *strongly monotone partition*.

In other words a partition is strongly monotone if its blocks can be written in increasing order of their least element and increasing order of their greatest element, simultaneously. Let us denote by a_n the number of strongly monotone partitions of $[n]$. The sequence $\{a_n\}_0^\infty$ starts with

$$1, 1, 2, 4, 9, 22, 58, 164, 496, 1601, 5502, 20075, 77531, 315947, 1354279.$$

It is routine to derive the continued fraction expansion

$$\sum_{n\geq 0} a_n x^n = \cfrac{1}{1-1\cdot x-\cfrac{x^2}{1-1\cdot x-\cfrac{x^2}{1-2\cdot x-\cfrac{x^2}{1-3\cdot x-\cfrac{x^2}{1-4\cdot x-\cfrac{x^2}{\ddots}}}}}}$$

using the standard machinery of Flajolet [3] and Françon and Viennot [5]. One can also note that there is a one-to-one correspondence between strongly monotone partitions and non-overlapping partition, π, such that if $\{x\}$ and B are blocks of π then either $x < \min B$ or $\max B < x$. In addition, we observe that

$$\sum_{n\geq 0} a_n x^n = \frac{1}{1-x-x^2B^*(x)},$$

where $B^*(x) = \sum_{n\geq 0} B_n^* x^n$ is the ordinary generating function for the Bessel numbers.

Proposition 26. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\big\{\, \{1\text{-}32, 21\text{-}3\}, \{3\text{-}12, 23\text{-}1\} \,\big\}$$

we have $|\mathcal{S}_n(p, q)| = a_n$, where a_n is the number of strongly monotone partitions of $[n]$ (see Definition 25).

Proof. Suppose $\pi \in \mathcal{S}_n$ has $k+1$ left-to-right minima $1, 1', 1'', \ldots, 1^{(k)}$ such that

$$1 < 1' < 1'' < \cdots < 1^{(k)}, \text{ and } \pi = 1^{(k)}\tau^{(k)} \cdots 1'\tau' 1\tau.$$

Then π avoids 1-32 if and only if, for each i, $\tau^{(i)} \in \mathcal{S}(21)$. If π avoids 1-32 and $x_i = \max 1^{(i)}\tau^{(i)}$ then π avoids 21-3 precisely when $x_0 < x_1 < \cdots < x_k$. This follows from observing that the only potential (21-3)-subwords of π are $x_{i+1}1^{(k)}x_j$ with $j \leq i$.

Mapping π to the partition $\{1\sigma, 1'\sigma', \ldots, 1^{(k)}\tau^{(k)}\}$ we thus get a one-to-one correspondence between permutations in $\mathcal{S}_n(1\text{-}32, 21\text{-}3)$ and strongly monotone partitions of $[n]$. □

4.7. **The Wilf-class corresponding to $\{1, 1, 2, 4, 9, 23, 65, 199, 654, 2296, \ldots\}$.**

Proposition 27. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\big\{\, \{1\text{-}23, 3\text{-}12\}, \{3\text{-}21, 1\text{-}32\}, \{23\text{-}1, 12\text{-}3\}, \{32\text{-}1, 21\text{-}3\} \,\big\}$$

we have $|\mathcal{S}_n(p, q)| = b_n$, where the sequence $\{b_n\}$ satisfies $b_0 = 1$ and, for $n \geq -2$,

$$b_{n+2} = b_{n+1} + \sum_{k=0}^{n} \binom{n}{k} b_k.$$

Proof. Suppose $\pi \in \mathcal{S}_n$ has $k+1$ left-to-right minima $1, 1', 1'', \ldots, 1^{(k)}$ such that

$$1 < 1' < 1'' < \cdots < 1^{(k)}, \text{ and } \pi = 1^{(k)}\tau^{(k)} \cdots 1'\tau' 1\tau.$$

Then π avoids 1-23 if and only if, for each i, $\tau^{(i)} \in \mathcal{S}(12)$. If π avoids 1-23 and $x_i = \max 1^{(i)}\tau^{(i)}$ then π avoids 3-12 precisely when

$$j > i \text{ and } x_i \neq 1^{(i)} \implies x_j < x_i.$$

This follows from observing that the only potential (3-12)-subwords of π are $x_j 1^{(k)} x_i$ with $j \leq i$. Thus we have established

$$\sigma 1 \tau \in \mathcal{S}_n(1\text{-}23, 3\text{-}12) \iff \begin{cases} \mathrm{proj}(\sigma) \in \mathcal{S}(1\text{-}23, 3\text{-}12) \\ \tau \neq \epsilon \ \Rightarrow \ \tau = \tau' n \text{ and } \mathrm{proj}(\tau') \in \mathcal{S}(12) \\ \sigma 1 \tau \in \mathcal{S}_n \end{cases}$$

If we know that $\sigma 1 \tau' n \in \mathcal{S}_n(1\text{-}23, 3\text{-}12)$ and $\mathrm{proj}(\tau') \in \mathcal{S}_k(12)$ then there are $\binom{n-2}{k}$ candidates for τ'. In this way the recursion follows. □

4.8. **The Wilf-class corresponding to I_n.**

Proposition 28. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 1\text{-}32\}, \{3\text{-}21, 3\text{-}12\}, \{21\text{-}3, 12\text{-}3\}, \{32\text{-}1, 23\text{-}1\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = I_n$, where I_n is the number of involutions in $\mathcal{S}_n$.

Proof. See Proposition 2. □

4.9. **The Wilf-class corresponding to C_n.**

Proposition 29. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}32, 13\text{-}2\}, \{3\text{-}12, 31\text{-}2\}, \{21\text{-}3, 2\text{-}13\}, \{23\text{-}1, 2\text{-}31\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = C_n$, where C_n is the nth Catalan number.

Proof. $\mathcal{S}_n(1\text{-}32, 13\text{-}2) = \mathcal{S}_n(1\text{-}3\text{-}2)$. □

4.10. **The Wilf-class corresponding to B_n^*.**

Proposition 30. *Let $n \in \mathbb{N}$. For any pair $\{p, q\}$ in the set*

$$\{\, \{1\text{-}23, 12\text{-}3\}, \{3\text{-}21, 32\text{-}1\} \,\}$$

we have $|\mathcal{S}_n(p, q)| = B_n^*$, where B_n^* is the nth Bessel number.

Proof. See Proposition 2. □

5. More than two patterns

Let P be a set of patterns of type $(1, 2)$ or $(2, 1)$. With the aid of a computer we have calculated the cardinality of $\mathcal{S}_n(P)$ for sets P of three ore more patterns. From these results we arrived at the plausible conjectures of table 1 (some of which are trivially true). We use the notation $m \times n$ to express that there are m symmetric classes each of which contains n sets. Moreover, we denote by F_n the nth *Fibonacci number* ($F_0 = F_1 = 1, F_{n+1} = F_n + F_{n-1}$).

Acknowledgements

The first author wishes to express his gratitude towards Einar Steingrímsson, Kimmo Eriksson, and Mireille Bousquet-Mélou; Einar for his guidance and infectious enthusiasm; Kimmo for useful suggestions and a very constructive discussion on the results of this paper; Mireille for her great hospitality during a stay at LaBRI, where some of the work on this paper was done.

We would like to thank N. J. A. Sloane for his excellent web site "The On-Line Encyclopedia of Integer Sequences"

`http://www.research.att.com/~njas/sequences/`.

It is simply an indispensable tool for all studies concerned with integer sequences.

For $|P| = 3$ there are 220 sets, 55 symmetry classes and 9 Wilf-classes.

cardinality	# sets
0	7×4
3	1×4
n	24×4
$1 + \binom{n}{2}$	2×4
F_n	7×4
$\binom{n}{[n/2]}$	1×4
$2^{n-2} + 1$	1×4
2^{n-1}	10×4
M_n	2×4

For $|P| = 4$ there are 495 sets, 135 symmetry classes, and 9 Wilf-classes.

cardinality	# sets
0	$1 \times 1 + 6 \times 2 + 30 \times 4$
2	$2 \times 1 + 5 \times 2 + 35 \times 4$
3	1×4
n	$37 \times 4 + 1 \times 2$
$1 + \binom{n}{2}$	1×4
F_n	$9 \times 4 + 1 \times 2$
$\binom{n}{[n/2]}$	1×2
$2^{n-2} + 1$	1×2
2^{n-1}	$1 \times 4 + 3 \times 2$

For $|P| = 5$ there are 792 sets, 198 symmetry classes, and 5 Wilf-classes.

cardinality	# sets
0	84×4
1	16×4
2	74×4
n	20×4
F_n	4×4

For $|P| = 6$ there are 924 sets, 246 symmetry classes, and 4 Wilf-classes.

cardinality	# sets
0	$17 \times 2 + 124 \times 4$
1	$4 \times 2 + 38 \times 4$
2	$7 \times 2 + 51 \times 4$
n	$1 \times 2 + 3 \times 4$
F_n	1×2

For $|P| = 7$ there are 792 sets, 198 symmetry classes, and 3 Wilf-classes.

cardinality	# sets
0	140×4
1	40×4
2	18×4

For $|P| = 8$ there are 495 sets, 135 symmetry classes, and 3 Wilf-classes.

cardinality	# sets
0	$2 \times 1 + 14 \times 2 + 94 \times 4$
1	$4 \times 2 + 18 \times 4$
2	$1 \times 1 + 2 \times 4$

For $|P| = 9$ there are 220 sets, 55 symmetry classes, and 2 Wilf-classes.

cardinality	# sets
0	50×4
1	5×4

For $|P| = 10$ there are 66 sets, 21 symmetry classes, and 2 Wilf-classes.

cardinality	# sets
0	$8 \times 2 + 12 \times 4$
1	1×2

For $|P| = 11$ there are 12 sets, 3 symmetry classes, and 1 Wilf-class.

cardinality	# sets
0	3×4

For $|P| = 12$ there is 1 set, 1 symmetry class, and 1 Wilf-class.

cardinality	# sets
0	1×1

TABLE 1. The cardinality of $\mathcal{S}_n(P)$ for $|P| > 2$.

REFERENCES

[1] E. Babson and E. Steingrímsson. Generalized permutation patterns and a classification of the Mahonian statistics. *Séminaire Lotharingien de Combinatoire*, B44b:18pp, 2000.

[2] A. Claesson. Generalized pattern avoidance. *To appear in: European Journal of Combinatorics*, 2001.

[3] P. Flajolet. Combinatorial aspects of continued fractions. *Annals of Discrete Mathematics*, 8:217–222, 1980.

[4] P. Flajolet and R. Schott. Non–overlapping partitions, continued fractions, Bessel functions and a divergent series. *European Journal of Combinatorics*, 11:421–432, 1990.
[5] J. Françon and G. Viennot. Permutations selon leurs pics, creux, doubles montées et double descentes, nombres d'Euler et nombres de Genocchi. *Discrete Math.*, 28(1):21–35, 1979.
[6] D. E. Knuth. *The art of computer programming*, volume 3. Addison-Wesley, 1973.
[7] R. Simion and F. W. Schmidt. Restricted permutations. *European Journal of Combinatorics*, 6:383–406, 1985.
[8] R. P. Stanley. *Enumerative Combinatorics*, volume 1. Cambridge University Press, 1997.

Matematik, Chalmers tekniska högskola och Göteborgs universitet, S-412 96 Göteborg, Sweden
E-mail address: claesson@math.chalmers.se

LaBRI, Univerité Bordeaux I, 351 cours de la Libération, 33405 Talence Cedex, France
E-mail address: toufik@labri.fr

Partitions of an Integer into Powers

Matthieu Latapy

LIAFA, *Université Paris 7, 2 place Jussieu, 75005 Paris. latapy@liafa.jussieu.fr*

In this paper, we use a simple discrete dynamical model to study partitions of integers into powers of another integer. We extend and generalize some known results about their enumeration and counting, and we give new structural results. In particular, we show that the set of these partitions can be ordered in a natural way which gives the distributive lattice structure to this set. We also give a tree structure which allow efficient and simple enumeration of the partitions of an integer.

Keywords: Integer partition, Composition, Lattice, Distributive Lattice, Discrete Dynamical Models, Chip Firing Game

1 Introduction

We study here the problem of writing a non-negative integer n as the sum of powers of another positive integer b:

$$n = p_0 b^0 + p_1 b^1 + \cdots + p_{k-1} b^{k-1}$$

with $p_{k-1} \neq 0$ and $p_i \in \mathbb{N}$ for all i. Following [Rod69], we call the k-tuple $(p_0, p_1, \ldots, p_{k-1})$ a *b-ary partition* of n. The integers p_i are called the *parts* of the partition and k is the *length* of the partition. A b-ary partition of n can be viewed as a representation of n in the basis b, with digits in $\mathbb{N}$. Conversely, given a k-tuple $(p_0, \ldots, p_{k-1})$ and a basis b, we will denote by $v_b(p_0, \ldots, p_{k-1})$ the integer $p_0 b^0 + p_1 b^1 + \cdots + p_{k-1} b^{k-1}$. There is a unique b-ary partition such that $p_i < b$ for all i, and it is the usual (canonical) representation of n in the basis b. Here, we consider the problem without any restriction over the parts: $p_i \in \mathbb{N}$, which is actually equivalent to say that $p_i \in \{0, 1, \ldots, n\}$ for all i. We will mainly be concerned with the enumeration and counting of the b-ary partitions of n, for given integers n and b.

This natural combinatorial problem has been introduced by Mahler [Mah40], who showed that the logarithm of the number of b-ary partitions of n grows as $\frac{(\log n)^2}{2 \log b}$. This asymptotic approximation was later improved by de Bruijn [dB48] and Pennington [Pen53]. Knuth [Knu66] studied the special case where $b = 2$. In this case, the function counting the b-ary partitions for a given n is called the *binary partition function*. This function has been widely studied. Euler and Tanturri [Eul50, Tan18a, Tan18b] studied its exact computation and Churchhouse [Chu69, Chu71] studied its congruence properties, while Fröberg [Fro77] gave a final solution to its asymptotical approximation. Later, Rödseth [Rod69] generalized some of these results to b-ary partitions for any b. Finally, Pfaltz [Pfa95] studied the subcase of the binary partitions of integers which are powers of two.

We are concerned here with the exact computation of the number of b-ary partitions of a given integer n, for any b. We will use a powerful technique we developped in [LP99] and [LMMP98]: incremental construction of the set of b-ary partitions of n, infinite extension and coding by an infinite tree. This method gives a deep understanding of the structure of the set of b-ary partitions of n. We will obtain this way a tree structure which permits the enumeration of all the b-ary partitions of n in linear time with respect to their number. We will also order these partitions in a natural way which gives the distributive lattice structure to this set. We recall that a *lattice* is a partially ordered set such that any two elements a and b have a least upper bound (called *supremum* of a and b and denoted by $a \vee b$) and a greatest lower bound (called *infimum* of a and b and denoted by $a \wedge b$). The element $a \vee b$ is the smallest element among the elements greater than both a and b. The element $a \wedge b$ is defined dually. A lattice is *distributive* if for all a, b and c: $(a \vee b) \wedge (a \vee c) = a \vee (b \wedge c)$ and $(a \wedge b) \vee (a \wedge c) = a \wedge (b \vee c)$. A distributive lattice is a strongly structured set, and many general results, for example efficient coding and algorithms, are known about such sets. For more details, see for example [DP90].

Notice that if we consider $b = 1$ and restrict the problem to partitions of length at most n, then we obtain the *compositions* of n, i.e. the series of at most n integers, the sum of which equals n. Many studies already deal with this special case. In particular, the (infinite) distributive lattice $R_1(\infty)$ which we will introduce in Section 4 is isomorphic to the well known Young lattice [Ber71]. Therefore, we will suppose $b > 1$ in the following. Notice however that some of the results we present here are already known in this special case (for example the distributive lattice structure), therefore they can be seen as an extension of the existing ones.

2 The lattice structure

In this section, we define a simple dynamical model which generates *all* the b-ary partitions of an integer. We will show that the set of b-ary partitions, ordered by the reflexive and transitive closure of the successor relation, has the distributive lattice structure.

Let us consider a b-ary partition $p = (p_0, p_1, \ldots, p_{k-1})$ of n, and let us define the following transition (or rewriting) rule: $p \xrightarrow{i} q$ if and only if for all $j \notin \{i, i+1\}$, $q_j = p_j$, $p_i \geq b$, $q_i = p_i - b$ and $q_{i+1} = p_{i+1} + 1$ (with the assumption that $p_k = 0$). In other words, if p_i is at least equal to b then q is obtained from p by removing b units from p_i and adding one unit to p_{i+1}. We call this operation *firing* i. The important point is to notice that q is then a b-ary partition of n. We call q a *successor* [†] of p, and we denote by $Succ_b(p)$ the set of all the successors of p, with respect to the rule. We denote by $R_b(n)$ the set of b-ary partitions of n reachable from (n) by iterating the evolution rule, ordered by the reflexive and transitive closure of the successor relation. Notice that the successor relation is the covering relation of the order, since it is defined as the transitive and reflexive closure of the successor relation, and one can easily verify that this relation has no reflexive ($x \longrightarrow x$) and no transitive ($x \longrightarrow z$ with $x \longrightarrow y$ and $y \longrightarrow z$) edge. See Figure 1 for some examples.

Given a sequence f of firings, we denote by $|f|_i$ the number of firings of i during f. Now, consider an element p of $R_b(n)$, and two sequences f and f' of firings which transform (n) into p. Then, $p_i = |f|_{i-1} - b \cdot |f|_i = |f'|_{i-1} - b \cdot |f'|_i$. Suppose that there exists an integer i such that $|f|_i \neq |f'|_i$, and let i be the smallest such integer. Then, $|f|_{i-1} = |f'|_{i-1}$ and the equality $|f|_{i-1} - b \cdot |f|_i = |f'|_{i-1} - b \cdot |f'|_i$ is

[†] Notice that the term *successor* can have many different meanings. We follow here the standard usage in discrete dynamical models, but in order theory the term has another meaning, and one may also consider that a *successor* of an integer n should be the integer $n+1$, which is not the case here.

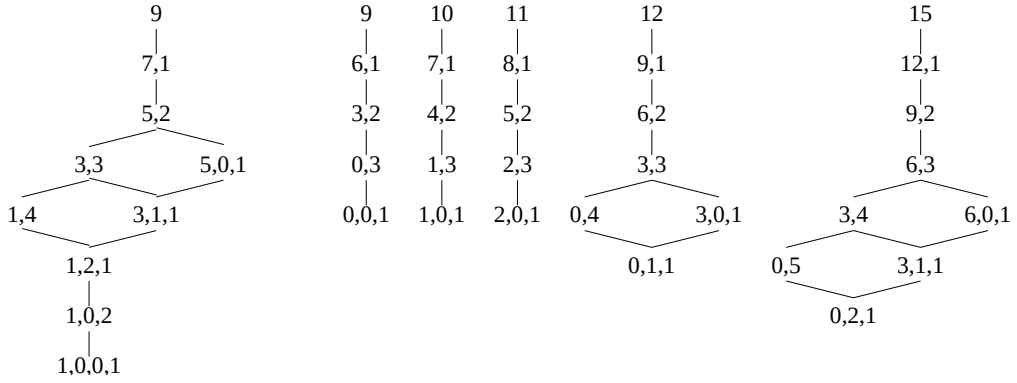


Fig. 1: From left to right, the sets $R_2(9)$, $R_3(9)$, $R_3(10)$, $R_3(11)$, $R_3(12)$ and $R_3(15)$. From Theorem 1, both of these sets is a distributive lattice.

impossible. Therefore, we have $|f|_i = |f'|_i$ for all i. This leads to the definition of the *shot vector* $s(p)$: $s(p)_i$ is the number of times one have to fire i in order to obtain p from (n). Now we can prove:

Lemma 1 *For all p and q in $R_b(n)$, $p \leq q$ if and only if for all i, $s(p)_i \geq s(q)_i$.*

Proof : If $p \leq q$, i.e. p is reachable from q then it is clear that for all i, $s(p)_i \geq s(q)_i$. Conversely, if there exists i such that $s(p)_i > s(q)_i$, then let j be the smallest such integer. Therefore, $q_j > p_j + b$ and so q can be fired at j. By iterating this process, we finally obtain p, and so $p \leq q$. □

Theorem 1 *For all integers b and n, the order $R_b(n)$ is a* distributive lattice *which contains* all *the b-ary partitions of n, with the infimum and supremum of any two elements p and q defined by:*

$$s(p \vee q)_i = \min(s(p)_i, s(q)_i) \text{ and } s(p \wedge q)_i = \max(s(p)_i, s(q)_i).$$

Proof : We first show that $R_b(n)$ contains all the b-ary partitions of n. Consider p a b-ary partition of n. If $p = (n)$, then $p \in R_b(n)$, so we suppose that $p \neq (n)$. Therefore, there must be an integer $i > 0$ such that $p_i > 0$. Let us define q such that $q_j = p_j$ for all $j \notin \{i-1, i\}$, $q_{i-1} = p_{i-1} + b$ and $q_i = p_i - 1$. It is clear that q is a b-ary partition of n, and that if $q \in R_b(n)$ then $p \in R_b(n)$ since $q \xrightarrow{i-1} p$. It is also obvious that, if we iterate this process, we go back to (n), and so $p \in R_b(n)$.

We now prove the formula for the infimum and the supremum. Let p and q be in $R_b(n)$, and r such that $s(r)_i = \min(s(p)_i, s(q)_i)$. From Lemma 1, p and q are reachable from r. Moreover, if p and q are reachable from $t \in R_b(n)$, then, from Lemma 1, r is reachable from t since we must have $s(t)_i \leq \min(s(p)_i, s(q)_i)$ (else one can not transform t into p or q). Therefore, r is the supremum of p and q, as claimed in the theorem. The argument for the infimum is symmetric. Finally, to prove that the lattice is *distributive*, we only have to check that the formulae satisfy the distributivity laws. □

We will now show that the dynamical model defined here can be viewed as a special Chip Firing Game (CFG). A CFG [BLS91, BL92] is defined over a directed multigraph. A configuration of the game is a repartition of a number of chips over the vertices of the graph, and it obeys the following evolution rule: if a vertex ν contains as many chips as its outgoing degree d, then one can transfer one chip along each of

its outgoing edges. In other words, the number of chips at ν is decreased by d and, for each vertex $v \neq \nu$, the number of chips at v is increased by the number of edges from ν to v. This model is very general and has been introduced in various contexts, such as physics, computer science, economics, and others. It is in particular very close to the famous Abelian Sandpile Model [LP00].

It is known that the set of reachable configurations of such a game, ordered with the reflexive and transitive closure of the transition rule, is a Lower Locally Distributive (LLD) lattice (see [Mon90] for a definition and properties), but it is not distributive in general [BL92, LP00, MPV01]. However, if a lattice is LLD and its dual, i.e. the lattice obtained by reversing the order relation, also is LLD, then the lattice is distributive. Therefore, we can give another proof of the fact that $R_b(n)$ is a distributive lattice by showing that it is the set of reachable configurations of a CFG, and that its dual too [‡].

Given two integers n and b, let us consider the following multigraph $G = (V,E)$ defined by: $V = \{0,\ldots,n\}$ and there are b^{i+1} edges from the i-th vertex to the $(i+1)$-th, for all $n < i \leq 0$. Now, let us consider the CFG C defined over G by the initial configuration where the vertex 0 contains n chips, the other ones being empty. Now, given a configuration c of the CFG, where c_i denotes the number of chips in the vertex number i, let us denote by $\bar{c}$ the vector such that $\bar{c}_i = \frac{c_i}{b^i}$. Then, if the CFG is in the configuration c, an application of the rule to the vertex number i gives the configuration c' such that $c'_i = c_i - b^{i+1}$, $c'_{i+1} = c_{i+1} + b^{i+1}$ and $c'_j = c_j$ for all $j \notin \{i, i+1\}$. Notice that this means exactly that $\bar{c}_i$ is decreased by b and that $\bar{c}_{i+1}$ is increased by 1, therefore an application of the CFG rule corresponds exactly to an application of the evolution rule we defined above, and so the set of reachable configurations of the CFG is isomorphic to $R_b(n)$. This leads to the fact that $R_b(n)$ is a LLD lattice.

Conversely, let G' be the multigraph obtained from G by reversing each edge, and let us consider the CFG C' over G' such that the initial configuration of C' is the final configuration of C. Then it is clear that the set of reachable configurations of C' is nothing but the dual of the one of C, therefore it is isomorphic to the dual of $R_b(n)$. This leads to the fact that the dual of $R_b(n)$ is a LLD lattice, which allows us to conclude that $R_b(n)$ is a distributive lattice.

3 From $R_b(n)$ to $R_b(n+1)$

In this section, we give a method to construct the transitive reduction (i.e. the successor relation) of $R_b(n+1)$ from the one of $R_b(n)$. In the following, we will simply call this the *construction of* $R_b(n+1)$ *from* $R_b(n)$. This will show the self-similarity of these sets, and give a new way, purely structural, to obtain a recursive formula for $|R_b(n)|$, which is previously known from [Rod69] (the special case where $b = 2$ is due to Euler [Eul50]). This construction will also show the special role played by certain b-ary partitions, which will be widely used in the rest of the paper. Therefore, we introduce a few notations about them. We denote by $P_i(b,n)$ the set of the partitions p in $R_b(n)$ such that $p_0 = p_1 = \cdots = p_{i-1} = b-1$. Notice that for all i we have $P_i(b,n) \subseteq P_{i+1}(b,n)$ and that $P_0(b,n) = R_b(n)$. If $p = (p_0,\ldots,p_{k-1})$ is in $P_i(b,n)$, we denote by $p^{\hookrightarrow_i}$ the k-uple $(0,\ldots,0,p_i+1,p_{i+1},\ldots,p_{k-1})$. In other words, $p^{\hookrightarrow_i}$ is obtained from p by switching all the i first components of p from $b-1$ to 0 and adding one unit to its i-th componend [§]. Notice that the k-uple $p^{\hookrightarrow_0}$, which is simply obtained from p by adding one unit to its first component, is always a b-ary partition of $n+1$. If S is a subset of $P_i(b,n)$, we denote by $S^{\hookrightarrow_i}$ the set $\{p^{\hookrightarrow_i} \mid p \in S\}$.

[‡] This idea is due to Clémence Magnien, who introduced this new way to prove that a set is a distributive lattice using two Chip Firing Games.

[§] This operator is known in numeration studies as an odometer. See [GLT95] for more precisions.

Notice that, if $p \xrightarrow{i} q$ in $R_b(n)$, then $p^{\hookrightarrow_0} \xrightarrow{i} q^{\hookrightarrow_0}$ in $R_b(n+1)$. This remark makes it possible to construct $R_b(n+1)$ from $R_b(n)$: the construction procedure starts with the lattice $R_b(n)^{\hookrightarrow_0}$ given by its diagram. Then, we look for those elements in $R_b(n)^{\hookrightarrow_0}$ that have a successor out of $R_b(n)^{\hookrightarrow_0}$. The set of these elements will be denoted by I_0, with $I_0 \subseteq R_b(n)^{\hookrightarrow_0}$. At this point, we add all the missing successors of the elements of I_0. The set of these new elements will be denoted by C_0. Now, we look for the elements in C_0 that have a successor out of the constructed set. The set of these elements is denoted by I_1. More generally, at the i-th step of the procedure we look for the elements in C_{i-1} with missing successors and call I_i the set of these elements. We add the new successors of the elements of I_i and call the set of these new elements C_i. At each step, when we add a new element, we also add its covering relations. Since $R_b(n+1)$ is a finite set, this procedure terminates. At the end, we obtain the whole set $R_b(n+1)$. In the rest of this section, we study more precisely this construction process.

Lemma 2 *Let p be a b-ary partition in $P_i(b,n)$. If $p_i \neq b-1$ then $Succ_b(p^{\hookrightarrow_i}) = Succ_b(p)^{\hookrightarrow_i}$. Else, $Succ_b(p^{\hookrightarrow_i}) = Succ_b(p)^{\hookrightarrow_i} \cup \{p^{\hookrightarrow_{i+1}}\}$.*

Proof : If a transition $p \xrightarrow{j} q$ is possible, then $p^{\hookrightarrow_i} \xrightarrow{j} q^{\hookrightarrow_i}$ is obviously possible. Moreover, an additional transition is possible from $p^{\hookrightarrow_i}$ if and only if $p_i = b-1$. In this case, $p^{\hookrightarrow_i} \xrightarrow{i} p^{\hookrightarrow_{i+1}}$. □

Lemma 3 *For all integer b, n and i, we define the function $r_i : P_i(b,n) \to R_b(\frac{n+1}{b^i} - 1)$ by: $r_i(p)$ is obtained from $p \in P_i(b,n)$ by removing its i first components (which are equal to $b-1$). Then, r_i is a bijection.*

Proof : Let us consider p in $P_i(b,n)$: $p = (b-1, b-1, \ldots, b-1, p_i, \ldots, p_k)$. Then, it is clear that $r_i(p) = (p_i, \ldots, p_k)$ is in $R_b(\frac{n-(b-1)-(b-1)b-\cdots-(b-1)b^{i-1}}{b^i}) = R_b(\frac{n+1-b^i}{b^i}) = R_b(\frac{n+1}{b^{i+1}} - 1)$. Conversely, if we consider p in $R_b(\frac{n+1}{b^i} - 1)$, then $r_i^{-1}(p) = (b-1, b-1, \ldots, b-1, p_0, p_1, \ldots, p_k)$ is a b-ary partition of $m = (b-1) + (b-1)b + \cdots + (b-1)b^{i-1} + \frac{n+1-b^i}{b^i}$, which is nothing but n. Therefore, $r_i^{-1}(p)$ is in $R_b(n)$. □

Lemma 4 *For all integer b, n and i, we have $I_i = P_{i+1}(b,n)^{\hookrightarrow_i}$ and $C_i = P_{i+1}(b,n)^{\hookrightarrow_{i+1}}$.*

Proof : By induction over i. For $i = 0$, it is clear from Lemma 2 that the set of elements in $R_b(n)^{\hookrightarrow_0}$ with a missing successor, namely I_0, is exactly $P_1(b,n)^{\hookrightarrow_0}$. Moreover, the set of these missing successors, namely C_0, is clearly $P_1(b,n)^{\hookrightarrow_1}$. Now, let us suppose that the claim is proved for i and let us prove it for $i+1$. The set I_{i+1} is the set of elements in C_i with one missing successor. By induction hypothesis, we have $C_i = P_{i+1}(b,n)^{\hookrightarrow_{i+1}}$ and so, from Lemma 2, $I_{i+1} = P_{i+2}(b,n)^{\hookrightarrow_{i+1}}$. Then, by application of the evolution rule, it is clear that the set C_{i+1} of the missing successor is $P_{i+2}(b,n)^{\hookrightarrow_{i+2}}$, which proves the claim. □

Theorem 2 *For any positive integer b and n, we have:*

$$R_b(n) = \bigsqcup_{i \geq 0} r_i^{-1}\left(R_b\left(\frac{n}{b^i} - 1\right)\right)^{\hookrightarrow_i}$$

$$|R_b(n)| = \sum_{i=0}^{\lfloor n/b \rfloor} \Big| \; R_b(\tfrac{i}{b}) \; \Big|$$

where $\bigsqcup$ denotes the disjoint union, where $R_b(n)$ is taken as $\emptyset$ when n is not a positive integer, and with $R_b(0) = \{0\}$.

Proof : From the construction procedure described above, we have $R_b(n) = R_b(n-1)^{\hookrightarrow_0} \sqcup \bigsqcup_{i\geq 0} C_i$. From Lemma 4, we obtain $R_b(n) = R_b(n-1)^{\hookrightarrow_0} \sqcup \bigsqcup_{i\geq 0} P_{i+1}(b,n)^{\hookrightarrow_{i+1}}$. Moreover, since $R_b(n-1)^{\hookrightarrow_0}$ is nothing but $P_0(b,n)^{\hookrightarrow_0}$, this is equivalent to $R_b(n) = \bigsqcup_{i\geq 0} P_i(b,n)^{\hookrightarrow_i}$. Finally, from Lemma 3, we obtain the announced formula.

From this formula, we have $R_b(\frac{n}{b}) = \bigsqcup_{i\geq 0} r^{-1}(R_b(\frac{n}{b^{i+1}}-1)^{\hookrightarrow_i})$. Therefore, $|R_b(n)| = \sum_{i\geq 0} |R_b(\frac{n}{b^i} - 1)| = |R_b(n-1)| + \sum_{i\geq 0} |R_b(\frac{n}{b^{i+1}}-1)| = |R_b(n-1)| + |R_b(\frac{n}{b})|$. We obtain the claim by iterating this last formula. □

The first formula given in this theorem can be used to compute the sets $R_b(n)$ efficiently since it only involves *disjoint* unions. We will give in Section 5 another method to compute $R_b(n)$ which is much simplier, as it gives $R_b(n)$ a tree structure. However, the formula is interesting since it points out the self-similar structure of the set (see Figure 4).

The second formula is previouly known from [Rod69], and from [Eul50] in the special case where $b = 2$. Notice that this does not give a way to compute $|R_b(n)|$ in linear time with respect to n, which is an unsolved problem in the general case, but it gives a very simple way to compute recursively $|R_b(n)|$.

4 Infinite extension

$R_b(n)$ is the lattice of the b-ary partitions of n reachable from (n) by iteration of the evolution rule. We now define $R_b(\infty)$ as the set of all b-ary partitions reachable from (∞). The order on $R_b(\infty)$ is the reflexive and transitive closure of the successor relation. For $b = 2$, the first b-ary partitions in $R_b(\infty)$ are given in Figure 2 along with their covering relation (the first component, which is always infinity, is not represented on this diagram). Notice that it is still possible to define the shot vector $s(p)$ of an element p of $R_b(\infty)$ by: $s(p)_i$ is the number of times one has to fire i in order to obtain p from (∞).

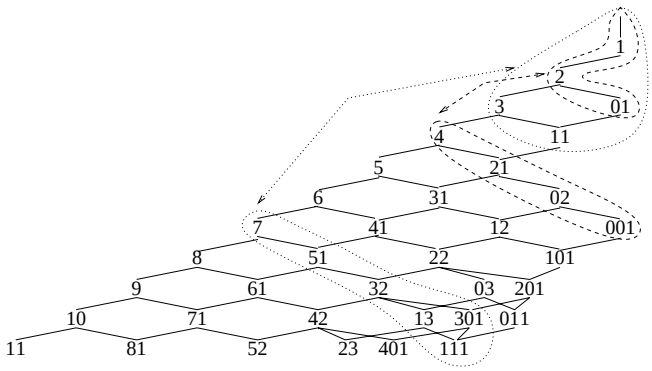


Fig. 2: The first b-ary partitions obtained in $R_b(\infty)$ when $b = 2$. Two parts isomorphic to $R_2(4)$ are distinguished, as well as two parts isomorphic to $R_2(7)$.

Theorem 3 *The set $R_b(\infty)$ is a distributive lattice with:*

$$s(p \vee q)_i = \min(s(p)_i, s(q)_i) \text{ and } s(p \wedge q)_i = \max(s(p)_i, s(q)_i)$$

for all p and q in $R_b(\infty)$. Moreover, for all n the functions

$$\pi : s = (s_1, s_2, \cdots, s_k) \longrightarrow \pi(s) = (\infty, s_2, \ldots, s_k)$$

and

$$\tau : s = (s_1, s_2, \cdots, s_k) \longrightarrow \tau(s) = (\infty, s_1, s_2, \ldots, s_k)$$

are lattice embeddings of $R_b(n)$ into $R_b(\infty)$.

Proof : The proof for the distributive lattice structure and for the formulae of the infimum and supremum is very similar to the proof of Theorem 1. Therefore, it is left to the reader.

Given p and q in $R_b(n)$, we now prove that $\pi(p) \vee \pi(q) = \pi(p \vee q)$. From Theorem 1, we have $s(p \vee q)_i = \min(s(p)_i, s(q)_i)$. Moreover, it is clear that $s(\pi(x))_i = s(x)_i$ for all x in $R_b(n)$. Therefore, $s(\pi(p \vee q))_i = \min(s(\pi(p))_i, s(\pi(q))_i))$, which shows that π preserves the supremum. The proof of $\pi(p) \wedge \pi(q) = \pi(p \wedge q)$ is symmetric. Therefore, π is a lattice embedding.

The proof for τ is very similar when one has noticed that the shot vector of $\tau(s)$ is obtained from the one of s by adding a new first component equal to n. □

With similar arguments, one can easily show that $\pi(R_b(n))$ is a sublattice of $\pi(R_b(n+1))$, and so we have an infinite chain of distributive lattices:

$$\pi(R_b(0)) \leq \pi(R_b(1)) \leq \cdots \leq \pi(R_b(n)) \leq \pi(R_b(n+1)) \leq \cdots \leq R_b(\infty),$$

where $\leq$ denotes the sublattice relation. Moreover, one can use the self-similarity estalished here to construct filters of $R_b(\infty)$ (a *filter* of a poset is an upper closed part of the poset). Indeed, if one defines $R_b(\leq n)$ as the sub-order of $R_b(\infty)$ over $\cup_{i \leq n} R_b(i)$, then one can construct efficiently $R_b(\leq n+1)$ from $R_b(\leq n)$ by extracting from $R_b(\leq n)$ a part isomorphic to $R_b(n+1)$ and pasting it to $R_b(\leq n)$. See Figures 2 and 4.

Notice that, for all integer b, $R_b(\infty)$ contains exactly all the finite sequences of integers, since any such sequence can be viewed as a b-ary partition of an integer n. Therefore, we provide infinitely many ways to give the set of finite sequences of integers the distributive lattice structure.

5 Infinite tree

As shown in our construction of $R_b(n+1)$ from $R_b(n)$, each b-ary partition p in $R_b(n+1)$ is obtained from another one $p' \in R_b(n)$ by application of the $\hookrightarrow$ operator: $p = p'^{\hookrightarrow i}$ with i an integer between 0 and $l(p')$, where $l(p')$ denotes the number of $b-1$ at the beginning of p'. Thus, we can define an infinite tree $T_b(\infty)$ whose nodes are the elements of $\bigsqcup_{n \geq 0} R_b(n)$ and in which the fatherhood relation is defined by:

$$q \text{ is the } (i+1)\text{-th son of } p \text{ if and only if } q = p^{\hookrightarrow i} \text{ for some } i,\ 0 \leq i \leq l(p).$$

The root of this tree is (0) and each node p of $T_b(\infty)$ has $l(p)+1$ sons. The first levels of $T_b(\infty)$ when $b = 2$ are shown in Figure 3 (we call the set of elements of depth n the "level n" of the tree).

Proposition 1 *The level n of $T_b(\infty)$ contains exactly the elements of $R_b(n)$.*

Proof : Straightforward from the construction of $R_b(n+1)$ from $R_b(n)$ given above and the definition of the tree. □

If we define $\overline{R_b(n)}$ as $\{(s_2, \ldots, s_k) \mid (s_1, s_2, \ldots, s_k) \in R_b(n)\}$, then:

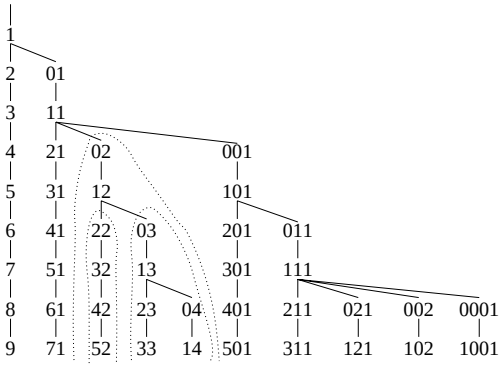


Fig. 3: The first levels of $T_b(\infty)$ when $b = 2$. We distinguished some special subtrees, which will play an important role in the following.

Proposition 2 *For all integer n, the elements of* $\overline{R_b(n)}$ *are exactly the elements of the* $\lfloor \frac{n}{b} \rfloor$ *first levels of* $T_b(\infty)$.

Proof : Let us first prove that the elements of $R_b(n)$ are the nodes of a subtree of $T_b(\infty)$ that contains its root. This is obviously true for $n = 0$. The general case follows by induction, since by construction the elements of $\overline{R_b(n+1)} \setminus \overline{R_b(n)}$ are sons of elements of $\overline{R_b(n)}$.

Now, let us consider an element e of the l-th level of $T_b(\infty)$. If there is a b-ary partition p of n such that $\overline{p} = e$, then clearly $p_i = e_{i-1}$ for all $i > 0$ and $p_0 = n - b \cdot l$. Therefore, if e is in $\overline{R_b(n)}$ then all the elements of the l-th level are in $\overline{R_b(n)}$, and this is clearly the case exactly when $0 \leq l < \lfloor \frac{n}{b} \rfloor$. This ends the proof. □

Notice that this proposition gives a simple way to enumerate the elements of $R_b(n)$ for any n in linear time with respect to their number, since it gives this set a tree structure. Algorithm 1 acheives this.

We will now show that $T_b(\infty)$ can be described recursively, which allows us to give a new recursive formula for $|R_b(n)|$. In order to do this, we will use a series known as the b-ary carry sequence [Slo73]: $c_b(n) = k$ if b^k divides n but b^{k+1} does not. Notice that this function is defined only for $n > 0$ (or one can consider that $c_b(0) = \infty$). These series appear in many contexts, and have many equivalent definitions [¶]. Here, we will mainly use the fact that the first n such that $c_b(n) = k$ is $n = b^k$, and the fact that $c_b(n)$ is nothing but the number of components equal to $b-1$ at the begining of the canonical representation of $n-1$ in the basis b.

Definition 1 *Let* $p \in T_b(\infty)$. *Let us consider the rightmost branch of* $T_b(\infty)$ *rooted at p (p is considered as the first node of the branch). We say that p is the root of a* $X_{b,k}$ subtree (of $T_b(\infty)$) *if this rightmost branch is as follows: for* $i \leq b^{k-1}$, *the i-th node on the branch has* $j = c_b(i) + 1$ *sons, and the l-th* ($1 \leq l < j$) *of these sons is the root of a* $X_{b,l}$ *subtree. Moreover, the* $(b^{k-1}+1)$*-th node of the branch is itself the root of a* $X_{b,k}$ *subtree.*

For example, we show in Figure 3 a $X_{2,2}$ subtree of $T_2(\infty)$, composed of a $X_{2,1}$ subtree and another $X_{2,2}$ subtree. Notice that a $X_{b,1}$ subtree is simply a chain.

[¶] For example, if one defines the series $C_{b,0} = 0$ and $C_{b,i} = C_{b,i-1}, \overbrace{i, C_{b,i-1}}^{b-1 \text{ times}}$, then $c_b(i)$ is nothing but the i-th integer of the series $C_{b,i}$. The ten first values for $c_2(i)$ are $0,1,0,2,0,1,0,3,0,1$ and the ten first ones for $c_3(i)$ are $0,0,1,0,0,1,0,0,2,0$.

Algorithm 1 Efficient enumeration of the elements of $R_b(n)$.

Input: An integer n and a basis b
Output: The elements of $R_b(n)$
begin
  Resu $\leftarrow \{(n)\}$;
  CurrentLevel $\leftarrow\leftarrow \{()\}$;
  OldLevel $\leftarrow \emptyset$; $l \leftarrow 0$;
  while $l < \lfloor \frac{n}{b} \rfloor$ **do**
    OldLevel $\leftarrow$ CurrentLevel;
    CurrentLevel $\leftarrow \emptyset$;
    $l \leftarrow l+1$;
    for each *p in OldLevel* **do**
      $i \leftarrow 0$;
      repeat
        Add $p^{\hookrightarrow i}$ to CurrentLevel;
        $i \leftarrow i+1$;
      until $p_{i-1} \neq b-1$;
    for each *e in CurrentLevel* **do**
      Create p such that $p_i = e_{i-1}$ for all $i > 0$ and $p_0 = n - b \cdot l$;
      Add p to Resu;
  Return(Resu);
end

Proposition 3 *Let $p = (0,0,\dots,0,p_k,\dots)$ in $T_b(\infty)$ with $p_k > b-1$. Then, p is the root of a $X_{b,k+1}$ subtree of $T_b(\infty)$.*

Proof : The proof is by induction over k and the depth of p. Let us consider the rightmost branch rooted at p. Since, for all q in $T_b(\infty)$, the rightmost son of q is $q^{\hookrightarrow_i}$ with i the number of $b-1$ at the beginning of q, it is clear that the j-th node of this branch for $j \leq b^k$ is $q = (q_0,\dots,q_{k-1},p_k,\dots)$ where $(q_0,\dots,q_{k-1})$ is the canonical representation of $j-1$ in the basis b. Therefore, q begins with $c_b(j)$ components equal to $b-1$, and so, for $l = 1,\dots,c_b(j)$, the l-th son of q starts with $l-1$ zeroes followed by a component equal to $b > b-1$. By induction hypothesis, we then have that the sons of q are the roots of $X_{b,l}$ subtrees. Moreover, the (b^k+1)-th node on the rightmost branch begins with exactly k zeroes followed by a component greater than $b-1$, and so it is the root of a $X_{b,k+1}$ subtree by induction hypothesis. □

Theorem 4 *The infinite tree $T_b(\infty)$ is a $X_{b,\infty}$ tree: it is a chain (its rightmost branch) such that its i-th node has $c_b(i)$ sons and the j-th of these sons, $1 \leq j \leq c_b(i)$, is the root of a $X_{b,j}$ subtree. Moreover, the i-th node of the chain is the canonical representation of $i-1$ in the basis b.*

Proof : Since the rightmost son of $p \in T_b(\infty)$ is $p^{\hookrightarrow_i}$, where i is the number of $b-1$ at the beginning of p, and since the root of $T_b(\infty)$ is nothing but the canonical representation of 0, it is clear by induction that the i-th node of the rightmost branch of $T_b(\infty)$ is the canonical representation of $i-1$ in the basis b. Then, the theorem follows from Proposition 3. □

We now have a recursive description of $T_b(\infty)$, which allows us to give recursive formula for the cardinal of some special sets. Let us denote by $\pi_b(l,k)$ the number of paths of length exactly l starting from the root of a $X_{b,k}$ subtree of $T_b(\infty)$. We have:

Theorem 5

$$\pi_b(l,k) = \begin{cases} 1 & \text{if } 0 \leq l < b \\ 1 + \sum_{i=1}^{l} \sum_{j=1}^{c_b(i)} \pi_b(l-i,j) & \text{if } b \leq l \leq b^{k-1} \\ \pi_b(l-b^{k-1},k) + \sum_{i=1}^{b^{k-1}} \sum_{j=1}^{c_b(i)} \pi_b(l-i,j) & \text{otherwise } (l > b^{k-1}) \end{cases}$$

Moreover, $|R_b(n)| = \pi_b(n,n)$ and the number of b-ary partitions of n into exactly l parts is $\pi_b(n-(b-1)^l,l)$.

Proof : The formula for $\pi_b(l,k)$ is directly deduced from the definition of the $X_{b,k}$ subtrees. The other formulae derive from Theorem 4 and from the fact that all the b-ary partitions of length l are in a $X_{b,l}$ subtree of $T_b(\infty)$which is rooted at the $(b-1)^l$-th node of the righmost branch of $T_b(\infty)$. □

6 Perspectives

The results presented in this paper mainly point out the strong self-similarity and the structure of the sets $R_b(n)$. As already noticed, it is an open question to compute the cardinal of $R_b(n)$ in linear time with respect to n, and one may expect to obtain a solution using these results.

Another interesting direction is to investigate how one can extend the dynamics we study. A first idea is to consider non-integer basis, in particular complex basis or Fibonnacci basis. For example, if we consider the complex basis $b = i-1$ then we can obtain all the ways to write an integer n as the sum of powers of b by iterating the following evolution rule from (n): q is a successor of p if $p - q =$

$(0,\dots,0,2,0,-1,-1,0\dots,0)$. In other words, we can decrease by two the j-th component of p and increase by one its $(j+2)$-th and its $(j+3)$-th components for some integer j. This gives to the set of representations of n in the complex basis $b = i-1$ the lattice structure, since this can be encoded by a Chip Firing Game [LP00] (notice however that in this case the lattice is no longer distributive). Another interesting case is when $b = 1$. As already noticed, we obtain the Young lattice, or equivalently the lattice of the compositions of n.

7 Acknowledgments

I thank Christiane Frougny and Clémence Magnien for many useful comments on preliminary versions, which deeply improved the manuscript.

References

[Ber71] Claude Berge. *Principles of Combinatorics*, volume 72 of *Mathematics in science and engineering*. Academic Press, 1971.

[BL92] A. Bjorner and L. Lovász. Chip-firing games on directed graphs. *J. Algebraic Combinatorics*, 1:305–328, 1992.

[BLS91] A. Bjorner, L. Lovász, and W. Shor. Chip-firing games on graphs. *E.J. Combinatorics*, 12:283–291, 1991.

[Chu69] R.F. Churchhouse. Congruence properties of the binary partition function. *Proc. Camb. Phil. Soc*, 66:371–375, 1969.

[Chu71] R.F. Churchhouse. Binary partitions. In A.O.L. Atkin and B.J. Birch, editors, *Computers in Number Theory*, pages 397–400. Academic Press, 1971.

[dB48] N.G. de Bruijn. *Nederl. Akad. Wetensch. Proc.*, 51:659–669, 1948.

[DP90] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Orders*. Cambridge university press, 1990.

[Eul50] L. Euler. *Novi Comm. Petrop.*, III, 1750.

[Fro77] C.-E. Froberg. Accurate estimation of the number of binary partitions. *BIT*, 17:386–391, 1977.

[GLT95] P.J. Grabner, P. Liarded, and R.F. Tichy. Odometers and systems of numeration. *Acta Arithmetica*, LXX.2:103–123, 1995.

[Knu66] D.E. Knuth. An almost linear recurrence. *Fib. Quart.*, 4:117–128, 1966.

[LMMP98] M. Latapy, R. Mantaci, M. Morvan, and H.D. Phan. Structure of some sand piles model. 1998. To appear in Theoretical Computer Science, preprint available at `http://www.liafa.jussieu.fr/~latapy/`.

[LP99] M. Latapy and H.D. Phan. The lattice of integer partitions and its infinite extension. 1999. To appear in DMTCS, special issue, proceedings of ORDAL'99. Preprint available at `http://www.liafa.jussieu.fr/~latapy/`.

[LP00] M. Latapy and H.D. Phan. The lattice structure of chip firing games. 2000. To appear in Physica D. Preprint available at `http://www.liafa.jussieu.fr/~latapy/`.

[Mah40] Kurt Mahler. On a special functional equation. *J. London Math. Soc*, 15:115–123, 1940.

[Mon90] B. Monjardet. The consequences of Dilworth's work on lattices with unique irreducible decompositions. In K.P.Bogart, R.Freese, and J.Kung, editors, *The Dilworth theorems. Selected papers of Robert p. Dilworth*, pages 192–201. Birkhauser, Boston, 1990.

[MPV01] C. Magnien, H.D. Phan, and L. Vuillon. An extension of the chip firing game. 2001. preprint.

[Pen53] W.B. Pennington. On Mahler's partition problem. *Annals of Math.*, 57:531–546, 1953.

[Pfa95] J.L. Pfaltz. Partitions of 2^n. *Congressus Numerantium*, 109:3–12, 1995.

[Rod69] Öystein Rodseth. Some arithmetical properties of m-ary partitions. *Proc. Camb. Phil. Soc*, 68:447–453, 1969.

[Slo73] N.J.A. Sloane. *A Handbook of Integer Sequences*. Academic Press, 1973. On-line version at `http://www.research.att.com/%7Enjas/`.

[Tan18a] A. Tanturri. *Atti R. Acad. Sci. Torino*, 54:69–82, 1918.

[Tan18b] A. Tanturri. *Atti R. Acad. Lincei*, 27:399–403, 1918.

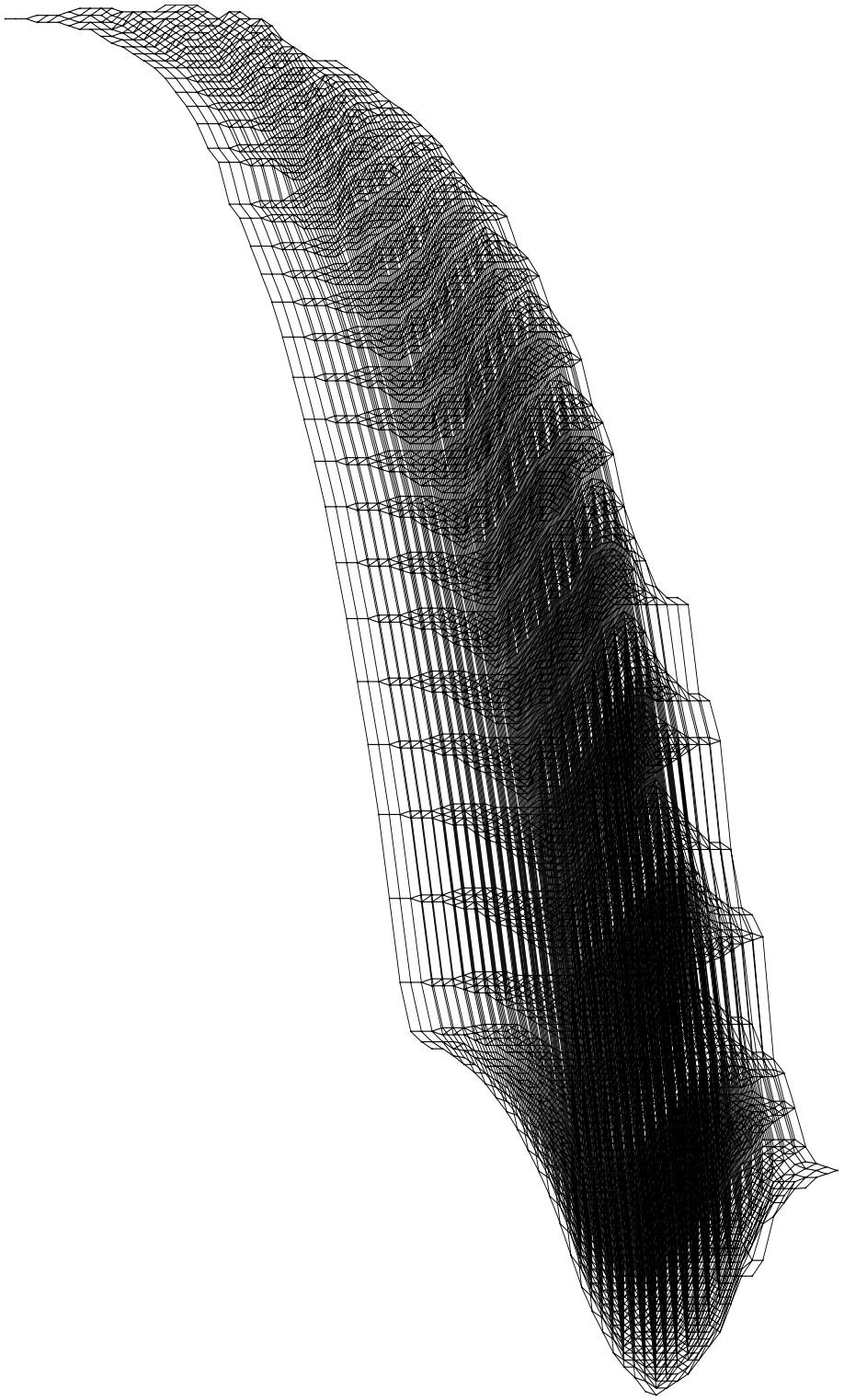


Fig. 4: The distributive lattice $R_2(80)$, which contains 4124 elements and 12484 edges. The self-similarity of the set clearly appears on this diagram.

Random Walks in Octants, and Related Structures

Heinrich Niederhausen
Florida Atlantic University, Boca Raton
Niederhausen@math.fau.edu

March 18, 2003

Abstract

A diffusion walk in $\mathbb{Z}^2$ is a (random) walk with unit step vectors $\rightarrow$, $\uparrow$, $\leftarrow$, and $\downarrow$. Particles from different sources with opposite charges cancel each other when they meet in the lattice. This cancellation principle is applied to enumerate diffusion walks in shifted half-planes, quadrants, and octants (a 3-D version is also considered). Summing over time we calculate expected numbers of visits and first passage probabilities. Comparing those quantities to analytically obtained expressions leads to interesting identities, many of them involving integrals over products of Chebyshev polynomials of the first and second kind. We also explore what the expected number of visits means when the diffusion in an octant is bijectively mapped onto other combinatorial structures, like pairs of non-intersecting Dyck paths, vicious walkers, bicolored Motzkin paths, staircase polygons in the second octant, and $\{\rightarrow\uparrow\}$-paths confined to the third hexadecant enumerated by left turns.

Keywords: Random walks, lattice path enumeration, first passage.
AMS subject classification: Primary 60J15, Secondary 05A15, 05A19

1 Introduction

There are many applications and therefore many names for random walks in the square integer lattice $\mathbb{Z}^2$ with unit step vectors $\rightarrow$, $\uparrow$, $\leftarrow$, and $\downarrow$; we will call them diffusion walks because of the fruitful physical interpretation of the walkers as particles spreading out from a source and being able to interact with particles coming from other sources. We find this "cancellation principle" of particles of opposite charges a better model for enumeration than the frequently applied "reflection principle". For example, let us start *two* diffusion processes at the same time, one from the source $\oplus$ at the origin, and a negatively charged synchronous diffusion from the source $\ominus$ at the "mirror" location $(-2l, 0)$, where l is a positive integer. The diagrams in Table 1 show the location of the sources and the number of ways a particle can reach a lattice point after $k = 1, 2, 3$ steps. The walks from the (virtual) negative source are counted as negative numbers; they annihilate the walks from the positive source when reaching the

boundary line $x = -l$. Thus the boundary is absorbing, and no particle that visits a point to the right of it has ever been to the boundary.

			$x=-l$		$x=0$						$x=-l$		$x=0$										
			■		⋮				-1		■		1			-3		-3	■	3	⋮	3
	-1		■		1			-2		-2	■	2	⋮	2			-9		0		9	
-1	⊖	-1	■	1	⊕	1	-1	···	-4	···	0	···	4	···	1	-9	⊖	-8	■	8	⊕	9
	-1		■		1			-2		-2	■	2	⋮	2			-9		0		9	
			■		⋮				-1		■		1			-3		-3	■	3	⋮	3
			■		⋮						■		⋮				-1		■		1	
$k=1$							$k=2$									$k=3$							

Table 1: Diffusion right of $x = -r$

Only eight such sources, four positive and four negative, are needed to keep the diffusion inside a shifted octant, to the right of $x = -l$ and strictly above $y = x - d$ (see Fig. 2). This will explain why the expression for the number of such diffusion walks from the origin to (n, m) in $m + n + 2k$ steps in the second octant (when $l = d = 1$) is so simple,

$$\frac{(n+1)\,(2+m)\,(m+3+n)\,(m+1-n)}{6\,(n+k+1)} \binom{n+m+2k+2}{n+k} \binom{n+m+2k}{k} / \binom{n+m+k+3}{3}.$$

To show how the complexity increases if the cancellation principle is applied in three dimensions, we solve a generalization of the above problem, the enumeration of 3-D diagonal diffusion with eight step vectors $(\pm 1, \pm 1, \pm 1)$ when the walks stay in the cone $z \geq y \geq x \geq 0$. Instead of 8 we need 48 sources; the formula is given in Subsection 2.3.1, equation (12). A special case of that formula is the number of walks returning to the origin in $2k$ steps,

$$20 C_k C_{k+1} C_{k+2} / \left(\binom{k+5}{3} \binom{k+4}{3} \right) \tag{1}$$

where C_k stands for the k-th Catalan numbers.

There are several statistics on walks in octants that lead to combinations of Catalan numbers, like $C_{\lfloor (k+1)/2 \rfloor} C_{\lfloor 1+k/2 \rfloor}$ in (8), the related $C_k C_{k+1}$ in (10)), and $C_k C_{k+2} - C_{k+1}^2$ in (11). Thus diffusion in an octant is only one of several visualizations of the same (unnamed) underlying combinatorial structure; they all deserve attention, but we will mention only a few in Section 4 on related structures,

- pairs (and triples) of non-intersecting Dyck paths (and three vicious walkers),
- bicolored Motzkin paths,
- staircase polygons in the (augmented) second octant, and
- $\{\rightarrow\uparrow\}$-paths in the (augmented) third hexadecant enumerated by left turns

(omitting Young tableaux [13], and skew Ferrer's diagrams [6]).

The physical approach (diffusion of particles) has a long history; a wealth of results can be found in *Random paths in two and three dimensions* by McCrea and Whipple [23, 1940]. Via

the expected number $\mathrm{E}(n,m)$ of visits to (n,m) they found numerous first passage probabilities for random walks in a rectangle by solving the difference equation $\mathrm{E}(n,m) = \frac{1}{4}(\mathrm{E}(n-1,m) + \mathrm{E}(n+1,m) + \mathrm{E}(n,m-1) + \mathrm{E}(n,m+1))$. The cancellation principle provides easy proofs of (hence) easy problems, using the enumeration of diffusion walks (with a given number of steps) in a half-plane as a building block for more restricted regions like quarter planes, octants, infinite strips, cylinders, rectangles and triangles, passing through formulae with increasing complexity. On the other hand, the analytic method of McCrea and Whipple starts for uniqueness sake with the diffusion in a bounded region like a rectangle, thus begins with the highest level of complexity, simplifying when parts of the boundary are removed. In Section 3 we let the two approaches meet, generating interesting identities. Here are a few examples:

$$\begin{aligned} & 4^{-m-l}\binom{m+l-1}{m}\ {}_4F_3\left[\begin{matrix}\frac{m+l+2}{2}, \frac{m+l+1}{2}, \frac{m+l+1}{2}, \frac{m+l}{2}; \\ m+l+1, l+1, m+1\end{matrix}\ 1\right] \qquad (2) \\ = & \frac{l}{\pi(m+l)}\int_0^\pi \cos((m-l)\theta)\cot^{l+m}\left(\frac{\pi+2\theta}{4}\right)d\theta \\ = & \frac{1}{2\pi}\int_0^\pi \cot^{m+l}\left(\frac{\pi+2\theta}{4}\right)\left(\cos((m-l)\theta) - \frac{\sin((m-l)\theta)}{\cos\theta}\right)d\theta \\ = & \frac{1}{\pi}\int_0^\pi \cos(m\lambda)\left(2-\cos\lambda-\sqrt{(2-\cos\lambda)^2-1}\right)^l d\lambda \end{aligned}$$

(see (21), (22), (23), and (24)), and

$$\begin{aligned} & \sum_{k=0}^\infty \binom{2k+m+l}{k}\binom{2k+m+l+1}{k+1+m}\frac{4^{-2k-m-l}l(m+1)}{(k+m+l+1)(2k+m+l)} \\ = & \frac{2}{\pi}\int_0^\pi \sin(lx)\sin(x)\left(2-\cos x-\sqrt{(2-\cos x)^2-1}\right)^{m+1}dx \end{aligned}$$

(see (29)) for all integers $m \geq 0$, $l \geq 1$, or

$$\begin{aligned} & 6\sum_{k=0}^\infty \frac{4^{-2k-1}}{(k+3)(k+2)}C_{k+1}C_k \\ = & \frac{1}{\pi}\int_0^\pi \frac{1-\sin x}{1+\sin x}\left(\frac{1}{3}(1+5\sin(x)) + \frac{(1-\sin x)^2\left(1+4\sin^2 x\right)}{5(1+\sin x)}\right)dx \end{aligned}$$

(the case $m = 0$ in (31)).

We did not derive these identities for the purpose of actual computations. In the process of numerically verifying the formulas, however, one notices that the sums are slowly converging. Only if the number of oscillations in the integrands gets very large, the numerical algorithms for evaluating the integrals can fail to produce a result.

A Few Historical Notes. It is the intention of this paper to show how far the cancellation principle can carry us with just a finite number of sources, and how little effort is required to obtain those beautiful results. However, an *infinite* number of sources is needed if we restrict the diffusion to bands with parallel boundaries, and subsets thereof. Another example requiring infinitely many sources is the $\{\leftarrow, \downarrow, \nearrow\}$-walk in the first quadrant, representing the ballot problem with three candidates where the winner ($\nearrow$) never falls behind the losers ($\leftarrow, \downarrow$) during the counting of the votes (Kreweras [19]). There is a wealth of approaches to the enumeration of walks bounded by hyperplanes, some of them attacking the problem (including all those with binomial results in this paper) from a very general angle [10],[2], or considering different kinds of boundary conditions and step sets [28]. In a recent paper, Bousquet-Mélou (2002) applies the *kernel method* to "Counting Walks in the Quarter Plane" [3]. Of course, these few references cannot even scratch the surface of the mountain of literature that has accumulated on the topic of planar walks; the situation gets worse when we discuss related structures in Section 19. Some references can be found in Janse van Rensburg's book [21], and a few others are interspersed among the results in Section 19. Mohanty's book on "Lattice Path Counting and Applications" [25] is still a valuable resource for a first introduction to that topic.

Acknowledgement This work began with the quest for a simpler proof of a much harder problem, the enumeration of diffusion walks in the second octant, conditioned on the number of visits to the diagonal[1]. For an analytic approach to this question see Janse van Rensburg [21]. I am also indebted to Y. Itoh for drawing my attention to the paper by McCrea and Whipple [23], and for the interpretation of diffusion in an octant as a gamblers' ruin problem. M.E.H. Ismail pointed out the connection to Chebyshev polynomials, which helps to "explain" some of the identities, and A.J. Guttmann showed me how to enumerate polyominoes by gap size. Finally, most of the references have been provided by one of the referees.

2 Restricted Diffusion

If the diffusion has only one source $\oplus$, at the origin, say, and no restrictions, then the number $U_k(n,m)$ of ways a particle can reach the point (n,m) in k steps is

$$U_k(n,m) = \binom{k}{\frac{k+n+m}{2}} \binom{k}{\frac{k+n-m}{2}}. \tag{3}$$

This is of course well-known; for a proof by picture see Fig. 3. No particle can reach (n,m) in k steps if $k+n+m$ is odd, or $|n|+|m|>k$; we must interpret the binomial coefficient $\binom{i}{j}$ as 0 if i or j are fractions or negative integers. Note the four axes of symmetry in diffusion walks: the x-axis, y-axis, and the diagonals $y=\pm x$. Thus

$$U_k(n,m) = U_k(|n|,|m|) = U_k(|m|,|n|).$$

[1] Since the completion of this paper, I have been able to prove by very different and much less elegant methods [26] that the number of such walks from the origin to (n,n) in $2k$ steps making d contacts with the diagonal equals $\frac{\binom{2k+2}{k+n+2}\binom{2k}{k}}{\binom{2k+1}{d+1}2(k+1)^2}\left((n+1)(d-1)\binom{k}{d-1}+\frac{(2k+1-d)(2n+d+2)}{d+1}\left(\binom{k-n}{d}-\binom{k}{d}+n\binom{k}{d-1}\right)\right)$.

Stirling approximation shows that for large k the walk ends at (n,m) after $2k+|n|+|m|$ steps with probability approximately $(\pi k)^{-1}$. Hence the expected number of visits to (n,m) is infinite.

We begin with a review of well known results in the enumeration of diffusion walks restricted to half- and quarter-planes. The pictures we show are not proofs in the strict sense; they are a suggestive "physical interpretation", based on the cancellation principle. However, the answers they suggest can be easily verified by checking the recursion and initial values. Another iteration of the "method of images" or cancellation principle leads from walks in quadrants to octants in Subsection 2.3. A more algebraic than geometric way of applying the cancellation principle is shown in Subsection 2.3.1.

2.1 Half-planes

Suppose l is a positive integer. As a prototype of diffusion restricted to half of the lattice we count the walks strictly to the right of the *l*eft boundary $x=-l$. We start *two* diffusion processes at the same time, one from the source $\oplus$ at the origin, and a synchronous negatively charged diffusion from the source $\ominus$ at the mirror location $(-2l,0)$. The diagrams in Table 1 show the location of the sources and the number of walks after $k=1,2,3$ steps. The walks from the (virtual) negative source are counted as negative numbers; they annihilate the walks from the positive source when reaching the boundary line $x=-l$. Thus the number $H_k^{l|}(n,m)$ of walks in a *H*alf plane to (n,m) from the origin in $k\geq|n|+|m|$ steps strictly to the right of the line $x=-l$ is

$$\begin{aligned} H_k^{l|}(n,m) &= U_k(n,m)-U_k(n+2l,m) \\ &= \binom{k}{\frac{k+n+m}{2}}\binom{k}{\frac{k+n-m}{2}}-\binom{k}{\frac{k+n+m}{2}+l}\binom{k}{\frac{k+n-m}{2}+l}. \end{aligned} \tag{4}$$

The case $l=1$ shows that for $n\geq 0$

$$H_{2k+n+|m|}^{1|}(n,m)=\frac{n+1}{2k+n+|m|+1}\binom{2k+n+|m|+1}{k}\binom{2k+n+|m|+1}{k+|m|}$$

walks reach (n,m) in $2k+n+|m|$ steps staying strictly in the right half plane.

Denote the number of walks to (n,m) strictly left of $x=r$ by $H_k^{|r}(n,m)$. By symmetry, $H_k^{|r}(n,m)=H_k^{r|}(-n,m)$. For more results on two-dimensional random walks in general see Csáki [4]; for diffusion in a quadrant Guy, Krattenthaler and Sagan [15], and for planar walks inside a rectangle [27].

2.2 Quadrants

If we want the diffusion to stay in the shifted first quadrant (a *quadrant walk*) strictly above the bottom line $y=-b$ and right of $x=-l$ we only have to study the scheme in Fig. 1.

One negative source $\mathfrak{H}$ of a virtual walk in the half-plane $x\geq -l$ is needed to cancel along $y=-b$ the same type of half-plane walk from the origin. Let $n>-l$ and $m>-b$. Thus

$$Q_k^{l\llcorner b}(n,m):=H_k^{l|}(n,m)-H_k^{l|}(n,m+2b)$$

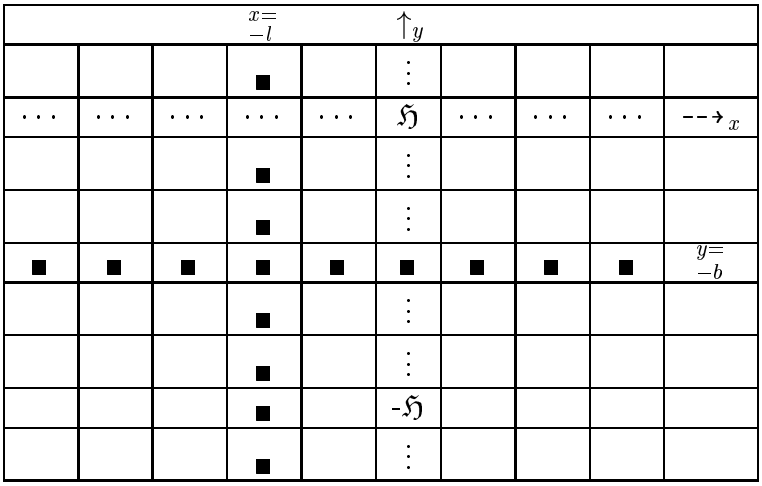


Figure 1: Diffusion in a shifted quadrant

is the number of Quadrant walks from the origin to (n,m) in k steps. Note that $Q_k^{l\llcorner b}(n,m) = Q_k^{b\llcorner l}(-m,-n)$.

The diagram also shows that $Q_k^{l\ulcorner b}(n,m) := H_k^{l|}(n,m) - H_k^{l|}(n,m-2b)$ where $Q^{l\ulcorner b}$ enumerates fourth-quadrant walks strictly right of $x=-l$ and below $y=b$. Note that for $m>0$

$$Q_k^{l\llcorner b}(n,m-b) = H_k^{l|}(n,b-m) - H_k^{l|}(n,-m-b) = Q_k^{l\ulcorner b}(n,b-m)\,.$$

For $l=b=1$ we obtain the number $\frac{(n+1)(m+1)}{(k+n+m+1)(k+n+m+2)}\binom{2k+n+m}{k}\binom{2k+n+m+2}{k+n+1}$ of quadrant walks to $(n,m)\in\mathbb{N}_0^2$ in $2k+n+m$ steps, applying (4)

$$\begin{aligned}
& H^{1|}_{2k+n+m}(n,m) - H^{1|}_{2k+n+m}(n,m+2)\\
= \;& \frac{n+1}{2k+n+m+1}\binom{2k+n+m+1}{k}\binom{2k+n+m+1}{k+m}\\
& -\frac{n+1}{2k+n+m+1}\binom{2k+n+m+1}{k-1}\binom{2k+n+m+1}{k+m+1}\\
= \;& \frac{(n+1)(m+1)}{(k+n+m+1)(k+n+m+2)}\binom{2k+n+m}{k}\binom{2k+n+m+2}{k+n+1}.
\end{aligned}$$

If $Q_k^{l\lrcorner b}(n,m)$ are the paths staying in the shifted second quadrant strictly above the line $y=-b$ and *left* of $x=l$ then $Q_k^{l\lrcorner b}(n,m) = Q_k^{l\llcorner b}(-n,m)$.

2.3 Octants

Are there any more results on bounded diffusion that are as beautiful and surprisingly simple as the diffusion in a quadrant, where the two positive sources exactly cancel the two negative sources at the right places? The answer is yes, because diffusion has another axis of symmetry that we can utilize, the first diagonal. Thus there is at least one more "nice" case, the diffusion inside an octant. Diffusion walks in an octant may be seen as the difference of two quadrant

walks (originating at $\mathfrak{Q}$ and -$\mathfrak{Q}$ on the right side of Fig. 2), or as the sum of an array of alternating unrestricted walks arranged along the corners of an octagon (left side of Fig. 2).

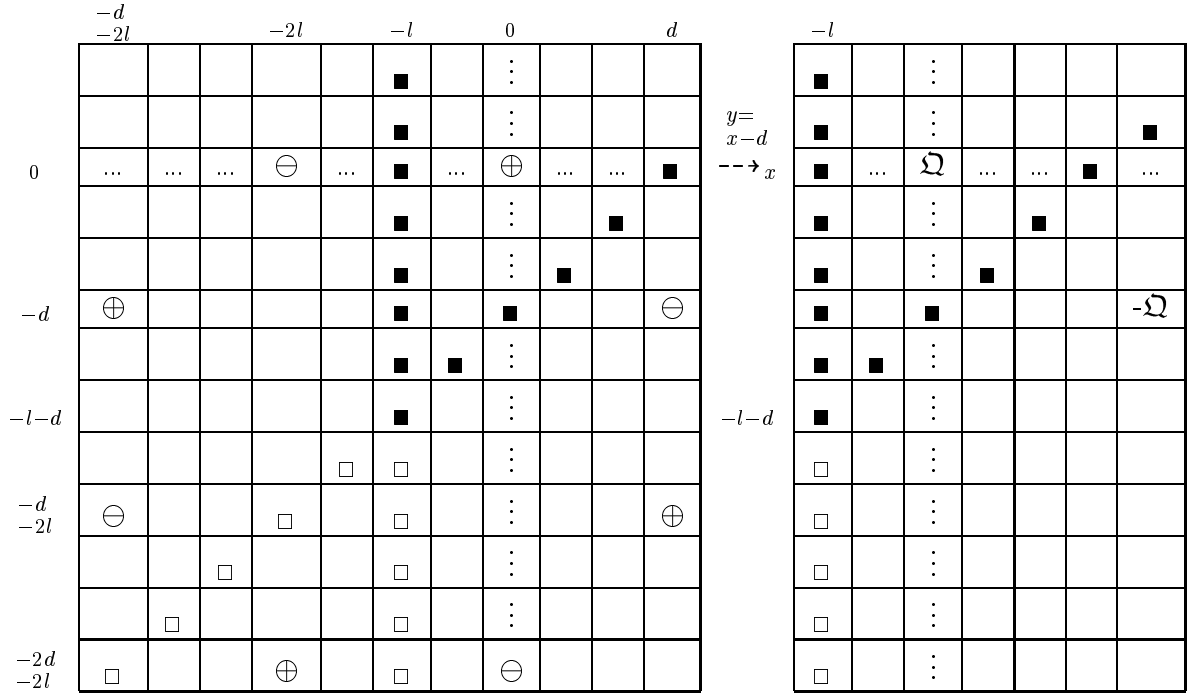


Figure 2: Only 7 virtual sources are needed to keep the diffusion in an octant!

In order to respect the boundary $x = -l$ and enable cancellation along $y = x - d$, the positive $\mathfrak{Q}$ must be $\mathfrak{Q}^{l\llcorner l+d}$, and the negative quadrant source must be $\mathfrak{Q}^{l+d\llcorner l}$.

Let $n > -l$ and $m > n - d$ for two given positive integers l and d. The number of paths from the origin to (n, m) in k steps strictly to the right of $x = -l$ and strictly above $y = x - d$ equals $O_k^{l|\!\diagup d}(n, m)$

$$\begin{aligned} &= U_k(n,m) - U_k(n+2l,m) - U_k(n-d,m+d) + U_k(n+d+2l,m+d) \\ &\quad + U_k(n-d,m+d+2l) - U_k(n+d+2l,m+d+2l) \\ &\quad - U_k(n,m+2d+2l) + U_k(n+2l,m+2d+2l) \\ &= H_k^{l|}(n,m) - H_k^{l|}(m+d,n-d) + H_k^{l|}(m+d,n+d+2l) - H_k^{l|}(n,m+2d+2l) \\ &= Q_k^{l\llcorner l+d}(n,m) - Q_k^{l\llcorner l+d}(m+d,n-d). \end{aligned} \tag{5}$$

If we extend formula (5) for $O_k^{l|\!\diagup d}(n, m)$ to *all* lattice points (n, m) we note that $O_k^{l|\!\diagup d}(n-l, m) = -O_k^{l|\!\diagup d}(-n-l, m)$, and $O_k^{l|\!\diagup d}(n, m) = -O_k^{l|\!\diagup d}(m+d, n-d)$.

Remark 1 *Diffusion in the second octant is related to a ruin problem where two players called E.W. and S.N. play, in random order against a bank. Player E.W. has a capital of l dollars, and the bank holds d dollars; player S.N. is of unlimited wealth in this version, and cannot be ruined. In every game the players either win or lose a dollar; the associated diffusion walk takes a step*

to the East, $\rightarrow$, if E.W. wins, to the West, $\leftarrow$, if E.W. loses,
to the South, $\downarrow$, if S.N. wins, and to the North, $\uparrow$, if S.N. loses.

Player E.W. is ruined when his capital is down to zero; the same holds for the bank. Thus $O_k^{l\diagup d}(1-l,m)$ is the number of ways gambler E.W. can get ruined in $k+1$ games when player S.N. has a gain (or loss) of m dollars. The banker can get ruined in $O_k^{l\diagup d}(n,n-d+1)+O_k^{l\diagup d}(n-1,n-d)$ ways in $k+1$ games when player S.N. has a loss (or gain) of n dollars, and E.W. has a gain (or loss) of $n-d$ dollars. Ruin probabilities can be obtained from the first passage probabilities in Subsection 3.3.

If player S.N. has limited capital a we must restrict the walk to the right triangle $-l<x<y-d$, $y<a$. It needs an infinite array of virtual octant walks to keep the diffusion inside that triangle. A more efficient approach starts with McCrea and Whipple's formula [23] for diffusion restricted to a rectangle, and views the triangle walks as the difference of to rectangular diffusions; see [20] for the corresponding ruin problems.

For walks in the second octant ($l=d=1$) we drop the superscript $l\diagup d$ from the notation. The number of such walks from the origin to (n,m) in $m+n+2k$ steps is for $m\geq n\geq 0$

$$\begin{aligned} & O_{n+m+2k}(n,m) \\ = & \left(\binom{n+m+2k}{k}-\binom{n+m+2k}{k-3}\right)\left(\binom{n+m+2k}{n+k}-\binom{n+m+2k}{n+k-1}\right) \\ & +\left(\binom{n+m+2k}{k-2}-\binom{n+m+2k}{k-1}\right)\left(\binom{n+m+2k}{n+k+1}-\binom{n+m+2k}{n+k-2}\right) \\ = & \frac{(n+1)(2+m)(m+3+n)(m+1-n)}{6(n+k+1)\binom{n+m+k+3}{3}}\binom{n+m+2k+2}{n+k}\binom{n+m+2k}{k} \end{aligned} \tag{6}$$

(there is a printing error in the corresponding formula (4.186) in [21]).

The number of paths in the second octant *ending on the y-axis* at height $m\geq 0$ in $m+2k$ steps is therefore

$$O_{m+2k}(0,m)=\frac{1}{k+1}\binom{m+3}{3}\binom{m+2k+2}{k}\binom{m+2k}{k}\Big/\binom{m+k+3}{3}. \tag{7}$$

Summing over the end point gives the number of walks in the second octant ending on the y-axis after k steps,

$$\sum_{j=0}^{k/2}\frac{1}{j+1}\binom{k-2j+3}{3}\binom{k+2}{j}\binom{k}{j}\Big/\binom{k-j+3}{3}=C_{\lfloor (k+1)/2\rfloor}C_{\lfloor 1+k/2\rfloor} \tag{8}$$

(sequence A005817 in the *On-Line Encyclopedia of Integer Sequences*), where $C_k=\binom{2k}{k}/(k+1)$ is the k-th Catalan number. To the diagonal, at (n,n), will return

$$O_{2n+2k}(n,n)=\frac{1}{4(n+k+1)}\binom{2n+4}{3}\binom{2n+2k+2}{n+k}\binom{2n+2k}{k}\Big/\binom{2n+k+3}{3} \tag{9}$$

paths after $2n+2k$ steps. This time summing over the end point gives the number of walks in the second octant ending on the diagonal after $2k$ steps,

$$\sum_{j=0}^{k} \frac{1}{4\,(k+1)} \binom{2k-2j+4}{3} \binom{2k+2}{k} \binom{2k}{j} \Big/ \binom{2k-j+3}{3} = C_k C_{k+1} \tag{10}$$

(sequence A005568). To the origin will return

$$O_{2k}\,(0,0) = 12 \frac{(2k)!\,(2k+1)!}{k!^2\,(k+3)!\,(k+2)!} = C_k C_{k+2} - C_{k+1}^2 \tag{11}$$

walks after $2k$ steps. More on the Catalan and other connections in Section 4.

Best suited for numerical experiments with planar walks are spreadsheets, but matrices are also a useful tool for displaying the effect of virtual sources. Start with a finite piece of the double infinite matrix $R = (R_{ij})_{i,j\in\mathbb{Z}}$ where $R_{ij} = 1$ if $|i+j| = 1$, and 0 else. This matrix represents the recursion in the sense that $O_{k+1} = RO_k + O_k R$. The initial matrix O_0 has ones and minus ones at the position of the sources. Table 2 shows an example for the case $l = d = 1$.

0	0	0	0	0	−1	**0**	1	0	0	0	0	**0**
0	0	0	0	−3	0	**0**	0	3	0	0	**0**	0
0	0	0	−2	0	−6	**0**	6	0	2	**0**	0	0
0	0	2	0	−5	0	**0**	0	5	**0**	−2	0	0
0	3	0	5	0	−3	**0**	3	**0**	−5	0	−3	0
1	0	6	0	3	0	**0**	**0**	−3	0	−6	0	−1
0	0	0	0	0	0	**0**	0	0	0	0	0	0
−1	0	−6	0	−3	**0**	**0**	0	3	0	6	0	1
0	−3	0	−5	**0**	3	**0**	−3	0	5	0	3	0
0	0	−2	**0**	5	0	**0**	0	−5	0	2	0	0
0	0	**0**	2	0	6	**0**	−6	0	−2	0	0	0
0	**0**	0	0	3	0	**0**	0	−3	0	0	0	0

Table 2: A piece of the matrix O_4, with boldface boundary values

2.3.1 Three dimensional diffusion

The same cancellation principle can be applied to solve 3-D diffusion problems. However, it is much harder to place the sources in space just by geometrical intuition. For example, consider the 3-D diagonal diffusion with eight step vectors $(\pm 1, \pm 1, \pm 1)$, and require that the walks (weakly) stay in the cone $z \geq y \geq x \geq 0$. Denote by $D_k\,(n,m,l)$ the number of walks from the origin to (n,m,l) in k steps. If n, m, l, and k are not of the same parity, this number will be zero. Thus the bounding planes to be avoided by the walk are of the form $x = -1$, $y = x - 2$, and $z = y - 2$. We derive the location of the sources by an algebraic instead of a visual argument.

Let S be the set of sources. If (a,b,c) is a source, then its effect on the boundaries $x=-1$, $y=x-2$, and $z=y-2$ must be canceled by the opposite sources at $f(a,b,c):=(-a-2,b,c)$, $g(a,b,c):=(b+2,a-2,c)$, and $h(a,b,c):=(a,c+2,b-2)$, respectively. Denote by C the noncommutative group generated by the three reflections f, g, and h. Hence $(a,b,c)\in S$ implies $p(a,b,c)\in S$ for all $p\in C$. In other words, $S=\{p(0,0,0):p\in C\}$. For a better understanding of S and C we temporarily move the origin into the intersection of the three planes, so that $(0,0,0)\mapsto(1,3,5)$. The three reflections are now $f'(a,b,c)=(-a,b,c)$, $g'(a,b,c)=(b,a,c)$, and $h'(a,b,c)=(a,c,b)$. Correspondingly, C' is the group generated by f', g' and h', and $S'=\{p'(1,3,5):p'\in C'\}$. Note that g' and h' are transpositions on three elements. In cycle notation, $g'=(a,b),(c)$ and $h'=(a),(b,c)$. The third transposition can be obtained as $h'g'h'=(a,c)(b)$. Hence g' and h' generate the group of all 3-permutations, $\mathfrak{S}_3$, which implies that any permutation of $(1,3,5)$ is in S'. The reflection f' changes the sign of the first coordinate, $g'f'g'$ the sign of the second, and $h'g'f'g'h'$ the sign of the third. Hence the 48 elements of S' are the permutations of $(\pm1,\pm3,\pm5)$. The group C' is well known as the group generated by the symmetries of the cube (the hyperoctahedral group on signed 3-permutations). Every element p' of C' can be written as a composition of transpositions $(g',h',h'g'h')$ and some sign changes induced by f'. The parity of the number of transpositions in p' is an invariant, and so is the parity of the number of sign changes f'. We call p' even or odd depending on the parity of the number of transpositions plus sign changes. An even (odd) p' can only be written as a composition of an even (odd) number of reflections. Now we return to our set of sources, $S=\{p'(1,3,5)-(1,3,5):p'\in C'\}$. A source $s=p'(1,3,5)-(1,3,5)$ is positive iff p' is even. We have shown the following lemma.

Lemma 2 *Let $P:=\{(\pm u,\pm v,\pm w)$, where (u,v,w) is a permutation of $(1,3,5)\}$. Then $\{(-1,-3,-5)+(x,y,z):(x,y,z)\in P\}$ is the set of sources. The sources $(-1,-3,-5)+(x,y,z)$ and $(-1,-3,-5)+\left((-1)^i x,(-1)^j y,(-1)^k z\right)$ have the same sign iff $i+j+k$ is even.*

The 3-D version of Fig.3 would show that unrestricted diagonal diffusion in three dimensions is generated by three independent random walks along the three coordinate axis. The number of unrestricted walks to (n,m,l) in k steps is therefore $U_k(n,m,l):=\binom{k}{(k+n)/2}\binom{k}{(k+m)/2}\binom{k}{(k+l)/2}$. After adding up the 48 unrestricted walks starting at the 48 sources in S with the appropriate signs we arrive at $D_k(n,m,l)$. It follows from the above Lemma that

$$D_k(n,m,l)=\sum_{i=0}^{7}(-1)^{\lfloor i/4\rfloor+\lfloor i/2\rfloor+i}\begin{vmatrix}\binom{k}{\frac{k+n+1-(-1)^{\lfloor i/4\rfloor}}{2}} & \binom{k}{\frac{k+n+1-3(-1)^{\lfloor i/2\rfloor}}{2}} & \binom{k}{\frac{k+n+1-5(-1)^{i}}{2}}\\ \binom{k}{\frac{k+m+3-(-1)^{\lfloor i/4\rfloor}}{2}} & \binom{k}{\frac{k+m+3-3(-1)^{\lfloor i/2\rfloor}}{2}} & \binom{k}{\frac{k+m+3-5(-1)^{i}}{2}}\\ \binom{k}{\frac{k+l+5-(-1)^{\lfloor i/4\rfloor}}{2}} & \binom{k}{\frac{k+l+5-3(-1)^{\lfloor i/2\rfloor}}{2}} & \binom{k}{\frac{k+l+5-5(-1)^{i}}{2}}\end{vmatrix}.$$

To shorten the expansion, we define $B_t^i:=\binom{k}{(k+i)/2}-\binom{k}{t+(k+i)/2}$ and find $D_k(n,m,l)=$

$$B_1^n\left(B_3^mB_5^l-B_5^{m-2}B_3^{l+2}\right)+B_1^{m+2}\left(B_5^{n-4}B_3^{l+2}-B_3^{n-2}B_5^l\right)+B_1^{l+4}\left(B_3^{n-2}B_5^{m-2}-B_5^{n-4}B_3^m\right)\quad(12)$$

The special case $D_{2k}(0,0,0)$ gives formula (1).

We chose the example of an eight-step diagonal diffusion in view of an application to counting watermelons at the end of Section 4.1. The more common "nearest neighbor walks" with six

steps $(\pm 1,0,0)$, $(0,\pm 1,0)$, and $(0,0,\pm 1)$ can be enumerated in the same way, with boundaries $x=-1$, $y=x-1$, and $z=y-1$, and corresponding sources at $(a,b,c)-(1,2,3)$, where (a,b,c) is a permutation of $(\pm 1,\pm 2,\pm 3)$. The same approach solves the boundary problem that restricts the walks to the region

$$x>-b,\ y>x-c,\ z>y-d$$

where b, c, and d are positive integers. If the six nearest neighbor steps are reduced to the three unit steps $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$, we obtain the more familiar "ballot problem with three candidates". The condition $x>-b$ is automatic in this case, thus there are only six sources, $(0,0,0)$, $(c,-c,0)$, $(c,d,-c-d)$, $(c+d,0,-c-d)$, $(c+d,-c,-d)$, and $(0,d,-d)$. In terms of trinomial coefficients, the well known number of ballot paths to (n,m,l) is $\binom{l+n+m}{n,m}-\binom{l+n+m}{n-c,m+c}+\binom{l+n+m}{n-c,m-d}-\binom{l+n+m}{n-c-d,m}+\binom{l+n+m}{n-c-d,m+c}-\binom{l+n+m}{n,m-d}$ which simplifies to $\frac{(m-n+1)(l+1-m)(l+2-n)}{(l+1)(l+2)(m+1)}\binom{l+n+m}{n,m}$ if $c=d=1$. In this "totally ordered" version of the ballot problem, the winner stays ahead of the second winner, who himself remains ahead of the looser throughout the counting of votes. Even with only three candidates the ballot problem becomes much more difficult if the boundary $z\geq y\geq x$ is replaced by $z\geq \max(x,y)$, which is the version mentioned in the Introduction (see [19],[18], and for the latest proof [3]). Lemma 2 is easily generalized to any dimension d. If we require that the walk stays in the chamber $x_i>x_{i-1}-c_i$ for all $i=1,\ldots,d$ (with $x_{-1}=0$, $c_i\geq 1$), then the sources must be placed at $\mathbf{p}-\mathbf{c}$, where $\mathbf{c}=(c_1,c_1+c_2,\ldots,c_1+c_2+\cdots+c_d)$, and $\mathbf{p}\in P:=\{(\pm u_1,\pm u_2,\ldots,\pm u_d)$, where $(u_1,u_2,\ldots,u_d)$ is a permutation of the components of $\mathbf{c}\}$. The source is positive if the permutation is even, and negative else. Note that the location of sources is independent of the step set of the walk. Of course, the admissible step sets must be symmetric with respect to all the bounding hyperplanes. The resulting solution to the totally ordered ballot problem was first solved by Frobenius and MacMahon [24]. For general $c_i's$ see [31]. For more on the general problem see [9],[10], and [14].

3 Expected Number of Visits and First Passage

Diffusion is not only a physical concept, modeled in combinatorics by discrete time and a lattice of discrete states, it is also an intensively studied area of probability theory. We compare in this section some probabilistic results on the long term behavior of diffusion to the corresponding expressions obtained from combinatorial enumeration. Most of the identities that are obtained this way seem to be hard to prove by other methods.

Denote by $V_{n,m}$ the random variable that reports the number of visits a random diffusion walk makes to (n,m) before being absorbed at some boundary, thus the expected number of visits to (n,m) equals

$$\mathrm{E}\left[V_{n,m}\right]=\sum_{j\geq 0}\Pr\left(V_{n,m}\geq j\right)=\sum_{k\geq 0}4^{-k}D_k\left(n,m\right) \tag{13}$$

if $D_k(n,m)$ is the number of paths to (n,m) in k steps under the same restrictions. Without any restrictions on the paths, the expectation is infinite. The enumeration results of Section

2 enable us to express the expected number of visits in half planes, quadrants and octants as sums; for half-planes and quadrants they also have been expressed as integrals in a paper by McCrea and Whipple [23] as limiting cases of planar walks in a rectangle. However, those integrals look different from the obvious integrals (16) obtained from the combinatorial sums!

3.1 Half-planes

Denote by $\mathrm{E}_H^{l|}[V_{n,m}]$ the expected number of visits to the point (n,m) of a random diffusion walk in the half-plane $x > -l$. Because of symmetry, $\mathrm{E}_H^{l|}[V_{n,m}] = \mathrm{E}_H^{l|}[V_{n,-m}]$, and the same holds for the first passage probability. All formulae in this subsection will be written for $m \geq 0$; every occurrence of m can be replaced by $|m|$. We find four formulas for $\mathrm{E}_H^{l|}[V_{n,m}]$: (14), (16), (19), and (20). We begin with an expression for $\mathrm{E}_H^{l|}[V_{n,m}]$ derived from (13) and (4)

$$\mathrm{E}_H^{l|}[V_{n,m}] = \sum_{k\geq 0} 4^{-k}\left(U_k(n,m) - U_k(n+2l,m)\right) \tag{14}$$

This sum can be written as an integral using the identities

$$\begin{aligned} \int_0^\pi \cos(mx)(\cos x)^n\,dx &= \begin{cases} \frac{\pi}{2^n}\binom{n}{(n-m)/2} & \text{if } n\geq m\geq 0 \text{ and } n-m \text{ is even} \\ 0 & \text{else} \end{cases} \\ \sum_{n=0}^{\infty}\binom{2n+x}{n}\xi^n &= \frac{2^x}{\sqrt{1-4\xi}}\left(1+\sqrt{1-4\xi}\right)^{-x}. \end{aligned} \tag{15}$$

We get for all integers n

$$\begin{aligned} 4^{-2k-|n|-m}U_{|n|+m+2k}(n,m) &= \frac{\binom{|n|+m+2k}{k}}{2^{|n|+m+2k}\pi}\int_0^\pi \cos\left((m-|n|)x\right)\cos^{|n|+m+2k}(x)\,dx \\ \sum_{k\geq 0}(\xi/4)^{2k+|n|+m}U_{|n|+m+2k}(n,m) &= \frac{1}{\pi}\int_0^\pi\left(\frac{\cos(x)}{1+\sqrt{1-\xi\cos(x)^2}}\right)^{|n|+m}\frac{\cos\left((m-|n|)x\right)}{\sqrt{1-\xi\cos(x)^2}}dx \end{aligned}$$

This expansion converges for $|\xi|<1$, diverges for $\xi=1$, but converges for $\xi=-1$. The power series

$\sum_{k\geq 0}(\xi/4)^k\left(U_k(n,m)-U_k(n+2l,m)\right)$

$= \int_0^\pi \frac{\left(\frac{\cos(x)}{1+\sqrt{1-\xi\cos(x)^2}}\right)^{|n|+m}\cos((m-|n|)x) - \left(\frac{\cos(x)}{1+\sqrt{1-\xi\cos(x)^2}}\right)^{n+2l+m}\cos((m-n-2l)x)}{\pi\sqrt{1-\xi\cos(x)^2}}dx$

converges for $\xi=1$, thus $\mathrm{E}_H^{l|}[V_{n,m}] =$

$$\int_0^\pi \frac{\left(\cot\left(\frac{\pi+2x}{4}\right)^{|n|+m}\cos\left((m-|n|)x\right) - \cot\left(\frac{\pi+2x}{4}\right)^{n+2l+m}\cos\left((m-n-2l)x\right)\right)}{\pi\sin(x)}dx \tag{16}$$

for $n \geq -l$.

Remark 3 *From* $U_k(n,m) = U_k(m,n)$ *follows*
$\mathrm{E}_H^{l|}[V_{n,m}] = \sum_{k\geq 0} 4^{-k}\left(U_k(n,m) - U_k(n+2l,m)\right) = \sum_{k\geq 0} 4^{-k}\left(U_k(m,n) - U_k(m,n+2l)\right)$.

McCrea and Whipple [23] determined $\mathrm{E}_H^{l}[V_{n,m}]$ from the recursion

$$\mathrm{E}_H^{l|}[V_{n,m}] = \frac{1}{4}\left(\mathrm{E}_H^{l|}[V_{n-1,m}] + \mathrm{E}_H^{l|}[V_{n+1,m}] + \mathrm{E}_H^{l|}[V_{n,m-1}] + \mathrm{E}_H^{l|}[V_{n,m+1}]\right) \tag{17}$$

when $(n,m) \neq (0,0)$, $n > -l$, and (because all walks start at the origin)

$$\mathrm{E}_H^{l|}[V_{0,0}] = 1 + \frac{1}{4}\left(\mathrm{E}_H^{l|}[V_{-1,0}] + \mathrm{E}_H^{l|}[V_{1,0}] + \mathrm{E}_H^{l|}[V_{0,-1}] + \mathrm{E}_H^{l|}[V_{0,1}]\right). \tag{18}$$

The recursion has a unique solution if the paths are restricted to the inside of a rectangle. Removing all but one side of the rectangle by limit processes, McCrea and Whipple found $\mathrm{E}_H^{l|}[V_{n,m}] =$

$$\frac{2}{\pi}\int_0^{\pi} \frac{\cos(\lambda m)\left(e^{-|n|\mu} - e^{-(n+2l)\mu}\right)}{\sinh(\mu)} d\lambda = \begin{cases} \frac{4}{\pi}\int_0^{\pi} \frac{\cos(\lambda m) e^{-l\mu}\sinh((n+l)\mu)}{\sinh(\mu)} d\lambda & \text{if } -l \leq n \leq 0 \\ \frac{4}{\pi}\int_0^{\pi} \frac{\cos(\lambda m) e^{-(n+l)\mu}\sinh(l\mu)}{\sinh(\mu)} d\lambda & \text{if } n \geq 0 \end{cases} \tag{19}$$

where $2 = \cos(\lambda) + \cosh(\mu)$. Denote by $\mathcal{T}_k$ and $\mathcal{U}_k$ the Chebyshev polynomials of the first and second kind, respectively, of degree k, i.e., $\mathcal{T}_k(x) = \cos(k\lambda)$ and $\mathcal{U}_{k-1}(x) = \sin(k\lambda)/\sin(\lambda)$ if $x = \cos(\lambda)$. It is easy to check that we can write (19) in this notation as

$$\mathrm{E}_H^{l|}[V_{n,m}] = \begin{cases} \frac{4}{\pi}\int_{-1}^{1} \mathcal{T}_m(x)\,\mathcal{U}_{n+l-1}(2-x)\,\frac{e^{-l\mu}dx}{(1-x^2)^{1/2}} & \text{if } -l \leq n \leq 0 \\ \frac{4}{\pi}\int_{-1}^{1} \mathcal{T}_m(x)\,\mathcal{U}_{l-1}(2-x)\,\frac{e^{-(n+l)\mu}dx}{(1-x^2)^{1/2}} & \text{if } n \geq 0 \end{cases} \tag{20}$$

Note that $e^{\mu} = e^{\operatorname{arccosh}(2-\cos(\lambda))} = \sqrt{(2-x)^2 - 1} + 2 - x$.

Remark 4 *Denote by* $\mathrm{E}_H^{|r}[V_{n,m}]$ *the expected number of visits at* (n,m) *of walks to the left of* $x = r$. *From* $\mathrm{E}_H^{|r}[V_{n,m}] = \mathrm{E}_H^{r|}[V_{-n,m}]$ *follows* $\mathrm{E}_H^{|r}[V_{n,m}] =$

$$\frac{2}{\pi}\int_0^{\pi} \frac{\cos(\lambda m)\left(e^{-|n|\mu} - e^{(n-2r)\mu}\right)}{\sinh(\mu)} d\lambda = \begin{cases} \frac{4}{\pi}\int_0^{\pi} \frac{\cos(\lambda m) e^{-r\mu}\sinh((r-n)\mu)}{\sinh(\mu)} d\lambda & \text{if } 0 \leq n \leq r \\ \frac{4}{\pi}\int_0^{\pi} \frac{\cos(\lambda m) e^{(n-r)\mu}\sinh(r\mu)}{\sinh(\mu)} d\lambda & \text{if } n \leq 0 \end{cases}.$$

3.1.1 First Passage

As before in this subsection about half-planes we assume that $m \geq 0$ and $l \geq 1$. A particle makes its "first passage" to the boundary point $(-l,m)$ at time k if it stayed away from $x = -l$ and reached $(1-l,m)$ at time $k-1$. By formula (5) there are

$$H_{k-1}^{l|}(1-l,m) = U_{k-1}(1-l,m) - U_{k-1}(1+l,m)$$

ways for the first passage at time k, and the probability of first passage (summed over time) at height $m \geq 0$ is $P_H^{l|}(-l,m) = \mathrm{E}_H^{l|}[V_{1-l,m}]$. We begin with a direct determination, using only formula (5), and find $P_H^{l|}(-l,m)$

$$\begin{aligned} &= \sum_{k \geq m+l} 4^{-k} H_{k-1}^{l|}(1-l,m) = \sum_{k=0}^{\infty} 4^{-2k-m-l} \binom{2k+m+l}{k} \binom{2k+m+l}{k+l} \frac{l}{2k+m+l} \\ &= 4^{-m-l} \binom{m+l-1}{m} \, {}_4F_3 \left[\begin{matrix} 1+\frac{m+l}{2}, \frac{1}{2}+\frac{m+l}{2}, \frac{1}{2}+\frac{m+l}{2}, \frac{m+l}{2}; \\ m+l+1, 1+l, 1+m \end{matrix} \; 1 \right]. \end{aligned} \tag{21}$$

Applying (15) together with the identity

$$\sum_{n=0}^{\infty} \binom{2n+x}{n} \frac{1}{2n+x} \xi^n = \frac{2^x}{x\left(1+\sqrt{1-4\xi}\right)^x},$$

we obtain $P_H^l(-l,m) =$

$$\begin{aligned} &\sum_{k=0}^{\infty} 4^{-2k-m-l} \binom{2k+m+l}{k} \frac{l}{2k+m+l} \frac{2^{2k+m+l}}{\pi} \int_0^{\pi} \cos\left((m-l)\theta\right) (\cos\theta)^{2k+m+l} \, d\theta \\ = \; &\frac{l}{\pi(m+l)} \int_0^{\pi} \cos\left((m-l)\theta\right) \left(\frac{\cos\theta}{1+\sin\theta}\right)^{m+l} d\theta. \end{aligned}$$

Hence

$$P_H^{l|}(-l,m) = \frac{l}{\pi(m+l)} \int_0^{\pi} \cos\left((m-l)\theta\right) \cot^{m+l}\left(\frac{\pi+2\theta}{4}\right) d\theta. \tag{22}$$

for all integers $m \geq 0$. With $x = \cos\theta$ and $dx/d\theta = -\sin\theta$ we get

$$P_H^{l|}(-l,m) = \frac{l}{\pi(m+l)} \int_{-1}^{1} \frac{\mathcal{T}_{|m-l|}(x) \left(1-\sqrt{1-x^2}\right)^{m+l}}{x\sqrt{1-x^2}} dx.$$

The next formula follows from (16), $P_H^{l|}(-l,m) =$

$$\begin{aligned} &\frac{1}{4\pi} \int_0^{\pi} \frac{\left(\cot^{l-1+m}\left(\frac{\pi+2\theta}{4}\right) \cos\left((m-l+1)\theta\right) - \cot^{1+l+m}\left(\frac{\pi+2\theta}{4}\right) \cos\left((m-1-l)\theta\right)\right)}{\sin\theta} d\theta \\ = \; &\frac{1}{2\pi} \int_0^{\pi} \cot^{m+l}\left(\frac{\pi+2\theta}{4}\right) \left(\cos\left((m-l)\theta\right) - \frac{\sin\left((m-l)\theta\right)}{\cos\theta}\right) d\theta. \end{aligned} \tag{23}$$

Finally, we know from (19) or [23] that $P_H^{l|}(-l,m) =$

$$\frac{1}{\pi} \int_0^{\pi} \cos(\lambda m) \, e^{-l\mu} d\lambda = \frac{1}{\pi} \int_{-1}^{1} \frac{\mathcal{T}_{|m|}(x)}{\left(\sqrt{(2-x)^2-1}+2-x\right)^l} \frac{dx}{\sqrt{1-x^2}}. \tag{24}$$

Thus (21), (22), (23), and (24) are the combinatorial/probabilistic reason for the identities (2).

3.2 Quadrants

We saw that the number $Q_k^{l\llcorner b}(n,m)$ of planar walks in the quadrant $-l < x$, $-b < y$ can be written in terms of half-plane walks as $Q_k^{l\llcorner b}(n,m) = H_k^{l|}(n,m) - H_k^{l|}(n,m+2b)$. Hence

$$\mathrm{E}_Q^{l\llcorner b}[V_{n,m}] = \mathrm{E}_H^{l|}[V_{n,m}] - \mathrm{E}_H^{l|}[V_{n,m+2b}]$$

can be calculated from any of the formulas for $\mathrm{E}_H^{l|}[V_{n,m}]$. In the following proposition we apply (19) because of an interesting and useful overlap in the domain of the two expressions for $\mathrm{E}_Q^{l\llcorner b}[V_{n,m}]$ (being identical for $-|m| \le n \le |m|$).

Proposition 5 *Let* $2 = \cos(\lambda) + \cosh(\mu)$, $x = \cos\lambda$, *thus* $e^{\mu} = \sqrt{(2-x)^2 - 1} + 2 - x$. *For* $n > -l$, *and* $m > -b$ *holds* $\mathrm{E}_Q^{l\llcorner b}[V_{n,m}]$

$$= \begin{cases} \frac{8}{\pi}\int_0^{\pi} \frac{\sin(b\lambda)\sin((m+b)\lambda)\sinh((n+l)\mu)}{\sinh\mu} d\lambda = \frac{4}{\pi}\int_{-1}^{1} \frac{\left(\mathcal{T}_{|m|} - \mathcal{T}_{m+2b}\right)(x)\mathcal{U}_{n+l-1}(2-x)}{e^{l\mu}(1-x^2)^{1/2}} dx & \text{if } n \le |m| \\ \frac{8}{\pi}\int_0^{\pi} \frac{\sin(b\lambda)\sin((m+b)\lambda)\sinh(l\mu)}{e^{(n+l)\mu}\sinh\mu} d\lambda = \frac{4}{\pi}\int_{-1}^{1} \frac{\left(\mathcal{T}_{|m|} - \mathcal{T}_{m+2b}\right)(x)\mathcal{U}_{l-1}(2-x)}{e^{(n+l)\mu}(1-x^2)^{1/2}} dx & \text{if } n \ge -|m| \end{cases}$$

Proof. Formula (19) implies

$$\mathrm{E}_Q^{l\llcorner b}[V_{n,m}] = \begin{cases} \frac{8}{\pi}\int_0^{\pi} \frac{\sin(b\lambda)\sin((m+b)\lambda)\sinh((n+l)\mu)e^{-l\mu}}{\sinh\mu} d\lambda & \text{if } n \le 0 \\ \frac{8}{\pi}\int_0^{\pi} \frac{\sin(b\lambda)\sin((m+b)\lambda)\sinh(l\mu)e^{-(n+l)\mu}}{\sinh\mu} d\lambda & \text{if } n \ge 0 \end{cases}$$

(see also [23]). The two integrals do not only agree for $n = 0$; if $m > -b$; they are the same for all $-|m| \le n \le |m|$, as shown in Proposition 7 below. ■

Lemma 6 *Let* $\cos\lambda + \cosh\mu = 2$.

$$\int_0^{\pi} \frac{\cos(m\lambda)\sinh(n\mu)}{\sinh\mu} d\lambda = 0 \text{ for } |n| \le |m|$$

Proof. (By M.E.H. Ismail) Let $0 \le n \le m$, $x = \cos\lambda$, and $z = \cos(i\mu) = \cosh(\mu)$, thus $z = 2 - x$. Denote by $\mathcal{T}_k$ and $\mathcal{U}_k$ the Chebyshev polynomials of the first and second kind, respectively, of degree k,

$$\mathcal{T}_k(x) = \cos(k\lambda) \text{ and } \mathcal{U}_{k-1}(z) = \frac{\sin(ki\mu)}{\sin(i\mu)} = \frac{\sinh(k\mu)}{\sinh(\mu)}.$$

From $dx/d\lambda = -\sin\lambda$ follows

$$\int_0^{\pi} \frac{\cos(m\lambda)\sinh(n\mu)}{\sinh\mu} d\lambda = \int_{-1}^{1} \frac{T_m(x)\,U_{n-1}(z)}{\sin\lambda} dx = \int_{-1}^{1} \frac{T_m(x)\,U_{n-1}(2-x)}{\sqrt{1-x^2}} dx$$

The Chebyshev polynomials are orthogonal on $[-1,1]$ with respect to $(1-x^2)^{-1/2}$, and $U_{n-1}(2-x)$ is of degree less than m, hence the integral is zero. The integral vanishes for all $0 \le n \le |m|$ because it is even in m, and it is odd in n. ■

Proposition 7 *For* $\max\{-|m|,-|m+2b|\} \leq n \leq \min\{|m|,|m+2b|\}$

$$\int_0^\pi \frac{\sin(\lambda b)\sin(\lambda(m+b))\sinh((n+l)\mu)e^{-l\mu}}{\sinh\mu}d\lambda = \int_0^\pi \frac{\sin(\lambda b)\sin(\lambda(m+b))\sinh(l\mu)e^{-(n+l)\mu}}{\sinh\mu}d\lambda$$

Proof. From $-|m| \leq n \leq |m|$ follows $\int_0^\pi (\cos(m\lambda)\sinh(n\mu)/\sinh\mu)\,d\lambda = 0$, and in the same way $\int_0^\pi (\cos(|m+2b|\lambda)\sinh(n\mu)/\sinh\mu)\,d\lambda = 0$. Hence

$$0 = \int_0^\pi \frac{(\cos(m\lambda)-\cos((m+2b)\lambda))\sinh(n\mu)}{\sinh\mu}d\lambda = 2\int_0^\pi \frac{\sin(\lambda b)\sin(\lambda(m+b))\sinh(n\mu)}{\sinh\mu}d\lambda$$

and $\int_0^\pi (\sin(\lambda b)\sin(\lambda(m+b))e^{n\mu}/\sinh\mu)\,d\lambda = \int_0^\pi (\sin(\lambda b)\sin(\lambda(m+b))e^{-n\mu}/\sinh\mu)\,d\lambda$. Subtract
$\int_0^\pi \left(\sin(\lambda b)\sin(\lambda(m+b))e^{-(n+2l)\mu}/\sinh\mu\right)d\lambda$ from both sides and get

$$\int_0^\pi \frac{\sin(\lambda b)\sin(\lambda(m+b))\left(e^{n\mu}-e^{-(n+2l)\mu}\right)}{\sinh\mu}d\lambda = \int_0^\pi \frac{\sin(\lambda b)\sin(\lambda(m+b))\left(e^{-n\mu}-e^{-(n+2l)\mu}\right)}{\sinh\mu}d\lambda.$$

■

3.2.1 First passage

The first passage probability $P_Q^{l\llcorner b}(-l,m)$ to the boundary $x=-l$ at height $m>-b$ equals $P_H^{l|}(-l,m) - P_H^{l|}(-l,m+2b) =$

$$\begin{aligned}
&\frac{1}{2\pi}\int_0^\pi \left(\cot^{l+|m|}\left(\frac{\pi+2\lambda}{4}\right)\left(\cos((|m|-l)\lambda) - \frac{\sin((|m|-l)\lambda)}{\cos\lambda}\right)\right. \qquad (25)\\
&\left.-\cot^{l+m+2b}\left(\frac{\pi+2\lambda}{4}\right)\left(\cos((m+2b-l)\lambda) - \frac{\sin((m+2b-l)\lambda)}{\cos\lambda}\right)\right)d\lambda\\
=\ &\frac{2}{\pi}\int_0^\pi \sin(\lambda b)\sin(\lambda(m+b))\left(2-\cos\lambda-\sqrt{(2-\cos\lambda)^2-1}\right)^l d\lambda \qquad (26)\\
=\ &\frac{l}{\pi}\int_0^\pi \left(\frac{\cos((|m|-l)\lambda)}{\left(\frac{1+\sin\lambda}{\cos\lambda}\right)^{|m|+l}(|m|+l)} - \frac{\cos((m+2b-l)\lambda)}{\left(\frac{1+\sin\lambda}{\cos\lambda}\right)^{m+2b+l}(m+2b+l)}\right)d\lambda. \qquad (27)
\end{aligned}$$

using (23), (24), and (22) (remember that $\cot\left(\frac{\pi+2\lambda}{4}\right) = \frac{\cos\lambda}{1+\sin\lambda}$). Note that $P_Q^{l\llcorner b}(n,-b) = P_Q^{b\llcorner l}(-b,n)$. Written as sums, $P_Q^{l\llcorner b}(-l,m) = P_H^l(-l,m) - P_H^l(-l,m+2b) =$

$$\begin{aligned}
=\ &\sum_{k=0}^{b-1} 4^{-2k-m-l}\binom{2k+m+l}{k+m+l}\binom{2k+m+l}{k+m}\frac{l}{2k+m+l} \qquad (28)\\
&+\sum_{k=b}^{\infty}\frac{4^{-2k-m-l}l}{2k+m+l}\left(\binom{2k+m+l}{k}\binom{2k+m+l}{k+m} - \binom{2k+m+l}{k-b}\binom{2k+m+l}{k+b+m}\right)
\end{aligned}$$

For example, if $b=1$ we get $P_Q^{l\llcorner 1}(-l,m)=$

$$\sum_{k=0}^{\infty}\binom{2k+m+l}{k}\binom{2k+m+l+1}{k+1+m}\frac{4^{-2k-m-l}\,l\,(m+1)}{(k+m+l+1)\,(2k+m+l)}$$
$$=\ \frac{2}{\pi}\int_0^{\pi}\sin(lx)\sin(x)\left(2-\cos x-\sqrt{(2-\cos x)^2-1}\right)^{m+1}dx. \qquad (29)$$

If $l=b=1$ we find the following five expressions for the first passage probability in the first quadrant to the y-axis at height $m\geq 0$,

from (28): $P_Q^{1\llcorner 1}(-1,m)=\sum_{k=0}^{\infty}4^{-2k-m-1}\frac{(m+1)}{(k+m+1)(k+1)}\binom{2k+m}{k}\binom{2k+m+2}{k}$,

from (29): $=\frac{2}{\pi}\int_0^{\pi}\sin^2(\lambda)\left(2-\cos\lambda-\sqrt{(2-\cos\lambda)^2-1}\right)^{m+1}d\lambda$,

from (26): $=\frac{2}{\pi}\int_0^{\pi}\sin(\lambda)\sin(\lambda(m+1))\left(2-\cos\lambda-\sqrt{(2-\cos\lambda)^2-1}\right)d\lambda$,

from (27): $=\frac{l}{\pi}\int_0^{\pi}\left(\frac{\cos\lambda}{1+\sin\lambda}\right)^{m+1}\left(\frac{\cos(\lambda(m-1))}{m+1}-\frac{\left(\frac{\cos\lambda}{1+\sin\lambda}\right)^2\cos(\lambda(m+1))}{m+3}\right)d\lambda$,

from (25): $=\frac{1}{\pi}\int_0^{\pi}\left(\frac{\cos\lambda}{1+\sin\lambda}\right)^{m}\cos(m\lambda)\sin(\lambda)\frac{1+\cos^2\lambda}{(1+\sin\lambda)^2}dx.$

3.3 Shifted Octants

The number $O_k^{l\not\llcorner d}(n,m)$ of planar walks in the shifted octant $-l<x,\ y>x-d$ can be written in terms of quadrant walks as $O_k^{l\not\llcorner d}(n,m)=Q_k^{l\llcorner l+d}(n,m)-Q_k^{l+d\llcorner l}(n-d,m+d)=Q_k^{l\llcorner l+d}(n,m)-Q_k^{l\llcorner l+d}(m+d,n-d)$. Hence

$$\mathrm{E}_O^{l\not\llcorner d}[V_{n,m}]=\mathrm{E}_Q^{l\llcorner l+d}[V_{n,m}]-\mathrm{E}_Q^{l+d\llcorner l}[V_{n-d,m+d}]=\mathrm{E}_Q^{l\llcorner l+d}[V_{n,m}]-\mathrm{E}_Q^{l\llcorner l+d}[V_{m+d,n-d}] \qquad (30)$$

for $-l-d<n-d<m$. We can calculate the probability $P_O^{l\not\llcorner d}(-l,m)$ of first passage to the line $x=-l$ from the first passage probabilities in shifted quadrants. This is not the case for the first passage to the diagonal line $y=x-d$, thus we need to know $\mathrm{E}_O^{l\not\llcorner d}[V_{n,m}]$. Formula (30) shows that any of the expressions for $\mathrm{E}_Q^{l\llcorner b}[V_{n,m}]$ can be used to find $\mathrm{E}_O^{l\not\llcorner d}[V_{n,m}]$. An example is worked out in the following proposition.

Proposition 8 *Let* $\cos\lambda+\cosh\mu=2$, $x=\cos\lambda$, *thus* $e^{\mu}=\sqrt{(2-x)^2-1}+2-x$. *For* $-l-d<n-d<m$ *holds* $\mathrm{E}_O^{l\not\llcorner d}[V_{n,m}]$

$$=\begin{cases}\frac{8}{\pi}\int_0^{\pi}\frac{\left(\sin(\lambda(l+d))-\sin(\lambda l)e^{-d\mu}\right)\sin(\lambda(m+l+d))\sinh((n+l)\mu)}{e^{l\mu}\sinh\mu}d\lambda & \text{if } n\leq|m|\\ \frac{8}{\pi}\int_0^{\pi}\frac{\sin(\lambda(m+l+d))\left(\sin(\lambda(l+d))\sinh(l\mu)e^{-n\mu}-\sin(\lambda l)\sinh((n+l)\mu)e^{-d\mu}\right)}{e^{l\mu}\sinh\mu}d\lambda & \text{if } n\geq-|m|\end{cases}$$
$$=\begin{cases}\frac{4}{\pi}\int_{-1}^{1}\left(\frac{\left(\mathcal{T}_{|m|}-\mathcal{T}_{m+2(l+d)}\right)(x)}{e^{l\mu}}-\frac{\left(\mathcal{T}_{|m+d|}-\mathcal{T}_{m+d+2l}\right)(x)}{e^{(l+d)\mu}}\right)\frac{\mathcal{U}_{n+l-1}(2-x)}{(1-x^2)^{1/2}}dx & \text{if } n\leq|m|\\ \frac{4}{\pi}\int_{-1}^{1}\left(\frac{\left(\mathcal{T}_{|m|}-\mathcal{T}_{m+2(l+d)}\right)\mathcal{U}_{l-1}(2-x)}{e^{(n+l)\mu}}-\frac{\left(\mathcal{T}_{|m+d|}-\mathcal{T}_{m+d+2l}\right)(x)\mathcal{U}_{n+l-1}(2-x)}{e^{(l+d)\mu}}\right)\frac{dx}{\sqrt{1-x^2}} & \text{if } n\geq-|m|\end{cases}$$

The proof is a straight forward application of Proposition 5 to $\mathrm{E}_O^{l\not\vee d}[V_{n,m}] = \mathrm{E}_Q^{l\llcorner l+d}[V_{n,m}] - \mathrm{E}_Q^{l+d\llcorner l}[V_{n-d,m+d}]$, noting that $n-d \le |m+d|$ inside the shifted octant $-l-d < n-d < m$. A different looking formula for $\mathrm{E}_O^{l\not\vee d}[V_{n,m}]$ can be derived in the same way if we write $\mathrm{E}_O^{l\not\vee d}[V_{n,m}] = \mathrm{E}_Q^{l\llcorner l+d}[V_{n,m}] - \mathrm{E}_Q^{l\llcorner l+d}[V_{m+d,n-d}]$.

3.3.1 First passage to $x = -l$

The probability $P_O^{l\not\vee d}(-l,m)$ of first passage strictly inside the shifted octant $x > -l$, $y > x-d$ to the vertical boundary $x = -l$ can be obtained in different variations from (25) - (28) because

$$P_O^{l\not\vee d}(-l,m) = \frac{1}{4}\,\mathrm{E}_O^{l\not\vee d}[V_{1-l,m}] = P_Q^{l\llcorner l+d}(-l,m) - P_Q^{l+d\llcorner l}(-l-d,m+d)$$

If $d = l = 1$, we write P_O instead of $P_O^{1\not\vee 1}$, and we get from (7) that $P_O(-1,m) =$

$$\begin{aligned} & 4^{-m-1}\binom{m+3}{3}\sum_{k=0}^{\infty}\frac{4^{-2k}}{k+1}\binom{m+2k+2}{k}\binom{m+2k}{k}\Big/\binom{m+k+3}{3} \qquad (31)\\ = \;& \frac{1}{\pi}\int_0^{\pi}\cos\left((m-1)\,x\right)\left(\frac{\cos x}{1+\sin x}\right)^{m+1}\left(\frac{1}{m+1}-\left(\frac{\cos x}{1+\sin x}\right)^{2}\frac{2}{m+3}\right)dx\\ & -\frac{1}{\pi}\int_0^{\pi}\left(\frac{\cos x}{1+\sin x}\right)^{m+5}\left(\frac{\cos\left((m+3)\,x\right)-2\cos\left((m+1)\,x\right)}{m+5}\right)dx. \end{aligned}$$

For some numerical examples see Table 3.

3.3.2 First passage to $y = x - d$

Let $n > 1-l$. There are two ways to get to the boundary point $(n, n-d)$, from above at $(n, n-d+1)$ and from the left at $(n-1, n-d)$. Thus $P_O^{l\not\vee d}(n,n-d) = \frac{1}{4}\,\mathrm{E}_O^{l\not\vee d}[V_{n,n-d+1}] + \frac{1}{4}\,\mathrm{E}_O^{l\not\vee d}[V_{n-1,n-d}]$. We apply Proposition 8, noting that $n \ge -|n-d+1|$ and $n-1 \ge -|n-d|$ for all n. Hence $P_O^{l\not\vee d}(n,n-d) =$

$$\begin{aligned} & \frac{2}{\pi}\int_0^{\pi} e^{-l\mu}\left[\sin\left(\lambda\left(l+d\right)\right)\sinh\left(l\mu\right)e^{-n\mu}\left(\sin\left(\lambda\left(n+1+l\right)\right)+e^{\mu}\sin\left(\lambda\left(n+l\right)\right)\right)\right.\\ & \left.-\sin\left(\lambda l\right)e^{-d\mu}\left(\sinh\left(\left(n+l\right)\mu\right)\sin\left(\lambda\left(n+1+l\right)\right)+\sinh\left(\left(n+l-1\right)\mu\right)\sin\left(\lambda\left(n+l\right)\right)\right)\right]\frac{d\lambda}{\sinh\mu}. \end{aligned}$$

In terms of Chebyshev polynomials, $P_O^{l\not\vee d}(n,n-d)$

$$\begin{aligned} = \;& \frac{1}{\pi}\int_{-1}^{1}\left[\frac{\left(\mathcal{T}_{|n-d+1|}-\mathcal{T}_{n+1+2l+d}+e^{\mu}\left(\mathcal{T}_{|n-d|}-\mathcal{T}_{n+2l+d}\right)\right)\mathcal{U}_{l-1}\left(2-x\right)}{e^{(l+n)\mu}}\right.\\ & \left.-\frac{\left(\mathcal{T}_{|n+1|}-\mathcal{T}_{n+1+2l}\right)(x)\,\mathcal{U}_{n+l-1}\left(2-x\right)+\left(\mathcal{T}_{|n|}-\mathcal{T}_{n+2l}\right)(x)\,\mathcal{U}_{n+l-2}\left(2-x\right)}{e^{(l+d)\mu}}\right]\frac{dx}{\sqrt{1-x^2}} \end{aligned}$$

Again, $\cos\lambda + \cosh\mu = 2$, $x = \cos\lambda$, thus $e^{\mu} = \sqrt{(2-x)^2 - 1} + 2 - x$.

For $n = 1 - l$, the boundary point $(1-l, 1-l-d)$ below the corner of the shifted octant can only be reached from above. Hence $P_O^{l\not\vee d}(1-l, 1-l-d) = \frac{1}{4}\,\mathrm{E}_O^{l\not\vee d}[V_{1-l,2-l-d}] = P_O^{l\not\vee d}(-l, 2-l-d)$, the probability of passage to the vertical boundary $x = -l$.

If the walk is restricted to the second octant ($l = b = 1$), first passage below the diagonal to $(n, n-1)$ happens with probability

$$\begin{aligned} P_O(n, n-1) &= \sum_{k=0}^{\infty} \left(4^{-2k-2(n-1)-1} O_{2(n-1)+2k}(n-1, n-1) + 4^{-2k-2n-1} O_{2n+2k}(n, n)\right) \qquad (32) \\ &= \sum_{k=0}^{\infty} \frac{4^{-2k-2n}(n+1)\binom{2n+2k}{n+k-1}\binom{2n+2k-2}{k}}{3(n+k)\binom{2n+k+2}{4}} \left((n+1)(n(2n+1)+2k(n+1))+k\right) \\ &= \frac{2}{\pi}\int_0^{\pi} e^{-\mu}\left[\sin(2\lambda)\, e^{-n\mu}\left(\sin(\lambda(n+2)) + e^{\mu}\sin(\lambda(n+1))\right)\right. \\ &\quad \left. -\frac{\sin(\lambda)\, e^{-\mu}}{\sinh\mu}\left(\sinh((n+1)\mu)\sin(\lambda(n+2)) + \sinh(n\mu)\sin(\lambda(n+1))\right)\right] d\lambda. \end{aligned}$$

$j =$	0	1	2	3	4	5	6	7	8	20
$P_O(-1, j) =$	.27005	.08018	.02658	.00991	.00416	.00194	$.0^3 99$	$.0^3 54$	$.0^3 32$	$.0^6 6$
$P_O(j+1, j) =$	.29414	.02935	.00691	.00229	.00093	.00044	$.0^3 23$	$.0^3 13$	$.0^4 7$	$.0^6 15$

Table 3: Some examples of first passage proabilities in the second octant

4 Related structures

Certain subsets of the diffusion walks in an octant can be visualized by structures that may look quite different. We list non-crossing pairs of Dyck paths, bicolored Motzkin paths, staircase polygons in the second octant, and $\{\rightarrow\uparrow\}$-paths enumerated by left turns. Of course, other aspects of such structures may not be efficiently represented by diffusion walks. A thorough discussion of these and other structures and their applications can be found in *The Statistical Mechanics of Interacting Walks, Polygons, Animals and Vesicles*, by Janse van Rensburg [21].

4.1 Pairs of Non-crossing Dyck paths

The diagonal diffusion, with step set $\{\nearrow, \swarrow, \searrow, \nwarrow\}$, is easily mapped onto the ordinary diffusion by the matrix $\frac{1}{2}\left(\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right)$. The matrix maps the diagonal steps $\nearrow, \swarrow, \searrow, \nwarrow$ onto $\uparrow, \downarrow, \rightarrow, \leftarrow$ (in this order). If we draw two independent random walks with steps ± 1 on the integers, a vertical walk V along the y-axis (marked by $\vdots$ in Fig. 3) and a horizontal walk H along the x-axis $(\cdots)$, then *the diagonal diffusion ($\bullet$) is the vector sum of the two integer walks*, i.e., if H and V are at the positions $(h_k, 0)$ and $(0, v_k)$ at time k, then (h_k, v_k) is the position of the diagonal diffusion walk (this proves formula (3)).

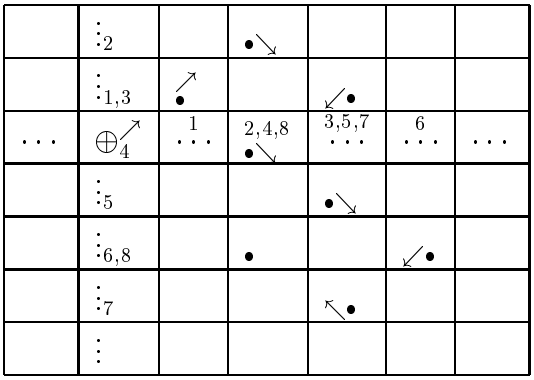


Figure 3: Diagonal diffusion generated by two perpendicular integer walks. The subscripts and superscripts indicate the position at step k of the vertical and horizontal walks, respectively.

If we restrict the one-dimensional walks to nonnegative integers and require that the i-th term v_i in the vertical walk is not larger than the i-th term h_i in the horizontal walk (making them dependent!), i.e., $h_i \geq v_i \geq 0$ for all i, then the diagonal diffusion stays in the first octant as in Fig. 4. Note that these restricted one-dimensional walks along the axes become Dyck paths (i.e., weakly above the x-axis) if we replace the steps $1, -1$ by $\nearrow$ and $\searrow$, respectively. In the pair P_H, P_V of paths we write the horizontal walk (P_H) first ; if the Dyck pair P_H, P_V ends at (k, h), (k, v) the diagonal diffusion ends at (h, v) after k steps (k, h and v are of the same parity). The image under $\frac{1}{2}\left(\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right)$ (ordinary diffusion) of the diagonal diffusion stays in the second octant.

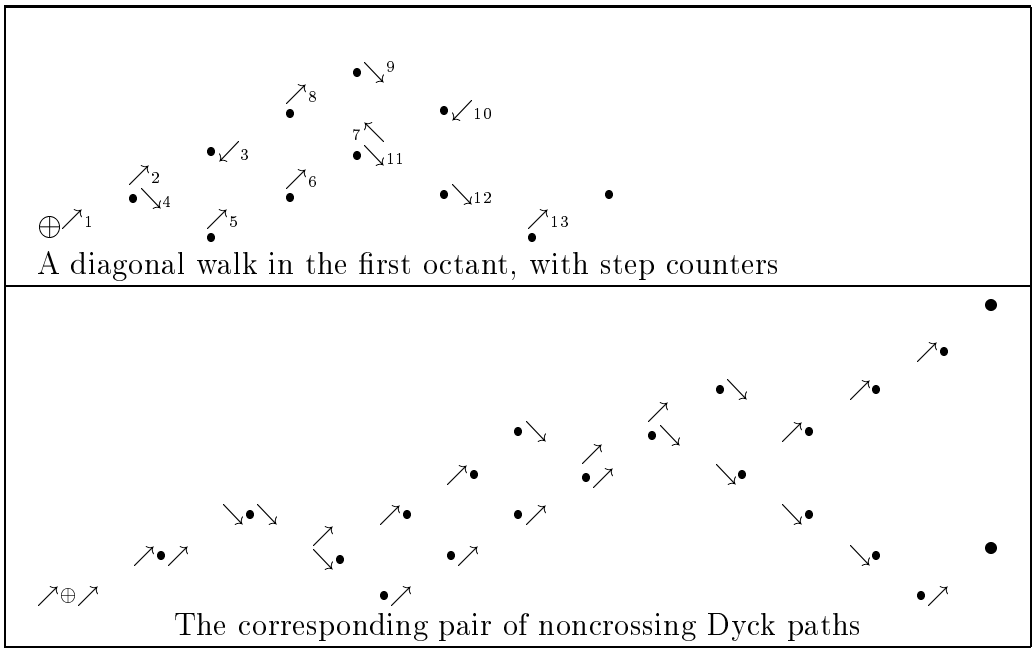


Figure 4: The correspondence between diagonal diffusion and Dyck pairs

Hence the number of pairs of noncrossing Dyck paths from the origin to $(k, h), (k, v)$ equals

the number of diagonal diffusion walks to (h,v) in k steps staying in the first octant, which in turn equals the number of ordinary diffusion walks in the second octant reaching $\frac{1}{2}\left(\begin{smallmatrix}1&-1\\1&1\end{smallmatrix}\right)\binom{h}{v} = ((h-v)/2,(h+v)/2)$ after k steps,

$$O_k((h-v)/2,(h+v)/2) = \frac{(2+h-v)(4+h+v)(h+3)(v+1)}{12(k-v+2)}\binom{k+2}{\frac{k-v}{2}}\binom{k}{\frac{h+k}{2}}/\binom{\frac{h+k}{2}+3}{3}$$

(see (6)). The pairs that end at a common point are equivalent to staircase polygons (parallel polyominoes); we discuss them in Subsection 4.3 in more detail.

It follows from the bijection between pairs of noncrossing Dyck paths and diffusion walks that

- the expected number of pairs of noncrossing paths P_H, P_V where the bottom path P_V falls below the x-axis for the first time when the top path P_H is at height h, equals $4P_O(h/2,(h-2)/2)$ (see (32)).
- the number of noncrossing Dyck pairs ending on the line $x = 2k$ with the bottom path on the x-axis equals C_kC_{k+1} (see (10)).

Remark 9 *There is also a connection between* single *Dyck paths and "short" walks in an octant. If any diffusion walk reaches the point (n,m) in the first quadrant in $n+m$ steps, all steps must be either $\rightarrow$ or $\uparrow$. The number of $\{\rightarrow\uparrow\}$-paths reaching (n,m) in $n+m$ steps* while staying in the second octant *is*

$$O_{n+m}(n,m) = \frac{m+1-n}{n+m+1}\binom{n+m+1}{n}.$$

Mapping $\rightarrow$ to $\searrow$, and $\uparrow$ to $\nearrow$ shows that this is also the number of (single) Dyck paths to $(n+m,m-n)$. If $m=n$, the Dyck paths end on their boundary, the x-axis, and their number is $C_n = \binom{2n}{n}/(n+1)$, the n-th Catalan number. These results are familiar from the classical ballot problem (with two candidates), first solved in 1887 by André [1].

Mapping diffusion walks to pairs of noncrossing Dyck paths goes back at least to Feller [8]. The method can be extended to n-tuples of Dyck paths. We only want to discuss triples. For this purpose we consider 3-D diagonal diffusion as in Subsection 2.3.1, generated by three independent random walks with steps ±1 on the integers, a vertical walk V along the y-axis, a horizontal walk H along the x-axis as before, and an additional up-down walk U along the z-axis. We map the walks H,V,U into a triple P_H,P_V,P_U of Dyck paths ($1 \mapsto \nearrow$ and $-1 \mapsto \searrow$), requiring that $v_i \geq 0$, $h_i \geq 0$, and $u_i \geq 0$ for all i. Formula (12) tells us how many diagonal diffusion walks reach (n,m,l) in k steps, restricted to lattice points (h_i,v_i,u_i), where $0 \leq h_i \leq v_i \leq u_i$ for all $i = 0,\ldots,k$ (and $h_0 = v_0 = u_0 = 0$, $h_k = n$, $v_k = m$, $u_k = l$). This number, $D_k(n,m,l)$, is therefore the same as the number of noncrossing Dyck triples from $(0,0)$ to (k,n), (k,m), and (k,l). Suppose we separate the triples by shifting the top path upwards two units, and then the upper pair again upwards two units, resulting in an unchanged bottom path from $(0,0)$ to (k,n), a shifted middle path from $(0,2)$ to $(k,m+2)$, and a shifted top

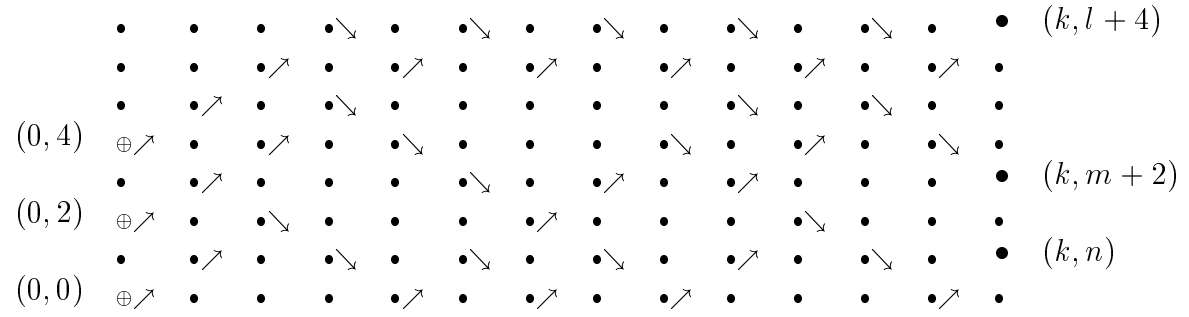


Figure 5: Three vicious walkers

path from $(0,4)$ to $(k, l+4)$. The three paths never occupy the same lattice point; the particles moving along those paths are called *vicious walkers* (Fig. 5).

If $n = m = l = 0$ such a configuration of nontouching Dyck paths is called a *watermelon* with three ribs. Note that k must be even in this case, $k = 2r$, say. Thus $D_{2r}(0,0,0) = \binom{6}{3} C_r C_{r+1} C_{r+2} / \left(\binom{r+5}{3} \binom{r+4}{3} \right)$ (see (1)) is the number of water melons. The number of watermelons with many ribs can be found by restricting the diagonal diffusion walk in many dimensions to the appropriate cone ([9],[14]). For more on this topic and additional references see [17]. The determinant enumerating several non-intersecting lattice paths (now called the Lindström-Gessel-Viennot formula) goes back to the work of Lindström [22], and Gessel and Viennot [12], [11].

4.2 Bicolored Motzkin paths

A Motzkin path has step set $\{\nearrow, \searrow, \longrightarrow\}$ and stays weakly above the x-axis. Map the step vectors of the diffusion walks onto $\{\nearrow, \searrow, \longrightarrow, \dashrightarrow\}$ according to Table 4.

Diffusion:	$\longrightarrow$	$\longleftarrow$	$\uparrow$	$\downarrow$
Bicolored Motzkin:	$\circ \nearrow$	$\bullet \searrow$	$\bullet \longrightarrow$	$\circ \longrightarrow$
color:	white	black	black	white

Table 4: The bijection

Diffusion in the right half plane is in one-to-one correspondence to bicolored Motzkin paths. We say that the Motzkin path has excess m if it ends with m more black $\longrightarrow$-steps than white $\dashrightarrow$-steps. Diffusion in the first quadrant is bijectively mapped onto bicolored Motzkin paths that reach any point in the first octant with at least as many black $\longrightarrow$-steps as white $\dashrightarrow$-steps, i.e., the excess is never negative along the path. The diffusion will stay in the second octant iff the corresponding bicolored Motzkin paths reach any point in the first octant with a total number of white steps ($\nearrow$ or $\dashrightarrow$) not exceeding the total number of black steps ($\searrow$ or $\longrightarrow$). We call them saturated. Let $m \geq n$. The number $O_k(n,m)$ of octant walks to (n,m) in k steps (see ((6))) equals the number of saturated Motzkin paths to (k,n) with excess m.

First passage through the x-axis at height m of the diffusion walk corresponds to the saturated path with excess m, crossing through the x-axis for the first time. First passage of the

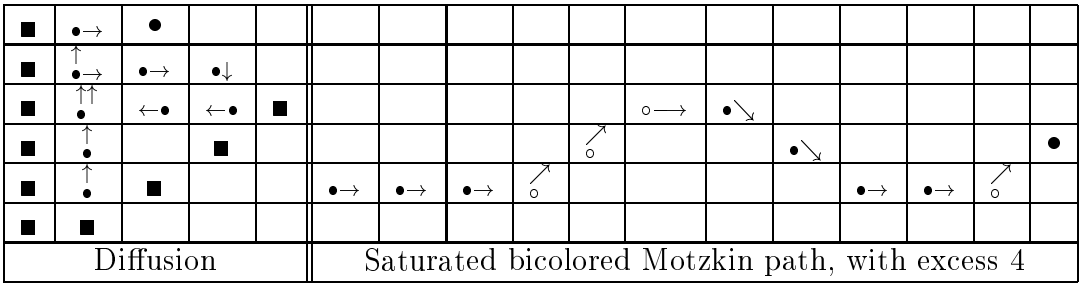


Figure 6: Diffusion in octant $\longleftrightarrow$ bicolored Motzkin

diffusion walk through the diagonal to $(n, n-1)$ corresponds to the bicolored Motzkin path of height n and excess $n-1$, having for the first time more white than black steps. See (31) and (32) for the first passage probabilities. Expression in terms of Catalan numbers are obtained if we enumerate all saturated paths ending at $(k, 0)$ with any excess (see (8)), and if we count all saturated paths ending on the line $x = 2k$ with an equal number of black and white steps (see (10)). For recent work on Motzkin paths see [30] and [29].

4.3 Staircase polygons in the augmented second octant

A staircase polygon (parallelogram polyomino) is a polygon bounded by two $\{\rightarrow\uparrow\}$-paths (staircases) that have only the beginning and the endpoint in common. Because staircase polygons are considered invariant under vertical and horizontal shifts, we can assume that the pair of bounding paths starts at the origin.

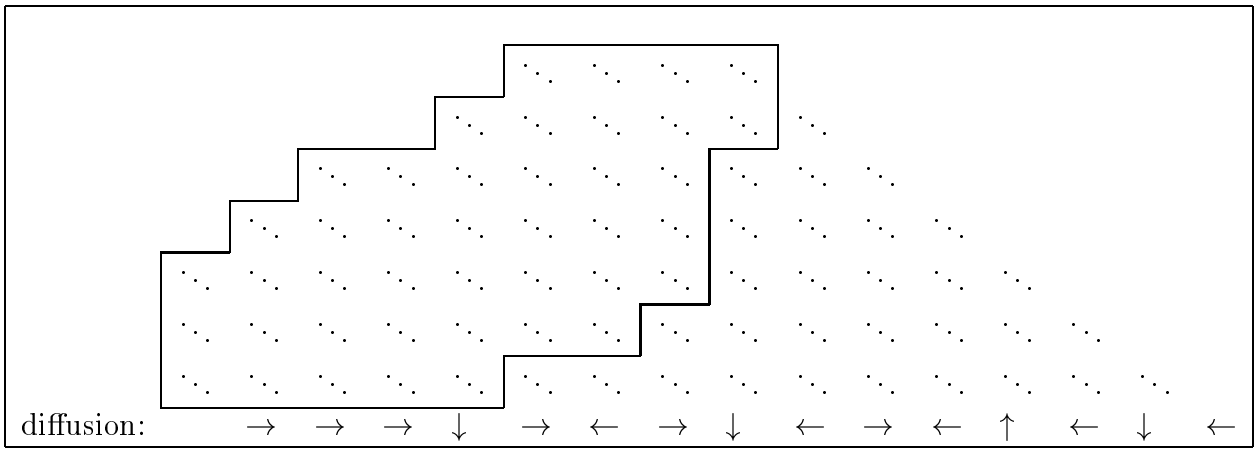


Figure 7: Staircase polygons and dotted diagonal gaps

If we look at a staircase polygon from the Northeast, we see (diagonal) gaps between the paths. We map the polygon into a diffusion according to the change of gaps. An increase (decrease) in gap width is mapped to a $\rightarrow$-step ($\leftarrow$-step). If the gap is just shifted to the right (diagonally shifted upwards) we map it onto a $\downarrow$-step ($\uparrow$-step). Thus a staircase polygon corresponds to a $\{\rightarrow\uparrow\leftarrow\downarrow\}$-planar walk (see also [7],[21]). Among the first k steps let l_k be the

number of $\leftarrow$-steps, and r_k the number of $\rightarrow$-steps. Because there cannot be less (expanding) $\rightarrow$-steps than (shrinking) $\leftarrow$-steps in any beginning part of the walk, we find $r_k \geq l_k$; the path stays in the right half-plane $x \geq 0$, and returns to the x-axis at the end. The vertical steps do not change the gap width of the polygon; there can be any number (u_k up, and d_k down) of them, at any location. Vice versa, any diffusion walk in the right half plane ending at $(0, j)$ after k steps can be mapped onto a staircase polygon with lower left corner $(0, 0)$ and upper right corner $(n + 1, m + 1)$ where $j = m - n$ and $k = m + n$. We can thus use equation (4) to find the number of all staircase polygons from $(0, 0)$ to (n, m),

$$\binom{n+m-2}{n-1}\binom{n+m-2}{m-1} - \binom{n+m-2}{m}\binom{n+m-1}{n}$$

a Narayana number. The enumeration by gap width allows for much deeper results than the above application (see [5]). A bijection between staircase polygons and bicolored Motzkin paths is described in [21]; for an approach via skew Ferrer's diagrams see [6].

We say that a staircase polygon stays in the augmented second octant if it stays weakly above $y = x - 1$. Any staircase polygon is bounded by two $\{\rightarrow\uparrow\}$-paths, starting with a lower left corner $\llcorner$ at $(0, 0)$,and ending with an upper right corner $\urcorner$ at $(n + 1, m + 1)$, say. If we remove those two corners from a staircase polygon in the augmented second octant, then shift both paths so that they start at the origin and end at (n, m), and turn the shifted pair downwards by 45°, we obtain a pair of non-crossing Dyck paths from the origin to the common endpoint $(n + m, m - n)$. We enumerated such pairs in Section 4.1; if we denote by $S(n, m)$ the number of staircase polygons to (n, m) in the augmented second octant, we find for $m \geq n$

$$S(n+1, m+1) = O_{m+n}(0, m-n) = 6\frac{(m+n)!\,(m+n+2)!}{n!\,(n+1)!\,(m+2)!\,(m+3)!}\binom{m-n+3}{3}$$

(or use formula (7)). Some special cases of staircase polygons in the augmented second octant:

- If $m = n$ the polygon ends at $(n + 1, n + 1)$, and by (11) there are

$$6\frac{(2n)!\,(2n+2)!}{n!\,(n+1)!\,(n+2)!\,(n+3)!} = C_n C_{n+2} - C_{n+1}^2$$

  such staircase polygons.

- From $O_{(n+k)+k}(0, n) = S(k + 1, n + k + 1)$ follows that the expected number of polygons ending n vertical steps above the diagonal equals $4P_O(-1, n)$ (see (31)).

- The number of polygons ending on the line $n + m = k$ for given integer $k \geq 2$ equals $C_{\lfloor (k-1)/2 \rfloor} C_{\lfloor k/2 \rfloor}$ (see (8)).

- The number of polygons ending on the diagonal at (n, n) equals $O_{2n-2}(0, 0) = C_{n-1}C_{n+1} - C_n^2$ (see (11)).

4.4 $\{\rightarrow\uparrow\}$-paths in the augmented third hexadecant enumerated by left turns

Denote by $[u,v]$ the discrete interval $u \le x \le v$, where $x \in \mathbb{Z}$, and by $\binom{[u,v]}{k}$ set set of all k-element subsets $[u,v]$. Several interesting combinatorial problems can be bijectively mapped to $\binom{[u,v]}{k} \times \binom{[p,q]}{l}$ (or subsets thereof) for certain choices of the parameters. The following examples are connected with diffusion walks in the second octant.

Lemma 10 *Let n_p, n_q and m be nonnegative integers. There exists a bijection between $\binom{[0,m+n_p-1]}{m} \times \binom{[0,m+n_q-1]}{m}$ and*

1. *pairs p,q of $\{\rightarrow\uparrow\}$-paths, starting at the origin and ending at the point (n_p,m) and (n_q,m), respectively.*

2. *$\{\rightarrow\uparrow\}$-paths, starting at the origin and ending at $(m+n_p, m+n_q)$, taking m left turns ($\begin{smallmatrix} & \uparrow \\ \rightarrow & \circ \end{smallmatrix}$).*

3. *$\{\rightarrow\uparrow\}$-paths, starting at the origin and ending at $(m+n_p, m+n_q)$, taking m right turns ($\begin{smallmatrix} \circ & \rightarrow \\ \uparrow & \end{smallmatrix}$).*

Proof. Consecutively label the $m+n_p$ steps of the path p with the numbers $0,\ldots,m+n_p-1$. Let x_i be the label of the i-th vertical step, thus $0 \le x_1 < \cdots < x_m \le m+n_p-1$, and $\{x_i \ : \ i \in [1,m]\} \in \binom{[0,m+n_p-1]}{m}$. In the same way, the labels $\{y_i \ : \ i \in [1,m]\}$ of the m vertical steps in the path q are elements of $\binom{[0,m+n_q-1]}{m}$. Vice versa, the m-subsets of $[0,m+n_p-1]$ and $[0,m+n_q-1]$ correspond to a unique pair of paths p,q.

If we interpret the sequence (x_i+1, y_i), $i=1,\ldots,m$ as the sequence of left turn coordinates we obtain a unique lattice path from the origin to $(m+n_p, m+n_q)$ with m left turns; vice versa, the left turn sequence of any such path defines a unique element from $\binom{[0,m+n_p-1]}{m} \times \binom{[0,m+n_q-1]}{m}$. ■

Corollary 11 *There exists a bijection between $\binom{[0,m+n]}{m+1} \times \binom{[1,m+n+1]}{m+1}$ and*

1. *pairs of $\{\rightarrow\uparrow\}$-paths, both starting at the origin and ending at the common point $(n,m+1)$.*

2. *pairs u,b of $\{\rightarrow\uparrow\}$-paths, both starting at the origin and ending at the common point $(n+1,m+1)$ such that u ends and b begins with a horizontal step (in the case of nonintersecting pairs, u would be the upper and b the bottom path).*

3. *diagonal diffusion walks from $(0,0)$ to $(n-m-1, n-m-1)$ in $n+m+1$ steps, weakly staying inside the rectangle $-m-1 \le x \le n$ and $-m-1 \le y \le n$.*

4. *$\{\rightarrow\uparrow\}$-paths, starting at the origin and ending at $(m+n+1, m+n+1)$, taking $m+1$ right turns ($\begin{smallmatrix} \circ & \rightarrow \\ \uparrow & \end{smallmatrix}$).*

5. *$\{\rightarrow\uparrow\}$-paths, starting at the origin and ending at $(m+n+1, m+n+1)$, taking $m+1$ left turns ($\begin{smallmatrix} & \uparrow \\ \rightarrow & \circ \end{smallmatrix}$).*

Proof. For the terminology we refer to the proof of Lemma 10. Take any pair u', b' of $\{\rightarrow\uparrow\}$-paths from the origin to the common endpoint $(n, m+1)$. By Lemma 10 such pairs can be bijectively mapped onto $\binom{[0,m+n]}{m+1} \times \binom{[0,m+n]}{m+1}$. Make b' into the "$b$ottom" path b by inserting a $\rightarrow$ step at the beginning of b, and u' into the $u$pper path u by appending a $\rightarrow$ step at the end of u (the paths may still intersect; they end at $(n+1, m+1)$). The vertical label subsets are now in $\binom{[0,m+n]}{m+1} \times \binom{[1,m+n+1]}{m+1}$, which shows parts 1 and 2. For part 3, interpret u' as the vertical, and b' as the horizontal walk defining a diagonal diffusion as in Fig. 3. The horizontal (vertical) walk moves $m+1$ steps to the left (downwards) and n steps to the right (upwards), which defines the boundary for the resulting diagonal diffusion. Parts 4 and 5 follow directly from Lemma 10 ■

In a staircase polygon to $(n+1, m+1)$ the upper and bottom paths make a pair u, b as in the above bijection, with the additional condition of no common points except at the beginning and end. The $m+1$ positions x_i and y_i of the vertical steps in the sequence of all steps determine the whole pair u, b. Note that $x_1 = 0$, $x_{m+1} \leq m+n$ and $y_1 \geq 1$, $y_{m+1} = m+n+1$ in every such staircase polygon. Therefore we disregard x_1 and y_{m+1}, and consider only $\{x_{i+1} \mid 1 \leq i \leq m\} \in \binom{[1,m+n]}{m}$ and $\{y_i \mid 1 \leq i \leq m\} \in \binom{[1,m+n]}{m}$. When the bottom path takes the i-th vertical step, it has taken $y_i - i + 1$ horizontal steps; the i-th vertical step leads from the vertex $(y_i - i + 1, i-1)$ to the vertex $(y_i - i + 1, i)$, for $i = 1, \ldots, m+1$. Exchange x with y and the same holds for the upper path u. The pair u, b is nontouching; when the bottom path b moves upwards, to $(y_i - i + 1, i)$, it must stay below the upper path, hence $x_{i+1} - (i+1) + 1 < y_i - i + 1$, i.e.,

$$x_{i+1} \leq y_i \tag{33}$$

for all $i = 1, \ldots, m$. This condition (together with $x_1 = 0$ and $y_{m+1} = m+n+1$) characterizes the staircase polygons. We now map them bijectively onto lattice paths enumerated by left turns in the augmented third hexadecant, where $x - 1 \leq y \leq 2x$. Instead of creating $m+1$ turns at (x_i, y_i) as described in the proof of Lemma 10, we place only m left turns at $(x_{i+1}, y_i - 1)_{i=1,\ldots,m}$ $\in \binom{[1,m+n]}{m} \times \binom{[0,m+n-1]}{m}$, because x_1 and y_{m+1} are fixed in any staircase polygon. The image path runs from $(0,0)$ to $(n+m, n+m)$ and stays weakly above $y = x-1$ because this condition holds at the end point and at all left turns, $y_i - 1 \geq x_{i+1} - 1$ (see (33)).

We said in the previous subsection that a staircase polygon to $(n+1, m+1)$ is in the augmented second octant iff the bottom path b stays weakly above $y = x - 1$. To keep b weakly above $y = x - 1$ we need $i - 1 \geq y_i - i$, i.e., $y_i + 1 \leq 2i$ for $i = 1, \ldots, m+1$. In the corresponding lattice path the sequence (s_i, t_i) of left turns must satisfy the condition $i \geq 1 + t_i/2$ for $i = 1, \ldots, m$.

The condition $i \geq t_{i+1}/2$ is equivalent to the restriction that every point (v, w) on the path is reached with at least $w/2$ left turns; equivalently, the path stays weakly below $y = 2x$. From $x - 1 \leq y \leq 2x$ for all points (x, y) on the path follows that the path stays in the augmented third hexadecant. Denote by $h(v, w; i)$ the number of $\{\rightarrow, \uparrow\}$-paths from the origin to (v, w) in the augmented third hexadecant with i left turns. We have shown that $h(n+m, n+m; m) = S(n+1, m+1) = O_{m+n}(0, m-n) = 6\frac{(m+n)!(m+n+2)!}{n!(n+1)!(m+2)!(m+3)!}\binom{m-n+3}{3}$ for all $m \geq n$. It is also easy to verify that $h(i, 2i; i) = C_i$, the i-th Catalan number. From $h(k, k; k-n) = O_k(0, k-2n)$

and (8) follows that $C_{\lfloor (k+1)/2 \rfloor} C_{\lfloor 1+k/2 \rfloor}$ ordinary $\{\rightarrow\uparrow\}$-paths stay in the augmented third hexadecant and end at (k,k) (independent of the number of left turns).

References

[1] André, D. (1887). Solution directe du problème résolu par M. Bertrand, *C. R. Acad. Sci. Paris*, **105**, 436-437.

[2] Biane, P. (1992). Minuscule weights and random walks on lattices, in *Quantum probability & related topics*, (Ed.: L. Accardi), World Sci. Publishing, River Edge, NJ, 51–65.

[3] Bousquet-Mélou, M. (2002). Counting walks in the quarter plane, *Trends in Mathematics*, 49-67, Birkhäuser, Basel.

[4] Csáki, E. (1997). Some results for two-dimensional random walk. *Advances in Combinatorial Methods and Applications to Probability and Statistics* (Ed.: N. Balakrishnan), 115-124, Boston, Birkhäuser.

[5] Conway,A., Delest, M., and Guttmann,A.J. (1997). The number of three-choice polygons. *Math. Comput. Modelling*, **26**, 51-58.

[6] Delest, M.-P. and Fedou, J.M. (1993). Enumeration of skew Ferrer's diagrams, Discrete Math., **112**, 65-79.

[7] Delest, M. and Viennot, G. (1984). Algebraic languages and polyominoes enumeration. *Theor. Comput. Sc.* **34**, 169-206.

[8] Feller, W. (1968). *An Introduction to Probability Theory and its Application.* Wiley, New York.

[9] Fisher, M.E. (1984). Walks, walls, wetting and melting, *J. Stat. Phys.* **34**, 667-729.

[10] Gessel, I.M. and Zeilberger, D. (1992). Random walk in a Weyl chamber, *Proc. Amer. Math. Soc.* **115**, 27-31.

[11] Gessel, I.M. and Viennot, X.G. (1989). Determinants, paths, and plane partitions, preprint, available at http://www.cs.brandeis.edu/~ira

[12] Gessel, I.M. and Viennot, G. (1985). Binomial determinants, paths, and hook length formulae. *Adv. in Math.* 58, 300–321.

[13] Gouyou-Beauchamps, D. (1986). Chemins sous-diagonaux et tableau de Young, in *Combinatoire Enumerative* (Montreal 1985), Lect. Notes Math. **1234**, 112-125.

[14] Grabiner, D.J. (2002). Random walk in an alcove of an affine Weyl group, and noncolliding random walks on an interval, *J. Combin. Theory Ser. A*, **97**, 285-306.

[15] Guy, R.K., Krattenthaler, C. and Sagan, B.E. (1992). Lattice paths, reflections, and dimension changing bijections. *Ars Combinatoria*, **34**, 3-15.

[16] Krattenthaler, C. (2002). Watermelon configurations with wall interaction: exact and asymptotic results. To appear in *J. of Statistical Planning and Inference.*

[17] Krattenthaler, C., Guttmann, A.J. and Viennot, X. G. (2000). Vicious walkers, friendly walkers and Young tableaux II: with a wall, *J. Phys. A: Math. Gen.* **33**, 8835-8866.

[18] Kreweras, G. and Niederhausen, H. (1981). Solution of an enumerative problem connected with lattice paths. *Europ. J. of Combinatorics* **2**, 55-60.

[19] Kreweras, G. (1965). Sur une classe de problèmes liés au treillis des partitions d'entiers, *Cahiers du B.U.R.O.* **6**, 5-105.

[20] Itoh, Y. and Maehara, H. (1998). A variation to the ruin problem. *Mathematica Japonica* **47**, 97-102.

[21] Janse van Rensburg, E.J. (2000). *The Statistical Mechanics of Interacting Walks, Polygons, Animals and Vesicles.* Oxford University Press, Oxford, UK.

[22] Lindström, B. (1973). On the vector representations of induced matroids, *Bull. London Math. Soc.* **5**, 85-90.

[23] McCrea, W.H. and Whipple, F.J.W. (1940). Random paths in two and three dimensions. *Proc. Royal Soc. Edinburgh* **60**, 281-298.

[24] MacMahon, P.A. (1916). *Combinatorial Analysis.* Reprinted by Chelsea, 1960.

[25] Mohanty, S.G. (1979). *Lattice Path Counting and Applications.* Academic Press, New York.

[26] Niederhausen, H. (2003). Enumeration of diffusion walks in the first octant by their number of contacts with the diagonal. Submitted to the *Electronic J. Combin.*

[27] Niederhausen, H. (1998). Planar random walks inside a rectangle, *Congr. Numerantium*, **132**, 125-144.

[28] Niederhausen, H. (2002). Planar walks with recursive initial conditions, *J. of Statistical Planning and Inference*, **101**, 229-253.

[29] Pergola, E., Pinzani, R., Rinaldi, S., and Sulanke, R. A. (2002). A bijective approach to the area of generalized Motzkin paths. *Adv. in Appl. Math.* **28**, 580–591.

[30] Sulanke, R. A. (2001). Bijective recurrences for Motzkin paths, Adv. in Appl. Math. **27**, 627–640.

[31] Zeilberger, D. (1983). Andre's reflection proof generalized to the many-candidate ballot problem, *Discrete Math.*, **44**, 325-326

References

[1] E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, A methodology for plane trees enumeration, *Discrete Mathematics*, 180 (1998) 45–64.

[2] E. Barcucci, A. Del Lungo, E. Pergola, Permutations with one forbidden subsequence of increasing length, *Proceedings of* 9^{th} *FPSAC*, Wien (1997) 49–60.

[3] E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, From C_n to $n!$: permutations avoiding $S_j(j+1)(j+2)$, *Proceedings of* 10^{th} *FPSAC*, Toronto (1998) 31–41.

[4] M. Bóna, Permutations avoiding certain patterns. The case of length 4 and some generalizations, *Discrete Mathematics*, 175 (1997) 55–67.

[5] M. Bóna, Exact enumeration of 1342–avoiding permutations; a close link with labelled trees and planar maps, *Journal of Combinatorial Theory Series A*, 80 (1997) 257–272.

[6] L. M. Butler, The q–log concavity of q–binomial coefficients, *Journal of Combinatorial Theory Series A*, 54 (1990) 53–62.

[7] F.R.K. Chung, R.L. Graham, V.E. Hoggat, M. Kleiman, The number of Baxter permutations, *Journal of Combinatorial Theory, Series A*, 24 (1978) 382–394.

[8] L. Comtet, *Advanced Combinatorics*, Reidel (1979).

[9] A. M. Garcia, J. B. Remmel, q–Counting rook configurations and a formula of Frobenius, *Journal of Combinatorial Theory Series A*, 41 (1986) 246–275.

[10] I. M. Gessel, Symmetric functions and P–recursiveness, *Journal of Combinatorial Theory Series A*, 53 (1990) 257–285.

[11] H. W. Gould, The q–Stirling numbers of first and second kinds, *Duke Math. Journal*, 28 (1961) 281–289.

[12] O. Guibert, Combinatoires des permutations a motifs exclus en liaison avec mots, cartes planaires et tableaux de Young, *Thèse de l'Univeristé de Bordeaux I* (1996).

[13] P. Leroux, Reduced matrices and q–log concavity properties of q–Stirling numbers, *Journal of Combinatorial Theory Series A*, 54 (1990) 64–84 (1990).

[14] A. De Médicis, P. Leroux, A unified combinatorial approach for q-(and p,q–) Stirling numbers, *Journal of Statistical Planning and Inference*, 34 (1993) 89–105.

[15] A. De Médicis, P. Leroux, Generalized Stirling numbers, convolution formulae and p,q–analogues, *Canadian Journal of Mathematics*, 47 (1995) 474–499.

[16] S.C. Milne, A q–analog of restricted growth functions, Dobinski's equality, and Charlier polynomials, *Trans. Amer. Math. Soc.*, 245 (1978) 89–118 (1978).

[17] S.C. Milne, Restricted growth functions, rank row matchings of partition lattices, and q–Stirling numbers, *Advanced in Mathematics*, 43 (1982) 173–196.

[18] S.C. Milne, Mapping of subspaces into subsets, *Journal of Combinatorial Theory Series A*, 33 (1982) 36–47.

[19] A. Regev, Asymptotic values for degrees associated with strips of Young diagrams, *Advances in Mathematics*, 41 (1981) 115–136.

[20] J. Riordan, *An introduction to combinatorial analysis*, Wiley (1958).

[21] R. Simion, F. W. Schmidt, Restricted permutations, *European Journal of Combinatorics*, 6 (1985) 383–406.

[22] N. Sloane, S. Plouffe, *Encyclopedia of Integer Sequences*, Academic Press, New York (1995).

Proof. The value of $a_{n,m}^{(k,j)}(q)$ in the case of $2 \le k \le j$ is an immediate consequence of (6.6). In the case of $k \ge j+1$, by applying (6.6) and (6.7) we can write:
$a_k^j(x,y,q) =$
$$= x^{k-1} \left(\sum_{t=0}^{k-j} \binom{k-j}{t} q^t \left([j-1]_q\right)^t y^{k-j-t} \right) \left(\sum_{i=0}^{j-1} c_q[j-1,i] q^{j-1-i} y^i \right) \left(\sum_{i \ge 0} S_q[i+k-j, k-j] \left(xq^j\right)^i \right),$$
and the second equality can then be easily proved. □

Let us now examine the polynomials $a_{n,m}^{(k,j)}(q)$ for some particular values of the parameter j.

- If $j=1$, then equation (6.9) should be used and the result is different from 0 if and only if the exponent of $([j-1]_q) = ([0]_q)$ is zero, that is $k = m+1$. Once n and m are fixed the only possibility is:
$$a_{n,m}^{(m+1,1)}(q) = S_q[n,m] q^{n-m}.$$
This confirms the results of Section 3 for the number of left–to–right minima in Bell permutations of length n. Moreover it shows that the classical q–analogue of the Stirling numbers of the second kind, $S_q[n,m]$, multiplied by q^{n-m}, count restricted permutations according to first kind inversions.

- If $j=2$ then equations (6.8) and (6.9) give:
$$\begin{cases} a_{1,1}^{(2,2)}(q) &= 1, \\ a_{n,m}^{(k,2)}(q) &= \binom{k-2}{m-1} S_q[n-1,k-2] q^{2n+1-k-m}, \quad k \ge 3. \end{cases}$$
By summing over k and m we obtain the polynomials for the permutations with forbidden subsequences $(5\bar{1}432, 5\bar{1}423)$ or $(5\bar{2}431, 5\bar{2}413)$ of length n according to the number of their second kind inversions:
$$\sum_{k \ge 2} \sum_{1 \le m \le n} a_{n,m}^{(k,2)}(q) = \sum_{k=0}^{n-1} S_q[n-1,k] q^{2(n-1-k)} (1+q)^k, \quad n \ge 2. \tag{6.10}$$
This expression reduces to a value of the $(n-1)$–th Bell polynomials: $\sum_{k \ge 0} 2^k S(n-1,k)$ for $q=1$ as said in Section 5, so (6.10) defines a q–analogue for these numbers.

- If $j=\infty$ then equation (6.8) gives:
$$a_{n,m}^{(n+1,\infty)}(q) = c_q[n,m] q^{n-m}, \qquad n \ge 1.$$
This means that the classical q–analogue of the first kind signless Stirling numbers, $c_q[n,m]$, multiplied by q^{n-m} correspond to q–counting the inversions in the permutations. A direct combinatorial explanation can be given.

The meaning of "Stirling numbers interpolation" lies in the observation that the permutations of length n having m left–to–right minima are counted by the second kind Stirling numbers for $j=1$ and by the first kind Stirling numbers for $j=\infty$. In the intermediate cases this number, $p_{n,m}^{(j)}$, is such that $S(n,m) \le p_{n,m}^{(j)} \le c(n,m)$, $c(n,m)$ denoting the first kind signless Stirling numbers, and it verifies the recursive relation:
$$p_{n,m}^{(j)} = p_{n-1,m-1}^{(j)} + \sum_{k=2}^{n} (k-1) a_{n-1,m}^{(k,j)}(1), \tag{6.11}$$
where:
$$a_{n-1,m}^{(k,j)}(1) = \begin{cases} c(n-1,m), & \text{for} \quad 2 \le k = n \le j, \\ S(n-j,k-j)(j-1)^{k-j-m} \sum_{i=0}^{j-1} \binom{k-j}{m-i} c(j-1,i))(j-1)^i, & \text{for} \quad k \ge j+1. \end{cases}$$
Note that the sum in equation (6.11) reduces to a single term if $j=1$ (namely, the term for $k=m+1$) and if $j=\infty$ (namely, the term for $k=n$). These two cases yield classical recurrence relations for the Stirling numbers of the second kind, $S(n,m)$, and unsigned first kind, $c(n,m)$, respectively.

A permutation π with $k > j$ active sites is the father of k permutations obtained by inserting the next element into each of its active sites: $i_1, i_2, \ldots, i_{k-j}, (n(\pi)-(j-2)), \ldots, (n(\pi)+1)$. Again the parameters in the new permutations change as follows:

- for the leftmost $(k-j)$ active sites:
  $\mathrm{n}(\pi^l) = \mathrm{n}(\pi)+1$; $\mathrm{rm}(\pi^l) = \mathrm{rm}(\pi)$; $\mathrm{inv}_j(\pi^l) = \mathrm{inv}_j(\pi) + k - l$;
  and the number of active sites is unchanged;

- for the remaining active sites:
  $$\mathrm{n}(\pi^l) = \mathrm{n}(\pi)+1; \quad \begin{cases} \mathrm{rm}(\pi^l) = \mathrm{rm}(\pi), & k-j+1 \le l \le k-1, \\ \mathrm{rm}(\pi^k) = \mathrm{rm}(\pi)+1, & l = k; \end{cases} \quad \mathrm{inv}_j(\pi^l) = \mathrm{inv}_j(\pi) + k - l;$$
  and the number of active sites increases by one unit.

Let $a_k^j(x,y,q)$ be the generating function of S^j–permutations with k active sites according to their length (x), the number of left–to–right minima (y) and the number of j–th kind inversions (q). The above considerations on the parameter modifications yield the following recursive relations for $a_k^j(x,y,q)$:

$$\begin{cases} a_2^j(x,y,q) &= xy, \\ a_k^j(x,y,q) &= xy a_{k-1}^j(x,y,q) + xq[k-2]_q a_{k-1}^j(x,y,q), & 3 \le k \le j, \\ a_k^j(x,y,q) &= xy a_{k-1}^j(x,y,q) + xq[j-1]_q a_{k-1}^j(x,y,q) + xq^j[k-j]_q a_k^j(x,y,q), & k \ge j+1; \end{cases} \tag{6.5}$$

where $[i]_q$ denotes the classical q–analogue of i that is $[i]_q = 1 + \cdots + q^{i-1} = \frac{q^i-1}{q-1}$.
Solving the recursions, we obtain the following:

Proposition 6.1 *The generating function $a_k^j(x,y,q)$ for S^j–permutations verify:*

$$\begin{aligned} a_k^j(x,y,q) &= x^{k-1} \prod_{i=0}^{k-2} (y + q[i]_q), & 2 \le k \le j; \\ a_k^j(x,y,q) &= x^{k-1} (y+q[j-1]_q)^{k-j} \frac{\prod_{i=0}^{j-2} (y+q[i]_q)}{\prod_{i=1}^{k-j} (1-xq^j[i]_q)}, & k \ge j+1. \end{aligned}$$

The coefficient $[x^n y^m]\, a_k^j(x,y,q)$ gives a polynomial in q–counting the S^j–permutations with length n, having m left–to–right minima and k active sites, according to their number of j–th kind inversions. Let $c_q[h,i]$ and $S_q[h,i]$ be the classical q–analogues of the (signless) Stirling numbers of the first and second kind respectively, as defined in [14, 15]. These polynomials are characterized by:

$$\sum_{i=0}^{h} c_q[h,i] z^{h-i} y^i = \prod_{i=0}^{h-1} (y + [i]_q z), \tag{6.6}$$

$$\sum_{i \ge h} S_q[i,h] z^{i-h} = \prod_{i=1}^{h} \frac{1}{1-z[i]_q}. \tag{6.7}$$

Corollary 6.2 *Let $a_{n,m}^{(k,j)}(q) = [x^n y^m]\, a_k^j(x,y,q)$, $m \le k-1$; then we have:*

$$a_{n,m}^{(k,j)}(q) = \delta_{n,k-1} c_q[k-1,m] q^{k-1-m}, \quad 2 \le k \le j; \tag{6.8}$$

$$a_{n,m}^{(k,j)}(q) = S_q[n+1-j, k-j] q^{j(n+1-k)+(k-m-1)} ([j-1]_q)^{k-j-m} \sum_{i=0}^{j-1} \binom{k-j}{m-i} c_q[j-1,i] ([j-1]_q)^i, \quad k \ge j+1; \tag{6.9}$$

where $\delta_{i,j}$ is the Kronecker delta.

We have the following parameter correspondences:

Bicolored set partitions	**S^2-permutations**
cardinality of the partitioned set	length of the permutations+1
number of black blocks	number of left–to–right minima-1
number of blocks	number of second kind left–to–right minima-1
number of red blocks + number of weighted inversions	number of second–kind inversions

We do not know of a direct bijection between these two classes of structures.

If $j = \infty$, then we obtain all permutations and $n!$ appears; for each other value of $j \geq 3$ we obtain sequences of numbers such the n–th term of each of them is between B_n and $n!$ (see Fig. 3). These sequences do not appear in the Sloane–Plouffe book [22]: "The Encyclopedia of Integer Sequences", and verify the following property: the $(j+2)$–th number of the $(j+1)$–th sequence is obtained from the $(j+2)$–th number of the (j)–th sequence by adding $j!$.

Index	Family of permutations	Numbers
j = 1	$S_n(4\bar{1}32)$	1 2 5 15 52 203 877 4140 21147 115975 678570 . . .
		+1!
j = 2	$S_n(5\bar{1}432,5\bar{1}423) \cup S_n(5\bar{2}431,5\bar{2}413)$	1 2 6 22 94 454 2430 14214 89918 610182 4412798 . . .
		+2!
j = 3	$S_n(6\bar{1}5432,6\bar{1}5423,6\bar{1}5342,6\bar{1}5324,6\bar{1}5243,6\bar{1}5234) \cup$ $S_n(6\bar{2}5431,6\bar{2}5413,6\bar{2}5341,6\bar{2}5314,6\bar{2}5143,6\bar{2}5134) \cup$ $S_n(6\bar{3}5421,6\bar{3}5412,6\bar{3}5241,6\bar{3}5214,6\bar{3}5142,6\bar{3}5124)$	1 2 6 24 114 618 3732 24702 177126 1363740 11195286. . .
		+3!
j = 4		1 2 6 24 120 696 4536 32568 254136 2133816 19130040 . . .
	. . .	+4!
j = 5		1 2 6 24 120 720 4920 37320 309120 2763720 26440920. . .
.	. . .	. . .
.	. . .	. . .
j = ∞	S_n	1 2 6 24 120 720 5040 40320 362880 3628800 39916800. . .

Figure 3: Table of permutations.

6 Enumerative results for S^j-permutations

For each j, we are interested in the enumeration of the permutations in S^j according to their length, the number of left–to–right minima and the number of j–th kind inversions. The reason we introduce this parameter is to give a combinatorial interpretation of the q–analogue that we obtain in a natural way from (4.3) by giving a "weight" to the label on the right–hand side of each inductive step in (4.3). More precisely the i–th child of a label (k) q–counts for $k-i$; the result of this "weight assignment procedure" is expressed in Proposition 6.1.

Let $\pi \in S^j$ and π^l be the permutation obtained from π by inserting the next element into the l^{th} active site, from left to right. We denote the length of π by $\mathrm{n}(\pi)$, the number of its left–to–right minima by $\mathrm{rm}(\pi)$ and the number of its j–th kind inversions by $\mathrm{inv}_j(\pi)$.

From (4.3) we deduce that the sites in a permutation $\pi \in S^j$ with length $k-1 \leq j$ are all active, so π is the father of k permutations obtained by inserting the element k into its first, second, ..., k^{th} sites. The parameters change as follows in the new permutation:

$$\mathrm{n}(\pi^l) = \mathrm{n}(\pi)+1; \quad \begin{cases} \mathrm{rm}(\pi^l) = \mathrm{rm}(\pi), & 1 \leq l \leq k-1, \\ \mathrm{rm}(\pi^k) = \mathrm{rm}(\pi)+1, & l = k; \end{cases} \quad \mathrm{inv}_j(\pi^l) = \mathrm{inv}_j(\pi) + k - l;$$

and the number of active sites becomes $k+1$.

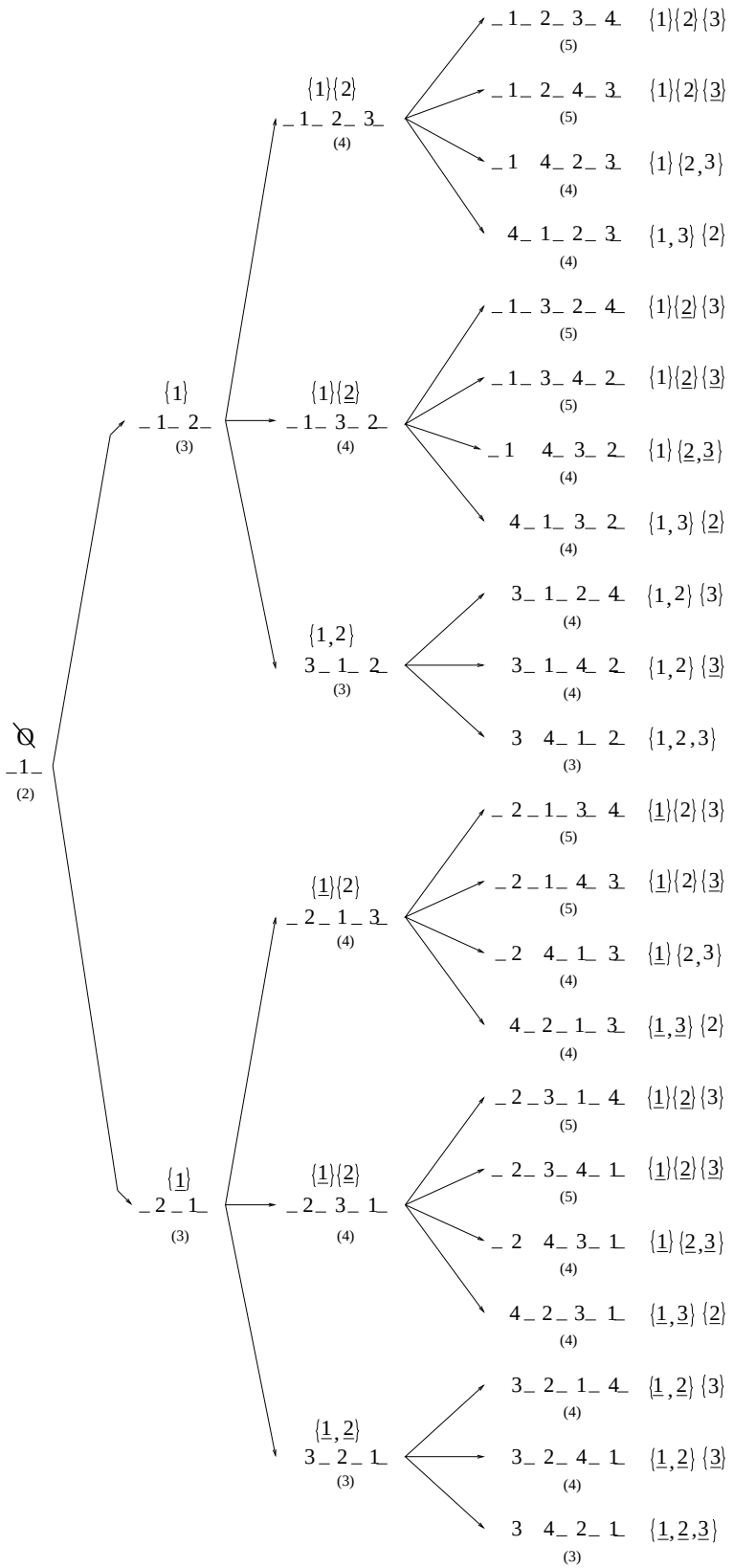


Figure 2: The first four levels of the generating tree for permutations in $S^2 = \bigcup_{n\geq 1} (S_n(5\bar{1}432, 5\bar{1}423) \bigcup S_n(5\bar{2}431, 5\bar{2}413))$ and the constructive bijection with the bicolored set partitions.

Proposition 4.1 *Let $\pi \in \bigcup_{i=1}^{j} S_n\left(\bar{F}_i^j\right)$, $j \geq 1$, be a permutation with $k \geq 2$ active sites: $i_1, \ldots, i_{k-j}$, $(n-(j-2)), \ldots, (n+1)$. Then the number of active sites does not change in the permutation obtained by inserting $(n+1)$ into the site i_t, $t = 1, \cdots, k-j$; the permutation obtained from π by inserting $(n+1)$ into the site $(n+1-t)$, $0 \leq t \leq j-1$, has $k+1$ active sites: $i_1, \ldots, i_{k-j}$, $(n-(j-2)), \ldots, (n+1), (n+2)$.*

Proof. The j rightmost sites of π that is $(n-(j-2)), \ldots, (n+1)$ are always active, if they exist, because the insertion of $(n+1)$ into the site $(n+1-t)$, $0 \leq t \leq j-1$, cannot create any occurrence of any forbidden subsequences. Thus, the element $(n+1)$ has exactly t elements on its right so it is the first and largest element of a sequence of length $(t+1) \leq j$ and any unbarred forbidden subsequence has length $(j+2)$. The site i_1 is still active if and only if a sequence of indices $i_2, \cdots, i_{j+1}$, $i_1 + 1 \leq i_2 < \cdots < i_{j+1} \leq n$, such that $\pi(i_1) > \pi(i_l)$, $2 \leq l \leq j+1$, does not exist, meaning that an active site must lie on the left of a j–th kind left–to–right minima.

Let the k active sites of a permutation π be $i_1, \ldots, i_{k-j}$, $(n-(j-2)), \ldots, (n+1)$. Observe that the site i_{k-j} is the site $(n-j+1)$, but it behaves as the sites i_t, $1 \leq t \leq k-j-1$. The active sites of the permutation obtained from π by inserting $(n+1)$ into the site $(n+1-t)$, $0 \leq t \leq j-1$, are: $i_1, \ldots, i_{k-j}$, $(n-(j-2)), \ldots, (n+1), (n+2)$. The site $(n+2-t)$, $0 \leq t \leq j$, is trivialy active; the remaining active sites are those that were active in the original permutation as the new inserted element $(n+1)$ plays no role in the creation of any forbidden subsequence. The sites that in π were inactive are always inactive because $(n+1)$ cannot play the role of the barred element in a forbidden subsequence.

By inserting $(n+1)$ into the site i_t, $1 \leq t \leq k-j$, the active sites in the new permutation are: $i_1, \ldots, i_{t-1}, (i_t+1), \ldots, (i_{k-j}+1), (n-(j-3)), \ldots, (n+2)$. The site on the left of $(n+1)$ is inactive because we would have $(n+2)(n+1)\sigma$, $|\sigma| = j$, which is forbidden. The sites that were active in π are always active because if they do not create any forbidden subsequences in π, then they do not create any problem in the new permutation and the inactive sites in π are still inactive in the new permutation. □

5 Bicolored set partitions and permutations

In Section 3 we illustrated the case $j = 1$, that is we showed that $4\bar{1}32$–avoiding permutations are counted by the Bell numbers and gave a bijection with set partitions. For $j = 2$ we show that the number of $(5\bar{1}432, 5\bar{1}423)$ or $(5\bar{2}431, 5\bar{2}413)$–avoiding permutations are the values of Bell polynomials whose $(n-1)$–th term is defined by $\sum_{k\geq 0} 2^k S(n-1,k)$ ([22], sequence M1662). These numbers count bicolored set partitions (that is to say each block can be red or black) and there is a bijection between these two classes of structures. This correspondence can be easily obtained by applying the succession rules

$$\begin{cases} basis: & (2) \\ inductive\ \ step: & (2) \rightarrow (3)(3), \\ inductive\ \ step: & (k) \rightarrow (k)^{k-2}(k+1)^2, \quad k > 2, \end{cases} \tag{5.4}$$

to the bicolored set partitions, obtaining a constructive bijection. In bicolored set partitions the label k represents the number of blocks plus two. Given an n–element set bicolored partition with $k-2$ blocks, labeled by (k), we can add on its right the block $\{(n+1)\}$ that can be red or black and in this case the number of blocks becomes $k-1$, so the label of these new partitions is $(k+1)$; or we can insert $(n+1)$ into any of the blocks of the partition, the color remaining the same. This bijection is represented in Fig. 2, where the red blocks are those with the underlined elements. In an n–element bicolored set partition with k blocks, let i be a number belonging to the m^{th} block, $1 \leq m \leq k$, which is different from the minimum element of the block. We then define the *weighted inversions* related to i as: the number of blocks on the right of its own block such that their minimum element is smaller than i, plus 2. The total weighted inversions of a partition is given by the sum, over each i satisfying the above condition, of its weighted inversions.

4 Generalized Bell permutations

In this Section we introduce a parameter j in the succession rule (3.1) giving the Bell numbers. Each value of j yields a number sequence such that the n–th term lies between B_n and $n!$. We are interested in characterizing the permutations enumerated by each number sequence.

Let us carefully examine the succession rule (3.1): the "exponents" of the terms on the right hand side of the inductive step are $k-1$ for the label (k) and 1 for the label $(k+1)$. We can make these "exponents" depend on a parameter j, thus giving the "exponent" $k-j$ to the label (k) and j to the label $(k+1)$; moreover if $k \leq j$ then only the label $(k+1)$ is obtained exactly k times. The exact form of the succession rule we obtain is

$$\begin{cases} basis: & (2) \\ inductive\ \ step: & (k) \rightarrow (k+1)^k, \qquad k \leq j \\ inductive\ \ step: & (k) \rightarrow (k)^{k-j}(k+1)^j, \quad k > j. \end{cases} \tag{4.3}$$

It is easy to verify that if $j=1$, then the succession rule (4.3) reduces to (3.1).

We recall that the "exponent" of a label in a succession rule means the number of times the label must be repeated. For example, the "exponent" of the label $(k+1)$ in the first inductive step is k because k is less or equal to j. Also the number of terms on the right hand side of the inductive step in a succession rule must be exactly k. The idea is to perform (4.3) on permutations and try to characterize the class we obtain. The first step is to give an interpretation of (4.3) in terms of active sites in a permutation; we have to decide how the active sites are modified when a new element is added into a permutation with a fixed number of active sites. The second step is to describe the resulting permutations in terms of forbidden subsequences. We refer to the first active site as the leftmost active site in the permutation and so on, and we make the following choices:

- if a new element is inserted in the l^{th} active site, $1 \leq l \leq k-j$, then the site on the left of the inserted element is inactive and the number of active sites do not change in the new permutation,
- if a new element is inserted in the l^{th} active site, $k \geq l \geq k-j+1$, then the site on the left of the inserted element is also active and the number of active sites grows by one.

In other words, the permutation obtained from π of length n with $k-1$ left–to–right minima of j–th kind, by inserting $(n+1)$ into its j rightmost active sites has its number of active sites increased by one; while the permutation obtained by inserting $(n+1)$ into the remaining active sites has an unchanged number of active sites.

We now show that the permutations we obtain avoid the subsequences $(j+2)(j+1)\sigma$ where $\sigma \in S_j$ and the elements corresponding to $(j+2)$ and $(j+1)$ are consecutive. In terms of permutations with forbidden subsequences such a condition is given by the union of j sets of permutations with forbidden subsequences: $\bigcup_{i=1}^{j} S\left(\bar{F}_i^j\right)$ where $\bar{F}_i^j$ is a set of barred subsequences $\bar{\tau} = (j+3)\bar{i}(j+2)\sigma_i$ with σ_i a permutation on the set $\{(j+1), \cdots, (i+1), (i-1), \cdots, 1\}$; so $\left|\bar{F}_i^j\right| = j!$ and $|\bar{\tau}| = j+3$.

Example 4.1 Let $j=2$ then $1 \leq i \leq 2$. The set $\bar{F}_2^2$ obtained for $i=2$ is $\{5\bar{2}431, 5\bar{2}413\}$.

Let us note that in the union i can assume all values between 1 and j. This means that we are not interested in the value of the element lying between $(j+3)$ and $(j+2)$, but at least one element must exist between $(j+3)$ and $(j+2)$. Such a condition avoids subpatterns of two adjacent decreasing elements having at least j smaller elements on their right. Moreover, i cannot be equal to $(j+1)$ because the subsequence $(j+3)(j+1)\sigma$ (σ being a permutation of length j) is of the forbidden type. Let S^j be the class of permutations defined by $S^j = \bigcup_{n \geq 1} \bigcup_{i=1}^{j} S_n\left(\bar{F}_i^j\right)$. We show that the class S^j has a recursive construction described by (4.3).

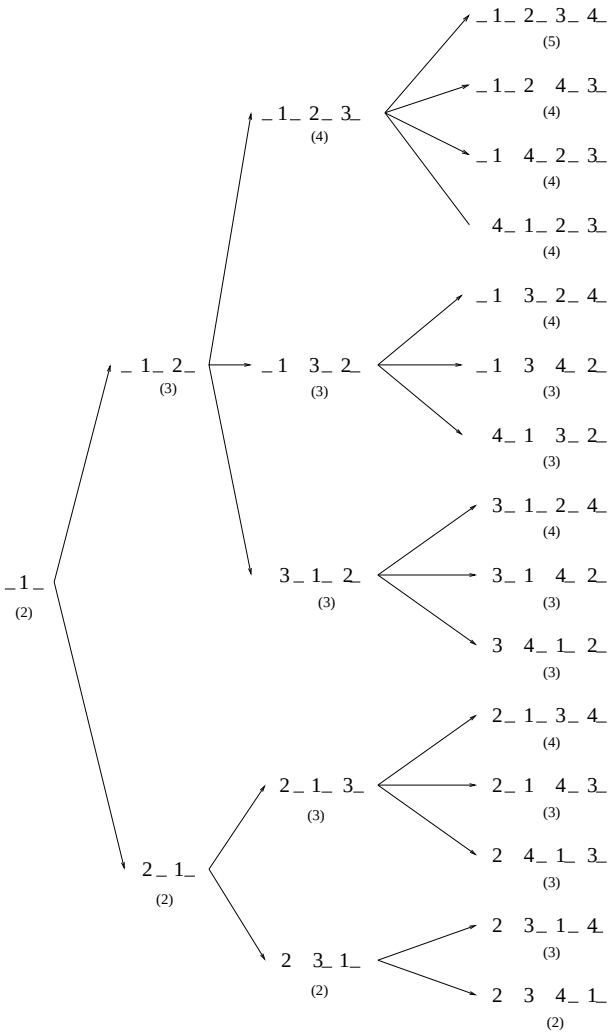


Figure 1: The generating tree for $4\bar{1}32$–avoiding permutations.

Let us note that the active sites of a permutation belonging to $S(4\bar{1}32)$ are the sites on the immediate left of each left–to–right minimum and the one on the right of the last element. Therefore the number of active sites in a permutation is the number of its left–to–right minima plus one.

The second approach we propose in order to generate $S(4\bar{1}32)$ permutations, is to construct $S_{n+1}(4\bar{1}32)$ starting from $S_1(4\bar{1}32)$, $S_2(4\bar{1}32)$, $\cdots$, $S_n(4\bar{1}32)$. The permutations in $S_{n+1}(4\bar{1}32)$ with k left–to–right minima can be obtained in the following way. For each value m such that $0 \leq m \leq n$:

- extract a subset of m elements from the set $\{2, \cdots, n+1\}$,
- construct the permutations in $S_m(4\bar{1}32)$ with $(k-1)$ left–to–right minima,
- add the element 1 on its left,
- place on the left of 1 the remaining $(n-m)$ elements in an increasing order.

The increasing order is required to avoid the forbidden subsequence 321 that would be obtained if there were two elements $\pi(t_1)$, $\pi(t_2)$ such that $\pi(t_1) > \pi(t_2)$ and $t_1 < t_2 \leq n-m$. This means that:

$$p_{n+1,k+1} = \sum_{m=0}^{n} \binom{n}{m} p_{m,k}, \qquad k \leq m.$$

As $p_{n,k} = S(n, k-1)$ we obtain a combinatorial interpretation of the well known relation involving the second kind signless Stirling numbers [8] by means of Bell permutations.

Example 3.1 Let us consider the following partition of an 8–element set into three blocks: $\{1,5,8\}$ $\{2,3\}$ $\{4,6,7\}$. The new representation described above is the permutation: 5 8 $\underline{1}$ 3 $\underline{2}$ 6 7 $\underline{4}$ which has exactly three (underlined) left–to–right minima.

Proposition 3.1 *Permutations in $S_n(4\bar{1}32)$ are counted by the n–th Bell number* (this is the reason why we call them Bell permutations)*, and $S(n,k)$ counts the permutations in $S_n(4\bar{1}32)$ with k left–to–right minima.*

Proof. Following the previous discussion, we observe that the permutation π obtained from a partition of an n–element set contains a subsequence of type $\hat{\tau} = 321$ if and only if it is a subsequence of any sequence of type $\tau = 4132$. In other words, three indices i_1, i_2, i_3, $i_1 < i_2 < i_3$, such that $\pi(i_1) > \pi(i_2) > \pi(i_3)$ can be found in π if and only if it exists an index j, $i_1 < j < i_2 < i_3$, such that $\pi(i_1)\pi(j)\pi(i_2)\pi(i_3)$ is of type 4132. Such a condition is described by the forbidden subsequence $4\bar{1}32$. Let $\pi(i_1), \cdots, \pi(i_k)$ be the k left–to–right minima of π, then $\pi(i_l)$, $1 \leq l \leq k$, is the first element of the l^{th} block in the corresponding partition; while the elements between $\pi(i_{l-1})$ and $\pi(i_l)$ are all the elements belonging to the l^{th} block of the partition. Permutations in $S_n(4\bar{1}32)$ with k left–to–tight minima are counted by the Stirling numbers. So, $S_n(4\bar{1}32)$ is enumerated by the Bell numbers. □

The first construction we take into consideration for the class $S(4\bar{1}32)$ is a recursive construction which allows to obtain $S_{n+1}(4\bar{1}32)$, starting with $S_n(4\bar{1}32)$. It uses the concept of *active site* of a permutation (see Fig. 1: the active sites are represented by "_").

Proposition 3.2 *Let $\pi \in S_n(4\bar{1}32)$ be a permutation with $k \geq 2$ active sites, that is the sites i_1, i_2, ..., i_{k-2}, n and $(n+1)$. Then the number of active sites is still k in the permutation obtained by inserting $(n+1)$ into any active site different from the rightmost one; the permutation obtained from π by inserting $(n+1)$ into the site $(n+1)$ has $k+1$ active sites: i_1, i_2, ..., i_{k-2}, n, $(n+1)$ and $(n+2)$.*

Proof. Let $i_1 < i_2 < \cdots < i_{k-2} < n$ be the indices of the $k-1$ left–to–right minima of π, namely $\pi(i_1), \pi(i_2), \cdots, \pi(i_{k-2}), \pi(n)$. The active sites of π are the sites on the immediate left of each left–to–right minimum and on the right of the last element, that is, active sites of π are i_1, i_2, ..., i_{k-2}, n and $(n+1)$. Indeed, the insertion of $(n+1)$ into the site $(n+1)$ does not cause any occurrence of the forbidden subsequence 321; by inserting $(n+1)$ into the site l, $l = i_1, \cdots, i_{k-2}, n$ we can obtain the forbidden subsequences 321 if and only if there exist two indices t_1, t_2 such that $l < t_1 < t_2$ and $(n+1) > \pi(t_1) > \pi(t_2)$, but in this case $(n+1)\pi(l)\pi(t_1)\pi(t_2)$ is of type 4132. Each other site is inactive: if a site lies on the left of $\pi(i)$ that is not a left–to–right minimum, then there exists $i_1 > i : \pi(i) > \pi(i_1)$, and the insertion of $(n+1)$ on the left of $\pi(i)$ gives $(n+1)\pi(i)\pi(i_1)$, that is a decreasing sequence of length three, with $(n+1)$ and $\pi(i)$ adjacent elements and we get a forbidden subsequence 321. Observe that the insertion of $(n+1)$ into the site $(n+1)$ increases the number of left–to–right minima of π while each other insertion does not change this number in the permutation. □

If we classify the permutations of $S_n(4\bar{1}32)$, $n \geq 1$, according to their number of active sites then we can synthetically describe the obtained recursive construction by the succession rule:

$$\begin{cases} basis: & (2) \\ inductive\ \ step: & (k) \rightarrow (k)^{k-1}(k+1), \end{cases} \tag{3.1}$$

since $S_1(4\bar{1}32) = \{1\}$ has two active sites.

The expansion of this succession rule gives the generating tree of Fig. 1. Consequently if $p_{n,k} = |\{\pi \in S_n(4\bar{1}32) : \pi \text{ has } k \text{ active sites}\}|$ then

$$\begin{cases} p_{1,2} & = & 1, \\ p_{n+1,k} & = & p_{n,k-1} + (k-1)p_{n,k}, \quad 2 \leq k \leq (n+2), \end{cases} \tag{3.2}$$

which is the recursive relation of the Stirling numbers of the second kind [8] (replace $p_{n,k}$ by $S(n,k-1)$).

$1 \leq i \leq n$, and on the right of $\pi(n)$, so the site i is on the left of $\pi(i)$ and the site $(n+1)$ on the right of $\pi(n)$. The site i of $\pi \in S_n(F)$ is *active* if the insertion of $(n+1)$ into the position between $\pi(i-1)$ and $\pi(i)$ gives a permutation belonging to the set $S_{n+1}(F)$; otherwise it is said to be *inactive*.

Example 2.3 The permutation $\pi = 58132674 \in S_8(4\bar{1}32)$ has 4 active sites that is the sites: 3, 5, 8 and 9. Indeed the permutations: 589132674, 581392674, 581326794 and 581326749 belong to $S_9(4\bar{1}32)$, while the remaining sites are inactive, for example 581326974 has the subsequence 974 of type 321 but it is not a subsequence of a sequence of type 4132.

Let π be a permutation on $[n]$. The element $\pi(i)$, $1 \leq i \leq n$, is a *left–to–right minimum* if $\pi(i) < \pi(t)$, for all $t \in [i+1, n]$. This means that an index i_1, $i+1 \leq i_1 \leq n$, such that $\pi(i) > \pi(i_1)$ does not exist. We propose to generalize the concept of left–to–right minimum as follows: let π be a permutation on $[n]$; the element $\pi(i)$, $1 \leq i \leq n$, is a *j–th kind left–to–right minimum* if and only if a sequence of indices of length j: $i_1, \cdots, i_j$, $i+1 \leq i_1 < \cdots < i_j \leq n$, such that $\pi(i) > \pi(i_l)$, $1 \leq l \leq j$ does not exist. This implies that the j rightmost elements of π are trivialy j–th kind left–to–right minima. Of course a left–to–right minimum is the same as a first kind left–to–right minimum while each element of the permutation is an ∞–kind left–to–right minimum. Hence the number of ∞–kind left–to–right minima is the length of the permutation.

Example 2.4 The permutation $\pi = 58132674$ has:

- 3 *left–to–right minima*: $\pi(3) = 1$, $\pi(5) = 2$ and $\pi(8) = 4$;
- 6 *second kind left–to–right minima*: $\pi(3) = 1$, $\pi(4) = 3$ $\pi(5) = 2$, $\pi(6) = 6$, $\pi(7) = 7$ and $\pi(8) = 4$;
- 8 *∞–kind left–to–right minima.*

Let π be a permutation on $[n]$. An *inversion* is an ordered pair of indices: (s,t), $1 \leq s < t \leq n$, such that $\pi(s) > \pi(t)$. We say that the couple of indices (s,t), $1 \leq s < t \leq n$, such that $\pi(s) > \pi(t)$, is a *j–th kind inversion* if $\pi(t)$ is a j–th kind left–to–right minimum. Following this definition the classical concept of an inversion becomes an ∞–kind inversion, while the number of inversions with respect to the left–to–right minima are first kind inversions.

Example 2.5 The permutation $\pi = 58132674$ of Example 2.4 has:

- 9 *first kind inversions*: $(1,3)$,$(1,5)$,$(1,8)$,$(2,3)$,$(2,5)$,$(2,8)$,$(4,5)$,$(6,8)$,$(7,8)$;
- 13 *second kind inversions*: $(1,3)$,$(1,4)$,$(1,5)$,$(1,8)$,$(2,3)$,$(2,4)$,$(2,5)$,$(2,6)$,$(2,7)$,$(2,8)$,$(4,5)$,$(6,8)$,$(7,8)$;
- 13 *∞–kind inversions*: $(1,3)$,$(1,4)$,$(1,5)$,$(1,8)$,$(2,3)$,$(2,4)$,$(2,5)$,$(2,6)$,$(2,7)$,$(2,8)$, $(4,5)$,$(6,8)$,$(7,8)$.

3 Bell permutations and set partitions

The Stirling numbers of the second kind, denoted by $S(n,k)$, for $n \geq k \geq 0$, count the ways of partitioning a set of n objects into k nonempty subsets, called blocks. The number of partitions of an n–element set is given by the sum over k, $0 \leq k \leq n$, of $S(n,k)$; this defines the n–th Bell number, denoted by B_n [20]. For example, there are 7 ways of partitioning a 4–element set into two blocks: $\{1,2,3\}$ $\{4\}$; $\{1,2,4\}$ $\{3\}$; $\{1,3,4\}$ $\{2\}$; $\{1,2\}$ $\{3,4\}$; $\{1,3\}$ $\{2,4\}$; $\{1,4\}$ $\{2,3\}$; $\{1\}$ $\{2,3,4\}$, and the total number of partitions is $B_4 = \sum_{k=0}^{4} S(4,k) = 0+1+7+6+1 = 15$. Note that $S(0,0) = B(0) = 1$.

The standard representation of a given set partition consists in using the increasing order within each block and, in listing the blocks according to the increasing order of their minimum elements. We consider a new representation of the partition by moving the minimum element from the first to the last position in each block and then erasing the curly braces. The sequence of elements thus obtained is a permutation such that its (first kind) left–to–right minima are exactly the minimum elements of the blocks in the partition.

2 Notations and Definitions

In this Section we recall the concepts of generating tree for a set of succession rules and of permutations with forbidden subsequences. Moreover, we generalize some classical definitions about permutations.

The concept of generating tree was introduced in [7] for the study of Baxter permutations and later applied to the study of various permutations with forbidden subsequences by different authors. Generating trees and succession rules can be used in combinatorics to deduce enumerative results about various combinatorial objects [1].

A *generating tree* is a rooted, labeled tree in which the size and labels of the set of children of each node x are determined solely from the label of x. Thus, any particular generating tree can be specified by a set of succession rules, that is, a recursive definition, consisting of

1. the basis, the label of the root,

2. the inductive step, a set of succession rules that yields a multiset of labeled children which depends solely on the label of the parent.

A succession rule can be used to describe the growth of the objects to which it is related and also to obtain the number sequence counting the objects themselves. The introduction of a parameter, say j, in a succession rule allows us to obtain a denumerable family of number sequences. In [2] the introduction of such a parameter into the classical succession rule for the Motzkin numbers allowed the authors to define number sequences such that the n–th number of each of them is lying between the n–th Motzkin and Catalan numbers. Moreover, the permutations enumerated by each number sequence are identified: they are permutations with two forbidden subsequences; the first, of length three, is fixed and the second has a length which increases with j. In [3] the introduction of the parameter j in the classical succession rule for the Catalan numbers defines number sequences such that the n–th term interpolates between the n–th Catalan number and $n!$. The objects that each sequence counts are permutations with $j!$ forbidden subsequences of length $(j+2)$.

A permutation $\pi = \pi(1)\pi(2)\ldots.\pi(n)$ on $[n] = \{1, 2, \ldots, n\}$ is a bijection from $[n]$ to $[n]$. Let S_n be the set of permutations on $[n]$. A permutation $\pi \in S_n$ *contains a subsequence of type* $\tau \in S_k$ iff a sequence of indices $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ exists such that $\pi(i_1)\pi(i_2)\ldots\pi(i_k)$ is ordered as τ. We denote the set of permutations of S_n *avoiding* subsequences of type τ by $S_n(\tau)$.

Example 2.1 The permutation 58132674 belongs to $S_8(4321)$ because none of its subsequences of length 4 are of type 4321. This permutation does not belong to $S_8(4132)$ because there exist some subsequences of type 4132 like, for example, $\pi(2)\pi(3)\pi(6)\pi(8) = 8164$.

A *barred* forbidden subsequence $\bar{\tau}$ on $[k]$ is a permutation of S_k having a bar over one of its elements. Let τ be a permutation on $[k]$ identical to $\bar{\tau}$ but unbarred and $\hat{\tau}$ be the permutation on $[k-1]$ made up of the $(k-1)$ unbarred elements of $\bar{\tau}$, rewritten to be a permutation on $[k-1]$. A permutation $\pi \in S_n$ *contains a type* $\bar{\tau}$ *subsequence* if π contains a type $\hat{\tau}$ subsequence that, in turn, is not a subsequence of a type τ subsequence. We denote the set of permutations of S_n *avoiding* type $\bar{\tau}$ subsequences by $S_n(\bar{\tau})$.

Example 2.2 If $\bar{\tau} = 4\bar{1}32$ then $\tau = 4132$ and $\hat{\tau} = 321$. The permutation $\pi = 58132674$ belongs to $S_8(\bar{\tau})$ because all its subsequences of type $\hat{\tau}$: $\pi(1)\pi(4)\pi(5) = 532$, $\pi(2)\pi(4)\pi(5) = 832$, $\pi(2)\pi(6)\pi(8) = 864$ and $\pi(2)\pi(7)\pi(8) = 874$ are subsequences of a sequence of type τ because: $\pi(1)\pi(3)\pi(4)\pi(5) = 5132$, $\pi(2)\pi(3)\pi(4)\pi(5) = 8132$, $\pi(2)(4)\pi(6)\pi(8) = 8364$ and $\pi(2)\pi(5)\pi(7)\pi(8) = 8274$ are of type τ.

If we have the set $\tau_1 \in S_{k_1}, \ldots, \tau_p \in S_{k_p}$ of barred or unbarred permutations, we denote the set $S_n(\tau_1) \cap \ldots \cap S_n(\tau_p)$ by $S_n(\tau_1, \ldots, \tau_p)$. We call the family $F = \{\tau_1, \ldots, \tau_p\}$ a *family of forbidden subsequences*, the set $S_n(F)$, a *family of permutations with forbidden subsequences* and $S(F) = \sum_{n \geq 1} S_n(F)$ a *class of permutations with forbidden subsequences*. For $\pi \in S_n$, we call *sites* the positions lying on the left of $\pi(i)$,

Bell permutations and Stirling numbers interpolation

E. Pergola [†], G. Labelle [*], P. Leroux [*], R. Pinzani [†]

Abstract. We present a family of number sequences which interpolates between the sequences $B(n)$, of Bell numbers, and $n!$. It is defined in terms of permutations with forbidden patterns. The introduction, as a parameter, of the number k of left–to–right minima yields an interpolation between Stirling numbers of the second kind $S(n,k)$ and of the first kind (signless) $c(n,k)$. Moreover, q-counting the restricted permutations by inversions gives an interpolation between the usual q-analogues of these numbers.

Résumé. Nous présentons une famille de suites de nombres qui interpole entre la suite $B(n)$ des nombres de Bell et la suite $n!$. Cette famille est définie en termes de permutations à motifs interdits. L'introduction comme paramètre du nombre d'éléments saillants minimums de gauche a droite donne une interpolation plus fine entre les nombres de Stirling de deuxième espèce $S(n,k)$ et de première espèce (sans signe) $c(n,k)$. De plus, un q-comptage des permutations selon leurs inversions donne une interpolation entre les q-analogues habituels de ces nombres.

1 Introduction

The study of Stirling numbers and their q–analogues dates back a long time; in the last twenty years mathematicians were interested in models giving combinatorial interpretations of classical relations involving the q–analogues of Stirling numbers. In 1961, Gould [11] gives his expression in terms of symmetric functions; a combinatorial treatment of q–Stirling numbers of second kind, involving finite dimensional vector spaces over a field $\mathcal{K}_q$ of cardinality q appears in [16, 17, 18]; Garsia and Remmel [9] introduce particular rook placements in Ferrers boards. Later, Leroux [13] introduces 0–1 tableaux to prove the conjecture of Butler [6] concerning the q–log concavity for q–Stirling numbers and De Médicis and Leroux [14, 15] study and generalize q–Stirling numbers of both kinds, using this interpretation.

On the other hand the study of permutations with forbidden subsequences made meaningful progresses in the last thirty years: the n–th Catalan number is the common value for the number of permutations with a single forbidden subsequence of length three [21]; some results for permutations avoiding a single forbidden subsequence of length four can be found in [4, 5, 10]. Concerning permutations avoiding a single subsequence of length greater than four, Regev [19] obtained an interesting result, that is: the number of permutations of length n avoiding the pattern $1\ldots(k+1)$ is asymptotically equal to $c(k-1)^{2n}n^{(2k-k^2)/2}$, where c is a constant. Pell, Fibonacci, Motzkin and Schröder numbers are sequences which count permutations avoiding more than one forbidden subsequence. We refer to [12] for an exhaustive survey on the results about permutations with forbidden subsequences.

In this paper we put these two research areas together and give combinatorial interpretations of q–analogues of Stirling numbers of both kinds in terms of permutations with forbidden subsequences. From another point of view it can be seen as a continuation of the two previous works [2, 3], here, the interpolation is between Bell numbers and factorials, and, moreover, between $S(n,k)$ and $c(n,k)$ and their q–analogues.

[*]LaCIM, Département de mathématiques, Université du Québec à Montréal, C.P. 8888, Succ. Centre–Ville, Montréal (Québec), Canada, H3C 3P8. e–mail: labelle.gilbert@uqam.ca, leroux.pierre@uqam.ca

[†]Dipartimento di Sistemi e Informatica, Universitá di Firenze, Via Lombroso 6/17, 50134 Firenze, Italy, e-mail: elisa@dsi.unifi.it, pinzani@dsi.unifi.it

Generating effective symmetry-breaking predicates for search problems ⋆

Ilya Shlyakhter

MIT Lab for Computer Science, Software Design Group

Abstract

Consider the problem of testing for the existence of an n-node graph G satisfying some condition P, expressed as a Boolean constraint among the $n \times n$ Boolean entries of the adjacency matrix M. This problem reduces to satisfiability of $P(M)$. If P is preserved by isomorphism, $P(M)$ is satisfiable iff $P(M) \wedge SB(M)$ is satisfiable, where $SB(M)$ is a *symmetry-breaking predicate* — a predicate satisfied by at least one matrix M in each isomorphism class. $P(M) \wedge SB(M)$ is more constrained than $P(M)$, so it's solved faster by backtracking than $P(M)$ – especially if $SB(M)$ rules out most matrices in each isomorphism class. This method, proposed by Crawford et al [1], applies not just to graphs but to testing existence of a combinatorial object satisfying any property that respects isomorphism, as long as the property can be compactly specified as a Boolean constraint on the object's binary representation.

We present methods for generating symmetry-breaking predicates for several classes of combinatorial objects: acyclic digraphs, permutations, functions, and arbitrary-arity relations (direct products). We define a uniform optimality measure for symmetry-breaking predicates, and evaluate our constraints according to this measure. Results indicate that these constraints are either optimal or near-optimal for their respective classes of objects. We also evaluate some previously published predicates according to our measure, and confirm that these predicates eliminate most isomorphism.

1 Introduction

Consider a universe U of combinatorial objects representable by m-bit binary numbers. We will speak interchangeably of an object and its binary representation. Let U be divided into equivalence classes of isomorphic objects. A

⋆ Expanded version of a paper published in Electronic Notes on Discrete Mathematics, Volume 9, June 2001, available online at
http://www.elsevier.nl/locate/endm/volume9.free

permutation θ of the m bits is a *symmetry* of the universe iff applying θ to any object $X \in U$ yields an object isomorphic to X. The set of all symmetries is the *symmetry group* of the universe U, denoted by Sym.

For example, n-node digraphs can be represented by $n \times n$ adjacency matrices, and two matrices A, B are isomorphic iff there exists a permutation θ of the n nodes such that $\theta(A) = B$, where $(\theta(A))_{i,j} = A_{\theta(i),\theta(j)}$. Note that θ is a permutation of the n *nodes* of the digraph, but it also acts on the n^2-bit *adjacency matrices*, because each permutation of the nodes induces a corresponding permutation of the adjacency matrix bits [2]. The symmetry group Sym has order $n!$ and is isomorphic to σ_n, the symmetric group of order n.

Suppose you need to find an object X from a universe U, satisfying a property $P(X)$ (or determine that no such object exists). Suppose also that P is preserved under isomorphism, i.e. is constant on each isomorphism class. Enumerating all elements of U and testing P on each is clearly wasteful: it's enough to test P on one object per isomorphism class. For some classes of objects, procedures exist for isomorph-free exhaustive generation [3–5]. Faster generation procedures may be developed at the cost of generating more than one labeled object per isomorphism class and/or repeating objects.

If no object in U satisfies P, the generate-and-test approach must explicitly generate a complete representation of at least one representative per isomorphism class to verify unsatisfiability. On the other hand, backtracking methods [6] can rule out entire sets of objects without explicit generation, by determining that no object extending a *partial* binary representation satisfies P. If P can be encoded as a polynomial-size Boolean constraint on the bits of the fixed-length binary representation (as opposed to black-box computer code), backtracking methods for satisfiability can be used. Such methods can significantly outperform explicit generate-and-test approaches, as demonstrated by satisfiability encoding of planning problems [7].

Crawford et al [1] have proposed an approach to taking advantage of isomorphism structure in this framework. We define a *symmetry-breaking predicate* on U, $SB(X)$, which is *true* on at least one *representative* object per isomorphism class. We then test for satisfiability of $P'(X) = P(X) \wedge SB(X)$. Since P is constant on each isomorphism class, P' is satisfiable iff P satisfiable. Moreover, P' is solved much faster than P by backtracking, because it is more constrained: the algorithm will backtrack if none of the extensions of its current partial instantiation are isomorphism class representatives selected by SB. Experiments show that symmetry-breaking predicates can reduce search time by orders of magnitude with no changes to the search algorithm [1,2].

The difficulty of this approach lies in generating the symmetry-breaking predicate. In general, generating a *complete* symmetry-breaking predicate (*true*

of exactly one representative per isomorphism class) is NP-complete [1]; the practical choice is between *partial* symmetry-breaking predicates, *true* of at least one (typically more than one) representative per isomorphism class. To be effective, the predicate must rule out a large fraction of objects from each isomorphism class. On the other hand, the predicate must be compact; otherwise, checking the predicate's constraints at each search node will slow down the search, erasing the benefit of expanding fewer search nodes. Balancing these contradictory requirements is the subject of this paper.

The rest of the paper is organized as follows. Section 2 summarizes prior approaches and points out their deficiencies. Section 3 describes the generation of symmetry-breaking predicates for several classes of combinatorial objects. Section 4 gives a uniform optimality measure for symmetry-breaking predicates, and evaluates the predicates from Section 3 according to this measure. Section 5 describes directions for future work.

2 Prior work

In his original paper on symmetry-breaking predicates, Crawford proposes the following general framework for predicate generation. Fix an ordering of the bits in the object's binary representation. This induces a strict lexicographical ordering on all objects. Construct a symmetry-breaking predicate which is true on the *lexicographically smallest* object in each isomorphism class, as follows.

Let V be a fixed ordering of the bits of the binary representation. Then

$$\bigwedge_{\Theta \in Sym} V \leq \theta(V)$$

is a symmetry-breaking predicate, true of only the lexicographically smallest object in each symmetry class. This predicate explicitly requires that *any* symmetry map either fix the the representative object, or map it to a lexicographically higher object – i.e. that the representative object be lexicographically smaller than any isomorphic object.

Unfortunately, in many important cases Sym is very large. For example, for n-node digraphs $|Sym| = n!$, because any permutation of the graph's nodes (and the corresponding permutation of adjacency matrix entries) leads to an isomorphic graph. Crawford suggests mitigating the problem by replacing Sym with a polynomial-size *subset* $Sym' \in Sym$, thus requiring that the object be lexicographically smallest with respect to only some of the symmetries.

Crawford gives no formal guidance on choosing the subset of symmetries to break or the fixed variable numbering to use. This paper begins to fill the gap

by describing polynomial-size symmetry-breaking predicates for some common combinatorial objects. For some objects, we refine Crawford's algorithm by determining Sym' and V. For others, we present new predicate constructions, giving the first concrete alternatives to Crawford's lexicographic approach.

Crawford uses empirical measurements to gauge the effectiveness of his symmetry-breaking predicates. While such end-to-end tests are certainly useful, they give no hint of optimality of a given predicate, and reflect peculiarities of a particular backtracking algorithm (such as the dynamic variable-ordering heuristic [6]) besides the inherent complexity reduction brought by the predicate. We present an alternative approach which directly measures predicate pruning power, and gives an optimality measure relative to a complete symmetry-breaking predicate.

3 Generating symmetry-breaking predicates

In this section, we present methods for generating symmetry-breaking predicates on several classes of combinatorial objects: acyclic digraphs, permutations, direct products, and functions. These objects commonly occur in formal descriptions of system designs [8], the analysis of which motivates this work. Each subsection deals with one class of combinatorial objects, describing the binary representation, the isomorphism classes, and the construction of the symmetry-breaking predicate in terms of the binary representation.

3.1 Acyclic digraphs

Let U be the set of $n \times n$ adjacency matrices representing acyclic digraphs. Two matrices representing isomorphic digraphs are isomorphic. The symmetry group Sym has order $n!$.

Any acyclic digraph has an isomorphic counterpart that is topologically sorted with respect to a given node ordering. In terms of adjacency matrices, this means that every isomorphism class of adjacency matrices representing acyclic digraphs includes an upper-triangular matrix (since the lower triangle represents "backwards" edges from higher-numbered to lower-numbered nodes). Our symmetry-breaking predicate simply requires all entries below the diagonal to be $false$. This does not completely eliminate all isomorphic matrices, but as measurements in section 4.1 show, eliminates most.

Additionally, this symmetry-breaking predicate, together with the requirement that diagonal entries be $false$ (eliminating self-loops), implies the acyclicity

constraint, so no additional constraints on the matrix are needed. By contrast, expressing the acyclicity constraint on general digraphs requires a constraint of size $\Omega(MatMult(n) \log n)$, where $MatMult(n)$ is the complexity of matrix multiplication. Shorter constraints require less time to check at every search node, leading to faster search. In general, in cases where not all binary representations represent valid combinatorial objects from our universe U, constraints restricting the object to valid values are separate from the symmetry-breaking predicate. This example illustrates a new use of symmetry-breaking predicates: to reduce the size of original problem constraints.

Note that this symmetry-breaking predicate does not use Crawford's methodology. It's not even clear that a single fixed variable ordering exists which corresponds to this predicate. The next section on permutations gives another example of a symmetry-breaking predicate not based on lexicographic comparison.

3.2 Permutations

Let U be the set of $n \times n$ binary matrices representing permutations of n items. Matrix A represents the permutation mapping i to j iff $A_{i,j}$ is true. A matrix A represents a valid permutation (is a *permutation matrix*) iff every column and every row has exactly one *true* bit.

Two permutations are isomorphic if they have the same cycle structure, i.e. the same multiset of cycle lengths. Thus, an isomorphism class of permutation matrices corresponds to one permutation on a set of n indistinguishale objects. We define a canonical representative of each isomorphism class, and give a polynomial-size Boolean predicate on permutation matrices which is true only of the canonical representatives. We thus achieve full symmetry-breaking with a polynomial-size predicate.

The canonical form is most easily explained using cycle notation for permutations [9]. We require that each cycle consist of a continuous segment of items, that each item map to the immediately succeeding one or (for highest-numbered item in a cycle) to the smallest item in the cycle, and that longer cycles use higher-numbered items than shorter ones. For example, the permutation (12)(345) is in canonical form, but the isomorphic permutations (123)(45), (12)(354) and (15)(234) are not. Formally, given an $n \times n$ permutation matrix A, we have the following predicate in terms of the Boolean entries $A_{i,j}$:

$$(\forall i, j | (j > i + 1) \Rightarrow \neg A_{i,j}) \bigwedge$$

$$((\forall i, j | ((j > i) \wedge A_{j,i}) \Rightarrow ((\wedge_{k=i..(j-1)} A_{k,k+1}) \bigwedge (\wedge_{k=(j+1)..(2j-i)} \neg A_{k,j}))))$$

In this predicate, the condition $(j > i+1) \Rightarrow \neg A_{i,j}$ requires that an item mapped to a higher-numbered item map to the immediately succeeding item: e.g. 3 must map either to 4 (in which case 3 is not the highest-numbered item in its cycle), or to an item numbered not higher than 3 (in which case 3 *is* the highest-numbered item in its cycle). The condition $\wedge_{k=i..(j-1)} A_{k,k+1}$, implied by a backward edge $A_{j,i} (i < j)$, says that every backward edge implies the corresponding forward cycle: e.g. if 5 maps to 3 then 5 must be the highest-numbered item in the cycle and the cycle must be (345). The condition $\wedge_{k=(j+1)..(2j-i)} \neg A_{k,j}$, implied by the presence of a cycle $(i\ i+1 \ldots j-1\ j)$, requires the immediately succeeding cycle to be no shorter, in effect sorting cycles by increasing length: e.g. the cycle (345) excludes the cycles (6) and (67). Together with the original constraints restricting A to be a permutation matrix, these constraints permit exactly one permutation with a given multiset of cycle lengths, i.e. one permutation from each isomorphism class.

The size of this predicate $O(n^3)$, which matches the order of growth of the original constraints. It may be possible to reduce this order of growth by introducing auxiliary Boolean variables, but since n is typically small (under 15) in our analyses, cubic growth has been acceptable.

3.3 Relations

Consider the direct product $D = D_1 \times \ldots \times D_k$ of k disjoint finite nonempty sets (we call them *domains*). We define our universe U to be $\mathrm{P}(D)$, the power set of D. Each element of U, called a *relation*, can be represented by $\prod_{i=1}^{k} |D_i|$ bits. Each bit corresponds to an ordered k-tuple $(d_1, \ldots, d_k)$, $d_i \in D_i$, and is *true* in the binary representation of a relation iff the relation contains the corresponding ordered k-tuple. We will speak interchangeably of the bits and corresponding ordered k-tuples.

Isomorphism classes are defined by treating elements within each domain as indistinguishable. The symmetry group Sym of our universe U is isomorphic to direct product of k symmetric groups: $Sym \cong \sigma_{|D_1|} \times \ldots \times \sigma_{|D_k|}$. An element $\Theta = (\theta_1, \ldots, \theta_k)$ of Sym maps a relation r to a relation r', such that r' contains an ordered tuple $(d_1, \ldots, d_k)$ iff r contains the ordered tuple $(\theta_1^{-1}(d_1), \ldots, \theta_k^{-1}(d_k))$.

With $|Sym| = \prod_{i=1}^{k} |D_i|!$, direct application of Crawford's method is impractical. Nevertheless, it is possible to break all symmetries which permute a *single* domain with a linear-size predicate. Even though such symmetries represent

only a tiny fraction of all symmetries, experiments show that this predicate rules out most of the isomorphic objects.

We start with an example for the case $k = 2$, then generalize to arbitrary k.

Consider a binary relation $r \in A \times B$, $A = \{a_0, a_1, a_2\}$, $B = \{b_0, b_1, b_2\}$. Let us use the following orderly numbering V for bits of the binary representation of r:

	b_0	b_1	b_2
a_0	1	2	3
a_1	4	5	6
a_2	7	8	9

Under this numbering, Crawford's symmetry-breaking condition for the symmetry exchanging a_0 with a_1 and fixing all other elements (denoted $a_0 \leftrightarrow a_1$) is

$$\overline{123456789} \leq \overline{456123789}$$

which simplifies to $\overline{123} \leq \overline{456}$. Together with the condition for $a_1 \leftrightarrow a_2$, we have

$$\overline{123} \leq \overline{456} \leq \overline{789}$$

which breaks all symmetries permuting only A. Similarly, the conditions for $b_0 \leftrightarrow b_1$ and $b_1 \leftrightarrow b_2$ together simplify to

$$\overline{147} \leq \overline{258} \leq \overline{369}$$

breaking all symmetries which permute only B. Together, these conditions allow only those relations for which permuting either the rows *or* the columns (but not both simultaneously) leads to a lexicographically higher (or the same) relation, according to the given bit ordering. These conditions still allow values of r mapped to lexicographically lower values by symmetries which permute *both* A and B.

In general, consider a relation $r \in D_1 \times D_2 \times \ldots \times D_k$. We use Crawford's lexicographic method with the following numbering. Denoting the elements of D_i as $a_{i,0}, a_{i,1}, \ldots, a_{i,|D_i|-1}$, we number the bit corresponding to tuple

$(a_{1,e_1}, \ldots, a_{k,e_k})$, $0 \leq e_i < |D_i|$, as

$$\sum_{i=1}^{k}(e_i \times \prod_{j=i+1}^{k} |D_j|)$$

Now consider a transposition $\theta = a_{i,p} \leftrightarrow a_{i,p+1}$. The effect of this transposition on the binary representation of r is to fix all k-tuples except those with p or $p+1$ as their i'th coordinate, and among the tuples with p or $p+1$ as their i'th coordinate, to swap k-tuples differing only in their i'th coordinate. Within each pair of swapped tuples, the tuple with $p+1$ in i'th coordinate is numbered *higher* than the tuple with p in i'th coordinate. Therefore, Crawford's $V \leq \theta(V)$ condition reduces to $P \leq P'$, where P lists the bits corresponding to k-tuples with p in i'th coordinate, in increasing order by number in our numbering, and P' lists the bits corresponding to k-tuples with $p+1$ in i'th coordinate, in increasing order by number in the numbering. Then the right-hand side of Crawford's $V \leq \theta(V)$ condition for $a_{i,p} \leftrightarrow a_{i,p+1}$ equals the left-hand side of the condition for $a_{i,p+1} \leftrightarrow a_{i,p+2}$, so asserting the condition for adjacent pairs of elements breaks *all* permutations which permute only D_i.

The size of this predicate, expressed in conjunctive normal form (CNF), is linear in the size of each domain. The size of a single n-bit comparator is $O(n)$ [1]. For each domain D_i, we have $|D_i| - 1$ comparators of length $\prod_{j \in 1,\ldots,i-1,i+1,\ldots,k} |D_j|$, for a total comparator size of $O(k \times \prod_{i=1}^{k} |D_i|)$. Measurements of symmetry-breaking coverage provided by this predicate is given in section 4.2.

3.4 Functions

A function is a restricted kind of relation: a two-dimensional relation $r \in A \times B$ with each element of A (the domain) related to *exactly one* element of B (the range). Two functions are isomorphic iff they have the same multiset of preimage sizes. In analyses of relational specifications [8], functions occur more frequently than general relations. For functions, we give a polynomial-size symmetry-breaking predicate which breaks *all* symmetries.

First, we break all symmetries permuting only A by sorting the rows of r as binary numbers, as in the preceding section. For notational convenience, here we make the leftmost column (the bits corresponding to b_0) the least significant bit. Second, we sort the columns by the count of *true* bits. Formally, the constraints on r read

$$(\forall i \in \{0, \ldots, |A| - 2\}\Big|$$

$$(\overline{r_{i,|B|-1}r_{i,|B|-2}\ldots r_{i,1}r_{i,0}} \leq \overline{r_{i+1,|B|-1}r_{i+1,|B|-2}\ldots r_{i+1,1}r_{i+1,0}})) \bigwedge$$
$$(\forall j \in \{0,\ldots,|B|-2\} \Big| (|\{i|r_{i,j}\}| \leq |\{i|r_{i,j+1}\}|))$$

We show that together, these constraints define a *complete* symmetry-breaking predicate.

Since r represents a function, there are $|B|$ possible values for a row of r. Sorting the rows of r makes identical rows adjacent, so that the preimage of each $b_j \in B$ occupies a continous segment of A. In addition, for $i < j$, rows mapped to b_i represent smaller binary numbers than rows mapped to b_j. Therefore, elements of A mapped to $b_j \in B$ have lower indices in A than elements of A mapped to b_{j+1}. Alternatively, listing the elements of A in increasing order by index, we first list the elements that map to b_0 (if any), followed by the elements that map to b_1 (if any), and so on, with the elements that map to $b_{|B|-1}$ (if any) at the end of the list.

We now show that adding the second requirement, that the columns be sorted by cardinality (the count of *true* bits in the column), forces a canonical form. Since all matrices in an isomorphism class have the same multiset of preimage sizes (i.e. column cardinalities), sorting the columns by cardinality uniquely determines the cardinality of each column. In other words, all matrices in an isomorphism class satisfying the column-sorting condition have the same cardinalities in the corresponding columns. But given the constraints described in the preceding paragraph, this uniquely determines the image in B of each $a_i \in A$. If $c_j = |\ \{i|r_{i,j}\}|$, i.e. c_j is the cardinality of th j'th column, then the first c_0 elements of A must map to $b_0 \in B$, the next c_1 elements of A must map to $b_1 \in B$, and so on.

For example, here are three isomorphic function matrices satisfying the row-sorting condition:

	b_0	b_1	b_2	b_3	b_4
a_0	1	0	0	0	0
a_1	1	0	0	0	0
a_2	1	0	0	0	0
a_3	0	0	1	0	0
a_4	0	0	0	0	1
a_5	0	0	0	0	1

	b_0	b_1	b_2	b_3	b_4
a_0	0	1	0	0	0
a_1	0	1	0	0	0
a_2	0	0	1	0	0
a_3	0	0	1	0	0
a_4	0	0	1	0	0
a_5	0	0	0	1	0

	b_0	b_1	b_2	b_3	b_4
a_0	0	0	1	0	0
a_1	0	0	0	1	0
a_2	0	0	0	1	0
a_3	0	0	0	0	1
a_4	0	0	0	0	1
a_5	0	0	0	0	1

Only the rightmost one also orders the column cardinalities, and is the only matrix in the isomorphism class allowed by our symmetry-breaking predicate.

The row-sorting constraint can be expressed as a CNF formula of size $O(|A||B|)$, as described in section 3.3. The column cardinality sorting constraint can be expressed by building a standard binary adder for each column, which adds the entries of that column as one-bit binary numbers. Such an adder for one column has size $O(|A|log|A|)$. We then use the standard binary comparator among the column adders to assert the column-sorting condition. The entire predicate then has size $O(|A||B| + |A|^2 log|A|)$.

4 Measuring effectiveness of symmetry-breaking predicates

Symmetry-breaking predicates are designed to speed up search, so it would seem natural to judge their effectiveness by measuring the reduction in search time. This approach has several problems, however. Search times can be highly dependent on the particular backtracking algorithm, and on parameter settings such as the splitting heuristic [6]. The addition of the symmetry-breaking predicate changes the whole search tree (since splitting choices are determined by the entire constraint set), so the comparison to the original constraint problem is not completely clean. Machine-dependent effects such as cache locality can also bias the measurements. Most importantly, end-to-end measurements provide no clue to optimality: how much of the reduction afforded by symmetry are we actually utilizing?

As an alternative measure of efficiency, we can directly measure the pruning power of a symmetry-breaking predicate by counting the number of objects satisfying the predicate. For a complete symmetry-breaking predicate, this number is the number of isomorphism classes. For a partial symmetry-breaking predicate, this number will be higher; the question is, how much higher? Where the number of isomorphism classes is known, we can obtain a precise measure of optimality of our partial symmetry-breaking predicate by comparing its pruning effect with the maximum possible pruning effect.

Table 1 describes the numbers computed to measure coverage of partial symmetry-breaking predicates.

The numbers of isomorphism classes are taken from [10], [11], [12] and [13]. The number of objects allowed by the predicate is computed by generating the corresponding satisfiability instance, and counting its solutions with the RELSAT solution counter [14]. Correctness of the implementation was verified by doing complete symmetry-breaking for several classes of objects by Crawford's explicit lexicographical method method, and checking that the number of allowed instances matches the number of isomorphism classes.

Table 1
Values used to measure coverage of partial symmetry-breaking predicates.

value	formula	meaning
labeled	$\|U\|$	the number of distinct binary representations
unlabeled	from [10,11]	the number of isomorphism classes
allowed	$\|\{X \in U \| SB(X)\}\|$	# of objects allowed by symmetry-breaking predicate
coverage	$\frac{labeled-allowed}{labeled-unlabeled}$	percentage of excludable objects actually excluded
slack	$\frac{allowed}{unlabeled}$	maximum possible improvement factor

Table 2
Acyclic digraphs: symmetry-breaking coverage.

n	*labeled*	*unlabeled*	*allowed*	*coverage*	*slack*
3	25	6	8	89.47%	1.3
4	543	31	64	93.55%	2.1
5	29,281	302	1024	97.51%	3.4
6	3,781,50	5,984	32,768	99.29%	5.5
7	1,138,779,265	243,668	2,097,152	99.84%	8.6

4.1 Acyclic digraphs

Table 2 gives coverage information for the DAG-specific symmetry-breaking predicate described in section 3.1.

4.2 Relations

Table 3 shows the results for binary relations, using the symmetry-breaking predicate described in section 3.3. For each n, the table gives aggregate results over $k_1 \times k_2$ binary relations such that $k_1 \leq k_2$ and $k_1 + k_2 = n$. The "unlabeled" counts in this table were obtained in 5 seconds using Brendan McKay's bipartite graph generator "makebg" [13]. The "allowed" counts were obtained in 8 minutes using the solution-counting function of the RELSAT satisfiability solver [14]. Both computations were done on a Linux machine with two Pentium III processors and 512MB of memory.

Table 3
Relations: symmetry-breaking coverage.

n	*labeled*	*unlabeled*	*allowed*	*coverage*	*slack*
8	102,528	565	1,059	99.516%	1.87
9	1,327,360	1,518	3,834	99.825%	2.53
10	52,494,848	9,713	38,254	99.946%	3.94
11	1,359,217,664	39,379	229,347	99.986%	5.82
12	107,509,450,752	416,032	3,978,677	99.997%	9.56

Table 4
Digraphs without self-loops: symmetry-breaking coverage.

n	*labeled*	*unlabeled*	*allowed*	*coverage*	*slack*
3	64	16	21	89.58%	1.3
4	4,096	218	473	93.42%	2.2
5	1,048,576	9,608	35,979	97.46%	3.7
6	1,073,741,824	1,540,944	9,228,259	99.28%	6.0

4.3 Permutations and Functions

In these cases, symmetry-breaking is complete. The only possible improvement would be in reducing the size of the predicate. However, this improvement would only matter in cases where the original problem constraints have a smaller order of growth than the predicate.

4.4 Digraphs: symmetry-breaking coverage

It has been proposed [1,15] that breaking symmetries for the generators of the symmetry group eliminates most isomorphs, even though the set of generators is exponentially smaller than the set of all symmetries. Here we evaluate this assertion by measuring symmetry-breaking coverage achieved by breaking generator symmetries in the case of a single digraph without self-loops. The results, shown in Table 4, confirm that most isomorphs are eliminated. In the special case of DAGs, we have found that breaking generator symmetries breaks eliminates almost as many isomorphs as using the the DAG-specific symmetry-breaking predicate from section 3.1. However, the DAG-specific predicate still has the advantage of being more compact and expressing the acyclicity constraint in addition to breaking symmetries.

5 Conclusion and future work

We have presented a uniform method to gauge the effectiveness *and* optimality of symmetry-breaking predicates. The method measures the inherent simplification of the constraint problem, which, unlike running-time measurements, does not depend on the details of a particular backtracking algorithm. The method hinges on our ability to lower-bound the number of isomorphism classes in the universe; these numbers are available for a wide variety of combinatorial objects.

The method also depends on the ability to count solutions to a CNF formula. The current implementation of solution counting in RELSAT suffices to obtain useful results. Combining RELSAT's counting algorithm with recent SAT solving techniques such as those introduced in [16] should extend the range of problems for which counting is feasible. Since approximate counting suffices for our application, it would interesting to see if approximate counting algorithms can be developed.

We have also presented specific polynomial-size symmetry-breaking predicates for the types of states commonly occurring in analysis of relational specifications. Measurements show that these predicate exclude over 99% of excludable assignments, and come within an order of magnitude of the optimum. These are the first formalized examples of predicates not derived from Crawford's conditions.

Experiments show that predicate coverage, defined as the fraction of excludable objects actually excluded, grows monotonically with the scope of the objects. In other words, as the search space grows, our use of the available symmetry becomes more complete. On the other hand, the slack factor representing the possible improvement also increases. With search space sizes growing exponentially, improving coverage by even a fraction of a percent can lead to significant reduction in absolute search time.

Most interestingly, breaking a random set of symmetries to small depth often leads to surprisingly effective predicates. Formalizing this observation into a formal randomized symmetry-breaking scheme will be a major goal of future work. Various ways to bias the random selection of symmetries will be investigated. For instance, Crawford's condition for a single symmetry Θ excludes $2^{n-|\Theta|}$ assignments, where $|\Theta|$ is the number of cycles in Θ. This suggests biasing selection towards symmetries with fewer cycles. On the other hand, overlap between sets of states excluded by the selected symmetries should be minimized. This work could relate to work on probabilistic isomorphism testing.

In this paper, we only cover objects consisting of a single DAG, relation,

function or permutation. In practice, the universe of objects may be the set of abstract states of a system, with each state described by a *collection* of combinatorial object components. For example, in a lock-based multitasking environment, the state can be represented by a *pair* of relations: which process waits on each mutex, and which process holds which mutex. Applying a symmetry-breaking predicate to one component destroys the symmetry of the domains related by that component: the elements of these domains stop being interchangeable. This raises the question: to which of the state components should we apply our symmetry-breaking predicates? A lookup table of known predicate coverages for the common component types, computed as described in this paper, could be used to make the decision that optimizes the pruning effect.

Finally, it is necessary to quantify the correlation between pruning power of the predicate and the search time under various backtracking algorithms. Since search time is directly affected by the size of the predicate, as well as by its pruning power, such measurements are necessary to determine the proper tradeoff values between predicate size and strength. Besides search time, one useful measure might be "symmetry-breaking density", that is, the number of assignments excluded per literal of the symmetry-breaking predicate. It would be useful to know whether this measure correlates with search time.

6 Related work

Since the publication of the conference version of this paper, a number of related results have appeared. Flener et al [17] have generalized the results of section 3.3 to matrices of arbitrary values (we only considered Boolean matrices in this paper). They also showed that the results hold for the case where only a subset of the rows/columns of the matrix is interchangable. Aloul et al [15] have proposed improved construction of symmetry-breaking predicates, which uses fewer CNF clauses and eliminates more isomorphs than Crawford's [1] construction. Luks and Roy [18] have shown how to construct small symmetry-breaking predicates when the symmetry group is commutative.

7 Acknowledgement

I'd like to thank Daniel Jackson and Manu Sridharan for helpful discussions, and Brendan McKay for providing an advance copy of his bipartite graph generator.

References

[1] J. Crawford, M. Ginsberg, E. Luks, and A. Roy. Symmetry-breaking predicates for search problems. In *Fifth International Conference on Principles of Knowledge Representation and Reasoning*, 1996.

[2] David Joslin and Amitabha Roy. Exploiting symmetry in lifted csps. In *AAAI97*, 1997.

[3] B. D. McKay. Isomorph-free exhaustive generation. *Journal of Algorithms*, 26:306 – 324, 1998.

[4] Daniel Jackson, Somesh Jha, and Craig A. Damon. Isomorph-free model enumeration: A new method for checking relational specifications. *ACM Transactions on Programming Languages and Systems*, 20(2):302–343, March 1998.

[5] C. Norris Ip and David L. Dill. Better verification through symmetry. *Formal Methods in System Design*, 9(1):41–75, August 1996.

[6] Rina Dechter and Daniel Frost. Backtracking algorithms for constraint satisfaction problems. Technical Report 56, UC-Irvine, 1999.

[7] Henry Kautz and Bart Selman. Planning as satisfiability. In *Proceedings of the 10th European Conference on Artificial Intelligence*, 1992. http://portal.research.bell-labs.com/orgs/ssr/people/kautz/papers-ftp/satplan.ps.

[8] Daniel Jackson, Ilya Shlyakhter, and Manu Sridharan. A micromodularity mechanism. In *Proc. ACM SIGSOFT Conf. Foundations of Software Engineering/European Software Engineering Conference (FSE/ESEC '01), Vienna*, 2001.

[9] Herbert Wilf. East side, west side: an introduction to combinatorial families with maple programming. http://www.cis.upenn.edu/ wilf/eastwest.pdf, 1999.

[10] R.C.Read. *An Atlas of Graphs.* Oxford University Press, 1998.

[11] Neil J. A. Sloane. Sloane's on-line encyclopedia of integer sequences. http://www.research.att.com/ njas/sequences/.

[12] F. Harary and E.M.Palmer. *Graphical Enumeration.* Academic Press, 1973.

[13] Brendan McKay. Personal communication. http://cs.anu.edu.au/people/bdm/nauty/, 2002.

[14] R. Bayardo and J. Pehoushek. Counting models using connected components. In *AAAI Proceedings*, 2000.

[15] Fadi A. Aloul, Arathi Ramani, Igor L. Markov, and Karem A. Sakallah. Solving difficult sat instances in the presence of symmetry. In *Proceedings of 39th ACM/IEEE Design Automation Conference, New Orleans, Louisiana*, 2002.

[16] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, 2001.

[17] Pierre Flener, Alan Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel, Justin Pearson, and Toby Walsh. Symmetry in matrix models. In *SymCon'01 – Symmetry in Constraints, CP'01 Post-Conference Workshop, Paphos, Cyprus*, 2001.

[18] Eugene Luks and Amitabha Roy. Symmetry breaking in constraint satisfaction. In *Proceedings of 7th International Conference of Artificial Intelligence and Mathematics, Ft. Lauderdale, Florida*, 2002. http://www.cs.bc.edu/ aroy/.

[19] Daniel Jackson. An intermediate design language and its analysis. In *Proceedings of International Conference on Foundations of Software Engineering, Orlando, FL*, 1998.

Evaluating the binary partition function when $N = 2^n$ *

John L. Pfaltz
University of Virginia

April 24, 1995

Abstract

We present a linear algorithm to count the number of binary partitions of 2^n. It is also shown how such binary partitions are related to closure spaces on n elements, thereby giving a lower bound on their enumeration as well.

1 Background

A *binary partition* of the integer N is a sequence of non-negative integers $< a_n, \cdots, a_0 >$, such that

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 = N. \tag{1}$$

The number of such sequences, denoted *b(N)*, is called the *binary partition function*. Both the function and its evaluation have been well investigated. It is described in Sloane's Handbook, [13]. A short history of the binary partition function can be found in [1], in which Churchhouse describes his calculation of *b(N)* on an early Atlas computer. Our method of evaluation improves on his only because we restrict ourselves to the special case in which $N = 2^n$. Consequently, we must first address the issue: "why consider such a special case?".

The concept of uniquely generated closure spaces has begun to be studied as a common thread emerging in computer applications, in graphs, and in discrete geometries. Briefly, a closure operator φ is said to be *uniquely generated* if in addition to the customary closure axioms[1]

$$X \quad \subseteq \quad X.\varphi$$

*Research supported in part by DOE grant DE-FG05-95ER25254.

[1] We will denote closure operators using a suffix notation.

$$\begin{aligned} X &\subseteq Y \text{ implies } X.\varphi \subseteq Y.\varphi \\ X.\varphi.\varphi &= X.\varphi^2 = X.\varphi \end{aligned} \tag{2}$$

we add a fourth which distinguishes this closure concept from more familiar topological closure,

$$X.\varphi = Y.\varphi \text{ implies } (X \cap Y).\varphi = X.\varphi = Y.\varphi \tag{3}$$

Closure operators satisfying (3) above are uniquely generated in the sense that for any set Z, there exists a unique minimal set $X \subseteq Z$, called its *generator*[2] and denoted $Z.gen$, such that $X.\varphi = Z.\varphi$. Such a closure operator acting on a set, or universe, of elements, $\mathbf{U}$, is said to be a *closure space* $(\mathbf{U}, \varphi)$, as in [7]. Readily, a subset X will be *closed* if $X.\varphi = X$.[3] The importance of uniquely generated closure spaces lies in the fact that in discrete systems they play a role that is in many respects analogous to the vector spaces of classical mathematics. We establish this parallel in the next paragraph.

A closure operator σ, satisfying the three closure axioms of (2), together with the Steinitz-MacLane *exchange* property

$$\text{if } y \notin X.\sigma \text{ then } y \in (X \cup \{x\}).\sigma \text{ implies } x \in (X \cup \{y\}).\sigma \tag{4}$$

can be shown to be the closure operator of a matroid, $\mathcal{M}$ [14]. Similarly, a closure φ satisfying the three closure axioms and the *anti-exchange* property

$$\text{if } x, y \notin X.\varphi \text{ then } y \in (X \cup \{x\}).\varphi \text{ implies } x \notin (X \cup \{y\}).\varphi \tag{5}$$

is the closure operator of an anti-matroid, $\mathcal{A}$ [3]. It can be shown [8] [12] that a closure operator is uniquely generated if and only if it satisfies the anti-exchange property (5). A matroid, $\mathcal{M}$, is a set system that generalizes the independent sets of a linear algebra. The closure of these sets, commonly called its *spanning* operator, is a *vector space*. Uniquely generated closure spaces, therefore, are the analogs of vector spaces, but with respect to anti-matroids. From now on, we will simply call them *closure spaces*.

Closure operators are fairly common, although they frequently have other names, for example "convexity". The convex hull of a discrete set is an uniquely generated closure. A theory of convex geometries is developed in [5]. Convexity in graphs has been examined in [11] [6]. The "lower ideals", or "down sets" of a partially ordered set are closed. In concurrent computing, the concept of a "transaction" is a simple closure operator. Algorithmic closure, in particular that of greedy algorithms is found in [9], which introduces the term "greedoid", a secial kind of anti-matroid.

[2] Readily, if X_1 and X_2 were distinct minimal generators of $Z.\varphi$, then because $X_1.\varphi = X_2.\varphi = Z.\varphi$, we must have, by (3), $(X_1 \cap X_2).\varphi = Z.\varphi$ contradicting minimality.

[3] The family $\mathcal{C}$ of closed sets is closed under intersection, and this characterization is equivalent to (2), *c.f.* [4].

The subsets of a closure space can be partially ordered to create a lattice [12], with many interesting properties. Of most importance is the observation that for any set $Z \subseteq \mathbf{U}$ the cardinality of $\{ X \mid X.\varphi = Z.\varphi\}$ must be a power of 2. Thus any uniquely generated closure operator φ partitions the subsets of $\mathbf{U}$ into a disjoint collection of subsets, each containing a single closed set and each consisting of 2^k subsets. Let a_k denote the number of collections with 2^k subsets. The sequence $< a^n, a^{n-1}, a^{n-2}, \cdots, a^2, a^1, a^0 >= 2^n$ is thus a compact description of a closure space $(\mathbf{U}, \varphi)$, where $|\mathbf{U}| = n$. Moreover, it is shown in [12] that for every such binary partition of 2^n there exists at least one closure space with that property. Consequently, the enumeration of binary partitions of 2^n becomes a lower bound on the enumeration of closure spaces over n elements.

2 Counting Partitions

Let $\mathbf{P}^n$ denote the set $\{\pi_i =< a_n, \cdots, a_k, \cdots a_0 >\}$ of all binary partitions of 2^n. Several characteristics of $\mathbf{P}^n$ are readily apparent. First, $a_n \neq 0$ if and only if $a_k = 0$ for all $0 \leq k < n$. Second, since the right hand side is even and all terms $a_k \cdot 2^k$, $k > 0$ must be even, the coefficient a_0 must be even. Third, if $< \cdots, a_k, a_{k-1}, \cdots >$ is a partition of $\mathbf{P}^n$, then $< \cdots, a_k - 1, a_{k-1} + 2, \cdots >$ must be as well. And fourth, if $< a_n, \cdots, a_k, \cdots, a_o >$ is a partition in $\mathbf{P}^n$ then $< a_n, \cdots, a_k, \cdots, a_0, 0 >$ is a partition in $\mathbf{P}^{n+1}$.

With these observations, it is not difficult to write a process which generates all partitions in lexicographic order. Doing so, and displaying each partition, generates the following enumerations of $\mathbf{P}^3$ and $\mathbf{P}^4$. It is quite easy to verify by inspection that each sequence is a

	n = 3		
1	0	0	0
0	2	0	0
0	1	2	0
0	1	1	2
0	1	0	4
0	0	4	0
0	0	3	2
0	0	2	4
0	0	1	6
0	0	0	8

			n = 4	
1	0	0	0	0
0	2	0	0	0
0	1	2	0	0
0	1	1	2	0
0	1	1	1	2
0	1	1	0	4
0	1	0	4	0
0	1	0	3	2
0	1	0	2	4
0	1	0	1	6
0	1	0	0	8
0	0	4	0	0
0	0	3	2	0
0	0	3	1	2
0	0	3	0	4
0	0	2	4	0
0	0	2	3	2
0	0	2	2	4

0	0	2	1	6
0	0	2	0	8
0	0	1	6	0
0	0	1	5	2
0	0	1	4	4
0	0	1	3	6
0	0	1	2	8
0	0	1	1	10
0	0	1	0	12
0	0	0	8	0
0	0	0	7	2
0	0	0	6	4
0	0	0	5	6
0	0	0	4	8
0	0	0	3	10
0	0	0	2	12
0	0	0	1	14
0	0	0	0	16

Figure 1: $\mathbf{P}^3$ and $\mathbf{P}^4$

partition of 2^n. And because they are in lexicographic order, one can verify that all possible

partitions have been generated.

Because $< a_{n-1}, \cdots, a_0 > \in \mathbf{P}^{n-1}$ implies $< a_{n-1}, \cdots, a_0, 0 > \in \mathbf{P}^n$, it follows that

$$b(2^n) = b(2^{n-1}) + p_n \tag{6}$$

where p_n denotes the number of partitions $\pi_i \in \mathbf{P}^n$ in which $a_0 \neq 0$. We say such partitions are *normal* because they correspond to closure spaces in which the empty set is closed.

In the lexicographic order of $\mathbf{P}^n$, if $\pi_i^n =< a_n, \cdots, a_2, a_1, 0 > \in \mathbf{P}^n, a_1 \neq 0$, then there must follow the sequence $S_{a_1}^n$ of partitions, $< a_n, \cdots, a_2, a_1 - 1, 2 >$, $< a_n, \cdots, a_2, a_1 - 2, 4 >$, $\cdots$, $< a_n, \cdots, a_2, 0, 2a_1 >$. There are two such sequences in $\mathbf{P}^3$; $< 0, 1, 2, 0 >$ followed by $< 0, 1, 1, 2 >$ and $< 0, 1, 0, 4 >$, and $< 0, 0, 4, 0 >$ followed by $< 0, 0, 3, 2 >$, $< 0, 0, 2, 4 >$, $< 0, 0, 1, 6 >$, and $< 0, 0, 0, 8 >$. In $\mathbf{P}^4$ there are 6 such subsequences because there are 6 normal partitions in $\mathbf{P}^3$; the last consists of 8 normal partitions following $< 0, 0, 0, 8, 0 >$. Once this pattern is perceived the counting process becomes evident. In Figure 2 we reinforce this pattern by showing just the first 8 and the last 34 (of 202) partitions in $\mathbf{P}^5$.

n = 5

1	0	0	0	0	0		0	1	1	1	2	0
0	2	0	0	0	0		0	1	1	1	1	2
0	1	2	0	0	0		0	1	1	1	0	4
0	1	1	2	0	0		0	1	1	0	4	0
		.								.		
		.								.		
		.								.		
0	0	0	2	1	22		0	0	0	0	16	0
0	0	0	2	0	24		0	0	0	0	15	2
0	0	0	1	14	0		0	0	0	0	14	4
0	0	0	1	13	2		0	0	0	0	13	6
0	0	0	1	12	4		0	0	0	0	12	8
0	0	0	1	11	6		0	0	0	0	11	10
0	0	0	1	10	8		0	0	0	0	10	12
0	0	0	1	9	10		0	0	0	0	9	14
0	0	0	1	8	12		0	0	0	0	8	16
0	0	0	1	7	14		0	0	0	0	7	18
0	0	0	1	6	16		0	0	0	0	6	20
0	0	0	1	5	18		0	0	0	0	5	22
0	0	0	1	4	20		0	0	0	0	4	24
0	0	0	1	3	22		0	0	0	0	3	26
0	0	0	1	2	24		0	0	0	0	2	28
0	0	0	1	1	26		0	0	0	0	1	30
0	0	0	1	0	28		0	0	0	0	0	32

Figure 2: First 8 and last 34 partitions of $\mathbf{P}^5$

Notice that these subsequences of normal partitions (with $a_0 \neq 0$) were generated by the three normal partitions $< 0, 1, 1, 1, 2 >$, $< 0, 0, 0, 1, 14 >$, and $< 0, 0, 0, 0, 16 >$ of $\mathbf{P}^4$.

The length of a sequence $S_{a_1}^n$ is a_1. Hence, each normal partition $\pi_i^{n-1} \in \mathbf{P}^{n-1}$ gives rise to a subsequence of $a_1^n = a_0^{n-1}$ normal partitions in $\mathbf{P}^n$. If one carefully keeps track of all normal permutations in $\mathbf{P}^{n-1}$, then one can use the mechanism above to generate all

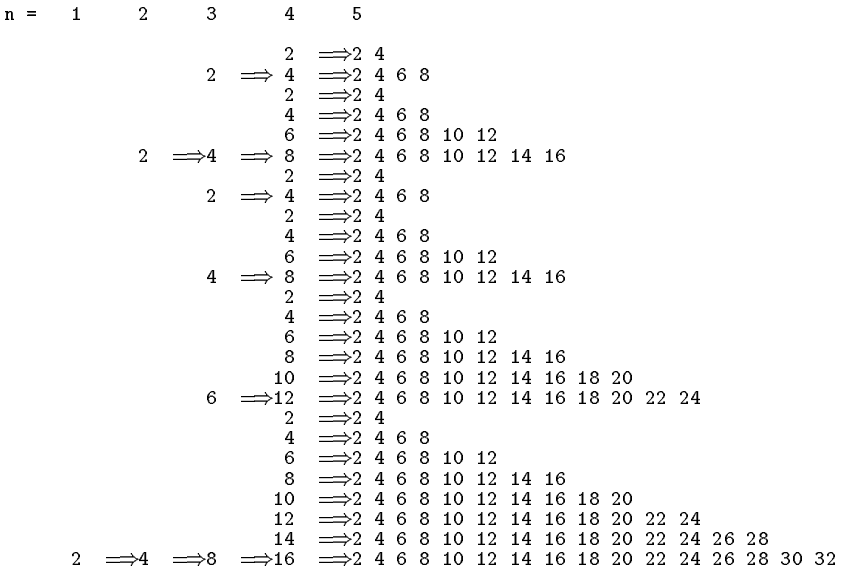


Figure 3: a_0 coefficient in sequences S_k^n of normal partitions

normal partitions in $\mathbf{P}^n$. This is illustrated in Figure 3 in which subsequences S_k^n of normal partitions are enumerated (by showing only the a_0 value) in vertical columns for $n = 1$ through 4, and horizontally (to conserve space) for $n = 5$. For $n = 1$ through 4, each entry a_0^n in S_i^n denotes to its right (with $\Rightarrow$) the *last* entry $< 2a_1, 0, \cdots, a_{n+1} >$ in the sequence $S_{a_0}^{n+1}$ that it generates.

Observe in this figure, that when $n = 3$, all 6 partitions with $a_0 \neq 0$ are enumerated in just two subsequences S_2^3 and S_4^3, which were generated by the two normal partitions in $\mathbf{P}^2$. With $n = 4$ the 26 normal partitions of $\mathbf{P}^4$ are enumerated in two occurrences of the subsequences S_2^4 and S_4^4, together with single occurrences of S_6^4 and S_8^4, which themselves were generated from the 6 normal partitions of $\mathbf{P}^3$. Fortunately, since all sequences S_k^n have the form 2, 4, $\cdots, k$, we need only keep track of the number of such sequences in $\mathbf{P}^n$, not their actual composition.

Let σ_k^n, k even, denote the *number* of subsequences S_k^n of normal partitions in $\mathbf{P}^n$. Based on Figure 3 we can construct Table 1.

Since every normal partition of $\mathbf{P}^n$ belongs to such a subsequence, we have

$$p_n = \sum_{even\ k}^{2^{n-1}} k \cdot \sigma_k^n \tag{7}$$

n	2	3	4	5	6
p_n	2	6	26	166	1,626
k			σ_k^n		
2	1	1	2	6	26
4		1	2	6	26
6			1	4	20
8			1	4	20
10				2	14
12				2	14
14				1	10
16				1	10
18					6
20					6
22					4
24					4
26					2
28					2
30					1
32					1

Table 1: Counts σ_k^n of subsequences S_k^n of normal partitions in $\mathbf{P}^n$

Using Table 1 and equation (7) one obtains $p_7 = 25,510$, and by (6) $b(2^6) = 1,828$, so $b(2^7) = b(2^6) + p_7 = 27,338$. It only remains to determine $\sigma_k^{n+1}, 2 \leq k \leq 2^n$ given $\sigma_j^n, 2 \leq j \leq 2^{n-1}$.

Since each sequence S_k^{n-1} of normal partitions in $\mathbf{P}^{n-1}$ generates the subsequences $S_2^n, S_4^n, \cdots, S_{2k}^n$ in $\mathbf{P}^n$, one can simply loop over all such subsequences σ_k^{n-1} and increment $\sigma_2^n, \cdots \sigma_{2k}^n$ as in the following code section

```
max_k = 2**(n-1);
for (k=2; k<=max_k; k+=2)
        {
        for (j=2; j<=2*k; j+=2)
                sigma[n][j] += sigma[n-1][k];
        }
```

The $O(k^2)$ behavior of this double loop can become expensive when $k = 2^{n-1}$ becomes large. We observe in Table 1, that the first two values of σ_k^n are determined by

$$\sigma_2^n = \sigma_4^n = p_{n-2} \tag{8}$$

and that subsequent values of σ_k^n can be calculated as

$$\sigma_k^n = \sigma_{k+2}^n = \sigma_{k-2}^n - \sigma_{\lfloor (k+2)/2 \rfloor - 2}^{n-1} \tag{9}$$

for $k = 6, 10, 14, \cdots$.

Putting together (6), (7), (8), and (9) one obtains

Theorem 2.1 *The number, p_n, of distinct partitions of 2^n is given by:*

$$p_n = p_{n-1} + \sum_{even k}^{2^{n-1}} k \cdot \sigma_k^n$$

$$\textit{where } \sigma_k^n = \begin{cases} \sum_{even\ i} k \cdot \sigma_i^{n-2} & : \quad k = 2, 4 \\ \sigma_{k-2}^n - \sigma_{\lfloor (k+2)/2 \rfloor - 2}^{n-1} & : \quad k = (6,8), (10,12) \cdots \end{cases}$$

The primary advantage of expressing p_n in this manner is that it permits the following counting procedure, which although somewhat more complex, has linear behavior.

```
long    sigma[MAX_N+1][POWER_MAX_N];

long    calculate_p (int n)
        /*
        ** Assumes sigma[n-1, 2**(n-2)] has been previously determined
        ** and globally stored.
        ** This procedure sets up sigma[n, 2**(n-1)], and returns
        ** the number p[n] of normal partitions with a[0] != 0
        */
        {
        int     k, k_calc, max_k;
        long    sum;

        max_k = 2**(n-1);
        switch (n)
          {
          case 1:
                return 1;
          case 2:
                sigma[2][2] = 1;
                break;
          case 3:
                sigma[3][2] = 1;
                sigma[3][4] = 1;
                break;
          default:
                sigma[n][2] = y[n-2];
                sigma[n][4] = y[n-2];

                for (k=6; k<=max_k; k+=4)
                        {
                        k_calc = (k+2)/2 - 2;
                        sigma[n][k]   = sigma[n][k-2] - sigma[n-1][k_calc];
                        sigma[n][k+2] = sigma[n][k-2] - sigma[n-1][k_calc];
```

```
                }
        break;
  }
sum = 0;
for (k=2; k<=max_k; k += 2)
        {
        sum = sum + sigma[n][k]*k;
        }
p[n] = sum;
return sum;
}
```

With this code one can generate the following Table 2 of partitions of 2^n. The values of

n	$b(2^n)$	p_n
3	10	6
4	36	26
5	202	166
6	1,828	1,626
7	27,338	25,510
8	692,004	664,666
9	30,251,722	29,559,718
10	2,320,518,948	2,290,267,226

Table 2: Total $b(2^n)$ and normal p_n partitions of 2^n

$b(2^7)$ and $b(2^8)$ can be verified by enumerating all partitions, using the program of section 1, or by reference to [13].

Readily, $b(2^n)$ must be even because, as observed, a_0 must be even, so every subsequence of normal partitions is even. It is not hard to show that $|\mathbf{P}^n|$ grows super exponentially with respect to n. Based on the expression $log\, b(n) \sim (log\, n)^2/2$ found in [10], Churchhouse [2] gives the asymptotic upper bound $b(n) \sim O(n^{1/2 \cdot log_2 n})$ or

$$b(2^n) = |\mathbf{P}^n| \sim O((2^n)^{n/2}). \tag{10}$$

The nature of this super exponential growth is difficult to intuitively comprehend because, unfortunately, equation (10) is a poor approximation for small values of n. In Table 3, we compare $b(2^n)$ with two lower bounding functions, n^n and $(2^n)^{n/3}$, and the upper bound $(2^n)^{n/2}$ to which it is eventually asymptotic. Besides giving some concrete feeling for the growth of the binary partition function, this table illustrates that a wealth of closure spaces exist for even small n.

n	n^n	$(2^n)^{n/3}$	$b(2^n)$	$(2^n)^{n/2}$
2	$4.000\ 10^0$	$2.519\ 10^0$	$4.000\ 10^0$	$4.000\ 10^0$
3	$2.700\ 10^1$	$8.000\ 10^0$	$1.000\ 10^1$	$2.262\ 10^1$
4	$2.560\ 10^2$	$4.031\ 10^1$	$3.600\ 10^1$	$2.560\ 10^2$
5	$3.125\ 10^3$	$3.225\ 10^2$	$2.020\ 10^2$	$5.792\ 10^3$
6	$4.665\ 10^4$	$4.096\ 10^3$	$1.828\ 10^3$	$2.621\ 10^5$
7	$8.235\ 10^5$	$8.257\ 10^4$	$2.733\ 10^4$	$2.372\ 10^7$
8	$1.677\ 10^7$	$2.642\ 10^6$	$6.920\ 10^5$	$4.294\ 10^9$
9	$3.874\ 10^8$	$1.342\ 10^8$	$3.025\ 10^7$	$1.554\ 10^{12}$
10	$1.001\ 10^{10}$	$1.082\ 10^{10}$	$2.320\ 10^9$	$1.125\ 10^{15}$
11	$2.853\ 10^{11}$	$1.385\ 10^{12}$	$3.163\ 10^{11}$	$1.630\ 10^{18}$
12	$8.916\ 10^{12}$	$2.814\ 10^{14}$	$7.747\ 10^{13}$	$4.722\ 10^{21}$
13	$3.088\ 10^{14}$	$9.078\ 10^{16}$	$3.439\ 10^{16}$	$2.735\ 10^{25}$
14	$1.111\ 10^{16}$	$4.648\ 10^{19}$	$2.789\ 10^{19}$	$3.169\ 10^{29}$
15	$4.379\ 10^{17}$	$3.777\ 10^{22}$	$4.160\ 10^{22}$	$7.343\ 10^{33}$
16	$1.844\ 10^{19}$	$4.874\ 10^{25}$	$4.874\ 10^{26}$	$3.402\ 10^{38}$
17	$8.272\ 10^{20}$	$9.982\ 10^{28}$	$5.888\ 10^{29}$	$3.153\ 10^{43}$

Table 3: $b(2^n)$ compared with upper and lower bounding functions

References

[1] R.F. Churchhouse. Congruence properties of the binary partition function. *Proc. Cambridge Phil. Soc.*, 66(2):371–376, 1969.

[2] R.F. Churchhouse. Binary partitions. In A.O.L. Atkin and B.J. Birch, editors, *Computers in Number Theory*, pages 397–400. Academic Press, 1971.

[3] Brenda L. Dietrich. Matroids and antimatroids — a survey. *Discrete Mathematics*, 78:223–237, 1989.

[4] Paul H. Edelman. Meet-distributive lattices and the anti-exchange closure. *Algebra Universalis*, 10(3):290–299, 1980.

[5] Paul H. Edelman and Robert E. Jamison. The theory of convex geometries. *Geometriae Dedicata*, 19(3):247–270, Dec. 1985.

[6] Martin Farber and Robert E. Jamison. Convexity in graphs and hypergraphs. *SIAM J. Algebra and Discrete Methods*, 7(3):433–444, July 1986.

[7] George Gratzer. *General Lattice Theory.* Academic Press, 1978.

[8] A. J. Hoffman. Binding constraints and Helly numbers. In *2nd Intern'l Conf. on Combinatorial Math.*, volume 319, pages 284–288. Annals of the N.Y. Acad. of Sciences, 1979.

[9] Bernhard Korte, Laszlo Lovasz, and Rainer Schrader. *Greedoids.* Springer-Verlag, Berlin, 1991.

[10] K. Mahler. On a special functional equation. *J. London Math. Soc.*, 15(58):115–123, Apr. 1940.

[11] John L. Pfaltz. Convexity in directed graphs. *J. of Comb. Theory*, 10(2):143–162, Apr. 1971.

[12] John L. Pfaltz. Closure lattices. *Discrete Mathematics*, 1995. (to appear), preprint available as Tech. Rpt. CS-94-02 through home page http://uvacs.cs.virginia.edu/.

[13] N. J. A. Sloane. *A Handbook of Integer Sequences.* Academic Press, 1973. On-line version at 'sequences@research.att.com'.

[14] D.J.A. Welsh. *Matroid Theory.* Academic Press, 1976.

Apollonian Circle Packings: Geometry and Group Theory III. Higher Dimensions

Ronald L. Graham [1]
Jeffrey C. Lagarias
Colin L. Mallows
Allan R. Wilks
AT&T Labs, Florham Park, NJ 07932-0971

Catherine H. Yan
Texas A&M University, College Station, TX 77843

(January 18, 2001 version)

ABSTRACT

Apollonian circle packings arise by repeatedly filling the interstices between four mutually tangent circles with further tangent circles. Such packings can be specified in terms of the Descartes configurations they contain, where an (ordered) Descartes configuration is an ordered set of four mutually tangent circles having disjoint interiors, on the Riemann sphere. Parts I and II considered groups of transformations preserving Apollonian packings or the Descartes configurations in such packings. The paper considers generalizations of these results to dimensions $n \geq 3$, for Apollonian ensembles of n-dimensional Descartes configurations. These are no longer packings for $n \geq 4$, but there are analogues of most of the properties in parts I and II for such ensembles of n-dimensional Descartes configurations. An Apollonian sphere ensemble is strongly rational if every sphere in it has a rational curvature and a rational center. We show that strongly rational Apollonian sphere ensembles exist in dimension n if and only if $n = 2k^2$ or $n = (2k+1)^2$ for some positive integer k.

Keywords: Circle packings, Apollonian circles, Diophantine equations, Lorentz group, Coxeter group

[1]Current address: Dept. of Computer Science, University of California at San Diego, La Jolla, CA 92110.

Apollonian Circle Packings: Geometry and Group Theory III. Higher Dimensions

1. Introduction

In parts I and II we studied Apollonian packings of circles in two-dimensional Euclidean space in terms of the Descartes configurations they contain. A Descartes configuration is an arrangement of four mutually tangent circles on the Riemann sphere which have disjoint interiors. We identify Apollonian packings $\mathcal{P}$ with the set $\mathbb{D}(\mathcal{P})$ of all ordered Descartes configurations they contain. We studied various groups acting on the ensemble of Descartes configurations, one action coming from the conformal group Möb(2) acting on the Riemann sphere, inducing an action on Descartes configurations, and a linear action on Descartes configurations represented by 4×3 matrices in "curvature-center coordinates", as described in part I. The group associated to the latter is the group $Iso^{\uparrow}(Q_2)$ which is a subgroup of index 2 in the group $Iso(Q_2)$ of real automorphs of the Descartes quadratic form $Q_2 = I_4 - \frac{1}{2}\mathbf{1}_4^T\mathbf{1}_4$. A certain discrete subgroup of the group $Iso^{\uparrow}(Q_2)$, which we called the Apollonian group, leaves Apollonian packings invariant. We also considered a larger group, the super-Apollonian group, that can be used to define super-Apollonian packings. These groups had a representation using 4×4 integer matrices, and we found there were distiguished circle packings in which all curvatures were integral, which we called integral Apollonian circle packings, and other packings where the curvatures were integral and the centers$\times$curvatures were also integral vectors, which we called strongly integral Apollonian packings.

In this paper we generalize these results to higher dimensions. We call any set $\mathcal{D} = (C_1, C_2, ..., C_{n+2})$ of $n+2$ mutually tangent spheres in n-dimensions, having disjoint interiors, an *n-dimensional Descartes configuration.* Given any set of $n+1$ mutually tangent (n-1)-spheres having disjoint interiors, there are exactly two spheres tangent to all of them, cf. Pedoe [25]. Such a set gives rise to two n-dimensional Descartes configurations. There is an inversion operation, given by an n-dimensional Möbius transformation in the n-dimensional conformal group Möb(n), mapping one to the other, cf. Pedoe [25, pp. 630-631]. Starting with

an initial Descartes configuration, we can now obtain an ensemble of spheres in n-dimensions by successively adding spheres by such reflection operations. We call the completed set of spheres an *Apollonian sphere ensemble*. It is a sphere packing in dimensions 2 and 3, but for $n \geq 4$ the spheres overlap and it is not a packing, cf. Boyd [3]. However the *Apollonian cluster ensemble* consisting of all n-dimensional Descartes configurations generated by the underlying group of inversions, makes sense in all dimensions. We show that most of the results which hold for 2-dimensional Descartes configurations and Apollonian circle packings viewed as sets of Descartes configurations have n-dimensional analogues.

In §2 we prove characterizations of n-dimensional Descartes configurations which generalize the Descartes circle theorem. We use the n-dimensional version of the curvature-center coordinates introduced in parts I and II. For related results in spherical and hyperbolic space, see [21].

In §3 we show that the group-theoretic constructions of parts I and II have n-dimensional analogues, even though the associated collections of Descartes configurations no longer correspond to packings. We call them *Apollonian sphere ensembles*. We construct the n-dimensional analogues of the Apollonian group, and super-Apollonian group. These groups consist of integer matrices in dimensions 2 and 3 but not for dimensions 4 and higher. However a related group, the dual Apollonian group, is a group of integer matrices in all dimensions. Given a finite set of primes S, an S-integer is any rational number whose denominator is divisible only by powers of primes in S. The entries of the Apollonian group and super-Apollonian group are S-integers where S consists of the prime divisors of $n-1$ if n is even and $\frac{1}{2}(n-1)$ if n is odd.

In §4 we consider integral and rational Apollonian sphere ensembles. In all dimensions $n \geq 2$ there exist Apollonian sphere ensembles in which all spheres have curvatures which are S-integers, where S consists of the prime divisors of $n-1$ if n is even and $\frac{1}{2}(n-1)$ if n is odd. An Apollonian sphere ensemble is *strongly rational* if the curvature of every sphere in the packing is rational, and the center of every sphere is a rational vector. We show that a necessary and sufficient condition for a strongly rational Apollonian sphere ensembles to exist in dimension n is that $n = 2k^2$ or $n = (2k+1)^2$ for some positive integer k. In these dimensions there exist Apollonian sphere ensembles in which all curvatures and curvature-center quantities

are S-integers for some fixed S. We do not determine an explicit choice of S, for allowable $n > 2$, however.

In §5 we consider a higher-dimensional analogue of the duality operation introduced in part II. The two-dimensional duality operator studied in part II was a geometric operation which led to a symmetry relating the generators of the super-Apollonian group under the transpose operation. In dimensions 3 and higher the "duality" operation no longer respects packings, and the generators of the associated super-Apollonian group are not preserved by the transpose operation. We show however that in higher dimensions the geometric analogue of the duality operation encodes an "equiangularity" property instead.

In §6 we briefly study the n-dimensional variant of Coxeter's study of loxodromic sequences of tangent spheres, inside an Apollonian sphere ensemble.

In the conclusion §7 we state some open problems.

Notation. In this paper, following earlier notation, the symbol C refers to an n-dimensional sphere ("n-dimensional circle"). The notion of augmented matrix $\tilde{N}_{\mathcal{D}}$ introduced in §2 adds the augmentation in the last column, while that used in [21] adds the augmentation as the second column of the matrix. For a row vector $\mathbf{x}$ in $\mathbb{R}^n$, its squared norm is $|\mathbf{x}|^2 = \mathbf{x}\mathbf{x}^T = \sum_{i=1}^n x_i^2$.

2. Generalized Descartes Theorem

The Descartes circle theorem generalizes to n-dimensional Euclidean space. A 3-dimensional analogue of the Descartes formula was found in 1886 by Lachlan [19, p. 498] and rediscovered in 1936 by Soddy [28]. The result of Soddy was extended to n-dimensions by Gossett [14]. It relates the (oriented) curvatures of $n+2$ mutually tangent n-spheres, forming an oriented Descartes configuration. Here an *orientation* of a sphere consists of a unit normal direction, pointing inward or outward. The *oriented curvature* of an oriented sphere is $a_i = \frac{1}{r_i}$ if it is inwardly oriented and is $a_i = -\frac{1}{r_i}$ if it is outwardly oriented. We define the *interior* of an oriented sphere to be either its interior or exterior according to the orientation being inward or outward, respectively.

Definition 2.1. An *oriented Descartes configuration* is a set of $n+2$ oriented spheres in $\mathbb{R}^n$, which are mutually tangent, such that either (i) each pair of oriented interiors are disjoint, or

(ii) each pair of oriented interiors intersect. We call these two cases (i) the positively oriented case and (ii) the negatively oriented case.

Given a positively oriented Descartes configuration, one obtains a negatively oriented Descartes configuration by reversing all orientations, and vice versa.

Theorem 2.1 (Soddy-Gossett Theorem) *Given an oriented Descartes configuration $\mathcal{D}$ in $\mathbb{R}^n$, its oriented curvatures $\{a_i \ : \ 1 \le i \le n+2\}$ satisfy*

$$\sum_{i=1}^{n+2} a_i^2 = \frac{1}{n}(\sum_{i=1}^{n+2} a_i)^2. \tag{2.1}$$

A proof of (2.2) can be found in Pedoe [25]; it also follows from Theorem 3.3 of [21]. This result can be rewritten as

$$\mathbf{a}^T \mathbf{Q}_n \mathbf{a} = 0, \tag{2.2}$$

where $\mathbf{a} := (a_1, \dots, a_{n+2})$ and $\mathbf{Q}_n$ is the symmetric matrix of the Descartes quadratic form, given by

$$\mathbf{Q}_n := I_{n+2} - \frac{1}{n}\mathbf{1}_{n+2}\mathbf{1}_{n+2}^T, \tag{2.3}$$

with $\mathbf{1}_{n+2}$ denoting a column of $n+2$ 1's.

Theorem 2.2 (Converse to Soddy-Gossett Theorem) *(i) Each nonzero real column vector $\mathbf{a} = (a_1, ..., a_{n+2})$ that satisfies the Descartes relation*

$$\sum_{i=1}^{n+2} a_i^2 = \frac{1}{n}(\sum_{i=1}^{n+2} a_i)^2. \tag{2.4}$$

is the set of oriented curvatures of some oriented Descartes configuration $\mathcal{D}$ in $\mathbb{R}^n$.

(ii) Any two oriented Descartes configurations with the same curvature vector are congruent, i.e. there is a Euclidean motion taking one to the other.

We will prove this result at the end of this section, after we have established some more general characterizations of oriented Descartes configurations.

For the next result we let $\mathcal{D}$ denote a general configuration of $n+2$ oriented spheres in $\mathbb{R}^n$, not necessarily an oriented Descartes configuration. If it is an oriented Descartes configuration

with $\sum_{i=1}^{n+2} a_i > 0$, then one of the following holds. (i) all of $a_1,\ a_1, ...,\ a_{n+2}$ are positive; (ii) $n+1$ are positive and one is negative; (iii) $n+1$ are positive and one is zero; or (iv) n are positive and equal and the other two are zero. These four cases correspond respectively to the following configurations of mutually tangent spheres: (i) $n+1$ spheres, with another in the curvilinear simplex that they enclose; (ii) $n+1$ spheres inscribed inside another larger sphere; (iii) n spheres with a common tangent plane (the $(n+1)$-st "sphere"), with another sphere between them; (iv) n equal spheres with two common parallel tangent planes.

Definition 2.2. (i) Given an oriented sphere C in $\mathbb{R}^n$ with oriented curvature $a = a(C)$, and center $(x_1, x_2, ..., x_n)$ its *curvature-center coordinates* $\mathbf{w}(C)$ are given by the row vector

$$\mathbf{w}(C) := (a, \mathbf{c}) = (a, ax_1, ax_2, ..., ax_n). \tag{2.5}$$

where $\mathbf{c} = (a(C)x_1, ..., a(C)x_n)$.

(ii) We regard a hyperplane as a "degenerate" sphere. Given an oriented hyperplane H with specified unit normal vector $\mathbf{h} := (h_1, h_2, \dots, h_n)$, its *curvature-center coordinates* $\mathbf{w}(H)$ are given by

$$\mathbf{w}(H) := (0, h_1, h_2, \dots, h_n). \tag{2.6}$$

Definition 2.3. Given a configuration $\mathcal{D} = (C_1, C_2, ..., C_{n+2})$ of $n+2$ oriented spheres in $\mathbb{R}^n$ (allowing some to be hyperplanes), define its *curvature-center matrix* $N_{\mathcal{D}}$ to be the $(n+2) \times (n+1)$ matrix whose rows are

$$(N_{\mathcal{D}})_i := \mathbf{w}(C_i) = (a(C_i), a(C_i)x_{i,1}, ..., a(C_i)x_{i,n}). \tag{2.7}$$

It is easy to see that the matrix $N = N_{\mathcal{D}}$ determines the oriented sphere configuration $\mathcal{D}$ uniquely.

Theorem 2.3 (n-Dimensional Euclidean Descartes Theorem) *An $(n+2)\times(n+1)$ real curvature-center matrix N has $N = N_{\mathcal{D}}$ for some oriented Descartes configuration $\mathcal{D}$ if and only if*

$$N^T \mathbf{Q}_n N = \begin{bmatrix} 0 & 0 \\ 0 & 2I_n \end{bmatrix} = diag(0,\ 2,\ 2, ...,\ 2). \tag{2.8}$$

Futhermore this Descartes configuration is positively oriented if and only if

$$\sum_{i=1}^{n+2} N_{i,1} > 0. \tag{2.9}$$

Remark. The Soddy-Gossett theorem (2.1)appears as the $(1,1)$-entry of the matrix equation (2.8). Note that in (2.9) the value $N_{i,1} = a_i$ is the oriented curvature of the i-th sphere in the oriented Descartes configuration.

Proof. Let $\mathcal{D}$ be an oriented Descartes configuration, and we must prove (2.8). We first treat the case where no curvature vanishes, i.e. the Descartes configuration contains no hyperplanes. Later we obtain the remaining cases by a limiting process. We use matrix notation. Recall that $J = \mathbf{1}\mathbf{1}^T$, where $\mathbf{1} = (1, 1, ..., 1)^T$ is an $(n+2) \times 1$ column vector. Let $X = [x_{i,j}]$ be the $(n+2) \times n$ matrix of sphere centers, and set $R = diag(r_1, r_2, \cdots, r_{n+2})$, where the r_i are the oriented radii of the spheres. We are assuming that all radii r_i are finite, so R is invertible. Note that one radius is assigned a negative sign if the sphere corresponding to it encloses the other spheres. Then $A := R^{-1}$ is the diagonal matrix of curvatures. and we have $N_{\mathcal{D}} = [R^{-1}\mathbf{1}, R^{-1}X] = [A\mathbf{1}, AX]$.

Without loss of generality we may rescale all coordinates by a positive constant factor λ, sending x_j to λx_j and r_j to λr_j. This rescales the first column of $N_{\mathcal{D}}$, leaving the other columns unchanged, and the relation (2.8) is preserved because the first column is an isotropic vector with respect to the indefinite bilinear form given by Q_n, i.e. it has inner product zero with all vectors. We choose the rescaling to make

$$\mathbf{1}^T A\mathbf{1} = \sum_{i=1}^{n+2} \frac{1}{r_i} = n. \tag{2.10}$$

The Soddy-Gossett relation, which is the $(1,1)$-entry of (2.8), then implies that

$$\mathbf{1}^T A^T A\mathbf{1} = \sum_{i=1}^{n+2} \frac{1}{r_i^2} = \frac{1}{n}(\sum_{i=1}^{n+2} \frac{1}{r_i})^2 = n, \tag{2.11}$$

see Theorem 3.3 of [21].

We next note that (2.8) is preserved under a translation of all sphere centers, because this subtracts a multiple of the first column of $N_{\mathcal{D}}$ from each other column, and this leaves

$N_{\mathcal{D}}^T Q_n N_{\mathcal{D}}$ unchanged, again because the first column is an isotropic vector. Without loss of generality we may now translate all the spheres to make

$$\mathbf{a}^T X = \mathbf{1}^T A X = \mathbf{0}^T, \tag{2.12}$$

and since the curvatures don't change, (2.10) and (2.11) still hold. It therefore suffices to prove the theorem in this special case.

Assuming that (2.10)- (2.12) all hold, we have

$$\begin{aligned} N_{\mathcal{D}}^T Q_n N_{\mathcal{D}} &= [A\mathbf{1}, AX]^T \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \right) [A\mathbf{1}, AX] \\ &= \begin{bmatrix} \mathbf{1}^T A^T A\mathbf{1} - \frac{1}{n}(\mathbf{1}^T A\mathbf{1})^2 & \mathbf{1}^T A^T AX - \frac{1}{n}(\mathbf{1}^T A^T \mathbf{1})(\mathbf{1}^T AX) \\ X^T A^T A\mathbf{1} - \frac{1}{n}(X^T A\mathbf{1})(\mathbf{1}^T AX) & X^T A^T AX - \frac{1}{n}(X^T A\mathbf{1})(\mathbf{1}^T AX) \end{bmatrix} \\ &= \begin{bmatrix} n - \frac{1}{n}(n^2) & \mathbf{1}^T A^T AX \\ X^T A^T A\mathbf{1} & X^T A^T AX \end{bmatrix}. \end{aligned} \tag{2.13}$$

The upper left corner of this block-partitioned matrix is zero, so to prove that it equals $diag(0, 2, 2, \ldots, 2)$ it remains to prove that

$$\mathbf{1}^T A^T AX = \mathbf{a}^T AX = \mathbf{0}^T, \tag{2.14}$$

and

$$X^T A^2 X = 2I_n. \tag{2.15}$$

The condition that two spheres with radii r_i and r_j touch is that

$$|\mathbf{x}_i - \mathbf{x}_j|^2 = (r_i + r_j)^2. \tag{2.16}$$

If we set

$$D = \mathrm{diag}(XX^T) = \mathrm{diag}(|\mathbf{x}_1|^2, ..., |\mathbf{x}_{n+2}|^2). \tag{2.17}$$

then the condition that all the spheres mutually touch is the matrix equality

$$D\mathbf{1}\mathbf{1}^T - 2XX^T + \mathbf{1}\mathbf{1}^T D = R^2\mathbf{1}\mathbf{1}^T + 2R\mathbf{1}\mathbf{1}^T R + \mathbf{1}\mathbf{1}^T R^2 - 4R^2, \tag{2.18}$$

in which the (i,j)-th entry is (2.16). Multiplying by A $:=R^{-1}$ on the left and on the right, we may rewrite this as

$$AD\mathbf{1}\mathbf{a}^T - 2AXX^TA + \mathbf{a}\mathbf{1}^TDA = A^{-1}\mathbf{1}\mathbf{a}^T + 2\mathbf{1}\mathbf{1}^T + \mathbf{a}\mathbf{1}^TA^{-1} - 4I. \tag{2.19}$$

Note that from (2.10),

$$\mathbf{f} := \frac{1}{\sqrt{2}}(\mathbf{1} - \mathbf{a}); \qquad \mathbf{g} := \frac{1}{\sqrt{n}}\ \mathbf{a} \tag{2.20}$$

are orthogonal unit vectors. Now define

$$\alpha := \mathbf{1}^TR\mathbf{1} = \mathbf{1}^TA^{-1}\mathbf{1}. \tag{2.21}$$

Pre-multiplying (2.19) by $\mathbf{1}^T$ and post-multiplying by $\mathbf{1}$, and using (2.10) and (2.12), we find that

$$\mathbf{a}^TD\mathbf{1} = \alpha + \ n + \ 2. \tag{2.22}$$

Similarly, pre-multiplying (2.19) by $\mathbf{1}^T$ and post-multiplying by $\mathbf{a}$, we obtain

$$\mathbf{a}^TD\mathbf{1} + \mathbf{1}^TDA\mathbf{a} = \alpha + \ 3n + \ 2. \tag{2.23}$$

and pre-multiplying (2.19) by $\mathbf{a}^T$ and post-multiplying by $\mathbf{a}$, we obtain

$$\mathbf{a}^TAD\mathbf{1} - \frac{1}{n}\mathbf{a}^TAXX^TA\mathbf{a} = 2n. \tag{2.24}$$

From (2.22) and (2.23), we obtain

$$\mathbf{1}^TDA\mathbf{a} = 2n, \tag{2.25}$$

so from (2.24) we get $\mathbf{a}^TAXX^TA\mathbf{a} = 0$, i.e.

$$\mathbf{a}^TAX = \mathbf{0}^T. \tag{2.26}$$

Now post-multiplying (2.19) by $\mathbf{1}$, we find

$$A^{-1}\mathbf{1} - AD\mathbf{1} = \frac{n+2}{n}\mathbf{a} - 2\ \mathbf{1}, \tag{2.27}$$

whence from (2.19), we have

$$AXX^TA = 2I_{n+2} - (\mathbf{1} - \mathbf{a})(\mathbf{1}^T - \mathbf{a}^T) - \frac{2}{n}\mathbf{a}\mathbf{a}^T = 2(I_{n+2} - \mathbf{f}\mathbf{f}^T - \mathbf{g}\mathbf{g}^T). \tag{2.28}$$

Thus $\frac{1}{\sqrt{2}}AX$ is a $(n+2)\times n$ section of an orthogonal matrix, so

$$X^T A^2 X = 2I_n, \tag{2.29}$$

which completes the proof that an n-dimensional Descartes configuration satisfies (2.8), when all curvatures are nonzero.

It remains to consider the limiting cases of configurations in which one sphere has curvature zero, say $a = 0$, i.e. it is a hyperplane H, with equation $\mathbf{x}^T\mathbf{h} = p$, where $\mathbf{h}$ is the oriented unit normal vector, pointing to the correct half-space. Choosing $\mathbf{x} \in H$, one can obtain H as a limit of a sequence of spheres with radius r centered at $\mathbf{x} + r\mathbf{h}$ as $r \to \infty$. The curvature×center of these spheres converges to $(0, h_1, \ldots h_n)$, independent of the choice of $\mathbf{x}$. Thus all is well provided we define $\mathbf{a}^T A := \mathbf{h}^T$. If two curvatures $a = b = 0$, then the remaining sphere centers must lie at the vertices of a regular simplex (with side $\frac{2}{c}$) and (2.8) is trivial in this case.

If a Descartes configuration is inwardly oriented, then either all curvatures are nonnegative, or exactly one is negative, corresponding to one sphere enclosing the others. Certainly if $\mathcal{D}$ has all (oriented) curvatures are positive or zero, then (2.9) holds. If one is negative, i.e. its sphere encloses the others, then the sphere having negative oriented curvature has the smallest curvature in absolute value, so condition (2.9) is satisfied. An outwardly oriented configuration reverses all signs of an inward one, so (2.9) does not hold.

That the conditions (2.8) and (2.9) always yield an n-dimensional Descartes configuration follows by reversing the above argument. First assume the first column of $N_{\mathcal{D}}$ has no zero entries. Given (2.8) holds, the curvatures satisfy the Soddy-Gossett relation (2.1), and by rescaling and translating as necessary we may assume (2.10) and (2.12) both hold. Here the rescaling is by a positive λ since (2.9) holds. Then $\mathbf{f}$ and $\mathbf{g}$ are orthogonal unit vectors. We now need to prove (2.19). From (2.8) we have (2.26), so that

$$\mathbf{f}^T AX = \mathbf{g}^T AX = \mathbf{0}. \tag{2.30}$$

From (2.8) again we have (2.29), so that the $(n+2)\times(n+2)$ matrix

$$[\mathbf{f}, \quad \mathbf{g}, \quad \frac{1}{\sqrt{2}}AX\ ] \tag{2.31}$$

is orthogonal, hence (2.28) holds. The diagonal of the matrix AXX^TA is

$$ADA = 2I_{n+2} - 2\cdot\frac{1}{2}\,(I_{n+2} - A)^2 - 2\cdot\frac{1}{n}A^2 \tag{2.32}$$

and it follows that

$$AD\mathbf{1} = A^{-1}\mathbf{1} + 2\ \mathbf{1} - \frac{n+2}{n}\ \mathbf{a}, \tag{2.33}$$

as required. This proves (2.19) in the case that no curvature vanishes.

In the remaining case where an element in the first column of $N_{\mathcal{D}}$ vanishes, any solution M satisfying (2.8) and (2.9) arises as a limit of such $N_{\mathcal{D}}$ in which all elements of the first column are nonzero. The limit of the corresponding Descartes configuration exists and gives the Descartes configuration corresponding to $N_{\mathcal{D}}$. ∎

Theorem 2.3 has a further generalization, which extends the $(n+2) \times (n+1)$ matrix $N_{\mathcal{D}}$ to an $(n+2) \times (n+2)$ augmented matrix $\tilde{N}_{\mathcal{D}}$ obtained by adding an additional column. The augmented matrix $\tilde{N}_{\mathcal{D}}$ involves information concerning two (oriented) Descartes configurations, the original one and one obtained from it by inversion in the unit sphere, as we now explain. This construction, which appears unmotivated here, was originally discovered in generalizing the Descartes theorem to spherical and hyperbolic geometry, as described in Lagarias, Mallows and Wilks [21, Section 4].

In n-dimensional Euclidean space, the operation of *inversion in the unit sphere* replaces the point $\mathbf{x}$ by $\mathbf{x}/|\mathbf{x}|^2$, where $|\mathbf{x}|^2 = \sum_{j=1}^{n} x_j^2$. Consider a general sphere C with center $\mathbf{x}$ and oriented radius r. Then inversion in the unit sphere takes C to the sphere $\bar{C}$ with center $\bar{\mathbf{x}} = \mathbf{x}/(|\mathbf{x}|^2 - r^2)$ and oriented radius $\bar{r} = r/(|\mathbf{x}|^2 - r^2)$. Note that if $|\mathbf{x}|^2 > r^2$, $\bar{C}$ has the same orientation as C. The inversion may take some spheres to hyperplanes, and vice-versa, as well as sending some hyperplanes to hyperplanes. In all cases,

$$\frac{\mathbf{x}}{r} = \frac{\bar{\mathbf{x}}}{\bar{r}}.$$

Definition 2.4. (i) Given an oriented sphere C with oriented curvature $a = a(C)$ and center $(x_1, ...x_n)$, with inverse oriented sphere $\bar{C}$ having oriented curvature $\bar{a} = a(\bar{C})$. Then its *augmented curvature-center coordinates* are

$$\tilde{\mathbf{w}}(C) := (a(C), \mathbf{c}(C), a(\bar{C})) = (a(C), a(C)x_1, ..., a(C)x_n, \bar{a}). \tag{2.34}$$

(ii) Given an oriented hyperplane H, with inverse $\bar{H}$ in the unit sphere, its *augmented curvature-center coordinates* $\tilde{\mathbf{w}}(H)$ are given by

$$\tilde{\mathbf{w}}(H) := (a(H), \mathbf{c}(H), a(\bar{H}) = (0, h_1, ..., h_n, a(\bar{H})). \tag{2.35}$$

Definition 2.5. Given a configuration $\mathcal{D} = (C_1, C_2, ..., C_{n+2})$ of $(n+2)$ oriented spheres in $\mathbb{R}^n$, in which some spheres may be hyperplanes, its *augmented curvature-center matrix* is the $(n+2) \times (n+2)$ matrix $\tilde{N}_{\mathcal{D}}$ with rows

$$(\tilde{N}_{\mathcal{D}})_i := (\ (N_{\mathcal{D}})_i, \bar{a}_i), \tag{2.36}$$

in which $\bar{a}_i$ is the signed curvature of the inverse sphere $\bar{C}_i$ to C_i.

Given an oriented sphere C, we have

$$\bar{a} = \frac{|\mathbf{x}|^2}{r} - r, \quad \mathbf{c}(\bar{C}) = \mathbf{c}(C) = \frac{\mathbf{x}}{r}. \tag{2.37}$$

Notice that the relation $\tilde{\mathbf{w}}(\bar{C}) = (\bar{a}, \mathbf{c}(C), a(C))$ enables us to extend the definition of $\mathbf{w}(C)$ to degenerate spheres with infinite radius; simply find $\mathbf{w}(\bar{C})$ and interchange the first and last coordinates. If H is a hyperplane containing the origin, then $H = \bar{H}$, $a = \bar{a} = 0$ and $\mathbf{c}$ is a unit vector orthogonal to H.

Theorem 2.4 (Augmented Euclidean Descartes Theorem) *An $(n+2) \times (n+2)$ real matrix $\tilde{N}$ is the augmented curvature$\times$center matrix $\tilde{N}_{\mathcal{D}}$ of some oriented Descartes configuration $\mathcal{D}$ in $\mathbb{R}^n$ if and only if*

$$\tilde{N}^T \mathbf{Q}_n \tilde{N} = \begin{bmatrix} 0 & 0 & -4 \\ 0 & 2I_n & 0 \\ -4 & 0 & 0 \end{bmatrix}. \tag{2.38}$$

The augmented Euclidean Descartes Theorem implies one direction of the n-dimensional Euclidean Descartes Theorem, namely that all oriented Descartes configurations satisfy (2.8). The converse direction, that all sphere configurations satisfying (2.8) are oriented Descartes configurations, requires an additional argument, given in the proof of Theorem 2.3.

We proceed to prove the augmented Euclidean Descartes theorem, via a preliminary lemma. Given a real number λ, define the matrix

$$\mathbf{K}_n(\lambda) := \begin{bmatrix} 0 & 0 & -\lambda \\ 0 & 2I_n & 0 \\ -\lambda & 0 & 0 \end{bmatrix}, \tag{2.39}$$

Note that $\mathbf{K}_n(4)$ appears in the theorem above, and a calculation reveals that

$$\mathbf{K}_n(\lambda)^{-1} = \frac{1}{4}\mathbf{K}_n(\frac{4}{\lambda}). \tag{2.40}$$

Lemma 2.5. *(i) For any $(n+2)$-vector $\tilde{\mathbf{w}}$, there is a sphere (or hyperplane) C in $\mathbb{R}^n$ with $\tilde{\mathbf{w}}(C) = \tilde{\mathbf{w}}$ if and only if*

$$\tilde{\mathbf{w}}\mathbf{K}_n(1)\tilde{\mathbf{w}}^T = 2.$$

(ii) The oriented spheres C and C' are externally tangent if and only if

$$\tilde{\mathbf{w}}(C)\mathbf{K}_n(1)\tilde{\mathbf{w}}(C')^T = -2.$$

Proof. . (i) This restates the relation $b\bar{b} = (|\mathbf{x}|^2 - r^2)/r^2 = |\mathbf{c}|^2 - 1$.

(ii) This is an immediate consequence of $|\mathbf{x} - \mathbf{x}'|^2 = (r + r')^2$. ∎

Proof of the Augmented Euclidean Descartes Theorem. Suppose $\tilde{N} = \tilde{N}_{\mathcal{D}}$ for some configuration of $(n+2)$ oriented spheres. From Lemma 2.5(ii), if the spheres all touch externally, we have

$$\tilde{N}\mathbf{K}_n(1)\tilde{N}^T = 4\mathbf{I}_{n+2} - 2\mathbf{1}_{n+2}\mathbf{1}_{n+2}^T = 4(\mathbf{Q}_n)^{-1} \tag{2.41}$$

Next, recall the the matrix identity that if A, B are symmetric non-singular $n \times n$ matrices satisfying $WAW^T = B$, then [2]

$$W^T B^{-1} W = A^{-1}. \tag{2.42}$$

Apply this identity with $A = \mathbf{K}_n(1)$ and $W = \tilde{N}$ noting that

$$4\mathbf{K}_n(1)^{-1} = \mathbf{K}_n(4), \tag{2.43}$$

to obtain (2.38).

For the converse direction, suppose $\tilde{N}$ satisfies (2.38). This matrix equation implies (2.41), using the identity (2.42). The diagonal terms in (2.41) imply, using Lemma 2.5(i), that $\tilde{N} = \tilde{N}_{\mathcal{D}}$ for some configuration of oriented spheres $\mathcal{D}$. Then Lemma 2.5(ii), implies that the spheres touch externally pairwise. ∎

Proof of Converse to Soddy-Gossett Theorem. (i) We start from a standardized Descartes configuration, which is the one with two parallel hyperplanes at distance 2 from each other, and n-unit spheres in between them, whose centers form a regular $(n-1)$-simplex lying

[2] Clearly W must be nonsingular. Invert both sides, and multiply on the left by W^T and on the right by W.

in the hyperplane parallel to the two hyperplanes in the configuration and halfway between them. Let its augmented matrix be W_0, and oriented curvature vector be $\mathbf{a}_0$. The automorphism group $Aut(\mathbf{Q}_n)$ is the set of all $(n+2)\times(n+2)$ matrices $\mathbf{U}$ such that $\mathbf{U}^T\mathbf{Q}_n\mathbf{U} = \mathbf{Q}_n$. The matrix $\mathbf{U}W_0$ clearly satisfies (2.38) since W_0 does, so by the augmented Euclidean Descartes Theorem, the matrix $\mathbf{U}W_0$ is itself the augmented matrix of some oriented Descartes configuration. In particular its first column $\mathbf{U}\mathbf{a}_0$ are the oriented curvatures of some oriented Descartes configuration.

Now since $\mathbf{Q}_n$ has signature $(1, n+1)$, there exists a real matrix $\mathbf{V}$ such that

$$\mathbf{V}^T\mathbf{Q}_n\mathbf{V} = \mathbf{Q}_{\mathcal{L}} := diag(-1, 1, 1, ..., 1).$$

This quadratic form, the $(n+2)$-dimensional Lorentzian form, has automorphism group $O(1, n+1)$, and it is known that $O(1, n+1)$ acts transitively on the nonzero elements of the *null cone* (or *light cone*)

$$\mathbf{b}^T\mathbf{Q}_{\mathcal{L}}\mathbf{b} = 0$$

of the Lorentzian form. Pulling back to $\mathbf{Q}_n$, we find that

$$Aut(\mathbf{Q}_n) = \mathbf{V}O(1, n+1)\mathbf{V}^{-1}$$

and that the action of $Aut(\mathbf{Q}_n)$ acts transitively on the nonzero solutions to $\mathbf{a}^T\mathbf{Q}_n\mathbf{a} = 0$. That is, for any non-zero vector $\mathbf{a}$ satisfying $\mathbf{a}^T\mathbf{Q}_n\mathbf{a} = 0$, there exists a matrix $\mathbf{U} \in Aut(\mathbf{Q}_n)$ such that $\mathbf{a} = \mathbf{U}\mathbf{a}_0$. Then by the argument at the beginning of the proof, $\mathbf{U}W_0$ is the augmented matrix of an oriented Descartes configuration whose vector of oriented curvatures is $\mathbf{a}$.

(ii) We may assume that the two Descartes configurations with the same curvature vector $\mathbf{a}$ are positively oriented. This is because the orientation of a Descartes configuration is determined by the signs of the a_i's, and any negatively oriented Descartes configuration can be obtained from a positively oriented one by reversing all orientations. By Definition 2.1, one of the following holds for $\mathbf{a} = (a_1, a_2, \ldots, a_{n+2})$. (a) all of $a_1, a_2, \ldots, a_{n+2}$ are positive; (b) $n+1$ are positive and one is negative; (c) $n+1$ are positive and one is zero; or (d) n are positive and equal and the other two are zero. For case (d), the theorem holds trivially. For cases (b) and (c), suppose $a_1 \leq 0$. Let C' be the sphere that is tangent to $C_2, \ldots, C_{n+2}$ but not equal to

C_1. By Soddy-Gossett Theorem, the curvature a' of C' equals $2(a_2 + \cdots + a_{n+2})/(n-1) - a_1$ which is positive and finite. Thus C' is positive oriented with finite radius. Since the position of C_1 is uniquely determined by that of $C_2, \ldots, C_{n+2}, C'$, cases (b) and (c) are reduced to (a).

Now we treat the case (a) in which all spheres have finite radius and positive oriented curvatures. We use the fact that an n-simplex is completely determined, up to congruence, by the lengths of its $\frac{n(n-1)}{2}$ edges (between labelled vertices). Any set of $n+1$ externally touching spheres is rigid, because the set of sphere centers forms an n-simplex in which the distance along the edge from center of S_i to center of S_j is $r_i + r_j$. Given two oriented Descartes configurations that have the same oriented curvature vector, the simplices determined by the first $n+1$ sphere centers are congruent, hence there is a Euclidean motion taking the first $n+1$ spheres of one to the first $n+1$ spheres of the other. Euclidean motions preserve tangencies, and the remaining sphere of the initial configuration must therefore be mapped to a sphere tangent to the other's first $n+1$ spheres. There are only two choices for the image sphere, and the Euclidean motion can map to the wrong image only if the second configuration has two tangent spheres of equal size. But if this happens, there is also a reflection of the second configuration taking this image configuration into the other. This finishes the proof. ∎

3. n-Dimensional Apollonian Ensembles and Group Actions

The n-dimensional (generalized) Möbius group Möb(n) is the set of conformal isomorphisms of the n-sphere $\hat{\mathbb{R}}^n = \mathbb{R}^n \cup \{\infty\}$ to itself, allowing orientation-reversing maps of the n-sphere. This notion of orientation is unrelated to the notion of an oriented Descartes configuration in §2; in fact the action of the n-dimensional Möbius group preserves orientation of Descartes configurations. In this section all Descartes configurations will be given the positive orientation [3] unless otherwise noted. Each group element $\mathfrak{g}$ acts as a permutation of individual $(n-1)$-spheres, and also induces a permutation action on the set $\mathbb{D}_n$ of all n-dimensional positively oriented Descartes configurations.

The following result is a straightforward generalization of Theorem 2.9 of part I. Given an oriented Descartes configuration $\mathcal{D}$, let $N_{\mathcal{D}}$ denote the $(n+2) \times (n+1)$ matrix assigned to it

[3] This conforms with the notation in parts I and II, which treated positively oriented Descartes configurations only.

in Theorem 2.3, and let $\tilde{N}_{\mathcal{D}}$ denote the $(n+2)\times(n+2)$ augmented matrix associated to $\mathcal{D}$ in Theorem 2.4.

Theorem 3.1. *There is a representation $\rho_n : M\ddot{o}b(n) \to GL(n+2,\mathbb{R})$ with each $\rho_n(\mathfrak{g}) = G_{\mathfrak{g}}$ having determinant ± 1, such that for all n-dimensional oriented Descartes configurations $\mathcal{D}$,*

$$\tilde{N}_{\mathfrak{g}(\mathcal{D})} = \tilde{N}_{\mathcal{D}} G_{\mathfrak{g}}^T. \tag{3.1}$$

Proof. It suffices to verify Theorem 3.1 on a set of generators of Möb(n). A general element of Möb(n) is either a similarity of $\mathbb{R}^n$ or the product of an inversion and an isometry of $\mathbb{R}^n$, cf. Wilker [31, §5, Corollary 1]. That is, Möb(n) is generated by the following operators: dilatations $\mathfrak{d}_k(v) = kv$ where $k \in \mathbb{R}$, $k \neq 0$, translations $\mathfrak{t}_{v_0}(v) = v + v_0$, where $v_0 \in \mathbb{R}^n$ is a row vector, rotations $\mathfrak{r}_O(v) = Ov$, where O is an orthogonal matrix of size n, and the inversion in the unit circle $\mathfrak{j}_C(v) = \frac{v}{|v|^2}$.

Direct computation shows that formula (3.1) holds for $\mathfrak{d}_k$, $\mathfrak{t}_{v_0}$, $\mathfrak{r}_O$, and $\mathfrak{j}_C$, where the right multiplication are given by the matrices

$$\begin{aligned}
G_{\mathfrak{d}_k}^T &:= \begin{bmatrix} \frac{1}{k} & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & k \end{bmatrix}, \\
G_{\mathfrak{t}_{v_0}}^T &:= \begin{bmatrix} 1 & v_0 & |v_0|^2 \\ 0 & I_n & 2v_0^T \\ 0 & 0 & 1 \end{bmatrix}, \\
G_{\mathfrak{r}_O}^T &:= \begin{bmatrix} 1 & 0 & 0 \\ 0 & O & 0 \\ 0 & 0 & 1 \end{bmatrix},
\end{aligned}$$

and $G_{\mathfrak{j}_C}^T := P_{1,n+2}$, the permutation matrix which permutes the first and $(n+2)$th entries. ∎

Given an n-dimensional Descartes configuration $\mathcal{D} = \{C_1, C_2, \ldots, C_{n+2}\}$ one can generate new Descartes configurations using *reflection operators* $\mathfrak{s}_i = \mathfrak{s}_i[\mathcal{D}] \in$ Möb(n) for $1 \leq i \leq n+2$, in which $\mathfrak{s}_i$ is the unique Möbius transformation that maps the sphere C_i to the other sphere C_i' which is tangent to all the remaining C_j, while leaving the other C_j invariant, cf. Pedoe [25, p. 630] and Wilker [31, Theorem 3]. The Möbius transformation $\mathfrak{s}_i[\mathcal{D}] := \mathfrak{j}_{C_i^{\perp}}$ is inversion with respect to the unique $(n-1)$-sphere $C_i^{\perp}$ which passes through the $\frac{n(n+1)}{2}$ points of tangency of the other $n+1$ circles $\{C_j : j \neq i\}$. The existence of $C_i^{\perp}$ is given by the following well-known result.

Proposition 3.2. *Given $n+1$ mutually tangent $(n-1)$-spheres $\{C_i : 1 \le j \le n+1\}$ in $\mathbb{R}^n$ having disjoint interiors, there exists a unique $(n-1)$-sphere $C^{\perp}$ passing through the $\frac{n(n-1)}{2}$ tangency points of these spheres. At each such tangency point the normal to the sphere $C^{\perp}$ is perpendicular to the normals of the two spheres C_i and C_j tangent there.*

Proof. The assumption of disjoint interiors (we allow interior to be defined as "exterior" for one sphere) is equivalent to all $\frac{n(n-1)}{2}$ tangency points of the spheres being distinct. For dimension $n = 2$ there is a unique circle through any three distinct points. For $n \ge 3$ the conditions are over-determined, since $n+1$ distinct points already determine a unique $(n-1)$-sphere, and the main issue is existence.

Both assertions of the theorem are invariant under Möbius transformations (which preserve angles), and there exists a Möbius transformation taking a set of $n+1$ mutually tangent $(n-1)$-spheres in $\mathbb{R}^n$ having disjoint interiors to any other such set, cf. Wilker [31, Theorem 3]. Thus it suffices to prove the result for a single such configuration, and we consider the configuration of $n+1$ mutually touching spheres of equal radius whose centers are at the vertices of a regular n-simplex, and tangency points of the spheres are the midpoints of its edges. The first assertion of the theorem holds in this case because there is an $(n-1)$- sphere whose center is at the center of gravity of this simplex, which passes through the midpoints of every edge of the simplex. Indeed the isometries preserving an n-simplex fix the center of gravity and act transitively on the edges. Note that for $n = 2$ the simplex is an equlateral triangle and $C^{\perp}$ is the inscribed circle; however for $n \ge 3$ the sphere $C^{\perp}$ is neither inscribed nor circumscribed about this simplex.

For the second assertion of the proposition, in this configuration the sphere $C^{\perp}$ has each edge of the n-simplex lying in a tangent plane to the sphere; so the normal to $C^{\perp}$ at the midpoint of an edge is perpendicular to that edge. Two spheres C_i and C_j intersect at the midpoint of an edge, and the normal to their tangent planes points along this edge; thus this normal is perpendicular to the normal to $C^{\perp}$ there. ∎

The second assertion in Proposition 3.2 explains why the sphere $C^{\perp}$ is termed " orthogonal." In §5 we give formulas for the curvature and center of $C^{\perp}$.

Definition 3.1. (i) The the *configuration group* $\mathcal{G}_{\mathcal{D}}^{0}$ of the Descartes configuration $\mathcal{D}$ is the

group generated by the operators $\mathfrak{s}_i$ associated to $\mathcal{D}$, i.e.

$$\mathcal{G}^0_{\mathcal{D}} := \langle \mathfrak{s}_1[\mathcal{D}], \dots, \mathfrak{s}_{n+2}[\mathcal{D}] \rangle \subseteq \text{Möb}(n) \ . \tag{3.2}$$

(ii) The *ordered configuration group* $\mathcal{G}_{\mathcal{D}}$ of $\mathcal{D}$ is the group obtained from $\mathcal{G}^0_{\mathcal{D}}$ by adjoining as generators the set of $(n+2)!$ different Möbius transformations that permute the spheres in $\mathcal{D}$.

The configuration group $\mathcal{G}^0_{\mathcal{D}}$ satisfies the relations

$$\mathfrak{s}_i[\mathcal{D}]^2 = I, \quad \text{for} \quad 1 \le i \le n+2 \ , \tag{3.3}$$

but may satisfy additional relations for certain $n \ge 3$. For $n = 3$ it satisfies the extra relations

$$(\mathfrak{s}_i \mathfrak{s}_j)^3 = 1, \qquad \text{when} \qquad i \ne j. \tag{3.4}$$

Definition 3.2. Given an n-dimensional Descartes configuration $\mathcal{D} = \{C_1, C_2, \dots C_{n+2}\}$, the *$n$-dimensional Apollonian sphere ensemble* $\mathcal{P}_{\mathcal{D}}$ is the set of $(n-1)-$spheres,

$$\mathcal{P}_{\mathcal{D}} := \{\mathfrak{s}(C_i) : \mathfrak{s} \in \mathcal{G}^0_{\mathcal{D}} \ \text{ and } C_i \in \mathcal{D}, 1 \le i \le n+2\} \ . \tag{3.5}$$

The *n-dimensional Apollonian cluster ensemble* $\mathbb{D}(\mathcal{P}_{\mathcal{D}})$ associated to $\mathcal{D}$ is the set of Descartes configurations

$$\mathbb{D}(\mathcal{P}_{\mathcal{D}}) = \{\mathfrak{s}(\mathcal{D}) : \mathfrak{s} \in \mathcal{G}_{\mathcal{D}}\}. \tag{3.6}$$

Boyd [3, Theorem 5] observed that the Apollonian sphere ensemble has spheres with disjoint interiors, thus giving an Apollonian sphere packing, if and only if the dimension is 2 or 3. The spheres overlap in higher dimensions, which motivates calling it an "ensemble", rather than a packing. In certain dimensions $n \ge 3$ Boyd [5] constructs configurations of $n+2$ spheres, not all touching, in which an associated group of inversions generates a packing of disjoint spheres. He finds examples up to dimension 9. Later Maxwell [24] classified the possible reflection groups involved.

On the level of Descartes configurations the Apollonian cluster ensembles $\mathbb{D}(\mathcal{P}_{\mathcal{D}})$ above make sense [4] in all dimensions. Furthermore we can recursively calculate the spheres appearing

[4] The set $\mathbb{D}(\mathcal{P}_{\mathcal{D}})$ is contained in the set $\mathbb{D}'(\mathcal{P}_{\mathcal{D}})$ of all Descartes configurations that consist of $n+2$ spheres from the set $\mathcal{P}_{\mathcal{D}}$, but it remains to be decided whether it is the entire set of such configurations.

in such an ensemble using the tree structure enumerating the elemants of $\mathcal{G}_{\mathcal{D}}$. That is, given a Descartes configuration $\mathcal{D}$ we can go to a neighboring configuration $\mathcal{D}'$ in the ensemble by deleting one sphere and adding a new sphere. The coordinates of the new sphere are easily calculated by linear operations, using the following result, derived from the n-dimensional Euclidean Descartes theorem 2.3.

Theorem 3.3. *(i) Given a configuration of $n+1$ tangent spheres $(C_1, C_2, ..., C_{n+1})$ in $\mathbb{R}^n$, with all tangents distinct, there are exactly two spheres, call them C_{n+2} and $C_{n+2}^{'}$, tangent to each of the $n+1$ spheres.*

(ii) Let $\mathcal{D}$ and $\mathcal{D}^{'}$ denote the positively oriented Descartes configurations associated to $(C_1, C_2, ..., C_{n+1})$ with C_{n+2} and $C_{n+2}^{'}$ added, respectively. Then the curvatures a, a' and centers $\mathbf{x}$ and $\mathbf{x}^{'}$ of C_{n+2} and $C_{n+2}^{'}$ are related by

$$a + a^{'} = \frac{2}{n-1}(a_1 + ... + a_{n+1}) \tag{3.7}$$

and

$$a\mathbf{x} + a'\mathbf{x}^{'} = \frac{2}{n-1}(a_1\mathbf{x}_1 + ... + a_{n+1}\mathbf{x}_{n+1}). \tag{3.8}$$

Proof. (i) This result is established in Pedoe [25], who gives references to earlier work, and who observes that in dimensions $n \geq 3$ it is a result in real algebraic geometry, rather than complex algebraic geometry. (In dimension $n \geq 3$ there may be more than two complex circles tangent to such a configuration.

(ii). This follows from the n-dimensional Euclidean Descartes theorem 2.3. If N and N' are the matrices corresponding to $\mathcal{D}$ and $\mathcal{D}'$, then Theorem 2.3 gives

$$N^T\mathbf{Q}_nN = (N')^T\mathbf{Q}_nN' = diag(0, 2I_n),$$

and their first $n+1$ rows agree.

Suppose, more generally, that $\mathbf{y}, \mathbf{z}, \mathbf{g}_1, \ldots, \mathbf{g}_{n+1}$ are row vectors in $\mathbb{R}^{n+1}$, and define the $(n+2) \times (n+1)$ matrices Y and Z by

$$Y^T := [\mathbf{g}_1^T, \ldots, \mathbf{g}_{n+1}^T, \mathbf{y}^T],$$
$$Z^T := [\mathbf{g}_1^T, \ldots, \mathbf{g}_{n+1}^T, \mathbf{z}^T].$$

Then we claim that

$$Y^T \mathbf{Q}_n Y = Z^T \mathbf{Q}_n Z \tag{3.9}$$

if and only if either $\mathbf{y} = \mathbf{z}$ or

$$\mathbf{y} + \mathbf{z} = \frac{2}{n-1}(\mathbf{g}_1 + \cdots + \mathbf{g}_{n+1}). \tag{3.10}$$

To see this, let $\mathbf{f}$ be an arbitrary $n+1$ row vector. Then $\mathbf{f}Y^T\mathbf{Q}_nY\mathbf{f}^T = \mathbf{f}Z^T\mathbf{Q}_nZ\mathbf{f}^T = c$ where c is a constant, so that

$$\begin{aligned} n((\mathbf{f}^T\mathbf{y})^2 + \sum_{i=1}^{n+1}(\mathbf{f}^T\mathbf{g}_i)^2) \quad &- \quad (\mathbf{f}^T\mathbf{y} + \sum_{i=1}^{n+1}\mathbf{f}^T\mathbf{g}_i)^2 = c, \\ n((\mathbf{f}^T\mathbf{z})^2 + \sum_{i=1}^{n+1}(\mathbf{f}^T\mathbf{g}_i)^2) \quad &- \quad (\mathbf{f}^T\mathbf{z} + \sum_{i=1}^{n+1}\mathbf{f}^T\mathbf{g}_i)^2 = c. \end{aligned}$$

That is, both $\mathbf{f}^T\mathbf{y}$ and $\mathbf{f}^T\mathbf{z}$ are solutions of the equation

$$n(x^2 + \sum_{i=1}^{n+1}(\mathbf{f}^T\mathbf{g}_i)^2) - (x + \sum_{i=1}^{n+1}\mathbf{f}^T\mathbf{g}_i)^2 = c.$$

It follows that either $\mathbf{f}^T\mathbf{y} = \mathbf{f}^T\mathbf{z}$ or

$$\mathbf{f}^T(\mathbf{y} + \mathbf{z}) = \frac{2}{n-1}\mathbf{f}^T(\mathbf{g}_1 + \cdots + \mathbf{g}_{n+1}).$$

for all $\mathbf{f} \in \mathbb{R}^{n+1}$. For $n \geq 2$ this is possible if and only if either $\mathbf{y} = \mathbf{z}$ or the equation (3.10) holds. ∎

This theorem shows that starting with the curvatures and centers of a set of $n+2$ mutually tangent spheres, we can step along a sequence of spheres, each tangent to some set of $n+1$ of the preceding spheres, simply by updating the a's and $a\mathbf{x}$'s using this linear recurrence. The special role of dimensions $n = 2$ and $n = 3$ is apparant here, in terms of integrality properties of this recurrence. In two or three dimensions, if we start with integer values of $\mathbf{a}$ and AX then all succeeding values will be integers. This fails to hold in higher dimensions, because then $\frac{2}{n-1}$ is not an integer.

Most results in parts I and II for Apollonian packings have n-dimensional analogues for Apollonian sphere ensembles. By definition the elements of $\mathcal{G}_{\mathcal{D}}^0$ leave the Apollonian sphere

ensemble $\mathcal{P}_{\mathcal{D}}$ invariant. If a Descartes configuration $\mathcal{D}' \in \mathcal{P}_{\mathcal{D}}$ then $\mathcal{P}_{\mathcal{D}} = \mathcal{P}_{\mathcal{D}'}$. The automorphism group $Aut(\mathcal{P}) \subseteq \text{Möb}(n)$ acts sharply transitively on the set $\mathbb{D}(\mathcal{P})$ of ordered Descartes configurations in the Apollonian cluster ensemble $\mathbb{D}(\mathcal{P})$ in $\mathbb{R}^n$. Under Möbius transformations all Apollonian sphere ensembles (resp. cluster ensembles) in $\mathbb{R}^n$ are the same: there exists $\mathfrak{g} \in \text{Möb}(n)$ with $\mathfrak{g}(\mathcal{P}) = \mathcal{P}'$, and $\mathfrak{g}(\mathbb{D}(\mathcal{P})) = \mathbb{D}(\mathcal{P}')$. This follows from Wilker [31, Theorem 3,p. 394].

One can also define Möbius transformations that move between Apollonian ensembles, which have natural geometric meanings. The *inversion operators* $\mathfrak{s}_i^{\perp} := \mathfrak{j}_{C_i}$ are the inversions with respect to the circles C_i, for $1 \le i \le n+2$. The *dual operator* does not generalize to $n \ge 3$ as a Möbius transformation, however.

We next consider the n-dimensional analogue of the Apollonian group. Recall that the *Descartes quadratic form* Q_n is

$$Q_n(\mathbf{x}) := \mathbf{x}^T \mathbf{Q}_n \mathbf{x} = \mathbf{x}^T (I_{n+2} - \frac{1}{n} \mathbf{1}_{n+2} \mathbf{1}_{n+2}^T) \mathbf{x} \tag{3.11}$$

where $\mathbf{1}_{n+2}^T = (1, 1, \ldots, 1)$. This is a rational quadratic form with $\det(Q_n) = -\frac{2}{n}$ (see Lemma 4.3 below) and it has signature $(1, n+1)$. Let $Iso^{\uparrow}(Q_n)$ denote the group

$$Iso^{\uparrow}(Q_n) := \{M \in GL(n+2, \mathbb{R}) : M^T \mathbf{Q}_n M = \mathbf{Q}_n, \text{ and } \mathbf{1}_{n+2}^T M \mathbf{1}_{n+2} > 0\} . \tag{3.12}$$

Theorem 2.3 shows that "curvature-center coordinates" describe an (oriented) Descartes configuration $\mathcal{D}$ in $\hat{\mathbb{R}}^n$ by an $(n+2) \times (n+1)$ matrix $N_{\mathcal{D}}$. Theorem 2.6 of part I generalizes as follows.

Theorem 3.4. *The group $Iso^{\uparrow}(Q_n)$ is sharply transitive on the set $\mathbb{D}_n$ of all ordered (positively oriented) n-dimensional Descartes configurations $\mathcal{D}$.*

This result is analogous to that of Wilker [31, Theorem 3, p.394], and we omit a proof.

The action of $Iso^{\uparrow}(Q_n)$ can be extended to the augmented matrices $\tilde{N}_{\mathcal{D}}$ by left linear multiplications. Similar to the 2-dimensional case, we have

Theorem 3.5. *The actions of $Iso^{\uparrow}(Q_n)$ and Möb(n) on the set $\{\tilde{N}_{\mathcal{D}} : \mathcal{D} \in \mathbb{D}_n\}$ commute with each other.*

The proof is similar to that of Theorem 2.8 of part I, and follows from Theorem 3.1.

Definition 3.3. The (unordered) *n-dimensional Apollonian group* $\mathcal{A}_n^0$ is the group of $(n+2) \times (n+2)$ matrices generated by

$$\mathcal{A}_n^0 = \langle S_1, S_2, \ldots, S_{n+2} \rangle, \tag{3.13}$$

in which

$$S_1 = \left[\begin{array}{c|ccc} -1 & \frac{2}{n-1} & \frac{2}{n-1} & \cdots \quad \frac{2}{n-1} \\ \hline 0 & & & \\ 0 & & & \\ \vdots & & I_{n+1} & \\ 0 & & & \end{array}\right] \tag{3.14}$$

and $S_i = P_{(1i)} S_1 P_{(1i)}$, where $P_{(1i)}$ is the permutation matrix for $(1i)$.

The action of S_1 on a Descartes configuration $\mathcal{D} = \{C_1, C_2, \ldots, C_{n+2}\}$ is to send it to the unique Descartes configuration $\mathcal{D}' = \{C_1', C_2, \ldots, C_{n+2}\}$ having $C_1' \neq C_1$, i.e. $S_1 N_{\mathcal{D}} = N_{\mathcal{D}'}$. It is easy to check that

$$S_i^T Q_n S_i = Q_n \ , \tag{3.15}$$

and $\mathbf{1}^T S_i \mathbf{1} > 0$, so that

$$\mathcal{A}_n^0 \subseteq Iso^{\uparrow}(Q_n) \ .$$

The group $\mathcal{A}_n^0$ preserves all Apollonian cluster ensembles in the sense that if $M \in \mathcal{A}_n^0$ and $\mathcal{D}$ is a Descartes configuration, then

$$M(\mathbb{D}(\mathcal{P}_{\mathcal{D}})) = \mathbb{D}(\mathcal{P}_{\mathcal{D}}) \ . \tag{3.16}$$

As in the two-dimensional case, one can find "integral" n-dimensional Apollonian ensembles all of whose curvature×center coordinates lie in $\mathbb{Z}[\frac{2}{n-2}]$. It remains to study number-theoretic properties of such packings, generalizing §5 and results in [18].

The n-dimensional Apollonian group $\mathcal{A}_n^0$ satisfies different relations than the two-dimensional case. For $n = 3$ its generators satisfy relations associated to the "Hexlet" noted by Soddy [29], [30]:

$$(S_i S_j)^3 = I \quad \text{for } i \neq j. \tag{3.17}$$

The results of Maxwell [24, Table I, p. 91] imply that for $n = 3$ the Apollonian group is a Coxeter group with the defining relations above. As far as we know it is an open problem to determine a complete set of relations for the generators of $\mathcal{A}_n^0$ for $n \geq 4$.

Definition 3.4. The (unordered) *inverse-Apollonian group* $(\mathcal{A}_n^0)^\perp$ on $\mathbb{R}^n$ is the group of $(n+2) \times (n+2)$ matrices

$$(\mathcal{A}_n^0)^\perp = \langle S_1^\perp, S_2^\perp, \dots S_{n+2}^\perp \rangle \tag{3.18}$$

in which

$$S_1^\perp = \left[\begin{array}{c|ccc} -1 & 0 & \dots & 0 \\ \hline 2 & & & \\ 2 & & & \\ \vdots & & I_{n+1} & \\ 2 & & & \end{array}\right] \tag{3.19}$$

and $S_i^\perp = P_{(1i)}^T S_1^\perp P_{(1i)}$, where $P_{(1i)}$ is the permutation matrix for $(1i)$.

The action of $S_1^\perp$ on the Descartes configuration $\mathcal{D} = \{C_1, C_2, \dots, C_{n+2}\}$ is to send it to $\mathcal{D}'' = \{C_1, C_2'', \dots, C_n''\}$ where C_j'' denotes the inversion of C_j in the circle C_1, i.e.

$$S_1^\perp N_{\mathcal{D}} = N_{\mathcal{D}''} \ .$$

One can directly check that

$$(S_i^\perp)^T Q_n S_i^\perp = Q_n \tag{3.20}$$

and $\mathbf{1}^T S_i \mathbf{1} > 0$, so that $(\mathcal{A}_n^0)^\perp \subseteq Iso^\uparrow(Q_n)$.

In dimension $n = 2$ one has the special relation

$$S_i^\perp = S_i^T \qquad \text{for} \qquad 1 \leq i \leq 4,$$

so that $(\mathcal{A}_2^0)^\perp = (\mathcal{A}_2^0)^T$. This symmetry, given by the transpose, no longer holds for $n \geq 3$. We also note that $(\mathcal{A}_n^0)^\perp$ is a group of integer matrices in all dimensions, while $\mathcal{A}_n^0$ is a group of integer matrices only for $n \leq 3$.

Definition 3.5. (i) The *(unordered) n-dimensional super-Apollonian group* $\tilde{\mathcal{A}}_n^0$ is the group generated by $\mathcal{A}^0$ and $(\mathcal{A}_n^0)^{\perp}$, with generators

$$\tilde{\mathcal{A}}_n^0 := \langle S_1, S_2, \ldots, S_{n+2}, S_1^{\perp}, S_2^{\perp}, \ldots, S_{n+2}^{\perp} \rangle \ . \tag{3.21}$$

(ii) The *(ordered) n-dimensional super-Apollonian group* $\tilde{\mathcal{A}}_n$ is obtained by adjoining to $\tilde{\mathcal{A}}_n^0$ the permutation matrices $\{P_\sigma : \ \sigma \in Sym(n+2)\}$.

The super-Apollonian group $\tilde{\mathcal{A}}_n^0$ is contained in the group of automorphs $Aut(Q_n, \mathbb{Z}[\frac{2}{n-2}])$ of Q_n with coefficients in the ring $\mathbb{Z}[\frac{2}{n-2}]$ by (3.15) and (3.20). It is natural ask whether it is a finite index subgroup of $Aut(Q_n, \mathbb{Z}[\frac{2}{n-2}])$ and, if so, to determine its index. It is also an open problem to find a complete set of relations among the generators of $\tilde{\mathcal{A}}_n^0$, for $n \geq 3$.

Finally, one may consider *n-dimensional super-packings*, which we define to be the set of n-dimensional Descartes configurations in the orbit of a single Descartes configuration under the action of the super-Apollonian group. The n-dimensional super-Apollonian group $\tilde{\mathcal{A}}_n$ lies in $Mat_{(n+2)\times(n+2)}(\mathbb{Z}[\frac{2}{n-2}])$. Considering curvatures alone, we can start with a Descartes configuration having curvatures $(0, 0, 1, 1, \ldots, 1)$ and construct a super-packing from it under the action of $\tilde{\mathcal{A}}_n^0$. We conjecture that this super-packing contains Descartes $(n+2)$-tuples similar to all integral Descartes $(n+2)$-tuples.

4. Integral and Rational Apollonian Sphere Ensembles

We now consider integrality and rationality properties for Apollonian sphere ensembles. The Apollonian group has an integral structure in dimensions 2 and 3, and retains an S-integral structure in all dimensions. Here S is a given finite set of primes and a rational number is *S-integral* if its denominator is divisible only by powers of primes in S.

Definition 4.1. An Apollonian sphere ensemble is *S-integral* if the curvature of every sphere in the ensemble is S-integral.

The recurrence relation between curvatures of two adjacent Descartes configurations, given in Theorem 3.3 as

$$a_1 + a_1^{'} = \frac{2}{n-1}(a_2 + ... + a_{n+2}).$$

shows that S-integrality is preserved under this operation, for any S containing all primes dividing the denominator of $\frac{2}{n-1}$. More generally all entries of all matrices in the Apollonian group are S-integral, where S consists of the primes dividing the denominator of $\frac{2}{n-1}$. The same property persists for the super-Apollonian group in n-dimensions, since its extra generators are all integral matrices.

Theorem 4.1. *In each dimension $n \geq 2$ there exists an S-integral Apollonian sphere ensembles with S being the set of primes dividing $n-1$ if n is even and dividing $\frac{n-1}{2}$ if n is odd.*

Proof. It suffices to show that the Descartes equation

$$\mathbf{a}^T \mathbf{Q}_n \mathbf{a} = 0 \tag{4.22}$$

has a non-zero S-integral solution $\mathbf{a}$ for each $n \geq 2$. There is such a configuration $\mathcal{D}$ which is not only S-integral, but integral, with curvatures $(0, 0, 1, 1, ..., 1)$. It consists of two parallel hyperplanes separated by distance 2 together with n unit spheres whose centers comprise the vertices an $(n-1)$-dimensional simplex in a hyperplane parallel to the two hyperplanes in the configuration, and lying midway between them.

The other Descartes configurations in the Apollonian sphere ensemble and super-Apollonian sphere ensemble generated by this configuration are S-integral, where S is the set of primes dividing the denominator of $\frac{2}{n-1}$, since they have associated matrices $GN_{\mathcal{D}}$ for some G in the Apollonian group. ∎

The Apollonian group and super-Apollonian group act on the set of S-integral Apollonian packings. For dimension $n = 2$ various number-theoretic questions related to the integers appearing in such packings were studied in [18]; in dimensions $n \geq 3$ the corresponding problems all remain open.

Next we consider S-integrality involving the sphere centers as well.

Definition 4.2. (i) An oriented Descartes configuration is *strongly S-integral* if if its associated matrix $N_{\mathcal{D}}$ has all entries S-integers.

(ii) An oriented Descartes configuration $\mathcal{D}$ is *super-strongly S-integral* if its associated augmented matrix $\tilde{N}_{\mathcal{D}}$ is S-integral.

We extend these definitions to Apollonian packings.

Definition 4.3. (i) An Apollonian sphere ensemble is *strongly S-integral* if every Descartes configuration $\mathcal{D}$ in the packing has S-integral matrix $N_{\mathcal{D}}$.

(ii) An Apollonian sphere ensemble is *super-strongly S-integral* if every Descartes configuration $\mathcal{D}$ in the packing has S-integral augmented matrix $\tilde{N}_{\mathcal{D}}$.

If a single Descartes configuration is strongly (resp. super-strongly) S-integral, then the Apollonian packing it generates is strongly (resp. super-strongly) S'-integral, where S' consists of S together with all primes dividing the denominator of $\frac{2}{n-1}$. For this reason it suffices to consider S-integrality for individual Descartes configurations.

For dimension $n = 2$, in part II we showed that strongly S-integral Descartes configurations existed, with $S= 1$, and that strongly integral Apollonian packings also existed. We also completely classified them, in the sense that we showed [17, Theorem 3.5] that under the action of the super-Apollonian group, the set of all strongly integral Descartes configurations formed exactly eight orbits ([17, Theorem 3.5]) .

In dimension $n = 2$, every strongly integral Descartes configuration is actually super-strongly integral! To show this it suffices to consider one Descartes configuration in each of the eight orbits above and verify that it has an integral augmented matrix $\tilde{N}_{\mathcal{D}}$, because the super-strong integrality property is preserved under the action of the super-Apollonian group ($n = 2$). For example, the strongly integral Descartes matrix

$$N_{\mathcal{D}} = \begin{bmatrix} -1 & 0 & 0 \\ 2 & -1 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 2 \end{bmatrix},$$

extends to the augmented Descartes matrix

$$\tilde{N}_{\mathcal{D}} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 2 & -1 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 2 & 1 \end{bmatrix}.$$

Similar integrality formulae hold for the other seven cases.

The existence of stongly S-integral Descartes configurations for some S is the same as the existence of Descartes configurations $\mathcal{D}$ having a rational augmented matrix $\tilde{N}_{\mathcal{D}}$.

Definition 4.4. A Descartes configuration $\mathcal{D}$ is *rational* if and only if its non-augmented matrix $N_{\mathcal{D}}$ is a rational matrix.

In this definition we could, alternatively, require that the augmented matrix $\tilde{N}_{\mathcal{D}}$ be rational, because the matrix $N_{\mathcal{D}}$ is rational if and only if the augmented matrix $\tilde{N}_{\mathcal{D}}$ is rational. Indeed, the last column of $\tilde{N}_{\mathcal{D}}$ is calculated from the entries of $N_{\mathcal{D}}$ using inversion in the unit circle, and this map sends the set of spheres with rational centers and rational curvatures into itself.

According to the augmented Euclidean Descartes theorem 2.4, rational Descartes configurations occur exactly in those dimensions n in which there exists an invertible rational matrix $\tilde{N}$ such that

$$\tilde{N}^T \mathbf{Q}_n \tilde{N} = \tilde{\mathbf{Q}}_n := \begin{bmatrix} 0 & 0 & -4 \\ 0 & 2I_n & 0 \\ -4 & 0 & 0 \end{bmatrix}, \tag{4.23}$$

that is, the quadratic form $\mathbf{Q}_n$ is rationally equivalent to the form $\tilde{\mathbf{Q}}_n$. We use this fact to show that in most higher dimensions rational Descartes configurations do not exist.

Theorem 4.2. *A necessary condition for a rational Descartes configuration to exist in dimension n is that $n = 2k^2$ or $(2k-1)^2$ for some positive integer k.*

To establish this result, we use the following lemma.

Lemma 4.3. *Given a Descartes configuration $\mathcal{D}$ in $\mathbb{R}^n$ its associated augmented matrix $\tilde{N}_{\mathcal{D}}$ has determinant satisfying*

$$\det(\tilde{N}_{\mathcal{D}})^2 = n2^{n+3}. \tag{4.24}$$

Proof. This follows from taking determinants in (2.38), since the right side has determinant -2^{n+4} while the left side has determinant $\det(\tilde{N}_{\mathcal{D}})^2 \det(\mathbf{Q}_n)$ and

$$\det(\mathbf{Q}_n) = -\frac{2}{n}. \tag{4.25}$$

To verify this last statement, we apply the following row operations to the matrix $\mathbf{Q}_n$. Add rows 2 through $n+2$ to the first row, to get a new first row that has all entries $-\frac{2}{n}$. Then add this row multiplied by $-\frac{1}{2}$ to each of the other rows. Aside from the first row, the first column is zero, and the lower right $(n+1) \times (n+1)$ matrix is the identity. But this matrix obviously has determinant $-\frac{2}{n}$. ∎

Proof of Theorem 4.2. A necessary condition for the existence of a Descartes configuration $\mathcal{D}$ whose augmented matrix $\tilde{N}_{\mathcal{D}}$ has rational entries is that $\det(\tilde{N}_{\mathcal{D}})$ be rational. This requires that $n2^{n+3}$ be the square of a rational number. By Lemma 4.3, this holds for even n if and only if n is twice a square, and for odd n if and only if n is an (odd) square. ∎

We now prove the converse to Theorem 4.2.

Theorem 4.4. *In each dimension $n \geq 2$ which has n of the form $n = 2k^2$ or $(2k-1)^2$ for some positive integer k, there exists a rational Descartes configuration.*

This theorem is proved using the well-developed theory of equivalence of rational quadratic forms, cf. Cassels [6] or Conway [8]. We write $\mathbf{Q} \simeq_{\mathbb{Q}} \mathbf{Q}'$ to mean that the (rational) quadratic form $\mathbf{Q}$ is rationally equivalent to $\mathbf{Q}'$. To apply the decision procedure, we first diagonalize $\mathbf{Q}_n$ over the rationals, which we do for all $n \geq 2$.

Lemma 4.5. *For each $n \geq 2$, the Descartes quadratic form $\mathbf{Q}_n = I_{n+2} - \frac{1}{n}\mathbf{1}_{n+2}\mathbf{1}_{n+2}^T$ has*

$$\mathbf{Q}_n \ \simeq_{\mathbb{Q}} \ diag(\frac{n-1}{n}, \frac{n-2}{n-1}, \cdots, \frac{2}{3}, 2, 2, 2, -2). \tag{4.26}$$

Proof. We diagonalize the quadratic form as in Conway [8, pp. 92–94]. Set

$$M^{(n+2)} := \mathbf{Q}_n = (x_0 + y_0)I_{n+2} - y_0\mathbf{1}_{n+2}\mathbf{1}_{n+2}^T\,,$$

where $x_0 = \frac{n-1}{n}$, $y_0 = \frac{1}{n}$. At the j-th stage of reduction we will have

$$\mathbf{Q}_n \simeq_{\mathbb{Q}} \mathrm{diag}(d_1, d_2, \ldots, d_j, M^{(n+2-j)})\,,$$

where

$$M^{(n+2-j)} = (x_j + y_j)I_{n+2-j} - y_j\mathbf{1}_{n+2-j}\mathbf{1}_{n+2-j}^T \tag{4.27}$$

for certain x_j, y_j. The reduction step is

$$(W^{(j)})^T M^{(n+2-j)} W^{(j)} = \mathrm{diag}(d_{j+1}, M^{(n+1-j)})\,. \tag{4.28}$$

To specify $W^{(j)}$ we first let $W_m(\alpha)$ be the $m \times m$ real matrix

$$W_m(\alpha) = \begin{bmatrix} 1 & \alpha \cdots \alpha \\ \mathbf{0} & I_{m-1} \end{bmatrix}.$$

and we set

$$W^{(j)} := W_{m+2-j}\left(\frac{y_j}{x_j}\right). \tag{4.29}$$

Substituting this in (4.28), its left side yields a matrix with the form of the right side with

$$d_{j+1} = x_j\,,$$

and with x_{j+1}, y_{j+1} given by the recursion

$$\begin{aligned} y_{j+1} &= y_j + \frac{y_j^2}{x_j}\,, & (4.30)\\ x_{j+1} + y_{j+1} &= x_j - \frac{y_j^2}{x_j}\,. & (4.31) \end{aligned}$$

Solving this recursion, by induction on j, one obtains

$$\begin{aligned} x_j &= \frac{n-j-1}{n-j}\,, \qquad 0 \le j \le n-2,\\ y_j &= \frac{1}{n-j}\,, \qquad 0 \le j \le n-2\,. \end{aligned}$$

This yields the diagonal elements

$$d_j = \frac{n-j-1}{n-j}, \quad 1 \le j \le n-3\,, \tag{4.32}$$

with

$$\mathbf{Q}_n \simeq_{\mathbb{Q}} \ \mathrm{diag}(\frac{n-1}{n}, \ldots, \frac{2}{3}, d_2, M^{(4)})\,.$$

We find $d_2 = x_3 = \frac{2}{3}$ and

$$M^{(4)} = (x_{n-2} + y_{n-2})I_4 - y_{n-2}\mathbf{1}_4\mathbf{1}_4^T = \frac{1}{2}\begin{bmatrix} 1 & -1 & -1 & -1\\ -1 & 1 & -1 & -1\\ -1 & -1 & 1 & -1\\ -1 & -1 & -1 & 1 \end{bmatrix} = \mathbf{Q}_2\,.$$

For the final step in the reduction we use

$$W^T\,(\mathbf{Q}_2)\,W = \mathrm{diag}(2,2,2,-2) \tag{4.33}$$

with

$$W = \begin{bmatrix} 1 & -1 & -1 & 1\\ -1 & 1 & -1 & 1\\ -1 & -1 & 1 & 1\\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

This completes the reduction. ∎

Proof of Theorem 4.4. The theorem is equivalent to proving that if $n = 2k^2$ and $n = (2k-1)^2$ then

$$\mathbf{Q}_n \simeq_{\mathbb{Q}} \tilde{\mathbf{Q}}_n := \begin{bmatrix} 0 & 0 & -4 \\ 0 & 2I_n & 0 \\ -4 & 0 & 0 \end{bmatrix}.$$

We begin by noting the rational equivalence

$$\tilde{\mathbf{Q}}_n \simeq_{\mathbb{Q}} \mathrm{diag}(2, 2, \cdots, 2, -2) = \mathrm{diag}(2I_{n+1}, -2) \tag{4.34}$$

via the matrix

$$W_0 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 2I_n & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Thus the theorem is equivalent to showing that $\mathbf{Q}_n$ is rationally equivalent to $\mathrm{diag}(2, 2, 2, ..., -2)$. Lemma 4.5 gives

$$\begin{aligned} \mathbf{Q}_n &\simeq_{\mathbb{Q}} (\frac{n-1}{n}, \frac{n-2}{n-1}, \dots, \frac{3}{2}, 2, 2, 2, -2), \\ &\simeq_{\mathbb{Q}} (n(n-1), (n-1)(n-2), \dots, 3 \cdot 2, 2, 2, 2, -2), \end{aligned} \tag{4.35}$$

using at the last step a conjugacy by $W = \mathrm{diag}(n, n-1, \dots, 2, 1, 1, 1, 1)$.

The Hasse-Minkowski theorem says that two rational quadratic forms of the same dimension are equivalent if and only they have the same signature, the ratio of their determinants is a nonzero square, and they are p-adically equivalent for all primes p, cf. Conway [8, p. 96ff]. Lemma 4.5 shows that the signatures of $\mathbf{Q}_n$ and $\mathrm{diag}(2, 2, 2, ..., 2, -2)$ agree, and the hypothesis $n = 2k^2$ or $n = (2k-1)^2$ is exactly the condition that the ratio of their determinants is a square of a rational, and it remains to check the p-adic invariants.

The p-adic invariants $\sigma_p(\mathbf{Q})$ are defined (mod 8), and for a diagonal form $\mathbf{Q} = \mathrm{diag}(d_1, d_2, \dots, d_n)$, one has

$$\sigma_p(\mathbf{Q}) \equiv \sum_{j=1}^{n} \sigma_p(d_j) \pmod 8. \tag{4.36}$$

We recall formulas for $\sigma_p(d)$ when $d \in \mathbb{Z}$, cf. Conway [8, pp. 94–96]. Writer $d = bp^l$ with $(b, p) = 1$. For $p \geq 3$, and an even power $l = 2j$,

$$\sigma_p(d) \equiv p^{2j} \equiv 1 \pmod 8, \tag{4.37}$$

while for an odd power $l = 2j+1$,

$$\sigma_p(d) \equiv \begin{cases} p \pmod 8 & \text{if } \left(\frac{b}{p}\right) = 1\,, \\ p+4 \pmod 8 & \text{if } \left(\frac{b}{p}\right) = -1\,. \end{cases} \tag{4.38}$$

If $p = 2$ then for an even power $l = 2j$,

$$\sigma_2(d) \equiv b \pmod 8\,, \tag{4.39}$$

while for an odd power$l = 2j+1$,

$$\sigma_2(d) \equiv \begin{cases} b & \text{if } b \equiv \pm 1 \pmod 8\,, \\ b+4 & \text{if } b \equiv \pm 3 \pmod 8\,. \end{cases} \tag{4.40}$$

Now (4.35) gives

$$\sigma_p(\mathbf{Q}_n) \equiv \sum_{j=0}^{n-3} \sigma_p((n-j)(n-j-1)) + 3\sigma_p(2) + \sigma_p(-2) \pmod 8$$

while (4.34) gives

$$\sigma_p(\tilde{\mathbf{Q}}_n) \equiv \sum_{j=0}^{n-3} \sigma_p(2) + 3\sigma_p(2) + \sigma_p(-2) \pmod 8\,.$$

To show equality of these, it suffices to show that for all p,

$$\sum_{j=0}^{n-3} \sigma_p(2) \equiv \sum_{j=0}^{n-3} \sigma_p((n-j)(n-j-1)) \pmod 8 \tag{4.41}$$

holds whenever $n = 2k^2$ or $n = (2k-1)^2$.

Consider first the case that $p \geq 3$ is odd. Then each $\sigma_p(2) = 1$, so

$$\sum_{j=0}^{n-3} \sigma_p(2) \equiv n-2 \pmod 8\,. \tag{4.42}$$

Now if $p \nmid (n-j)(n-j-1)$ then $\sigma_p((n-j)(n-j-1)) = 1$. The terms divisible by p occur in blocks of two consecutive terms, and we claim that if p divides j then

$$\sigma_p((j+1)j) + \sigma_p(j(j-1)) \equiv 2 \pmod 8. \tag{4.43}$$

Suppose $j = bp^l$, with where $(b,p) = 1$ and $l \geq 1$. If l is even, both terms on the left side of (4.43) are 1 (mod 8) by (4.37), while if K is odd, then if $p \equiv 1 \pmod 4$, then $\left(\frac{-1}{p}\right) = 1$, so

the two terms both have values p (resp. $p+4$) according as $\left(\frac{b}{p}\right) = 1$ (resp. -1), and their sum is $2p \equiv 2 \pmod 8$. If $p \equiv 3 \pmod 4$, then $\left(\frac{-1}{p}\right) = -1$, so exactly one of $\left(\frac{\pm b}{p}\right)$ takes the value -1, and the two terms add up to $2p+4 \equiv 2 \pmod 8$. Thus (4.43) follows. Thus adding up the right side of (4.41) and grouping terms divisible by p in consecutive pairs gives

$$\sum_{j=0}^{n-3} \sigma_p((n-j)(n-j-1)) \equiv \sum_{j=0}^{n-3} 1 \equiv n-2 \,(\mathrm{mod} 8)\,. \tag{4.44}$$

There remains an exceptional case where $p|n$, in which case $n(n-1)$ is divisible by p and is an un-paired term. Since $n = 2k^2$ or $(2k-1)^2$, thus $p^l \| n$ with l even, hence $\sigma_p(n(n-1)) = 1$ in this case, and (4.44) holds. This establishes (4.41) for $p \geq 3$.

Now consider the case $p = 2$. Certainly $\sigma_2(2) = 1$ so (4.42) holds. We claim that

$$\sigma_2((2j+1)2j) + \sigma_2(2j(2j-1)) \equiv 0 \pmod 8\,. \tag{4.45}$$

Write $2j = 2^l b$ with b odd, and by checking all possible cases using (4.39) and (4.40), one verifies (4.45). Suppose $n = 2k^2$. Then in the right side of (4.41) all terms pair except the first and last, and (4.45) yields

$$\begin{aligned}\sum_{j=0}^{n-3} \sigma_2((n-j(n-j-1)) &\equiv \sigma_2(n(n-1)) + \sigma_2(3\cdot 2) \\ &= \begin{cases} -1+-1 & \text{if} \quad k \equiv 0 \pmod 2\,, \\ \ \ 1+-1 & \text{if} \quad k \equiv 1 \pmod 2 \end{cases} \\ &= n-2 \pmod 8\,,\end{aligned}$$

so (4.41) holds. If $n = (2k-1)^2 \equiv 1 \pmod 8$ then all term pair except the last term, and (4.45) yields

$$\sum_{j=0}^{n-3} \sigma_2((n-j)(n-j-1)) = \sigma_2(3\cdot 2) \equiv -1 \pmod 8\,,$$

so (4.41) holds in this case. ∎

In the dimensions covered by Theorem 4.4, rational Descartes configurations exist. Therefore there exist finite sets S for which S-integral configurations exist. As long as such an S is enlarged to include all prime divisors of $n-1$, all configurations in the Apollonian cluster ensemble generated by such a configuration will also be S-integral. One can then raise the question of classifying such ensembles; this appears difficult.

Theorem 4.4 establishes the existence of rational Descartes configurations in the given dimensions, but does not give a bound for the denominators of the rationals appearing in these configurations, i.e. an explicit value for S. As far as we know, it could be that in dimensions $n = 2k^2$ and $(2k+1)^2$ there exist super-strongly integral Descartes configurations, i.e. one could take $S = 1$. (Note that, if such configurations exist for $n > 2$, the Apollonian packing containing them would not inherit the super-strong integrality property.) We leave this as an open problem; results on it should be attainable using the theory of integral quadratic forms.

5. Duality Operator

In two dimensions we studied in part II a duality operation $\mathfrak{d}$ based on orthogonal spheres. This operator had an analogue operator D which was contained in the normalizer of the super-Apollonian group.

The duality operation based on orthogonal spheres generalizes to higher dimensions as follows. Given $n+1$ mutually tangent spheres in n dimensions, there is a unique sphere through their points of tangency, and this sphere is orthogonal to each of the given $n+1$ spheres, see Proposition 3.2. Thus, given a Descartes configuration of $n+2$ spheres C_i, we get a system of $n+2$ "orthogonal" spheres

$$\mathcal{D}^{\perp} := \{C_1^{\perp}, \ldots, C_{n+2}^{\perp}\},$$

where $C_i^{\perp}$ is associated to the $n+1$ spheres obtained by deleting C_i. When $n = 2$ the new spheres are mutually tangent and give a new Descartes configuration; this gives the "duality" operation D studied in parts I and II. For $n \geq 3$, however, the spheres are not mutually tangent. In fact for all n their curvatures satisfy a relation similar in form to the original (two-dimensional) Descartes relation, namely

$$\sum_{i=1}^{n+2} q_i^2 = \frac{1}{2}(\sum_{i=1}^{n+2} q_i)^2, \tag{5.1}$$

and not the Soddy-Gossett relation (2.1). (We omit a proof of this formula.) In particular, for $n \geq 3$ given a Descartes configuration $\mathcal{D}$, the set $\mathcal{D}^{\perp} := \{C_1^{\perp}, \ldots, C_{n+2}^{\perp}\}$ of orthogonal spheres is *not* a Descartes configuration, and the duality operation is *not* in $Iso^{\uparrow}(Q_n)$.

The question arises, are these $n+2$ "orthogonal" spheres in any special relation to one another? We answer this in terms of an inversive invariant of two arbitrary (not necessarily tangent) oriented spheres.

Definition 5.1. (i) The *separation* between two oriented spheres C_1 and C_2 with finite radii r_1 and r_2, and with centers distance d apart, as

$$\Delta(C_1, C_2) := \frac{d^2 - r_1^2 - r_2^2}{2r_1 r_2}. \tag{5.2}$$

provided both spheres are inwardly oriented or outwardly oriented, and is otherwise the negative of the right side of this formula.

(ii) The *separation* of an oriented sphere C_1 of finite radius r_1 and an oriented hyperplane C_2 is

$$\Delta(C_1, C_2) := \frac{d}{r_1}. \tag{5.3}$$

where d is the (signed) distance from the center $\mathbf{a}_1$ of C_1 to C_2, measured so that $d \geq 0$ if $\mathbf{a}_1$ is not in the interior of C_2 and C_1 is inwardly oriented, or if $\mathbf{a}_1$ is in the interior of C_2 and C_1 is outwardly oriented, and $d < 0$ otherwise.

(iii) The *separation* between two oriented hyperplanes C_1 and C_2 is

$$\Delta(C_1, C_2) := -\cos\theta. \tag{5.4}$$

where θ is the dihedral angle between the designated normals at a point of intersection.

The separation of two spheres is an inversive invariant ; that is,

$$\Delta(\mathfrak{g}(C_1), \mathfrak{g}(C_2)) = \Delta(C_1, C_2), \tag{5.5}$$

holds for any Möbius transformation $\mathfrak{g}$. This concept appears in Boyd [3], who introduced the term separation for it, but the concept [5] was used earlier by Mauldon [23] in 1962, who used the term *inclination* to mean the negative of $\Delta(C_1, C_2)$, and showed it was an inversive invariant.

[5] The idea of considering such an inversive invariant traces back to work of Clifford [7] in 1868 and of Darboux [13] in 1872. However, neither Clifford's nor Darboux' definition was precisely $\Delta(C_1, C_2)$. Clifford defines the *power of two spheres* to be the square distance of their centers less the sum of the squares of their radii, i.e., $d^2 - r_1^2 - r_2^2$, and Darboux also uses the same quantity, [13, p.350].

The separation $\Delta(C_1, C_2)$ of two spheres can be expressed in terms of their augmented curvature-center coordinates as

$$\begin{aligned}\Delta(C_1, C_2) &= \frac{1}{2}\tilde{\mathbf{w}}(C_1)^T \mathbf{K}_n(1)\tilde{\mathbf{w}}(C_2) \\ &= -\frac{1}{2}(\bar{a}(C_1)a(C_2) + a(C_1)\bar{a}(C_2)) + a(C_1)a(C_2)\sum_{j=1}^{n} x_j(C_1)x_j(C_2), \quad (5.6)\end{aligned}$$

where $\mathbf{K}_n(1)$ is given in (2.39). This formula can be proved by a simple algebraic calculation, cf. [20]. Using it, one can check that for two tangent spheres C_1 and C_2, $\Delta(C_1, C_2) = 1$, if (1) C_1 and C_2 are externally tangent, and both are inwardly oriented or outwardly oriented, or (2) C_1 and C_2 are internally tangent and one is inwardly oriented, the other is outwardly oriented. In all other cases two tangent spheres have $\Delta(C_1, C_2) = -1$, and orthogonal spheres are those with $\Delta(C_1, C_2) = 0$.

From Proposition 3.2 one obtains

$$\Delta(C^{\perp}, C_j) = 0 \qquad \text{for} \qquad 1 \le j \le n+1, \tag{5.7}$$

using (5.4), and these relations determine $C^{\perp}$ up to orientation. It can also be shown that if a set of tangent spheres $\{C_1, ..., C_{n+1}\}$ have oriented curvatures $\mathbf{a}_{n+1} = (a_1, ..., a_{n+1})$, and centers $\mathbf{x}_j$, then for either orientation the orthogonal sphere $C^{\perp}$ has oriented curvature q satisfying

$$q^2 = \frac{1}{2}\left(\frac{1}{n-1}(\sum_{j=1}^{n+1} a_j)^2 - \sum_{j=1}^{n+1} a_j^2\right), \tag{5.8}$$

and (oriented) center $\mathbf{x}$ satisfying

$$q\mathbf{x} = -\mathbf{a}_{n+1}(\frac{1}{2}\mathbf{Q}_{n-1})\mathbf{C}, \tag{5.9}$$

in which $\mathbf{C}$ is an $(n+1) \times n$ matrix whose j-th row is $a_j\mathbf{x}_j$, and $\mathbf{Q}_{n-1}$ is the Descartes form.

An oriented Descartes configuration in $\mathbb{R}^n$ is characterized in terms of separation as a set of $n+2$ oriented spheres each pair of which has $\Delta(C_i, C_j) = 1$, when $i \neq j$. Thus such a configuration has the following property.

Definition 5.2. A collection of oriented spheres is *equiseparated* if all values $\Delta(C_j, C_k)$ with $j \neq k$ are equal.

The equiseparation property can also be viewed as an *equiangularity* property, because for two oriented circles that intersect or touch one has

$$\Delta(C_1, C_2) = -\cos\theta, \tag{5.10}$$

where θ is the angle between oriented normals at a point of intersection of the two circles. We now show the duality operation preserves equiseparability in all dimensions; a further generalization appears in [20].

Theorem 5.1 (Equiseparation Theorem) *Given an oriented Descartes configuration* $\mathcal{D} = (C_1, C_2, ..., C_{n+2})$ *in* $\mathbb{R}^n$*, if the dual spheres are properly oriented then the (oriented) dual configuration* $(C_1^\perp, C_2^\perp, ..., C_{n+2}^\perp)$ *is equiseparated, with*

$$\Delta(C_j^\perp, C_k^\perp) = \frac{1}{n-1} \qquad \textit{if} \qquad j \neq k. \tag{5.11}$$

Proof. In this result, the orientation assigned to the dual spheres in the theorem depends on all $n+2$ spheres in the Descartes configuration, and the orientation of $C_j^\perp$ cannot be consistently assigned from the $n+1$ oriented spheres $\{C_i : i \neq j\}$ alone. If all $n+2$ spheres C_j are inwardly oriented, then $n+1$ of the spheres $C_j^\perp$ will be inwardly oriented and one outwardly oriented, the last being the one of largest radius. If all but one of the $n+2$ spheres are inwardly oriented, and one outwardly oriented, then all $n+2$ spheres $C_j^\perp$ will be inwardly oriented.

Since the result is invariant under inversion, it suffices to prove it for a single Descartes configuration. We consider the special oriented Descartes configuration where the curvatures are (0,0,1,1, ... , 1). Here we have two parallel planes, which we take as $x_1 = \pm 1$, and n unit spheres, all with centers on the plane $x_1 = 0$. Their centers form a regular simplex in this plane. We may take one of these centers at $(0, \xi, 0, 0, \dots)$ where $\xi^2 = 2(n-1)/n$. Consider the "orthogonal" spheres that pass through the point $T = (1, \xi, 0, 0, \dots, 0)$. There are n such, and all but one of them is a plane containing T, $(-1, \xi, 0, 0 \dots, 0)$, and the centers of all but one of the original unit spheres. Since these centers are the vertices of a regular simplex, these $n-1$ "orthogonal" planes are equiangular satisfying (5.10), where θ is the angle between the normals of two facets of a regular n-simplex. It follows that these orthogonal planes satisfy (5.11). The final "orthogonal" sphere through T is orthogonal to the plane $x_1 = 1$ and all the n original unit spheres. Its center is thus $(1, 0, 0, \dots)$ and its radius is ξ. Hence it is also

equiangular with the $n-1$ "orthogonal" planes, with $\cos\theta = -\frac{1}{n-1}$. (These angles are all eaqual to the one formed by connecting the vertices of a regular simplex to its center, i.e. the angle in a triangle of sides ξ, ξ and 2.) Finally, the last two "orthogonal" spheres meet at the same angle in the plane $x_1 = 0$. ∎

6. Loxodromic Sequences of Tangent Spheres

As our final topic we turn to a concept studied by Coxeter [11], concerning loxodromic sequences of tangent spheres. Coxeter defines a *loxodromic sequence* of spheres as being a sequence where each successive set of $n+2$ spheres are mutually tangent. Thus if the sequence of curvatures is

$$\ldots\ a_{-2},\ a_{-1},\ a_0,\ a_1,\ a_2,\ \ldots$$

then each successive set of $n+2$ spheres satisfies the Soddy relation $\mathbf{a}^T Q_n \mathbf{a} = 0$, so that by Theorem 2.3 they also satisfy a linear recurrence

$$a_i + a_{i+n+2} = \frac{2}{n-1}(a_{i+1} + \ldots + a_{i+n+1}). \tag{6.1}$$

For $n = 3$, Coxeter proves in [12] that the sequence

$$\ldots \Delta_{0,-2}\ ,\quad \Delta_{0,-1}\ ,\quad \Delta_{0,0}\ ,\quad \Delta_{0,1}\ ,\quad \Delta_{0,2}\ ,\ \ldots$$

where $\Delta_{ij} = \Delta(C_i, C_j)$ is the separation between circles C_i and C_j, also satisfies the linear recurrence (6.1). This is slightly unexpected, since while Δ is dimensionless, it involves the square of the distance between the centers, while the other quantities that obey the recurrence are the curvatures a_i and $a_i \mathbf{x}_i$, which is the product of curvatures and centers. We prove a slightly more general result.

Theorem 6.1 (Separation Formula) *Given $n+1$ mutually tangent spheres $C_1, \ldots C_{n+1}$ with disjoint interiors, let C_0 and C_{n+2} be the two spheres that are tangent to each of these. Let C' be an arbitrary sphere, and let Δ_i be the separation between C_i and C'. Then*

$$\Delta_0 + \Delta_{n+2} = \frac{2}{n-1}(\Delta_1 + \ldots + \Delta_{n+1}). \tag{6.2}$$

Proof. Without loss of generality, we may assume that all the curvatures qare non-zero, because the separation is invariant under inversions. Furthermore, we may assume that the center of C_0 is the origin, and that its curvature a_0 is not zero. Let the curvatures of $C_1, \ldots, C_{n+2}, C'$ be $a_1, \ldots, a_{n+2}, a'$, and let their centers be $\mathbf{x}_1, \ldots, \mathbf{x}_{n+2}, \mathbf{x}'$. Then

$$\begin{aligned}\Delta_i &= \frac{1}{2}\left(a_i a'|\mathbf{x}_i - \mathbf{x}'|^2 - \frac{a_i}{a'} - \frac{a'}{a_i}\right) \\ &= \frac{1}{2}\left(a'\left(a_i|\mathbf{x}_i|^2 - \frac{1}{a_i}\right) + a_i\left(a'|\mathbf{x}'|^2 - \frac{1}{a'}\right) - 2a'a_i\mathbf{x}_i^T\mathbf{x}'\right).\end{aligned}$$

We know that each of the sequences $(a_0, \ldots, a_{n+2})$, $(a_0\mathbf{x}_0, \ldots, a_{n+2}\mathbf{x}_{n+2})$ satisfy the linear recurrence, so it is sufficient to prove that the quantities

$$t_i = a_i|\mathbf{x}_i|^2 - \frac{1}{a_i}$$

do also. Notice that the dependence on a' and $\mathbf{x}'$ has been eliminated. Thus $t_0 = -1/a_0$, and for $i > 0$,

$$\begin{aligned}t_i &= a_i(\frac{1}{a_0} + \frac{1}{a_i})^2 - \frac{1}{a_i} \\ &= \frac{a_i + 2a_0}{a_0^2}.\end{aligned}$$

Therefore the vector $\mathbf{t}^T = (t_0, t_1, \ldots, t_{n+1})$ is given by

$$\mathbf{t} = \frac{1}{a_0^2}(\mathbf{a} + 2a_0(\mathbf{1}_{n+2} - 2\mathbf{e}_0)),$$

where $\mathbf{a}^T = (a_0, \ldots, a_{n+1})$, ${\mathbf{e}_0}^T = (1, 0, \ldots, 0)$. Simple algebra now verifies that $\mathbf{t}^T Q_n \mathbf{t} = 0$. Hence the t's also satisfy the linear recurrence (6.1). ∎

As Coxeter (1997,[12]) points out, if we are in two dimensions and $S' = S_0$, then $(\Delta_0, \Delta_1, \Delta_2, \Delta_3) = (-1, 1, 1, 1)$ and the sequence extends uniquely to

$$-1, 1, 1, 1, 7, 17, 49, 145, 415, 1201, 3473, 10033, 28999, 83809, 242209, 700001, 2023039, \ldots$$

(sequence A045821 in Sloane [27]). In three dimensions the corresponding (unique) sequence is

$$-1, 1, 1, 1, 1, 5, 7, 13, 25, 49, 89, 169, 319, 601, 1129, 2129, 4009, \ldots$$

(this is sequence A027674 [27]).

7. Conclusion

This series of papers studied various group theoretic problems raised by geometrically defined groups associated to Apollonian packings. It obtained fairly complete answers when the dimension $n = 2$, but left many open problems, particularly in dimensions $n \geq 3$.

In the case of the Apollonian group and super-Apollonian groups in n-dimensions there remain a number of open questions. One is the problem of determining their exact normalizers. Another is that of establishing the index of the super-Apollonian group in the automorphism group $Aut(Q_n,\ \mathbb{Z}[\frac{2}{n-1}]\ )$. It is also an open problem to obtain finite presentations for these groups, for $n \geq 3$. We noted that for $n \geq 4$ the Apollonian group no longer produced a sphere-packing. Is this related to the non-integral nature of the matrices in the Apollonian group, for $n \geq 4$? Can one define a discontinuous action of this group on a real space $\times$ p-adic spaces corresponding to primes dividing the denominator of $\frac{2}{n-1}$? There also remain open questions connected with the fact that these groups are integral over the ring $\mathbb{Z}[\frac{2}{n-1}]$. Various number-theoretic questions in this direction are raised in the concluding section of the companion paper [18]. Finally, in §4 we showed that S-integral Descartes configurations exist in dimensions of the form $n = 2k^2$ or $(2k-1)^2$, for some finite set S of primes, which depends on the dimension, but we did not determine an explicit set S that can be used in such dimensions. It is an open problem to find a minimal set S. In particular do there exist Descartes configurations which are super-strongly integral ($S = \{1\}$) in all such dimensions?

This paper treated Apollonian packings in Euclidean space. Such packings can also be constructed in spherical n-space (positive curvature), and in hyperbolic n-space (negative curvature). In spherical and hyperbolic space the notion of center and radius of a sphere change, but there exist suitable analogues of (augmented) curvature-center coordinates for Descartes configurations, see [21]. Various questions raised in this paper may have interesting analogues in these geometries.

Acknowledgments. The authors are grateful for helpful comments from Andrew Odlyzko, Eric Rains, Jim Reeds and Neil Sloane during this work.

References

[1] A. F. Beardon, *The Geometry of Discrete Groups,* Springer-Verlag: New York 1983.

[2] M. Berger, *Geometry II*, Springer-Verlag: Berlin 1987.

[3] D. W. Boyd, The osculatory packing of a three-dimensional sphere. Canadian J. Math. **25** (1973), 303–322.

[4] D. W. Boyd, The residual set dimension of the Apollonian packing. Mathematika **20** (1973), 170–174.

[5] D. W. Boyd, A new class of infinite sphere packings, Pacific J. Math. **50** (1974), 383–398.

[6] J. W. S. Cassels, *Rational Quadratic Forms*, Academic Press: New York 1978.

[7] W. K. Clifford, On the powers of spheres (1868), in: *Mathematical Papers of William Kingdon Clifford,* MacMillan and Co., London 1882, pp. 332-336.

[8] J. H. Conway, with F. Fung, *The sensual (quadratic) form,* Carus Monograph No. 26, Math. Assoc. America, Washington DC, 1997.

[9] H. S. M. Coxeter, The problem of Apollonius. Amer. Math. Monthly **75** (1968), 5–15.

[10] H. S. M. Coxeter, *Introduction to Geometry, Second Edition*, John Wiley and Sons, New York, 1969.

[11] H. S. M. Coxeter, Loxodromic sequences of tangent spheres. Aequationes Mathematicae **1** (1968), 104–121.

[12] H. S. M. Coxeter, Numerical distances among the spheres in a loxodromic sequence. The Mathematical Intelligencer **19** (1997), 41–47.

[13] G. Darboux, Sur les relations entre les groupes de points, de cercles et de sphéres dans le plan et dans l'espace, Ann. Sci. École Norm. Sup. 1 (1872), 323–392.

[14] T. Gossett, The Kiss Precise, Nature **139**(1937), 62.

[15] T. Gossett, The Hexlet, Nature **139** (1937), 251.

[16] R. L. Graham, J. C. Lagarias, C. L. Mallows, A. Wilks and C. Yan, Apollonian circle packings: geometry and group theory I. The Apollonian group, preprint.

[17] R. L. Graham, J. C. Lagarias, C. L. Mallows, A. Wilks and C. Yan, Apollonian circle packings: geometry and group theory II. Super-Apollonian group and integral packings, preprint.

[18] R. L. Graham, J. C. Lagarias, C. L. Mallows, A. Wilks and C. Yan, Apollonian circle packings: number theory, eprint: `arXiv math.NT/0009113.`

[19] R. Lachlan, On systems of circles and spheres, Phil. Trans. Roy. Soc. London, Ser. A **177** (1886), 481–625.

[20] J. C. Lagarias and C. L. Mallows, paper in preparation.

[21] J. C. Lagarias, C. L. Mallows and A. Wilks, Beyond the Descartes circle theorem, eprint: `arXiv math.MG/0101066, 9 Jan 2001`.

[22] D. G. Larman, On the exponent of convergence of a packing of spheres, Mathematika **13** (1966), 57–59.

[23] J.G. Mauldon, Sets of equally inclined spheres, Canadian J. Math. 14 (1962), 509–516.

[24] G. Maxwell, Sphere packings and hyperbolic reflection groups. J. Algebra **79** (1982), 78–97.

[25] D. Pedoe, On a theorem in geometry, Amer. Math. Monthly **74** (1967), 627–640.

[26] W. Rühl, *The Lorentz group and harmonic analysis*, W. A. Benjamin: New York 1970.

[27] N. J. A. Sloane, The on-line encyclopedia of integer sequences.
(URL is http://www.research.att.com/~njas/sequences/index.html)

[28] F. Soddy, The Kiss Precise. Nature **137** (1936), 1021.

[29] F. Soddy, The Hexlet. Nature **138** (1936), 958.

[30] F. Soddy, The bowl of integers and the hexlet, Nature **139** (1937), 77-79.

[31] J. B. Wilker, Inversive Geometry, in: *The Geometric Vein*, (C. Davis, B. Grünbaum, F. A. Sherk, Eds.), Springer-Verlag: New York 1981, pp. 379–442.

email: graham@ucsd.edu
jcl@research.att.com
clm@research.att.com
allan@research.att.com
cyan@math.tamu.edu

Tractability of parameterized completion problems on chordal, strongly chordal and proper interval graphs*

Haim Kaplan† Ron Shamir‡ Robert E. Tarjan§

May 1, 1996

Abstract

We study the parameterized complexity of three NP-hard graph completion problems.

The MINIMUM FILL-IN problem is to decide if a graph can be triangulated by adding at most k edges. We develop an $O(k^2mn + f(k))$ algorithm for this problem on a graph with n vertices and m edges. In particular, this implies that the problem is fixed-parameter tractable (FPT).

The PROPER INTERVAL GRAPH COMPLETION problem, motivated by molecular biology, asks if a graph can be made proper interval by adding no more than k edges. We show that the problem is FPT by providing a simple search-tree-based algorithm that solves it in linear time. Similarly, we show that the parameterized version of the STRONGLY CHORDAL GRAPH COMPLETION problem is FPT by giving an $O(m \log n)$-time algorithm for it.

All our algorithms can actually enumerate all possible k-completions within the same time bounds.

AMS (MOS) subject classification: 68Q20, 68R15, 05C85

Key words: Design and analysis of algorithms, parameterized complexity, chordal graphs, proper interval graphs, strongly chordal graphs, minimum fill-in, physical mapping of DNA.

Abbreviated title: Parameterized Completion.

*Portions of this paper were presented at the 34th Annual IEEE Symp. on the Foundations of Computer Science, Santa Fe, NM 1994 [20].

†Department of Computer Science, Princeton University, Princeton, NJ 08544 USA. Research at Princeton University partially supported by the NSF, Grant No. CCR-8920505, and the Office of Naval Research, Contract No. N00014-91-J-1463. hkl@cs.princeton.edu.

‡Department of Computer Science, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel-Aviv 69978 ISRAEL. Research supported in part by a grant from the Ministry of Science and the Arts, Israel. shamir@math.tau.ac.il

§Department of Computer Science, Princeton University, Princeton, NJ 08544 USA and NEC Institute, Princeton, NJ. Research at Princeton University partially supported by the NSF, Grant No. CCR-8920505, and the Office of Naval Research, Contract No. N00014-91-J-1463. ret@cs.princeton.edu.

1 Introduction.

The focus of this paper is the parameterized complexity of several graph completion problems. Many well-known NP-hard problems can be stated with a parameter k so that they have polynomial-time algorithms when k is fixed. (For example, given a graph, decide if it has a vertex cover of size at most k, an independent set of size at least k, or pathwidth at most k.) The way the complexity depends on k varies dramatically, however. Some problems (eg. VERTEX COVER and PATHWIDTH) can be solved in linear time when k is fixed, but for others (like INDEPENDENT SET) the best known algorithms require $\Omega(n^k)$ steps. How the complexity depends on k can be crucial for applications in which small, fixed parameter values are important, as in the problems we study here.

Downey and Fellows initiated a systematic complexity analysis of such problems [1, 8, 9]. They called those parameterized problems that have algorithms of complexity $O(f(k)n^\alpha)$ (with α a constant) *fixed parameter tractable* (FPT), and defined a hierarchy of parameterized decision problem classes, $FPT \subseteq W[1] \subseteq W[2] \subseteq \ldots$, with appropriate reducibility and completeness notions. They also conjectured that each of the containments in this hierarchy is proper. (cf. [1, 8, 9] for definitions and details.) Thus, for example, VERTEX COVER and PATHWIDTH are in FPT [3, 10, 23] but INDEPENDENT SET is $W[1]$-complete [1], and BANDWIDTH is W[t]-hard for all t [4].

Let Π be a family of graphs such that $K_n \in \Pi$ for every n. The Π-*COMPLETION* problem is defined as follows. Given a graph $G = (V, E)$ find a smallest set of edges A such that $G = (V, E \cup A) \in \Pi$. The parameterized version of the Π-COMPLETION problem, denoted by Π-*COMPLETION(k)*, asks whether there exists an edge set A such that $|A| \leq k$ and $G = (V, E \cup A) \in \Pi$.

In this paper we study the parameterized complexity of Π-COMPLETION(k) for three graph families Π; namely, chordal, proper interval and strongly chordal graphs.

A graph is *chordal* (or *triangulated*) if every cycle of length four or more contains a chord (an edge between nonadjacent vertices on the cycle). The CHORDAL COMPLETION problem is also known as the MINIMUM FILL-IN problem and has received a lot of attention in the past due to its importance in sparse matrix computation (cf. [16]). Rose [32] has shown that for a sparse, symmetric matrix, finding an order of Gaussian elimination steps on diagonal elements that minimizes the number of non-zeros generated in the elimination process (assuming no lucky cancelation of non-zeros) is equivalent to solving the minimum fill-in problem on a corresponding graph.

Yannakakis [40] has shown that minimum fill-in is NP-complete. We focus here on CHORDAL

COMPLETION(k) or FILL-IN(k), the parametrized version of the problem as defined above. Here k is fixed (to be thought of as a small constant) and is not part of the input. For a graph with n vertices and m edges, the problem can be solved by enumeration in $O(n^{2k}m)$-time, but we seek an algorithm with better dependence on k. In section 2 we describe two such algorithms. We first present a fairly simple, $O(c^k(m+n))$-time search-tree-based algorithm, which already implies that the problem is in FPT. The same technique was previously used by Downey and Fellows [10] to prove parameterized tractability of VERTEX COVER, DOMINATING SET IN PLANAR GRAPH, FEEDBACK VERTEX SET, and FACE COVER NUMBER OF PLANAR GRAPH. We then develop a more involved algorithm that gives a stronger complexity result: its multiplicative factor depending on k is *polynomial*, and the exponential in k appears only as an *additive* factor. Specifically, this algorithm has complexity $O(k^2nm + f(k))$.

The second part of the paper deals with the parameterized complexity of the PROPER INTERVAL GRAPH COMPLETION (PIGC) problem. An *interval graph* is a graph for which one can assign an interval on the real line to each vertex so that two vertices are adjacent if and only if their intervals intersect. It is a *unit interval graph* if all intervals assigned have equal length. It is *proper interval* if it has an assignment in which no interval properly contains another. The last two notions are equivalent for finite graphs [29]. Interval completion problems arise in molecular biology and in the Human Genome Project. In *physical mapping* of DNA, a set of long contiguous intervals of the DNA chain (called *clones*) is given, together with experimental information on their pairwise overlaps. The goal is to build a map describing the relative position of the clones [6, 26, 21, 4]. We concentrate here on the biologically important case in which all clones have equal length. In the presence of "false negative" errors (unidentified overlaps) the problem of building a map with fewest errors is equivalent to PIGC. This problem is NP-hard [17]. But what about its complexity for a small fixed number of errors? Let PIGC(k) be the parameterized version of the problem, in which one asks for an augmenting set with no more than k edges if one exists. We prove parameterized tractability of PIGC(k) by providing a linear-time algorithm for fixed k.

The third part of the paper considers the parameterized version of the STRONGLY CHORDAL COMPLETION problem, denoted by SCC(k). The class of strongly chordal graphs was defined and characterized by Farber [11]. Denote by $N(v)$ the set of neighbors of a vertex v, including v itself. A *perfect elimination ordering* of a graph $G = (V, E)$ is an ordering $v_1, v_2, \ldots, v_n$ of V with the property that for each i, j and l, if $i < j, i < l$, and $v_l, v_j \in N(v_i)$, then $v_l \in N(v_j)$. Rose [30] has shown that a graph is chordal iff it admits a perfect elimination ordering. A *strong elimination ordering* of a graph $G = (V, E)$ is an ordering $v_1, v_2, \ldots, v_n$ of V with the property that for each i, j, k and l, if $i < j$,$k < l$, $v_k, v_l \in N(v_i)$, and $v_k \in N(v_j)$,

then $v_l \in N(v_j)$. A graph is *strongly chordal* if it admits a strong elimination ordering. It is easy to see that every strong elimination ordering is also a perfect elimination ordering, and thus every strongly chordal graph is also a chordal graph. In addition every interval graph is strongly chordal. One can obtain a strong elimination order for an interval graph G by fixing a representation R of G and ordering the vertices in increasing right-endpoint order of their intervals in R. Interest in strongly chordal graphs arises in several ways. First, the problems of locating minimum weight dominating sets and minimum weight independent dominating sets in strongly chordal graphs with real vertex weights can be solved in polynomial time, whereas each of these problems is NP-hard for chordal graphs [12]. Second, these graphs have surprisingly nice structural properties and are intimately related to the class of totally balanced matrices [2]. We show that SCC(k) is fixed parameter tractable by describing an $O(m \log n)$-time algorithm for it.

Section 2 contains the algorithms for chordal completion. Section 2.1 describes the simple search-tree-based algorithm and Section 2.2 gives the details of the more involved $O(k^2nm + f(k))$-time algorithm. Section 3 extends the search tree algorithm of Section 2.1 to solve PIGC(k) and Section 4 extends it to solve SCC(k). These extensions require additional ideas in order to handle the obstructions characterizing each particular graph family. Section 5 contains a summary and suggestions for some further research.

2 Minimum Fill-In.

In this section we present two algorithms for FILL-IN(k). In Section 2.1 we begin by describing an $O(c^k m)$-time time algorithm for the problem. Then in Section 2.2 we use additional new ideas to develop an $O(k^2nm + f(k))$-time algorithm. Both algorithms can actually enumerate all minimal k-triangulations of the input graph within the same time bounds.

We will use the following notation. Let $G = (V, E)$ be an undirected graph. For $X \subseteq V$, we denote by G_X the subgraph of G induced by the vertex set X. We define the *length* of a path (cycle) as the number of edges on the path (cycle). A *triangulation* of a graph $G = (V, E)$ is a set of edges F where $E \cap F = \emptyset$ and $\tilde{G} = (V, E \cup F)$ is a chordal graph. We will also say that the set of edges F *triangulates* G. If $|F| \leq k$ then F is a *k-triangulation*. We shall also refer to $\tilde{G}$ as a triangulation of G, when there is no confusion. We assume without loss of generality that G is connected and $n \geq 2$; thus $n = O(m)$. A triangulation F is *minimal* if no proper subset of F triangulates G.

2.1 A linear algorithm for fixed k.

A *triangulation of a chordless cycle* C is a set T of chords of C such that there is no induced chordless cycle in $C \cup T$. We shall characterize and count the number of minimal triangulations of a cycle C. We call a cycle an *l-cycle* if it contains l vertices. A *triangle* is a 3-cycle. The proof of the following lemma is straightforward by induction.

Lemma 2.1 *A minimal triangulation T of an n-cycle C consists of $n-3$ chords. It partitions C into $n-2$ triangles. Any two of these triangles are either disjoint or share a chord. Every chord in T is shared by exactly two triangles.* ■

The following lemma is well known (cf. [33] and the proof of Lemma 4.3, which is similar).

Lemma 2.2 *There is a 1-1 correspondence between the minimal triangulations of a cycle with l vertices and the binary trees with $l-2$ internal nodes.* ■

Denote by c_l the l-th Catalan number, i.e., $c_l = \binom{2l}{l}\frac{1}{l+1}$. Note that $c_l < 4^l$. Denote the number of binary trees with n internal nodes by b_n. The value b_n satisfies the recurrence $b_0 = 1$, $b_n = \sum_{i+j=n-1} b_i b_j$ for $n \geq 1$. The solution to this recurrence is $b_n = c_n$ (cf. [18]). Thus the following lemma is implied by Lemma 2.2.

Lemma 2.3 *The number of minimal triangulations of an l-cycle is $c_{l-2} \leq 4^{l-2}$.* ■

The algorithm will traverse part of a search tree in which each node corresponds to a supergraph of G. This search tree is defined as follows. The graph G itself corresponds to the root of the tree. In order to generate the children of an internal node x that corresponds to a graph G', one needs to find a chordless cycle C in G'. Node x will have a child for each minimal triangulation of C. The graph corresponding to a child is obtained by adding the corresponding minimal triangulation to G'. If $|C| = l$, by Lemma 2.3 node x will have c_{l-2} children. Each leaf of the tree corresponds to a chordal supergraph of G. Note that every minimal triangulation of G is represented by at least one leaf.

One can find a chordless cycle C in a nonchordal graph with m edges in $O(m+n)$ time by the maximum cardinality search (MCS) algorithm described in [36, 37]. Using the algorithm described in [34] and the mapping described in Lemma 2.2, one can generate all minimal triangulations in $O(|C|)$ time per triangulation.

The algorithm actually visits only the nodes of the search tree that correspond to supergraphs of G with no more than k additional edges. If one such node is a leaf then we have found a k-triangulation. Otherwise, no such triangulation exists.

Theorem 2.4 *All minimal k-triangulations of a graph G can be found in $O(2^{4k}m)$ time.*

Proof Let T be the subtree of the search tree traversed by the algorithm. For a node $x \in T$ let $G_x = (V, E_x)$ be the corresponding supergraph of G, d_x the maximum length of a path from x to a leaf of T and $a_x = \max(\{|E_l| - |E_x| \mid l \text{ is a leaf descendant of } x\})$. Denote by l_x the total number of leaves among the descendants of x. By induction we prove that $l_x \leq 4^{d_x + a_x}$. Thus the total number of nodes in T is bounded by $2 \cdot 4^{2k}$. For each such node a linear amount of time is spent, consisting of the time to generate it and the time to find a chordless cycle in the graph corresponding to it.

Here is the induction argument. Assume the claim is true for all the children of a node x. Let l be the length of the cycle detected at x. Let $d_{max} = \max\{d_y \mid y \text{ is a child of x}\}$ and Let $a_{max} = \max\{a_y \mid y \text{ is a child of x}\}$. Using the induction hypothesis the number of leaf descendants of any of the c_{l-2} children of x is bounded by $4^{d_{max}+a_{max}}$. Thus the total number of leaf descendants of x is bounded by $4^{l-2}4^{d_{max}+a_{max}} = 4^{d_{max}+1+a_{max}+l-3} = 4^{d_x+a_x}$. The last equality follows from the fact that the size of a minimal triangulation of a chordless l-cycle is $l-3$ as stated in Lemma 2.1. ■

The algorithm for enumerating minimal k-triangulations can actually list the same triangulation several times. We can eliminate this redundancy by storing solutions in a table and checking each new solution to see if it has been found already. If we use a k-dimensional search tree to store solutions, the extra time per search tree node to test for redundancy is $O(k \log k)$. Using universal hashing [38] or dynamic perfect hashing [15], the extra time per search tree node is $O(k)$, but the algorithm becomes randomized. These ideas apply equally well to the other enumeration algorithms proposed in this paper.

2.2 An algorithm with a polynomial multiplicative factor.

To achieve an $O(k^2nm + f(k))$ time bound for minimal k-triangulation we first describe an algorithm such that if G can be triangulated with no more than k edges, the algorithm partitions the vertex set of G into two subsets A, B, such that the size of A is $O(k^3)$ and there are no chordless cycles in G that contain vertices in B. Then we prove that obtaining a k-triangulation of G is equivalent to obtaining a $(k-a)$-triangulation of A for some $a \geq 0$.

Partitioning the graph The algorithm uses three main procedures, denoted by P_1, P_2, P_3, executed in sequence. These procedures are described below.

P_1) *Extracting independent chordless cycles :*
This procedure starts with $B = V, A = \emptyset$ and repeatedly finds a chordless cycle in G_B using the MCS algorithm and moves its vertices to A. Note that when P_1 is finished, the induced subgraph on B is chordal.

Let $C_1, \ldots, C_j$ be the cycles extracted. The minimum number of chords needed to triangulate each C_i is $|C_i| - 3$. The algorithm maintains a dynamic lower bound cc on the minimum number of chords needed to triangulate G. After detecting the chordless cycle C_i it increases cc by $|C_i| - 3$. Thus, if at some point $cc > k$ the algorithm can stop with a negative answer. Otherwise procedure P_1 ends when there are no more chordless cycles in B and $cc = \sum_{i=1}^{j}(|C_i| - 3) \leq k$.

The complexity of this part is $O(km)$. The MCS algorithm runs in linear time and the number of cycles detected is not greater than k since each cycle adds at least one to the dynamic lower bound cc. The size of the set A after performing this procedure is $O(k)$.

P_2) *Extracting related chordless cycles with independent paths:*
This procedure looks for chordless cycles that intersect both parts of the current partition, A and B, and contain at least two consecutive vertices in B, as long as such cycles exist. Let C be such a cycle, $|C| = l$. If $l > k + 3$ the algorithm stops with a negative answer. Otherwise every maximal subpath of C containing only vertices from B is moved into A if its length is at least one. The increase to cc depends on the structure of C. We need the following lemma in order to specify this increase precisely.

Lemma 2.5 *Let C be a chordless cycle and let p be a path in C of length l with $1 \leq l \leq |C| - 2$. If $l = |C| - 2$, then in every minimal triangulation of C there are at least $l - 1$ chords incident with at least one vertex of p. If $l < |C| - 2$ then in every minimal triangulation of C there are at least l chords incident with at least one vertex on p.* ■

Proof If $l = |C| - 2$ then every chord in a minimal triangulation of C is incident with some vertex of p; thus the first part of the lemma holds. We prove the second part by induction on the path length. Obviously there must be a chord incident with at least one of the vertices on p; thus the lemma holds for paths of length one. Assume the result is true for every path with length less than l. Let p be a path with length l. Let (a, b) be a chord incident with p dividing the cycle C into two cycles C_1, C_2.
Case 1: $a, b \in p$. Let l_1 be the length of the subpath of p that connects a and b. Without loss

of generality we can assume that $l_1 = |C_1| - 1$. Let p' be the path between the endpoints of p passing through (a,b) in C_2, and let $l_2 = |p'|$. There must be at least $l_1 - 2$ chords incident with p in C_1, and according to the induction hypothesis l_2 chords incident with p' in C_2. Thus the total number of chords incident with p will be at least $(l_1 - 2) + l_2 + 1 = l$.
Case 2: $a \in p$, $b \notin p$. Let $p_1 = p \cap C_1$, $p_2 = p \cap C_2$. For at least one $i = 1, 2$, $|p_i| < |C_i| - 2$. W.l.o.g. assume that $|p_1| < |C_1| - 2$. By applying the induction hypothesis and using the previous part of the lemma we find that the total number of chords incident with p is at least $l_1 + (l_2 - 1) + 1 = l$. ■

Suppose that C is a chordless l-cycle that contains $j \geq 1$ disjoint maximal subpaths $p_1, \ldots, p_j$, each of length at least one, that are in B. Let $l_i = |p_i|$, $i = 1, \ldots, j$. Obviously if $l_1 = l - 2$ then $j = 1$, i.e. there is only one such subpath. Otherwise $l_i < l - 2$ for every $1 \leq i \leq j$. Using the previous lemma we can increase our lower bound *cc* as follows. If there is only one such subpath, *cc* is increased by either $(l_1 - 1)$ if $l_1 = l - 2$ or l_1 if $l_1 < l - 2$. Otherwise *cc* is increased by the larger of $\frac{1}{2}\sum_{i=1}^{j} l_i$ (the factor $\frac{1}{2}$ is needed because a chord can be counted twice in the sum) and $\max\{l_i \mid 1 \leq i \leq j\}$. P_2 terminates whenever either *cc* is greater than k, in which case it stops with a negative answer, or when there are no more cycles of the appropriate kind.

In order to complete the description of P_2 we need to specify how to detect a chordless cycle C with consecutive vertices in B if such a cycle exists. The following observation is useful.

Observation 2.6 *There exists a chordless cycle C with at least two consecutive vertices in B if and only if there exists an edge (x,y), $x \in A$, $y \in B$ and a path between a vertex in $(N(y) - N(x)) \cap B$ and a vertex in $N(x) - N(y)$ that avoids any other vertices in $N(x) \cup N(y)$.*

One can detect whether such a path exists as follows: Delete $N(x) \cap N(y)$ and $(N(y) - N(x)) \cap A$ from G. Find the connected components of G induced on the other vertices. Check whether there is a vertex in $(N(y) - N(x)) \cap B$ and a vertex in $N(x) - N(y)$ in the same connected component. This process requires $O(m)$ time per edge (x,y) and can be implemented so that if the path exists then the process will output a chordless cycle through (x,y) for which the other neighbor of y is also in B.

Recall that the size of A after the execution of P_1 is $O(k)$. The number of vertices added to A after the detection of each cycle by P_2 is at most twice the increase to *cc*. Since *cc* is never greater than k the total number of vertices in A when P_2 ends remains $O(k)$.

P_3) Adding essential edges in G_A:
For every nonadjacent pair of vertices $y, z \in V$ define $A_{y,z}$ to be the set of all vertices x such

that y, x, z appear consecutively on some chordless cycle in G.

Lemma 2.7 *If for some pair $y, z \in A$, $(y, z) \notin E$, $|A_{y,z}| > 2k$ then the edge (y, z) is in every k-triangulation of G.*

Proof Assume that (y, z) is not in a k-triangulation $\overline{G} = (V, \overline{E})$ of G. Then there must be a chord in $\overline{E} - E$ incident with each vertex in $A_{y,z}$. Since no more than two such vertices can share a chord, $|\overline{E} - E| > k$, which is a contradiction. ■

Edges (y, z) satisfying the lemma are called *essential*. For a triple y, x, z such that $(y, x) \in E$, $(x, z) \in E$, $(y, z) \notin E$ one can determine whether y, x, z appear consecutively on some chordless cycle in linear time: They appear consecutively on a chordless cycle if and only if y and z are in the same connected component after deleting $N(x) - \{y, z\}$ from G.

P_3 first calculates the sets $A_{y,z}$ for every pair $y, z \in A$, $(y, z) \notin E$. Then for each pair $y, z \in A$ such that $|A_{y,z}| > 2k$, we add (y, z) to G'. Finally, we add to A all vertices in each computed set $A_{y,z}$ such that $|A_{y,z}| \leq 2k$.

We now analyze the overall complexity of the partitioning scheme.

Lemma 2.8 *1) The execution of P_2 takes $O(knm)$ time. 2) The execution of P_3 takes $O(k^2nm)$ time.*

Proof 1) For each edge (x, y), $x \in A$, $y \in B$, it takes linear time to find a chordless cycle through (x, y) with consecutive vertices in B. The size of A is always $O(k)$; thus the total number of edges incident with vertices of A is always $O(kn)$. For each such edge we may have to run the test mentioned above once, giving a total time complexity $O(knm)$.

2) Since the size of A when P_3 begins its execution is $O(k)$, the number of triples y, x, z such that $(y, x), (z, x) \in E$, $(y, z) \notin E$, $y, z \in A$, is $O(k^2n)$. For each triple we need to check whether there exists a path between y and z after deleting $N(x) - \{y, z\}$ from G. As mentioned above, this can be done by identifying connected components of G on the remaining vertices and then checking whether y and z are in the same connected component. ■

Thus the overall complexity of the partitioning procedure is dominated by the complexity of P_3, which is $O(k^2nm)$. Before the call to P_3 the size of the set A is $O(k)$. Procedure P_3 may add $O(k)$ additional vertices to A for each pair of vertices in A prior to its execution, so that we end up with $O(k^3)$ vertices in A.

Let E_s be the set of essential edges detected by P_3, and let $G' = (V, E \cup E_s)$. Denote by A_2, B_2 the partition of the vertex set before the execution of P_3 and by A, B the final partition.

The following lemma will be useful in establishing the correctness of the partitioning scheme and the completion algorithm.

Lemma 2.9 *Let $G = (V, E)$ be a graph and $v \in V$. Let F be a set of edges between vertices of G such that*
1) Each $e \in F$ is a chord in a chordless cycle C_e of G.
2) $F \cap E = \emptyset$.
3) v is not an endpoint of any $e \in F$.
Denote by G^+ the graph obtained from G by adding the edges in F. If there exists a chordless cycle C in G^+ with v_1, v, v_2 occurring consecutively on C then either there exists a chordless cycle in G on which v_1, v, v_2 occur consecutively, or there exists a chordless cycle $D_e = v, x_1, \ldots, x_t, v$ in G such that the path $p = x_1, \ldots, x_t$ is part of a cycle C_e for some $e \in F$, and p contains one of the endpoints of e.

Proof For an $e = (x, y) \in F$ let P_e^1 and P_e^2 denote the two paths on C_e between x and y, with x and y removed from each path. Since C_e is chordless, for every e such that $v \in C_e$, v is not adjacent to any vertex either on P_e^1 or on P_e^2.
Case 1: For every $e \in F$ such that $v \notin C_e$, there exists a path $P_e \in \{P_e^1, P_e^2\}$ such that v is not adjacent in G to any vertex on P_e. Consider the cycle C. Replacing every edge $e \in F$ along C by P_e, one gets a cycle C' in G (not necessarily chordless or simple) with the property that v is not adjacent to any vertex in $C' - \{v_1, v_2\}$. Since edges in F are not incident with v, the edges (v, v_1) and (v, v_2) exist in G. Since C is chordless, $(v_1, v_2) \notin E$. Thus C' contains a chordless cycle in G on which v_1, v, v_2 occur consecutively.
Case 2: For some $e = (x, y) \in F$, v is adjacent to a vertex $u_1 \in P_e^1$ and a vertex $u_2 \in P_e^2$. Since C is chordless in G^+, v must be nonadjacent either to x or to y in G. W.l.o.g. assume v is not adjacent to x and that u_1 and u_2 are the closest to x among all vertices on P_e^1 and P_e^2 respectively that are adjacent to v. D_e is the chordless cycle in G consisting of the path between u_1 and u_2 through x on C_e and v. ■

The correctness of the partitioning scheme is captured by the following theorem.

Theorem 2.10 *When the partitioning procedure ends, the graph G' has no chordless cycles with vertices in B.*

Proof The proof is by contradiction. Suppose that there is a chordless cycle $C \in G'$ such that $C \cap B \neq \emptyset$ and let v be a vertex in $C \cap B$. Denote by v_1 and v_2 the two neighbors of v on C. Cycle C must contain at least one essential edge since otherwise C exists in G and either v

would have been moved to A or (v_1, v_2) would have been added as an essential edge. Let F be the set of essential edges on C. By the definition of an essential edge, for each $e = (x, y) \in F$ there is a chordless cycle C_e in G in which e is a chord. Moreover, if P_e^1 and P_e^2 are the two paths connecting x and y on C_e then either P_e^1 or P_e^2 consists of a single vertex $z_e \in B_2$. Since v is in B it is not incident with any essential edge. Applying Lemma 2.9 we find that one of the following things must happen.
1) There exists in G a chordless cycle on which v_1, v, v_2 occur consecutively. Thus either v should have been in A or (v_1, v_2) should have been added as an essential edge, and we obtain a contradiction.
2) There exists a chordless cycle D_e in G on which v and z_e occur consecutively for some $e \in F$. Since both v and z_e are in B_2, they both should have been moved to A by P_2, and again we obtain a contradiction. ■

Triangulating the graph All that remains is to show that once we have partitioned the graph, it suffices to look for $(k - a)$-triangulations of the smaller graph with vertex set A, where a is the number of essential edges added during the partitioning algorithm. This is the content of Theorem 2.13 below. In order to prove the theorem we need some background and preliminary results.

We define the *elimination* of a vertex v from G as the operation that deletes v from G and adds an edge between every nonadjacent pair among v's neighbors. Let $\alpha = v_1, \ldots, v_n$ be an ordering of the vertices of a graph $G = (V, E)$. We denote by G_i, $0 \leq i \leq n$ the graph obtained from G after eliminating the first i vertices in α ($G_0 = G$). Let x and y be two vertices in $G = (V, E)$. An *x,y separator* of G is a set $S \subseteq V - \{x, y\}$ such that when S is deleted from G, x and y occur in different connected components.

The following characterization of minimal triangulations was proved by Ohtsuki, Cheung and Fujisawa.

Theorem 2.11 ([27]) *A triangulation F of $G = (V, E)$ is minimal if and only if, for each $(x, y) \in F$, there exists no x, y separator S of G such that S is a clique of the triangulated graph $\tilde{G} = (V, E \cup F)$.* ■

Using this theorem we prove the following lemma that is needed for the proof of Theorem 2.13.

Lemma 2.12 *Let F be a minimal triangulation of a graph $G = (V, E)$. Any edge in F is a chord in a chordless cycle of G.*

Proof Let $v_1, \ldots, v_n$ be a perfect elimination ordering of $\tilde{G} = (V, E \cup F)$. We use this ordering to eliminate vertices from G. Since F is minimal, for each edge $e = (u, w) \in F$ there exists an index k, $k \geq 1$, such that $e \in G_k$ but $e \notin G_{k-1}$.

We claim that u and w are connected in G_{k-1} by a path such that none of its vertices is adjacent to v_k. Here is the proof of the claim. Assume that no such path exists. Then the set $N_{G_{k-1}}(v_k) - \{u, w\}$ separates u and w in G_{k-1}. But it follows from the definition of a perfect elimination ordering that this set is a clique in $\tilde{G}_{k-1}$. Since F is a minimal triangulation of G we must also have that $\tilde{G}_{k-1}$ is a minimal triangulation of G_{k-1}. This contradicts Theorem 2.11 and the claim follows.

We obtain that e is a chord of a chordless cycle in G_{k-1}. This cycle consists of u, v_k and w occurring consecutively and a shortest path between u and w that avoids the neighborhood of v_k in G_{k-1}. We finish the proof by showing that e is also a chord of a chordless cycle of G. This is done by arguing that if C is a chordless cycle in G_j for some j, $1 \leq j \leq n$ then there exists a chordless cycle C' in G_{j-1} that is either identical to C or contains one additional vertex. If all the edges in C are in G_{j-1}, then $C' = C$ is a chordless cycle in G_{j-1}. Otherwise there is an edge (x, y) in C that is not in G_{j-1}. For each such edge both its endpoints must be adjacent to v_j. Of the vertices on C only x and y can be adjacent to v_j, since otherwise C is not chordless in G_j. Take C' to be C with the vertex v_j added between x and y. ■

Theorem 2.13 *Let A, B be a partition of the vertex set V of a graph $G = (V, E)$ such that the vertices of every chordless cycle in G are contained in A. A set of edges F is a minimal triangulation of G if and only if F is a minimal triangulation of G_A.*

Proof Let F be a minimal triangulation of G_A. We need to prove that $\tilde{G} = (V, E \cup F)$ is chordal. Assume that $\tilde{G}$ is not chordal. Let C be a chordless cycle in $\tilde{G}$. Since $\tilde{G}$ induced on A is chordal, $C \cap B \neq \emptyset$. By assumption, G does not contain chordless cycles with vertices in B; hence C must not exist in G and thus it contains at least one edge from F and $|C \cap A| \geq 2$. Let v be a vertex in $C \cap B$. According to Lemma 2.12 each edge $e \in F$ is a chord in a chordless cycle C_e of G whose vertices are in A. Since F is a minimal triangulation of G_A, v is not an endpoint of any edge in F. Using Lemma 2.9 we conclude that there must be a chordless cycle with v on it in G, contradicting the assumptions of the theorem.

To prove the other direction let F be a minimal triangulation of G. There exists $F' \subseteq F$ that is a minimal triangulation of G_A. According to the first part of the proof F' also triangulates G. Since F is minimal we conclude that $F' = F$. ■

Overall Running Time The final step of the algorithm is to look for $(k-a)$-triangulations

in vertex set A, as justified by Theorem 2.13. One can find one or all such triangulations by the algorithm described in Section 2.1. Since the size of A is $O(k^3)$, the running time for this step is $O(k^6 2^{4k})$. The total time for the three-step partitioning process is $O(k^2 nm)$, giving a time bound for the entire algorithm of $O(k^2 nm + k^6 2^{4k})$.

3 Unit Interval Completion

A proper interval supergraph $G = (V, E \cup F)$ of a graph $G = (V, E)$ with $|F| \leq k$ is called a *k-proper interval supergraph* of G.

The algorithm presented in Section 2.1 can be easily modified to produce all possible k-proper interval supergraphs of a graph, using the following observations. Proper interval graphs are exactly the chordal graphs that do not contain any of the three obstructions in Figure 1 as an induced subgraph [39]. Deng, Hell and Huang [7] have recently described an algorithm that checks whether a graph G is a proper interval graph. In case G is indeed a proper interval graph the algorithm can provide a proper interval representation for G. The running time of the algorithm is $O(m)$, and it does not use complicated data structures such as PQ-trees [5]. It is straightforward to check that in case the input graph is not a proper interval graph, one can use the information maintained by the algorithm to extract either a chordless cycle or one of the obstructions in Figure 1 in linear time.

The k-completion algorithm will traverse part of a search tree defined as follows. The graph G itself corresponds to the root of the tree. Let x be a node of the search tree corresponding to a supergraph G_x of G that is not a proper interval graph. The children of x are obtained as follows. The algorithm by Deng, Hell and Huang is applied to G_x to find either a chordless cycle or one of the obstructions in Figure 1. If a chordless cycle C is found in G_x then every minimal triangulation of C gives rise to a child of x as in Section 2.1. In case an obstruction is found, x has a child for every edge e between vertices of the obstruction that is not part of the obstruction. The supergraph corresponding to such a child is $G_x \cup \{e\}$. Thus if the obstruction found is a tent the node has six children, if it is a claw it has three and if it is a net it has nine.

Each leaf in the search tree thus defined corresponds to a proper interval supergraph of G. Note that every minimal proper interval supergraph of G is represented by at least one leaf. As in section 2.1 the nodes of the search tree that are actually traversed correspond to supergraphs with no more than k additional edges. If one such node is a leaf then we have found a k-proper interval supergraph. Otherwise, no such supergraph exists.

We summarize the result presented in this section in the following theorem. Its proof is

analogous to the proof of Theorem 2.4 and hence omitted.

Theorem 3.1 *All k-proper interval supergraphs of a graph can be found in* $O(2^{4k}m)$ *time.* ■

Remark Rose, Tarjan and Lueker proved that if $G = (V, E)$ is triangulated and $G = (V, E \cup F)$, with $F \neq \emptyset$, $F \cap E = \emptyset$ is triangulated, then there exists an edge $e \in F$ such that $G = (V, E \cup \{e\})$ is also triangulated [31, Lemma 2]. Using this lemma, while traversing the search tree as described above one can avoid generating non-triangulated children of nodes that correspond to triangulations of G. Each minimal proper interval completion of G is still guaranteed to be represented by at least one leaf. In this version of the algorithm one uses the MCS algorithm to detect chordality and find a chordless cycle as long as a chordal supergraph has not been reached. When reaching a chordal supergraph, the algorithm by Deng, Hell and Huang is applied to get one of the obstructions in Figure 1. The children of the node are then generated as described above. Finally the MCS algorithm is applied to each of the children in order to avoid traversing those that are not chordal. Those that are chordal are further expanded. Such an implementation would use the algorithm of Deng, Hell and Huang only on chordal graphs and hence a somewhat simpler version of it would suffice.

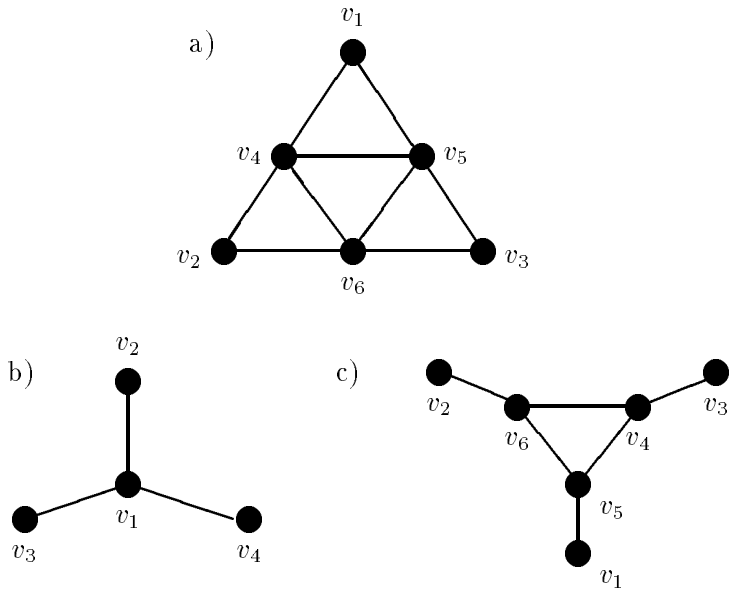


Figure 1: Obstructions for chordal graphs that are not proper interval. a) Tent. b) Claw. c) Net.

4 Strongly Chordal Completion

A chord (v, w) in an even cycle C is *odd* if the paths connecting v and w on C contain an odd number of edges.

The following characterization of strongly chordal graphs is due to Farber [11].

Theorem 4.1 *A graph G is strongly chordal if and only if G is chordal and every even cycle of length at least six in G has an odd chord.* ■

An odd chord in an even cycle C partitions C into two smaller even cycles C_1 and C_2. Any odd chord in C_1 or C_2 is an odd chord in C as well. A *4-cycle decomposition* of an even chordless cycle C is a minimal set T of odd chords in C such that there is no induced even chordless cycle of length at least six in $C + T$.

Next we characterize and count the number of minimal 4-cycle decompositions of an even cycle C. Let $|C| = n$.

The proof of the following lemma is straightforward by induction.

Lemma 4.2 *A minimal 4-cycle decomposition T of an even n-cycle C consists of $(\frac{n}{2} - 2)$ chords. It partitions C into $(\frac{n}{2} - 1)$ 4-cycles. Every two of these 4-cycles are either disjoint or share a chord. Every chord is shared by exactly two 4-cycles.* ■

A *ternary tree* is a tree in which each internal node has three children. The following theorem establishes a correspondence between the set of 4-cycle decompositions of an even n-cycle and the set of ternary trees with $n - 1$ leaves and $\frac{n}{2} - 1$ internal nodes. This correspondence is similar to the one stated in Lemma 2.2 between minimal triangulations of a chordless n-cycle and binary trees with $n - 1$ leaves.

Lemma 4.3 *The number of 4-cycle decompositions of an even n-cycle C is equal to the number of ternary trees with $\frac{n}{2} - 1$ internal nodes.*

Proof For every even n-cycle C construct an invertible mapping from the set of 4-cycle decompositions of C to the set of ternary trees with $\frac{n}{2} - 1$ internal nodes, as follows. The construction is by induction on the length of the cycle. Assume that one has constructed an invertible mapping for every cycle C' where $|C'| \leq n-2$. Let C be an n-cycle and let e be a fixed edge on C. Let T be a 4-cycle decomposition of C, and let $C_e = \{e, e_1, e_2, e_3\}$ be the 4-cycle in

$C + T$ which includes e. If e_i, $i \in \{1,2,3\}$ is a chord, let C_i be the cycle $C - C_e + \{e_i\}$. The 4-cycle decomposition T induces a 4-cycle decomposition T_i of C_i. The tree which corresponds to T has a root (associated with the edge e); the i-th child of the root is a leaf if $e_i \in C$ or the root of the ternary tree which corresponds to T_i under the mapping associated with C_i if e_i is a chord. It is straightforward to verify that the mapping defined above is indeed invertible. ■

Denote the number of ternary trees with n internal nodes by t_n. The value t_n satisfies the following recurrence: $t_0 = 1$ and $t_n = \Sigma_{\{i+j+k=n-1\}} t_i t_j t_k$ if $n \geq 1$. According to Graham, Knuth and Patashnick [18, p. 349] the solution to this recurrence is

$$t_n = \binom{3n+1}{n} \frac{1}{3n+1}$$

which is no greater than $2^{3n} = 8^n$. Together with Lemma 4.3 we obtain

Lemma 4.4 *The number of 4-cycle decompositions of an even n-cycle C is no greater than $8^{\frac{n}{2}-1}$.* ■

4.1 Finding an even cycle without odd chords

The *neighborhood matrix* of a graph is a symmetric 0-1 matrix with rows and columns indexed by the set of vertices of the graph and with an entry of 1 if and only if the corresponding two vertices are equal or adjacent in the graph. A *doubly lexical ordering* of a matrix is an ordering of the rows and of the columns so that the rows, as vectors, are lexically increasing and the columns, as vectors, are lexically increasing. *Lexical ordering* of vectors is the standard dictionary ordering, except that vectors will be read from highest to lowest coordinate. Thus row vectors will be compared from right to left, and column vectors from bottom to top. A matrix M is *symmetric* if its rows and columns are indexed by the same set S and $M(s,t) = M(t,s)$ for all $s, t \in S$. A *symmetric ordering* of such an M is an ordering of S. It is not true that every symmetric matrix has a symmetric doubly lexical ordering. But it was proved by Lubiw [25] that a symmetric matrix that has a *dominant diagonal*, meaning that $M(s,s) \geq M(s,t)$ for all $s, t \in S$, has a symmetric doubly lexical ordering. In particular, the neighborhood matrix of any graph has a symmetric doubly lexical ordering.

A *cycle* matrix is a 0-1 $n \times n$ matrix, $n \geq 3$, with exactly two 1's in each row and in each column and such that no proper submatrix has this property. A *totally balanced matrix* is a 0-1 matrix with no cycle submatrices.

Farber [11] proved the following characterization of strongly chordal graphs.

Theorem 4.5 *A graph is strongly chordal if and only if its neighborhood matrix is totally balanced.*

A Γ is an ordered 0-1-valued 2×2 matrix with exactly one 0, in the bottom right corner:

$$\Gamma = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Lubiw proved the following property[25, 5.2] of a doubly lexical 0-1 matrix M with rows R and columns C.

Theorem 4.6 *Let M be an ordered doubly lexical 0-1 matrix with rows R and columns C. Any 2×2 submatrix of M formed by $r_1 < r_2 \in R$ and $c_1 < c_2 \in C$ with $M(r_1, c_2) = M(r_2, c_1) = 1$, $M(r_2, c_2) = 0$ is, for some $k \geq 3$, embedded in a $k \times k$ submatrix of M formed by $r_1 < r_2 < \ldots < r_k \in R$ and $c_1 < c_2 < \ldots < c_k \in C$ with $M(r_i, c_{i+1}) = M(r_{i+1}, c_i) = 1$ for $i = 1, \ldots, k-1$, $M(r_k, c_k) = 1$, and $M(r_i, c_j) = 0$ for other i, j except possibly $i = j = 1$. In particular any Γ submatrix is embedded in a cycle submatrix. See Figure 2.*

	c_1	c_2	c_3	c_4			c_i			c_k
r_1	?	1	0	0						
r_2	1	0	1	0						
r_3	0	1	0	1			0			
r_4	0	0	1	0	.					
				.	.	.				
					.	0	1			
r_i						1	0	.		
			0				.	.	.	
								.	0	1
r_k									1	1

Figure 2: Every Γ submatrix can be embedded in a cycle submatrix.

Together with the observation that in any ordering of a cycle submatrix there is a Γ submatrix, Theorem 4.6 reestablishes the following result.

Theorem 4.7 ([19, 2]) *A 0-1 matrix has a Γ-free ordering if and only if it is totally balanced. Moreover, a doubly lexical ordering of a totally balanced matrix is Γ-free.*

The following theorem makes a link between cycle submatrices in a neighborhood matrix of a graph G and chordless cycles or even cycles without odd chords in G.

Theorem 4.8 *Let M be a neighborhood matrix of a graph G and N a $k \times k$ cycle submatrix of M with rows $r_1 < r_2 < \ldots < r_k$ and columns $c_1 < c_2 \ldots < c_k$. Let $V_N = \{v_l \mid l = r_i$ or $l = c_j, 1 \leq i, j \leq k\}$. Then either the vertices of V_N form an even cycle without odd chords or there exists a subset $C \subseteq V_N$ that induces a chordless cycle.*

Proof If $r_i \neq c_j$ for every $1 \leq i, j \leq k$, V_N clearly forms an even cycle without odd chords. Assume $r_i = c_j$ for some i and j. This implies that $N(i,j) = M(r_i, c_j) = 1$. Let i' be the other row in which column j has a one and j' the other column in which row i has a one. $N(i', j') = 0$ since otherwise we get a contradiction to the fact that N is a cycle submatrix. Thus $r_{i'} \neq c_{j'}$. Among the vertices in V_N, v_{r_i} is adjacent only to $v_{r_{i'}}$ and $v_{c_{j'}}$. These two are not adjacent but there is a path connecting them in $V - \{v_{r_i}\}$. Thus there exists a chordless cycle $C \subseteq V_N$ through v_{r_i}. ■

Let M be a symmetric $n \times n$ neighborhood matrix of a connected graph G with m edges and n vertices. Using Paige and Tarjan's implementation [28] of the algorithm described by Lubiw [25] one can obtain a doubly lexical ordering of M in $O(m \log n)$ time. Lubiw [25] also shows how to search for a Γ submatrix in a doubly lexically ordered M in $O(m)$ time. Given a Γ submatrix in a doubly lexically ordered M, a cycle submatrix that contains it can also be found in $O(m)$ time [25]. According to Theorem 4.8, either the rows and columns of this cycle submatrix induce an even cycle without odd chords, or a subset of them induce a chordless cycle in G. As suggested by the proof of Theorem 4.8 this cycle can be extracted from the cycle submatrix in $O(m)$ time.

4.2 The k-completion algorithm

As in Sections 2.1 and 3 the k-completion algorithm will traverse part of a search tree in which each node corresponds to a supergraph of G. This search tree is defined as follows. The graph G itself corresponds to the root of the tree. In order to generate the children of an internal node x that corresponds to a graph G' one needs to find either a chordless cycle or an even cycle without odd chords in G'. In case a chordless cycle C is found, node x will have a child for each minimal triangulation T of C. If an even cycle without odd chords, C, is found, x will have a child for each 4-cycle decomposition of C. The graph corresponding to a child is obtained by adding the corresponding minimal triangulation or 4-cycle decomposition to G'. If C is a chordless l-cycle, by Lemma 2.3 node x will have at most c_{l-2} children. If C is an even

l-cycle without odd chords then x will have $t_{\frac{l}{2}-1}$ children. Each leaf of the tree corresponds to a strongly chordal supergraph of G. Note that every such supergraph of G that is minimal is represented by at least one leaf.

Remark In the case that a chordless cycle C is found in the graph corresponding to a node x, it will be more efficient to generate a child only for each triangulation T of C such that $C+T$ has no even cycles without odd chords.

One can find a chordless cycle C in a nonchordal graph with m edges and n vertices in $O(m)$ time by using the MCS algorithm described in [36, 37]. An even cycle without odd chords can be found in a chordal graph that is not strongly chordal in $O(m \log n)$ time using Paige and Tarjan's implementation [28] of Lubiw's algorithm [25] as described in Section 4.1. Obviously, one can use Paige and Tarjan's algorithm for both tasks in order to simplify the implementation, while getting some penalty in the performance. The algorithm described in [34] can be easily extended to enumerate all ternary trees with n internal nodes, spending $O(n)$ time for each. Applying Lemma 4.3 one obtains an algorithm that enumerates all 4-cycle decompositions of an even cycle C in $O(|C|)$ time for each. It is straightforward to check that a more involved enumeration procedure that enumerates all minimal strongly chordal triangulations of a chordless even cycle C in $O(|C|)$ time for each could be designed as well, based on the ideas in [34].

The nodes of this search tree that are actually traversed correspond to supergraphs of G with no more than k additional edges. If one such node is a leaf then we have found a strongly chordal supergraph with no more than k additional edges. Otherwise, no such supergraph exists. The proof of the following theorem is analogous to the proof of Theorem 2.4.

Theorem 4.9 *All minimal strongly chordal supergraphs of a graph G with no more than k additional edges can be found in $O(8^{2k} m \log n)$ time.* ■

Remark An alternative implementation that avoids traversing nonchordal children of chordal supergraphs can be designed as described in the remark at the end of Section 3.

Remark For dense matrices, Spinrad describes a faster algorithm which can obtain a doubly lexical ordering in $O(n^2)$ time [35]. Hence the complexity of the algorithm described above can be improved for dense graphs to $O(8^{2k} \min(n^2, m \log n))$ time.

5 Concluding Remarks

We have presented polynomial algorithms for the fixed-parameter version of three graph completion problems: CHORDAL COMPLETION(k), STRONGLY CHORDAL COMPLETION(k) and PROPER INTERVAL COMPLETION(k). Note that the class of proper interval graphs is a subset of the strongly chordal graphs, which are a subset of the chordal graphs.

Another important graph family that we have not discussed in this paper is interval graphs. The INTERVAL COMPLETION(k) problem has an important application in molecular biology, as discussed in Section 1. Its NP-completeness was proved in [22]. NP-completeness is also implied by the proof of Yannakakis [40] for chordal graph completion, as the graphs generated in that proof are chordal if and only if they are interval. To date the complexity status of the parametric version of the problem is open. It is not known whether the problem is in FPT or hard for some level of the W-hierarchy. The obstructions that have to exist in a chordal graph that is not interval are described in [24]. An arbitrarily large obstruction X could exist in a graph that is not interval but could be made interval with the addition of any one out of $O(|X|)$ edges. This causes difficulties when one tries to apply the techniques of this paper to this graph class.

When the input is restricted to bounded-degree interval graphs for some fixed bound d, the obstruction size is bounded by $O(d)$ and the search tree technique applies to get a quadratic FPT result using the characterization of [24]. It is an open problem whether this obvious bound can be improved.

For the molecular biology application in physical mapping, one can assume that the ratio of sizes of the largest and the smallest clones is at most a small constant c (in practice, $c = 10$ suffices). Fishburn and Graham [14] (see also [13, 8.2]) provided characterizations for interval graphs which have such length restrictions. Their results, together with the characterizations of [24], imply that the obstruction size is $O(c)$ and thus for this case too the search tree technique applies and the k-completion problem is FPT.

References

[1] K. Abrahamson, R. Downey, and M. Fellows. Fixed-parameter intractability II. In *Proceedings of the 10th Symposium on Theoretical Aspects of Computer Science (STACS'93)*, Lecture Notes in Computer Science vol. 665, pages 374–385. Springer-Verlag, Berlin, 1993.

[2] R. P. Anstee and M. Farber. Characterizations of totally balanced matrices. *Journal of Algorithms*, 5:215–230, 1984.

[3] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, pages 226–234. ACM Press, New York, 1993.

[4] H. L. Bodlaender, M. R. Fellows, and M. T. Hallet. Beyond NP-Completeness for problems of bounded width: Hardness for the W hierarchy (extended abstract). In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, pages 449–458. ACM Press, New York, 1994.

[5] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms. *J. Comput. Sys. Sci.*, 13:335–379, 1976.

[6] A. V. Carrano. Establishing the order of human chromosome-specific DNA fragments. In A. D. Woodhead and B. J. Barnhart, editors, *Biotechnology and the Human Genome*, pages 37–50. Plenum Press, New York, 1988.

[7] X. Deng, P. Hell, and J. Huang. Linear time representation algorithms for proper circular arc graphs and proper interval graphs. Technical report, School of Computing Science, Simon Fraser University, 1993.

[8] R. G. Downey and M. R. Fellows. Fixed-parameter intractability. In *Proceedings of the Seventh Annual Structure in Complexity Theory Conference (Structures'92)*, pages 36–49, Boston Massachusetts, 1992. IEEE Computer Society Press, Los Alamitos, California.

[9] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness III: Some structural aspects of the W hierarchy. In *Complexity Theory: Current Research (Proceedings of the 1992 Dagstuhl Workshop on Structural Complexity)*, pages 191–226. Cambridge University Press, Cambridge, 1993.

[10] R. G. Downey and M. R. Fellows. Parameterized computational feasibility. In K. Ambos-Spies, S. Homer, and U. Schoning, editors, *Complexity Theory: Current Research*, pages 166–191. Cambridge University Press, New York, 1993.

[11] M. Farber. Characterizations of strongly chordal graphs. *Discrete Math.*, 43:173–189, 1983.

[12] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Appl. Math.*, 7:115–130, 1984.

[13] P. Fishburn. *Interval Orders and Interval Graphs.* Wiley, New York, 1985.

[14] P. Fishburn and R. L. Graham. Classes of interval graphs under expanding length restrictions. *J. Graph Theory*, 9:459–472, 1985.

[15] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $o(1)$ worst case access time. *Journal of the ACM*, 31:538–544, 1984.

[16] A. George and J. W. Liu. *Computer solution of large sparse positive definite systems.* Prentice Hall, Englewood Cliffs, NJ, 1981.

[17] M. C. Golumbic, H. Kaplan, and R. Shamir. On the complexity of DNA physical mapping. *Advances in Applied Mathematics*, 15:251–261, 1994.

[18] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete mathematics : a foundation for computer science*. Addison-Wesley, Reading, Massachusetts, 1989.

[19] A. J. Hoffman, A. W. J. Kolen, and M. Sakarovitch. Totally balanced and greedy matrices. *SIAM J. Algebraic and Disc. Methods*, 6:721–730, 1985.

[20] H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping. In *Proceedings of the 35th Symposium on Foundations of Computer Science*, pages 780–791. IEEE Computer Science Press, Los Alamitos, California, 1994.

[21] R. M. Karp. Mapping the genome: some combinatorial problems arising in molecular biology. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, pages 278–285. ACM Press, New York, 1993.

[22] T. Kashiwabara and T. Fujisawa. An NP-complete problem on interval graphs. In *IEEE International Symposium on Circuits and Systems (12th)*, pages 82–83. Institute of Electrical and Electronics Engineers; Piscataway, N.J., 1979.

[23] T. Kloks. *Treewidth*. PhD thesis, Dept. of Computer Science, Utrecht University, 1993.

[24] C. G. Lekkerkerker and J. Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundam. Math.*, 51:45–64, 1962.

[25] A. Lubiw. Doubly lexical ordering of matrices. *SIAM J. Computing*, 16:854–879, 1987.

[26] R. Nagaraja. Current approaches to long-range physical mapping of the human genome. In R. Anand, editor, *Techniques for the Analysis of Complex Genomes*, pages 1–18. Academic Press, London, 1992.

[27] T. Ohtsuki, L. K. Cheung, and T. Fujisawa. Minimal triangulation of a graph and optimal pivoting order in a sparse matrix. *Journal of Math. Anal. Appl.*, 54:622–633, 1976.

[28] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM J. Computing*, 16(6):973–989, 1987.

[29] F. S. Roberts. Indifference graphs. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, New York, 1969.

[30] D. J. Rose. Triangulated graphs and the elimination process. *J. Math. Anal. Appl.*, 32:597–609, 1970.

[31] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination of graphs. *SIAM J. Computing*, 5:266–283, 1976.

[32] J. D. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In R. C. Reed, editor, *Graph Theory and Computing*, pages 183–217. Academic Press, N.Y., 1972.

[33] D. D. Sleator, R. E. Tarjan, and W. P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *Journal of the AMS*, 1(3):647–681, 1988.

[34] M. Solomon and R. A. Finkel. A note on enumerating binary trees. *Journal of the ACM*, 27:3–5, 1980.

[35] J. Spinrad. Doubly lexical ordering of dense 0-1 matrices. *Inf. Proc. Letts.*, 45:229–235, 1993.

[36] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, text acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Computing*, 13:566–579, 1984.

[37] R. E. Tarjan and M. Yannakakis. Addendum: Simple linear-time algorithms to test chordality of graphs, text acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Computing*, 14:254–255, 1985.

[38] M. N. Wegman and J. L Carter. New classes and applications of hash functions. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pages 175–182. IEEE Computer Society Press, Los Alamitos, California, 1979.

[39] G. Wegner. *Eigenschaften der nerven homologische eihfacher familien in* R^n. PhD thesis, Götingen, 1967.

[40] M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2, 1981.

Isolating critical cases for reciprocals using integer factorization

John Harrison
Intel Corporation, JF1-13
2111 NE 25th Avenue
Hillsboro OR, USA
johnh@ichips.intel.com

Abstract

One approach to testing and/or proving correctness of a floating-point algorithm computing a function f is based on finding input floating-point numbers a such that the exact result $f(a)$ is very close to a "rounding boundary", i.e. a floating-point number or a midpoint between them. In the present paper we show how to do this for the reciprocal function by utilizing prime factorizations. We present the method and show examples, as well as making a fairly detailed study of its expected and worst-case behavior. We point out how this analysis of reciprocals can be useful in analyzing certain reciprocal algorithms, and also show how the approach can be trivially adapted to the reciprocal square root function.

1 Background

Suppose we have a floating-point algorithm computing a function that approximates a true mathematical function $f : \mathbb{R} \to \mathbb{R}$. For example, consider the following algorithm for the Intel® Itanium® architecture designed to compute a floating-point square root $\sqrt{a}$ using an initial reciprocal square root approximation followed by a sequence of fused multiply-adds. (In the actual implementation, the initial approximation instruction deals with special cases including $a = 0$.)

1.	$y_0 = \texttt{frsqrta}(a)$	
2.	$H_0 = \frac{1}{2} y_0$	$S_0 = a y_0$
3.	$d_0 = \frac{1}{2} - S_0 H_0$	
4.	$H_1 = H_0 + d_0 H_0$	$S_1 = S_0 + d_0 S_0$
5.	$d_1 = \frac{1}{2} - S_1 H_1$	
6.	$H_2 = H_1 + d_1 H_1$	$S_2 = S_1 + d_1 S_1$
7.	$d_2 = \frac{1}{2} - S_2 H_2$	$e_2 = a - S_2 S_2$
8.	$H_3 = H_2 + d_2 H_2$	$S_3 = S_2 + e_2 H_2$
9.	$e_3 = a - S_3 S_3$	
10.	$S = S_3 + e_3 H_3$	

If an algorithm is, like this one, implemented by composing basic floating-point operations (rather than, say, some more complicated analysis of bit-patterns), then the value computed can usually be represented as the result of rounding some approximation $f^*(x) \approx f(x)$, the value before the final rounding. In this case, the final S results from rounding the exact value $S_3 + e_3 H_3$.

The algorithm will therefore round correctly for all inputs x such that $f^*(x)$ and $f(x)$ round to the same number (for all the rounding modes under consideration). In the concrete square root example, this means that $\sqrt{a}$ and $S_3 + e_3 H_3$ should always round the same way.

A sufficient condition for equivalent rounding behavior is that the two values $f^*(x)$ and $f(x)$ should never be separated by a rounding boundary, i.e. a floating-point number (for directed rounding) or a midpoint (for round-to-nearest). That is, there is never a rounding boundary m with $f(x) \leq m \leq f^*(x)$ or $f^*(x) \leq m \leq f(x)$, unless $f^*(x) = f(x)$. (Not quite a necessary condition in the round-to-nearest mode since if one is exactly equal to the rounding boundary and the other on the "right" side, the correct result will be obtained.) This is usually hard to establish by analytic reasoning. However, it is usually easy to establish some sort of relative error bound ϵ such that:

$$|f^*(x) - f(x)| \leq \epsilon |f(x)|$$

Therefore, misrounding can occur only when

$$|f(x) - m| \leq \epsilon |f(x)|$$

It is therefore interesting for purposes of both testing and proving correctness to deliberately concoct test points x to make the relative distance from a rounding boundary $|f(x) - m|/|f(x)|$ as small as possible. Indeed, irrespective of the details of the algorithms we are concerned with, these test points might be expected to display greatest sensitivity to the accuracy of $f^*(x)$ and so show up errors most easily.

For some basic algebraic functions, such special x can be found analytically using number-theoretic techniques [14, 11], in such a way that the very worst examples (having the smallest relative distance from a rounding boundary) are isolated. For transcendental functions, this is more difficult, but one can still generate good cases by exploiting local linearity and solving congruences. For double-precision it is feasible, though costly, to isolate the very worst examples [6].

One use of the points so obtained is to test floating-point functions. Indeed, Parks [11] reports that such testing exposed a bug in a commercial microprocessor. A more ambitious goal, realized for square root algorithms by Cornea [1], is to isolate a sufficiently large set of points that the correct behavior of the algorithm on these, in conjunction with an analytical proof that covers all other cases, gives a complete correctness proof of the algorithm in all cases. For example, if we can prove analytically that for all floating-point numbers x we have:

$$|f^*(x) - f(x)| \leq \epsilon |f(x)|$$

and that some set S_ϵ contains all points x where $|m - f(x)| \leq \epsilon |f(x)|$ for some rounding boundary m, the correctness of the algorithm in all cases is equivalent to the correctness just for the points in S_ϵ. If such sets can be found easily and they are not too large, this gives a very effective methodology for proofs of algorithms. The goal of this paper is to show how to isolate such special cases for the reciprocal (and reciprocal square root) function and demonstrate their applicability in such correctness proofs of algorithms.

2 Critical cases for quotient and reciprocal

We will in what follows consider a single floating-point format with precision p, which contains all the floating-point numbers concerned and is also the destination format for the result. We also ignore the possibility of overflow and underflow in computation sequences. This keeps the presentation simpler and accords well with the intended applications where all input numbers are double-extended and additional exponent range (but not precision) is available for intermediate computations. The results that follow can straightforwardly be refined for mixed-precision applications.

It's instructive to examine the problem for the general case of quotients, and then contrast the restriction to the reciprocal. In general, we seek floating-point numbers x and y such that x/y lies close to some w that is either itself a floating-point number or a midpoint between two floating-point numbers. Without loss of generality, we can assume:

$$\begin{array}{ll} x = 2^{e_x} a & 2^{p-1} \leq a < 2^p \\ y = 2^{e_y} b & 2^{p-1} \leq b < 2^p \\ w = 2^{e_w} m & 2^p \leq m < 2^{p+1} \end{array}$$

where p is the floating-point precision and a, b and m, as well as the various e_i, are integers. Note that even values of m correspond to floating-point numbers and odd values correspond to midpoints. We are interested in how small the relative difference $|x/y - w|/|x/y|$ can become. This relative difference can be rewritten as:

$$\frac{|x/y - w|}{|x/y|} = |1 - wy/x| = |1 - 2^{-q} mb/a|$$

where $q = e_x - (e_w + e_y)$, and so

$$\frac{|mb - 2^q a|}{2^q a}$$

Given the ranges of the values a, b and m, we have

$$2^{2p-1} \leq mb < 2^{2p+1}$$

and

$$2^{q+p-1} \leq 2^q a < 2^{q+p}$$

It turns out that the only interesting cases are when $q = p$ or $q = p + 1$. For if $q \leq p - 1$ then $q + p \leq 2p - 1$ so we have

$$2^q a \leq 2^q (2^p - 1) < 2^{2p-1} \leq mb$$

(remember that the values a, b and m are integers so when $< 2^r$ they are actually $\leq 2^r - 1$) and so

$$\frac{|mb - 2^q a|}{2^q a} \geq 2^q / (2^q a) = 1/a > 2^{-p}$$

.

Similarly if $q = p + 2$ we have:

$$mb \leq (2^p - 1)(2^{p+1} - 1) < 2^{2p+1} \leq 2^{p+2} a < 2^{2p+2}$$

and therefore

$$\frac{|mb - 2^q a|}{2^q a} \geq (2^{p+1} + 2^p - 1)/(2^q a) > 2^{p+1}/2^{2p+2} = 2^{-(p+1)}$$

Finally, if $q \geq p + 3$ then $2^q a > 2mb$ and so

$$\frac{|mb - 2^q a|}{2^q a} > 1/2$$

In all these cases, the distance is at least $2^{-(p+1)}$. Therefore, when seeking cases where the distance is of order 2^{-2p} (for realistic p) we need only consider $q \in \{p, p + 1\}$. This

being the case, the denominator $2^q a$ is constrained to within a factor of 4, so the essential problem is to find how small

$$|mb - 2^q a|$$

can become for $q \in \{p, p+1\}$. Since the value is an integer, we can try to find small values by explicit consideration of the various possibilities in succession:

$$\begin{aligned} mb &= 2^p a + 1 \\ mb &= 2^p a - 1 \\ mb &= 2^{p+1} a + 1 \\ mb &= 2^{p+1} a - 1 \\ mb &= 2^p a + 2 \\ mb &= 2^p a - 2 \\ mb &= 2^{p+1} a + 2 \\ mb &= 2^{p+1} a - 2 \\ mb &= 2^p a + 3 \\ &\dots \end{aligned}$$

It seems that the number of possible solutions of these equations is too large for this to be a practical approach. On the other hand, if we fix any one of the values a, b and m, the problem becomes tractable. If we fix either m or b then the problem becomes a set of linear congruences (with additional range restrictions filtering the possible solution set), which are easy to solve. If we consider the special case of the reciprocal, then we fix $a = 2^{p-1}$. This problem is also tractable, as we shall see, but has a somewhat different character. We just need to consider

$$\begin{aligned} mb &= 2^{2p-1} + \delta \\ mb &= 2^{2p} + \delta \end{aligned}$$

for successive small integers δ. In fact, the situation is even better, because once again no small values can arise in the former case because of the range limitation, except for the trivial $mb = 2^{2p-1}$; the next case must be $(2^p + 1)2^{p-1} = 2^{2p-1} + 2^{p-1}$. So we need only be concerned with solutions to

$$mb = 2^{2p} + \delta$$

for integers $2^{p-1} \le b < 2^p$ and $2^p \le m < 2^{p+1}$. Indeed, for small δ, it is easy to see that the two upper bounds imply the lower ones.

3 Factorization distribution

Our approach to the problem of finding all solutions to $mb = 2^{2p} + \delta$ (with p and δ fixed) is quite straightforward. We find the prime factorization of $2^{2p} + \delta$, and consider all possible ways of distributing these prime factors into two parts m and b subject to the appropriate range limitation $m < 2^{p+1}$ and $b < 2^p$. In general, we will refer to a factorization $n = ab$ of n with $a < A$ and $b < B$ as an (A, B)-*balanced* factorization.

Consider, for illustration, the case $p = 6$ and $\delta \in \{\pm 1, \pm 2, \pm 3\}$. In each case we find the prime factorization of $2^{2p} + \delta$:

$$\begin{aligned} 2^{12} + 1 &= 17 \cdot 241 \\ 2^{12} - 1 &= 3^2 \cdot 5 \cdot 7 \cdot 13 \\ 2^{12} + 2 &= 2 \cdot 3 \cdot 683 \\ 2^{12} - 2 &= 2 \cdot 23 \cdot 89 \\ 2^{12} + 3 &= 4099 \\ 2^{12} - 3 &= 4093 \end{aligned}$$

In the cases $2^{12}+1$, $2^{12}+2$, $2^{12}+3$ and $2^{12}-3$, the largest factor is already $> 2^{p+1} = 128$, so there is no possible distribution obeying the range restrictions. For $2^{12} - 2$ there is exactly one such distribution:

$$m \cdot b = 89 \cdot (2 \cdot 23) = 89 \cdot 46$$

Note that the 'symmetrical' distribution is not admissible because $89 > 2^p$. For $2^{12} - 1$, there are four possible distributions:

$$\begin{aligned} m \cdot b &= (3^2 \cdot 13) \cdot (5 \cdot 7) = 117 \cdot 35 \\ m \cdot b &= (3 \cdot 5 \cdot 7) \cdot (3 \cdot 13) = 105 \cdot 39 \\ m \cdot b &= (7 \cdot 13) \cdot (3^2 \cdot 5) = 91 \cdot 45 \\ m \cdot b &= (5 \cdot 13) \cdot (3^2 \cdot 7) = 65 \cdot 63 \end{aligned}$$

Note that the corresponding m are all odd, and therefore represent midpoints. Thus, we can say that $|1/y - w| \ge 4/2^{12}|1/y|$ for any midpoint w except in the cases where y's significand b is in the set $\{35, 39, 45, 46, 63\}$; for $b = 46$ we get a $2/2^{12}$ relative distance and for 35, 39, 45 and 63 we get $1/2^{12}$. Since the above lists exhausts all m, even or odd, we see that $|1/y - w| \ge 4/2^{12}|1/y|$ for any floating-point number w, except for the special cases when y is a power of 2 and so its reciprocal is exactly representable (i.e. $1/y = w$).

4 Implementation

The implementation of the above idea is straightforward, given any reasonable programming language. We have used Objective CAML, a very high-level functional language that we have previously used extensively for implementation of theorem proving code:

`http://www.ocaml.org/`

This already has a multiprecision integer and rational function datatype available. It does not, however, have a built-in library for factoring numbers, and we did not want to write our own code for this operation — since the numbers can be as large as 2^{226} (for quad precision reciprocals), factorization is a non-trivial problem. We used the factoring code included in the PARI / GP system:

`http://www.parigp-home.de/`

The documentation says:

> **factorint**$(n, \{flag = 0\})$: factors the integer n using a combination of the Shanks SQUFOF and Pollard Rho method (with modifications due to Brent), Lenstra's ECM (with modifications by Montgomery), and MPQS (the latter adapted from the LiDIA code with the kind permission of the LiDIA maintainers), as well as a search for pure powers with exponents≤ 10.

We are not experts in the topic of factorization, but have been quite impressed with how fast it usually factors numbers. Only for quad precision, when the numbers are of the order 2^{226}, does it start to slow down noticeably. Rather than a strict primality test, the factors are only subjected to a strong probabilistic primality test. Therefore, out of paranoia, we have developed our own code to certify primality, by constructing prime certificates in the style of Pratt [12], appealing to Lucas's theorem. That is, to certify that each p occurring in PARI/GP's factorization is prime, we show that there is a primitive root a modulo p such that $a^{p-1} \equiv 1 \pmod{p}$ but $a^{\frac{p-1}{q}} \not\equiv 1 \pmod{p}$ for any prime factor q of $p-1$. (The primitive root a is found randomly, and the factors q of $p-1$ are found by using PARI/GP's factorization recursively, certifying those factors as primes too.) This certification slows down the factorization process by a moderate amount, so we sometimes switch it off when experimenting.

Once we have the prime factors, we need to test all ways of distributing them over two numbers subject to range restrictions. As noted, we need only apply the upper range restrictions $m < 2^{p+1}$ and $b < 2^p$. Roughly, we just naively enumerate all possibilities. In order to cut off choice points as soon as possible, we start distributing from the largest prime factors, i.e. consider the prime factors $p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \cdots \cdot p_k^{\alpha_k}$ in decreasing order $p_1 > p_2 > \cdots > p_k$. We first consider all $\alpha_1 + 1$ ways of distributing $p_1^{\alpha_1}$ into two parts. If any of these distributions already violate the range restriction, they are abandoned. Otherwise, for each one, we consider the $\alpha_2 + 1$ ways of distributing $p_2^{\alpha_2}$, and so on. The algorithm is very straightforward to program recursively in OCaml.

It might be doubted whether such a naive distribution algorithm is acceptably efficient. At least it has been adequate to obtain some results quite quickly for the main precisions that interest us, $p \in \{24, 53, 63, 113\}$. We first look at some of these results and then turn to a detailed performance analysis.

5 Results

Table 1 presents a small sample of the results obtained using the methods outlined above. For each of the four major precisions $p = 24, 53, 64, 113$, we list the 66 floating-point significands whose reciprocals are closest either to floating-point numbers or midpoints. This distance, as a multiple of the corresponding 2^{-2p}, is given in the 'd' columns. When, as often happens, several reciprocals have the same 'd' value we order them in decreasing order, and cut the table off on that basis. The asterisk means that the distance is from a floating-point number (and hence may be unimportant if we are concerned only with round-to-nearest).

Larger lists for d up to a few thousand can be generated for all these precisions without requiring more than a few days of runtime on a modern machine. And of course, it is trivial to parallelize the task since it consists of a separate subtask for each d considered.

6 Applications

We can use the techniques set out above in the design and verification of algorithms for correctly rounded reciprocals. These might be substituted by the programmer, or by the compiler if it can recognize that in an expression a/b, the constant a is guaranteed to be 1. (This could be generalized to any power of 2.) For example, the following algorithm is normally used for double-extended precision division (precision $p = 64$) on Intel® Itanium® processors.

1.	$y_0 = \texttt{frcpa}(b)$	
2.	$d = 1 - by_0$	$q_0 = ay_0$
3.	$d_2 = dd$	$d_3 = dd + d$
4.	$y_1 = y_0 + y_0 d_3$	$d_5 = d_2 d_2 + d$
5.	$y_2 = y_0 + y_1 d_5$	$r_0 = a - bq_0$
6.	$e = 1 - by_2$	$q_1 = q_0 + r_0 y_2$
7.	$y_3 = y_2 + ey_2$	$r = a - bq_1$
8.	$q = q_1 + ry_3$	

Single precision		*Double precision*		*Extended precision*		*Quad precision*	
Mantissa	d	Mantissa	d	Mantissa	d	Mantissa	d
0x800000	0^*	0x10000000000000	0^*	0x8000000000000000	0^*	0x10000000000000000000000000000	0^*
0xFFFFFF	1	0x1FFFFFFFFFFFFF	1	0xFFFFFFFFFFFFFFFF	1	0x1FFFFFFFFFFFFFFFFFFFFFFFFFFF	1
0xFE01FF	1	0x1FFFFFF8000001	1	0xD6329033D6329033	1	0x1FFFFFFFFFFFFFE00000000000001	1
0xFC3237	1	0x1FD8CD299E8D79	2^*	0xB7938C6947D97303	1	0x1B52F1BB6F8DC3F0D920E2F3D449B	1
0xF0FF0F	1	0x1FC94266515BC9	2^*	0x99D0C486A0FAD481	1	0x19C1ECF3420D27F8729BA7E1AB31D	1
0xF02A3B	1	0x1F739BD459BEA2	2	0x989E556CADAC2D7F	1	0x17ABDE305BAC595488190B4AD7657	1
0xF00FF1	1	0x1F65FAD23B0D86	2	0x8E05E117D9E786D5	1	0x14367E6C7D1CD9E2833D2900EE8D5	1
0xEE4BC5	1	0x1EF7930608393E	2	0xFFFFFFFFFFFFFFFE	4^*	0x1FFBA28E4810FB56A9FDD85058227	2^*
0xEC7EC7	1	0x1EDA43AEE3120F	2^*	0xFFFFFFFE00000002	4^*	0x1FF0231E35F73DFF14F89AADF10C2	2
0xE25473	1	0x1ED31F284BA183	2^*	0xFFFFF000007FFFFE	4^*	0x1FE5A1913A4EF66DEF762D8053282	2
0xE1368B	1	0x1E9A9473949BF6	2	0xFF801FFA00FFE002	4^*	0x1FE1696E4EFFB6A84655C0D432D92	2
0xE05475	1	0x1E8D517D09C5C2	2	0xFF007FC01FF007FE	4^*	0x1FDA070427995BB524AB4B13DC457	2^*
0xDE86A9	1	0x1E756F08DF1792	2	0xFE421D63446A3B34	4	0x1FAA42B2AE532A32F819FE18EDEAF	2^*
0xDC23DD	1	0x1E4599DD926B71	2^*	0xFDC1EAD583108905	4^*	0x1F97117BE0A19F4B8279CEBB8A682	2
0xD43D43	1	0x1E20ADBC4078A2	2	0xFC41DF1077C41DF3	4^*	0x1F76B18346B7182CE92732C773FB2	2
0xD25D25	1	0x1DE4A0D00FA9B2	2	0xFC07FFE03FFF01FE	4^*	0x1F742DB89E4A0B81D5A2FE647E4EB	2^*
0xD0DD0D	1	0x1DE441D5331432	2	0xFC07F01FC07F01FE	4^*	0x1F490212CC8000000003E92042599	2^*
0xD0AC19	1	0x1DA83EEDD80267	2^*	0xFBFC17DFE0BEFF04	4	0x1F40436566B31BF75C99DF44F291F	2^*
0xC23DC3	1	0x1DA210DAEB138E	2	0xFB20F95555168D17	4^*	0x1F361A9D498B669732A60AFCF9461	2^*
0xC100C1	1	0x1D7F8AC20F7A3F	2^*	0xFB0089D7241D10FC	4	0x1F25136B121FE2DD08B9975B8DBD2	2
0xB84A93	1	0x1D7B72B82BAE23	2^*	0xFA0BF7D05FBE82FC	4	0x1F1DFB37ABDE94B6800C5550CD152	2
0xAD1367	1	0x1D5B9032F086BE	2	0xF98AF433A85E62BF	4^*	0x1F182E16A52503DCEDEBC24CA2B4E	2
0xAB8BE1	1	0x1D5616F7BA44B9	2^*	0xF96BA24DC930852A	4^*	0x1F140333F6B5946A06272DBD508B7	2^*
0xA6449F	1	0x1C8ECFA282734B	2^*	0xF93AB02081C9D1D6	4^*	0x1F0DD51725F05CC5C752AA05A4311	2^*
0xA24CF7	1	0x1C69BF28EBA166	2	0xF912590F016D6D04	4	0x1F001BEA0DE009CE597A0A8CE4B02	2
0xA0DDD1	1	0x1C67CF42F20D11	2^*	0xF858A9FE5A20550D	4^*	0x1EC36516E1240A243EF66232D4BA7	2^*
0x9BEAAF	1	0x1C4D3AABD478F6	2	0xF84CEE8E701FC266	4^*	0x1EBC4C4507F9CE8304761C8F703D2	2
0x986799	1	0x1C2693DCF34742	2	0xF774DD7F912E1F54	4	0x1E9EDBD047D1D813FB315AB469B2E	2
0x909909	1	0x1BEA3278B789D2	2	0xF7444DFBF7B20EAC	4	0x1E89A9332E4A8C84E2AA22A6DF7F1	2^*
0x8EFA43	1	0x1BB2278C9B2F97	2^*	0xF6F0243D8121FB7A	4^*	0x1E8119576512C73436A03607DCB9B	2^*
0x87CC45	1	0x1B962F9EBB9659	2^*	0xF6640F754B4E709A	4^*	0x1E765D90D920CEEEBD7F5E0E0BBA9	2^*
0x869913	1	0x1B227794E85702	2	0xF39EB657E24734AC	4	0x1E5DF4E7C4BB29C00588956CF0009	2^*
0xCA6691	2^*	0x1B0FD7099EB189	2^*	0xF36EE790DE069D54	4	0x1E5879735065906E472C4A72A35CF1	2^*
0xA1E58F	2^*	0x1B0942AAAE0BD3	2^*	0xF363A464E2DCD8EB	4^*	0x1E40DECFCF36257C367CACDAD3F77	2^*
0xFFFFFE	4^*	0x1AE6849E786AD2	2	0xF286AD7943D79434	4	0x1E2BE9D384CE2D85FD8013E21ECF2	2
0xFFE002	4^*	0x1ABBEB8E009CE1	2^*	0xEF9DA1D868469215	4^*	0x1E15BB4DD7AA987E15487C533C649	2^*
0xAAAAAC	4	0x1AA7C88EE59082	2	0xEDF09CCC53942014	4	0x1E0C9181ECD8418355A1A49887852	2
0x8C1284	4	0x1A6F41DAB98CB2	2	0xEDE957FFFFC485AA	4^*	0x1E0697C8651B43A9309DE9E6F021E	2
0x801001	4^*	0x1A2CE4D7478A06	2	0xEDE95090B57B7A56	4^*	0x1DEE59C5D9CC8CB8613C8C6AD453F	2^*
0x800001	4^*	0x1A149BAD85DE72	2	0xEDBAA0922AFB6EAA	4^*	0x1DBA39CE33CA8DEF599F5DA2A534F	2^*
0x94F105	5	0x1A0E795098FF63	2^*	0xEC4B058D0F7155BC	4	0x1DAC85098ABA5E144E44187FB5467	2^*
0x92ABAB	5	0x1A0B8FFFFCBE8E	2	0xEC1CA6DB6D7BD444	4	0x1D99C15392893EA6B5200AB3E8819	2^*
0xE401C8	8	0x19F142D24E1352	2	0xEB443F5A21FAD10E	4^*	0x1D983DBB99EAC626064F81D7BF4D2	2
0xE071F9	8^*	0x19BD2D9FD24AD7	2^*	0xEA6EE2D972746ED1	4^*	0x1D86CC4938A03D4525C152AB8505E	2
0xD6D764	8^*	0x19B8D7C084EE43	2^*	0xEA40E197842DA6AF	4^*	0x1D573D7B5CFE2D277AD5E05BAC65E	2
0xD443F2	8^*	0x199E1B447E99C2	2	0xE934A8E070ACB65D	4^*	0x1D4D79E1F152354E10F583D4A65C9	2^*
0xCFBA38	8	0x19939800033273	2^*	0xE84BDA12F684BDA3	4^*	0x1D4CF86C34F75247D8FA16202FA29	2^*
0xB5C2F1	8^*	0x1975E059B82E49	2^*	0xE775FF856986AE74	4	0x1D4562A76F879A38EEE86F526D231	2^*
0xB447BC	8^*	0x1960A45D1A71E6	2	0xE6944AE6502F8A22	4^*	0x1D3168C71EC1068F69A433D0DF9B7	2^*
0xAE4F88	8	0x19385F4F83B2B1	2^*	0xE5CB972E5CB972E4	4	0x1D26EDCA8F70B604EC3E7797A93A1	2^*
0x9A5F6E	8^*	0x190759A7F39561	2^*	0xE597116BD81B26A3	4^*	0x1D16B51A196CFFF2477078355A9AE	2
0x988597	8^*	0x18F187FFFCE1CF	2^*	0xE58C38D1342FBE3A	4^*	0x1D12CF093CC27703DA21DC7D68CE7	2^*
0x91FEDC	8^*	0x18DFA37A569E47	2^*	0xE58469F0234F72C4	4	0x1CEC5ADBF01E9685CD487AD8F3327	2^*
0xFFFFFD	9	0x1879574AF5FBB1	2^*	0xE511C4648E2332C4	4	0x1CE72221273FE0035FEC64CBB3DBF	2^*
0xA013D1	11	0x184A12EFEF626E	2	0xE3FC771FE3B8FF1C	4	0x1CE4C2D686D170738B75E2AFECF3E	2
0xF56DA7	12^*	0x18401CBCDB5596	2	0xE3C845B18BD25EC6	4^*	0x1CD0C2468D84F6ACF871D5E1FCBA9	2^*
0x858376	12^*	0x181EFE51EAD722	2	0xE318DE3C8E6370E4	4	0x1CCB50FE42CD1B95A59CA8AD6EB99	2^*
0xCF7D05	13	0x1806C89FCB9452	2	0xE301201062C997DE	4^*	0x1CABCCB01B54CE7E2A63A99B9D9C2	2
0xEE5223	14^*	0x17F52093014F0E	2	0xE23B9711DCB88EE4	4	0x1C9ED60CFD93F55F117571C3FDA0E	2
0xE528AB	14^*	0x17D93736C115FF	2^*	0xE231188C46231187	4^*	0x1C820E19C1A66CE04C62A562E9111	2^*
0xBF3621	14^*	0x17A6B0778D60C1	2^*	0xE1F00785C1FF0F82	4^*	0x1C673D52FCD6E005D2A2B3D40EDCF	2^*
0xAB5ED9	14^*	0x178CB7D5D6E322	2	0xE159BE4A8763011C	4	0x1C670773DF1678DF0A4336D3FE21E	2
0x8EFE15	14^*	0x17641C46F799EE	2	0xE0A72F05397829CA	4^*	0x1C4D4290F01337DEE39B3A7862BDE	2
0x897ECD	14^*	0x175D929C3C2FC9	2^*	0xE0A720FAC6F829CA	4^*	0x1C4BD3136A6DB6DB6DB351F3546E2	2
0xFFFFFC	16^*	0x1733B8284238F1	2^*	0xE073C0EE938231F9	4^*	0x1C49777FF62E0B0DA1A4CDB58587F	2^*
0xF83F04	16^*	0x17255CA25B68E1	2^*	0xDF738B7CF7F482E4	4	0x1C4814DFE06ECB4EB88D00BA934F2	2

Table 1. Some numbers with reciprocals closest to numbers(*) and midpoints

As usual in algorithms of this kind, each operation uses a fused multiply-add (*not* a separate multiplication and addition), all steps but the last are performed in round-to-nearest mode with additional exponent range precluding the possibility of intermediate overflow or underflow, and the last operation is done in the intended rounding mode and target precision.

Embedded in this algorithm is the computation of a very accurate reciprocal approximation y_3. Originally, in the design of algorithms of this kind, the correctness of the final rounding of q was justified by a theorem whose precondition requires perfect rounding of y_3 [9], and only later was it noted by the present author that a relative error $y_3 = \frac{1}{b}(1+\epsilon)$ for $|\epsilon| < 2^{-p}$ suffices, which can be satisfied by a relatively weak error condition on y_2 and the analysis of a few special cases [3, 8]. However, if we are in a situation where $a = 1$ we might consider, instead of using the entire sequence, unpicking the algorithm for reciprocation to be used directly, since its latency is shorter by 1 operation, and it uses only 9 floating-point operations instead of 14:

1. $y_0 = \texttt{frcpa}(b)$
2. $d = 1 - by_0$
3. $d_2 = dd$ $\qquad$ $d_3 = dd + d$
4. $y_1 = y_0 + y_0 d_3$ $\qquad$ $d_5 = d_2 d_2 + d$
5. $y_2 = y_0 + y_1 d_5$
6. $e = 1 - by_2$
7. $y = y_2 + ey_2$

Now the question of whether y is always correctly rounded becomes critical. First we will consider round-to-nearest. The initial approximation returned by `frcpa` will satisfy $y_0 = \frac{1}{b}(1+\epsilon_0)$ for some $|\epsilon_0| \leq 2^{-8.886}$. A routine relative error analysis, assuming each rounding $rn(x)$ yields $x(1+\epsilon)$ for some $|\epsilon| \leq 2^{-64}$, shows that y^*, the value of y before the last rounding, satisfies

$$y^* = \frac{1}{b}(1+\epsilon)$$

where $|\epsilon| \leq 2^{-123.37}$. Therefore, the only cases where incorrect rounding can occur are those closer than this relative distance to a midpoint. The potentially failing significands b can be isolated by finding all $(2^{65}, 2^{64})$-balanced factorizations $mb = 2^{128} + d$ for integers $|d| \leq 24$ (since $24 + 1 > 2^{-123.37}/2^{-128}$) and m odd. The set of b values that we need to consider are the following 134 (ordered in decreasing size, not according to their closeness to a midpoint):

```
0xFFFFFFFFFFFFFFFF 0xFFFFFFFFFFFFFFFD 0xFE421D63446A3B34
0xFBFC17DFE0BEFF04 0xFB940B119826E598 0xFB0089D7241D10FC
0xFA0BF7D05FBE82FC 0xF912590F016D6D04 0xF774DD7F912E1F54
0xF7444DFBF7B20EAC 0xF39EB657E24734AC 0xF36EE790DE069D54
0xF286AD7943D79434 0xEDF09CCC53942014 0xEC4B058D0F7155BC
0xEC1CA6DB6D7BD444 0xE775FF856986AE74 0xE5CB972E5CB972E4
0xE58469F0234F72C4 0xE511C4648E2332C4 0xE3FC771FE3B8FF1C
0xE318DE3C8E6370E4 0xE23B9711DCB88EE4 0xE159BE4A8763011C
0xDF738B7CF7F482E4 0xDEE256F712B7B894 0xDEE24908EDB7B894
0xDE86505A77F81B25 0xDE03D5F96C8A976C 0xDDFF059997C451E5
0xDB73060F0C3B6170 0xDB6DB6DB6DB6DB6C 0xDB6DA92492B6DB6C
0xDA92B6A4ADA92B6C 0xD9986492DD18DB7C 0xD72F32D1C0CC4094
0xD6329033D6329033 0xD5A004AE261AB3DC 0xD4D43A30F2645D7C
0xD33131D2408C6084 0xD23F53B88EADABB4 0xCCCE6669999CCCD0
0xCCCE666666633330 0xCCCCCCCCCCCCCCD0 0xCBC489A1DBB2F124
0xCB21076817350724 0xCAF92AC7A6F19EDC 0xC9A8364D41B26A0C
0xC687D6343EB1A1F4 0xC54EDD8E76EC6764 0xC4EC4EC362762764
0xC3FCF61FE7B0FF3C 0xC3FCE9E018B0FF3C 0xC344F8A627C53D74
0xC27B1613D8B09EC4 0xC27B09EC27B09EC4 0xC07756F170EAFBEC
0xBDF3CD1B9E68E8D4 0xBD5EAF57ABD5EAF4 0xBCA1AF286BCA1AF4
0xB9B501C68DD6D90C 0xB880B72F050B57FC 0xB85C824924643204
0xB7C8928A28749804 0xB7A481C71C43DDFC 0xB7938C6947D97303
0xB38A7755BB835F24 0xB152958A94AC54A4 0xAFF5757FABABFD5C
0xAF4D99ADFEFCAAFC 0xAF2B32F270835F04 0xAE235074CF5BAE64
0xAE0866F90799F954 0xADCC548E46756E64 0xAD5AB56AD5AB56AC
0xAD5AAA952AAB56AC 0xAB55AAD56AB55AAC 0xAAAAB55555AAAAAC
0xAAAAAAAAAAAAAAAC 0xAAAAA00000555554 0xA93CFF3E629F347D
0xA80555402AAA0154 0xA8054ABFD5AA0154 0xA7F94913CA4893D4
0xA62E84F95819C3BC 0xA5889F09A0152C44 0xA4E75446CA6A1A44
0xA442B4F8DCDEF5BC 0xA27E096B503396EE 0x9E9B8FFFFFD8591C
0x9E9B8B0B23A7A6E4 0x9E7C6B0C1CA79F1C 0x9DFC78A4EEEE4DCB
0x9C15954988E121AB 0x9A585968B4F4D2C4 0x99D0C486A0FAD481
0x99B831EEE01FB16C 0x990C8B8926172254 0x990825E0CD75297C
0x989E556CADAC2D7F 0x97DAD92107E19484 0x9756156041DBBA94
0x95C4C0A72F501BDC 0x94E1AE991B4B4EB4 0x949DE0B0664FD224
0x942755353AA9A094 0x9349AE0703CB65B4 0x92B6A4ADA92B6A4C
0x9101187A01C04E4C 0x907056B6E018E1B4 0x8F808E79E77A99C4
0x8F64655555317C3C 0x8E988B8B3BA3A624 0x8E05E117D9E786D5
0x8BEB067D130382A4 0x8B679E2B7FB0532C 0x887C8B2B1F1081C4
0x8858CCDCA9E0F6C4 0x881BB1CAB40AE884 0x87715550DCDE29E4
0x875BDE4FE977C1EC 0x86F71861FDF38714 0x85DBEE9FB93EA864
0x8542A9A4D2ABD5EC 0x8542A150A8542A14 0x84BDA12F684BDA14
0x83AB6A090756D410 0x83AB6A06F8A92BF0 0x83A7B5D13DAE81B4
0x8365F2672F9341B4 0x8331C0CFE9341614 0x82A5F5692FAB4154
0x8140A05028140A04 0x8042251A9D6EF7FC
```

One can show by explicit computation that the algorithm works correctly on these values. It therefore rounds correctly on all values in round-to-nearest.

For directed rounding modes, the situation is less good. Once again the relative error condition gives rise to a set of test points, this time 227 of them. The algorithm works correctly on 220 of them, but not on floating-point numbers with one of the following 7 significands, the last of these representing exact powers of 2, for which the true result is exactly representable. Cognoscenti who perform a back-of-envelope calculation will not be surprised by the failure on exactly representable results, since correctness here would require y_2 already to be the correct result, which our relative error cannot quite guarantee.

```
0x8c82da588adc6416 0x84fdf027ef813f7b 0x827b9b8059090ab2
0x8080402010080401 0x8000080000400001 0x8000000000000001
0x8000000000000000
```

This analysis indicates that the algorithm will produce correctly rounded results if the ambient rounding mode is known to be round-to-nearest, but will not always guarantee correct rounding in other rounding modes. Moreover, note that for the same reason, the 'inexact' flag will be incorrectly set in round-to-nearest mode in the special cases where b is a power of 2. (As noted, the penultimate approximation y_2 cannot be the exact reciprocal in such cases, for otherwise we would obtain $e = 0$ and correct rounding in all

modes.) However, if this is considered important, it would be easy to detect and fix the problem with special case code without affecting overall latency.

7 Feasibility study

Although the previous sections show that the method is usefully applicable to some real problems, it's worth analyzing how practical the approach is likely to be in general. In attempting to use the method, three potential practical problems might arise

- Too many special points are isolated for further analysis to be feasible
- The factorization of some of the numbers is not feasible
- The distribution of prime factors is not feasible

We will not analyze the feasibility of factorization, since we do not understand the details of its implementation. We will however make the empirical observation that all factorizations for precisions up to $p = 64$ seem to be very straightforward for PARI / GP, taking a fraction of a second, while those for $p = 113$ usually take several seconds and, exceptionally, minutes.

Average density of balanced factorizations

It is not difficult to see that "on average" we obtain a fairly modest number of balanced factorizations per value examined. First note that the number of (A, B)-balanced products of numbers $\leq n$ is the number of lattice points contained both within the rectangle $0 \leq x \leq A$, $0 \leq y \leq B$ and under the curve $xy = n$. We can get a good estimate by ignoring "edge effects" and just considering the plane area, integrating to obtain:

$$C(n) = n(1 + \ln(\frac{AB}{n}))$$

Differentiating with respect to n yields the expected density, i.e. the average number of (A, B)-balanced product representations of a number close to n:

$$D(n) = \ln(AB/n)$$

Of course, these gross averages do not reflect small-scale fluctuations. Nevertheless, the agreement is fairly good with some empirical results obtained by sampling. In the following table, we examine the density of $(2^{\,p}, 2^p)$-balanced products for several p, looking in each case at 31 regions close to $\frac{k+1/2}{32}2^{2p}$ for $0 \leq k \leq 31$ and sampling 1024 successive points in each. The final figures at the bottom give the mean value. This indicates how accurate the sampling process is on average; perfectly representative sampling would give exactly 1 here. (We avoid sampling at $\frac{k}{32}2^p$ because that would lead to strong correlations between the sets of numbers at different k.)

$\ln(2^{2p}/n)$	$p = 24$	$p = 53$	$p = 64$
4.1588	4.4785	4.6835	3.3300
3.0602	2.8496	5.6621	3.2734
2.5494	2.4570	2.7070	2.2753
2.2129	2.0332	2.2421	2.2089
1.9616	2.0000	1.6953	2.3417
1.7609	1.9101	1.5664	1.5585
1.5939	1.5742	1.9140	1.2128
1.4508	1.3632	1.4765	1.5625
1.3256	1.3144	1.0839	1.2558
1.2144	1.2050	1.2187	1.2890
1.1143	1.0175	1.0996	1.4296
1.0233	1.0273	0.9335	0.9687
0.9400	0.7539	0.9062	0.8828
0.8630	0.7636	0.8613	0.8789
0.7915	0.6875	0.7187	0.6875
0.7248	0.6933	0.6621	0.7832
0.6623	0.6621	0.5976	0.7656
0.6035	0.5878	0.5468	0.6445
0.5479	0.5546	0.6210	0.5683
0.4953	0.4941	0.5136	0.6289
0.4453	0.4394	0.3847	0.3652
0.3976	0.3984	0.4453	0.4277
0.3522	0.3417	0.3476	0.3242
0.3087	0.3203	0.2890	0.3593
0.2670	0.2382	0.2285	0.2773
0.2270	0.2480	0.2070	0.3007
0.1885	0.1347	0.2207	0.2148
0.1515	0.1347	0.1640	0.1562
0.1158	0.0839	0.0976	0.1015
0.0813	0.0917	0.1054	0.0761
0.0480	0.0449	0.0371	0.0527
0.0157	0.0078	0.0078	0.0156
1.0000	0.9660	1.0701	0.9755

So much for the average case. What about the worst case? This seems a more difficult question to address theoretically, but in the next section we will show how to obtain a pessimistic upper bound.

Feasibility of distribution algorithm

Although the final number of values produced depends on the number of balanced factorizations, the process by which the balanced factorizations are enumerated involves examination of many dead-end paths, so the runtime of the distribution process may be very large relative to the final number of possibilities produced. A reasonable, though pessimistic, bound on the runtime of the distribution algorithm for a value n is $d(n)$, the *total* number of divisors of n, regardless of balance. For even without early cutoffs owing to range limitations, the algorithm cannot examine, given

$$n = \Pi_{i=1}^{i=k} p_i^{\alpha_i}$$

more than

$$d(n) = \Pi_{i=1}^{i=k} (1 + \alpha_i)^n$$

possibilities, since each factor $p_i^{\alpha_i}$ can, without range cutoffs, be distributed in $1 + \alpha_i$ ways.

It is well known that the average number of divisors $d(n)$ of a number near n is approximately $d(n) = \ln(n)$. This can easily be derived using the same sort of argument as we used above for balanced products [2]. This suggests that on average, the distribution process will not have many cases to examine; even for quad precision, we have $n \leq 2^{230}$ and so $\ln(n) \leq 160$.

What about the worst case? The number of divisors of a number can be much larger than $\ln(n)$. In fact [2], *almost all* numbers (in a precise sense) have about $\ln(n)^{\ln(2)}$ divisors, with the larger overall average of $\ln(n)$ resulting from a small proportion of numbers with many more divisors. Asymptotically, it is known [2] that $d(n)$ has an upper limit of exactly $2^{\ln(n)/\ln(\ln(n))}$, or more precisely, that if $\epsilon > 0$ then $d(n) < 2^{(1+\epsilon)\ln(n)/\ln(\ln(n))}$ for all sufficiently large n, while $d(n) > 2^{(1-\epsilon)\ln(n)/\ln(\ln(n))}$ for infinitely many n.

This asymptotic limit needs refinement to be useful to us for the concrete ranges we are interested in. We can obtain a more refined estimate of the maximum $d(n)$ for all n below some limit N we are interested in as follows. The key to efficient search is to seek the *minimal* n with the *maximal* number of divisors possible for $n \leq N$. The minimality constraint forces strong patterns onto the prime factorization. Suppose that n has the following prime factorization:

$$n = \Pi_{i=1}^{i=k} p_i^{\alpha_i}$$

Let $p_i < p_j$ be two primes (not necessarily appearing with nonzero index in the above factorization) such that $p_i^{\beta} < p_j < p_i^{\beta+1}$ for some nonnegative integer β. Then it is easy to see that if n has the minimality property, the following relationships hold between the α's:

$$\beta\alpha_j \leq \alpha_i \leq (\beta + 1)\alpha_j + 2\beta$$

For if the first inequality failed we could get a smaller number with at least as many divisors by replacing $p_i^{\alpha_i} p_j^{\alpha_j}$ with $p_i^{\alpha_i+\beta} p_j^{\alpha_j-1}$, while if the second inequality failed we could likewise replace it with $p_i^{\alpha_i-(\beta+1)} p_j^{\alpha_j+1}$.

This observation includes the case where p_j is the first prime beyond those appearing in the factorization, and in this case $\alpha_i \leq 2\beta$. For example, if 17^{α} appears in the factorization, so must $3^{2\alpha}$ and $2^{4\alpha}$, while if no power of 17 appears in the factorization then the highest possible power of 2 appearing is 2^8, and the highest power of 3 is 3^6. Note in particular that the factorization of the minimal n must contain the first k consecutive primes without gaps, for some k.

These observations cut down the search space dramatically enough that we can easily perform an exhaustive search for the precise worst numbers up to quite large values, say 2^{3000}. The following table shows, for various values of p up to 230, the minimal $n \leq 2^p$ with the largest number of divisors possible in that range. For each such n, we show $\log_2(n)$ and $\log_2(d(n))$ (where $d(n)$ is the number of divisors of n), as well as the ratio with the expected limit superior $r(n) = \log_2(d(n))/(\ln(n)/\ln(\ln(n)))$ and the actual factorization of n.

p	$\log_2(n)$	$\log_2(d(n))$	$r(n)$	Factorization of that worst n
10	9.71	5.00	1.416	$2^3\ 3\ 5\ 7$
20	19.45	7.90	1.525	$2^4\ 3^2\ 5 \cdots 13$
30	29.45	10.39	1.535	$2^6\ 3^3\ 5^2\ 7 \cdots 17$
40	39.80	12.71	1.528	$2^6\ 3^4\ 5^2\ 7 \cdots 23$
50	49.84	14.75	1.512	$2^5\ 3^3\ 5^2\ 7^2\ 11 \cdots 31$
60	59.96	16.71	1.498	$2^6\ 3^4\ 5^3\ 7^2\ 11 \cdots 37$
70	69.42	18.49	1.488	$2^7\ 3^4\ 5^2\ 7^2\ 11 \cdots 43$
80	79.88	20.33	1.474	$2^8\ 3^5\ 5^3\ 7^2\ 11 \cdots 47$
90	89.90	22.07	1.463	$2^8\ 3^4\ 5^3\ 7^2\ 11 \cdots 59$
100	99.88	23.75	1.453	$2^7\ 3^5\ 5^3\ 7^2\ 11^2\ 13 \cdots 61$
110	109.64	25.33	1.443	$2^8\ 3^5\ 5^3\ 7^2\ 11 \cdots 71$
120	119.87	26.97	1.435	$2^7\ 3^6\ 5^3\ 7^2\ 11^2\ 13 \cdots 73$
130	129.87	28.56	1.427	$2^7\ 3^6\ 5^3\ 7^2\ 11^2\ 13^2\ 17 \cdots 79$
140	139.99	30.12	1.420	$2^{10}\ 3^5\ 5^4\ 7^2\ 11^2\ 13^2\ 17 \cdots 83$
150	149.74	31.66	1.416	$2^9\ 3^5\ 5^3\ 7^2\ 11^2\ 13^2\ 17 \cdots 97$
160	159.79	33.14	1.408	$2^8\ 3^6\ 5^3\ 7^3\ 11^2\ 13^2\ 17 \cdots 101$
170	169.83	34.66	1.404	$2^9\ 3^5\ 5^3\ 7^2\ 11^2\ 13^2\ 17 \cdots 107$
180	179.99	36.14	1.398	$2^8\ 3^6\ 5^3\ 7^3\ 11^2\ 13^2\ 17 \cdots 109$
190	189.82	37.56	1.393	$2^9\ 3^5\ 5^4\ 7^2\ 11^2\ 13^2\ 17^2\ 19 \cdots 113$
200	199.88	39.02	1.388	$2^{10}\ 3^6\ 5^3\ 7^3\ 11^2\ 13^2\ 17^2\ 19 \cdots 127$
210	209.93	40.43	1.383	$2^{10}\ 3^6\ 5^3\ 7^3\ 11^2\ 13^2\ 17 \cdots 137$
220	219.87	41.83	1.379	$2^8\ 3^5\ 5^4\ 7^3\ 11^2\ 13^2\ 17^2\ 19 \cdots 139$
230	229.92	43.21	1.375	$2^{10}\ 3^5\ 5^3\ 7^3\ 11^2\ 13^2\ 17 \cdots 151$

We can see that even for double-extended precision, the number of factorizations that could possibly need to be examined is about 2^{28}. Although a fairly large number, this is definitely feasible. (And of course in practice such cases are exceptional and not all factorizations would be examined.) For quad precision, on the other hand, it is entirely possible for the search to be infeasible. We have not yet encountered this phenomenon in practice, however.

Note that $d(n)$ also gives an upper bound to the number of balanced factorizations. It is, of course, pessimistic, but testing on some of the values above suggests that the the number of balanced factorizations is a reasonable proportion (say 10%) of the total number of divisors. Naturally, it would be better to refine all these estimates to consider only numbers very close to the powers of 2, which is what we are really interested in.

The special numbers that we searched for above are particular cases of *highly composite numbers* [13]. For a detailed survey of the subject see [10], while Achim Flammenkamp's Web page seems to give a more efficient algorithm for generating HCNs:

`http://wwwhomes.uni-bielefeld.de/achim/highly.html`

The sequence of highly composite numbers is A002182 in Sloane's Encyclopedia of Integer Sequences.

8 Extension to reciprocal square root

It is interesting to note that a similar factor distribution technique can be used to attempt to find exceptional cases

for the reciprocal square root. In this case, we seek floating-point numbers or midpoints w and floating-point numbers y such that

$$\frac{|w - \frac{1}{\sqrt{y}}|}{|\frac{1}{\sqrt{y}}|}$$

is small. We can rewrite this as:

$$|\sqrt{y}(w - \frac{1}{\sqrt{y}})| = |w\sqrt{y} - 1|$$

In the critical cases where $w\sqrt{y} - 1$ is very small, then $w\sqrt{y} + 1$ is almost exactly 2 and so:

$$|w\sqrt{y} - 1| = \frac{|w^2 y - 1|}{|w\sqrt{y} + 1|} \approx \frac{|w^2 y - 1|}{2}$$

Once again, let us scale the values w and y to integers m and b:

$$\begin{array}{ll} y = 2^{e_y} b & 2^{p-1} \leq b < 2^p \\ w = 2^{e_w} m & 2^p \leq m < 2^{p+1} \end{array}$$

and then the distance we are interested in is then:

$$\frac{|m^2 b - 2^q|}{2^{q+1}}$$

where $q = -(2e_w + e_y)$. So we seek cases where $d = m^2 b - 2^q$ is as small as possible. Keeping in mind the range restrictions, we see that $2^{3p-1} \leq m^2 b < 2^{3p+2}$. As with simple reciprocals, it is impossible to come very close to the extremal powers of 2, but we do now need to consider two cases, $q = 3p$ and $q = 3p + 1$.

The reciprocal square root function is of some theoretical interest because it seems *prima facie* possible that $d = m^2 b - 2^q$ could be very small, perhaps even ± 1, yet no precisions where it is much smaller than 2^p have ever been found, and one might expect on naive statistical grounds that it is unlikely. (We only have 2^{2p} different choices of pairs m and b, and are scattering the resulting $m^2 b$'s somehow over an interval of size about 2^{3p}.) Li [7] proves that *assuming* the ABC conjecture from number theory holds, the distance is indeed of order 2^p for all sufficiently large p. Even if the ABC conjecture were proven, however, it's not clear whether it would be possible to constructivize the proof in order to obtain useful bounds for specific precisions. Iordache and Matula [4] observe that $d = 1$ is impossible in general, allowing the accuracy required to be lowered slightly, but add that 'trying to lower it is not an easy problem, even for a fixed p'. Although the present work does not touch the general case, and nor can it fully bridge the gap between expected and provable bounds, it *does* allow us quite easily to improve the provable bound for the typical p we are interested in by a reasonable factor.

We can take over the prime distribution function with little change. The only difference is that we now need to distribute the prime factors among $m^2 b$. This has the immediate consequence that only even powers of primes can be allocated to the m^2 part, and so any prime appearing to an odd power in the prime factorization of $2^q + d$ must be allocated at least once to b. This is almost always enough to render the distribution immediately impossible. We have made some searches for double-extended precision ($p = 64$) and quad precision ($p = 113$). For double-extended, we have shown that $d \leq 1024$ is impossible, and it would be easy to continue the search much further. For quad precision, the cost of factoring numbers is now a serious bottleneck, with a single number sometimes taking a day of CPU time and one of the factorizations for the $d = 6$ case apparently defeating factorization in a reasonable time. Nevertheless we have at least shown that $d < 6$ is impossible, which represents some improvement. For smaller precisions, it seems likely that other algorithms based on an (intelligent) exhaustive analysis of the whole space of significands would be more efficient. For example Lang and Muller [5] have performed a complete analysis of the double-precision case $p = 53$ (and found that the minimal distance is about 2^{-110}).

9 Conclusion

The methods described here allow reasonably effective isolation of the 'worst cases' for the reciprocal function. This opens the way to correctness proofs of reciprocal algorithms using the same kind of two-part approach used by Cornea [1] for square roots. In the absence of new theoretical advances, the method described may also be the best available means of improving the difficulty bounds on the reciprocal square root functions for larger precisions. Although our method has feasibility problems for the extreme case of quad-precision reciprocal square roots, it would be possible to explore alternative factoring algorithms. The numbers we are interested in factoring are very close (in relative terms) to powers of 2, so it is possible that algorithms such as the Special Number Field Sieve (SNFS) would give much better results.

Acknowledgements

The author is grateful to the anonymous referees, who made a number of excellent suggestions, and pointed out connections of which the author had been unaware.

References

[1] M. Cornea-Hasegan. Proving the IEEE correctness of iterative floating-point square root, divide

and remainder algorithms. *Intel Technology Journal*, 1998-Q2:1–11, 1998. Available on the Web as `http://developer.intel.com/technology/itj/q21998/articles/art_3.htm`.

[2] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Clarendon Press, 5th edition, 1979.

[3] J. Harrison. Formal verification of IA-64 division algorithms. In M. Aagaard and J. Harrison, editors, *Theorem Proving in Higher Order Logics: 13th International Conference, TPHOLs 2000*, volume 1869 of *Lecture Notes in Computer Science*, pages 234–251. Springer-Verlag, 2000.

[4] C. Iordache and D. W. Matula. On infinitely precise rounding for division, square root, reciprocal and square root reciprocal. In I. Koren and P. Kornerup, editors, *Proceedings, 14th IEEE symposium on on computer arithmetic*, pages 233–240, Adelaide, Australia, 1999. IEEE Computer Society. See also Technical Report 99-CSE-1, Southern Methodist University.

[5] T. Lang and J.-M. Muller. Bounds on runs of zeros and ones for algebraic functions. Research Report 4045, INRIA, 2000.

[6] V. Lefèvre and J.-M. Muller. Worst cases for correct rounding of the elementary functions in double precision. Research Report 4044, INRIA, 2000.

[7] R.-C. Li. The ABC conjecture and correctly rounded reciprocal square root. Preprint, 2002.

[8] P. Markstein. *IA-64 and Elementary Functions: Speed and Precision*. Prentice-Hall, 2000.

[9] P. W. Markstein. Computation of elementary functions on the IBM RISC System/6000 processor. *IBM Journal of Research and Development*, 34:111–119, 1990.

[10] J.-L. Nicholas. On highly composite numbers. In *Ramanujan Revisited: Proceedings of the Centenery Conference*, pages 215–244. Academic Press, 1988.

[11] M. Parks. Number-theoretic test generation for directed rounding. *IEEE Transactions on Computers*, 49:651–658, 2000.

[12] V. Pratt. Every prime has a succinct certificate. *SIAM Journal of Computing*, 4:214–220, 1975.

[13] S. Ramanujan. Highly composite numbers. *Proceedings of the London Mathematical Society*, 14:347–409, 1915.

[14] P. T. P. Tang. Testing computer arithmetic by elementary number theory. Preprint MCS-P84-0889, Mathematics and Computer Science Division, Argonne National Labs, 1989.

Challenges in Mathematical Computing

Jonathan M. Borwein and Peter B. Borwein*

February 19, 2001

ABSTRACT. Almost all interesting mathematical algorithmic questions relate to NP-hard questions and such computation is prone to explode exponentially. More space, more speed and processors, and even say massive parallelism will have an impact but it will be largely at a 'micro not macro' level. We anticipate the greatest benefit accruing from mathematical platforms that allow for highly computer assisted insight generation (more 'aha's' per cycle), not from solution of grand challenge problems.

1 Mathematics Embraces Computing

It is often said that pure mathematicians invented digital computers and then proceeded to ignore them for the better part of half a century. In the past two decades this situation has started to change with a vengeance.

Major *symbolic mathematics* or *computer algebra* packages, most notably Maple and Mathematica, have over the last fifteen years reached a remarkable degree of sophistication. We should also allude to counterparts such as Axiom, Macsyma, Reduce, MuPad and Derive and to many other more specialized packages such as GAP, Magma or Cayley (for group theoretic computation), Pari (for number theory), KnotPlot (for knot theory) SnapPea

*Centre for Experimental & Constructive Mathematics, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada. Email: jborwein@cecm.sfu.ca,pborwein@cecm.sfu.ca. Research supported by NSERC and by the Network of Centres of Excellence Program.

(for hyperbolic 3-manifolds) and SPlus (for statistics), and many more. This sophistication has relied on a confluence of algorithmic breakthroughs, dramatically increased processor power, almost limitless storage capacity, and most recently network communication, excellent online data bases and web-distributed (often Java-based) computational tools. We mention: the mathematics front end to the Los Alamos Preprint ArXiv (`front.math.ucdavis.edu/`), Mathematical Reviews on the Web (`e-math.ams.org/mathscinet`), Neil Sloane's Encyclopedia of Integer Sequences (`www.research.att.com/personal /njas/sequences/eisonline.html`), our own Inverse Symbolic Calculator (`www.cecm.sfu.ca/projects/ISC/ISCmain.html`) which infers symbolic structure from numerical input, and an Integer Relation Finder (`www.cecm.sfu.ca/projects/IntegerRelations/`).

The relatively seamless *integration* of all these components arguably represents *the* challenge for 21st Century computational mathematics. By contrast, it is hard to think of mathematical problems where a dramatic increase in speed and scale of computation would make possible a presently intractable line of research. It is easy to give examples where it would not. Thus, consider Lam's 1991 proof (`www.cecm.sfu.ca/organics/papers/lam/index.html`) of the nonexistence of a finite projective plane of order 10.[1] It involved thousands of hours of CRAY and other computation. Lam's estimate is that the next case ($n = 18$) susceptible to his methods would take millions of years on any conceivable architecture. While a certain class of mathematical enquiries is susceptible to massively parallel, even web based 'embarrassingly parallel', computation[2] these tend, however interesting, not to to be problems central to mathematics.

2 Computational Excursions in Contemporary Mathematics

Rather difficult problems, previously viewed as intractable, such as exact integration of elementary functions have been significantly attacked. A number of the most important mathematical algorithms of the twentieth century are (i) the Fast Fourier Transform, (ii) Lattice Basis Reduction methods and

[1] A hunt for a configuration of $n^2 + n^1 + 1$ points and lines.

[2] For example, discovering Mersenne primes: those of the form $2^n - 1$.

related Integer Relation algorithms, (iii) the Risch algorithm for indefinite integration, (iv) Gröbner basis computation for solving algebraic equations, and (v) the Wilf/Zeilberger Algorithms for 'hypergeometric' summation and integration that rigorously prove very large classes of identities. All these are, or soon will be, centrally incorporated in such packages.[3]

Such packages, and powerful more numerical relatives such as MatLab, can now substantially deal with large parts of the standard mathematics curriculum – and can out-perform most of our undergraduates to boot. They provide extraordinary opportunities for research that most mathematicians are only beginning to appreciate and digest. They also allow access to sophisticated mathematics to a very broad cross-section of scientists and engineers.

There is a coherent argument that the emergence of such packages, and their integration into mathematical parole, represents the most significant part in a paradigm shift in how mathematics is done; and certainly they have already become a central research tool in many subareas of mathematics both from an exploratory and from a formal point of view. (It is acceptable now to see a line in a proof that begins "by a large calculation in Maple we see ...".) The first objective of symbolic algebra packages was to do as much exact mathematics as possible. A second increasingly important objective is to do it very fast and to deal in an arbitrary precision environment with the more standard algorithms of mathematical analysis. Roughly, one would like to be able to incorporate the usual methods of numerical analysis into an exact environment or at least into an arbitrary precision environment.

The problems are obvious and hard. For example, how does one do arbitrary precision numerical quadrature? When does one switch methods with precision required or with different analytic properties of the integrand? How does one deal with branch cuts of analytic functions? How does one deal consistently with log? (Even this isn't completely worked out.) More ambitiously how does one do a similar analysis for differential equations? The goal is to marry the algorithms of analysis with symbolic and exact computation and to do this with as little loss of speed as possible. Sometimes this means we must first go back and speed up the core algebraic calculations. And ultimately, can we provide any 'certificates' that a given numeric or

[3]The first two were among the ten algorithms with "the greatest influence on the development and practice of science and engineering in the 20th century" described in the previous volume of this journal. Of course many of the others, such as sorting algorithms, are fundamental to the needs of contemporary mathematics.

symbolic computation is indeed a proof or even just correct?

Within this context a number of very interesting problems concerning the visualization of mathematics arise. How does one actually "see" what one is doing. It has been argued that Cartesian graphing was the most important invention of the last millennium. Certainly it changed how we thought about mathematics – the subsequent development of differential calculus rested on it. More subtle and complicated graphics, like those of fractals, allow for a kind of exploration that was previously impossible. There are many issues to be worked out here that live at the interface of mathematics, pedagogy and even psychology but are very timely to get right. (Think of how one visualizes the human genome and its patterns – which is after all just a particular several billion digit number base four.) An instructive example is afforded by the growing reliance of numerical analysts on graphic representation of large sparse matrices – the pictures show structure, numerical measurements very little.[4]

The great success of the symbolic algebra packages has been their mathematical generality and ease of use. These packages deal most successfully with algebraic problems while many (perhaps most) serious applications require analytic objects such as definite integrals, series and differential equations. All the elementary notions of analysis, like continuity and differentiability have to be given precise computational meaning. The first challenge involves mathematical algorithmic developments to allow the handling of a variety of these only partially handled problems – including the analysis of functions given by programs. Many of these relate to the difficult mathematical problems involved in automatic simplification of complicated analytic formulae and recognition of when two very different such expressions represent the same object. There is also an intrinsic need to mix numeric and symbolic (exact and inexact) methods. Human mathematicians often criticize programs for making dumb errors but often these errors (such as over simplifying expressions, leaving out hypotheses or 'dividing by zero') are precisely how one begins oneself. As Hadamard noted almost a century ago:

> "The object of mathematical rigor is to sanction and legitimize the conquests of intuition, and there was never any other object for it."

[4] A nice example is JavaView (`www-sfb288.math.tu-berlin.de/vgp/javaview/index.html`) for doing 3D Geometry on the web.

3 Challenge Problems for Computational Pure Mathematics

1. The question that a pure mathematician might trade his soul with the devil to solve is most likely the so called "Riemann–Hypothesis" of 1859. The bounty on its solution now exceeds $\$1,000,000$ – the amount offered by the *Millennium Prize* of the Clay Mathematics Institute (`www.claymath.org/prize_problems/rules.htm`).

At the Clay Institute website the problem is described in the following form:

> "Some numbers have the special property that they cannot be expressed as the product of two smaller numbers, e.g., $2, 3, 5, 7$, etc. Such numbers are called prime numbers, and they play an important role, both in pure mathematics and its applications. The distribution of such prime numbers among all natural numbers does not follow any regular pattern, however the German mathematician G.F.B. Riemann (1826–1866) observed that the frequency of prime numbers is very closely related to the behavior of an elaborate function $\zeta(s)$ called the Riemann Zeta function. The Riemann hypothesis asserts that all interesting solutions of the equation $\zeta(s) = 0$ lie on a straight line. This has been checked for the first 1,500,000,000 solutions. A proof that it is true for every interesting solution would shed light on many of the mysteries surrounding the distribution of prime numbers."

A little more precisely the Riemann Hypothesis is usually formulated as:

> *All the zeros in the right half of the complex plane of the analytic continuation of*
>
> $$\zeta(s) := \sum_{n=0}^{\infty} \frac{1}{n^s}$$
>
> *lie on the vertical line* $\Re(s) = \frac{1}{2}$.

We observe in passing that one of the most famous results in elementary mathematics is Bernoulli's evaluation of $\zeta(2) = \pi^2/6$.

Without doubt this is one of the 'grand challenge' problems of mathematics and for good reason. Large tracts of mathematics fall into place if the Riemann Hypothesis is true. Unlike problems such as Fermat's last problem (now theorem) which may prove to be an isolated mountain peak, even if the proof methods are tremendously significant, [5] the truth of the Riemann Hypothesis is central – its falseness would be disquieting. Most mathematicians believe the Riemann Hypothesis is true though there have been notable dissenters. Littlewood, one of the great analytic number theorists of last century is in print hypothesizing its falseness [6] . Of course, finding just one zero off the line $\Re(s) = \frac{1}{2}$,[7] should it exist, is worth a million dollars (although perhaps the prize is only for a proof not a disproof – certainly a proof is more interesting) and this may provide additional motivation to extend the climb of this particular mountain. The fact that more than the first billion zeros are known, by computation, to satisfy the Riemann hypothesis might be considered "strong numerical evidence" as it is the article by Enrico Bombieri that accompanies the prize citation. But it is far from overwhelming – there are subtle phenomena in this branch of mathematics that only manifest themselves far outside of present computer range.

One reason to extend such computations, which are neither easy nor obvious and rely on some fairly subtle mathematics, is the hope that one will uncover delicate phenomena that give insight for a proof. Greatly more ambitious is the possibility that, in the very long run, it will be possible to machine generate a proof even for problems clearly as difficult as this one.

2. Of the seven million-dollar Millennium Prize problems, the one that is most germane to this discussion is the so called $P \neq NP$ *problem.* Again, we quote from the discussion on the Clay website:

> "It is Saturday evening and you arrive at a big party. Feeling shy, you wonder whether you already know anyone in the room. Your host proposes that you must certainly know Rose, the lady in the corner next to the dessert tray. In a fraction of a second you are able to cast a glance and verify that your host is correct. However,

[5] A much deeper community understanding of modular and elliptic functions may also pay dividends.

[6] J.E. Littlewood, "Some Problems in Real and Complex Analysis," Heath Mathematical Monographs, 1968.

[7] And off the real line where there are 'trivial' zeros at negative even integers.

> in the absence of such a suggestion, you are obliged to make a tour of the whole room, checking out each person one by one, to see if there is anyone you recognize. This is an example of the general phenomenon that generating a solution to a problem often takes far longer than verifying that a given solution is correct. Similarly, if someone tells you that the number $13,717,421$ can be written as the product of two smaller numbers, you might not know whether to believe him, but if he tells you that it can be factored as 3607 times 3803 then you can easily check that it is true using a hand calculator. One of the outstanding problems in logic and computer science is determining whether questions exist whose answer can be quickly checked (for example by computer), but which require a much longer time to solve from scratch (without knowing the answer). There certainly seem to be many such questions. But so far no one has proved that any of them really does require a long time to solve; it may be that we simply have not yet discovered how to solve them quickly. Stephen Cook formulated the P versus NP problem in 1971."

Although in many instances one may question the practical distinction between polynomial and non polynomial algorithms, this problem really is central to our current understanding of computing. Roughly it conjectures that many of the problems we currently find computationally difficult must per force be be that way. It is a question about methods, not about actual computations, but it underlies many of the challenge problems one can imagine posing. A question that requests one to "compute such and such a sized incidence of this or that phenomena" always risks having the answer "it's just not possible" because $P \neq NP$.

4 Two Specific Challenges

With the 'NP' caveat,[8] let us offer two challenges that are let us offer two challenges that are far fetched but not inconceivable goals for the next few decades.

[8]Though factoring is difficult it is not generally assumed to be in the class of NP-hard problems.

3. *Design an algorithm that can reliably factor a random thousand digit integer.*

Current algorithms even with a huge effort get stuck at about 150 digits. Details lie at `www.rsasecurity.com/rsalabs/challenges/factoring/index.html` where the current factoring challenges are listed. Again, in the cash prize game there is also a $\$100,000$ offered for any honest $10,000,000$ digit prime (`www.mersenne.org/prime.htm`.)

Primality checking is currently easier than factoring, and there are some very fast and powerful *probabilistic* primality tests – much faster than those providing 'certificates'. Given that any computation has potential errors due to: (i) subtle (or even not-so-subtle) programming bugs, (ii) compiler errors, (iii) other software errors, (iv) and undetected hardware integrity errors, it seems increasingly pointless to distinguish between these two types of primality tests. Many would take their chances with a $(1 - 10^{-100})$ probability statistic over a 'proof' any day.

The above questions are intimately related to the Riemann Hypothesis, though not obviously so to the non afficionado. They are also critical to issues of internet security. Learn how to factor large numbers and most current security systems are crackable.

There are many old plum problems that lend themselves to extensive numerical exploration. To name but one other: a problem that arose originally in signal processing called the *Merit Factor problem* that is due in large part to Marcel Golay with closely related versions to Littlewood and Erdős. It has a long pedigree though certainly not as long as the Riemann hypothesis. Recent references and records can be found at (`itp.nat.uni-magdeburg.de/ mertens/`.)

It can be formulated as follows. Suppose $(a_0 := 1, a_1, ..., a_n)$ is a sequence of length $n + 1$ where each a_i is either 1 or -1. If

$$c_k = \sum_{j=0}^{n-k} a_j a_{j+k}$$

then the problem is, for each fixed n, to minimize,

$$\sum_{k=-n}^{n} c_k^2.$$

Minima have be found up to about about $n = 50$. The search space of sequences at size 50 is 2^{50} which is about todays limit of a very very large

scale calculation. In fact the records use a branch and bound algorithm which grows more or less like 1.8^n. This is marginally better than the naive 2^n of a completely exhaustive search but is still painfully exponential.

4. *Find the minima in the merit factor problem up to size 100.*

The best hope for a solution is better algorithms. The problem is widely acknowledged as a very hard problem in combinatorial optimization but it isn't known to be in one of the recognized hard classes like NP. The next best hope is radically different computers, perhaps quantum computers. And there is always a remote chance that analysis will lead to a mathematical solution.

5 A Concrete Example

In this section we illustrate some of the mathematical challenges with a specific problem, proposed in the *American Mathematical Monthly* (November, 2000).

10832. *Donald E. Knuth, Stanford University, Stanford, CA.* Evaluate

$$\sum_{k=1}^{\infty} \left(\frac{k^k}{k!\, e^k} - \frac{1}{\sqrt{2\pi k}} \right).$$

1. A very rapid Maple computation yielded $-0.08406950872765600\ldots$ as the first 16 digits of the sum.

2. The Inverse Symbolic Calculator has a 'smart lookup' feature[9] that replied that this was probably $-\frac{2}{3} - \zeta(\frac{1}{2})/\sqrt{2\pi}$.

3. Ample experimental confirmation was provided by checking this to 50 digits. Thus within minutes we *knew* the answer.

4. As to why? A clue was provided by the surprising speed with which Maple computed the slowly convergent infinite sum. The package clearly knew something the user did not. Peering under the covers

[9] Alternatively, a sufficiently robust integer relation finder could be used.

revealed that it was using the *LambertW* function, W, which is the inverse of $w = z\exp(z)$.[10]

5. The presence of $\zeta(1/2)$ and standard Euler-MacLaurin techniques, using Stirling's formula (as might be anticipated from the question), led to

$$\sum_{k=1}^{\infty}\left(\frac{1}{\sqrt{2\pi k}} - \frac{1}{\sqrt{2}}\frac{(\frac{1}{2})_{k-1}}{(k-1)!}\right) = \frac{\zeta(\frac{1}{2})}{\sqrt{2\pi}}, \tag{1}$$

   where the binomial coefficients in (1) are those of $\frac{1}{\sqrt{2-2\,z}}$. Now (1) is a formula Maple can 'prove'.

6. It remains to show

$$\sum_{k=1}^{\infty}\left(\frac{k^k}{k!\;e^k} - \frac{1}{\sqrt{2}}\frac{(\frac{1}{2})_{k-1}}{(k-1)!}\right) = -\frac{2}{3}. \tag{2}$$

7. Guided by the presence of W and its series $\sum_{k=1}^{\infty}\frac{(-k)^{k-1}z^k}{k!}$, an appeal to Abel's limit theorem lets one deduce the need to evaluate

$$\lim_{z\to 1}\left(\frac{d}{dz}\mathrm{W}(-\frac{z}{e}) + \frac{1}{\sqrt{2-2\,z}}\right) = \frac{2}{3}. \tag{3}$$

   Again Maple happily does know (3).

Of course this all took a fair amount of human mediation and insight.

6 Conclusion

In 1996, discussing the philosophy and practice of Experimental Mathematics, we wrote:[11]

[10] A search for 'Lambert W function' on MathSciNet provided 9 references – all since 1997 when the function appears named for the first time in Maple and Mathematica.

[11] J.M. Borwein, P.B. Borwein, R. Girgensohn and S. Parnes, "Making Sense of Experimental Mathematics," Mathematical Intelligencer, 18, Number 4 (Fall 1996), 12-18. The quotes from Zeilberger and Chaitin are also cited therein.

> "As mathematics has continued to grow there has been a recognition that the age of the mathematical generalist is long over. What has not been so readily acknowledged is just how specialized mathematics has become. As we have already observed, sub-fields of mathematics have become more and more isolated from each other. At some level, this isolation is inherent but it is imperative that communications between fields should be left as wide open as possible. As fields mature, speciation occurs. The communication of sophisticated proofs will never transcend all boundaries since many boundaries mark true conceptual difficulties. But experimental mathematics, centering on the use of computers in mathematics, would seem to provide a common ground for the transmission of many insights."

This common ground continues to increase and extends throughout the sciences and engineering.

The corresponding need is to retain the robustness and unusually long-livedness of the rigorous mathematical literature. Doron Zeilberger's proposed *Abstract of the future* (1993) challenges this in many ways.

> "We show in a certain precise sense that the Goldbach conjecture[12] is true with probability larger than 0.99999 and that its complete truth could be determined with a budget of 10 billion."

He goes on to suggest that only the Riemann hypothesis merits paying really big bucks for certainty. Relatedly, Greg Chaitin (1994) argued that we should introduce the Riemann hypothesis as an 'axiom'.

> "I believe that elementary number theory and the rest of mathematics should be pursued more in the spirit of experimental science, and that you should be willing to adopt new principles. I believe that Euclid's statement that an axiom is a self-evident truth is a big mistake[13]. The Schrödinger equation certainly isn't a self-evident truth! And the Riemann hypothesis isn't self-evident either, but it's very useful. A physicist would say that

[12] Every even number is the sum of two primes.

[13] There is no evidence that Euclid ever made such a statement. However, the statement does have an undeniable emotional appeal.

there is ample experimental evidence for the Riemann hypothesis and would go ahead and take it as a working assumption."

How do we reconcile these somewhat combative challenges with the inarguable power of the deductive method? How do we continue to produce rigorous mathematics when more and more research will be performed in large computational environments where one may or not be able to determine what the system has done or why?[14]

At another level we see the core challenge for mathematical computing to be the construction of work spaces that largely or completely automate the diverse steps illustrated in Knuth's and like problems.

[14]This has often been described as "relying on proof by 'Von Neumann says'."

Riccati meets Fibonacci

Wolfdieter L a n g

Institut für Theoretische Physik

Universität Karlsruhe

Kaiserstrasse 12, D-76128 Karlsruhe, Germany

E-mail: wolfdieter.lang@physik.uni-karlsruhe.de,

http://www-itp.physik.uni-karlsruhe.de/˜ wl

1 Introduction

Consider the *Riccati* differential equation

$$f(x)\,G'(x) \;=\; f_0(x)\,G^2(x) \;+\; f_1(x)\,G(x) \;+\; f_2(x)\;. \tag{1}$$

For $f_2(x) \equiv 0$ this reduces to a special *Bernoulli* equation (with exponent 2) which will be treated separately. For the history of such eqs. see [9], ch.I, 1.1. If $f_0(x)$ does not vanish we speak of the non-degenerate case, and

$$G^2(x) \quad = \alpha(x)\,G'(x) \;-\; \beta(x)\,G(x) \;-\; \gamma(x)\;, \tag{2}$$

with $\alpha(x) = f(x)/f_0(x)$, $\beta(x) = f_1(x)/f_0(x)$, and $\gamma(x) = f_2(x)/f_0(x)$.

Let $G(x)$ generate the number sequence $\{G_n\}_0^\infty$, *i.e.* $G(x) = \sum_{n=0}^\infty G_n\ x^n$. Because $G^2(x)$ is the generating function for the convolution of the sequence $\{G_n\}_0^\infty$ with itself, *i.e.* of $G_n^{(1)} := \sum_{k=0}^n G_k\, G_{n-k}$, one can use eq. 2 in order to express the convolution numbers $G_n^{(1)}$ in terms of $\{G_k\}_0^{n+1}$ and the numbers $\{\alpha_k\}_0^n$, $\{\beta_k\}_0^n$, and γ_n, which are generated by the functions $\alpha(x)$, $\beta(x)$, and $\gamma(x)$, respectively, as follows.

$$\begin{aligned} G_n^{(1)} &= \sum_{q=0}^{n} \left((n+1-q)\, G_{n+1-q}\, \alpha_q \ - \ G_{n-q}\beta_q\right) \ - \ \gamma_n \ , \\ &= \sum_{q=0}^{n} \left((q+1)\, G_{q+1}\, \alpha_{n-q} \ - \ G_q\beta_{n-q}\right) \ - \ \gamma_n \ . \end{aligned} \tag{3}$$

The $k-$th order convolution sequence $\{G_n^{(k)}\}_{n=0}^\infty$ is generated by $G^{k+1}(x)$, and can be obtained recursively if one first writes $G^{k+1}(x) = G^{k-1}(x)\, G^2(x)$ and then employs *Riccati* eq. 2:

$$G^{k+1}(x) \ = \ \left(\alpha(x)\,\frac{1}{k}\,\frac{d}{dx} \ - \ \beta(x)\right)\, G^k(x) \ - \ \gamma(x)\, G^{k-1}(x) \tag{4}$$

for $k \in \mathbb{N}$. This yields in terms of the expansion coefficients, from $G^{k+1}(x) =: \sum_{n=0}^\infty G_n^{(k)}\ x^n$,

$$G_n^{(k)} \ = \ \sum_{q=0}^{n} \left(\frac{1}{k}\,(q+1)\, G_{q+1}^{(k-1)}\, \alpha_{n-q} \ - \ G_q^{(k-1)}\,\beta_{n-q} \ - \ G_q^{(k-2)}\,\gamma_{n-q}\right) \ . \tag{5}$$

for $k \in \mathbb{N}$, with $G_q^{(-1)} := \delta_{q,0}$ (*Kronecker* symbol) and $G_q^{(0)} = G_q$.

As is well-known, *Riccati* eq. 2 can be transformed into a homogeneous second order differential equation of the type (we use $\alpha(x) \neq 0$)

$$\alpha(x)\, H^{''}(x) \ + \ (\alpha'(x) \ - \ \beta(x))\, H'(x) \ + \ (\gamma(x)/\alpha(x))\, H(x) \ = \ 0 \ . \tag{6}$$

This transformation is accomplished by

$$G(x) \ = \ -\alpha(x)(ln\, H(x))' \quad or \quad H(x) \ = \ exp\left(-\int \frac{G(x)}{\alpha(x)}\, dx\right) \ , \tag{7}$$

Therefore, if a function $H(x)$ satisfies the differential eq. of type 6 with certain initial conditions for $H(0)$ and $H'(0)$ we can use recursion eq. 5 for the $k-$th convolution of the sequence $\{G_n\}_0^\infty$ generated by $G(x) = -\alpha(x)(ln\, H(x))'$, and $\alpha(x)$, $\beta(x)$, and $\gamma(x)$ generate the coefficients in eq. 5

In the special *Bernoulli* case, when $\gamma(x) \equiv 0$, and $f(x) \neq 0$ and $\alpha(x) \neq 0$, the eq.

$$G'(x) \;=\; \frac{f_1(x)}{f(x)}\, G(x) \;+\; \frac{f_0(x)}{f(x)}\, G^2(x) \;=\; (\beta(x)\, G(x) \;+\; G^2(x))/\alpha(x)\;, \tag{8}$$

can be transformed into an inhomogeneous first order linear differential eq. for the inverse of $G(x)$; *i.e.* $H(x) := 1/G(x)$ satisfies

$$\alpha(x)\, H'(x) \;+\; \beta(x)\, H(x) \;=\; -1\;, \tag{9}$$

with the solution

$$H(x) \;=\; \frac{1}{G(x)} \;=\; e^{F(x)} \left[C \;-\; \int \frac{e^{-F(x)}}{\alpha(x)}\, dx \right]\;, \tag{10}$$

where C is an integration constant, and $F(x) := -\int (\beta(x)/\alpha(x))\, dx$.

Therefore, if $H(x)$ satisfies a differential equation of type 9 with a certain initial condition for $H(0)$ we can use recursion eq. 5, with $\gamma_{n-q} \equiv 0$, for the $k-$th convolution of the sequence $\{G_n\}_0^\infty$ generated by $G(x) = 1/H(x)$. $\alpha(x)$ and $\beta(x)$ generate the remaining coefficients in eq. 5.

From this set-up we do not gain direct information about convolutions of the sequence of numbers generated by the functions $H(x)$ in both cases. This method becomes particularly useful if the coefficient functions $\alpha(x)$, $\beta(x)$ and $\gamma(x)$ are simple, for example if they are polynomials.

In this paper we concentrate on examples of *Riccati* equation 8 of the special *Bernoulli* type. It is shown that the generalized *Fibonacci* and corresponding *Lucas* numbers are generated by functions which satisfy such a *Riccati* equation. We discuss the resulting expressions for the $k-$th convolution of these number sequences. At the end we extend this method to the so-called generalized $p-$*Fibonacci* numbers which appeared in a recent paper [6].

2 Summary

The generating function for the generalized *Fibonacci* numbers $\{F_n(a,b)\}_0^\infty$, defined by the three term recurrence relation

$$F_n(a,b) \;=\; a\,F_{n-1}(a,b) \;+\; b\,F_{n-2}(a,b), \quad F_0(a,b) \;=\; 0,\; F_1(a,b) \;=\; 1, \tag{11}$$

with given real $a \neq 0$ and $b \neq 0$, is well-known. For arbitrary a and b, $F_n(a,b)$ can be considered as a polynomial in two variables. If we introduce the numbers, or polynomials, $U_n(a,b) := F_{n+1}(a,b)$ we have from the recursion with input $U_0(a,b) = 1$ and $U_1(a,b) = a$ (or $U_{-1}(a,b) = 0$)

$$U(a,b;x) \;:=\; \sum_{n=0}^{\infty} U_n(a,b)\,x^n \;=\; \frac{1}{1 \,-\, a\,x \,-\, b\,x^2}\;. \tag{12}$$

Similarly, for the generalized *Lucas* numbers $\{L_n(a,b)\}_0^\infty$ which satisfy the same recursion eq. 11 but with inputs $L_0(a,b) = 2, L_1(a,b) = a$, we find, with $V_n(a,b) \;:=\; L_{n+1}(a,b)/a$, remembering that $a \neq 0$,

$$V(a,b;x) \;:=\; \sum_{n=0}^{\infty} V_n(a,b)\,x^n \;=\; \frac{1 + 2\,b\,x/a}{1 \,-\, a\,x \,-\, b\,x^2}\;. \tag{13}$$

The input is now $V_0(a,b) = 1$ and $V_1(a,b) = (a^2 \,+\, 2\,b)/a$ (or $V_{-1}(a,b) = 2/a$).

These (ordinary) generating functions can also be written in terms of the characteristic roots corresponding to recursion relation eq. 11

$$\lambda_\pm \;\equiv\; \lambda_\pm(a,b) \;:=\; \frac{1}{2}\,(a \,\pm\, \sqrt{a^2 + 4\,b}) \tag{14}$$

as follows.

$$U(a,b;x) \quad = \quad \frac{1}{x\,(\lambda_+ \,-\, \lambda_-)}\left(\frac{1}{1 \,-\, \lambda_+\,x} \;-\; \frac{1}{1 \,-\, \lambda_-\,x}\right)\;, \tag{15}$$

$$V(a,b;x) \quad = \quad \frac{1}{\lambda_+ + \lambda_-}\left(\frac{\lambda_+}{1 \,-\, \lambda_+\,x} \;+\; \frac{\lambda_-}{1 \,-\, \lambda_-\,x}\right)\;. \tag{16}$$

The corresponding *Binet* forms of the generated numbers are, in the non-degenerate case $\lambda_+ \neq \lambda_-$, *i.e.* $D(a,b) := a^2 + 4\,b \neq 0$,

$$U_n(a,b) \quad = \quad \frac{\lambda_+^{n+1} \; - \; \lambda_-^{n+1}}{\lambda_+ - \lambda_-} \; , \tag{17}$$

$$V_n(a,b) \quad = \quad \frac{\lambda_+^{n+1} \; + \; \lambda_-^{n+1}}{\lambda_+ + \lambda_-} \; . \tag{18}$$

In the degenerate case we have

$$U_n(a) \quad := \quad U_n(a, -\frac{a^2}{4}) \; = \; (n+1)\left(\frac{a}{2}\right)^n \; , \tag{19}$$

$$V_n(a) \quad := \quad V_n(a, -\frac{a^2}{4}) \; = \; \left(\frac{a}{2}\right)^n \; . \tag{20}$$

A sum representation of these polynomials is obtained by expanding the generating functions.

$$U_n(a,b) \quad = \quad \sum_{l=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n-l}{l} a^{n-2l}\, b^l, \tag{21}$$

$$V_n(a,b) \quad = \quad \sum_{l=0}^{\lfloor \frac{n+1}{2} \rfloor} \frac{n+1}{n+1-l} \binom{n+1-l}{l} a^{n-2l}\, b^l. \tag{22}$$

This result for $U_n(a,b)$ follows also from a combinatorial interpretation of the recurrence relation, and the one for $V_n(a,b)$ is also due to the *Girard - Waring* formula in its simplest version (for this *cf.* [4], [3], also for original refs.).

The generating functions eq. 12 (or eq. 15) and eq. 13 (or eq. 16) are found to be the unique solutions of *Riccati* eqs. (simultaneously a special type of *Bernoulli* eq.) of the type shown in eq. 8. To be precise we have, identically in a and b,

$$(a \; + \; 2\,b\,x)\,\frac{\partial}{\partial x} U(a,b;x) \; + \; 4\,b\,U(a,b;x) \; - \; (a^2 \; + \; 4\,b)\,U^2(a,b;x) \; = \; 0, \tag{23}$$

with the initial condition $U(a,b;0) = 1$. Similarly,

$$(1 \; + \; 2\,\frac{b}{a}\,x)^2\,\frac{\partial}{\partial x} V(a,b;x) \; + \; 2\,\frac{b}{a}\,(1 \; + \; 2\,\frac{b}{a}\,x)\,V(a,b;x) \; - \; (a \; + \; 4\,\frac{b}{a})\,V^2(a,b;x) \; = \; 0, \tag{24}$$

with the initial condition $V(a,b;0) = 1$

Hence the coefficient functions from eq. 9 are at most first degree polynomials, namely $\alpha(x) \equiv \alpha(a,b;x) = (a + 2\,b\,x)/(a^2 + 4\,b)$ and $\beta(x) \equiv \beta(a,b;x) = -4\,b/(a^2 + 4\,b)$ in the *Fibonacci* case, and $\alpha(x) \equiv \alpha(a,b;x) = (1 + 2\,b\,x/a)^2/(a + 4\,b/a)$ and $\beta(x) \equiv \beta(a,b;x) = -2\,(b/a)\,(1 + 2\,b\,x/a)/(a + 4\,b/a)$ in the *Lucas* case, provided $a \neq 0$ and $a^2 + 4\,b \neq 0$.

The degenerate case $D(a,b) := a^2 + 4\,b = 0$, for which the above given differential eqs. become linear, will be considered separately. This case corresponds to vanishing $f_0(x)$ in *section 1.*

From the general results given in *section 1* the generating functions for the $k-$th convolution of these sequences satisfy

$$U^{k+1}(a,b;x) = \frac{1}{(a^2 + 4\,b)\,k}\left((a + 2\,b\,x)\,\frac{\partial}{\partial x} + 4\,k\,b\right) U^k(a,b;x)\ , \tag{25}$$

and

$$V^{k+1}(a,b;x) = \frac{a}{(a^2 + 4\,b)\,k}\left((1 + 2\,\frac{b}{a}\,x)^2\,\frac{\partial}{\partial x} + 2\,k\,\frac{b}{a}(1 + 2\,\frac{b}{a}\,x)\right) V^k(a,b;x)\ . \tag{26}$$

This implies, from eq. 5, that the $k-$th convolution $U_n^{(k)}$, defined by $U^{k+1}(a,b;x) =: \sum_{n=0}^{\infty} U_n^{(k)}(a,b)\,x^n$ can be expressed in terms of the $k-1$-st one according to

$$U_n^{(k)}(a,b) = \frac{1}{k\,(a^2 + 4\,b)}\left(a\,(n+1)\,U_{n+1}^{(k-1)}(a,b) + 2\,b\,(n+2\,k)\,U_n^{(k-1)}(a,b)\right), \tag{27}$$

with input $U_n^{(0)}(a,b) = U_n(a,b)$, and similarly

$$\begin{aligned} V_n^{(k)}(a,b) = \frac{1}{k\,a\,(a^2 + 4\,b)}\Big((n+1)\,a^2\,V_{n+1}^{(k-1)}(a,b) + 2\,a\,b\,(2\,n + k)\,V_n^{(k-1)}(a,b) \\ + 4\,b^2\,(n + k - 1)\,V_{n-1}^{(k-1)}(a,b)\Big)\ , \end{aligned} \tag{28}$$

with input $V_n^{(0)}(a,b) = V_n(a,b)$. The formula given in eq. 27 has been found earlier in [1] (p. 202, III and p.213, eq. (30)) without using the defining *Riccati* eq. for $U(a,b;x)$. The notations have to be translated with the help of $F_n^{(k)} \mathrel{\hat{=}} U_n^{(k-1)}$, $a_1 \mathrel{\hat{=}} a$, and $a_2 \mathrel{\hat{=}} b$.

For example, the convolution of $\{V_n(a,b)\}_0^\infty$ with itself ($k=1$) becomes, after use of recursion eq. 11

$$V_n^{(1)}(a,b) \;=\; \frac{1}{a\,(a^2\,+\,4\,b)}\left(\left[a^2\,(n+1)\;\;+\;\;4\,b\,n\right]V_{n+1}(a,b)\;\;+\;\;2\,b\,a\,V_n(a,b)\right)\,. \tag{29}$$

For $a=b=1$ one recovers well-known formulae for the first convolutions of ordinary *Fibonacci*, *resp. Lucas* numbers (*e.g.* [7], p.183, eqs. (98) and (99) (with corrected $L_{n-1}\to L_{n-i}$)). To see this, observe that $U_n^{(1)}(1,1) \;=\; F_{n+2}^{(1)}$ and $V_n^{(1)}(1,1) \;=\; L_{n+2}^{(1)} - 4\,L_{n+2}$.

$$F_n^{(1)} \;\;=\;\; U_{n-2}^{(1)}(1,1) \;=\; \frac{1}{5}\left((n-1)\,F_n \;\;+\;\;2\,n\,F_{n-1}\right) \;=\; \frac{1}{5}\,(n\,L_n \;-\; F_n) \tag{30}$$

$$L_n^{(1)} \;\;=\;\; V_{n-2}^{(1)}(1,1) \;+\; 4\,L_n \;=\; \frac{1}{5}\left((5\,n-9)\,L_n \;+\;2\,L_{n-1}\right) \;+\; 4\,L_n \;=\; (n+2)\,L_n \;\;+\;\; F_n\;. \tag{31}$$

We note, in passing, a sum representation of these convolutions obtained from the expansion of the generating functions which is valid for $k\in\mathbb{N}_0$.

$$U_n^{(k)}(a,b) \;\;=\;\; \sum_{l=0}^{\lfloor\frac{n}{2}\rfloor}\binom{k+n-l}{k}\binom{n-l}{l}\,a^{n-2l}\,b^l, \tag{32}$$

$$V_n^{(k)}(a,b) \;\;=\;\; \sum_{p=0}^{min(n,k+1)}2^p\binom{k+1}{p}\sum_{l=p}^{n}\binom{n-l+k}{k}\binom{n-l}{l-p}\,a^{n-2l}\,b^l. \tag{33}$$

Before discussing iteration of recursion relations 27 and 28 we state results for the degenerate case $D(a,b) \;:=\; a^2\;+\;4\,b\;=\;0$. *Riccati* eqs. 23 and 24 collapse to linear differential eqs. for $U(a;x) \;:=\; U(a,-a^2/4;x)$ and $V(a;x) \;:=\; V(a,-a^2/4;x)$

$$(1\;-\;\frac{a}{2}\,x)\,\frac{\partial}{\partial x}U(a;x) \;\;=\;\; a\,U(a;x) \quad,\quad U(a;0)\;=\;1, \tag{34}$$

$$(1\;-\;\frac{a}{2}\,x)\,\frac{\partial}{\partial x}V(a;x) \;\;=\;\; \frac{a}{2}\,V(a;x) \quad,\quad V(a;0)\;=\;1\;. \tag{35}$$

For the last eq. $x\neq 2/a$ was assumed. Because the solutions to these eqs. imply

$$\frac{\partial^2}{\partial x^2}U(a;x) \;=\; \frac{3}{2}\,a^2\,U^2(a;x) \quad,\quad \frac{\partial}{\partial x}V(a;x) \;=\; \frac{a}{2}\,V^2(a;x)\;, \tag{36}$$

the corresponding first ($k=1$) convolutions of these numbers $U_n(a) \;:=\; U_n(a,-a^2/4)$ and

$V_n(a) := V_n(a, -a^2/4)$ are given by

$$U_n^{(1)}(a) \;=\; \frac{2}{3\,a^2}\,(n\,+\,2)\,(n\,+\,1)\,U_{n+2}(a) \quad , \quad V_n^{(1)}(a) \;=\; \frac{2}{a}\,(n\,+\,1)\,V_{n+1}(a)\;, \tag{37}$$

with eqs. 21 and 22.

In order to derive the result for the $k-$th convolution we start with identities which follow from the solutions of eqs. 34 and 35, namely

$$U^{k+1}(a;x) \;=\; \frac{2}{a^2\,k\,(2\,k+1)}\,\frac{\partial^2}{\partial x^2}\Big(U^k(a;x)\Big)\;, \tag{38}$$

$$V^{k+1}(a;x) \;=\; \frac{2}{a\,k}\,\frac{\partial}{\partial x}\Big(V^k(a;x)\Big)\;. \tag{39}$$

These identities imply for the $k-$th convolutions

$$U_n^{(k)}(a) \;=\; \frac{2}{a^2\,k\,(2\,k+1)}\,(n+2)\,(n+1)\,U_{n+2}^{(k-1)}(a)\;, \tag{40}$$

$$V_n^{(k)}(a) \;=\; \frac{2}{a\,k}\,(n+1)\,V_{n+1}^{(k-1)}(a)\;, \tag{41}$$

with inputs $U_n^{(0)}(a) = U_n(a) = (n+1)\,(a/2)^n$ and $V_n^{(0)}(a) = V_n(a) = (a/2)^n$. See eqs. 19 and 20. The iteration of these eqs. yields the final result, which for $k \in \mathbb{N}_0$, and in the degenerate case $b = -a^2/4$, is

$$U_n^{(k)}(a) \;=\; \binom{n+2\,k+1}{2\,k+1}\,\Big(\frac{a}{2}\Big)^n\;, \tag{42}$$

$$V_n^{(k)}(a) \;=\; \binom{n+k}{k}\,\Big(\frac{a}{2}\Big)^n\;. \tag{43}$$

Thus $V_n^{(2l+1)}(a) = U_n^{(l)}(a)$, and it suffices to treat $V_n^{(k)}(a)$. For even a these are non-negative integer sequences. For $n, k \in \mathbb{N}_0$, $V_{n+k}^{(k)}(2\,l)$ constitutes a convolution triangle of numbers based on the $k = 0$ column sequence $V_n^{(0)}(2\,l) = l^n$ (powers of l). See [5] for these triangles of numbers.

In the non-degenerate case recursion eq. 27 can be iterated in order to express the $k-$th convolution

of $U_n(a,b)$ as linear combination of these numbers according to

$$U_n^{(k)}(a,b) \;=\; \frac{1}{k!\,(a^2+4\,b)^k}\left(AU_{k-1}(a,b;n)\,(n+1)\,a\,U_{n+1}(a,b) \;+\; BU_{k-1}(a,b;n)\,(n+2)\,b\,U_n(a,b)\right)\,, \tag{44}$$

with certain polynomials $AU_{k-1}(a,b;n)$ and $BU_{k-1}(a,b;n)$ of degree $k-1$ in the variable n, for arbitrary, but fixed, $a\neq 0$, $b\neq 0$, and $b\neq -a^2/4$.

The (mixed) recursion relations for these polynomials are deduced from eq. 27, and for $k=1,2,...$, they are

$$\begin{aligned} AU_k(a,b;n) \;&=\; a^2\,(n+2)\,AU_{k-1}(a,b;n+1) \;+\; 2\,b\,(n+2(k+1))\,AU_{k-1}(a,b;n) \;+ \\ &\quad\; b\,(n+3)\,BU_{k-1}(a,b;n+1)\,, \end{aligned} \tag{45}$$

$$BU_k(a,b;n) \;=\; a^2\,(n+1)\,AU_{k-1}(a,b;n+1) \;+\; 2\,b\,(n+2(k+1))\,BU_{k-1}(a,b;n)\,, \tag{46}$$

with inputs $AU_0(a,b;n)=1$ and $BU_0(a,b;n)=2$.

In eqs. (26), *resp.* (27), of [1] one can find explicit results for $U_n^{(k)}(a,b)$ for the instances $k=2$, *resp.* $k=3$ (in eq. (26) of this ref. one has to multiply the *lhs* with 2!, and in the second line of N of eq.(27) it should read $B(2,n+1)$).

For the case $a=1=b$ the triangles of the coefficients of these polynomials can be viewed under the nrs. A057995 and A057280 in [5]. For $a=2$, $b=1$ see A058402 and A058403, and for $a=1$, $b=2$ A073401 and A073402.

Similarly, iteration of recursion eq. 28 results, with the help of recursion eq. 11, in

$$V_n^{(k)}(a,b) \;=\; \frac{1}{k!\,a\,(a^2+4\,b)^k}\left(AV_k(a,b;n)\,V_{n+1}(a,b) \;+\; BV_k(a,b;n)\,V_n(a,b)\right)\,, \tag{47}$$

with certain polynomials $AV_k(a,b;n)$ and $BV_k(a,b;n)$ of generic degree k in the variable n, for fixed $a\neq 0$, $b\neq 0$, with $b\neq -a^2/4$.

The (mixed) recursion relations for these polynomials are found from eq. 28, and for $k = 1, 2, ...,$ they are

$$\begin{aligned} AV_k(a,b;n) &= a^2\,(n+1)\,AV_{k-1}(a,b;n+1) \;+\; 2\,b\,(2\,n+k)\,AV_{k-1}(a,b;n) \;+ \\ & \quad a\,(n+1)\,BV_{k-1}(a,b;n+1) \;+\; 4\,\frac{b}{a}\,(n+k-1)\,BV_{k-1}(a,b;n-1)\;, \qquad (48) \\ BV_k(a,b;n) &= 2\,b\,(2\,n+k)\,BV_{k-1}(a,b;n) \;-\; 4\;b\,(n+k-1)\,BV_{k-1}(a,b;n-1) \;+ \\ & \quad a\,b\,(n+1)\,AV_{k-1}(a,b;n+1) \;+\; 4\,\frac{b^2}{a}\,(n+k-1)\,AV_{k-1}(a,b;n-1)\;, \qquad (49) \end{aligned}$$

with inputs $AV_0(a,b;n) = 0$ and $BV_0(a,b;n) = a$.

For $a = 1 = b$ the triangles of coefficients of these polynomials in n can be found under the nrs. A061188 and A061189 in [5]. Observe that $BV_1(1,1;n)$ is accidentally of degree 0. For $a = 2, b = 1$ see nrs. A062133 and A062134.

Motivated by a recent paper [6] we consider also the following generalized *p−Fibonacci* numbers $U_n(p;a,b)$ defined for $p \in \mathbb{N}_0$ by the generating function

$$U(p;a,b;x) \;:=\; \frac{1}{1 \;-\; a\,x \;-\; b\,x^{p+1}} \;=\; \sum_{n=0}^{\infty} U_n(p;a,b)\,x^n\;. \qquad (50)$$

Of course, we assume $b \neq 0$ and also take $a \neq 0$. For $p = 1$ these numbers reduce to the $U_n(a,b)$ treated above, and for $p = 0$ they become the powers $(a+b)^n$. $U(p;1,1;x)$ appears in eq. 71 of [2]. The recursion relations are

$$U_n(p;a,b) \;=\; a\,U_{n-1}(p;a,b) \;+\; b\,U_{n-(p+1)}(p;a,b)\;, \qquad (51)$$

with inputs $U_j(p;a,b) \;=\; a^j$ for $j = 0, 1, ..., p$. In order to derive expressions for the $k-$th convolution of these *p−Fibonacci* numbers consider first the following *Riccati* eq. of type 8 satisfied by $U(p;a,b;x)$ written for the non-degenerate case $D(p;a,b) \;:=\; (p+1)^{p+1}\,b \;+\; a\,(a\,p)^p \;\neq\; 0$ if $p \in \mathbb{N}$, and $a + b \neq 0$ if $p = 0$ (*i.e.* one puts $(a\,p)^p \;=\; 1$ if $p = 0$).

$$U^2(p;a,b;x) \;=\; \frac{1}{(p+1)^{p+1}\,b \;+\; a\,(a\,p)^p}\left\{A_p(a,b;x)\,\frac{\partial}{\partial x} \;+\; b\,(p+1)^2\,B_{p-1}(a;x)\right\}U(p;a,b;x)\;, \qquad (52)$$

with

$$A_p(a,b;x) \;=\; (a\,p)^p \;+\; b\,(p+1)\,x\,B_{p-1}(a;x)\;, \tag{53}$$

$$B_{p-1}(a;x) \;=\; (p+1)^{p-1}\sum_{i=0}^{p-1}\Big(\frac{a\,p}{p+1}\Big)^i\,x^i \;=\; \frac{(p+1)^p \;-\; (a\,p\,x)^p}{p+1 \;-\; a\,p\,x}. \tag{54}$$

Hence, the coefficient functions $\alpha(x)$, *resp.* $\beta(x)$ from the general set-up in *section 1* are polynomials of degree p *resp.* $p-1$, namely $\alpha(x) \equiv \alpha_p(a,b;x) = A_p(a,b;x)/D(p;a,b)$ *resp.* $\beta(x) \equiv \beta_p(a,b;x) = -b\,(p+1)^2\,B_{p-1}(a;x)/D(p;a,b)$, and $\gamma(x) \equiv 0$. For $p=0$ one has to use $A_0(a,b;x) = 1$ and $B_{-1}(a;x) = 0$. For given non-vanishing a and b these polynomials $A_p(a,b;x)$, *resp.* $B_{p-1}(a;x)$, in the variable x of degree p, *resp.* $p-1$, have therefore the following explicit form.

$$A_p(a,b;x) \;=\; \sum_{m=0}^{p} A(a,b;p,m)\;x^m \quad , \quad B_{p-1}(a;x) \;=\; \sum_{m=0}^{p-1} B(a;p-1,m)\;x^m\;, \tag{55}$$

with the coefficients

$$A(a,b;p,m) \;=\; \begin{cases} 0 & \text{if } m \;>\; p, \\ 1 & \text{if } m = 0 \text{ and } p = 0, \\ (a\,p)^p & \text{if } m = 0 \text{ and } p \geq 1, \\ b\,(p+1)^p\,\Big(\frac{a\,p}{p+1}\Big)^{m-1} & \text{if } m \geq 1\;. \end{cases} \tag{56}$$

$$B(a;p,m) \;=\; \begin{cases} 0 & \text{if } p \;<\; m \text{ or } p = -1\;, \\ (p+2)^p\,\Big(\frac{a\,(p+1)}{p+2}\Big)^{m} & \text{if } p \geq m \geq 0\;. \end{cases} \tag{57}$$

For $a = 1 = b$ these triangles of coefficients can be viewed under the numbers $A055858$ and $A055864$ in [5] where further details may be found.

Even though we cannot compute the integral in the solution eq. 10 of the linear differential eq. 9, which is equivalent to *Riccati* eq. 52 for $p \neq 0, 1$, $H(x) = 1 - a\,x - b\,x^{p+1}$ is the unique solution due to the existence and uniqueness theorem for the linear first order differential eq. 9 with initial value $H(p;a,b;0) = 1$.

The result for the first ($k = 1$) convolution of the numbers $U_n(p; a, b)$ which flows from *Riccati* eq. 52 is

$$U_n^{(1)}(p; a, b) \;=\; \frac{1}{b\,(p+1)^{p+1} \;+\; a\,(a\,p)^p} \sum_{j=0}^{p} C_j(n; p; a, b)\, U_{n+1-j}(p; a, b)\;, \tag{58}$$

with

$$C_j(n; p; a, b) \;=\; \begin{cases} n+1 & \text{if } p = 0 = j, \\ (n+1)\,(a\,p)^p & \text{if } p \geq 1 \text{ and } j = 0\,, \\ b\,(p+1)^p\,(n+p+2-j)\left(\frac{a\,p}{p+1}\right)^{j-1} & \text{if } p \geq 1 \text{ and } j = 1, ..., p. \end{cases} \tag{59}$$

The $U_n(p; a, b)$ recursion cannot be used to simplify the sum in eq. 58.

This result can now be compared, after putting $a = 1 = b$, with a different formula for the same convolution found in [6], eq.(14). For given $p \in \mathbb{N}_0$ and $k = 2, 3, ...$, the recursion for $F_p^{(2)}(k) \hat{=} U_{k-2}^{(1)}(p; 1, 1)$ in [6] involves all $k-1$ terms $F_p(n) \hat{=} U_{n-1}(p; 1, 1)$, for $n = 1, ..., k-1$, whereas our result needs only $p+1$ terms for all k. For example, $F_3^{(2)}(7) \hat{=} U_5^{(1)}(3; 1, 1)$ is reduced to six terms involving $F_3(1) \hat{=} U_0(3; 1, 1)$,..., $F_3(6) \hat{=} U_5(3; 1, 1)$ in [6], but only to four terms, involving $U_8(3; 1, 1) \hat{=} F_3(9)$, $U_7(3; 1, 1) \hat{=} F_3(8)$,..., $U_5(3; 1, 1) \hat{=} F_3(6)$ in eq. 58.

For the $k-$th convolution we use eq. 4 with $\gamma(x) \equiv 0$ and the above given functions $\alpha_p(a, b; x)$ and $\beta_p(a, b; x)$. For the non-degenerate case, and for $k \in \mathbb{N}$, we have

$$U^{k+1}(p; a, b; x) \;=\; \frac{1}{k\left(b\,(p+1)^{p+1} \;+\; a\,(a\,p)^p\right)} \left\{ A_p(a, b; x)\,\frac{\partial}{\partial x} \;+\; k\,b\,(p+1)^2\,B_{p-1}(a; x) \right\} U^k(p; a, b; x)\;. \tag{60}$$

For $p \in \mathbb{N}_0$ the corresponding recursion relation for the $k-$th convolution is (remember that we put $(a\,p)^p \;=\; 1$ if $p = 0$)

$$U_n^{(k)}(p; a, b) \;=\; \frac{1}{k\left(b\,(p+1)^{p+1} \;+\; a\,(a\,p)^p\right)} \sum_{j=0}^{p} C_j^{(k)}(n; p; a, b)\, U_{n+1-j}^{(k-1)}(p; a, b)\;. \tag{61}$$

with

$$C_j^{(k)}(n;p;a,b) \;=\; \begin{cases} n+1 & \text{if } p=0=j, \\ (n+1)\,(a\,p)^p & \text{if } p\geq 1 \text{ and } j=0\,, \\ b\,(p+1)^p\,(n+1+k\,(p+1)-j)\left(\frac{a\,p}{p+1}\right)^{j-1} & \text{if } p\geq 1 \text{ and } j=1,...,p. \end{cases} \tag{62}$$

Instead of showing the rather unwieldy formula for the iteration of this recursion relation, after employing the fundamental recursion eq. 51, we prefer to state the result for the instance $p=2, k=2, a=1=b$, with the notation $U_n^{(1)}(2;1,1) \equiv U_n^{(1)}(2)$ and $U_n(2;1,1) \equiv U_n(2)$:

$$\begin{aligned} U_n^{(2)}(2) \;&=\; \frac{1}{2\cdot 31}\left(4\,(n+1)\,U_{n+1}^{(1)}(2) \;+\; 9\,(n+6)\,U_n^{(1)}(2) \;+\; 6\,(n+5)\,U_{n-1}^{(1)}(2)\right) \\ &=\; \frac{1}{2\cdot 31^2}\Big(\,(217\,n^2 \;+\; 1425\,n \;+\; 1922)\,U_n(2) \;+\; 2\,(n+2)\,(62\,n \;+\; 305)\,U_{n-1}(2) \;+ \\ &\qquad\qquad 4\,(n+1)\,(31\,n+143)\,U_{n-2}(2)\Big)\;. \end{aligned} \tag{63}$$

Recursion relation eq. 51 has been used twice.

In the degenerate case $D(p;a,b) := (p+1)^{p+1}\,b \;+\; a\,(a\,p)^p \;=\; 0$ (where we put $(a\,p)^p \equiv 1$ if $p=0$) we find for $U(p;a,b=b(p;a);x) =: U(p;a;x)$, where $b(p;a) := -p^p\,(a/(p+1))^{p+1}$, the linear differential eq.

$$\left\{A_p(a,b(p;a);x)\,\frac{\partial}{\partial x} \;+\; b(p;a)\,(p+1)^2\,B_{p-1}(a;x)\right\}U(p;a;x) \;=\; 0 \tag{64}$$

with B_{p-1} and A_p taken in their explicit form known from eqs. 55 with 57 and 56. For general p and $D(p;a,b) = 0$ we cannot say anything about convolutions because we have no suitable expression for $U^2(p;a,b;x)$. Recurrence eq. 51 with depth $p+1$ can be replaced by one with only depth p. See eq. 65.

In the non-degenerate case we could also consider the other p linear independent (*Lucas*-type) sequences defined by recurrence eq. 51 with appropriate inputs, but we will not do this here.

The remainder of this paper provides proofs for the above given statements.

3 Riccati equations for Fibonacci and Lucas generating functions

Proposition 1: $U(a,b;x)$ defined in eq. 12 is for $a^2 + 4\,b \neq 0$ equivalent to *Riccati* eq. 23 with initial condition $U(a,b;0) = 1$.

Proof: a) $H(a,b;x) = 1/U(a,b;x) = 1 - a\,x - b\,x^2$ satisfies eq. 9 with $\alpha(x) \equiv \alpha(a,b;x) = (a + 2\,b\,x)/(a^2 + 4\,b)$ and $\beta(x) \equiv \beta(a,b;x) = -4\,b/(a^2 + 4\,b)$. Therefore, $U(a,b;x)$ obeys eq. 8 which coincides with eq. 23.

b) With $\alpha(a,b;x)$ and $\beta(a,b;x)$ from eq. 23, as given in part a) we can compute the integral in eq. 10 and determine the constant C from the initial condition. This produces $1/U(a,b;x)$. □

Lemma 1: In the degenerate case $U(a;x) := U(a,-a^2/4;x)$ yields the first order linear differential eq. 34 as well as the second order non-linear differential eq. given as the first of eqs. 36.

Proof: Elementary. □

Note 1: The degenerate case is equivalent to $\lambda_+(a,b) = \lambda_-(a,b)$ with the definition of the characteristic roots of the recursion relation eq. 11 given in eq. 14. We may assume that not both, a and b, vanish and $x \neq 1/\lambda_\pm(a,b) = -\lambda_\mp/b$. In each case $U(a,b;0) = 1$.

Proposition 2: $V(a,b;x)$, defined in eq. 13 for $a \neq 0$, is for $a^2 + 4\,b \neq 0$ equivalent to *Riccati* eq. 24 with initial condition $V(a,b;0) = 1$.

Proof: Analogous to the proof of *Proposition 1.* □

Lemma 2: In the degenerate case $V(a;x) := V(a,-a^2/4;x)$ satisfies the first order linear differential eq. 35 as well as the first order non-linear differential eq. given as the second of eqs. 36.

In each case $V(a,b,0) = 1$.

Proof: Elementary. □

4 Convolutions of generalized Fibonacci and Lucas sequences

Because the $k+1$st power of the (ordinary) generating functions of a sequence generates $k-$fold convolutions of this sequence we obtain in the non-degenerate case, $a^2+4\,b \neq 0$, according to the general set-up of *section 1*, for the generalized *Fibonacci resp. Lucas* case, expression eq. 27, *resp.* eq. 28. For the definition of the $k-$th convolutions $U_n^{(k)}(a,b)$ (similarly of $V_n^{(k)}(a,b)$) see the line after eq. 26. The first convolutions ($k=1$) can be determined in each case from linear combinations of the two independent original sequences. See eq. 29 for the *Lucas* case. For $a=1=b$ these formulae are well-known (see *section 2* after eq. 29).

Lemma 3 (Recurrence for $k-$fold convolution, degenerate case):

For $b=-\frac{a^2}{4} \neq 0$ the recurrence formulae for the $k-$fold convolution of the generalzed *Fibonacci*,*resp. Lucas*, sequences are those stated in eqs. 40, *resp.* 41.

Proof: This statement is equivalent to eq. 25, *resp.* eq. 26 for the powers of the corresponding generating functions. They are deduced from the the second, *resp.* first, order differential eq., given in eqs. 36, which coincides with the $k=1$ assertion. To verify the general k case, eq. 38 *resp.* eq. 39, one may use $U(a;x)=U(a,-\frac{a^2}{4};x)=1/(1-a\,x/2)^2$ from eq. 12, *resp.* $V(a;x)=V(a,-\frac{a^2}{4};x)=1/(1-a\,x/2)$ from eq. 13. □

Lemma 4: The explicit form for the $k-$fold convolution in the degenerate case is given by eq. 42, *resp.* 43, for the generalized *Fibonacci*, *resp. Lucas*, case.

Proof: Iteration of recurrence eq. 40, *resp.* eq. 41, with input $U_n^{(0)}(a)=U_n(a)=(a/2)^n$, *resp.* $V_n^{(0)}(a)=V_n(a)=(a/2)^n$, which originates from the generating functions $U(a,-\frac{a^2}{4};x)$, *resp.* $V(a,-\frac{a^2}{4};x)$. □

Proposition 3 (Iteration of recurrence for $k-$fold convolutions; non-degenerate *Fibonacci* case):

For $a^2+4\,b \neq 0$ the $k-$fold convolution of the generalized *Fibonacci* sequence $\{U_n(a,b)\}$ is expressed

as linear combinations of the two independent solutions of recurrence eq. 11 as given in eq. 44. The coefficient polynomials $AU_k(a,b;n)$ and $BU_k(a,b;n)$ satisfy the mixed recurrence relations eqs. 45 and 46.

Proof: If one considers eq. 44 as *ansatz* and puts it into recurrence eq. 27 we find, after elimination of $U_{n+2}(a,b)$ *via* its recursion relation and a comparison of the coefficients of the linear independent $U_n(a,b)$ and $U_{n-1}(a,b)$ sequences, the mixed recurrence relations for $AU_k(a,b;n)$ and $BU_k(a,b;n)$. The inputs $AU_0(a,b;n) = 1$ and $BU_0(a,b;n) = 2$ are necessary in order that for $k = 1$ eq. 44 coincides with eq. 27. With these inputs and the mixed recurrence one proves, by induction over k, that $AU_k(a,b;n)$ and $BU_k(a,b;n)$ are polynomials in n of degree k, provided a and b are fixed with $b \neq -a^2/4$, $b \neq 0$ and $a \neq 0$. □

Note 2: For fixed integers a and $b \neq -a^2/4$ the coefficients of the polynomials $AU_n(a,b;x)$ and $BU_n(a,b;x)$ furnish two lower triangular (infinite) integer matrices. For the ordinary *Fibonacci* case $a = 1 = b$ these positive integer triangles can be found in [5] under the nrs. A057995 and A057280. For the *Pell* case $a = 2, b = 1$ see nrs. A058402 and A058403, and for the case $a = 1, b = 2$ see nrs. A073401 and A073402.

Proposition 4 (Iteration of recurrence for $k-$fold convolutions; non-degenerate *Lucas* case):

For $a^2 + 4b \neq 0$ the $k-$fold convolution of the generalized *Lucas* sequence $\{V_n(a,b)\}$ is expressed as linear combination of the two independent solutions of recurrence eq. 11 as given in eq. 47. The coefficient polynomials $AV_k(a,b;n)$ and $BV_k(a,b;n)$ satisfy the mixed recurrence relations eq. 48 and eq. 49.

Proof: Analogous to the proof of *Proposition 3.* □

Note 3: For fixed integers a and $b \neq -a^2/4$ the coefficients of the polynomials $AV_n(a,b;x)$ and $BV_n(a,b;x)$ furnish two lower triangular (infinite) integer matrices. For the ordinary *Lucas* case $a = 1 = b$ these positive integer triangles can be found in [5] under the nrs. A061188 and A061189. For the *Pell*

case $a = 2, b = 1$ see nrs. A062133 and A062134.

5 Convolutions of generalized $p-$Fibonacci sequences

Generalized $p-$*Fibonacci* numbers $U_n(p; a, b)$ are defined by eq. 51 for $p \in \mathbb{N}_0$, $b \neq 0$ and $a \neq 0$, together with the inputs $U_j(p; a, b) := a^j$ for $j = 0, ..., p$. For $p = 1$ we recover the generalized *Fibonacci* numbers $U_n(a, b)$ treated above.

Lemma 5: The generating function $U(p; a, b; x)$ for the generalized $p-$*Fibonacci* numbers is given by eq. 50.

Proof: From the recurrence with inputs given in eq. 51. □

Lemma 6 (*Riccati* eq. for the generalized $p-$Fibonacci case):

If $D(p; a, b) := (p+1)^{p+1}\, b + a\,(a\,p)^p \neq 0$ (non-degenerate case) then $U(p; a, b; x)$ satisfies *Riccati* eq. 52 with the polynomials $A_p(a, b; x)$ and $B_{p-1}(a, b; x)$ defined in eqs. 53 and 54.

Proof: $H(p; a, b; x) = 1/U(p; a, b; x) = 1 - a\,x - b\,x^{p+1}$ satisfies eq. 9 with $\alpha(x) \equiv \alpha_p(a, b; x) = A_p(a, b; x)/D(p; a, b)$ and $\beta(x) \equiv \beta_p(a, b; x) = -b\,(p+1)^2\, B_{p-1}(a; x)/D(p; a, b)$ with $A_p(a, b; x)$ and $B_{p-1}(a; x)$ given by eq. 53 and 54. This is shown by comparing coefficients of powers x^i for $i = 0, 1, ..., 2\,p$. According to *section 1 Riccati* eq. 8 ensues which becomes eq. 52. □

Note 4: i) If $p = 0$, $U(0; a, b; x) = 1/(1 - (a + b)\,x)$ generates powers of $a + b$, and one has to put $A_0(a, b; x) \equiv 1$ and $B_{-1}(a; x) \equiv 0$. This means that one puts $(a\,p)^p = 1$ for $p = 0$.

ii) For given non-vanishing a and b $A_p(a, b; x)$ is a polynomial in x of degree p, and $B_{p-1}(a; x)$ is one of degree $p - 1$. The sum in $B_{p-1}(a; x)$ can be evaluated to yield the second of eqs. 54 provided $p \neq 0$.

Lemma 7 (Coefficient triangles of numbers for polynomials $A_p(a, b; x)$ and $B_p(a; x)$):

The coefficients of the polynomials defined in eqs. 55 are given by eqs. 56 and 57.

Proof: $B_p(a;x)$ from eq. 54 leads immediately to eq. 57, remembering that $B_{-1}(a;x) \equiv 0$. Then eq. 56 follows from eq. 53 and $A_0(a,b;x) \equiv 1$. $\square$

Proposition 5 (Uniqueness of *Riccati* solution; non-degenerate case):

If $D(a,b) \neq 0$ then $y \equiv U(p;a,b;x) = 1/(1 - a\,x - b\,x^{p+1})$ is the unique solution of *Riccati* eq. 52 with eqs. 53, 54 and initial value $U(p;a,b;0) = 1$.

Proof: From *section 1* we know that the *Riccati* eq. is equivalent to the inhomogeneous linear differential eq. for the inverse $H = 1/U$: $H' \equiv F(x,H) = (-\beta(x)/\alpha(x))\,H - 1/\alpha(x)$. Because $F(x,H)$ is continuous in the strip $0 \leq x \leq A < \infty$, $|H| < \infty$ and is there ($K = K(p;a,b;A)$)-*Lipschitz*, the existence and uniqueness theorem for linear differential eqs. proves the assertion (see *e.g.*[8], § 6,I, p.62ff). In order to find K we use the summed expression for B_{p-1} from eq. 54 and apply the triangle inequality repeatedly. $\square$

Proposition 6 (Recursion for $k-$th convolution of $\{U_n(p;a,b)\}$; non-degenerate case):

The $k-$th convolution of the sequence $\{U_n(p;a,b)\}$ is given in the non-degenerate case

$D(p;a,b) := b\,(p+1)^{p+1} + a\,(a\,p)^p \neq 0$ recursively by eq. 61 with eq. 62.

Proof: This follows from the general set-up of *section 1*, eq. 5 with $\gamma_{n-q} \equiv 0$ and the appropriate coefficient functions $\alpha(x) = \alpha_p(a,b;x)$ and $\beta(x) = \beta_p(a,b;x)$ given after eq. 54. See the corresponding eq. 60 for the $k+1$-st power of the generating function. $\square$

Lemma 8 (Degenerate case $D(p;a,b) = 0$):

If $D(p;a,b) := (p+1)^{p+1}\,b + a\,(a\,p)^p = 0$ then $U(p;a;x) = 1/(1 - a\,x - b(p;a)\,x^{p+1}) =$

$1/(1 - a\,x\,(p/(p+1)^p\,(a\,x)^p)$ satisfies the first order linear differential eq. 64.

Proof: We prove $(a + (p+1)\,b\,x^p)\,A_p(a,b;x) + b\,(p+1)^2\,(1 - a\,x - b\,x^{p+1})\,B_{p-1}(a;x) = 0$ with eqs. 54 and 53 in the version where the sum has been evaluated (the case $p = 0$ is treated separately). If we factor out $b/(p+1-a\,p\,x)$ we see that all terms cancel provided we replace $a\,(a\,p)^p$ by $-b\,(p+1)^{p+1}$.$\square$

Note 5: The solution $1/(1 - a\,x - b\,x^{p+1})$ of this linear differential eq. 64 with input $U(p; a, b; 0) = 1$ is unique. The proof is analogous to the one of *Proposition 5*.

Note 6: If $U(p; a, b; x) = 1/(1 - a\,x + (((a\,p\,x/(p+1))^{p+1})/p)$ we do not have a formula for $U^2(p; a, b; x)$, valid for all p, like in the non-degenerate case. Therefore, we cannot derive results for convolutions along the line shown above.

Lemma 9 (Recurrence in the degenerate case):

If $D(p; a, b) := (p+1)^{p+1}\,b + a\,(a\,p)^p = 0$ (and $b \neq 0$) then one can replace recurrence eq. 51 which has depth $p+1$, by the following one with depth $p \in \mathbb{N}$.

$$U_{n+1}(p; a) = \frac{a}{(p+1)\,(n+1)} \sum_{j=1}^{p} \left(\frac{a\,p}{p+1}\right)^{j-1} (n + p + 2 - j)\, U_{n+1-j}(p; a) \ , \tag{65}$$

where one uses the inputs $U_j(p; a) = a^j$ for $j = 0, 1, ..., p-1$.

Proof: This derives from the sum on the *rhs* of eq. 58 which now vanishes. If the coefficients C_j from eq. 59 are used with the replacement of $a\,(a\,p)^p$ by $-b\,(p+1)^{p+1}$ one arrives at the desired recurrence, after the common factor b has been dropped. The inputs are adopted from the original recurrence except that U_p can now be computed to be a^p. □

Acknowledgements

The author thanks Dr. L. Turban for sending him his preprint [6]. He also thanks Mr. M. Frank, Dr. T. Hahn and Mr. G. Jahn for advice on how to keep conversations with the machine going. An anonymous referee suggested to include the general background now found in the *Introduction*.

References

[1] J. Arkin and V. E. Hoggatt, Jr.: An Extension of the Fibonacci Numbers (Part II), The Fibonacci Quarterly **8** (1970),199-216

[2] M. Bicknell: A Primer for the Fibonacci Numbers (Part VIII), The Fibonacci Quarterly **9** (1971),74-81

[3] H. W. Gould: The Girard-Waring Power Sum Formulas for Symmetric Functions and Fibonacci Sequences, The Fibonacci Quarterly **37**, **2** (1999),135-140

[4] W. Lang: On Sums of Powers of Zeros of Polynomials, Journal of Computational and Applied Mathematics **89** (1998),237-256

[5] N.J.A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, 1995; see also N.J.A. Sloane's On-Line Encyclopedia of Integer Sequences, http//:www.research.att.com/~njas/sequences/index.html

[6] L. Turban: Lattice animals on a staircase and generalized Fibonacci numbers, Henri Poincaré Université, Nancy, France, preprint, May 2000; http://xxx.lanl.gov/form/cond-mat under Number 0106595

[7] S. Vajda: *Fibonacci & Lucas Numbers, and the Golden Section*, Ellis and Horwood Ltd., Chichester, 1989

[8] W. Walter: *Ordinary differential equations*, Springer, New York-Berlin-Heidelberg, 1998

[9] G. N. Watson: *A Treatise on the Theory of Bessel Functions*, 2nd ed., Cambridge University Press, Cambridge, 1958

AMS MSC numbers: 11B83, 11B38, 11C08

On Polynomials Related to Derivatives of the Generating Function of Catalan Numbers

Wolfdieter L a n g [1]

Institut für Theoretische Physik
Universität Karlsruhe
Kaiserstrasse 12, D-76128 Karlsruhe, Germany

1 Introduction and summary

In [3] it has been shown that powers of the generating function $c(x)$ of *Catalan* numbers $\{C_n\}_{n\in\mathbf{N}_0} = \{1,1,2,5,14,42,...\}$ where $\mathbf{N}_0 := \{0,1,2,...\}$ (nr.1459 and *A*000108 of [8], and references of [3]) can be expressed in terms of a linear combination of 1 and $c(x)$ with coefficients replaced by certain scaled *Chebyshev* polynomials of the second kind. In this paper derivatives of $c(x)$ are studied in a similar manner. The starting point is the following expression for the first derivative.

$$\frac{d\ c(x)}{dx} \equiv c'(x) \ = \ \frac{1}{x(1-4x)}\Big(1+(-1+2x)\ c(x)\Big)\ . \tag{1}$$

This equation is equivalent to the simple recurrence relation valid for C_n:

$$(n+2)\ C_{n+1} \ - \ 2(2n+1)\ C_n \ = \ 0 \ \ , \ \ n=-1,0,1,..., \ \text{with} \ \ C_{-1}=-1/2 \ \ . \tag{2}$$

Equation (1) can, of course, also be found from the explicit form $c(x) \ = \ (1-\sqrt{1-4x}\ )/(2x)$. The result for the n-th derivative is of the form

$$\frac{1}{n!}\frac{d^n c(x)}{dx^n} \ = \ \frac{1}{(x(1-4x))^n}\Big(a_{n-1}(x) \ + b_n(x)\ c(x)\Big), \tag{3}$$

with certain polynomials $a_{n-1}(x)$ of degree $n-1$ and $b_n(x)$ of degree n. These polynomials are found to be

$$b_n(x) \ \ = \ \ \sum_{m=0}^{n}(-1)^m B(n,m)\ x^{n-m}$$

with

$$B(n,m) \ := \ \binom{2n}{n}\binom{n}{m}\Big/\binom{2m}{m} \ \ , \tag{4}$$

which defines a triangle of numbers for $n,m\in\mathbf{N}$, $n\geq m\geq 0$, where $\mathbf{N} := \{1,2,3,...\}$. The first terms are depicted in *TAB. 1* with $B(n,m)=0$ for $n<m$. Another representation for the polynomials $b_n(x)$ is also found, *viz.*

$$b_n(x) \ = \ -2\sum_{k=0}^{n} C_{k-1}\ x^k\ (4x-1)^{n-k}\ . \tag{5}$$

[1] E-mail: wolfdieter.lang@physik.uni-karlsruhe.de, http://www-itp.physik.uni-karlsruhe.de/~wl

Equating both forms of $b_n(x)$ leads to a formula involving convolutions of Catalan numbers with powers of an arbitrary constant $\lambda := (4x-1)/x$. This formula is given in (31). Equation (5) reveals the generating function of the polynomials $b_n(x)$ because it is a convolution of two functional sequences. The result is

$$g_b(x;z) \ := \ \sum_{n=0}^{\infty} \ b_n(x) \ z^n \ = \ \frac{\sqrt{1-4xz}}{1+(1-4x) \ z} \ . \tag{6}$$

The other family of polynomials is

$$a_n(x) \quad = \quad \sum_{k=0}^{n} (-1)^k \ A(n+1,k+1) \ x^{n-k}$$

with the triangular array $A(n,m)$ defined for $m=0$ by $A(n,0)=C_n$, and for $n,m \in \mathbf{N}$ with $n \geq m > 0$ by the numbers

$$A(n,m) \ = \ \frac{1}{2}\binom{n}{m-1}\left[4^{n-m+1} \ - \ \binom{2n}{n} \Big/ \binom{2(m-1)}{m-1}\right] \ . \tag{7}$$

The first terms of this triangular array of numbers are shown in *TAB. 2* with $A(n,m)=0$ for $n<m$. Both results, (4) and (7), are solutions to recurrence relations which hold for $b_n(x)$ and $a_n(x)$ and their respective coefficients $B(n,m)$ and $A(n,m)$.

Another representation for the polynomials $a_n(x)$ is found to be

$$a_n(x) \ = \ \sum_{k=0}^{n} C_k \ x^k \ (4x-1)^{n-k} \ , \tag{8}$$

which shows that the generating function of these polynomials is

$$g_a(x;z) \ := \ \sum_{n=0}^{\infty} \ a_n(x) \ z^n \ = \ \frac{c(xz)}{1+(1-4x) \ z} \ . \tag{9}$$

Comparing (5) with (8) yields the following relation between these two types of polynomials

$$b_n(x) \ = \ (4x-1)^n \ - \ 2x \ a_{n-1}(x) \quad , \quad n \in \mathbf{N}_0 \quad \text{with} \ \ a_{-1}(x) \equiv 0 \ , \tag{10}$$

and between the coefficients

$$B(n,m) \ = \ \binom{n}{m} \ 4^{n-m} \ - \ 2 \ A(n,m+1) \ . \tag{11}$$

The triangle of numbers $A(n,m)$ is related to a rectangular array of integers $\hat{A}(n,m)$, with $\hat{A}(0,m) \equiv 1$, $\hat{A}(n,0) = -C_n$ for $n \in \mathbf{N}$, and for $n \geq m \geq 1$ by

$$A(n,m) \ = \ -\hat{A}(n-m,m) \ + \ 2^{2(n-m)+1}\binom{n-1}{m-1} \ , \tag{12}$$

or with (7), for $m \in \mathbf{N}, n \in \mathbf{N}_0$, by

$$\hat{A}(n,m) \ = \ \frac{1}{2}\binom{n+m}{n+1}\left[\binom{2(n+m)}{n+m} \Big/ \binom{2(m-1)}{m-1} \ - \ 4^{n+1} \ \frac{m-1}{n+m}\right] . \tag{13}$$

Part of the array $A(n,m)$ is shown in *TAB. 3*, where it is called $C4(n,m)$.

It turns out that the $m-$th column of the triangle of numbers $A(n,m)$ for $m = 0, 1, ...$ is determined by the generating function $c(x)(\frac{x}{1-4x})^m$. The $m-$th column of the triangle of numbers $B(n,m)$ for $m = 0, 1, ...,$ is generated by $\frac{1}{\sqrt{1-4x}}\ (\frac{x}{1-4x})^m$. This fact identifies the infinite dimensional matrices $\mathbf{A}$ and $\mathbf{B}$ as examples of *Riordan* matrices in the terminology of [7]. The matrix $\hat{\mathbf{A}}$ associated with $\hat{A}(n,m)$ is an example of a *Riordan* array.

Because differentiation of $c(x) = \sum_{k=0}^{\infty}\ C_k\ x^k$ leads to

$$\frac{1}{n!}\ \frac{d^n c(x)}{dx^n}\ =\ \sum_{k=0}^{\infty} C(n,k)\ x^k\ ,\ \text{with}\ C(n,k)\ :=\ \frac{1}{n!}\ \prod_{j=1}^{n}(k+j)\ C_{n+k}\ =\ \frac{(2(n+k))!}{n!k!(n+k+1)!}\ , \tag{14}$$

where $C(0,k) = C_k$, one finds, together with (3), the following identities, for $n \in \mathbf{N}$, $p \in \{0, 1, 2, ..., n-1\}$

$$(D1):\ \sum_{k=0}^{p}(-1)^k\ C_k \binom{n}{p-k} \Big/ \binom{2(n-p+k)}{n-p+k}\ =\ \frac{1}{2}\binom{n}{p+1}\left\{2^{2(p+1)} \Big/ \binom{2n}{n}\ -\ 1 \Big/ \binom{2(n-p-1)}{n-p-1}\right\}$$

$$=\ A(n,n-p) \Big/ \binom{2n}{n}\ , \tag{15}$$

and, for $n \in \mathbf{N}, k \in \mathbf{N}_0$,

$$(D2):\ \sum_{j=0}^{n}(-1)^j\ \left(\binom{n}{j} \Big/ \binom{2j}{j}\right)\ \sum_{l=0}^{k} 4^l \binom{n+l-1}{n-1}\ C_{k+j-l}\ =\ C(n,k) \Big/ \binom{2n}{n}\ . \tag{16}$$

The remainder of this paper provides proofs for the above statements.

2 Derivatives

The starting point is equation (1) which can either be verified from the explicit form of the generating function $c(x)$, or by converting the recursion relation (2) for *Catalan* numbers into an equation for their generating function. A computation of

$$\frac{1}{(n+1)!}\ \frac{d^{n+1}c(x)}{dx^{n+1}}\quad =\quad \frac{1}{n+1}\ \frac{d}{dx}\left(\frac{1}{n!}\ \frac{d^n c(x)}{dx^n}\right)$$

with (3) taken as granted and equation (1) produces the following mixed relations between the quantities $a_n(x)$ and $b_n(x)$ and their first derivatives, valid for $n \in \mathbf{N}_0$,

$$(n+1)\ a_n(x)\ =\ x(1-4x)\ a'_{n-1}(x)\ +\ b_n(x)\ +\ n(8x-1)\ a_{n-1}(x)\ , \tag{17}$$

$$(n+1)\ b_{n+1}(x)\ =\ x(1-4x)\ b'_n(x)\ +\ (-(n+1)+2(1+4n)x)\ b_n(x)\ , \tag{18}$$

with inputs $a_{-1}(x) \equiv 0$ and $b_0(x) \equiv 1$.

From (18) it is clear by induction that $b_n(x)$ is a polynomial of degree n. Again by induction, the same statement holds for $a_n(x)$ in (17). Therefore we write, for $n \in \mathbf{N}_0$,

$$a_n(x)\ =\ \sum_{k=0}^{n}(-1)^k\ a(n,k)\ x^{n-k}\ , \tag{19}$$

$$b_n(x) \;=\; \sum_{k=0}(-1)^k\; B(n,k)\; x^{n-k}\;, \tag{20}$$

with the triangular arrays of numbers $a(n,k)$ and $B(n,k)$ with row number n and column number $k \leq n$. The triangular array $a(n,k)$ will later be enlarged to another one which will then be called $A(n,k)$.

We first solve $b_n(x)$ in (18) by inserting (20) and deriving the recursion relation for the coefficients $B(n,m)$ after comparing coefficients of x^{n+1}, x^0, and x^{n-k} for $k = 0, 1, ..., n-1$.

$$x^{n+1}\;: \qquad (n+1)\; B(n+1,0) \;=\; 2(2n+1)\; B(n,0)\;, \tag{21}$$

$$x^{0}\;: \qquad B(n+1,n+1) \;=\; B(n,n)\;, \tag{22}$$

$$x^{n-k}\;: \qquad (n+1)\; B(n+1,k+1) \;=\; (k+1)\; B(n,k) \;+\; 2\Big(2(n+k)+3\Big)\; B(n,k+1)\;. \tag{23}$$

With the input $B(0,0) = 1$ one deduces from (21) for the leading coefficient of $b_n(x)$

$$B(n,0) \;=\; 2^n\frac{(2n-1)!!}{n!} \;=\; \frac{(2n)!}{n!\;n!} \;=\; \binom{2n}{n}\;, \tag{24}$$

and from (22)

$$B(n,n) \;\equiv 1\,, \quad \text{i.e.,} \quad b_n(0) = (-1)^n\;. \tag{25}$$

In (24) the double factorial $(2n-1)!! := 1\cdot 3\cdot 5\cdots(2n-1)$ appeared.

In order to solve (23) we conjecture from *TAB. 1* that for $n, m \in \mathbf{N}$

$$B(n,m) \;=\; 4\; B(n-1,m) \;+\; B(n-1,m-1)\;, \tag{26}$$

with input $B(n,0) = \binom{2n}{n}$ from (24).

If in (23) we use this conjecture, written with $n \to n-1,\; k \to m-1,$ we are led to consider the simple recursion

$$B(n,m) \;=\; \frac{n+1-m}{2(2m-1)}\; B(n,m-1)\;. \tag{27}$$

The solution of this recursion is, for $n, m \in \mathbf{N}_0$,

$$B(n,m) \;=\; \frac{1}{2^m(2m-1)!!}\frac{n!}{(n-m)!}\binom{2n}{n} \;=\; \frac{m!\;n!}{(2m)!\;(n-m)!}\binom{2n}{n} \;=\; \binom{2n}{n}\binom{n}{m}\Big/\binom{2m}{m}\;. \tag{28}$$

With the *Pochhammer* symbol $(a)_n := \Gamma(n+a)/\Gamma(a)$ this result can also be written as

$$B(n,m) \quad = \quad ((2m+1)/2)_{n-m}4^{m-n}/(n-m)!\;.$$

This result satisfies (21), *i.e.*, (24), as well as (22), *i.e.*, (25). It is also the solution to (23) provided we prove the conjecture (26) using $B(n,m)$ in (28). This can be done by using the equality $B(n,m) = \frac{(2n)!\;m!}{(2m)!\;n!\;(n-m)!}$ in (26). Thus we have proved:

Proposition 1: *We have*

$$b_n(x) \quad = \quad \sum_{k=0}^{n}(-1)^k\, B(n,k)\, x^{n-k}$$

where $B(n,k) \;=\; \binom{2n}{n}\,\binom{n}{k}\Big/\binom{2k}{k}\;.$

This triangle of numbers as shown in *TAB. 1* appears as A046521 in the database [8].

One can derive another explicit representation for the polynomials $b_n(x)$ by using (27) in (20):

$$(1-4x)\ b_n'(x)\ +\ 2(2n-1)\ b_n(x)\ +\ 2\binom{2n}{n}\ x^n\ =\ 0\ . \tag{29}$$

This leads, together with (18), to the following inhomogeneous recursion relation for $b_n(x)$.

$$b_{n+1}(x)\ =\ (4x-1)\ b_n(x)\ -\ 2C_n\ x^{n+1}\ \ ,\ \ b_0(x)\equiv 1\ . \tag{30}$$

Equation (29) can also be solved as a first order linear and inhomogeneous differential equation for $b_n(x)$.

Proposition 2: *We have*

$$b_n(x)\ \ =\ \ -2\sum_{k=0}^{n} C_{k-1}\ x^k\ (4x-1)^{n-k}\ ,$$

where the C_k'*s are the* Catalan *numbers for* $k\in\mathbf{N}_0$*, and* $C_{-1}=-1/2$.

Proof: Iteration of (30). □

Proposition 3: *The generating function* $g_b(x;z)\ :=\sum_{n=0}^{\infty}\ b_n(x)\ x^n$ *for* $\{b_n(x)\}$ *is given by* (14).

Proof: The alternative form of $b_n(x)$, given by (5), is a convolution of the functional sequences $\{-2C_{k-1}\ x^k\}_{n\in\mathbf{N}_0}$ and $\{(4x-1)^n\}_{n\in\mathbf{N}_0}$, with generating functions $1-2xz\ c(xz)=\sqrt{1-4xz}$ and $1/(1+(1-4x)z)$, respectively. Therefore, $g_b(x;z)$ is the product of these two generating functions. □

Comparing this alternative form (5) for $b_n(x)$ with the one given by (20), together with (28), proves the following identity in n and $\lambda:=(4x-1)/x$. The term $k=0$ in the sum (5) has been written separately.

Corollary 1 (Convolution of *Catalan* sequence and the sequence of powers of λ):

For $n\in\mathbf{N}$ *and* $\lambda\neq\infty$,

$$s_{n-1}(\lambda)\ :=\lambda^{n-1}\ \sum_{k=0}^{n-1}\ \frac{C_k}{\lambda^k}\ =\frac{1}{2}\left(\lambda^n\ -\ \binom{2n}{n}\ \sum_{k=0}^{n}\ (-1)^k\ (4-\lambda)^k\ \binom{n}{k}\ \Big/\binom{2k}{k}\right)\ . \tag{31}$$

Therefore, the generating function for the sequence $s_n(\lambda)$ is

$$g(\lambda;x)\ \ :=\ \ \sum_{n=0}^{\infty}\ s_n(\lambda)\ x^n\ =\ c(x)/(1-\lambda\ x)\ .$$

From the generating function the recurrence relation is found to be $s_n(\lambda)=\lambda\ s_{n-1}(\lambda)+C_n$, $s_{-1}(\lambda)\equiv 0$. The connection with the polynomial $b_n(x)$ is $s_n(\lambda)\ =\ \frac{1}{2}\Big(\lambda^{n+1}\ -\ (4-\lambda)^{n+1}\ b_{n+1}(1/(4-\lambda))\Big)$.

The case $\lambda=0$ $(x=1/4)$ is also covered by this formula. It produces from $s_n(0)=C_n$ the following identity.

Example 1: Case $\lambda=0$ $(x=1/4)$

$$\sum_{k=0}^{n}\ (-1)^{k+1}\ \binom{n}{k}\ 4^k\ \Big/\binom{2k}{k}\ \ =\ \ \frac{1}{2n-1}\ . \tag{32}$$

This identity occurs in one of the exercises 2.7, 2, p. 32 of [4].

We note that from (5) one has $-2b_{n+1}(1/4) = C_n/4^n$. The large n behaviour of this sequence is known to be $C_n/4^n \sim \frac{1}{\sqrt{\pi}}\ \frac{1}{n^{3/2}}$; *cf.* [2], Exercise 9.60.

If one puts in (5) $4x - 1 = x$, *i.e.* $x = 1/3$, one can identify the partial sum $s_n(1)$ of *Catalan* numbers:

$$s_n(1) \ := \ \sum_{k=0}^{n} \ C_k \ = \ \frac{1}{2}(1 \ - \ 3^{n+1} \ b_{n+1}(1/3) \ ). \tag{33}$$

This sequence $\{1, 2, 4, 9, 23, 65, 197, 626, 2056, ...\}$ appears as A014137 in the web encyclopedia [8]. If one puts $\lambda = 1$ in *Corollary 1* one also finds the following

Example 2:

$$2 \ s_{n-1}(1) \ = \ 1 \ + \ \binom{2n}{n} \ \sum_{k=0}^{n} \ (-1)^{k+1} \ \binom{n}{k} \ 3^k \ \Big/ \binom{2k}{k} \ . \tag{34}$$

Another interesting example is the case $\lambda = 4$ ($x = \infty$). Here one finds a simple result for the convolution of *Catalan*'s sequence with powers of 4, *viz.*

Example 3: $\lambda = 4 \ \ (x = \infty)$

$$2 \ s_{n-1}(4) \ = \ 4^n \ - \ \binom{2n}{n} \ . \tag{35}$$

This sequence $\{1, 5, 22, 93, 386, 1586, 6476, ...\}$ appears in the book [8] as Nr. 3920 and as A000346 in the web encyclopedia. It will show up again in this work as $A(n+1, 1)$, the second column in the $A(n, m)$ triangle (*cf. TAB. 2*).

The sequence for $\lambda = -1$ $(x = 1/5)$ is also non-negative, as can be seen by writing $s_{2k}(-1) = C_2 + \sum_{l=2}^{k}(C_{2l} - C_{2l-1})$ for $k \in \mathbf{N}$ and $s_{2k+1}(-1) = \sum_{l=1}^{k}(C_{2l+1} - C_{2l})$, and using $\triangle C_n := C_n - C_{n-1} = 3\frac{n-1}{n+1} \ C_{n-1} \geq 0$. This is the sequence $\{1, 0, 2, 3, 11, 31, 101, 328, 1102, 3760, ...\}$ which appears now as A032357 in the web encyclopedia [8] .

Recursion (26) for $B(n, m)$ can be transformed into an equation for the generating function for the sequence appearing in the $m-$th column of the $B(n, m)$ triangle

$$G_B(m; x) := \sum_{n=m}^{\infty} \ B(n, m) \ x^n \ \ , \tag{36}$$

with input $G_B(0; x) \ = \ \sum_{n=0}^{\infty} \binom{2n}{n} \ x^n \ = \ 1/\sqrt{1-4x}$, the generating function for the central binomial numbers. So (26) implies for $m \in \mathbf{N}_0$,

$$G_B(m; x) \ = \ \Big(\frac{x}{1-4x}\Big)^m \ \frac{1}{\sqrt{1-4x}}. \tag{37}$$

For $x\frac{d}{dx}G_B(m; x)$ see (53). Therefore, we have proved:

Proposition 4 (Column sequences of the $B(n, m)$ triangle)

The sequence $\{B(n, m)\}_{n=m}^{\infty}$, *defined, for fixed* $m \in \mathbf{N}_0$ *and* $n \in \mathbf{N}_0$ by (28) *is the convolution of the central binomial sequence* $\{\binom{2k}{k}\}_{k \in \mathbf{N}_0}$ *and the* $m-$*th convolution of the (shifted) power sequence*$\{0, 1, 4^1, 4^2, ...\}$.

Note 1: The infinite dimensional matrix $\mathbf{B}$ with elements $B(n,m)$ given for $n \geq m \geq 0$ by (28) and $B(n,m) \equiv 0$ for $n < m$ is an example of a *Riordan* matrix [7]. With the notation of this reference $\mathbf{B} = (\frac{1}{\sqrt{1-4x}}, \frac{x}{1-4x})$.

Note 2: *Sheffer*-type identities from *Riordan*-matrices

Triangular *Riordan*-matrices $\mathbf{M} = (M_{i,j})_{i \geq j \geq 0} = (g(x), f(x))$, $M_{i,j} = 0$ for $j > i$, in the notation of [7], lead to polynomials which satisfy *Sheffer*-type identities (see [5] and its references, and [1])

$$S_n(x+y) \;=\; \sum_{k=0}^{n} \; S_k(y)\, P_{n-k}(x) \quad = \sum_{k=0}^{n} \; P_k(y)\, S_{n-k}(x) \;, \tag{38}$$

$$P_n(x+y) \;=\; \sum_{k=0}^{n} \; P_k(y)\, P_{n-k}(x) \quad = \sum_{k=0}^{n} \; P_k(x)\, P_{n-k}(y) \;, \tag{39}$$

where the polynomials $S_n(x)$ and $P_n(x)$ are defined by

$$S_n(x) \;=\; \sum_{m=0}^{n} \; M_{n,m}\, \frac{x^m}{m!},\; n \in \mathbf{N}_0 \quad , \quad P_n(x) \;=\; \sum_{m=1}^{n} \; P_{n,m}\, \frac{x^m}{m!},\; n \in \mathbf{N},\; P_0(x) \equiv 1 \;, \tag{40}$$

with $P_{n,m} := [z^n]\,(f^m(z))$, $n \geq m \geq 1$. Here $g(x)$ defines the first column of $\mathbf{M}$: $M_{n,0} = [x^n]\, g(x)$.

If one uses $s_n(x) := n!\, S_n(x)$ and $p_n(x) := n!\, P_n(x)$ one obtains the *Sheffer*-identities (also called binomial identities) treated in [5]. Then $s_n(x)$ is *Sheffer* for $(1/g(\bar{f}(t)), \bar{f}(t))$, and $p_n(x)$ is associated to $\bar{f}(t)$ (or *Sheffer* for $(1, \bar{f}(t))$) in the terminology of [5]. Here $\bar{f}(t)$ stands for the inverse of $f(t)$ as a function.

Let us give the relation between $g_b(x;z)$ and $G_B(m;x)$.

Proposition 5: *We have*

$$g_b(x;z) \;=\; \sum_{m=0}^{\infty} (-1)^m \; G_B(m;xz) \; \Big(\frac{1}{x}\Big)^m \quad . \tag{41}$$

Proof: One inserts the value of $b_n(x)$ given in (20) into the definition (6) of $g_b(x;z)$ and rewrites the *Cauchy*-sum as two infinite sums which are then interchanged. Finally, the definition of $G_B(m;x)$ in (36) is used. □

One can check (41) by using the explicit form of $G_B(m;xz)$ given in (36) and comparing with (6).

In a similar vein we can solve $a_n(x)$ in (17) with $b_n(x)$ given by (20) and (28). The coefficients $a(n,k)$, defined by (19), have to satisfy, after comparing coefficients of x^n, x^0, and x^{n-k} for $k = 1, 2, ..., n-1$ and $n \in \mathbf{N}_0$:

$$x^n \;:\quad a(n,0) \;=\; 4\; a(n-1,0) \;+\; C_n, \tag{42}$$

$$x^0 \;:\quad (n+1)\; a(n,n) \;=\; 1 \;+\; n\; a(n-1,n-1) \;, \tag{43}$$

$$x^{n-k} \;:\quad (n+1)\; a(n,k) \;=\; k\; a(n-1,k-1) \;+\; 4(n+1+k)\; a(n-1,k) \;+\; B(n,k) \;. \tag{44}$$

In (42) we have used (24), *i.e.*, $B(n,0) = (n+1)\; C_n$, and in 43 we have used (25), *i.e.*, $B(n,n) \equiv 1$. From (42) one finds with input $a(0,0) = 1$

$$a(n,0) \;=\; \sum_{k=0}^{n} \; C_k \; 4^{n-k} \;, \tag{45}$$

and from (43)

$$a(n,n) \equiv 1, \text{ or } a_n(0) = (-1)^n . \tag{46}$$

Note that $a(n,0) = s_n(4)$ of (31) with solution (35). It is convenient to define $a(n-1,-1) := C_n$, $n \in \mathbf{N}_0$. Then the sequence $\{a(n,0)\}_{-1}^{\infty}$ is, with $a(-1,0) := 0$, the convolution of the sequence $\{a(k,-1)\}_{-1}^{\infty}$ and the shifted power sequence $\{0, 1, 4^1, 4^2, ...\}$. Before solving (44), with $B(n,k)$ from (28) inserted, we add to the trianglular array of numbers $a(n,m)$ the $m = -1$ column and an extra row for $n = -1$, and define a new enlarged triangular array for $n, m \in \mathbf{N}_0$ as

$$A(n,m) := a(n-1, m-1) \tag{47}$$

with $A(n,0) = a(n-1,-1) = C_n$ and $A(0,m) = a(-1,m-1) = \delta_{0,m}$. An inspection of the $A(n,m)$ triangular array, partly depicted in *TAB. 2*, leads to the conjecture

$$A(n,m) = 4\, A(n-1,m) + A(n-1,m-1) , \tag{48}$$

with $A(n,0) = C_n$ and $A(n,m) \equiv 0$ for $n < m$. This recursion relation can be used to extend the array $A(n,m)$ to negative integer values of m. This conjecture is correct for $A(n+1,1) = a(n,0)$ found in (45), as well as for $A(n+1,n+1) = a(n,n) \equiv 1$ known from (46). The generating function for the sequence appearing in the $m-$th column,

$$G_A(m;x) := \sum_{n=m}^{\infty} A(n,m)\, x^n , \tag{49}$$

due to (48) satisfies $G_A(m;x) = \frac{x}{1-4x}\, G_A(m-1;x)$, remembering that $A(m-1,m) \equiv 0$, and that $G_A(0;x) = c(x)$. Therefore

$$G_A(m;x) = \Big(\frac{x}{1-4x}\Big)^m c(x) . \tag{50}$$

Note 3: The infinite dimensional matrix $\mathbf{A}$ with elements $A(n,m)$ given for $n \geq m \geq 0$ by (48) and $A(n,m) \equiv 0$ for $n < m$ is another example of a *Riordan* matrix, written in the notation of [7] as $(c(x), x/(1-4x))$.

Because of (37) and $\sqrt{1-4x}\; c(x) = 2 - c(x)$, these generating functions of the conjectured $A(n,m)$ column sequences obey

$$G_A(m;x) = (2 - c(x))\, G_B(m;x) . \tag{51}$$

If we use the conjecture (48) in (44) which is written with (47) in the form

$$(n+1)\, A(n+1,m+1) = \&m\, A(n,m) + 4(n+m+1)\, A(n,m+1) + B(n,m) ,$$

for $n \in \mathbf{N}_0$, $m \in \{1, 2, .., n-1\}$, we have

$$m\, A(n+1,m+1) - (n+1)\, A(n,m) + B(n,m) = 0 . \tag{52}$$

This recursion relation can be written with the help of the generating functions (36) and (49) as

$$\Big(x\frac{d}{dx} + 1\Big)\, G_A(m;x) - \frac{m}{x}\, G_A(m+1;x) = G_B(m;x) , \tag{53}$$

or with (50) (*i.e.* the conjecture) as

$$\Big(x\frac{d}{dx} + 1 - \frac{m}{1-4x}\Big)\, G_A(m;x) = G_B(m;x) . \tag{54}$$

Together with (51) this means

$$x\frac{d}{dx}\Big((2-c(x))\ G_B(m;x)\Big) \ = \ \Big[\Big(\frac{m}{1-4x}-1\Big)(2-c(x)) \ +1\Big] \ G_B(m;x)\ . \tag{55}$$

If we can prove this equation with $G_B(x)$ given by (37) we have shown that (44) is equivalent to the conjecture (48). In order to prove (55) we first compute from (37), for $m \in \mathbf{N}_0$,

$$x\frac{d}{dx}G_B(m;x) \ = (2+\frac{m}{x})\ G_B(m+1;x) \ = \ \frac{2x+m}{1-4x}\ G_B(m;x)\ . \tag{56}$$

With this result, (55) reduces to

$$\Big(-x\ c'(x) \ + \ (2-c(x))\ \frac{1-2x}{1-4x} \ - \ 1\Big)\ G_B(m;x) \ = \ 0\ , \tag{57}$$

and with (1) the factor in front of $G_B(m;x)$ vanishes identically for $x \neq 1/4$. Therefore, we have proved the following two propositions concerning the column sequences of the $A(n,m)$ triangular array and the triangular $A(n,m)$ array respectively.

Proposition 6: *The triangular array of numbers* $A(n,m)$, *defined for* $n,m \in \mathbf{N}_0$ *by equation* (48), $A(n,0) = C_n$, $A(n,m) \equiv 0$ *for* $n < m$ *has as* $m-$*th column sequence* $\{A(n,m)\}_{n=m}^{\infty}$ *the convolution of the* Catalan *sequence and the* $m-$*th convolution of the shifted power sequence* $\{0,1,4^1,4^2,...\}$.

Proof: Use (50) with (49). □

Proposition 7: *The triangular array* $A(n,m)$ *of* Proposition 6 *coincides with the one defined by* (47) *and* (42), (43) *and* (44) *with* $B(n,m)$ *given by* (28).

Proof: On the one hand $a(n,0) = A(n+1,1)$ and $a(n,n) = A(n+1,n+1) \equiv 1$ of (42) and (43), *i.e.*, (45) and (46), respectively, satisfy (45). On the other hand (44) is rewritten with the aid of (47) as (52), and (52) has been proved by (53) to (57). □

Alternatively, one can use the now proven conjecture (48), together with (47), in (44) and derive for $n \in \mathbf{N}_0$, $m \in \mathbf{N}_0$,

$$4m\ a(n-1,m) \ = \ (n+1-m)\ a(n-1,m-1) \ - \ B(n,m)\ . \tag{58}$$

This is written in terms of the polynomials $a_{n-1}(x)$ of (19) and $b_n(x)$ of (20) as

$$x(1-4x)\ a'_{n-1}(x) \ + \ (1-4x+4nx)\ a_{n-1}(x) \ - \ \binom{2n}{n}\ x^n \ + \ b_n(x) \ = \ 0\ . \tag{59}$$

With this result (17) becomes an inhomogeneous recursion relation for $a_n(x)$, *viz.*

$$a_n(x) \ = \ (4x-1)\ a_{n-1}(x) \ + \ C_n\ x^n \quad , \quad a_0(x) \equiv 1\ . \tag{60}$$

Moreover, (59) can also be considered as an inhomogeneous linear differential equation for $a_{n-1}(x)$ with given $b_n(x)$. To find the solution this way is, however, a bit tedious. Let us give an alternative form for $a_n(x)$:

Proposition 8: *The solution of the recursion relation* (60) *is given by* (8).

Proof: Iteration of (60). □

Next we give a

Corollary 2: *The generating function* $g_a(x;z) := \sum_{n=0}^{\infty} a_n(x)\, z^n$ *is given by* (9).

Proof: Equation (8) shows that $a_n(x)$ is a convolution of the functional sequences $\{C_k\ x^k\}_{n\in\mathbf{N}_0}$ and $\{(4x-1)^k\}_{k\in\mathbf{N}_0}$ with generating functions $c(xz)$ and $1/(1+(1-4x)z)$. Therefore, $g_a(x;z)$ is the product of these generating functions. □

We now have a relation between $g_a(x;z)$ and $G_A(m;x)$:

Proposition 9:

$$g_a(x;z) \;=\; \frac{1}{1-4xz}\ \sum_{m=0}^{\infty}(-1)^m\ G_A(m;xz)\ \left(\frac{1}{x}\right)^m\ . \tag{61}$$

Proof: Analogous to the proof of *Proposition 5.* □

One can ckeck (61) by putting in the explicit form (50) of $G_A(m;x)$ and compare with (9). Let us state the relation between $b_n(x)$ and $a_{n-1}(x)$ as

Proposition 10: *For* $n\in\mathbf{N}_0$ *and* $a_{-1}(x)\equiv 0$, *the relation between* $b_n(x)$ *and* $a_{n-1}(x)$ *is given by* (10) .

Proof: The alternative expressions (5) and (8) for these two families of polynomials are used. One splits off the $k=0$ term in (5) with $C_{-1}=-1/2$ from the sum and shifts the summation variable. □

Corollary 3: The coefficients of the triangular arrays $A(n,m)$ and $B(n,m)$ are related as given by (11).

Proof: The relation (10) between the polynomials is, with the help of (19) and (20), written for the coefficients $a(n-1,m)$, or by (47) for $A(n,m+1)$, and $B(n,m)$. □

It remains to compute the explicit expression for the coefficients $a(n,k)$ of $a_n(x)$ defined by (19). Because of (47) it suffices to determine $A(n,m)$.

Corollary 4: *The triangular array numbers* $A(n,m)$ *are given explicitely by formula* (7).

Proof: The formula (4) written for $B(n,m-1)$ is used in relation (11). □

Note 4: This formula for $A(n,m)$ satisfies indeed the recursion relation (48) with the given input. The first term, $\frac{1}{2}\ 4^{n-m+1}\ \binom{n}{m-1}$, satisfies it because of the binomial identity $\binom{n}{m-1} = \binom{n-1}{m-1} + \binom{n-1}{m-2}$. For the second term of $A(n,m)$ in (7) one has to prove

$$\binom{n}{m-1}\binom{2n}{n} \;=\; 4\binom{n-1}{m-1}\binom{2(n-1)}{n-1} + \binom{n-1}{m-2}\binom{2(n-1)}{n-1}\frac{2(2m-3)}{m-1}\ ,$$

or after division by $\binom{2(n-1)}{n-1}$,

$$\frac{2n-1}{n}\binom{n}{m-1} \;=\; 2\binom{n-1}{m-1} + \binom{n-1}{m-2}\frac{2m-3}{m-1}\ ,$$

which reduces to the trivial identity $2n-1 = 2\,(n-m+1)+2m-3$. Both terms together, *i.e.*, (7), satisfy the input $A(n,n) \equiv 1$.

Note 5: $A(n,m)$ was found originally after iteration in the form (with $n \geq m > 0$ and $(-1)!! := 1$)

$$A(n,m) \;=\; 2\cdot 4^{n-m}\binom{n}{m-1} \;-\; \frac{\prod_{k=1}^{m}(2(n-m)+2k-1)}{(2m-3)!!}\,C_{n-m}\;. \tag{62}$$

$A(n,0) = C_n$. It is easy to establish the equivalence with (7).

In the original derivation of the formula (7) for $A(n,m)$ it turned out to be convenient to introduce a rectangular array of integers $\hat{A}(n,m)$ for $n,m \in \mathbf{N}_0$ as follows: $\hat{A}(0,m) \equiv 1$, $\hat{A}(n,0) := -C_n$ for $n \in \mathbf{N}$, and for $m \in \mathbf{N}$ and $n \in \mathbf{N}_0$, $\hat{A}(n,m)$ is defined by (7), or equivalently, by (8). The $A(n,m)$ recursion (48) translates (with the help of the *Pascal*-triangle identity) into

$$\hat{A}(n,m) \;=\; 4\;\hat{A}(n-1,m) \;+\; \hat{A}(n,m-1)\;. \tag{63}$$

This leads, after iteration and use of $\hat{A}(0,m) \equiv 1$ from (12) with $A(n,n) \equiv 1$, to

$$\hat{A}(n,m) \;=\; 4^n\;\sum_{k=0}^{n}\;\hat{A}(k,m-1)/4^k\;. \tag{64}$$

Thus, the following proposition describes column sequences of the $\hat{A}(n,m) \equiv C4(n,m)$ array.

Proposition 11: *The $m-$th column sequence of the $\hat{A}(n,m)$ array, $\{\hat{A}(n,m)\}_{n\in\mathbf{N}_0}$, is the convolution of the sequence $\{\hat{A}(n,0)\}_{n\in\mathbf{N}_0} = \{1,-1,-2,-5,...\}$, generated by $2-c(x)$, and the $m-$th convolution of the power sequence $\{4^k\}_{k\in\mathbf{N}_0}$.*

Proof: Iteration of (64) with the $\hat{A}(n,0)$ input. □

Corollary 5: *The ordinary generating function of the $m-$th column sequence of the $\hat{A}(n,m)$* array (13) *is for $m \in \mathbf{N}_0$ given by*

$$G_{\hat{A}}(m;x) \;:=\; \sum_{n=0}^{\infty}\;\hat{A}(n,m)\;x^n \;=\; (2-c(x))\;\Big(\frac{1}{1-4x}\Big)^m\;. \tag{65}$$

Proof: Use *Proposition 11* written for generating functions. □

Because of the convolution of the (negative) *Catalan* sequence with powers of 4 we shall call this $\hat{A}(n,m)$ array also $C4(n,m)$. A part of it is shown in *TAB. 3.* The second column sequence is given by $\hat{A}(n,1) \equiv C4(n,1) = \binom{2n+1}{n}$ and appears as nr.2848 in the book [8], or as $A001700$ in the web encyclopedia [8]. The sequence of the third column $\{\hat{A}(n,2) \equiv C4(n,2)\}_{n\in\mathbf{N}_0} = \{1,7,38,187,...\}$ is from (64) and (62) with (12) determined by $4^n\;\sum_{k=0}^{n}\binom{2k+1}{k}/4^k \;=\; (2n+3)\;(2n+1)\;C_n \;-\; 2^{2n+1}$, and is listed as $A000531$ in [8]. There the fourth column sequence is now listed as $A029887$.

Note 6: The infinite dimensional lower triangular matrix $\tilde{\mathbf{A}}$ related to the array $\hat{A}(n,m) \equiv C4(n,m)$ by $\tilde{A}(n,m) := \hat{A}(n-m,m+1)$ for $n \geq m \geq 0$ and $\tilde{A}(n,m) := 0$ for $n < m$ is again an example of a *Riordan* matrix [7]. In the notation of [7], $\tilde{\mathbf{A}} \;=\; (c(x)/\sqrt{1-4x}, x/\sqrt{1-4x})$.

Finally, we derive identities by using, for $n \in \mathbf{N}_0$, equation (14) for the left hand side of (3) and the results for $a_{n-1}(x)$ and $b_n(x)$ for the right hand side.

Because there are no negative powers of x on the left hand side of (3), such powers have to vanish on the right hand side. This leads to the first family of identities. Because $(1-4x)^{-n} = \sum_{k=0}^{\infty} \frac{(n)_k}{k!}\, 4^k\, x^k$, with *Pochhammer*'s symbol defined after (28), this means that $[x^p]\ (a_{n-1}(x)+b_n(x)\ c(x))$, the coefficient proportional to x^p, has to vanish for $p = 0, 1, ..., n-1,\ n \in \mathbf{N}$. This requirement reads

$$(-1)^{n-1-p}\ a(n-1, n-1-p)\ +\ \sum_{k=0}^{p}(-1)^{n-k}\ B(n, n-k)\ C_{p-k}\ \equiv 0\ . \tag{66}$$

The sum is restricted to $k \leq p\ (< n)$ because no number C_l with negative index is found in $c(x)$. Inserting the known coefficients, this produces (15).

Proposition 12: *For $n \in \mathbf{N}$ and $p \in \{0, 1, ..., n-1\}$ identity $(D1)$, given by (15), holds.*

Proof: With (47), (66) becomes

$$\sum_{k=0}^{p}\ (-1)^{p-k}\ C_{p-k}\ B(n, n-k)\ =\ A(n, n-p)\ , \tag{67}$$

which is $(D1)$ of (15) if the summation index k is changed into $p-k$, and the symmetry of the binomial coefficients is used. □.

Example 4: Take $p = n-1 \in \mathbf{N}_0$:

$$\sum_{k=0}^{n-1}(-1)^k \binom{n}{k+1} \frac{1}{2k+1}\ =\ 4^n \Big/ \binom{2n}{n}\ -\ 1\ =\ 2A(n,1) \Big/ \binom{2n}{n}\quad . \tag{68}$$

With this identity we have found a sum representation for the convolution of the *Catalan* sequence and powers of 4:

$$s_{n-1}(4)\quad :=\quad 4^{n-1}\ \sum_{k=0}^{n-1}\ C_k/4^k\ =\ \frac{1}{2}\binom{2n}{n}\ \sum_{k=0}^{n-1}(-1)^k \binom{n}{k+1}\ \frac{1}{2k+1}$$

(*cf.* (35) with (31)).

The second family of identities, $(D2)$ of (16), results from comparing powers x^k with $k \in \mathbf{N}_0$ on both sides of (3) after expansion of $(1-4x)^{-n}$ as given above in the text before (66). Only the second term $b_n(x)\ c(x)$ contributes because $a_{n-1}(x)/x^n$ has only negative powers of x. Thus, with definition (14), one finds for $k \in \mathbf{N}_0$ and $n \in \mathbf{N}$,

$$C(n,k)\ =\ \sum_{l=0}^{k}\ \frac{(n)_l\ 4^l}{l!}\ \sum_{j=0}^{n}(-1)^{n-j}\, B(n, n-j)\, C_{n-j+k-l} \tag{69}$$

which is, after interchange of the summations and insertion of $B(n, n-j)$ from (4) the desired identity $(D2)$ if also the summation index j is changed to $n-q$.

Thus we have shown:

Proposition 13: *For $k \in \mathbf{N}_0$ and $n \in \mathbf{N}$ identity $(D2)$ of (16) with $C(n,k)$ defined by (14) holds true.*

Example 5: Take $k = 0,\ n \in \mathbf{N}$. So we have

$$\sum_{j=0}^{n}(-1)^j \binom{n+1}{j+1}\ \equiv 1\ , \tag{70}$$

which is elementary.

Acknowledgements

The author thanks the referees of this and of [3] for remarks and some references, namely [7] and [1].

TAB. 1: B(n,m) Central Binomial Triangle

n \ m	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0	0	0	0
2	6	6	1	0	0	0	0	0	0	0	0
3	20	30	10	1	0	0	0	0	0	0	0
4	70	140	70	14	1	0	0	0	0	0	0
5	252	630	420	126	18	1	0	0	0	0	0
6	924	2772	2310	924	198	22	1	0	0	0	0
7	3432	12012	12012	6006	1716	286	26	1	0	0	0
8	12870	51480	60060	36036	12870	2860	390	30	1	0	0
9	48620	218790	291720	204204	87516	24310	4420	510	34	1	0
10	184756	923780	1385670	1108536	554268	184756	41990	6460	646	38	1

TAB. 2: A(n,m) Catalan Triangle

n \ m	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0
2	2	5	1	0	0	0	0	0	0	0	0
3	5	22	9	1	0	0	0	0	0	0	0
4	14	93	58	13	1	0	0	0	0	0	0
5	42	386	325	110	17	1	0	0	0	0	0
6	132	1586	1686	765	178	21	1	0	0	0	0
7	429	6476	8330	4746	1477	262	25	1	0	0	0
8	1430	26333	39796	27314	10654	2525	362	29	1	0	0
9	4862	106762	185517	149052	69930	20754	3973	478	33	1	0
10	16796	431910	848830	781725	428772	152946	36646	5885	610	37	1

TAB. 3: C4(n,m) Catalan array

n \ m	0	1	2	3	4	5	6
0	1	1	1	1	1	1	1
1	-1	3	7	11	15	19	23
2	-2	10	38	82	142	218	310
3	-5	35	187	515	1083	1955	3195
4	-14	126	874	2934	7266	15086	27866
5	-42	462	3958	15694	44758	105102	216566
6	-132	1716	17548	80324	259356	679764	1546028
7	-429	6435	76627	397923	1435347	4154403	10338515
8	-1430	24310	330818	1922510	7663898	24281510	65635570
9	-4862	92378	1415650	9105690	39761282	136887322	399429602
10	-16796	352716	6015316	42438076	201483204	749032492	2346750900

References

[1] M. Barnabei, A. Brini, and G. Nicoletti: "Recursive Matrices and Umbral Calculus", J. Algebra 75 (1982), 546-573

[2] R.L. Graham, D.E. Knuth, and O. Patashnik: *"Concrete Mathematics"*, Addison-Wesley, Reading MA, 1989

[3] W. Lang: "On Polynomials Related to Powers of the Generating Function of Catalan Numbers", Karlsruhe preprint 1999, tbp The Fibonacci Quarterly

[4] M. Petkovšek, H.S. Wilf, and D. Zeilberger: *"A=B"*, AK Peters, Wellesley, MA, 1996

[5] S. Roman: *"The Umbral Calculus"*, Academic Press, New York, 1984

[6] L.W. Shapiro: "A Catalan Triangle", Discrete Mathematics 14 (1976), 83-90

[7] L. W. Shapiro, S. Getu, W.-J. Woan and L. C. Woodson:"The Riordan Group", Discrete Appl. Math. 34 (1991), 229-239

[8] N.J.A. Sloane and S. Plouffe: *" The Encyclopedia of Integer Sequences"*, Academic Press, San Diego, 1995; see also N.J.A. Sloane's On-Line Encyclopedia of Integer Sequences, http//:www.research.att.com/~njas/sequences/index.html

[9] W.-J. Woan, L. Shapiro, and D.G. Rogers: "The Catalan Numbers, the Lebesgue Integral, and 4^{n-2}", Am. Math. Monthly 101 (1997), 926-931

AMS MSC numbers: 11B83, 11B37, 33C45

On Polynomials Related to Powers of the Generating Function of Catalan's Numbers

Wolfdieter L a n g [1]

Institut für Theoretische Physik
Universität Karlsruhe
Kaiserstrasse 12, D-76128 Karlsruhe, Germany

1 Introduction and Summary

Catalan's sequence of numbers $\{C_n\}_0^\infty = \{1, 1, 2, 5, 14, 42, ...\}$ (nr.1459 and $A000108$ of [14]) emerges in the solution of many combinatorial problems (see [2],[4],[5],[16] (also for further references)). The moments $\mu_{2\,k}$ of the normalized weight function of *Chebyshev*'s polynomials of the second kind are given by $C_k/2^k$ (see e.g. [3] Lemma 4.3, p. 160 for $l = 0$, [17], p.II-3). This sequence also shows up in the asymptotic moments of zeros of scaled *Laguerre* and *Hermite* polynomials [9] eqs.(3.34) and (3.35). The generating function $c(x) = \sum_{n=0}^\infty C_n\ x^n$ is the solution of the quadratic equation $x\ c^2(x) - c(x) + 1 = 0$ with $c(0) = 1$. Therefore, every positive integer power of $c(x)$ can be written as

$$c^n(x) = p_{n-1}(x)1 + q_{n-1}(x)\ c(x) \ , \tag{1}$$

with certain polynomials p_{n-1} and q_{n-1}, both of degree $(n-1)$, in $1/x$. In *section* 2 they are shown to be related to *Chebyshev*'s polynomials of the second kind:

$$p_{n-1}(x) = -(\frac{1}{\sqrt{x}})^n\ S_{n-2}(\frac{1}{\sqrt{x}}) \ , \quad q_{n-1}(x) = (\frac{1}{\sqrt{x}})^{n-1}\ S_{n-1}(\frac{1}{\sqrt{x}}) = -x\ p_n(x) \ , \tag{2}$$

with $S_n(y) = U_n(y/2)$. It is therefore possible to extend the range of the power n to negative integers (or to real or complex numbers). Tables for the $U_n(x)$ polynomials can be found, *e.g.*, in [1]. Because powers of a generating function correspond to convolutions of the generated number sequence the given decomposition of $c^n(x)$ will determine convolutions of the *Catalan* sequence. In passing, an explicit expression for general convolutions in the form of nested sums will also be given. Contact with the works of refs. [6],[12],[18], [5] will be made.

Together with the known (*e.g.* [4],[11]) result (valid for real n)

$$c^n(x) = \sum_{k=0}^\infty C_k(n)\ x^k \ \text{, with } C_k(n) = \frac{n}{n+2k}\binom{n+2k}{k} = \frac{n}{k+n}\binom{n-1+2k}{k} \ , \tag{3}$$

[1] E-mail: wolfdieter.lang@physik.uni-karlsruhe.de, http://www-itp.physik.uni-karlsruhe.de/~wl

one finds from the alternative expression (1) for positive n two sets of identities:

$$(P1) \qquad \sum_{l=0}^{p}(-1)^l \binom{n+1-p+l}{p-l}\, C_l \;=\; \binom{n-p}{p}\,, \tag{4}$$

for $n \in \mathbf{N}_0,\ p \in \{0,1,2,...\lfloor \frac{n}{2} \rfloor\}$, and

$$(P2) \qquad \sum_{l=0}^{\lfloor \frac{n-1}{2} \rfloor}(-1)^l \binom{n-1-l}{l}\, C_{k+n-1-l} \;=\; C_k(n)\,, \tag{5}$$

for $n \in \mathbf{N}$, $k \in \mathbf{N}_0$.

For negative powers in (1) two other sets of identities result:

$$(P3) \qquad \sum_{l=0}^{min(\lfloor \frac{n-1}{2} \rfloor, k-1)}(-1)^l \binom{n-1-l}{l}\, C_{k-1-l} \;=\; (-1)^{k+1} \binom{n-k-1}{k-1}\,, \tag{6}$$

for $n \in \mathbf{N},\ k \in \{0,1,2,...\lfloor \frac{n}{2} \rfloor\}$, (for $k=0$ both sides are by definition zero) and

$$(P4) \qquad \sum_{l=0}^{\lfloor \frac{n-1}{2} \rfloor}(-1)^l \binom{n-1-l}{l}\, C_{k-1-l} \;=\; -C_k(-n) \;=\; \frac{n}{k}\binom{2k-n-1}{k-1}, \tag{7}$$

for $n \in \mathbf{N}$, $k \in \mathbf{N}$ with $k \geq \lfloor \frac{n}{2} \rfloor + 1$. These identities can be continued for appropriate values of real n.

Another expression for the coefficients of negative powers of $c(x)$ is

$$C_k(-n) \;=\; \sum_{l=1}^{min(n,k)}(-1)^l \binom{n}{l}\, C_{k-l}(n)\,, \tag{8}$$

for $n,k \in \mathbf{N}$, and $C_0(-n) = 1$, $C_n(0) = \delta_{n,0}$. Also, from (3) $C_k(-n) = -C_{k-n}(n)$ for $n,k \in \mathbf{N}$ with $k \geq n$.

The remainder of this paper provides proofs for the above given statements. *Section 2* deals with integer (and real) powers of the generating function $c(x)$. Convolutions of general sequences are expressed there in terms of nested sums. In *Section 3* some families of integer sequences related to the polynomials $q_n(x)$ (2) evaluated for $x = 1/m$ for $m = 4,5,...$ and $(-1)^n\ q_n(x)$ evaluated at $x = -1/m$, $m \in \mathbf{N}$ are considered.

2 Powers

The equation $x\ c^2(x) - c(x) + 1 = 0$ whose solution defines the generating function of *Catalan*'s numbers if $c(0) = 1$ can be considered as characteristic equation for the recursion relation

$$x\ r_{n+1} \;-\; r_n \;+\; r_{n-1} \;=\; 0 \quad , \quad n = 0,1,... \quad , \tag{9}$$

with arbitrary inputs $r_{-1}(x)$ and $r_0(x)$. A basis of two linearly independent solutions is given by the *Lucas*-type polynomials $\{\mathcal{U}_n\}$ and $\{\mathcal{V}_n\}$, with standard inputs $\mathcal{U}_{-1} = 0$, $\mathcal{U}_0 = 1$, $(\mathcal{U}_{-2} = -x)$, and $\mathcal{V}_{-1} = 1$, $\mathcal{V}_0 = 2$, $(\mathcal{V}_1 = 1/x)$, in the *Binet* form

$$\mathcal{U}_{n-1}(x) \;=\; \frac{c_+^n(x) \;-\; c_-^n(x)}{c_+(x) \;-\; c_-(x)} \;, \tag{10}$$

$$\mathcal{V}_n(x) \;=\; c_+^n(x) \;+\; c_-^n(x) \;=\; \frac{1}{x}(\mathcal{U}_{n-1}(x) \;-\; 2\,\mathcal{U}_{n-2}(x)) \;, \tag{11}$$

with the two solutions of the characteristic equation, *viz* $c_\pm(x) := (1 \pm \sqrt{1-4x}\;)/(2x)$. $c(x) := c_-(x)$ satisfies $c(0) = 1$, and $c_+(x) = 1/(xc(x))$, as well as $c_+(x) + c(x) \;=\; 1/x$. From the recurrence (9) it is clear that for positive $n \neq 0$ $\mathcal{U}_n$ is a polynomial in $1/x$ of degree $n-1$. If $c_+(x) - c_-(x) = 0$, *i.e.* $x = 1/4$, eq.(10) is replaced by $\mathcal{U}_n(1/4) = 2^n(n+1)$. The second eq. in (11) holds because both sides of the eq. satisfy recurrence (9) and the inputs for $\mathcal{V}_0$ and $\mathcal{V}_1$ match. One may associate with the recurrence relation (9) a transfer matrix

$$\mathbf{T}(x) \;=\; \begin{pmatrix} 1/x & -1/x \\ 1 & 0 \end{pmatrix} \quad , \quad Det\;\mathbf{T}(x) \;=\; 1/x \;. \tag{12}$$

With this matrix one can rewrite (9) as

$$\begin{pmatrix} r_n \\ r_{n-1} \end{pmatrix} \;=\; \mathbf{T}(x) \begin{pmatrix} r_{n-1} \\ r_{n-2} \end{pmatrix} \;=\; \mathbf{T}^n(x) \begin{pmatrix} r_0(x) \\ r_{-1}(x) \end{pmatrix} \tag{13}$$

Because $\mathbf{T}^n \;=\; \mathbf{T}\;\mathbf{T}^{n-1}$ with input $\mathbf{T}^1 = \mathbf{T}(\mathbf{x})$ given by (12), one finds from the recurrence relation (9) with $r_n = \mathcal{U}_n$

$$\mathbf{T}^n(x) \;=\; \begin{pmatrix} \mathcal{U}_n(x) & -\frac{1}{x}\,\mathcal{U}_{n-1}(x) \\ \\ \mathcal{U}_{n-1}(x) & -\frac{1}{x}\,\mathcal{U}_{n-2}(x) \end{pmatrix} \quad . \tag{14}$$

Note that for $x = 1$ one has $c_\pm(1) = (1 \pm i\sqrt{3})/2$, which are 6th roots of unity, and the related period 6 sequences are $\{\mathcal{U}_n(1)\}_{-1}^{\infty} \;=\; \{\overline{0,1,1,0,-1,-1,}\}$, as well as $\{\mathcal{V}_n(1)\}_0^{\infty} \;=\; \{\overline{2,1,-1,-2,-1,1,}\}$. This follows from eqs. (10) and (11). It is convenient to map the recursion relation (9) to the familiar one for *Chebyshev*'s $S_n(x) = U_n(x/2)$ polynomials of the second kind, *viz*

$$S_n(x) \;=\; x\;S_{n-1}(x) \;-\; S_{n-2}(x) \quad , \quad S_{-1} = 0,\; S_0 = 1 \;, \tag{15}$$

with characteristic equation $\lambda^2 - x\lambda + 1 = 0$ and solutions $\lambda_\pm(x) = \frac{x}{2}(1 \pm \sqrt{1-(2/x)^2}\;)$, satisfying $\lambda_+(x)\;\lambda_-(x) = 1$ and $\lambda_+(x) + \lambda_-(x) = x$. The relation to $c_\pm(x)$ is

$$\sqrt{x}\;c_\pm(x) \;=\; \lambda_\pm(1/\sqrt{x}) \;. \tag{16}$$

The *Binet* form of the corresponding two independent polynomial systems is

$$S_{n-1}(x) \;=\; \frac{\lambda_+^n(x) \;-\; \lambda_-^n(x)}{\lambda_+(x) \;-\; \lambda_-(x)} \;, \tag{17}$$

$$2\;T_n(x/2) \;=\; \lambda_+^n(x) \;+\; \lambda_-^n(x) \;, \tag{18}$$

and $T_n(x/2) \;=\; (S_n(x) - S_{n-2}(x))/2$ are *Chebyshev*'s polynomials of the first kind. Tables of *Chebyshev*'s polynomials can be found in [1]. The coefficient triangles of the $S_n(x)$, $U_n(x)$ and $T_n(x)$ polynomials can

also be viewed under the numbers A049310, A053117 and A053120, respectively, in the on-line data-base [14].

The extension to negative integer indices runs as follows

$$\mathcal{U}_{-n}(x) = -x^{n-1}\,\mathcal{U}_{n-2}(x)\ , \tag{19}$$

$$S_{-(n+2)}(x) = -S_n(x)\ . \tag{20}$$

This follows from (10) and (17). Note that from (9) $\mathcal{U}_n$ is for positive n a monic polynomial in $1/x$ of degree n, and for negative n in general a non-monic polynomial in x of degree $\lfloor -\frac{n}{2} \rfloor$. It is possible to extend the range of n to complex numbers using the *Binet* forms.

A connection between both systems of polynomials is made, after using (10), (16) and (17),by

$$\mathcal{U}_n(x) = \left(\frac{1}{\sqrt{x}}\right)^n\ S_n(1/\sqrt{x})\ . \tag{21}$$

This holds for $n \in \mathbf{Z}$, in accordance with (19) and (20).

After these preliminaries we are ready to state:

Proposition 1: The nth power of $c(x)$, the generating function of *Catalan*'s numbers, can, for $n \in \mathbf{Z}$, be written as

$$c^n(x) = -\frac{1}{x}\,\mathcal{U}_{n-2}(x) + \mathcal{U}_{n-1}(x)\,c(x)\ , \tag{22}$$

$$= -(\frac{1}{\sqrt{x}})^n\ S_{n-2}(1/\sqrt{x}) + (\frac{1}{\sqrt{x}})^{n-1}\ S_{n-1}(1/\sqrt{x})\ c(x). \tag{23}$$

Proof: Due to $c^2(x) = (c(x)-1)/x$ and $c^{-1}(x) = 1 - x\,c(x)$ one can, for $n \in \mathbf{Z}$, write $c^n(x) = p_{n-1}(x) + q_{n-1}(x)\,c(x)$. From $c^n(x) = c(x)\,c^{n-1}(x)$ one is led to $q_{n-1} = p_{n-2} + \frac{1}{x}\,q_{n-2}$ and $p_{n-1} = -\frac{1}{x}\,q_{n-2}$, or $q_{n-1} = (q_{n-2} - q_{n-3})/x$ with input $q_{-1} = 0,\ q_0 = 1$. Therefore, $q_{n-1}(x) = \mathcal{U}_{n-1}(x)$ and $p_{n-1}(x) = -\,\mathcal{U}_{n-2}(x)/x$. (23) then follows from (21). □

Note 1: Because $S_n(y) = \sum_{j=0}^{\lfloor n/2 \rfloor}(-1)^j \binom{n-j}{j} y^{n-2j}$ the explicit form of these polynomials (2) is $p_{n-1}(x) = \sum_{j=0}^{\lfloor n/2 \rfloor - 1}(-1)^{j+1}\binom{n-2-j}{j}x^{-(n-1-j)}$, $p_{-1} = 1,\ p_0 = 0$, and $q_{n-1}(x) = \sum_{j=0}^{\lfloor (n-1)/2 \rfloor}(-1)^j\binom{n-1-j}{j}x^{-(n-1-j)}$, $q_{-1} = 0$. For negative index one has, due to (20), $p_{-(n+1)}(x) = (\sqrt{x})^n\ S_n(1/\sqrt{x}) = \sum_{j=0}^{\lfloor n/2 \rfloor}(-1)^j\ \binom{n-j}{j}\ x^j$, and $q_{-(n+1)}(x) = -(\sqrt{x})^{n+1}\ S_{n-1}(1/\sqrt{x}) = -x\ \sum_{j=0}^{\lfloor (n-1)/2 \rfloor}(-1)^j\ \binom{n-1-j}{j}\ x^j$.

In the *Table* one can find the coefficient triangle for the polynomials $\{p_n(x)\}_{-1}^{12}$ with column m corresponding to $(\frac{1}{x})^m$, $m \geq 0$.

Note 2: An alternative proof of *proposition 1* can be given starting with eqs.(17) and (18) which show, together with $\lambda_+(x) - \lambda_-(x) = \sqrt{x^2-4}$, that

$$\lambda_\pm^n(x) = T_n(x/2) \pm \sqrt{(x/2)^2 - 1}\ S_{n-1}(x)\ , \tag{24}$$

or, from $\pm\sqrt{(x/2)^2-1} = \lambda_\pm(x) - x/2$ and the S_n recurrence relation (15)

$$\lambda_\pm^n(x) = T_n(x/2) - \tfrac{1}{2}\,(S_n(x) + S_{n-2}(x)) + S_{n-1}(x)\,\lambda_\pm(x) \tag{25}$$

$$= -S_{n-2}(x) + S_{n-1}(x)\,\lambda_\pm(x)\ . \tag{26}$$

Now (23) follows from (16). This also proves that one may replace in *proposition 1* $c(x)$ by $c_+(x) = 1/(xc(x))$ from which one recovers the c^{-n} formula for $n \in \mathbf{N}$ in accordance with (19) and (20).

Note 3: For the transfer matrix $\mathbf{T}(\mathbf{x})$, defined in (12), one can prove for $n \in \mathbf{N}$ in an analogous manner

$$\mathbf{T}^n \;=\; -(\frac{1}{\sqrt{x}})^n \; S_{n-2}(1/\sqrt{x}) \; \mathbf{1} \;+\; (\frac{1}{\sqrt{x}})^{n-1} \; S_{n-1}(1/\sqrt{x}) \; \mathbf{T}(x), \tag{27}$$

by employing the *Cayley-Hamilton* theorem for the 2×2 matrix $\mathbf{T}$ with $tr\ \mathbf{T} \;=\; \frac{1}{x} \;=\; det\ \mathbf{T}$ which states that $\mathbf{T}$ satisfies the characteristic equation $\mathbf{T}^2 \;-\; \frac{1}{x}\ \mathbf{T} \;+\; \frac{1}{x}\ \mathbf{1} \;=\; 0$.

Powers of a function which generates a sequence generate convolutions of this sequence. Therefore, *proposition 1* implies that convolutions of the *Catalan* sequence can be expressed in terms of *Catalan* numbers and binomial coefficients. Before giving this result we shall present an explicit formula for the nth convolution of a general sequence $\{C_n\}$ generated by $c(x) \;=\; \sum_{l=0}^{\infty}\ C_l\ x^l$. Usually the convolution coefficients $C_l(n)$, defined by $c^n(x) = \sum_{l=0}^{\infty}\ C_l(n)\ x^l$, are written as

$$C_l(n) \;=\; \sum_{\sum_{j=1}^{n} i_j \;=\; l} \; C_{i_1}\ C_{i_2} \cdots C_{i_n} \quad , \quad \text{with } i_j \in \mathbf{N}_0 \ . \tag{28}$$

An explicit formula with $(l-1)$ nested sums is the content of the next lemma.

Lemma 1: General convolutions

For $l = 2, 3, ...$

$$C_l(n) \;=\; C_0^{n-l}\ C_1^l \Big(\prod_{k=2}^{l} \sum_{i_k = a_k}^{\lfloor b_k \rfloor} \Big) \;< n, l, \{i_j\}_2^l > \; \prod_{j=2}^{l} \Big((\frac{C_j\ C_0}{C_1^j})^{i_j}\ \frac{1}{i_j!} \Big) \ , \tag{29}$$

with

$$b_2 = l/2 \quad , \quad b_k \;=\; (l - \sum_{j=2}^{k-1}\ j\ i_j)/k \ , \tag{30}$$

$$a_k = 0 \quad , \quad \text{for } k = 2, 3, ..., l-1; \quad a_l \;=\; max\Big(0, \lceil \frac{l - n - \sum_{j=2}^{l-1}(j-1)\ i_j)}{l-1} \rceil \Big) \tag{31}$$

$$< n, l, \{i_j\}_2^l > \;=\; \frac{n!}{(n - l + \sum_{j=2}^{l}(j-1)\ i_j)!(l - \sum_{j=2}^{l} j\ i_j)!}. \tag{32}$$

The first product in (29) is understood to be ordered such that the sums have indices $i_2, i_3, ...i_l$ when written from the left to the right. In addition: $C_0(n) = C_0^n$ and $C_1(n) = n\ C_0^{n-1}\ C_1$.

Proof: $C_l(n)$ of (28) is rewritten first as

$$C_l(n) \;=\; \sum\ (n, l, \{i_j\}_0^l)\ C_0^{i_0}\ C_1^{i_1} \cdots C_l^{i_l} \quad , \quad i_j \in \mathbf{N}_0 \quad , \tag{33}$$

where the sum is restricted by

$$(i): \qquad \sum_{j=0}^{l}\ j\ i_j \;=\; l \qquad \text{and} \qquad (ii): \qquad \sum_{j=0}^{l}\ i_j \;=\; n \ . \tag{34}$$

$(n, l, \{i_j\}_0^l)$ is a combinatorial factor to be determined later on. (*E.g.* for $n = 3, l = 5$ one has 4 terms in the sum: $i_5 = 1, i_0 = 2$; $i_4 = 1, i_1 = 1, i_0 = 1$; $i_3 = 1, i_2 = 1, i_0 = 1$; $i_3 = 1, i_2 = 2$,

with other indices vanishing, and the combinatorial factors are $3, 6, 6, 3$, respectively.) (ii) restricts the sum to terms with n factors, and (i) produces the correct weight l. These restrictions are solved by (i') : $\quad i_1 \;=\; l - \sum_{j=2}^{l} j\ i_j$ and (ii') : $\quad i_0 \;=\; n - i_1 - \sum_{j=2}^{l} i_j \;=\; n - l + \sum_{j=2}^{l}(j-1)\ i_j$. From $i_1 \geq 0$,*i.e.* $l - \sum_{j=2}^{l} j\ i_j \;\geq 0$, one infers $i_2 \leq \lfloor \frac{l}{2} \rfloor$, thus $i_2 \in [0, \lfloor \frac{l}{2} \rfloor\ ]$. For given i_2 in this range $i_3 \;\leq \lfloor \frac{l-2i_2}{3} \rfloor$, *etc.*, in general $0 \leq i_k \leq \lfloor (l - \sum_{j=2}^{k-1}\ j\ i_j)/k \rfloor$ for $k = 2, 3, ..., l$ with the sum replaced by zero for $k = 2$. This accounts for the upper boundaries $\lfloor b_k \rfloor$ in (30). Now, because $i_0 \geq 0$ (ii') implies a lower bound for i_l, the index of the last sum, *viz* $i_l \geq \lceil (l - n - \sum_{j=2}^{l-1}\ (j-1)\ i_j)/(l-1) \rceil$ with the ceiling function $\lceil \cdot \rceil$. In any case $i_l \geq 0$, therefore, the lower boundary for the i_l-sum is a_l as given in (31). All restrictions have then be solved and the lower boundaries of the other sums are given by $a_k = 0$, for $k = i_2, ..., i_{l-1}$. As to the combinatorial factor, it now depends only on $n, l, \{i_j\}_2^l$ and is written as $< n, l, \{i_j\}_2^l >$. It counts the number of possibilities for the occurence of the considered term of the sum which is given by $\binom{n}{i_0}\ \binom{n-i_0}{i_1} \cdots \binom{n-\sum_{j=2}^{l-1}\ i_j}{i_l} \;=\; n!/(\prod_{j=0}^{l}\ i_j!\ (n - \sum_{j=0}^{l}\ i_j)!$. Inserting i_0 and i_1 from (ii') and (i'), respectively, remembering (ii), produces $< n, l, \{i_j\}_2^l >$ as given in (32). Finally, $\sum < n, l, \{i_j\}_2^l >\ C_0^{i_0}\ C_1^{i_1} \cdots C_l^{i_l}$ is transformed into $(l-1)$ nested sums with boundaries a_k and $\lfloor b_k \rfloor$ after replacement of i_1 and i_0. This completes the proof of (29) for the non-trivial $l \geq 2$ cases. □

Corollary 1: *Catalan* convolutions

For *Catalan*'s sequence $\{C_n\}_0^\infty$ the n-th convolution sequence is for $n \in \mathbf{N}$ given by $C_0(n) = 1$, $C_1(n) = n$ and, for $l = 2, 3, ..$, by

$$C_l(n) \;=\; \Big(\prod_{k=2}^{l} \sum_{i_k = a_k}^{\lfloor b_k \rfloor}\Big)\ < n, l, \{i_j\}_2^l >\ \prod_{j=2}^{l} \Big(\frac{C_j^{i_j}}{i_j!}\Big)\ , \tag{35}$$

with (30), (31) and (32).

Proof: This is *lemma 1* with $C_0 = 1 = C_1$. □

Example 1: $C_4(3) \;=\; 3C_4 \;+\; 6C_3 \;+\; 3C_2^2 \;+\; 3C_2 \;=\; 90$.

Corollary 2: With the *Catalan* generating function $c(x)$ and the definition

$c^{-n}(x) =: \sum_{l=0}^{\infty}\ C_l(-n)\ x^l$, for $n \in \mathbf{N}$, one has for $l = 2, 3, ...$

$$C_l(-n) \;=\; (-1)^l \Big(\prod_{k=2}^{l} \sum_{i_k = a_k}^{\lfloor b_k \rfloor} \frac{(-1)^{(k-1)i_k}}{i_k!}\Big)\ < n, l, \{i_j\}_2^l >\ \prod_{j=2}^{l-1} C_j^{i_{j+1}}\ , \tag{36}$$

with (30), (31), (32) and *Catalan*'s numbers C_k. In addition: $C_0(-n) = 1$, $C_1(-n) = -n$.

Proof: *Lemma 1* is used for powers of $c(x)$ replaced by those of $c^{-1}(x) = 1 - x\ c(x)$, with the *Catalan* generating function $c(x)$. Hence $c^{-1}(x) = \sum_{k=0}^{\infty}\ C_k(-1)\ x^k$ with

$C_k(-1) = \begin{cases} 1 & \text{for } k = 0 \\ -C_{k-1} & \text{for } k = 1, 2, ... \end{cases}$. Then in *lemma 1* C_k is replaced by $C_k(-1)$. □

Example 2: $C_4(-3) \;=\; -3C_3 \;+\; 6C_2 \;-\; 3 \;+\; 3 \;=\; -3$.

Convolutions of *Catalan*'s sequence have been encountered in various contexts, for example, in the enumeration of non-intersecting path pairs on a square lattice [12], [18], [5], and in the problem of inverting triangular matrices with *Pascal* triangle entries [6] (and earlier works cited there). They also appear in [15], p.148.

Note 4: *Shapiro*'s *Catalan* triangle has entries $B_{n,k} = \frac{n}{n}\binom{n}{n-k}$ for $n \geq k \geq 1$, and $B_{n,k} = [x^n](x\ c^2(x))^k$, with $[x^n]f(x)$ denoting the coefficient of x^n in the expansion of $f(x)$ around $x = 0$. Here $\hat{c}(x) = (c(x)-1)/x = c^2(x)$. (See [12], propositions (2.1) and (3.3) with $i_j \in \mathbf{N}$, *not* $\mathbf{N}_0$.) In [18] this triangle of numbers from [12] reappears as $b(n,k)$ and it is shown there that $B_{n,k} \equiv b(n,k) = [x^n](x\ c^2(x))^k$, in accordance with the identity $\hat{c}(x) = c^2(x)$. Therefore, only even powers of $c(x)$ appear in *Shapiro*'s *Catalan* triangle. In [5] $C_l(n)$ appears as special case ${}_2d_{2-n,l+1}$. In [6] all powers of $c(x)$ show up as convolutions for the special case of the S_1 sequence there. The entries of the S_1-array, p. 397, are $[x^n]c^{k+1}(x)$ for $n, k \in \mathbf{N}_0$.

The referee of this paper noticed that the inverse of the lower triangular matrix $S_{n,k} = [x^k]\,S_n(x)$, for $n, k \in \mathbf{N}_0$, with *Chebyshev*'s $S_n(x) = U_n(x/2)$ polynomials is the lower triangular convolution matrix obtained from its first (k=0) column sequence generated by $c(x^2)$ (*Catalan* numbers alternating with zeros). This follows from the fact that the $\mathbf{S}-$matrix is also a lower triangular convolution matrix with generating function $1/(1+x^2)$ of its first column. See [13] for such type of matrices $\mathbf{M}$ and the relation between the generating functions of the first columns of $\mathbf{M}$ and $\mathbf{M}^{-1}$. The head of this *Catalan* triangle can be viewed under number A053121 in the on-line data-base [14]. See also [6] for inverses of *Pascal*-type arrays.

Lemma 2: Explicit form of *Catalan* convolutions [12],[18],[6],[4],[11],[5]

For $n \in \mathbf{R}$, $l \in \mathbf{N}_0$:

$$C_l(n) = \frac{n}{l}\binom{2l+n-1}{l-1} = \frac{n}{n+2l}\binom{n+2l}{l} = \frac{n}{l+n}\binom{2l+n-1}{l}. \tag{37}$$

Proof: Three equivalent expressions have been given for convenience. See [4], p. 201, eq.(5.60), with $\mathcal{B}_2(z) = c(z)$, $t \to 2, k \to l, r \to n$. The proof of this eq.(5.60) appears as (7.69) on p.349, with $m = 2, n = l \in \mathbf{R}$.

The same formula occurs as exercise nr. 213 in Vol.1 of [11] for $\beta = 2$ as a special instance of exercises nrs. 211, 212. Put $\alpha = n$ and $n = l$ in the solution of exercise nr. 213 on p. 301.

In order to prove this lemma from [12] or [18] one can use $C_l(n) = \sum_{j=0}^{\min(l,n)}\binom{n}{j}\hat{C}_l(j)$ obtained from $c(x) =: 1 + \hat{c}(x)$ with $\hat{c}^n(x) =: \sum_{k=n}^{\infty}\hat{C}_k(n)\ x^{k-n}$. The result in [12] and [18] is, with this notation, $\hat{C}_l(j) = B_{l,j} = b(l,j) = \frac{j}{l}\binom{2l}{l-j}$. Inserting this in the given sum, making use of the identity $j\binom{n}{j} = n\binom{n-1}{j-1}$ and the *Vandermonde* convolution identity, leads to *lemma 2* at least for positive integer n but one can continue this formula to real (or complex) n.

In [6] one finds this result as eq.(3.1), p.402, for $i = 1$: $s_1(l,n) = C_l(n)$.

In [5] ${}_2d_{2-n,l+1} = C_l(n)$ with the result given in theorem 2.3, eq. (2.6), p.71. □

Note 5: As a side remark we mention that from (37) $E_l(x) := l!\ C_l(x)$ (with real $n = x$) is a polynomial of degree l, *viz* $\prod_{j=0}^{l-1}(x+l+1+j)$. These polynomials which are not the subject of this work are known (see [8] and references given there) as exponential convolution polynomials satisfying $E_l(x+y) = \sum_{k=0}^{l}\binom{l}{k}\ E_k(x)\ E_{l-k}(y)$.

We now compute the coefficients $C_l(n) = [x^l]c^n(x)$ (see *Note 4* for this notation) from our formula given in *proposition 1*. This can be done for $n \in \mathbf{Z}$.

First consider $n \in \mathbf{N}_0$. For $n = 0$ and $n = 1$ there is nothing new due to the inputs $S_{-2} = -1$, $S_{-1} = 0$ and $S_0 = 1$. $C_l(n) = 0$ for negative integer l. Therefore, terms proportional to $1/x^l$ with $l \in \mathbf{N}$ have to cancel in (23), or (1). For $n = 2, 3, ...$ terms of the type $1/x^{n-j}$ occur for $j \in \{1, 2, ..., \lfloor n/2 \rfloor\}$. The coefficient of $1/x^{n-j}$ in $p_{n-1}(x)$ is $(-1)^j \binom{n-1-j}{j-1}$ (see Note 3 for the explicit form of p_{n-1}). For the $1/x^{n-j}$ coefficient in $q_{n-1}(x)\ c(x)$ one finds the convolution $\sum_{l=0}^{j-1} (-1)^{j-l-1} \binom{n-(j-l)}{j-l-1} C_l$. Compensation of both coefficients leads to identity $(P1)$ given in (4), after $(j-1)$ has been traded for p. Thus, after a shift $n \to n+2$:

Proposition 2: Identity $(P1)$

For $n \in \mathbf{N}_0$ and $p = 0, 1, ..., \lfloor \frac{n}{2} \rfloor$ identity $(P1)$, given in eq.(4) holds.

Example 3: $n = 2(k-1),\ p = k-1$, and $n = 2k-1,\ p = k-1$ for $k \in \mathbf{N}$

$\sum_{l=0}^{k-1} (-1)^l \binom{k+l}{2l+1} C_l = 1$, $\sum_{l=0}^{k-1} (-1)^l \binom{k+l+1}{2(l+1)} C_l = k$.

e.g. $k = 3$: $3\,C_0 - 4\,C_1 + 1\,C_2 = 1$, $6\,C_0 - 5\,C_1 + 1\,C_2 = 3$.

For $n = 2, 3, ...$ terms in (1), or (23), proportional to x^k with $k \in \mathbf{N}_0$ arise only from $q_{n-1}(x)\ c(x)$, and they are given by the convolution (cf. Note 4) $\sum_{l=0}^{\lfloor (n-1)/2 \rfloor} (-1)^l \binom{n-1-l}{l} C_{k+n-1-l}$. For $n = 1$ this is C_k. The *lhs.* of (1) contributes $C_k(n)$, and $C_k(1) = C_k$. Therefore:

Proposition 3: Identity $(P3)$

For $n \in \mathbf{N}$, $k \in \mathbf{N}_0$ identity $P(2)$, given in eq.(5) with (3) holds.

Example 4: $k = 0$, $(n-1) \to n$: $\sum_{l=1}^{\lfloor n/2 \rfloor} (-1)^{l+1} \binom{n-l}{l} C_{n-l} = C_n - 1$,

e.g. $n = 3$: $2\,C_2 = C_3 - 1$, $n = 4$: $3\,C_3 - 1\,C_2 = C_4 - 1$.

Now consider negative powers in (1), *i.e.* $c^{-n}(x)$, $n \in \mathbf{N}$. No negative powers of x appear (*cf.* footnote 4 for the explicit form of $p_{-(n+1)}(x)$ and $q_{-(n+1)}(x)$). The coefficient of x^k, $k \in \mathbf{N}_0$, of the *rhs.* of (1) is $(-1)^k \binom{n-k}{k} - \sum_{l=0}^{\lfloor (n-1)/2 \rfloor} (-1)^l \binom{n-1-l}{l} C_{k-1-l}$, where the first term, arising from $p_{-(n+1)}(x)$, contributes only for $k \in \{0, 1, ..., \lfloor n/2 \rfloor\}$. In the summation one also needs $l \leq k-1$ because no *Catalan* numbers with negative index occur in (1). The *lhs.* of (1) has $[x^k]c^{-n}(x) = C_k(-n)$. From the last eq. in (37) one finds $C_k(-n) = \frac{n}{n-k} \binom{2k-n-1}{k} = (-1)^k \frac{n}{n-k} \binom{n-k}{k}$. In the last eq. the upper index in the binomial has been negated (*cf.* [4], (5.14)). Two sets of identities follow, depending on the range of k:

Proposition 4: Identity $(P3)$

For $n \in \mathbf{N}$, $k \in \{0, 1, ..., \lfloor \frac{n}{2} \rfloor\}$ identity $(P3)$, given in eq.(6) holds.

Example 5: $k = 3,\ n \geq 6$: $C_2 - (n-2)\,C_1 + \binom{n+3}{2} C_0 = \binom{n-4}{2}$.

Proposition 5: Identity $(P4)$

For $n \in \mathbf{N}$, $k \in \mathbf{N}$ with $k \geq \lfloor \frac{n}{2} \rfloor + 1$ identity $(P4)$, given in eq.(7) holds.

In $(P4)$ only the $q_{-(n+1)}(x)\ c(x)$ part of (1) contributed and we used the first expression for $C_k(-n)$ in (37). In $(P3)$, where also $p_{-(n+1)}(x)$ contributed, we used the negated binomial coefficient for $C_l(-n)$ and absorption in the resulting one.

Note that (37) implies $C_k(-n) = -C_{k-n}(n)$ for $k, n \in \mathbf{N}$, and $k \geq n$. $C_k(0) = \delta_{k,0}$.

Example 6: $n = 5,\ k \geq 3:\ C_{k-1} - 3\,C_{k-2} + C_{k-3} = \frac{5}{k}\binom{2\,k-6}{k-1},$ *e.g.* $k = 7:\ C_6 - 3\,C_5 + C_4 = 20.$

If one uses the binomial formula for $c^{-n}(x) = (1 - x\ c(x))^n$ and $c^n(x) = \sum_{k=0}^{\infty} C_k(n)\ x^k$ one arrives at eq.(8).

3 Some families of integer sequences

In this section we present some sequences of positive integers which are defined with the help of the $\mathcal{U}_n$ polynomials (10).

$$u_n(m) := \mathcal{U}_n(1/m) = (\sqrt{m}\,)^n\ S_n(\sqrt{m}\,)\ . \tag{38}$$

The last eq. is due to (21). It will be shown that $u_n(m)$ is for each $m = 4, 5, ...$ and $n = -1, 0, ...$ a non-negative integer. Also negative integers $-m,\ m \in \mathbf{N}$ are of interest. In this case we add a sign factor.

$$v_n(m) := (-1)^n\,\mathcal{U}_n(-1/m) = (-i\sqrt{m}\,)^n\ S_n(i\sqrt{m}\,)\ . \tag{39}$$

From the S_n recursion relation (15) one infers those for the $u_n(m)$ and $v_n(m)$ sequences.

$$u_n(m) = m\ (u_{n-1}(m) - u_{n-2}(m))\ ,\quad u_{-1}(m) \equiv 0\ ,\quad u_0(m) \equiv 1\ , \tag{40}$$

$$v_n(m) = m\ (v_{n-1}(m) + v_{n-2}(m))\ ,\quad v_{-1}(m) \equiv 0\ ,\quad v_0(m) \equiv 1\ . \tag{41}$$

This shows that $v_n(m)$ constitutes a non-negative integer sequences for positive integer m. It describes certain generalized *Fibonacci* sequences (see *e.g.* [7] with $v_n(m) = W_{n+1}(0, 1; m, m)$). $v_n(m)$ counts, for example, the length of the binary word $W(m; n)$ obtained at step n from the substitution rule $1 \to 1^m0$, $0 \to 1^m$, starting at step $n = 0$ with 0. The number of 1's, resp. 0's in $W(m; n)$ is $2v_{n-1}(m)$, resp. $2v_{n-2}(m)$. *E.g.* $W(2; 3) = (110)^2 1^2 (110)^2 1^2$ and $v_3(2) = 16$, $2v_2(2) = 6$ and $2v_1(2) = 4$. For $m = 1$ this substitution rule produces the well-known Fibonacci-tree. Of course, one can define in a similar manner generalized *Lucas* sequences using the polynomials $\{\mathcal{V}_n\}$ given in (11). Each $u_n(m)$ sequence (which is identified with $W_{n+1}(0, 1; m, -m)$ of [7]) turns out to be composed of two simpler sequences, *viz* $u_{2k}(m) =: m^k\ \alpha_k(m)$ and $u_{2k-1}(m) =: m^k\ \beta_k(m)$, $k \in \mathbf{N}_0$. These new sequences, which are, due to (38), given by $\alpha_k = S_{2k}(\sqrt{m}\,)$ and $\beta_k(m) = S_{2k-1}(\sqrt{m}\,)/\sqrt{m}$, satisfy therefore the following relations.

$$\beta_{k+1}(m) = (m-2)\ \beta_k(m) - \beta_{k-1}(m)\ ,\quad \beta_0(m) \equiv 0\ ,\quad \beta_1(m) \equiv 1\ , \tag{42}$$

and

$$\alpha_{k-1}(m) = \beta_k(m) + \beta_{k-1}(m)\ . \tag{43}$$

From (42) it is now clear that $\beta_n(m)$ is a non-negative integer sequence for $m = 4, 5, ...$ (In [7] $\beta_n(m) = W_n(0, 1; m-2, -1)$.) This property is then inherited by the $\alpha_n(m)$ sequences due to (43), and then by the composed sequence $u_n(m)$.

The ordinary generating functions are

$$g_\beta(m; x) := \sum_{n=0}^{\infty} \beta_n(m)\ x^n = \frac{1}{x^2 - (m-2)x + 1}\ ,\quad g_\alpha(m; x) := \sum_{n=0}^{\infty} \alpha_n(m)\ x^n = \frac{1+x}{x^2 - (m-2)x + 1}\ , \tag{44}$$

$$g_u(m;x) := \sum_{n=0} u_n(m)\, x^n = \frac{1}{1 - m\, x + m\, x^2} \quad , \quad g_v(m;x) := \sum_{n=0} v_n(m)\, x^n = \frac{1}{1 - m\, x - m\, x^2} \, . \quad (45)$$

Note 6: The $\{\beta_n(m)\}$ sequences for $m = 4, 5, 6, 7, 8, 10$ appear in the book [14]. The case $m = 4$ produces the sequence of non-negative integers, $m = 5$ are the even indexed *Fibonacci* numbers. The $m = 9$ sequence appears in *Sloane's* On-Line-Encyclopedia [14] as $A004187$. The $\{\alpha_n(m)\}$ sequences for $m = 4, 5, 6$ and 8 appear in the book [14]. $m = 4$ yields the positive odd integer sequence; $m = 5$ is the odd indexed *Lucas* number sequence. The $m = 7$ sequence appears now as $A030221$ in the database [14]. The composed sequences $\{u_n(m)\}$ are not in the book but some of them are found in the database [14]. $m = 4$ is the sequence $(n + 1)\ 2^n$, $A001787$, and $m = 5, 6, 7$ appear now as $A030191, A030192, A030240$, respectively. As mentioned above $\{v_{n+1}(1)\}$ is the *Fibonacci* sequence. The instances $m = 2$ and 3 appear as $A002605$ and $A030195$, respectively, in the database [14].

Acknowledgements

The author likes to thank Dr. Stephen Bedding for a collaboration on powers of matrices. In *section 2* a result for 2×2 matrices (here $\mathbf{T}$) was recovered. The referee of this paper asked for a combinatorial interpretation of the $v_n(m)$ numbers. She or he also pointed out refs.[15], [13], [3], [17], and noticed that the inverse of the coefficient matrix for *Chebyshev's* S polynomials furnishes a *Catalan* triangle (see *Note 4*).

References

[1] M. Abramowitz and I. A. Stegun: " *Handbook of mathematical functions* ", Dover, 1968

[2] M. Gardner: " *Time Travel And Other Mathematical Bewilderments* ", ch. Twenty, W.H. Freeman, New York, 1988

[3] C.D. Godsil: " *Algebraic Combinatorics* ", Chapman & Hall, New York, London, 1993

[4] R.L. Graham, D.E. Knuth, and O. Patashnik: " *Concrete Mathematics* ", Addison-Wesley, Reading MA, 1989

[5] P. Hilton and J. Pedersen: " Catalan Numbers, Their Generalization, and Their Uses ", The Mathematical Intelligencer 13 (1991) 64-75

[6] V.E. Hoggatt, Jr. and M. Bicknell: " Catalan and Related Sequences Arising from Inverses of Pascal's Triangle Matrices ", The Fibonacci Quarterly 14 (1976) 395-405

[7] A.F. Horadam: " Special Properties of the Sequence $W_n(a, b; p, q)$ ", The Fibonacci Quarterly 5, 5 (1967) 424-434

[8] D.E. Knuth: " Convolution Polynomials ", The Mathematica J. 2, 1 (1992) 67-78

[9] W. Lang: " On Sums of Powers of Zeros of Polynomials ", Journal of Computational and Applied Mathematics 89 (1998) 237-256

[10] M. Petkovšek, H.S. Wilf, and D. Zeilberger: *" A=B "*, A K Peters, Wellesley, MA, 1996

[11] G. Pólya and G. Szegő: *" Aufgaben und Lehrsätze aus der Analysis I "*, Springer, Berlin, 1970, 4.ed.

[12] L.W. Shapiro: " A Catalan Triangle ", Discrete Mathematics 14 (1976) 83-90

[13] Louis W. Shapiro, Seyoum Getu, Wen-Jin Woan and Leon C. Woodson:" The Riordan Group ", Discrete Appl. Maths. 34 (1991) 229-239

[14] N.J.A. Sloane and S. Plouffe: *" The Encyclopedia of Integer Sequences "*, Academic Press, San Diego, 1995; see also N.J.A. Sloane's On-Line Encyclopedia of Integer Sequences, http//:www.research.att.com/ ~njas/sequences/index.html

[15] D.R. Snow: " Spreadsheets, Power Series, Generating Functions, and Integers ", The College Mathematics Journal 20 (1989) 143-152

[16] R.P. Stanley: *" Enumerative Combinatorics "*, vol. 2, Cambridge University Press, 1999; excerpt 'Problems on Catalan and Related Numbers', available from http/www-math.mit-edu/ ~rstan/ec/ec.html

[17] G. Viennot: " Une théorie combinatoire des polynômes orthogonaux generaux ", Notes de conférence donnée au Département de mathématique et d'informatique, Université du Québec à Montreal, Septembre- Octobre 1983

[18] Wen-Jin Woan, Lou Shapiro, and D.G. Rogers: " The Catalan Numbers, the Lebesgue Integral, and 4^{n-2} ", American Mathematical Monthly 101 (1997) 926-931

AMS MSC numbers: 11B83, 11B37, 33C45

TABLE : $p(n,m) = [1/x^m]\, p\_\{n\}(x)$ *coefficient matrix*

$n = -1..12,\ m = 0..12$

n\m	0	1	2	3	4	5	6	7	8	9	10	11	12
-1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	-1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	-1	0	0	0	0	0	0	0	0	0	0
3	0	0	1	-1	0	0	0	0	0	0	0	0	0
4	0	0	0	2	-1	0	0	0	0	0	0	0	0
5	0	0	0	-1	3	-1	0	0	0	0	0	0	0
6	0	0	0	0	-3	4	-1	0	0	0	0	0	0
7	0	0	0	0	1	-6	5	-1	0	0	0	0	0
8	0	0	0	0	0	4	-10	6	-1	0	0	0	0
9	0	0	0	0	0	-1	10	-15	7	-1	0	0	0
10	0	0	0	0	0	0	-5	20	-21	8	-1	0	0
11	0	0	0	0	0	0	1	-15	35	-28	9	-1	0
12	0	0	0	0	0	0	0	6	-35	56	-36	10	-1

KA-TP-4-1998

April 1998

On Polynomials Related to Powers and Derivatives of the Generating Function of Catalan's Numbers

Wolfdieter L a n g [1]

Institut für Theoretische Physik
Universität Karlsruhe
Kaiserstrasse 12, D-76128 Karlsruhe, Germany

Abstract

Arbitrary powers of the generating function $c(x)$ of *Catalan*'s numbers are written as $c^n(x) := -(\frac{1}{\sqrt{x}})^n \; S_{n-2}(\frac{1}{\sqrt{x}}) \; + \; (\frac{1}{\sqrt{x}})^{n-1} \; S_{n-1}(\frac{1}{\sqrt{x}}) \; c(x)$, with *Chebyshev*'s polynomials of the second kind $S_n(y) = U_n(y/2)$ which are also defined for real (or complex) n. This formula leads to four sets of identities involving *Catalan* numbers.

The nth derivative of this generating function $c(x)$ is expressed as $\frac{1}{n!}\frac{d^n c(x)}{dx^n} \; = \; \big(a_{n-1}(x) \; + \; b_n(x) \; c(x)\big)/(x(1-4x))^n$, with certain polynomial systems $\{a_n\}$ and $\{b_n\}$ which are given explicitly. The coefficients of the $\{a_n\}$ polynomials furnish a triangle of numbers $A(n,k)$ which generalizes *Catalan*'s numbers. It is related to a convolution of the *Catalan* sequence with $2k$-fold convolutions of the central binomial coefficient sequence. Also, an associated rectangular array $\hat{A}(n,k)$ of numbers is defined. The triangle of numbers of the $\{b_n\}$ coefficients is related to the $(2k+1)$-fold convolution of the central binomial number sequence. This formula for the derivatives of $c(x)$ implies identities involving *Catalan*'s numbers as well as central binomial coefficients.

1 Introduction and Summary

Catalan's sequence of numbers $\{C_n\}_0^\infty \; = \; \{1, 1, 2, 5, 14, 42, ...\}$ (nr.1459 and $A000108$ of [10]) emerges in the solution of many combinatorial problems (see [1],[2],[3],[11] (also for further references). It also shows up in the asymptotic moments of zeros of scaled *Laguerre* and *Hermite* polynomials [6]. The ordinary generating function $c(x) = \sum_{n=0}^\infty C_n \; x^n$ is the solution of the quadratic equation $x \; c^2(x) - c(x) + 1 \; = \; 0$ with $c(0) = 1$. Therefore, every positive integer power of $c(x)$ can be written as

$$c^n(x) \; = \; p_{n-1}(x)1 \; + \; q_{n-1}(x) \; c(x) \quad , \tag{1}$$

with certain polynomials p_{n-1} and q_{n-1}, both of degree $(n-1)$, in $1/x$. In *section* 2 they are shown to be related to Chebyshev's polynomials of the second kind:

$$p_{n-1}(x) \; = \; -(\frac{1}{\sqrt{x}})^n \; S_{n-2}(\frac{1}{\sqrt{x}}) \quad , \quad q_{n-1}(x) \; = \; (\frac{1}{\sqrt{x}})^{n-1} \; S_{n-1}(\frac{1}{\sqrt{x}}) \; = \; -x \; p_n(x) \; , \tag{2}$$

[1]E-mail: wolfdieter.lang@physik.uni-karlsruhe.de http://www-itp.physik.uni-karlsruhe.de/~wl

with $S_n(y) = U_n(y/2)$. It is therefore possible to extend the range of the power n to integers (or to real or complex numbers). Because powers of a generating function correspond to convolutions of the generated number sequence the given decomposition of $c^n(x)$ will determine convolutions of the *Catalan* sequence. In passing, an explicit expression for general convolutions in the form of nested sums will also be given. Contact with the works of refs. [4],[9],[12], [3] will be made.

Together with the known (*e.g.* [2],[8]) result (valid for real n)

$$c^n(x) \;=\; \sum_{k=0}^{\infty} C_k(n)\; x^k \quad \text{, with } C_k(n) \;:=\; \frac{n}{n+2k}\binom{n+2k}{k} \;=\; \frac{n}{k+n}\binom{n-1+2k}{k}\,, \tag{3}$$

one finds from the alternative expression (1) for positive n two sets of identities:

$$(P1) \qquad \sum_{l=0}^{p}(-1)^l\binom{n-1-p+l}{p-l}\; C_l \;=\; \binom{n-2-p}{p}\,, \tag{4}$$

for $n \in \{2,3,...\},\ p \in \{0,1,2,...\lfloor\frac{n}{2}\rfloor - 1\}$, and

$$(P2) \qquad \sum_{l=0}^{\lfloor\frac{n-1}{2}\rfloor}(-1)^l\,\binom{n-1-l}{l}\; C_{k+n-1-l} \;=\; C_k(n)\,, \tag{5}$$

for $n \in \mathbf{N}$, $k \in \mathbf{N}_0$.

For negative powers in (1) two other sets of identities result:

$$(P3) \qquad \sum_{l=0}^{min(\lfloor\frac{n-1}{2}\rfloor,k-1)}(-1)^l\,\binom{n-1-l}{l}\; C_{k-1-l} \;=\; (-1)^{k+1}\binom{n-k-1}{k-1}\,, \tag{6}$$

for $n \in \mathbf{N},\ k \in \{0,1,2,...\lfloor\frac{n}{2}\rfloor\}$, and

$$(P4) \qquad \sum_{l=0}^{\lfloor\frac{n-1}{2}\rfloor}(-1)^l\,\binom{n-1-l}{l}\; C_{k-1-l} \;=\; -C_k(-n) \;=\; \frac{n}{k}\binom{2k-n-1}{k-1}, \tag{7}$$

for $n \in \mathbf{N}$, $k \in \mathbf{N}$ with $k \geq \lfloor\frac{n}{2}\rfloor + 1$. [2]

Another expression for the coefficients of negative powers of $c(x)$ is

$$C_k(-n) \;=\; \sum_{l=1}^{min(n,k)}(-1)^l\,\binom{n}{l}\; C_{k-l}(n)\,, \tag{8}$$

for $n,k \in \mathbf{N}$, and $C_0(-n) = 1$, $C_n(0) = \delta_{n,0}$. Also, from (3) $C_k(-n) = -C_{k-n}(n)$ for $n,k \in \mathbf{N}$ with $k \geq n$.

Section 3 deals with the derivatives of $c(x)$ where the following basic equation is used.

$$\frac{d\ c(x)}{dx} \equiv c'(x) \;=\; \frac{1}{x(1-4x)}\Big(1 + (-1+2x)\ c(x)\Big)\,. \tag{9}$$

[2]These identities can be continued for appropriate values of real n.

This eq. is equivalent to the simple recurrence relation valid for C_n: [3]

$$(n+2)\ C_{n+1}\ -\ 2(2n+1)\ C_n\ =\ 0\ ,\quad n=-1,0,1,...,\ \text{with}\ \ C_{-1}=-1/2\ \ . \tag{10}$$

The result for the n-th derivative is of the form

$$\frac{1}{n!}\frac{d^n c(x)}{dx^n}\ =\ \frac{1}{(x(1-4x))^n}\Big(a_{n-1}(x)\ +b_n(x)\ c(x)\Big), \tag{11}$$

with certain polynomials a_{n-1} of degree $n-1$ and b_n of degree n. These polynomials are found to be $b_n(x)\ =\ \sum_{m=0}^{n}(-1)^m B(n,m)\ x^{n-m}$ with

$$B(n,m)\ :=\ \binom{2n}{n}\ \binom{n}{m}/\binom{2m}{m}\ , \tag{12}$$

which defines a triangle of numbers for $n,m\in\mathbf{N}$, $n\geq m\geq 0$. Its head is depicted in *TAB. 1* with $B(n,m)=0$ for $n<m$. Another representation for these b_n polynomials is also found, *viz*

$$b_n(x)\ =\ -2\sum_{k=0}^{n} C_{k-1}\ x^k\ (4x-1)^{n-k}\ . \tag{13}$$

Equating both forms of $b_n(x)$ leads to a formula involving convolutions of Catalan numbers with powers of an arbitrary constant $\lambda:=(4x-1)/x$. This formula is given in *section 3* as eq.(71).

The other family of polynomials is $a_n(x)=\sum_{k=0}^{n}(-1)^k\ A(n+1,k+1)\ x^{n-k}$ with the triangular array $A(n,m)$ defined for $m=0$ by $A(n,0)=C_n$, and for $n\in\mathbf{N}, m\in\mathbf{N}$ with $n\geq m>0$ by the numbers

$$A(n,m)\ =\ \frac{1}{2}\binom{n}{m-1}\Big[4^{n-m+1}\ -\ \binom{2n}{n}/\binom{2(m-1)}{m-1}\Big]\ . \tag{14}$$

The head of this triangular array of numbers is shown in *TAB.2* with $A(n,m)=0$ for $n<m$. These results are solutions to recurrence relations which hold for $b_n(x)$ and $a_n(x)$ and their respective coefficients $B(n,m)$ and $A(n,m)$.

The triangle of numbers $A(n,m)$ is related to a rectangular array of numbers $\hat{A}(n,m)$, with $\hat{A}(0,0)=1$, $\hat{A}(n,0)=-C_n$ for $n\in\mathbf{N}$, and for $m\in\mathbf{N}, n\in\mathbf{N}_0$ by

$$A(n,m)\ =\ -\hat{A}(n-m,m)\ +\ 2^{2(n-m)+1}\binom{n-1}{m-1}\ , \tag{15}$$

or with (14), for $m\in\mathbf{N}, n\in\mathbf{N}_0$, by

$$\hat{A}(n,m)\ =\ \frac{1}{2}\binom{n+m}{n+1}\Big[\binom{2(n+m)}{n+m}/\binom{2(m-1)}{m-1}\ -\ 4^{n+1}\ \frac{m-1}{n+m}\Big]. \tag{16}$$

Part of the array $\hat{A}(n,m)$ is shown in *TAB. 3*, where it is called $C4(n,m)$.

It turns out that the mth column of the number triangle $A(n,m)$ is for $m=0,1,...$ determined by the generating function $c(x)(\frac{x}{1-4x})^m$. The mth column of the number triangle $B(n,m)$ is, for $m=0,1,...,$ generated by $\frac{1}{\sqrt{1-4x}}\ (\frac{x}{1-4x})^m$.

[3] Eq.(9) can, of course, also be found from the explicit form $c(x)\ =\ (1-\sqrt{1-4x}\ )/(2x)$.

Because differentiation of $c(x) = \sum_{k=0}^{\infty}\ C_k\ x^k$ leads to

$$\frac{1}{n!}\ \frac{d^n c(x)}{dx^n}\ =\ \sum_{k=0}^{\infty} C(n,k)\ x^k\ ,\ \text{with}\ C(n,k)\ :=\ \frac{1}{n!}\ \prod_{j=1}^{n}(k+j)\ C_{n+k}\ =\ \frac{(2(n+k))!}{n!k!(n+k+1)!}\ , \tag{17}$$

with $C(0,k) = C_k$, one finds, together with (11), the following identities, for $n \in \mathbf{N}$,

$p \in \{0, 1, 2, ..., n-1\}$

$$(D1):\ \ \sum_{k=0}^{p}(-1)^k\ C_k\ \binom{n}{p-k}\Big/\binom{2(n-p+k)}{n-p+k}\ =\ \frac{1}{2}\binom{n}{p+1}\Big\{2^{2(p+1)}\Big/\binom{2n}{n}\ -\ 1\Big/\binom{2(n-p-1)}{n-p-1}\Big\}$$

$$=\ A(n,n-p)\Big/\binom{2n}{n}\ , \tag{18}$$

and, for $n \in \mathbf{N}, k \in \mathbf{N}_0$,

$$(D2):\ \ \sum_{j=0}^{n}(-1)^j\ \Big(\binom{n}{j}\Big/\binom{2j}{j}\Big)\ \sum_{l=0}^{k}4^l\ \binom{n+l-1}{n-1}\ C_{k+j-l}\ =\ C(n,k)\Big/\binom{2n}{n}\ . \tag{19}$$

The remainder of this paper provides proofs for the above given statements. *Section 2* deals with integer (and real) powers of the generating function $c(x)$. Convolutions of general sequences are expressed there in terms of nested sums. In *Section 3* derivatives of $c(x)$ are treated.

2 Powers

The equation $x\ c^2(x) - c(x) + 1 = 0$ whose solution defines the generating function of *Catalan*'s numbers if $c(0) = 1$ can be considered as characteristic equation for the recursion relation

$$x\ r_{n+1}\ -\ r_n\ +\ r_{n-1}\ =\ 0\ \ ,\ \ \ n = 0, 1, ...\ \ , \tag{20}$$

with arbitrary inputs $r_{-1}(x)$ and $r_0(x)$. A basis of two linearly independent solutions is given by the *Lucas*-type polynomials $\{\mathcal{U}_n\}$ and $\{\mathcal{V}_n\}$, with standard inputs $\mathcal{U}_{-1} = 0$, $\mathcal{U}_0 = 1$, $(\mathcal{U}_{-2} = -x)$, and $\mathcal{V}_{-1} = 1$, $\mathcal{V}_0 = 2$, $(\mathcal{V}_1 = 1/x)$, in the *Binet* form

$$\mathcal{U}_{n-1}(x)\ =\ \frac{c_+^n(x)\ -\ c_-^n(x)}{c_+(x)\ -\ c_-(x)}\ , \tag{21}$$

$$\mathcal{V}_n(x)\ =\ c_+^n(x)\ +\ c_-^n(x)\ =\ \frac{1}{x}(\mathcal{U}_{n-1}(x)\ -\ 2\,\mathcal{U}_{n-2}(x))\ , \tag{22}$$

with the two solutions of the characteristic equation, *viz* $c_\pm(x) := (1 \pm \sqrt{1-4x}\ )/(2x)$. $c(x) := c_-(x)$ satisfies $c(0) = 1$, and $c_+(x) = 1/(xc(x))$, as well as $c_+(x) + c(x)\ =\ 1/x$. From the recurrence (20) it is clear that for positive $n \neq 0$ $\mathcal{U}_n$ is a polynomial in $1/x$ of degree $n-1$. If $c_+(x) - c_-(x) = 0$, *i.e.* $x = 1/4$, eq.(21) is replaced by $\mathcal{U}_n(1/4) = 2^n(n+1)$. The second eq. in (22) holds because both sides of the eq. satisfy recurrence (20) and the inputs for $\mathcal{V}_0$ and $\mathcal{V}_1$ match. One may associate with the recurrence relation (20) a transfer matrix

$$\mathbf{C}(x)\ =\ \begin{pmatrix} 1/x & -1/x \\ 1 & 0 \end{pmatrix}\ \ ,\ \ Det\ \mathbf{C}(x)\ =\ 1/x\ . \tag{23}$$

With this matrix one can rewrite (20) as

$$\begin{pmatrix} r_n \\ r_{n-1} \end{pmatrix} \;=\; \mathbf{C}(x) \begin{pmatrix} r_{n-1} \\ r_{n-2} \end{pmatrix} \;=\; \mathbf{C}^n(x) \begin{pmatrix} r_0(x) \\ r_{-1}(x) \end{pmatrix} \tag{24}$$

Because $\mathbf{C}^n \;=\; \mathbf{C}\;\mathbf{C}^{n-1}$ with input $\mathbf{C}^1 = \mathbf{C}(\mathbf{x})$ given by (23), one finds from the recurrence relation (20) with $r_n = \mathcal{U}_n$

$$\mathbf{C}^n(x) \;=\; \begin{pmatrix} \mathcal{U}_n(x) & -\frac{1}{x}\,\mathcal{U}_{n-1}(x) \\ \\ \mathcal{U}_{n-1}(x) & -\frac{1}{x}\,\mathcal{U}_{n-2}(x) \end{pmatrix} \quad . \tag{25}$$

Note that for $x = 1$ one has $c_{\pm}(1) = (1 \pm i\sqrt{3})/2$, which are 6th roots of unity, and the related period 6 sequences are $\{\mathcal{U}_n(1)\}_{-1}^{\infty} \;=\; \{\overline{0, 1, 1, 0, -1, -1,}\}$, as well as $\{\mathcal{V}_n(1)\}_0^{\infty} \;=\; \{\overline{2, 1, -1, -2, -1, 1,}\}$. This follows from eqs. (21) and (22). It is convenient to map the recursion relation (20) to the familiar one for *Chebyshev*'s $S_n(x) = U_n(x/2)$ polynomials of the second kind, *viz*

$$S_n(x) \;=\; x\;S_{n-1}(x) \;-\; S_{n-2}(x) \quad , \quad S_{-1} = 0,\; S_0 = 1 \;, \tag{26}$$

with characteristic equation $\lambda^2 - x\lambda + 1 = 0$ and solutions $\lambda_{\pm}(x) = \frac{x}{2}(1 \pm \sqrt{1 - (2/x)^2}\;)$, satisfying $\lambda_+(x)\;\lambda_-(x) = 1$ and $\lambda_+(x) + \lambda_-(x) = x$. The relation to $c_{\pm}(x)$ is

$$\sqrt{x}\;c_{\pm}(x) \;=\; \lambda_{\pm}(1/\sqrt{x}) \;. \tag{27}$$

The *Binet* form of the corresponding two independent polynomial systems is

$$S_{n-1}(x) \;=\; \frac{\lambda_+^n(x) \;-\; \lambda_-^n(x)}{\lambda_+(x) \;-\; \lambda_-(x)} \;, \tag{28}$$

$$2\;T_n(x/2) \;=\; \lambda_+^n(x) \;+\; \lambda_-^n(x) \;, \tag{29}$$

and $T_n(x/2) \;=\; (S_n(x) - S_{n-2}(x))/2$ are *Chebyshev*'s polynomials of the first kind.

The extension to negative integer indices runs as follows

$$\mathcal{U}_{-n}(x) \quad = \quad -x^{n-1}\,\mathcal{U}_{n-2}(x) \;, \tag{30}$$

$$S_{-(n+2)}(x) \quad = \quad -S_n(x) \;. \tag{31}$$

This follows from (21) and (28). Note that from (20) $\mathcal{U}_n$ is for positive n a monic polynomial in $1/x$ of degree n, and for negative n in general a non-monic polynomial in x of degree $\lfloor -\frac{n}{2} \rfloor$. It is possible to extend the range of n to complex numbers using the *Binet* forms.

Connection between both systems of polynomials is made, after using (21), (27) and (28),by

$$\mathcal{U}_n(x) \;=\; \left(\frac{1}{\sqrt{x}}\right)^n \;S_n(1/\sqrt{x}) \;. \tag{32}$$

This holds for $n \in \mathbf{Z}$, in accordance with (30) and (31).

After these preliminaries we are ready to state:

Proposition 1: The nth power of $c(x)$, the generating function of *Catalan*'s numbers, can, for $n \in \mathbf{Z}$, be written as

$$c^n(x) \quad = \quad -\frac{1}{x}\,\mathcal{U}_{n-2}(x) \;+\; \mathcal{U}_{n-1}(x)\;c(x) \;, \tag{33}$$

$$\quad = \quad -(\frac{1}{\sqrt{x}})^n \;S_{n-2}(1/\sqrt{x}) \;+\; (\frac{1}{\sqrt{x}})^{n-1} \;S_{n-1}(1/\sqrt{x})\;c(x). \tag{34}$$

Proof: Due to $c^2(x) = (c(x)-1)/x$ and $c^{-1}(x) = 1 - x\ c(x)$ one can, for $n \in \mathbf{Z}$, write $c^n(x) = p_{n-1}(x) + q_{n-1}(x)\ c(x)$. From $c^n(x) = c(x)\ c^{n-1}(x)$ one is led to $q_{n-1} = p_{n-2} + \frac{1}{x}\ q_{n-2}$ and $p_{n-1} = -\frac{1}{x}\ q_{n-2}$, or $q_{n-1} = (q_{n-2} - q_{n-3})/x$ with input $q_{-1} = 0,\ q_0 = 1$. Therefore, $q_{n-1}(x) = \mathcal{U}_{n-1}(x)$ and $p_{n-1}(x) = -\ \mathcal{U}_{n-2}(x)/x$. (34) then follows from (32). [4] □

Note 1: An alternative proof of *proposition 1* can be given starting with eqs.(28) and (29) which show, together with $\lambda_+(x) - \lambda_-(x) = \sqrt{x^2-4}$, that

$$\lambda_\pm^n(x) \ = \ T_n(x/2) \ \pm \ \sqrt{(x/2)^2 - 1}\ S_{n-1}(x)\ , \tag{35}$$

or, from $\pm\sqrt{(x/2)^2-1} \ = \ \lambda_\pm(x) - x/2$ and the S_n recurrence relation (26)

$$\lambda_\pm^n(x) \ = \ T_n(x/2) \ - \ \tfrac{1}{2}\ (S_n(x) + S_{n-2}(x)) \ + \ S_{n-1}(x)\ \lambda_\pm(x) \tag{36}$$

$$= \ -S_{n-2}(x) \ + \ S_{n-1}(x)\ \lambda_\pm(x)\ . \tag{37}$$

Now (34) follows from (27). This also proves that one may replace in *proposition 1* $c(x)$ by $c_+(x) = 1/(xc(x))$ from which one recovers the c^{-n} formula for $n \in \mathbf{N}$ in accordance with (30) and (31).

Note 2: For the transfer matrix $\mathbf{C}(\mathbf{x})$, defined in (25), one can prove for $n \in \mathbf{N}$ in an analogous manner

$$\mathbf{C}^n \ = \ -(\frac{1}{\sqrt{x}})^n\ S_{n-2}(1/\sqrt{x})\ \mathbf{1} \ + \ (\frac{1}{\sqrt{x}})^{n-1}\ S_{n-1}(1/\sqrt{x})\ \mathbf{C}(x), \tag{38}$$

by employing the *Cayley-Hamilton* theorem for the 2×2 matrix $\mathbf{C}$ with $tr\ \mathbf{C} \ = \ \frac{1}{x} \ = \ det\ \mathbf{C}$ which states that $\mathbf{C}$ satisfies the characteristic equation $\mathbf{C}^2 \ - \ \frac{1}{x}\ \mathbf{C} \ + \ \frac{1}{x}\ \mathbf{1} \ = \ 0$.

Powers of a function which generates a sequence generate convolutions of this sequence. Therefore, *proposition 1* implies that convolutions of the *Catalan* sequence can be expressed in terms of *Catalan* numbers and binomial coefficients. Before giving this result we shall present an explicit formula for the nth convolution of a general sequence $\{C_n\}$ generated by $c(x) \ = \ \sum_{l=0}^\infty\ C_l\ x^l$. Usually the convolution coefficients $C_l(n)$, defined by $c^n(x) = \sum_{l=0}^\infty\ C_l(n)\ x^l$, are written as

$$C_l(n) \ = \ \sum_{\sum_{j=1}^n i_j \ = \ l} C_{i_1}\ C_{i_2} \cdots C_{i_n} \quad , \quad \text{with } i_j \in \mathbf{N}_0\ . \tag{39}$$

An explicit formula with $(l-1)$ nested sums is the content of the next lemma.

Lemma 1: General convolutions

For $l = 2, 3, ...$

$$C_l(n) \ = \ C_0^{n-l}\ C_1^l\Big(\prod_{k=2}^{l}\ \sum_{i_k = a_k}^{\lfloor b_k \rfloor}\Big) \ < n, l, \{i_j\}_2^l > \ \prod_{j=2}^{l}\Big((\frac{C_j\ C_0}{C_1^j})^{i_j}\ \frac{1}{i_j!}\Big)\ , \tag{40}$$

with

$$b_2 = l/2 \quad , \quad b_k \ = \ (l - \sum_{j=2}^{k-1}\ j\ i_j)/k\ , \tag{41}$$

[4] Because $S_n(y) = \sum_{j=0}^{\lfloor n/2 \rfloor}(-1)^j \binom{n-j}{j} y^{n-2j}$ the explicit form of these polynomials (2) is $p_{n-1}(x) \ = \ \sum_{j=0}^{\lfloor n/2 \rfloor - 1}(-1)^{j+1}\binom{n-2-j}{j} x^{-(n-1-j)}$, $p_{-1} = 1,\ p_0 = 0$, and $q_{n-1}(x) \ = \ \sum_{j=0}^{\lfloor (n-1)/2 \rfloor}(-1)^j \binom{n-1-j}{j} x^{-(n-1-j)}$, $q_{-1} = 0$. For negative index one has, due to (31), $p_{-(n+1)}(x) \ = \ (\sqrt{x})^n\ S_n(1/\sqrt{x}) \ = \ \sum_{j=0}^{\lfloor n/2 \rfloor}(-1)^j\ \binom{n-j}{j}\ x^j$, and $q_{-(n+1)}(x) \ = \ -(\sqrt{x})^{n+1}\ S_{n-1}(1/\sqrt{x}) \ = \ -x\ \sum_{j=0}^{\lfloor (n-1)/2 \rfloor}(-1)^j\ \binom{n-1-j}{j}\ x^j$.

$$a_k = 0 \;\;,\;\; \text{for } k = 2,3,...,l-1;\;\; a_l \;=\; max\Big(0, \lceil \frac{l - n - \sum_{j=2}^{l-1}(j-1)\; i_j)}{l-1} \rceil\Big) \tag{42}$$

$$< n, l, \{i_j\}_2^l > \;=\; \frac{n!}{(n - l + \sum_{j=2}^{l}(j-1)\; i_j)!(l - \sum_{j=2}^{l} j\; i_j)!}. \tag{43}$$

The first product in (40) is understood to be ordered such that the sums have indices $i_2, i_3, ...i_l$ when written from the left to the right. In addition: $C_0(n) = C_0^n$ and $C_1(n) = n\; C_0^{n-1}\; C_1$.

Proof: $C_l(n)$ of (39) is rewritten first as

$$C_l(n) \;=\; \sum\; (n, l, \{i_j\}_0^l)\; C_0^{i_0}\; C_1^{i_1} \cdots C_l^{i_l} \;\;,\;\; i_j \in \mathbf{N}_0 \;\;, \tag{44}$$

where the sum is restricted by

$$(i): \qquad \sum_{j=0}^{l} j\; i_j \;=\; l \qquad \text{and} \qquad (ii): \qquad \sum_{j=0}^{l} i_j \;=\; n\;. \tag{45}$$

$(n, l, \{i_j\}_0^l)$ is a combinatorial factor to be determined later on. (*E.g.* for $n = 3, l = 5$ one has 4 terms in the sum: $i_5 = 1, i_0 = 2$; $i_4 = 1, i_1 = 1, i_0 = 1$; $i_3 = 1, i_2 = 1, i_0 = 1$; $i_3 = 1, i_2 = 2$, with other indices vanishing, and the combinatorial factors are $3, 6, 6, 3$, respectively.) (ii) restricts the sum to terms with n factors, and (i) produces the correct weight l. These restrictions are solved by (i') : $\quad i_1 \;=\; l - \sum_{j=2}^{l} j\; i_j$ and (ii') : $\quad i_0 \;=\; n - i_1 - \sum_{j=2}^{l} i_j \;=\; n - l + \sum_{j=2}^{l}(j-1)\; i_j$. From $i_1 \geq 0$,*i.e.* $l - \sum_{j=2}^{l} j\; i_j \;\geq 0$, one infers $i_2 \leq \lfloor \frac{l}{2} \rfloor$, thus $i_2 \in [0, \lfloor \frac{l}{2} \rfloor\,]$. For given i_2 in this range $i_3 \;\leq \lfloor \frac{l-2i_2}{3} \rfloor$, *etc.*, in general $0 \leq i_k \leq \lfloor (l - \sum_{j=2}^{k-1}\; j\; i_j)/k \rfloor$ for $k = 2, 3, ..., l$ with the sum replaced by zero for $k = 2$. This accounts for the upper boundaries $\lfloor b_k \rfloor$ in (41). Now, because $i_0 \geq 0$ (ii') implies a lower bound for i_l, the index of the last sum, *viz* $i_l \geq \lceil (l - n - \sum_{j=2}^{l-1}\; (j-1)\; i_j)/(l-1) \rceil$ with the ceiling function $\lceil \cdot \rceil$. In any case $i_l \geq 0$, therefore, the lower boundary for the i_l-sum is a_l as given in (42). All restrictions have then be solved and the lower boundaries of the other sums are given by $a_k = 0$, for $k = i_2, ..., i_{l-1}$. As to the combinatorial factor, it now depends only on $n, l, \{i_j\}_2^l$ and is written as $< n, l, \{i_j\}_2^l >$. It counts the number of possibilities for the occurence of the considered term of the sum which is given by $\binom{n}{i_0}\; \binom{n-i_0}{i_1} \cdots \binom{n - \sum_{j=2}^{l-1}\; i_j}{i_l} \;=\; n!/(\prod_{j=0}^{l}\; i_j!\; (n - \sum_{j=0}^{l}\; i_j)!$. Inserting i_0 and i_1 from (ii') and (i'), respectively, remembering (ii), produces $< n, l, \{i_j\}_2^l >$ as given in (43). Finally, $\sum\; < n, l, \{i_j\}_2^l >\; C_0^{i_0}\; C_1^{i_1} \cdots C_l^{i_l}$ is transformed into $(l-1)$ nested sums with boundaries a_k and $\lfloor b_k \rfloor$ after replacement of i_1 and i_0. This completes the proof of (40) for the non-trivial $l \geq 2$ cases. □

Corollary 1: *Catalan* convolutions

For *Catalan*'s sequence $\{C_n\}_0^\infty$ the n-th convolution sequence is for $n \in \mathbf{N}$ given by $C_0(n) = 1$, $C_1(n) = n$ and, for $l = 2, 3, ..$, by

$$C_l(n) \;=\; \Big(\prod_{k=2}^{l} \sum_{i_k = a_k}^{\lfloor b_k \rfloor}\Big)\; < n, l, \{i_j\}_2^l >\; \prod_{j=2}^{l} \Big(\frac{C_j^{i_j}}{i_j!}\Big)\;, \tag{46}$$

with (41), (42) and (43).

Proof: This is *lemma 1* with $C_0 = 1 = C_1$. □

Example 1: $C_4(3) \;=\; 3C_4 \;+\; 6C_3 \;+\; 3C_2^2 \;+\; 3C_2 \;=\; 90$.

Corollary 2: With the *Catalan* generating function $c(x)$ and the definition $c^{-n}(x) =: \sum_{l=0}^{\infty}\; C_l(-n)\; x^l$,

for $n \in \mathbf{N}$, one has for $l = 2, 3, ...$

$$C_l(-n) \;=\; (-1)^l \Big(\prod_{k=2}^{l} \sum_{i_k=a_k}^{\lfloor b_k \rfloor} \frac{(-1)^{(k-1)i_k}}{i_k!} \Big) \;< n, l, \{i_j\}_2^l > \; \prod_{j=2}^{l-1} C_j^{i_{j+1}} \;, \tag{47}$$

with (41), (42), (43) and *Catalan*'s numbers C_k. In addition: $C_0(-n) = 1$, $C_1(-n) = -n$.

Proof: *Lemma 1* is used for powers of $c(x)$ replaced by those of $c^{-1}(x) = 1 - x\, c(x)$, with the *Catalan* generating function $c(x)$. Hence $c^{-1}(x) = \sum_{k=0}^{\infty} C_k(-1)\, x^k$ with

$C_k(-1) = \begin{cases} 1 & \text{for } k = 0 \\ -C_{k-1} & \text{for } k = 1, 2, ... \end{cases}$. Then in *lemma 1* C_k is replaced by $C_k(-1)$. □

Example 2: $C_4(-3) \;=\; -3C_3 \;+\; 6C_2 \;-\; 3 \;+\; 3 \;=\; -3$.

Convolutions of *Catalan*'s sequence have been encountered in various contexts. For example, in the enumeration of non-intersecting path pairs on a square lattice [9], [12], [3], and in the problem of inverting triangular matrices with *Pascal* triangle entries [4] (and earlier works cited there). [5]

Lemma 2: Explicit form of *Catalan* convolutions [9],[12], [4],[2],[8],[3]

For $n \in \mathbf{R}$, $l \in \mathbf{N}_0$:

$$C_l(n) \;=\; \frac{n}{l} \binom{2l+n-1}{l-1} \;=\; \frac{n}{n+2l} \binom{n+2l}{l} \;=\; \frac{n}{l+n} \binom{2l+n-1}{l} . \tag{48}$$

Proof: Three equivalent expressions have been given for convenience. See [2], p. 201, eq.(5.60), with $\mathcal{B}_2(z) \;=\; c(z)$, $t \to 2, k \to l, r \to n$. The proof of this eq.(5.60) appears as (7.69) on p.349, with $m = 2, n = l \in \mathbf{R}$.

The same formula occurs as exercise nr. 213 in Vol.1 of [8] for $\beta = 2$ as a special instant of exercises nrs. 211, 212. Put $\alpha = n$ and $n = l$ in the solution of exercise nr. 213 on p. 301.

In order to prove this lemma from [9] or [12] one can use $C_l(n) \;=\; \sum_{j=0}^{\min(l,n)} \binom{n}{j}\, \hat{C}_l(j)$ obtained from $c(x) =: 1 + \hat{c}(x)$ with $\hat{c}^n(x) =: \sum_{k=n}^{\infty} \hat{C}_k(n)\, x^{k-n}$. The result in [9] and [12] is, with this notation, $\hat{C}_l(j) \;=\; B_{l,j} \;=\; b(l,j) \;=\; \frac{j}{l} \binom{2l}{l-j}$. Inserting this in the given sum, making use of the identity $j \binom{n}{j} \;=\; n \binom{n-1}{j-1}$ and the *Vandermonde* convolution identity, leads to *lemma 2* at least for positive integer n but one can continue this formula to real (or complex) n.

In [4] one finds this result as eq.(3.1), p.402, for $i = 1$: $s_1(l, n) = C_l(n)$.

In [3] ${}_2d_{2-n,l+1} \;=\; C_l(n)$ with the result given in theorem 2.3, eq. (2.6), p.71. □

[5] *Shapiro*'s *Catalan* triangle has entries $B_{n,k} \;=\; \frac{k}{n} \binom{2n}{n-k}$ for $n \geq k \geq 1$, and $B_{n,k} = [x^n]\big(x^k\ \hat{c}^k(x)\big)$, with $[x^n]f(x)$ denoting the coefficient of x^n in the expansion of $f(x)$ around $x = 0$. Here $\hat{c}(x) = (c(x) - 1)/x \;=\; c^2(x)$. (See [9], propositions (2.1) and (3.3) with $i_j \in \mathbf{N}$, *not* $\mathbf{N}_0$.) In [12] this triangle of numbers from [9] reappears as $b(n,k)$ and it is shown there that $B_{n,k} \equiv b(n,k) \;=\; [x^n](x\ c^2(x))^k$, in accordance with the identity $\hat{c}(x) \;=\; c^2(x)$. Therefore, only even powers of $c(x)$ appear in *Shapiro*'s *Catalan* triangle. In [3] $C_l(n)$ appears as special case ${}_2d_{2-n,l+1}$. In [4] all powers of $c(x)$ show up as convolutions for the special case of the S_1 sequence there. The entries of the S_1-array, p. 397, are $[x^n]c^{k+1}(x)$ for $n, k \in \mathbf{N}_0$.

We now compute the coefficients $C_l(n) = [x^l]c^n(x)$ (see footnote 5 for this notation) from our formula given in *proposition 1.* This can be done for $n \in \mathbf{Z}$.

First consider $n \in \mathbf{N}_0$. For $n = 0$ and $n = 1$ there is nothing new due to the inputs $S_{-2} = -1$, $S_{-1} = 0$ and $S_0 = 1$. $C_l(n) = 0$ for negative integer l. Therefore, terms proportional to $1/x^l$ with $l \in \mathbf{N}$ have to cancel in (34). For $n = 2, 3, ...$ terms of the type $1/x^{n-j}$ occur for $j \in \{1, 2, ..., \lfloor n/2 \rfloor\}$. The coefficient of $1/x^{n-j}$ in $p_{n-1}(x)$ is $(-1)^j \binom{n-1-j}{j-1}$ (see footnote 4 for the explicit form of p_{n-1}). For the $1/x^{n-j}$ coefficient in $q_{n-1}(x)\ c(x)$ one finds the convolution $\sum_{l=0}^{j-1}(-1)^{j-l-1}\ \binom{n-(j-l)}{j-l-1}\ C_l$. Compensation of both coefficients leads to identity $(P1)$ given in (4), after $(j-1)$ has been traded for p. Thus:

Proposition 2: Identity $(P1)$

For $n = 2, 3, ...$ and $p\ =\ 0, 1, ..., \lfloor \frac{n}{2} \rfloor - 1$ identity $(P1)$, given in eq.(4) holds.

Example 3: $n = 2k,\ p = k-1$, and $n = 2k+1,\ p = k-1$ for $k \in \mathbf{N}$

$\sum_{l=0}^{k-1}(-1)^l\ \binom{k+l}{2l+1}\ C_l\ = 1 \qquad , \qquad \sum_{l=0}^{k-1}(-1)^l\ \binom{k+l+1}{2(l+1)}\ C_l\ = k$.

For $n = 2, 3, ...$ terms in (1), or (31), proportional to x^k with $k \in \mathbf{N}_0$ arise only from $q_{n-1}(x)\ c(x)$, and they are given by the convolution (cf. footnote 4) $\sum_{l=0}^{\lfloor (n-1)/2 \rfloor}\ (-1)^l\ \binom{n-1-l}{l}\ C_{k+n-1-l}$. For $n = 1$ this is C_k. The l.h.s. of (1) contributes $C_k(n)$, and $C_k(1) = C_k$. Therefore:

Proposition 3: Identity $(P3)$

For $n \in \mathbf{N}$, $k \in \mathbf{N}_0$ identity $P(2)$, given in eq.(5) with (3) holds.

Example 4: $k = 0$, $(n-1) \to n$: $\sum_{l=1}^{\lfloor n/2 \rfloor}\ (-1)^{l+1}\ \binom{n-l}{l}\ C_{n-l}\ =\ C_n\ -\ 1$

Now consider negative powers in (1), *i.e.* $c^{-n}(x)$, $n \in \mathbf{N}$. No negative powers of x appear (*cf.* footnote 4 for the explicit form of $p_{-(n+1)}(x)$ and $q_{-(n+1)}(x)$). The coefficient of x^k, $k \in \mathbf{N}_0$, of the *rhs.* of (1) is $(-1)^k\ \binom{n-k}{k}\ -\ \sum_{l=0}^{\lfloor (n-1)/2 \rfloor}\ (-1)^l\ \binom{n-1-l}{l}\ C_{k-1-l}$, where the first term, arising from $p_{-(n+1)}(x)$, contributes only for $k \in \{0, 1, ..., \lfloor n/2 \rfloor\}$. The *lhs.* of (1) has $[x^k]c^{-n}(x) = C_k(-n)$. From the last eq. in (48) one finds $C_k(-n) = \frac{n}{n-k}\ \binom{2k-n-1}{k}\ =\ (-1)^k\ \frac{n}{n-k}\ \binom{n-k}{k}$. In the last eq. the upper index in the binomial has been negated (*cf.* [2], (5.14)). Two sets of identities follow, depending on the range of k:

Proposition 4: Identity $(P3)$

For $n \in \mathbf{N}$, $k \in \{0, 1, ..., \lfloor \frac{n}{2} \rfloor\}$ identity $(P3)$, given in eq.(6) holds.

Proposition 5: Identity $(P4)$

For $n \in \mathbf{N}$, $k \in \mathbf{N}$ with $k \geq \lfloor \frac{n}{2} \rfloor\ + 1$ identity $(P4)$, given in eq.(7) holds.

In $(P4)$ only the $q_{-(n+1)}(x)\ c(x)$ part of (1) contributed and we used the first expression for $C_k(-n)$ in (48). In $(P3)$, where also $p_{-(n+1)}(x)$ contributed, we used the negated binomial coefficient for $C_l(-n)$ and absorption in the resulting one.

Note that (48) implies $C_k(-n)\ =\ -C_{k-n}(n)$ for $k, n \in \mathbf{N}$, and $k \geq n$. $C_k(0) = \delta_{k,0}$.

If one uses the binomial formula for $c^{-n}(x)\ =\ (1 - x\ c(x))^n$ and $c^n(x) = \sum_{k=0}^{\infty}\ C_k(n)\ x^k$ one arrives at eq.(8).

We close this section by presenting some sequences of positive integers which are defined with the help of the $\mathcal{U}_n$ polynomials (21).

$$a_n(m) \ := \ \mathcal{U}_n(1/m) \ = \ (\sqrt{m}\ )^n \ S_n(\sqrt{m}\ ) \ . \tag{49}$$

The last eq. is due to (32). It will be shown that $a_n(m)$ is for each $m = 4, 5, ...$ and $n = -1, 0, ...$ a non-negative integer. Also negative integers $-m$, $m \in \mathbf{N}$ are of interest. In this case we add a sign factor.

$$b_n(m) \ := \ (-1)^n \ \mathcal{U}_n(-1/m) \ = \ (-i\sqrt{m}\ )^n \ S_n(i\sqrt{m}\ ) \ . \tag{50}$$

From the S_n recursion relation (26) one infers those for the $a_n(m)$ and $b_n(m)$ sequences.

$$a_n(m) \ = \ m \ (a_{n-1}(m) \ - \ a_{n-2}(m)) \ \ , \ \ a_{-1}(m) \equiv 0 \ \ , \ \ a_0(m) \equiv 1 \ , \tag{51}$$

$$b_n(m) \ = \ m \ (b_{n-1}(m) \ + \ b_{n-2}(m)) \ \ , \ \ b_{-1}(m) \equiv 0 \ \ , \ \ b_0(m) \equiv 1 \ . \tag{52}$$

This shows that $b_n(m)$ constitutes a non-negative integer sequences for positive integer m. It describes certain generalized *Fibonacci* sequences (see *e.g.* [5] with $b_n(m) \ = \ W_{n+1}(0, 1; m, m)$). Of course, one can define in a similar manner generalized *Lucas* sequences using the polynomials $\{\mathcal{V}_n\}$ given in (22). Each $a_n(m)$ sequence (which is identified with $W_{n+1}(0, 1; m, -m)$ of [5]) turns out to be composed of two simpler sequences, *viz* $a_{2k}(m) =: m^k \ \alpha_k(m)$ and $a_{2k-1} =: m^k \ \beta_k(m)$, $k \in \mathbf{N}_0$. These new sequences, which are, due to (49) and (50), given by $\alpha_k = S_{2k}(\sqrt{m}\ )$ and $\beta_k(m) = S_{2k-1}(\sqrt{m}\ )/\sqrt{m}$, satisfy therefore the following relations.

$$\beta_{k+1}(m) \ = \ (m-2) \ \beta_k(m) \ - \ \beta_{k-1}(m) \ \ , \ \ \beta_0(m) \equiv 0 \ \ , \ \ \beta_1(m) \equiv 1 \ , \tag{53}$$

and

$$\alpha_{k-1}(m) \ = \ \beta_k(m) \ + \beta_{k-1}(m) \ \ . \tag{54}$$

From (53) it is now clear that $\beta_n(m)$ is a non-negative integer sequence for $m = 4, 5, ...$ (In [5] $\beta_n(m) = W_n(0, 1; m-2, -1)$.) This property is then inherited by the $\alpha_n(m)$ sequences due to (54), and then by the composed sequence $a_n(m)$. (Of course, one could also consider sequences built from negative and positive numbers, but we refrain from doing so here).

The ordinary generating functions are [6]

$$g_\beta(m; x) := \sum_{n=0}^{\infty} \ \beta_n(m) \ x^n = \frac{1}{x^2 - (m-2)x + 1} \ \ , \ \ g_\alpha(m; x) := \sum_{n=0}^{\infty} \ \alpha_n(m) \ x^n = \frac{1+x}{x^2 - (m-2)x + 1} \ , \tag{55}$$

$$g_a(m; x) \ := \sum_{n=0}^{\infty} \ a_n(m) \ x^n \ = \frac{1}{1 - m \ x + m \ x^2} \ \ , \ \ g_b(m; x) \ := \sum_{n=0}^{\infty} \ b_n(m) \ x^n \ = \frac{1}{1 - m \ x - m \ x^2} \ . \tag{56}$$

[6] The $\{\beta_n(m)\}$ sequences for $m = 4, 5, 6, 7, 8, 10$ appear in the book [10]. The case $m = 4$ produces the sequence of non-negative integers, $m = 5$ are the even indexed *Fibonacci* numbers. The $m = 9$ sequence appears only in *Sloane's* On-Line-Encyclopedia [10] as $A004187$. The $\{\alpha_n(m)\}$ sequences for $m = 4, 5, 6$ and 8 appear in the book [10]. $m = 4$ yields the positive odd integer sequence, $m = 5$ the odd indexed *Lucas* number sequence. The $m = 7$ sequence appears now as $A030221$ in the data bank [10]. The composed sequences $\{a_n(m)\}$ are not in the book but some of them are found in the data bank [10]. $m = 4$ is the sequence $(n+1) \ 2^n$, $A001787$, and $m = 5, 6, 7$ appear now as $A030191, A030192,$ $A030240$, respectively. As mentioned above $\{b_{n+1}(1)\}$ is the *Fibonacci* sequence. The instances $m = 2$ and 3 appear as $A002605$ and $A030195$, respectively, in the data bank [10].

3 Derivatives

The starting point is eq.(9) which can either be verified from the explicit form of the generating function $c(x)$ (*cf.* footnote 3), or by converting the recursion relation (10) for *Catalan*'s numbers into an eq. for their generating function. A computation of $\frac{1}{(n+1)!}\,\frac{d^{n+1}c(x)}{dx^{n+1}} = \frac{1}{n+1}\,\frac{d}{dx}(\frac{1}{n!}\,\frac{d^n c(x)}{dx^n})$ with Ansatz (11) and eq. (9) produces the following mixed relations between the quantities $a_n(x)$ and $b_n(x)$ and their first derivatives, valid for $n \in \mathbf{N}_0$,

$$(n+1)\; a_n(x) \;=\; x(1-4x)\; a'_{n-1}(x) \;+\; b_n(x) \;+\; n(8x-1)\; a_{n-1}(x)\;, \tag{57}$$

$$(n+1)\; b_{n+1}(x) \;=\; x(1-4x)\; b'_n(x) \;+\; (-(n+1)+2(1+4n)x)\; b_n(x)\;, \tag{58}$$

with inputs $a_{-1}(x) \equiv 0$ and $b_0(x) \equiv 1$.

From (58) and the input it is clear by induction that $b_n(x)$ is a polynomial in x of degree n. With this information (57) and the input show, again by induction, that the same statement holds for $a_n(x)$. Therefore we write, for $n \in \mathbf{N}_0$, [7]

$$a_n(x) \;=\; \sum_{k=0}^{n}(-1)^k\; a(n,k)\; x^{n-k}\;, \tag{59}$$

$$b_n(x) \;=\; \sum_{k=0}^{n}(-1)^k\; B(n,k)\; x^{n-k}\;, \tag{60}$$

with the triangular arrays of numbers $a(n,k)$ and $B(n,k)$ with row number n and column number $k \le n$.

We first solve the $b_n(x)$ eq.(58) by inserting (60) and deriving the recursion relation for the coefficients $B(n,m)$ after comparing coefficients of x^{n+1}, x^0, and x^{n-k} for $k = 0, 1, ..., n-1$.

$$x^{n+1}\;: \qquad (n+1)\; B(n+1,0) \;=\; 2(2n+1)\; B(n,0)\;, \tag{61}$$

$$x^{0}\;: \qquad B(n+1,n+1) \;=\; B(n,n)\;, \tag{62}$$

$$x^{n-k}\;: \qquad (n+1)\; B(n+1,k+1) \;=\; (k+1)\; B(n,k) \;+\; 2(2(n+k)+3)\; B(n,k+1)\;. \tag{63}$$

With the input $B(0,0) = 1$ one deduces from (61) for the leading coefficient of $b_n(x)$

$$B(n,0) \;=\; 2^n\frac{(2n-1)!!}{n!} \;=\; \frac{(2n)!}{n!\;n!} \;=\; \binom{2n}{n}\;, \tag{64}$$

and from (62)

$$B(n,n) \;\equiv 1 \quad,\quad \text{i.e.}\;\; b_n(0) = (-1)^n\;. \tag{65}$$

In order to solve (63) we inspect the $B(n,m)$ triangle of numbers *TAB.1*, and conjecture that for $n, m \in \mathbf{N}$

$$B(n,m) \;=\; 4\; B(n-1,m) \;+\; B(n-1,m-1)\;, \tag{66}$$

with input $B(n,0) = \binom{2n}{n}$ from (64).

If we use this conjecture in (63), written with $n \to n-1,\; k \to m-1$ we are led to consider the simple recursion

$$B(n,m) \;=\; \frac{n+1-m}{2(2m-1)}\; B(n,m-1)\;, \tag{67}$$

[7] The triangular array $a(n,k)$ will later be enlarged to another one which will then be called $A(n,k)$.

with input $B(n,0) = \binom{2n}{n}$ from (64).

The solution of this recursion is, for $n, m \in \mathbf{N}_0$, [8]

$$B(n,m) \;=\; \frac{1}{2^m(2m-1)!!}\,\frac{n!}{(n-m)!}\,\binom{2n}{n} \;=\; \frac{m!\;n!}{(2m)!\;(n-m)!}\,\binom{2n}{n} \;=\; \binom{2n}{n}\,\binom{n}{m}/\binom{2m}{m}\;. \qquad (68)$$

This result satisfies (61), *i.e.* (64), as well as (62), *i.e.* (65). It is also the solution to (63) provided we prove the conjecture (66) for $B(n,m)$ of (68). This can be done by using the form $B(n,m) = \frac{(2n)!\;m!}{(2m)!\;n!\;(n-m)!}$ and extracting this expression on the *rhs.* of (66). Then one is left to prove $1 \;=\; \frac{4}{2}\,\frac{n-m-1}{2n-1} \;+\; \frac{2m-1}{2n-1}$, which is trivial. Thus we have proved:

Proposition 6: Explicit form of $b_n(x)$

$B(n,m)$ given by eq. (68) is the solution to eqs.(61), (62), and (63). Hence $b_n(x)$, defined by (60) with $B(n,m)$ from (68), solves eq. (58) with $b_0(x) \equiv 1$.

One can derive another explicit representation for the $b_n(x)$ polynomials by converting the simple recurrence relation (67) into the following eq. for $b_n(x)$ defined by (60).

$$(1-4x)\;b_n'(x) \;+\; 2(2n-1)\;b_n(x) \;+\; 2\,\binom{2n}{n}\,x^n \;=\; 0\;. \qquad (69)$$

Now this first order linear and inhomogeneous differential eq. for $b_n(x)$ can be solved.

Proposition 7: Alternative form for $b_n(x)$

The solution to eq.(69) with input $b_n(0) \equiv (-1)^n$ is given by eq.(13), with $C_{-1} = -1/2$ and the *Catalan* numbers C_k for $k \in \mathbf{N}_0$.

Proof: This eq. is of the standard type $y' + f(x)\;y = g(x)$ with $y \equiv b_n$, $f(x) = 2(2n-1)/(1-4x)$ and $g(x) = 2(n+1)C_n x^n/(1-4x)$. $F(x) := \int dx\; f(x) \;=\; -\frac{1}{2}(2n-1)\;ln(1-4x) + const(n)$. $y = exp(-F(x))\;\{Const(n) - 2\;(n+1)C_n I_n(x)\}$ with $I_n(x) := \int dx\; x^n/(1-4x)^{n+1/2}$ and $exp(-F(x)) = (1-4x)^{n-1/2}$. The integral $I_n(x)$ can be computed by repeated partial integration, and it is found to be

$$I_n(x) \;=\; \frac{1}{n+1}\;\sum_{k=0}^{n}\;(-1)^k\;\frac{C_{n-k-1}}{C_n}\;x^{n-k}/(1-4x)^{n-k-1/2}\;, \qquad (70)$$

where we used $C_{-1} := -1/2$, compatible with the recursion (10). This leads to the desired result for $y \equiv b_n(x)$ if the integration constant $Const(n)$ is put to zero in order to satisfy $b_n(0) = (-1)^n$ and a resummation $k \to k-n$ is performed. □

Comparing this alternative form (13) for $b_n(x)$ with the one given by (60), together with (68), proves the following identity in n and $\lambda := (4x-1)/x$. The term $k=0$ in the sum (13) has been written separately.

Corollary 3: Convolution of *Catalan* sequence and powers of λ

$$s_{n-1}(\lambda) \;:= \lambda^{n-1}\;\sum_{k=0}^{n-1}\;\frac{C_k}{\lambda^k} \;=\frac{1}{2}\Big(\lambda^n \;-\; \binom{2n}{n}\;\sum_{k=0}^{n}\;(-1)^k\;(4-\lambda)^k\;\binom{n}{k}/\binom{2k}{k}\Big)\;, \qquad (71)$$

[8] With the *Pochhammer* symbol $(a)_n := \Gamma(n+a)/\Gamma(a)$ this result can also be written as $B(n,m) = ((2m+1)/2)_{n-m}\;4^{m-n}/(n-m)!$.

for $n \in \mathbf{N}$ and $\lambda \neq \infty$. Observe that $s_n(\lambda)$ is the convolution of the *Catalan* sequence with the sequence of powers of λ. Therefore, the (ordinary) generating function for the sequence $s_n(\lambda)$ is $g(\lambda; x) := \sum_{n=0}^{\infty} s_n(\lambda)\, x^n = c(x)/(1-\lambda\, x)$. [9]

The case $\lambda = 0$ ($x = 1/4$) is also covered by this formula. It produces from $s_n(0) = C_n$ the following identity.

Example 5: Case $\lambda = 0$ $(x = 1/4)$ [10]

$$\sum_{k=0}^{n} (-1)^{k+1} \binom{n}{k} 4^k / \binom{2k}{k} = \frac{1}{2n-1} . \tag{72}$$

We note that from (13) one has $-2b_{n+1}(1/4) = C_n/4^n$. [11]

If one puts in (13) $4x - 1 = x$, *i.e.* $x = 1/3$, one can identify the partial sum of *Catalan* numbers, $s_n(1)$ [12], as follows.

$$s_n(1) = \sum_{k=0}^{n} C_k = \frac{1}{2}(1 - 3^{n+1}\, b_{n+1}(1/3)\,). \tag{73}$$

If one puts $\lambda = 1$ in *Corollary 3* one finds also

Example 6:

$$2\, s_{n-1}(1) = 1 + \binom{2n}{n} \sum_{k=0}^{n} (-1)^{k+1} \binom{n}{k} 3^k / \binom{2k}{k} . \tag{74}$$

Another interesting example is the case $\lambda = 4$ ($x = \infty$). Here one finds a simple result for the convolution of *Catalan*'s sequence with powers of 4, *viz* [13]

Example 7: $\lambda = 4$ $(x = \infty)$

$$2\, s_{n-1}(4) = 4^n - \binom{2n}{n} . \tag{75}$$

The sequence for $\lambda = -1$ $(x = 1/5)$ is also non-negative, as can be seen by writing $s_{2k}(-1) = C_2 + \sum_{l=2}^{k} (C_{2l} - C_{2l-1})$ for $k \in \mathbf{N}$ and $s_{2k+1}(-1) = \sum_{l=1}^{k} (C_{2l+1} - C_{2l})$, and using $\triangle C_n := C_n - C_{n-1} = 3\frac{n-1}{n+1}\, C_{n-1} \geq 0$. [14]

Recursion (66) for $B(n, m)$ can be transformed into an eq. for the (ordinary) generating function for the sequence appearing in the mth column of the $B(n, m)$ triangle

$$G_B(m; x) := \sum_{n \geq m} B(n, m)\, x^n \quad , \tag{76}$$

---

[9] From the generating function the recurrence relation is found to be $s_n(\lambda) = \lambda\, s_{n-1}(\lambda) + C_n$, $s_{-1}(\lambda) \equiv 0$. The connection to the $b_n(x)$ polynomial is $s_n(\lambda) = \frac{1}{2}\Big(\lambda^{n+1} - (4-\lambda)^{n+1}\, b_{n+1}(1/(4-\lambda))\Big)$.

[10] This identity occurs in one of the exercises 2.7, 2, p.32, in [7].

[11] The large n behaviour of this sequence is known to be $C_n/4^n \sim \frac{1}{\sqrt{\pi}}\, \frac{1}{n^{3/2}}$, *cf.* [2], Exercise 9.60.

[12] This sequence $\{1, 2, 4, 9, 23, 65, 197, 626, 2056, ...\}$, appears as A014137 in the on-line encyclopedia [10].

[13] This sequence $\{1, 5, 22, 93, 386, 1586, 6476, ...\}$ appears in the book [10] as Nr. 3920 and as A000346 in the on-line encyclopedia. It will show up again in this work as $A(n+1, 1)$, the second column in the $A(n, m)$ triangle (*cf. TAB.2*).

[14] This is the sequence $\{1, 0, 2, 3, 11, 31, 101, 328, 1102, 3760, ...\}$ which appears now as $A032357$ in the on-line encyclopedia [10] .

with input $G_B(0;x) = \sum_{n=0}^{\infty} \binom{2n}{n} x^n = 1/\sqrt{1-4x}$, the generating function for the central binomial numbers. (66) implies for $m \in \mathbf{N}_0$ [15]

$$G_B(m;x) = \left(\frac{x}{1-4x}\right)^m \frac{1}{\sqrt{1-4x}}. \tag{77}$$

Therefore, we have proved:

Proposition 8: Column sequences of the $B(n,m)$ triangle

The sequence $\{B(n,m)\}_{n=m}^{\infty}$, defined, for fixed $m \in \mathbf{N}_0$, by (68) for $n \in \mathbf{N}_0$ is the convolution of the central binomial sequence $\{\binom{2k}{k}\}_0^{\infty}$ and the mth convolution of the (shifted) power sequence $\{0, 1, 4^1, 4^2, ...\}$.

In a similar vein we solve the $a_n(x)$ eq.(57) with $b_n(x)$ given by (60) and (68). The coefficients $a(n,k)$, defined by (59), have to satisfy, after comparing coefficients of x^n, x^0, and x^{n-k} for $k = 1, 2, ..., n-1$ and $n \in \mathbf{N}_0$:

$$x^n : \quad a(n,0) = 4\, a(n-1,0) + C_n, \tag{78}$$

$$x^0 : \quad (n+1)\, a(n,n) = 1 + n\, a(n-1,n-1)\ , \tag{79}$$

$$x^{n-k} : \quad (n+1)\, a(n,k) = k\, a(n-1,k-1) + 4(n+1+k)\, a(n-1,k) + B(n,k)\ . \tag{80}$$

We have used (64), *i.e.* $B(n,0) = (n+1)\, C_n$ in (78), as well as (65), *i.e.* $B(n,n) \equiv 1$, in (79). From (78) one finds with input $a(0,0) = 1$ [16]

$$a(n,0) = \sum_{k=0}^{n} C_k\, 4^{n-k}\ , \tag{81}$$

and from (79)

$$a(n,n) \equiv 1, \text{ or } a_n(0) = (-1)^n\ . \tag{82}$$

It is convenient to define $a(n-1,-1) := C_n$, $n \in \mathbf{N}_0$. Then the sequence $\{a(n,0)\}_{-1}^{\infty}$ is, with $a(-1,0) := 0$, the convolution of the sequence $\{a(k,-1)\}_{-1}^{\infty}$ and the shifted power sequence $\{0, 1, 4^1, 4^2, ...\}$. Before solving (80) with inserted $B(n,k)$ from (68) we therefore add to the trianglular array of numbers $a(n,m)$ the $m = -1$ column and an extra row for $n = -1$, and define a new enlarged triangular array for $n, m \in \mathbf{N}_0$ as

$$A(n,m) := a(n-1,m-1) \tag{83}$$

with $A(n,0) = a(n-1,-1) = C_n$ and $A(0,m) = a(-1,m-1) = \delta_{0,m}$. An inspection of the $A(n,m)$ triangular array, partly depicted in *TAB. 2*, leads to the conjecture

$$A(n,m) = 4\, A(n-1,m) + A(n-1,m-1)\ , \tag{84}$$

with $A(n,0) = C_n$ and $A(n,m) \equiv 0$ for $n < m$. [17] This conjecture is correct for $A(n+1,1) = a(n,0)$ found in (81), as well as for $A(n+1,n+1) = a(n,n) \equiv 1$ known from (82). The (ordinary) generating function for the sequence appearing in the mth column,

$$G_A(m;x) = \sum_{n=m}^{\infty} A(n,m)\, x^n\ , \tag{85}$$

[15] For $x\frac{d}{dx} G_B(m;x)$ see (92).

[16] $a(n,0) = s_n(4)$ of (71) with solution (75).

[17] This recursion relation can be employed to extend the array $A(n,m)$ to negative integer m values.

satisfies due to (84) $G_A(m;x) = \frac{x}{1-4x}\, G_A(m-1;x)$, remembering that $A(m-1,m) \equiv 0$, or because of $G_A(0;x) \;=\; c(x)$

$$G_A(m;x) \;=\; \Big(\frac{x}{1-4x}\Big)^m \; c(x) \; . \tag{86}$$

Because of (77) and $\sqrt{1-4x}\; c(x) \;=\; 2 - c(x)$ these generating functions of the conjectured $A(n,m)$ column sequences obey

$$G_A(m;x) \;=\; (2-c(x))\; G_B(m;x) \; . \tag{87}$$

If we use the conjecture (84) in (80) which is written with (83) in the form $(n+1)\; A(n+1,m+1) \;=\; m\; A(n,m) \;+\; 4(n+m+1)\; A(n,m+1) \;+\; B(n,m)$, for $n \in \mathbf{N}_0,\; m \in \{1,2,..,n-1\}$, we have

$$m\; A(n+1,m+1) \;-\; (n+1)\; A(n,m) \;+\; B(n,m) \;=\; 0 \; . \tag{88}$$

This recursion relation can be written with the help of the generating functions (76) and (85) as

$$(x\frac{d}{dx} + 1)\; G_A(m;x) \;-\; \frac{m}{x}\; G_A(m+1;x) \;=\; G_B(m;x) \; , \tag{89}$$

or with (86) (*i.e.* the conjecture) as

$$\Big(x\frac{d}{dx} + 1 \;-\; \frac{m}{1-4x}\Big)\; G_A(m;x) \;=\; G_B(m;x) \; . \tag{90}$$

Together with (87) this means

$$x\frac{d}{dx}\Big((2-c(x))\; G_B(m;x)\Big) \;=\; \Big[(\frac{m}{1-4x} - 1)(2-c(x)) \;+1\Big]\; G_B(m;x) \; . \tag{91}$$

If we can prove this eq. with $G_B(x)$ given by (77) we have shown that (80) is equivalent to the conjecture (84). In order to prove (91) we first compute from (77), for $m \in \mathbf{N}_0$,

$$x\frac{d}{dx}G_B(m;x) \;= (2 + \frac{m}{x})\; G_B(m+1;x) \;=\; \frac{2x+m}{1-4x}\; G_B(m;x) \; . \tag{92}$$

With this result (91) reduces to

$$\Big(-x\; c'(x) \;+\; (2-c(x))\; \frac{1-2x}{1-4x} \;-\; 1\Big)\; G_B(m;x) \;=\; 0 \; , \tag{93}$$

and with (9) the factor in front of $G_B(m;x)$ finally vanishes identically for $x \neq 1/4$. Therefore, we have proved the following two propositions.

Proposition 9: Column sequences of the $A(n,m)$ triangular array

The triangular array of numbers $A(n,m)$, defined for $n,m \in \mathbf{N}_0$ by eq.(84), $A(n,0) = C_n,\; A(n,m) \equiv 0$ for $n < m$ has as mth column sequence $\{A(n,m)\}_{n=m}^{\infty}$ the convolution of *Catalan*'s sequence and the mth convolution of the shifted power sequence $\{0,1,4^1,4^2,...\}$.

Proof: (86) with (85). □

Proposition 10: Triangular $A(n,m)$ array

The triangular array $A(n,m)$ of *proposition 9* coincides with the one defined by (83) and (78), (79) and (80) with $B(n,m)$ given by (68).

Proof: $a(n,0) = A(n+1,1)$ and $a(n,n) = A(n+1,n+1) \equiv 1$ of (78) and (79), *i.e.* (81) and (82), respectively, coincide with (84). (80) is rewritten with the aid of (83) as (88), and (88) has been proved by (89) to (93). □

It remains to find the explicit expression for the $a_n(x)$ coefficients $a(n,k)$ defined by (59). Because of (83) we try to find a formula for $A(n,m)$. By *propositions 9* and *10* we may consider the recursion (84) with inputs $A(n,0) = C_n$, $A(n,m) \equiv 0$ for $n < m$, and $A(n,n) \equiv 1$ from (83) and (82).

Proposition 11: Explicit form of $a_n(x)$

$A(n,m)$ given by $A(n,0) = C_n$, $A(n,m) \equiv 0$ for $n < m$, and (14) is the solution to (84) with $A(n,n) \equiv 1$.

Therefore, $a_n(x)$ is given by (59) with $a(n,k) = A(n+1,k+1)$ from (14).

Proof: The first term of $A(n,m)$, $\frac{1}{2}\, 4^{n-m+1} \binom{n}{m-1}$, satisfies the recursion (84) because of the binomial identity $\binom{n}{m-1} = \binom{n-1}{m-1} + \binom{n-1}{m-2}$ (*Pascal*'s triangle). For the second term of $A(n,m)$ in (14) one has to prove

$$\binom{n}{m-1}\binom{2n}{n} = 4\binom{n-1}{m-1}\binom{2(n-1)}{n-1} + \binom{n-1}{m-2}\binom{2(n-1)}{n-1}\frac{2(2m-3)}{m-1} , \tag{94}$$

or after division by $\binom{2(n-1)}{n-1}$

$$\frac{2n-1}{n}\binom{n}{m-1} = 2\binom{n-1}{m-1} + \binom{n-1}{m-2}\frac{2m-3}{m-1} , \tag{95}$$

which reduces to the trivial identity $2n-1 = 2\,(n-m+1) + 2m - 3$.

Both terms together, *i.e.* (14), satisfy the input $A(n,n) \equiv 1$. □

Note 3: $A(n,m)$ was found originally after iteration in the form (with $n \geq m > 0$ and $(-1)!! := 1$)

$$A(n,m) = 2 \cdot 4^{n-m}\binom{n}{m-1} - \frac{\prod_{k=1}^{m}(2(n-m)+2k-1)}{(2m-3)!!}\, C_{n-m} . \tag{96}$$

$A(n,0) = C_n$. It is easy to establish equivalence with (14).

In the original derivation of the $A(n,m)$ formula (14) it turned out to be convenient to introduce a rectangular array of integers $\hat{A}(n,m)$ for $n, m \in \mathbf{N}_0$ as follows. $\hat{A}(0,0) := 1$, $\hat{A}(n,0) := -C_n$ for $n \in \mathbf{N}$, and for $m \in \mathbf{N}$ and $n \in \mathbf{N}_0$ $\hat{A}(n,m)$ is defined by (15), or equivalently, by (16). The $A(n,m)$ recursion (84) translates (with the help of the above mentioned *Pascal*-triangle identity) to

$$\hat{A}(n,m) = 4\,\hat{A}(n-1,m) + \hat{A}(n,m-1) . \tag{97}$$

This leads, after iteration and use of $\hat{A}(0,m) \equiv 1$ from (15) with $A(n,n) \equiv 1$, to

$$\hat{A}(n,m) = 4^n \sum_{k=0}^{n} \hat{A}(k,m-1)/4^k . \tag{98}$$

Thus, the following proposition holds.

Proposition 12: Column sequences of the $\hat{A}(n,m) \equiv C4(n,m)$ array

The mth column sequence of the $\hat{A}(n,m)$ array, $\{\hat{A}(n,m)\}_{n=0}^{\infty}$, is the convolution of the sequence $\{\hat{A}(n,0)\}_0^{\infty} = \{1,-1,-2,-5,...\}$, generated by $2-c(x)$, and the mth convolution of the power sequence $\{4^k\}_0^{\infty}$.

Proof: Iteration of (98) with the $\hat{A}(n,0)$ input. □

Corollary 4: Generating functions for columns of the $\hat{A}(n,m) \equiv C4(n,m)$ array

The ordinary generating function of the mth column sequence of the $\hat{A}(n,m)$ array (16) is for $m \in \mathbf{N}_0$ given by

$$G_{\hat{A}}(m;x) \ := \ \sum_{n=0}^{\infty} \hat{A}(n,m)\ x^n \ = \ (2-c(x))\ \Big(\frac{1}{1-4x}\Big)^m \ . \tag{99}$$

Proof: Proposition 12 written for generating functions. □

Because of the convolution of the (negative) *Catalan* sequence with powers of 4 we shall call this array $\hat{A}(n,m)$ also $C4(n,m)$. A part of it is shown in *TAB.3.* [18]

Finally, we derive identities by using, for $n \in \mathbf{N}_0$, eq.(17) for the *lhs.* of (11) and the results for a_{n-1} and b_n for the *rhs.*

Because there are no negative powers of x on the *lhs.* of (11), such powers have to vanish on the *rhs.* This leads to the first family of identities. Because $(1-4x)^{-n} \ = \ \sum_{k=0}^{\infty} \frac{(n)_k}{k!}\ 4^k\ x^k$, with *Pochhammer*'s symbol defined in footnote 8, this means that $[x^p]\ (a_{n-1}(x)+b_n(x)\ c(x))$, the coefficient proportional to x^p, has to vanish for $p=0,1,...,n-1,\ n \in \mathbf{N}$. This requirement reads

$$(-1)^{n-1-p}\ a(n-1,n-1-p)\ +\ \sum_{k=0}^{p}(-1)^{n-k}\ B(n,n-k)\ C_{p-k}\ \equiv 0\ . \tag{100}$$

The sum is restricted to $k \le p(<n)$ because no C_l number with negative index is found in $c(x)$. Inserting the known coefficients this produces identity $(D1)$ of (18).

Proposition 13: Identity $(D1)$ of (18)

For $n \in \mathbf{N}$ and $p \in \{0,1,...,n-1\}$ identity $(D1)$, given by (18), holds.

Proof: With (83) (100) becomes

$$\sum_{k=0}^{p}\ (-1)^{p-k}\ C_{p-k}\ B(n,n-k)\ =\ A(n,n-p)\ , \tag{101}$$

which is $(D1)$ of (18) if the summation index k is changed into $p-k$, and symmetry of the binomial coefficients is used. □.

[18] The second column sequence is given by $\hat{A}(n,1) \equiv C4(n,1) = \binom{2n+1}{n}$ and appears as nr.2848 in the book [10], or as $A001700$ in the on-line encyclopedia [10]. The sequence of the third column $\{\hat{A}(n,2) \equiv C4(n,2)\}_0^{\infty} \ = \ \{1,7,38,187,...\}$ is from (98) and (96) with (15) determined by $4^n\ \sum_{k=0}^{n}\binom{2k+1}{k}/4^k \ = \ (2n+3)\ (2n+1)\ C_n \ - \ 2^{2n+1}$, and is listed as $A000531$ in the mentioned on-line encyclopedia. There the fourth column sequence is now listed as $A029887$.

Example 8: $(D1)$ identity for $p = n-1 \in \mathbf{N}_0$ [19]

$$\sum_{k=0}^{n-1}(-1)^k \binom{n}{k+1} \frac{1}{2k+1} \;=\; 4^n/\binom{2n}{n} \;-\; 1 \;=\; 2A(n,1)/\binom{2n}{n} \quad . \tag{102}$$

The second family of identities, $(D2)$ of (19), results from comparing powers x^k with $k \in \mathbf{N}_0$ on both sides of eq.(11) after expansion of $(1-4x)^{-n}$ as given above in the text before eq. (100). Only the second term $b_n(x)\; c(x)$ contributes because $a_{n-1}(x)/x^n$ has only negative powers of x. Thus, with definition (17) one finds for $k \in \mathbf{N}_0$ and $n \in \mathbf{N}$,

$$C(n,k) \;=\; \sum_{l=0}^{k} \frac{(n)_l\; 4^l}{l!} \sum_{j=0}^{n}(-1)^{n-j}\; B(n,n-j)\; C_{n-j+k-l} \tag{103}$$

which is, after interchange of the summations and insertion of $B(n,n-j)$ from (12) the desired identity $(D2)$ if also the summation index j is changed to $n-q$.

Thus we have shown:

Proposition 14: Identity $(D2)$ of (19)

For $k \in \mathbf{N}_0$ and $n \in \mathbf{N}$ identity $(D2)$ of (19) with $C(n,k)$ defined by (17) holds.

Example 9: Identity $(D2)$ for $k=0$, $n \in \mathbf{N}$

$$\sum_{j=0}^{n}(-1)^j \binom{n+1}{j+1} \;\equiv 1 \;, \tag{104}$$

which is elementary.

Acknowledgements

The author likes to thank Dr. Stephen Bedding for a collaboration on power of matrices. In *section 2* a result for 2×2 matrices (here $\mathbf{C}$) was recovered.

[19] With this identity we have found a sum representation for the convolution of the *Catalan* sequence and powers of 4: $s_{n-1}(4) := 4^{n-1} \sum_{k=0}^{n-1} C_k/4^k \;=\; \frac{1}{2}\binom{2n}{n} \sum_{k=0}^{n-1}(-1)^k \binom{n}{k+1} \frac{1}{2k+1}$ (*cf.* (75) with (71)).

TAB. 1 : $B(n,m)$ Central Binomial Triangle

n\m	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0	0	0	0
2	6	6	1	0	0	0	0	0	0	0	0
3	20	30	10	1	0	0	0	0	0	0	0
4	70	140	70	14	1	0	0	0	0	0	0
5	252	630	420	126	18	1	0	0	0	0	0
6	924	2772	2310	924	198	22	1	0	0	0	0
7	3432	12012	12012	6006	1716	286	26	1	0	0	0
8	12870	51480	60060	36036	12870	2860	390	30	1	0	0
9	48620	218790	291720	204204	87516	24310	4420	510	34	1	0
10	184756	923780	1385670	1108536	554268	184756	41990	6460	646	38	1

TAB. 2 : $A(n,m)$ Catalan triangle

n\m	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0
2	2	5	1	0	0	0	0	0	0	0	0
3	5	22	9	1	0	0	0	0	0	0	0
4	14	93	58	13	1	0	0	0	0	0	0
5	42	386	325	110	17	1	0	0	0	0	0
6	132	1586	1686	765	178	21	1	0	0	0	0
7	429	6476	8330	4746	1477	262	25	1	0	0	0
8	1430	26333	39796	27314	10654	2525	362	29	1	0	0
9	4862	106762	185517	149052	69930	20754	3973	478	33	1	0
10	16796	431910	848830	781725	428772	152946	36646	5885	610	37	1

TAB. 3 : $C4(n,m)$ Catalan array

n\m	0	1	2	3	4	5	6
0	1	1	1	1	1	1	1
1	-1	3	7	11	15	19	23
2	-2	10	38	82	142	218	310
3	-5	35	187	515	1083	1955	3195
4	-14	126	874	2934	7266	15086	27866
5	-42	462	3958	15694	44758	105102	216566
6	-132	1716	17548	80324	259356	679764	1546028
7	-429	6435	76627	397923	1435347	4154403	10338515
8	-1430	24310	330818	1922510	7663898	24281510	65635570
9	-4862	92378	1415650	9105690	39761282	136887322	399429602
10	-16796	352716	6015316	42438076	201483204	749032492	2346750900

References

[1] M. Gardner: *"Time Travel And Other Mathematical Bewilderments"*, ch. Twenty, W.H. Freeman, New York, 1988

[2] R.L. Graham, D.E. Knuth, and O. Patashnik: *" Concrete Mathematics "*, Addison-Wesley, Reading MA, 1989

[3] P. Hilton and J. Pedersen: "Catalan Numbers, Their Generalization, and Their Uses ", The Mathematical Intelligencer 13 (1991) 64-75

[4] V.E. Hoggatt, Jr. and M. Bicknell: "Catalan and Related Sequences Arising from Inverses of Pascal's Triangle Matrices", The Fibonacci Quarterly 14 (1976) 395-405

[5] A.F. Horadam: "Special Properties of the Sequence $W_n(a,b;p,q)$", The Fibonacci Quarterly 5, 5 (1967) 424-434

[6] W. Lang: "On Sums of Powers of Zeros of Polynomials", Journal of Computational and Applied Mathematics 89 (1998) 237-256

[7] M. Petkovšek, H.S. Wilf, and D. Zeilberger: *" A=B "*, A K Peters, Wellesley, MA, 1996

[8] G. Pólya and G. Szegő: *"Aufgaben und Lehrsätze aus der Analysis I"*, Springer, Berlin, 1970, 4.ed.

[9] L.W. Shapiro: "A Catalan Triangle", Discrete Mathematics 14 (1976) 83-90

[10] N.J.A. Sloane and S. Plouffe: *"The Encyclopedia of Integer Sequences"*, Academic Press, San Diego, 1995; see also N.J.A. Sloane's On-Line Encyclopedia of Integer Sequences, http//:www.research.att.com/ ~njas/sequences/index.html

[11] R.P. Stanley: *"Enumerative Combinatorics"*, vol. II, tbp Cambridge University Press, excerpt 'Problems on Catalan and Related Numbers', available from http/www-math.mit-edu/ ~rstan/ec/ec.html

[12] Wen-Jin Woan, Lou Shapiro, and D.G. Rogers: "The Catalan Numbers, the Lebesgue Integral, and 4^{n-2}", American Mathematical Monthly 101 (1997) 926-931

AMS MSC numbers: 11B83, 11B37, 33C45

Beyond Mere Convergence

James A. Sellers

Department of Mathematics
The Pennsylvania State University
107 Whitmore Laboratory
University Park, PA 16802
sellersj@math.psu.edu

February 5, 2002 – REVISED

Abstract

In this article, I suggest that calculus instruction should include a wider variety of examples of convergent and divergent series than is usually demonstrated. In particular, a number of convergent series, such as $\sum_{k\geq 1} \frac{k^3}{2^k}$, are considered, and their exact values are found in a straightforward manner. We explore and utilize a number of mathematical topics, including manipulation of certain power series and recurrences.

During my most recent spring break, I read William Dunham's book *Euler: The Master of Us All* [3]. I was thoroughly intrigued by the material presented and am certainly glad I selected it as part of the week's reading.

Of special interest were Dunham's comments on series manipulations and the power series identities developed by Euler and his contemporaries, for I had just completed teaching convergence and divergence of infinite series in my calculus class. In particular, Dunham [3, p. 47-48] presents Euler's proof of the Basel Problem, a challenge from Jakob Bernoulli to determine the

exact value of the sum $\sum_{k\geq 1} \frac{1}{k^2}$. Euler was the first to solve this problem by proving that the sum equals $\frac{\pi^2}{6}$.

I was reminded of my students' interest in this result when I shared it with them just weeks before. I had already mentioned to them that exact values for relatively few families of convergent series could be determined. The obvious examples are geometric series $\sum_{k\geq 0} r^k$ (with $|r| < 1$) and telescoping series. I also remembered their disappointment when I observed that the exact numerical value of most convergent series cannot be determined in a straightforward way. I tried to excite them with the notion that the convergence or divergence of a given series could be determined via the Integral Test, Limit Comparison Test, Ratio or Root Test, but this was received with little enthusiasm.

But now I return to Dunham's book. In [3, p. 41], Dunham notes that Jakob Bernoulli [2, p. 248-249] proved

$$\sum_{k\geq 1} \frac{k^2}{2^k} = 6 \tag{1}$$

and

$$\sum_{k\geq 1} \frac{k^3}{2^k} = 26. \tag{2}$$

Many teachers of calculus will recognize at least two things about (1) and (2). First, these series are made-to-order examples to demonstrate convergence with the Ratio Test. Such examples, where the summands are defined by the ratio of a polynomial and an exponential function, can be found in a number of calculus texts, such as [4] and [5]. Second - a much more negative admission - is that we rarely teach students how to prove equalities like (1) and (2). We usually stop at demonstrating that such series converge, and move on to other matters. This is the case with the two calculus texts mentioned above, and it is an unfortunate situation to say the least.

I contend that students of first-year calculus would be better served if we provided a few more tools to them for finding **exact** values of convergent infinite series. Oddly enough, the series in (1) and (2) are ideal for such a task.

My goal in this note is to present two approaches to finding the exact value of

$$a(m,n) := \sum_{k\geq 1} \frac{k^n}{m^k}$$

with $|m| > 1$ and $n \in \mathbb{N} \cup \{0\}$ (of which Bernoulli's examples (1) and (2) are special cases).

We begin by noting that, for each $|m| > 1$, $\left|\frac{1}{m}\right| < 1$, so that $a(m,0)$ is a convergent geometric series. Moreover,

$$\begin{aligned} a(m,0) &= \sum_{k\geq 1} \frac{1}{m^k} \\ &= \frac{1}{m} + \sum_{k\geq 2} \left(\frac{1}{m}\right)^k \\ &= \frac{1}{m} + \frac{1}{m}\sum_{k\geq 1} \left(\frac{1}{m}\right)^k \\ &= \frac{1}{m} + \frac{1}{m} a(m,0). \end{aligned}$$

Solving for $a(m,0)$, we see that it equals $\frac{1}{m-1}$. Of course, this result easily follows from the usual formula for the sum of a convergent geometric series.

Next, we obtain a recurrence for $a(m,n),\ n \geq 1$, in terms of $a(m,j)$ for $j < n$. Note that

$$\begin{aligned} a(m,n) &= \sum_{k\geq 1} \frac{k^n}{m^k} \\ &= \frac{1}{m} + \sum_{k\geq 2} \frac{k^n}{m^k} \\ &= \frac{1}{m} + \frac{1}{m}\sum_{k\geq 1} \frac{(k+1)^n}{m^k} \\ &= \frac{1}{m}\left[1 + \sum_{k\geq 1} \frac{(k+1)^n}{m^k}\right]. \end{aligned}$$

The argument up to this point is exactly that used in finding the formula for $a(m,0)$ above. We now employ the binomial theorem, a tool that should be in the repertoire of first-year calculus students.

$$\begin{aligned} a(m,n) &= \frac{1}{m}\left[1+\sum_{k\geq 1}\frac{\left(\sum_{j=0}^{n}\binom{n}{j}k^j\right)}{m^k}\right] \\ &= \frac{1}{m}\left[1+\sum_{j=0}^{n}\binom{n}{j}\sum_{k\geq 1}\frac{k^j}{m^k}\right] \\ &= \frac{1}{m}\left[1+\sum_{j=0}^{n-1}\binom{n}{j}\sum_{k\geq 1}\frac{k^j}{m^k}+\sum_{k\geq 1}\frac{k^n}{m^k}\right] \\ &= \frac{1}{m}\left[1+\sum_{j=0}^{n-1}\binom{n}{j}a(m,j)+a(m,n)\right] \\ &= \frac{1}{m}a(m,n)+\frac{1}{m}\left[1+\sum_{j=0}^{n-1}\binom{n}{j}a(m,j)\right] \end{aligned}$$

Solving for $a(m,n)$ yields

$$\left(1-\frac{1}{m}\right)a(m,n)=\frac{1}{m}\left[1+\sum_{j=0}^{n-1}\binom{n}{j}a(m,j)\right]$$

or

$$a(m,n)=\left(\frac{1}{m-1}\right)\left[1+\sum_{j=0}^{n-1}\binom{n}{j}a(m,j)\right]. \tag{3}$$

As a sidenote, it is interesting to see from (3) that, for rational values of m, the numerical value of $a(m,n)$ must be rational for all $n \geq 0$. This can be proven via induction on n. We noted above that $a(m,0)=\frac{1}{m-1}$ which is rational as long as m is rational. Then, assuming $a(m,j)$ is rational for $0 \leq j \leq n-1$, (3) implies $a(m,n)$ is also rational. Hence, no values such as $\frac{\pi^2}{6}$ will arise as values for $a(m,n)$ whenever m is rational.

The recurrence in (3) can be used to calculate with relative ease the **exact** value of

$$a(m,n)=\sum_{k\geq 1}\frac{k^n}{m^k}$$

for all $|m| > 1$ and $n \in \mathbb{N} \cup \{0\}$. For example, since

$$a(2,0) = \sum_{k \geq 1} \frac{1}{2^k} = 1,$$

we have

$$\begin{aligned} a(2,1) &= \sum_{k \geq 1} \frac{k}{2^k} \\ &= \left(\frac{1}{2-1}\right)\left[1 + \binom{1}{0} a(2,0)\right] \\ &= 1 + 1 = 2, \end{aligned}$$

and

$$\begin{aligned} a(2,2) &= \sum_{k \geq 1} \frac{k^2}{2^k} \\ &= 1 + \binom{2}{0} a(2,0) + \binom{2}{1} a(2,1) \\ &= 1 + 1 + 2 \cdot 2 = 6, \end{aligned}$$

which is the result labeled (1). Finally,

$$\begin{aligned} a(2,3) &= \sum_{k \geq 1} \frac{k^3}{2^k} \\ &= 1 + \binom{3}{0} a(2,0) + \binom{3}{1} a(2,1) + \binom{3}{2} a(2,2) \\ &= 1 + 1 + 3 \cdot 2 + 3 \cdot 6 = 26, \end{aligned}$$

which is (2).

Of course, recurrence (3) could be used to calculate $a(m,n)$ for larger values of m and n. However, this might prove tedious for extremely large values of n. With this in mind, we now approach the calculation of $a(m,n)$ from a second point of view.

We begin with the familiar power series representation for the function $\dfrac{1}{1-x}$:

(4) $$\frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + \ldots, \text{ where } |x| < 1$$

Andrews [1] recently extolled the virtues of (4) in the study of calculus. Our goal in this section is to manipulate (4) via differentiation and multiplication to obtain a new power series of the form

$$f_n(x) := x + 2^n x^2 + 3^n x^3 + 4^n x^4 + \ldots = \sum_{k\geq 1} k^n x^k$$

for a fixed positive integer n. This is done by applying the $x\frac{d}{dx}$ operator to $\frac{1}{1-x}$ n times. Then $a(m,n)$ equals $f_n\left(\frac{1}{m}\right)$, which is easily computed once $f_n(x)$ is written as a rational function. (Note that we define $f_0(x)$ by $f_0(x) := x\left(\frac{1}{1-x}\right) = \sum_{k\geq 1} x^k$.)

As an example, we apply the $x\frac{d}{dx}$ operator to $\frac{1}{1-x}$ and get

$$x\frac{d}{dx}\left(\frac{1}{1-x}\right) = x\frac{d}{dx}(1 + x + x^2 + x^3 + x^4 + \ldots)$$

or

$$f_1(x) = \frac{x}{(1-x)^2} = x + 2x^2 + 3x^3 + 4x^4 + \ldots = \sum_{k\geq 1} kx^k.$$

Hence,

$$\sum_{k\geq 1} \frac{k}{2^k} = f_1\left(\frac{1}{2}\right) = \frac{\frac{1}{2}}{\left(1-\frac{1}{2}\right)^2} = 2.$$

We can apply the $x\frac{d}{dx}$ operator to $\frac{1}{1-x}$ twice to obtain $f_2(x)$:

$$\begin{aligned} f_2(x) &= x\frac{d}{dx}\left(x\frac{d}{dx}\left(\frac{1}{1-x}\right)\right) \\ &= x\frac{d}{dx}\left(\frac{x}{(1-x)^2}\right) \\ &= \frac{x^2+x}{(1-x)^3}. \end{aligned}$$

Thus,

$$f_2(x) = \frac{x^2+x}{(1-x)^3} = x + 2^2x^2 + 3^2x^3 + 4^2x^4 + \ldots = \sum_{k\geq 1} k^2x^k.$$

Hence,

$$\sum_{k\geq 1}\frac{k^2}{2^k} = f_2\left(\frac{1}{2}\right) = \frac{\frac{1}{2}+\left(\frac{1}{2}\right)^2}{\left(1-\frac{1}{2}\right)^3} = 6$$

upon simplification. This, as we have already seen, is (1).

Additional applications of the $x\frac{d}{dx}$ operator can be performed to yield

$$\begin{aligned}
f_1(x) &= \frac{x}{(1-x)^2} = \sum_{k\geq 1} kx^k,\\
f_2(x) &= \frac{x^2+x}{(1-x)^3} = \sum_{k\geq 1} k^2x^k,\\
f_3(x) &= \frac{x^3+4x^2+x}{(1-x)^4} = \sum_{k\geq 1} k^3x^k,\\
f_4(x) &= \frac{x^4+11x^3+11x^2+x}{(1-x)^5} = \sum_{k\geq 1} k^4x^k,\\
f_5(x) &= \frac{x^5+26x^4+66x^3+26x^2+x}{(1-x)^6} = \sum_{k\geq 1} k^5x^k, \text{ and}\\
f_6(x) &= \frac{x^6+57x^5+302x^4+302x^3+57x^2+x}{(1-x)^7} = \sum_{k\geq 1} k^6x^k.
\end{aligned}$$

We see that

$$f_n(x) = \frac{g_n(x)}{(1-x)^{n+1}}$$

for each $n \geq 1$ where $g_n(x)$ is a certain polynomial of degree n. Indeed, the functions $g_n(x)$ are well-known. Upon searching N.J.A. Sloane's On-Line Encyclopedia of Integer Sequences [6] for the sequence

$$1, 1, 1, 1, 4, 1, 1, 11, 11, 1, 1, 26, 66, 26, 1, \ldots,$$

which is the sequence of coefficients of the polynomials $g_n(x)$, we discover that these are the **Eulerian numbers** $e(n, j)$. They are defined, for each value of j and n satisfying $1 \leq j \leq n$, by

(5) $e(n, j) = je(n-1, j) + (n-j+1)e(n-1, j-1)$ with $e(1, 1) = 1$.

With this notation, it appears that, for $n \geq 1$,

$$f_n(x) = \frac{\sum_{j=1}^{n} e(n,j)x^j}{(1-x)^{n+1}}.$$

Using (5), this assertion can be proven in a straightforward manner via induction. Moreover, we know from [6, Sequence A008292] that

$$e(n,j) = \sum_{\ell=0}^{j} (-1)^\ell (j-\ell)^n \binom{n+1}{\ell}.$$

This can be used to write the rational version of $f_n(x)$ for any $n \geq 1$ in a timely way. So, for example, we see that

$$f_8(x) = \frac{x^8 + 247x^7 + 4293x^6 + 15619x^5 + 15619x^4 + 4293x^3 + 247x^2 + x}{(1-x)^9},$$

which implies

$$\sum_{k\geq 1} \frac{k^8}{5^k} = f_8\left(\frac{1}{5}\right) = \frac{1139685}{2048}.$$

We have thus seen two different ways to compute the exact value of $\sum_{k\geq 1} \frac{k^n}{m^k}$ with $|m| > 1$ and $n \in \mathbb{N} \cup \{0\}$, one with a recurrence and one with power series. I encourage us all to share at least one of these techniques with our students the next time we are exploring infinite series.

References

[1] G. Andrews, *The Geometric Series in Calculus*, American Mathematical Monthly **105**, no. 1 (1998), 36-40.

[2] J. Bernoulli, *Tractatus de seriebus infinitis*, 1689.

[3] W. Dunham, *Euler: The Master of Us All*, The Dolciani Mathematical Expositions, no. 22, Mathematical Association of America, Washington, D.C., 1999.

[4] C. Edwards and D. Penney, *Calculus with Analytic Geometry*, Fifth Edition, Prentice Hall, 1998.

[5] R. Larson, R. Hostetler, and B. Edwards, *Calculus: Early Transcendental Functions*, Second Edition, Houghton Mifflin Company, 1999.

[6] N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences, published electronically at http://www.research.att.com/∼njas/sequences/.

Keywords: infinite series, convergence, divergence, Euler, Bernoulli, ratio test, recurrence, binomial theorem, Eulerian numbers

Biographical Note: James A. Sellers is currently the Director of Undergraduate Mathematics at the Pennsylvania State University. Before accepting this position he served for nine years as a mathematics professor at Cedarville University in Ohio. As a mathematics professor, James loves to teach mathematics to undergraduates and perform research with them.

Prior to going to Cedarville he received his Ph.D. in mathematics in 1992 from Penn State, where he met his wife Mary. James truly enjoys spending time with Mary and their five children. He agrees with Euler that mathematics can often be enjoyed and discovered with a child in his arms or playing round his feet.

A Probabilistic View of Certain Weighted Fibonacci Sums

Arthur T. Benjamin
Dept. of Mathematics, Harvey Mudd College, Claremont, CA 91711
benjamin@hmc.edu

Judson D. Neer
Dept. of Science and Mathematics, Cedarville University, 251 N. Main St., Cedarville, OH 45314-0601
jud@poboxes.com

Daniel E. Otero
Dept. of Mathematics and Computer Science, Xavier University, Cincinnati, OH 45207-4441
otero@xu.edu

James A. Sellers
Dept. of Mathematics, Penn State University, University Park, PA 16802
sellersj@math.psu.edu

1 Introduction

In this paper we investigate sums of the form

$$a_n := \sum_{k \geq 1} \frac{k^n F_k}{2^{k+1}}. \tag{1}$$

For any given n, such a sum can be determined [3] by applying the $x\frac{d}{dx}$ operator n times to the generating function

$$G(x) := \sum_{k\geq 1} F_k x^k = \frac{x}{1-x-x^2},$$

then evaluating the resulting expression at $x = 1/2$. This leads to $a_0 = 1$, $a_1 = 5$, $a_2 = 47$, and so on. These sums may be used to determine the expected value and higher moments of the number of flips needed of a fair coin until two consecutive heads appear [3]. In this article, we pursue the reverse strategy of using probability to derive a_n and develop an exponential generating function for a_n in Section 3. In Section 4, we present a method for finding an exact, non-recursive, formula for a_n.

2 Probabilistic Interpretation

Consider an infinitely long binary sequence of independent random variables $b_1, b_2, b_3, \ldots$ where $P(b_i = 0) = P(b_i = 1) = 1/2$. Let Y denote the random variable denoting the beginning of the first 00 substring. That is, $b_Y = b_{Y+1} = 0$ and no 00 occurs before then. Thus $P(Y = 1) = 1/4$. For $k \geq 2$, we have $P(Y = k)$ is equal to the probability that our sequence begins $b_1, b_2, \ldots, b_{k-2}, 1, 0, 0$, where no 00 occurs among the first $k-2$ terms. Since the probability of occurence of each such string is $(1/2)^{k+1}$, and it is well known [1] that there are exactly F_k binary strings of length $k-2$ with no consecutive 0's, we have for $k \geq 1$,

$$P(Y = k) = \frac{F_k}{2^{k+1}}.$$

Since Y is finite with probability 1, it follows that

$$\sum_{k\geq 1} \frac{F_k}{2^{k+1}} = \sum_{k\geq 1} P(Y=k) = 1.$$

For $n \geq 0$, the expected value of Y^n is

$$a_n := E(Y^n) = \sum_{k\geq 1} \frac{k^n F_k}{2^{k+1}}. \tag{2}$$

Thus $a_0 = 1$. For $n \geq 1$, we use conditional expectation to find a recursive formula for a_n. We illustrate our argument with $n = 1$ and $n = 2$ before proceeding with the general case.

For a random sequence $b_1, b_2, \ldots$, we compute $E(Y)$ by conditioning on b_1 and b_2. If $b_1 = b_2 = 0$, then $Y = 1$. If $b_1 = 1$, then we have wasted a flip, and we are back to the drawing board; let Y' denote the number of remaining flips needed. If $b_1 = 0$ and $b_2 = 1$, then we have wasted two flips, and we are back to the drawing board; let Y'' denote the number of remaining flips needed in this case. Now by conditional expectation we have

$$\begin{aligned} E(Y) &= \frac{1}{4}(1) + \frac{1}{2}E(1+Y') + \frac{1}{4}E(2+Y'') \\ &= \frac{1}{4} + \frac{1}{2} + \frac{1}{2}E(Y') + \frac{1}{2} + \frac{1}{4}E(Y'') \\ &= \frac{5}{4} + \frac{3}{4}E(Y) \end{aligned}$$

since $E(Y') = E(Y'') = E(Y)$. Solving for $E(Y)$ gives us $E(Y) = 5$. Hence,

$$a_1 = \sum_{k\geq 1} \frac{kF_k}{2^{k+1}} = 5.$$

Conditioning on the first two outcomes again allows us to compute

$$\begin{aligned} E(Y^2) &= \frac{1}{4}(1^2) + \frac{1}{2}E\left[(1+Y')^2\right] + \frac{1}{4}E\left[(2+Y'')^2\right] \\ &= \frac{1}{4} + \frac{1}{2}E(1+2Y+Y^2) + \frac{1}{4}E(4+4Y+Y^2) \\ &= \frac{7}{4} + 2E(Y) + \frac{3}{4}E(Y^2). \end{aligned}$$

Since $E(Y) = 5$, it follows that $E(Y^2) = 47$. Thus,

$$a_2 = \sum_{k\geq 1} \frac{k^2 F_k}{2^{k+1}} = 47.$$

Following the same logic for higher moments, we derive for $n \geq 1$,

$$\begin{aligned} E(Y^n) &= \frac{1}{4}(1^n) + \frac{1}{2}E[(1+Y)^n] + \frac{1}{4}E[(2+Y)^n] \\ &= \frac{1}{4} + \frac{3}{4}E(Y^n) + \frac{1}{2}\sum_{k=0}^{n-1}\binom{n}{k}E(Y^k) + \frac{1}{4}\sum_{k=0}^{n-1}\binom{n}{k}2^{n-k}E(Y^k). \end{aligned}$$

Consequently, we have the following recursive equation:

$$E(Y^n) = 1 + \sum_{k=0}^{n-1}\binom{n}{k}[2+2^{n-k}]E(Y^k)$$

Thus for all $n \geq 1$,

$$a_n = 1 + \sum_{k=0}^{n-1}\binom{n}{k}[2+2^{n-k}]a_k. \tag{3}$$

Using equation (3), one can easily derive $a_3 = 665$, $a_4 = 12,551$, and so on.

3 Generating Function and Asymptotics

For $n \geq 0$, define the exponential generating function

$$a(x) = \sum_{n\geq 0} \frac{a_n}{n!}x^n.$$

It follows from equation (3) that

$$\begin{aligned} a(x) &= 1 + \sum_{n\geq 1} \frac{\left(1 + \sum_{k=0}^{n-1} \binom{n}{k}[2 + 2^{n-k}]a_k\right)}{n!} x^n \\ &= e^x + 2a(x)(e^x - 1) + a(x)(e^{2x} - 1). \end{aligned}$$

Consequently,

$$a(x) = \frac{e^x}{4 - 2e^x - e^{2x}}. \tag{4}$$

For the asymptotic growth of a_n, one need only look at the leading term of the Laurent series expansion [4] of $a(x)$. This leads to

$$a_n \approx \frac{\sqrt{5} - 1}{10 - 2\sqrt{5}} \left(\frac{1}{\ln(\sqrt{5} - 1)}\right)^{n+1} n!. \tag{5}$$

4 Closed Form

While the recurrence (3), generating function (4), and asymptotic result (5) are satisfying, a closed form for a_n might also be desired. For the sake of completeness, we demonstrate such a closed form here.

To calculate

$$a_n = \sum_{k\geq 1} \frac{k^n F_k}{2^{k+1}},$$

we first recall the Binet formula for F_k [3]:

$$F_k = \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2}\right)^k - \left(\frac{1 - \sqrt{5}}{2}\right)^k\right) \tag{6}$$

Then (6) implies that (1) can be rewritten as

$$a_n = \frac{1}{2\sqrt{5}} \sum_{k\geq 1} k^n \left(\frac{1 + \sqrt{5}}{4}\right)^k - \frac{1}{2\sqrt{5}} \sum_{k\geq 1} k^n \left(\frac{1 - \sqrt{5}}{4}\right)^k. \tag{7}$$

Next, we remember the formula for the geometric series:

$$\sum_{k\geq 0} x^k = \frac{1}{1-x} \tag{8}$$

This holds for all real numbers x such that $|x| < 1$. We now apply the $x\frac{d}{dx}$ operator n times to (8). It is clear that the left-hand side of (8) will then become

$$\sum_{k\geq 1} k^n x^k.$$

The right-hand side of (8) is transformed into the rational function

$$\frac{1}{(1-x)^{n+1}} \times \sum_{j=1}^{n} e(n,j)x^j, \tag{9}$$

where the coefficients $e(n,j)$ are the Eulerian numbers [2, Sequence A008292], defined by

$$e(n,j) = j \cdot e(n-1,j) + (n-j+1) \cdot e(n-1,j-1) \text{ with } e(1,1) = 1.$$

(The fact that these are indeed the coefficients of the polynomial in the numerator of (9) can be proven quickly by induction.) From the information found in [2, Sequence A008292], we know

$$e(n,j) = \sum_{\ell=0}^{j} (-1)^\ell (j-\ell)^n \tbinom{n+1}{\ell}.$$

Therefore,

$$\sum_{k\geq 1} k^n x^k = \frac{1}{(1-x)^{n+1}} \times \sum_{j=1}^{n} \left[\sum_{\ell=0}^{j} (-1)^\ell (j-\ell)^n \tbinom{n+1}{\ell} \right] x^j. \tag{10}$$

Thus the two sums

$$\sum_{k\geq 1} k^n \left(\frac{1+\sqrt{5}}{4}\right)^k \quad \text{and} \quad \sum_{k\geq 1} k^n \left(\frac{1-\sqrt{5}}{4}\right)^k$$

that appear in (7) can be determined explicitly using (10) since

$$\left|\frac{1+\sqrt{5}}{4}\right| < 1 \quad \text{and} \quad \left|\frac{1-\sqrt{5}}{4}\right| < 1.$$

Hence, an exact, non-recursive, formula for a_n can be developed.

References

[1] A. T. Benjamin and J. J. Quinn, Recounting Fibonacci and Lucas Identities, *College Mathematics Journal*, Vol. 30, No. 5, pp. 359-366, 1999.

[2] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, published electronically at http://www.research.att.com/∼njas/sequences/, 2000.

[3] S. Vajda, *Fibonacci and Lucas Numbers, and the Golden Section*, John Wiley and Sons, New York, 1989.

[4] H. S. Wilf, *Generatingfunctionology*, Academic Press, Boston, 1994.

AMS Subject Classification Number: 11B39.

Unspecified Journal
Volume 00, Number 0, Xxxx XXXX, Pages 000–000
S ????-????(XX)0000-0

POLYDIAGONAL COMPACTIFICATION OF CONFIGURATION SPACES

ALEXANDER P. ULYANOV

Abstract. A smooth compactification $X\langle n\rangle$ of the configuration space of n distinct labeled points in a smooth algebraic variety X is constructed by a natural sequence of blowups, with the full symmetry of the permutation group $\mathbb{S}_n$ manifest at each stage. The strata of the normal crossing divisor at infinity are labeled by *leveled* trees and their structure is studied. This is the maximal wonderful compactification in the sense of De Concini–Procesi, and it has a strata-compatible surjection onto the Fulton–MacPherson compactification. The degenerate configurations added in the compactification are geometrically described by *polyscreens* similar to the screens of Fulton and MacPherson.

In characteristic 0, isotropy subgroups of the action of $\mathbb{S}_n$ on $X\langle n\rangle$ are abelian, thus $X\langle n\rangle$ may be a step toward an explicit resolution of singularities of the symmetric products $X^n/\mathbb{S}_n$.

Introduction

The configuration space $\mathrm{F}(X,n)$ of n distinct labeled points in a topological space X is the complement in the Cartesian product X^n of the union of the large diagonals $\Delta^{ij} = \{(x_1,\dots,x_n)\,|\,x_i = x_j\}$. Pioneering studies of these spaces by Fadell, Neuwirth, Arnold and Cohen [Ar, C, Fa, FaN] evolved into a still active area of algebraic topology; Totaro opens his paper with a brief review [Tot]. Somewhat later, a compactification of $\mathrm{F}(\mathbb{C},n)$ modulo affine automorphisms, known as the Grothendieck–Knudsen moduli space of stable n-pointed curves of genus 0, rose to prominence in modern algebraic geometry [De2, Ka1, Ke, Kn].

Then Fulton and MacPherson devised a powerful construction that works for any nonsingular algebraic variety and produces a compactification $X[n]$ of $\mathrm{F}(X,n)$ with a remarkable combination of properties [FM]:

- ▷ $X[n]$ is nonsingular.
- ▷ $X[n]$ naturally comes equipped with a proper map onto X^n.
- ▷ $X[n]$ is symmetric: it carries an action of the symmetric group $\mathbb{S}_n$ by permuting the labels.

Received by the editors December 16, 1999.

1991 *Mathematics Subject Classification.* Primary 14C99, 14M99, secondary 05A18, 14E15, 14M25.

Research partially supported by NSF grant DMS–9803593.

- ▷ The complement $D = X[n] \smallsetminus \mathrm{F}(X, n)$ is a normal crossing divisor.
- ▷ The combinatorial structure of D and of the resulting stratification of $X[n]$ is explicitly described: the components of D correspond to the subsets of $[n] = \{1, \dots, n\}$ with at least 2 elements; their intersections, the strata, correspond to nested collections of such subsets, and the latter are just a reincarnation of rooted trees with n marked leaves.
- ▷ Degenerate configurations have simple geometric descriptions.

Further results of Fulton and MacPherson include: a functorial description of $X[n]$, used to prove many of its properties listed above; a fact that all isotropy subgroups of $\mathbb{S}_n$ acting on $X[n]$ are solvable; some intersection theory, namely, a presentation of the intersection rings of $X[n]$ and of its strata, and, as an application, a computation of the rational cohomology ring of $\mathrm{F}(X, n)$ for X a smooth compact complex variety.

About the same time, constructions related to the Fulton–MacPherson compactification appeared, all motivated by, and suited to, some problems of mathematical physics: for real manifolds [AS, Ko]; for complex curves [BG], with later extension to higher dimensions [Gi].

The compactifications $X[n]$ are defined inductively, with the step from $X[n]$ to $X[n+1]$ performed by a sequence of blowups

$$X[n+1] = Y_n \xrightarrow{\alpha_{n-1}} Y_{n-1} \xrightarrow{\alpha_{n-2}} \cdots \xrightarrow{\alpha_1} Y_1 \xrightarrow{\alpha_0} Y_0 = X[n] \times X,$$

where the center of the blowup α_k is a disjoint union of subvarieties in Y_k corresponding in a specified way to the subsets of $[n]$ of cardinality $n - k$. Thus, the symmetry of $\mathbb{S}_{n+1}$ is not present at the intermediate stages. An alternative, and completely symmetric, description of $X[n]$ as the closure of $\mathrm{F}(X, n)$ in a product of blowups does not provide much insight into the structure of $X[n]$, so the inductive sequence of blowups is essential for that.

Fulton and MacPherson remark:

> It would be interesting to see if other sequences of blowups give compactifications that are symmetric, and whose points have explicit and concise descriptions [FM, bottom of p. 196].

An example of such a compactification, for any nonsingular algebraic variety X, is studied in the present paper. I denote it by $X\langle n\rangle$ and call it a *polydiagonal compactification*, because the blowup loci are not only the diagonals of X^n, but also their *intersections*. The idea is very simple: one who tries to blow up all diagonals of the same dimension simultaneously is forced to blow up all their intersections prior to that, and this prescribes the sequence. Following Fulton and MacPherson's terminology, $X\langle n\rangle$ is a compactification even though it is only compact when X itself is compact. In general, it is equipped with a canonical proper map onto X^n.

This construction applies also to real manifolds, with real blowups replacing algebraic blowups. The compactification is then a manifold with corners, and the results about the strata presented here can be rephrased to describe the combinatorics of its boundary.

The construction of $X\langle n\rangle$ is in some respects similar to that of $X[n]$, with one important difference: the former is completely symmetric *at each stage*. This reduces logical complexity of the construction even though it involves (considerably) more blowups. From this last fact stems another feature of $X\langle n\rangle$: it distinguishes some collisions that are treated as equal by Fulton and MacPherson. There is a surjection $\vartheta_n\colon X\langle n\rangle \to X[n]$ that essentially retreats from making these distinctions, and it is completely symmetric as well. Regardless of X, this map, derived from a description of $X\langle n\rangle$ as the closure of $\mathrm{F}(X,n)$ in a product of blowups, is an isomorphism for $n \leqslant 3$ only, and an iterated blowup otherwise. The fibers of ϑ_n have purely combinatorial nature and do not depend even on the dimension of X; their detailed description will appear in a separate paper [U].

Geometrically, the limiting configurations in the Fulton–MacPherson compactification are viewed in terms of tree-like successions of *screens*, each of which is a tangent space to X with several labeled points in it, considered modulo translations and dilations. In a similar visualization for points in $X\langle n\rangle$, labels of a new kind, necessary because $X\langle n\rangle$ has 'more' points than $X[n]$, augment the screens. This rests on a study of the strata: they are bundles over $X\langle r\rangle$, $r < n$, with fibers decomposable into products of certain projective varieties. Named *bricks*, they form a family indexed by integer partitions that includes, for example, permutahedral varieties. The latter in fact show up in each brick as constituents that account for those new labels.

As for the combinatorics underlying $X\langle n\rangle$, here the place of subsets of $[n]$, nested collections of such subsets, and plain rooted trees is taken by partitions of the set $[n]$, chains of such partitions, and rooted trees whose vertices are assigned integer numbers, called *levels*. With these changes, the natural stratification of $X\langle n\rangle$ is quite similar to that of $X[n]$; moreover, ϑ_n is a strata-compatible map corresponding to the forgetful map from leveled trees to usual rooted trees.

Analogues for $X\langle n\rangle$ of most results of Fulton and MacPherson follow purely geometrically. Since the proofs do not require a functorial description of the space, it is omitted.

The action of the symmetric group $\mathbb{S}_n$ on X^n by permuting the labels has fixed points. Fulton and MacPherson showed that the isotropy subgroups of the label permutation action of $\mathbb{S}_n$ on $X[n]$ are solvable [FM, Theorem 5]. It turns out that in characteristic 0 the similar action of $\mathbb{S}_n$ on $X\langle n\rangle$ has only abelian isotropy subgroups; thus, singularities of $X\langle n\rangle/\mathbb{S}_n$ can in principle be resolved by toric methods [AMRT, Br, KKMS, O]. The resulting space will provide an explicit desingularization of the symmetric product $X^n/\mathbb{S}_n$, as well as a smooth compactification of $\mathrm{B}(X,n) = \mathrm{F}(X,n)/\mathbb{S}_n$, the configuration space of n *unlabeled* points in X.

De Concini and Procesi developed a general approach to compactifying complements of linear subspace arrangements by iterated blowups [DP]. For each arrangement, it yields a family of *wonderful* blowups with minimal and

maximal elements. Although they work with linear subspaces, their technique is local and can be applied to $X^n \smallsetminus \mathrm{F}(X,n)$ for any smooth variety X; in this case, the Fulton–MacPherson compactification is the minimal one, while the polydiagonal compactification is the maximal one. Along the lines of De Concini, MacPherson and Procesi [MP], Yi Hu has extended many results presented here in Sections 4, 5 and 6 to blowups of arrangements of smooth subvarieties and then recovered Kirwan's partial desingularization of geometric invariant theory quotients [Hu, Ki].

In addition, Hu computed the intersection rings in that general context of arrangements. In the case of $X\langle n\rangle$ these rings may be used to build a differential graded algebra model of $\mathrm{F}(X,n)$ for X a compact complex algebraic manifold, as Fulton and MacPherson did. After that, Kriz streamlined their differential graded algebra, while Totaro extracted a presentation of the cohomology ring of the configuration space from the Leray spectral sequence of its embedding into its 'naive' compactification X^n [Kr, Tot].

Historical note. (Communicated by W. Fulton.) Fulton and MacPherson sought to build the space whose points would be described by screens; early attempts led them to consider the spaces denoted here by $X\langle 4\rangle$ and $X\langle 5\rangle$, and to identify what to blow down to create the desired $X[4]$ and $X[5]$. Seeing that as n grows, the blowdown description quickly becomes unwieldy, they chose not to pursue this in general and finally settled on a non-symmetric procedure. D. Thurston pointed out a symmetric construction of $X[n]$ and used its real analogue in his work on knot invariants [Th].

Standing assumptions. Throughout the paper, X is a smooth irreducible m-dimensional ($m > 0$) algebraic variety over some field $\Bbbk$, and n is the number of labeled points in X. The section on Hodge polynomials applies only to complex varieties, and that on the symmetric group action, only to the characteristic 0 case.

Outline of the paper. The first section is informal and serves to introduce the basic ideas of the polydiagonal compactification on the simplest example. A combinatorial interlude of Section 2 is followed by a discussion of polyscreens and color screens that represent points in $X\langle n\rangle$.

Formally stated and proved results begin in Section 4 that contains: construction of $X\langle n\rangle$ by a symmetric sequence of blowups, a description of the combinatorics of the complement $X\langle n\rangle \smallsetminus \mathrm{F}(X,n)$ as a divisor with normal crossings and of the ensuing stratification of $X\langle n\rangle$, and a recurrent formula for the number of the strata. If X is a complex variety, the blowup construction translates into a formula for the (virtual) Hodge polynomial $e(X\langle n\rangle)$ in terms of $e(X)$ derived in the next section. In Section 6, a consideration of $X\langle n\rangle$ as the closure of $\mathrm{F}(X,n)$ in a product of blowups implies a surjection $\vartheta_n\colon X\langle n\rangle \to X[n]$, written then as an iterated blowup. Technical analysis of the strata of $X\langle n\rangle$ occupies Section 7, and the last section deals with the isotropy subgroups of $\mathbb{S}_n$ acting on $X\langle n\rangle$.

Acknowledgements. In many ways, I am indebted to Jean–Luc Brylinski, my Ph.D. advisor. I am most grateful to William Fulton and Jim Stasheff for sharing their advice and for many valuable comments and insights. The referee's suggestions helped me improve exposition. I would also like to thank Dmitry Tamarkin for useful discussions.

1. Small numbers of colliding points

The purpose of this section is to introduce the main ideas of the paper by looking at the case of 4 points—the smallest integer n for which $X\langle n\rangle$ is different from $X[n]$ is 4.

To begin with, consider an example of two collisions of four points in $X = \mathbb{C}^2$. The corresponding two limiting configurations arising in the approach of Fulton and MacPherson coincide; however, the polydiagonal compactification will distinguish them. Take four points labeled by 1 through 4 and make them collide as $t \to 0$ in the following way:

- ⋄ the distance between 1 and 2 is $O(t^3)$,
- ⋄ the distance between 3 and 4 is $O(t^2)$,
- ⋄ the distance between the two pairs (12) and (34) is $O(t)$.

Then do the same thing, except for a small exchange:

- ⋄ the distance between 1 and 2 is $O(t^2)$,
- ⋄ the distance between 3 and 4 is $O(t^3)$,

and call the two limiting points $\mathbf{x}_1$ and $\mathbf{x}_2$.

Both limiting points lie in the same stratum of $X[4]$, the intersection of three divisors $D(1234)$, $D(12)$, and $D(34)$. The dimension of this stratum is 5; the dimension of its fiber over a point in the small diagonal $\Delta \subset X^4$ is 3. The three parameters record the 'directions' of collisions encoded by the middle tree in Figure 1. Specifying these directions for the two approach curves, that is, vectors hidden behind the symbol O, one can arrange that $\mathbf{x}_1 = \mathbf{x}_2$ in $X[4]$.

These approach curves actually belong to a whole family $\mathcal{F}$ of curves in $\mathrm{F}(X, 4)$ whose limits in $X[4]$ may coincide. Indeed, consider the diagonals Δ^{12} and Δ^{34} in X^4, and their intersection $\Delta^{12|34}$. Both curves approach this intersection, but the first one does it while having a 3rd degree osculation to Δ^{12}, and the second one does the same with Δ^{34}. The projectivized normal space $\mathbb{P}(T_pX^4/T_p\Delta^{12|34})$ parametrizes the family, and the two curves above correspond to normal directions going along Δ^{12} and Δ^{34} respectively. This suggests looking into a possibility of involving blowups of subvarieties like $\Delta^{12} \cap \Delta^{34}$, if the objective is to obtain a compactification that would distinguish from one another collisions produced by curves in such families.

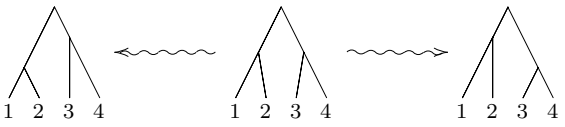


Figure 1

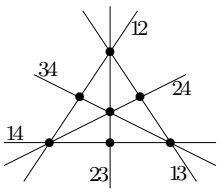


FIGURE 2

The space that achieves this results from implementing a simple idea of blowing up 'from the bottom to the top'. Although the dominant feature of the general case first comes to light when $n = 4$, it may be useful to begin with the cases of two and three colliding points.

Assume that $\dim X > 1$. There is no ambiguity about the case of $n = 2$ points: the compactification is the blowup of the diagonal in X^2. If $n = 3$, blowing up the small diagonal $\Delta \subset X^3$ creates disjoint proper transforms of Δ^{12}, Δ^{13} and Δ^{23} that can then be blown up in any order. The resulting compactification coincides with $X[3]$. For $n > 3$, however, this strategy will not work, and some additional blowups are needed [FM, bottom of p. 196], but what they are Fulton and MacPherson do not specify.

The left graph in Figure 3 shows the diagonals in X^4, including the space itself, as vertices, and (non-refinable) inclusions of the diagonals into each other as edges. As before, blow up the small diagonal first, then blow up the (disjoint) proper transforms of the four larger diagonals, like Δ^{123}. Now try to blow up the next level below them simultaneously. It does not work: these six largest diagonals have not been made disjoint. How can this be fixed?

The six lines intersecting at seven points depicted in Figure 2 are the images of the large diagonals of $\mathbb{R}^4$ in the real projective plane $\mathbb{P}(\mathbb{R}^4/\Delta)$, where Δ is the small diagonal. Four of the points correspond to diagonals like Δ^{123}, and the other three, where the intersections are normal, represent additional loci that need to be blown up to make the large diagonals disjoint. The second graph in Figure 3 is obtained from the first one by adding these three intersections $\Delta^{12} \cap \Delta^{34}$, $\Delta^{13} \cap \Delta^{24}$ and $\Delta^{14} \cap \Delta^{23}$. All seven vertices in the second row correspond to subvarieties pairwise disjoint after the blowup of the small diagonal $\Delta \subset X^4$, so they can be blown up simultaneously, and—crucially—after that the subvarieties from the row just below become disjoint and can be blown up simultaneously. This gives a compactification $X\langle 4 \rangle$ of $\mathrm{F}(X, 4)$.

FIGURE 3. Diagonals (left) and polydiagonals (right) in X^4

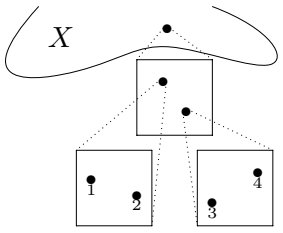


FIGURE 4. A point in $X[4]$

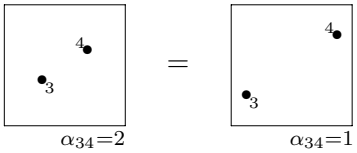


FIGURE 5. Dilation

The construction of $X\langle 4\rangle$ involves three more blowups than that of $X[4]$, so the complement of $\mathrm{F}(X,4)$ in $X\langle 4\rangle$ has three additional components $D^{12|34}$, $D^{13|24}$ and $D^{14|23}$. Collisions belonging to the family $\mathcal{F}$ discussed above result in points in $Z = D^{1234} \cap D^{12|34}$. To accommodate these, as well as more complicated degenerations of the same nature that appear for $n > 4$, two new features are added to Fulton–MacPherson screens: the screens are grouped into **levels**, and the group on each level bears a new parameter living in a projective space.

Figure 4 illustrates the screen description of the limiting points in $X[4]$ of the family $\mathcal{F}$: its macroscopic part is a single point in X and its microscopic part consists of three screens, one for each of the subsets 1234, 12 and 34 of $\{1,2,3,4\}$. A screen is a tangent space T_pX with a configuration of points in it, considered up to dilations and translations. In particular, the last two screens, S_{12} and S_{34}, are completely independent of each other.

Pictures like the left one in Figure 6 will represent generic points of Z. It consists of three levels:

(0) one point in X,
(1) a screen for 1234 with two distinct points, and
(2) a pair of screens S_{12} and S_{34} together with their **scale factors** α_{12} and α_{34}, where the pair $[\alpha_{12}:\alpha_{34}]$ is considered as a point in $\mathbb{P}^1$.

The scale factors serve to compare the approach speeds of the pairs 12 and 34 by keeping track of independent dilations of their respective screens: for all non-zero scalars ϕ_i, the pairs (S_i, α_i) and $(\phi_i S_i, \alpha_i/\phi_i)$ are identified, where the screen in the second pair is the dilation of S_i by the factor of ϕ_i, as in Figure 5.

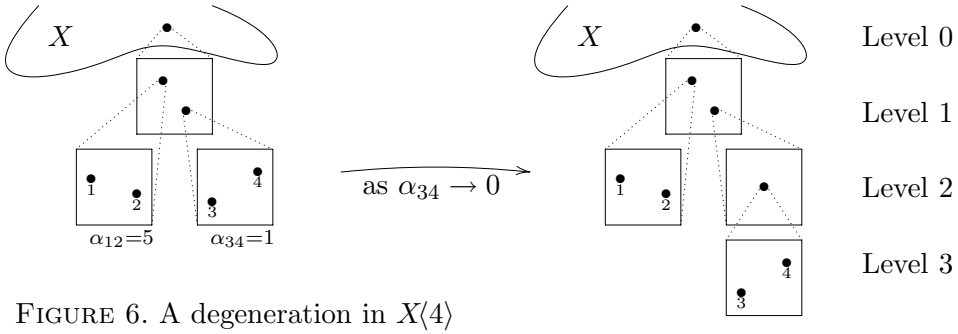


FIGURE 6. A degeneration in $X\langle 4\rangle$

Non-generic points of Z, which lie in $Z \cap D^{12}$ and $Z \cap D^{34}$, correspond to incomparable speeds and to the points $[0:1]$ and $[1:0]$ in $\mathbb{P}^1$. They result from collisions mentioned in the beginning of the section. Keeping the screen S_{34} fixed while letting $\alpha_{34} \to 0$ is the same thing as keeping α_{34} fixed while contracting the screen. In the limit the two points in it collide, but a new screen appearing on level 3 separates them. Trivial screens, which contain a single point, may be omitted from the pictures.

Similarly, points in $D^{12|34}$ away from D^{1234} are represented by configurations of two distinct points in X, labeled 12 and 34, plus screens S_{12} and S_{34} together with their scale factors, generically on the same level and degenerating to two levels.

The microscopic levels in Figure 6 correspond to the intersecting divisors: the first to D^{1234}, the second to $D^{12|34}$ and, in the right half of the figure, the third to D^{34}. Accordingly, trees that link screens together acquire some extra structure: levels of vertices. For example, the two pictures in Figure 6 correspond to the middle and right trees in Figure 1. Such trees index the strata of $X\langle 4\rangle$.

Scale factors are redundant on any level that contains only one nontrivial screen. Since the middle tree in Figure 1 is, up to relabeling, the only tree with four leaves in which two vertices may be on the same level, points in $X\langle 4\rangle$ outside the three additional divisors will have exactly the same screen description as for $X[4]$. In fact, forgetting the scale factors gives a map $\vartheta_4 \colon X\langle 4\rangle \to X[4]$ that blows down the divisor $D^{12|34}$ to the stratum $D(12) \cap D(34)$, and respectively for $D^{13|24}$ and $D^{14|23}$.

A combinatorial basis is necessary in order to generalize these ideas to an arbitrary number of points, and it is very easy to find. The definition of Δ^S for any subset S of $[n] = \{1, \dots, n\}$ applied to $S = \{k\}$ gives $\Delta^{\{k\}} = X^n$, hence $\Delta^{123} = \Delta^{123} \cap \Delta^4$ and so on. The true combinatorial basis will thus be *the partitions of the set* $[n]$. Indeed, when $n = 4$, the first blowup is that of $\Delta = \Delta^{1234}$, which corresponds to the only partition into one block; the next stage blowup centers correspond to all partitions into two blocks; finally, all those corresponding to partitions into three blocks are blown up: $\Delta^{12} = \Delta^{12} \cap \Delta^3 \cap \Delta^4$ and so on.

2. Combinatorial background

This section is a short primer on the language of the rest of the paper: it deals with basic properties of set partitions and a bijection between partition chains and leveled trees.

Let $[n]$ denote the set $\{1, \dots, n\}$ of integers. A partition π of $[n]$ is a set of disjoint subsets of $[n]$, called the blocks of π, whose union is $[n]$. Non-singleton blocks are called **essential**. The two functions of partitions that are most important for this work are $\rho(\pi)$, the number of blocks, and $\epsilon(\pi)$, the number of essential blocks. The integer partition whose parts are *one less than* the cardinalities of the essential blocks of π is called the **essential shape** of π and denoted by $\lambda(\pi)$. For example, $\pi_1 = \{12357, 9, 468\}$ and

$\pi_2 = \{15, 23, 7, 9, 468\}$ are two partitions of $[9]$ with

$$\rho(\pi_1) = 3, \qquad \epsilon(\pi_1) = 2, \qquad \lambda(\pi_1) = (4, 2),$$
$$\rho(\pi_2) = 5, \qquad \epsilon(\pi_2) = 3, \qquad \lambda(\pi_2) = (2, 1, 1).$$

Let $L_{[n]}$ be the set of all partitions of $[n]$. There is a refinement partial order on $L_{[n]}$: $\pi_1 \leqslant \pi_2$ whenever each block of π_2 is contained in a block of π_1, as in the example. This makes $L_{[n]}$ a ranked lattice, with $\rho(\pi)$ being the rank function. The minimal (bottom) and maximal (top) elements of $L_{[n]}$ are denoted by $\bot$ and $\top$ respectively.

The Stirling number of the second kind $S(n, k)$ is the number of partitions of $[n]$ into exactly k blocks. Many textbooks on combinatorics discuss these numbers and the partition lattice, for instance, Andrews [An] and Stanley [Sta1].

An interval $[\pi', \pi'']$ in a lattice L is its subset $\{\pi \,|\, \pi' \leqslant \pi \leqslant \pi''\}$. In $L_{[n]}$, every lower interval $[\bot, \pi]$ is isomorphic to $L_{[\rho(\pi)]}$ and every upper interval $[\pi, \top]$ is isomorphic to $L_{[\nu_1+1]} \times \cdots \times L_{[\nu_r+1]}$, where $\lambda = \lambda(\pi) = (\nu_1, \ldots, \nu_r)$ is the essential shape of π. This product will be denoted by L_λ.

A totally ordered subset of a partially ordered set is called a chain. The length of a chain is the number of its elements. Half of the chains in $L_{[n]}$ contain the top (finest) partition, and the other half do not; from now on, a chain will mean a partition chain of the latter kind.

Lengyel represented [Le] partition chains as trees. If $\gamma = \{\pi_1, \ldots, \pi_k\}$, where $\pi_i < \pi_{i+1}$ for $1 \leqslant i \leqslant k$, then the associated tree has the blocks of each partition as its interior vertices, one additional vertex (the root) and leaves labeled by $1, \ldots, n$. Edges indicate inclusions of blocks of π_{i+1} into those of π_i and of the elements of $[n]$ into the blocks of π_k; they also connect the blocks of π_1 to the root. The left tree in Figure 7 goes with the chain $\gamma = \{\pi_1, \pi_2, \pi_3\}$, where

$$\pi_1 = \{12357, 9, 468\}, \quad \pi_2 = \{15, 23, 7, 9, 468\}, \quad \pi_3 = \{1, 5, 23, 7, 9, 46, 8\}.$$

The 2-valent vertices (except for the root if it happens to be such) may be called the **phantom vertices** because it is often convenient to omit them; this gives trees like the middle one in the same figure. Furthermore, labels of interior vertices are also unnecessary. In thus simplified tree the set of interior vertices is the set $\{12357, 468, 23, 46, 15\}$ of all essential blocks in the three partitions, and they appear to be on different *levels* reflecting how far in the chain they survive unsubdivided. This leads to the following

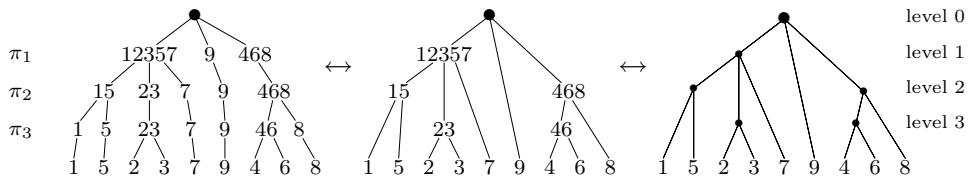


FIGURE 7. From a partition chain to a leveled tree (and back)

Definition. A k-**leveled tree** is a pair (T, η), where T is a rooted tree without 2-valent vertices, except possibly for the root, and η is a surjective poset map from the set of vertices of T with the parent-descendant partial order to the set of integers $\{0, \dots, k\}$ with its standard order. (The root goes to 0.) The number $\eta(v)$ is called the **level** of the vertex v. The map from leveled trees with marked leaves to usual rooted trees with marked leaves by $(T, \eta) \mapsto T$ is denoted by θ.

The term *leveled tree* belongs to Loday, although his trees are binary [Lo]. An inspiring picture evinces that Tonks used leveled trees implicitly [Ton]. In both references the leaves are not marked. The sole purpose of the root is to simplify wording: without it, we would be dealing not only with trees, but also with groves (disjoint unions of trees).

The example above demonstrates how to pass from a k-chain γ of partitions of $[n]$ to a k-leveled tree (T_γ, η_γ) with n marked leaves; this is actually a bijection when restricted to such chains γ that $\top \notin \gamma$. There is a unique (shortest) path from the root of (T, η) to each leaf, and each pair of such separate at a vertex on certain level j. The labels of the two leaves will be in the same block in the partitions π_i for $i \leqslant j$, and they will be in different blocks in π_i for $i > j$. This defines the k-chain $\gamma(T, \eta)$.

It will also be useful to associate with a k-leveled tree (T, η) a sequence $\{\lambda_i(T, \eta)\} = \{\lambda_i(\gamma)\}$ of integer partitions as follows. While λ_0 has just one part, equal to the valency of the root of T, the partition λ_i, $1 \leqslant i \leqslant k$, is to have as many parts as there are vertices of (T, η) on level i, and each part is to be one less than the number of direct descendants of the corresponding vertex. With that, $\rho(\pi_1) = \lambda_0(\gamma)$ and $[\pi_i, \pi_{i+1}] \simeq L_{\lambda_i(\gamma)}$ for $1 \leqslant i \leqslant k$, where $\gamma = \gamma(T, \eta) = \{\pi_1, \dots, \pi_k\}$ and $\pi_{k+1} = \top$. For the example above, $\lambda_0 = (3)$, $\lambda_1 = (2)$, $\lambda_2 = (1, 1)$ and $\lambda_3 = (1, 1)$.

3. Polyscreens and color screens

Partition chains and leveled trees of the previous section play in the poly-diagonal compactification $X\langle n\rangle$ the same role as nests of subsets of $[n]$ and usual trees (groves) do in the Fulton–MacPherson compactification $X[n]$. They index the strata and are an integral part of the geometric description of points in $X\langle n\rangle$, explained in this section without any proofs. It is implied by the technical work of Section 7.

For a chain $\gamma = \{\pi_1, \dots, \pi_k\}$, each point $\mathbf{x}$ in the stratum S_γ of $X\langle n\rangle$ is represented by a configuration $\mathbf{x}'$ of distinct points in X labeled by the blocks of π_1 and a coherent sequence of polyscreens $\mathrm{PS}^{\pi_1}, \dots, \mathrm{PS}^{\pi_k}$ at $\mathbf{x}'$. Let $p(\mathbf{x}', \beta)$ be the point in the configuration $\mathbf{x}'$ labeled by the block of π_1 that contains $\beta \subseteq [n]$. This makes sense for every block β of every $\pi \geqslant \pi_1$.

Definition. A **polyscreen** PS^π at $\mathbf{x}'$ is given by: for each block β_i of π, a configuration $\mathbf{S}_i$ of $\mathrm{card}(\beta_i)$ points in the tangent space to X at $p(\mathbf{x}', \beta_i)$, labeled by the elements of β_i, and a non-zero scalar α_i, called the **scale factor** of $\mathbf{S}_i$. The data is considered modulo the following relations:

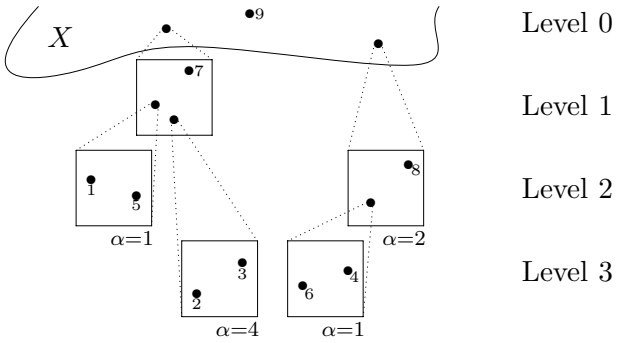


FIGURE 8

(a) translation of any screen $\mathbf{S}_i$;
(b) dilation of any screen $\mathbf{S}_i$ with compensating change of its scale factor: $(\mathbf{S}_i, \alpha_i) \sim (\phi\mathbf{S}_i, \phi^{-1}\alpha_i)$, $\phi \in \Bbbk^\times$;
(c) simultaneous multiplication of all α_i by an element of $\Bbbk^\times$ (rescaling).

A sequence of polyscreens $\mathrm{PS}^{\pi_1}, \ldots, \mathrm{PS}^{\pi_k}$ is **coherent** if, for all $j = 1, \ldots, k-1$, two labeled points in PS^{π_j} coincide if and only if their labels belong to the same block of π_{j+1}, and all labeled points in PS^{π_k} are distinct.

Coherence makes a sequence PS^γ conform to the leveled tree (T_γ, η_γ), as the example in Figure 8 of a point in $X\langle 9\rangle$ does to the (right) tree in Figure 7. This means that the root of the tree corresponds to X, each internal vertex has a screen attached to it and the direct descendants of each vertex form a configuration of distinct points in X or in the respective screen. The screens in PS^γ attached to the phantom vertices of (T_γ, η_γ) contain just one distinct labeled point and are called **trivial**; they carry no information and are left out of the pictures.

Non-trivial screens in a polyscreen PS^{π_j} are exactly Fulton–MacPherson screens for those blocks of π_j that are subdivided in π_{j+1} (all essential blocks of π_j if $j = k$). The data of PS^{π_j} is equivalent to this collection of screens together with the point in the projective space $\mathbb{P}^{r_j - 1}$ given by the r_j-tuple of scale factors, where r_j is the number of non-trivial screens in PS^{π_j}.

If $\gamma = \{\pi_1, \ldots, \pi_k\}$ starts with the bottom partition $\perp$, then for all points $\mathbf{x}$ in S_γ the configuration $\mathbf{x}'$ is a single point p in X, and all screens in $\mathrm{PS}^\gamma(\mathbf{x})$ are based on the *same* tangent space T_pX. Under the additional assumption that $\operatorname{char}\Bbbk = 0$, now made for the rest of this section, the data of each polyscreen $\mathrm{PS}^{\pi_j}(\mathbf{x})$ then fits into a single *color screen* $\mathrm{CS}^{\pi_j}(\mathbf{x})$.

Definition. Let a **color** be any non-empty subset of $[n]$. A **color screen** CS^π at p is a configuration of n colored points $x_1, \ldots, x_n$ in T_pX, considered modulo dilations of T_pX, where the color of x_i is the block of π that contains i, such that the points of each color are centered around the origin (their vector sum is 0).

A sequence $\mathrm{CS}^{\pi_1}, \ldots, \mathrm{CS}^{\pi_k}$ is **coherent** if, for all $j = 1, \ldots, k-1$, two points of the same color coincide in CS^{π_j} if and only if they have the same color in $\mathrm{CS}^{\pi_{j+1}}$, and in CS^{π_k} no points of the same color coincide.

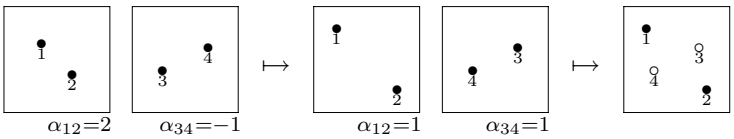


FIGURE 9. Conversion to color

To convert a polyscreen PS^π into a color screen CS^π, first translate the representative screens of PS^π to center the points around the origin, then dilate them to make all scale factors equal. Identifying now the underlying spaces of the screens, place several configurations in the same T_pX. To tell them apart, colors of points are added as a way of recording which one of the screens each point comes from. Figure 9 shows the simplest non-trivial example.

Since this conversion of polyscreens into color screens respects coherence, points in $X\langle n\rangle$ corresponding to collisions at a single point in X can be viewed in terms of coherent sequences of color screens. This interpretation is useful in Section 8 for studying the natural action of $\mathbb{S}_n$ on $X\langle n\rangle$.

4. Construction of the compactification

For a partition π of $[n]$, denote by $\Delta^\pi \subseteq X^n$ the subset of all points $(x_1, \ldots, x_n)$ with $x_i = x_j$ whenever i and j are in the same block of π, and call Δ^π a **polydiagonal**. The diagonals of X^n correspond to partitions with only one essential block. The set of all polydiagonals in X^n is naturally a lattice isomorphic to $L_{[n]}$, with its top element X^n itself.

Theorem 1. *The following $(n-1)$-stage sequence of blowups results in a smooth compactification $X\langle n\rangle$ of the configuration space of n distinct labeled points in a smooth algebraic variety X:*

- *the first stage is the blowup of Δ, the small diagonal of X^n;*
- *the k-th stage, $1 < k < n$, is the blowup of the disjoint union of the previous stage proper transforms Y^π_{k-1} of Δ^π, for all partitions π of the set $[n] = \{1, \ldots, n\}$ into exactly k blocks.*

Remark. In the language of De Concini and Procesi [DP], the building set for this iterated blowup construction consists of *all* possible intersections of the diagonals of X^n, and therefore it is maximal. The building set of the Fulton–MacPherson compactification includes only those intersections that fail to be normal, so $X[n]$ is the minimal compactification of $\mathrm{F}(X, n)$ with the property that the complement to the configuration space is a divisor with normal crossings.

Two smooth subvarieties U and V of a smooth algebraic variety W are said to **intersect cleanly** if $U \not\subset V \not\subset U$, their scheme-theoretic intersection is smooth and the tangent bundles satisfy $T(U \cap V) = TU \cap TV$. Two polydiagonals Δ^{π_1} and Δ^{π_2} in X^n intersect cleanly unless one of them contains

the other; the noncontainment condition is that the partitions π_1 and π_2 are incomparable in $L_{[n]}$.

Recall two standard results about the behaviour of clean intersections under blowups:

Lemma 1. *Let W be a smooth algebraic variety and let U, V be smooth subvarieties of W intersecting cleanly. Then*

(a) *the proper transforms of U and V in $\mathrm{Bl}_{U\cap V} W$ are disjoint;*
(b) *if Z is a smooth subvariety of $U \cap V$, then the proper transforms of U and V in $\mathrm{Bl}_Z W$ intersect cleanly.* □

Proof of Theorem 1. Denote the space obtained at stage k by Y_k and organize the projections of the fiber squares of all stages as

$$X\langle n\rangle = Y_{n-1} \longrightarrow Y_{n-2} \longrightarrow \cdots \longrightarrow Y_1 \longrightarrow Y_0 = X^n. \tag{1}$$

Then $Y_0^\pi = \Delta^\pi$ and Y_k^π is the proper transform of Y_{k-1}^π in Y_k if $\rho(\pi) \neq k$, while $Y_{\rho(\pi)}^\pi$ is the component of the exceptional divisor over $Y_{\rho(\pi)-1}^\pi$.

The statement will follow once it has been shown that the stated sequence of blowups can indeed be performed. For this, it suffices to check that the centers of those simultaneous blowups will have indeed become disjoint after the previous stages of the construction. The proof will be done by induction on k, for all $X\langle n\rangle$ at the same time; after stage k, the induction will stop for $X\langle k+1\rangle$, and it will continue on for $X\langle n\rangle$ with $n > k+1$.

For any pair of distinct partitions π_1 and π_2 of $[n]$ into two blocks, their meet $\pi_1 \wedge \pi_2$ is the 'non-partition', so $\Delta^{\pi_1} \cap \Delta^{\pi_2} = \Delta^{\pi_1\wedge\pi_2} = \Delta$, the small diagonal of X^n. By Lemma 1a, the transforms $Y_1^{\pi_1}$ and $Y_1^{\pi_2}$ will be disjoint, making the second stage possible.

Assume that stage $k-1$ has been performed; this means that the varieties $X\langle n\rangle$ have been constructed for $1 \leqslant n \leqslant k$, and only those for $n > k$ are still being built. Also assume that the proper transforms Y_{k-1}^π for π with $\rho(\pi) = k$ are disjoint.

For each partition $\pi \in L_{[n]}$ with $\rho(\pi) = k$, the projection $X\langle k\rangle \to X^k$ pulls back the obvious isomorphism $X^k \simeq \Delta^\pi \subset X^n$ to an isomorphism $X\langle k\rangle \simeq Y_{k-1}^\pi \subset Y_{k-1}$. All these subvarieties are disjoint by the inductive assumption, and can all be blown up at the same time. This defines the variety $X\langle k+1\rangle$.

To provide the inductive step necessary to continue the construction of $X\langle n\rangle$ for $n > k+1$, the intersection $Y_k^{\pi_1} \cap Y_k^{\pi_2}$ must be empty for all pairs of distinct π_1, π_2 in $L_{[n]}$ with $\rho(\pi_1) = \rho(\pi_2) = k+1$. Such a pair automatically satisfies the noncontainment condition; so $\Delta^{\pi_1} \cap \Delta^{\pi_2} = \Delta^{\pi_1\wedge\pi_2}$ is a clean intersection. Since $\rho = \rho(\pi_1 \wedge \pi_2) < k+1$, a repeated use of Lemma 1b shows that $Y_{\rho-1}^{\pi_1} \cap Y_{\rho-1}^{\pi_2} = Y_{\rho-1}^{\pi_1\wedge\pi_2}$ is a clean intersection, and then Lemma 1a implies that $Y_\rho^{\pi_1} \cap Y_\rho^{\pi_2}$ is empty. The proper transforms of Δ^{π_1} and Δ^{π_2} become disjoint after stage $\rho \leqslant k$, and the proof is complete. □

Corollary 1. *For each $\pi \in L_{[n]}$ we have $Y_{\rho(\pi)-1}^\pi \simeq X\langle\rho(\pi)\rangle$.*

Proof. This has been obtained while proving the theorem, and is formulated separately only for the ease of future reference. □

Flag Blowup Lemma. *Let $V_0^1 \subset V_0^2 \subset \cdots \subset V_0^s \subset W_0$ be a flag of smooth subvarieties in a smooth algebraic variety W_0. For $k = 1, \dots, s$, define inductively: W_k as the blowup of W_{k-1} along V_{k-1}^k; V_k^k as the exceptional divisor in W_k; and V_k^i, for $i \neq k$, as the proper transform of V_{k-1}^i in W_k. Then the preimage of V_0^s in the resulting variety W_s is a normal crossing divisor $V_s^1 \cup \cdots \cup V_s^s$.*

Remark. This auxiliary result is implicit in earlier works [FM, Ka2].

Proof. In a blowup $p\colon \mathrm{Bl}_Z W \to W$ of a smooth algebraic variety W along a smooth center Z, if $\tilde{V}$ is the proper transform of a smooth variety $V \supset Z$, then in terms of ideal sheaves $\mathcal{I}(p^{-1}(V)) = \mathcal{I}(\tilde{V}) \cdot \mathcal{I}(E)$. Applied at each step, this equality yields $\mathcal{I}(p_s^{-1}(V_0^s)) = \mathcal{I}(V_s^1) \times \cdots \times \mathcal{I}(V_s^s)$, where $p_s\colon W_s \to W_0$ denotes the composition of the stated blowups. □

Proposition 1. *For each partition π of $[n]$ with at least one essential block, there is a smooth divisor $D^\pi \subset X\langle n\rangle$ such that:*

(a) *The union of these divisors is $D = X\langle n\rangle \smallsetminus \mathrm{F}(X, n)$.*
(b) *Any set of these divisors meets transversally.*
(c) *An intersection $D^{\pi_1} \cap \cdots \cap D^{\pi_k}$ of divisors is nonempty exactly when the partitions form a chain. In other words, the incidence graph of D coincides with the comparability graph of the lattice $L_{[n]}$ with the top partition removed.*

Corollary 2. (a) *$X\langle n\rangle$ is stratified by strata $S_\gamma = \bigcap_{\pi \in \gamma} D^\pi$ parametrized by all chains γ in $L_{[n]}$.*
(b) *The codimension of S_γ in $X\langle n\rangle$ is equal to the length of γ.*
(c) *The intersection of two strata S_γ and $S_{\gamma'}$ is nonempty exactly when $\gamma \cup \gamma'$ is a chain, in which case $S_\gamma \cap S_{\gamma'} = S_{\gamma \cup \gamma'}$. In particular, $S_\gamma \supset S_{\gamma'}$ if and only if $\gamma \subset \gamma'$.*

Proof. We concentrate on the normal crossing property, which implies the other claims.

By construction, the proper transform of every polydiagonal $\Delta^\pi \subset X^n$ under $X\langle n\rangle \to X^n$ is a smooth divisor; it will be denoted by D^π. The proper transforms of Δ^{π_1} and Δ^{π_2} become disjoint when that of their intersection $\Delta^{\pi_1 \wedge \pi_2}$ is blown up, unless one of Δ^{π_1} and Δ^{π_2} contains the other, that is, unless $\{\pi_1, \pi_2\}$ is a chain.

In order to show that for any saturated (maximal length) chain $\gamma = \{\pi_i\}$, the union $D^\gamma = D^{\pi_1} \cup \cdots \cup D^{\pi_{n-1}}$ is a normal crossing divisor in $X\langle n\rangle$, consider the flag of polydiagonals $\Delta^{\pi_1} \subset \cdots \subset \Delta^{\pi_{n-1}} \subset X^n$. The blowups of $Y^\pi_{\rho(\pi)-1}$ for $\pi \notin \gamma$ are irrelevant for the intersection of the components of D^γ because their centers are disjoint from

$$\bigcap_{i=1}^{n-1} Y^{\pi_i}_{\rho(\pi)-1};$$

hence, the Flag Blowup Lemma can be applied. The normal crossing property of D^γ follows by the lemma, and so does the proposition: since any chain γ' is refined by a saturated chain γ, the components of $\bigcup_{\pi\in\gamma'} D^\pi$ form a subset of components of $\bigcup_{\pi\in\gamma} D^\pi$. □

Enumeration of the strata. The number of strata in $X\langle n\rangle$, $n > 1$, is equal to the number $2Z(n)$ of chains in $L_{[n]}$. There is a factor of 2 here because half of the chains contain $\perp$ and half do not (the top $\top$ is always excluded). Sloane and Plouffe [SP] catalogued the sequence $\{Z(n)\}$ of integers as M3649. Since the following recurrence relation is immediate:

$$Z(n) = \sum_{k=1}^{n-1} S(n,k)Z(k),$$

the first few values of $Z(n)$ are easy to compute. No closed general formula is known, although Babai and Lengyel described the asymptotics of $Z(n)$, up to yet undetermined constant [BL, Le].

Here is a small table of the numbers of strata in $X[n]$ and $X\langle n\rangle$:

n	2	3	4	5	6	7	8	9
$X[n]$	2	8	52	472	5504	78416	1320064	25637824
$X\langle n\rangle$	2	8	64	872	18024	525520	20541392	1036555120

As codimension-1 strata are the components D^π of the divisor at infinity, there are $B(n) - 1$ of them, where $B(n)$ is the Bell number, equal to the number of partitions of $[n]$. The minimal strata have codimension $n-1$ and correspond to saturated chains in $L_{[n]}$, whose number is $2^{1-n}n!(n-1)!$.

5. The Hodge polynomial of $X\langle n\rangle$

If X is a smooth complex algebraic variety, the construction of $X\langle n\rangle$ allows an easy derivation of a formula for the Hodge polynomial, hence, for the Poincaré polynomial of $X\langle n\rangle$ in terms of those of X.

The notion of a *virtual Poincaré polynomial* extends the usual one to all complex algebraic varieties and provides a good tool for computing the Poincaré polynomials of blowup constructions.

Lemma 2. (a) *If Y is smooth and compact, the virtual Poincaré polynomial $P(Y)$ coincides with the usual Poincaré polynomial of Y.*
(b) *If Z is a closed subvariety of Y, then $P(Y) = P(Z) + P(Y \smallsetminus Z)$.*
(c) *If $Y' \to Y$ is a bundle with fiber F which is locally trivial in the Zariski topology, then $P(Y') = P(Y)P(F)$.* □

Using Deligne's mixed Hodge theory [De1, De3], Danilov and Khovanskii defined a refinement of $P(X)$, the *virtual Hodge polynomial* $e(X)$, also called the *Serre polynomial*, and proved [DKh] that it has the properties listed in Lemma 2, also independently found by Durfee [Du]. Cheah, Getzler and Manin computed the Hodge polynomials of the Fulton–MacPherson

compactifications via generating functions [Ch, Ge, M], while the original paper dealt with summation over trees (groves).

Proposition 2. *For any two positive integers m and n, there is a polynomial $U_n^m(t,x)$ such that for any smooth m-dimensional complex algebraic variety X the Hodge polynomial of $X\langle n\rangle$ is $e(X\langle n\rangle;z,\bar z)=U_n^m(z\bar z,e(X;z,\bar z))$, and in particular, $P(X\langle n\rangle;t)=U_n^m(t,P(X;t))$. The polynomials $U_n^m(t,x)$ satisfy the recurrence relation*

$$U_n^m(t,x)=x^n+\sum_{k=1}^{n-1}S(n,k)h_{(n-k)m}(t)U_k^m(t,x),$$

where $h_d(t)=P(\mathbb{CP}^{d-1})-1=t^{2d-2}+\cdots+t^4+t^2$.

Proof. Straightforwardly from the construction of $X\langle n\rangle$ and Lemma 2,

$$e(Y_k)=e(Y_{k-1})+\sum_{\rho(\pi)=k}\big(e(\mathbb{P}(N^\pi))-1\big)e(Y_{k-1}^\pi),$$

where N^π is the fiber of the normal bundle to Y_{k-1}^π in Y_{k-1}, which is $\mathbb{C}^{(n-k)m}$ by an easy dimension count. Corollary 1 converts this formula into

$$e(Y_k)=e(Y_{k-1})+S(n,k)h_{(n-k)m}(z\bar z)e(X\langle k\rangle).$$

Since $Y_0=X^n$ and $Y_{n-1}=X\langle n\rangle$, there results a recurrence relation

$$e(X\langle n\rangle)=e(X)^n+\sum_{k=1}^{n-1}S(n,k)h_{(n-k)m}(z\bar z)e(X\langle k\rangle),$$

and both claims immediately follow. □

A nonrecursive expression for $U_n^m(t,x)$ can be found by expanding in the right-hand side of the recurrence the terms with the highest k present in a loop down to $k=2$ terms:

$$U_n^m(t,x)=x^n+\sum_{s=1}^{n-1}\left(x^s\sum_{r=1}^{n-s}\sum_{\mathrm{J}_{s,n}^r}\prod_{i=1}^{r}S(j_i,j_{i-1})h_{(j_i-j_{i-1})m}(t)\right),$$

where $\mathrm{J}_{s,n}^r=\{(j_0,\dots,j_r)\in\mathbb{Z}^{r+1}\,|\,s=j_0<\cdots<j_r=n\}$.

Similar computations of the Hodge polynomials of the strata in the stratification of $X\langle n\rangle$ from Corollary 2 can be carried out using the description of their structure given in Section 7.

6. $X\langle n\rangle$ as a closure and a surjection $X\langle n\rangle\to X[n]$

In this section I present $X\langle n\rangle$ as the closure of the configuration space embedded in a product of blowups, exhibit a surjection $X\langle n\rangle\to X[n]$, and write it as an iterated blowup.

First, the results of Section 4 about the structure of $X\langle n\rangle$ at infinity should be rephrased in terms of ideal sheaves. Let $\mathcal{I}(\Delta^\pi)$ be the ideal sheaf of Δ^π in $\mathcal{O}_{X^n}$. For any k, $1\leqslant k\leqslant n-1$, let $\tau_k\colon Y_k\to X^n$ be the appropriate

composition of projections from Eq. (1), let $\mathcal{I}_k(\pi)$ be the ideal sheaf in $\mathcal{O}_{Y_k}$ generated by $\tau_k^*(\mathcal{I}(\Delta^\pi))$, and also let $\mathcal{I}(Y_k^\pi)$ be the ideal sheaf of Y_k^π in $\mathcal{O}_{Y_k}$. This notation, although similar to Fulton and MacPherson's, is not quite the same. The assertions of Proposition 1 can be restated as

$$\mathcal{I}_{n-1}(\pi) = \prod_{\pi' \leqslant \pi} \mathcal{I}(D^{\pi'}),$$

while at the intermediate stages

$$\mathcal{I}_k(\pi) = \prod_{\pi' \leqslant \pi \text{ with } \rho(\pi') \leqslant k} \mathcal{I}(Y_k^{\pi'}).$$

Since $Y_k^{\pi'} \subset Y_k$ is a divisor if $\rho(\pi') < k$, it follows that $\mathcal{I}_k(\pi) = \mathcal{I}(Y_k^\pi) \cdot \mathcal{J}$, where $\mathcal{J}$ is an invertible ideal sheaf.

Proposition 3. *The variety $X\langle n\rangle$ constructed by blowing up is the closure of the configuration space $\mathrm{F}(X,n)$ in*

$$\prod_{\pi \in L_{[n]}} \mathrm{Bl}_{\Delta^\pi} X^n.$$

Remark. The top partition contributes the factor X^n.

Proof. By induction on k, each Y_k is the closure of $\mathrm{F}(X,n)$ in

$$X^n \times \prod_{\rho(\pi) \leqslant k} \mathrm{Bl}_{\Delta^\pi} X^n.$$

The basis is clear: $Y_0 = X^n$. Then, Y_k is the blowup of Y_{k-1} along

$$\coprod_{\rho(\pi)=k} Y_{k-1}^\pi,$$

or in other terms, along

$$\mathcal{I}\left(\coprod_{\rho(\pi)=k} Y_{k-1}^\pi \right) = \prod_{\rho(\pi)=k} \mathcal{I}(Y_{k-1}^\pi).$$

This ideal sheaf becomes

$$\mathcal{I}_{k-1} = \prod_{\rho(\pi)=k} \mathcal{I}_{k-1}(\pi)$$

upon multiplying by an invertible ideal sheaf, and blowing up $\mathcal{I}_{k-1}$ is equivalent to taking the closure of the graph of the rational map from Y_{k-1} to

$$\prod_{\rho(\pi)=k} \mathrm{Bl}_{\Delta^\pi} X^n.$$

This provides the inductive step, and eventually $Y_{n-1} = X\langle n\rangle$. □

Both the statement and its proof parallel those by Fulton and MacPherson [FM, Prop. 4.1], who use pullbacks by $X[n] \to X^n \to X^S$, for $S \subset [n]$, $\#S > 1$, and also by $f_S \colon Y_k \to X^n \to X^S$ at the intermediate stages, while here $\tau_k \colon Y_k \to X^n$. A slight reformulation of their characterization of $X[n]$ as a closure elucidates its similarity with $X\langle n\rangle$. For each S as before, take the diagonal $\Delta^S \subset X^n$ and pull back its ideal sheaf by the first of the two arrows whose composition is f_S; this gives the same ideal sheaf $f_S^*(\mathcal{I}(\Delta))$.

Proposition 4. *The variety $X[n]$ is the closure of $\mathrm{F}(X,n)$ in*

$$X^n \times \prod_{S\subset[n],\#S>1} \mathrm{Bl}_{\Delta^S} X^n.$$

The two compactifications can now be related.

Proposition 5. *For each $n \geqslant 1$, there is a surjection $\vartheta_n \colon X\langle n\rangle \to X[n]$.*

Proof. Start with notation for the products from Propositions 3 and 4:

$$\Pi = \prod_{\epsilon(\pi)\geqslant 1} \mathrm{Bl}_{\Delta^\pi} X^n, \quad \text{and} \quad \Pi' = \prod_{\epsilon(\pi)=1} \mathrm{Bl}_{\Delta^\pi} X^n,$$

where $\epsilon(\pi)$ is the number of essential blocks in a partition π. If S is the only essential block of π, then $\Delta^S = \Delta^\pi$, so Π' can indeed be used for $X[n]$.

Now take the left of these two diagrams, where ϕ and ψ are rational maps defined on $\mathrm{F}(X,n)$, and notice that the projection $\mathrm{id} \times p$ maps the closure $\overline{G(\phi)}$ of the graph of ϕ onto the closure $\overline{G(\psi)}$ of the graph of ψ. □

This surjection ϑ_n admits a more explicit description. For $n \leqslant 3$, it is the identity map; otherwise, it can be written as a composition

$$X\langle n\rangle = W_{n-2} \xrightarrow{\beta_{n-2}} W_{n-3} \xrightarrow{\beta_{n-3}} \cdots \xrightarrow{\beta_3} W_2 \xrightarrow{\beta_2} W_1 = X[n], \tag{2}$$

where $W_k \xrightarrow{\beta_k} W_{k-1}$ is the blowup in W_{k-1} of (the disjoint union of the proper transforms under $\beta_{k-1} \circ \cdots \circ \beta_2$ of) some strata $X(\mathcal{S})$ of $X[n]$; their encoding nests $\mathcal{S}$ are characterized below. Favoring imprecision over repetitiveness, I will neglect to reiterate the ritual phrase that in the previous sentence appears in parentheses.

Let $U \subset X[n]$ be the union of all strata $X(\mathcal{S})$ such that the nest $\mathcal{S}$ contains two disjoint subsets of $[n]$. The irreducible components of this codimension 2 reduced subscheme are $X(\mathcal{S})$ for all nests $\mathcal{S} = \{S_1, S_2\}$ with $S_1 \cap S_2 = \varnothing$, which intersect transversally [FM, Theorem 3]. The map ϑ_n is an iterated blowup of $X[n]$ along U, but not all the strata contained in U are centers

of a blowup β_k. The components of the center of β_k are the strata $X(\mathcal{S})$ such that $\mathcal{S}$ is the set of all essential blocks of a partition $\pi \in L_{[n]}$ with $\rho(\pi) = k$ and $\epsilon(\pi) > 1$, which is always a nest. The transversality of the strata guarantees that, whenever the sequence $\beta_2, \dots, \beta_{n-2}$ calls for two intersecting strata to be in the center of the same β_k, the previous stages will have made them disjoint. The sequence itself implies that, whenever $X(\mathcal{S}) \subset X(\mathcal{S}')$ are both to become centers, the smaller stratum is blown up before the larger one.

Alternatively, the variety W_k can be defined as the closure of $\mathrm{F}(X, n)$ in

$$X^n \times \prod_{\rho(\pi)\leqslant k \text{ or } \epsilon(\pi)=1} \mathrm{Bl}_{\Delta^\pi} X^n,$$

and an argument similar to Proposition 3 shows that this is equivalent to the blowup description.

Examples. Here $X(S_1, \dots, S_k) = D(S_1) \cap \dots \cap D(S_k)$ refers to strata of $X[n]$.

The map ϑ_4 is the blowup of 3 disjoint codimension-2 strata $X(12, 34)$ and alike, for the nests obtained from the 3 partitions of shape $(2, 2)$. The divisor $D^{12|34} \subset X\langle 4\rangle$ is a $\mathbb{P}^1$-bundle over $X(12, 34)$.

For $n = 5$, there are two maps in Eq. (2). The first blows up 10 disjoint codimension-2 strata, like $X(123, 45)$, corresponding to the partitions of shape $(3, 2)$. The second blows up 15 disjoint codimension-2 strata, like $X(12, 34)$, corresponding to $(2, 2, 1)$.

For $n = 6$, there are three stages according to the partitions

$$(4, 2), \quad (3, 3); \qquad (3, 2, 1), \quad (2, 2, 2); \qquad (2, 2, 1, 1).$$

Here we encounter inclusions like $X(12, 34, 56) \subset X(12, 34)$. Interestingly, the proper transform by ϑ_6 of $X(12, 34, 56)$, which is the divisor $D^{12|34|56}$, is a bundle over $X(12, 34, 56)$ with fiber $\mathbb{P}^2$ blown up at three points. Proposition 11 generalizes this observation.

The preimage in $X\langle n\rangle$ of a stratum of $X[n]$ is

$$\vartheta_n^{-1}(X(\mathcal{S})) = \bigcup_{(T,\eta)\in\theta^{-1}(T(\mathcal{S}))} S_{(T,\eta)},$$

the union of all strata encoded by the leveled trees (T, η) with the same base tree $T(\mathcal{S})$ and any legal assignment of levels to its interior vertices. The map ϑ_n is thus strata-compatible.

Proposition 6. *The fibers of $\vartheta_n \colon X\langle n\rangle \to X[n]$ are independent of X and even of its dimension. The fiber over a point in $X(\mathcal{S})$ that is not in any smaller stratum is completely determined by the nest $\mathcal{S}$.*

Proof. The normal space $N_{\mathbf{x}}$ at a point $\mathbf{x}$ to $X(\mathcal{S}) \subset X[n]$ is independent of $\dim X$ (assumed positive): its dimension is equal to the cardinality of the nest $\mathcal{S}$. The nest alone determines the iterated blowup of $N_{\mathbf{x}}$ induced from Eq. (2), and the preimage of the origin under it is isomorphic to $\vartheta_n^{-1}(\mathbf{x})$. □

7. Structure of the strata

This section begins by discussing a family of linear subspace arrangements indexed by integer partitions; each of them leads to a projective variety that will be called a brick. Points of a brick correspond to polyscreens, and by presenting the strata of $X\langle n\rangle$ as bundles over $X\langle k\rangle$ whose fibers are products of bricks, the polyscreen description of $X\langle n\rangle$ is established here.

The configuration space $\mathrm{F}(\mathbb{A}^1, n)$ is the complement to the braid arrangement of hyperplanes in $\mathbb{A}^n$, the motivating example for much of the theory of hyperplane arrangements [OT]. The analogue for $\mathbb{A}^m$, denoted by $\bar{\mathcal{B}}^m_n$, is an arrangement of codimension m linear subspaces of $(\mathbb{A}^m)^n$. Its strata are various intersections of the large diagonals, so the partitions of $[n]$ index them, for each $m \geqslant 1$; in other words, the intersection lattice of $\bar{\mathcal{B}}^m_n$ is isomorphic to the partition lattice $L_{[n]}$. These and all other subspace arrangements encountered in this section are c-plexifications of hyperplane arrangements [Bj]. This means practically that most information about $\bar{\mathcal{B}}^m_n$ can be extracted from the braid arrangement $\bar{\mathcal{B}}^1_n$.

For any partition π of $[n]$, the images in the quotient $C^m_\pi = (\mathbb{A}^m)^n/\Delta^\pi$ of those large diagonals that contain Δ^π form an induced arrangement $\mathcal{B}^m_\pi$. For $\pi = \perp(L_{[n]})$, it is denoted by $\mathcal{B}^m_{n-1}$ (actual subscripts will be integers ν_i); if $m = 1$, this is the Coxeter arrangement of type A_n. For other partitions, $\mathcal{B}^m_\pi$ is a product arrangement, as Lemma 3 shows below.

For two subspace arrangements $\mathcal{A}_i = \{K^i_1, \dots, K^i_{s_i}\}$ in $\Bbbk$-vector spaces V_i, $i = 1, 2$, the product arrangement $\mathcal{A}_1 \times \mathcal{A}_2$ in $V_1 \oplus V_2$ is the collection of subspaces $\{K^1_1 \oplus V_2, \dots, K^1_{s_1} \oplus V_2, K^2_1 \oplus V_1, \dots, K^2_{s_2} \oplus V_1\}$. For each integer partition $\lambda = (\nu_1, \dots, \nu_r)$, define $\mathcal{B}^m_\lambda$ as the product $\mathcal{B}^m_{\nu_1} \times \cdots \times \mathcal{B}^m_{\nu_r}$. The intersection lattice of a product is the product of those of the factors; for $\mathcal{B}^m_\lambda$ this gives the lattice $L_\lambda = L_{[\nu_1+1]} \times \cdots \times L_{[\nu_r+1]}$.

As an example, take for λ the finest partition $(1, \dots, 1)$ of r, often denoted by 1^r. Since $\mathcal{B}^1_1$ is the arrangement $\{0\}$ in $\Bbbk$, its r-th power $\mathcal{B}^1_{1^r}$ is the arrangement of coordinate hyperplanes in $\Bbbk^r$.

Lemma 3. *Up to a change of coordinates $\mathcal{B}^m_\pi \simeq \mathcal{B}^m_\lambda$, where $\lambda = \lambda(\pi)$ is the essential shape of π.*

Proof. Look at the equations of the large diagonals containing Δ^π, that is, $\Delta^{ij} = \{(x_1, \dots, x_n) \in (\mathbb{A}^m)^n \,|\, x_i = x_j\}$ for all pairs of i and j belonging to the same block of π. Equations coming from different blocks of π are independent of each other, leading to the product decomposition. □

The polydiagonal compactification $\mathbb{A}^m\langle n\rangle$ is the maximal blowup of the arrangement $\bar{\mathcal{B}}^m_n$, in the sense that all strata of $\bar{\mathcal{B}}^m_n$ are blown up in the course of its construction. In the same fashion, all strata of the arrangement $\mathcal{B}^m_\lambda$ can be blown up in the ascending order given by their dimensions. The first stage is always the blowup of the origin, creating the exceptional divisor $\mathbb{P}(C^m_\lambda) \simeq \mathbb{P}^{m|\lambda|-1}$, where $|\lambda|$ is the sum of all parts of λ.

The main objects of interest for this section are defined as follows.

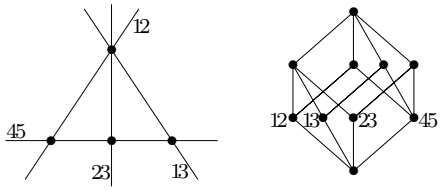


FIGURE 10

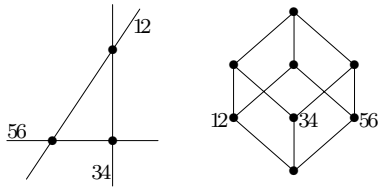


FIGURE 11

Definition. For any integer partition λ, a **brick** M^m_λ is the proper transform of $\mathbb{P}(C^m_\lambda)$ in the maximal blowup of $\mathcal{B}^m_\lambda$. If λ has only one part, the brick M^m_λ is **simple**, otherwise it is **compound**. The **open** brick $^\circ M^m_\lambda$ is the complement in $\mathbb{P}(C^m_\lambda)$ of the projectivization of $\mathcal{B}^m_\lambda$.

Examples. The brick M^m_1 is just $\mathbb{P}^{m-1}$ (a single point if $m = 1$).

The bricks M^m_2 and $M^m_{1,1}$ are blowups of $\mathbb{P}^{2m-1}$; their centers are, respectively, three and two copies of M^m_1.

The bricks M^m_3, $M^m_{2,1}$ and $M^m_{1,1,1}$ are 2-stage blowups of $\mathbb{P}^{3m-1}$; the lower intervals in L_3, $L_{2,1}$ and $L_{1,1,1}$ determine their centers, respectively:

7 copies of M^m_1, then 6 copies of M^m_2;
4 copies of M^m_1, then 3 copies of $M^m_{1,1}$ and 1 copy of M^m_2;
3 copies of M^m_1, then 3 copies of $M^m_{1,1}$.

For M^1_3, look again at Figures 2 and 3 on page 6. Similar pictures for $M^1_{2,1}$ and $M^1_{1,1,1}$ are in Figures 10 and 11. Comparison of these figures suggests that refining the indexing partition corresponds to omitting some subspaces from the arrangement. This is proved in general in Proposition 12.

Of special importance is the brick $M^1_{1^r}$ that arises from the coordinate arrangement in $\Bbbk^r$: blow up r points in $\mathbb{P}^{r-1}$ in general position, then blow up the proper transforms of all lines spanned by pairs of these points, then blow up those of all planes spanned by triples, and so on. Thus $M^1_{1^r}$ is isomorphic to the space Π_r that Kapranov called the *permutahedral space* [Ka2, p. 105]. It is the compact projective toric variety whose encoding polytope is the permutahedron P_r, usually defined as the convex hull of the set of $r!$ points in $\mathbb{R}^r$ with coordinates $(\sigma^{-1}(1), \dots, \sigma^{-1}(r))$, for all $\sigma \in \mathbb{S}_r$. This polytope can also be obtained from the standard $(r-1)$-simplex by chopping off first all its vertices, then all that remains of its edges, then faces, and so on; this corresponds to the sequence of blowups producing Π_r. In addition, this variety is the closure of a principal toric orbit in the complete flag variety and it has been extensively studied from various perspectives [At, DL, GS, P, Sta2, Ste1, Ste2].

For each $m \geqslant 1$, the brick $M^m_{1^r}$ is a toric variety because all strata of $\mathcal{B}^m_{1^r}$, sitting in $\mathbb{P}^{rm-1}$, are $(\Bbbk^\times)^{rm}$-invariant.

Proposition 7. *Every open compound brick has the structure of a bundle*

$$^\circ M^1_{1^r} \longrightarrow {}^\circ M^m_\lambda \longrightarrow {}^\circ M^m_{\nu_1} \times \cdots \times {}^\circ M^m_{\nu_r}, \tag{3}$$

where λ is the integer partition $(\nu_1, \dots, \nu_r)$.

Proof. The complement to $\mathcal{B}^m_{\nu_i}$ in $(\mathbb{A}^m)^{\nu_i}$ is $\mathrm{F}(\mathbb{A}^m, \nu_i + 1)/\mathbb{A}^m$, the configuration space of $\nu_i + 1$ distinct labeled points in $\mathbb{A}^m$ modulo translations. Since ${}^\circ M^m_\lambda$ is the complement in $\mathbb{P}(C^m_\lambda)$ to the projectivization of the arrangement $\mathcal{B}^m_\lambda = \mathcal{B}^m_{\nu_1} \times \cdots \times \mathcal{B}^m_{\nu_r}$, it follows that

$$ {}^\circ M^m_\lambda = \mathbb{P}({}^\circ C^m_\lambda), \quad \text{where} \quad {}^\circ C^m_\lambda = \prod_{i=1}^{r} \Big(\mathrm{F}(\mathbb{A}^m, \nu_i + 1) \Big/ \mathbb{A}^m \Big), \tag{4} $$

is the orbit space of the diagonal action of $\Bbbk^\times$ on this product by dilations.

Separate actions of $\Bbbk^\times$ on each factor together give that of $(\Bbbk^\times)^r$ on ${}^\circ C^m_\lambda$. Its total orbit space is isomorphic to the product of those coming from the factors, which is ${}^\circ M^m_{\nu_1} \times \cdots \times {}^\circ M^m_{\nu_r}$. The orbit space ${}^\circ M^m_\lambda$ maps into this product, with fiber $(\Bbbk^\times)^r/\Bbbk^\times \simeq {}^\circ M^1_{1^r}$. □

The next two propositions follow from the general work of De Concini and Procesi [DP, pages 480–482], but they can also be proved directly.

Proposition 8. (a) *The compactification $\mathbb{A}^m\langle n\rangle$ is the product $\mathbb{A}^m \times \mathcal{L}$, where $\mathcal{L}$ is the total space of a line bundle over the simple brick M^m_{n-1}.*
(b) *The simple brick M^m_{n-1} is a compactification of $\mathrm{F}(\mathbb{A}^m, n)/\mathrm{Aff}$, where* Aff *is the group of all affine transformations in $\mathbb{A}^m$.*

Proof. (a) The direct factor $\mathbb{A}^m$ is the small diagonal $\Delta \subset (\mathbb{A}^m)^n$. The essential shape of the bottom partition of $[n]$ is the integer $n-1$, thus by definition, there is a map $\psi\colon M^m_{n-1} \to P = \mathbb{P}((\mathbb{A}^m)^n/\Delta)$. The bundle $\mathcal{L}$ is the pullback by ψ of the tautologial line bundle over P; since ψ is an iterated blowup, Lemma 4 (formulated below) has to be used at each stage.

(b) Affine transformations identify any non-degenerate configuration in $\mathbb{A}^m$ with a degenerate one in which all n points collide at 0, cancelling both the direct factor $\mathbb{A}^m$ and the fiber of the line bundle $\mathcal{L}^n$. □

Lemma 4. *Let V be a smooth subvariety of a smooth algebraic variety W, let $h\colon F \to W$ be a vector bundle over W, and E its restriction onto V. Then $\mathrm{Bl}_E F$ is a vector bundle over $\mathrm{Bl}_V W$ isomorphic to the pullback of F by the blowup projection.*

Proof. The normal bundle $N_{E/F}$ is the pullback $h^* N_{V/W}$. □

There are two differences between the construction of $\mathbb{A}^m\langle n\rangle$ and that of the bricks: different arrangements to start with and projectivization; both are minor enough that some basic facts about bricks follow by the same arguments that apply to $\mathbb{A}^m\langle n\rangle$. In turn, describing first the strata of the bricks provides a quick way of doing the same for $X\langle n\rangle$.

Proposition 9. *For any integer partition λ, fix a partition π of $[n]$ of essential shape λ and an isomorphism $[\pi, \top] \simeq L_\lambda$.*
(a) *For each partition π_1, $\pi < \pi_1 < \top$, there is a divisor E^{π_1} in M^m_λ. The union of these divisors is the complement $M^m_\lambda \smallsetminus {}^\circ M^m_\lambda$, and any set of them meets transversally.*

(b) *An intersection $E^{\pi_1} \cap \cdots \cap E^{\pi_k}$ is nonempty exactly when the partitions form a chain. Thus M^m_λ is stratified by strata parametrized by all chains in L_λ that include neither its bottom nor its top.*

(c) *For any such chain $\{\pi_1, \dots, \pi_k\}$, the corresponding stratum of M^m_λ is isomorphic to $M^m_{\lambda_0} \times \cdots \times M^m_{\lambda_k}$, where the integer partitions $\lambda_0, \dots, \lambda_k$ are determined by $L_{\lambda_i} \simeq [\pi_i, \pi_{i+1}]$, with $\pi_0 = \pi$ and $\pi_{k+1} = \top$.*

Definition. A smooth subvariety V of a smooth algebraic variety W will be called **straight** if the normal bundle $N_V W$ is isomorphic to a direct sum of copies of a single line bundle. In this case, the exceptional divisor of the blowup $\mathrm{Bl}_V W$ is a trivial bundle.

Lemma 5. (a) *For any two positive integers k and l, any linear subvariety $\mathbb{P}^k$ of $\mathbb{P}^{k+l+1}$ is straight. (Whence the term.)*

(b) *Let Z and V be smooth subvarieties of a smooth algebraic variety W, such that either $Z \cap V = \varnothing$ or $Z \subset V$. If V is straight in W, then so is its proper transform $\tilde{V}$ in $\tilde{W} = \mathrm{Bl}_Z W$.*

Proof. Part (a) follows directly from the definition.

(b) Nothing to be done when V and Z are disjoint. When $Z \subset V$, denote by E the exceptional divisor of $\tilde{W}$, and by p the projection $\tilde{V} \to V$, then

$$N_{\tilde{V}/\tilde{W}} \simeq p^* N_{V/W} \otimes \mathcal{O}(-E)\big|_{\tilde{V}},$$

and the claim follows. □

Proof of Proposition 9. Similarly to Proposition 1, the definition of M^m_λ implies parts (a) and (b).

Part (c) can be checked by induction on k, where the inductive step follows by applying the case $k = 1$. Thus, it is enough to show that each divisor E^π is isomorphic to $M^m_{\lambda_0} \times M^m_{\lambda_1}$, where $L_{\lambda_0} \simeq [\pi, \pi_1]$ and $L_{\lambda_1} \simeq [\pi_1, \top]$. The argument is based on Lemmas 4 and 5.

Every partition from $[\pi, \top]$ belongs to one of the following six groups:

(i) $\{\pi\}$, (ii) $\{\pi' \,|\, \pi < \pi' < \pi_1\}$,

(iii) $\{\pi_1\}$, (iv) $\{\pi' \,|\, \pi_1 < \pi' < \top\}$,

(v) $\{\top\}$, (vi) incomparable with π_1.

The proof will be completed by studying the impact of blowups corresponding to partitions in each group on the stratum $\Delta^{\pi_1}/\Delta^\pi$ of the arrangement $\mathcal{B}^m_\lambda$. Before the blowups, the arrangements induced in $\Delta^{\pi_1}/\Delta^\pi$ and $(\mathbb{A}^m)^n/\Delta^{\pi_1}$ are isomorphic respectively to $\mathcal{B}^m_{\lambda_0}$ and $\mathcal{B}^m_{\lambda_1}$.

First group, first stage. The exceptional divisor $\mathbb{P}(C^m_\lambda)$ of the first stage has a straight subvariety $\mathbb{P}(\Delta^{\pi_1}/\Delta^\pi) \simeq \mathbb{P}(C^m_{\lambda_0})$ with the projectivization of $\mathcal{B}^m_{\lambda_0}$ in it, and with the arrangement $\mathcal{B}^m_{\lambda_1}$ in each normal space to it (Lemma 4). Lemmas 4 and 5 also apply at the subsequent stages, pulling back arrangements inside normal spaces and preserving the straightness of blowup centers. Group (vi) blowups are irrelevant for the divisor E^π at all stages, and no blowup corresponds to $\top$.

Group (ii) blowups turn $\mathbb{P}(C^m_{\lambda_0})$ into $M^m_{\lambda_0}$. Then the group (iii) blowup makes it into a divisor isomorphic to $M^m_{\lambda_0} \times \mathbb{P}(C^m_{\lambda_1})$. The second factor inherits the projectivization of $\mathcal{B}^m_{\lambda_1}$, and blowups of the remaining group (iv) transform this divisor into $E^{\pi_1} \simeq M^m_{\lambda_0} \times M^m_{\lambda_1}$. □

Lemma 6. *Each divisor D^π of $X\langle n\rangle$ is isomorphic to a bundle over $X\langle\rho(\pi)\rangle$ with fiber $M^m_{\lambda(\pi)}$. In addition, this bundle is trivial if $X = \mathbb{A}^m$.*

Proof. Corollary 1 gives $Y^\pi_{r-1} \simeq X\langle r\rangle$, where $r = \rho(\pi)$. By Lemma 4, the arrangements $\mathcal{B}^m_\pi$ transform isomorphically from the normal spaces to Δ^π in $X\langle n\rangle$ into the normal spaces to Y^π_{r-1} in Y_{r-1}. At the next stage, Y^π_r is a bundle over $X\langle r\rangle$ with fibers isomorphic to $P = \mathbb{P}((\mathbb{A}^m)^n/\Delta)$. The relevant blowup centers of the subsequent stages are its subbundles; their fibers form in every fiber of Y^π_r an arrangement isomorphic to the projectivization of $\mathcal{B}^m_\pi$ in P. Thus in the end, the fibers of Y^π_r transform into $M^m_{\lambda(\pi)}$.

If in addition $X = \mathbb{A}^m$, a repeated application of Lemma 5 shows that Y^π_{r-1} is straight in Y^π, so $Y^\pi_r = \mathbb{A}^m\langle r\rangle \times P$ and the result follows. □

Proposition 10. *Let $\gamma = \{\pi_1, \dots, \pi_k\}$ be a chain of partitions of $[n]$ and let $\{\lambda_0, \dots, \lambda_k\}$ be its associated sequence of integer partitions (Section 2).*

(a) *The stratum S_γ of $X\langle n\rangle$ is a bundle over $X\langle\lambda_0\rangle$ with fiber isomorphic to $M^m_{\lambda_1} \times \dots \times M^m_{\lambda_k}$.*
(b) *Consequently, the complement in S_γ to the union of smaller strata, the open stratum ${}^\circ S_\gamma$, is a bundle over $\mathrm{F}(X, \lambda_0)$ with fiber isomorphic to ${}^\circ M^m_{\lambda_1} \times \dots \times {}^\circ M^m_{\lambda_k}$.*

Proof. Put together Lemma 6 and Proposition 9. □

By this proposition, a point in a stratum ${}^\circ S_\gamma$ is given by a configuration of r distinct points in X (where the collision occurs) and a sequence consisting of one point in each open brick ${}^\circ M^m_{\lambda_i}$. Equations (3) and (4) in Proposition 7 show that such points can be represented by suitable polyscreens: points in each constituent open simple brick are Fulton–MacPherson screens, and points in ${}^\circ M^1_{1^r}$ are r-tuples of scale factors. Thus, points of $X\langle n\rangle$ indeed have the geometric description explained in Section 3.

Proposition 11. *The compound brick $M^m_{1^r}$ has the structure of a bundle*

$$\Pi_r \longrightarrow M^m_{1^r} \longrightarrow (M^m_1)^r. \tag{5}$$

Proof. Fix a partition π of $[2r]$ into two-element blocks and let the nest $\mathcal{S}$ be the set $\{\beta_1, \dots, \beta_r\}$ of blocks of π. The map $\vartheta_{2r} \colon \mathbb{A}^m\langle 2r\rangle \to \mathbb{A}^m[2r]$ takes the divisor D^π of $\mathbb{A}^m\langle 2r\rangle$ into the stratum $\mathbb{A}^m(\mathcal{S})$ of $\mathbb{A}^m[2r]$. The divisor is isomorphic to $\mathbb{A}^m\langle r\rangle \times M^m_{1^r}$ by Lemma 6 and the stratum is isomorphic to $\mathbb{A}^m[r] \times (\mathbb{P}^{m-1})^r$. Since $M^m_1 \simeq \mathbb{P}^{m-1}$, it follows that $M^m_{1^r}$ maps to $(M^m_1)^r$.

Tracing ϑ^{-1}_{2r} stage by stage, first transform the factor $\mathbb{A}^m[r]$ into $\mathbb{A}^m\langle r\rangle$; then at stage r blow up the proper transform of $\mathbb{A}^m(\mathcal{S})$, turning the second factor into a $\mathbb{P}^{r-1}$-bundle over $(M^m_1)^r$. Since $\mathbb{A}^m[n] \smallsetminus \mathrm{F}(\mathbb{A}^m, n)$ is a normal

FIGURE 12. One level splits into two

crossing divisor, the divisors $D(\beta_i)$ induce in each fiber $\mathbb{P}^{r-1}$ the projectivized coordinate hyperplane arrangement. All of its strata are blown up at the subsequent stages, turning $\mathbb{P}^{r-1}$ into $M^1_{1^r} \simeq \Pi_r$. □

The fiber Π_r in Eq. (5) stores scale factors; points in its open part $^\circ M^1_{1^r}$ are generic and each is a part of one polyscreen. The divisor $E_r = \Pi_r \smallsetminus {}^\circ M^1_{1^r}$ has components isomorphic to $\Pi_s \times \Pi_{r-s}$, whose points represent degenerations with s scale factors tending to zero, and therefore polyscreens that split into two: s screens form a new level. For example, the left leveled tree in Figure 12 may degenerate into the right one, corresponding to a divisor $\Pi_4 \times \Pi_3 \subset \Pi_7$. The new levels may of course split further; intersections of components of E_r give a stratification of Π_r, and each stratum is a product of a number of smaller permutahedral varieties. This corresponds to the well-known fact that all faces of the permutahedron P_r are products of lower-dimensional permutahedra [BS].

Other compound bricks, that is, M^m_λ for those integer partitions λ that have parts greater than 1, do not admit decompositions similar to Eq. (5), but each of them is a blowup of $M^m_{1^r}$ for $r = |\lambda|$. Let Λ_r be the set of all partitions of an integer r partially ordered by refinement: $(5,3) < (4,2,1,1)$ in Λ_8 because $5 = 4+1$ and $3 = 2+1$. It turns out that the set of bricks $\{M^m_\lambda \,|\, \lambda \in \Lambda_r\}$ has a compatible (reverse) 'blowing-up' partial order.

Proposition 12. *Suppose that $\lambda, \lambda' \in \Lambda_r$ and $\lambda < \lambda'$.*

(a) *The lattice $L_{\lambda'}$ contains a sublattice isomorphic to L_λ.*
(b) *The subarrangement of $\mathcal{B}^m_{\lambda'}$ formed by the subspaces that correspond to this sublattice is $\mathcal{B}^m_\lambda$, up to coordinate change.*
(c) *The brick $M^m_{\lambda'}$ is an iterated blowup of M^m_λ.*

Proof. (a) It is enough to show this for $\lambda' = (r-1)$ and $\lambda = (s-1, r-s)$. The required sublattice of $L_{[r]}$ is generated by the union $[\pi_1, \top] \cup [\pi_2, \top]$, where the only essential block of π_1 (π_2) is $\{k \,|\, k \leqslant s\}$ (resp. $\{k \,|\, k \geqslant s\}$).

(b) It is enough to consider the same λ and λ' as in (a) and then write explicitly the equations for the large diagonals.

(c) The two lattices $L_\lambda \subset L_{\lambda'}$ determine the sequences of blowups of $\mathbb{P}^{rm-1}$ creating M^m_λ and $M^m_{\lambda'}$. It suffices to show that the blowups making $M^m_{\lambda'}$ can be rearranged, without changing the outcome (up to an isomorphism), into a different sequence so that an intermediate stage is M^m_λ. This situation is quite similar to the consideration of $\vartheta_n \colon X\langle n\rangle \to X[n]$ in Section 6, and similar is the solution. □

8. Isotropy of the permutation action

Assume that the ground field $\Bbbk$ is of characteristic 0. Reading carefully into Fulton and MacPherson's proof of the solvability of the isotropy subgroups of $\mathbb{S}_n$ acting on $X[n]$, one soon realizes that every point where the isotropy subgroup fails to be abelian lies in a stratum whose encoding nest contains a pair of disjoint subsets of $[n]$. Exactly these strata are blown up by $\vartheta_n \colon X\langle n\rangle \to X[n]$, and this observation raises hopes that are not false.

Theorem 2. *If X is a smooth algebraic variety over a field $\Bbbk$ of characteristic 0, then all isotropy subgroups of $\mathbb{S}_n$ acting on $X\langle n\rangle$ by permutations of labels are abelian.*

Proof. First, reduce to the case of all n points colliding at the same point in X. Suppose a collision $\mathbf{x}$ occurs at $p_1, \dots, p_r \in X$. If it could be studied near each p_i independently of the other points, as for $X[n]$, the isotropy subgroup would have been $G^{p_1} \times \cdots \times G^{p_r}$, where G^{p_i} is the isotropy subgroup of the collision near p_i. It would have corresponded to r independent sequences of color screens and reduced the proof to the case of one collision point, but this does not suit $X\langle n\rangle$. Fortunately, interdependencies among the corresponding levels in those r sequences only put more restrictions on a permutation aspiring to fix $\mathbf{x}$. It means the isotropy subgroup will be a subgroup of the above product, which still does the trick.

Pick a k-chain $\gamma \ni \perp$ and a coherent sequence of color screens $\mathrm{CS}^j(\mathbf{x})$ for γ. A permutation $\sigma \in \mathbb{S}_n$ fixes $\mathbf{x} \in {}^\circ S_\gamma$ if and only if it fixes all $\mathrm{CS}^j(\mathbf{x})$. A color screen is fixed by σ exactly when these two conditions are fulfilled:

(F1) it is fixed modulo colors;
(F2) any two points of the same color go to two points of the same color, not necessarily the original one.

Let G be the isotropy subgroup at $\mathbf{x}$. A permutation $\sigma \in G$ satisfies (F1) for $\mathrm{CS}^j(\mathbf{x})$, therefore it induces the scaling of T_pX underlying $\mathrm{CS}^j(\mathbf{x})$ by a scale factor $f_j(\sigma) \in \Bbbk^\times$. The map $f_j \colon G \to \Bbbk^\times$ is a group homomorphism, thus there is a group homomorphism $(f_1, \dots, f_k) = f \colon G \to (\Bbbk^\times)^k$, and to show that it is injective suffices to complete the theorem.

Take $\sigma \in \ker f$, then σ does not move points in any of the color screens $\mathrm{CS}^j(\mathbf{x})$, $j = 1, \dots, k$. By coherence, every color in $\mathrm{CS}^j(\mathbf{x})$ is a point in $\mathrm{CS}^{j-1}(\mathbf{x})$, since both are but blocks of the partition $\pi_j \in \gamma$. Thus, σ cannot change colors either, in any $\mathrm{CS}^j(\mathbf{x})$ for $j = 2, \dots, k$. Colors in $\mathrm{CS}^1(\mathbf{x})$ stay unchanged because there is only one such.

Thus σ does not move anything at all, and there is only one such permutation. Indeed, if $\sigma \neq \mathrm{id}$ and $\sigma(a) = b$, then σ must induce non-trivial scaling on $\mathrm{CS}^l(\mathbf{x})$, where l is the maximal index j for which a and b are in the same block of $\pi_j \in \gamma$. □

Remark. This version of the original proof is one substantially simplified with a key idea due to Jean–Luc Brylinski.

References

[An] G. Andrews, The Theory of Partitions, Addison–Wesley, 1976.

[Ar] V. I. Arnold, *The cohomology ring of the colored braid group*, Mat. Zametki **5** (1969), 227–231.

[AMRT] A. Ash, D. Mumford, M. Rapoport, Y. Tai, Smooth Compactification of Locally Symmetric Spaces, Math. Sci. Press, 1975.

[At] M. Atiyah, *Convexity and commuting Hamiltonians*, Bull. London Math. Soc. **14** (1982), 1–15.

[AS] S. Axelrod, I. Singer, *Chern–Simons perturbation theory II*, J. Diff. Geom. **39** (1994), 173–213.

[BL] L. Babai, T. Lengyel, *A convergence criterion for recurrent sequences with applications to the partition lattice*, Analysis **12** (1992), 109–119.

[BG] A. Beilinson, V. Ginzburg, *Infinitesimal structure on moduli space of G-bundles*, Intl. Math. Res. Notices **4** (1992), 63–74.

[BS] L. J. Billera, A. Sarangarajan, *The combinatorics of permutation polytopes*, in: Formal Power Series and Algebraic Combinatorics, DIMACS Ser. on Discrete Math. Comp. Sci., **24** (1994), 1–25.

[Bj] A. Björner, *Subspace arrangements*, in: First Europ. Congress of Math. (Paris 1992), v. I, Progress in Math. **119**, Birkhauser, 1994, pp. 321–370.

[Br] J.–L. Brylinski, *Eventails et variétés toriques*, in: Séminaire sur les Singularités des Surfaces, Lect. Notes in Math. **777**, Springer, 1980, 247–288.

[Ch] J. Cheah, *The Hogde polynomial of the Fulton–MacPherson compactification of configuration spaces*, Amer. J. Math. **118** (1996), 963–977.

[C] F. R. Cohen, *The homology of $\mathcal{C}_{n+1}$-spaces*, in The Homology of Iterated Loop Spaces, Lect. Notes in Math. **533**, Springer, 1976, pp. 207–351.

[DKh] V. I. Danilov, A.G. Khovanskii, *Newton polyhedra and an algorithm for computing Hodge–Deligne numbers*, Math. U.S.S.R. Izvestiya **29** (1987), 279–298.

[DP] C. De Concini, C. Procesi, *Wonderful models for subspace arrangements*, Selecta Math., New ser. **1** (1995), 459–494.

[De1] P. Deligne, *Théorie de Hodge I, II, III*, in: Proc. I.C.M. 1970, v. 1, 425–430; Publ. Math. I.H.E.S. **40** (1971), 5–58; ibid. **44** (1974), 5–77.

[De2] P. Deligne, *Resumé des premiers exposés de A. Grothendieck*, in: Groupes de Monodromie en Géométrie Algébrique, SGA 7, Lect. Notes in Math. **288**, Springer, 1972, pp. 1–24.

[De3] P. Deligne, *Poids dans la cohomologie des variétés algébriques*, in: Proc. I.C.M. 1974, v. 1, 79–85.

[DL] I. Dolgachev, V. Lunts, *A character formula for the representation of a Weyl group in the cohomology of the associated toric variety*, J. Alg. **168** (1994), 741–772.

[Du] A. H. Durfee, *Algebraic varieties which are a disjoint union of subvarieties*, in: Geometry and Topology: Manifolds, Varieties and Knots, Lect. Notes in Pure Appl. Math. **105**, Marcel Dekker, 1987, pp. 99–192.

[Fa] E. Fadell, *Homotopy groups of configuration spaces and the string problem of Dirac*, Duke Math. J. **29** (1962), 231–242.

[FaN] E. Fadell, L. Neuwirth, *Configuration spaces*, Math. Scand. **10** (1962), 111–118.

[FM] W. Fulton, R. MacPherson, *A compactification of configuration spaces*, Ann. Math. **139** (1994), 183–225.

[GS] I. M. Gelfand, V. V. Serganova, *Combinatorial geometries and torus strata on homogeneous compact manifolds*, Russian Math. Surveys **42:2** (1987), 133–168.

[Ge] E. Getzler, *Mixed Hodge structures of configuration spaces*, q-alg/9510018.

[Gi] V. Ginzburg, *Resolution of diagonals and moduli spaces*, in: The Moduli Space of Curves, Progress in Math. **129**, Birkhauser, 1995, pp. 231–266.

[Hu] Y. Hu, *A compactification of open varieties*, math.AG/9910181.

[Ka1] M. M. Kapranov, *Veronese curves and Grothendieck–Knudsen moduli space* $\overline{M}_{0,n}$, J. Alg. Geom. **2** (1992), 236–262.

[Ka2] M. M. Kapranov, *Chow quotients of Grassmannians, I*, in: I. M. Gelfand Seminar, Adv. in Soviet Math. **16** (1993), 29–111.

[Ke] S. Keel, *Intersection theory of the moduli space of stable n-pointed curves*, Trans. Amer. Math. Soc. **330** (1992), 545–574.

[KKMS] G. Kempf, F. Knudsen, D. Mumford, B. Saint-Donat, Toroidal Embeddings I, Lect. Notes in Math. **339**, Springer, 1973.

[Ki] F. Kirwan, *Partial desingularization of quotients of nonsingular varieties and their Betti numbers*, Ann. Math. **122** (1985), 41–85.

[Kn] F. Knudsen, *Projectivity of the moduli space of stable curves, II: the stacks* $M_{g,n}$, Math. Scand. **52** (1983), 161–199.

[Ko] M. Kontsevich, *Feynman diagrams and low-dimensional topology*, in: First Europ. Congress of Math. (Paris 1992), v. II, Progress in Math. **120**, Birkhauser, 1994, pp. 97–121.

[Kr] I. Kriz, *On the rational homotopy type of configuration spaces*, Ann. Math. **139** (1994), 227–237.

[Le] T. Lengyel, *On a recurrence involving Stirling numbers*, Europ. J. Combin. **5** (1984), 313–321.

[Lo] J.-L. Loday, *Overview on Leibniz algebras, dialgebras and their homology*, Fields Inst. Comm. **17**, (1997), 91–102.

[MP] R. MacPherson, C. Procesi, *Making conical compactifications wonderful*, Selecta Math., New ser. **4** (1998), 125–137.

[M] Yu. I. Manin, *Generating functions in algebraic geometry and sums over trees*, in: The Moduli Space of Curves, Progress in Math. **129**, Birkhauser, 1995, pp. 401–417.

[O] T. Oda, Convex Bodies and Algebraic Geometry, Springer, 1987.

[OT] P. Orlik, H. Terao, Arrangements of Hyperplanes, Springer, 1992.

[P] C. Procesi, *The toric variety associated to Weyl chambers*, in: Mots, M. Lothaire, ed., Hermés, Paris, 1990, pp. 153–161.

[SP] N. J. A. Sloane, S. Plouffe, The Encyclopedia of Integer Sequences, Acad. Press, 1995.

[Sta1] R. Stanley, Enumerative Combinatorics, v. 1, Wadsworth & Brooks/Cole, 1986.

[Sta2] R. Stanley, *Log-concave and unimodal sequences in algebra, combinatorics and geometry*, Ann. New York Acad. Sci. **576** (1989), 500–535.

[Ste1] J. Stembridge, *Eulerian numbers, tableaux, and the Betti numbers of a toric variety*, Discrete Math. **99** (1992), 307–320.

[Ste2] J. Stembridge, *Some permutation representations of Weyl groups associated with the cohomology of toric varieties*, Adv. Math. **106** (1994), 244–307.

[Th] D. Thurston, *Integral Expressions for the Vassiliev Knot Invariants*, math.AG/9901110.

[Ton] A. Tonks, *Relating the associahedron and the permutohedron*, in: Operads, Proceedings of the Renaissance Conferences, Contemp. Math. **202** (1997), 33–36.

[Tot] B. Totaro, *Configuration spaces of algebraic varieties*, Topology **35** (1996), 1057–1067.

[U] A. Ulyanov, *Polydiagonal compactification and the permutahedra*, in preparation.

Department of Mathematics, The Pennsylvania State University
218 McAllister Building, University Park, PA 16802
E-mail address: ulyanov@math.psu.edu

Apollonian Circle Packings: Number Theory

Ronald L. Graham [1]
Jeffrey C. Lagarias [2]
Colin L. Mallows
Allan R. Wilks
AT&T Labs, Florham Park, NJ 07932-0971

Catherine H. Yan
Texas A&M University, College Station, TX 77843

(August 6, 2001 version)

ABSTRACT

Apollonian circle packings arise by repeatedly filling the interstices between mutually tangent circles with further tangent circles. It is possible for every circle in such a packing to have integer radius of curvature, and we call such a packing an *integral Apollonian circle packing.* This paper studies number-theoretic properties of the set of integer curvatures appearing in such packings. Each Descartes quadruple of four tangent circles in the packing gives an integer solution to the Descartes equation, which relates the radii of curvature of four mutually tangent circles: $x^2 + y^2 + z^2 + w^2 = \frac{1}{2}(x + y + z + w)^2$. Each integral Apollonian circle packing is classified by a certain *root quadruple* of integers that satisfies the Descartes equation, and that corresponds to a particular quadruple of circles appearing in the packing. We determine asymptotics for the number of root quadruples of size below T. We study which integers occur in a given integer packing, and determine congruence restrictions which sometimes apply. Finally, we present evidence suggesting that the set of integer radii of curvatures that appear in an integral Apollonian circle packing has positive density, and in fact represents all sufficiently large integers not excluded by congruence conditions. In a series of companion papers "Apollonian Circle Packings: Geometry and Group Theory," we investigate a variety of group-theoretic properties of these configurations, as well as various extensions to higher dimensions and other spaces, such as hyperbolic space.

Keywords: Circle packings, Apollonian circles, Diophantine equations

[1]Current address: Dept. of Computer Science, Univ. of Calif. at San Diego, La Jolla, CA 92093
[2]Work partly done during a visit to the Institute for Advanced Study.

Apollonian Circle Packings: Number Theory

1. Introduction

Place two tangent circles of radius 1/2 inside and tangent to a circle of radius 1. In the two resulting curvilinear triangles fit tangent circles as large as possible. Repeat this process for the six new curvilinear triangles, and so on. The result is Figure 1, where each circle has been labeled with its curvature—the reciprocal of its radius.

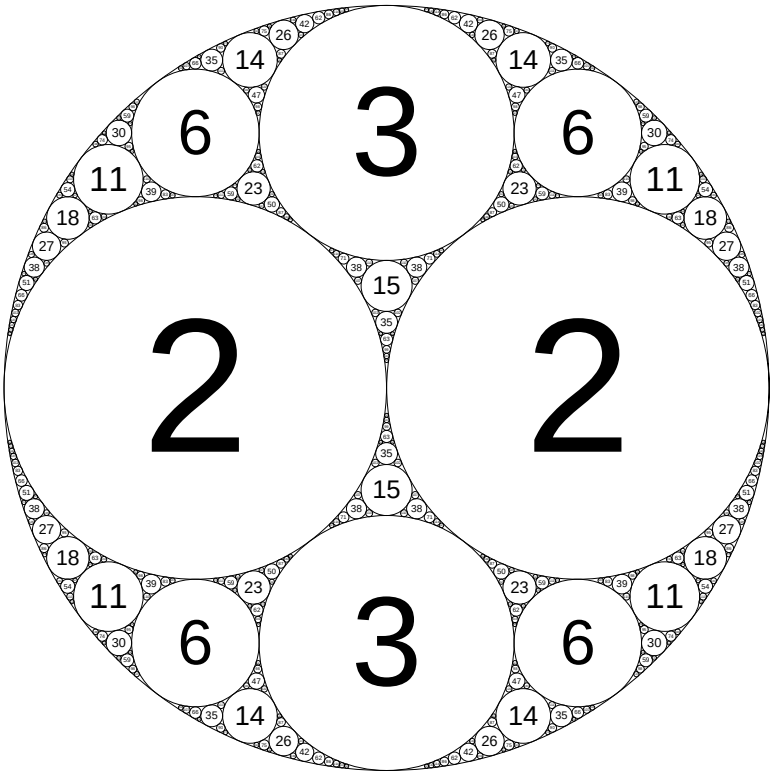


Figure 1: The integral Apollonian circle packing (-1, 2, 2, 3)

Remarkably, every circle in Figure 1 has integer curvature. Even more remarkable is that if the picture is centered at the origin of the Euclidean plane with the centers of the "2" circles on the x-axis, then each circle in the picture has the property that the coordinates of its center, multiplied by its curvature, are also integers. In this paper we are concerned with circle packings having the first of these properties; the latter property is addressed in a companion

paper [20, Section 3].

An *Apollonian circle packing* is any packing of circles constructed recursively from an initial configuration of four mutually tangent circles by the procedure above. More precisely, one starts from a *Descartes configuration*, which is a set of four mutually tangent circles with disjoint interiors, suitably defined. In the example above, the enclosing circle has "interior" equal to its exterior, and its curvature is given a negative sign. Recall that in a quadruple of mutually touching circles the curvatures $(a,\ b,\ c,\ d)$ satisfy the *Descartes equation*

$$a^2 + b^2 + c^2 + d^2 = \frac{1}{2}(a + b + c + d)^2, \tag{1.1}$$

as observed by Descartes in 1643 (in an equivalent form). Any quadruple $(a,\ b,\ c,\ d)$ satisfying this equation is called a *Descartes quadruple.* An *integral Apollonian circle packing* is an Apollonian circle packing in which every circle has an integer curvature. The starting point of this paper is the observation that if an initial Descartes configuration has all integral curvatures, then the whole packing is integral, and conversely. This integrality property of packings has been discovered repeatedly; perhaps the first observation of it is in the 1937 note of F. Soddy [44] "The bowl of integers and the Hexlet". It is discussed in some detail in Aharonov and Stephenson [1].

In this paper we study integral Apollonian circle packings viewed as equivalent under Euclidean motions, an operation which preserves the curvatures of all circles. Such packings are classified by their root quadruple, a notion defined in §3. This is the "smallest" quadruple in the packing as measured in terms of curvatures of the circles. In the packing above the root quadruple is $(-1,\ 2,\ 2,\ 3)$, where -1 represents the (negative) curvature of the bounding circle. We study the set of integers (curvatures) represented by a packing using the *Apollonian group* $\mathcal{A}$, which is a subgroup of $GL(4, \mathbb{Z})$ which acts on integer Descartes quadruples. This action permits one to "walk around" on a fixed Apollonian packing, moving from one Descartes quadruple to any other quadruple in the same packing, as shown in [19, Theorem 3.6]. The Apollonian group was introduced by Hirst [23] in 1967, who used it bounding the Hausdorff dimension of the residual set of an Apollonian packing; it was used in Söderberg,[45] and Aharonov and Stephenson [1]. Descartes quadruples associated to different root quadruples cannot be reached by the action of $\mathcal{A}$, and the action of the Apollonian group partitions the set of integer Descartes quadruples into infinitely many equivalence classes (according to which

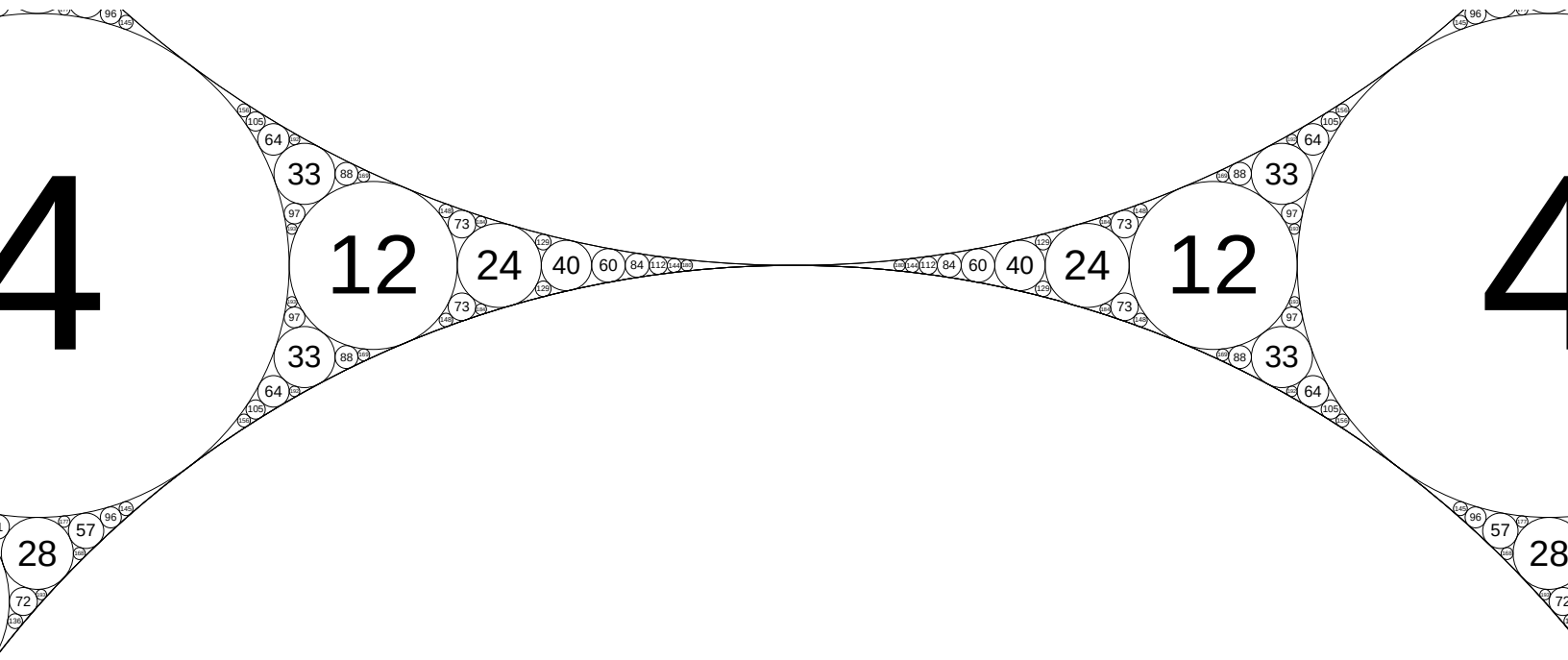
integral Apollonian packing they belong.) By scaling an integer Apollonian packing by an appropriate homothety, one may obtain a *primitive integral Apollonian packing,* which is one whose Descartes quadruples have integer curvatures with greatest common divisor 1. Thus the study of integral Apollonian packings essentially reduces to the study of primitive packings.

The simplest integral Apollonian circle packing is the one with root quadruple $(0,\ 0,\ 1,\ 1)$, which is pictured in Figure 2. This packing is special in several ways. It is degenerate in that it has two circles with "center at infinity", whose boundaries are straight lines, and it is the only primitive integral Apollonian circle packing that is unbounded. It is also the only primitive integral Apollonian circle packing that contains infinitely many copies of the root quadruple. This particular packing has already played a role in number theory. That part of the packing in an interval of length two between the tangencies of two adjacent circles of radius one, consisting of the (infinite) set of circles tangent to one of the straight lines, forms a set of "Ford circles", after shrinking all circles by a factor of two. These circles, introduced by Ford in 1916 (see [16], [17]), can be labelled by the Farey fractions on the interval $(0, 1)$ and used to prove basic results in one-dimensional Diophantine approximation connected with the Markoff spectrum, see Rademacher [37] and Nicholls [36].

In this paper our interest is in those properties of the set of integral Apollonian circle packings that are of a Diophantine nature. These include the distribution of integer Descartes quadruples, of integer root quadruples, and the representation and the distribution of the integers (curvatures) occurring in a fixed integral Apollonian circle packing. Finally we consider the size distribution of elements in the Apollonian group, a group of integer matrices associated to such packings.

To begin with, the full set of all integer Descartes quadruples (taken over all integral Apollonian packings) is enumerated by the integer solutions to the Descartes equation (up to a sign.) In §2 we determine asymptotics for the total number of integer solutions to the Descartes equation of Euclidean norm below a given bound.

In §3 we define the Apollonian group. We describe a reduction theory which multiplies Descartes quadruples by elements of this group and uses it to find a quadruple of smallest size in a given packing, called a *root quadruple.* We prove the existence and uniqueness of a root quadruple associated to each integral Apollonian packing.


4
12
24
40
60
84
112
73
97
33
88
64
105
28
57
72
96
148
129
125
120
108
100
92
85
80
76
69
65
61
53
49
45
41
37
32
29
25
21
17
13
9
5

1

In §4 we study the root quadruples of primitive integer packings. We give upper and lower bounds for the number of such quadruples having a given negative integer $-n$ as its smallest element, as $n \to \infty$. We obtain the upper bound $O(n \log n)$ and lower bound $\Omega(\frac{n}{(\log \log n)^2})$, respectively.

In §5 we study the integer curvatures appearing in a single integral Apollonian packing, counting integers with multiplicity. D. Boyd [7] showed that the number of circles occurring in a bounded Apollonian packing having curvature less than a bound T grows like $T^{\alpha+o(1)}$, where $\alpha \approx 1.30...$ is the Hausdorff dimension of the residual set of any Apollonian circle packing. This result applies to integral Apollonian packings. We observe that these integers can be put in one-to-one correspondence with elements of the Apollonian group, using the root quadruple. Using this result we show that the number of elements of the Apollonian group which have norm less than T is of order $T^{\alpha+o(1)}$, as $T \to \infty$.

In §6 we study the integer curvatures appearing in a packing, counted without multiplicity. We show that there are always nontrivial congruence restrictions (*mod* 24) on the integers that occur. We give some evidence suggesting that such congruence restrictions can only involve powers of the primes 2 and 3. We conjecture that in any integral Apollonian packing, all sufficiently large integers occur, provided they are not excluded by a congruence condition. This may be a hard problem, however, since we show that it is analogous to Zaremba's conjecture stating that there is a fixed integer K such that for all denominators $n \geq 2$ there is a rational $\frac{a}{n}$ in lowest terms whose continued fraction expansion has all partial quotients bounded by K.

In §7 we study the set of integer curvatures at "depth n" in an integral Apollonian packing, where n measures the distance to the root quadruple. There are exactly $4 \times 3^{n-1}$ such elements. We determine the maximal and minimal curvature in this set, and also formulate a conjecture concerning the asymptotic size of the median curvature as $n \to \infty$. These problems are related to the joint spectral radius of the matrix generators $\Sigma = \{S_1, S_2, S_3, S_4\}$ of the Apollonian group, which we determine.

In §8 we conclude the paper with some directions for further work and a list of open problems.

There has been extensive previous work on various aspects of Apollonian circle packings, related to geometry, group theory and fractals. The name "Apollonian packing" traces back

at least to Kasner and Supnick [25] in 1943. and has been popularized by Mandlebrot [31, p. 169ff], who observed a connection with work of Apollonius of Perga, around 200BC. Further discussion and references can be found in Aharonov and Stephenson [1] and Wilker [51]. Also see the companion papers [19], [20],[21] and [26].

2. Integral Descartes Quadruples

An Apollonian circle packing is *integral* if every circle of the packing has an integer curvature. From (1.1) it follows that if a, b, c, are given, the curvatures d, d' of the two circles that are tangent to all three satisfy

$$d,\ d' = a + b + c \pm 2q_{abc},$$

where

$$q_{abc} = \sqrt{ab + bc + ac}.$$

Hence

$$d + d' = 2(a + b + c). \tag{2.1}$$

In other words, given four mutually tangent circles with curvatures a, b, c, d, the curvature of the other circle that touches the first three is given by

$$d' = 2a + 2b + 2c - d. \tag{2.2}$$

It follows that an Apollonian packing is integral if the starting Descartes quadruple consists entirely of integers.

The relation (2.1) is the basis of the integrality property of Apollonian packings. It generalizes to n dimensions, where the curvatures X_i of a set of $n + 1$ mutually tangent spheres in $\mathbb{R}^n$ (having distinct tangents) are related to the curvatures X_0 and X_{n+2} of the two spheres that are tangent of all of these by

$$X_0 + X_{n+2} = \frac{2}{n-1}(X_1 + X_2 + \cdots + X_n).$$

This relation gives integrality in dimensions $n = 2$ and $n = 3$; the three dimensional case is studied in Boyd [5]. It even generalizes further to sets of equally inclined spheres with

inclination parameter γ, with the constant $\frac{2}{n+\frac{1}{\gamma}}$; the case $\gamma = -1$ is the mutually tangent case, cf. Mauldon [32] and Weiss [49, Theorem 3].

Definition 2.1. (i) An *integer Descartes quadruple* $\mathbf{a} = (a, b, c, d) \in \mathbb{Z}^4$ is any integer representation of zero by the indefinite integral quaternary quadratic form,

$$Q_{\mathcal{D}}(w, x, y, z) := 2(w^2 + x^2 + y^2 + z^2) - (w + x + y + z)^2,$$

which we call the *Descartes integral form*. That is, writing $\mathbf{v} = (w,\ x,\ y,\ z)^T$, we have $Q_{\mathcal{D}}(w, x, y, z) = \mathbf{v}^T Q_{\mathcal{D}} \mathbf{v}$, where

$$Q_{\mathcal{D}} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}. \tag{2.3}$$

This quadratic form has determinant $det(Q_{\mathcal{D}}) = -16$ and, on identifying the form $Q_{\mathcal{D}}$ with its symmetric integral matrix, it satisfies $Q_{\mathcal{D}}^2 = 4I$.

(ii) An integer Descartes quadruple is *primitive* if $gcd(a, b, c, d) = 1$.

I n studying the geometry of Apollonian packings ([19]- [21], [26]) we use instead a scaled version of this quadratic form, namely the —em Descartes quadratic form $Q_2 := \frac{1}{2} Q_{\mathcal{D}}$.

Definition 2.2. The size of any real quadruple $(a, b, c, d) \in \mathbb{R}^4$ is measured by the *Euclidean height* $H(\mathbf{a})$, which is:

$$H(\mathbf{a}) := (a^2 + b^2 + c^2 + d^2)^{1/2}. \tag{2.4}$$

Now let $N_{\mathcal{D}}(T)$ count the number of integer Descartes quadruples with Euclidean height at most T. We shall relate this quantity to the number $N_{\mathcal{L}}(T)$ of integer Lorentz quadruples of height at most T, where *Lorentz quadruples* are those quadruples that satisfy the Lorentz equation

$$-W^2 + X^2 + Y^2 + Z^2 = 0. \tag{2.5}$$

These are the zero vectors of the *Lorentz quadratic form*

$$Q_{\mathcal{L}}(W, X, Y, Z) = -W^2 + X^2 + Y^2 + Z^2, \tag{2.6}$$

whose matrix representation is

$$Q_{\mathcal{L}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Similarly we shall relate the number of primitive integer Descarte quadruples, denoted $N^*_{\mathcal{D}}(T)$, to the number of primitive integer Lorentz quadruples of height at most T, denoted $N^*_{\mathcal{L}}(T)$. We show that there is is a one-to-one height preserving correspondence between integer Descartes quadruples and integer Lorentzian quadruples. Introduce the matrix J_0 defined by

$$J_0 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{2.7}$$

and note that ${J_0}^2 = I$. The Descartes and Lorentz forms are related by

$$Q_{\mathcal{D}} = 2J_0^T Q_{\mathcal{L}} J_0, \tag{2.8}$$

which leads to a relation between their zero vectors.

Lemma 2.1. *The mapping* $(W,\ X,\ Y,\ Z)^T = J_0(w,\ x,\ y,\ z)^T$ *gives a bijection from the set* $(w,\ x,\ y,\ z)$ *of real Descartes quadruples to that of real Lorentz quadruples* $(W,\ X,\ Y,\ Z)$ *which preserves height. It restricts to a bijection from the set of integer Descartes quadruples to integer Lorentz quadruples, so that* $N_{\mathcal{D}}(T) = N_{\mathcal{L}}(T)$, *for all* $T > 0$, *and from primitive integer Descartes quadruples to primitive integer Lorentz quadruples, so that* $N^*_{\mathcal{D}}(T) = N^*_{\mathcal{L}}(T)$, *for all* $T > 0$.

Proof. An easy calculation shows that the mapping takes real solutions of one equation to solutions of the other and that the inverse mapping is $(w,\ x,\ y,\ z)^T = J_0(W,\ X,\ Y,\ Z)^T$, so that it is a bijection. The mapping takes integer Descartes quadruples to integer Lorentz quadruples because any integer solution to the Descartes equation satisfies $w+x+y+z \equiv 0\ (mod\ 2)$. This also holds in the reverse direction because integer solutions to the Lorentz form also satisfy $W + X + Y + Z \equiv 0\ (mod\ 2)$, as follows by reducing (2.5) $(mod\ 2)$. It is easy to check that primitive integer Descartes quadruples correspond to primitive integer Lorentz quadruples. □

Counting the number of integer Descartes quadruples of height below a given bound T is the same as counting integer Lorentz quadruples. This is a special case of the classical problem of estimating the number of representations of a fixed integer by a fixed diagonal quadratic form, on which there is an enormous literature. For example Ratcliffe and Tschantz [38] give asymptotic estimates with good error terms for the number of solutions for the equation $X^2 + Y^2 + Z^2 - W^2 = k$, of Euclidean height below a given bound, for all $k \neq 0$. (They treat Lorentzian forms in n variables.) Rather surprisingly the case $k = 0$ seems not to have been treated in the published literature. The main term in the asymptotic formula below was found in 1993 by W. Duke [14](unpublished) in the course of establishing an equidistribution result for its solutions.

Theorem 2.2. *The number of integer Descartes quadruples $N_{\mathcal{D}}(T)$ of Euclidean height at most T satisfies $N_{\mathcal{D}}(T) = N_{\mathcal{L}}(T)$, and*

$$N_{\mathcal{L}}(T) = \frac{\pi^2}{4L(2, \chi_{-4})}\ T^2 + O(T^{3/2}(logT)^3), \tag{2.9}$$

as $T \to \infty$, in which

$$L(2, \chi_{-4}) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2} \approx 0.9159.$$

The number $N^*_{\mathcal{D}}(T)$ of primitive integer Apollonian quadruples of Euclidean height less than T satisfies $N^*_{\mathcal{D}}(T) = N^*_{\mathcal{L}}(T)$ and

$$N^*_{\mathcal{L}}(T) = \frac{1}{24L(2, \chi_{-4})}\ T^2 + O(T^{3/2}(logT)^3), \tag{2.10}$$

as $T \to \infty$.

Proof. By Lemma 2.1 it suffices to estimate $N_{\mathcal{L}}(T)$. Let $r_3(m)$ denote the number of integer representations of m as a sum of three squares, allowing positive, negative and zero integers. Rewriting the Lorentz equation as $X^2 + Y^2 + Z^2 = W^2$ we obtain for integer T that

$$N_{\mathcal{L}}(\sqrt{2}T) = 1 + 2\sum_{m=1}^{T} r_3(m^2), \tag{2.11}$$

since there are two choices for W whenever $W \neq 0$. A general form for $r_3(m)$ was obtained in 1801 by Gauss [18, Articles 291-292], while in the special case $r_3(m^2)$ a simpler form holds,

given in 1906 by Hurwitz [24]. This is reformulated in Sandham [41, p. 231] in the form: if $m = \prod_p p^{e_p(m)}$, and p runs over the primes and $m_{odd} = m2^{-e_2(m)}$, then

$$\begin{aligned} r_3(m^2) &= 6m_{odd} \prod_{p\equiv 3 \pmod 4} (1 + \frac{2}{p} + ... + \frac{2}{p^{e_p(m)}}) \\ &= 6 \prod_{p\equiv 1 \pmod 2} \frac{(p^{e_p(m))+1} - 1 - (\frac{-4}{p})(p^{e_p(m))} - 1))}{p-1}. \end{aligned} \tag{2.12}$$

Sandham observes that this formula is equivalent to

$$\sum_{m=1}^{\infty} \frac{r_3(m^2)}{m^s} = 6(1 - 2^{1-s}) \frac{\zeta(s)\zeta(s-1)}{L(s, \chi_{-4})} \tag{2.13}$$

where

$$L(s, \chi_{-4}) := \sum_{m=1}^{\infty} (\frac{-4}{m}) m^{-s} = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^s}. \tag{2.14}$$

The right hand side of (2.13) is a meromorphic function in the s-plane, which has a simple pole at $s = 2$ with residue

$$c_1 = \frac{3\zeta(2)}{L(2, \chi_{-4})} = \frac{\pi^2}{2\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}},$$

and has no other poles for $\Re s > 1$. A standard contour integral argument indicates that $N_{\mathcal{L}}(\sqrt{2}T)$ will be $\frac{1}{2}c_1 T^2$ plus an error term. We can directly estimate the error term using the exact formula (2.12). We have,

$$\begin{aligned} N_{\mathcal{L}}(\sqrt{2}T) &= \sum_{1\le j\le log_2 T} \sum_{n=1}^{\lceil T/2^j \rceil} r_3((2n-1)^2) \\ &= 6 \sum_{1\le j\le log_2 T} \sum_{n=1}^{\lceil T/2^j \rceil} (2n-1) \prod_{p\equiv 3 \pmod 4} (1 + \frac{2}{p} + ... + \frac{2}{p^{e_p(2n-1)}}). \end{aligned}$$

Expanding the products above and using $\sum_{n=1}^{U}(2n-1) = U^2 + O(U)$, one obtains

$$N_{\mathcal{L}}(\sqrt{2}T) = \frac{6}{4}T^2 (\sum_{k\ge 0} 2^k \sum_{j, P_k : P_k < \sqrt{T/2^j}} 2^{-2j} P_k^{-2}) + O(T^{3/2} + \frac{1}{\sqrt{T}} \sum_{\sqrt{T}<m<T} d(m)^2 m \ ), \tag{2.15}$$

in which P_k denotes any integer of the form $p_1^{e_1}...p_k^{e_k}$ with all $p_i \equiv 3 \ (mod\ 4)$ and all $e_i \ge 1$, and any $j \ge 0$ is allowed. If the condition $P_k 2^{j/2} < \sqrt{T}$ were dropped in the first sum in

parentheses above, then it would sum to $\frac{\zeta(2)}{L(2,\chi_{-4})}$, as one sees by examining the Euler product, which is

$$(1-2^{-s})^{-1} \prod_{p \equiv 3 \pmod 4} \frac{1+p^{-s}}{1-p^{-s}},$$

evaluated at $s=2$, using $\frac{1+p^{-s}}{1-p^{-s}} = 1 + 2p^{-s} + 2p^{-2s} + ...$ The error introduced in this term by truncating at $P_k 2^{j/2} < \sqrt{T}$ is $O(\frac{1}{\sqrt{T}})$. Since[3] $\sum_{1 \le m \le T} d(m)^2 \; m = \; O(\; T \; (logT)^3)$, and since $\zeta(2) = \frac{\pi^2}{6}$, these estimates combine to give

$$N_{\mathcal{L}}(\sqrt{2}T) = \frac{\pi^2}{4L(2,\chi_{-4})} T^2 + O(T^{3/2} \; (logT)^3),$$

as claimed.

To handle the case of primitive Lorentz quadruples, we use the function $r_3^*(m)$ which counts the number of primitive integer representations of m as a sum of three squares, using positive, negative and zero integers. One has the formula $r_3(m^2) = \sum_{d \mid m} r_3^*(d^2)$, which by Möbius inversion yields

$$r_3^*(m^2) = \sum_{d \mid m} \mu(d) r_3((\frac{m}{d})^2)$$

Summing over m up to T and applying the asymptotic formula (2.9) easily yields (2.10). □

Remarks. (1) Various Dirichlet series associated to zero solutions of indefinite quadratic forms have meromorphic continuations to $\mathbb{C}$, cf. Andrianov [2]. These can used to obtain asymptotics for the number of solutions satisfying various side conditions.

(2) The real solutions of the homogeneous equation $Q_{\mathcal{L}}(w,x,y,z) = -w^2+x^2+y^2+z^2 = 0$ form the *light cone* in special relativity.

3. Reduction Theory and Root Quadruples

In this section we describe, for each Apollonian circle packing with a given Descartes quadruple in it a reduction procedure which, if it halts, identifies within it a unique Descartes quadruple (a,b,c,d) which is "minimal". This quadruple is called the *root quadruple* of the packing. This procedure always halts for integral Apollonian packings.

[3] We use the formula

$$\frac{(\zeta(s))^4}{\zeta(2s)} = \sum_n \frac{d(n)^2}{n^s},$$

see Titchmarsh and Heath-Brown [46, (1.2.10)], which has a fourth order pole at $s=1$.

Definition 3.1. The *Apollonian group* $\mathcal{A}$ is the group generated by the four integer 4×4 matrices

$$S_1 = \begin{bmatrix} -1 & 2 & 2 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad S_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & -1 & 2 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 2 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad S_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 2 & 2 & -1 \end{bmatrix}$$

As mentioned earlier, the Apollonian group was introduced in the 1967 paper of Hirst[23], and was later used in Söderberg [45] and Aharonov and Stephenson [1] in studying Apollonian packings.

We view real Descartes quadruples $\mathbf{v} = (a,\ b,\ c,\ d)^T$ as column vectors, and the Apollonian group $\mathcal{A}$ acts by matrix multiplication, sending $\mathbf{v}$ to $M\mathbf{v}$, for $M \in \mathcal{A}$. The action takes Descartes quadruples to Descartes quadruples, because $\mathcal{A} \subset Aut_{\mathbb{Z}}(Q_{\mathcal{D}})$, the set of real automorphs of the where $Q_{\mathcal{D}}$ is the Descartes integral quadratic form $Q_{\mathcal{D}}$ given in (2.3). That is, each such M satisfies

$$M^T Q_{\mathcal{D}} M = Q_{\mathcal{D}}, \qquad \text{for all } M \in \mathcal{A},$$

a relation which it suffices to check on the four generators $S_i \in \mathcal{A}$.

The elements S_j have a geometric meaning as corresponding to inversion in one of the four circles of a Descartes quadruple to give a new quadruple in the same circle packing, as explained in [19, Section 2]. That paper showed that this group with the given generators is a Coxeter group whose only relations are $S_1^2 = S_2^2 = S_3^2 = S_4^2 = I$.

The reduction procedure attempts to reduce the size of the elements in a Descartes quadruple by applying one of the generators S_i to take the quadruple $\mathbf{v} = (a,\ b,\ c,\ d)$ viewed as a column vector to the new quadruple $S_j\mathbf{v}$, until further decrease is not possible. We always suppose $|\mathbf{v}| = a + b + c + d > 0$ and for simplicity we consider the case where the quadruple is ordered $a \le b \le c \le d$. We consider which $S_i\mathbf{v}$ can decrease the sum $|\mathbf{v}| = a + b + c + d$, which it turns out is possible only using S_4, as the following lemma asserts. Note that $S_4(a,\ b,\ c,\ d)^T = (a,\ b,\ c,\ d')^T$ where $d' = 2(a + b + c) - d$.

Lemma 3.1. *Suppose that* $\mathbf{v} = (a,\ b,\ c,\ d)^T$ *is a real Descartes quadruple and set* $|\mathbf{v}| = a+b+c+d$. *Suppose that* $\mathbf{v}$ *has elements ordered* $a \le b \le c \le d$, *and set* $d' = 2(a+b+c) - d$.

(i) If $a+b+c+d > 0$, *then* $a+b \ge 0$, *with equality holding only if* $a = b = 0$ *and* $c = d$. *As a consequence, we always have* $b \ge 0$.

(ii) If $a+b+c+d > 0$, *then* $a+b+c+d' > 0$.

(iii) If $a \ge 0$, *so that* $a+b+c+d \ge 0$, *then* $d' \le c \le d$. *If* $d' < c$ *then the matrix* S_4 *that changes* d *to* d' *strictly decreases the sum* $|\mathbf{v}|$, *and it is the only generator of* $\mathcal{A}$ *that does so. If* $d' = c$ *then necessarily* $c = d = d'$ *and no generator* S_i *of* $\mathcal{A}$ *decreases* $|\mathbf{v}|$.

Proof. (i) If $a \ge 0$ then we are done, so assume $a < 0$. It is easy to check that in any real Descartes quadruple, at least three terms have the same sign. First, suppose $0 \le b \le c \le d$. Let $x = -(a+b)$, $y = -ab$. Note that $y \ge 0$. From the Descartes equation,

$$\begin{aligned} 2(x^2 + 2y + c^2 + d^2) &= (c + d - x)^2, \\ 2c^2 + 2d^2 + 2x^2 + 4y &= c^2 + d^2 + x^2 + 2cd - 2cx - 2dx, \\ (c-d)^2 + x^2 + 4y + 2cx + 2dx &= 0. \end{aligned} \tag{3.1}$$

The last equation (3.1) cannot hold if $x > 0$. If $x = 0$, then (3.1) implies $y = 0$ and $c = d$. It follows that $a = b = 0$ and $c = d$.

Now assume that $a \le b \le c \le 0 < d$. In this case, consider $(-d, -c, -b, -a)$. The preceding argument shows that $d + c \le 0$. Then $a+b+c+d \le 0$, contradicting the fact that $a+b+c+d > 0$. This completes the proof of (i).

(ii) The Descartes equation implies that

$$d, d' = a + b + c \pm 2q_{abc}, \qquad \text{where} \qquad q_{abc} = \sqrt{ab + bc + ca}.$$

We have $a+b+c+d' = 2(a+b+c) - 2\sqrt{ab+bc+ac} > 0$ because $a+b+c \ge 0$ (using (i)) and

$$(a+b+c)^2 - (ab+bc+ac) = \frac{1}{2}((a+b)^2 + (b+c)^2 + (a+c)^2) > 0.$$

(iii) The Descartes equation (1.1) gives

$$d' = a + b + c - 2\sqrt{ab + bc + ac}.$$

Thus

$$d' - c = a + b - 2\sqrt{ab + bc + ac} \leq a + b - \sqrt{4(a+b)c} \leq a + b - \sqrt{(a+b)^2} = 0.$$

If $d' < c \leq d$ then the sum $|\mathbf{v}'| = a + b + c + d' < |\mathbf{v}|$, so the sum decreases. If S_i changes c to c', then $c' = 2(a + b + d) - c \geq 2(a + b + c) - c \geq c$ because $a + b \geq 0$ by (i), so the sum $|\mathbf{v}|$ does not decrease in this case. Similarly the sum does not decrease if S_i changes b to b' or a to a'. In the case of equality $d' = c$, one easily checks that $c = d = d'$, which forces $a = b = 0$, and no S_i decreases the sum $|\mathbf{v}|$. □

Definition 3.2. A Descartes quadruple (a, b, c, d) with $a + b + c + d > 0$ is a *root quadruple* if $a \leq 0 \leq b \leq c \leq d$ and $2(a + b + c) \geq d$.

Note that the last inequality above is equivalent to the condition $d' \geq d$.

Reduction algorithm.

Input: A real Descartes quadruple $(a,\ b,\ c,\ d)$ *with* $a + b + c + d > 0$.

(1) Test in order $1 \leq i \leq 4$ *whether some* S_i *decreases the sum* $a + b + c + d$. *It so, apply it to produce a new quadruple and continue.*

(2) If no S_i *decreases the sum, halt.*

The reduction algorithm takes real quadruples as input, and is not always guaranteed to halt. The following theorem shows that when the algorithm is given an integer Descartes quadruple as input, it always halts, and outputs a root quadruple. In the algorithm, the element S_i which decreases the sum necessarily decreases the largest element in the quadruple, leaving the other three elements unchanged. The proof below establishes that in all cases where a reduction is possible, the largest element of the quadruple is unique, so that the choice of S_i in the reduction step is unique. There do exist quadruples with a tie in the largest element, such as $(0.\ 0,\ 1,\ 1)$, but the vector $(a,\ b,\ c,\ d)$ then cannot be further reduced.

Theorem 3.2. (1) *If the reduction algorithm ever encounters some element* $a < 0$, *then it will halt at a root quadruple in finitely many more steps.*

(2) *If* a, b, c, d *are integers, then the reduction algorithm will halt at a root quadruple in finitely many steps.*

(3) *A root quadruple is unique if it exists. However an Apollonian circle packing may contain more than one Descartes configuration yielding this quadruple.*

Proof.

(1) Geometrically a Descartes quadruple with $a < 0$ describes a circle of of radius $1/a$ enclosing three mutually tangent circles of radii $1/b, 1/c, 1/d$. All circles in the packing lie inside this bounding circle of radius $1/a$. Each non-halting reduction produces a new circle of radius $1/d' > 1/d$, which covers an area of π/d'^2, and this is at least π/d^2. Since there is a total area of π/a^2 which can be covered, and all circles except the one with radius $1/a$ have disjoint interiors, this process must halt in at most $\left\lfloor (\frac{d}{a})^2 \right\rfloor$ steps.

(2) Let $q_{abc} = \sqrt{ab+bc+ac} = (a+b+c-d)/2 \in \mathbb{N}$. After each reduction, the sum $a+b+c+d$ decreases by $4q_{abc}$. By Lemma 3.1, the sum $a+b+c+d$ is bounded below by 0. Therefore this process halts after finitely many steps.

(3) If (a,b,c,d) is a root quadruple of an Apollonian packing, then the numbers a,b,c,d are the curvatures of the largest circles contained in this packing, hence they are unique. On the other hand, the Apollonian packing may contain more than one copy of this quadruple, for example, $(-1,2,2,3)$ appears twice in the packing shown in Figure 1, and $(0,0,1,1)$ appears infinitely many times in the packing in Figure 1 generated by it. □

Root quadruples lead to a partition of the set $Q(\mathbb{Z})$ of all integer Descartes quadruples. This set partitions into $Q(\mathbb{Z})^+ \cup \{(0,0,0,0)\} \cup Q(\mathbb{Z})^-$, where

$$Q(\mathbb{Z})^+ = \{(a,b,c,d) \in Q(\mathbb{Z}) : a+b+c+d > 0\} \tag{3.2}$$

and $Q(\mathbb{Z})^- = -Q(\mathbb{Z})^+$. Next we have the partition

$$Q(\mathbb{Z})^+ = \bigcup_{k=1}^{\infty} kQ(\mathbb{Z})^+_{prim}\ , \tag{3.3}$$

where $Q(\mathbb{Z})^+_{prim}$ enumerates all primitive integer Descartes quadruples in $Q(\mathbb{Z})^+$. These latter are exactly the Descartes quadruples occurring in all primitive integer Apollonian packings, so we may further partition $Q(\mathbb{Z})^+_{prim}$ into a union of the sets $Q(\mathcal{P}_{\mathcal{D}})$, where $Q(\mathcal{P}_{\mathcal{D}})$ denotes the set of all Descartes quadruples in the circle packing $\mathcal{P}_{\mathcal{D}}$ having primitive root quadruple $\mathcal{D}$, i.e.

$$Q(\mathbb{Z})^+_{prim} = \bigcup_{\substack{\text{primitive root} \\ \text{quadruple } \mathcal{D}}} Q(\mathcal{P}_{\mathcal{D}}). \tag{3.4}$$

We study the distribution of root quadruples in §4 and the set of integers in a given packing $\mathcal{P}_{\mathcal{D}}$ in §5 and §6.

By definition the Apollonian group labels all the (unordered) Descartes quadruples in a fixed Apollonian packing. We now show that it has the additional property that for a given integral Apollonian packing, the integer curvatures of all circles not in the root quaduple lie in one-to-one correspondence with the non-identity elements of the Apollonian group.

Theorem 3.3. *Let $\mathcal{P}_{\mathbf{v}}$ be the integer Apollonian circle packing with root quadruple $\mathbf{v} = (a,\ b,\ c,\ d,)^T$, and suppose $a < 0$. Then the set of integer curvatures occurring in $\mathcal{P}$, counted with multiplicity, consists of the four elements of $\mathbf{v}$ plus the largest elements of each vector $M\mathbf{v}$, where M runs over all elements of the Apollonian group $\mathcal{A}$.*

Proof. Let $M = S_{i_n} \cdots S_{i_1}$ be a reduced word in the generators of $\mathcal{A}$, that is $S_{i_k} \neq S_{i_{k+1}}$ for $1 \leq k < n$. The main point of the proof is that if $\mathbf{w}^{(n)} = S_{i_n} \cdots S_i \mathbf{v}$, then $\mathbf{w}^{(n)}$ is obtained from $\mathbf{w}^{(n-1)}$ by changing one entry, and the new entry inserted is always the largest entry in the new vector. (It may be tied for largest value.) We prove this by induction on n. In the base case $n = 1$, there are four possible vectors $S_i\mathbf{v}$, whose inserted entries are $a' = 2(b+c+d) - a$, $b' = 2(a+c+d) - b$, $c' = 2(a+b+d) - c$, and $d' = 2(a+b+c) - d$, respectively. Since $a \leq b \leq c \leq d$ we have $d' \leq c' \leq b' \leq a'$, and since $\mathbf{v}$ is a root quadruple with $a \leq 0$, we have $d' \geq d$, as asserted.

For the induction step, where $n \geq 2$, there are only three choices for S_{i_n} since $S_{i_n} \neq S_{i_{n-1}}$. If the elements of $\mathbf{w}^{(n-1)}$ are labelled in increasing order as $w_1^{(n-1)} \leq w_2^{(n-1)} \leq w_3^{(n-1)} \leq w_4^{(n-1)}$, then we may choose the labels (in case of a tie for the largest element) so that $w_4^{(n-1)}$ was produced at step $n-1$, by the induction hypothesis. Thus exchanging $w_4^{(n-1)}$ is forbidden at step n, hence if $w_4^{(n)}$ denotes the new value produced at the next step, then

$$\begin{aligned} w_4^{(n)} &\geq 2(w_1^{(n-1)} + w_2^{(n-1)} + w_4^{(n-1)} - w_3^{(n-1)} \\ &\geq 2w_1^{(n-1)} + 2w_2^{(n-1)} + w_4^{(n-1)} > w_4^{(n-1)}, \end{aligned} \tag{3.5}$$

because $w_1^{(n-1)} + w_2^{(n-1)} > 0$ by Lemma 3.1(i). This completes the induction step.

The inversion operation produces one new circle in the packing, namely the new value added in the Descartes quadruple, and (3.5) shows that its curvature is $|M\mathbf{v}|_\infty$, where $|\cdot|_\infty$ is the supremum norm.

Every circle in the packing is produced in this procedure, by definition of the Apollonian group. That all words $M \in \mathcal{A}$ label distinct circles is clear geometrically from the tree structure of the packing. $\square$

4. Distribution of Primitive Integer Root Quadruples

In this section we count integer Apollonian circle packings in terms of the size of their root quadruples. Recall that a Descartes quadruple $(a,\ b,\ c,\ d)$ is a *root quadruple* if $a \le 0 \le b \le c \le d$ and $d' = 2(a+b+c) - d \ge d > 0$. It suffices to study primitive packings, i.e. ones whose integer quadruples are relatively prime. We begin with a Diophantine characterization of root quadruples.

Theorem 4.1. *Given a solution $(a,b,c,d) \in \mathbb{Z}^4$ to the Descartes equation*

$$(a+b+c+d)^2 = 2(a^2+b^2+c^2+d^2),$$

define (x, d_1, d_2, m) by

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ d_1 \\ d_2 \\ m \end{bmatrix} = \begin{bmatrix} x \\ d_1 - x \\ d_2 - x \\ -2m + d_1 + d_2 - x \end{bmatrix}. \tag{4.1}$$

Then $(x, d_1, d_2, m) \in \mathbb{Z}^4$ satisfies

$$x^2 + m^2 = d_1 d_2\ . \tag{4.2}$$

Conversely, any solution $(x, d_1, d_2, m) \in \mathbb{Z}^4$ to this equation yields an integer solution to the Descartes equation as above. In addition:

(i) The solution (a,b,c,d) is primitive if and only if $\gcd(x, d_1, d_2) = 1$.

(ii) The solution is a root quadruple with $a < 0 \le b \le c \le d$ if and only if $x < 0 \le 2m \le d_1 \le d_2$.

Proof. The first part of the theorem requires, to have $m \in \mathbb{Z}$, that $a+b+c+d \equiv 0 \pmod 2$. This follows from the Descartes equation by reduction (mod 2).

For (i), note that $\gcd(x, d_1, d_2) = \gcd(a,b,c) = \gcd(a,b,c,d)$.

For (ii), the condition $a < 0 \le b \le c \le d$ implies successively $x < 0$, $d_1 \le d_2$, $d_1 - 2x = b - a \ge 0$, and $-2m + d_1 = d - c \ge 0$. Finally the root condition $d' = 2(a+b+c) - d \ge d \ge 0$ gives $d' = 2m \ge 0$. Thus $x < 0 \le 2m \le d_1 \le d_2$. The converse implication follows similarly. $\square$

We proceed to study primitive integer root quadruples with $a = -n$, for $n \in \mathbb{Z}_{\ge 0}$. Let $N^*_{root}(n)$ denote the number of such quadruples. Theorem 4.1 shows that they are in one-to-one correspondence with the set of integer solutions (m, d_1, d_2) to

$$n^2 + m^2 = d_1 d_2 \tag{4.3}$$

$$0 \le 2m \le d_1 \le d_2 \quad \text{and} \quad \gcd(n, d_1, d_2) = 1 \ . \tag{4.4}$$

For each of $n = 0, 1, 2$, there is only one primitive root quadruple with $a = -n$, namely, $(0, 0, 1, 1)$, $(-1, 2, 2, 3)$, $(-2, 3, 6, 7)$. For $n = 3$, there are two, $(-3, 4, 12, 13)$ and $(-3, 5, 8, 8)$. As an example of a nonsymmetric integral Apollonian circle packing, Figure 3 pictures the packing $(-6,\ 11,\ 14,\ 15)$.

Table 1 below presents a list of $N^*_{root}(n)$ for small n. One easily sees that $N^*_{root}(n) \ge 1$ for all $n \ge 0$, since $(x,\ d_1,\ d_2,\ m) = (-n, 1, n^2, 0)$ in Theorem 4.1 produces the primitive root quadruple $(a,\ b,\ c,\ d) = (-n, n+1, n(n+1), n(n+1)+1)$ with $a = -n$. Table 1 and Table 2 present selected values of $N^*_{root}(n)$ for small n.

n	$N(n)$	n	$N(n)$	n	$N(n)$	n	$N(n)$	n	$N(n)$
1	1	11	4	21	10	31	9	41	11
2	1	12	6	22	7	32	9	42	18
3	2	13	4	23	7	33	14	43	12
4	2	14	5	24	10	34	9	44	14
5	2	15	6	25	6	35	10	45	14
6	3	16	5	26	7	36	14	46	13
7	3	17	5	27	10	37	10	47	13
8	3	18	7	28	10	38	11	48	18
9	4	19	6	29	8	39	14	49	15
10	3	20	6	30	10	40	10	50	11

Table 1: $N^*_{root}(n)$ for small n

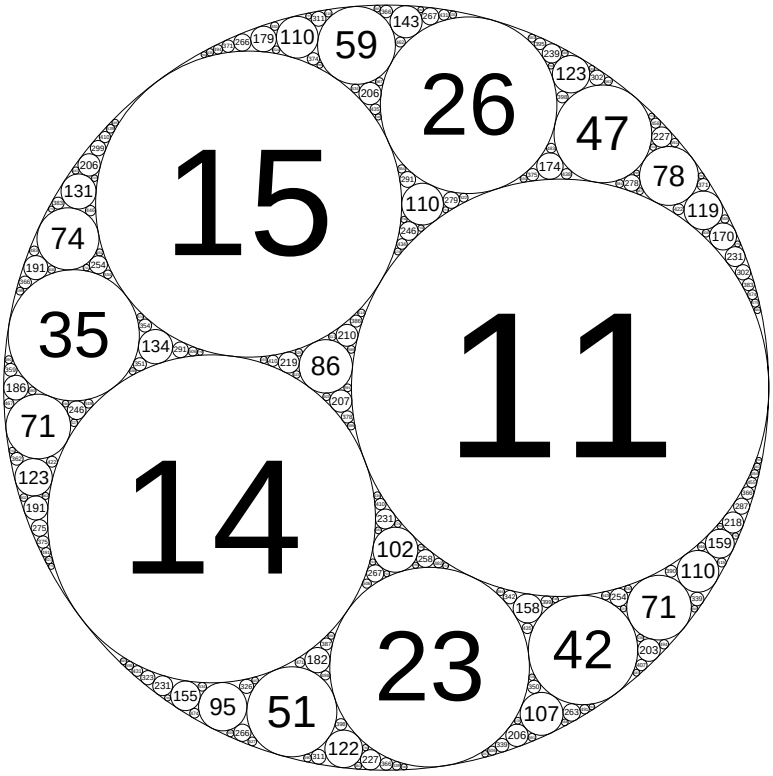


Figure 3: The nonsymmetric packing $(-6,\ 11,\ 14,\ 15)$.

n	$N(n)$	n	$N(n)$	n	$N(n)$	n	$N(n)$
1009	253	3001	751	4007	1003	5011	1254
1013	254	3011	754	4013	1004	10007	2503
2003	502	4001	1001	5003	1252	10009	2503
2011	504	4003	1002	5009	1253	20011	5004

Table 2: $N^*_{root}(n)$ for selected prime n.

4.1. Lower Bound for $N^*_{root}(n)$

We will establish:

Theorem 4.2. *The number $N^*_{root}(n)$ of primitive integer root quadruples (a,b,c,d) with $a=-n$ satisfies*

$$N^*_{root}(n) \geq \frac{1}{8}\#\{(x,y) : x^2+y^2 \leq n, \quad \textit{with} \quad \gcd(x,y)=1,\ \gcd(x^2+y^2,n)=1\}\ . \tag{4.5}$$

For $n=p$ a prime, the condition $\gcd(x^2+y^2,n)=1$ excludes at most four points in the disk $x^2+y^2 \leq n$, and one obtains

$$N^*_{root}(p) \geq \frac{3}{4\pi}p(1+o(1)) \quad \text{as} \quad p \to \infty\ ,$$

see Lemma 4.6 below. Since $\frac{3}{4\pi} \approx .237$ this lower bound compares favorably with numerical data given in Table 2, which one observes is unaccountably close to $\frac{1}{4}n$. In the general case we obtain the following bound:

Theorem 4.3. *The number $N^*_{root}(n)$ of primitive integer root quadruples (a,b,c,d) with $a=-n$ satisfies*

$$N^*_{root}(n) \geq C_0\frac{n}{(\log\log n)^2} \quad \textit{for} \quad n \geq 3\ , \tag{4.6}$$

where C_0 is a positive constant independent of n.

We prove Theorem 4.2 using two preliminary lemmas which study solutions to (4.3), (4.4) using arithmetic in the ring $\mathbb{Z}[i]$ of Gaussian integers. Afterwards we deduce Theorem 4.3.

We study solutions to (4.3) which have

$$0 \leq 2m \leq d_1 \leq n\ .$$

Since $d_1d_2 = n^2+m^2 \geq n^2$ we automatically have $d_2 \geq n \geq d_1$.

Lemma 4.4. *Given $n \geq 1$, if $\gcd(x,y)=1$ then there is exactly one integer m with $0 \leq m < x^2+y^2$ such that*

$$x+yi \mid n+mi \quad \textit{in} \quad \mathbb{Z}[i]\ . \tag{4.7}$$

Proof. The ideal $(x+yi)$ has norm x^2+y^2, and since $\gcd(x,y)=1$, x^2+y^2 is not divisible by any prime $p \equiv 3 \ (mod\ 4)$. Thus it factors over $\mathbb{Z}$ as

$$x^2+y^2 = 2^{e_2}\prod p_i^{\alpha_i},$$

where each $p_i \equiv 1 \ (\mod 4)$. A prime $p \equiv 1 \ (mod\ 4)$ has the prime ideal factorization $(p)=\pi_p\bar{\pi}_p$ in $\mathbb{Z}[i]$, and exactly one of π_p or $\bar{\pi}_p$ can divide $(x+yi)$, for if they both did then $(p)|(x+yi)$ hence $p|\gcd(x,y)$, a contradiction. Also $(2)=\pi_2^2$, so a divisor of π_2 can occur only to the power 0 or 1. Thus we have: $\gcd(x,y)=1$ if and only if the ideal $(x+yi)$ in $\mathbb{Z}[i]$ has the factorization

$$(x+yi) = \pi_2^{e_2} \prod_{p\equiv 1 (\mod 4)} \pi_p^{e_p}\bar{\pi}_p^{\bar{e}_p} \tag{4.8}$$

with $e_2 = 0$ or 1 and at least one of each e_p and $\bar{e}_p$ is zero. We conclude that when $\gcd(x,y)=1$ the ring $R := \mathbb{Z}[i]/(x+yi)\mathbb{Z}[i]$ has an additive group structure which is cyclic of order x^2+y^2, and that the residue classes $1, 2, 3, \dots, x^2+y^2$ are all distinct. Consequently the residue classes n, $n+i$, $n+2i, \dots, n+(x^2+y^2-1)i$ are all distinct, so exactly one of them is the zero class in R, which is (4.7). (More generally, an arithmetic progression $\{m+ki \ : \ 0 \le k \le t\}$ contains at most $\lceil (t+1)/(x^2+y^2)\rceil$ elements that are in the zero class in R.) $\square$

Given $n \ge 1$, we define a function which assigns to each pair (x,y) with $\gcd(x,y)=1$ the pair (d_1,m) associated to

$$n^2+m^2 = d_1d_2, \quad 0 \le m < d_1 \ ,$$

by setting $d_1 = x^2+y^2$, with m given by Lemma 4.4. We call (d_1,m) the *value* of (x,y).

Several different (x,y) may have the same value (d_1,m). In the reverse direction, we have:

Lemma 4.5. *Given $n \ge 1$, let (d_1,m) satisfy*

$$n^2+m^2 = d_1d_2, \quad 0 < m \le d_1 \ ,$$

and suppose that $\gcd(n,m,d_1)=1$. Then there are exactly four pairs (x,y) with $(x,y)=1$ which have value (d_1,m), and each pair generates the same ideal $(x+yi)$ in $\mathbb{Z}[i]$.

Proof. The ring $\mathbb{Z}[i]$ has unique factorization, so we obtain a factorization

$$n + mi = (n' + m'i) \times \quad \text{(other factors)}$$

in which $n' + m'i$ is the product of all prime ideal factors of $(n + mi)$ which have norm dividing d_1, counted with multiplicity. If $\gcd(n, m, d_1) = 1$ then $\gcd(n', m') = 1$. However any Gaussian integer $n' + m'i$ with $\gcd(n', m') = 1$ has the property that for each k with $1 \le k \le (n')^2 + (m')^2$ there is at most one ideal divisor $(x + yi)$ of $n' + m'i$ with norm

$$N(x + yi) = x^2 + y^2 = k \ .$$

(This follows from the factorization (4.8) in Lemma 4.4.) Now $\gcd(n, m, d_1) = 1$ implies that all $p \mid d_1$ have $p = 2$ or $p \equiv 1 \pmod 4$. The ideal $(n + mi)$ has a prime ideal factorization into degree one prime ideals above such primes, hence there exists a product of such ideals of norm exactly d_1, which by the above argument is unique. This gives $(x + yi) \mid (n + mi)$ with $x^2 + y^2 = d_1$. This yields four solutions (x, y), $(-x, -y)$, $(y, -x)$ and $(-y, x)$. $\square$

Proof of Theorem 4.2. Given $n \ge 1$, the conditions (4.3), (4.4) imply that $N^*_{root}(n)$ is lower bounded by the number of solutions (m, d_1, d_2) to

$$n^2 + m^2 = d_1 d_2, \quad 0 \le d_1 \le n \ ,$$

such that

(i) $\gcd(n, m, d_1) = 1$

(ii) $0 \le 2m \le d_1$.

Here we use the fact that (i) implies $(n, d_1, d_2) = 1$. By Lemma 4.5 there are exactly four pairs (x, y), $(-x, -y)$, $(y, -x)$, $(-y, x)$ which have $(x, y) = 1$ and value (d_1, m), with $x^2 + y^2 = d_1$. We claim that the four pairs $(x, -y)$, $(-x, y)$, (y, x), $(-y, -x)$ have value $(d_1, d_1 - m)$, and that $\gcd(n, d_1 - m, d_1) = 1$. To see this, note that $x + yi \mid n + mi$ implies $x - yi \mid n - mi$ by applying complex conjugation to $(x + yi)(a + bi) = n + mi$, and since $x - yi \mid x^2 + y^2 = d_1$, we obtain $x - yi \mid n + (d_1 - m)i$ and $0 \le d_1 - m \le d_1$, which proves the claim, using Lemma 4.4.

We next observe that since m and $m' = d_1 - m$ satisfy $m + m' = d_1$, at least one of them lies between 0 and $\frac{d_1}{2}$, say m for definiteness. The equality $m = \frac{d_1}{2}$ requires $d_1 = x^2 + y^2 \mid n + \frac{d_1}{2}i$,

which contradicts $\gcd(n, m, d_1) = 1$, except when $d_1 = 2$ and $m = 1$, in which case $x^2 = y^2 = 1$ and n must be odd. If $m \neq d_1/2$ then $d_1 - m > d_1/2$ and we conclude: The pairs (x, y) with $0 < x^2 + y^2 \leq n$, $\gcd(x, y) = 1$ and $\gcd(x^2 + y^2, n) = 1$ can be partitioned into groups of eight $\{(\pm x, y), (\pm x, -y), (\pm y, x), (\pm y_1, -x)\}$, four of which give a value (d_1, m) satisfying conditions (i), (ii) above, and the other four giving a value $(d_1, d_1 - m)$ which does not satisfy (i),(ii), with one exceptional case when n is odd, and $x = y = 1$, in which case the group of eight collapses to a group of four (since $x = y$), and gives a value (d_1, m) that satisfies (i),(ii). Thus each such group of eight contributes a primitive solution, and these solutions are all distinct by Lemma 4.5, so we have

$$\begin{aligned} N^*_{root}(n) &\geq \frac{1}{8}\#\{(x,y) : x^2 + y^2 \leq n, \gcd(x,y) = 1, \text{ and } \gcd(x^2 + y^2, m, n) = 1\} \\ &\geq \frac{1}{8}\#\{(x,y) : x^2 + y^2 \leq n, \gcd(x,y) = 1, \text{ and } (x^2 + y^2, n) = 1\}\ , \end{aligned} \tag{4.9}$$

as required. □

We now turn to the proof of Theorem 4.3. The major part of the proof is contained in the following lemma.

Lemma 4.6. *The function*

$$M(n) := \#\{(x,y) \in \mathbb{Z}^2 : x^2 + y^2 \leq n \text{ with } \gcd(x,y) = 1, \gcd(x^2 + y^2, n) = 1\}\,. \tag{4.10}$$

satisfies

$$M(n) = \frac{6n}{\pi}\Phi^*(n) + O(n^{19/20}) \tag{4.11}$$

where $\Phi^*(n)$ is the multiplicative function given by

$$\Phi^*(n) := \left(\frac{2}{3}\right)^{\omega_2(n)} \prod_{\substack{p|n \\ p \equiv 1 (\mathrm{mod}\ 4)}} \left(\frac{1 - \frac{1}{p}}{1 + \frac{1}{p}}\right) \tag{4.12}$$

where $\omega_2(n) = 1$ if 2 divides n, and is 0 otherwise.

Proof. We construct a Dirichlet series $G_n(s)$ whose coefficient of m^{-s} counts a constant multiple of

$$r_2^{**}(m; n) = \#\{(x,y) \in \mathbb{Z}^2 : x^2 + y^2 = m \text{ with } \gcd(x,y) = 1, \gcd(x^2 + y^2, n) = 1\}.$$

We then estimate $M(n)$ using a contour integral of $G_n(s)$ multiplied by a suitable kernel function.

Recall that the zeta function $\zeta_{\mathbb{Q}(i)}(s)$ of the Gaussian field $\mathbb{Q}(i)$ is given by

$$\begin{aligned}\zeta_{\mathbb{Q}(i)}(s) &= (1-2^{-s})^{-1}\prod_{p\equiv 1(\mathrm{mod}\ 4)}(1-p^{-s})^{-2}\prod_{p\equiv 3(\mathrm{mod}\ 4)}(1-p^{-2s})^{-1}\\ &= \sum_{n=1}^{\infty} r_2(m)m^{-s}\ ,\end{aligned}$$

where $r_2(m)$ counts the number of ordered representations (x,y) of $m=x^2+y^2$ with $x>0$, $y\geq 0$. The Dirichlet series

$$G(s)=\frac{\zeta_{\mathbb{Q}(i)}(s)}{\zeta(2s)}=\sum_{m=1}^{\infty} r_2^*(m)m^{-s}\ , \tag{4.13}$$

has the property that $r_2^*(m)$ counts[4] the number of ordered representations (x,y) of $m=x^2+y^2$ with $\gcd(x,y)=1$, and $x>0$, $y\geq 0$. For $m\geq 2$ this is exactly $\frac{1}{4}$ of the number of signed, ordered representations, since $\gcd(x,y)=1$ implies $x\neq 0$, $y\neq 0$, $x\neq y$ for $m\geq 2$. We set

$$\Phi_n(s):=\left(\frac{1}{1+2^{-s}}\right)^{\omega_2(n)}\prod_{\substack{p|n\\ p\equiv 1(\mathrm{mod}\ 4)}}\left(\frac{1-p^{-s}}{1+p^{-s}}\right)\ , \tag{4.14}$$

and then define

$$G_n(s):=\Phi_n(s)G(s)=1+\sum_{m=2}^{\infty}\frac{1}{4}r_2^{**}(m;n)m^{-s}\ . \tag{4.15}$$

We next describe the kernel function and contour integral. Following [27, Lemma 1] we let $f_1(t)=6t(1-t)$, and define the kernel function $F_{x,y}(s)$ by

$$\begin{aligned}F_{x,y}(s) &:= \frac{1}{s}\int_0^1 (x-ty)^s f_1(t)dt\\ &= \frac{-12}{y^3 s(s+1)(s+2)(s+3)}[x^{s+3}-(x-y)^{s+3}]\\ &\quad + \frac{6}{y^2 s(s+1)(s+2)}[x^{s+2}+(x-y)^{s+2}]\ .\end{aligned} \tag{4.16}$$

[4]Since $\zeta(2s)=\prod_p(1-p^{-2s})^{-1}$ we have $G(s)=(1+2^{-s})\prod_{p\equiv 1(\mathrm{mod}\ 4)}\left(\frac{1+p^{-s}}{1-p^{-s}}\right)$. Since $r_2^*(m)$ and the coefficients of the Dirichlet series for $G(s)$ are multiplicative, it suffices to check their equality on prime powers. We have $\frac{1+p^{-s}}{1-p^{-s}}=1+2p^{-s}+2p^{-2s}+\cdots$. A given $m=x^2+y^2$ with $\gcd(x,y)=1$ has $m=2^{\tilde{e}_2}\prod_{p\equiv 1(\mathrm{mod}\ 4)}p^{\tilde{e}_p}$. $\tilde{e}_2=0$ or 1, and the ideal $(x+yi)=\pi_2^{e_2}\prod_{p\equiv 1(\mathrm{mod}\ 4)}\pi_p^{e_p}\bar{\pi}_p^{\bar{e}_p}$, where one of e_p and $\bar{e}_p$ is zero and the other is $\tilde{e}_p$ (see (4.8)). So the number of representations of a prime power $p\equiv 1(\mathrm{mod}\ 4)$ is 2, as required, and the case $p=2$ is also covered.

Lemma 1 of [27] shows that, on any vertical line $\Re(s) = \sigma > 0$, the integral

$$\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty} F_{x,y}(s)u^{-s}ds = \begin{cases} 1 & \text{if} \quad 1 \le u \le x-y \ , \\ g_1\left(\frac{x-y}{y}\right) & \text{if} \quad x-y \le u \le x \ , \\ 0 & \text{if} \quad u \ge x \ , \end{cases} \tag{4.17}$$

where

$$0 \le g_1(w) \le 1 \quad \text{for} \quad 0 \le w \le 1, \tag{4.18}$$

and $g_1(w)$ is given explicitly by

$$g_1(w) := 6\int_0^w t(1-t)dt \quad \text{for} \quad 0 \le w \le 1. \tag{4.19}$$

The formula (4.16) shows that, for $\sigma = \Re(s) > 0$,

$$|F_{x,y}(s)| \le 2x^\sigma \left(\frac{x}{y}\right)|s|^{-3} \quad \text{for} \quad |s| \ge 1 \ , \tag{4.20}$$

We choose $\sigma = \frac{9}{8}$, and compute that

$$J^n_{x,y} := \frac{1}{2\pi i}\int_{9/8^{-i\infty}}^{9/8^{+i\infty}} F_{x,y}(s)G_n(s)ds = 1 + \sum_{2\le m\le x-y}\frac{1}{4}r_2^{**}(m;n) + \sum_{x-y\le m\le x}\frac{1}{4}g_1\left(\frac{x-m}{y}\right)r_2^{**}(m;n)$$

by applying (4.17) to the Dirichlet series $G_n(s)$ term-by-term, which is justified in the region of absolute convergence $\sigma > 1$. We choose $x = n$ and $0 < y \le n$, in which case the last equation with (4.17) and (4.18) yields

$$\frac{1}{4}M(n) + 1 \ge J^n_{n,y} \ge \frac{1}{4}(M(n) - M(n-y)) \ . \tag{4.21}$$

We choose $y = n^{19/20}$, in which case (4.21) yields

$$J^n_{n,y} = \frac{1}{4}M(n) + O(n^{19/20}) \ . \tag{4.22}$$

We next estimate $J^n_{n,y}$ using the contour integral along a rectangular contour $\mathcal{C}_U$ with corners at $\frac{3}{2} \pm iU$ and $\frac{3}{4} \pm iU$, oriented counterclockwise. The function $F_{x,y}(s)G_n(s)$ is analytic inside $\mathcal{C}_U$ except for a simple pole at $s = 1$. (Indeed, the functions $\zeta(2s)$ and $\Phi_n(s)$ are holomorphic in the half-plane $\Re(s) > \frac{1}{2}$, while $\zeta_{\mathbb{Q}(i)}(s)$ is holomorphic in $\mathbb{C}$ except for a simple pole at $s = 1$.) This pole comes from $\zeta_{\mathbb{Q}(i)}(s)$, which satisfies

$$\zeta_{\mathbb{Q}(i)}(s-1) = \frac{\pi/4}{s-1} + O(1) \quad \text{as} \quad s \to 1 \, .$$

Since $\Phi^*(n) = \Phi_n(1)$, we have

$$\tilde{J}^n_{n,y} := \frac{1}{2\pi i}\oint_{\mathcal{C}_U} F^n_{n,y}(s)G_n(s)ds = F^n_{n,y}(1)\left(\frac{\pi}{4}\right)\left(\frac{\pi^2}{6}\right)^{-1}\Phi^*(n)\ .$$

A computation using (4.16) yields $F_{x,y}(1) = x - \frac{y}{2}$, which implies, for $y = n^{19/20}$, that

$$\tilde{J}^n_{n,y} = \frac{3n}{2\pi}\Phi^*(n) + O(n^{19/20})\ , \tag{4.23}$$

using $0 < \Phi^*(n) \le 1$. This integral differs from $\tilde{J}^n_{n,y}$ by the contributions of five integrals: $I_0^+(U)$ over the vertical line segment $[\frac{9}{8}+iU, \frac{9}{8}+i\infty]$, $I_0^-(U)$ over $[\frac{9}{8}-i\infty, \frac{9}{8}-iU]$, $I_1(U)$ over the horizontal line segment $[\frac{9}{8}+iU, \frac{3}{4}+iU]$, $I_2(U)$ over the vertical line segment $[\frac{3}{4}+iU, \frac{3}{4}-iU]$, and $I_3(U)$ over the horizontal line segment $[\frac{3}{4}-iU, \frac{9}{8}-iU]$. We bound these integrals separately, showing for a proper choice of U that they contribute $O(n^{19/20})$ in total. We first find, using (4.17), that

$$\begin{aligned} |I_0^+(U)| &\le \int_U^\infty \left|F^n_{n,y}\left(\frac{9}{8}+it\right)\right|\left|G_n\left(\frac{9}{8}+it\right)\right|dt \\ &\ll n^{\frac{9}{8}+\frac{3}{20}}\int_U^\infty |t|^{-3}dt \ll n^{\frac{15}{8}}U^{-2} \end{aligned} \tag{4.24}$$

and the same estimate applies to $|I_0^-(U)|$. To estimate $I_2(U)$ we use the Phragmen-Lindelöf estimate

$$\left|\zeta_{\mathbb{Q}(i)}\left(\frac{3}{4}+it\right)\right| = O(|t|^{1/4+\epsilon})\ , \tag{4.25}$$

valid for $|t| \ge 1$ and any fixed $\epsilon > 0$. (This bound is the standard convexity bound applied to $\zeta_{\mathbb{Q}(i)}(s)$ and is similar to the bound for $\zeta(s)$ given in [46, p. 95], with the modification that the gamma factor $\Gamma(s)$ for $\zeta_{\mathbb{Q}(i)}(s)$ contributes the exponent 1/4.) Since $|\frac{1}{\zeta(2s)}| \le \frac{1}{\zeta(\frac{3}{2})}$ for $\Re(s) \ge \frac{3}{4}$, and since

$$\left|\Phi_n\left(\frac{3}{4}+it\right)\right| \le \frac{3}{2}\prod_{\substack{p|n\\ p\equiv 1 (\mathrm{mod}\ 4)}}\left(1-\frac{1}{p^{\frac{3}{4}}}\right)^{-2} \le \exp(C_2(\log n)^{\frac{1}{4}}) \le C_4(\epsilon)n^\epsilon$$

for any fixed $\epsilon > 0$ with suitable positive $C_2(\epsilon)$, we obtain

$$\begin{aligned} |I_2(U)| &\le n^{\frac{3}{4}+\frac{3}{20}+\epsilon}\left(\int_1^U |t|^{-3}dt + C_4\right)(C_3(\epsilon)n^\epsilon) \\ &\le C_5 n^{19/20}\ , \end{aligned} \tag{4.26}$$

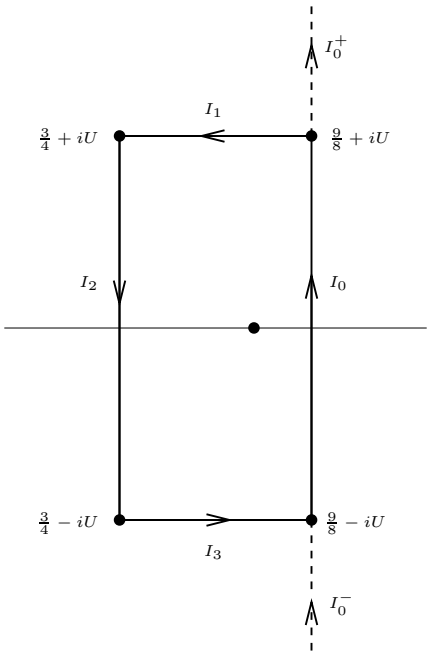


Figure 4: Contour Integrals $\mathcal{C}_U = I_0 \cup I_1 \cup I_2 \cup I_3$.

provided that we choose $\epsilon = \frac{1}{50}$, say.

We have not yet chosen U. To get a reasonable upper bound for $I_1(U)$ and $I_3(U)$ we choose $U = n$. We have the estimate

$$|\zeta_{\mathbb{Q}(i)}(\sigma + iU)| = O(|U|^{\frac{1}{2}+\epsilon}), \quad \text{for} \quad \frac{3}{4} \le \sigma \le \frac{9}{8}, \quad |U| \ge 1\,,$$

valid for any fixed positive ϵ; hence for $y = n^{19/20}$,

$$\begin{aligned} |I_1(U)| &\le \int_{3/4}^{9/8} C_5(n)^{\frac{1}{2}+\epsilon} 2n^{\sigma} \left(\frac{n}{y}\right)^3 n^{-3} d\sigma \\ &\le C_6(\epsilon) n^{\frac{1}{2}+\frac{9}{8}+\frac{3}{20}-3+\epsilon} \le \frac{C_7}{n}\,. \end{aligned} \tag{4.27}$$

A similar estimate holds for $|I_3(n)|$. Combining the estimates (4.24), (4.26), (4.27) yields

$$|J_{n,y}^n - \tilde{J}_{n,y}^n| = O\left(n^{\frac{19}{20}}\right)\,.$$

Combining this with (4.22) and (4.23) yields

$$\frac{1}{4}M(n) - \frac{6n}{4\pi}\Phi^*(n) = O\left(n^{\frac{19}{20}}\right)\,,$$

which proves the lemma. □

Proof of Theorem 4.3. We establish existence of a positive absolute constant C_0 such that

$$\Phi^*(n) \geq \frac{C_0}{(\log\log n)^2} \quad \text{for} \quad n \geq 3 \ . \tag{4.28}$$

To do this, we use

$$\Phi^*(m) \geq \frac{1}{2} \prod_{\substack{p|m \\ p\equiv 1 (\mathrm{mod}\ 4)}} \left(1 - \frac{1}{p}\right)^2 \ .$$

To minimize the right side for $m \leq n$ one takes m to be the product of the smallest primes $p = 1 \pmod 4$ that first exceed n in size. Asymptotically one takes at most $\frac{\log n}{\log\log n}(1 + o(1))$ such primes. Using Merten's theorem [22, Theorem 429], which states that

$$\prod_{p\leq x} \left(1 - \frac{1}{p}\right) \sim \frac{e^{-\gamma}}{\log x} \, ,$$

and choosing $x = \frac{\log n}{\log\log n}$ we easily obtain (4.28). Combining the lower bound (4.28) with the asymptotic formula of Lemma 4.6 finishes the proof. □

4.2. Upper Bound for $N^*_{root}(n)$

Theorem 4.7. *The number $N^*_{root}(n)$ of primitive integer root quadruples with $a = -n$ satisfies*

$$N^*_{root}(n) \leq C_1 n \log n \quad \textit{for} \quad n \geq 3 \ , \tag{4.29}$$

where C_1 is a positive constant independent of n.

Proof. The conditions (4.3), (4.4) for a primitive integer root quadruple imply that $n^2 + m^2 = d_1 d_2 \geq (2m)^2$, and hence $n^2 \geq 3m^2$. Thus $0 \leq m < n$ and since $n^2 + m^2 \leq 4n^2$, we have $d_1 \leq \sqrt{(}3)n < 2n$. Thus an upper bound for $N^*_{root}(n)$ is given by

$$\begin{aligned} N^*_{root}(n) &\leq \#\{(m, d_1, d_2) : 0 \leq m < n, \ n^2 + m^2 = d_1 d_2, \ \gcd(n, d_1, d_2) = 1 \text{ and } d_1 \leq 2n\} \\ &\leq \sum_{m=1}^{n} d^*(n^2 + m^2) \ , \end{aligned} \tag{4.30}$$

where the function $d^*(k)$ counts the number of divisors of k for which all prime factors $p \equiv 3 \pmod 4$ occur to an even power. (To see this, note that any prime $p \equiv 3 \pmod 4$ that divides d_1 divides $\gcd(m, n)$, and the condition $\gcd(n, d_1, d_2) = 1$ shows that it does not divide d_2. Such a prime divides $n^2 + m^2$ to an even power, and it necessarily divides d_1 to that same power.)

The number of divisors d_1 of n^2+m^2 with all prime factors $p \equiv 3 \pmod 4$ occurring to an even power is less than or equal to the number of distinct ideals in $\mathbb{Z}[i]$ that divide $(n+mi)$, where we take $n > 0$, $m \geq 0$. We count such ideals. Each such ideal is principal, and has a unique generator of the form $\alpha = (x+yi)z$, with $x > 0$, $y \geq 0$, $\gcd(x,y) = 1$, and $z | \gcd(m,n)$. Note that $\gcd(n+mi, n-mi) = \gcd(n,m)$ in $\mathbb{Z}[i]$, and that

$$d_1 = N(\alpha) = (x^2+y^2)z^2 \, .$$

The side condition $d_1 < 2n$ requires that $x^2+y^2 < \frac{2n}{z^2}$. The proof of Lemma 4.5 shows that the number of $0 \leq m < \frac{2n}{z^2}$ such that $x+yi$ divides $n+mi$ is at most $\left\lceil \frac{2n}{(x^2+y^2)z^2} \right\rceil$. From this we obtain

$$\sum_{m=0}^{n} d^*(n^2+m^2) \leq \sum_{z|n} \left(\sum_{\substack{(x^2+y^2)z^2 \leq 2n \\ (x,y)=1 \\ x>0, y \geq 0}} \left\lceil \frac{2n}{(x^2+y^2)z^2} \right\rceil \right) \, , \tag{4.31}$$

where we note that the left side of (4.31) requires an extra factor of 2 to count both $d_1 d_2$ and $d_2 d_1$, whereas the right side accounts for this factor since x and y are unordered in (x,y). We have

$$\left\lceil \frac{2n}{(x^2+y^2)z^2} \right\rceil \leq \frac{4n}{(x^2+y^2)z^2},$$

since $\frac{2n}{(x^2+y^2)z^2} \geq 1$. Thus (4.31) gives

$$\begin{aligned} \sum_{m=0}^{n} d^*(n^2+m^2) &\leq 2 \left(\sum_{z=1}^{\infty} \frac{1}{z^2} \right) \left(\sum_{x^2+y^2<2n} \frac{4n}{x^2+y^2} \right) \\ &\leq \frac{4\pi^2}{3} \left(\sum_{k=1}^{2n} r_2(k) \frac{n}{k} \right) \, , \end{aligned} \tag{4.32}$$

where $r_2(k)$ is the number of representations $k = x^2+y^2$ with $x > 0$, $y \geq 0$. If we set

$$M^*(k) := \#\{(x,y) : x^2+y^2 \leq k, x > 0, y \geq 0\}$$

then by partial summation

$$\begin{aligned} \sum_{k=1}^{m} \frac{r_2(k)}{k} &= \sum_{k=1}^{m} M^*(k) \left(\frac{1}{k} - \frac{1}{k+1} \right) + \frac{1}{m+1} M^*(m) \\ &= \sum_{k=1}^{m} \left(\frac{\pi k}{4} + O(k^{1/2}) \right) \frac{1}{k(k+1)} + \frac{1}{m+1} \left(\frac{\pi m}{4} + O(m^{1/2}) \right) \\ &= \frac{\pi}{4} \log m + O(1) \, , \end{aligned} \tag{4.33}$$

as $m \to \infty$. Combining this with (4.32) and (4.30) implies (4.29). We can clearly take $C_1 = \frac{\pi^3}{3} + \epsilon$ for $n > n_0(\epsilon)$. $\square$

Remark. The bounds of this section show that $N^*_{root}(n)$ grows like $n^{1+o(1)}$, so that the number of root quadruples with least element of size at most T grows like $T^{1+o(1)}$. This does not conflict with the results of §3 because size of a root quadruple as measured by its first element $a = -n$ can be quite different from its Euclidean height. Theorem 4.1 allows one to show that the Euclidean height of a root quadruple with $a = -n$ can be as large as $\Omega(n^2)$, and that this occurs when m and d_1 are both small and d_2 is of order n^2.

5. Integers Represented by a Packing: Asymptotics

In this section we study the ensemble of integer curvatures that occur in an integer Apollonian circle packing, where integers are counted with the multiplicity that they occur in the packing. Their asymptotics are known to be related to the Hausdorff dimension of the residual set of the packing, as follows from work of Boyd described in Theorem 5.2 below. At the end of the section we begin the study of the set of integer curvatures that occur, counted without multiplicity.

The *residual set* of a disk packing $\mathcal{P}$ (not necessarily an Apollonian packing) is the set remaining after all the (open) disks in the packing are removed, including any disks with "center at infinity." For a general disk packing $\mathcal{P}$, we denote the Hausdorff dimension of the residual set by $\alpha(\mathcal{P})$ and call it the *residual set dimension* of the packing. The definition of Hausdorff dimension can be found in Falconer [15], who also studies the residual sets of Apollonian packings in [15, pp. 125–131.].

The residual sets of Apollonian packings all have the same Hausdorff dimension, which we denote by α. This is a consequence of the equivalence of such residual sets under Möbius transformations (see [19, Sect. 2]), using also the fact that the Hausdorff dimension strictly exceeds one, as follows from results described below.

The *exponent* or *packing constant* $e(\mathcal{P})$ of a bounded circle packing $\mathcal{P}$ (not necessarily an Apollonian packing) is defined to be

$$e(\mathcal{P}) := \sup\{e : \sum_{C \in \mathcal{P}} r(C)^e = \infty\} = \inf\{e : \sum_{C \in \mathcal{P}} r(C)^e < \infty\},$$

in which $r(C)$ denotes the radius of the circle C. This number has been extensively studied in the literature, beginning in 1966 with the work of Melzak [34, Theorem 3], who showed that in any circle packing that covers all but a set of measure zero one has $\sum_{C\in\mathcal{P}} r(C) = \infty$. He constructed a circle packing with $e(\mathcal{P}) = 2$ and showed for Apollonian packings that $e(\mathcal{P})$ lies strictly between 1.035 and 1.99971. He conjectured that the minimal value of $e(\mathcal{P})$ is attained by an Apollonian circle packing. In 1967 J. Wilker [50] showed that all osculatory circle packings $\mathcal{P}$, which include all Apollonian circle packings, have the same exponent $e(\mathcal{P})$, which we call the *osculatory packing exponent* e. He also showed that $e \geq 1.059$. Later Boyd [3], [4], [7] improved this to $1.300 < e < 1.314$. Recent non-rigorous computations of Thomas and Dhar [47] estimate the Apollonian packing exponent to be 1.30568673 with a possible error of 1 in the last digit.

The relation between the packing exponent and the residual set dimension of Apollonian packings was resolved by an elegant result of D. Boyd [6].

Theorem 5.1. (Boyd) *The exponent e of any bounded Apollonian circle packing is equal to the Hausdorff dimension α of the residual set of any Apollonian circle packing.*

The inequality $e \geq \alpha$ follows from a 1966 result of Larman [29], and in 1973 Boyd proved the matching upper bound $\alpha \geq e$. A simpler proof of the upper bound was later given by C. Tricot [48].

Given a bounded circle packing $\mathcal{P}$ we define the *circle-counting function* $N_{\mathcal{P}}(T)$ to count the number of circles in the packing whose radius of curvature is no larger than T, i.e., whose radius is at least $\frac{1}{T}$. Boyd [7] proved the following improvement of the result above.

Theorem 5.2. (Boyd) *For a bounded Apollonian circle packing $\mathcal{P}$, the circle-counting function $N_{\mathcal{P}}(T)$ satisfies*

$$\lim_{T\to\infty} \frac{log\ N_{\mathcal{P}}(T)}{log\ T} = \alpha, \tag{5.1}$$

where α is the Hausdorff dimension of the residual set. That is, $N_{\mathcal{P}}(T) = T^{\alpha+o(1)}$ as $T \to \infty$.

. Theorem 3.3 showed that the curvatures of all circles in the packing, excluding the root quadruple, can be enumerated by the elements of the Apollonian group $\mathcal{A}$. ¿From this one

can derive a relation between the number of elements of $\mathcal{A}$ having height below a given bound T and the Hausdorff dimension α. We measure the *height* of an element $M \in \mathcal{A}$ using the Frobenius norm

$$||M||_F := (tr[M^T M])^{1/2} = (\sum_{i,j} M_{ij}^2)^{1/2}. \tag{5.2}$$

Theorem 5.3. *The number of elements $N_T(\mathcal{A})$ of height at most T in the Apollonian group $\mathcal{A}$ satisfies*

$$N_T(\mathcal{A}) = T^{\alpha + o(1)}, \tag{5.3}$$

as $T \to \infty$, where α is the Hausdorff dimension of the residual set of any Apollonian packing.

In order to prove this result, we establish two preliminary lemmas.

Lemma 5.4. *Let $M = S_{i_m} \cdots S_{i_2} S_{i_1} \in \mathcal{A}$, the Apollonian group, and suppose that $i_j \neq i_{j+1}$ for $1 \leq j \leq m-1$, and $m \geq 2$. In each row k of M,*

(i) $M_{kl} \leq 0$ if $l = i_1$,

(ii) $M_{kj} \geq |M_{kl}|$ for $l = i_1$ and $j \neq l$.

Proof. The lemma follows by induction on m. It is true for $m = 1$, since each matrix S_i has i^{th} column negative (or zero).

Suppose (i)–(ii) hold for $M' = S_{i_m} \cdots S_{i_2}$. Suppose, for convenience, that $i_1 = 1$. Then

$$M = M' S_{i_1} = \begin{bmatrix} -M'_{11} & 2M'_{11} + M'_{12} & 2M'_{11} + M'_{13} & 2M'_{11} + M'_{14} \\ -M'_{21} & 2M'_{21} + M'_{22} & 2M'_{21} + M'_{23} & 2M'_{21} + M'_{24} \\ -M'_{31} & 2M'_{31} + M'_{32} & 2M'_{31} + M'_{33} & 2M'_{31} + M'_{34} \\ -M'_{41} & 2M'_{41} + M'_{42} & 2M'_{41} + M'_{43} & 2M'_{41} + M'_{44} \end{bmatrix}.$$

Since $i_2 \neq i_1 = 1$ all $M'_{i1} \geq 0$ by (ii) of the induction hypothesis, so $M_{i1} = M'_{i1} \leq 0$ gives (i). Next, note that

$$M_{kj} = 2M'_{k1} + M'_{kj} \geq 2M'_{k1} - |M'_{kl}| \geq M'_{k1} = |M_{k1}|$$

since $M'_{kj} \geq |M'_{kj}|$ and $M'_{kj} \geq -|M'_{kl}|$ in all cases by (ii). This completes the induction step in this case. The arguments when $i_1 = 2, 3$, or 4 are similar. $\square$

Lemma 5.5. *Let* $\mathbf{v} = (a, b, c, d)^T$ *be an integer root quadruple with* $a < 0$. *Then there are positive constants* $c_0 = c_0(\mathbf{v})$ *and* $c_1 = c_1(\mathbf{v})$ *depending on* $\mathbf{v}$ *such that*

$$c_0\|M\|_F \leq |M\mathbf{v}|_\infty \leq c_1\|M\|_F, \quad \textit{for all} \quad M \in \mathcal{A} \ . \tag{5.4}$$

Proof. For the upper bound, we have

$$|M\mathbf{v}|_\infty \leq 2|M\mathbf{v}|_2 \leq 2\|M\|_F|\mathbf{v}|_2 \ , \tag{5.5}$$

so we may take $c_1 = 2|\mathbf{v}|_2$.

For the lower bound, we first show that if $M = S_{i_m} \cdots S_{i_2} S_{i_1}$ with $i_j \neq i_{j+1}$ and $i_1 = 1$, we have

$$|M\mathbf{v}|_\infty \geq \frac{1}{2}\|M\|_F \ . \tag{5.6}$$

The vector $\mathbf{v}$ has sign pattern $(-,+,+,+)$ and Lemma 5.4 shows that M has first column nonpositive elements and other columns nonnegative. Thus all terms in the product $M\mathbf{v}$ are nonnegative, and hence

$$(M\mathbf{v})_i \geq \sum_{j=1}^{4} |M_{ij}||\mathbf{v}_j| \geq \sum_{j=1}^{4} |M_{ij}| \ ,$$

because $a < 0$ implies $\min(|a|, |b|, |c|, |d|) \geq 1$. Thus

$$|M\mathbf{v}|_\infty \geq \frac{1}{4}\sum_{i,j} |M_{ij}| \geq \frac{1}{2}\|M\|_F \ .$$

It remains to deal with the cases where $i_1 = 2,\ 3$ or 4. By Theorem 3.3, the value $|M\mathbf{v}|_\infty$ gives the curvature of a particular circle in the packing, and this circle lies in one of the four lunes pictured in Figure 3 according to the value of i_m.

The bound (5.6) applies to all circles in the central lune corresponding to $i_m = 1$. For the remaining cases, we use the fact that there exists a Möbius transformation $\phi : \hat{\mathbb{C}} \to \hat{\mathbb{C}}$ with $\phi \in Aut(\mathcal{P})$, which fixes the Descartes configuration corresponding to $\mathbf{v}$ but cyclically permutes the four circles $a \to b \to c \to d$. In particular ϕ also cyclically permutes the four lunes $i_1 = 1 \to i_1 = 2 \to i_1 = 3 \to i_1 = 4$. Now ϕ maps the center of circle d to the center of circle a, which is the point at infinity, and maps the point at infinity to the center of circle b. It follows that the stretching factor of the map ϕ inside the four lunes is bounded above and

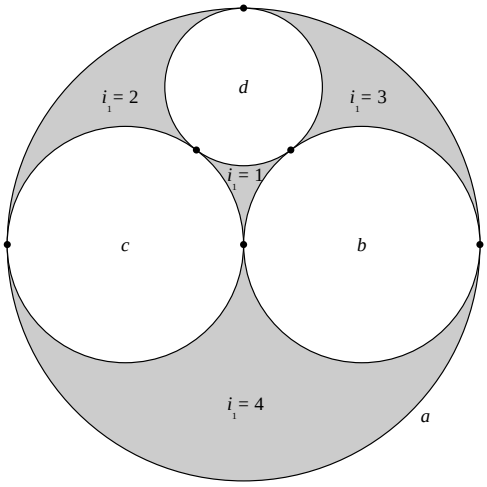


Figure 5: Four lunes of Descartes quadruple.

below by positive absolute constants c_2 and c_2^{-1}. Since ϕ maps the lune $i_1 = 4$ to $i_1 = 1$ we conclude for cases where $i_1 = 4$ that

$$|M\mathbf{v}|_\infty \geq \frac{1}{2c_2}\|M\|_F \ . \tag{5.7}$$

Applying the same argument to ϕ^2 and ϕ^3 gives the similar bound for the cases $i_1 = 3$ and $i_1 = 2$. We conclude that the lower bound in (5.4) holds with $c_0 = \frac{1}{2c_2}$. □

Proof of Theorem 5.3. Pick a fixed quadruple having $a < 0$, say $\mathbf{v} = (-1, 2, 2, 3)$, and let $\mathcal{P}_\mathbf{v}$ be the associated Apollonian packing. By Theorem 3.3, each $M \in \mathcal{A}$ corresponds to a circle of curvature $|M\mathbf{v}|_\infty$ in $\mathcal{P}_\mathbf{v}$, and all circles are so labelled except the four circles in $\mathbf{v}$. Lemma 5.5 shows that each $\|M\|_F < T$ produces a circle of curvature at most $c_1 T$. Now Theorem 5.2 asserts there are at most $T^{\alpha+o(1)}$ such circles, hence $N_T(\mathcal{A}) \leq T^{\alpha+o(1)}$. Conversely Lemma 5.5 implies that each circle of curvature $|M\mathbf{v}|_\infty \leq T$ comes from a matrix $M \in \mathcal{A}$ with $\|M\|_F \leq \frac{1}{c_0}T$. Since there are at least $T^{\alpha+o(1)}$ such circles, we obtain $N_T(\mathcal{A}) \geq T^{\alpha+o(1)}$, as desired. □

Can the estimate of Theorem 5.3 be sharpened to obtain an asymptotic formula? A. Gamburd has pointed out to us that the method of Lax and Phillips [30] might prove useful in studying this question.

We now turn to a different question: How many different integers occur, counted without multiplicity, in a given integral Apollonian circle packing $\mathcal{P}_\mathbf{v}$? This seems to be a difficult

problem. It is easy to prove that at least $cT^{1/2}$ of all integers less than T occur in a given packing. This comes from considering the largest elements of the vectors $\{(S_1S_2)^j\mathbf{v} : j = 1, 2, ...\}$, where $\mathbf{v}$ is a root quadruple, which are curvatures in the packing, by Theorem 3.3 above. These values grow like j^2 (see the example (1) in §7). Concerning the true answer to the question above, we propose the following conjecture.

Positive Density Conjecture. *Each integral Apollonian packing represents a positive fraction of all integers.*

Theorem 5.2 shows that the average number of representations of an integer n grows like $n^{\alpha-1}$, which goes rapidly to infinity as $n \to \infty$. Therefore one might guess that all sufficiently large integers are represented. However in the next section we will show there are always some congruence restrictions on which integers occur. There we formulate a stronger version of this conjecture and present numerical evidence concerning it.

6. Integers Represented by a Packing: Congruence Conditions

In this section we study congruence restrictions on the set of integer curvatures which occur in a primitive integral Apollonian packing.

We first show that there are always congruence restrictions (mod 12).

Theorem 6.1. *In any primitive integral Apollonian packing, the (unordered) Descartes quadruples (mod 12) that occur fall in one of four possible orbits, which are* $Y, 3-Y, 6+Y,$ *and* $9-Y$ *(mod 12), where*

$$Y = \{(0,0,1,1),\ (0,1,1,4), (0,1,4,9), (1,4,4,9), (4,4,9,9)\} \tag{6.1}$$

Remark. Each orbit contains only 4 different residue classes (mod 12), hence 8 residue classes (mod 12) are excluded as curvature values.

Proof. A straightforward computation, using the action of the Apollonian group (mod 12), shows that the set of all quadruples without common factors of 2 or 3 (mod 12) consists of the list below, which are grouped into eight orbits under the action of the Apollonian group (mod 12).

$$
\begin{array}{llllllll}
(1) & Y & = & (0,0,1,1) & (0,1,1,4) & (0,1,4,9) & (1,4,4,9) & (4,4,9,9); \\
(2) & 3-Y & = & (6,6,11,11) & (2,6,11,11) & (2,3,6,11) & (2,2,3,11) & (2,2,3,3); \\
(3) & 6+Y & = & (3,3,10,10) & (3,6,7,10) & (3,7,10,10) & (6,7,7,10) & (6,6,7,7); \\
(4) & 9-Y & = & (0,0,5,5) & (0,5,5,8) & (0,5,8,9) & (5,8,8,9) & (8,8,9,9); \\
(5) & -Y & = & (0,0,11,11) & (0,8,11,11) & (0,3,8,11) & (3,8,8,11) & (3,3,8,8); \\
(6) & 3+Y & = & (0,0,7,7) & (0,4,7,7) & (0,3,4,7) & (3,4,4,7) & (3,3,4,4); \\
(7) & 6-Y & = & (2,2,9,9) & (2,2,5,9) & (2,5,6,9) & (2,5,5,6) & (5,5,6,6); \\
(8) & 9+Y & = & (9,9,10,10) & (1,9,10,10) & (1,6,9,10) & (1,1,6,10) & (1,1,6,6);
\end{array}
\qquad (\text{mod } 12)
$$

To check the orbit structure is as given, note that the action of the four generators of the Apollonian group on the five elements of the orbit Y is summarized in the following transition matrix:

$$
\frac{1}{4}\begin{pmatrix}
2 & 2 & 0 & 0 & 0 \\
1 & 1 & 2 & 0 & 0 \\
0 & 1 & 2 & 1 & 0 \\
0 & 0 & 2 & 1 & 1 \\
0 & 0 & 0 & 2 & 2
\end{pmatrix}.
$$

We may view this matrix as the transition matrix of a Markov chain (after rescaling each row to be stochastic), and find that the action is transitive and the stationary distribution is $(\frac{1}{10}, \frac{1}{5}, \frac{2}{5}, \frac{1}{5}, \frac{1}{10})$. The other seven orbits have the same transition matrix and the same stationary distribution as Y.

There exist integral solutions to the Descartes equation in all of the congruence classes (mod 12) in the list above. However we recall that a Descartes quadruple (a, b, c, d) coming from an Apollonian packing satisfies the extra condition

$$a + b + c + d > 0. \tag{6.2}$$

In the rest of the proof we show that this extra condition excludes half of the orbits above, namely orbits (5)- (8).

As a preliminary, we observe that any integer solution (a, b, c, d) to the Descartes equation (1.1) yields a unique integer solution to the equation

$$4m^2 + 4a^2 + n^2 = l^2, \tag{6.3}$$

and vice-versa. Here the solution to (6.3) is given by

$$
\begin{bmatrix} a \\ n \\ l \\ m \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 2 & 1 & 1 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a \\ b-c \\ 2a+b+c \\ \frac{1}{2}(d-a-b-c) \end{bmatrix}. \tag{6.4}
$$

In the reverse direction, an integer solution to (6.3) gives one to the Descartes equation via

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & \frac{1}{2} & \frac{1}{2} & 0 \\ -1 & -\frac{1}{2} & \frac{1}{2} & 0 \\ -1 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} a \\ n \\ l \\ m \end{bmatrix} = \begin{bmatrix} a \\ \frac{1}{2}(l-2a+n) \\ \frac{1}{2}(l-2a-n) \\ 2m+l-a \end{bmatrix}. \tag{6.5}$$

Solutions to the Descartes equation satisfy a congruence (mod 2) which guarantee that the maps above take integral solutions to integral solutions, in both directions. Now (6.3) gives

$$l^2 \geq 4a^2 + m^2 \geq 2(|a| + |m|)^2 \geq (|a| + |m|)^2, \tag{6.6}$$

and equality holds if and only if $\ell = a = m = 0$. In particular, if $\ell > 0$, then (6.6) gives

$$\ell > |a| + |m|. \tag{6.7}$$

We assert that any integer solution (a, b, c, d) to the Descartes equation has

$$a + b + c + d > 0 \qquad \text{if and only if} \quad l > 0. \tag{6.8}$$

To prove this, note that if $a + b + c + d > 0$ then by Lemma 3.1 (i) we have

$$\ell = 2a + b + c = (a + b) + (a + c) \geq 0.$$

Equality can hold here only if $a = b = c = 0$, which implies $d = 0$, which contradicts the assumption $a + b + c + d > 0$. Conversely, if $\ell > 0$, then, using (6.7),

$$a + b + c + d = 2\ell + 2m - 2a \geq 2(\ell - |a| - |m|) > 0,$$

so (6.8) is proved.

Claim : No primitive integer Descartes quadruples with $a + b + c + d > 0$ occur in the orbits (5)–(8).

We prove the claim for orbit (8); the arguments to rule out orbits (5), (6), (7) are similar. We argue by contradiction. Suppose there were such a solution in orbit (8). Since the Apollonian group acts transitively on the orbit, and preserves the condition $a + b + c + d > 0$, there would be such a quadruple $(a, b, c, d) \equiv (1, 1, 6, 6) (\text{mod } 12)$. In this case $l = 2a + b + c \equiv 9 (\text{mod } 12)$ and

$$m \equiv \frac{1}{2}(6 - 6 + 1 + 1) \equiv 1 (\text{mod } 6),$$

which gives $m^2 \equiv 1 (\mathrm{mod}\ 12)$. Now (6.3) gives

$$(l+2m)(l-2m) = l^2 - 4m^2 = 4a^2 + n^2 > 0. \tag{6.9}$$

Since $a+b+c+d > 0$ we have $l > 0$ by (6.8). Then in the equation above at least one of the factors on the left side must be positive, hence they both are. Consider $l + 2m > 0$. We have

$$l + 2m \equiv 9 \pm 2 (\mathrm{mod}\ 12) \equiv 3 (\mathrm{mod} 4). \tag{6.10}$$

Consider any prime $p \equiv 3 (\mathrm{mod}\ 4)$ dividing $l + 2m$. Then it divides $4a^2 + n^2$, which it must divide to an even power, say p^{2e}, with $a \equiv n \equiv 0 (\mathrm{mod}\ p^e)$. If p also divides $l - 2m$, then it would divide both l and m, and then (6.5) would imply that it divides $\gcd(a,\ b,\ c,\ d)$, which contradicts the primitivity assumption $\gcd(a,\ b,\ c,\ d) = 1$. Therefore p does not divide $l - 2m$, and we conclude from (6.9) that $p^{2e} || l + 2m$. It follows that all primes $p \equiv 3 (\mathrm{mod}\ 4)$ that divide $l + 2m$ do so to an even power, hence we must have have $l + 2m \equiv 1 (\mathrm{mod}\ 4)$, a contradiction. This rules out orbit (8), which proves the claim in this case.

Theorem 6.1 follows from the claim. □

At the end of this section we present numberical evidence that suggests that these congruences (mod 12) are the only congruence restrictions for the integer packing $(-1, 2, 2, 3)$. However there are stronger modular restrictions (mod 24) that apply to other integer packings. For example, in the packing $(0, 0, 1, 1)$ (Fig. 2), any curvature which occurs must be congruent to $0, 1, 4, 9, 12$ or 16 (mod 24) (these are the quadratic residues modulo 24). Thus only 6 classes (mod 24) can occur rather than the 8 classes allowed by Theorem 6.1.

It seems likely that the full set of congruence restrictions possible (mod m) is attained for m a small fixed power $2^a 3^b$, perhaps even $m = 24$. We are a long way from proving this. As evidence in its favor, we prove the following result, which shows that all residue classes modulo m do occur for any m relatively prime to 30.

Theorem 6.2. *Let $\mathcal{P}$ be a primitive integral Apollonian circle packing. For any integer m with gcd$(m, 30) = 1$, every residue class modulo m occurs as the value of some circle curvature in the packing $\mathcal{P}$.*

Proof. Observe that the s-term product $W(s) = \ldots S_2S_1S_2S_1$ is

$$\begin{pmatrix} -s & s+1 & s(s+1) & s(s+1) \\ -(s-1) & s & s(s-1) & s(s-1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(where the top two rows are interchanged if s is even). Of course, the two non-trivial rows can be placed anywhere by choosing the two matrices from the set S_1, S_2, S_3, S_4 appropriately.

If $(a, b, c, d)^T$ is a quadruple in $\mathcal{P}$ then the product

$$W(s)(a, b, c, d)^T =$$

$$(-sa + (s+1)b + s(s+1)c + s(s+1)d, -(s-1)a + sb + s(s-1)c + s(s-1)d, c, d)^T \quad (6.11)$$

is also in $\mathcal{P}$ as well. Let $\mathcal{J}$ denote the set of all rows $(\alpha, \beta, \gamma, \delta)$ which can occur in a product of matrices taken from S_1, S_2, S_3, S_4. Thus, if $(\alpha, \beta, \gamma, \delta) \in \mathcal{J}$ then so are:
$(-\alpha, 2\alpha + \beta, 2\alpha + \gamma, 2\alpha + \delta)$, $(2\beta + \alpha, -\beta, 2\beta + \gamma, 2\beta + \delta)$, $(2\gamma + \alpha, 2\gamma + \beta, -\gamma, 2\gamma + \delta)$,
and $(2\delta + \alpha, 2\delta + \beta, 2\delta + \gamma, -\delta)$. Therefore,
$(-\alpha, 2\alpha + \beta, 2\alpha + \gamma, 2\alpha + \delta), (3\alpha + 2\beta, -2\alpha - \beta, 6\alpha + 2\beta + \gamma, 6\alpha + 2\beta + \delta)$,
$(-3\alpha - 2\beta, 4\alpha + 3\beta, 12\alpha + 6\beta + \gamma, 12\alpha + 6\beta + \delta), \ldots$, and in general,

$$(-r\alpha - (r-1)\beta, (r+1)\alpha + r\beta, r(r+1)\alpha + r(r-1)\beta + \gamma, r(r+1)\alpha + r(r-1)\beta + \delta) \quad (6.12)$$

are all in $\mathcal{J}$ for all r (as well as all permutations of these). Now substitute $(\alpha, \beta, \gamma, \delta) = (s(s+1), s(s+1), -s, s+1) \in \mathcal{J}$ into (6.12). This shows that the row

$$\rho = (-(2r-1)s(s+1), (2r+1)s(s+1), 2r^2s(s+1) - s, 2r^2s(s+1) + s + 1) \in \mathcal{J}. \quad (6.13)$$

The sum of the last two coordinates of ρ is

$$4r^2s(s+1) + 1 = r^2((2s+1)^2 - 1) + 1 = r^2(x^2 - 1) + 1 = u^2 - r^2 + 1 \quad (6.14)$$

where $u = rx$ and $x = 2s + 1$. Note that the g.c.d. of these two summands must divide their difference, which is $2s + 1$. It is well known (and easy to show) that for any prime power p^w with $p > 5$, in at least one of the pairs $\{1, 2\}, \{4, 5\}$ and $\{9, 10\}$ are both nonzero quadratic residues modulo p^w. For each $p \mid m$, let $\{a_p, a_p + 1\}$ denote such a pair. Define u_p and r_p so that

$$u_p^2 \equiv a_p, \qquad r_p^2 \equiv a_p + 1 \pmod{p^w} \quad (6.15)$$

where p^w is the largest power of p dividing m. Since $\gcd(r_p, p) = 1$ then we can define $x_p \equiv u_p r_p^{-1} \pmod{p^w}$. We can guarantee that x_p is odd by adding a multiple of p_w if necessary. Hence, for these choices, the expression in (6.14) is 0 modulo p^w, i.e.,

$$r_p^2(x_p^2 - 1) + 1 \equiv 0 \pmod{p^w}. \tag{6.16}$$

Of course, we can use the values $r_p + kp^w$ and $x_p + lp^w$ in place of r_p and x_p in (6.16) for any k and l. Note that $\gcd(x_p, p) = 1$. Letting p range over all prime divisors of m, then by the Chinese Remainder Theorem, there exist X (odd) and R such that

$$R^2(X^2 - 1) + 1 \equiv 0 \pmod{p^w} \tag{6.17}$$

for all $p^w | \ m$. Thus,

$$R^2(X^2 - 1) + 1 \equiv 0 \pmod{m}, \quad gcd(X, m) = 1. \tag{6.18}$$

Hence, by (6.13) and (6.14) we can find a row modulo m in $\mathcal{J}$ of the form $(C, D, A, -A) \pmod{m}$ where it easy to check that $\gcd(A, m) = 1$.

We can now apply the transformation preceding (6.12) to $(C, D, A, -A)$ to get the following rows modulo m in $\mathcal{J}$:

$$\begin{aligned}
(C, &\quad D, &\quad A, -A) &\pmod{m} \\
(2A + C, &\quad 2A + D, &\quad -A, A) &\pmod{m} \\
(4A + C, &\quad 4A + D, &\quad A, -A) &\pmod{m} \\
(6A + C, &\quad 6A + D, &\quad -A, A) &\pmod{m} \\
&\cdots
\end{aligned}$$

and more generally

$$(4tA + C, 4tA + D, A, -A) \pmod{m} \in \mathcal{J} \text{ for all } t \geq 0. \tag{6.19}$$

Suppose for the moment (and we will prove this shortly) that we can find $(a, b, c, d)^T \in \mathcal{P}$ with $\gcd(a + b, m) = 1$. Taking the inner product of the row in (6.19) with $(a, b, c, d)^T$, we get the curvature value

$$(4tA + C, 4tA + D, A, -A) \cdot (a, b, c, d)^T \pmod{m} \tag{6.20}$$

$$\equiv 4A(a + b)t + Ca + Db + Ac - Ad \pmod{m}. \tag{6.21}$$

Since $\gcd(4A(a+b), m) = 1$ then these values range over a complete residue system modulo m as t runs over all positive integers. The proof will be complete now if we can establish the following result.

Claim. If $\mathcal{P}$ is a primitive packing then for any odd $m \geq 1$, there exists $(a, b, c, d)^T \in \mathcal{P}$ with $\gcd(a + b, m) = 1$.

Proof of Claim. First recall that for any $(a, b, c, d) \in \mathcal{P}$, we have $\gcd(a, b, c) = 1$. We have also seen by (6.11), if $(a, b, c, d) \in \mathcal{P}$ then for any $r > 1$,

$$(A(r), B(r), C(r), D(r)) :=$$
$$(-ra + (r+1)b + r(r+1)(c+d), -(r-1)a + rb + r(r-1)(c+d), c, d) \in \mathcal{P} \tag{6.22}$$

as well. Define $q(r)$ to be the sum of the first two components of this vector:

$$q(r) := A(r) + B(r) = 2(c+d)r^2 - 2(a-b)r + a + b.$$

Let p denote a fixed odd prime. We show that

$$q(r) \not\equiv 0 \pmod{p} \quad \text{for some} \quad r \geq 1. \tag{6.23}$$

Suppose to the contrary that $q(r) \equiv 0 \pmod{p}$ for all r. Thus,

$$\begin{aligned} q(0) &\equiv a + b \equiv 0 \pmod{p} \\ q(1) &\equiv 2(c+d) - 2(a-b) + (a+b) \equiv 2(c+d) - a + 3b \equiv 0 \pmod{p} \\ q(2) &\equiv 8(c+d) - 4(a-b) + (a+b) \equiv 8(c+d) - 3a + 5b \equiv 0 \pmod{p} \end{aligned}$$

which implies $a \equiv b \equiv c + d \equiv 0 \pmod{p}$. However, since
$a = b + c + d \pm 2\sqrt{b(c+d) + c\ d}$ then $cd \equiv 0 \pmod{p}$, i.e., $c \equiv 0$ or $d \equiv 0 \pmod{p}$.
This would imply that $\mathcal{P}$ is not primitive, a contradiction which establishes (6.23).

To finish proving the claim, for each $p|m$ let r_p satisfy $q(r_p) \not\equiv 0 \pmod{p}$. Then we have

$$q(r_p + kp) \equiv q(r_p) \not\equiv 0 \pmod{p}$$

for all $k \geq 0$. By the Chinese Remainder Theorem one can find R and S such that $q(R + kS) \not\equiv 0 \pmod{p}$ for all $p \mid m$ and all k. In particular, $\gcd(q(R), m) = \gcd(A(R) + B(R), m) = 1$, and the Claim is proved. $\square$

$n \equiv 3 \pmod{12}$

159	207	243	435	603	711	1923	2175	2319	3711
4167	4959	4995	5283	6015	6879	7863	10095	10923	11295
12063	16311	16515	18051	19815	21135	23175	28323	41655	48075
68055	97287								

$n \equiv 6 \pmod{12}$

78	246	342	834	1422	2010	2022	2454	2718	2766
3150	3402	3510	3774	4854	6018	6666	7470	10638	12534
13154	13206	20406	24270	32670	42186	45258	55878		

$n \equiv 2 \pmod{12}$

13154

Table 3: Missing integers in the packing $(-1, 2, 2, 3)$ up to 10^6

Which integers occur as curvatures, when the congruence conditions are taken into account? We consider numerical data for two cases. The first case is the packing with root quadruple $(-1, 2, 2, 3)$, where Theorem 6.1 permits only values $2, 3, 6$, or 11 (mod 12). Not all such integers appear in the Apollonian packing $(-1, 2, 2, 3)$, for example in the class 6 (mod 12) the value 78 is missed. In Table 3 we present the missing values in these residue classes for the first million integers. Only 61 integers congruent to $2, 3$, or 6 do not occur in the packing $(-1, 2, 2, 3)$, the largest being 97287 (see Table 3), and no integers 11 (mod 12) are missed. This data suggests that there are finitely many missing values in total, with 97287 being the largest one.

Our second example is the packing with root quadruple $(0, 0, 1, 1)$. As mentioned above, there are congruence conditions (mod 24) in this case. Table 4 presents numerical data on exceptional values for the allowed congruence classes (mod 24) up to $T = 10^7$. There is a much larger set of exceptional values, and it appears more equivocal whether the full list of exceptional values is finite. However we think it is.

The numerical examples above support the idea that for any fixed integer Apollonian packing and for sufficiently large integers a finite list of congruence conditions will be the only obstruction to existence. We therefore propose the following strengthening of the Density Conjecture.

$n \equiv 0 \pmod{24}$

48	120	360	528	552	720	888	912	1080
1176	1272	1392	1560	1704	1848	1968	2184	2208
2736	2880	3240	3408	3552	4080	4392	4464	4584
4680	4896	5040	5088	5760	6192	6888	7272	8280
8880	9792	10680	10920	10944	11760	11928	13152	14160
14328	16008	17160	17232	17520	18000	19320	20712	23160
25896	26472	26760	27552	27600	27768	29424	29688	30288
31440	34440	34488	35232	36408	36648	36816	37968	38928
39168	43056	43392	45240	46056	50448	52800	58728	59400
66120	74976	80280	82200	87192	93216	96912	96960	107016
108240	117480	121680	133392	137280	138360	165360	201480	399000
424560	496080							

$n \equiv 12 \pmod{24}$

132	252	300	468	636	780	1140	1476	1572
1980	2100	2148	2628	2820	2868	3012	3492	3828
3900	4212	4692	5028	5148	5340	5796	6516	6684
6900	7380	7908	8772	10020	10212	10260	10380	10548
11268	11868	12876	13572	14100	14244	14724	14916	15300
15588	19260	19620	20940	21732	22908	23652	24252	24804
25140	25812	26100	26124	27660	28860	29532	30540	31092
31932	36564	37908	38772	39780	41460	41964	44988	46980
52260	52788	61596	67308	69324	69420	75900	76908	79740
88140	101940	120300	135252	185580	188748	220308	228780	234660
354540	422820	472548	926820	1199820				

$n \equiv 1, 4 \text{ or } 9 \pmod{24}$

241	340	748	2980	5452	11380	45652	16617	21825

$n \equiv 16 \pmod{24}$

208	328	712	1168	2488	3400	5200	13600	15088	116896

Table 4: Missing integers in the packing $(0, 0, 1, 1)$, up to 10^7

Strong Density Conjecture. *In any primitive integral Apollonian packing, all sufficiently large integers occur, provided they are not excluded by congruence conditions.*

In further support of the Strong Density Conjecture, we note an analogy to a number-theoretic conjecture of Zaremba [52], who conjectured that there exists an absolute constant b (possibly $b = 5$) such that each sufficiently large positive integer can be represented by some continuant with digits bounded above by b. In other words, given any integer $m > 1$, there exists an integer $a < m$ (a relatively prime to m) such that the simple continued fraction $[0, c_1, \cdots, c_r] = a/m$ has partial denominators $c_i \leq b$. Fix b, and let M be the set of all pairs (a, m) with the above property. There is a linear recurrence for the pairs (a, m) which is similar to that of the Descartes quadruples, since if the terms in the continued fraction of a/m are bounded by b, then so are those for the fractions $1/(i + a/m)$, $i = 1, 2, \ldots, b$. Zaremba's conjecture is saying that all the integers m will appear in some pair of M. This conjecture is currently still open. But as in the Apollonian packing, consideration of the Hausdorff dimension of the set $E_b = \{a/m : (a, m) \in M\}$ is suggestive. Namely, let $S_b(m)$ be the number of a's such that $(a, m) \in M$. If $S_b(m) \sim m^\beta$, then $\sum m^\beta m^{-x}$ converges iff $x \geq \beta + 1$. Since the abscissa of convergence of the series $\sum S_b(m)m^{-x}$ is equal to twice the Hausdorff dimension γ of E_b (see T. Cusick [12]), then $\beta = 2\gamma - 1 \approx .0624 > 0$. Thus the "expected" number of appearances of m in the pairs of M is $m^\beta \gg 1$.

7. The Growth of Descartes Quadruples in a Packing

The circles in an integral Apollonian circle packing, starting from the root quadruple, are enumerated by the elements of the Apollonian group. The graph of this group is a rooted infinite tree with four edges meeting each vertex, with each vertex labelled by a nontrivial word in the generators of the Apollonian group. (Such a word satisfies the condition that any two adjacent generators in the word are unequal.) Starting from the root node, there are 4 nodes at depth 1, and at each subsequent level there are three choices of generators at each node, so there are $4 \times 3^{n-1}$ words of length n labelling depth n circles. How are the curvatures of the circles at depth n distributed? We consider the maximum value, the minimum value, and the median value. In the process we also determine the joint spectral radius of the generators of the Apollonian group.

We begin with the maximum value. We define for $n = 4m + i$ with $0 \le i \le 3$, the reduced word T_n of length n given by

$$T_n := T_i(S_4S_3S_2S_1)^m, \tag{7.1}$$

with $T_i = I, S_1, S_2S_1, S_3S_2S_1$ for $0 \le i \le 3$, respectively.

Theorem 7.1. *Let* $\mathbf{v} = (a, b, c, d)$ *be any root quadruple with* $a \le b \le c \le d$ *and* $a < 0$, $a + b + c + d > 0$. *Then for any reduced word* W *of length* n *in the generators* $\{S_1, S_2, S_3, S_4\}$ *of the Apollonian group,*

$$|W\mathbf{v}|_\infty \le |T_n\mathbf{v}|_\infty \ . \tag{7.2}$$

Proof. Write $W = S_{i_n}S_{i_{n-1}} \cdots S_{i_1}$ and set $\mathbf{w}^{(n)} = W\mathbf{v}$ and $\mathbf{v}^{(n)} = T_n\mathbf{v}$. Write the elements of $\mathbf{w}^{(n)}$ and $\mathbf{v}^{(n)}$ in increasing order as

$$w_1^{(n)} \le w_2^{(n)} \le w_3^{(n)} \le w_4^{(n)} \quad \text{and} \quad v_1^{(n)} \le v_2^{(n)} \le v_3^{(n)} \le v_4^{(n)} \ .$$

The idea of the proof is that T_n always inverts with respect to the circle of smallest curvature, and in fact produces the largest curvature vector in a strong lexicographic sense. More precisely, we prove by induction on $n \ge 1$ that

$$w_i^{(n)} \le v_i^{(n)} \quad \text{for} \quad 1 \le i \le 4 \tag{7.3}$$

and

$$w_4^{(n)} - w_1^{(n)} \le v_4^{(n)} - v_1^{(n)} \ . \tag{7.4}$$

For the base case $n = 1$, we have $\mathbf{v}^{(1)} = (a', b, c, d)$ where $a' = 2(b + c + d) - a = |S_1\mathbf{v}|_\infty$. If $b' = 2(a + c + d) - b = |S_2\mathbf{v}|_\infty$ and $c' = |S_3\mathbf{v}|_\infty$, $d' = |S_4\mathbf{v}|_\infty$ then $a \le b \le c \le d$ gives $d' \le c' \le b' \le a'$, and (7.3) holds for $n = 1$, since $d' \ge d$ because $\mathbf{v}$ is a root quadruple.

For the induction step, a reduced word has $i_n \ne i_{n-1}$. The forbidden move $S_{i_{n-1}}$ is the one that replaces $w_4^{(n-1)}$ with $2(w_1^{(n-1)} + w_2^{(n-1)} + w_3^{(n-1)}) - w_4^{(n-1)}$. Now the induction hypothesis gives

$$\begin{aligned} w_1^{(n)} &\le w_2^{(n-1)} \le v_2^{(n-1)} = v_1^{(n)} \\ w_2^{(n)} &\le w_3^{(n-1)} \le v_3^{(n-1)} = v_2^{(n)} \\ w_3^{(n)} &\le w_4^{(n-1)} \le v_4^{(n-1)} = v_3^{(n)} \end{aligned}$$

and

$$\begin{aligned} w_4^{(n)} &\le 2(w_2^{(n-1)} + w_3^{(n-1)} + w_4^{(n-1)}) - w_1^{(n-1)} \\ &\le 2(w_2^{(n-1)} + w_3^{(n-1)}) + w_4^{(n-1)} + (w_4^{(n-1)} - w_1^{(n-1)}) \\ &\le 2(v_2^{(n-1)} + v_3^{(n-1)}) + v_4^{(n-1)} + (v_4^{(n-1)} - v_1^{(n-1)}) \\ &= v_4^{(n)} \ . \end{aligned}$$

For the remaining inequality, suppose first that $w_1^{(n)} = w_2^{(n-1)}$. Then

$$\begin{aligned} w_4^{(n)} - w_1^{(n)} &= [2(w_2^{(n-1)} + w_3^{(n-1)} + w_4^{(n-1)}) - w_1^{(n-1)}] - w_2^{(n-1)} \\ &= w_2^{(n-1)} + 2w_3^{(n-1)} + w_4^{(n-1)} + (w_4^{(n-1)} - w_1^{(n-1)}) \\ &\le v_2^{(n-1)} + 2v_3^{(n-1)} + v_4^{(n-1)} + (v_4^{(n-1)} - v_1^{(n-1)}) \\ &= v_4^{(n)} - v_1^{(n)} \ . \end{aligned}$$

If, however, $w_1^{(n)} = w_1^{(n-1)}$, then

$$\begin{aligned} w_4^{(n)} - w_1^{(n)} &\le [2(w_1^{(n-1)} + w_3^{(n-1)} + w_4^{(n-1)}) - w_2^{(n-1)}] - w_1^{(n-1)} \\ &\le 2(w_2^{(n-1)} + w_3^{(n-1)} + w_4^{(n-1)}) - w_1^{(n-1)} - w_2^{(n-1)} \\ &\le v_4^{(n)} - v_1^{(n)} \ , \end{aligned}$$

using the previous inequality. This completes the induction step. □

The maximum growth rate of the elements at level n of the Apollonian group is also describable in terms of the joint spectral radius of the generators $\{S_1, S_2, S_3, S_4\}$ of the Apollonian group.

Definition 7.1. Given a finite set of $n \times n$ matrices $\Sigma = \{M_1, \ldots, M_s\}$ the *joint spectral radius* $\sigma(\Sigma)$ is

$$\sigma(\Sigma) := \limsup_{k \to \infty} \left\{ \max_{1 \le i_1, \cdots, i_k \le s} \sigma(M_{i_1} \cdots M_{i_k})^{1/k} \right\} \ ,$$

where $\sigma(M) := \max\{|\lambda| : \lambda \text{ eigenvalue of } M\}$ is the spectral radius of M.

The notion of joint spectral radius has appeared in many contexts, including wavelets and fractals; see Daubechies and Lagarias [13] for a discussion and references. In general it is hard to compute, but here we can obtain an explicit answer.

Theorem 7.2. *The joint spectral radius for the generators $\Sigma = \{S_1, S_2, S_3, S_4\}$ of the Apollonian group is $\sigma(\Sigma) = \theta^{1/4}$ where*

$$\theta = \frac{1}{2}\left(1 + \sqrt{5} + \sqrt{2 + 2\sqrt{5}}\right) \approx 2.890 \ . \tag{7.5}$$

It is attained by $M = S_4S_3S_2S_1$.

Proof. Pick a fixed root quadruple with $a < 0$, say $\mathbf{v} = (-1, 2, 2, 3)$, and consider the associated packing $\mathcal{P}_{\mathbf{v}}$. Lemma 5.5 asserts that

$$c_0\|M\|_F \le |M\mathbf{v}|_\infty \le c_1\|M\|_F, \quad \text{all} \quad M \in \mathcal{A} \ . \tag{7.6}$$

We use the well-known fact that, for any real $n \times n$ matrix M,

$$\sigma(M) = \lim_{k\to\infty} \|M^k\|_F^{1/k} \ . \tag{7.7}$$

Now (7.6) gives for any reduced word $M = S_{i_s} \cdots S_{i_2}S_{i_1} \in \mathcal{A}$ with $i_k \neq i_{k-1}$ that

$$\sigma(M)^{1/s} = \lim_{k\to\infty} (|M^k\mathbf{v}|_\infty)^{\frac{1}{ks}} \ ,$$

Choosing $k = 4n$, Theorem 7.1 yields

$$\sigma(M)^{1/s} \le \lim_{n\to\infty} |T_{4ns}\mathbf{v}|_\infty^{\frac{1}{4ns}} \ .$$

Since $T_{4ns} = (S_4S_3S_2S_1)^{ns}$, this gives

$$\begin{aligned} \sigma(M) &\le \lim_{n\to\infty} |(S_4S_3S_2S_1)^{ns}\mathbf{v}|_\infty^{\frac{1}{4ns}} \\ &\le \sigma(S_4S_3S_2S_1)^{1/4} \ . \end{aligned}$$

Choosing $M = S_4S_3S_2S_1 \in \mathcal{A}$ attains equality (with $s = 4$), which determines the joint spectral radius. A computation reveals that the characteristic polynomial of $M = S_4S_3S_2S_1$ is $X^4 - 2X^3 - 2X^2 - 2X + 1 = 0$ which factors as

$$(X^2 + (-1 + \sqrt{5})X + 1)(X^2 - (1 + \sqrt{5})X + 1) = 0 \ .$$

Its spectral radius is given by (7.5). $\square$

The minimal growth rate of any reduced word of length $2n$ is attained by the word $W_{2n} = (S_4S_3)^n$. If $\mathbf{v} = (a, b, c, d)$ is a root quadruple with $a < 0$, then

$$|W_{2n}\mathbf{v}|_\infty = n(n + 1)(a + b) - nc + (n - 1)d \tag{7.8}$$

which grows quadratically with n. We omit the easy proof of this fact.

To conclude this section, we consider the "average value" of $|W\mathbf{v}|_\infty$ over all reduced words W in $\mathcal{A}$ of length n, which we define to be the *median* of this distribution. (The elements of the distribution are exponentially large, so the median is a more appropriate quantity to consider than the mean value.) Let T_n denote the median. We expect that its growth rate should be related to the Hausdorff dimension α of the limit set of the Apollonian packing. The results of §5 lead to the heuristic that one should expect

$$\lim_{n\to\infty} \frac{1}{n} \log T_n = \frac{\log 3}{\alpha} \ . \tag{7.9}$$

We leave the proof (or disproof) of this as an open problem.

8. Open questions

There remain many open questions concerning integral Apollonian circle packings. We list a few of these here.

(1) In any primitive integral Apollonian packing $\mathcal{P}$, just four distinct residue classes modulo 12 can occur as curvature values in $\mathcal{P}$. For example, for $\mathcal{P} = (0, 0, 1, 1)$, these values are $0, 1, 4, 9 \pmod{12}$ while for $\mathcal{P} = (-1, 2, 2, 3)$, they are $2, 3, 6, 11 \pmod{12}$. As we noted in Section 6, it seems likely that in the packing $(-1, 2, 2, 3)$, all sufficiently large integers congruent to $2, 3, 6$ and $11 \pmod{12}$ actually do occur. However, in the packing $(0, 0, 1, 1)$, instead of the 8 residue classes $0, 1, 4, 9, 12, 13, 16, 22 \pmod{24}$ which we might expect to occur, the classes 13 and 22 are completely missing. Again, computation suggests that only finitely many values in the other 6 are missing in $(0, 0, 1, 1)$. Is it true that in any integral Apollonian packing, the only congruence restrictions on the curvature values are for the modulus 24? However, in no case can we even show that the set of values which do occur has positive upper density.

(2) Is there a direct way for determining the root quadruple to which a given Descartes quadruple belongs? The only way we currently know involves using the recursive reduction algorithm described in Section 3.

(3) With regards to $N^*_{prim}(n)$, the number of primitive integer root quadruples (a, b, c, d) with $a = -n$, is it true that $N^*_{prim}(p) \sim p/4$ for p prime? Does this also hold for general n?

(4) Concerning root quadruples, what are the asymptotics of the total number of root quadruples having Euclidean height below T?

(5) We have not proved any reasonable lower bound on the number of integers below T that occur as curvatures in a fixed integral Apollonian packing. For how large a β can one prove asymptotically that at least $T^{\beta+o(1)}$ integers occur in every such packing?

(6) All of the preceding questions can also be raised for integral Apollonian packing of spheres in 3 dimensions, as discussed in [21]. For example, what are the modular restrictions (if any) for the Descartes quintuples (a, b, c, d, e) occurring in the packing with root quintuple $(0, 0, 1, 1, 1)$?

(7) In [20] it was shown that there exist strongly integral Apollonian packings, in which the circles all have integer curvatures and also the curvature×centers of the circles are Gaussian integers. Here the circle centers are coordinatized as complex numbers. The questions we investigated in this paper for integral packings can also be asked for strongly integral packings. If we write (x, xX) for a circle with curvature x and (complex) center X, then the pairs (x, xX) must also satisfy various modular constraints. For example, modulo 12, the standard integral packing (i.e., starting with the circles $(-1, 0), (2, 1), (2, -1)$ has just 20 types of circles, namely,

$$\begin{aligned}(x, xX) \;=\; & (2,1), (2,3), (2,5), (2,7), (2,9), (2,11) \\ & (3,2i), (3,4i), (3,8i), (3,10i) \\ & (6,3+4i), (6,3+8i), (6,9+4i), (6,9+8i) \\ & (11,0), (11,4), (11,8), (11,6i), (11,4+6i), (11,8+6i)\end{aligned}$$

and there are just 120 different four-circle configurations. What are the asymptotics of these types and configurations? What is the characterization of the integral (complex) vectors (x, xX) that can appear in a given packing?

We hope to return to some of these issues in a future paper.

Acknowledgments. The authors wish to acknowledge the insightful comments of Arthur Baragar, William Duke, Andrew Odlyzko, Eric Rains, and Neil Sloane at various stages of this work. We also thank the referee for many useful comments and historical references.

References

[1] D. Aharonov and K. Stephenson, Geometric sequences of discs in the Apollonian packing, Algebra i Analiz **9** (1997), No. 3, 104–140. [English version: St. Petersburg Math. J. **9** (1998), 509–545.]

[2] A. N. Andrianov, Dirichlet series that correspond to representations of zero by indefinite quadratic forms, Algebra i Analiz **1** (1989), No. 3, 71–82. [English Version: St. Petersburg Math. J. **1** (1990), 635–646.]

[3] D. Boyd, The disk-packing constant, Aequationes Math. **7** (1971), 182–193.

[4] D. Boyd, Improved bounds for the disk-packing constant, Aequationes Math. **9** (1973), 99–106.

[5] D. Boyd, The osculatory packing of a three-dimensional sphere, Canadian J. Math. **25** (1973), 303–322.

[6] D. Boyd, The residual set dimension of the Apollonian packing, Mathematika **20** (1973), 170–174.

[7] D. Boyd, The sequence of radii of the Apollonian packing, Math. Comp. **39** (1982), 249–254.

[8] H. S. M. Coxeter, The problem of Apollonius, Amer. Math. Monthly **75** (1968), 5–15.

[9] H. S. M. Coxeter, *Introduction to Geometry, Second Edition*, John Wiley and Sons, New York, 1969.

[10] H. S. M. Coxeter, Loxodromic sequences of tangent spheres, Aequationes Mathematicae **51** (1996), 104–121.

[11] H. S. M. Coxeter, Numerical distances among the spheres in a loxodromic sequence, The Mathematical Intelligencer **19** (1997), 41–47.

[12] T. W. Cusick, Continuants with bounded digits, Mathematika **24** (1977), 166–172.

[13] I. Daubechies and J. C. Lagarias, Sets of matrices all infinite products of which converge, Lin. Alg. Appl. **161** (1992), 227–263.

[14] W. Duke, Notes on the distribution of points on $x^2+y^2+z^2 = w^2$, unpublished manuscript, Feb. 1993.

[15] K. J. Falconer, *The Geometry of Fractal Sets*, Cambridge Tracts in Math., vol. 85, Camb. Univ. Press, Cambridge, 1986.

[16] L. R. Ford, Proc. Edinburgh Math. Soc. **35** (1916/17), 59–65.

[17] L. R. Ford, Fractions, Amer. Math. Monthly **45** (1938), 586–601.

[18] C. F. Gauss, *Disquisitiones Arithmeticae,* Leipzig 1801. (Reprinted in: Werke.) English translation: Springer-Verlag.

[19] R. L. Graham, J. C. Lagarias, C. L. Mallows, A. Wilks and C. Yan, Apollonian Packings: Geometry and Group Theory, I. Apollonian Group, `eprint: arXiv math.MG/0010298`

[20] R. L. Graham, J. C. Lagarias, C. L. Mallows, A. Wilks and C. Yan, Apollonian Packings: Geometry and Group Theory, II. Super-Apollonian Group and Integral Packings, `eprint: arXiv math.MG/0010302`

[21] R. L. Graham, J. C. Lagarias, C. L. Mallows, A. Wilks and C. Yan, Apollonian Packings: Geometry and Group Theory, III. Higher Dimensions, `eprint: arXiv math.MG/0010324`

[22] G. H. Hardy and E. M. Wright, *Introduction to the Theory of Numbers*, (4th ed.) Oxford University Press, 1960.

[23] K.E. Hirst, The Apollonian packing of circles, J. Lond. Math. Soc., **42** (1967), 281–291.

[24] A. Hurwitz, Solution to Problem 3084, *13* (1906), 164. In: *Mathematische Werke*, Vol II, Birkhäuser: Basle 1934, p. 751.

[25] E. Kasner and F. Supnick, The Apollonian packing of circles, Proc. Nat. Acad. Sci. USA **29** (1943), 378–384.

[26] J. C. Lagarias, C. L. Mallows and A. Wilks, Beyond the Descartes circle theorem, Amer. Math. Monthly, to appear. `eprint: arXiv math.MG/0101066`

[27] J. C. Lagarias and A. M. Odlyzko, Computing $\pi(x)$: an analytic method, J. Algorithms **8** (1987), 173–191.

[28] S. Lang, *Algebraic Number Theory*, (2nd ed.),1967 [Ch. VIII §2 Theorem 5, p. 161].

[29] D. G. Larman, On the exponent of convergence of a packing of spheres, Mathematika **13** (1966), 57–59.

[30] P. D. Lax and R. S. Phillips, The asymptotic distribution of lattice points in Euclidean and non-Euclidean spaces, J. Funct. Anal. **46** (1982), 280–350.

[31] B. B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman: New York, 1982.

[32] J. G. Mauldon, Sets of equally inclined spheres, Canad. J. Math. **14** (1962), 509–516.

[33] G. Maxwell, Sphere packings and hyperbolic reflection groups. J. Algebra **79** (1982), 78–97.

[34] Z. A. Melzak, Infinite packings of disks, Canad. J. Math. **18** (1966), 838–853.

[35] Z. A. Melzak, On the solid-packing constant for circles, Math. Comp. **23** (1969), 169–172.

[36] P. J. Nicholls, Diophantine approximation via the modular group, J. London Math. Soc. **17** (1978), 11–17.

[37] H. Rademacher, *Lectures on Elementary Number Theory,* Blaisdell: New York 1964.

[38] J. G. Ratcliffe and S. T. Tschantz, On the representation of integers by the Lorentzian quadratic form, J. Funct. Anal. **150** (1997), 498–525.

[39] B. Rodin and D. Sullivan, The convergence of circle packings to the Riemann mapping, J. Differential Geometry **26** (1987), 349–360.

[40] T. Rothman, Japanese temple geometry, Scientific American, May 1998, 84–91.

[41] H. F. Sandham, A square as the sum of seven squares, Quart. J. Math. (Oxford) **4** (1953), 230–236.

[42] N. J. A. Sloane, not just integer packings. The on-line encyclopedia of integer sequences. (URL is http://www.research.att.com/~njas/sequences/index.html)

[43] F. Soddy, The Kiss Precise, Nature **137** (June 20, 1936), 1021.

[44] F. Soddy, The bowl of integers and the Hexlet, Nature **139** (1937), 77–79.

[45] B. Söderberg, Apollonian tiling, the Lorentz group, and regular trees, Phys. Rev. A **46** (1992), No. 4, 1859–1866.

[46] E. C. Titchmarsh, *The Theory of the Riemann Zeta Function,* (Revised by D. R. Heath-Brown) Oxford, 1986.

[47] P. B. Thomas and D. Dhar, The Hausdorff dimension of the Apollonian packing of circles, J. Phys. A: Math. Gen. **27** (1994), 2257–2268.

[48] C. Tricot, A new proof for the residual set dimension of the Apollonian packing, Math. Proc. Cambridge Phil. Soc. **96** (1984), 413–423.

[49] A. I. Weiss, On isoclinal sequences of spheres, Proc. Amer. Math. Soc. **88** (1983), 665–671.

[50] J. B. Wilker, Open disk packings of a disk, Canad. Math. Bull., **10** (1967), 395–415.

[51] J. B. Wilker, Inversive Geometry, in: *The Geometric Vein*, (C. Davis, B. Grünbaum, F. A. Sherk, Eds.), Springer-Verlag: New York 1981, pp. 379–442.

[52] S. K. Zaremba, La methode des "bonnes treillis" pour le calcul des integrales multiples, in *Applications of number theory to numerical analysis,* (Montreal, 1971), (S. K. Zaremba, Ed.) Academic Press, New York, 1972, pp. 39–119.

email: graham@ucsd.edu
jcl@research.att.com
clm@research.att.com
allan@research.att.com
Catherine.Yan@math.tamu.edu

On numbers of Davenport-Schinzel sequences

Martin Klazar*

Abstract

One class of Davenport-Schinzel sequences consists of finite sequences over n symbols without immediate repetitions and without any subsequence of the type *abab*. We present a bijective encoding of such sequences by rooted plane trees with distinguished nonleaves and we give a combinatorial proof of the formula

$$\frac{1}{k-n+1}\binom{2k-2n}{k-n}\binom{k-1}{2n-k-1}$$

for the number of such normalized sequences of length k. The formula was found by Gardy and Gouyou-Beauchamps by means of generating functions. We survey previous results concerning counting of DS sequences and mention several equivalent enumerative problems.

1 Introduction

The set $DS(n)$ of *Davenport-Schinzel sequences* over n symbols is formed by finite sequences $u = a_1a_2 \dots a_k$ satisfying

1. $a_i \in [n] = \{1, 2, \dots, n\}$ for all i, each integer $j \in [n]$ appears in u.
2. For each pair $i < j$ of $[n]$ the first appearance of i in u precedes that of j.
3. $a_i \neq a_{i+1}$ for all $i = 1, 2, \dots, k-1$.

*During his stay on ASU the author was partially supported by Office of Naval Research grant NOOO14-90-J-1206.

4. $a_{i_1} = a_{i_3} = a \neq b = a_{i_2} = a_{i_4}$ holds for no four indices $1 \leq i_1 < \ldots < i_4 \leq k$.

Condition 3 forbids immediate repetitions while condition 4 does not allow any subsequence of the type $\ldots a \ldots b \ldots a \ldots b \ldots$ where a and b are two distinct numbers. Conditions 1 and 2 normalize sequences for purposes of enumeration.

One can consider *maximal* $DS(n)$ sequences, denoted as $MDS(n)$, which end with 1. For instance,

$$DS(3) = \{123, 1231, 1232, 12321, 1213, 12131\}$$

and

$$MDS(3) = \{1231, 12321, 12131\}.$$

The number of $MDS(n)$ sequences of length k is denoted by $f_{n,k}$ and their total number by f_n. Similarly, $b_{n,k}$ is the number of $DS(n)$ sequences of length k and $b_n = |DS(n)|$. Clearly, $b_1 = f_1 = 1$. The mapping $u \to u1$ is a bijection between $DS(n) \backslash MDS(n)$ and $MDS(n)$, $n > 1$. We see that

$$b_n = 2f_n \text{ and } b_{n,k} = f_{n,k} + f_{n,k+1}. \tag{1}$$

The minimum length of a $DS(n)$ sequence is n and the maximum length is $2n - 1$ (see [4]).

Our aim is to give a combinatorial proof of the formula

$$b_{n,k} = C_{k-n} \cdot \binom{k-1}{2n-k-1} = \frac{\binom{2k-2n}{k-n}\binom{k-1}{2n-k-1}}{k-n+1} \tag{2}$$

established by Gardy and Gouyou-Beauchamps in [6] by means of generating functions. Here $C_n = \binom{2n}{n}/(n+1)$ stands for the n-th *Catalan number* that counts, among other structures, the number of rooted plane trees on $n+1$ vertices.

The paper is organized as follows. In the next section we list several (classical) enumerative problems which are equivalent to counting of $MDS(n)$. In the third section a combinatorial proof of (2) is given. We introduce a new representation of $DS(n)$ by rooted plane trees on n vertices with distinguished nonleaves. To count such trees we encode them bijectively by another tree structure. The bijection is described in the fourth section.

We recall briefly some basic features of a *rooted plane tree* $T = (V, E)$, shortly an *rp tree*. It is a finite rooted tree with edges directed away from the *root* $r \in V$. For an edge $(u, v) \in E$ of T we call u the *parent* of v while v is a *child* of u. The order of children of u matters, we think of T as drawn in the plane with r at the lowest and all edges drawn as straight segments directed up. The number of children of $u \in V$ is denoted by $deg(u)$. A *leaf* is a vertex with no child. The number of leaves of T is denoted by $l(T)$. *Principal subtrees* of T are the trees which arise by deleting the root of T.

To conclude the present section we should say that Davenport-Schinzel sequences were introduced by Davenport and Schinzel [4] in a more general context where alternating subsequences $ababab\ldots$ of length d were excluded. The most important results of the theory of Davenport-Schinzel sequences are upper and lower bounds on their maximum length when d is fixed — [20], [8], and [2]. Applications include both computational and combinatorial geometry. From the enumerative point of view cases $d > 4$ have proven so far intractable. Surveys can be found in [1], [18], [13], and also in [9].

2 The Schröder family

There is an old *Schröder family* of mutually equivalent enumerative problems and the sequence of finite sets $\{MDS(n)\}_{n\geq 1}$ is a relatively new and less known member of it. As such $MDS(n)$ sequences had been enumerated and the generating function had been found well before they were defined. Since this is not articulated in other enumerative papers about $DS(n)$ sequences, it appears useful here to give a brief description of these problems bearing in mind $DS(n)$ sequences. Our list of references is by no means exhaustive.

The sequence of numbers $\{f_n\}_{n\geq 1}$ is the enumerator of the family. There is no closed formula for f_n but it can be computed by a recurrence relation, by a generating function, by sums with positive terms or by alternating sums. We list some of these expressions below.

Special rooted plane maps. The first enumerative paper about $DS(n)$ sequences is due to Mullin and Stanton [11]. They proved, not mentioning so, the membership of the problem to the Schröder family. We describe briefly their bijection between $MDS(n)$ and the set of special rooted plane maps which we will call *fences*.

By a *plane* multigraph we mean a planar multigraph with a specific embedding in the plane. We say it is *totally outerplane* if all edges lie on the boundary of the outer face. A *cut* edge in a connected multigraph G is an edge whose removal disconnects G. A *fence* (F, r, e) is a connected totally outerplane multigraph with no cut edges, with distinguished edge e and vertex r. The vertex r is incident with e and for an observer on r the outer face lies to the left of e.

Note that in a fence no two vertices are connected by three or more edges and that any fence arises from a connected totally outerplane graph by doubling the cut edges.

In F there is a unique closed Eulerian walk C which goes around F clockwise, starts at r, and uses e as its first edge. C produces an $MDS(n)$ sequence. We label r as 1 and we write down the labels of vertices in the order of C. Whenever an unlabeled vertex is encountered, it is given the least unused label.

Counting $MDS(n)$ or fences on n vertices is therefore equivalent. Mullin and Stanton proved the formula

$$b_{n,2n-1} = f_{n,2n-1} = C_{n-1} = \frac{1}{n}\binom{2n-2}{n-1} \tag{3}$$

by observing that fences on n vertices with maximum number of edges are rp trees on n vertices with all edges doubled. They also proved that

$$(n+1)f_{n+1} - (6n-3)f_n + (n-2)f_{n-1} = 0 \quad (n \geq 3), \tag{4}$$

using the generating function

$$\sum_{n=1}^{\infty} f_n x^n = \frac{1 + x - \sqrt{1 - 6x + x^2}}{4}. \tag{5}$$

They derived, for $n \geq 2$, the formula

$$f_n = \sum_{0 \leq k \leq n/2-1} 3^{n-2-2k}\ 2^k \binom{n-2}{2k} C_k. \tag{6}$$

Equation (5) together with the first ten values of f_n appear already in [17]. Interestingly, numbers f_n and equation (4) can also be found (without any combinatorial interpretation) in [15], p. 168.

Dissections of a convex polygon. A *dissection* of a convex polygon P with labeled vertices is a set of diagonals, no two of them crossing. Dissections with various restrictions on the face sizes were enumerated by Etherington [5]. Etherington pointed out that the case when there is no restriction at all is equivalent to Schröder's bracketing problem. Similar problems were investigated by Motzkin [10].

Roselle [16] gave the following bijection that matches dissections of a convex $(n+1)$-gon and $MDS(n)$ sequences. Let D be a dissection of P with vertices labeled by $1, 2, \ldots, n+1$ clockwise. Start with the sequence $12 \ldots n1$. Then insert between $j-1$ and j in the decreasing order the numbers k where $k < j$ and kj is a diagonal of D. Similarly insert between n and 1 the decreasing list of numbers k joined by a diagonal to $n+1$. What you get is an $MDS(n)$ sequence.

In fact, Roselle described this bijection only for the case of triangulations and $MDS(n)$ sequences with maximum length. It is well known that triangulations are counted by Catalan numbers and Roselle gave this way an alternative proof of (3). However, it is easy to see that the bijection works in general and that it matches the elements of $MDS(n)$ of length k with dissections of a convex $(n+1)$-gon with $k-n-1$ diagonals. And this implies already (2) because as early as 1866 Prouhet [14] (see [3], p. 75) counted the number, $r(n,d)$, of dissections of a convex n-gon by d diagonals:

$$r(n,d) = \frac{1}{d+1}\binom{n-3}{d}\binom{n+d-1}{d}. \tag{7}$$

Thus $f_{n,k} = r(n+1, k-n-1)$, and (7) combined with (1) give (2). Since this combination leading to a combinatorial proof of (2) went unnoticed, we take the freedom to present another combinatorial proof.

Bracketings of a product. Schröder [17] discovered the family in 1870 by solving the following problem. Given a noncomutative product of n terms, in how many ways can one bracket them so that each bracket contains at least two factors? The outer bracket is not allowed. The answer is again given by the numbers f_n.

A nice exposition of (4) and (5) is in Comtet [3] on p. 56 who gives the expression, $n > 2$,

$$f_n = \sum_{0 \le k \le n/2} (-1)^k \, \frac{(2n-2k-3)!!}{k!(n-2k)!} \, 3^{n-2k} \, 2^{-k-2}. \tag{8}$$

Here $(2n-2k-3)!!$ denotes the odd factorial $1 \cdot 3 \cdot 5 \cdots (2n-2k-3)$. Standard Lagrange inversion (see Goulden and Jackson [7], problem 2.7.12) yields a simpler alternating expression

$$f_n = \frac{1}{n} \sum_{i=0}^{n-1} (-1)^i 2^{n-1-i} \binom{n}{i} \binom{2n-2-i}{n-1}. \tag{9}$$

Other disguises. There is an obvious tree disguise of the problem. It was noticed already by Etherington that bracketings of n terms can be visualized by rooted plane trees having n leaves and no vertex with degree 1. Two other, less obvious, tree disguises are given in the next two sections.

Besides (2) Gardy and Gouyou-Beauchamps in [6] determined the average length and average number of symbols of a $DS(n)$ sequence and found the bivariate generating function for $b_{n,k}$'s. They gave also a bijection between $DS(n)$ and Schröder words of length $2n-2$. These are words over the alphabet $\{x, \bar{x}, y\}$ given by the language equation

$$X = 1 + yyX + xX\bar{x}X.$$

3 Coding and counting

The first step in our combinatorial proof of (2) is an encoding of $DS(n)$ by the set $CT(n)$ of pairs $\mathcal{T} = (T, S)$, where T is an rp tree on n vertices and S is a subset of nonleaves of T. We call them *circled rooted plane trees*, or shortly *crp trees*, since we visualize the distinguished nonleaves as being circled. See Figure 1. The encoding is easier to describe recursively but the nonrecursive version is easier to perform.

Recursive version. Suppose $u = a_1 a_2 \dots a_k$ is a $DS(n)$ sequence. If $k = 1$ then u is encoded by a single uncircled vertex. Otherwise we use the decomposition $u = 1u_1 1u_2 \dots 1u_l$ of u by all appearances of 1. A moment of thought reveals that the segments u_i are nonempty, except possibly for u_l, they do not share symbols, and each u_i satisfies conditions 3 and 4 of the definition of $DS(n)$. We rename the symbols so that u_i complies with conditions 1 and 2 as well and we encode u_i by $\mathcal{T}_i$. The sequence u is encoded by the crp tree $\mathcal{T}$ with principal subtrees from left to right $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_l$, the

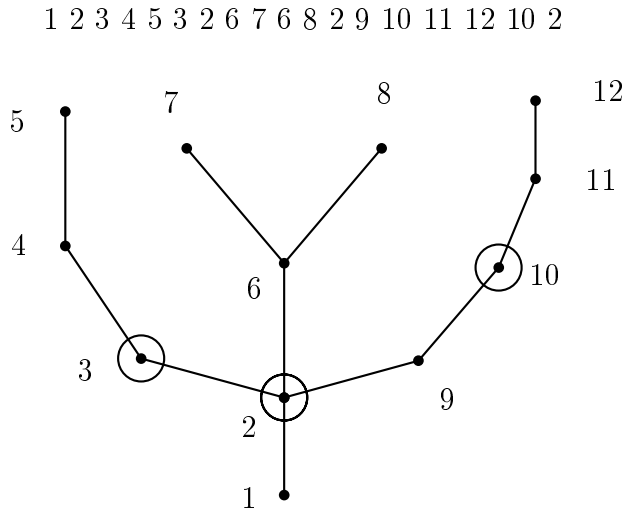


Figure 1: Encoding by crp trees

root of $\mathcal{T}$ is circled iff u_l is empty. We leave the inverse decoding to the reader.

Nonrecursive version. Suppose $u = a_1 a_2 \dots a_k$ is a $DS(n)$ sequence. A crp tree (T, S) on n vertices is generated, the algorithm uses three auxiliary variables: i is the index of the currently processed term of u, v denotes the currently processed vertex, and C is either empty or a singleton set containing a candidate for an element of the set S.

We initialize the variables by setting $i := 1, v := p$, and $S := C := \emptyset$, where p, the root, is an arbitrary point in the plane labeled by $a_1 = 1$. In the general step if $i = k$ we are done. If $i < k$ then there is to distinguish two cases.

1. a_{i+1} has appeared earlier in the sequence. We denote by q the unique vertex on the path joining the root and v which is labeled by a_{i+1}. We put

$$i := i + 1,\ v := q,\ S := S \cup C, \text{ and } C := \{v\} = \{q\}.$$

In the case that now $i = k$ (we did the last step) we add q to S.

2. a_{i+1} is a new symbol. We join to v, above v and to the right of the

children of v, a new child q and give it the label a_{i+1}. Then we put

$$i := i + 1,\ v := q,\ S := S, \text{ and } C := \emptyset.$$

So S consists of vertices which were reached by a jump from above, and from which we jumped down again or for which the procedure terminated. In the end we can discard the labels. Even so it is easy to reconstruct u from the crp tree (T, S). We describe it now.

If (T, S) is a crp tree then the corresponding $DS(n)$ sequence $u = a_1 a_2 \dots a_k$ arises by climbing up and jumping down around T clockwise and writing down the labels of vertices. On the beginning the vertices are unlabeled. We start at the root r and give it the label 1. Whenever an unlabeled vertex is encountered it is given the least unused label. We go up without jumps to the leftmost leaf z. For the crp tree on Figure 1 we produce 12345. Then we jump down on the r-z path P in jumps following elements of $P \cap S$ until we reach a vertex $v \in P$ that has a child to the right of P. In our example we perform the jumps 53 and 32. It is irrelevant now that 2 is circled, we would end in it anyway. From v we continue in consecutive steps upward to the second leftmost leaf and so on. For the rightmost leaf w, which is the last one to be visited, there is no such vertex v and we finish jumping at the lowest element of $Q \cap S$ where Q is the r-w path. If $Q \cap S = \emptyset$ then we finish at w. In our example we finish at 2 and only now it matters that 2 is circled.

We recall that $l(T)$ is the number of leaves in T. The following theorem summarizes the above encoding procedures.

Theorem 3.1 *The above encodings give a bijection between the sets $DS(n)$ and $CT(n)$. It follows that $b_{n,k}$ equals to the number of crp trees (T, S) on n vertices with $2n - k - 1$ uncircled nonleaves, i. e. crp trees (T, S) with $|V(T)| = n$ and $n - l(T) - |S| = 2n - k - 1$.*

Proof. Using our recursive version we can easily prove the bijectivity. If $u \in DS(n)$ has length k then it is encoded by a crp tree (T, S) on n vertices such that $k = n + l(T) + |S| - 1$. So the set of circled nonleaves S has $k - n - l(T) + 1$ elements and the complement S^c (complement in the set of nonleaves) has $n - l(T) - |S| = 2n - k - 1$ elements. □

It is easier to count the pairs (T, S^c) than the pairs (T, S) because the cardinality $|S^c|$ is independent of the structure of T. Therefore (formally we

switch between circled and uncircled nonleaves) it suffices to count crp trees with a fixed number of vertices and circles. The next step is an encoding of crp trees by *rooted plane trees with dots*, shortly *drp trees*. We need few definitions.

Consider an rp tree T with n vertices drawn as a picture in the plane. Let v be a vertex with $d = deg(v)$ children. The $d+1$ edges incident with v, which are drawn as straight segments, split the neighborhood of v into $d+1$ wedge-shaped areas which we call *gaps* of v. For the root of T there is no difference, we imagine an edge joining it to a virtual parent. The set $g(T)$ of all gaps in T has $\sum_V (deg(v)+1) = 2n-1$ elements. A *drp tree* is a pair (T, D) where T is an rp tree and D is a finite multisubset of $g(T)$. This means that we distinguish, possibly with repetitions, some gaps of T. We visualize a drp tree (T, D) as an rp tree T with D determined by dots distributed in the gaps of T. The number of dots in a gap g is then the multiplicity of g in D. Look at the picture on Figure 2.

There is a bijection between crp trees with n vertices and m circles and drp trees with $n-m$ vertices and m dots, the proof is given in the next section. Since it is easy to count drp trees with a given number of vertices and dots, we are done.

Theorem 3.2 *The number of crp trees with n vertices and m circles is*

$$C_{n-m-1} \cdot \binom{2n-m-2}{m}.$$

Proof. From Lemma 4.2 of the next section we know that the number of crp trees with n vertices and m circles is the same as the number of drp trees with $n-m$ vertices and m dots. But this is equal to the number of rp trees on $n-m$ vertices times the number of m element multisubsets of a $2n-2m-1$ element set. □

The proof of (2) is finished, (2) follows immediately from Theorems 3.1 and 3.2 by setting $m = 2n-k-1$.

The total number b_n of $DS(n)$ sequences can be counted in two ways. One can sum (2) for all $k = n, n+1, \ldots, 2n-1$. Changing the summation range the expression found in [6] follows:

$$b_n = \sum_{j=0}^{n-1} \frac{1}{j+1} \binom{2j}{j} \binom{j+n-1}{2j}. \tag{10}$$

The other way is to form groups of crp trees on n vertices with the same number of leaves. The number, $p(n,l)$, of rooted plane trees on n vertices with l leaves is given by the well known formula (first appearing implicitly in [12])

$$p(n,l) = \frac{1}{n-l}\binom{n-1}{l}\binom{n-2}{l-1}.$$

Note that $p(n,l) = p(n,n-l)$. The number of crp trees with the same underlying rp tree is 2^{n-l}. Hence

$$b_n = \sum_{l=1}^{n-1} p(n,l)\cdot 2^{n-l} = \sum_{l=1}^{n-1} \frac{2^l}{n-l}\binom{n-1}{l}\binom{n-2}{l-1}. \tag{11}$$

Well, how many $MDS(n)$ sequences are there then? From either (4), (6), (8), (9), (10) or (11), taking (1) into account, we get

$$\{f_n\}_{n\geq 1} = \{1,1,3,11,45,197,903,4279,20793,103049,\ldots\}.$$

This is the 1163-rd sequence in the phenomenal Sloane's handbook [19].

4 Contractions and expansions

We show that there is a natural bijection between crp trees with n vertices and m circles and drp trees with $n-m$ vertices and m dots. As an example to illustrate our idea we consider first crp and drp trees with one circle and one dot. Let $(T,\{v\})$ be such a crp tree, let e join v to its leftmost child. We put one dot d in the gap of v lying to the right of e and contract e. The drp tree obtained is denoted by $(T^*,\{d\})$. It is easy to see how to recover $(T,\{v\})$ from $(T^*,\{d\})$. Hence the mapping $(T,\{v\}) \to (T^*,\{d\})$ is the desired bijection in the case $m=1$.

To generalize this to $m>1$ we need to define a more general tree structure with both circles and dots and we need to define an order to perform the contractions. First we recall the standard linear order $(V,\prec)$ on the vertex set of an rp tree T. For two distinct vertices $u,v\in V$ one considers the paths P_u and P_v joining the root to u and v. Two cases arise.

1. One path — say P_u — is an initial segment of the other path. Then $u \prec v$.

2. Otherwise there is a branching point and one path — say P_u — branches to the right. Then again $u \prec v$.

Suppose (T, S, D) is a triple where (T, S), resp. (T, D), is a crp tree, resp. a drp tree. We define a partial ordering $(S \cup D, \prec)$. If $x \in S \cup D$ then x is either a circled vertex v or a dot in a gap of a vertex v, in both cases the expression the *vertex of* x refers to v. Let $x, y \in S \cup D$ be two distinct elements, let u be the vertex of x, and let v be the vertex of y.

1. $u \neq v$. We set $x \prec y$ iff $u \prec v$.

2. $u = v$. If x is a dot in a gap g and y is a dot in a gap h, g and h belong to the same vertex, we set $x \prec y$ iff g lies to the right of h. In the two remaining cases — both x and y are dots in the same gap or one of them is a dot and the other is a circled vertex — x and y are set to be incomparable.

A *circled rooted plane tree with dots*, shortly a *cdrp tree*, is a triple $\mathcal{T} = (T, S, D)$ where (T, S), resp. (T, D), is a crp tree, resp. a drp tree, and such that $S \prec D$. In other words, $v \prec d$ for any $v \in S$ and any $d \in D$. In particular, each gap of a circled vertex is empty. We define two mutually inverse operations on $\mathcal{T}$ with an example to illustrate them on Figure 2. The operations preserve the sum $|S| + |D|$. Let v be the largest, with respect to $\prec$, vertex of S and w be its leftmost child. Let d be one of the minimal dots.

Contraction of $\mathcal{T}$ contracts the edge $e = \{v, w\}$, i.e. e is deleted and v and w are identified. The new vertex z created by the identification is not circled. All other circles are preserved. The dots of the leftmost gap of w appear now in the leftmost gap of z and the dots of the rightmost gap of w appear now in what was the second leftmost gap of v. Furthermore we add to the latter one more dot. The distribution of dots in other gaps is preserved. Resulting cdrp tree is denoted by $C(\mathcal{T})$.

Expansion of $\mathcal{T}$ expands d. Suppose d is located in a gap g of a vertex z. We delete d and split z into two vertices w and v. The vertex w is slightly to the left of v and is joined only to those children of z which were to the left of g. Vertex v is joined to the remaining children and to the parent of z. Now w is moved upward a bit with all the dots it bears and is joined to v as its new leftmost child. The dots of g appear now in the rightmost gap of w. All gaps of v are empty. Vertex v is circled, vertex w is not circled. Dots in other gaps and other circles are preserved. Resulting cdrp tree is denoted by $E(\mathcal{T})$.

Lemma 4.1 *$C(\mathcal{T})$ and $E(\mathcal{T})$ are cdrp trees again. Also $C(E(\mathcal{T})) =$*

$E(C(\mathcal{T})) = \mathcal{T}$ *whenever the operations involved are defined.*

Proof. The lemma can be easily proved by an inspection of the above definitions. The proof is left to an interested reader. □

Let $\mathcal{T} = (T, S)$ be a crp tree with n vertices and m circles. We assign to $\mathcal{T}$ a drp tree $\mathcal{U} = C^m(\mathcal{T})$ which arises by m iterations of the contraction operation on $\mathcal{T}$.

Lemma 4.2 *The above assignment is a bijection between crp trees with n vertices and m circles and drp trees with $n - m$ vertices and m dots.*

Proof. It follows immediately from the previous lemma that the mappings $\mathcal{T} \to \mathcal{U} = C^m(\mathcal{T})$ and $\mathcal{U} \to \mathcal{T} = E^m(\mathcal{U})$ are inverses of one another. □

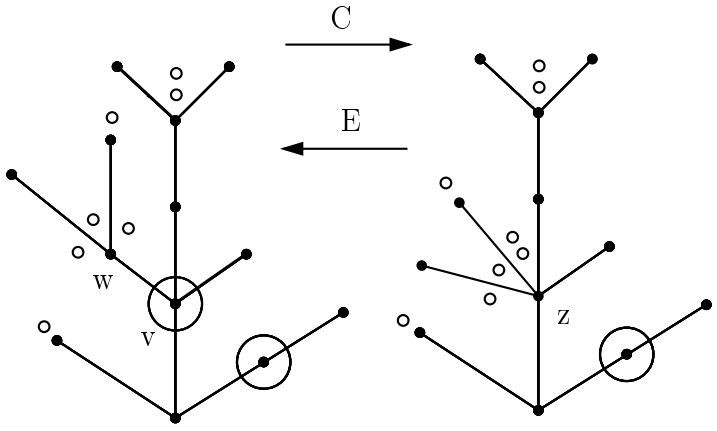


Figure 2: A contraction and an expansion

Acknowledgments. The author is grateful to prof. H. Barcelo, Arizona State University, for reading the manuscript and for her valuable comments. He thanks also to M. Zeman for sending him copies of some references. The comments of two anonymous referees helped to improve the readability of the paper.

References

[1] P. K. Agarwal, *Intersection and decomposition algorithms for planar arrangements*, Cambridge University Press, 1991.

[2] P. K. Agarwal, M. Sharir and P. Shor, Sharp upper and lower bounds on the lengths of general Davenport-Schinzel sequences, *J. Combin. Theory A* **52** (1989), 228–274.

[3] L. Comtet, *Advanced Combinatorics*, D. Reidel Publishing Company, 1974.

[4] H. Davenport and A. Schinzel, A combinatorial problem connected with differential equations, *Amer. J. Math.* **87** (1965), 684–694.

[5] I. M. H. Etherington, Some problems of non-associative combinatorics, *The Edinburgh Math. Notes* **32** (1940), 1–6.

[6] D. Gardy and D. Gouyou-Beauchamps, Enumerating Davenport-Schinzel sequences, *Informatique théorique et Applications / Theoretical Informatics and Applications* **26** (1992), 387-402.

[7] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, J. Wiley, 1983.

[8] S. Hart and M. Sharir, Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes, *Combinatorica* **6** (1986), 151–177.

[9] M. Klazar, *Combinatorial aspects of Davenport-Schinzel sequences*, thesis, Charles University, Prague 1995.

[10] Th. Motzkin, Relations between hypersurfaces crossratio, and a combinatorial formula for partitions of a polygon, for a permanent preponderance and for nonassociative products, *Bull. of the American Math. Soc.* **54** (1948), 362–370.

[11] R. C. Mullin and R. G. Stanton, A map-theoretic approach to Davenport-Schinzel sequences, *Pacific J. Math.* **40** (1972), 167–172.

[12] V. T. Narayana, A partial order and its application to probability, *Sankhyá* **21** (1959), 91–98.

[13] J. Pach (Editor), *New Trends in Discrete and Computational Geometry*, Springer, 1993.

[14] E. Prouhet, *Nouvelles Annales Mathematiques* **5** (1866), 384

[15] J. Riordan, *Combinatorial Identities*, John Wiley, 1968.

[16] D. P. Roselle, An algorithmic approach to Davenport-Schinzel sequences, *Utilitas Math.* **6** (1974), 91–93.

[17] E. Schröder, Vier combinatorische Probleme, *Zeitschrift für Mathematik und Physik* **15** (1870), 361–376.

[18] M. Sharir and P. K. Agarwal, *Davenport-Schinzel sequences and their geometric applications*, Cambridge University Press, 1995.

[19] N. J. A. Sloane, *A Handbook of Integer Sequences*, Academic Press, 1973. (new updated edition in Academic Press, 1995)

[20] E. Szemerédi, On a problem by Davenport and Schinzel, *Acta Arith.* **25** (1974), 213–224.

Department of Applied Mathematics
Charles University
Malostranské náměstí 25
11800 Praha 1
Czech Republic
klazar@kam.ms.mff.cuni.cz

and

Department of Mathematics
Arizona State University
Tempe 85281 Arizona
USA

NUMBER THEORY AND FORMAL LANGUAGES

JEFFREY SHALLIT*

Abstract. I survey some of the connections between formal languages and number theory. Topics discussed include applications of representation in base k, representation by sums of Fibonacci numbers, automatic sequences, transcendence in finite characteristic, automatic real numbers, fixed points of homomorphisms, automaticity, and k-regular sequences.

Key words. finite automata, automatic sequences, transcendence, automaticity

AMS(MOS) subject classifications. Primary 11B85, Secondary 11A63 11A55 11J81

1. Introduction. In this paper, I survey some interesting connections between number theory and the theory of formal languages. This is a very large and rapidly growing area, and I focus on a few areas that interest me, rather than attempting to be comprehensive. (An earlier survey of this area, written in French, is [1].) I also give a number of open questions.

Number theory deals with the properties of integers, and formal language theory deals with the properties of strings. At the intersection lies

(a) the study of the properties of integers based on their *representation* in some manner — for example, representation in base k; and

(b) the study of the properties of strings of digits based on the integers they represent.

An example of a theorem of type (a) — perhaps the first significant one — is the famous theorem of Kummer [60, pp. 115–116], which states that the exponent of the highest power of a prime p which divides the binomial coefficient $\binom{n}{m}$ is equal to the number of "carries" when m is added to $n-m$ in base p.

For type (b) I do not know a theorem as fundamental as Kummer's. But here is a little problem that some may find amusing. Call a set of strings *sparse* if, as $n \to \infty$, it contains a vanishingly small fraction of all possible strings of length n. Then can one find a sparse set S of strings of 0's and 1's such that every string of 0's and 1's can be written as the concatenation of two strings from S? One solution is to let S be the set of all strings of 0's and 1's such that the number of 1's is a sum of two squares. Then by a famous theorem in number theory — Lagrange's theorem — every number n is the sum of *four* squares, so every string of 0's and 1's is a concatenation of two strings chosen from S. The sparseness of S follows

* Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. E-mail: shallit@graceland.uwaterloo.ca . Research supported in part by a grant from NSERC.

from an estimate in sieve theory [38]. Further examples of theorems of type (b) can be found in Section 8.1.

It may be objected that studying the formal language aspects of number theory is somewhat artificial, in the sense that it depends on choosing one particular representation — such as representation in base 2 — and that there is no reason to choose base 2 over any other base. For example, recall the famous objection of Hardy to certain kinds of digital problems[1]:

> These are odd facts, very suitable for puzzle columns and likely to amuse amateurs, but there is nothing in them which appeals much to a mathematician. The proofs are neither difficult nor interesting — merely a little tiresome. The theorems are not serious; and it is plain that one reason (though perhaps not the most important) is the extreme speciality of both the enunciations and the proofs, which are not capable of significant generalization. [46, p. 105]

I offer four answers to Hardy's objection. First, we attempt to make our theorems as general as possible. For example, we can try to prove theorems for all bases k rather than just a single base. Second, sometimes some bases *do* occur naturally in problems, and base 2 is one of them; see Section 4. Third, the area has proved to have many applications; perhaps the most dramatic examples are the recent simple proofs of transcendence in finite characteristic by Allouche and others; see Section 5. Finally, the area is "natural", and I submit as evidence the fact that many good mathematicians throughout history have worked in it, including Kummer, Lucas, and Carlitz.

2. Notation. I begin with some notation for formal languages, for which a good reference is the book of Hopcroft and Ullman [49].

Let Σ be a finite list of symbols, or *alphabet*, and let Σ^* denote the free monoid over Σ, that is, the set of all finite strings of symbols chosen from Σ, with concatenation as the monoid operation. Thus, if $\Sigma = \{0, 1\}$, then

$$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\},$$

where ϵ is the notation for the empty string. A *formal language*, or just *language*, is defined to be any subset of Σ^*.

Let L, L_1, L_2 be languages. We define the concatenation of languages as follows:

$$L_1 L_2 = \{x_1 x_2 \ : \ x_1 \in L_1, \ x_2 \in L_2\}.$$

[1] The two problems he cited as examples were (a) show that 8712 and 9801 are the only four-digit numbers which are nontrivial integral multiples of their reversals and (b) show that 153, 370, 371, and 407 are the only integers > 1 which are equal to the sum of the cubes of their decimal digits. Today, digital problems continue to attract attention and criticism; see, for example, [35].

Define $L^0 = \{\epsilon\}$, and $L^i = LL^{i-1}$ for $i \geq 1$. We define the *Kleene closure* of a language by

$$L^* = \bigcup_{i \geq 0} L^i.$$

A *regular expression* over an alphabet Σ is a way to denote certain languages — a finite expression using the symbols in Σ together with $+$ (to denote union), $*$ (to denote Kleene closure), ϵ (to denote the empty string), $\emptyset$ (to denote the empty set), and parentheses for grouping. For example, the regular expression $(\epsilon + 1)(0 + 01)^*$ denotes the set of all strings over $\{0, 1\}$ containing no two consecutive 1's. If a language can be represented by a regular expression, it is said to be *regular*.

3. Number representations. In order to talk about numbers in formal language theory terms, we need a way to represent numbers as strings of symbols over a finite alphabet. Let us begin with the integers. A classical way to do this is the canonical representation in base k:

THEOREM 3.1. *Let k be an integer ≥ 2. Then every positive integer n can be represented uniquely in the form $n = \sum_{0 \leq i \leq r} a_i k^i$, where the a_i are integers with $0 \leq a_i < k$, and $a_r \neq 0$.*

By associating n with the string $a_r a_{r-1} \cdots a_1 a_0$, this theorem gives a bijection between the positive integers and the set of strings given by the regular expression $(\Sigma_k - \{0\})\Sigma_k^*$, where $\Sigma_k = \{0, 1, 2, \ldots, k-1\}$. We define $(n)_k$ to be the string $a_r a_{r-1} \cdots a_1 a_0$ representing n in base k. We also define the inverse map $[w]_k$ to be the value of the string w when interpreted as a base-k number. We define $(0)_k = \epsilon$ and $[\epsilon]_k = 0$.

There are many relationships between base-k representation and elementary number theory. Here is just one example. Given an integer n, we may form $s_k(n)$, the sum of its base-k digits. For a prime p, let $\nu_p(n)$ denote the exponent of the highest power of p dividing n. Then we have the following classical theorem of Legendre [61, Vol. I, p. 10]:

THEOREM 3.2. *Let p be a prime number. Then for all $n \geq 0$ we have*

$$\nu_p(n!) = \frac{n - s_p(n)}{p - 1}.$$

One annoyance is that the canonical representation in base k suffers from the "leading zeros" problem — that is, the map $w \to [w]_k$ is not one-one if $w \in \Sigma_k^*$. For example, $[101]_2 = [0101]_2 = [00101]_2 = 5$. One way around this difficulty is the following simple "folk theorem", whose precise origins are unknown to me (but see [87, Note 9.1, pp. 90–91], [101, p. 24], and [40]):

THEOREM 3.3. *Let k be an integer ≥ 2. Then every non-negative integer can be represented uniquely in the form $n = \sum_{0 \leq i \leq r} a_i k^i$, where the a_i are integers with $1 \leq a_i \leq k$.*

For example, $13 = 2 \cdot 4 + 2 \cdot 2 + 1 \cdot 1$. This theorem gives a bijection between $\mathbb{N}$, the non-negative integers, and the regular language $(1 + 2 + \cdots + k)^*$.

There are many other ways to represent the non-negative integers. For example, let the Fibonacci numbers be defined by $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$. The following theorem gives the so-called *Zeckendorf* or *Fibonacci* representation [65,107]:

THEOREM 3.4. *Every non-negative integer can be represented uniquely in the form* $\sum_{2 \le i \le r} a_i F_i$, *where* $a_i \in \{0, 1\}$, *and* $a_i a_{i+1} \neq 1$.

This theorem gives a bijection between $\mathbb{N}$ and the regular language $\epsilon + 1(0 + 01)^*$. Notice that in all three cases we have examined, the set of "valid" representations is a regular language. This observation raises the question, for what numeration systems is the set of valid representations regular? See, for example, [91,48,67].

As above, if m and n are integers, then we can uniquely write $m = 2^{a_1} + \cdots + 2^{a_c}$ and $n = 2^{b_1} + \cdots + 2^{b_d}$, where $a_1 < \cdots < a_c$ and $b_1 < \cdots < b_d$. We clearly have

$$mn = \sum_{1 \le i \le c} \sum_{1 \le j \le d} 2^{a_i + b_j}.$$

Knuth [57] found a surprising generalization of this identity: if the Zeckendorf representation of m is $F_{a_1} + F_{a_2} + \cdots + F_{a_c}$, and the Zeckendorf representation of n is $F_{b_1} + F_{b_2} + \cdots + F_{b_d}$, define

$$m \circ n = \sum_{1 \le i \le c} \sum_{1 \le j \le d} F_{a_i + b_j}.$$

Then the $\circ$ multiplication is associative! Also see [7,43].

We now turn to the representation of rational numbers. Let $[a_0, \ldots, a_n]$ be an abbreviation for the *continued fraction*

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots + \cfrac{1}{a_n}}}. \tag{3.1}$$

THEOREM 3.5. *Every rational number in* $(0, 1)$ *can be expressed uniquely in the form*

$$[0, a_1, a_2, \ldots, a_n]$$

where the a_i *are positive integers and* $a_n \ge 2$.

As an application of this theorem, we prove the following theorem, inspired by [77]:

THEOREM 3.6. *There is a bijection* $r : \mathbb{N} \to \mathbb{Q}$ *such that both* r *and* r^{-1} *are computable in polynomial time.*

Proof. It suffices to give such a bijection between $\mathbb{N}$ and $\mathbb{Q} \cap (0,1)$.

Let $f_k : \mathbb{N} \to (1+2+\cdots+k)^*$ be the map that takes a non-negative integer to its representation in base k using digits $\{1, 2 \ldots, k\}$, as discussed in Theorem 3.3, and let f_k^{-1} be the inverse map. Let g be the map which takes a string over $(1+2+3)^*$ as an argument and returns a list of strings, where the 3's are treated as delimiters. For example, $g(121313322) = (121, 1, \epsilon, 22)$. Let h be the map such that

$$h(a_1, a_2, \ldots, a_k) = (0, a_1 + 1, \ldots, a_{k-1} + 1, a_k + 2).$$

Then we define the bijection r as follows:

$$r(n) = [h(f_2^{-1}(g(f_3(n))))],$$

where the function f_2^{-1} is extended in the obvious way to operate on lists of strings.

For example, consider the case $n = 12590$. Then its representation in base 3 using digits $\{1, 2, 3\}$ is 121313322. This is transformed by g into the list $(121, 1, \epsilon, 22)$, which is mapped by f_2^{-1} into $(9, 1, 0, 6)$. Then h maps this to $(0, 10, 2, 1, 8)$. Hence $r(12590) = [0, 10, 2, 1, 8] = 26/269$.

It remains to see that r and r^{-1} can be computed in polynomial time. That f_3 and f_2^{-1} can be computed in polynomial time is easy, and is left to the reader. For the polynomial time computability of continued fractions, see, for example, [8, Chapter 4]. □

There are many other formal language aspects of continued fractions. Some of these deal with the so-called "LR" or "Stern-Brocot" representation of rational numbers [44]. If

$$\theta = [a_0, a_1, a_2, \ldots],$$

then the LR-representation of θ is the string

$$R^{a_0} L^{a_1} R^{a_2} L^{a_3} \cdots.$$

Let a, b, c, d be integers with $ad - bc \neq 0$. Raney [83] gave a finite-state transducer to compute the LR-expansion of $\tau = (a\theta + b)/(c\theta + d)$ from that of θ. Using Raney's theorem, one can give a purely formal-language-theoretic proof of the fact that θ has bounded partial quotients iff τ does [90].

4. The Thue-Morse sequence. Recall from the previous section that $s_2(n)$ denotes the sum of the bits in the base-2 representation of n.

Now define an infinite word $\mathbf{t} = t_0 t_1 t_2 \cdots$ over $\{0, 1\}$, as follows: $t_n = s_2(n) \bmod 2$. This infinite word is sometimes called the Thue-Morse sequence, because both Thue [99] and Morse [75] examined its properties near the beginning of this century. But Prouhet implicitly used the definition of $\mathbf{t}$ in an 1851 paper ([82]; also see [104]) that gave a solution to the multigrade problem.

The *multigrade problem* (or Tarry-Escott problem; see [62]) is to find disjoint sets U, V that $\sum_{u \in U} u^i = \sum_{v \in V} v^i$ for $i = 0, 1, \ldots, k-1$. Prouhet observed that one could take $U = \{0 \leq n < 2^k : t_n = 0\}$ and $V = \{0 \leq n < 2^k : t_n = 1\}$. For example, we have

$$0^i + 3^i + 5^i + 6^i = 1^i + 2^i + 4^i + 7^i$$

for $i = 0, 1, 2$.

Another result of number-theoretic interest related to the Thue-Morse sequence is the following. Woods [103] and Robbins [85] observed that

$$\prod_{n \geq 0} \left(\frac{2n+1}{2n+2} \right)^{(-1)^{t_n}} = \frac{\sqrt{2}}{2}. \tag{4.1}$$

Here is a simple proof, due to Jean-Paul Allouche: Let $P = \prod_{n \geq 0} \left(\frac{2n+1}{2n+2} \right)^{(-1)^{t_n}}$ and let $Q = \prod_{n \geq 1} \left(\frac{2n}{2n+1} \right)^{(-1)^{t_n}}$. Clearly

$$PQ = \frac{1}{2} \prod_{n \geq 1} \left(\frac{n}{n+1} \right)^{(-1)^{t_n}}.$$

Now break this infinite product into separate products over odd and even indices; we find

$$\begin{aligned} PQ &= \frac{1}{2} \prod_{n \geq 0} \left(\frac{2n+1}{2n+2} \right)^{(-1)^{t_{2n+1}}} \prod_{n \geq 1} \left(\frac{2n}{2n+1} \right)^{(-1)^{t_n}} \\ &= \frac{1}{2} P^{-1} Q. \end{aligned}$$

It follows that $P^2 = \frac{1}{2}$. (Convergence and correctness of the rearrangements are left to the reader.)

But in fact, even more is true. Suppose one tries to express $\frac{\sqrt{2}}{2}$ as an infinite product of terms of the form $(\frac{2n+1}{2n+2})^{\pm 1}$, where the sign for $n = 0$ is chosen to be $+1$, and then iteratively chosen according to a greedy algorithm: if the product constructed so far is greater than $\frac{\sqrt{2}}{2}$, choose the sign $+1$, and if the product constructed so far is smaller than $\frac{\sqrt{2}}{2}$, choose the sign -1. Then the sequence of signs chosen is exactly $(-1)^{t_n}$. I conjectured this in 1983 [89], and it was proved by Allouche and Cohen in 1985 [5].

Notice that the technique used above does not let us conclude anything about the number Q. In analogy with (4.1), one may ask the following

Open Question 1. *Is the number*

$$Q = \prod_{n \geq 1} \left(\frac{2n}{2n+1} \right)^{(-1)^{t_n}} \doteq 1.6281601297189$$

algebraic?
No simple formula for the number Q is known, although it appears in a somewhat disguised form in a paper of Flajolet and Martin [39, Theorem 3.A], where (using their notation) $\varphi = 2^{-1/2} e^{\gamma} Q^{-1}$.

5. Automatic sequences. The Thue-Morse sequence is a member of a much larger class of sequences called *k-automatic sequences*; more precisely, the Thue-Morse sequence is 2-automatic.

Let us recall the basics of finite automata. A *deterministic finite automaton*, or DFA, is a simple model of a computer. Formally it is a quintuple, $M = (Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set of *states*;
- Σ is a finite set of symbols, called the *input alphabet*;
- $q_0 \in Q$ is the *initial state*;
- $F \subseteq Q$ is the set of *final states*;
- $\delta : Q \times \Sigma \to Q$ is the *transition function*.

The transition function δ is extended in the obvious way to a map from $Q \times \Sigma^*$ into Q.

The *language accepted by* M is denoted by $L(M)$ and is given by $\{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. As an example, consider the automaton in Figure 5.1, which accepts exactly the strings over $\{0, 1\}$ that are the base-2 representations of the primes between 2 and 11.

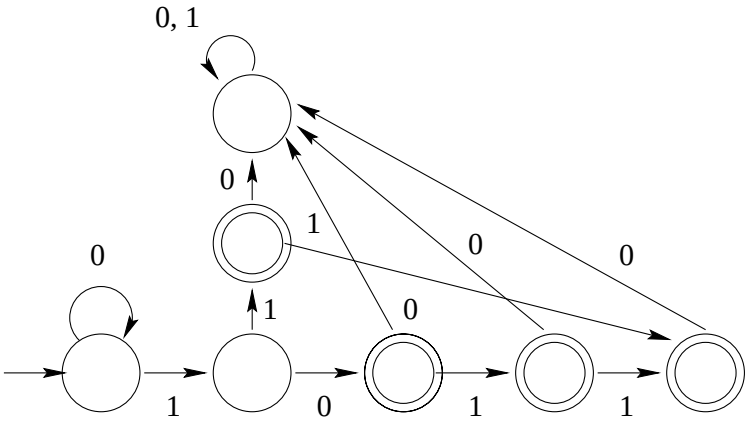


FIG. 5.1. *Automaton accepting the base-2 representations of the primes p where* $2 \leq p \leq 11$

Note that the start state is at the lower left, and is indicated, as is customary, by an unlabeled arrow with no source. Also, final states are denoted by double circles.

We may also provide our automaton with output. In this case we discard the set of final states from the definition of the DFA and add back Δ (the output alphabet) and $\tau : Q \to \Delta$ is the output mapping.

DEFINITION 5.1. *We say a sequence $(s_i)_{i\geq 0}$ over a finite alphabet Δ is k-automatic if there exists a deterministic finite automaton with output (DFAO) $M = (Q, \Sigma_k, \Delta, \delta, \tau, q_0)$ (where τ is a mapping taking Q to Δ) such that $\tau(\delta(q_0, (n)_k)) = s_n$ for all $n \geq 0$.*

These sequences are sometimes called *uniform tag sequences* [27] or k-recognizable sequences [37, p. 106] in the literature.

Another characterization of automatic sequences is the following. Suppose $(s(n))_{n\geq 0}$ is a sequence over a finite alphabet. Define $K_k(s)$, the k-kernel of s, to be the set of subsequences

$$K_k(s) = \{(s(k^i n + a))_{n\geq 0} \ : \ i \geq 0, \ 0 \leq a < k^i\}.$$

Then $(s(n))_{n\geq 0}$ is k-automatic iff the set $K_k(s)$ is finite.

Many sequences that occur in number theory turn out to be k-automatic for some small integer k. For example, let B be an integer ≥ 3, and consider the real number $f(B) = \sum_{k\geq 0} B^{-2^k}$. This is a transcendental number[2] ([53,15,71,68,56]; [76, Thm. 1.1.2]) whose continued fraction has bounded partial quotients [88,34]:

$$\begin{aligned} f(B) &= [a_0, a_1, a_2, \ldots] \\ &= [0, \ B-1, \ B+2, \ B, \ B, \ B-2, \ B, \ B+2, \ B, \ \ldots]. \end{aligned}$$

In fact, its continued fraction can be generated by the simple finite automaton with ten states in Figure 5.2.

For example, to compute a_{12}, we compute $(12)_2 = 1100$, and then feed the digits into the automaton, starting at the top. The output is the label of the last state reached, which is $B - 2$.

Probably the most interesting and useful number-theoretic aspect of automatic sequences is the following theorem of Christol [23,24]:

THEOREM 5.1. *Let Δ be a nonempty finite set, $(a_i)_{i\geq 0}$ be a sequence over Δ, and p be a prime number. Then $(a_n)_{n\geq 0}$ is p-automatic iff there exists an integer $m \geq 1$ and an injection $\beta : \Delta \rightarrow GF(p^m)$ such that the formal power series $\sum_{n\geq 0} \beta(a_n) X^n$ is algebraic over $GF(p^m)(X)$.*

As an example, consider the Thue-Morse sequence $(t_n)_{n\geq 0}$, which is 2-automatic. Let $T(X) = \sum_{n\geq 0} t_n X^n$.

$$T(X) = X + X^2 + X^4 + X^7 + X^8 + X^{11} + \cdots$$

Now

$$\begin{aligned} T(X) &= \sum_{n\geq 0} t_n X^n \\ &= \sum_{n\geq 0} t_{2n} X^{2n} + \sum_{n\geq 0} t_{2n+1} X^{2n+1} \end{aligned}$$

[2] Sometimes called the 'Fredholm number', although Fredholm apparently never worked on it.

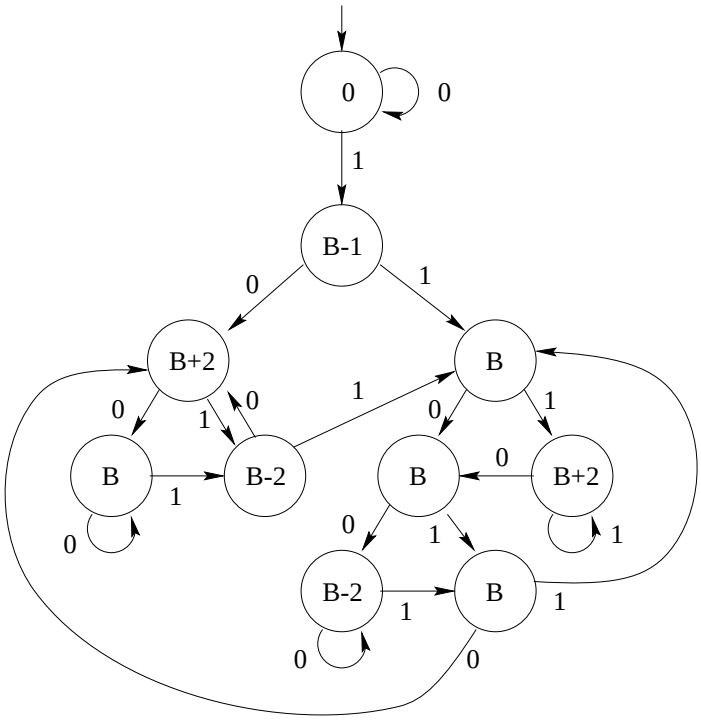


FIG. 5.2. *Automaton generating the continued fraction expansion of* $f(B)$

$$\begin{aligned} &= \sum_{n\geq 0} t_n X^{2n} + X\sum_{n\geq 0}(t_n+1)X^{2n} \\ &= T(X^2) + XT(X^2) + X\frac{1}{1-X^2}. \end{aligned}$$

Hence we have, over $GF(2)$,

$$(1+X)^3T(X)^2 + (1+X)^2T(X) + X = 0.$$

The theorem of Christol is remarkable because it relates a purely number-theoretic fact (algebraicity in finite characteristic) to a purely machine-theoretic fact (generation by a finite automaton). As a consequence, one may obtain transcendence results in finite characteristic by proving that no finite automaton can generate the sequence of coefficients of an appropriate formal power series. For example, Allouche [2] used this technique to give a new proof of the transcendence of π_q, the analogue of π in the field of formal Laurent series over $GF(q)$.

Other results along this line include those of Berthé [11,12], who proved that $\frac{\zeta_q(n)}{\pi_q^n}$ is transcendental for $1 \leq n \leq q-2$, a result previously proved

by Yu [105] for every n such that $(q-1) \not| n$. Here ζ_q is the Carlitz zeta-function, the formal power series analogue of the ordinary zeta-function. Recher [84] obtained transcendence results for periods of generalized Carlitz exponentials, i.e., of generalizations of π_q. Berthé [13] proved transcendence results for the Carlitz logarithm and gave results on linear expressions in $\frac{\zeta_q(n)}{\pi_q^n}$ for $1 \leq n \leq q-2$ [14]. Allouche [3] proved the transcendence of the values of the Carlitz-Goss gamma function for all p-adic rational arguments that are not natural numbers, and Mendès France and Yao [73] extended the result to *all* the values of the Carlitz-Goss gamma function at p-adic arguments that are not natural numbers. Thakur proved [98] that the period of the Tate elliptic curve is transcendental.

6. Automatic real numbers. Given a k-automatic sequence $(s_i)_{i \geq 0}$ over the alphabet $\Sigma = \{0, 1, 2, \ldots, b-1\}$, we may consider the sequence to represent the base-b representation of a real number. The number $\sum_{i \geq 0} b^{-2^i}$ is an example of such a number, discussed in the previous section.

Or consider the Thue-Morse real number $\sum_{i \geq 1} t_{i-1} 2^{-i}$, whose base-2 representation is

$$\mathcal{T} = .0110100110010110 \cdots.$$

It follows from a general result of Mahler [71] that $\mathcal{T}$ is transcendental. Mahler's proof technique was later rediscovered by Cobham [26] and Dekking [30].[3]

It may be amusing to note that the number $\mathcal{T}$ appears "naturally" as a certain probability in formal language theory. Let $\mathcal{P}$ be the probability that a randomly-chosen language over $\{0, 1\}$ contains at least one word of every possible length. (Our model is to decide the membership of each word in L uniformly at random, with probability $\frac{1}{2}$.) Then

$$\mathcal{P} = \prod_{i \geq 0} (1 - 2^{-2^i}) = \sum_{j \geq 0} \frac{(-1)^{t_j}}{2^j} = \sum_{j \geq 0} \frac{1 - 2t_j}{2^j} = 2 - 4\mathcal{T}.$$

This result suggests the following

CONJECTURE 2. *Let k, b be integers ≥ 2. If $(s_i)_{i \geq 0}$ is a non-ultimately-periodic k-automatic sequence over the alphabet $\Sigma = \{0, 1, 2, \ldots, b-1\}$, then the number $\sum_{i \geq 0} s_i b^{-i}$ is transcendental.*

For some time it was believed that Loxton and van der Poorten had completely resolved this problem [69,70], but gaps in the proof have been pointed out by Paul-Georg Becker.

CONJECTURE 3. *No number of the form $\sum_{i \geq 0} s_i b^{-i}$, where $(s_i)_{i \geq 0}$ is a k-automatic sequence, and b is an integer ≥ 2, is a Liouville number.*

[3] Michel Dekking has kindly pointed out a minor, easily-repairable flaw in his proof.

Becker conjectures (personal communication, 1993) that in fact these numbers, when transcendental, are S-numbers in Mahler's classification ([72], [58, p. 63]).

Recently there have been some other interesting results on real numbers whose base-b expansions are k-automatic. Denoting the set of such numbers as $L(k,b)$, we have the following theorem of Lehr [63]:

THEOREM 6.1. *The set $L(k,b)$ forms a $\mathbb{Q}$-vector space.*

However, it can be shown that the set $L(k,b)$ is not closed under product; that is, $L(k,b)$ is not a ring [64]. The structure of $L(k,b)$ is still somewhat mysterious, although it is known that $L(k,b)$ is infinite dimensional over $\mathbb{Q}$. In fact, for each $B \geq 2$, we have $\mathbb{Q}[f(B)] \subset L(2,B)$, where f is the function defined in Section 5. Since $f(B)$ is transcendental over $\mathbb{Q}$, we have $\mathbb{Q}[f(B)]$ is infinite dimensional over $\mathbb{Q}$. See [64].

It would be nice to prove that some classical real numbers are not automatic numbers. For example, we have

CONJECTURE 4. *The numbers π, e, and $\ln 2$ are not in $L(k,b)$ for any $k, b \geq 2$.*

This conjecture would follow, for example, if it were proved that these numbers were normal.

7. Fixed points of homomorphisms. As Cobham observed [27], the k-automatic sequences discussed in the previous section can also be characterized as images (under a length-preserving homomorphism, or *coding*) of fixed points of *uniform* homomorphisms (i.e., homomorphisms φ with $|\varphi(a)| = k$ for all $a \in \Sigma$). For example, the Thue-Morse word is the unique fixed point, starting with 0, of the map which sends 0 to 01 and 1 to 10.

One can also study the fixed points of homomorphisms that are not necessarily uniform. The *depth* of a homomorphism $\varphi : \Sigma \to \Sigma^*$ is defined to be $|\Sigma|$, and the *width* is $\max_{a \in \Sigma} |\varphi(a)|$.

Suppose that $\varphi : \Sigma \to \Sigma^*$ is a homomorphism with the property that $\varphi(a) = ax$ for some letter $a \in \Sigma$. (We call such a homomorphism *extendable* on a.) Then

$$ax\varphi(x)\varphi^2(x)\varphi^3(x)\cdots$$

is a fixed point of φ, and if x contains at least one letter which is not ultimately sent to ϵ by repeated applications of φ, then this fixed point is infinite.

OPEN QUESTION 5. *Given a homomorphism φ extendable on a, of depth m and width n, can one compute the ith letter of the fixed point starting with a in time polynomial in m, n, and $\log i$?*

Note that this question is easily answerable in the affirmative when the homomorphism is uniform.

A particular fixed point that has been studied extensively is the so-called infinite *Fibonacci word*

$$\mathbf{f} = f_1 f_2 f_3 \cdots = 0100101001001\cdots,$$

which is the fixed point of the map $\varphi(0) = 01$ and $\varphi(1) = 0$ [9,10]. It can be shown that

$$f_n = 1 - \lfloor (n+1)\alpha \rfloor + \lfloor n\alpha \rfloor,$$

where $\alpha = (\sqrt{5} - 1)/2$.

One may generalize the concept of fixed points of homomorphisms by considering fixed points of finite-state transducers. The most famous example of this type is the Kolakoski word [59]

$$\mathbf{k} = 122112122122112112212112122\cdots$$

which is a fixed point of the transducer in Figure 7.1.

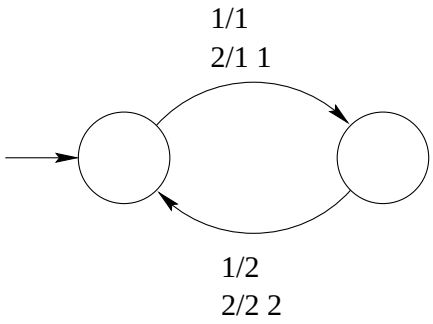


FIG. 7.1. *The Kolakoski transducer*

Despite much work on this sequence (e.g., [54,31,32,102,52,29,28] and [79,20,66,25,21,33,96]), the following conjecture is still open:

CONJECTURE 6. *The limiting frequencies of* 1 *and* 2 *in* $\mathbf{k}$ *exist, and are equal to* $\frac{1}{2}$.

8. Automaticity. In Section 5 we discussed languages that are accepted by finite automata and sequences that are generated by finite automata. However, "most" languages and sequences are not of this type. For the rest of these languages and sequences, can we somehow evaluate how "close" these objects are to being regular or automatic?

In this section, we introduce a measure of descriptional complexity called *automaticity*. Our complexity measure is a *function*, and is designed so that regular languages have $O(1)$ automaticity, and languages "close" to regular have "small" automaticity.

Let

$$\Sigma^{\leq n} = \epsilon + \Sigma + \Sigma^2 + \cdots + \Sigma^n,$$

the set of all strings in Σ^* of length $\leq n$. We say a language $L \subseteq \Sigma^*$ is an *nth order approximation* to a language L' if $L \cap \Sigma^{\leq n} = L' \cap \Sigma^{\leq n}$. Let DFA be the class of all deterministic finite automata over a finite alphabet Σ. We can now informally define the automaticity of a language L to be the function which counts the number of states in the smallest DFA that accepts some nth order approximation to L. Formally, if $|M|$ is defined to be the number of states in the DFA M, we define the automaticity $A_L(n)$ of a language L as follows:

$$A_L(n) = \min\{|M| \ : \ M \in \text{DFA} \ \text{ and } L(M) \cap \Sigma^{\leq n} = L \cap \Sigma^{\leq n}\}.$$

The following basic properties of the function $A_L(n)$ are easy to prove:

1. $A_L(n) \leq A_L(n+1)$.
2. L is regular iff $A_L(n) = O(1)$.
3. $A_L(n) = A_{\overline{L}}(n)$.
4. $A_L(n) \leq 2 + \Sigma_{w \in L \cap \Sigma^{\leq n}} |w|$.

We now make the following

DEFINITION 8.1. *Two strings w, w' are called* n–dissimilar for L *if there exists a string v with $|wv|, |w'v| \leq n$ and either*

(i) $wv \in L$, $w'v \notin L$; or

(ii) $wv \notin L$, $w'v \in L$.

Then we have [36,50,94]:

THEOREM 8.1. *$A_L(n)$ = the maximum number of distinct pairwise n–dissimilar strings for L.*

As an example, consider the language

$$L = \{0^n 1^n \ : \ n \geq 0\}.$$

This language is clearly not regular. What is its automaticity?

It can be shown that the automaticity of L is $A_L(n) = 2\lfloor n/2 \rfloor + 1$ for $n \geq 2$. To see the upper bound, note that we can accept an nth order approximation to L (for $n = 9$) with DFA in Figure 8.1.

To get the lower bound for $n = 9$, note that we may take

$$\{\epsilon,\ 0,\ 00,\ 000,\ 0000,\ 1,\ 01,\ 001,\ 0001\}$$

as our set of n-dissimilar strings. This easily generalizes to larger n.

Now, let's turn to another example. Consider the set

$$P = \{10, 11, 101, 111, 1011, 1101, 10001, 10011, \ldots\},$$

the set of primes expressed in base 2. A classical (1966) theorem due to Minsky and Papert [74] shows that P is not a regular language. However, this raises the question, how "far" from regular is P? We have the following theorem [92]:

THEOREM 8.2. *The automaticity of P^R is $\Omega(2^{n/43})$.*

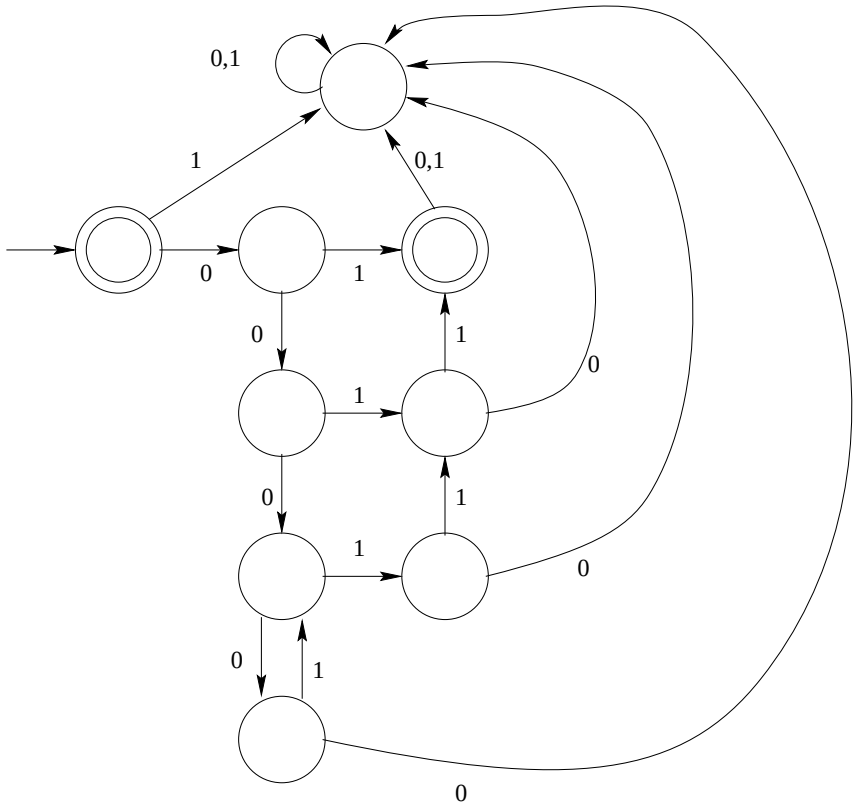


FIG. 8.1. *Automaton acccepting 9th order approximation to L*

(Here P^R denotes the reversal of the set P, i.e., the primes expressed with least significant digit first.)

The basic idea is to prove the following

LEMMA 8.1. *Given integers r, a, b with $r \geq 2$, $1 \leq a, b < r$ with $\gcd(r, a) = \gcd(r, b) = 1$, and $a \neq b$, there exists $m = O(r^{165/4})$ such that $rm + a$ is prime and $rm + b$ is composite.*

The proof of this lemma is an easy consequence of a deep theorem of Heath-Brown [47] on the distribution of primes in arithmetic progressions ("Linnik's Theorem").

Taking $r = 2^n$, the lemma implies that there are at least $2^{n/43}$ n-dissimilar strings for the language P^R.

Automaticity has been examined by Trakhtenbrot [100]; Grinberg & Korshunov [45]; Karp [51]; Breitbart [16,17,18]; Dwork and Stockmeyer [36]; Kaneps & Freivalds [50]; Shallit & Breitbart [93,94], Pomerance, Robson, & Shallit [80], Glaister & Shallit [42], and Shallit [92]. Koskas and de Mathan (work in progress, 1996) show how to apply automaticity to obtain

irrationality measures in finite characteristic.

One of the nicest results in the area is Karp's theorem [51]:

THEOREM 8.3. *Let $L \subseteq \Sigma^*$ be a nonregular language. Then*

$$A_L(n) \geq (n+3)/2$$

for infinitely many n.

It can be shown that the constants 3 and 2 in Karp's theorem are best possible, in the sense that the theorem would be false if 2 were replaced with any smaller number, or if 3 were replaced with any larger number [94].

The case of unary alphabets has only recently begun to be studied. In this case, we have $A_L(n) \leq n+1$, for all L and for all n. The following theorems can be proved [80]:

THEOREM 8.4. *Let $L \subseteq 0^*$. Then*

$$A_L(n) \leq n + 1 - \lfloor \log_2 n \rfloor$$

for infinitely many n.

THEOREM 8.5. *Let $L \subseteq 0^*$. Then for "almost all" L we have*

$$A_L(n) > n - 2\log_2 n - 2\log_2 \log_2 n$$

for all sufficiently large n.

Recall that Karp proved that if L is not regular, then $A_L(n) \geq (n+3)/2$ infinitely often. This implies that

$$\limsup_{n\to\infty} \frac{A_L(n)}{n} \geq \frac{1}{2}$$

for all nonregular L. However, it seems that one can do better in the unary case. In 1994, I made the following conjecture [93,80]:

CONJECTURE 7. *There exists a real number $\gamma > 1/2$ such that if $L \subseteq 0^*$ is not regular, then*

$$\limsup_{n\to\infty} \frac{A_L(n)}{n} \geq \gamma.$$

In fact, I had conjectured that $\gamma = (\sqrt{5}-1)/2 \doteq .61803$. However, recently J. Cassaigne has shown that the proper constant is

$$\gamma = (60 - 2\sqrt{10})/89 \doteq .60309$$

and this constant is best possible [22]. (Partial results had previously been obtained by Allouche and Bousquet-Mélou [4].)

Finally, it is known that the maximum possible automaticity for a language $L \subseteq (0+1)^*$ is $O(2^n/n)$. An example of a context-free language (CFL) with automaticity $\Omega(2^n/n)$ is not known, although there are examples with automaticity $\Omega(2^{n^{(1-\epsilon)}})$ for all $\epsilon > 0$ [42]. This suggests the following open problem:

OPEN PROBLEM 8. *Develop an efficient algorithm for computing the automaticity of a CFL, given its representation as a context-free grammar.*

8.1. Nondeterministic Automaticity. Let NFA be the class of all nondeterministic finite automata.

A *nondeterministic finite automaton (NFA)* is like a deterministic one, except now there can be $0, 1, 2$, or more arrows with the same label leaving any state. A string w is accepted by an NFA if there exists some path labeled w from the initial state to some final state.

The function $N_L(n)$ is the *nondeterministic automaticity* of the language L, where

$$N_L(n) = \min\{|M| \ : \ M \in \text{NFA} \ \text{ and } L(M) \ \cap \ \Sigma^{\leq n} = L \ \cap \ \Sigma^{\leq n}\}.$$

Then by the classical subset construction, we have

THEOREM 8.6. *Suppose $L \subseteq \Sigma^*$. If L is not regular, then $N_L(n) \geq \log_2((n+3)/2)$ for infinitely many n.*

This lower bound is best possible, up to a constant, since the Stearns-Hartmanis-Lewis language

$$\{2\ (1)_2^R\ 2\ (2)_2^R\ 2\ (3)_2^R\ 2\ (4)_2^R\ 2 \cdots\ 2\ (n)_2^R : n \geq 1\}$$

has nondeterministic automaticity $O(\log n)$. Here, as in Section 3, $(k)_2$ is the representation of k in base 2, and w^R denotes the reversal of the string w.

We can use some classical estimates from number theory to produce an example of a language with low nondeterministic automaticity [94]:

THEOREM 8.7. *Define*

$$L = \{w \in (0+1)^* \ : \ |w|_0 \neq |w|_1\}.$$

Then L is nonregular and

$$N_L(n) = O((\log n)^2/(\log\log n)).$$

Proof. We need the following fact from number theory:

LEMMA 8.2. *Let $n \geq 2$ and suppose $0 \leq i, j < n$. Then $i \neq j$ iff there exists a prime $p \leq 4.4 \log n$ such that $i \not\equiv j \pmod p$.*

Thus, to nondeterministically accept some nth order approximation to L, we can

- guess the correct prime $p \leq 4.4 \log n$;
- verify that $|w|_0 \not\equiv |w|_1 \pmod p$.

This construction uses at most

$$1 + \sum_{p \leq 4.4 \log n} p = O((\log n)^2/(\log\log n))$$

states. The construction is illustrated in Figure 8.2.

We now turn to the question of lower bounds for nondeterministic automaticity in the unary case [80]:

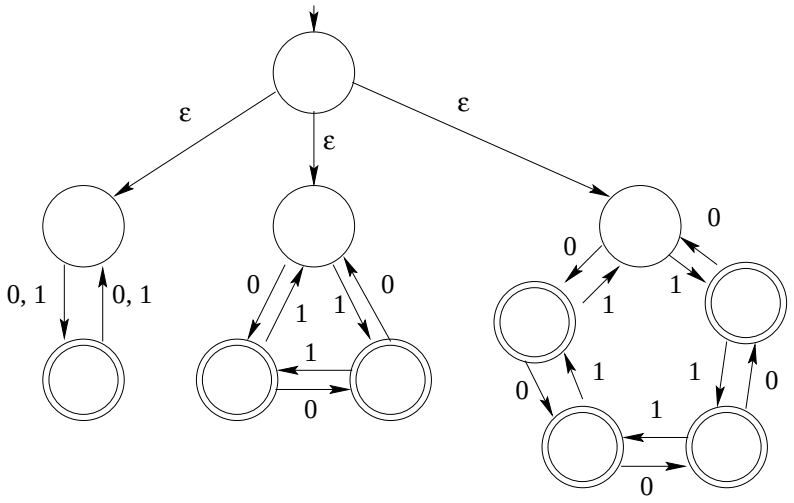


FIG. 8.2. *30th order approximation to L*

THEOREM 8.8. *There exists a constant c (which does not depend on L) such that if $L \subseteq 0^*$ is not regular, then*

$$N_L(n) \geq c(\log n)^2/(\log\log n)$$

infinitely often.

Pomerance has shown [80] that for all monotonically increasing functions f, there exists a language $L = L(f)$ such that

$$N_L(n) = O(f(n)(\log n)^2/(\log\log n)),$$

thus showing the lower bound is essentially tight. To give the flavor of his construction, we prove the following weaker result:

THEOREM 8.9. *Define $L = \{0^n : n \geq 1$ and the least positive integer not dividing n is not a power of $2\}$. Then L is nonregular and*

$$N_L(n) = O((\log n)^3/(\log\log n)).$$

Proof. The construction depends on the following two facts:

LEMMA 8.3. *If $0^n \in L$, then there exists a prime power p^k, $p \geq 3$, $k \geq 1$, $p^k \leq 5\log n$, such that $n \not\equiv 0 \pmod{p^k}$, and $n \equiv 0 \pmod{2^s}$, with $2^s < p^k < 2^{s+1}$. Further, if such a prime power p^k exists, then $0^n \in L$.*

An NFA accepting an n-th order approximation to L can now be constructed as follows:

- guess the correct odd prime power $p^k \leq 5\log n$;
- verify that, on input 0^r, we have
  - $r \not\equiv 0 \pmod{p^k}$;

$*$ $r \equiv 0 \pmod{2^s}$, with $2^s < p^k < 2^{s+1}$.

This construction uses at most $O((\log n)^3/(\log\log n))$ states.

OPEN QUESTION 9. *What is a good lower bound on the nondeterministic automaticity of the set P^R, the (reversed) representations of primes in base 2?*

9. k-regular sequences. The last topic I wish to consider in this survey is k-regular sequences. These are generalizations of the automatic sequences mentioned above in Section 5.

While there are many examples of automatic sequences in number theory, their expressive power is somewhat limited because of the requirement that they take only a finite number of values. How can this be generalized? As we have seen above in Section 5, a sequence is k-automatic iff its k-kernel is finite. This suggests studying the class of sequences where the $\mathbb{Z}$-module generated by the k-kernel is *finitely generated*. We call such a sequence k-regular. The properties of such sequences and many examples were given in [6].

Here are some examples of k-regular sequences in number theory.

Example 1. The 3-adic valuation of a sum of binomial coefficients. Let $r(n) := \sum_{0 \le i < n} \binom{2i}{i}$. Then $\nu_3(r(n))$ is 3-regular, as it can be shown that

$$\nu_3(r(n)) = \nu_3\left(n^2 \binom{2n}{n}\right); \tag{9.1}$$

see [97]. In fact, Eq. (9.1) was first conjectured by applying a program which attempts to deduce the k-regularity of a given sequence. Zagier [106] found a beautiful proof based on 3-adic analysis.

Example 2. Propp's sequence. Jim Propp [81] introduced the sequence $(s(n))_{n \ge 0}$, defined to be the unique monotone sequence such that $s(s(n)) = 3n$. The table below gives the first few terms:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$s(n)$	0	2	3	6	7	8	9	12	15	18	19	20	21	22

It is sequence M0747 in the book of Sloane and Plouffe [95]. Patruno [78] showed that

$$s(n) = \begin{cases} n + 3^k, & \text{if } 3^k \le n < 2 \cdot 3^k; \\ 3(n - 3^k), & \text{if } 2 \cdot 3^k \le n < 3^{k+1}. \end{cases}$$

This sequence is 3-regular, and satisfies the recurrence

$$\begin{aligned} s(3n) &= 3s(n); \\ s(9n+1) &= 6s(n) + s(3n+1); \end{aligned}$$

$$\begin{aligned}
s(9n+2) &= 6s(n) + s(3n+2);\\
s(9n+4) &= 2s(3n+1) + s(3n+2);\\
s(9n+5) &= s(3n+1) + s(3n+2);\\
s(9n+7) &= -6s(n) + 3s(3n+1) + 2s(3n+2);\\
s(9n+8) &= -12s(n) + 6s(3n+1) + s(3n+2).
\end{aligned}$$

Example 3. A greedy partition of the natural numbers into sets avoiding arithmetic progressions. Suppose we consider the integers $0, 1, 2, \ldots$ in turn, and place each new integer i into the set of lowest index S_k ($k \geq 0$) so that S_k never contains three integers in arithmetic progression. For example, we put 0 and 1 in S_0, but placing 2 in S_0 would create an arithmetic progression of size 3 (namely, $\{0, 1, 2\}$), so we put 2 in S_1, etc.

Now define the sequence $(a_k)_{k\geq 0}$ as follows: $a_k = n$ if k is placed into set S_n. Here are the first few terms of this sequence:

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13
a_k	0	0	1	0	0	1	1	2	2	0	0	1	0	0

This is Sloane and Plouffe's sequence M0185.

Gerver, Propp, and Simpson [41] showed that $a_{3k+r} = \lfloor (3a_k + r)/2 \rfloor$ for $k \geq 0$, $0 \leq r < 3$. It follows that $(a_k)_{k\geq 0}$ is 3-regular.

We now give some open problems on k-regular sequences.

CONJECTURE 10. *Suppose $(A(n))_{n\geq 0}$ and $(B(n))_{n\geq 0}$ are k-regular sequences with $B(n) \neq 0$ for all n. If $A(n)/B(n)$ is always an integer, then $(A(n)/B(n))_{n\geq 0}$ is also k-regular.*

OPEN QUESTION 11. *Show that $(\lfloor \frac{1}{2} + \log_2 n \rfloor)_{n\geq 0}$ is not a 2-regular sequence.*

We may also consider an extension of k-regular sequences to other types of representation; e.g., Fibonacci representation. Let us consider, for example, the problem of determining the number of partitions k_n of a number n as a sum of distinct Fibonacci numbers [55,19,86]. In other words, we are interested in the coefficient k_n of X^n in the infinite product

$$(1+X)(1+X^2)(1+X^3)(1+X^5)(1+X^8)(1+X^{13})\cdots.$$

Here are the first few terms of this sequence:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
k_n	1	1	1	2	1	2	2	1	3	2	2	3	1	3

Then it is not hard to see that

$$k_n = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot M_{w^R} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \tag{9.2}$$

where w is the Fibonacci expansion of n, and

$$M_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix}; \quad M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}. \tag{9.3}$$

In particular, this allows computation of k_n in time polynomial in $\log n$, and gives a simple proof of Theorem 1 of [86].

10. Conclusions. Both number theory and formal language theory have a large body of research associated with them. At their intersection, however, is a new and growing area which promises to enrich them both.

11. Acknowledgments. Jean-Paul Allouche read a draft of this survey and made many helpful comments. I also express my gratitude to the referee, who read this survey with care and corrected several errors.

I would like to express my appreciation to the Institute for Mathematics and its Applications at the University of Minneapolis, where he spent a very pleasant and productive week in the summer of 1996 as an invited speaker at the conference entitled Emerging Applications of Number Theory.

REFERENCES

[1] J.-P. Allouche. Automates finis en théorie des nombres. *Exposition. Math.* **5** (1987), 239–266.
[2] J.-P. Allouche. Sur la transcendance de la série formelle π. *Séminaire de Théorie des Nombres de Bordeaux* **2** (1990), 103–117.
[3] J.-P. Allouche. Transcendence of the Carlitz-Goss Gamma function at rational arguments. *J. Number Theory* **60** (1996), 318–328.
[4] J.-P. Allouche and M. Bousquet-Mélou. On the conjectures of Rauzy and Shallit for infinite words. *Comment. Math. Univ. Carolinae* **36** (1995), 705–711.
[5] J.-P. Allouche and H. Cohen. Dirichlet series and curious infinite products. *Bull. Lond. Math. Soc.* **17** (1985), 531–538.
[6] J.-P. Allouche and J. O. Shallit. The ring of k–regular sequences. *Theoret. Comput. Sci.* **98** (1992), 163–187.
[7] P. Arnoux. Some remarks about Fibonacci multiplication. *Appl. Math. Letters* **2** (1989), 319–320.
[8] E. Bach and J. Shallit. *Algorithmic Number Theory.* MIT Press, 1996.
[9] J. Berstel. Mots de Fibonacci. In *Séminaire d'Informatique Théorique*, pages 57–78. Laboratoire Informatique Théorique, Institut Henri Poincaré, 1980/81.
[10] J. Berstel. Fibonacci words—a survey. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 13–27. Springer-Verlag, 1986.
[11] V. Berthé. De nouvelles preuves "automatiques" de transcendance pour la fonction zêta de Carlitz. In D. F. Coray and Y.-F. S. Pétermann, editors, *Journées Arithmétiques de Genève*, Vol. 209 of *Astérisque*, pages 159–168, 1992.
[12] V. Berthé. Fonction ζ de Carlitz et automates. *J. Théorie Nombres Bordeaux* **5** (1993), 53–77.
[13] V. Berthé. Automates et valeurs de transcendance du logarithme de Carlitz. *Acta Arith.* **66** (1994), 369–390.
[14] V. Berthé. Combinaisons linéaires de $\zeta(s)/\pi^s$ sur $\mathbb{F}_q(x)$, pour $1 \leq s \leq q-2$. *J. Number Theory* **53** (1995), 272–299.
[15] H. Blumberg. Note on a theorem of Kempner concerning transcendental numbers. *Bull. Amer. Math. Soc.* **32** (1926), 351–356.

[16] Y. Breitbart. Realization of boolean functions by finite automata. *NTL (Novosti Technicheskoi Literature), Seria Automatica, Telemechanika i Priborostroenie* No. 4 (1970).

[17] Y. Breitbart. On automaton and "zone" complexity of the predicate "to be a kth power of an integer". *Dokl. Akad. Nauk SSSR* **196** (1971), 16–19. In Russian. English translation in *Soviet Math. Dokl.* **12** (1971), 10–14.

[18] Y. Breitbart. *Complexity of the calculation of predicates by finite automata.* PhD thesis, Technion, Haifa, Israel, June 1973.

[19] L. Carlitz. Fibonacci representations. *Fibonacci Quart.* **6** (1968), 193–220.

[20] A. Carpi. Repetitions in the Kolakovski [sic] sequence. *Bull. European Assoc. Theor. Comput. Sci.* (50) (1993), 194–196.

[21] A. Carpi. On repeated factors in C^∞-words. *Inform. Process. Lett.* **52** (1994), 289–294.

[22] J. Cassaigne. On a conjecture of J. Shallit. In *Proc. 24th Int'l. Conf. on Automata, Languages, and Programming (ICALP)*, Vol. 1256 of *Lecture Notes in Computer Science*, pages 693–704. Springer-Verlag, 1997.

[23] G. Christol. Ensembles presque périodiques k-reconnaissables. *Theoret. Comput. Sci.* **9** (1979), 141–145.

[24] G. Christol, T. Kamae, M. Mendès France, and G. Rauzy. Suites algébriques, automates et substitutions. *Bull. Soc. Math. France* **108** (1980), 401–419.

[25] V. Chvátal. Notes on the Kolakoski sequence. Technical Report 93-84, DIMACS, March 1994. Revised.

[26] A. Cobham. A proof of transcendence based on functional equations. Technical Report RC-2041, IBM Yorktown Heights, March 25 1968.

[27] A. Cobham. Uniform tag sequences. *Math. Systems Theory* **6** (1972), 164–192.

[28] K. Culik II and J. Karhumäki. Iterative devices generating infinite words. In *STACS 92, Proc. 9th Symp. Theoretical Aspects of Comp. Sci.*, Vol. 577 of *Lecture Notes in Computer Science*, pages 531–543. Springer-Verlag, 1992.

[29] K. Culik II, J. Karhumäki, and A. Lepistö. Alternating iteration of morphisms and the Kolakovski [sic] sequence. In G. Rozenberg and A. Salomaa, editors, *Lindenmayer Systems*, pages 93–103. Springer-Verlag, 1992.

[30] F. M. Dekking. Transcendance du nombre de Thue-Morse. *C. R. Acad. Sci. Paris* **285** (1977), 157–160.

[31] F. M. Dekking. Regularity and irregularity of sequences generated by automata. In *Séminaire de Théorie des Nombres de Bordeaux*, pages 9.01–9.10, 1979–1980.

[32] F. M. Dekking. On the structure of self-generating sequences. In *Séminaire de Théorie des Nombres de Bordeaux*, pages 31.01–31.06, 1980–1981.

[33] F. M. Dekking. What is the long range order in the Kolakoski sequence? Technical report, Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1995.

[34] F. M. Dekking, M. Mendès France, and A. J. van der Poorten. Folds! *Math. Intelligencer* **4** (1982), 130–138, 173–181, 190–195.

[35] U. Dudley. Smith numbers. *Math. Mag.* **67** (1994), 62–65.

[36] C. Dwork and L. Stockmeyer. On the power of 2-way probabilistic finite state automata. In *Proc. 30th Ann. Symp. Found. Comput. Sci.*, pages 480–485. IEEE Press, 1989.

[37] S. Eilenberg. *Automata, Languages, and Machines*, Vol. A. Academic Press, 1974.

[38] P. Enflo, A. Granville, J. Shallit, and S. Yu. On sparse languages L such that $LL = \Sigma^*$. *Disc. Appl. Math.* **52** (1994), 275–285.

[39] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. System Sci.* **31** (1985), 182–209.

[40] R. R. Forslund. A logical alternative to the existing positional number system. *Southwest J. Pure Appl. Math.* **1** (1995), 27–29.

[41] J. Gerver, J. Propp, and J. Simpson. Greedily partitioning the natural numbers

into sets free of arithmetic progressions. *Proc. Amer. Math. Soc.* **102** (1988), 765–772.

[42] I. Glaister and J. Shallit. Automaticity III: Polynomial automaticity, context-free languages, and fixed points of morphisms. To appear, *Computational Complexity*, 1996.

[43] P. J. Grabner, A. Pethö, R. F. Tichy, and G. J. Woeginger. Associativity of reccurrence multiplication. *Appl. Math. Letters* **7**(4) (1994), 85–90.

[44] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 1989.

[45] V. S. Grinberg and A. D. Korshunov. Asymptotic behavior of the maximum of the weight of a finite tree. *Problemy Peredachi Informatsii* **2** (1966), 96–99. In Russian. English translation in *Problems of Information Transmission* **2** (1966), 75–78.

[46] G. H. Hardy. *A Mathematician's Apology*. Cambridge University Press, 1967.

[47] D. R. Heath-Brown. Zero-free regions for Dirichlet L-functions and the least prime in an arithmetic progression. *Proc. Lond. Math. Soc.* **64** (1992), 265–338.

[48] M. Hollander. Greedy numeration systems and recognizability. Unpublished manuscript, 1995.

[49] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[50] J. Kaneps and R. Freivalds. Minimal nontrivial space complexity of probabilistic one-way Turing machines. In B. Rovan, editor, *MFCS '90 (Mathematical Foundations of Computer Science)*, Vol. 452 of *Lecture Notes in Computer Science*, pages 355–361. Springer-Verlag, 1990.

[51] R. M. Karp. Some bounds on the storage requirements of sequential machines and Turing machines. *J. Assoc. Comput. Mach.* **14** (1967), 478–489.

[52] M. S. Keane. Ergodic theory and subshifts of finite type. In T. Bedford, M. Keane, and C. Series, editors, *Ergodic Theory, Symbolic Dynamics, and Hyperbolic Spaces*, pages 35–70. Oxford University Press, 1991.

[53] A. J. Kempner. On transcendental numbers. *Trans. Amer. Math. Soc.* **17** (1916), 476–482.

[54] C. Kimberling. Advanced problem 6281. *Amer. Math. Monthly* **86** (1979), 793.

[55] D. Klarner. Partitions of N into distinct Fibonacci numbers. *Fibonacci Quart.* **6** (1968), 235–243.

[56] M. J. Knight. An "ocean of zeros" proof that a certain non-Liouville number is transcendental. *Amer. Math. Monthly* **98** (1991), 947–949.

[57] D. E. Knuth. Fibonacci multiplication. *Appl. Math. Letters* **1** (1988), 57–60.

[58] J. F. Koksma. *Diophantische Approximationen*. Springer, 1936.

[59] W. Kolakoski. Elementary problem 5304. *Amer. Math. Monthly* **72** (1965), 674. Solution in **73** (1966), 681–682.

[60] E. E. Kummer. Über die Ergänzungssätze zu den allgemeinen Reciprocitätsgesetzen. *J. Reine Angew. Math.* **44** (1852), 93–146.

[61] A.-M. Legendre. *Théorie des Nombres*. Firmin Didot Frères, Paris, 1830.

[62] D. H. Lehmer. The Tarry-Escott problem. *Scripta Math.* **13** (1947), 37–41.

[63] S. Lehr. Sums and rational multiples of q–automatic sequences are q–automatic. *Theoret. Comput. Sci.* **108** (1993), 385–391.

[64] S. Lehr, J. Shallit, and J. Tromp. On the vector space of the automatic reals. *Theoret. Comput. Sci.* **163** (1996), 193–210.

[65] C. G. Lekkerkerker. Voorstelling van natuurlijke getallen door een som van getallen van Fibonacci. *Simon Stevin* **29** (1952), 190–195.

[66] A. Lepistö. Repetitions in Kolakoski sequence. In G. Rozenberg and A. Salomaa, editors, *Developments in Language Theory*, pages 130–143. World Scientific, 1994.

[67] N. Loraud. β-shift, systèmes de numération et automates. *J. Théorie Nombres Bordeaux* **7** (1995), 473–498.

[68] J. H. Loxton and A. J. van der Poorten. Algebraic independence properties of

the Fredholm series. *J. Austral. Math. Soc. A* **26** (1978), 31–45.

[69] J. H. Loxton and A. J. van der Poorten. Arithmetic properties of the solutions of a class of functional equations. *J. Reine Angew. Math.* **330** (1982), 159–172.

[70] J. H. Loxton and A. J. van der Poorten. Arithmetic properties of automata: regular sequences. *J. Reine Angew. Math.* **392** (1988), 57–69.

[71] K. Mahler. Arithmetische Eigenschaften der Lösungen einer Klasse von Funktionalgleichungen. *Math. Annalen* **101** (1929), 342–366. Corrigendum, *Math. Annalen* **103** (1930), 532.

[72] K. Mahler. Zur Approximation der Exponentialfunktion und des Logarithmus. I. *J. Reine Angew. Math.* **166** (1931/32), 118–136.

[73] M. Mendès France and J.-Y. Yao. Transcendence and the Carlitz-Goss gamma function. *J. Number Theory* **63** (1997), 396–402.

[74] M. Minsky and S. Papert. Unrecognizable sets of numbers. *J. Assoc. Comput. Mach.* **13** (1966), 281–286.

[75] M. Morse. Recurrent geodesics on a surface of negative curvature. *Trans. Amer. Math. Soc.* **22** (1921), 84–100.

[76] K. Nishioka. *Mahler Functions and Transcendence.* Lecture Notes in Mathematics, Vol. 1631, Springer-Verlag, 1996.

[77] J. Paradís, L. Bibiloni, and P. Viader. On actually computable bijections between $\mathbb{N}$ and $\mathbb{Q}^+$. *Order* **13** (1996), 369–377.

[78] G. Patruno. Solution to problem proposal 474. *Crux Math.* **6** (1980), 198.

[79] G. Păun. How much Thue is Kolakovski? [sic]. *Bull. European Assoc. Theor. Comput. Sci.* (49) (February 1993), 183–185.

[80] C. Pomerance, J. M. Robson, and J. Shallit. Automaticity II: Descriptional complexity in the unary case. *Theoret. Comput. Sci.* **180** (1997), 181–201.

[81] J. Propp. Problem proposal 474. *Crux Math.* **5** (1979), 229.

[82] E. Prouhet. Mémoire sur quelques relations entre les puissances des nombres. *C. R. Acad. Sci. Paris* **33** (1851), 225.

[83] G. N. Raney. On continued fractions and finite automata. *Math. Annalen* **206** (1973), 265–283.

[84] F. Recher. Propriétés de transcendance de séries formelles provenant de l'exponentielle de Carlitz. *C. R. Acad. Sci. Paris* **315** (1992), 245–250.

[85] D. Robbins. Solution to problem E 2692. *Amer. Math. Monthly* **86** (1979), 394–395.

[86] N. Robbins. Fibonacci partitions. *Fibonacci Quart.* **34** (1996), 306–313.

[87] A. Salomaa. *Formal Languages.* Academic Press, 1973.

[88] J. O. Shallit. Simple continued fractions for some irrational numbers. *J. Number Theory* **11** (1979), 209–217.

[89] J. O. Shallit. On infinite products associated with sums of digits. *J. Number Theory* **21** (1985), 128–134.

[90] J. O. Shallit. Some facts about continued fractions that should be better known. Technical Report CS-91-30, Department of Computer Science, University of Waterloo, July 1991.

[91] J. O. Shallit. Numeration systems, linear recurrences, and regular sets. *Inform. Comput.* **113** (1994), 331–347.

[92] J. O. Shallit. Automaticity IV: Sets, sequences, and diversity. *J. Théorie Nombres Bordeaux* **8** (1996), 347–367.

[93] J. Shallit and Y. Breitbart. Automaticity: properties of a measure of descriptional complexity. In P. Enjalbert et al., editor, *STACS '94: 11th Annual Symposium on Theoretical Aspects of Computer Science*, Vol. 775 of *Lecture Notes in Computer Science*, pages 619–630. Springer-Verlag, 1994.

[94] J. Shallit and Y. Breitbart. Automaticity I: Properties of a measure of descriptional complexity. *J. Comput. System Sci.* **53** (1996), 10–25.

[95] N. J. A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences.* Academic Press, 1995.

[96] R. Steacy. Structure in the Kolakoski sequence. *Bull. European Assoc. Theor.*

Comput. Sci. (59) (1996), 173–182.

[97] N. Strauss and J. Shallit. Advanced problem 6625. *Amer. Math. Monthly* **97** (1990), 252.

[98] D. S. Thakur. Automata-style proof of Voloch's result on transcendence. *J. Number Theory* **58** (1996), 60–63.

[99] A. Thue. Über unendliche Zeichenreihen. *Norske vid. Selsk. Skr. I. Mat. Nat. Kl. Christiana* **7** (1906), 1–22. Reprinted in *Selected Mathematical Papers of Axel Thue*, T. Nagell, editor, Universitetaforlaget, Oslo, 1977, pp. 139–158.

[100] B. A. Trakhtenbrot. On an estimate for the weight of a finite tree. *Sibirskii Matematicheskii Zhurnal* **5** (1964), 186–191. In Russian.

[101] K. Wagner and G. Wechsung. *Computational Complexity.* D. Reidel, 1986.

[102] W. D. Weakley. On the number of C^∞-words of each length. *J. Combin. Theory. Ser. A* **51** (1989), 55–62.

[103] D. R. Woods. Elementary problem proposal E 2692. *Amer. Math. Monthly* **85** (1978), 48.

[104] E. M. Wright. Prouhet's 1851 solution of the Tarry-Escott problem of 1910. *Amer. Math. Monthly* **66** (1959), 199–201.

[105] J. Yu. Transcendence and special zeta values in characteristic p. *Ann. Math.* **134** (1991), 1–23.

[106] D. Zagier. Solution to advanced problem 6625. *Amer. Math. Monthly* **99** (1992), 66–69.

[107] E. Zeckendorf. Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas. *Bull. Soc. Royale des Sciences de Liège* **41**(3–4) (1972), 179–182.

A bijection between nonnegative words and sparse *abba*-free partitions

Jan Němeček[a] and Martin Klazar[b,1,2]

[a] *Ve Stráni 87, 560 02 Česká Třebová, Czech Republic*
[b] *Department of Applied Mathematics and Institute for Theoretical Computer Science, Charles University, Malostranské náměstí 25, 118 00 Praha, Czech Republic*

Abstract

We give a bijective proof that the following two sets are equinumerous: (i) the set of words over $\{-1,0,1\}$ of length $m-2$ which have every initial sum nonnegative, and (ii) the set of partitions of $\{1,2,\ldots,m\}$ such that no two consecutive numbers lie in the same block and for no four numbers the middle two are in one block and the end two are in another block. The words were considered by Gouyou-Beauchamps and Viennot who enumerated by means of them certain animals. The identity connecting (i) and (ii) was observed by Klazar who proved it by generating functions.

Keywords: set partition; bijection; directed animal

Let us denote, for $m>0$, $[m]=\{1,2,\ldots,m\}$. A sequence $a=a_1a_2\ldots a_k$ is a *nonnegative word* if $a_i\in\{-1,0,1\}$ for each i and for each initial segment of a the sum of its elements is nonnegative. Recall that $\mathcal{A}=\{A_1,A_2,\ldots,A_n\}$ is a partition of $[m]$ if the A_is (called *blocks*) are nonempty disjoint subsets of $[m]$ and their union is $[m]$. We say that $\mathcal{A}$ is *sparse* if for every $i\in[m-1]$ the elements i and $i+1$ lie in two distinct blocks. $\mathcal{A}$ is called *abba-free* if it does not happen for any four elements $i<j<k<l$ of $[m]$ that i,l lie in a common block and j,k in another common block. For

[1]Supported by the project LN00A056 of the Ministry of Education of the Czech Republic.

[2]Corresponding author, klazar@kam.ms.mff.cuni.cz

example, $\{\{1,5,7\},\{2,4\},\{3,6\}\}$ is a sparse partition that is not *abba*-free. The partition $\{\{1,2,5,7\},\{4\},\{3\},\{6,8\}\}$ is *abba*-free but it is not sparse. We prove the following theorem.

Theorem. For every $m \geq 3$ there exists a bijection G between the set of sparse *abba*-free partitions of $[m]$ and the set of nonnegative words of length $m-2$.

Gouyou-Beauchamps and Viennot [1] were interested in counting certain animals (certain sets of plane lattice points) and showed that their animal problem is equivalent to enumeration of nonnegative words (they use slightly different terminology). Klazar [2] was interested in counting set partitions subject to structural restrictions and obtained as a byproduct the above identity. His derivation uses substantially generating functions. Speaking of them, if r_m is the number of sparse *abba*-free partitions of $[m]$, then ([2])

$$\sum_{m=0}^{\infty} r_m x^m = 1 + \frac{x}{2}\sqrt{\frac{1+x}{1-3x}}.$$

Analogous formula for nonnegative words was derived before in [1]. Our aim is to avoid the use of generating functions and to give a bijective proof of the identity.

We need few more definitions. A nonnegative word is a *correct word* if the first letter is 1, the last letter is -1, the sum of all letters is zero, and each proper initial segment has positive sum. We say that the letter a_j in a word over $\{-1,0,1\}$ is *dominant* if $a_j = 1$ and the sum of letters in every interval beginning in a_j is positive. For a a correct word of length at least three, a' is obtained from a by deleting the first and the last letter. Obviously, a' is a nonnegative word. For a partition $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ of $[m]$ we denote $|\mathcal{A}| = m$. Similarly, for a sequence a we denote $|a|$ its length. We say that $j \in [m]$ is *covered* in $\mathcal{A}$ if there exist $i,k \in [m]$ and $A_r \in \mathcal{A}$ so that $i < j < k$, $i,k \in A_r$, and $j \notin A_r$. If every element of $\{2,\ldots,m-1\}$ is covered in $\mathcal{A}$, we say that $\mathcal{A}$ is a *connected partition*. Any partition $\mathcal{A}$, $|\mathcal{A}| = m$, can be written in a *sequential form*. This is a sequence $b = b_1 b_2 \ldots b_m$ of length m over some alphabet such that $b_i = b_j$ if and only if i,j lie in the same block of $\mathcal{A}$. A partition has many sequential forms. One of them is the *canonical sequential form* in which the alphabet is $[n]$ (n is the number of blocks in $\mathcal{A}$) and the first occurrence of every $i \in [n]$, $i > 1$, in b is preceded by the

first occurrence of $i-1$. In particular, b starts with 1. Each partition has a unique canonical sequential form. It is convenient to write specific partitions in (canonical) sequential form. For example, the canonical sequential form of

$$\{\{1,5,7\},\{2,4\},\{3,6\}\} \text{ is } 1232131$$

(we will omit commas in the sequential forms of partitions).

Lemma 1. Each block of a connected sparse *abba*-free partition of $[m]$, $m \geq 3$, has at most two elements. Moreover, the block containing 1 and the block containing m have exactly two elements.

Proof. Suppose that $j < k < l$ belong to the same block, say B, of $\mathcal{A}$. Since k is covered, there exist s and t, $s < k < t$, belonging to the same block that is different from B. It is easy to check that each of the four possitions of s, j and t, l leads to the forbidden pattern *abba*. If $\{1\}$ were a block, 2 would not be covered. Similarly $\{m\}$ cannot be a block. □

We consider the following mapping F from the set of partitions of $[m]$ with no block with more than two elements to $\{-1, 0, 1\}$. $F(\mathcal{A}) = a_1 a_2 \ldots a_m$ where $a_i = 0$ if $\{i\} \in \mathcal{A}$, $a_i = 1$ if i is the first element of the two-element block containing i, and $a_i = -1$ if i is the second element. For example, $F(1234153) = 1, 0, 1, 0, -1, 0, -1$.

Lemma 2. For every $m \geq 3$, F is a bijection between the set of connected sparse *abba*-free partitions of $[m]$ and the set of correct words of lengt m.

Proof. By the previous lemma, if $\mathcal{A}$ is a connected sparse *abba*-free partition, $F(\mathcal{A})$ is defined and is a sequence beginning with 1 and ending with -1. Every initial sum of $F(\mathcal{A})$ is nonnegative for else we would have in the corresponding initial segment more second elements of two-element blocks than the first elements, which is impossible. Moreover, for no i, $1 < i < m$, the sum of the first i letters is zero because then i would not be covered. Thus $F(\mathcal{A})$ is a correct word.

We define the inverse mapping F'. Let $a = a_1 a_2 \ldots a_m$ be a correct word and let the partition $F'(a) = \mathcal{A}$ be defined in the following way. If $a_i = 0$ then $\{i\}$ is a (singleton) block of $\mathcal{A}$ and if a_i is the kth occurrence of 1 in a and a_j is the kth occurrence of -1, then $\{i, j\}$ is a block of $\mathcal{A}$. Note that always $i < j$ and that the second elements of two-element blocks come in

the same order as the first elements. Thus $\mathcal{A}$ is *abba*-free. $\mathcal{A}$ is connected because if an inner element i were not covered, then the sum of the first i letters of a would be zero. $\mathcal{A}$ is sparse because $\{i, i+1\} \in \mathcal{A}$ implies that $a_1 + a_2 + \cdots + a_{i-1} = 0$ and $a_1 + a_2 + \cdots + a_{i+1} = 0$. Finally, it is easy to check that F and F' are inverses of one another and thus F is a bijection. □

For a sparse *abba*-free partition $\mathcal{A}$ of $[m]$, $m \geq 3$, consider the collection $\mathcal{A}^*$ of maximal subintervals $I \subset [m]$ of length at least three for which the induced partition $\mathcal{A}|I$ is connected.

Lemma 3. Every two distinct intervals $I_1, I_2 \in \mathcal{A}^*$ are disjoint or they overlap in one element only.

Proof. Any other position of I_1 and I_2 means that every inner element of $I = I_1 \cup I_2$ is inner in I_1 or in I_2 and thus $\mathcal{A}|I$ is connected. This contradicts the maximality of I_1 or of I_2. □

Thus we can order $\mathcal{A}^*$ as $\mathcal{A}^* = \{I_1, I_2, \ldots, I_n\}_<$ where $I_i = [u_i, v_i]$ and $1 \leq u_1 < v_1 \leq u_2 < v_2 \leq u_3 < \ldots \leq u_n < v_n \leq m$. We define the numbers a_i, $0 \leq i \leq n$, by $a_i = u_{i+1} - v_i - 1$ where we set $v_0 = 0$ and $u_{n+1} = m + 1$. Clearly, $a_i \geq -1$ and a_i is the number of elements between I_i and I_{i+1}, where $a_i = -1$ means that the intervals overlap. Note that every element between I_i and I_{i+1} forms a singleton block.

Now we can define the desired bijection G:

$$G(\mathcal{A}) = 1^{a_0} F(\mathcal{A}_1)' 1^{a_1+2} F(\mathcal{A}_2)' 1^{a_2+2} \ldots 1^{a_{n-1}+2} F(\mathcal{A}_n)' 1^{a_n}.$$

Here $\mathcal{A}$ is a sparse *abba*-free partition of $[m]$, $m \geq 3$, 1^i abbreviates the sequence $1, 1, \ldots, 1$ of i 1s, a_i are the above numbers, $\mathcal{A}_i$ is the restriction of $\mathcal{A}$ to I_i (where $\mathcal{A}^* = \{I_1, I_2, \ldots, I_n\}_<$) normalized so that the ground set is an initial segment of positive integers (of lentgh $v_i - u_i + 1$), F is the mapping of Lemma 2, and $'$ means the deletion of the first and last letter. If $n = 0$, that is if $\mathcal{A}^* = \emptyset$ and $\mathcal{A}$ has only singleton blocks, we set

$$G(\mathcal{A}) = 1^{a_0 - 2} = 1^{m-2}.$$

We prove that G is indeed a bijection between all sparse *abba*-free partitions of $[m]$ and all nonnegative words of length $m-2$. By Lemma 2, $F(\mathcal{A}_i)$

is a correct word. Hence $F(\mathcal{A}_i)'$ is a nonnegative word and the whole $G(\mathcal{A})$ is a nonnegative word. Its length is $m-2$ if $\mathcal{A}^* = \emptyset$ and

$$\sum_{i=1}^{n}(a_{i-1} + |F(\mathcal{A}_i)| - 2) + a_n + 2(n-1) = \sum_{i=1}^{n}(a_{i-1} + |I_i|) + a_n - 2 = m - 2$$

if $\mathcal{A}^* \neq \emptyset$.

We define the inverse mapping G'. Let $b = b_1 b_2 \ldots b_{m-2}$, $m \geq 3$, be a nonnegative word. There is a unique decomposition of b into intervals

$$b = c_0 d_1 c_1 d_2 \ldots c_{n-1} d_n c_n$$

such that c_0 is the longest initial interval in which every element is dominant, d_1 is the longest interval starting immediately after c_0 whose elements sum up to zero, c_1 is the longest interval starting immediately after d_1 in which every element is dominant and so on. Note that c_0 and c_n may be empty but the other intervals are nonempty, $c_i = 1^{e_i}$ where e_i is a nonnegative integer, and every d_i is a nonnegative word. If $b = c_0$, b consists only of 1s, and we set $G'(b)$ to be the partition of $[m]$ having just the singleton blocks $\{1\}, \{2\}, \ldots, \{m\}$. If $n > 0$, we define $\mathcal{A}_i = F'(1, d_i, -1)$ where F' is the inverse mapping to F of Lemma 2, defined in its proof. Word $1, d_i, -1$ is a correct word and $\mathcal{A}_i$ is a connected sparse *abba*-free partition of some initial interval of positive integers. We define the numbers a_i as $a_0 = e_0$, $a_n = e_n$, and $a_i = e_i - 2$ for $0 < i < n$. Finally, we set

$$G'(b) = \mathcal{B}_0 \mathcal{A}_1 \mathcal{B}_1 \mathcal{A}_2 \ldots \mathcal{B}_{n-1} \mathcal{A}_n \mathcal{B}_n$$

where $\mathcal{B}_i$ is, for $a_i > 0$, a partition consisting of a_i singleton blocks. If $a_i = 0$, $\mathcal{B}_i = \emptyset$ and $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ are neighbours. If $a_i = -1$, $\mathcal{B}_i = \emptyset$ and $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ are made to overlap in the last element of $\mathcal{A}_i$ and the first element of $\mathcal{A}_{i+1}$. The two blocks which now intersect merge into one block. Needless to say, the ground intervals of $\mathcal{A}_i$ and $\mathcal{B}_i$ are in the concatenation shifted appropriately to make up an initial interval of positive integers.

It is easy to check that the resulting partition $G'(b)$ is a sparse *abba*-free partition of $[m]$ and that for every $\mathcal{A}$ and b we have $G'(G(\mathcal{A})) = \mathcal{A}$ and $G(G'(b)) = b$. Thus G and G' are bijections. The theorem is proved.

As an example, we list in the lexicographical order all 13 sparse *abba*-free partitions of $[5]$ in their canonical sequential form and the corresponding nonnegative words with length 3:

$$
\begin{aligned}
G(12123) &= F(1212)', 1 = (1,1,-1,-1)', 1 = 1,-1,1.\\
G(12131) &= F(121)', 1, F(131)' = (1,0,-1)', 1, (1,0,-1)' = 0,1,0.\\
G(12132) &= F(12132)' = (1,1,-1,0,-1)' = 1,-1,0.\\
G(12134) &= F(121)', 1^2 = (1,0,-1)', 1, 1 = 0,1,1.\\
G(12312) &= F(12312)' = (1,1,0,-1,-1)' = 1,0,-1.\\
G(12313) &= F(12313)' = (1,0,1,-1,-1)' = 0,1,-1.\\
G(12314) &= F(1231)', 1 = (1,0,0,-1)', 1 = 0,0,1.\\
G(12323) &= 1, F(2323)' = 1, (1,1,-1,-1)' = 1,1,-1.\\
G(12324) &= 1, F(232)', 1 = 1, (1,0,-1)', 1 = 1,0,1.\\
G(12341) &= F(12341)' = (1,0,0,0,-1)' = 0,0,0.\\
G(12342) &= 1, F(2342)' = 1, (1,0,0,-1)' = 1,0,0.\\
G(12343) &= 1^2, F(343)' = 1,1,(1,0,-1)' = 1,1,0.\\
G(12345) &= 1^3 = 1,1,1.
\end{aligned}
$$

References

[1] D. Gouyou-Beauchamps and G. Viennot, Equivalence of the two-dimensional directed animal problem to a one-dimensional path problem, Adv. Appl. Math. 9 (1988) 334–357.

[2] M. Klazar, On *abab*-free and *abba*-free set partitions, Europ. J. Comb. 17 (1996) 53–68.

THE NUMBER OF INTERSECTION POINTS MADE BY THE DIAGONALS OF A REGULAR POLYGON

BJORN POONEN AND MICHAEL RUBINSTEIN

ABSTRACT. We give a formula for the number of interior intersection points made by the diagonals of a regular n-gon. The answer is a polynomial on each residue class modulo 2520. We also compute the number of regions formed by the diagonals, by using Euler's formula $V - E + F = 2$.

1. INTRODUCTION

We will find a formula for the number $I(n)$ of intersection points formed inside a regular n-gon by its diagonals. The case $n = 30$ is depicted in Figure 1. For a *generic* convex n-gon, the answer would be $\binom{n}{4}$, because every four vertices would be the endpoints of a unique pair of intersecting diagonals. But $I(n)$ can be less, because in a regular n-gon it may happen that three or more diagonals meet at an interior point, and then some of the $\binom{n}{4}$ intersection points will coincide. In fact, if n is even and at least 6, $I(n)$ will always be less than $\binom{n}{4}$, because there will be $n/2 \geq 3$ diagonals meeting at the center point. It will result from our analysis that for $n > 4$, the maximum number of diagonals of the regular n-gon that meet at a point other than the center is

2 if n is odd,
3 if n is even but not divisible by 6,
5 if n is divisible by 6 but not 30, and,
7 if n is divisible by 30.

with two exceptions: this number is 2 if $n = 6$, and 4 if $n = 12$. In particular, it is impossible to have 8 or more diagonals of a regular n-gon meeting at a point other than the center. Also, by our earlier remarks, the fact that no three diagonals meet when n is odd will imply that $I(n) = \binom{n}{4}$ for odd n.

A careful analysis of the possible configurations of three diagonals meeting will provide enough information to permit us in theory to deduce a formula for $I(n)$. But because the explicit description of these configurations is so complex, our strategy will be instead to use this information to deduce only the *form* of the answer, and then to compute the answer for enough small n that we can determine the result precisely. The computations are done in Mathematica, Maple and C, and annotated source codes can be obtained via anonymous ftp at `http://math.berkeley.edu/~poonen`.

Date: November 18, 1997.

1991 *Mathematics Subject Classification.* Primary 51M04; Secondary 11R18.

Key words and phrases. regular polygons, diagonals, intersection points, roots of unity, adventitious quadrangles.

The first author is supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship. Part of this work was done at MSRI, where research is supported in part by NSF grant DMS-9022140.

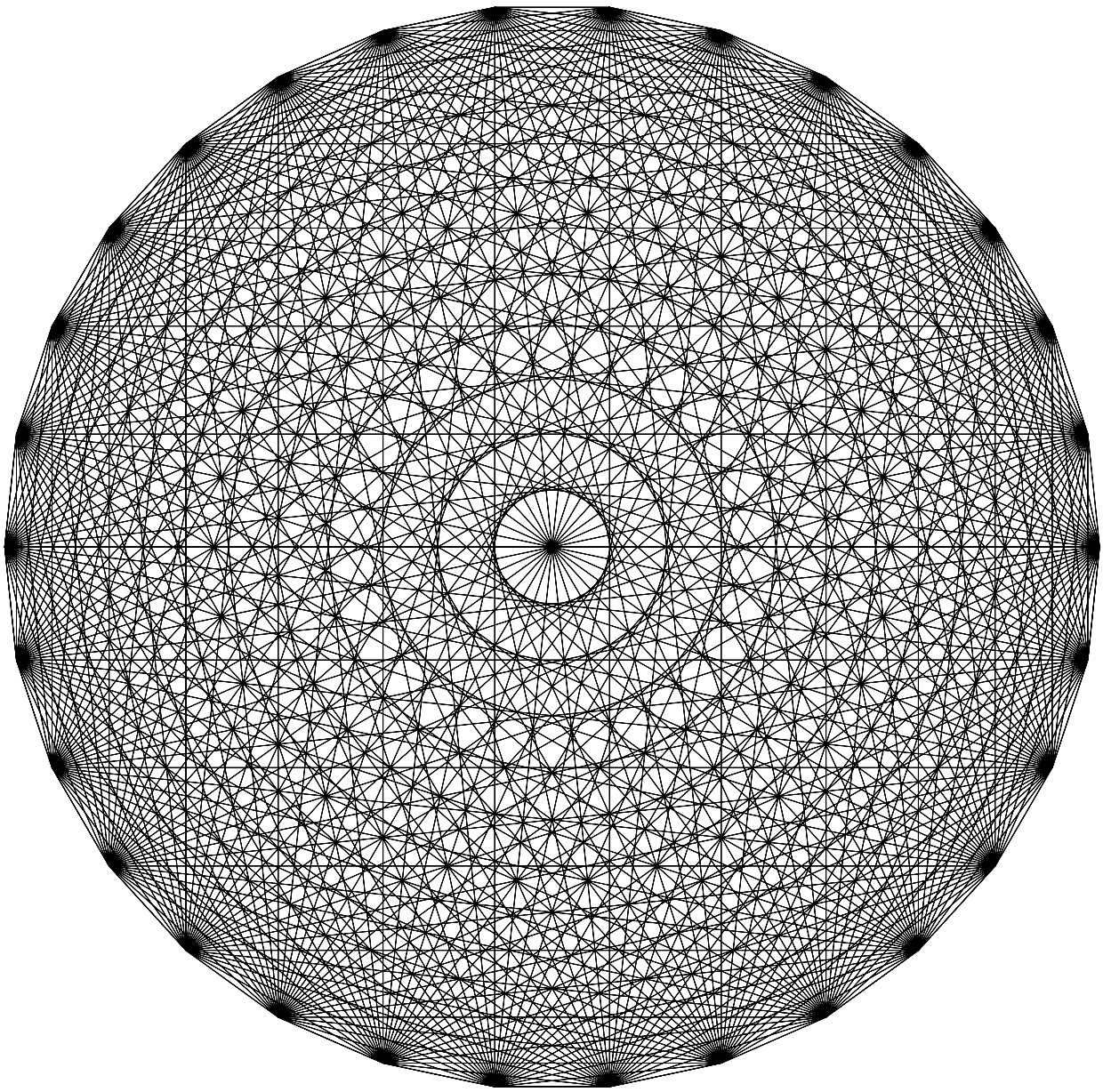


FIGURE 1. The 30-gon with its diagonals. There are 16801 interior intersection points: 13800 two line intersections, 2250 three line intersections, 420 four line intersections, 180 five line intersections, 120 six line intersections, 30 seven line intersections, and 1 fifteen line intersection.

In order to write the answer in a reasonable form, we define

$$\delta_m(n) = \begin{cases} 1 & \text{if } n \equiv 0 \pmod{m}, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 1. *For $n \geq 3$,*

$$\begin{aligned} I(n) &= \binom{n}{4} + (-5n^3 + 45n^2 - 70n + 24)/24 \cdot \delta_2(n) - (3n/2) \cdot \delta_4(n) \\ &\quad + (-45n^2 + 262n)/6 \cdot \delta_6(n) + 42n \cdot \delta_{12}(n) + 60n \cdot \delta_{18}(n) \\ &\quad + 35n \cdot \delta_{24}(n) - 38n \cdot \delta_{30}(n) - 82n \cdot \delta_{42}(n) - 330n \cdot \delta_{60}(n) \\ &\quad - 144n \cdot \delta_{84}(n) - 96n \cdot \delta_{90}(n) - 144n \cdot \delta_{120}(n) - 96n \cdot \delta_{210}(n). \end{aligned}$$

Further analysis, involving Euler's formula $V - E + F = 2$, will yield a formula for the number $R(n)$ of regions that the diagonals cut the n-gon into.

Theorem 2. *For $n \geq 3$,*

$$\begin{aligned} R(n) &= (n^4 - 6n^3 + 23n^2 - 42n + 24)/24 \\ &\quad + (-5n^3 + 42n^2 - 40n - 48)/48 \cdot \delta_2(n) - (3n/4) \cdot \delta_4(n) \\ &\quad + (-53n^2 + 310n)/12 \cdot \delta_6(n) + (49n/2) \cdot \delta_{12}(n) + 32n \cdot \delta_{18}(n) \\ &\quad + 19n \cdot \delta_{24}(n) - 36n \cdot \delta_{30}(n) - 50n \cdot \delta_{42}(n) - 190n \cdot \delta_{60}(n) \\ &\quad - 78n \cdot \delta_{84}(n) - 48n \cdot \delta_{90}(n) - 78n \cdot \delta_{120}(n) - 48n \cdot \delta_{210}(n). \end{aligned}$$

These problems have been studied by many authors before, but this is apparently the first time the correct formulas have been obtained. The Dutch mathematician Gerrit Bol [1] gave a complete solution in 1936, except that a few of the coefficients in his formulas are wrong. (A few misprints and omissions in Bol's paper are mentioned in [11].)

The approaches used by us and Bol are similar in many ways. One difference (which is not too substantial) is that we work as much as possible with roots of unity whereas Bol tended to use more trigonometry (integer relations between sines of rational multiples of π). Also, we relegate much of the work to the computer, whereas Bol had to enumerate the many cases by hand. The task is so formidable that it is amazing to us that Bol was able to complete it, and at the same time not so surprising that it would contain a few errors!

Bol's work was largely forgotten. In fact, even we were not aware of his paper until after deriving the formulas ourselves. Many other authors in the interim solved special cases of the problem. Steinhaus [14] posed the problem of showing that no three diagonals meet internally when n is prime, and this was solved by Croft and Fowler [3]. (Steinhaus also mentions this in [13], which includes a picture of the 23-gon and its diagonals.) In the 1960s, Heineken [6] gave a delightful argument which generalized this to all odd n, and later he [7] and Harborth [4] independently enumerated all three-diagonal intersections for n not divisible by 6.

The classification of three-diagonal intersections also solves Colin Tripp's problem [15] of enumerating "adventitious quadrilaterals," those convex quadrilaterals for which the angles formed by sides and diagonals are all rational multiples of π. See Rigby's paper [11] or the summary [10] for details. Rigby, who was aware of Bol's work, mentions that Monsky and Pleasants also each independently classified all three-diagonal intersections of regular n-gons. Rigby's papers partially solve

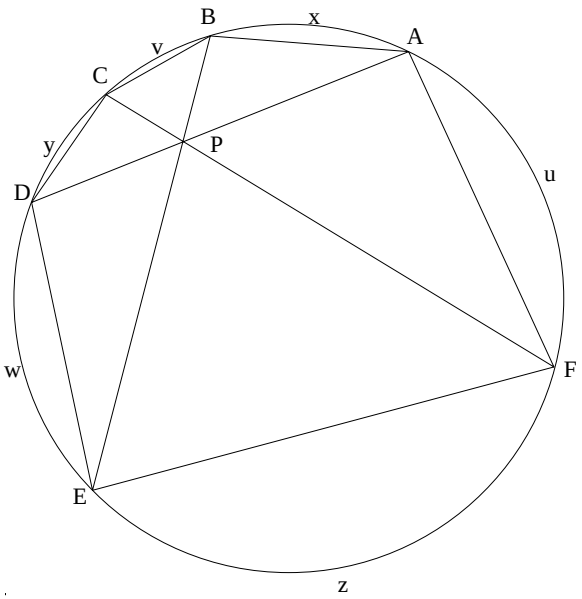


FIGURE 2.

Tripp's further problem of proving the existence of all adventitious quadrangles using only elementary geometry; i.e., without resorting to trigonometry.

All the questions so far have been in the Euclidean plane. What happens if we count the interior intersections made by the diagonals of a hyperbolic regular n-gon? The answers are exactly the same, as pointed out in [11], because if we use Beltrami's representation of points of the hyperbolic plane by points inside a circle in the Euclidean plane, we can assume that the center of the hyperbolic n-gon corresponds to the center of the circle, and then the hyperbolic n-gon with its diagonals looks in the model exactly like a Euclidean regular n-gon with its diagonals. It is equally easy to see that the answers will be the same in elliptic geometry.

2. When do three diagonals meet?

We now begin our derivations of the formulas for $I(n)$ and $R(n)$. The first step will be to find a criterion for the concurrency of three diagonals. Let A, B, C, D, E, F be six distinct points in order on a unit circle dividing up the circumference into arc lengths u, x, v, y, w, z and assume that the three chords AD, BE, CF meet at P (see Figure 2).

By similar triangles, $AF/CD = PF/PD$, $BC/EF = PB/PF$, $DE/AB = PD/PB$. Multiplying these together yields

$$(AF \cdot BC \cdot DE)/(CD \cdot EF \cdot AB) = 1,$$

and so

$$\sin(u/2)\sin(v/2)\sin(w/2) = \sin(x/2)\sin(y/2)\sin(z/2). \tag{1}$$

Conversely, suppose six distinct points A, B, C, D, E, F partition the circumference of a unit circle into arc lengths u, x, v, y, w, z and suppose that (1) holds. Then the three diagonals AD, BE, CF meet in a single point which we see as follows. Let lines AD and BE intersect at P_0 . Form the line through F and P_0 and let C' be the other intersection point of FP_0 with the circle. This partitions the circumference into arc lengths u, x, v', y', w, z. As shown above, we have

$$\sin(u/2)\sin(v'/2)\sin(w/2) = \sin(x/2)\sin(y'/2)\sin(z/2)$$

and since we are assuming that (1) holds for u, x, v, y, w, z we get

$$\frac{\sin(v'/2)}{\sin(y'/2)} = \frac{\sin(v/2)}{\sin(y/2)}.$$

Let $\alpha = v + y = v' + y'$. Substituting $v = \alpha - y$, $v' = \alpha - y'$ above we get

$$\frac{\sin(\alpha/2)\cos(y'/2) - \cos(\alpha/2)\sin(y'/2)}{\sin(y'/2)} = \frac{\sin(\alpha/2)\cos(y/2) - \cos(\alpha/2)\sin(y/2)}{\sin(y/2)}$$

and so

$$\cot(y'/2) = \cot(y/2).$$

Now $0 < \alpha/2 < \pi$, so $y = y'$ and hence $C = C'$. Thus, the three diagonals AD, BE, CF meet at a single point.

So (1) gives a necessary and sufficient condition (in terms of arc lengths) for the chords AD, BE, CF formed by six distinct points A, B, C, D, E, F on a unit circle to meet at a single point. In other words, to give an explicit answer to the question in the section title, we need to characterize the positive rational solutions to

$$\begin{aligned} \sin(\pi U)\sin(\pi V)\sin(\pi W) &= \sin(\pi X)\sin(\pi Y)\sin(\pi Z) \qquad (2)\\ U + V + W + X + Y + Z &= 1. \end{aligned}$$

(Here $U = u/(2\pi)$, etc.) This is a trigonometric diophantine equation in the sense of [2], where it is shown that in theory, there is a finite computation which reduces the solution of such equations to ordinary diophantine equations. The solutions to the analogous equation with only two sines on each side are listed in [9].

If in (2), we substitute $\sin(\theta) = (e^{i\theta} - e^{-i\theta})/(2i)$, multiply both sides by $(2i)^3$, and expand, we get a sum of eight terms on the left equalling a similar sum on the right, but two terms on the left cancel with two terms on the right since $U+V+W = 1-(X+Y+Z)$, leaving

$$-e^{i\pi(V+W-U)} + e^{-i\pi(V+W-U)} - e^{i\pi(W+U-V)} + e^{-i\pi(W+U-V)} - e^{i\pi(U+V-W)} + e^{-i\pi(U+V-W)} =$$
$$-e^{i\pi(Y+Z-X)} + e^{-i\pi(Y+Z-X)} - e^{i\pi(Z+X-Y)} + e^{-i\pi(Z+X-Y)} - e^{i\pi(X+Y-Z)} + e^{-i\pi(X+Y-Z)}.$$

If we move all terms to the left hand side, convert minus signs into $e^{-i\pi}$, multiply by $i = e^{i\pi/2}$, and let

$$\begin{aligned} \alpha_1 &= V + W - U - 1/2\\ \alpha_2 &= W + U - V - 1/2\\ \alpha_3 &= U + V - W - 1/2\\ \alpha_4 &= Y + Z - X + 1/2\\ \alpha_5 &= Z + X - Y + 1/2\\ \alpha_6 &= X + Y - Z + 1/2, \end{aligned}$$

we obtain

$$\sum_{j=1}^{6} e^{i\pi\alpha_j} + \sum_{j=1}^{6} e^{-i\pi\alpha_j} = 0, \tag{3}$$

in which $\sum_{j=1}^{6} \alpha_j = U+V+W+X+Y+Z = 1$. Conversely, given rational numbers $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ (not necessarily positive) which sum to 1 and satisfy (3), we can recover U, V, W, X, Y, Z, (for example, $U = (\alpha_2 + \alpha_3)/2 + 1/2$), but we must check that they turn out positive.

3. Zero as a sum of 12 roots of unity

In order to enumerate the solutions to (2), we are led, as in the end of the last section, to classify the ways in which 12 roots of unity can sum to zero. More generally, we will study relations of the form

$$\sum_{i=1}^{k} a_i \eta_i = 0, \tag{4}$$

where the a_i are positive integers, and the η_i are distinct roots of unity. (These have been studied previously by Schoenberg [12], Mann [8], Conway and Jones [2], and others.) We call $w(S) = \sum_{i=1}^{k} a_i$ the *weight* of the relation S. (So we shall be particularly interested in relations of weight 12.) We shall say the relation (4) is *minimal* if it has no nontrivial subrelation; i.e., if

$$\sum_{i=1}^{k} b_i \eta_i = 0, \quad a_i \geq b_i \geq 0$$

implies either $b_i = a_i$ for all i or $b_i = 0$ for all i. By induction on the weight, any relation can be represented as a sum of minimal relations (but the representation need not be unique).

Let us give some examples of minimal relations. For each $n \geq 1$, let $\zeta_n = \exp(2\pi i/n)$ be the standard primitive n-th root of unity. For each prime p, let R_p be the relation

$$1 + \zeta_p + \zeta_p^2 + \cdots + \zeta_p^{p-1} = 0.$$

Its minimality follows from the irreducibility of the cyclotomic polynomial. Also we can "rotate" any relation by multiplying through by an arbitrary root of unity to obtain a new relation. In fact, Schoenberg [12] proved that every relation (even those with possibly negative coefficients) can be obtained as a linear combination with positive and negative integral coefficients of the R_p and their rotations. But we are only allowing positive combinations, so it is not clear that these are enough to generate all relations.

In fact it is not even true! In other words, there are other minimal relations. If we subtract R_3 from R_5, cancel the 1's and incorporate the minus signs into the roots of unity, we obtain a new relation

$$\zeta_6 + \zeta_6^{-1} + \zeta_5 + \zeta_5^2 + \zeta_5^3 + \zeta_5^4 = 0, \tag{5}$$

which we will denote $(R_5 : R_3)$. In general, if S and $T_1, T_2, \ldots, T_j$ are relations, we will use the notation $(S : T_1, T_2, \ldots, T_j)$ to denote any relation obtained by rotating the T_i so that each shares exactly one root of unity with S which is different for each i, subtracting them from S, and incorporating the minus signs into the

Weight	Relation type	Number of relations of that type
2	R_2	1
3	R_3	1
5	R_5	1
6	$(R_5 : R_3)$	1
7	$(R_5 : 2R_3)$	2
	R_7	1
8	$(R_5 : 3R_3)$	2
	$(R_7 : R_3)$	1
9	$(R_5 : 4R_3)$	1
	$(R_7 : 2R_3)$	3
10	$(R_7 : 3R_3)$	5
	$(R_7 : R_5)$	1
11	$(R_7 : 4R_3)$	5
	$(R_7 : R_5, R_3)$	6
	$(R_7 : (R_5 : R_3))$	6
	R_{11}	1
12	$(R_7 : 5R_3)$	3
	$(R_7 : R_5, 2R_3)$	15
	$(R_7 : (R_5 : R_3), R_3)$	36
	$(R_7 : (R_5 : 2R_3))$	14
	$(R_{11} : R_3)$	1

TABLE 1. The 107 minimal relations of weight up to 12.

roots of unity. For notational convenience, we will write $(R_5 : 4R_3)$ for $(R_5 : R_3, R_3, R_3, R_3)$, for example. Note that although $(R_5 : R_3)$ denotes unambiguously (up to rotation) the relation listed in (5), in general there will be many relations of type $(S : T_1, T_2, \ldots, T_j)$ up to rotational equivalence. Let us also remark that including R_2's in the list of T's has no effect.

It turns out that recursive use of the construction above is enough to generate all minimal relations of weight up to 12. These are listed in Table 1. The completeness and correctness of the table will be proved in Theorem 3 below. Although there are 107 minimal relations up to rotational equivalence, often the minimal relations within one of our classes are Galois conjugates. For example, the two minimal relations of type $(R_5 : 2R_3)$ are conjugate under $\mathrm{Gal}(\mathbb{Q}(\zeta_{15})/\mathbb{Q})$, as pointed out in [8].

The minimal relations with $k \leq 7$ (k defined as in (4)) had been previously catalogued in [8], and those with $k \leq 9$ in [2]. In fact, the a_i in these never exceed 1, so these also have weight less than or equal to 9.

Theorem 3. *Table 1 is a complete listing of the minimal relations of weight up to 12 (up to rotation).*

The following three lemmas will be needed in the proof.

Lemma 1. *If the relation (4) is minimal, then there are distinct primes $p_1 < p_2 < \cdots < p_s \leq k$ so that each η_i is a $p_1 p_2 \cdots p_s$-th root of unity, after the relation has been suitably rotated.*

Proof. This is a corollary of Theorem 1 in [8]. □

Lemma 2. *The only minimal relations (up to rotation) involving only the $2p$-th roots of unity, for p prime, are R_2 and R_p.*

Proof. Any $2p$-th root of unity is of the form $\pm\zeta^i$. If both $+\zeta^i$ and $-\zeta^i$ occurred in the same relation, then R_2 occurs as a subrelation. So the relation has the form

$$\sum_{i=0}^{p-1} c_i \zeta_p^i = 0$$

By the irreducibility of the cyclotomic polynomial, $\{1, \zeta_p, \ldots, \zeta_p^{p-1}\}$ are independent over $\mathbb{Q}$ save for the relation that their sum is zero, so all the c_i must be equal. If they are all positive, then R_p occurs as a subrelation. If they are all negative, then R_p rotated by -1 (i.e., 180 degrees) occurs as a subrelation. □

Lemma 3. *Suppose S is a minimal relation, and $p_1 < p_2 < \cdots < p_s$ are picked as in Lemma 1 with $p_1 = 2$ and p_s minimal. If $w(S) < 2p_s$, then S (or a rotation) is of the form $(R_{p_s} : T_1, T_2, \ldots, T_j)$ where the T_i are minimal relations not equal to R_2 and involving only $p_1 p_2 \cdots p_{s-1}$-th roots of unity, such that $j < p_s$ and*

$$\sum_{i=1}^{j} [w(T_i) - 2] = w(S) - p_s .$$

Proof. Since every $p_1 p_2 \ldots p_s$-th root of unity is uniquely expressible as the product of a $p_1 p_2 \ldots p_{s-1}$-th root of unity and a p_s-th root of unity, the relation can be rewritten as

$$\sum_{i=0}^{p_s-1} f_i \zeta_{p_s}^i = 0, \tag{6}$$

where each f_i is a sum of $p_1 p_2 \ldots p_{s-1}$-th roots of unity, which we will think of as a sum (not just its value).

Let K_m be the field obtained by adjoining the $p_1 p_2 \ldots p_m$-th roots of unity to $\mathbb{Q}$. Since $[K_s : K_{s-1}] = \phi(p_1 p_2 \cdots p_s)/\phi(p_1 p_2 \cdots p_{s-1}) = \phi(p_s) = p_s - 1$, the only linear relation satisfied by $1, \zeta_{p_s}, \ldots, \zeta_{p_s}^{p_s-1}$ over K_{s-1} is that their sum is zero. Hence (6) forces the values of the f_i to be equal.

The total number of roots of unity in all the f_i's is $w(S) < 2p_s$, so by the pigeonhole principle, some f_i is zero or consists of a single root of unity. In the former case, each f_j sums to zero, but at least two of these sums contain at least one root of unity, since otherwise s was not minimal, so one of these sums gives a subrelation of S, contradicting its minimality. So some f_i consists of a single root of unity. By rotation, we may assume $f_0 = 1$. Then each f_i sums to 1, and if it is not simply the single root of unity 1, the negatives of the roots of unity in f_i together with 1 form a relation T_i which is not R_2 and involves only $p_1 p_2 \cdots p_{s-1}$-th roots of unity, and it is clear that S is of type $(R_{p_s} : T_{i_1}, T_{i_2}, \ldots, T_{i_j})$. If one of the T's were not minimal, then it could be decomposed into two nontrivial subrelations, one of which would not share a root of unity with the R_{p_s}, and this would give a nontrivial subrelation of S, contradicting the minimality of S. Finally, $w(S)$ must equal the sum of the weights of R_{p_s} and the T's, minus $2j$ to account for the roots of unity that are cancelled in the construction of $(R_{p_s} : T_{i_1}, T_{i_2}, \ldots, T_{i_j})$. □

Proof of Theorem 3. We will content ourselves with proving that every relation of weight up to 12 can be decomposed into a sum of the ones listed in Table 1, it then being straightforward to check that the entries in the table are distinct, and that none of them can be further decomposed into relations higher up in the table.

Let S be a minimal relation with $w(S) \leq 12$. Pick $p_1 < p_2 < \cdots < p_s$ as in Lemma 1 with $p_1 = 2$ and p_s minimal. In particular, $p_s \leq 12$, so $p_s = 2,3,5,7$, or 11.

Case 1: $p_s \leq 3$

Here the only minimal relations are R_2 and R_3, by Lemma 2.

Case 2: $p_s = 5$

If $w(S) < 10$, then we may apply Lemma 3 to deduce that S is of type $(R_5 : T_1, T_2, \ldots, T_j)$ Each T must be R_3 (since $p_{s-1} \leq 3$), and $j = w(S) - 5$ by the last equation in Lemma 3. The number of relations of type $(R_5 : jR_3)$, up to rotation, is $\binom{5}{j}/5$. (There are $\binom{5}{j}$ ways to place the R_3's, but one must divide by 5 to avoid counting rotations of the same relation.)

If $10 \leq w(S) \leq 12$, then write S as in (6). If some f_i consists of zero or one roots of unity, then the argument of Lemma 3 applies, and S must be of the form $(R_5 : jR_3)$ with $j \leq 4$, which contradicts the last equation in the Lemma. Otherwise the numbers of (sixth) roots of unity occurring in f_0, f_1, f_2, f_3, f_4 must be 2,2,2,2,2 or 2,2,2,2,3 or 2,2,2,3,3 or 2,2,2,2,4 in some order. So the common value of the f_i is a sum of two sixth roots of unity. By rotating by a sixth root of unity, we may assume this value is 0, 1, $1 + \zeta_6$, or 2. If it is 0 or 1, then the arguments in the proof of Lemma 3 apply. Next assume it is $1 + \zeta_6$. The only way two sixth roots of unity can sum to $1 + \zeta_6$ is if they are 1 and ζ_6 in some order. The only ways three sixth roots of unity can sum to $1 + \zeta_6$ is if they are $1, 1, \zeta_6^2$ or $\zeta_6, \zeta_6, \zeta_6^{-1}$. So if the numbers of roots of unity occurring in f_0, f_1, f_2, f_3, f_4 are 2,2,2,2,2 or 2,2,2,2,3, then S will contain R_5 or its rotation by ζ_6, and the same will be true for 2,2,2,3,3 unless the two f_i with three terms are $1 + 1 + \zeta_6^2$ and $\zeta_6 + \zeta_6 + \zeta_6^{-1}$, in which case S contains $(R_5 : R_3)$. It is impossible to write $1 + \zeta_6$ as a sum of sixth roots of unity without using 1 or ζ_6, so if the numbers are 2,2,2,2,4, then again S contains R_5 or its rotation by ζ_6. Thus we get no new relations where the common value of the f_i is $1 + \zeta_6$. Lastly, assume this common value is 2. Any representation of 2 as a sum of four or fewer sixth roots of unity contains 1, unless it is $\zeta_6 + \zeta_6 + \zeta_6^{-1} + \zeta_6^{-1}$, so S will contain R_5 except possibly in the case where f_0, f_1, f_2, f_3, f_4 are 2,2,2,2,4 in some order, and the 4 is as above. But in this final remaining case, S contains $(R_5 : R_3)$. Thus there are no minimal relations S with $p_s = 5$ and $10 \leq w(S) \leq 12$.

Case 3: $p_s = 7$

Since $w(S) \leq 12 < 2 \cdot 7$, we can apply Lemma 3. Now the sum of $w(T_i) - 2$ is required to be $w(S) - 7$ which is at most 5, so the T's that may be used are R_3, R_5, $(R_5 : R_3)$, and the two of type $(R_5 : 2R_3)$, for which weight minus 2 equals 1, 3, 4, and 5, respectively. So the problem is reduced to listing the partitions of $w(S) - 7$ into parts of size 1, 3, 4, and 5.

If all parts used are 1, then we get $(R_7 : jR_3)$ with $j = w(S) - 7$, and there are $\binom{7}{j}/7$ distinct relations in this class. Otherwise exactly one part of size 3, 4, or 5 is used, and the possibilities are as follows. If a part of size 3 is used, we get $(R_7 : R_5)$, $(R_7 : R_5, R_3)$, or $(R_7 : R_5, 2R_3)$, of weights 10, 11, 12 respectively. By rotation, the R_5 may be assumed to share the 1 in the R_7, and then there are $\binom{6}{i}$

Partition	Relation type
12	$(R_7 : 5R_3)$
	$(R_7 : R_5, 2R_3)$
	$(R_7 : (R_5 : R_3), R_3)$
	$(R_7 : (R_5 : 2R_3))$
	$(R_{11} : R_3)$
10,2	$(R_7 : 3R_3) + R_2$
	$(R_7 : R_5) + R_2$
9,3	$(R_5 : 4R_3) + R_3$
	$(R_7 : 2R_3) + R_3$
8,2,2	$(R_5 : 3R_3) + 2R_2$
	$(R_7 : R_3) + 2R_2$
7,5	$(R_5 : 2R_3) + R_5$
	$R_7 + R_5$
7,3,2	$(R_5 : 2R_3) + R_3 + R_2$
	$R_7 + R_3 + R_2$
6,6	$2(R_5 : R_3)$
6,3,3	$(R_5 : R_3) + 2R_3$
6,2,2,2	$(R_5 : R_3) + 3R_2$
5,5,2	$2R_5 + R_2$
5,3,2,2	$R_5 + R_3 + 2R_2$
3,3,3,3	$4R_3$
3,3,2,2,2	$2R_3 + 3R_2$
2,2,2,2,2,2	$6R_2$

TABLE 2. The types of relations of weight 12.

ways to place the R_3's where i is the number of R_3's. If a part of size 4 is used, we get $(R_7 : (R_5 : R_3))$ of weight 11 or $(R_7 : (R_5 : R_3), R_3)$ of weight 12. By rotation, the $(R_5 : R_3)$ may be assumed to share the 1 in the R_7, but any of the six roots of unity in the $(R_5 : R_3)$ may be rotated to be 1. The R_3 can then overlap any of the other 6 seventh roots of unity. Finally, if a part of size 5 is used, we get $(R_7 : (R_5 : 2R_3))$. There are two different relations of type $(R_5 : 2R_3)$ that may be used, and each has seven roots of unity which may be rotated to be the 1 shared by the R_7, so there are 14 of these all together.

Case 4: $p_s = 11$

Applying Lemma 3 shows that the only possibilities are R_{11} of weight 11, and $(R_{11} : R_3)$ of weight 12. □

Now a general relation of weight 12 is a sum of the minimal ones of weight up to 12, and we can classify them according to the weights of the minimal relations, which form a partition of 12 with no parts of size 1 or 4. We will use the notation $(R_5 : 2R_3) + 2R_3$, for example, to denote a sum of three minimal relations of type $(R_5 : 2R_3)$, R_3, and R_3. Table 2 lists the possibilities. The parts may be rotated independently, so any category involving more than one minimal relation contains infinitely many relations, even up to rotation (of the entire relation). Also, the categories are not mutually exclusive, because of the non-uniqueness of the decomposition into minimal relations.

4. Solutions to the trigonometric equation

Here we use the classification of the previous section to give a complete listing of the solutions to the trigonometric equation (2). There are some obvious solutions to (2), namely those in which U, V, W are arbitary positive rational numbers with sum $1/2$, and X, Y, Z are a permutation of U, V, W. We will call these the trivial solutions, even though the three-diagonal intersections they give rise to can look surprising. See Figure 3 for an example on the 16-gon.

The twelve roots of unity occurring in (3) are not arbitrary; therefore we must go through Table 2 to see which relations are of the correct form, i.e., expressible

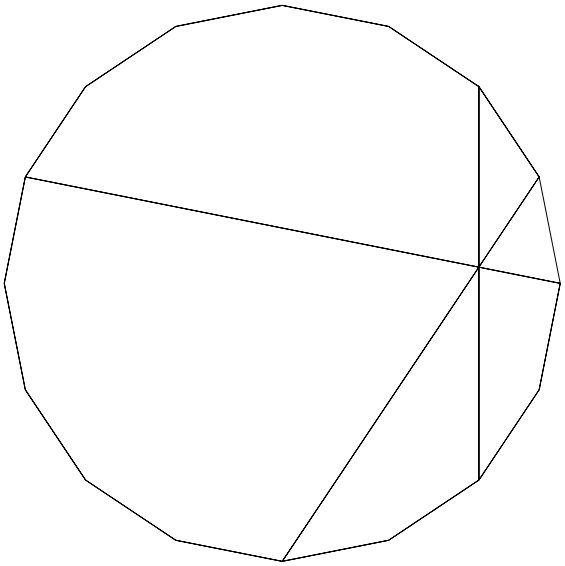


FIGURE 3. A surprising trivial solution for the 16-gon. The intersection point does not lie on any of the 16 lines of symmetry of the 16-gon.

as a sum of six roots of unity and their inverses, where the product of the six is -1. First let us prove a few lemmas that will greatly reduce the number of cases.

Lemma 4. *Let S be a relation of weight $k \leq 12$. Suppose S is stable under complex conjugation (i.e., under $\zeta \mapsto \zeta^{-1}$). Then S has a complex conjugation-stable decomposition into minimal relations; i.e., each minimal relation occurring is itself stable under complex conjugation, or can be paired with another minimal relation which is its complex conjugate.*

Proof. We will use induction on k. If S is minimal, there is nothing to prove. Otherwise let T be a (minimal) subrelation of S of minimal weight, so T is of weight at most 6. The complex conjugate $\overline{T}$ of T is another minimal relation in S. If they do not intersect, then we take the decomposition of S into T, $\overline{T}$, and a decomposition of $S \setminus (T \cup \overline{T})$ given by the inductive hypothesis. If they do overlap and the weight of T is at most 5, then $T = R_p$ for some prime p, and the fact that T intersects $\overline{T}$ implies that $T = \overline{T}$, and we get the result by applying the inductive hypothesis to $S \setminus T$.

The only remaining case is where S is of type $2(R_5 : R_3)$. If the two $(R_5 : R_3)$'s are not conjugate to each other, then for each there is a root of unity ζ such that ζ and ζ^{-1} occur in that (rotation of) $(R_5 : R_3)$. The quotient ζ^2 is then a 30-th root of unity, so ζ itself is a 60-th root of unity. Thus each $(R_5 : R_3)$ is a rotation of the "standard" $(R_5 : R_3)$ as in (5) by a 60-th root of unity, and we let Mathematica check the 60^2 possibilities. □

We do not know if the preceding lemma holds for relations of weight greater than 12.

U	V	W	X	Y	Z	Range
$1/6$	t	$1/3-2t$	$1/3+t$	t	$1/6-t$	$0<t<1/6$
$1/6$	$1/2-3t$	t	$1/6-t$	$2t$	$1/6+t$	$0<t<1/6$
$1/6$	$1/6-2t$	$2t$	$1/6-2t$	t	$1/2+t$	$0<t<1/12$
$1/3-4t$	t	$1/3+t$	$1/6-2t$	$3t$	$1/6+t$	$0<t<1/12$

TABLE 3. The nontrivial infinite families of solutions to (2).

Lemma 5. *Let S be a minimal relation of type $(R_p : T_1, \dots, T_j)$, $p \geq 5$, where the T_i involve roots of unity of order prime to p, and $j < p$. If S is stable under complex conjugation, then the particular rotation of R_p from which the T_i were "subtracted" is also stable (and hence so is the collection of the relations subtracted).*

Proof. Let ℓ be the product of the orders of the roots of unity in all the T_i. The elements of S in the original R_p can be characterized as those terms of S that are unique in their coset of μ_ℓ (the ℓ-th roots of unity), and this condition is stable under complex conjugation, so the set of terms of the R_p that were not subtracted is stable. Since $j < p$, we can pick one such term ζ. Then the quotient ζ/ζ^{-1} is a p-th root of unity, so ζ is a $2p$-th root of unity, and hence the R_p containing it is stable. □

Corollary 1. *A relation of type $(R_7 : (R_5 : R_3), R_3)$ cannot be stable under complex conjugation.*

Even with these restrictions, a very large number of cases remain, so we perform the calculation using Mathematica. Each entry of Table 2 represents a finite number of linearly parameterized (in the exponents) families of relations of weight 12. For each parameterized family, we check to see what additional constraints must be put on the parameters for the relation to be of the form of (3). Next, for each parameterized family of solutions to (3), we calculate the corresponding U, V, W, X, Y, Z and throw away solutions in which some of these are nonpositive. Finally, we sort U, V, W and X, Y, Z and interchange the two triples if $U > X$, in order to count the solutions only up to symmetry.

The results of this computation are recorded in the following theorem.

Theorem 4. *The positive rational solutions to (2), up to symmetry, can be classified as follows:*

1. *The trivial solutions, which arise from relations of type $6R_2$.*
2. *Four one-parameter families of solutions, listed in Table 3. The first arises from relations of type $4R_3$, and the other three arise from relations of type $2R_3 + 3R_2$.*
3. *Sixty-five "sporadic" solutions, listed in Table 4, which arise from the other types of weight 12 relations listed in Table 2.*

The only duplications in this list are that the second family of Table 3 gives a trivial solution for $t = 1/12$, the first and fourth families of Table 3 give the same solution when $t = 1/18$ in both, and the second and fourth families of Table 3 give the same solution when $t = 1/24$ in both.

Some explanation of the tables is in order. The last column of Table 3 gives the allowable range for the rational parameter t. The entries of Table 4 are sorted

Denominator	U	V	W	X	Y	Z	Relation type
30	1/10	2/15	3/10	2/15	1/6	1/6	$2(R_5 : R_3)$
	1/15	1/15	7/15	1/15	1/10	7/30	
	1/30	7/30	4/15	1/15	1/10	3/10	
	1/30	1/10	7/15	1/15	1/15	4/15	
	1/30	1/15	19/30	1/15	1/10	1/10	
	1/15	1/6	4/15	1/10	1/10	3/10	$(R_5 : R_3) + 2R_3$
	1/15	2/15	11/30	1/10	1/6	1/6	
	1/30	1/6	13/30	1/10	2/15	2/15	
	1/30	1/30	7/10	1/30	1/15	2/15	
	1/30	7/30	3/10	1/15	2/15	7/30	$R_5 + R_3 + 2R_2$
	1/30	1/6	11/30	1/15	1/10	4/15	
	1/30	1/10	13/30	1/30	2/15	4/15	
	1/30	1/15	8/15	1/30	1/10	7/30	
42	1/14	5/42	5/14	2/21	5/42	5/21	$(R_7 : 5R_3)$
	1/21	4/21	13/42	1/14	1/6	3/14	
	1/42	3/14	5/14	1/21	1/6	4/21	
	1/42	1/6	19/42	1/14	2/21	4/21	
	1/42	1/6	13/42	1/21	1/14	8/21	
	1/42	1/21	13/21	1/42	1/14	3/14	
60	1/20	1/12	29/60	1/15	1/10	13/60	$2(R_5 : R_3)$
	1/20	1/12	9/20	1/15	1/12	4/15	
	1/20	1/12	5/12	1/20	1/10	3/10	
	1/60	4/15	3/10	1/20	1/12	17/60	
	1/60	13/60	9/20	1/12	1/10	2/15	
	1/60	13/60	5/12	1/20	2/15	1/6	
	1/12	1/6	17/60	2/15	3/20	11/60	$(R_5 : 3R_3) + 2R_2$
	1/12	2/15	19/60	1/10	3/20	13/60	
	1/15	11/60	13/60	1/12	1/10	7/20	
	1/20	11/60	3/10	1/12	7/60	4/15	
	1/20	1/10	23/60	1/15	1/12	19/60	
	1/30	7/60	19/60	1/20	1/15	5/12	
	1/30	1/12	7/12	1/15	1/10	2/15	
	1/30	1/20	11/20	1/30	1/15	4/15	
	1/60	3/10	7/20	1/12	7/60	2/15	
	1/60	4/15	23/60	1/12	1/10	3/20	
	1/60	7/30	5/12	1/15	7/60	3/20	
	1/60	13/60	11/30	1/20	1/12	4/15	
	1/60	1/6	31/60	1/15	1/10	2/15	
	1/60	1/6	5/12	1/20	1/15	17/60	
	1/60	2/15	9/20	1/30	1/12	17/60	
	1/60	1/10	31/60	1/30	1/15	4/15	
84	1/12	3/14	19/84	11/84	13/84	4/21	$(R_7 : R_3) + 2R_2$
	1/14	11/84	23/84	1/12	2/21	29/84	
	1/21	13/84	23/84	1/14	1/12	31/84	
	1/42	1/12	7/12	1/21	1/14	4/21	
	1/84	25/84	5/14	5/84	1/12	4/21	
	1/84	5/21	5/12	5/84	1/14	17/84	
	1/84	3/14	37/84	1/21	1/12	17/84	
	1/84	1/6	43/84	1/21	1/14	4/21	
90	1/18	13/90	7/18	11/90	2/15	7/45	$(R_5 : R_3) + 2R_3$
	1/45	19/90	16/45	1/18	1/10	23/90	
	1/90	23/90	31/90	2/45	1/15	5/18	
	1/90	17/90	47/90	1/18	4/45	2/15	
120	13/120	3/20	31/120	2/15	19/120	23/120	$(R_5 : R_3) + 3R_2$
	1/12	19/120	29/120	1/10	13/120	37/120	
	1/20	23/120	29/120	1/15	13/120	41/120	
	1/60	13/120	73/120	1/20	1/12	2/15	
	1/120	7/20	43/120	7/120	11/120	2/15	
	1/120	3/10	49/120	7/120	1/12	17/120	
	1/120	4/15	53/120	1/20	11/120	17/120	
	1/120	13/60	61/120	1/20	1/12	2/15	
210	1/15	41/210	8/35	1/14	31/210	61/210	$(R_7 : (R_5 : 2R_3))$
	13/210	1/10	83/210	1/14	4/35	9/35	
	1/35	2/15	97/210	1/14	17/210	47/210	
	1/210	3/14	121/210	11/210	1/15	3/35	

TABLE 4. The 65 sporadic solutions to (2).

according to the least common denominator of U, V, W, X, Y, Z, which is also the least n for which diagonals of a regular n-gon can create arcs of the corresponding lengths. The relation type from which each solution derives is also given. The reason 11 does not appear in the least common denominator for any sporadic solution is that the relation $(R_{11} : R_3)$ cannot be put in the form of (3) with the α_j summing to 1, and hence leads to no solutions of (2). (Several other types of relations also give rise to no solutions.)

Tables 3 and 4 are the same as Bol's tables at the bottom of page 40 and on page 41 of [1], in a slightly different format.

The arcs cut by diagonals of a regular n-gon have lengths which are multiples of $2\pi/n$, so U, V, W, X, Y and Z corresponding to any configuration of three diagonals meeting must be multiples of $1/n$. With this additional restriction, trivial solutions to (2) occur only when n is even (and at least 6). Solutions within the infinite families of Table 3 occur when n is a multiple of 6 (and at least 12), and there t must be a multiple of $1/n$. Sporadic solutions with least common denominator d occur if and only if n is a multiple of d.

5. Intersections of more than three diagonals

Now that we know the configurations of three diagonals meeting, we can check how they overlap to produce configurations of more than three diagonals meeting. We will disregard configurations in which the intersection point is the center of the n-gon, since these are easily described: there are exactly $n/2$ diagonals (diameters) through the center when n is even, and none otherwise.

When k diagonals meet, they form $2k$ arcs, whose lengths we will measure as a fraction of the whole circumference (so they will be multiples of $1/n$) and list in counterclockwise order. (Warning: this is different from the order used in Tables 3 and 4.) The least common denominator of the numbers in this list will be called the denominator of the configuration. It is the least n for which the configuration can be realized as diagonals of a regular n-gon.

Lemma 6. *If a configuration of $k \geq 2$ diagonals meeting at an interior point other than the center has denominator dividing d, then any configuration of diagonals meeting at that point has denominator dividing* $\mathrm{LCM}(2d, 3)$.

Proof. We may assume $k = 2$. Any other configuration of diagonals through the intersection point is contained in the union of configurations obtained by adding one diagonal to the original two, so we may assume the final configuration consists of three diagonals, two of which were the original two. Now we need only go through our list of three-diagonal intersections.

It can be checked (using Mathematica) that removing any diagonal from a sporadic configuration of three intersecting diagonals yields a configuration whose denominator is the same or half as much, except that it is possible that removing a diagonal from a three-diagonal configuration of denominator 210 or 60 yields one of denominator 70 or 20, respectively, which proves the desired result for these cases. The additive group generated by 1/6 and the normalized arc lengths of a configuration obtained by removing a diagonal from a configuration corresponding to one of the families of Table 3 contains $2t$ where t is the parameter, (as can be verified using Mathematica again), which means that adding that third diagonal can at most double the denominator (and throw in a factor of 3, if it isn't already

								Range
t	t	t	$1/6-2t$	$1/6$	$1/3+t$	$1/6$	$1/6-2t$	$0<t<1/12$
t	$1/6-t$	$1/6-t$	$1/6-t$	t	$1/6$	$1/6+t$	$1/6$	$0<t<1/6$
$1/6-4t$	$2t$	t	$3t$	$1/6-4t$	$1/6$	$1/6+t$	$1/3+t$	$0<t<1/24$
$2t$	$1/2-t$	$2t$	$1/6-2t$	t	$1/6-t$	t	$1/6-2t$	$0<t<1/12$
$1/3-4t$	$1/6+t$	$1/2-3t$	$-1/6+4t$	$1/6-2t$	t	$1/6-t$	$-1/6+4t$	$1/24<t<1/12$
$2t$	t	$3t$	$1/6-2t$	$1/6$	$1/6-t$	$1/3-t$	$1/6-2t$	$0<t<1/12$
t	t	$2t$	$1/3-t$	$1/6$	$1/6-t$	$1/6-t$	$1/6-t$	$0<t<1/6$
$1/3-4t$	$1/6$	t	t	$1/6-2t$	$1/3-2t$	$3t$	$3t$	$0<t<1/12$
$2t$	$1/3-2t$	$1/6-t$	$1/6-t$	$1/6$	$1/6$	t	t	$0<t<1/6$
$1/3-4t$	$2t$	t	t	$1/6-2t$	$1/6$	$1/6+t$	$1/6+t$	$0<t<1/12$
$1/3-4t$	$2t$	$1/6-t$	t	$1/6-2t$	$2t$	$1/3-t$	$3t$	$0<t<1/12$
$2t$	$1/6-t$	t	$1/6-t$	t	$1/6-t$	$2t$	$1/2-3t$	$0<t<1/6$

TABLE 5. The one-parameter families of four-diagonal configurations.

there). Similarly, it is easily checked (even by hand), that the subgroup generated by the normalized arc lengths of a configuration obtained by removing one of the three diagonals of a configuration corresponding to a trivial solution to (2) but with intersection point not the center, contains twice the arc lengths of the original configuration. □

Corollary 2. *If a configuration of three or more diagonals meeting includes three forming a sporadic configuration, then its denominator is 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, or 420.*

Proof. Combine the lemma with the list of denominators of sporadic configurations listed in Table 4. □

For $k \geq 4$, a list of $2k$ positive rational numbers summing to 1 arises this way if and only if the lists of length $2k-2$ which would arise by removing the first or second diagonal actually correspond to $k-1$ intersecting diagonals. Suppose $k=4$. If we specify the sporadic configuration or parameterized family of configurations that arise when we remove the first or second diagonal, we get a set of linear conditions on the eight arc lengths. Corollary 2 tells us that we get a configuration with denominator among 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, and 420, if one of these two is sporadic. Using Mathematica to perform this computation for the rest of possibilities in Theorem 4 shows that the other four-diagonal configurations, up to rotation and reflection, fall into 12 one-parameter families, which are listed in Table 5 by the eight normalized arc lengths and the range for the parameter t, with a finite number of exceptions of denominators among 12, 18, 24, 30, 36, 42, 48, 60, 84, and 120.

We will use a similar argument when $k=5$. Any five-diagonal configuration containing a sporadic three-diagonal configuration will again have denominator among 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, and 420. Any other five-diagonal configuration containing one of the exceptional four-diagonal configurations will have denominator among 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 96, 120, 168, and 240, by Lemma 6. Finally, another Mathematica computation shows that the one-parameter families of four-diagonal configurations overlap to produce the

										Range
t	$2t$	$1/6-2t$	$1/6$	$1/6-t$	$1/6-t$	$1/6$	$1/6-2t$	$2t$	t	$0<t<1/12$
t	$2t$	$1/6-4t$	$1/6$	$1/6+t$	$1/6+t$	$1/6$	$1/6-4t$	$2t$	t	$0<t<1/24$
t	$1/6-2t$	$-1/6+4t$	$1/3-4t$	$1/6+t$	$1/6+t$	$1/3-4t$	$-1/6+4t$	$1/6-2t$	t	$1/24<t<1/12$
t	$1/6-2t$	$2t$	$1/3-4t$	$3t$	$3t$	$1/3-4t$	$2t$	$1/6-2t$	t	$0<t<1/12$

TABLE 6. The one-parameter families of five-diagonal configurations.

one-parameter families listed (up to rotation and reflection) in Table 6, and a finite number of exceptions of denominators among 18, 24, and 30.

For $k=6$, any six-diagonal configuration containing a sporadic three-diagonal configuration will again have denominator among 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, and 420. Any six-diagonal configuration containing one of the exceptional four-diagonal configurations will have denominator among 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 96, 120, 168, and 240. Any six-diagonal configuration containing one of the exceptional five-diagonal configurations will have denominator among 18, 24, 30, 36, 48, and 60. Another Mathematica computation shows that the one-parameter families of five-diagonal configurations cannot combine to give a six-diagonal configuration.

Finally for $k \geq 7$, any k-diagonal configuration must contain an exceptional configuration of 3, 4, or 5 diagonals, and hence by Lemma 6 has denominator among 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 90, 96, 120, 168, 180, 210, 240, and 420.

We summarize the results of this section in the following.

Proposition 1. *The configurations of $k \geq 4$ diagonals meeting at a point not the center, up to rotation and reflection, fall into the one-parameter families listed in Tables 5 and 6, with finitely many exceptions (for fixed k) of denominators among 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 90, 96, 120, 168, 180, 210, 240, and 420.*

In fact, many of the numbers listed in the proposition do not actually occur as denominators of exceptional configurations. For example, it will turn out that the only denominator greater than 120 that occurs is 210.

6. The formula for intersection points

Let $a_k(n)$ denote the number of points inside the regular n-gon other than the center where exactly k lines meet. Let $b_k(n)$ denote the number of k-tuples of diagonals which meet at a point inside the n-gon other than the center. Each interior point at which exactly m diagonals meet gives rise to $\binom{m}{k}$ such k-tuples, so we have the relationship

$$b_k(n) = \sum_{m \geq k} \binom{m}{k} a_m(n) \tag{7}$$

Since every four distinct vertices of the n-gon determine one pair of diagonals which intersect inside, the number of such pairs is exactly $\binom{n}{4}$, but if n is even, then $\binom{n/2}{2}$ of these are pairs which meet at the center, so

$$b_2(n) = \binom{n}{4} - \binom{n/2}{2}\delta_2(n). \tag{8}$$

(Recall that $\delta_m(n)$ is defined to be 1 if n is a multiple of m, and 0 otherwise.)

We will use the results of the previous two sections to deduce the form of $b_k(n)$ and then the form of $a_k(n)$. To avoid having to repeat the following, let us make a definition.

Definition . A function on integers $n \geq 3$ will be called *tame* if it is a linear combination (with rational coefficients) of the functions n^3, n^2, n, 1, $n^2\delta_2(n)$, $n\delta_2(n)$, $\delta_2(n)$, $\delta_4(n)$, $n\delta_6(n)$, $\delta_6(n)$, $\delta_{12}(n)$, $\delta_{18}(n)$, $\delta_{24}(n)$, $\delta_{24}(n-6)$, $\delta_{30}(n)$, $\delta_{36}(n)$, $\delta_{42}(n)$, $\delta_{48}(n)$, $\delta_{60}(n)$, $\delta_{72}(n)$, $\delta_{84}(n)$, $\delta_{90}(n)$, $\delta_{96}(n)$, $\delta_{120}(n)$, $\delta_{168}(n)$, $\delta_{180}(n)$, $\delta_{210}(n)$, and $\delta_{420}(n)$.

Proposition 2. *For each $k \geq 2$, the function $b_k(n)/n$ on integers $n \geq 3$ is tame.*

Proof. The case $k = 2$ is handled by (8), so assume $k \geq 3$. Each list of $2k$ normalized arc lengths as in Section 5 corresponding to a configuration of k diagonals meeting at a point other than the center, considered up to rotation (but not reflection), contributes n to $b_k(n)$. (There are n places to start measuring the arcs from, and these n configurations are distinct, because the corresponding intersection points differ by rotations of multiples of $2\pi/n$, and by assumption they are not at the center.) So $b_k(n)/n$ counts such lists.

Suppose $k = 3$. When n is even, the family of trivial solutions to the trigonometric equation (2) has $U = a/n$, $V = b/n$, $W = c/n$, where a, b, and c are positive integers with sum $n/2$, and X, Y, and Z are some permutation of U, V, W. Each permutation gives rise to a two-parameter family of six-long lists of arc lengths, and the number of lists within each family is the number of partitions of $n/2$ into three positive parts, which is a quadratic polynomial in n. Similarly each family of solutions in Table 3 gives rise to a number of one-parameter families of lists, when n is a multiple of 6, each containing $\lceil n/6 \rceil - 1$ or $\lceil n/12 \rceil - 1$ lists. These functions of n (extended to be 0 when 6 does not divide n) are expressible as a linear combination of $n\delta_6(n)$, $\delta_6(n)$, and $\delta_{12}(n)$. Finally the sporadic solutions to 2 give rise to a finite number of lists, having denominators among 30, 42, 60, 84, 90, 120, and 210, so their contribution to $b_3(n)/n$ is a linear combination of $\delta_{30}(n), \ldots, \delta_{210}(n)$.

But these families of lists overlap, so we must use the Principle of Inclusion-Exclusion to count them properly. To show that the result is a tame function, it suffices to show that the number of lists in any intersection of these families is a tame function. When two of the trivial families overlap but do not coincide, they overlap where two of the a, b, and c above are equal, and the corresponding lists lie in one of the one-parameter families $(t, t, t, t, 1/2 - 2t, 1/2 - 2t)$ or $(t, t, t, 1/2 - 2t, t, 1/2 - 2t)$ (with $0 < t < 1/4$), each of which contain $\lceil n/4 \rceil - 1$ lists (for n even). This function of n is a combination of $n\delta_2(n)$, $\delta_2(n)$, and $\delta_4(n)$, hence it is tame. Any other intersection of the infinite families must contain the intersection of two one-parameter families which are among the two above or arise from Table 3, and a Mathematica computation shows that such an intersection consists of at most a single list of denominator among 6, 12, 18, 24, and 30. And, of course, any intersection involving a single sporadic list, can contain at most that sporadic list. Thus the number of lists within any intersection is a tame function of n. Finally we must delete the lists which correspond to configurations of diagonals meeting at the center. These are the lists within the trivial two-parameter family $(t, u, 1/2 - t - u, t, u, 1/2 - t - u)$, so their number is also a tame function of n, by the Principle of Inclusion-Exclusion again. Thus $b_3(n)/n$ is tame.

Next suppose $k = 4$. The number of lists within each family listed in Table 5, or the reflection of such a family, is (when n is divisible by 6) the number of multiples of $1/n$ strictly between α and β, where the range for the parameter t is $\alpha < t < \beta$. This number is $\lceil \beta n \rceil - 1 - \lfloor \alpha n \rfloor$. Since the table shows that α and β are always multiples of $1/24$, this function of n is expressible as a combination of $n\delta_6(n)$ and a function on multiples of 6 depending only on n mod 24, and the latter can be written as a combination of $\delta_6(n)$, $\delta_{12}(n)$, $\delta_{24}(n)$, and $\delta_{24}(n-6)$, so it is tame. Mathematica shows that when two of these families are not the same, they intersect in at most a single list of denominator among 6, 12, 18, and 24. So these and the exceptions of Proposition 1 can be counted by a tame function. Thus, again by the Principle of Inclusion-Exclusion, $b_4(n)/n$ is tame.

The proof for $k = 5$ is identical to that of $k = 4$, using Table 6 instead of Table 5, and using another Mathematica computation which shows that the intersections of two one-parameter families of lists consist of at most a single list of denominator 24.

The proof for $k \geq 6$ is even simpler, because then there are only the exceptional lists. By Proposition 1, $b_k(n)/n$ is a linear combination of $\delta_m(n)$ where m ranges over the possible denominators of exceptional lists listed in the proposition, so it is tame. □

Lemma 7. *A tame function is determined by its values at $n = 3, 4, 5, 6, 7, 8, 9, 10, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 84, 90, 96, 120, 168, 180, 210,$ and 420.*

Proof. By linearity, it suffices to show that if a tame function f is zero at those values, then f is the zero linear combination of the functions in the definition of a tame function. The vanishing at $n = 3, 5, 7$, and 9 forces the coefficients of n^3, n^2, n, and 1 to vanish, by Lagrange interpolation. Then comparing the values at $n = 4$ and $n = 10$ shows that the coefficient of $\delta_4(n)$ is zero. The vanishing at $n = 4$, 8, and 10 forces the coefficients of $n^2\delta_2(n)$, $n\delta_2(n)$, and $\delta_2(n)$ to vanish. Comparing the values at $n = 6$ and $n = 54$ shows that the coefficient of $n\delta_6$ is zero. Comparing the values at $n = 6$ and $n = 66$ shows that the coefficient of $\delta_2 4(n-6)$ is zero.

At this point, we know that $f(n)$ is a combination of $\delta_m(n)$, for $m = 6$, 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 90, 96, 120, 168, 180, 210, and 420. For each m in turn, $f(m) = 0$ now implies that the coefficient of $\delta_m(n)$ is zero. □

Proof of Theorem 1. Computation (see the appendix) shows that the tame function $b_8(n)/n$ vanishes at all the numbers listed in Lemma 7. Hence by that lemma, $b_8(n) = 0$ for all n. Thus by (7), $a_k(n)$ and $b_k(n)$ are identically zero for all $k \geq 8$ as well.

By reverse induction on k, we can invert (7) to express $a_k(n)$ as a linear combination of $b_m(n)$ with $m \geq k$. Hence $a_k(n)/n$ is tame as well for each $k \geq 2$.

Computation shows that the equations

$$\begin{aligned}
a_2(n)/n &= (n^3-6n^2+11n-6)/24+(-5n^2+46n-72)/16\cdot\delta_2(n)\\
&\quad -9/4\cdot\delta_4(n)+(-19n+110)/2\cdot\delta_6(n)+54\cdot\delta_{12}(n)+84\cdot\delta_{18}(n)\\
&\quad +50\cdot\delta_{24}(n)-24\cdot\delta_{30}(n)-100\cdot\delta_{42}(n)-432\cdot\delta_{60}(n)\\
&\quad -204\cdot\delta_{84}(n)-144\cdot\delta_{90}(n)-204\cdot\delta_{120}(n)-144\cdot\delta_{210}(n)\\
a_3(n)/n &= (5n^2-48n+76)/48\cdot\delta_2(n)+3/4\cdot\delta_4(n)+(7n-38)/6\cdot\delta_6(n)\\
&\quad -8\cdot\delta_{12}(n)-20\cdot\delta_{18}(n)-16\cdot\delta_{24}(n)-19\cdot\delta_{30}(n)+8\cdot\delta_{42}(n)\\
&\quad +68\cdot\delta_{60}(n)+60\cdot\delta_{84}(n)+48\cdot\delta_{90}(n)+60\cdot\delta_{120}(n)+48\cdot\delta_{210}(n)\\
a_4(n)/n &= (7n-42)/12\cdot\delta_6(n)-5/2\cdot\delta_{12}(n)-4\cdot\delta_{18}(n)+3\cdot\delta_{24}(n)\\
&\quad +6\cdot\delta_{42}(n)+34\cdot\delta_{60}(n)-6\cdot\delta_{84}(n)-6\cdot\delta_{120}(n)\\
a_5(n)/n &= (n-6)/4\cdot\delta_6(n)-3/2\cdot\delta_{12}(n)-2\cdot\delta_{24}(n)+4\cdot\delta_{42}(n)\\
&\quad +6\cdot\delta_{84}(n)+6\cdot\delta_{120}(n)\\
a_6(n)/n &= 4\cdot\delta_{30}(n)-4\cdot\delta_{60}(n)\\
a_7(n)/n &= \delta_{30}(n)+4\cdot\delta_{60}(n)
\end{aligned}$$

hold for all the n listed in Lemma 7, so the lemma implies that they hold for all $n \geq 3$. These formulas imply the remarks in the introduction about the maximum number of diagonals meeting at an interior point other than the center. Finally

$$\begin{aligned}
I(n) &= \delta_2(n)+\sum_{k=2}^{\infty} a_k(n)\\
&= \delta_2(n)+\sum_{k=2}^{7} a_k(n),
\end{aligned}$$

which gives the desired formula. (The $\delta_2(n)$ in the expression for $I(n)$ is to account for the center point when n is even, which is the only point not counted by the a_k.) □

7. The formula for regions

We now use the knowledge obtained in the proof of Theorem 1 about the number of interior points through which exactly k diagonals pass to calculate the number of regions formed by the diagonals.

Proof of Theorem 2. Consider the graph formed from the configuration of a regular n-gon with its diagonals, in which the vertices are the vertices of the n-gon together with the interior intersection points, and the edges are the sides of the n-gon together with the segments that the diagonals cut themselves into. As usual, let V denote the number of vertices of the graph, E the number of edges, and F the number of regions formed, including the region outside the n-gon. We will employ Euler's Formula $V - E + F = 2$.

Clearly $V = n + I(n)$. We will count edges by counting their ends, which are $2E$ in number. Each vertex has $n-1$ edge ends, the center (if n is even) has n edge ends, and any other interior point through which exactly k diagonals pass has $2k$

edge ends, so

$$2E = n(n-1) + n\delta_2(n) + \sum_{k=2}^{\infty} 2k a_k(n).$$

So the desired number of regions, not counting the region outside the n-gon, is

$$\begin{aligned} F - 1 &= E - V + 1 \\ &= \left[n(n-1)/2 + n\delta_2(n)/2 + \sum_{k=2}^{\infty} k a_k(n) \right] - [n + I(n)] + 1. \end{aligned}$$

Substitution of the formulas derived in the proof of Theorem 1 for $a_k(n)$ and $I(n)$ yields the desired result. □

Appendix: computations and tables

In Table 7 we list $I(n), R(n), a_2(n), \ldots, a_7(n)$ for $n = 4, 5, \ldots, 30$. To determine the polynomials listed in Theorem 1 more data was needed especially for $n \equiv 0 \bmod 6$. The largest n for which this was required was 420. For speed and memory conservation, we took advantage of the regular n-gon's rotational symmetry and focused our attention on only $2\pi/n$ radians of the n-gon. The data from this computation is found in Table 8. Although we only needed to know the values at those n listed in Lemma 7 of Section 6, we give a list for $n = 6, 12, \ldots, 420$ so that the nice patterns can be seen.

The numbers in these tables were found by numerically computing (using a C program and 64 bit precision) all possible $\binom{n}{4}$ intersections, and sorting them by their x coordinate. We then focused on runs of points with close x coordinates, looking for points with close y coordinates.

Several checks were made to eliminate any fears (arising from round-off errors) of distinct points being mistaken as close. First, the C program sent data to Maple which checked that the coordinates of close points agreed to at least 40 decimal places. Second, we verified for each n that close points came in counts of the form $\binom{k}{2}$ (k diagonals meeting at a point give rise to $\binom{k}{2}$ close points. Hence, any run whose length is not of this form indicates a computational error).

A second program was then written and run on a second machine to make the computations completely rigorous. It also found the intersection points numerically, sorted them and looked for close points, but, to be absolutely sure that a pair of close points p_1 and p_2 were actually the same, it checked that for the two pairs of diagonals (l_1, l_2) and (l_3, l_4) determining p_1 and p_2, respectively, the triples l_1, l_2, l_3 and l_1, l_2, l_4 each divided the circle into arcs of lengths consistent with Theorem 4. Since this test only involves comparing rational numbers, it could be performed exactly.

A word should also be said concerning limiting the search to $2\pi/n$ radians of the n-gon. Both programs looked at slightly smaller slices of the n-gon to avoid problems caused by points near the boundary. We further subdivided this region into twenty smaller pieces to make the task of sorting the intersection points manageable. More precisely, we limited our search to points whose angle with the origin fell between $[c_1 + 2\pi(m-1)/(20n) + \varepsilon, c_1 + 2\pi m/(20n) - \varepsilon)$, $m = 1, 2, \ldots 20$, and also made sure not to include the origin in the count. Here ε was chosen to be .00000000001 and c_1 was chosen to be .00000123 ($c_1 = 0$ would have led to problems since there are many intersection points with angle 0 or $2\pi/n$). To make sure

n	$a_2(n)$	$a_3(n)$	$a_4(n)$	$a_5(n)$	$a_6(n)$	$a_7(n)$	$I(n)$	$R(n)$
3							0	1
4							1	4
5	5						5	11
6	12						13	24
7	35						35	50
8	40	8					49	80
9	126						126	154
10	140	20					161	220
11	330						330	375
12	228	60	12				301	444
13	715						715	781
14	644	112					757	952
15	1365						1365	1456
16	1168	208					1377	1696
17	2380						2380	2500
18	1512	216	54	54			1837	2466
19	3876						3876	4029
20	3360	480					3841	4500
21	5985						5985	6175
22	5280	660					5941	6820
23	8855						8855	9086
24	6144	864	264	24			7297	9024
25	12650						12650	12926
26	11284	1196					12481	13988
27	17550						17550	17875
28	15680	1568					17249	19180
29	23751						23751	24129
30	13800	2250	420	180	120	30	16801	21480

TABLE 7. A listing of $I(n)$,$R(n)$ and $a_2(n), \ldots, a_7(n)$, $n = 3, 4, \ldots, 30$. Note that, when n is even, $I(n)$ also counts the point in the center.

that no intersection points were omitted, the number of points found (counting multiplicity) was compared with $(\binom{n}{4} - \binom{n/2}{2}\delta_2)/n$.

ACKNOWLEDGEMENTS

We thank Joel Spencer and Noga Alon for helpful conversations. Also we thank Jerry Alexanderson, Jeff Lagarias, Hendrik Lenstra, and Gerry Myerson for pointing out to us many of the references below.

REFERENCES

[1] G. Bol: Beantwoording van prijsvraag no. 17, *Nieuw Archief voor Wiskunde* **18** (1936), 14–66.
[2] J. H. Conway and A. J. Jones: Trigonometric Diophantine equations (On vanishing sums of roots of unity), *Acta Arith.* **30** (1976), 229–240.
[3] H. T. Croft and M. Fowler: On a problem of Steinhaus about polygons, *Proc. Camb. Phil. Soc.* **57** (1961), 686–688.
[4] H. Harborth: Diagonalen im regularen n-Eck, *Elem. Math.* **24** (1969), 104–109.

n	$\frac{a_2(n)}{n}$	$\frac{a_3(n)}{n}$	$\frac{a_4(n)}{n}$	$\frac{a_5(n)}{n}$	$\frac{a_6(n)}{n}$	$\frac{a_7(n)}{n}$	$\frac{I(n)-1}{n}$	n	$\frac{a_2(n)}{n}$	$\frac{a_3(n)}{n}$	$\frac{a_4(n)}{n}$	$\frac{a_5(n)}{n}$	$\frac{a_6(n)}{n}$	$\frac{a_7(n)}{n}$	$\frac{I(n)-1}{n}$
6	2						2	216	392564	4848	119	49			397580
12	19	5	1				25	222	426836	5166	126	54			432182
18	84	12	3	3			102	228	463303	5441	127	54			468925
24	256	36	11	1			304	234	501762	5718	129	57			507666
30	460	75	14	6	4	1	560	240	541612	6121	165	61		5	547964
36	1179	109	11	6			1305	246	584782	6340	140	60			591322
42	1786	194	27	13			2020	252	629399	6693	137	70			636299
48	3168	220	25	7			3420	258	676580	6972	147	63			683762
54	4722	288	24	12			5046	264	725976	7276	151	61			733464
60	6251	422	63	12		5	6753	270	777420	7643	150	66	4	1	785284
66	9172	460	35	15			9682	276	831575	7969	155	66			839765
72	12428	504	35	13			12980	282	887986	8326	161	69			896542
78	15920	642	42	18			16622	288	947132	8640	161	67			956000
84	20007	805	43	28			20883	294	1008358	9056	174	76			1017664
90	25230	863	45	21	4	1	26164	300	1072171	9462	203	72		5	1081913
96	31240	948	53	19			32260	306	1139436	9780	171	75			1149462
102	37786	1096	56	24			38962	312	1208944	10164	179	73			1219360
108	45447	1201	53	24			46725	318	1281100	10582	182	78			1291942
114	53768	1368	63	27			55226	324	1356315	10957	179	78			1367529
120	62652	1601	95	31		5	64384	330	1434110	11375	189	81	4	1	1445760
126	73676	1658	72	34			75440	336	1514816	11856	193	89			1526954
132	85319	1825	71	30			87245	342	1598970	12216	192	84			1611462
138	97990	2002	77	33			100102	348	1685843	12661	197	84			1698785
144	112100	2136	77	31			114344	354	1775788	13108	203	87			1789186
150	127070	2345	84	36	4	1	129540	360	1868312	13669	231	91		5	1882308
156	143635	2549	85	36			146305	366	1965272	14010	210	90			1979582
162	161520	2736	87	39			164382	372	2064919	14465	211	90			2079685
168	180504	3008	95	47			183654	378	2167754	14930	219	97			2183000
174	201448	3178	98	42			204766	384	2274136	15396	221	91			2289844
180	223251	3470	129	42		5	226897	390	2383690	15885	224	96	4	1	2399900
186	247562	3630	105	45			251342	396	2496999	16369	221	96			2513685
192	273144	3844	109	43			277140	402	2613536	16896	231	99			2630762
198	300294	4092	108	48			304542	408	2733888	17380	235	97			2751600
204	329171	4357	113	48			333689	414	2857752	17898	234	102			2875986
210	359556	4661	125	55	4	1	364402	420	2984383	18598	273	112		5	3003371

TABLE 8. The number of intersection points for one piece of the pie (i.e. $2\pi/n$ radians), $n = 6, 12, \ldots, 420$.

[5] H. Harborth: Number of intersections of diagonals in regular n-gons, *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, 151–153.
[6] H. Heineken: Regelmässige Vielecke und ihre Diagonalen, *Enseignement Math.* (2), sér. 8 (1962), 275–278.
[7] H. Heineken: Regelmässige Vielecke und ihre Diagonalen II, *Rend. Sem. Mat. Univ. Padova* **41** (1968), 332–344.
[8] H. Mann: On linear relations between roots of unity, *Mathematika* **12** (1965), 107–117.
[9] G. Myerson: Rational products of sines of rational angles, *Aequationes Math.* **45** (1993), 70–82.
Math. Gaz. **61** (1977), 55–58.
[10] J. F. Rigby: Adventitious quadrangles: a geometrical approach, *Math. Gaz.* **62** (1978), 183–191.
[11] J. F. Rigby: Multiple intersections of diagonals of regular polygons, and related topics, *Geom. Dedicata* **9** (1980), 207–238.
[12] I. J. Schoenberg: A note on the cyclotomic polynomial, *Mathematika* **11** (1964), 131–136.
[13] H. Steinhaus: *Mathematical Snapshots*, Oxford University Press, 1983, 259–260.
[14] H. Steinhaus: Problem 225, *Colloq. Math.* **5** (1958).
[15] C. E. Tripp: Adventitious angles, *Math. Gaz.* **59** (1975), 98–106.

AT&T Bell Laboratories, Murray Hill, NJ 07974, USA
Current address: University of California at Berkeley, Berkeley, CA 94720-3840, USA
E-mail address: poonen@math.berkeley.edu

AT&T Bell Laboratories, Murray Hill, NJ 07974, USA
Current address: Princeton University, Princeton, NJ 08544-1000, USA
E-mail address: miker@math.princeton.edu

Labelled and unlabelled enumeration of k-gonal 2-trees

Gilbert Labelle, Cédric Lamathe and Pierre Leroux

April 1, 2003

Abstract

In this paper[1], we generalize 2-trees by replacing triangles by quadrilaterals, pentagons or k-sided polygons (k-gons), where $k \geq 3$ is given. This generalization, to k-gonal 2-trees, is natural and is closely related, in the planar case, to some specializations of the cell-growth problem. Our goal is the labelled and unlabelled enumeration, of k-gonal 2-trees according to the number n of k-gons. We give explicit formulas in the labelled case, and, in the unlabelled case, recursive and asymptotic formulas. We also enumerate these structures according to their perimeter.

1 Introduction

The class of *bidimensional trees*, or in brief *2-trees*, is extensively studied in the literature. For instance, see [7] and [5, 6] and their references; see also [10, 11]. Essentially, a 2-tree is a connected simple graph composed by triangles glued along their edges in a tree-like fashion, that is, without cycles (of triangles). In [8], Harary et al. enumerated a variant of the cell-growth problem, namely plane and planar (in the sense that all faces, except possibly the external face, are also k-sided polygons, also called outerplanar) 2-trees, in which triangles have been replaced by quadrilaterals, pentagons or k-sided polygons (k-gons), where $k \geq 3$ is fixed. Such 2-trees, built on k-gons, are called *k-gonal 2-trees*. This generalization is natural and the purpose of this work is the enumeration of free k-gonal 2-trees, *i.e.*, seen as simple graphs, without any condition of planarity. Figure 1, a) and b), and Figure 2 a) show examples of k-gonal 2-trees, for $k = 3, 5$ and 4, respectively.

Our goal is the labelled and unlabelled enumeraton of k-gonal 2-trees, according to the number of k-gons. We give explicit formulas in the labelled case and recursive and asymptotic formulas in the unlabelled case. This is the full version of a paper presented at the "Mathematics and computer science" conference in Versailles, France, in September 2002 (see [15]). More complete proofs are given, in particular for the asymptotic formulas, and a section has been added on the enumeration of k-gonal 2-trees according to their perimeter.

[1]With the support of FCAR (Québec) and NSERC (Canada).

It was recently brought to our attention that Ton Kloks [10, 11] had enumerated unlabelled *biconnected partial 2-trees* according to the number of vertices, in his 1993 thesis. These structures are more general than k-gonal 2-trees since various size of polygons can occur in the same graph and some polygons may have missing edges.

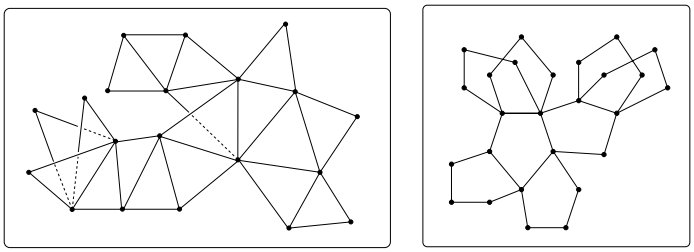


Figure 1: k-gonal 2-trees with $k = 3$ and $k = 5$

We say that a k-gonal 2-tree is *oriented* if its edges are oriented in such a way that each k-gon forms an oriented cycle; see Figure 2 b). In fact, for any k-gonal 2-tree s, the orientation of any one of its edges can be extended uniquely to all of s by first orienting all the polygons to which the edge belongs and then continuing recursively on all adjacent polygons. The coherence of the extension is ensured by the arborescent (acyclic) nature of 2-trees.

We denote by $\mathcal{A}$ and $\mathcal{A}_o$ the species of k-gonal 2-trees and of oriented k-gonal 2-trees. For these species, we use the symbols $-$, $\diamond$ and $\underline{\diamond}$ as upper indices to indicate that the structures are pointed at an edge, at a k-gon, and at a k-gon having itself a distinguished edge, respectively.

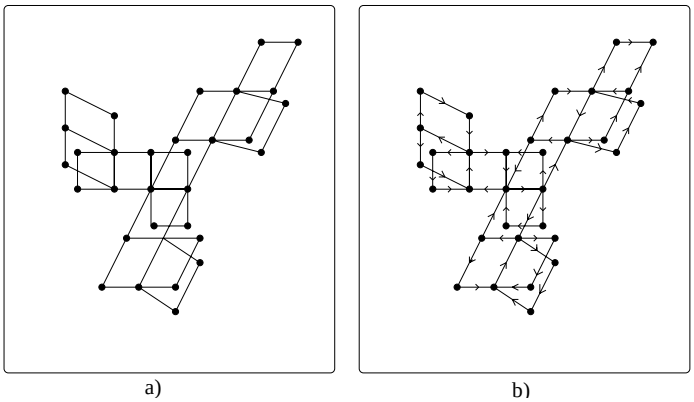


Figure 2: A unoriented and oriented 4-gonal 2-tree

Following the approach of Fowler et al. in [5, 6], which corresponds to the case $k = 3$, we label the 2-trees at their k-gons and give functional equations

which relate these various pointed species together and eventually lead to their enumeration. The main difficulty of this extension from triangles to k-gons comes, as we will see later, from the case where k is an even integer.

The first step is an extension of the dissymmetry theorem for 2-trees to the k-gonal case. The proof is similar to the case $k = 3$ and is omitted (see [5, 6]).

Theorem 1. DISSYMMETRY THEOREM FOR k-GONAL 2-TREES. The species $\mathcal{A}_o$ and $\mathcal{A}$ of oriented and unoriented k-gonal 2-trees, respectively, satisfy the following isomorphisms of species:

$$\mathcal{A}_o^- + \mathcal{A}_o^\diamond = \mathcal{A}_o + \mathcal{A}_{\overline{o}}^{\overline{\diamond}}, \tag{1}$$

$$\mathcal{A}^- + \mathcal{A}^\diamond = \mathcal{A} + \mathcal{A}^{\overline{\diamond}}. \tag{2}$$

There is yet another species to introduce, which plays an essential role in the process. It is the species $B = \mathcal{A}^\rightarrow$ of oriented-edge rooted (k-gonal) 2-trees, that is of 2-trees where an edge is selected and oriented. As mentionned above, the orientation of the rooted edge can be extended uniquely to an orientation of the 2-tree so that there is a canonical isomorphism $B = \mathcal{A}_o^-$ which can be used for all enumerative purposes. However, it is often useful not to perform this extension and to consider that only the rooted edge is oriented, as we will see.

In the next section, we characterize the species $B = \mathcal{A}^\rightarrow$ by a combinatorial functional equation and state some of its properties. The goal is to express the various pointed species occuring in the dissymmetry theorem in terms of B and to deduce enumerative results for the species $\mathcal{A}_o$ and $\mathcal{A}$. The oriented case is simpler, and carried out first, in Section 3. The unoriented case is analyzed in Section 4, distinguishing two cases according to the parity of the integer k. Enumeration of k-gonal 2-trees according to the perimeter is carried out in Section 5. Finally, asymptotic results are presented in Section 6.

This paper uses the framework of species theory. See Chapter 1 of [3] for an introduction. The main tool for our purposes is the composition theorem which can be stated as follows: let the species F be the (partitionnal) composition of two species, $F = G \circ H$. Then, the exponential generating function

$$F(x) = \sum_{n \geq 0} f_n \frac{x^n}{n!},$$

where $f_n = |F[n]|$ is the number of labelled F-structures of order n, and the tilde generating function

$$\widetilde{F}(x) = \sum_{n \geq 0} \widetilde{f}_n x^n,$$

where $\widetilde{f}_n = |F[n]/\mathbb{S}_n|$ is the number of unlabelled F-structures of order n, satisfy the following equations:

$$F(x) = G(H((x)), \tag{3}$$

$$\widetilde{F}(x) = Z_G(\widetilde{H}(x), \widetilde{H}(x^2), \ldots), \tag{4}$$

where $Z_G(x_1, x_2, \ldots)$ is the cycle index series of G.

2 The species B of oriented-edge rooted 2-trees

The species $B = \mathcal{A}^{\rightarrow}$ plays a central role in the study of k-gonal 2-trees. The following theorem is an extension to a general k of the case $k = 3$. Note that formula (5) also makes sense for $k = 2$ and corresponds to edge-labelled (ordinary) rooted trees.

Theorem 2. The species $B = \mathcal{A}^{\rightarrow}$ of oriented-edge rooted k-gonal 2-trees satisfies the following functional equation (isomorphism):

$$B = E(XB^{k-1}), \tag{5}$$

where E represents the species of sets and X is the species of singleton k-gons.

Proof. We decompose an $\mathcal{A}^{\rightarrow}$-structure as a set of *pages*, that is, of maximal subgraphs sharing only one k-gon with the rooted edge. For each page, the orientation of the rooted edge permits to define a linear order and an orientation on the $k-1$ remaining edges of the polygon having this edge, in some conventional way, for example in the fashion illustrated in Figure 3 a), for the odd case, and b), for the even case. These edges being oriented, we can glue on them some B-structures. We then deduce relation (5). ■

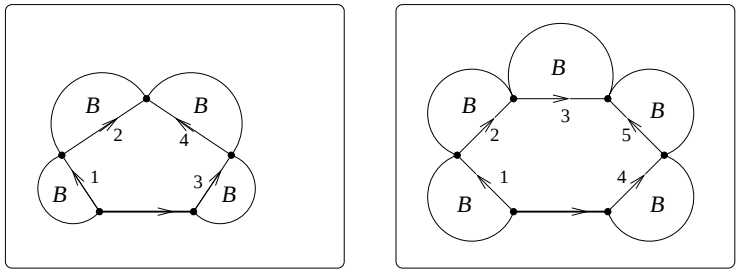


Figure 3: An oriented page for a) $k = 5$, b) $k = 6$

We can easily relate the species $B = \mathcal{A}^{\rightarrow}$ to the species of rooted trees denoted by A, characterized by the functional equation $A = XE(A)$, where X is now the species of singleton vertices. Indeed from (5), we deduce successively

$$(k-1)XB^{k-1} = (k-1)XE((k-1)XB^{k-1}), \tag{6}$$

knowing that $E^m(X) = E(mX)$, and, by unicity,

$$(k-1)XB^{k-1} = A((k-1)X). \tag{7}$$

Finally, we obtain the following expression for the species B in terms of the species of rooted trees.

Proposition 1. The species $B = \mathcal{A}^{\rightarrow}$ of oriented-edge-rooted k-gonal 2-trees satisfies

$$B = \sqrt[k-1]{\frac{A((k-1)X)}{(k-1)X}}. \tag{8}$$

Corollary 1. The numbers $a_n^{\rightarrow}$, $a_{n_1,n_2,\ldots}^{\rightarrow}$, and $b_n = \widetilde{a}_n^{\rightarrow}$ of k-gonal 2-trees pointed at an oriented edge and having n k-gons, respectively labelled, fixed by a permutation of cycle type $1^{n_1}2^{n_2}\ldots$ and unlabelled, satisfy the following formulas and recurrence:

$$a_n^{\rightarrow} = ((k-1)n+1)^{n-1} = m^{n-1}, \tag{9}$$

where $m = (k-1)n+1$ is the number of edges,

$$a_{n_1,n_2,\ldots}^{\rightarrow} = \prod_{i=1}^{\infty}(1+(k-1)\sum_{d|i} dn_d)^{n_i-1}(1+(k-1)\sum_{d|i,d<i} dn_d), \tag{10}$$

and

$$b_n = \frac{1}{n}\sum_{1\le j\le n}\sum_{\alpha}(|\alpha|+1)b_{\alpha_1}b_{\alpha_2}\ldots b_{\alpha_{k-1}}b_{n-j}, \qquad b_0 = 1, \tag{11}$$

the last sum is running over $(k-1)$-tuples of integers $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{k-1})$ such that $|\alpha|+1$ divides the integer j, where $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_{k-1}$.

Proof. Formulas (9) and (10) are obtained by specializing with $\mu = (k-1)^{-1}$ the following formulas, given by Fowler et al. in [5, 6],

$$\left(\frac{A(x)}{x}\right)^{\mu} = \sum_{n\ge 0}\mu(\mu+n)^{n-1}\frac{x^n}{n!}, \tag{12}$$

$Z_{\left(\frac{A(X/\mu)}{X/\mu}\right)^{\mu}} =$

$$\sum_{n_1,n_2,\ldots}\frac{x_1^{n_1}x_2^{n_2}\ldots}{1^{n_1}n_1!2^{n_2}n_2!\ldots}\prod_{i=1}^{\infty}(1+\frac{1}{\mu}\sum_{d|i}dn_d)^{n_i-1}(1+\frac{1}{\mu}\sum_{d|i,d<i}dn_d). \tag{13}$$

Formula (9) can also be established by a Prüfer-like bijection. To obtain the recurrence (11), it suffices to take the logarithmic derivative of the equation

$$\widetilde{B}(x) = \exp\left(\sum_{i\ge 1}\frac{x^i\widetilde{B}^{k-1}(x^i)}{i}\right), \tag{14}$$

where $\widetilde{B}(x) = \sum_{n\ge 0} b_n x^n$, which follows from relation (5), using (4). ∎

It is interesting to note that the sequences $\{b_n\}_{n\in\mathbb{N}}$, for $k = 2, 3, 4, 5$, are listed in the encyclopedia of integer sequences [18] and the equation (5), in the encyclopedia of combinatorial structures [9]. Also remark that, for each $n \geq 1$, b_n is a polynomial in k of degree $n-1$. This follows from (10) and the following explicit formula for b_n,

$$b_n = \sum_{n_1+2n_2+\cdots=n} \frac{a^{\rightarrow}_{n_1,n_2,\ldots}}{1^{n_1}n_1!2^{n_2}n_2!\ldots}, \tag{15}$$

which is a consequence of Burnside's lemma. The asymptotic behavior of the numbers b_n as $n \to \infty$, is studied, in particular as a function of k, in Section 7.

Remark 1. Equation (8) can also be used to compute the molecular expansion of the species B from the molecular expansion of A, using the binomial theorem. See [1] for more details.

3 Oriented case

We begin by determining relations for the pointed species appearing in the dissymmetry theorem. These relations are quite direct and the proof is left to the reader.

Proposition 2. The species a_o^-, $a_o^\diamond$, and $a_o^{\overline{\diamond}}$ are characterized by the following isomorphisms:

$$a_o^- = B, \quad a_o^\diamond = XC_k(B), \quad a_o^{\overline{\diamond}} = XB^k, \tag{16}$$

where $B = a^{\rightarrow}$ and C_k represents the species of oriented cycles of length k.

The dissymmetry theorem permits us to express the ordinary generating series $\widetilde{a}_o(x)$ of unlabelled oriented k-gonal 2-trees in terms of the corresponding series for the rooted species:

$$\widetilde{a}_o(x) = \widetilde{a}_o^-(x) + \widetilde{a}_o^\diamond(x) - \widetilde{a}_o^{\overline{\diamond}}(x). \tag{17}$$

By Proposition 2, we can then express $\widetilde{a}_o(x)$ as function of $\widetilde{B}(x) = \widetilde{a}^{\rightarrow}(x)$.

Proposition 3. The ordinary generating series $\widetilde{a}_o(x)$ of unlabelled oriented k-gonal 2-trees is given by

$$\widetilde{a}_o(x) = \widetilde{B}(x) + \frac{x}{k}\sum_{\substack{d|k\\ d>1}} \phi(d)\widetilde{B}^{\frac{k}{d}}(x^d) - \frac{k-1}{k}x\widetilde{B}^k(x). \tag{18}$$

Corollary 2. The numbers $a_{o,n}$ and $\widetilde{a}_{o,n}$ of oriented k-gonal 2-trees labelled and unlabelled, over n k-gons, respectively, are given by

$$a_{o,n} = ((k-1)n+1)^{n-2} = m^{n-2}, \quad n \geq 2, \tag{19}$$

$$\widetilde{a}_{o,n} = b_n - \frac{k-1}{k}b^{(k)}_{n-1} + \frac{1}{k}\sum_{\substack{d|k\\ d>1}} \phi(d)b^{(\frac{k}{d})}_{\frac{n-1}{d}}, \tag{20}$$

where

$$b_i^{(j)} \;=\; [x^i]\widetilde{B}^j(x) \;=\; \sum_{i_1+\cdots+i_j=i} b_{i_1} b_{i_2} \dots b_{i_j},$$

denotes the coefficient of x^i in the series $\widetilde{B}^j(x)$, with $b_r^{(j)} = 0$ if r is non-integral or negative.

Proof. For the labelled case, it suffices to remark that $a_n^{\rightarrow} = m a_{o,n}$. In the unlabelled case, equation (20) is directly obtained from (18). ∎

4 Unoriented case

In the unoriented case, the number a_n of k-gonal 2-trees labelled over n polygons satisfies $2a_n = a_{o,n}+1$, since the only k-gonal 2-tree left fixed by a reversal of the orientation, for a given number of polygons, is the one in which every polygon share one common edge. We get

Proposition 4. The number a_n of labelled k-gonal 2-trees on n k-gons is given by

$$a_n = \frac{1}{2}\left(m^{n-2}+1\right), \qquad n \geq 2, \tag{21}$$

where $m = (k-1)n+1$.

For the unlabelled enumeration of k-gonal 2-trees (unoriented), we have to consider quotient species of the form $F/\mathbb{Z}_2$, where F is any species of "oriented" structures and $\mathbb{Z}_2 = \{1, \tau\}$, is the group where the action of τ is to reverse the orientation of the structure. A structure of such a species then consists in an orbit $\{s, \tau \cdot s\}$ of F-structures under the action of $\mathbb{Z}_2$.

For instance, the different pointed species of unoriented k-gonal 2-trees a^-, $a^\diamond$ and $a^{\underline{\diamond}}$, can be expressed as quotient species of the corresponding species of oriented k-gonal 2-trees:

$$a^- = \frac{a^{\rightarrow}}{\mathbb{Z}_2}, \quad a^\diamond = \frac{a_o^\diamond}{\mathbb{Z}_2} \;=\; \frac{XC_k(B)}{\mathbb{Z}_2}, \quad a^{\underline{\diamond}} = \frac{a_o^{\underline{\diamond}}}{\mathbb{Z}_2} \;=\; \frac{XB^k}{\mathbb{Z}_2}. \tag{22}$$

For the ordinary generating series (unlabelled structures) associated to such quotient species, we use the following formula, which is quite obvious,

$$(F/\mathbb{Z}_2)^\sim(x) = \frac{1}{2}(\widetilde{F}(x) + \widetilde{F}_\tau(x)), \tag{23}$$

where $\widetilde{F}_\tau(x) = \sum_{n\geq 0} |\mathrm{Fix}_{\widetilde{F}_n}(\tau)| x^n$ is the ordinary generating series of unlabelled F-structures left fixed by the action of τ, that is, by orientation reversal. However, the computation of the series $\widetilde{F}_\tau(x)$ is quite complicated and it is better to treat separately two cases according to the parity of k.

4.1 Case k odd

We can notice, observing Figures 3 a) and b), that in every k-gon containing the pointed (but not oriented) edge of an $\mathcal{A}^{-}$-structure, it is possible to orient the $k-1$ other edges in a "going away (from the root edge) direction" as in Figure 3 a), when k is odd, but there remains an ambiguous edge if k is even. This phenomenon permits us to introduce *skeleton* species, when k is odd, in analogy with the approach of Fowler et al. in [5, 6], where $k=3$. They are the two-sort quotient species $Q(X,Y)$, $S(X,Y)$ and $U(X,Y)$, where X represents the species of k-gons and Y the species of oriented edges, defined by Figures 4 a), b) and c), where $k=5$. In analogy with the case $k=3$, we get the following

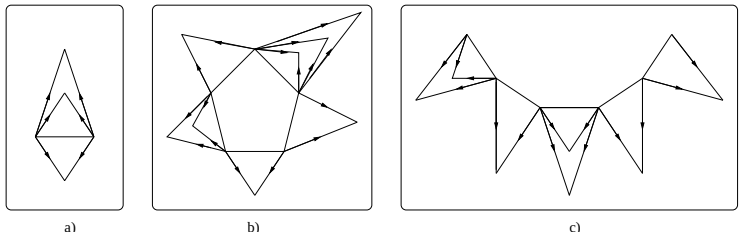


Figure 4: Skeleton species a) $Q(X,Y)$, b) $S(X,Y)$ and c) $U(X,Y)$

propositions.

Proposition 5. The skeleton species Q, S and U admit the following expressions in terms of quotients species

$$\begin{aligned} Q(X,Y) &= E(XY^2)/\mathbb{Z}_2, && (24)\\ S(X,Y) &= C_k(E(XY^2))/\mathbb{Z}_2, && (25)\\ U(X,Y) &= (E(XY^2))^k/\mathbb{Z}_2. && (26) \end{aligned}$$

Proposition 6. For k odd, $k \geq 3$, we have the following expressions for the pointed species of k-gonal 2-trees, where $B = \mathcal{A}^{\rightarrow}$:

$$\mathcal{A}^{-} = Q(X, B^{\frac{k-1}{2}}), \quad \mathcal{A}^{\circ} = X \cdot S(X, B^{\frac{k-1}{2}}), \quad \mathcal{A}^{\diamond} = X \cdot U(X, B^{\frac{k-1}{2}}). \quad (27)$$

In order to obtain enumeration formulas, we have first to compute the cycle index series of the species Q, S and U.

Proposition 7. The cycle index series of the species $Q(X,Y)$, $S(X,Y)$ and $U(X,Y)$ are given by

$$\begin{aligned} Z_Q &= \frac{1}{2}\left(Z_{E(XY^2)} + q\right), && (28)\\ Z_S &= \frac{1}{2}\left(Z_{C_k(E(XY^2))} + q \cdot (p_2 \circ Z_{E(XY^2)})^{\frac{k-1}{2}}\right), && (29)\\ Z_U &= \frac{1}{2}\left(Z_{(E(XY^2))^k} + q \cdot (p_2 \circ Z_{E(XY^2)})^{\frac{k-1}{2}}\right), && (30) \end{aligned}$$

where $q = h \circ (x_1 y_2 + p_2 \circ (x_1 \frac{y_1^2 - y_2}{2}))$, p_2 represents the power sum function of degree two, h the homogeneous symmetric function and $\circ$, the plethystic substitution.

Proof. Formula (28) and the method used can be found in [5, 6]. It is a matter of counting colored unlabelled $F(X,Y)$-structures left fixed by τ. In the case of S, we have to leave fixed a colored $C_k(E(XY^2))$-structure. For this, the basis cycle of length k must possess (at least) one symmetry axis passing through the middle of one of its sides. We can see that when such a structure has several axis of symmetry, the choice of the axis is arbitrary. On both sides of the axis, each colored $E(XY^2)$-structure must have its mirror image; this contributes for a term of $(p_2 \circ Z_{E(XY^2)})^{\frac{k-1}{2}}$. Next, the attached structure on the distinguished edge must be globally left fixed; this gives the factor q. The reasoning is very similar for the species U ■

Combining the dissymmetry theorem, equations (28), (29), (30) and the substitution rules of unlabelled enumeration, we obtain the ordinary generating series of the species of k-gonal 2-trees.

Proposition 8. Let $k \geq 3$ be an odd integer. The ordinary generating series $\widetilde{a}(x)$ of unlabelled k-gonal 2-trees is given by

$$\widetilde{a}(x) = \frac{1}{2}\left(\widetilde{a}_o(x) + \exp\Big(\sum_{i\geq 1} \frac{1}{2i}(2x^i \widetilde{B}^{\frac{k-1}{2}}(x^{2i}) + x^{2i}\widetilde{B}^{k-1}(x^{2i}) - x^{2i}\widetilde{B}^{\frac{k-1}{2}}(x^{4i}))\Big)\right). \tag{31}$$

Corollary 3. For $k \geq 3$, odd, the number $\widetilde{a}_n$ of unlabelled k-gonal 2-trees over n k-gons, satisfy the following recurrence

$$\widetilde{a}_n = \frac{1}{2n}\sum_{j=1}^{n}\Big(\sum_{l|j} l\omega_l\Big)\Big(\widetilde{a}_{n-j} - \frac{1}{2}\widetilde{a}_{o,n-j}\Big) + \frac{1}{2}\widetilde{a}_{o,n}, \quad \widetilde{a}_0 = 1, \tag{32}$$

where, for all $n \geq 1$,

$$\omega_n = 2b_{\frac{n-1}{2}}^{(\frac{k-1}{2})} + b_{\frac{n-2}{2}}^{(k-1)} - b_{\frac{n-2}{4}}^{(\frac{k-1}{2})}, \tag{33}$$

and $b_i^{(j)}$ is defined in Corollary 2.

4.2 Case k even

The case k even is much more delicate. In order to express the ordinary generating functions of the three species a^-, $a^\diamond$ and $a^{\hat{\diamond}}$, we apply relation (23) to formulas (22). For the species a^-, we have

$$\widetilde{a}^-(x) = \frac{1}{2}(\widetilde{a}^{\rightarrow}(x) + \widetilde{a}_\tau^{\rightarrow}(x)), \tag{34}$$

where $\widetilde{a}_{\tau}^{\rightarrow}(x) = \sum_{n\geq 0} |\mathrm{Fix}\, \widetilde{a}_n^{\rightarrow}(\tau)| x^n$ is the ordinary generating series of unlabelled oriented-edge-rooted 2-trees which are left fixed by reversing the orientation. Let a_{S} denotes the subspecies of $a^{\rightarrow}$ consisting of $a^{\rightarrow}$-structures s which are isomorphic to their image $\tau \cdot s$ under the orientation reversing map. We have to compute $\widetilde{a}_{\mathrm{S}}(x) = \widetilde{a}_{\tau}^{\rightarrow}(x)$. For this, let us introduce some auxiliary species. The first one, denoted a_{TS}, is the class of a_{S}-structures for which every page attached to the rooted edge is vertically symmetric without crossed symmetries (see below); we say *totally symmetric*. We can characterize this species by the

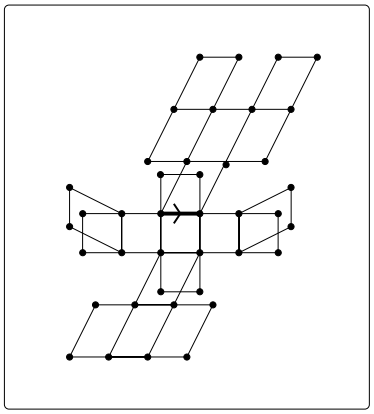


Figure 5: A structure of the species a_{TS}

following functional equation (see Figure 5)

$$a_{\mathrm{TS}} = E(X \cdot X_{=}^2 < B^{\frac{k-2}{2}} > \cdot a_{\mathrm{TS}}) = E(P_{\mathrm{TS}}), \tag{35}$$

where $X_{=}^2 < F >$ represents the species of ordered pairs of isomorphic F-structures and P_{TS} is the species of *totally symmetric pages*. Translating this equation in terms of generating series, we get

$$\widetilde{a}_{\mathrm{TS}}(x) = \exp\left(\sum_{i\geq 1} \frac{1}{i} x^i \widetilde{B}^{\frac{k-2}{2}}(x^{2i}) \widetilde{a}_{\mathrm{TS}}(x^i)\right). \tag{36}$$

Proposition 9. The numbers $\beta_n = |\widetilde{a}_{\mathrm{TS}}[n]|$ of unlabelled a_{TS}-structures on n polygons satisfy the recurrence

$$\beta_n = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{d|i} d\pi_d\right) \beta_{n-i}, \ n \geq 1 \quad \beta_0 = 1, \tag{37}$$

where

$$\pi_n = \widetilde{P}_{\mathrm{TS},n} = \sum_{\substack{i+j=n-1 \\ i \ even}} b_{\frac{i}{2}}^{(\frac{k-2}{2})} \beta_j. \tag{38}$$

Proof. It suffices to take the logarithmic derivative of (36), that is

$$x\frac{(\widetilde{a}_{\mathrm{TS}})'(x)}{\widetilde{a}_{\mathrm{TS}}(x)} = x\cdot\sum_{i\geq 1}\Omega'(x^i)x^{i-1}, \tag{39}$$

where $\Omega(x)=\sum_{n\geq 1}\omega_n x^n = x\widetilde{B}^{\frac{k-2}{2}}(x^2)\widetilde{a}_{\mathrm{TS}}(x)$. Next, extracting the coefficient of x^n in both sides of

$$x(\widetilde{a}_{\mathrm{TS}})'(x) = \Big(\sum_{i\geq 1}\Omega'(x^i)x^i\Big)\widetilde{a}_{\mathrm{TS}}(x) \tag{40}$$

leads to (37) using (35) since $\Omega(x)=\widetilde{P}_{\mathrm{TS}}(x)$. ■

Let us now introduce two other species, namely P_{AL} and P_{M}, of *pairs of alternated pages* and of *mixed pages*. A pair of *alternated pages* is, by definition, an unordered pair of oriented pages ($a^{\rightarrow}$-structures having only one page) of the form $\{s,\tau\cdot s\}$ with s and $\tau\cdot s$ non-isomorphic. Figure 6 a) shows a structure belonging to this species. A *mixed page* is a symmetric page having at least one alternated symmetry. Such a structure is drawn in Figure 6 b). We can then express each of these two species in terms of the other, as follows:

$$P_{\mathrm{AL}} = \Phi_2 < XB^{k-1} - (P_{\mathrm{TS}}+P_{\mathrm{M}}) >, \tag{41}$$

$$P_{\mathrm{M}} = X\cdot X_{=}^2 < B^{\frac{k-2}{2}} > \cdot(a_{\mathrm{S}} - a_{\mathrm{TS}}), \tag{42}$$

where $\Phi_2 < F >$ represents the species of pairs of F-structures of the form $\{s,\tau\cdot s\}$ and E_+ is the species of non empty sets. At the level of ordinary generating series, we get

$$\widetilde{P}_{\mathrm{AL}}(x) = \frac{1}{2}\big(x^2\widetilde{B}^{k-1}(x^2) - \widetilde{P}_{\mathrm{TS}}(x^2) - \widetilde{P}_{\mathrm{M}}(x^2)\big), \tag{43}$$

$$\widetilde{P}_{\mathrm{M}}(x) = \Big(XX_{=}^2 < B^{\frac{k-2}{2}} > \cdot a_{\mathrm{TS}}\cdot E_+(P_{\mathrm{AL}}+P_{\mathrm{M}})\Big)^{\sim}(x) \tag{44}$$

$$= x\widetilde{B}^{\frac{k-2}{2}}(x^2)\widetilde{a}_{\mathrm{TS}}(x)\left(\exp\left(\sum_{i\geq 1}\frac{1}{i}(\widetilde{P}_{\mathrm{AL}}(x^i)+\widetilde{P}_{\mathrm{M}}(x^i))\right)-1\right) \tag{45}$$

$$= x\widetilde{B}^{\frac{k-2}{2}}(x^2)(\widetilde{a}_{\mathrm{S}}(x) - \widetilde{a}_{\mathrm{TS}}(x)) \tag{46}$$

Let $\widetilde{a}_{\mathrm{S}}(x)$ denote the ordinary generating series of unlabelled symmetric $a^{\rightarrow}$-structures. We have (see Figure 7)

$$\widetilde{a}_{\mathrm{S}}(x) = E(P_{\mathrm{TS}}+P_{\mathrm{AL}}+P_{\mathrm{M}})^{\sim}(x), \tag{47}$$

$$= \exp\left(\sum_{i\geq 1}\frac{1}{i}(\widetilde{P}_{\mathrm{TS}}(x^i)+\widetilde{P}_{\mathrm{AL}}(x^i)+\widetilde{P}_{\mathrm{M}}(x^i))\right). \tag{48}$$

We then deduce a recurrence for the numbers $\alpha_n = \widetilde{a}_{\mathrm{S},n}$ of symmetric k-gonal 2-trees rooted at an edge left fixed by orientation reversing, $\widetilde{P}_{\mathrm{AL},n}$ and

$\widetilde{P}_{\mathrm{M},n}$ of alternated and mixed pages, respectively, on n k-gons:

$$\alpha_n = \frac{1}{n}\sum_{i=1}^{n}\Big(\sum_{d|i} d\omega_d\Big)\alpha_{n-i}, \quad \alpha_0 = 1, \tag{49}$$

$$\widetilde{P}_{\mathrm{M},n} = \sum_{i=0}^{n-1} b_{\frac{i}{2}}^{(\frac{k-2}{2})}\alpha_{n-1-i} - \widetilde{P}_{\mathrm{TS},n}, \tag{50}$$

$$\widetilde{P}_{\mathrm{AL},n} = \frac{1}{2}\left(b_{\frac{n-2}{2}}^{(k-1)} - \widetilde{P}_{\mathrm{TS},n/2} - \widetilde{P}_{\mathrm{M},n/2}\right), \tag{51}$$

where

$$\omega_k = \widetilde{P}_{\mathrm{TS},k} + \widetilde{P}_{\mathrm{AL},k} + \widetilde{P}_{\mathrm{M},k},$$

and $\widetilde{P}_{\mathrm{TS},n} = \pi_n$ is given by (38).

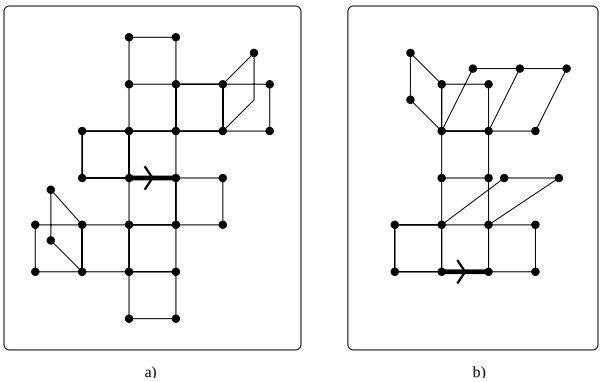


Figure 6: A pair of alternated pages and a mixed page

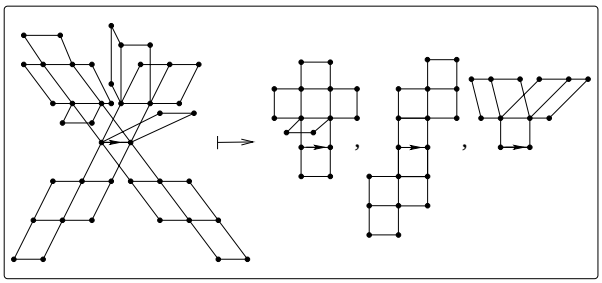


Figure 7: Decomposition of an $\widetilde{a}^{\rightarrow}$-structure fixed under τ

Proposition 10. If k is an even integer, then the number of edge rooted (unoriented) k-gonal 2-trees over n k-gons is given by

$$\widetilde{a}_n^{-} = \frac{1}{2}(b_n + \alpha_n). \tag{52}$$

Let us now turn to the species $\mathcal{A}^{\underline{\diamond}}$ of k-gonal 2-trees rooted at an edge-pointed k-gon.

Proposition 11. We have

$$\widetilde{a}^{\underline{\diamond}}(x) = \frac{1}{2}\left(\widetilde{a}_o^{\underline{\diamond}}(x) + \widetilde{a}_{o,\tau}^{\underline{\diamond}}(x)\right), \tag{53}$$

where

$$\widetilde{a}_{o,\tau}^{\underline{\diamond}}(x) = x\widetilde{a}_{\mathrm{S}}^2(x)\widetilde{B}^{\frac{k-2}{2}}(x^2).$$

Proof. An unlabelled τ-symmetric $\mathcal{A}_o^{\underline{\diamond}}$-structure possesses an axis of symmetry which is, in fact, the mediatrix of the distinguished edge of the rooted polygon, and also the mediatrix of the edge facing the rooted one, see Figure 8. The two structures s and t glued on these two edges are thus symmetric, which leads to the term $(\widetilde{a}_{\mathrm{S}}(x))^2$. Then, on each side of the axis, are found two $B^{\frac{k-2}{2}}$-structures α and β, which by symmetry satisfy $\beta = \tau \cdot \alpha$, contributing to the factor $\widetilde{B}^{\frac{k-2}{2}}(x^2)$. ■

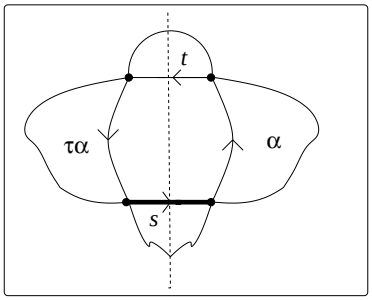


Figure 8: A τ-symmetric unlabelled $\mathcal{A}_o^{\underline{\diamond}}$-structures

Corollary 4. We have the following expression for the number $\widetilde{a}_n^{\underline{\diamond}}$ of unlabelled $\mathcal{A}^{\underline{\diamond}}$-structures,

$$\widetilde{a}_n^{\underline{\diamond}} = \frac{1}{2}\left(\widetilde{a}_{o,n}^{\underline{\diamond}} + \sum_{i+j=n-1} \alpha_i^{(2)} \cdot b_{\frac{j}{2}}^{(\frac{k-2}{2})}\right), \tag{54}$$

where $\alpha_i^{(2)} = [x^i]\widetilde{a}_S^2(x)$. □

We proceed in a similar way for the species $\mathcal{A}^{\diamond}$, of k-gon rooted k-gonal 2-trees. Once again, we use relation (23), giving

$$\widetilde{a}^{\diamond}(x) = \frac{1}{2}\left(\widetilde{a}_o^{\diamond}(x) + \widetilde{a}_{o,\tau}^{\diamond}(x)\right). \tag{55}$$

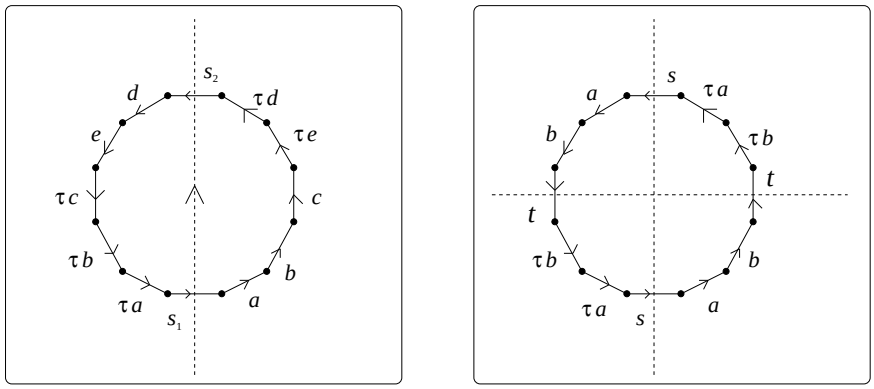
Proposition 12. Let $\widetilde{a}^{\diamond}_{o,\tau}(x)$ be the generating series of unlabelled $a^{\diamond}_{o}$-structures left fixed by orientation reversing. Then, we have

$$\widetilde{a}^{\diamond}_{o,\tau}(x) = \frac{x}{2}\widetilde{a}^{2}_{\mathrm{S}}(x)\widetilde{B}^{\frac{k-2}{2}}(x^2) + \frac{x}{2}\widetilde{B}^{\frac{k}{2}}(x^2). \tag{56}$$

Proof. Notice first that to be left fixed by orientation reversing, an $a^{\diamond}_{o}$-structure must admit at least one axis of symmetry, which can be of two kinds:

1. an axis passing through the middle of two opposite edges, or
2. an axis passing through two opposite vertices,

of the pointed polygon. The enumeration is carried out by first orienting the axis of symmetry. The first term of (56) then corresponds to a symmetry of the first kind, and the second term to a symmetry of the second kind. The structures having both symmetries are precisely those which are counted one half time in both of these terms. This is established for a general k by considering the largest power of 2, 2^m, such that $k/2^m$ is odd. We illustrate the proof in the following lines with $k = 12$; the reader will easily convince himself of the validity of this argument for any k.



a) b)

Figure 9: $\widetilde{a}^{\diamond}_{o,\tau}$-structures with an edge–edge symmetry

For $k = 12$, a general unlabelled τ-symmetric polygon-rooted oriented k-gonal 2-tree with an oriented edge–edge axis will be of the form illustrated in Figure 9 a), where s_1 and s_2 represent unlabelled a_{S}-structures, a, b, c, d and e are general unlabelled B-structures and τx represents the opposite of the B-structures x, obtained by reversing their orientation. Most of these structures are enumerated exactly by $\frac{1}{2}x\widetilde{a}^{2}_{\mathrm{S}}(x)\widetilde{B}^{5}(x^2)$. Indeed, the factor $x\widetilde{a}^{2}_{\mathrm{S}}(x)\widetilde{B}^{5}(x^2)$ is obtained in the same way as for $a^{\diamond}_{o,\tau}$-structures and the division by two is justified in the following cases:

1. $s_1 \neq s_2$ (two orientations of the axis),

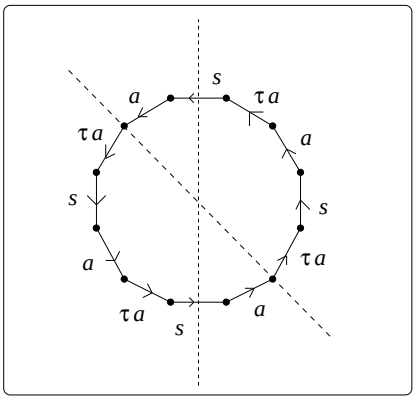


Figure 10: $\widetilde{a}^{\diamond}_{o,\tau}$-structures with edge–edge and vertex–vertex symmetries

2. $s_1 = s_2 = s$, $(a, b, c) \neq (d, e, \tau \cdot c)$ (two orientations),

3. $s_1 = s_2 = s$, $(a, b, c) = (d, e, \tau \cdot c)$, so that $c = \tau \cdot c = t \in \widetilde{a}_{\mathrm{S}}$, and either $i)$ $s \neq t$ or $ii)$ $s = t$ and $(a, b) \neq (\tau \cdot b, \tau \cdot a)$ (two choices for the symmetry axis, see Figure 9 b)),

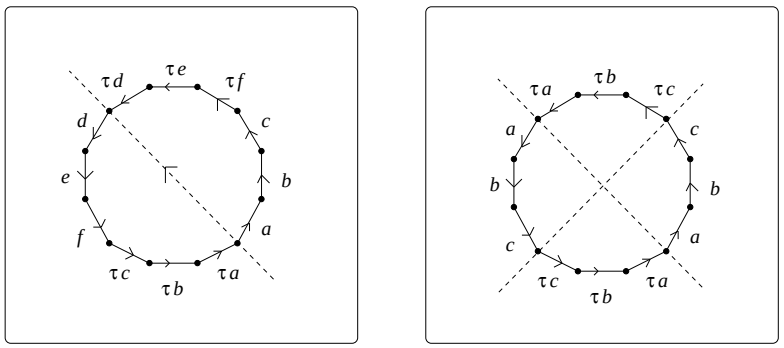


Figure 11: $\widetilde{a}^{\diamond}_{o,\tau}$-structure with a vertex–vertex symmetry axis

However, the structures with $s = t$ and $b = \tau \cdot a$ (see Figure 10) will occur only once and are counted only one half time in the formula. But, notice that these structures also admit a vertex–vertex symmetry axis and, as it will turn out, are also counted one half time in the second term of (56).

Similarly, an unlabelled $a^{\diamond}_{o,\tau}$-structure with an oriented vertex–vertex symmetry axis will be of the form illustrated in Figure 11 a), where a, b, ..., f are arbitrary unlabelled B-structures. Most of these terms are enumerated exactly by $\frac{1}{2}x\widetilde{B}^6(x^2)$, the division by two being justified in the following cases:

1. $(a,b,c) \neq (d,e,f)$ (two orientations of the symmetry axis),

2. $(a,b,c) = (d,e,f)$ and $(a,b,c) \neq (\tau\cdot c, \tau\cdot b, \tau\cdot a)$ (two choices for the symmetry axis, see Figure 11 b)),

However, the structures with $(a,b,c) = (d,e,f)$, $c = \tau\cdot a$ and $b = \tau\cdot b = s \in \widetilde{\mathcal{A}}_{\mathrm{S}}$ appear only once and are counted one half time here. But they also have an edge-edge symmetry axis and were also counted one half time in the first term of (56) (exchange a and $\tau\cdot a$ in Figure 10). ■

The dissymmetry theorem yields, for $k \geq 4$ even,

$$\widetilde{a}(x) = \frac{1}{2}\widetilde{a}_o(x) + \frac{1}{2}\widetilde{a}_{\mathrm{S}}(x) + \frac{1}{2}\widetilde{a}^{\diamond}_{o,\tau}(x) - \frac{1}{2}\widetilde{a}^{\diamond}_{\overline{o,\tau}}(x), \tag{57}$$

So, we have the following result.

Proposition 13. Let k be an even integer, $k \geq 4$. Then,the generating series $\widetilde{a}(x)$ of unlabelled k-gonal 2-trees is given by

$$\widetilde{a}(x) = \frac{1}{2}\widetilde{a}_o(x) + \frac{1}{2}\widetilde{a}_{\mathrm{S}}(x) + \frac{x}{4}(\widetilde{B}^{\frac{k}{2}}(x^2) - \widetilde{a}^2_{\mathrm{S}}(x)\widetilde{B}^{\frac{k-2}{2}}(x^2)), \tag{58}$$

where $\widetilde{a}_o(x)$ is given by (18) and $\widetilde{a}_{\mathrm{S}}(x)$ by (48). □

Corollary 5. If $k \geq 4$, is an even integer, then the number of unlabelled k-gonal 2-trees over n k-gons is given by

$$\widetilde{a}_n = \frac{1}{2}\widetilde{a}_{o,n} + \frac{1}{2}\alpha_n + \frac{1}{4}b^{(\frac{k}{2})}_{\frac{n-1}{2}} - \frac{1}{4}\sum_{i+j=n-1}\alpha^{(2)}_i \cdot b^{(\frac{k-2}{2})}_{\frac{j}{2}}, \tag{59}$$

with

$$b^{(m)}_l = [x^l]\widetilde{B}^m(x), \quad \alpha^{(2)}_i = [x^i]\widetilde{a}^2_{\mathrm{S}}(x).$$

5 Enumeration according to the perimeter

In this section, we are interested in the enumeration of k-gonal 2-trees according to the perimeter. The *perimeter* of a k-gonal 2-tree is the number of external edges (edges of degree at most one). In particular, if the structure s is the single edge, the perimeter is 1. In order to keep track of the perimeter, we introduce a weight function w over k-gonal 2-tree, defined by:

$$\begin{array}{rccc} w: & \mathcal{A} & \longrightarrow & \mathbb{Q}[t] \\ & s & \longmapsto & w(s) = t^{\mathrm{p}(s)}, \end{array} \tag{60}$$

where $\mathrm{p}(s)$ denotes the perimeter of the structure $s \in \mathcal{A}$. For example, the 2-tree of Figure 1 a) has perimeter 28.

5.1 A weighted version of the species B

Our first task is to determine the functional equation satisfied by the species B_w of k-gonal 2-trees pointed at an oriented edge and weighted by the perimeter counter t, with the precision that the rooted edge does not contribute to the perimeter of a B-structure except in the case of a single edge, which has perimeter 1. We have

Proposition 14. The weighted species B_w is characterized by the following functional equation

$$B_w(X) = t + E_+(XB_w^{k-1}(X)), \tag{61}$$

where E_+ is the species of non-empty sets.

Proof. The (unweighted) species B satisfies

$$B = E(XB^{k-1}) = 1 + E_+(XB^{k-1}(X)),$$

where the term 1 corresponds to the single edge. By taking into account the perimeter weight w and the fact that a single edge has weight t, we obtain (61). ■

Note that (61) is also valid for $k = 2$. The species B_w then represents weighted edge-labelled (ordinary) rooted trees where the variables t acts as a leaf counter.

We write the generating series associated to the weighted species B_w as follows:

$$B_w(x) = B(x,t) = \sum_{\substack{n\geq 0\\ \ell\geq 1}} a^{\rightarrow}_{n,\ell}\, t^\ell \frac{x^n}{n!}, = \sum_{n\geq 0} a^{\rightarrow}_n(t)\frac{x^n}{n!} \tag{62}$$

$$\widetilde{B}_w(x) = \widetilde{B}(x,t) = \sum_{\substack{n\geq 0\\ \ell\geq 1}} b_{n,\ell}\, t^\ell x^n = \sum_{n\geq 0} b_n(t)x^n, \tag{63}$$

where $a^{\rightarrow}_{n,\ell}$ and $b_{n,\ell}$ are the numbers of labelled and unlabelled k-gonal 2-trees rooted at an oriented edge having n k-gons and perimeter ℓ. From equation (61), we can deduce explicit formulas for $a^{\rightarrow}_n(t)$ and $a^{\rightarrow}_{n,\ell}$ and recursive formulas for $b_n(t)$ and $b_{n,\ell}$. Notice that, because of the nature of the structures, the integer ℓ is bounded: $(k-2)n + 1 \leq \ell \leq (k-1)n$.

Proposition 15. The polynomial $a^{\rightarrow}_n(t)$, giving the labelled weighted enumeration of B_w-structures over n k-gons is given by $a^{\rightarrow}_0(t) = t$ and, for $n \geq 1$,

$$a^{\rightarrow}_n(t) = \frac{n!}{m}\sum_{\ell=m-n}^{m-1}\ \sum_{i+j=m-\ell} (-1)^j i^n \binom{m}{\ell, i, j} t^\ell, \tag{64}$$

$$= \frac{1}{m}\sum_{i=1}^{n}\frac{m!}{(m-i)!}S(n,i)t^{m-i}, \tag{65}$$

where $m = (k-1)n+1$ is the number of edges and $S(n,j)$ denotes the Stirling numbers of the second kind, giving the number of partitions of an n-set in j blocks.

Proof. From (61), we have $B(x,t) = t + \exp(xB^{k-1}(x,t)) - 1$. So, we get

$$xB^{k-1}(x,t) = x(t + \exp(xB^{k-1}(x,t)) - 1)^{k-1}.$$

Putting $\mathcal{B}(x,t) = xB^{k-1}(x,t)$, we obtain that the series $\mathcal{B}(x,t)$ satisfies the functional equation $\mathcal{B}(x,t) = xR(\mathcal{B}(x,t))$, where $R(y) = (t + \exp(y) - 1)^{k-1}$. Moreover,

$$B(x,t) = \left(\frac{\mathcal{B}(x,t)}{x}\right)^{\frac{1}{k-1}}. \tag{66}$$

The composite form of Lagrange inversion applied to equation (66) gives (64). To obtain now (65), we apply the same method but we use the following well-known relation

$$\frac{(\mathrm{e}^x - 1)^j}{j!} = \sum_{n \geq j} S(n,j)\frac{x^n}{n!},$$

see [4] page 63. ■

We obtain now, in a straightforward way, expressions for $a^{\rightarrow}_{n,\ell}$. Formula (65) can also be given a Prüfer-type bijective proof.

Corollary 6. The number $a^{\rightarrow}_{n,\ell}$ of labelled B_w-structures over n k-gons and having perimeter ℓ, for $(k-2)n+1 \leq \ell \leq (k-1)n$ (a weight t^ℓ), is given by

$$a^{\rightarrow}_{n,\ell} = \frac{n!}{m} \sum_{i+j=m-\ell} (-1)^j i^n \binom{m}{\ell, i, j}, \tag{67}$$

$$= \frac{(m-1)!}{\ell!} S(n, m-\ell), \tag{68}$$

where $m = (k-1)n+1$ is the number of edges. □

We notice that, when $k = 3$, $\ell = n+1$ is the minimal perimeter and $a^{\rightarrow}_{n,n+1} = n!\mathbf{c_n}$, where $\mathbf{c_n}$ is the famous Catalan number, since, in this case, the B_w-structures obtained are outerplanar, see Labelle et al. [14]. These structures are the basic ones in the computation of the molecular expansion (a classification according to symmetries) of the species of outerplanar k-gonal 2-trees. For general k, $a^{\rightarrow}_{n,(k-2)n+1} = n!C_{k,n}$, where $C_{k,n} = \frac{1}{n}\binom{n(k-1)}{n-1}$ is the generalized Catalan numbers. See [16].

As in the unweighted case, we cannot obtain an explicit formula for the number $b_{n,\ell}$ as well as for the polynomial $b_n(t)$. However, we give recursive formulas.

Proposition 16. The polynomials $b_n(t)$, $n \geq 1$, satisfy the following recurrence

$$\begin{aligned} b_0(t) &= t, \\ b_n(t) &= \frac{1}{n}\left(\sum_{d|n} d \cdot b_{d-1}^{(k-1)}(t^{\frac{n}{d}}) + \sum_{i=1}^{n-1}\left(\sum_{d|i} d \cdot b_{d-1}^{(k-1)}(t^{\frac{i}{d}})\right) b_{n-i}(t)\right), \end{aligned} \tag{69}$$

where the summations are taken over integers $i, d \geq 1$, and where

$$b_n^{(k-1)}(t) = [x^n]\widetilde{B}^{k-1}(x,t) = \sum_{i_1+i_2+\cdots+i_{k-1}=n} b_{i_1}(t)b_{i_2}(t)\ldots b_{i_{k-1}}(t). \tag{70}$$

Proof. We obtain recurrence (69) by taking the derivative (with respect to x) of the following expression

$$\widetilde{B}(x,t) = t + \exp\left(\sum_{i\geq 1}\frac{1}{i}x^i\widetilde{B}^{k-1}(x^i,t^i)\right) - 1,$$

obtained from (61) by passing to the ordinary generating series for unlabelled enumeration. ■

We obtain the next proposition quite directly from the previous one.

Corollary 7. The number $b_{n,\ell}$ of unlabelled B_w-structures over n k-gons and having perimeter ℓ satisfies the following recurrence

$$b_{0,\ell} = \delta_{1,\ell}, \qquad b_{n,\ell} = \frac{1}{n}\omega_{n,\ell} + \frac{1}{n}\sum_{\substack{\nu+\mu=n \\ \nu,\mu\geq 1}}\sum_{\substack{p+q=\ell \\ p,q\geq 1}} \omega_{\nu,p}\cdot b_{\mu,q}, \tag{71}$$

where $\delta_{i,j}$ is the Kronecker symbol and

$$\omega_{n,\ell} = \sum_{d|(n,\ell)} \frac{n}{d} b_{\frac{n}{d}-1,\frac{\ell}{d}}^{(k-1)}. \tag{72}$$

□

As for the unweighted case, we can express the pointed weighted species of k-gonal 2-trees as function of the species B_w. We begin with the oriented case, which is simpler, and use it to obtain the unoriented case.

5.2 Oriented case

Let us denote by $a_w^- = (a_w)^-$, $a_w^\diamond = (a_w)^\diamond$, $a_w^{\diamond\!\!\diamond} = (a_w)^{\diamond\!\!\diamond}$, and $a_{o,w}^- = (a_{o,w})^-$, $a_{o,w}^\diamond = (a_{o,w})^\diamond$, $a_{o,w}^{\diamond\!\!\diamond} = (a_{o,w})^{\diamond\!\!\diamond}$, where w is defined by (60). Note in particular that $a_{o,w}^- \neq B_w$. The dissymetry theorem remains valid in this weighted context, for both the oriented and unoriented cases:

$$a_{o,w}^- + a_{o,w}^\diamond = a_{o,w} + a_{o,w}^{\diamond\!\!\diamond}, \tag{73}$$

$$a_w^- + a_w^\diamond = a_w + a_w^{\diamond\!\!\diamond}. \tag{74}$$

As in the unweighted case, we have to express these species in terms of the weighted species B_w. Enumeration formulas will then follow. The following proposition is quite obvious and the proof is omitted.

Proposition 17. The weighted species $\mathcal{A}^{-}_{o,w}$, $\mathcal{A}^{\diamond}_{o,w}$ and $\mathcal{A}^{\diamond}_{o,w}$ are characterized by

$$\begin{aligned}
\mathcal{A}^{-}_{o,w} &= B_w + (t-1)XB_w^{k-1}, && (75)\\
\mathcal{A}^{\diamond}_{o,w} &= XC_k(B_w), && (76)\\
\mathcal{A}^{\diamond}_{o,w} &= XB_w^k. && (77)
\end{aligned}$$

We then deduce easily the associated generating series of these species

$$\mathcal{A}^{-}_{o}(x,t) = B(x,t) + (t-1)xB^{k-1}(x,t) \tag{78}$$

and

$$\begin{aligned}
\widetilde{\mathcal{A}}^{-}_{o}(x,t) &= \widetilde{B}(x,t) + (t-1)x\widetilde{B}^{k-1}(x,t), && (79)\\
\widetilde{\mathcal{A}}^{\diamond}_{o}(x,t) &= \frac{x}{k}\sum_{d|k}\phi(d)\widetilde{B}^{\frac{k}{d}}(x^d,t^d), && (80)\\
\widetilde{\mathcal{A}}^{\diamond}_{o}(x,t) &= x(\widetilde{B}^{k}(x,t) + (t-1)\widetilde{B}^{k-1}(x,t)), && (81)
\end{aligned}$$

from which we deduce

$$a^{-}_{o,n}(t) = n![x^n]\mathcal{A}^{-}_{o}(x,t) = a^{\rightarrow}_{n}(t) + (t-1)na^{\rightarrow\,(k-1)}_{n-1}(t), \tag{82}$$

and, using the dissymmetry theorem,

$$\widetilde{\mathcal{A}}_{o}(x,t) = \widetilde{B}(x,t) + \frac{x}{k}\sum_{d|k}\phi(d)\widetilde{B}^{\frac{k}{d}}(x^d,t^d) - x\widetilde{B}^{k}(x,t) + (t-1)x\widetilde{B}^{k-1}(x,t). \tag{83}$$

We then get:

Proposition 18. We have, for $n \geq 2$,

$$\begin{aligned}
a_{o,n}(t) &= \frac{a^{-}_{o,n}(t)}{m}, && (84)\\
\widetilde{a}_{o,n}(t) &= [x^n]\widetilde{\mathcal{A}}_{o}(x,t) && (85)\\
&= b_n(t) - b^{(k)}_{n-1}(t) + \frac{1}{k}\sum_{\substack{d|k\\ d\geq 1}}\phi(d)b^{(\frac{k}{d})}_{\frac{n-1}{d}}(t^d) + (t-1)b^{(k-1)}_{n-1}(t), && (86)
\end{aligned}$$

where $m = (k-1)n+1$ is the number of edges and $b^{(i)}_n(t)$ is defined by (70).

Corollary 8. The numbers $a_o(n,\ell)$ and $\widetilde{a}_o(n,\ell)$ of labelled and unlabelled oriented k-gonal 2-trees , over n k-gons and having perimeter ℓ are given by

$$\begin{aligned}
a_o(n,\ell) &= \frac{1}{m}a^{-}_{o}(n,\ell) = \frac{1}{m}(a^{\rightarrow}_{n,\ell} + na^{\rightarrow\,(k-1)}_{n-1.\ell-1} - na^{\rightarrow\,(k-1)}_{n-1,\ell}), && (87)\\
\widetilde{a}_o(n,\ell) &= b_{n,\ell} - b^{(k)}_{n-1,\ell} + \frac{1}{k}\sum_{d|(k,\ell)}\phi(d)b^{(\frac{k}{d})}_{\frac{n-1}{d},\frac{\ell}{d}} + b^{(k-1)}_{n-1,\ell-1} - b^{(k-1)}_{n-1,\ell}. && (88)
\end{aligned}$$

5.3 Unoriented case

As in the unweighted case, unoriented species of k-gonal 2-trees can be expressed as quotient species of the oriented ones, as follows, where notations are obvious,

$$a_w^- = \frac{a_{o,w}^-}{\mathbb{Z}_2}, \qquad a_w^\diamond = \frac{a_{o,w}^\diamond}{\mathbb{Z}_2} \qquad a_w^\diamond = \frac{a_{o,w}^\diamond}{\mathbb{Z}_2} \tag{89}$$

It is very easy to obtain the number $a_{n,\ell}$ of labelled k-gonal 2-trees over n k-gons and having a perimeter of lenght ℓ,

$$a(n,\ell) = \begin{cases} \frac{1}{2}(a_o(n,\ell+1), & \text{if } \ell = (k-1)n, \\ \frac{1}{2}a_o(n,\ell), & \text{otherwise.} \end{cases} \tag{90}$$

since the only labelled k-gonal 2-trees fixed by orientation reversal for a given perimeter and number of polygons, is the one in which each k-gon share a common edge, which has $(k-1)n$ external edges (illustrated by Figure 12). So, the polynomial $a_n(t)$, giving the weighted enumeration of labelled k-gonal 2-trees, is given by

$$a_n(t) = \sum a_{n,\ell}\, t^\ell \;=\; \frac{1}{2}(a_{o,n}(t) + t^{(k-1)n}). \tag{91}$$

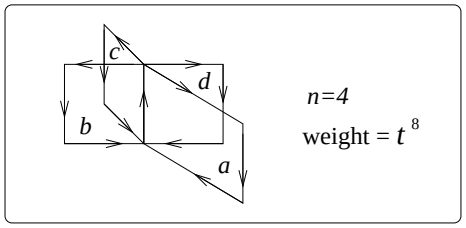


Figure 12: Labelled oriented 4-gonal 2-tree which is fixed by orientation reversion

For the unlabelled (weighted) enumeration, we have to adapt the results obtained in Section 4.2 and 4.3 to take into account the perimeter.

• k odd.

For k odd, we can easily see that the species a_w^-, $a_w^\diamond$ and $a_w^\diamond$ satisfy the following expressions in terms of the weighted quotient species Q_w, S_w and U_w, which are adapted from Section 4.1:

$$a_w^- = Q_w(X, B_{w^2}^{\frac{k-1}{2}}), \tag{92}$$

$$a_w^\diamond = X \cdot S_w(X, B_{w^2}^{\frac{k-1}{2}}), \tag{93}$$

$$a_w^\diamond = X \cdot U_w(X, B_{w^2}^{\frac{k-1}{2}}), \tag{94}$$

with

$$\begin{aligned} Q_w(X,Y) &= \left(t + tXY^2 + E_{\geq 2}(XY^2)\right)/\mathbb{Z}_2, & (95)\\ S_w(X,Y) &= C_k(t + E_+(XY^2))/\mathbb{Z}_2, & (96)\\ U_w(X,Y) &= \left((t + E_+(XY^2))^k\right)/\mathbb{Z}_2, & (97)\end{aligned}$$

where $E_{\geq 2}$ is the species of sets of cardinality at least two. The cycle index series of these species are given by:

$$\begin{aligned} Z_{Q_w} &= \frac{1}{2}(Z_{E_w(XY^2)} + q_w), & (98)\\ Z_{S_w} &= \frac{1}{2}\left(Z_{C_k(t+E_+(XY^2))} + q_w \cdot (p_2 \circ (t + Z_{E_+(XY^2)})^{\frac{k-1}{2}})\right), & (99)\\ Z_{U_w} &= \frac{1}{2}\left(Z_{(t+E_+(XY^2))^k} + q_w \cdot (p_2 \circ (t + Z_{E_+(XY^2)})^{\frac{k-1}{2}})\right), & (100)\end{aligned}$$

where $q_w = (t-1)(1+x_1y_2) + h \circ (x_1y_2 + p_2 \circ (x_1\frac{y_1^2-y_2}{2})) = q + (t-1)(1+x_1y_2)$, h being the homogeneous symmetric function, and p_i, $i \geq 1$, denotes the i^{th} power sum and $E_w(XY^2) = E(XY^2) + (t-1)(1+XY^2)$.

Another use of the dissymmetry theorem gives the ordinary generating series of unlabelled k-gonal 2-trees weighted by their perimeter:

$$\widetilde{a}(x,t) = \frac{1}{2}\left(\widetilde{a}_o(x,t) + q_w[x, \widetilde{B}^{\frac{k-1}{2}}(x,t)] + (t-1)(1 + x\widetilde{B}^{\frac{k-2}{2}}(x^2,t^2)\right), \quad (101)$$

where

$$q_w[x, \widetilde{B}^{\frac{k-1}{2}}(x,t)] := q_w(x, x^2, \ldots; \widetilde{B}^{\frac{k-1}{2}}(x,t), \widetilde{B}^{\frac{k-1}{2}}(x^2,t^2), \ldots).$$

- k **even**.

When k is even, it suffices to adapt all species introduced in Section 4.2 in the present weighted context. This is easily done, as follows, the index w meaning that the species are weighted according to perimeter. Note that the species $\mathcal{A}_{\mathrm{S},w}$ is a sub weighted-species of $\mathcal{A}^-_{o,w}$ by definition. We have:

$$\widetilde{a}_{\mathrm{S}}(x,t) = \left(E(P_{\mathrm{TS},w} + P_{\mathrm{M},w} + P_{\mathrm{AL},w}) + (t-1)(1 + P_{\mathrm{TS},w} + P_{\mathrm{M},w})\right)^{\sim}(x), \quad (102)$$

where

$$\begin{aligned} \mathcal{A}_{\mathrm{TS},w} &= t + t \cdot P_{\mathrm{TS},w} + E_{\geq 2}(P_{\mathrm{TS},w}) & (103)\\ &= (t-1)(1 + P_{\mathrm{TS},w}) + E(P_{\mathrm{TS},w}), & (104)\\ P_{\mathrm{TS},w} &= X \cdot X_=^2 < B^{\frac{k-2}{2}} > \cdot (\mathcal{A}_{\mathrm{TS},w} + (1-t)P_{\mathrm{TS},w}), & (105)\\ P_{\mathrm{AL},w} &= \Phi_2 < XB_w^{k-1} - (P_{\mathrm{TS},w} + P_{\mathrm{M},w}) >, & (106)\\ P_{\mathrm{M},w} &= X \cdot X_=^2 < B^{\frac{k-2}{2}} > \cdot (\mathcal{A}_{\mathrm{S},w} + (1-t)P_{\mathrm{M},w} - \mathcal{A}_{\mathrm{TS},w}). & (107)\end{aligned}$$

We then have

$$\widetilde{a}_{\mathrm{S}}(x,t) = \exp(\sum_{i\geq 1}\frac{1}{i}(\widetilde{P}_{\mathrm{TS}}(x^i,t^i)+\widetilde{P}_{\mathrm{M}}(x^i,t^i)+\widetilde{P}_{\mathrm{AL}}(x^i,t^i)))+(t-1)(1+\widetilde{P}_{\mathrm{TS}}(x,t)+\widetilde{P}_{\mathrm{M}}(x,t)), \tag{108}$$

where

$$\widetilde{a}_{\mathrm{TS}}(x,t) = (t-1)(1+\widetilde{P}_{\mathrm{TS}}(x,t)) + \exp\left(\sum_{i\geq 1}\frac{1}{i}\widetilde{P}_{\mathrm{TS}}(x^i,t^i)\right), \tag{109}$$

$$\widetilde{P}_{\mathrm{TS}}(x,t) = x\widetilde{B}^{\frac{k-2}{2}}(x^2,t^2)\left(\widetilde{a}_{\mathrm{TS}}(x,t)+(1-t)\widetilde{P}_{\mathrm{TS}}(x,t)\right), \tag{110}$$

$$\widetilde{P}_{\mathrm{AL}}(x,t) = \frac{1}{2}\big(x^2\widetilde{B}^{k-1}(x^2,t^2) - \widetilde{P}_{\mathrm{TS}}(x^2,t^2) - \widetilde{P}_{\mathrm{M}}(x^2,t^2)\big), \tag{111}$$

and

$$\begin{aligned}\widetilde{P}_{\mathrm{M}}(x,t) &= \left(XX_{=}^2 < B_w^{\frac{k-2}{2}} > \cdot(a_{\mathrm{TS},w}+(1-t)(1+P_{\mathrm{TS},w}))\cdot E_+(P_{\mathrm{AL},w}+P_{\mathrm{M},w})\right)^{\sim}(x)\\ &= x\widetilde{B}^{\frac{k-2}{2}}(x^2,t^2)\left(\widetilde{a}_{\mathrm{S}}(x,t)+(1-t)\widetilde{P}_{\mathrm{M}}(x,t)-\widetilde{a}_{\mathrm{TS}}(x,t)\right).\end{aligned} \tag{112}$$

It is then possible to compute the tilde generating functions of unlabelled structures associated to the species (89):

$$\begin{aligned}\widetilde{a}^{-}_{o,\tau}(x,t) &= \widetilde{a}_S(x,t),\\ \widetilde{a}^{\diamond}_{\cdot,\tau}(x,t) &= x\left(\widetilde{a}_S(x,t)+(1-t)(\widetilde{P}_{\mathrm{TS}}(x,t)+\widetilde{P}_{\mathrm{M}}(x,t)\right)^2\cdot\widetilde{B}^{\frac{k-2}{2}}(x^2,t^2),\\ \widetilde{a}^{\diamond}_{o,\tau}(x,t) &= \frac{x}{2}\left(\widetilde{a}_S(x,t)+(1-t)(\widetilde{P}_{\mathrm{TS}}(x,t)+\widetilde{P}_{\mathrm{M}}(x,t)\right)^2\cdot\widetilde{B}^{\frac{k-2}{2}}(x^2,t^2)+\frac{x}{2}\widetilde{B}^{\frac{k}{2}}(x^2,t^2).\end{aligned}$$

Finally, we obtain

$$\begin{aligned}\widetilde{a}(x,t) = \frac{1}{2}\widetilde{a}_o(x,t)+\frac{1}{2}\widetilde{a}_{\mathrm{S}}(x,t)+\frac{x}{4}\widetilde{B}^{\frac{k}{2}}(x^2,t^2)\\ -\frac{x}{4}\left(\widetilde{a}_S(x,t)+(1-t)(\widetilde{P}_{\mathrm{TS}}(x,t)+\widetilde{P}_{\mathrm{M}}(x,t)\right)^2\cdot\widetilde{B}^{\frac{k-2}{2}}(x^2,t^2).\end{aligned} \tag{113}$$

6 Asymptotics

Thanks to the dissymmetry theorem and to the various combinatorial equations related to it, the asymptotic enumeration of (labelled or unlabelled) k-gonal 2-trees depends essentially on the asymptotic enumeration of B-structures where B is the auxiliary species characterized by the functional equation (5). In the labelled case, the asymptotics is trivial since we have the simple explicit formulas (9), (19) and (21). The unlabelled case is more eleborate and makes use of the funtional equation (14) satisfied by the series $\widetilde{B}(x)$.

We need first the following result, which is a consequence of the classical theorem of Bender (see [2]) and is inspired form the approach of Fowler et al. for 2-trees (see [5, 6]).

Proposition 19. Let $p = k - 1$ and $\widetilde{B}(x) = \sum b_n(p)x^n$. Then, there exist constants α_p and β_p such that

$$b_n(p) \sim \alpha_p \beta_p^n n^{-3/2}, \qquad \text{as } n \to \infty. \tag{114}$$

Moreover,

$$\alpha_p = \alpha(\xi_p) = \frac{1}{\sqrt{2\pi}} \frac{1}{p^{1+\frac{1}{p}}} \xi_p^{-\frac{1}{p}} \left(1 + \frac{p\xi_p \omega'(\xi_p)}{\omega(\xi_p)}\right)^{\frac{1}{2}} \tag{115}$$

and

$$\beta_p = \frac{1}{\xi_p}, \tag{116}$$

where ξ_p is the smallest root of the equation

$$\xi = \frac{1}{ep} \omega^{-p}(\xi), \tag{117}$$

where $\omega(x)$ is the series given by

$$\omega(x) = \mathrm{e}^{\frac{1}{2}x^2 b^p(x^2) + \frac{1}{3}x^3 b^p(x^3) + \cdots}. \tag{118}$$

Proof. Write, for simplicity, $b(x) = \widetilde{B}(x)$. Then, thanks to (14), $y = b(x)$ satisfies the relation

$$y = e^{xy^p}\omega(x), \quad \text{where} \quad \omega(x) = e^{\frac{1}{2}x^2 b^p(x^2) + \frac{1}{3}x^3 b^p(x^3) + \cdots}. \tag{119}$$

By Bender's theorem applied to the function $f(x, y) = y - e^{xy^p}\omega(x)$, we have to find a solution (ξ_p, τ_p) of the system

$$f(x, y) = 0 \qquad \text{and} \qquad f_y(x, y) = 0. \tag{120}$$

It is equivalent to say that ξ_p is solution of (117) and that $p\xi_p\tau_p^p = 1$.

Since $f_{yy}(\xi_p, \tau_p) \neq 0$, ξ_p is an algebraic singularity of degree 2 of $b(x)$ and, for x near ξ_p, we have an expression of the form

$$b(x) = \tau_{p.0} + \tau_{p.1}(1 - \frac{x}{\xi_p})^{\frac{1}{2}} + \tau_{p,2}(1 - \frac{x}{\xi_p}) + \tau_{p.3}(1 - \frac{x}{\xi_p})^{\frac{3}{2}} + \cdots \tag{121}$$

where

$$\tau_{p,0} = \tau_p = b(\xi_p) = \left(\frac{1}{p\xi_p}\right)^{\frac{1}{p}}, \tag{122}$$

$$\tau_{p,1} = -\frac{\sqrt{2}}{p^{1+\frac{1}{p}}} \xi_p^{-\frac{1}{p}} \left(1 + \frac{p\xi_p \omega'(\xi_p)}{\omega(\xi_p)}\right)^{\frac{1}{2}}, \tag{123}$$

$$\tau_{p,2} = \frac{1}{3p^{2+\frac{1}{p}}} \xi_p^{-\frac{1}{p}} \left((2p+3) - p(p-3)\frac{\xi_p \omega'(\xi_p)}{\omega(\xi_p)}\right). \tag{124}$$

The asymptotic formula (114) with α_p and β_p given by (115) and (116) then follow from the fact that the main term of the asymptotic behavior of the coefficients $b_n(p)$ of x^n in (121) depends only on the term $\tau_{p,1}(1-\frac{x}{\xi_p})^{\frac{1}{2}}$ in (121) and is given by

$$b_n(p) \sim \binom{\frac{1}{2}}{n} \tau_{p,1}(-1)^n \frac{1}{\xi_p^n} \sim \alpha_p \beta_p^n n^{-\frac{3}{2}} \quad \text{as} \quad n \to \infty. \tag{125}$$

■

Note that ξ_p is the radius of convergence of $b(x)$ and that the radius of convergence of $\omega(x)$ is $\sqrt{\xi_p}$. It can be shown that $0 < \xi_p < \sqrt{\xi_p} < 1$. This implies that numerical approximations of ξ_p, for fixed p, can be computed by iteration using (117), and a suitable truncated polynomial approximations of $b(x)$. We now state our main asymptotic result.

Proposition 20. Let $p = k - 1$. Then, the number $\widetilde{a}_n$ of k-gonal 2-trees on n unlabelled k-gons satisfy

$$\widetilde{a}_n \sim \frac{1}{2}\widetilde{a}_{o,n}, \quad n \to \infty, \tag{126}$$

where $\widetilde{a}_{o,n}$ is the number of oriented k-gonal 2-trees over n unlabelled polygons. Moreover,

$$\widetilde{a}_{o,n} \sim \overline{\alpha}_p \beta_p^n n^{-5/2}, \quad n \to \infty, \tag{127}$$

where

$$\overline{\alpha_p} = 2\pi p^{1+\frac{2}{p}} \xi_p^{\frac{2}{p}} \alpha_p^3, \tag{128}$$

$$= \frac{1}{\sqrt{2\pi}} \frac{1}{p^{2+\frac{1}{p}}} \xi_p^{-\frac{1}{p}} \left(1 + p\frac{\omega'(\xi_p)}{\omega(\xi_p)}\right)^{\frac{3}{2}}, \tag{129}$$

and $\beta_p = \frac{1}{\xi_p}$ is the same growth as in Proposition 19.

Proof. The asymptotic formula (127) follows from the fact that the radius of convergence, ξ_p, of $\widetilde{a}(x)$, given by (31) for k odd and by (58) for k even, is equal to the radius of convergence of the dominating term $\frac{1}{2}\widetilde{a}_o(x)$. This is due to the easily checked fact that all terms in (31) and (58), except $\frac{1}{2}\widetilde{a}_o(x)$, have a radius of convergence greater or equal to $\sqrt{\xi_p} > \xi_p$. To establish (127), note first that, because of equation (18), the radius of convergence of $\widetilde{a}_o(x)$ is equal to the radius of covergence, ξ_p, of

$$b(x) - \frac{k-1}{k} x b^k(x), \tag{130}$$

where $b(x) = \widetilde{B}(x)$ and $k = p+1$. This implies that the asymptotic behavior of the coefficients $\widetilde{a}_{o,n}$ of $\widetilde{a}_o(x)$ is completely determined by that of (130). Substituting (121) into (130) and making use of (124) gives the following expansion

$$b(x) - \frac{k-1}{k} x b^k(x) = \overline{\tau}_{p,0} + \overline{\tau}_{p,1}\left(1 - \frac{x}{\xi_p}\right)^{\frac{1}{2}} + \overline{\tau}_{p,2}\left(1 - \frac{x}{\xi_p}\right) + \overline{\tau}_{p,3}\left(1 - \frac{x}{\xi_p}\right)^{\frac{3}{2}} + \cdots \tag{131}$$

where

$$\begin{aligned}
\overline{\tau}_{p,0} &= \frac{p}{p+1}\tau_{p,0}, && (132)\\
\overline{\tau}_{p,1} &= 0, && (133)\\
\overline{\tau}_{p,2} &= -\frac{1}{2}\frac{p(p+1)\tau_{p,1}^2 - 2\tau_{p,0}^2}{(p+1)\tau_{p,0}}, && (134)\\
\overline{\tau}_{p,3} &= -\frac{1}{6}\frac{\tau_{p,1}(6p\tau_{p,0}\tau_{p,2} + p(p-1)\tau_{p,1}^2 - 6\tau_{p,0}^2)}{\tau_{p,0}^2}, && (135)\\
&= -\frac{p}{3}\frac{\tau_{p,1}^3}{\tau_{p,0}^2}. && (136)
\end{aligned}$$

This implies that the dominating term for the asymptotic behavior of the coefficients $\widetilde{a}_{n,o}$ of x^n in $\widetilde{a}_o(x)$ depends only on the term $\overline{\tau}_{p,3}\left(1 - \frac{x}{\xi_p}\right)^{\frac{3}{2}}$ in (131) and is given by

$$\widetilde{a}_{n,o} \sim \binom{\frac{3}{2}}{n}\overline{\tau}_{p,3}(-1)^n \frac{1}{\xi_p^n} \sim \overline{\alpha}_p \beta_p n^{-\frac{5}{2}}, \quad \text{as } n \to \infty. \tag{137}$$

Computations making use of (136), (122) and (123), show that $\overline{\alpha}_p$ is indeed given by (128) and (129). ■

Our final result gives an explicit formula in terms of integer partitions for the common radius of convergence ξ_p of the series $\widetilde{B}(x)$, $\widetilde{a}(x)$ and $\widetilde{a}_o(x)$ from which the growth constant $\beta_p = \frac{1}{\xi_p}$ is obtained. We need the following special notations. If $\lambda = (\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_\nu)$ is a partition of an integer n in ν parts, we write $\lambda \vdash n$, $n = |\lambda|$, $\nu = l(\lambda)$, $m_i(\lambda) = |\{j : \lambda_j = i\}|$ = number of parts of size i in λ. Furthermore, we put

$$\sigma_i(\lambda) = \sum_{d|i} d m_d(\lambda), \quad \sigma_i^*(\lambda) = \sum_{d|i, d<i} d m_d(\lambda), \tag{138}$$

$$\widehat{\lambda} = 1 + |\lambda| + l(\lambda), \quad \widehat{z}(\lambda) = 2^{m_1(\lambda)} m_1(\lambda)! 3^{m_2(\lambda)} m_2(\lambda)! \ldots. \tag{139}$$

Proposition 21. We have the convergent expansion

$$\xi_p = \sum_{n=1}^{\infty} \frac{c_n}{p^n}, \tag{140}$$

where the coefficients c_n are constants, independent of p, explicitely given by

$$c_n = \sum_{\lambda \vdash n} \frac{e^{-\widehat{\lambda}}}{\widehat{\lambda}\widehat{z}(\lambda)} \prod_{i \geq 1} (\sigma_i(\lambda) - \widehat{\lambda})^{m_i(\lambda)-1} (\sigma_i^*(\lambda) - \widehat{\lambda}), \tag{141}$$

where λ runs over the set of partitions of n.

Proof. We establish the explicit formulas (140) and (141) by applying first Lagrange inversion to the equation $\xi = zR(\xi)$ where $z = \frac{1}{\mathrm{e}p}$ and $R(t) = \omega^{-p}(t)$, to get

$$\xi_p = \xi = \sum_{n \geq 1} \gamma_n \left(\frac{1}{\mathrm{e}p}\right)^n, \quad \text{and} \quad \gamma_n = \frac{1}{n}[t^{n-1}]\omega^{-np}(t). \tag{142}$$

Next, to explicitely evaluate $\omega^{-np}(x)$, we use Labelle's version ([12]) of the Good inversion formula in the context of cycle index series as follows. We begin with

$$\omega^p(x) = \exp(\frac{1}{2}px^2 b^p(x^2) + \frac{1}{3}px^3 b^p(x^3) + \cdots), \tag{143}$$

$$= \exp(\frac{1}{2}px_2 + \frac{1}{3}px_3 + \cdots) \circ Z_{XB^p(X)}\Big|_{x_i := x^i} \tag{144}$$

where the $\circ$ denotes the plethystic substitution. Using (7), we can then write $XB^p(X) = \frac{A(pX)}{p}$. This implies that

$$\omega^p(x) = \exp(\frac{1}{2}px_2 + \frac{1}{3}px_3 + \cdots) \circ \frac{Z_A(px_1, px_2, \ldots)}{p}\Big|_{x_i := x^i}, \tag{145}$$

and we get

$$\omega^{-np}(x) = \exp(-\frac{n}{2}px_2 - \frac{n}{3}px_3 - \cdots) \circ \left(\frac{1}{p} Z_A(px_1, px_2, \ldots)\right)\Big|_{x_i := x^i} \tag{146}$$

$$= \exp(-\frac{n}{2}x_2 - \frac{n}{3}x_3 - \cdots) \circ Z_A(x_1, x_2, \ldots)\Big|_{x_i := px^i}. \tag{147}$$

Then, using Labelle's inversion formula for cycle index series, we have, for any formal cycle index series $g(x_1, x_2, \ldots)$

$$[x_1^{n_1} x_2^{n_2} \ldots]\, g \circ Z_A(x_1, x_2, \ldots) = [t_1^{n_1} t_2^{n_2} \ldots] g(t_1, t_2, \ldots) \prod_{i=1}^{\infty} (1 - t_i) \exp(n_i(t_i + \frac{1}{2}t_{2i} + \cdots)), \tag{148}$$

and

$$\prod_{j=1}^{\infty} \exp(n_j(t_j + \frac{1}{2}t_{2j} + \cdots)) = \prod_{i=1}^{\infty} \exp(\sum_{d|i} d n_d \frac{t_i}{i}). \tag{149}$$

Taking $g(x_1, x_2, \ldots) = \exp(-\frac{\nu}{2}px_2 - \frac{\nu}{3}px_3 - \cdots)$, gives, after some computations,

$$[x_1^{n_1} x_2^{n_2} \ldots] \left(\exp(-\frac{\nu}{2}x_2 - \frac{\nu}{3}x_3 - \cdots) \circ Z_A\right) =$$

$$\left\{ \begin{array}{ll} 0 & \text{if} \quad n_1 > 0, \\ \left(\dfrac{\prod\limits_{i\geq 2}(-\nu+\sum\limits_{d|i} dn_d)^{n_i-1}(-\nu+\sum\limits_{d|i,d<i} dn_d)}{2^{n_2}n_2!3^{n_3}n_3!\ldots} \right) & \text{if} \quad n_1 = 0. \end{array} \right. \tag{150}$$

Making the substitution $x_i := px^i$, for $i = 1, 2, 3, \ldots$, gives the explicit formula

$$\omega^{-\nu p}(x) = \sum_{n\geq 0} \left(\sum_{2n_2+3n_3+\cdots=n} p^{n_2+n_3+\cdots} \frac{\prod\limits_{i\geq 2}(-\nu+\sum\limits_{d|i} dn_d)^{n_i-1}(-\nu+\sum\limits_{d|i,d<i} dn_d)}{2^{n_2}n_2!3^{n_3}n_3!\ldots} \right) x^n.$$

This implies, taking $\nu = n$ and using (142), that

$$\begin{aligned} \xi_p &= \sum_{n\geq 1} \frac{1}{n} \left(\sum_{2n_2+3n_3+\cdots=n-1} p^{n_2+n_3+\cdots} \frac{\prod\limits_{i\geq 2}(1-n+\sum\limits_{d|i} dn_d)^{n_i-1}(1-n+\sum\limits_{d|i,d<i} dn_d)}{2^{n_2}n_2!3^{n_3}n_3!\ldots} \right) \left(\frac{1}{ep}\right)^n, \\ &= \sum_{n\geq 1} \frac{c_n}{p^n}, \end{aligned}$$

where the coefficients c_n, $n \geq 1$, are given by (141). ■

Table 1, in the Appendix, gives, to 20 decimal places, the constants ξ_p, α_p, $\overline{\alpha}_p$ and $\beta_p = \frac{1}{\xi_p}$ for $p = 1, \ldots, 5$. Table 2 gives the exact values of the numbers $\widetilde{a}_n$, for k from 2 up to 12 and for $n = 0, 1, \ldots 20$, of the number of unlabelled k-gonal 2-trees built over n k-gons.

Here are the first few values of the universal constants c_n occuring in (140), for $n = 1, \ldots, 5$.

$$\begin{aligned} c_1 &= \frac{1}{\mathrm{e}} = 0.36787944117144232160, \\ c_2 &= -\frac{1}{2}\frac{1}{\mathrm{e}^3} = -0.02489353418393197149, \\ c_3 &= \frac{1}{8}\frac{1}{\mathrm{e}^5} - \frac{1}{3}\frac{1}{\mathrm{e}^4} = -0.00526296958802571004, \\ c_4 &= -\frac{1}{48}\frac{1}{\mathrm{e}^7} + \frac{1}{\mathrm{e}^6} - \frac{1}{4}\frac{1}{\mathrm{e}^5} = 0.00077526788594593923, \\ c_5 &= \frac{1}{384}\frac{1}{\mathrm{e}^9} - \frac{4}{3}\frac{1}{\mathrm{e}^8} + \frac{49}{72}\frac{1}{\mathrm{e}^7} - \frac{1}{5}\frac{1}{\mathrm{e}^6} = 0.00032212622183609932. \end{aligned} \tag{151}$$

Remark 2. The computations of this section are also valid for the case $k = 2$ ($p = 1$), corresponding to the case of classical rooted trees (*Cayley trees*) defined

by the functional equation $A = XE(A)$. In this case, the growth constant $\beta = \beta_1$, in (114), is known as the Otter constant (see [17]). It is interesting to note that this constant takes the explicit form $\beta = \frac{1}{\xi_1}$, with

$$\xi_1 = \sum_{n \geq 1} c_n. \tag{152}$$

Notice also that, when $k = 3$, we recover the asymptotic results of Fowler et al. in [5, 6].

References

[1] P. Auger, G. Labelle, P. Leroux, *Computing the molecular expansion of species with the Maple package Devmol*, 49$^{\text{th}}$ Séminaire Lotharingien de Combinatoire, submitted.

[2] E. A. Bender *Asymptotic methods in enumeration*, SIAM Rev., **16**, 485–515, (1974).

[3] F. Bergeron, G. Labelle, and P. Leroux, *Combinatorial Species and Tree-like Structures*, Encyclopedia of Mathematics and its Applications, vol. **67**, Cambridge University Press, (1998).

[4] L. Comtet, *Analyse Combinatoire*, tome premier, Presses Universitaires de France, (1970).

[5] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *Specifying 2-trees*, Proceedings FPSAC'00, Moscow, June 26-30 2000, D. Krob, A. A. Mikhalev, A. V. Mikhalev Eds, Springer-Verlag, 202–213.

[6] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *The Specification of 2-trees*, Advances in Applied Mathematics, **28**, 145–168, (2002).

[7] F. Harary and E. Palmer, *Graphical Enumeration*, Academic Press, New York, (1973).

[8] F. Harary, E. Palmer and R. Read, *On the cell-growth problem for arbitrary polygons*, Discrete Mathematics, **11**, 371–389, (1975).

[9] INRIA, *Encyclopedia of combinatorial structures.* http://algo.inria.fr/encyclopedia/index.html.

[10] T. Kloks, *Enumeration of biconnected partial 2-trees*, 26$^{\text{th}}$ Dutch Mathematical Conference, 1990.

[11] T. Kloks, *Treewidth*, Ph.D. Thesis, Royal University of Utrecht, Holland, (1993).

[12] G. Labelle, *Some new computational methods in the theory of species*, Combinatoire énumérative, Proceedings, Montréal, Québec, Lectures Notes in Mathematics, vol. 1234, Springer-Verlag, New-York/Berlin, 160–176, (1985).

[13] G. Labelle, C. Lamathe and P. Leroux, *Développement moléculaire de l'espèce des 2-arbres planaires*, Proceedings GASCom'01, 41–46, (2001).

[14] G. Labelle, C. Lamathe and P. Leroux, *A classification of plane and planar 2-trees*, 26 pages, to appear in Theoretical Computer Science.

[15] G. Labelle, C. Lamathe et P. Leroux, *Enumération des 2-arbres k-gonaux*, Second Colloquium on Mathematics and Computer Science, Versailles, September, 16–19, 2002, Trends in Mathematics, Éd. B. Chauvin, P. Flajolet et al., Birkhauser Verlag Basel Switzwerland, 95–109, (2002).

[16] C. Lamathe, *Molecular expansion of planar k-gonal 2-trees*, in preparation.

[17] R. Otter, *The number of trees*, Annals of Mathematics, **49**, 583–599, (1948).

[18] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, (1995).
`http://www.research.att.com/~njas/sequences`

E-mail adresses: [gilbert,lamathe,leroux]@lacim.uqam.ca

Appendix

Table 1 gives, to 20 decimal places, the constants ξ_p, α_p, $\overline{\alpha}_p$ and $\beta_p = \frac{1}{\xi_p}$ for $p = 1, \ldots, 5$.

p	ξ_p	α_p	$\overline{\alpha}_p$	β_p
1	0.338321856899	1.300312124682	1.581185475409	2.955765285652
2	0.177099522303	0.349261381742	0.349261381742	5.646542616233
3	0.119674100436	0.191997258650	0.067390781222	8.356026879296
4	0.090334539604	0.131073637349	0.034020667269	11.069962877759
5	0.072539192528	0.099178841365	0.020427915489	13.785651110085
6	0.060597948397	0.079660456931	0.013601784466	16.502208844693
7	0.052031135998	0.066517090385	0.009699566188	19.219261329064
8	0.045585869619	0.057075912245	0.007262873797	21.936622211299
9	0.040561059517	0.049970993036	0.005640546218	24.654188324989
10	0.036533820306	0.044433135893	0.004506504206	27.371897918664
11	0.033233950789	0.039996691773	0.003682863427	30.089711763681

Table 1: Numerical values of ξ_p, α_p, $\overline{\alpha}_p$ and β_p, $p = 1, \ldots, 5$

Table 2 gives the exact values of the numbers $\widetilde{a}_n$, for k from 2 up to 12 and for $n = 0, 1, \ldots 20$, of the number of unlabelled k-gonal 2-trees built over n k-gons.

Tables 3 and 4 give the polynomials $b_n(t)$, for $n = 0, 1, \ldots, 9$ and for k from 2 up to 9, of the weighted (by their perimeter) unlabelled oriented-edge-rooted k-gonal 2-trees over n k-gons.

$\boldsymbol{k = 2}$
1, 1, 1, 2, 3, 6, 11, 23, 47, 106, 235, 551, 1301, 3159, 7741, 19320, 48629, 123867, 317955, 823065, 2144505
$\boldsymbol{k = 3}$
1, 1, 1, 2, 5, 12, 39, 136, 529, 2171, 9368, 41534, 188942, 874906, 4115060, 19602156, 94419351, 459183768, 2252217207, 11130545494, 55382155396
$\boldsymbol{k = 4}$
1, 1, 1, 3, 8, 32, 141, 749, 4304, 26492, 169263, 1115015, 7507211, 51466500, 358100288, 2523472751, 17978488711, 129325796854, 938234533024, 6858551493579, 50478955083341
$\boldsymbol{k = 5}$
1, 1, 1, 3, 11, 56, 359, 2597, 20386, 167819, 1429815, 12500748, 111595289, 1013544057, 9340950309, 87176935700, 822559721606, 7836316493485, 75293711520236, 728968295958626, 7105984356424859
$\boldsymbol{k = 6}$
1, 1, 1, 4, 16, 103, 799, 7286, 71094, 729974, 7743818, 84307887, 937002302, 10595117272, 121568251909, 1412555701804, 16594126114458, 196829590326284, 2354703777373055, 28385225424840078, 344524656398655124
$\boldsymbol{k = 7}$
1, 1, 1, 4, 20, 158, 1539, 16970, 199879, 2460350, 31266165, 407461893, 5420228329, 73352481577, 1007312969202, 14008437540003, 196963172193733, 2796235114720116, 40038505601111596, 577693117173844307, 8392528734991449808
$\boldsymbol{k = 8}$
1, 1, 1, 5, 26, 245, 2737, 35291, 483819, 6937913, 102666626, 1558022255, 24133790815, 380320794122, 6081804068869, 98490990290897, 1612634990857755, 26660840123167203, 444560998431678554, 7469779489114328514, 126375763235359105446
$\boldsymbol{k = 9}$
1, 1, 1, 5, 32, 343, 4505, 66603, 1045335, 17115162, 289107854, 5007144433, 88516438360, 1591949961503, 29053438148676, 536972307386326, 10034276171127780, 189331187319203010, 3603141751525175854, 69097496637591215442, 1334213677527481808220
$\boldsymbol{k = 10}$
1, 1, 1, 6, 39, 482, 7053, 117399, 2070289, 38097139, 723169329, 14074851642, 279609377638, 5651139037570, 115901006038377, 2407291353219949, 50553753543016719, 1071971262516091572, 22926544048209731554, 494103705426160765546, 10722146465907412669810
$\boldsymbol{k = 11}$
1, 1, 1, 6, 46, 636, 10527, 194997, 3823327, 78118107, 1646300388, 35570427615, 784467060622, 17601062294302, 400750115756742, 9240636709048733, 215435023547580882, 5071520482516388865, 120417032326341878672, 2881134828445365441407, 6941046822030714862026
$\boldsymbol{k = 12}$
1, 1, 1, 7, 55, 840, 15189, 309607, 6671842, 149850849, 3471296793, 82442359291, 1998559329142, 49290785442796, 1233639304644946, 31268489727956101, 801335133177932829, 20736286803363051714, 541224489038545084067, 14234799536039481373552, 376974819516101224941091

Table 2: Values of $\widetilde{a}_n$ for $k = 2, \ldots,\ 12$ and $n = 0, \ldots,\ 20$

$\boldsymbol{k = 2}$
$t,$
$t,$
$t + t^2,$
$t + 2\,t^2 + t^3,$
$t + 4\,t^2 + 3\,t^3 + t^4,$
$t + 6\,t^2 + 8\,t^3 + 4\,t^4 + t^5,$
$t + 9\,t^2 + 18\,t^3 + 14\,t^4 + 5\,t^5 + t^6,$
$t + 12\,t^2 + 35\,t^3 + 39\,t^4 + 21\,t^5 + 6\,t^6 + t^7,$
$t + 16\,t^2 + 62\,t^3 + 97\,t^4 + 72\,t^5 + 30\,t^6 + 7\,t^7 + t^8,$
$t + 20\,t^2 + 103\,t^3 + 212\,t^4 + 214\,t^5 + 120\,t^6 + 40\,t^7 + 8\,t^8 + t^9$

$\boldsymbol{k = 3}$
t
t^2
$2\,t^3 + t^4$
$5\,t^4 + 4\,t^5 + t^6$
$14\,t^5 + 18\,t^6 + 6\,t^7 + t^8$
$42\,t^6 + 72\,t^7 + 37\,t^8 + 8\,t^9 + t^{10}$
$132\,t^7 + 291\,t^8 + 204\,t^9 + 64\,t^{10} + 10\,t^{11} + t^{12}$
$429\,t^8 + 1152\,t^9 + 1048\,t^{10} + 438\,t^{11} + 97\,t^{12} + 12\,t^{13} + t^{14}$
$1430\,t^9 + 4558\,t^{10} + 5128\,t^{11} + 2757\,t^{12} + 804\,t^{13} + 138\,t^{14} + 14\,t^{15} + t^{16}$
$4862\,t^{10} + 17944\,t^{11} + 24249\,t^{12} + 16108\,t^{13} + 5981\,t^{14} + 1332\,t^{15} + 185\,t^{16} + 16\,t^{17} + t^{18}$

$\boldsymbol{k = 4}$
t
t^3
$3\,t^5 + t^6$
$12\,t^7 + 6\,t^8 + t^9$
$55\,t^9 + 42\,t^{10} + 9\,t^{11} + t^{12}$
$273\,t^{11} + 274\,t^{12} + 87\,t^{13} + 12\,t^{14} + t^{15}$
$1428\,t^{13} + 1806\,t^{14} + 767\,t^{15} + 150\,t^{16} + 15\,t^{17} + t^{18}$
$7752\,t^{15} + 11820\,t^{16} + 6387\,t^{17} + 1641\,t^{18} + 228\,t^{19} + 18\,t^{20} + t^{21}$
$43263\,t^{17} + 77440\,t^{18} + 51078\,t^{19} + 16614\,t^{20} + 3006\,t^{21} + 324\,t^{22} + 21\,t^{23} + t^{24}$
$246675\,t^{19} + 507246\,t^{20} + 396905\,t^{21} + 157638\,t^{22} + 35847\,t^{23} + 4972\,t^{24} + 435\,t^{25} + 24\,t^{26} + t^{27}$

$\boldsymbol{k = 5}$
t
t^4
$4\,t^7 + t^8$
$22\,t^{10} + 8\,t^{11} + t^{12}$
$140\,t^{13} + 76\,t^{14} + 12\,t^{15} + t^{16}$
$969\,t^{16} + 688\,t^{17} + 158\,t^{18} + 16\,t^{19} + t^{20}$
$7084\,t^{19} + 6290\,t^{20} + 1916\,t^{21} + 272\,t^{22} + 20\,t^{23} + t^{24}$
$53820\,t^{22} + 57376\,t^{23} + 22064\,t^{24} + 4092\,t^{25} + 414\,t^{26} + 24\,t^{27} + t^{28}$
$420732\,t^{25} + 524412\,t^{26} + 244840\,t^{27} + 57113\,t^{28} + 7488\,t^{29} + 588\,t^{30} + 28\,t^{31} + t^{32}$
$3362260\,t^{28} + 4799568\,t^{29} + 2645854\,t^{30} + 749908\,t^{31} + 122908\,t^{32} + 12376\,t^{33} + 790\,t^{34} + 32\,t^{35} + t^{36}$

Table 3: Polynomials $b_n(t)$ for $k = 2, 3, 4, 5$ and $n = 0, \ldots 9$

$\boldsymbol{k=6}$

t

t^5

$5\,t^9+t^{10}$

$35\,t^{13}+10\,t^{14}+t^{15}$

$285\,t^{17}+120\,t^{18}+15\,t^{19}+t^{20}$

$2530\,t^{21}+1390\,t^{22}+250\,t^{23}+20\,t^{24}+t^{25}$

$23751\,t^{25}+16255\,t^{26}+3860\,t^{27}+430\,t^{28}+25\,t29+t^{30}$

$231880\,t^{29}+190106\,t^{30}+56755\,t^{31}+8235\,t^{32}+655\,t^{33}+30\,t^{34}+t^{35}$

$2330445\,t^{33}+2229120\,t^{34}+805621\,t^{35}+146510\,t^{36}+15060\,t^{37}+930\,t^{38}+35\,t^{39}+t^{40}$

$23950355\,t^{37}+26193570\,t^{38}+11149900\,t^{39}+2457081\,t^{40}+314810\,t^{41}+24880\,t^{42}+1250\,t^{43}+40\,t^{44}+t^{45}$

$\boldsymbol{k=7}$

t

t^6

$6\,t^{11}+t^{12}$

$51\,t^{16}+12\,t^{17}+t^{18}$

$506\,t^{21}+174\,t^{22}+18\,t^{23}+t^{24}$

$5481\,t^{26}+2456\,t^{27}+363\,t^{28}+24\,t^{29}+t^{30}$

$62832\,t^{31}+34989\,t^{32}+6808\,t^{33}+624\,t^{34}+30\,t^{35}+t^{36}$

$749398\,t^{36}+499188\,t^{37}+121800\,t^{38}+14514\,t^{39}+951\,t^{40}+36\,t^{41}+t^{42}$

$9203634\,t^{41}+7143466\,t^{42}+2106138\,t^{43}+313872\,t^{44}+26532\,t^{45}+1350\,t^{46}+42\,t^{47}+t^{48}$

$115607310\,t^{46}+102489288\,t^{47}+35536296\,t^{48}+6406278\,t^{49}+673749\,t^{50}+43820\,t^{51}+1815\,t^{52}+48\,t^{53}+t^{54}$

$\boldsymbol{k=8}$

t

t^7

$7\,t^{13}+t^{14}$

$70\,t^{19}+14\,t^{20}+t^{21}$

$819\,t^{25}+238\,t^{26}+21\,t^{27}+t^{28}$

$10472\,t^{31}+3962\,t^{32}+497\,t^{33}+28\,t^{34}+t^{35}$

$141778\,t^{37}+66556\,t^{38}+10969\,t^{39}+854\,t^{40}+35\,t^{41}+t^{42}$

$1997688\,t^{43}+1120658\,t^{44}+231203\,t^{45}+23373\,t^{46}+1302\,t^{47}+42\,t^{48}+t^{49}$

$28989675\,t^{49}+18932368\,t^{50}+4713849\,t^{51}+595077\,t^{52}+42714\,t^{53}+1848\,t^{54}+49\,t^{55}+t^{56}$

$430321633\,t^{55}+320771256\,t^{56}+93827895\,t^{57}+14311479\,t^{58}+1276471\,t^{59}+70532\,t^{60}+2485\,t^{61}+56\,t^{62}+t^{63}$

$\boldsymbol{k=9}$

t

t^8

$8\,t^{15}+t^{16}$

$92\,t^{22}+16\,t^{23}+t^{24}$

$1240\,t^{29}+312\,t^{30}+24\,t^{31}+t^{32}$

$18278\,t^{36}+5984\,t^{37}+652\,t^{38}+32\,t^{39}+t^{40}$

$285384\,t^{43}+115796\,t^{44}+16552\,t^{45}+1120\,t^{46}+40\,t^{47}+t^{48}$

$4638348\,t^{50}+2247376\,t^{51}+401632\,t^{52}+35256\,t^{53}+1708\,t^{54}+48\,t^{55}+t^{56}$

$77652024\,t^{57}+43772920\,t^{58}+9432184\,t^{59}+1032814\,t^{60}+64416\,t^{61}+2424\,t^{62}+56\,t^{63}+t^{64}$

$1329890705\,t^{64}+855243648\,t^{65}+216340024\,t^{66}+28597424\,t^{67}+2214272\,t^{68}+106352\,t^{69}+3260\,t^{70}+64\,t^{71}+t^{72}$

Table 4: Polynomials $b_n(t)$ for $k=6,7,8,9$ and $n=0,\ldots,9$

$\boldsymbol{k=2}$

t
t^2
t^2
t^2+t^3
$t^2+t^3+t^4$
$t^2+2\,t^3+2\,t^4+t^5$
$t^2+3\,t^3+4\,t^4+2\,t^5+t^6$
$t^2+4\,t^3+8\,t^4+6\,t^5+3\,t^6+t^7$
$t^2+5\,t^3+14\,t^4+14\,t^5+9\,t^6+3\,t^7+t^8$
$t^2+7\,t^3+23\,t^4+32\,t^5+26\,t^6+12\,t^7+4\,t^8+t^9$
$t^2+8\,t^3+36\,t^4+64\,t^5+66\,t^6+39\,t^7+16\,t^8+4\,t^9+t^{10}$

$\boldsymbol{k=3}$

t
t^3
t^4
t^5+t^6
$3\,t^6+t^7+t^8$
$4\,t^7+5\,t^8+2\,t^9+t^{10}$
$12\,t^8+14\,t^9+10\,t^{10}+2\,t^{11}+t^{12}$
$27\,t^9+53\,t^{10}+37\,t^{11}+15\,t^{12}+3\,t^{13}+t^{14}$
$82\,t^{10}+179\,t^{11}+171\,t^{12}+71\,t^{13}+22\,t^{14}+3\,t^{15}+t^{16}$
$228\,t^{11}+664\,t^{12}+716\,t^{13}+401\,t^{14}+128\,t^{15}+29\,t^{16}+4\,t^{17}+t^{18}$
$733\,t^{12}+2386\,t^{13}+3128\,t^{14}+2051\,t^{15}+825\,t^{16}+201\,t^{17}+39\,t^{18}+4\,t^{19}+t^{20}$

$\boldsymbol{k=4}$

t
t^4
t^6
$2\,t^8+t^9$
$7\,t^{10}+3\,t^{11}+t^{12}$
$25\,t^{12}+18\,t^{13}+5\,t^{14}+t^{15}$
$108\,t^{14}+101\,t^{15}+36\,t^{16}+6\,t^{17}+t^{18}$
$492\,t^{16}+588\,t^{17}+259\,t^{18}+58\,t^{19}+8\,t^{20}+t^{21}$
$2431\,t^{18}+3471\,t^{19}+1887\,t^{20}+519\,t^{21}+87\,t^{22}+9\,t^{23}+t^{24}$
$12371\,t^{20}+20834\,t^{21}+13521\,t^{22}+4569\,t^{23}+921\,t^{24}+120\,t^{25}+11\,t^{26}+t^{27}$
$65169\,t^{22}+125976\,t^{23}+96096\,t^{24}+38730\,t^{25}+9411\,t^{26}+1474\,t^{27}+160\,t^{28}+12\,t^{29}+t^{30}$

$\boldsymbol{k=5}$

t
t^5
t^8
$2\,t^{11}+t^{12}$
$8\,t^{14}+2\,t^{15}+t^{16}$
$33\,t^{17}+18\,t^{18}+4\,t^{19}+t^{20}$
$194\,t^{20}+124\,t^{21}+36\,t^{22}+4\,t^{23}+t^{24}$
$1196\,t^{23}+1014\,t^{24}+324\,t^{25}+56\,t^{26}+6\,t^{27}+t^{28}$
$8196\,t^{26}+8226\,t^{27}+3233\,t^{28}+640\,t^{29}+84\,t^{30}+6\,t^{31}+t^{32}$
$58140\,t^{29}+68780\,t^{30}+31846\,t^{31}+7787\,t^{32}+1143\,t^{33}+114\,t^{34}+8\,t^{35}+t^{36}$
$427975\,t^{32}+579266\,t^{33}+313832\,t^{34}+90742\,t^{35}+16019\,t^{36}+1820\,t^{37}+152\,t^{38}+8\,t^{39}+t^{40}$

Table 5: Coefficients of $\widetilde{a}_o(x,t)$ for $k=2,3,4,5$ and $n=0,\ldots,10$

$\boldsymbol{k = 6}$

t

t^{6}

t^{10}

$3\,t^{14} + t^{15}$

$19\,t^{18} + 5\,t^{19} + t^{20}$

$118\,t^{22} + 50\,t^{23} + 8\,t^{24} + t^{25}$

$931\,t^{26} + 495\,t^{27} + 100\,t^{28} + 10\,t^{29} + t^{30}$

$7756\,t^{30} + 5110\,t^{31} + 1266\,t^{32} + 164\,t^{33} + 13\,t^{34} + t^{35}$

$68685\,t^{34} + 53801\,t^{35} + 16275\,t^{36} + 2560\,t^{37} + 245\,t^{38} + 15\,t^{39} + t^{40}$

$630465\,t^{38} + 575535\,t^{39} + 206954\,t^{40} + 39445\,t^{41} + 4529\,t^{42} + 340\,t^{43} + 18\,t^{44} + t^{45}$

$5966610\,t^{42} + 6224520\,t^{43} + 2611405\,t^{44} + 589676\,t^{45} + 81145\,t^{46} + 7285\,t^{47} + 454\,t^{48} + 20\,t^{49} + t^{50}$

$\boldsymbol{k = 7}$

t

t^{7}

t^{12}

$3\,t^{17} + t^{18}$

$16\,t^{22} + 3\,t^{23} + t^{24}$

$112\,t^{27} + 39\,t^{28} + 6\,t^{29} + t^{30}$

$1020\,t^{32} + 434\,t^{33} + 78\,t^{34} + 6\,t^{35} + t^{36}$

$10222\,t^{37} + 5487\,t^{38} + 1127\,t^{39} + 124\,t^{40} + 9\,t^{41} + t^{42}$

$109947\,t^{42} + 70053\,t^{43} + 17436\,t^{44} + 2247\,t^{45} + 186\,t^{46} + 9\,t^{47} + t^{48}$

$1230840\,t^{47} + 914103\,t^{48} + 268995\,t^{49} + 42144\,t^{50} + 4000\,t^{51} + 255\,t^{52} + 12\,t^{53} + t^{54}$

$14218671\,t^{52} + 12057540\,t^{53} + 4131929\,t^{54} + 764623\,t^{55} + 86652\,t^{56} + 6397\,t^{57} + 340\,t^{58} + 12\,t^{59} + t^{60}$

$\boldsymbol{k = 8}$

t

t^{8}

t^{14}

$4\,t^{20} + t^{21}$

$35\,t^{26} + 7\,t^{27} + t^{28}$

$332\,t^{32} + 98\,t^{33} + 11\,t^{34} + t^{35}$

$3766\,t^{38} + 1393\,t^{39} + 196\,t^{40} + 14\,t^{41} + t^{42}$

$45448\,t^{44} + 20650\,t^{45} + 3561\,t^{46} + 322\,t^{47} + 18\,t^{48} + t^{49}$

$580203\,t^{50} + 312739\,t^{51} + 65590\,t^{52} + 7217\,t^{53} + 483\,t^{54} + 21\,t^{55} + t^{56}$

$7684881\,t^{56} + 4813130\,t^{57} + 1197467\,t^{58} + 158928\,t^{59} + 12762\,t^{60} + 672\,t^{61} + 25\,t^{62} + t^{63}$

$104898024\,t^{62} + 74961328\,t^{63} + 21701960\,t^{64} + 3403708\,t^{65} + 326760\,t^{66} + 20552\,t^{67} + 896\,t^{68} + 28\,t^{69} + t^{70}$

$\boldsymbol{k = 9}$

t

t^{9}

t^{16}

$4\,t^{23} + t^{24}$

$27\,t^{30} + 4\,t^{31} + t^{32}$

$266\,t^{37} + 68\,t^{38} + 8\,t^{39} + t^{40}$

$3312\,t^{44} + 1048\,t^{45} + 136\,t^{46} + 8\,t^{47} + t^{48}$

$45711\,t^{51} + 17948\,t^{52} + 2712\,t^{53} + 219\,t^{54} + 12\,t^{55} + t^{56}$

$670344\,t^{58} + 312276\,t^{59} + 56942\,t^{60} + 5432\,t^{61} + 328\,t^{62} + 12\,t^{63} + t^{64}$

$10233201\,t^{65} + 5539348\,t^{66} + 1194736\,t^{67} + 137754\,t^{68} + 9654\,t^{69} + 452\,t^{70} + 16\,t^{71} + t^{72}$

$161055618\,t^{72} + 99432684\,t^{73} + 24928832\,t^{74} + 3391482\,t^{75} + 283146\,t^{76} + 15472\,t^{77} + 603\,t^{78} + 16\,t^{79} + t^{80}$

Table 6: Coefficients of $\widetilde{\alpha}_o(x,t)$ for $k = 6, 7, 8, 9$ and $n = 0, \ldots, 10$

$\boldsymbol{k=2}$

t
t^2
t^2
t^2+t^3
$t^2+t^3+t^4$
$t^2+2\,t^3+2\,t^4+t^5$
$t^2+3\,t^3+4\,t^4+2\,t^5+t^6$
$t^2+4\,t^3+8\,t^4+6\,t^5+3\,t^6+t^7$
$t^2+5\,t^3+14\,t^4+14\,t^5+9\,t^6+3\,t^7+t^8$
$t^2+7\,t^3+23\,t^4+32\,t^5+26\,t^6+12\,t^7+4\,t^8+t^9$
$t^2+8\,t^3+36\,t^4+64\,t^5+66\,t^6+39\,t^7+16\,t^8+4\,t^9+t^{10}$

$\boldsymbol{k=3}$

t
t^3
t^4
t^5+t^6
$4\,t^6+2\,t^7+t^8$
$6\,t^7+8\,t^8+3\,t^9+t^{10}$
$19\,t^8+28\,t^9+16\,t^{10}+4\,t^{11}+t^{12}$
$49\,t^9+100\,t^{10}+70\,t^{11}+26\,t^{12}+5\,t^{13}+t^{14}$
$150\,t^{10}+358\,t^{11}+325\,t^{12}+142\,t^{13}+38\,t^{14}+6\,t^{15}+t^{16}$
$442\,t^{11}+1309\,t^{12}+1414\,t^{13}+783\,t^{14}+250\,t^{15}+52\,t^{16}+7\,t^{17}+t^{18}$
$1424\,t^{12}+4772\,t^{13}+6186\,t^{14}+4102\,t^{15}+1615\,t^{16}+402\,t^{17}+70\,t^{18}+8\,t^{19}+t^{20}$

$\boldsymbol{k=4}$

t
t^4
t^6
$2\,t^8+t^9$
$5\,t^{10}+2\,t^{11}+t^{12}$
$16\,t^{12}+11\,t^{13}+4\,t^{14}+t^{15}$
$60\,t^{14}+54\,t^{15}+22\,t^{16}+4\,t^{17}+t^{18}$
$261\,t^{16}+305\,t^{17}+142\,t^{18}+34\,t^{19}+6\,t^{20}+t^{21}$
$1243\,t^{18}+1755\,t^{19}+975\,t^{20}+273\,t^{21}+51\,t^{22}+6\,t^{23}+t^{24}$
$6257\,t^{20}+10478\,t^{21}+6853\,t^{22}+2336\,t^{23}+490\,t^{24}+69\,t^{25}+8\,t^{26}+t^{27}$
$32721\,t^{22}+63100\,t^{23}+48271\,t^{24}+19497\,t^{25}+4803\,t^{26}+770\,t^{27}+92\,t^{28}+8\,t^{29}+t^{30}$

$\boldsymbol{k=5}$

t
t^5
t^8
$2\,t^{11}+t^{12}$
$12\,t^{14}+4\,t^{15}+t^{16}$
$57\,t^{17}+32\,t^{18}+6\,t^{19}+t^{20}$
$366\,t^{20}+248\,t^{21}+64\,t^{22}+8\,t^{23}+t^{24}$
$2340\,t^{23}+2002\,t^{24}+630\,t^{25}+104\,t^{26}+10\,t^{27}+t^{28}$
$16252\,t^{26}+16452\,t^{27}+6393\,t^{28}+1280\,t^{29}+156\,t^{30}+12\,t^{31}+t^{32}$
$115940\,t^{29}+137378\,t^{30}+63516\,t^{31}+15493\,t^{32}+2259\,t^{33}+216\,t^{34}+14\,t^{35}+t^{36}$
$854981\,t^{32}+1158532\,t^{33}+626996\,t^{34}+181484\,t^{35}+31887\,t^{36}+3640\,t^{37}+288\,t^{38}+16\,t^{39}+t^{40}$

Table 7: Coefficients of $\widetilde{a}(x,t)$ for $k=2,3,4,5$ and $n=0,\ldots,10$

$\boldsymbol{k = 6}$

t

t^{6}

t^{10}

$3\,t^{14} + t^{15}$

$12\,t^{18} + 3\,t^{19} + t^{20}$

$68\,t^{22} + 28\,t^{23} + 6\,t^{24} + t^{25}$

$483\,t^{26} + 253\,t^{27} + 56\,t^{28} + 6\,t^{29} + t^{30}$

$3946\,t^{30} + 2582\,t^{31} + 659\,t^{32} + 89\,t^{33} + 9\,t^{34} + t^{35}$

$34485\,t^{34} + 26953\,t^{35} + 8213\,t^{36} + 1300\,t^{37} + 133\,t^{38} + 9\,t^{39} + t^{40}$

$315810\,t^{38} + 288021\,t^{39} + 103799\,t^{40} + 19831\,t^{41} + 2318\,t^{42} + 182\,t^{43} + 12\,t^{44} + t^{45}$

$2984570\,t^{42} + 3112780\,t^{43} + 1306605\,t^{44} + 295143\,t^{45} + 40775\,t^{46} + 3689\,t^{47} + 243\,t^{48} + 12\,t^{49} + t^{50}$

$\boldsymbol{k = 7}$

t

t^{7}

t^{12}

$3\,t^{17} + t^{18}$

$26\,t^{22} + 6\,t^{23} + t^{24}$

$203\,t^{27} + 72\,t^{28} + 9\,t^{29} + t^{30}$

$41989\,t^{32} + 868\,t^{33} + 144\,t^{34} + 12\,t^{35} + t^{36}$

$20254\,t^{37} + 10914\,t^{38} + 2212\,t^{39} + 236\,t^{40} + 15\,t^{41} + t^{42}$

$219388\,t^{42} + 140106\,t^{43} + 34704\,t^{44} + 4494\,t^{45} + 354\,t^{46} + 18\,t^{47} + t^{48}$

$2459730\,t^{47} + 1827555\,t^{48} + 537357\,t^{49} + 84102\,t^{50} + 7937\,t^{51} + 492\,t^{52} + 21\,t^{53} + t^{54}$

$28431861\,t^{52} + 24115080\,t^{53} + 8261473\,t^{54} + 1529246\,t^{55} + 172956\,t^{56} + 12794\,t^{57} + 656\,t^{58} + 24\,t^{59} + t^{60}$

$\boldsymbol{k = 8}$

t

t^{8}

t^{14}

$4\,t^{20} + t^{21}$

$21\,t^{26} + 4\,t^{27} + t^{28}$

$183\,t^{32} + 53\,t^{33} + 8\,t^{34} + t^{35}$

$1918\,t^{38} + 704\,t^{39} + 106\,t^{40} + 8\,t^{41} + t^{42}$

$22908\,t^{44} + 10375\,t^{45} + 1825\,t^{46} + 170\,t^{47} + 12\,t^{48} + t^{49}$

$290511\,t^{50} + 156471\,t^{51} + 32934\,t^{52} + 3635\,t^{53} + 255\,t^{54} + 12\,t^{55} + t^{56}$

$3844688\,t^{56} + 2407227\,t^{57} + 599513\,t^{58} + 79651\,t^{59} + 6466\,t^{60} + 351\,t^{61} + 16\,t^{62} + t^{63}$

$52454248\,t^{62} + 37482092\,t^{63} + 10853332\,t^{64} + 1702405\,t^{65} + 163728\,t^{66} + 10336\,t^{67} + 468\,t^{68} + 16\,t^{69} + t^{70}$

$\boldsymbol{k = 9}$

t

t^{9}

t^{16}

$4\,t^{23} + t^{24}$

$46\,t^{30} + 8\,t^{31} + t^{32}$

$494\,t^{37} + 128\,t^{38} + 12\,t^{39} + t^{40}$

$6532\,t^{44} + 2096\,t^{45} + 256\,t^{46} + 16\,t^{47} + t^{48}$

$90954\,t^{51} + 35788\,t^{52} + 5348\,t^{53} + 422\,t^{54} + 20\,t^{55} + t^{56}$

$1339448\,t^{58} + 624552\,t^{59} + 113582\,t^{60} + 10864\,t^{61} + 632\,t^{62} + 24\,t^{63} + t^{64}$

$20459857\,t^{65} + 11077108\,t^{66} + 2387924\,t^{67} + 275174\,t^{68} + 19194\,t^{69} + 880\,t^{70} + 28\,t^{71} + t^{72}$

$322092958\,t^{72} + 198865368\,t^{73} + 49851852\,t^{74} + 6782964\,t^{75} + 565666\,t^{76} + 30944\,t^{77} + 1174\,t^{78} + 32\,t^{79} + t^{80}$

Table 8: Coefficients of $\widetilde{a}(x,t)$ for $k = 6, 7, 8, 9$ and $n = 0, \ldots, 10$

ANALYTIC CONTINUATION OF MULTIPLE ZETA-FUNCTIONS AND THEIR VALUES AT NON-POSITIVE INTEGERS

SHIGEKI AKIYAMA, SHIGEKI EGAMI AND YOSHIO TANIGAWA

ABSTRACT. Analytic continuation of the multiple zeta-function is established by a simple application of the Euler-Maclaurin summation formula. Multiple zeta values at non-positive integers are defined and their properties are investigated.

1. INTRODUCTION

The multiple zeta values due to D. Zagier are defined by

$$\zeta_k(s_1, s_2, \dots, s_k) = \sum_{0<n_1<n_2<\dots<n_k} \frac{1}{n_1^{s_1} n_2^{s_2} \dots n_k^{s_k}}$$

with positive integers s_i $(i = 1, 2, \dots, k)$ and $s_k \geq 2$. These values have a certain connection with topology and physics, and algebraic relations among them are extensively studied (see [18], [19], [6], [7] and [14]). Recently, Y. Ohno developed a unified algebraic relation in [16]. It is also interesting to consider it for complex variables s_i.

In this paper, we treat analytic continuation of $\zeta_k(s_1, s_2, \dots, s_k)$. Analytic continuation of $\zeta_2(s_1, s_2)$ was proved by F.V. Atkinson [5] with applications to the study of the asymptotic behavior of the 'mean values' of zeta-functions. See also Y. Motohashi [15] and M. Katsurada & K. Matsumoto [13]. In [4], T. Arakawa & M. Kaneko used analytic continuation of $\zeta_k(s_1, s_2, \dots, s_k)$ as a function of one variable s_k when $s_1, s_2, \dots, s_{k-1}$ are positive integers, and discussed the relation among generalized Bernoulli numbers. On the other hand, S. Egami discussed the relationship among various multiple zeta-functions introduced by E.W. Barnes, T. Shintani and D. Zagier. (See [9] and [10].)

However, for a general k, we cannot find the proof of analytic continuation of $\zeta_k(s_1, s_2, \dots, s_k)$ as a function of k variables in literature (but see the comment of Zagier [18, p. 509, lines 14–19]). We shall show that the multiple zeta-function can be continued analytically to $\mathbb{C}^k$ and discuss interesting properties of multiple zeta values at non-positive integers.

1991 *Mathematics Subject Classification.* Primary 11M41; Secondary 30B40,32Dxx.

The authors wish to express their gratitude to the referee for valuable comments on the earlier version of the present paper. The third author also thanks Professor Aleksandar Ivić for giving him useful comments.

Remark 1. After submitting the first version of our paper, we found a recent work of J. Zhao [20] treating analytic continuation of multiple zeta-function. This fact was also pointed out by the referee. With the help of the theory of generalized function in the sense of I.M. Gel'fand and G.E. Shilov, he gave their possible singularities as well as the residues. However our method is apparently simple and reveals the *exact* location of singularities, which seems to be an advantage.

2. Analytic continuation

Let l and m be positive integers. Define an entire function:

$$\phi_l(m,s)=\sum_{n=1}^{m}\frac{1}{n^s}-\left\{\frac{m^{1-s}-1}{1-s}+\frac{1}{2m^s}-\sum_{q=1}^{l}\frac{(s)_q a_q}{m^{s+q}}+\zeta(s)-\frac{1}{s-1}\right\} \tag{1}$$

with $(s)_n=s(s+1)\cdots(s+n-1)$ and $a_q=B_{q+1}/(q+1)!$. Here B_q are Bernoulli numbers defined by $z/(e^z-1)=\sum_{q=0}^{\infty}B_q z^q/q!$ and $\zeta(s)$ is the Riemann zeta-function. By using the Euler-Maclaurin summation formula, we have $\phi_l(m,s)=O(|(s)_{l+1}|m^{-\Re(s)-l-1})$ when s is a complex number. Considering s as a complex variable and $m\to\infty$, we get an analytic continuation of $\zeta(s)$ in $\Re(s+l+1)>0$. Note that (1) is also valid when $s\to 1$, if we replace $(m^{1-s}-1)/(1-s)$ by its limit $\log m$.

This is one of the oldest way of the analytic continuation of the Riemann zeta-function, which provides us with a method of numerical calculations in the critical strip $0<\Re s<1$. (c.f. [8], [12]). It does not give us the celebrated functional equation of $\zeta(s)$ directly, but it is possible to derive it by more precise observations (see Chapter 2 of [17]). Hereafter we will use (1) in the form:

$$\sum_{n=m+1}^{\infty}\frac{1}{n^s}=-\phi_l(m,s)+\frac{m^{1-s}}{s-1}-\frac{1}{2m^s}+\sum_{q=1}^{l}\frac{(s)_q a_q}{m^{s+q}}, \tag{2}$$

for $\Re(s)>1$. Consider the multiple zeta-function in two variables:

$$\zeta_2(s_1,s_2)=\sum_{0<n_1<n_2}\frac{1}{n_1^{s_1}n_2^{s_2}}$$

with $\Re s_i > 1$ $(i = 1, 2)$. By (2),

$$\begin{aligned}
\zeta_2(s_1, s_2) &= \sum_{n_1=1}^{\infty} \frac{1}{n_1^{s_1}} \sum_{n_2=n_1+1}^{\infty} \frac{1}{n_2^{s_2}} \\
&= \sum_{n_1=1}^{\infty} \frac{1}{n_1^{s_1}} \left\{ -\phi_l(n_1, s_2) + \frac{n_1^{1-s_2}}{s_2 - 1} - \frac{1}{2n_1^{s_2}} + \sum_{q=1}^{l} \frac{(s_2)_q a_q}{n_1^{s_2+q}} \right\} \\
&= \frac{\zeta(s_1 + s_2 - 1)}{s_2 - 1} - \frac{\zeta(s_1 + s_2)}{2} \\
&\quad + \sum_{q=1}^{l} (s_2)_q a_q \zeta(s_1 + s_2 + q) - \sum_{n_1=1}^{\infty} \frac{\phi_l(n_1, s_2)}{n_1^{s_1}} \qquad (3)
\end{aligned}$$

holds for $\Re(s_i) > 1$ $(i = 1, 2)$. The terms on the right hand side have meromorphic continuations except the last one. The last sum is absolutely convergent, and hence holomorphic, in $\Re(s_1 + s_2 + l) > 0$. Thus we now have a meromorphic continuation of $\zeta_2(s_1, s_2)$ to $\Re(s_1 + s_2 + l) > 0$. Since we can choose arbitrary large l, we get a meromorphic continuation of $\zeta_2(s_1, s_2)$ to $\mathbb{C}^2$, which is holomorphic in

$$\left\{ (s_1, s_2) \in \mathbb{C}^2 \mid s_2 \neq 1, s_1 + s_2 \notin \{2, 1, 0, -2, -4, -6, \dots\} \right\}.$$

One can see easily that this trick can be applied to a multiple zeta-function with k variables. In fact,

$$\begin{aligned}
&\zeta_k(s_1, s_2, \dots, s_k) = \sum_{n_1=1}^{\infty} \frac{1}{n_1^{s_1}} \sum_{n_2=n_1+1}^{\infty} \frac{1}{n_2^{s_2}} \cdots \sum_{n_{k-1}=n_{k-2}+1}^{\infty} \frac{1}{n_{k-1}^{s_{k-1}}} \sum_{n_k=n_{k-1}+1}^{\infty} \frac{1}{n_k^{s_k}} \\
&= \sum_{n_1=1}^{\infty} \frac{1}{n_1^{s_1}} \sum_{n_2=n_1+1}^{\infty} \frac{1}{n_2^{s_2}} \cdots \sum_{n_{k-1}=n_{k-2}+1}^{\infty} \frac{1}{n_{k-1}^{s_{k-1}}} \times \\
&\qquad \times \left\{ -\phi_l(n_{k-1}, s_k) + \frac{n_{k-1}^{1-s_k}}{s_k - 1} - \frac{1}{2n_{k-1}^{s_k}} + \sum_{q=1}^{l} \frac{(s_k)_q a_q}{n_{k-1}^{s_k+q}} \right\} \\
&= \frac{\zeta_{k-1}(s_1, s_2, \dots, s_{k-2}, s_{k-1} + s_k - 1)}{s_k - 1} - \frac{\zeta_{k-1}(s_1, s_2, \dots, s_{k-2}, s_{k-1} + s_k)}{2} \\
&\quad + \sum_{q=1}^{l} (s_k)_q a_q \zeta_{k-1}(s_1, s_2, \dots, s_{k-2}, s_{k-1} + s_k + q) \\
&\quad - \sum_{0 < n_1 < n_2 < \dots < n_{k-1}} \frac{\phi_l(n_{k-1}, s_k)}{n_1^{s_1} n_2^{s_2} \dots n_{k-1}^{s_{k-1}}} \qquad (4)
\end{aligned}$$

for $\Re(s_i) > 1$ $(i = 1, 2, \dots, k)$. Since

$$\sum_{0<n_1<n_2<\ldots<n_{k-1}} \frac{\phi_l(n_{k-1}, s_k)}{n_1^{s_1} n_2^{s_2} \dots n_{k-1}^{s_{k-1}}} \ll \sum_{n_{k-1}} \frac{n_{k-1}^{-l-\Re(s_k)+k-3}}{n_{k-1}^{L}}$$

with $L = \Re(s_{k-1}) + \sum_{\substack{1\le j\le k-2,\\ \Re(s_i)\le 0}} \Re(s_i)$, the last summation is convergent absolutely in

$$l - k + 2 + \Re(s_k) + \Re(s_{k-1}) + \sum_{\substack{1\le i\le k-2,\\ \Re(s_i)\le 0}} \Re(s_i) > 0. \tag{5}$$

Since l can be taken arbitrarily large, we get an analytic continuation of $\zeta_k(s_1, s_2, \dots, s_k)$ to $\mathbb{C}^k$. Now we consider the set of singularities. For simplicity, we put $(s)_0 = 1$. Then the 'singular part' of $\zeta_2(s_1, s_2)$ is given by

$$\frac{\zeta(s_1 + s_2 - 1)}{s_2 - 1} + \sum_{q_1\ge 0} \frac{a_{q_1}(s_2)_{q_1}}{s_1 + s_2 + q_1 - 1}.$$

Note that this sum is formal and only indicates local singularities. From this expression, we see

$$s_2 = 1, \quad s_1 + s_2 \in \{2, 1, 0, -2, -4, -6, \dots\}$$

forms the set of whole singularities. For the case $\zeta_3(s_1, s_2, s_3)$, by using the singular part of ζ_2, we see that singularities lie on

$$s_3 = 1, \quad s_2 + s_3 \in \{2, 1, 0, -2, -4, -6, \dots\}$$

and

$$s_1 + s_2 + s_3 \in \{3, 2, 1, 0, -1, -2, -3, \dots\}.$$

We want to show that these are the whole singularities. It suffices to show that no singularities defined by one of above equations will identically vanish. This can be shown by replacing variables:

$$u_1 = s_1, \quad u_2 = s_2 + s_3, \quad u_3 = s_3.$$

In fact, we see that the singular part of $\zeta_3(u_1, u_2 - u_3, u_3)$ is given by

$$\frac{1}{u_3 - 1}\zeta_2(u_1, u_2 - 1) + \sum_{q_2\ge 0} (u_3)_{q_2} a_{q_2} \zeta_2(u_1, u_2 + q_2).$$

By this expression we see that the singularities of $\zeta_2(u_1, u_2 + q)$ are summed with functions of u_3 of different degree. Thus these singularities, as a weighted sum by another variable u_3, will not vanish identically. Similarly, we see

Theorem 1. *The multiple zeta-function* $\zeta_k(s_1, s_2, \dots, s_k)$ *is meromorphically continued to* $\mathbb{C}^k$ *and has singularities on*

$$s_k = 1, \quad s_{k-1} + s_k = 2, 1, 0, -2, -4, \dots,$$

and

$$\sum_{i=1}^{j} s_{k-i+1} \in \mathbb{Z}_{\leq j}\,(j = 3, 4, \dots, k),$$

where $\mathbb{Z}_{\leq j}$ *is the set of integers less than or equal to* j.

3. Zeta values at non-positive integers

In this section, we use the notation $(s)_0 = 1$ and $(s)_{-1} = 1/(s-1)$ for the sake of simplicity. We also put $a_q = B_{q+1}/(q+1)!$ for $q = 0$ and -1 as in §2. A point of $\mathbb{C}^n$ $(n \geq 2)$ is said to be *a point of indeterminacy* of a meromorphic function if both the local denominator and the local numerator vanish there. See p.164 of [11] for the precise definition. For instance, let $f(s_1, s_2) = s_1/(s_1 + s_2)$. Then $s_1 = s_2 = 0$ is a point of indeterminacy of f. So the value of f at $(0, 0)$ depends on a limiting process, for example $\lim_{s_2 \to 0} \lim_{s_1 \to 0} f(s_1, s_2) = 0$ while $\lim_{s_1 \to 0} \lim_{s_2 \to 0} f(s_1, s_2) = 1$.

Let r_i $(i = 1, 2, \dots, k)$ be non-negative integers. Recall from Theorem 1 that each point $(-r_1, -r_2, \dots, -r_k)$ except when $k = 2$ and $r_1 + r_2$ is odd, lies on the set of singularities. Moreover, such a point is a point of indeterminacy. To prove this, it suffices to show that ζ_k has a finite value at $(-r_1, -r_2, \dots, -r_k)$ by a specific limiting process. Now we give the definition which we will employ in this paper.

Definition . We define the multiple zeta values at non-positive integers by

$$\zeta_k(-r_1, -r_2, \dots, -r_k) = \lim_{s_1 \to -r_1} \lim_{s_2 \to -r_2} \cdots \lim_{s_k \to -r_k} \zeta_k(s_1, s_1, \dots, s_k).$$

From (4) and the above definition, we have

$$\begin{aligned} &\zeta_k(-r_1, -r_2, \dots, -r_k) \\ &= \sum_{q=-1}^{r_k} (-r_k)_q a_q \zeta_{k-1}(-r_1, -r_2, \dots, -r_{k-2}, -r_{k-1} - r_k + q). \end{aligned} \tag{6}$$

Here we used the fact that $\phi_r(m, l) = 0$ for $l \geq r$. This formula shows that the value $\zeta_k(-r_1, -r_2, \dots, -r_k)$ is determined recursively as a finite number, hence each point $(-r_1, -r_2, \dots, -r_k)$ is a point of indeterminacy. The formula (6) also gives us a simple way of calculation of multiple zeta values $\zeta_k(-r_1, -r_2, \dots, -r_k)$. For example, we have $\zeta_2(0, 0) = 1/3$, $\zeta_3(0, 0, 0) = -1/4$, $\zeta_4(0, 0, 0, 0) = 1/5$, $\zeta_2(-1, -1) = 1/360$, $\zeta_3(-1, -1, -1) = 83/30240$. One may expect that $\zeta_k(0, 0, \dots, 0) = (-1)^k/(1 + k)$. This assertion will be

proved in the forthcoming paper [2]. Here we shall show some other interesting properties.

Theorem 2. *Let r_i $(i = 1, 2, \dots, k)$ be non-negative integers. Then the value $\zeta_k(-r_1, -r_2, \dots, -r_k)$ is a rational number whose denominator has prime factors less than or equal to $1 + k + \sum_{i=1}^{k} r_i$.*

Proof. It is well known that $\zeta(0) = -1/2, \zeta(-2r) = 0$ and $\zeta(1-2r) = -B_{2r}/2r$ for positive integers r. By using the theorem of von Staudt & Clausen, the assertion for $k = 1$ is obvious. From (6), the proof is completed by the induction on k. □

Theorem 3. *Let r_i $(i = 1, 2, \dots, k)$ be positive integers and n_i $(i = 1, 2, \dots, k)$ be non-negative integers. If $\sum_{i=1}^{k}(r_i + n_i + 1)$ is odd then*

$$\sum_{\sigma \in \mathfrak{S}_k} \operatorname{sgn}(\sigma) \zeta_k(-r_{\sigma(1)} - n_1, -r_{\sigma(2)} - n_2, \dots, -r_{\sigma(k)} - n_k) = 0, \tag{7}$$

where $\mathfrak{S}_k$ is the symmetric group of degree k and $\operatorname{sgn}(\sigma)$ is the signature of $\sigma \in \mathfrak{S}_k$

The statement is trivial when r_i are not distinct. We will show some examples when $n_1 = n_2 = n_3 = 0$ before proving the theorem:

$$\begin{aligned}
&\zeta_2(-1,-2) - \zeta_2(-2,-1) = -\frac{1}{240} + \frac{1}{240} = 0,\\
&\zeta_3(-1,-2,-3) + \zeta_3(-2,-3,-1) + \zeta_3(-3,-1,-2)\\
&\qquad - \zeta_3(-1,-3,-2) - \zeta_3(-2,-1,-3) - \zeta_3(-3,-2,-1)\\
&= -\frac{101}{100800} + \frac{149}{302400} + \frac{107}{302400} + \frac{19}{30240} + \frac{17}{43200} - \frac{131}{151200} = 0.
\end{aligned}$$

Proof. In the following, we shall only prove the case $n_1 = n_2 = \ldots = n_k = 0$. The generalization is quite easy and is left to the reader. Let

$$I_k = \left\{ (-r_1, \dots, -r_k) \,\middle|\, r_i \text{ are positive integers and } \textstyle\sum_{i=1}^{k}(r_i + 1) \text{ is odd.} \right\}.$$

For $1 \le a < b \le k$, we define a vector space $\mathfrak{F}_k(a, b)$ consisting of $\mathbb{C}$-valued functions $f(\xi_1, \xi_2, \dots, \xi_k)$ such that

$$\sum_{\sigma \in \mathfrak{S}_{a:b}} \operatorname{sgn}(\sigma) f(-r_{\sigma(1)}, -r_{\sigma(2)}, \dots, -r_{\sigma(k)}) = 0$$

for any $(-r_1, \dots, -r_k) \in I_k$, where $\mathfrak{S}_{a:b}$ is a subgroup of $\mathfrak{S}_k$ whose elements stabilize $\{1, 2, \dots, k\} \setminus \{a, a+1, \dots, b\}$. Our task is to show that the function $\zeta_k(\xi_1, \xi_2, \dots, \xi_k)$ is contained in $\mathfrak{F}_k(1, k)$. Considering the coset decomposition $\mathfrak{S}_k / \mathfrak{S}_{a:b}$, we see $\mathfrak{F}_k(a, b)$ is a subspace of $\mathfrak{F}_k(1, k)$. Thus it

is enough to show that the multiple zeta-function is contained in a sum of subspaces $\mathfrak{F}_k(a,b)$.

First we prove the case $k \leq 3$. The assertion (7) is valid when $k=1$, since $\zeta(-2r)=0$ for any positive integers r. When $k=2$ and $(-r_1,-r_2) \in I_2$, we have

$$\begin{aligned} \zeta_2(-r_1,-r_2) &= \sum_{q=-1}^{r_2} (-r_2)_q a_q \zeta(-r_1-r_2+q) \\ &= -\frac{1}{2}\zeta(-r_1-r_2), \end{aligned} \tag{8}$$

which shows the assertion for $k=2$. When $k=3$ and $(-r_1,-r_2,-r_3) \in I_3$, we have from (8) that

$$\begin{aligned} \zeta_3(-r_1,-r_2,-r_3) &= \sum_{q=-1}^{r_3} (-r_3)_q a_q \zeta_2(-r_1,-r_2-r_3+q) \\ &= -\frac{1}{2}\zeta_2(-r_1,-r_2-r_3) - \frac{1}{2}\sum_{\substack{q=-1 \\ q:\,\text{odd}}}^{r_3} (-r_3)_q a_q \zeta(-r_1-r_2-r_3+q). \end{aligned} \tag{9}$$

Hence $\zeta_3(\xi_1,\xi_2,\xi_3) \in \mathfrak{F}_3(2,3)+\mathfrak{F}_3(1,2)$.

Let $k \geq 3$ and $(-r_1,\ldots,-r_k) \in I_k$. Then by induction on k, we can easily see that

$$\begin{aligned} &\zeta_k(\xi_1,\xi_2,\ldots,\xi_k) + \frac{1}{2}\zeta_{k-1}(\xi_1,\xi_2,\ldots,\xi_{k-2},\xi_{k-1}+\xi_k) \\ &\qquad \in \mathfrak{F}_k(1,2)+\mathfrak{F}_k(2,3)+\ldots+\mathfrak{F}_k(k-2,k-1). \end{aligned} \tag{10}$$

The assertion of the theorem follows immediately from (10). □

Suppose that $k=3$, r_i are non-negative integers, $r_1>0$ and $r_1+r_2+r_3$ is even. Then from (8) and (9), we have

$$\zeta_3(-r_1,-r_2,-r_3) = -\frac{1}{2}\Big\{\zeta_2(-r_1-r_2,-r_3)+\zeta_2(-r_1,-r_2-r_3)\Big\}. \tag{11}$$

One may expect that symmetric expressions like (8) and (11) would give us a deeper understanding of Theorem 3. Further calculation suggests us the following conjecture. To state it, we shall prepare some notation. Let S be the ordered index set $\{1,2,\cdots,k\}$ of k elements and let $\mathcal{D}_l^k$ be the set of all ways of dividing S into l parts. Clearly the set $\mathcal{D}_l^k$ consists of $\binom{k-1}{l-1}$ elements. The element J in $\mathcal{D}_l^k$ can be expressed as

$$J=(1,\cdots,i_1\,|\,i_1+1,\cdots,i_2\,|\,i_2+1,\cdots,i_{l-1}\,|\,i_{l-1}+1,\cdots,k).$$

Let $A = (-r_1, -r_2, \cdots, -r_k)$ be a sequence of k non-positive integers. For $J \in \mathcal{D}_l^k$ as above, we set

$$A^J = (-r_1 - r_2 - \cdots - r_{i_1},\ -r_{i_1+1} - \cdots - r_{i_2}, \cdots, -r_{i_{l-1}+1} - \cdots - r_k)$$

and

$$\zeta_l(A^J) = \zeta_l\big(-r_1 - r_2 - \cdots - r_{i_1},\ -r_{i_1+1} - \cdots - r_{i_2}, \cdots, -r_{i_{l-1}+1} - \cdots - r_k\big).$$

Now we can state our

Conjecture . *Let r_i be non-negative integers, $r_1 > 0$ and $\sum_{i=1}^k (r_i + 1)$ is odd. Let $A = (-r_1, -r_2, \cdots, -r_k)$. Then we have*

$$\zeta_k(A) = -2\sum_{j=1}^{k-1}(2^{j+1} - 1)\frac{B_{j+1}}{j+1}\Big(\sum_{J \in \mathcal{D}_{k-j}^k} \zeta_{k-j}(A^J)\Big). \tag{12}$$

Further discussion [1] will be found in [2]. We would like to note that we could find the conjectural form of (12) by the home page 'Sloane's On-Line Encyclopedia of Integer Sequences'.

Theorem 4. *For a positive integer r, we have*

$$\frac{\zeta(-4r-1)}{\zeta_2(-2r,-2r)} = (2r+1)\binom{4r+2}{2r+1}.$$

Proof. From (6) and the definition of a_q, we have

$$\zeta_2(-2r,-2r) = \frac{B_{4r+2}}{2(2r+1)^2} + \frac{1}{2r+1}\sum_{j=1}^{r}\binom{2r+1}{2j}\frac{B_{2j}B_{4r+2-2j}}{4r+2-2j}.$$

We note the following identity of Bernoulli numbers:

$$2(2r+1)\sum_{j=0}^{r}\binom{2r+1}{2j}\frac{B_{2j}B_{4r+2-2j}}{4r+2-2j} + \frac{((2r+1)!)^2}{(4r+2)!}B_{4r+2} = 0,$$

obtained by putting $m = n = 2r+1$ and $x = 0$ in Apostol [3, p.276, 19 (b)]. Hence,we have

$$\zeta_2(-2r,-2r) = -\frac{1}{2(2r+1)^2}\frac{((2r+1)!)^2}{(4r+2)!}B_{4r+2}.$$

On the other hand, $\zeta(-4r-1) = -B_{4r+2}/(4r+2)$, and this gives the assertion of Theorem 4. □

Finally we want to add several remarks.

[1] Addendum for the revised version: Finally we have succeeded in proving the validity of this Conjecture. See [2] for details.

Remark 2. There are many other possibilities for the definition of multiple zeta values at non-positive integers. For instance, define the value ζ_k^* by

$$\zeta_k^*(-r_1,-r_2,\dots,-r_k)=\lim_{\varepsilon\to 0}\zeta_k(-r_1+\varepsilon,-r_2+\varepsilon,\dots,-r_k+\varepsilon).$$

When $k=2$, this is equivalent to define by

$$\zeta_2^*(-r_1,-r_2)=\sum_{q=-1}^{r_1}(-r_2)_q a_q\zeta(-r_1-r_2+q)+\frac{(-1)^{r_1}r_1!\,r_2!\,a_{r_1+r_2+1}}{2} \tag{13}$$

for non-negative integers r_i $(i=1,2)$. This definition seems to be better than our former definition at least when $k=2,3$. In fact, when r_1+r_2 is odd we have $\zeta_2(-r_1,-r_2)=\zeta_2^*(-r_1,-r_2)$ and

$$\zeta_2^*(-2u_1,-2u_2)+\zeta_2^*(-2u_2,-2u_1)=0$$

for positive integers u_i $(i=1,2)$. Especially we have $\zeta_2^*(-2u,-2u)=0$ with a positive integer u. We can also find a recursive formula for $k=3$ and show that

$$\zeta_3^*(-2u,-2v,-2w)+\zeta_3^*(-2w,-2v,-2u)=0$$

for positive integers u,v,w. However in the general case, it seems difficult to construct a recursive formula like (13), since there exist a lot of singularities to take into account. One may hope that

$$\zeta_k^*(-2u,-2u,\dots,-2u)=0$$

for a positive integer u.

Remark 3. The set of points of indeterminacy forms a $k-2$ dimensional holomorphic subvariety of $\mathbb{C}^k$, by p.166 of [11]. We have shown that each non-positive points $(-r_1,\dots,-r_k)$ are actually on this subvariety, but there are another type of integer points on this set. For instance, it will be shown in [2] that $(-r_1,\dots,-r_{k-1},1)$ is a point of indeterminacy whose multiple zeta value in our sense is rational, when $r_i\in\mathbb{Z}_{\geq 0}$ and not all r_i is zero. Also we have

$$\zeta_3(4,-3,-2)=-\frac{461}{2520}-\frac{\pi^2}{144}+\frac{\pi^4}{45360}+\frac{\zeta(3)}{420}.$$

We could not determined yet the whole set of such integer points.

Remark 4. We can easily apply the Euler-Maclaurin summation formula to more general zeta-functions. For instance, let $\alpha_i>-1$ $(i=1,2,\dots,k)$ be real numbers and χ_i $(i=1,2,\dots,k)$ the Dirichlet characters. Define a function $\xi(s_1,s_2,\dots,s_k)$ for $\Re(s_i)>1$ $(i=1,2,\dots,k)$, by a convergent sum:

$$\sum_{0<n_1<n_2<\dots<n_k}\frac{\chi_1(n_1)\chi_2(n_2)\cdots\chi_k(n_k)}{(n_1+\alpha_1)^{s_1}(n_2+\alpha_2)^{s_2}\dots(n_k+\alpha_k)^{s_k}}.$$

Then ξ is meromorphically continued to $\mathbb{C}^k$. In fact, using binomial series expansion of $(n+\beta)^{-s} = n^{-s}(1+\beta/n)^{-s}$ for each variable, we see that ξ can be expressed in terms of absolutely convergent sums of multiple zeta functions. See [1] for further study of this kind of function.

References

[1] S. Akiyama and H. Ishikawa, An analytic continuation of multiple L-functions and related zeta-functions, to appear in 'Analytic Number Theory' ed. by C.Jia and K.Matsumoto.

[2] S. Akiyama and Y. Tanigawa, Multiple zeta values at non-positive integers, submitted.

[3] T.M. Apostol, *Introduction to Analytic Number Theory*, Springer 1976.

[4] T. Arakawa and M. Kaneko, Multiple zeta values, poly-Bernoulli numbers, and related zeta-functions, *Nagoya Math. J.* **153** (1999), 189–209.

[5] F.V. Atkinson, The mean value of the Riemann zeta-function, *Acta Math.*, **81** (1949), 353–376.

[6] J.M. Borwein, D.M. Bradley, D.J. Broadhurst and P. Lisoněk, Combinatorial aspects of multiple zeta values, *Electron. J. Combin.*, **5** (1998), no. 1, Research Paper 38, 12 pp.

[7] D.J. Broadhurst and D. Kreimer, Association of multiple zeta values with positive knots via Feynman diagrams up to 9 loops, *Phys. Lett.*, B 393 (1997), no. 3-4, 403–412.

[8] H.M. Edwards,*Riemann's Zeta Function*, Academic Press, New York and London 1974.

[9] S. Egami, Reciprocity laws of multiple zeta functions and generalized Dedekind sums, *Analytic number theory and related topics* (Tokyo, 1991), 17–27, World Sci. Publishing, River Edge, NJ, 1993.

[10] S. Egami, *Introduction to multiple zeta function*, Lecture Note at Niigata University (in Japanese), DVI and TeX files are available at http://mathalg.ge.niigata-u.ac.jp/Seminar/Intensive/Egami.html

[11] R.C. Gunning, *Introduction to holomorphic functions of several variables II*, Wadsworth & Brooks/Cole Mathematics Series.

[12] A. Ivić, *The Riemann Zeta-Function*, A Wiley-Interscience Publ. John Wiley & Sons 1985.

[13] M. Katsurada and K. Matsumoto, Asymptotic expansions of the mean values of Dirichlet L-functions. *Math. Z.*, **208** (1991), 23–39.

[14] T.Q.T. Le and J. Murakami, Kontsevich's integral for the Homfly polynomial and relations between values of multiple zeta functions, *Topology Appl.*, **62** (1995), 193–206.

[15] Y. Motohashi, A note on the mean value of the zeta and L-functions. I, *Proc. Japan Acad.*, Ser. A Math. Sci. **61** (1985), 222–224.

[16] Y. Ohno, A generalization of the duality and sum formulas on the multiple zeta values, *J. Number Theory*, **74** (1999), 39–43.

[17] E.C. Titchmarsh (revised by D.R. Heath-Brown), *The theory of the Riemann Zeta-function*, Clarendon Press, Oxford 1986.

[18] D. Zagier, Values of zeta-functions and their applications, *First European Congress of Mathematics*, Vol. II, Birkhäuser, (1994) 210–220

[19] D. Zagier, Periods of modular forms, traces of Hecke operators, and multiple zeta values, *Research into automorphic forms and L functions (in Japanese) (Kyoto, 1992), Sūrikaisekikenkyūsho Kōkyūroku*, **843** (1993), 162–170.

[20] J. Zhao, Analytic continuation of multiple zeta function, Proc. Amer. Math. Soc. **128** (2000), no. 5, 1275–1283.

Shigeki AKIYAMA
Department of Mathematics, Faculty of Science, Niigata University,
Ikarashi 2-8050,Niigata 950-2181, Japan
e-mail: akiyama@mathalg.ge.niigata-u.ac.jp

Shigeki EGAMI
Department of Mechanical and Intelligent Systems Engineering,
Faculty of Engineering, Toyama University
Gofuku 3190, Toyama 930-8555,Japan
e-mail: megami@eng.toyama-u.ac.jp

Yoshio TANIGAWA
Graduate School of Mathematics, Nagoya University
Chikusa-ku, Nagoya 464-8602, Japan
e-mail: tanigawa@math.nagoya-u.ac.jp

Robert P. Dobrow
Department of Mathematical Sciences
The Johns Hopkins University
Baltimore, MD 21218-2689

James Allen Fill
Department of Mathematical Sciences
The Johns Hopkins University
Baltimore, MD 21218-2689

Bitner, J. R. (1979). Heuristics that dynamically organize data structures. *SIAM J. Comp.,* **8** 82–110.

Fill, J. A. (1993). An exact formula for the move-to-front rule for self-organizing lists. Technical Report #529, Department of Mathematical Sciences, The Johns Hopkins University.

Fine, T. (1970). Extrapolation when very little is known about the source. *Info. and Control.*, **16** 331–359.

Hendricks, W. J. (1972). The stationary distribution of an interesting Markov chain. *J. Appl. Probab.*, **9** 231–233.

Hendricks, W. J. (1989). *Self-organizing Markov Chains.* MITRE Corp., McLean, Va.

Hester, J. H. and Hirschberg, D. S. (1985). Self-organizing linear search. *Comp. Surveys,* **17** 295–311.

Kemeny, J. G. and Snell, J. L. (1965). *Finite Markov Chains.* D. Van Nostrand Co., Princeton, N.J.

Knuth, D. E. (1973). *The Art of Computer Programming.* Vols. 1, 3. Addison-Wesley, Reading, Mass.

Mahmoud, H. M. (1992). *Evolution of Random Search Trees.* John Wiley & Sons, Inc., New York.

Phatarfod, R. M. (1991). On the matrix occurring in a linear search problem. *J. Appl. Prob.*, **28** 336–346.

Ruskey, F., and Hu, T. C. (1977). Generating binary trees lexicographically. *SIAM J. Comp.*, **6** 745–758.

Sleator, D. D., and Tarjan, R. E. (1985). Self-adjusting binary search trees. *J. ACM*, **32** 652–686.

Sloane, N. J. A. (1973). *A Handbook of Integer Sequences,* Academic Press, New York.

Trojanowski, A. E. (1978). Ranking and listing algorithms for k-ary trees. *SIAM J. Comp.*, **7** 492–509.

Summing over $T \in B_n$ gives

$$\sum_{m=|R|}^{n} \sum_{\boldsymbol{\sigma}_m : [\boldsymbol{\sigma}_{|R|}]=R} (-1)^{m-|R|} P^{\infty}(\sigma_1, \ldots, \sigma_{|R|}) P^{\infty}(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{|R|+1}) \tau(\boldsymbol{\sigma}_m), \tag{19}$$

where $\tau(\boldsymbol{\sigma}_m) = |\{T \in B_n : \boldsymbol{\sigma}_m \in \Pi_m(T)\}|$.

It is easily seen that $\tau(\boldsymbol{\sigma}_m)$ depends only on the unordered set $[\boldsymbol{\sigma}_m]$ and equals

$$\tau(\boldsymbol{\sigma}_m) = \prod_{i=0}^{m} \beta_{g_i([\boldsymbol{\sigma}_m])}. \tag{20}$$

Therefore (19) equals

$$\sum_{m=|R|}^{n} (-1)^{m-|R|} \sum_{\substack{U \supseteq R \\ |U|=m}} \tau(U) = \sum_{U \supseteq R} (-1)^{|U|-|R|} \tau(U). \tag{21}$$

But $\tau(U)$ is, by (20), precisely the number of trees that fix *at least* the points in U. By Möbius inversion, (21) reduces to $\alpha_n(R)$. ■

Remarks:

1. For $n \geq 2$, the second largest eigenvalue is the sum which leaves out the consecutive pair $\{i, i+1\}$ with the smallest total weight. Its multiplicity (assuming no ties) is $\alpha_2 = 1$.

2. As in the case of MTF, when the weights are uniform the eigenvalues of MTR are the numbers

$$0,\ 1/n,\ 2/n, \ldots, (n-2)/n,\ 1.$$

The multiplicity of the eigenvalue m/n is the number of trees which fix exactly m points.

7 References

Aho, A. V. and Sloane, N. J. A. (1973). Some doubly exponential sequences. *Fibonacci Quarterly*, **11** 429–437.

Allen, B. and Munro, I. (1978). Self-organizing binary search trees. *J. ACM*, **25** 526–535.

short, α_n is the number of admissible closeness relations (in Fine's terminology) on $[n]$. Fine gave a method of calculating α_n but did not produce an explicit formula like our (15).

3. It is easy to show that α_n satisfies the following recursive relationship with respect to β_n:

$$\alpha_n = \frac{1}{2}(\beta_n - \alpha_{n-1}),\ n \geq 1; \tag{17}$$

furthermore, β_n satisfies

$$\beta_n = \frac{2(2n-1)}{n+1}\beta_{n-1}, \quad n \geq 1. \tag{18}$$

We feel that the simplest method for calculating α_n is to calculate β_n iteratively and then use (17) iteratively to get α_n.

4. Combining (17) and (18) gives a simple recurrence relation satisfied by (α_n):

$$2(n+1)\alpha_n = (7n-5)\alpha_{n-1} + 2(2n-1)\alpha_{n-2}, \quad n \geq 2.$$

We now give a tree-based description of the spectral structure of MTR.

Theorem 5 *The transition matrix for MTR is diagonalizable. The eigenvalues of Q are those values*

$$\lambda_R := p(R) = \sum_{j \in R} p_j$$

for which R has no gaps of size 1. The multiplicity μ_R of λ_R is the number of n-node trees which fix exactly those points in R. That is,

$$\mu_R = \alpha_n(R),$$

which can be computed directly from (13) and the formula for the number of derangement trees.

Proof We identify the eigenvalues and their multiplicities by calculating the trace of Q^k. Consider formula (7). When $S = T$, $d(S, T; R') = 1$ for all $R' \subseteq [n]$ and the coefficient of $(p(R))^k$ simplifies to

$$\sum_{m=|R|}^{n} \sum_{\substack{\boldsymbol{\sigma}_m \in \Pi_m(T): \\ [\boldsymbol{\sigma}_{|R|}]=R}} (-1)^{m-|R|}\, P^{\infty}(\sigma_1, \ldots, \sigma_{|R|})\, P^{\infty}(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{|R|+1}).$$

It now follows by iteration that

$$\alpha_n(R) = \prod_{i=0}^{|R|} \alpha_{g_i(R)}. \tag{13}$$

Similarly, with the conventions $\alpha_0 = \beta_0 = 1$,

Lemma 6.1 *Let $\beta_n = \binom{2n}{n}/(n+1)$ denote the number of binary search trees on n nodes. Then (β_n) satisfies the following recursive relationship with respect to (α_n):*

$$\beta_n = \alpha_n + \sum_{j=1}^{n} \alpha_{j-1}\beta_{n-j}, \quad n \geq 1. \tag{14}$$

Corollary 6.1 *The following formula gives the number of derangement trees on n nodes:*

$$\alpha_n = \frac{1}{2}\left[\left(-\frac{1}{2}\right)^n + \sum_{j=0}^{n}\left(-\frac{1}{2}\right)^j \beta_{n-j}\right], \quad n \geq 0. \tag{15}$$

Proof Recall that the generating function for the nth Catalan number β_n is

$$\mathcal{B}(z) = \frac{1-\sqrt{1-4z}}{2z}.$$

From (14) it follows that the generating function for the number of derangement trees is

$$\mathcal{A}(z) = \frac{\mathcal{B}(z)}{1+z\mathcal{B}(z)} = \frac{1+\mathcal{B}(z)}{2+z}, \tag{16}$$

and the result follows by computing the coefficient of z^n. ■

Remarks:

1. Values of α_n up through $n = 21$ can be found in Sloane (1973), sequence number 635. The first 10 numbers, starting with α_1, are: 0, 1, 2, 6, 18, 57, 186, 622, 2120, 7338.

2. The sequence (α_n) has arisen in the context of Fine's (1970) work on closeness relations. We shall not go into detail about the connections. In

Phatarfod (1991) derived the eigenvalues and multiplicities for MTF. Suppose for simplicity throughout this section that sums of distinct collections of weights are distinct. Then the eigenvalues are all the partial sums of the weights, excluding the n cases where the summation is over $n-1$ weights. The multiplicity of each eigenvalue $\lambda_R = \sum_{j \in R} p_j$ corresponding to a sum of $|R| = m$ weights is the number of permutations in S_n fixing exactly those points in R, namely, the number of derangements (permutations with no fixed points) in S_{n-m}.

Our results for MTR exhibit an interesting parallelism to those for MTF. In brief, we shall define the notions of unit gap and fixed point of a tree and show (i) that the eigenvalues for MTR are the partial sums of weights excluding sets which have unit gaps, and (ii) that the multiplicity of the eigenvalue λ_R is the number of trees in B_n fixing exactly those points in R.

For $R \subseteq [n]$, write $r_1 < r_2 < \cdots < r_m$ for the elements of R. Define $r_0 := 0$ and $r_{m+1} := n+1$. Let

$$g_i(R) := r_{i+1} - r_i - 1, \quad i = 0, \ldots, m,$$

denote the number of integers in the interval (r_i, r_{i+1}). Then $g_i(R)$ is called the *i-th gap* of R.

We say that a tree T *fixes* a record j if the records $j+1, \ldots, n$ are all in the right subtree of j and the left subtree of j is empty. Equivalently, T fixes j if there exists $\pi \in \Pi(T)$ such that $\pi(j) = j$ and π maps $\{1, \ldots, j-1\}$ to itself and $\{j+1, \ldots, n\}$ to itself.

We say that a tree fixes a set of records R if the tree fixes each of the records in R. Denote the number of trees which fix exactly R by $\alpha_n(R)$. We call a tree which fixes none of its records a *derangement tree.* Write $\alpha_n := \alpha_n(\emptyset)$ for the number of n-node derangement trees.

Note that if a tree T fixes exactly one record j, then the nodes of T which contain records $1, \ldots, j-1$ form a derangement tree. Similarly, the nodes of T which contain records $j+1, \ldots, n$ also form a derangement tree. Conversely, any derangement tree on $\{1, \ldots, j-1\}$ can be joined with any derangement tree on $\{j+1, \ldots, n\}$ to obtain a tree with j as its unique fixed point.

Remarks:

1. When T is "long and skinny"—that is, when T is "close" to the tree obtained by the identity or reversal permutation—$Q^k(S,T)$ can be computed in time polynomial in n using any of the methods we have discussed. For example, one can use (1) and the formula from Fill (1993) for the MTF transition probability $P^k(\pi,\sigma)$. The latter can be computed in polynomial time for fixed $\sigma \in S_n$, and it is not hard to show that if $T \in B_n$ has height $n-1-k$, then $N(T) = |\Pi(T)| \leq n^k$.

2. Let $u(T)$ denote the number of uptrees for tree T. Thus $u(T)$ is the number of terms in the sum in (8). Then u satisfies the recursion

$$u(T) = 1 + u(L(T))u(R(T)). \tag{11}$$

For example, for the perfect binary tree on $n = 2^m - 1$ nodes let u_m denote the number of uptrees. Then

$$u_{m+1} = u_m^2 + 1, \ m \geq 0, \tag{12}$$

which generates the sequence $1, 2, 5, 26, 677, 458330, 210066388901, \ldots$.

Note that u_m is the number of binary search trees with height at most $m-1$ and (12) has been studied from this point of view. While no closed form solution to (12) is known, one can show that $u_m = \lfloor K^{2^m} \rfloor = \lfloor K^{n+1} \rfloor$ where K is approximately 1.502837. (See Aho and Sloane (1973) for a discussion of this and other nonlinear recurrences of the form $x_{n+1} = x_n^2 + g_n$, where g_n is a slowly growing function of n.)

3. One approach to computing $D(S,T;\cdot)$ begins by constructing tables of ancestry relations for S and T. It is easy to see how to construct such tables in time—and space—$O(n^2)$. By constructing an ancestry table as a hash table, a single ancestry relation can be checked in constant time and thus $D(S,T;R)$ computed in time $O((n-|R|)^2) = O(n^2)$ for fixed S, T, R.

6 Eigenanalysis of MTR

The fact that MTF is lumpable gives us little to go on in trying to determine the eigenstructure of MTR. From lumpability it follows that the eigenvalues for MTR are some subset of those for MTF. But determining *which* subset and the corresponding multiplicities requires more detailed analysis.

programming approach. Beginning with $\mathcal{Q}_\emptyset(z) = 1$, (9) yields $\mathcal{Q}_{\{x\}}(z) = e^{w_x z} - 1$ for a tree $\{x\}$ of height 0. Having computed $\mathcal{Q}_U$ for all trees U with height at most $h-1$, the recursion (9) can be used to find $\mathcal{Q}_U$ for trees U with height h. In each instance, (9) is a first-order linear differential equation involving only linear combinations of exponentials.

For fixed n and $T \in B_n$, the process of solving for $\mathcal{Q}_U$ for all $U \in \mathcal{U}(T)$ can be less tedious. As an illustration, let T be the tree of 3 nodes corresponding to the reversal permutation. (In the notation of Figure 1, $T = T_5$.) The uptrees of T are the empty tree, the singleton tree T' storing 3, the tree T'' induced by records 2 and 3, and T itself. We have

$$\begin{aligned}
\mathcal{Q}_\emptyset(z) &= 1, \\
\mathcal{Q}_{T'}(z) &= e^{p_3 z} - 1, \\
\mathcal{Q}_{T''}(z) &= \frac{p_3}{p_2+p_3}(e^{(p_2+p_3)z} - 1) - (e^{p_3 z} - 1), \\
\mathcal{Q}_T(z) &= \frac{p_2 p_3}{p_1+p_2}(e^z - 1) - \frac{p_3}{p_2+p_3}(e^{(p_2+p_3)z} - 1) + \frac{p_1}{p_1+p_2}(e^{p_3 z} - 1).
\end{aligned}$$

Solving for the coefficients in the generating functions gives, for $k \geq 0$,

$$\begin{aligned}
Q_k(\emptyset) &= \delta_{0k}, \\
Q_k(T') &= p_3^k - \delta_{0k}, \\
Q_k(T'') &= \frac{p_3}{p_2+p_3}((p_2+p_3)^k - \delta_{0k}) - (p_3^k - \delta_{0k}), \\
Q_k(T) &= \frac{p_2 p_3}{p_1+p_2}(1 - \delta_{0k}) - \frac{p_3}{p_2+p_3}((p_2+p_3)^k - \delta_{0k}) + \frac{p_1}{p_1+p_2}(p_3^k - \delta_{0k}),
\end{aligned}$$

where δ_{ij} equals 1 if $i = j$ and 0 otherwise.

Now suppose $S \in B_3$ corresponds to the identity permutation. (In terms of Figure 1, $S = T_1$.) Then

$$\begin{aligned}
D(S,T;\emptyset) &= D(S,T;\{3\}) = 0 \quad \text{and} \\
D(S,T;\{2,3\}) &= D(S,T;\{1,2,3\}) = 1,
\end{aligned}$$

and so

$$\begin{aligned}
Q^k(S,T) &= Q_k(T'') + Q_k(T) \\
&= \begin{cases} \frac{p_2 p_3}{p_1+p_2}(1 - p_3^{k-1}) & \text{if } k \geq 1 \\ 0 & \text{if } k = 0. \end{cases}
\end{aligned}$$

exactly $|U|$ distinct records to the root, with these $|U|$ records forming the tree U as a result.

We will derive a recursive (in U) functional relationship for the exponential generating function $\mathcal{Q}_U(z) := \sum_{k=0}^{\infty} Q_k(U) z^k/k!$. Its solution will give a straightforward method for computing the k-step probabilities simultaneously for *all* k. In the remarks at the end of this section we will discuss issues related to the complexity of the calculations.

Theorem 4 *Let U be a binary search tree. Let $\mathcal{Q}_U(z) := \sum_{k=0}^{\infty} Q_k(U) z^k/k!$ be the exponential generating function of the sequence $(Q_k(U))_{k\geq 0}$. Define $\mathcal{Q}_\emptyset(z) := 1$. Then*

$$\mathcal{Q}'_U(z) = w_{\mathrm{root}(U)} e^{w_{\mathrm{root}(U)} z} \mathcal{Q}_{L(U)}(z) \mathcal{Q}_{R(U)}(z), \tag{9}$$

with the initial condition

$$\mathcal{Q}_U(0) = \begin{cases} 1 & \text{if } U = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Proof For $k \geq 1$, $Q_k(U)$ is the probability that in k requests: (a) the last request is for root(U); (b) the request for records in $L(U)$ are such that after the k steps they form $L(U)$; and (c) the request for records in $R(U)$ are such that after the k steps they form $R(U)$. Thus, for $k \geq 1$,

$$Q_k(U) = w_{\mathrm{root}(U)} \sum_{j_1, j_2, j_3} \binom{k-1}{j_1, j_2, j_3} w_{\mathrm{root}(U)}^{j_1} Q_{j_2}(L(U)) Q_{j_3}(R(U)), \tag{10}$$

where the sum is over all non-negative triples which sum to $k-1$. For $k = 0$ it is clear that

$$Q_0(U) = \begin{cases} 1 & \text{if } U = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Multiplying both sides of (10) by $z^{k-1}/(k-1)!$ and summing from 1 to ∞ gives the result. ∎

We do not know a tree-based closed-form solution to (9) (except in the case of equal weights). The process of solving (9) for $\mathcal{Q}_U(z)$ for all trees U with height at most h is best implemented using a "bottom-up" dynamic

Remarks:

1. From (5) and rearrangement, (6) can alternatively be written as

$$\begin{aligned} Q^k(S,T) \\ &= \sum_{R\subset[n]} (p(R))^k \sum_{m=|R|}^{n} \sum_{\substack{\boldsymbol{\sigma}_m\in\Pi_m(T):\\ [\boldsymbol{\sigma}_{|R|}]=R}} D(S,T;[\boldsymbol{\sigma}_m]) \\ &\qquad \times (-1)^{m-|R|}\, P^\infty(\sigma_1,\ldots,\sigma_{|R|})\, P^\infty(\sigma_m,\sigma_{m-1},\ldots,\sigma_{|R|+1}), \end{aligned} \tag{7}$$

where $p(R) := \sum_{i\in R} p_i$. This form of Q^k will be useful for the spectral analysis of MTR given in Section 6.

2. From (7) we can derive the stationary distribution as given in (2). Let $k\to\infty$ and note that the only term in the outer sum which doesn't vanish is the one corresponding to $R=[n]$. This gives $Q^\infty(T) = \sum_{\sigma\in\Pi(T)} P^\infty(\sigma)$.

3. In the case of equal weights ($p_i \equiv 1/n$),

$$P^k(\boldsymbol{\sigma}_m) = \sum_{i=0}^{m} \left(\frac{i}{n}\right)^k \frac{(-1)^{m-i}}{i!(m-i)!} =: P^k(m).$$

Thus

$$Q^k(S,T) = \sum_{m=0}^{n} P^k(m) C_m(S,T),$$

where $C_m(S,T) := |\{\boldsymbol{\sigma}_m \in \Pi_m(T) : D(S,T;[\boldsymbol{\sigma}_m]) = 1\}|$.

5.2 Computation of k-step probabilities

While formula (6) is useful for deriving certain characteristics of the MTR chain, we next consider a version that seems better suited for numerical computations. For any tree T, let $\mathrm{rec}(T)$ be the set of records stored at the nodes of T. By rearranging (6) we find that for $S,T\in B_n$,

$$Q^k(S,T) = \sum_{U\in\mathcal{U}(T)} D(S,T;\mathrm{rec}(U)) Q_k(U), \tag{8}$$

where $\mathcal{U}(T)$ is the collection of uptrees of T and $Q_k(U) := \sum_{\tau\in\Pi(U)} P^k(\tau)$. Observe that $Q_k(U)$ is the probability that k requests using MTR move

$$P^k(\boldsymbol{\sigma}_m) = w_m^* \sum_{i=0}^{m} (w_i^+)^k w_{m,i},$$

where, for $0 \le i \le m \le n$,

$$w_i^+ := \sum_{h=1}^{i} w_h, \quad w_i^* := \prod_{h=1}^{i} w_h, \quad \textit{and} \quad w_{m,i} := 1/ \prod_{\substack{j \ne i \\ 0 \le j \le m}} (w_i^+ - w_j^+),$$

with the natural conventions $w_0^+ := 0$, $w_0^* := 1$, *and* $w_{0,0} := 1$.

Proof Noting that the top m records must have their last requests occur in the order $\sigma_m, \sigma_{m-1}, \ldots, \sigma_1$, and conditioning on the times of these requests, we find

$$P^k(\boldsymbol{\sigma}_m) = \sum_{\mathbf{j}_m} (w_m^+)^{j_m} w_m (w_{m-1}^+)^{j_{m-1}} w_{m-1} \cdots (w_1^+)^{j_1} w_1 = w_m^* \sum_{\mathbf{j}_m} \prod_{r=1}^{m} (w_r^+)^{j_r},$$

where the sum is over all m-tuples of nonnegative integers summing to $k-m$. The result follows from an algebraic identity derived in the Appendix of Fill (1993). ■

As discussed in Remark 2.2(a) of Fill (1993), $P^k(\boldsymbol{\sigma}_m)$ can also be written in the form

$$P^k(\boldsymbol{\sigma}_m) = \sum_{i=0}^{m} (w_i^+)^k \, (-1)^{m-i} \, P^\infty(\sigma_1, \ldots, \sigma_i) \, P^\infty(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{i+1}), \quad (5)$$

where $P^\infty(\sigma_1, \ldots, \sigma_i)$ is the probability that sampling without replacement from $[n]$ selects the elements of $\{\sigma_1, \ldots, \sigma_i\}$ in the relative order $(\sigma_1, \ldots, \sigma_i)$; similarly for $P^\infty(\sigma_m, \sigma_{m-1}, \ldots, \sigma_{i+1})$.

The main Theorem 3 now follows directly:

Theorem 3 *Let* $S, T \in B_n$. *Then*

$$Q^k(S,T) = \sum_{m=0}^{n} \sum_{\boldsymbol{\sigma}_m \in \Pi_m(T)} D(S,T;[\boldsymbol{\sigma}_m]) P^k(\boldsymbol{\sigma}_m), \quad (6)$$

where

$$D(S,T;[\boldsymbol{\sigma}_m]) = \prod_{j=0}^{m} d(S,T;(\sigma_{(j)}, \sigma_{(j+1)})).$$

of the event that the ancestry relations of the two trees agree for the records in R. That is, $d(S,T;R) = 1$ if $i <_a^T j$ exactly when $i <_a^S j$ for all $i, j \in R$, and $d(S,T;R) = 0$ otherwise.

For a permutation $\sigma \in S_n$, let $\boldsymbol{\sigma}_m := (\sigma_1, \ldots, \sigma_m)$ for $1 \le m \le n$. Thus $\boldsymbol{\sigma}_m$ is the projection of σ onto its first m coordinates. Recall the definition of $\Pi(T)$ given in Section 2. Let $\Pi_m(T)$ be the projection of the elements of $\Pi(T)$ onto their first m coordinates. Thus $\Pi_n(T) = \Pi(T)$ and $\Pi_1(T)$ is the singleton $\{\mathrm{root}(T)\}$. Finally, let $[\boldsymbol{\sigma}_m]$ denote the *unordered* set $\{\sigma_1, \ldots, \sigma_m\}$ and $\sigma_{(1)} < \cdots < \sigma_{(m)}$ the corresponding order statistics with $\sigma_{(0)} := 0$ and $\sigma_{(m+1)} := n + 1$.

An *upset* in a tree $T \in B_n$ is a set U of nodes with the property that if $j \in U$, then the parent (equivalently, all ancestors) of j is in U. Note that the graph in T induced by an upset in T is itself a tree containing (if nonempty) the root of T. We shall refer to this induced tree as the *uptree* U.

It follows from the discussion of the tree-building operation in Section 2 that the uptrees of T consisting of m elements are precisely the trees $t(\boldsymbol{\sigma}_m)$ with $\boldsymbol{\sigma}_m \in \Pi_m(T)$. The discussion in Sections 2 and 3, especially the proof of Lemma 3.1, also yields the following lemma. We leave the simple proof to the reader.

Lemma 5.1 *Consider a sequence Σ of k record requests that contains m distinct records. Suppose that application of Σ to the list $(1, \ldots, n)$ using MTF results in $\boldsymbol{\sigma}_m = (\sigma_1, \ldots, \sigma_m)$ as the m-tuple of frontmost elements. Then application of Σ to a given tree $S \in B_n$ using MTR results in the tree $T \in B_n$ characterized by the following two statements:*

(a) The tree $t(\boldsymbol{\sigma}_m)$ is an uptree of T.

(b) For each $j = 0, \ldots, m$, the T-ancestry relations among the records in $(\sigma_{(j)}, \sigma_{(j+1)})$ are the same as in S.

Here we use the notation (a, b) for integers a and b to mean the interval of integers strictly between a and b. Note that we take the initial list in Lemma 5.1 to be $(1, \ldots, n)$ only for definiteness. The same result clearly holds for any initial permutation π.

Next we reproduce a result from Fill (1993) concerning MTF:

Lemma 5.2 *Let $P^k(\boldsymbol{\sigma}_m)$ denote the probability, starting in the list $(1, \ldots, n)$, that k requests using MTF move exactly m distinct records to the front and result in $\boldsymbol{\sigma}_m$ as the m-tuple of frontmost elements. Then*

and partial product for a node after these quantities have been calculated for its children.

3. The distribution (4) arises in the study of random trees. It is the distribution of $t(\sigma)$, where $\sigma \in S_n$ is uniformly distributed. See Mahmoud (1992). The distribution (3) is the distribution of $t(\sigma)$, where $\sigma \in S_n$ has the weighted-sampling-without-replacement stationary distribution of MTF, and so is a generalization of the random permutation model.

4. Unlike the uniform distribution on B_n, the distribution (4) favors trees which are "short and fat." Suppose for ease of di scussion that $n = 2^m - 1$ for integer m. The perfect binary tree is the tree for which all nodes, except for leaves, have 2 children. Call this tree T_m. We can show that the mode of (4) is T_m. It is not hard to derive the asymptotic behavior of $Q^\infty(T_m)$. In particular, the rate of decay for $Q^\infty(T_m)$ is exponential in n. In contrast, $\min_{T \in B_n} Q^\infty(T) = Q^\infty(t(1, \ldots, n)) = 1/n!$ decays at a superexponential rate.

5 Transition probabilities

5.1 A tree-based approach

Our goal in this section is to derive a formula for the k-step transition probabilities $Q^k(S, T)$, where $S, T \in B_n$. The k-step probabilities for MTF were derived by Fill (1993). Thus in light of Corollary 3.1 it would seem that we are done.

Fill's formula, however, is necessarily permutation-based. It depends, for instance, on permutation statistics which are not invariant under the mapping Π. And while the MTF probabilities can be computed in polynomial time, the number of summands in (1) is $N(T)$, which by the pigeonhole principle is, for some T, at least $n!/|B_n| \sim \pi\sqrt{2}\, n^{n+2}(4e)^{-n}$.

The formulas (6) and (8) below have the advantage that they are, at least partially, "tree-based" and can be used to derive numerous characteristics of the chain, including (see Section 6) the eigenvalues and their multiplicities.

Before proceeding to the main theorem of this section (Theorem 3) we establish some notation and preliminary results. It will be necessary to distinguish between the nodes in a tree and the records stored there. Let $R \subseteq [n]$ be a subset of records. For $S, T \in B_n$, define $d(S, T; R)$ to be the indicator

The stationary distribution for MTF, originally derived by Hendricks (1972), is given by

$$P^{\infty}(\sigma) = \prod_{i=1}^{n} \frac{w_i}{\sum_{j=i}^{n} w_j}.$$

Observe that P^{∞} is the distribution of the order obtained by sampling n items without replacement. It follows from Corollary 3.2 that $Q^{\infty}(T)$ is the probability of sampling n items without replacement in such a way that the first item is at the root of T and the order of choosing the remaining items is consistent with the ancestry relations in $L(T)$ and $R(T)$. Since the root and the two subtrees partition the n items, (3) follows by recursion. ■

As a corollary, we obtain the number of terms in the sum (2).

Corollary 4.1 *For a tree T, let $N(T) = |\Pi(T)|$. Then*

$$N(T) = \binom{|T|-1}{|L(T)|} N(L(T))N(R(T)) = \frac{|T|!}{\prod_{x \in T} |T_x|},$$

where $|T|$ is the number of nodes of T.

Proof The first equation follows from the recursive argument in the proof above. Now iterate to obtain the second equation. ■

Corollary 4.2 *Let T be a nonempty binary search tree. Under MTR if records are accessed uniformly (each with probabily $1/|T|$), then*

$$Q^{\infty}(T) = \frac{1}{\prod_{x \in T} |T_x|}. \tag{4}$$

Remarks:

1. Another way to think about Corollary 4.1 is with respect to partial orderings. The lemma gives the number of linear extensions for a set of elements in a partial order which satisfy some given relations. These relations, of course, must be consistent with the relations satisfied by a binary tree.

2. Computing (3) in linear time requires a simple algorithm which starts at a leaf, working its way up the tree, iteratively computing the partial sum

$P(\pi', \sigma') = p_{\pi'_{k'}} = p_{\pi_k}$. By the previous lemma, $t(\sigma')$ is identical to the tree obtained by moving $\pi'_{k'} = \pi_k$ to the root in S. But since $\pi \in \Pi(S)$, it follows that $t(\sigma)$ is identical to the tree obtained by moving π_k to the root in S. That is, $t(\sigma) = t(\sigma')$, so $\sigma' \in \Pi(T)$ and hence

$$\sum_{\tau \in \Pi(T)} P(\pi', \tau) = P(\pi', \sigma') = P(\pi, \sigma) = p_{\pi_k}.$$

It follows from the foregoing that if $\pi \in \Pi(S)$ and $P(\pi, \sigma) = 0$ for all $\sigma \in \Pi(T)$, then for each $\pi' \in \Pi(S)$, $P(\pi', \sigma) = 0$ for all $\sigma \in \Pi(T)$. ■

4 Stationary distribution

While Corollary 3.1 gives an exact formula for the transition probabilities of MTR, explicit calculation of these numbers for specific trees is another matter. In the case of the stationary distribution Q^∞, however, exploiting the recursive character of binary trees and using a simple property of sampling without replacement gives a result analogous to that for MTF.

For x a node of a given tree, we use the notation w_x for the probability of accessing the record at that node. For i an index of a given permutation σ, we write w_i for p_{σ_i}.

Theorem 2 *For a tree T,*

$$Q^\infty(T) = \prod_{x \in T} \left(\frac{w_x}{\sum_{y \in T_x} w_y} \right), \tag{3}$$

where T_x is the subtree of T with root x.

Proof For a binary search tree T let $L(T)$ denote the left subtree of T. For a permutation τ let τ^L be the subpermutation of τ induced by the elements of $L(T)$. That is, for $k = 1, \ldots, |L(T)|$, τ^L_k is the kth element of τ which is contained in $L(T)$. Similarly define $R(T)$ and τ^R.

A necessary and sufficient condition for $\tau \in \Pi(T)$ is that the following three conditions hold: (i) τ_1 is the record at the root of T; (ii) $\tau^L \in \Pi(L(T))$; and (iii) $\tau^R \in \Pi(R(T))$.

is obtained by requesting $\sigma_n, \ldots, \sigma_1$. Thus $t(\sigma')$ is obtained from $t(\sigma)$ by moving σ_k to the root. ■

The reader may consult Figure 3 for an illustration of Lemma 3.4. Note that we can read off the ancestry relations of a tree from any one of its associated permutations by looking at the ordering relations. For instance, for $\sigma = (3, 6, 1, 4, 2, 5)$, 3 is an ancestor of everyone; 6 is an ancestor of 4 and 5, but not of 3, 2, or 1 (since 6 is to the left of 4 and 5 but not to the left of 3); 1 is an ancestor of 2, but of no others; 4 is an ancestor of 5 and of no others. This will be true for all equivalent permutations, which include, for instance, $(3, 1, 2, 6, 4, 5)$ and $(3, 1, 6, 4, 5, 2)$.

Figure 3.

We now prove Theorem 1.

Proof Define root(T) to be the record at the root of T. Note that for any $\sigma, \sigma' \in \Pi(T)$, $\sigma_1 = \sigma'_1 =$root(T), since $t(\sigma) = t(\sigma')$. Thus for any fixed π and $T \in B_n$, $P(\pi, \sigma) = 0$ for all but possibly one $\sigma \in \Pi(T)$. If $P(\pi, \sigma) > 0$, then σ is uniquely determined. In particular, since MTF corresponds to composition with a cycle, $\sigma = \pi \circ (k \cdots 1)$, where $k = \pi^{-1}(\text{root}(T))$.

Let $S, T \in B_n$ and $\pi \in \Pi(S)$. Then

$$\sum_{\tau \in \Pi(T)} P(\pi, \tau) = \begin{cases} P(\pi, \sigma) & \text{if there exists } \sigma \in \Pi(T) \text{ such that } P(\pi, \sigma) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The theorem will follow if we can show that the righthand side above is the same for all $\pi' \in \Pi(S)$.

Suppose there exists $\sigma \in \Pi(T)$ such that $P(\pi, \sigma) > 0$. Let $k = \pi^{-1}(\sigma_1)$. Thus $P(\pi, \sigma) = p_{\pi_k} = p_{\sigma_1}$ and σ is just what results from π after moving record π_k to the front. Let $\pi' \in \Pi(S)$ and $k' = \pi'^{-1}(\pi_k)$. Put $\sigma' = \pi' \circ (k' \cdots 1)$. Thus σ' is just what results from π' after moving $\pi'_{k'} = \pi_k$ to the front and

Corollary 3.3 *Given $\sigma \in S_n$, $t(\sigma)$ is the tree obtained from any n-node tree after making the sequence of requests $\sigma_n, \sigma_{n-1}, \ldots, \sigma_1$.*

Proof For any k, after deleting records $\sigma_n, \ldots, \sigma_{k+1}$ from $t(\sigma)$, σ_k will be a leaf. ■

This gives a characterization of $\Pi(T)$. For a set S, we use the notation $a < S$ to mean that a is less than every element of S, i.e., that $a < \min S$.

Lemma 3.3 *For $T \in B_n$,*

$$\Pi(T) = \{\sigma : i <_a^T j \;\; \Leftrightarrow \;\; \sigma^{-1}(i) < \{\sigma^{-1}(i+1), \ldots, \sigma^{-1}(j)\}, \text{ for } i < j, \\ \sigma^{-1}(i) < \{\sigma^{-1}(j), \ldots, \sigma^{-1}(i-1)\}, \text{ for } i > j\}.$$

In words, $\Pi(T)$ is the set of all permutations σ such that i is a T-ancestor of j if and only if i is to the left of $i+1, \ldots, j$ when $i < j$ and to the left of $j, \ldots, i-1$ when $i > j$.

Proof The proof, after sorting through the notation, is a direct consequence of the above lemmas. Call the set on the righthand side $\Pi'(T)$. Let $\sigma \in \Pi(T)$. Then $t(\sigma) = T$, and so, by Corollary 3.3, T can be obtained from any tree by requesting $\sigma_n, \ldots, \sigma_1$. Suppose $i <_a^T j$. Then, by Lemma 3.1, i is requested after $i+1, \ldots, j$, so $\sigma^{-1}(i) < \{\sigma^{-1}(i+1), \ldots, \sigma^{-1}(j)\}$. Similarly, the converse holds, showing that $\Pi(T)$ is contained in $\Pi'(T)$.

Suppose $\sigma \notin \Pi(T)$. Then there exist some i and j such that $i <_a^T j$ but $i \not<_a^{t(\sigma)} j$. Since $\sigma \in \Pi(t(\sigma))$, by the first part of the proof $\sigma^{-1}(i) \not< \{\sigma^{-1}(i+1), \ldots, \sigma^{-1}(j)\}$. Hence $\sigma \notin \Pi'(T)$. ■

A direct consequence is that MTF for permutations corresponds to MTR for the associated binary search trees.

Lemma 3.4 *Fix $\sigma \in S_n$ and let $\sigma' \in S_n$ be the permutation obtained by moving σ_k to the front. Then $t(\sigma')$ is the tree obtained from $t(\sigma)$ by moving σ_k to the root.*

Proof By assumption, $\sigma' = (\sigma_k \sigma_1 \cdots \sigma_{k-1} \sigma_{k+1} \cdots \sigma_n)$. By Corollary 3.3, $t(\sigma')$ is obtained by requesting $\sigma_n, \ldots, \sigma_{k+1}, \sigma_{k-1}, \ldots, \sigma_1, \sigma_k$. But by Lemma 3.1, $t(\sigma')$ is also gotten by requesting $\sigma_n, \ldots, \sigma_1, \sigma_k$. On the other hand, $t(\sigma)$

Lemma 3.1 *Suppose records i and j have been requested at least once each in a tree modified according to MTR. Let $i < j$. Then $i <_a j$ if and only if the most recent request for i has occurred since the most recent request for any of $i+1, \ldots, j$. Similarly, $j <_a i$ if and only if the most recent request for j has occurred since the most recent request for any of $i, \ldots, j-1$.*

Proof When either simple exchange or MTR is used, if i is requested then i is the only record which *becomes* an ancestor of any records. Also, i will *cease* to be an ancestor of j if and only if an element k, where $i < k \leq j$, is requested. This gives the first part of the lemma. The second part is shown similarly. ■

For the remaining proofs in this section, as in the preceding proof, the condition that $i > j$ can be handled in the same fashion as the case $i < j$, so we will tacitly restrict ourselves to the latter.

The binary search tree obtained from some permutation σ by the tree-building process is also, as shown by Corollary 3.3 below, the tree obtained from any other binary search tree by successively requesting records in the reverse order of σ. In this regard, note that any binary search tree can be obtained from any other in at most n operations. (That n steps might be necessary is shown by considering the "degenerate" trees corresponding to the identity and reversal permutations.)

Lemma 3.2 *Let $S, T \in B_n$. Consider the following sequence of operations: Choose a leaf in T and within S move the corresponding record to the root. After the record has been moved in S, delete it from T by eliminating its node and the incident branch. Continue until T is empty. Then, after any such sequence of n moves applied to S, the transformed tree will be identical to T before the operation.*

Proof Let S' be the tree obtained from S after the n moves. If $i <_a^T j$ then any k such that $i < k < j$ will be in the subtree of T with root i. Thus the request for i will come after the requests for $i+1, \ldots, j$ because only then will i become a leaf. Thus $i <_a^{S'} j$ by Lemma 3.1. The result now follows from Lemma 2.1. ■

Theorem 1 *Let Q be the $|B_n| \times |B_n|$ transition matrix for MTR and let P be the $n! \times n!$ transition matrix for MTF. Then for $S, T \in B_n$,*

$$Q(S,T) = \sum_{\sigma \in \Pi(T)} P(\pi, \sigma),$$

where π is any permutation in $\Pi(S)$.

The theorem is equivalent to the statement that the Markov chain corresponding to P is lumpable (see Kemeny and Snell (1965)) with respect to the map t. From the properties of lumpable chains the following corollaries are immediate:

Corollary 3.1

$$Q^k(S,T) = \sum_{\sigma \in \Pi(T)} P^k(\pi, \sigma), \tag{1}$$

for each $k \geq 0$, where π is any permutation in $\Pi(S)$.

Corollary 3.2

$$Q^\infty(T) = \sum_{\tau \in \Pi(T)} P^\infty(\tau), \tag{2}$$

where P^∞ and Q^∞ are the stationary distributions for the MTF and MTR chains, respectively.

Remark: It is easily seen (e.g., from the case $n = 3$) that the Markov chain corresponding to the simple exchange heuristic is not lumpable with respect to the map t. In addition, for general weights the chain is not time-reversible, unlike the chain corresponding to the transposition heuristic for lists. When all the weights are identical ($p_i \equiv 1/n$), the SE transition matrix is symmetric and so the stationary distribution is uniform on B_n. However, for general weights the stationary distribution—not to mention the k-step transition probabilities and the spectral structure of the transition matrix—is unknown.

The proof of Theorem 1 is based on several observations and lemmas, which follow. A key result is Lemma 3.2 in Allen and Munro (1978), which we reproduce:

For $i \neq j$, we say that i is an *ancestor* of j in T, and write $i <_a^T j$, if j is an element of the subtree which has i as its root. We will suppress the superscript if it is obvious to what tree we are referring. Note that $<_a$ defines a partial order on the nodes of T. A tree is uniquely determined by its ancestry relations and, as the next lemma implies, among trees with the same number of nodes the set of ancestry relations for one tree can never be a proper subset of those for another.

Lemma 2.1 *Let $S, T \in B_n$. Then $T = S$ if and only if $i <_a^T j$ implies $i <_a^S j$ for all $i, j \in [n]$.*

Proof Necessity is trivial; sufficiency follows by a simple induction on n using the recursive definition of a tree. ■

3 Main result: lumping

The mappings t and Π between S_n and B_n make it easy to translate tree operations into operations on permutations. In fact we will show (Lemma 3.4) that MTR for a binary search tree T corresponds to MTF for all of the permutations in $\Pi(T)$.

For n-node trees it is easily shown that the sequence of operations generated by MTR gives an ergodic (aperiodic, irreducible, and positive recurrent) Markov chain on the space B_n.

In the case $n = 3$, the transition matrix for MTR corresponding to the trees in Figure 1 is

$$Q = \begin{array}{c|ccccc} & T_1 & T_2 & T_3 & T_4 & T_5 \\ \hline T_1 & p_1 & 0 & p_2 & p_3 & 0 \\ T_2 & 0 & p_1 & p_2 & p_3 & 0 \\ T_3 & p_1 & 0 & p_2 & 0 & p_3 \\ T_4 & 0 & p_1 & p_2 & p_3 & 0 \\ T_5 & 0 & p_1 & p_2 & 0 & p_3 \end{array}$$

The correspondence between trees and permutations makes it possible to read off the exact transition probabilities for the Markov chain for trees from those for MTF.

well-defined and onto, and determines an equivalence relation on S_n. We say that two permutations σ and σ' are equivalent if $t(\sigma) = t(\sigma')$, that is, if they correspond to the same tree in the tree-building operation.

Let $\Pi : B_n \to 2^{S_n}$ be the set-valued inverse of t. That is, $\Pi(T) = \{\sigma \in S_n : t(\sigma) = T\}$. The $\Pi(T)$'s are the equivalence classes of S_n.

Note that some authors have considered 1-to-1 mappings between the symmetric group and the space of binary trees in a way that gives a method for ordering and ranking trees. See, for instance, Ruskey and Hu (1977) and Trojanowski (1978). By contrast, here we are considering the set of *all* permutations which can be identified with a particular tree.

The *move-to-root* (MTR) operation is defined as a series of simple exchanges between nodes. A *simple exchange* (SE) for a requested record j is as follows:

(i) Do nothing if j is the root.

(ii) If j is the left child of its parent m, the resulting tree will be the same as the original except for the subtree whose root was m. Record j is "rotated" up to m so that j becomes the root of this subtree. The old left subtree of j doesn't change in relation to j. The old right subtree of j becomes the left subtree of m. The old right subtree of m keeps its relation to m. The transformation is best understood by examining Figure 2-L.

(iii) If j is the right child of m, perform the analogous transformation. (See Figure 2-R.)

The MTR operation performs a sequence of simple exchanges until the requested record is moved to the root of the tree.

Thus MTR and SE are natural analogues of the move-to-front (MTF) and transposition (TR) rules for linear lists. In MTF, an accessed record is brought to the top of the list. In TR, it is transposed with its immediate predecessor.

Figure 2.

The move-to-root heuristic—described in Section 2—is one self-organizing method which has been studied by several authors. Allen and Munro (1978) introduced the heuristic and gave an exact formula for stationary expected search cost (the asymptotic average cost of retrieving a record). Other treatments of self-organizing trees include Bitner (1979), who considers various search rules, and Sleator and Tarjan (1985), who introduce splay trees and develop (non-probabilistic) amortized analysis of search cost.

This paper is organized as follows: In Section 2 we set notation and describe a many-to-1 mapping between the set of permutations and the set of binary search trees which permits tree operations to be expressed in terms of operations on permutations. We show in Section 3 that the Markov chain for MTR can be obtained by lumping the MTF chain. In Section 4, by exploiting the recursive definition of binary trees and using a simple property of sampling without replacement, we derive the stationary distribution for MTR in a form that is intrinsically tree-based and computationally simple. In Section 5 we give formulas for the k-step transition probabilities, and in Section 6 we analyze the eigenstructure of MTR. In so doing we note interesting parallels with the spectral structure of MTF.

We will treat rates of convergence to stationarity in future work.

2 Notation and preliminaries

Consider an ordered, indexed set of n records. For ease of notation and exposition we identify the records with their indices and just consider $[n] := \{1, 2, \ldots, n\}$ as the set of records.

Let B_n be the set of all labeled binary search trees on n nodes. It can be shown, by exploiting the recursive definition and using generating functions, that $|B_n| = \binom{2n}{n} / (n+1)$. In what follows we use the term "tree" for binary search tree.

Let $\sigma = (\sigma_1, \ldots, \sigma_n) \in S_n$ be a permutation of $[n]$. We will consider σ_k to be the record at the kth position of σ. Define a "tree-building" function $t : S_n \to B_n$ as follows: $t(\sigma)$ is the tree obtained by inserting $\sigma_1, \ldots, \sigma_n$ successively into an empty tree. While technically the function t depends on n, notationally there is no need to distinguish among t for various n.

The function t corresponds to inserting new records into a tree. It is

1 Introduction and Summary

There has been much interest in recent years in self-organizing search methods. Hester and Hirschberg (1985) survey the field. Hendricks (1989) is a good introduction with numerous applications and open problems.

While most research in this area has been devoted to sequential search techniques for linear lists, a growing body of work addresses heuristics for other data structures. In particular, the binary search tree is a very common and important structure which exploits the ordering of records to achieve faster search time. Records are stored at the nodes of a tree in such a way that a traversal of the tree produces the records in their linear order.

A *binary tree* is a finite tree with at most two "children" for each node and in which each child is distinguished as either a left or right child. By defining an empty binary tree as a binary tree with no nodes we can give a useful recursive definition: a binary tree either is empty or is a node with left and right subtrees, each of which is a binary tree.

Consider a binary tree in which the nodes are labeled with elements of some linearly ordered set. Inorder traversal is a common method for traversing the tree: visit the root after visiting the left subtree and before visiting the right subtree. If this traversal yields the labels in order, the tree is called a *binary search tree.* For example, the set of all binary search trees on 3 nodes is given by:

Figure 1.

Consider a set of n records stored at the nodes of a binary search tree. Assume that record i is accessed with unknown probability p_i and independently of past requests. For simplicity, assume that all the p_i's are strictly positive. As records are accessed we would like to alter the tree dynamically so that the average search cost is made small, where the search cost of a record is defined as one more than the length of the unique path from the root to the node containing the record.

On the Markov chain for the move-to-root rule for binary search trees

[short title: Move-to-root rule for binary search trees]

by Robert P. Dobrow and James Allen Fill*
The Johns Hopkins University

January 4, 1998

Abstract

The move-to-root (MTR) heuristic is a self-organizing rule which attempts to keep a binary search tree in near-optimal form. It is a tree analogue of the move-to-front (MTF) scheme for self-organizing lists. Both heuristics can be modeled as Markov chains. We show that the MTR chain can be derived by lumping the MTF chain and give exact formulas for the transition probabilities and stationary distribution for MTR. We also derive the eigenvalues and their multiplicities for MTR.

[1]Research for both authors supported by NSF grant DMS-9311367.

[2]*AMS 1991 subject classifications.* Primary 60J10; secondary 68P10, 68P05.

[3]*Keywords and phrases.* Markov chains, self-organizing search, binary search trees, move-to-root rule, lumping, eigenvalues, simple exchange, move-to-front rule.

Rational Numbers with Non-Terminating, Non-Periodic Modified Engel-Type Expansions

Jeffrey Shallit
Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada
shallit@graceland.waterloo.edu

Abstract.

Recently, Kalpazidou, Knopfmacher, and Knopfmacher asked if there exist rational numbers whose "modified Engel-type" expansion is neither finite nor ultimately periodic. In this note we answer their question by explicitly providing an infinite sequence of such numbers.

In a recent paper [3], Kalpazidou, Knopfmacher, and Knopfmacher discussed expansions for real numbers of the form

$$A = a_0 + \frac{1}{a_1} - \frac{1}{a_1+1} \cdot \frac{1}{a_2} + \frac{1}{(a_1+1)(a_2+1)} \cdot \frac{1}{a_3} - \cdots \tag{1}$$

which they called a "modified Engel-type" alternating expansion. Here a_0 is an integer and a_i is a positive integer for $i \geq 1$. If $a_{i+1} \geq a_i$, this expansion is essentially unique. To save space we will abbreviate Eq. (1) by

$$A = \{a_0, a_1, a_2, \ldots\}.$$

They say, "The question of whether or not all rationals have a finite or recurring expansion has not been settled." (By "recurring" we understand "ultimately periodic".)

In this note, we prove that the rational numbers $\frac{2}{2r+1}$ (r an integer ≥ 2) have modified Engel-type expansions that are neither finite nor ultimately periodic.

Theorem.

Let r be an integer ≥ 1. Then

$$\frac{2}{2r+1} = \{a_0, a_1, a_2, \ldots\}$$

where $a_0 = 0$, and $a_{2i-1} = b_i$, $a_{2i} = 2b_i - 1$ for $i \geq 1$, and $b_1 = r$, $b_{n+1} = 2b_n^2 - 1$ for $n \geq 1$.

Proof.

As in [3], we have $a_0 = \lfloor A \rfloor$, $A_1 = A - a_0$, $a_n = \lfloor 1/A_n \rfloor$ for $n \geq 1$ and $A_{n+1} = (1/a_n - A_n)(a_n + 1)$ for $n \geq 1$.

From this we see that $a_0 = \lfloor \frac{2}{2r+1} \rfloor = 0$.

We now prove the following four assertions by induction on n: (i) $A_{2n-1} = \frac{2}{2b_n+1}$; (ii) $a_{2n-1} = b_n$; (iii) $A_{2n} = \frac{b_n+1}{b_n(2b_n+1)}$; and (iv) $a_{2n} = 2b_n - 1$.

It is easy to verify these assertions for $n = 1$, as we find

(i) $A_1 = \frac{2}{2r+1} = \frac{2}{2b_1+1}$;

(ii) $a_1 = \left\lfloor \frac{1}{A_1} \right\rfloor = r = b_1$;

(iii) $A_2 = (\frac{1}{r} - \frac{2}{2r+1})(r+1) = \frac{r+1}{r(2r+1)} = \frac{b_1+1}{b_1(2b_1+1)}$;

(iv) $a_2 = \left\lfloor \frac{1}{A_2} \right\rfloor = \left\lfloor \frac{r(2r+1)}{r+1} \right\rfloor = \left\lfloor 2r - 1 + \frac{1}{r+1} \right\rfloor = 2r - 1 = 2b_1 - 1$.

Now assume the result is true for all $i \leq n$. We prove it for $n+1$:

(i)

$$\begin{aligned} A_{2n+1} &= \left(\frac{1}{a_{2n}} - A_{2n} \right) (a_{2n} + 1) \\ &= \left(\frac{1}{2b_n - 1} - \frac{b_n + 1}{b_n(2b_n + 1)} \right) (2b_n) \\ &= \frac{2}{4b_n^2 - 1} \\ &= \frac{2}{2b_{n+1} + 1}. \end{aligned}$$

(ii)

$$a_{2\,n+1} = \left\lfloor \frac{1}{A_{2\,n+1}} \right\rfloor = \left\lfloor \frac{2b_{n+1}+1}{2} \right\rfloor = b_{n+1}.$$

(iii)

$$\begin{aligned} A_{2\,n+2} &= \left(\frac{1}{a_{2\,n+1}} - A_{2\,n+1} \right) (a_{2\,n+1}+1) \\ &= \left(\frac{1}{b_{n+1}} - \frac{2}{2b_{n+1}+1} \right) (b_{n+1}+1) \\ &= \frac{b_{n+1}+1}{b_{n+1}(2b_{n+1}+1)}. \end{aligned}$$

(iv)

$$\begin{aligned} a_{2\,n+2} &= \left\lfloor \frac{1}{A_{2\,n+2}} \right\rfloor \\ &= \left\lfloor \frac{b_{n+1}(2b_{n+1}+1)}{b_{n+1}+1} \right\rfloor \\ &= \left\lfloor 2b_{n+1} - 1 + \frac{1}{b_{n+1}+1} \right\rfloor \\ &= 2b_{n+1} - 1. \end{aligned}$$

This completes the proof. ■

Corollary.

For $r \geq 2$, the rational numbers $\frac{2}{2r+1}$ have non-terminating, non-ultimately-periodic modified Engel-type expansions.

Additional Remarks.

- For $r = 1$, the theorem gives the ultimately periodic expansion

$$2/3 = \{0, 1, 1, 1, 1, \ldots\}.$$

- For $r \geq 2$, the expansion is not ultimately periodic; e.g.

$$2/5 = \{0, 2, 3, 7, 13, 97, 193, 18817, \ldots\}.$$

In this case, we have the following brief table:

n	a_n	b_n	A_n
1	2	2	2/5
2	3	7	3/10
3	7	97	2/15
4	13	18817	8/105
5	97	708158977	2/195
6	193	1002978273411373057	98/18915

• The sequence $b_1, b_2, \ldots = 2, 7, 97, 18817, 708158977, \ldots$, corresponding to $r = 2$, appears to have been discussed first by G. Cantor in 1869 [1], who gave the infinite product

$$\sqrt{3} = \left(1 + \frac{1}{2}\right)\left(1 + \frac{1}{7}\right)\left(1 + \frac{1}{97}\right)\cdots.$$

For more on this product of Cantor, see Spiess [9], Sierpiński [7], Engel [2], Stratemeyer [10,11], Ostrowski [6], and Mendès France and van der Poorten [5]. The sequence $2, 7, 97, 18817, \ldots$ was also discussed by Lucas [4]. It is sequence #720 in Sloane [8].

• The sequence $b_1, b_2, \ldots = 3, 17, 577, 665857, \ldots$, corresponding to $r = 3$, was also discussed by Cantor [1], who gave the infinite product

$$\sqrt{2} = \left(1 + \frac{1}{3}\right)\left(1 + \frac{1}{17}\right)\left(1 + \frac{1}{577}\right)\cdots.$$

Also see the papers mentioned above. The sequence was also discussed by Wilf [12]. It sequence #1234 in Sloane [8].

• It is easy to prove that $b_{n+1} = B_{2^n}$ where $B_0 = 1$, $B_1 = r$, and $B_n = 2rB_{n-1} - B_{n-2}$ for $n \geq 2$. This gives a closed form for the sequence (b_n):

$$b_{n+1} = \frac{(r + \sqrt{r^2 - 1})^{2^n} + (r - \sqrt{r^2 - 1})^{2^n}}{2}.$$

• 3/7 is the "simplest" rational for which no simple description of the terms in its modified Engel-type expansion is known. The first forty terms are as follows:

$$3/7 = \{0, 2, 4, 5, 7, 8, 10, 25, 53, 62, 134, 574, 2431, 13147, 27167, 229073, 315416,$$

$$435474, 771789, 1522716, 3853889, 7878986, 7922488, 8844776, 9182596, 9388467,$$

$$14781524, 135097360, 1374449987, 1561240840, 4408239956, 11166053604, 12014224315,$$

$$23110106464, 553192836372, 900447772231, 1189661630241, 2058097840143484,$$

$$6730348855426376, 12928512475357529, \cdots\}.$$

More generally, it would be of interest to know whether it is possible to characterize the modified Engel expansion of every rational number.

References

[1] G. Cantor, Zwei Sätze über eine gewisse Zerlegung der Zahlen in undendliche Producte, *Z. Math. Phys.* **14** (1869), 152–158.

[2] F. Engel, Entwicklung der Zahlen nach Stammbrüchen, *Verhandlungen der 52sten Versammlung deutscher Philologen und Schulmänner*, 1913, pp. 190-191.

[3] S. Kalpazidou, A. Knopfmacher, and J. Knopfmacher, Lüroth-type alternating series representations for real numbers, *Acta Arithmetica* **55** (1990), 311-322.

[4] E. Lucas, Considérations nouvelles sur la théorie des nombres premiers et sur la division géométrique de la circonférence en parties égales, *Assoc. Française Pour L'Avancement des Sciences* **6** (1877), 159-166.

[5] M. Mendès France and A. van der Poorten, From geometry to Euler identities, *Theor. Comput. Sci.* **65** (1989), 213–220.

[6] A. Ostrowski, Über einige Verallgemeinerungen des Eulerschen Produktes $\prod_{v=0}^{\infty}(1 + x^{2^v}) = \frac{1}{1-x}$, *Verh. Naturforsch. Gesell. Basel* **11** (1929), 153-214.

[7] W. Sierpiński, O kilku algorytmach dla rozwijania liczb rzeczywistych na szeregi, *C. R. Soc. Sic. Varsovie* **4** (1911), 56-77. In Polish; French version appeared as Sur quelques algorithmes pour dévelopeer les nombres réels en séries, in *Oeuvres Choisies*, V. I, PWN, Warsaw, 1974, pp. 236-254.

[8] N. J. A. Sloane, *A Handbook of Integer Sequences*, Academic Press, 1973.

[9] O. Spiess, Über eine Klasse unendlicher Reihen, *Arch. Math. Phys.* **12** (1907), 124-134.

[10] G. Stratemeyer, Stammbruchentwickelungen für die Quadratwurzel aus einer rationalen Zahl, *Math. Zeit.* **31** (1930), 767–768.

[11] G. Stratemeyer, Entwicklung positiver Zahlen nach Stammbrüchen, *Mitt. Math. Sem. Univ. Giessen* **20** (1931), 3–27.

[12] H. Wilf, Limit of a sequence, Elementary Problem E 1093, *Amer. Math. Monthly* **61** (1954), 424-425.

Énumération des méandres : une approche à partir des méthodes de physique théorique

Jérémie Bouttier

Mémoire d'exposé bibliographique
du DEA de Physique Théorique

Réalisé sous la direction de
Jesper Lykke Jacobsen (LPTMS Orsay)

Juin 2001

Table des matières

1 Introduction

1.1 Le problème des méandres

Le problème des méandres est l'un de ces problèmes mathématiques à l'énoncé fort simple, qui reste malgré tout encore sans solution complète aujourd'hui : de combien de façons une route fermée sans croisement peut-elle traverser une rivière rectiligne en un nombre donné de ponts ?

De façon plus mathématique, on définit la notion de *méandre fermé* : étant donné une droite (*rivière*), un méandre fermé d'ordre n est un chemin fermé auto-évitant ou lacet de Jordan (*route*) qui coupe la rivière en $2n$ points[1] (*ponts*). Deux méandres seront dits équivalents si on peut passer de l'un à l'autre par déformation continue sans déplacer la rivière ni changer l'ordre des ponts.

Le problème revient donc à chercher le nombre M_n de classes d'équivalence de méandres fermés d'ordre n. Pour clarifier les choses, la figure 1 montre les $M_3 = 8$ classes de méandres d'ordre 3.

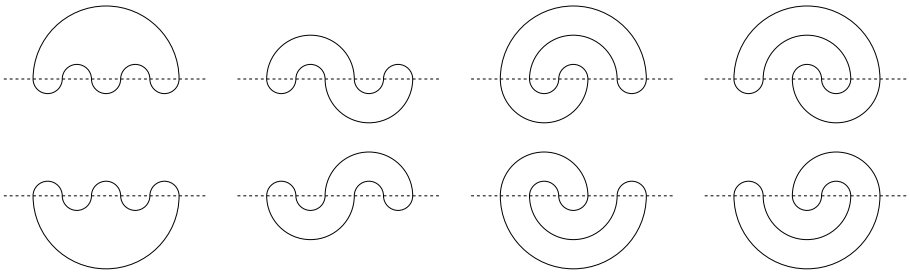


FIG. 1: Les 8 classes de méandres d'ordre 3 (6 ponts)

On ne connaît aujourd'hui aucune formule explicite donnant M_n en fonction de n. Une vingtaine de termes de la suite sont connus par énumération (tableau 1), ce qui nécessite déjà des méthodes avancées en raison de la croissance exponentielle de M_n. On trouve d'autre part des estimations du comportement asymptotique de cette suite.

1.2 Historique

Le problème des méandres est connu au moins depuis la fin du XIX^e siècle, sous diverses formulations. La version la plus ancienne semble être connue sous le nom de *problème des timbres-poste* : de combien de façons peut-on replier totalement une bande de n timbres sur un seul timbre[2] ? Ce problème est mentionné par Lucas en 1891 [2]. Plus tard, Sainte-Lagüe s'y intéresse [3] et y consacre notamment en 1937 un chapitre dans un ouvrage de

[1] Il y a nécessairement un nombre pair d'intersections.

[2] L'équivalence entre différentes formulations sera discutée dans la partie suivante.

n	M_n
1	1
2	2
3	8
4	42
5	262
6	1828
7	13820
8	110954
9	933458
10	8152860
⋮	⋮
20	64477712119584604

TAB. 1: Premières valeurs de la suite M_n (valeurs tirées de [1])

récréations mathématiques [4], où on trouve une représentation du problème en terme de méandres (même si l'auteur n'utilise pas explicitement ce mot). Notons par la suite les contributions de Touchard [5], Koehler [6], Lunnon [7] à la recherche de relations de récurrence entre termes et d'algorithmes d'énumération par des méthodes combinatoires « classiques ».

Une réactualisation du problème a été faite par Arnol'd dans les années 80 [8]. Dans des travaux en rapport avec le seizième problème de Hilbert (énumération des ovales de courbes algébriques planes), il s'intéresse aux propriétés géométrico-différentielles de la variété des zéros de polynômes hyperboliques dans le plan projectif. En particulier, connaissant le nombre d'intersections d'un lacet et d'une droite projective, il parvient à une estimation du nombre de points d'inflexion de ce lacet par des considérations sur le « méandre » ainsi formé. Arnol'd est le premier à introduire la dénomination de méandres, même s'il s'intéresse alors plus particulièrement aux méandres *projectifs* (tracés dans le plan projectif). À sa suite, les travaux de Lando et Zvonkin [9] [10] introduisent la formulation moderne du problème des méandres dans le plan affine.

À côté du problème de l'énumération, les méandres apparaissent chez Poincaré dans une tentative de démonstration d'un théorème de point fixe[3] [11]. Si la preuve définitive de ce théorème par Birkhoff en 1913 n'y fait plus allusion, les méandres réapparaissent dans une généralisation par Eliashberg en 1978 [12]. Ils interviennent également dans des travaux de classification des 3-surfaces [13], par des résultats combinatoires pour la recherche d'invariants

[3] L'énoncé est le suivant : toute transformation d'une couronne circulaire préservant l'aire et déplaçant les bords en des directions opposées admet au moins deux points fixes. Dans son approche, les positions relatives d'un cercle et son image par la transformation sont représentées comme des méandres.

de Witten-Reshetikhin-Turaev.

En informatique, on étudie les *permutations planes* équivalentes aux méandres [14]. Ces permutations peuvent être ordonnées en temps linéaire [15]. Il existe également un lien avec la théorie des langages formels car un méandre peut se représenter en terme de parenthésage.

D'un point de vue esthétique, le problème des méandres est d'une grande beauté. Phillips [16] s'est intéressé aux labyrinthes présents dans l'art de différentes civilisations, et observe des analogies structurelles avec les méandres.

Finalement, dans les années 90, le problème a commencé à intéresser les physiciens théoriciens. Le problème des timbres-poste peut s'interpréter de façon évidente comme l'énumération des configurations d'un polymère totalement replié, d'où une connexion avec la physique statistique. D'autre part, Kazakov et Kostov[4] ont suggéré le lien entre méandres et graphes de Feynman, dans la cadre des modèles de matrices. Di Francesco, Golinelli et Guitter approfondissent ce lien [17], tout en présentant d'autres approches récursives et des résultats exacts dans certains cas particuliers. Les approches énumératives [18] sont poursuivies par l'utilisation de méthodes de matrice de transfert ayant permis un calcul de M_n jusqu'à $n = 24$ [19], ou bien par une approche Monte-Carlo pour des estimations jusqu'à $n = 400$ [20]. Des méthodes algébriques fondées sur l'algèbre de Temperley-Lieb sont entreprises [21]. Enfin, le comportement asymptotique est estimé [22] en interprétant le problème des méandres comme le couplage d'un gaz de boucles compactes à la gravité quantique bidimensionnelle. Des exposants peuvent ainsi être prédits et vérifiés numériquement [23].

1.3 Plan

Ce mémoire a pour objet de présenter brièvement les méthodes récentes tirées de la physique théorique, qui ont pu être appliquées au problème des méandres. Mon étude bibliographique a donc principalement porté sur les publications récentes sur le sujet, presque toutes citées dans le dernier paragraphe de la section précédente. Je dois ajouter à celles-ci un article de revue [24] comportant une partie consacrée aux méandres.

En une première partie je présenterai quelques variantes du problème des méandres et des formulations équivalentes. Puis je montrerai un modèle de matrice lié aux méandres. Enfin j'exposerai l'approche la plus récente à partir du couplage d'un gaz de boucles à la gravité quantique, qui permet de prédire certains exposants critiques.

[4] Leurs résultats ont été indirectement publiés dans [10].

2 Généralisations du problème, formulations équivalentes

2.1 Autres types de méandres

Jusqu'ici, nous n'avons parlé que des méandres fermés. On peut bien entendu définir de nombreuses autres classes. Citons d'abord (fig. 2) :

- les méandres *ouverts* pour lesquels on ne suppose plus la route fermée (route à deux extrémités),
- les méandres ouverts pour lesquels on fixe les extrémités de la route à l'infini (il est alors loisible d'inverser les dénominations de route et rivière, ce qui donne une image plus réaliste),
- les *semi-méandres* pour lesquels la route est fermée mais la rivière est une demi-droite : la route peut alors contourner la rivière autour de son extrémité (*source*).

Pour ces types de méandres, le nombre de ponts n'est pas nécessairement pair.

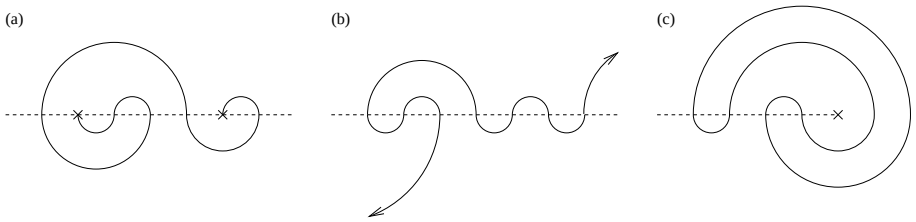


FIG. 2: Autres types de méandres. a : méandre ouvert (par déformation, on peut toujours ramener les extrémités de la route en un pont) ; b : méandre ouvert avec extrémités à l'infini ; c : semi-méandre.

Dans le sens d'Arnol'd [8], un méandre est un méandre ouvert avec extrémités à l'infini, et où la route va du « sud-ouest » au « nord-est » ou « sud-est » selon la parité du nombre de ponts. La figure 2-b montre en fait un tel méandre avec 7 ponts. Suivant [10], on note m_n le nombre de classes de méandres au sens d'Arnol'd à n ponts. On a alors la relation $M_n = m_{2n-1}$, qui résulte de la bijection réalisée en « refermant » un méandre d'Arnol'd par un pont supplémentaire situé à gauche des $2n-1$ autres.

Généralisons encore la notion de méandre en relâchant la contrainte de connexité de la route : on peut s'intéresser aux configurations avec une rivière toujours rectiligne, $2n$ ponts, et un nombre $k \geq 1$ de routes fermées ne se croisant pas mutuellement mais traversant la rivière au moins une fois. De telles configurations sont appelées *systèmes de méandres fermés* à $2n$ ponts et k composantes connexes, le nombre de configurations inéquivalentes est noté $M_n^{(k)}$. Pour $k = 1$ on retrouve évidemment les méandres fermés usuels :

$M_n^{(1)} = M_n$. Comme chaque route comporte au moins deux ponts, on a l'inégalité $k \leq n$; la quantité $M_n(q) = \sum_{k=0}^{n} M_n^{(k)} q^k$ est un polynôme en q appelé *polynôme méandrique*. Le cas $q = 1$ correspond au nombre de systèmes de méandres à $2n$ ponts sans contrainte sur le nombre de composantes connexes, et ce nombre peut-être calculé explicitement : $M_n(1) = c_n^2$ où $c_n = \frac{2n!}{n!(n+1)!}$ est le n^{ieme} nombre de Catalan.

2.2 Formulations équivalentes

2.2.1 Problème des timbres-poste

En introduction, nous avons mentionné le problème des timbres-poste : combien existe-t-il de façons différentes de replier totalement une bande de n timbres, de sorte que l'empilement occupe la surface d'un seul timbre ?

Nous supposerons ici que les deux faces d'un timbre ainsi que les deux extrémités de la bande sont indistingables. Par contre, l'empilement des timbres est supposé placé dans un plan vertical, les plis sont horizontaux et on regarde horizontalement, parallèlement au plan (fig. 3-gauche) ; nous supposerons que dans ces conditions on peut distinguer gauche et droite, haut et bas de l'empilement. Comme le montre la figure 3, la correspondance avec les méandres peut alors se voir façon directe : on représente les timbres comme points alignés sur une droite et les plis entre timbres comme des arcs de cercles joignant les points correspondants (fig. 3).

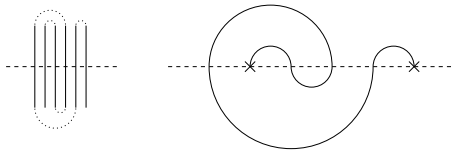


FIG. 3: Relation entre timbres et méandres : à gauche, une bande de 6 timbres pliée (les charnières entre timbres sont représentées en arcs pointillés) ; à droite, le méandre ouvert correspondant.

Ainsi il y correspondance entre le pliage d'une bande ouverte de n timbres et un méandre ouvert à n ponts ; de la même façon, le pliage d'une bande fermée de $2n$ timbres correspond à un méandre fermé d'ordre n. Pour une bande ouverte mais dont une extrémité est attachée à un support fixe, la correspondance naïve serait avec un méandre ouvert dont une extrémité serait à l'infini. En fait, on peut trouver une correspondance avec les semi-méandres, mais de façon plus difficile à voir (fig. 4).

On remarque qu'à la différence des autres correspondances, la bande de timbres devient la rivière et non la route. Il est possible de définir une correspondance analogue pour les bandes fermées, qui est *duale* à la correspondance directe vue plus haut, au sens où route et rivière sont échangées.

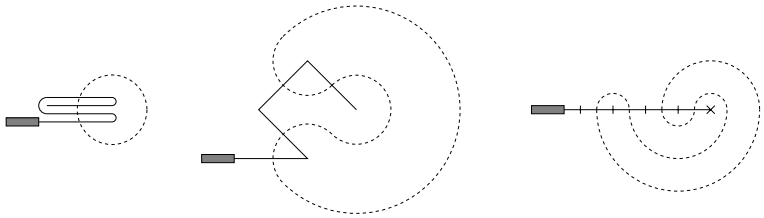


FIG. 4: Relation entre pliage d'une bande avec extrémité fixe et semi-méandres. On trace tout d'abord un cercle traversant chaque timbre de la bande pliée (figure de gauche) ; on « déplie » ensuite la bande par déformation (au milieu) ; le cercle déformé devient alors un semi-méandre (à droite).

2.2.2 Permutations planes

Les permutations planes sont en fait des représentations des méandres sous formes de permutations. Considérons un méandre fermé d'ordre n et un marcheur parcourant la route avec un point de départ et un sens de parcours arbitraires ; le premier pont qu'il rencontre sera numéroté 1, le second 2, et ainsi de suite, jusqu'au moment où le marcheur repasse par son point de départ, alors les ponts auront été numérotés de façon biunivoque par un entier dans $\{1, \ldots, 2n\}$. Lorsqu'on lit la numérotation des ponts le long de la rivière et non plus la route, on obtient une permutation *plane* de $\{1, \ldots, 2n\}$.

Réciproquement, pour vérifier qu'une permutation σ de $\{1, \ldots, 2n\}$ est plane, on numérote $2n$ ponts successifs sur une rivière horizontale par $\sigma(1)$, $\ldots$, $\sigma(2n)$ et on construit la route en reliant chaque point à son successeur dans la numérotation, par un demi-cercle orienté alternativement de part et d'autre de la rivière ; si la permutation est plane on construit ainsi un méandre, dans le cas contraire, certains cercles se coupent (fig 5).

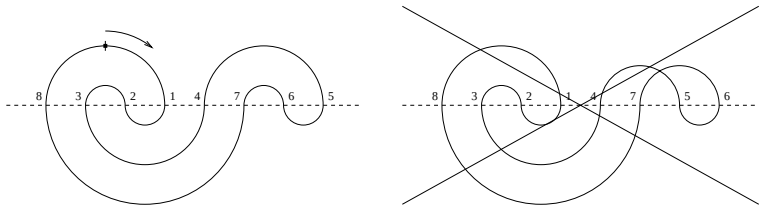


FIG. 5: Équivalence avec les permutations planes : (83214765) est une permutation plane, (83214756) ne l'est pas.

L'inverse d'une permutation plane est également plane, on peut considérer que les méandres correspondants sont liés par une réflexion d'axe la rivière. Deux permutations planes définissent un même méandre si elles diffèrent d'un facteur à droite ω^k où ω est le cycle $(2, 3, \ldots, 2n, 1)$; ceci correspond simplement à un changement de point de départ du marcheur sur la

route, sans changement de sens de parcours. Il y a donc exactement $2nM_n$ permutations planes.

2.3 Méandres sur la sphère

On peut trouver des correspondances entre les différents types de méandres plans rencontrés jusqu'ici en généralisant les méandres à la sphère. Il s'agit en effet d'une généralisation puisque le plan est compactifié en une sphère par ajout du point à l'infini.

Cependant, sur la sphère, le point à l'infini perd sa spécificité et on autorise alors toute déformation continue. Dans le cas des méandres fermés, rivière et route deviennent alors des lacets simples, sans plus permettre de distinction entre les deux. Les méandres ouverts et les semi-méandres correspondent à des configurations avec un lacet et un chemin ouvert.

Inversement, les méandres sur la sphère peuvent être projetés dans le plan après choix d'un point à l'infini (fig 6).

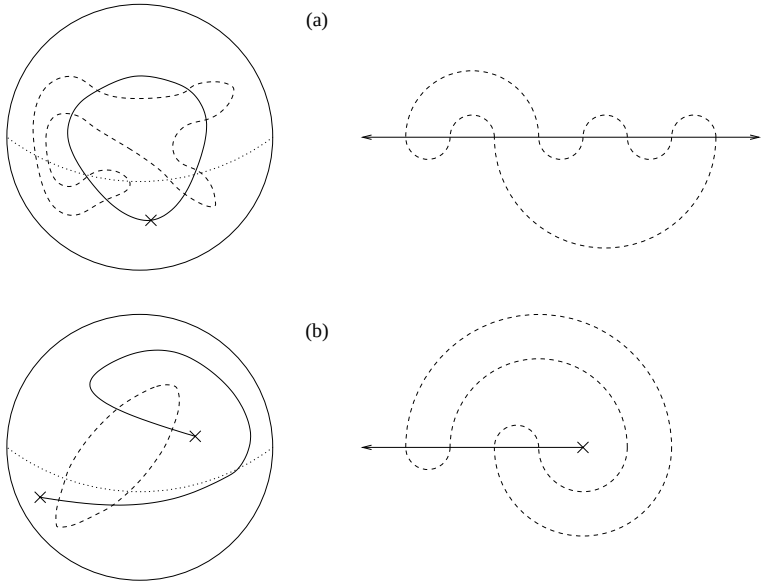


FIG. 6: Méandres sur la sphère et leurs correspondants plans. a : méandre fermé, projeté dans le plan par choix du point à l'infini sur un lacet ; b : méandre semi-ouvert, projeté en un semi-méandre dans le plan par envoi d'une extrémité du chemin ouvert à l'infini.

On ne peut toutefois aisément relier le nombre de classes de méandres sphériques et plans. Par exemple, un méandre fermé sur la sphère peut générer *a priori* autant de méandres fermés plans qu'il y a de façons de choisir le point à l'infini puis une orientation de la rivière, mais certains de ces méandres peuvent être en fait équivalents en raison des « symétries » du

méandre de départ.

3 Formalisme des modèles de matrices

3.1 Introduction du formalisme

Les modèles de matrices [25] permettent de reformuler virtuellement tout problème d'énumération de graphes en terme de calcul d'une intégrale sur des matrices. Nous présentons brièvement dans cette section le cas le plus simple de modèle de matrice, destiné à introduire le formalisme, avant de passer au modèle correspondant aux méandres.

Considérons une intégrale sur les matrices hermitiennes $N \times N$ de la forme :

$$Z(V,N) = \frac{1}{Z_0(N)} \int dM e^{-N \operatorname{Tr} V(M)} \tag{1}$$

où $V(x)$ est un « potentiel » polynomial $V(x) = \frac{x^2}{2} - \sum_{i \geq 3} t_i \frac{x^i}{i}$, et $Z_0(N)$ un facteur de normalisation tel que $Z(V_0, N) = 1$ où $V_0 = \frac{x^2}{2}$.

On développe alors en série formelle en les t_i la partie non-gaussienne de l'exponentielle ; le coefficient de $\prod t_i^{v_i}$ est :

$$\frac{1}{Z_0(N)} \int dM e^{-N \operatorname{Tr} \frac{M^2}{2}} N^{\Sigma v_i} \prod_i \frac{\operatorname{Tr}(M^i)^{v_i}}{i^{v_i} v_i!}. \tag{2}$$

Il s'agit d'une valeur moyenne d'une fonction de matrice avec un poids gaussien. Comme pour toute mesure gaussienne, une telle valeur moyenne peut être calculée à partir des règles de Feynman. Sans détailler le processus d'élaboration de ces règles, mentionnons-les brièvement (fig 7) :

- Le propagateur (a) est $\langle M_{ij} M_{kl} \rangle = \delta_{il} \delta_{jk} / N$ et est représenté par une ligne double, chaque ligne étant orientée et « portant » un indice de matrice qui est conservé.
- Chaque $\operatorname{Tr}(M^i)$ dans (2) introduit un vertex i-valent analogue à (b), affecté d'un poids N/i. Sur les lignes connectées, l'indice de matrice est le même.
- Lorsqu'un graphe de Feynman est constitué en connectant les vertex par les propagateurs, les lignes connectées forment des boucles (c) et portent toutes le même indice matriciel. La sommation sur ces indices de boucle fait apparaître un facteur N supplémentaire par boucle.

Le calcul du poids global d'un graphe de Feynman est finalement très simple : compter un facteur N par vertex, un facteur $1/N$ par propagateur (arête du graphe), et enfin un facteur N par boucle, d'où un facteur total $N^{f-a+v} = N^{2-2g}$, où f, a, v, g sont respectivement les nombres de faces, d'arêtes et de vertex et le genre du graphe[5]. Tous les autres facteurs se

[5] $\chi \equiv f - a + v$ est la caractéristique d'Euler du graphe, et on a la relation $\chi = 2 - 2g$.

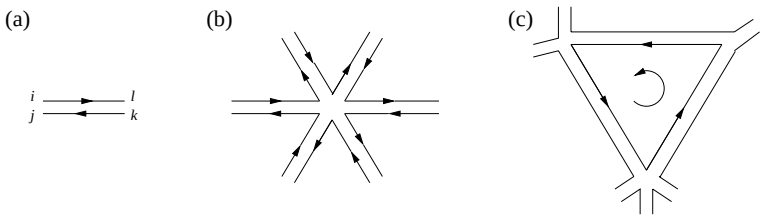


FIG. 7: Constituants des diagrammes de Feynman de modèles de matrices. a : propagateur ; b : vertex 6-valent ; c : boucle d'indice libre.

groupent en un simple facteur de symétrie, ce qui aboutit finalement au développement :

$$Z(V,N) = \sum_{\text{graphes } \Gamma} \frac{1}{|\mathrm{Aut}(\Gamma)|} N^{2-2g(\Gamma)} \prod_i t_i^{v_i(\Gamma)} \tag{3}$$

où la somme s'étend sur tous les diagrammes non nécessairement connexes, et $|\mathrm{Aut}(\Gamma)|$, $g(\Gamma)$, $v_i(\Gamma)$ désignent respectivement l'ordre du groupe de symétrie, le genre et le nombre de vertex i-valents du graphe Γ. La somme est restreinte aux diagrammes connexes si on considère le logarithme de Z.

Le fait remarquable est qu'on obtient ainsi une série formelle en N, le terme en N^{2-2g} étant donné par une sommation sur les graphes de genre g. Dans la limite $N \to \infty$, la contribution dominante correspond aux graphes de genre 0, c'est-à-dire les graphes pouvant être tracés sur une surface de genre 0 comme la sphère ou le plan. Pour cette raison, la limite $N \to \infty$ est appelée limite *planaire*. L'énergie libre planaire est définie par :

$$F_{pl}(V) \equiv \lim_{N\to\infty} \frac{\ln Z(V,N)}{N^2} \tag{4}$$

et se développe en :

$$F_{pl}(V) = \sum_{\Gamma} \frac{\prod_i t_i^{v_i(\Gamma)}}{|\mathrm{Aut}(\Gamma)|} \tag{5}$$

où la somme porte sur tous les graphes Γ connexes planaires.

3.2 Application aux méandres

Le modèle présenté ci-dessus ne comporte qu'une seule intégration sur les matrices. Une généralisation évidente du modèle se fait en considérant une intégration sur n matrices de taille $N \times N$:

$$Z(V,N) = \frac{1}{Z_0(N)} \int \prod_{i=1}^{n} dM_i \; e^{-N \mathrm{Tr}\, V(M_1,\ldots,M_n)}. \tag{6}$$

En prenant pour V un potentiel de partie quadratique diagonale ($V_0(\{M_i\}) = \frac{1}{2}\sum M_i^2$), l'indice de « couleur » i est conservé le long d'un propagateur. La forme des vertex est quant à elle déterminée par la partie quadratique.

Pour représenter les méandres en terme de graphe d'un modèle de matrice, l'idée évidente est de représenter les ponts comme vertex et les éléments de route et de rivière comme propagateurs. La limite planaire du modèle de matrice est utile pour ne retenir que les diagrammes correspondant effectivement à des méandres du plan.

Il faut *a priori* deux couleurs de matrice dans le modèle, pour permettre de distinguer entre les éléments de route et de rivière. Cependant, il est également souhaitable de garder un moyen de contrôler le nombre de composantes connexes de route et de rivière ; ceci peut être réalisé en utilisant une méthode de « répliques ». Notre modèle contiendra donc n matrices « noires » $N_1, \ldots, N_n$ et b matrices « blanches » $B_1, \ldots, B_b$. L'intégrale de matrice considérée sera alors :

$$Z(N; n, b, x) = \frac{1}{Z_0(N; n, b)} \int \prod_{i=1}^{n} dN_i \prod_{j=1}^{b} dB_j \; e^{-N \mathrm{Tr}\, V(\{N_i\},\{B_j\})} \qquad (7)$$

avec le potentiel :

$$V(\{N_i\},\{B_j\}) = \frac{1}{2}(\sum_{i=1}^{n} N_i^2 + \sum_{j=1}^{b} B_j^2 - x \sum_{i=1}^{n}\sum_{j=1}^{b} N_i B_j N_i B_j). \qquad (8)$$

Sur les graphes de Feynman, il y a donc des arêtes noires (représentées en trait continu) portant un indice de réplique compris entre 1 et n, et des arêtes blanches (trait pointillé) portant un indice de réplique compris entre 1 et b. Le seul type de vertex possible est présenté sur la figure 8.

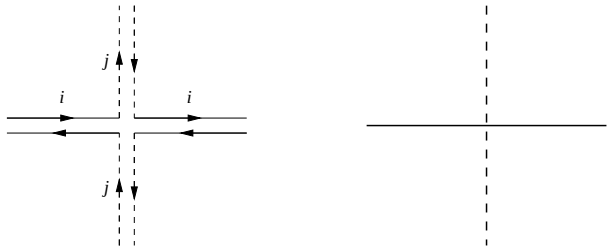


FIG. 8: Vertex du modèle, et son squelette.

Dans la limite planaire, il n'est plus nécessaire de représenter la propagation des indices matriciels, et le squelette du graphe fournit une représentation très explicite des routes et rivières. Notons toutefois que la présence des indices reste sous-jacente car elle « force » la route à traverser la rivière. De plus, l'indice de réplique de chaque type d'arête est conservé au niveau

des vertex : la sommation de ces indices fournit un poids n par boucle noire et b par boucle blanche dans un graphe de Feynman.

Finalement, les graphes de Feynman sont constitués de boucles noires et blanches se coupant sans tangence, tracés sur la sphère. L'énergie libre planaire se développe comme somme sur les tels graphes connexes :

$$F_{pl}(n,b,x) = \sum_{\Gamma} \frac{1}{|\mathrm{Aut}(\Gamma)|} x^{v(\Gamma)} n^{L_n(\Gamma)} b^{L_b(\Gamma)} \tag{9}$$

où $v(\Gamma), L_n(\Gamma), L_b(\Gamma)$ sont respectivement le nombre de vertex, le nombre de boucles noires, le nombre de boucles blanches du graphe Γ.

Pour revenir aux configurations de méandres vues plus haut, il faut imposer la connexité de la rivière, c'est-à-dire imposer $L_b(\Gamma) = 1$. Cela revient à prendre le terme linéaire en b dans l'énergie libre planaire. Dans ces conditions, la somme peut être vue comme portant sur les configurations de systèmes de méandres :

$$\left.\frac{\partial F_{pl}(n,b,x)}{\partial b}\right|_{b=0} = \sum_{k=1}^{\infty} \frac{x^{2k}}{4k} M_k(n) \tag{10}$$

où apparaissent les polynômes méandriques vus plus hauts. Le nombre de méandres fermés apparaît alors en retenant la partie linéaire en n de cette expression. Le facteur $1/4n$ est identifié en notant que les systèmes de méandres sur la sphère se projettent en $4n$ systèmes de méandres plans modulo le facteur de symétrie.

Ainsi, l'énergie libre planaire du modèle de matrice étudié est, dans certaines limites, fonction génératrice d'une suite reliée aux nombres de méandres fermés. Ce modèle de matrice permet d'obtenir également des fonctions génératrices pour d'autres nombres méandriques. Pour générer les méandres ouverts de la sphère (en correspondance avec les semi-méandres plans), il suffit de considérer la fonction de corrélation à deux points $\langle \phi_1 \phi_1 \rangle$ avec $\phi_1 = \frac{1}{N} \sum_{i=1}^{b} \mathrm{Tr}\, B_i$. En effet les graphes de Feynman contribuant à cette fonction de corrélation possèdent deux « pattes » externes blanches soit deux extrémités de rivière. On trouve finalement :

$$\left.\frac{\partial^2 \langle \phi_1 \phi_1 \rangle_{pl}}{\partial b\, \partial n}\right|_{b=n=0} = 2 \sum_{k=0}^{\infty} x^k \bar{M}_k \tag{11}$$

où $\bar{M}_k$ est le nombre de semi-méandres inéquivalents à k ponts.

Malheureusement, l'intégrale de matrice (7) n'a pu être calculée explicitement jusqu'à présent. Si des méthodes de calcul existent dans le cas de modèles à une matrice, elles ne peuvent être généralisées à notre modèle. On peut toutefois réaliser une intégration partielle, par exemple sur les matrices blanches, puisque cette intégrale partielle est gaussienne. Le développement du résultat fournit des méthodes d'énumération de méandres. D'autre part,

l'interprétation en terme de modèle de matrices fournit des comparaisons « physiques » lors de la reformulation en terme de gaz de boucles couplé à la gravité quantique.

4 Interprétation en terme de gaz de boucles couplé à la gravité quantique

La relation (9) suggère l'analogie avec les modèles de gaz de boucles, puisqu'elle fait apparaître un développement en terme de nombre de boucles sur un graphe. Plus précisément, un tel gaz de boucles est « couplé à la gravité quantique », ce qui ne signifie autre chose que la fonction de partition (ou l'énergie libre) est obtenue en sommant sur différents graphes sur lesquels sont tracés les boucles.

Les gaz de boucles sur des réseaux réguliers ont été intensivement étudiés, car ils apparaissent notamment comme équivalents à des problèmes de coloriage de réseaux. Pour le cas des méandres, le modèle sous-jacent de gaz de boucles sur réseau régulier comporte des boucles de deux couleurs, chaque type remplissant de façon dense le réseau carré. Ce modèle est connu sous le nom de *gaz de boucles denses sur le réseau carré*, ou modèle *FPL*^{2} [6], et a été étudié par Jacobsen et Kondev [26] [27].

Nous présenterons brièvement quelques propriétés du modèle FPL^2 avant d'examiner le mécanisme de couplage à la gravité quantique (modèle GFPL^2), qui permet finalement de prédire des exposants apparaissant dans le comportement asymptotique des méandres.

4.1 Le modèle FPL^2

4.1.1 Définition

Les configurations du modèle de gaz de boucles denses sont obtenues en coloriant de deux couleurs (noir et blanc) les arêtes du réseau carré, de sorte qu'à chaque site du réseau, on a exactement 2 arêtes noires et 2 blanches incidentes (fig 9). Il y a donc, à rotation près, deux types de sites possibles : intersection et évitement.

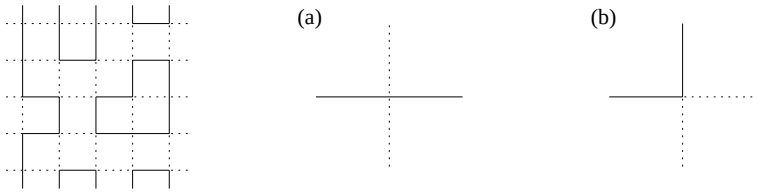


FIG. 9: À gauche, exemple de configuration de gaz de boucles denses. Avec des conditions aux limites périodiques, il y a 4 boucles noires (trait continu) et 2 boucles blanches (trait pointillé). À des rotations près, les sites du réseau sont soit de type a (intersection) ou de type b (évitement).

[6] Fully Packed Loop model.

Avec des conditions aux limites périodiques, les arêtes de même couleurs forment des boucles fermées. Chaque site du réseau est visité par une boucle de chaque couleur, d'où la dénomination de boucles *denses*.

On définit alors la fonction de partition du modèle en introduisant la fugacité n des boucles noires et b des boucles blanches :

$$Z_{\mathrm{FPL}}(n,b) = \sum_{\text{configurations } \mathcal{C}} n^{L_n(\mathcal{C})} b^{L_b(\mathcal{C})} \tag{12}$$

où $L_n(\mathcal{C}), L_b(\mathcal{C})$ désignent respectivement le nombre de boucles noires et blanches dans la configuration $\mathcal{C}$.

Une remarque importante est que cette fonction de partition peut être réexprimée en fonction de l'état *local* de chaque site. Décidons en effet d'orienter arbitrairement chaque boucle noire ou blanche, et d'associer à chaque site du réseau un poids $e^{i\pi(\epsilon_n e_n + \epsilon_b e_b)/4}$, où ϵ_i vaut -1, 0 ou 1 selon que la boucle de couleur i tourne à gauche, va tout droit ou tourne à droite au site considéré. Une boucle se refermant sur elle-même tournant 4 fois plus dans une direction que dans l'autre[7], on obtient un facteur $e^{\pm i\pi e_i}$ par boucle orientée de couleur i, et après sommation sur les différentes orientations possibles des boucles, on obtient un poids $n = 2\cos\pi e_n$ par boucle noire, $b = 2\cos\pi e_b$ par boucle blanche.

4.1.2 Description par une théorie conforme

Pour des fugacités des boucles comprises entre -2 et 2, les facteurs de phase locaux e_n et e_b sont réels, et le modèle est critique. Il peut être décrit dans la limite continue par une théorie conforme de champs scalaires libres. Afin d'identifier les degrés de liberté de la théorie conforme, il est utile de reformuler le modèle en terme de modèle de hauteur d'interface (modèle Solid On Solid). Cette reformulation part d'une interprétation des boucles comme lignes de niveau.

Considérons ainsi une configuration de boucles denses orientées et partitionnons le réseau en sites pairs et impairs, que l'on repère respectivement par des points noirs et blancs (fig 10). Alors il existe 4 types d'arêtes suivant l'orientation de la boucle et sa couleur, que nous noterons A,B,C,D. Selon les conventions de la figure 10, des arêtes de type A et B alternent le long des boucles noires, et des arêtes C et D le long des boucles blanches. De plus, on trouve *exactement une* arête de chaque type autour de chaque site du réseau.

À présent, on définit le modèle de hauteur d'interface de la façon suivante : sur chaque facette du réseau carré est définie une hauteur (pour l'instant quantité algébrique abstraite), et la différence de hauteur entre facettes

[7] Une telle propriété est fausse pour une boucle appartenant à une classe d'homotopie non triviale, ce qui est possible par les conditions aux limites périodiques. Nous ne tiendrons pas compte ce phénomène, qui est correctement pris en compte dans l'étude de Jacobsen et Kondev.

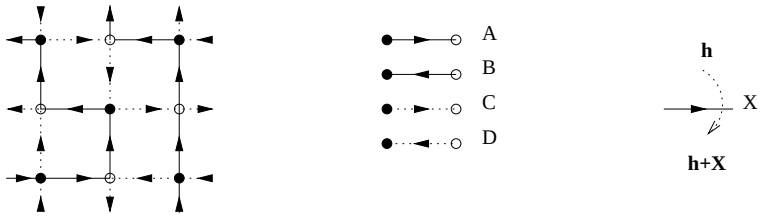


FIG. 10: À gauche, une configuration de boucles denses orientées, sur un réseau bicolorié (damier en noir et blanc). On peut alors distinguer 4 types d'arêtes représentées au milieu. À droite, la convention dite d'Ampère pour la variation de hauteur entre facettes partageant une arête commune de type X.

adjacentes est déterminée par le type de l'arête commune. Plus précisément, on adopte la convention « d'Ampère » : la hauteur est augmentée (respectivement diminuée) d'une quantité $\mathbf{X}$ lorsqu'on traverse une arête de type X orientée vers la gauche (resp. vers la droite) par rapport au déplacement (fig 10 droite).

Pour que la hauteur soit une quantité bien définie, il suffit que la variation totale de hauteur soit nulle lorsqu'on entoure un vertex. Il est facile de voir que ceci ne dégage qu'une seule contrainte sur les quantités algébriques $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$, à savoir $\mathbf{A} - \mathbf{B} + \mathbf{C} - \mathbf{D} = 0$. La situation générique est celle où ces quantités appartiennent à un espace vectoriel de dimension 3. Pour une formulation symétrique, on peut décider que les vecteurs $\mathbf{A}, -\mathbf{B}, \mathbf{C}, -\mathbf{D}$ sont les vecteurs unitaires pointant du centre d'un tétragone régulier vers ses sommets.

Dans la limite continue, Jacobsen et Kondev montrent que cette hauteur tridimensionnelle devient un champ scalaire à trois composantes. De plus, l'action pour ces trois champ est entièrement fixée par les symétries. La théorie conforme obtenue est une théorie de Liouville, et la charge centrale est

$$c_{\mathrm{FPL}}(n,b) = 3 - 6\left(\frac{e_n^2}{1-e_n} + \frac{e_b^2}{1-e_b}\right). \tag{13}$$

Cette théorie peut être interprétée en terme de gaz de Coulomb, et la diminution de charge centrale par rapport à 3 est due à l'introduction d'une charge électrique de fond, destinée à assurer un poids correct aux boucles d'homotopie non nulle. Cette interprétation permet également le calcul des dimensions conformes des opérateurs de la théorie.

4.2 Le modèle GFPL^2

4.2.1 Couplage à la gravité quantique

Le couplage du modèle à la gravité quantique se fait en définissant le modèle non plus sur le réseau carré régulier mais sur un graphe tétravalent arbitraire. Sur un tel graphe, les arêtes sont à nouveau coloriées en noir ou en blanc, et les vertex restent des mêmes types que ceux de la figure 9.

Introduisons un poids x par vertex de type a et y par vertex de type b. La fonction de partition du gaz de boucles denses couplé à la gravité quantique (modèle GFPL) est

$$Z_{\mathrm{GFPL}}(n,b,x,y) = \sum_{\Gamma} \sum_{\mathcal{C}} \frac{1}{|\mathrm{Aut}(\Gamma,\mathcal{C})|} n^{L_n(\mathcal{C})} b^{L_b(\mathcal{C})} x^{v_a(\Gamma)} y^{v_b(\Gamma)} \qquad (14)$$

où la première somme porte sur les graphes Γ planaires tétravalents, la seconde sur les configurations $\mathcal{C}$ de boucles denses tracées sur le graphe Γ, $|\mathrm{Aut}(\Gamma,\mathcal{C})|$ est l'ordre du groupe de symétrie du graphe Γ muni de la configuration $\mathcal{C}$, L_i et v_j comptent respectivement le nombre de boucles de couleur i et le nombre de vertex de type j.

Dans la gravité quantique usuelle, les poids x et y sont tous deux égaux à la « constante cosmologique » qui est conjuguée au nombre total de vertex, c'est-à-dire à l'aire du graphe dual.

Il est à noter une différence notable lors du passage du réseau régulier au réseau aléatoire : on ne peut plus à présent reformuler le modèle en terme de hauteur tridimensionnelle d'une interface. En effet cette reformulation repose sur l'hypothèse cruciale qu'on peut bicolorier le graphe, c'est-à-dire colorier chaque vertex en noir ou blanc de sorte que deux vertex adjacents soient toujours de couleurs opposées. Ceci n'est pas le cas pour un graphe tétravalent arbitraire[8]. Cependant, on peut contourner cette difficulté, en renonçant à la distinction entre les types d'arêtes respectifs A et B, C et D. Plus précisément, on doit imposer les contraintes $\mathbf{A} = \mathbf{B}$, $\mathbf{C} = \mathbf{D}$, et cela est suffisant puisque la convention d'Ampère reste applicable au modèle sur réseau aléatoire.. Il s'ensuit une réduction dimensionnelle du modèle : la hauteur devient une quantité bidimensionnelle au lieu de tridimensionnelle. Au niveau de la théorie conforme sous-jacente, la réduction du nombre de degrés de liberté entraîne une diminution globale de 1 de la charge centrale :

$$c(n,b) = 2 - 6\left(\frac{e_n^2}{1-e_n} + \frac{e_b^2}{1-e_b}\right). \qquad (15)$$

4.2.2 Lien avec les méandres

Les configurations intervenant dans le modèle de gaz de boucles denses couplé à la gravité quantique sont des graphes planaires dont les arêtes sont

[8] Les graphes bicoloriables sont appelés eulériens.

coloriées en noir ou blanc. Elles sont très analogues aux graphes de Feynman du modèle de matrice vu plus haut. La différence principale réside dans le fait qu'on autorise un évitement des boucles par le vertex de type b, tandis que pour le modèle de matrice on impose aux boucles de se couper. En relaxant cette contrainte, on autorise des configurations de méandres où rivière et route se touchent sans se couper, et que l'on appelle méandres *tangents*.

En faisant tendre les fugacités n et b vers 0 pour imposer la connexité de la rivière et de la route, il vient :

$$\lim_{n,b\to 0} \frac{Z_{\mathrm{GFPL}}(n,b,x,y)-1}{nb} = \sum_{\substack{k,p\geq 0\\ k+p\geq 1}} \frac{x^{2k}y^p}{2(2k+p)}\mu_{k,p} \tag{16}$$

où $\mu_{k,p}$ désigne le nombre de classes de méandres fermés tangents avec $2k$ points de croisement et p points de tangence. Le facteur de symétrie a une justification analogue à celui vu dans le cadre des modèles de matrices. On retrouve les méandres fermés usuels dans la limite $y = 0$: $\mu_{k,0} = M_k$.

Mais de plus il est conjecturé, à partir de raisonnements physiques, que les méandres tangents appartiennent à la même classe d'universalité que les méandres. Dans le modèle GFPL, un point de tangence correspond à un vertex de type b. La figure 11 nous montre que ce vertex est sans influence du point de vue du modèle de hauteur, on s'attend donc à ce qu'il soit non-pertinent au sens du groupe de renormalisation. Nous en verrons plus loin les conséquences.

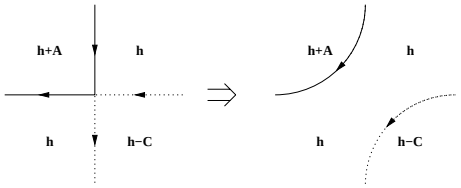


FIG. 11: Non-pertinence du vertex de type b : la hauteur en haut à droite et en bas à gauche sont identiques, on peut donc « défaire » le vertex sans altérer les hauteurs.

4.2.3 Résultats

Quelques résultats sont connus au sujet du couplage entre une théorie conforme avec la gravité quantique bidimensionnelle [28] [29] [30]. En particulier, certains exposants de la théorie couplée sont peuvent être déterminés à partir de la charge centrale et des dimensions d'opérateurs de la théorie non couplée.

En particulier, selon la formule KPZ, la fonction de partition du modèle couplé possède une singularité au voisinage d'une valeur x_c de la constante

cosmologique et la divergence est en :

$$Z(x) \sim (x_c - x)^{2-\gamma(c)} \tag{17}$$

et l'exposant γ dit de susceptibilité de corde est relié à la charge centrale du modèle non couplé selon :

$$\gamma(c) = \frac{c - 1 - \sqrt{(1-c)(25-c)}}{12} \tag{18}$$

Dans le cadre du modèle GFPL, un tel résultat s'applique *a priori* seulement au modèle avec constante cosmologique unique :

$$Z(x) = Z_{\mathrm{GFPL}}(n, b, x = y) \sim (x_c - x)^{2-\gamma(c(n,b))}. \tag{19}$$

Mais comment le vertex de type b est non-pertinent au sens du groupe de renormalisation, les méandres appartiennent à la même classe d'universalité et on s'attend à ce que :

$$\lim_{n,b\to 0} \frac{Z_{\mathrm{GFPL}}(n, b, x, y = 0) - 1}{nb} = \sum_{k=1}^{\infty} \frac{x^{2k}}{4k} M_k \sim (x'_c - x)^{2-\gamma(c(0,0))} \tag{20}$$

où la position du point critique x'_c non universelle a été changée alors que l'exposant n'est pas modifié. Ceci entraîne le comportement asymptotique :

$$M_k \sim \frac{x_c'^{-2k}}{k^{\alpha}} \tag{21}$$

avec

$$\alpha = 2 - \gamma(c(0,0)) = 2 - \gamma(-4) = \frac{29 + \sqrt{145}}{12}. \tag{22}$$

La prédiction d'un tel exposant irrationnel est assez extraordinaire. Numériquement, il a pu être vérifié avec un accord de 5 chiffres après la virgule !

De façon analogue, on peut estimer le comportement asymptotique des polynômes méandriques $M_k(n)$ pour $n \leq 2$ fixé :

$$\lim_{b\to 0} \frac{Z_{\mathrm{GFPL}}(n, b, x, y = 0) - 1}{b} = \sum_{k=1}^{\infty} \frac{x^{2k}}{4k} M_k(n) \sim (x_c(n) - x)^{2-\gamma(c(n,0))} \tag{23}$$

d'où

$$M_k(n) \sim_{k\to\infty} \frac{x_c(n)^{-2k}}{k^{2-\gamma(c(n,0))}} \tag{24}$$

avec $c(n,0) = -1 - 6e_n^2/(1 - e_n)$ et $n = 2cos\pi e_n$.

La formule KPZ pour l'exposant de susceptibilité fournit donc des exposants du comportement asymptotique pour les méandres ou systèmes de méandres fermés. Mais il est possible de tirer d'autre résultats, par exemple pour le cas des semi-méandres.

Dans la partie précédente, nous avons vu que la fonction génératrice des nombres de semi-méandres $\bar{M}_k$ est liée à une fonction de corrélation à deux points du modèle de matrice étudié. Une relation analogue existe dans le cas du modèle GFPL, puisqu'on génère les méandres semi-ouverts sur la sphère par insertion d'une extrémité de ligne noire en deux points distincts. Dans le modèle FPL, l'insertion d'une extrémité de boucle orientée correspond à un opérateur de défaut topologique. Plus précisément, soit donc $\psi_1(z)$ (resp. $\psi_{-1}(z)$) l'opérateur de la théorie conforme FPL réalisant au point z une insertion de boucle noire orientée vers l'extérieur (resp. vers l'intérieur) ; dans l'analogie avec les gaz de Coulomb, un tel opérateur de défaut topologique possède une charge magnétique $\pm 1/2$ (vortex) mais également une charge électrique e_n. La dimension conforme de ces opérateurs est connue est vaut :

$$h_1 = -\frac{e_n^2}{4(1-e_n)} + \frac{1-e_n}{16}. \tag{25}$$

Lors du couplage à la gravité quantique, la théorie KPZ nous apprend que les opérateurs sont « habillés » par la gravité et acquièrent une dimension habillée :

$$\Delta_1 = \frac{\sqrt{1-c+24h_1} - \sqrt{1-c}}{\sqrt{25-c} - \sqrt{1-c}}. \tag{26}$$

Au voisinage du point critique du modèle GFPL, la fonction de corrélation des opérateurs habillés se comporte en :

$$\langle \tilde{\psi}_1 \tilde{\psi}_{-1} \rangle \sim (x_c - x)^{2\Delta_1 - \gamma}. \tag{27}$$

Cette fonction de corrélation correspond aux fonctions génératrices pour les systèmes de méandres avec deux extrémités, les semi-méandres sont obtenus en imposant la connexité de la route et de la rivière, c'est-à-dire dans la limite $n, b \to 0$. La charge centrale est alors $c = -4$ et la dimension conforme des opérateurs est $h_1 = -3/32$ d'où la dimension habillée :

$$\Delta_1 = \frac{1}{2}\frac{\sqrt{11} - \sqrt{5}}{\sqrt{29} - \sqrt{5}} \tag{28}$$

et on tire ainsi le comportement asymptotique du nombre de semi-méandres :

$$\bar{M}_k \sim \frac{{x'_c}^{-k}}{k^{\bar{\alpha}}} \tag{29}$$

avec

$$\bar{\alpha} = 2\Delta_1 - \gamma(-4) + 1 = 1 + \frac{1}{24}\sqrt{11}(\sqrt{29} + \sqrt{5}) \tag{30}$$

et l'« entropie » par pont des semi-méandres est $-\log x'_c$, égale à celle des méandres fermés car ces quantités interviennent toutes deux au niveau du point critique du modèle GFPL dans la limite $n, b \to 0$. À nouveau, l'exposant $\bar{\alpha}$ a pu être vérifié numériquement.

5 Conclusion

Cet exposé reste encore très incomplet. Il est en effet possible de poursuivre la méthode de cette dernière partie pour calculer une infinité d'exposants apparaissant pour des classes de méandres exotiques (méandres avec croisements de rivière, ...).

Mais le but principal était de montrer, à partir de l'exemple des méandres, comment les méthodes récentes développées en physique théorique peuvent trouver des applications dans des problèmes combinatoires d'une formulation extrêment simple. Pourtant, les modèles de gaz de boucles ont été originellement introduits en physique statistique en relation avec des modèles de polymères, et les résultats de KPZ pour la gravité quantique ont été motivés par la théorie des cordes! On perçoit bien la richesse des applications possibles de ces méthodes.

Je dois remercier pour l'élaboration de cet exposé Jesper Jacobsen pour son encadrement, ainsi que Philippe Di Francesco et Emmanuel Guitter qui m'ont initialement présenté ce sujet fort intéressant et qui ont fourni une littérature abondante!

Références

[1] The On-Line Encyclopedia of Integer Sequences, sequence A005315, `http://www.research.att.com/~njas/sequences/`.

[2] Édouard Lucas. *Théorie des nombres I*, page 120. A. Blanchard, Paris, 1961.

[3] André Sainte-Lagüe. *Les Réseaux (ou Graphes)*, page 39. Fasc. 18 de *Mémorial des Sciences Mathématiques*. Gauthier-Villars, Paris, 1926.

[4] André Sainte-Lagüe. *Avec des nombres et des lignes : récréations mathématiques*, page 147. Vuibert, Paris, 1937.

[5] J. Touchard. Contributions à l'étude du problème des timbres poste. *Canad. J. Math.*, **2** (1950) 385–398.

[6] J.E. Koehler. Folding a strip of stamps. *J. Combinat. Theory*, **5** (1968) 135–152.

[7] W. Lunnon. A map-folding problem. *Math. of Computation*, **22** (1968) 193–199.

[8] V.I. Arnol'd. The branched covering of $CP^2 \to S^4$, hyperbolicity and projective topology. *Siberian Math. J.*, **29** (1988) 717–725.

[9] S.K. Lando et A.K. Zvonkin. Meanders. *Selecta Math. Sov.*, **11** (1992) 117–144.

[10] S.K. Lando et A.K. Zvonkin. Plane and projective meanders. *Theor. Comp. Science*, **117** (1993) 227–241.

[11] Henri Poincaré. *Sur un théorème de géométrie.* In *Oeuvres VI*, p. 499–538. Gauthier-Villars, Paris, 1953.

[12] Y.M. Eliashberg. Estimates of number of fixed points of area preserving transformations. *VINITI, Syktyvkar*, **104** (1979).

[13] K.H. Ko et L. Smolinsky. A combinatorial matrix in 3-manifold theory. *Pacific J. Math.*, **149** (1991) 319–336.

[14] P. Rosenstiehl. *Planar permutations defined by two intersecting Jordan curves.* In *Graph Theory and Combinatorics.* Academic Press, London, 1984.

[15] K. Hoffman, K. Mehlhorn, P. Rosenstiehl et R. Tarjan. Sorting Jordan sequences in linear time using level-linked search trees. *Information and Control*, **68** (1986) 170-184.

[16] A. Phillips. The topology of Roman mosaic mazes. In *The visual mind*, pages 65–73. MIT Press, Cambridge, MA, 1993. Voir aussi `http://www.math.sunysb.edu/~tony/mazes/`.

[17] P. Di Francesco, O. Golinelli et E. Guitter. Meander, folding and arch statistics. *Math. Comput. Modelling*, **26** (1997) 97–147. `hep-th/9506030`.

[18] P. Di Francesco, O. Golinelli et E. Guitter. Meanders : a direct enumeration approach. *Nucl. Phys.*, **B 482**[FS] (1996) 497–535. `hep-th/9607039`.

[19] I. Jensen. Enumerations of plane meanders. preprint `cond-mat/9910313`.

[20] O. Golinelli. A Monte-Carlo study of meanders. *Eur. Phys. J.*, **B 14** (2000) 145–155. `cond-mat/9906329`.

[21] P. Di Francesco, O. Golinelli et E. Guitter. Meanders and the Temperley-Lieb algebra. *Commun. Math. Phys.*, **186** (1997) 1–59. `hep-th/9602025`.

[22] P. Di Francesco, O. Golinelli et E. Guitter. Meanders : exact esymptotics. *Nucl. Phys.*, **B 570** (2000) 699–712. `cond-mat/9910453`.

[23] P. Di Francesco, E. Guitter et J.L. Jacobsen. Exact meander asymptotics : a numerical check. *Nucl. Phys.*, **B 580**[FS] (2000) 757–795. `cond-mat/0003008`.

[24] P. Di Francesco. Folding and coloring problems in mathematics and physics. *Bull. Am. Math. Soc.*, **37** (2000) 251–307.

[25] E. Brézin, C. Itzykson, G. Parisi et J.-B. Zuber. Planar diagrams. *Comm. Math. Phys.*, **69** (1979) 147.

[26] J.L. Jacobsen et J. Kondev. Field theory of compact polymers on the square lattice. *Nucl. Phys.*, **B 532**[FS] (1998) 635–688. `cond-mat/9804048`.

[27] J.L. Jacobsen et J. Kondev. Transition from the compact to the dense phase of two-dimensional polymers. *J. Stat. Phys.*, **96** (1999) 21–48. `cond-mat/9811085`.

[28] V.G. Knizhnik, A.M. Polyakov et A.B. Zamolodchikov. Fractal structure of 2d quantum gravity. *Mod. Phys. Lett.*, **A 3** (1988) 819–826.

[29] F. David. Conformal field theories coupled to 2d quantum gravity in the conformal gauge. *Mod. Phys. Lett.*, **A 3** (1988) 1651–1656.

[30] J. Distler et H. Kawai. Conformal field theory and 2d quantum gravity. *Nucl. Phys.*, **B 321** (1989) 509.

Multisectioning, Rational Poly-Exponential Functions and Parallel Computation.

by

Kevin Hare
B.Math, University of Waterloo, 1997.

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the Department
of
Mathematics & Statistics.

SIMON FRASER UNIVERSITY
February 2001

APPROVAL

Name: Kevin Hare

Degree: Master of Science

Title of thesis: Multisectioning, Rational Poly-Exponential Functions and Parallel Computation.

Examining Committee:

Dr. R. Lockhart
Chair

Dr. J. M. Borwein
Senior Supervisor

Dr. M. Monagan

Dr. L. Goddyn

Dr. A. Gupta
Department of Computing Science
External Examiner

Date Approved: _____

Abstract.

Bernoulli numbers and similar arithmetic objects have long been of interest in mathematics. Historically, people have been interested in different recursion formulae that can be derived for the Bernoulli numbers, and the use of these recursion formulae for the calculation of Bernoulli numbers. Some of these methods, which in the past have only been of theoretical interest, are now practical with the availability of high-powered computation.

This thesis explores some of these techniques of deriving new recursion formulae, and expands upon these methods. The main technique that is explored is that of "*multisectioning*". Typically, the calculation of a Bernoulli number requires the calculation of all previous Bernoulli numbers. The method of multisectioning is such that only a fraction of the previous Bernoulli numbers are needed. In exchange, a more complicated recursion formula, called a "*lacunary recursion formula*", must be derived and used.

Dedication.

I would like to dedicate this thesis to my parents, who always supported me with my interest in mathematics.

Acknowledgments.

I would like to thank my supervisor, Jon Borwein, for all his help and insight with respect to this area of research. Also, I would like to thank Marni Mishna, Cindy Loten and Jeff Graham for their proof reading of my thesis, Greg Fee for all of his suggestions on how to improve my Maple code, and numerous other people both within the CECM, and at SFU who made my time here enjoyable.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and preliminaries.

1.1 Introduction.

Bernoulli numbers and similar arithmetic objects have long been of interest in mathematics. Historically, people have been interested in different recursion formulae that can be derived for the Bernoulli numbers, and the use of these recursion formulae for the calculation of Bernoulli numbers. Some of these methods, which in the past have only been of theoretical interest, are now practical with the availability of high-powered computation.

This thesis explores some of these techniques of deriving new recursion formulae, and expands upon these methods. The main technique that is explored is that of "*multisectioning*". Typically, the calculation of a Bernoulli number requires the calculation of all previous Bernoulli numbers. The method of multisectioning is such that only a fraction of the previous Bernoulli numbers are needed. In exchange, a more complicated recursion formula, called a "*lacunary recursion formula*", must be derived and used.

There is a simple formula for $\zeta(n)$, the "*Riemann zeta function*" evaluated at n, for positive even integers n and for negative odd integers n in terms of the Bernoulli numbers. Also, there are numerous constants, (π^{2n}, $\log 2$, γ - the Euler gamma function, τ - the golden mean, G - Catalan's constant) that admit identities of infinite sums of zeta values. Thus the calculations of Bernoulli numbers can be used for certain high precision evaluations of other constants [6].

Bernoulli numbers were first introduced by Jacques Bernoulli (1654-1705), in the second part of his treatise published in 1713, *Ars conjectandi* ("Art of Conjecturing"). At the time, Bernoulli numbers were used for writing the infinite series expansions of hyperbolic and trigonometric functions [7].

Von Staudt and Clausen independently discovered a rapid means of determining the denominator of the Bernoulli numbers [17]. This is very useful for testing to see if the calculation was done without errors. (Any error will most likely return a result for which the Clausen - von Staudt theorem does not hold.)

Van den Berg was the first to discuss finding recurrence formulae for the Bernoulli numbers with arbitrary sized gaps (1881) [19]. (Gaps of size m implies that only $\frac{1}{m}$-th of the information is required, and is the result of multisectioning by m.) Haussner worked on this again, 12 years later (1893) giving the results in terms of hypergeometric functions [19]. Ramanujan, in 1911, is given credit for first giving the formulae for small gaps explicitly. Ramanujan showed how gaps of size 7 could be found, and explicitly wrote out the recursion for gaps of size 6 [4, 19, 22]. These methods were extended to the Euler numbers in 1914 by Glaisher, who used these to compute the first 27 non-zero Euler numbers [14].

Nielsen in 1922, gave an improved notation from a computational point of view to deal with gaps of large sizes [19].

Lehmer in 1934 extended these methods to Euler numbers, Genocchi numbers, and Lucas numbers (1934) [19], and calculated the 196-th Bernoulli number.

The goal in this thesis is to expand these techniques to much more than just Bernoulli and Euler numbers. In general anything that is in the form $\frac{\sum_{i=1}^{n} p_i(x)e^{\lambda_i x}}{\sum_{j=1}^{m} q_j(x)e^{\mu_j x}}$ for polynomials $p_i(x)$, $q_j(x) \in \mathbb{C}[x]$ and constants λ_i, $\mu_j \in \mathbb{C}$ can have the terms of its exponential generating function calculated quickly via multisectioning. This type of function is called a "*rational poly-exponential function*".

This thesis will be looking at examples that are derived from Bernoulli numbers, such as Euler numbers, Genocchi numbers and Lucas numbers. But there are a large variety of other situations where rational poly-exponential functions occur. Some are listed below:

- $(1+x)(\tan(x)+\sec(x))$ - Boustrophedon transform of sequence 1,1,0,0,0,0,... [21]. Reference number A000756 [25, 26].
- $e^{2x}(\tan(x)+\sec(x))$ - Boustrophedon transform of powers of 2 [21]. Reference number A000752 [25, 26].
- $e^{x}(\tan(x)+\sec(x))$ - Boustrophedon transform of all-1's sequence [21]. Reference number A000667 [25, 26].
- $(1+x)e^{x}(\tan(x)+\sec(x))$ - Boustrophedon transform of natural numbers [21]. Reference number A000737 [25, 26].
- $\frac{e^{-x}}{(1-x)^3}$ - $a(n) = na(n-1) + (n-2)a(n-2)$ [23]. Reference numbers A000153, M1791, N0706 [25, 26].

- $\frac{e^{-x}}{(1-x)^2}$ - $a(n) = na(n-1) + (n-1)a(n-2)$ [11, 23]. Reference numbers A000255, M2905, N1166 [25, 26].
- $\frac{e^{x}}{(1-x)^2}$ - $\sum_{k=0}^{n}(k+1)!\binom{n}{k}$ [3, 29]. Reference numbers A001339, M2901, N1164 [25, 26].
- $\frac{e^{-x}}{(1-x)^4}$ - $a(n) = na(n-1) + (n-3)a(n-2)$ [23]. Reference numbers A000261, M2949, N1189 [25, 26].
- $\frac{1-e^{x}}{1-2e^{-x}}$ - Simplices in barycentric subdivisions of n-simplex. Reference numbers A002050, M3939, N1622 [25, 26].
- $\frac{1}{2+x-e^{x}}$ - Partition n labeled elements into sets of sizes of at least 2 and order the sets. Reference number A032032 [25, 26].
- The tangent numbers T_n where $\tan z = \sum_{i=0}^{\infty}(-1)^{n+1}\frac{T_{2n+1}z^{2n+1}}{(2n+1)!}$ [5].

These examples, with the exception of the last one, were all found with the help of *The Encyclopedia of Integer Sequences* and its online counterpart [25, 26]. The reference number is the number associated with the sequence within *The Encyclopedia of Integer Sequences*.

Also, although most of the techniques discussed in this thesis are for rational poly-exponential functions in one variable, it is possible to perform multisectioning in a more general setting, such as for the Bernoulli polynomials, or Euler polynomials (the exponential generating function with respect to x of $\frac{xe^{tx}}{e^{x}-1}$ and $\frac{2e^{xt}}{e^{x}+1}$ give the Bernoulli and Euler polynomials respectively as polynomials in t) [2].

The goal of multisectioning by m is to calculate a lacunary recursion formula so that to calculate a term of the exponential generating function of the rational poly-exponential function requires only $\frac{1}{m}$-th of the time and an $\frac{1}{m}$-th of the information when compared with the standard recursion formula. This allows the calculation on m different machines to achieve a theoretical speed up of a factor of m. (In actual fact, experience shows that the speed up will be greater than this, as the reduction in memory requirements will delay thrashing, and the system can better utilize memory management.) Unfortunately for large m it becomes impractical to determine what these lacunary recursion formulae are as the time to determine the recursion formulae and the complexity of these recursion formulae far exceeds the time to calculate these values with smaller gaps.

Hence multisectioning is a method to compute the Bernoulli numbers that does not require any shared memory. This method is limited by the growth in the cost of determining the lacunary recursion formulae. Conversely there are methods which make use of shared memory (or limited message passing) that are not limited by any increase in the complexity of the lacunary recursion formulae. These methods are limited by the effectiveness of the communication between processes. These techniques are called "*recycling methods*" [6].

Included with this thesis are are a description of the computer programs to determine the lacunary recurrence relations for multisectioned poly-exponential functions, programs to determine the lacunary recursion formulae for multisectioned rational poly-exponential function, as well as algorithms to perform these calculations by recycling. For space consideration the actual code was not included within the thesis. These programs can be found on the web at [1]. This is all written in Maple [13].

1.2 Outline.

Chapter 2 defines and explores poly-exponential functions. This chapter examines some closure properties and metrics upon these functions. As well, this chapter looks at some examples of multisectioning functions of this type.

Rational poly-exponential functions are defined and explored in Chapter 3. Again some closure properties, and metrics upon these functions are examined. As well, examples of how to calculate the coefficients of the exponential generating functions of rational poly-exponential functions and multisectioned rational poly-exponential functions via their lacunary recursion formulae are looked at.

Chapter 4 examines different techniques of calculating lacunary recursion formulae for multisectioned poly-exponential functions.

Different techniques of calculating lacunary recursion formulae for multisectioned rational poly-exponential functions are examined at in Chapter 5.

Chapter 6 looks at different methods to perform the calculation of the coefficients of the exponential generating functions of rational poly-exponential functions, after the lacunary recursion formulae are determined. These different techniques take advantage of multi-processor computers, and distributed computer networks.

The last chapter, Chapter 7 discusses some of the results of this thesis, and makes some conclusions as to what has been learned as a result of these investigations.

Appendix A is an outline of the code. Appendix B lists the common notation and page references. Appendix C contains a list of definitions along with the page reference where the definition is first made. Appendix D is for the bugs reports of bugs found in Maple during the course of these investigations. The last appendix, Appendix E is the code.

Chapter 2

Poly-exponential functions.

2.1 Poly-exponential functions.

The study of rational poly-exponential functions is begun with the exploration of a simpler model; that of poly-exponential functions. To that end define:

Definition 2.1 (Poly-exponential function.) *Let* λ_1, ..., $\lambda_n \in \mathbb{C}$ *be constants and* $p_1(x)$, ..., $p_n(x) \in \mathbb{C}[x]$ *be polynomials. Then*

$$\sum_{i=1}^{n} p_i(x) e^{\lambda_i x},$$

*is a "*poly-exponential function*". Denote the set of all such functions by* $\mathcal{P}$.

This definition along with Lemma 2.1 and Theorem 2.1 are generalization of examples found in Wilf's *Generating Functionology* [30].

Many results for poly-exponential functions can be extended to ratios of poly-exponential functions, thus allowing a simpler setting for developing techniques for the calculations that are the goal of this thesis. Section 2.2 examines the relationship between exponential generating functions and poly-exponential functions. In Section 2.3 the recurrence polynomial corresponding to a linear recurrence relation is defined and explored. Section 2.4 examines in detail the structure and some of the substructure of $\mathcal{P}$, defining both $\mathcal{P}^{R_1,R_2}$ and $\mathcal{P}_{R_1,R_2}$ ($\mathcal{P}^{R_1,R_2}$ and $\mathcal{P}_{R_1,R_2}$ being subrings of $\mathcal{P}$ where the certain coefficients lie within R_1 or R_2). The relationship between two subrings of $\mathcal{P}$, $\mathcal{P}^{R_1,R_2}$ and $\mathcal{P}_{R_1,R_2}$, and showing that these subrings are distinct are shown in Section 2.5. (The subrings are defined by restricting the coefficients to certain rings.) In Section 2.6 some metrics of complexity are introduced for the functions in $\mathcal{P}$, and the relationships between these metrics, with

each other and with standard operations such as addition or multiplication are explored. Section 2.7 contains three detailed examples. The last section, Section 2.8, summarizes the main points of this chapter into a final theorem.

2.2 Exponential generating functions.

The main result of this section is the detailing of the relationship between poly-exponential functions and exponential generating functions.

Lemma 2.1 *Let $s(x)$ be a complex valued function. Then $s(x)$ can be written as an exponential generating function $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$, where the b_i satisfies an N-term linear recurrence relation with constant terms if and only if s(x) can be written as $\sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$ for polynomials $p_i(x) \in \mathbb{C}[x]$ and non-zero constants $\lambda_i \in \mathbb{C}$.*

Proof: Let $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ where the b_i satisfy the linear recurrence relation $b_i = \beta_1 b_{i-1} + ... + \beta_N b_{i-N}$, $\beta_N \neq 0$ for $i \geq N$. Let λ_1, ..., λ_N be roots of the polynomial $x^N - \beta_1 x^{N-1} - ... - \beta_N$ (not necessarily distinct). It is worth noting here that $\lambda_i \neq 0$ for all i. From a standard result on linear recurrence relations [16], it follows that $b_j = \sum_{i=1}^{N} \alpha_i j^{(r_i)} \lambda_i^{j-r_i}$ for some $r_i \in \mathbb{Z}$, and some $\alpha_i \in \mathbb{C}$. Here the notation of Comtet [10] is used, where $j^{(r)} = j(j-1)(j-2)...(j-r+1)$ and $j^{(0)} = 1$. Thus:

$$\begin{aligned} s(x) &= \sum_{j=0}^{\infty} b_j \frac{x^j}{j!} = \sum_{j=0}^{\infty} \sum_{i=1}^{N} \frac{\alpha_i j^{(r_i)} \lambda_i^{j-r_i} x^j}{j!} = \sum_{i=1}^{N} \sum_{j=0}^{\infty} \alpha_i x^{r_i} (\frac{j^{(r_i)} \lambda_i^{j-r_i} x^{j-r_i}}{j!}) \\ &= \sum_{i=1}^{N} \alpha_i x^{r_i} \sum_{j=0}^{\infty} (\frac{j^{(r_i)} \lambda_i^{j-r_i} x^{j-r_i}}{j!}) = \sum_{i=1}^{N} \alpha_i x^{r_i} \sum_{j=r_i}^{\infty} (\frac{\lambda_i^{j-r_i} x^{j-r_i}}{(j-r_i)!}) = \sum_{i=1}^{N} \alpha_i x^{r_i} e^{\lambda_i x}. \end{aligned}$$

Now combine the $\alpha_i x^{r_i}$ which have the same λ_i, and relabel to get $s(x) = \sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$, where the λ_i are distinct and non-zero.

To prove the other direction, let $t(x) = \sum_{j=1}^{m} q_j(x)e^{\mu_j x}$, where $\mu_j \neq 0$, $\mu_j \in \mathbb{C}$ and $q_j(x) \in \mathbb{C}[x]$ are polynomials. Consider the polynomial:

$$P(x) = \prod_{j=1}^{n} (x - \mu_j)^{\deg(q_j(x))} = x^n - \alpha_1 x^{n-1} - ... - \alpha_n.$$

Then $t(x) = \sum_{j=0}^{\infty} d_j \frac{x^j}{j!}$ where the d_j satisfies the n term linear recurrence relation $d_j = \alpha_1 d_{j-1} + ... + \alpha_n d_{j-n}$. Later, in Section 2.3 it will be shown that $P(x)$ is the recurrence polynomial of $t(x)$.

■

Theorem 2.1 *Let $s(x)$ be a complex valued function. Then $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ where there exists an m, such that for $i > m$ the b_i satisfy an N-term linear recurrence relation with constant terms if and only if $s(x) \in \mathcal{P}$.*

Proof: First consider $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ where after some m, the b_i satisfy an N-term linear recurrence relation. A degree m polynomial can be extracted, say $p_0(x)$ $(= \sum_{i=0}^{m} \beta_i \frac{x^i}{i!})$ such that the resulting $\bar{b}_i$ $(= b_i - \beta_i)$ satisfy an N-term linear recurrence relation. Then by Lemma 2.1 $s(x)$ can be written as:

$$s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} = \sum_{i=0}^{\infty} \bar{b}_i \frac{x^i}{i!} + p_0(x) = \sum_{i=1}^{n} p_i(x) e^{\lambda_i x} + p_0(x) e^{0x},$$

for some polynomials $p_i(x)$ and constants λ_i.

Similarly, if $t(x) = \sum_{j=1}^{m} q_j(x) e^{\mu_j x} + p_0(x)$, for polynomials $p_0(x)$, $q_j(x)$, and non-zero constants μ_j, by Lemma 2.1, $t(x)$ can be rewritten as:

$$t(x) = \sum_{j=0}^{\infty} d_j \frac{x^j}{j!} + p_0(x) = \sum_{j=0}^{\infty} \bar{d}_j \frac{x^j}{j!},$$

where the d_j satisfy an N-term linear recurrence relation and where the $\bar{d}_j$ (which are derived by adding the d_j to the coefficients of the polynomial $p_0(x)$) satisfy an N-term linear recurrence relation for $j \geq N + \deg(p_0(x))$.

■

Example 1 *Consider the following example in Maple. For more information about the Maple code, see Appendix A. For the Maple code see Appendix E. The Maple code and help files (including information about syntax) are available on the web at [1].*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the function $s_1(x) = x + x\,e^x$. Converting this to an exponential generating function gives:

```
> \mapleinline{active}{1d}{s[1] := x + x * exp(x):}{%
> }
```

```
> \mapleinline{active}{1d}{convert_egf(s[1], b, x);}{%
> }
```

$$\mathrm{b}(x) = 2\,\mathrm{b}(x-1) - \mathrm{b}(x-2),\ b,\ x,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 2,\ \mathrm{b}(2) = 2,\ \mathrm{b}(3) = 3]$$

So $s_1(x)$ can be written as $\sum_{i=0}^{\infty} \frac{b_i\, x^i}{i!}$ where $b_i = 2\, b_{i-1} - b_{i-2}$, with $b_0 = 0$, $b_1 = 2$, $b_2 = 2$ and $b_3 = 3$.

Example 2 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

*Consider the function $s_2(x) = \sum_{i=0}^{\infty} \frac{b_i\, x^i}{i!}$, where $b_i = b_{i-1} + b_{i-2}$ with $b_0 = 0$ and $b_1 = 1$. These b_i are the "*Fibonacci numbers*" [2]. Converting this to a poly-exponential function gives.*

```
> \mapleinline{active}{1d}{s[2] := b(x) = b(x-1) + b(x-2), b, x,
> [b(0) = 0, b(1) = 1];}{%
> }
```

$$s_2 := \mathrm{b}(x) = \mathrm{b}(x-1) + \mathrm{b}(x-2),\, b,\, x,\, [\mathrm{b}(0) = 0,\, \mathrm{b}(1) = 1]$$

```
> \mapleinline{active}{1d}{convert_pe(s[2]);}{%
> }
```

$$-\frac{1}{5}\sqrt{5}\, e^{(x\,(1/2-1/2\sqrt{5}))} + \frac{1}{5}\sqrt{5}\, e^{(x\,(1/2+1/2\sqrt{5}))},\, x$$

So this can be written as a poly-exponential function, as demonstrated above.

2.3 The recurrence polynomial.

Identifying linear recurrence relations with polynomials will be useful for the further exploration of poly-exponential functions and rational poly-exponential functions. To this end define:

Definition 2.2 (Recurrence polynomial $P^s(x)$.) *Let $s(x) \in \mathcal{P}$, where $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$, where the b_i satisfy an N-term linear recurrence relation for all b_i, $i \geq m+N$, say $b_i = \alpha_1 b_{i-1} + ... + \alpha_N b_{i-N}$. For $m \geq 1$ assume that for $i = m+N-1$, that $b_i \neq \alpha_1 b_{i-1} + ... + \alpha_N b_{i-N}$. Define the "*recurrence polynomial*" $P^s(x)$ by:*

$$P^s(x) = x^m (x^N - \alpha_1 x^{N-1} - ... - \alpha_{N-1} x - \alpha_N).$$

Example 3 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Again consider $s_1(x) = x + e^x\, x$ from Example 1. This example determines what $s_1(x)$'s recurrence polynomial is.

```
> \mapleinline{active}{1d}{s[1] := x + exp(x)*x;}{%
> }
```

$$s_1 := x + x\,e^x$$

```
> \mapleinline{active}{1d}{egf := convert_egf(s[1], b, x);}{%
> }
```

$$egf := \mathrm{b}(x) = 2\,\mathrm{b}(x-1) - \mathrm{b}(x-2),\ b,\ x,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 2,\ \mathrm{b}(2) = 2,\ \mathrm{b}(3) = 3]$$

```
> \mapleinline{active}{1d}{convert_poly(egf);}{%
> }
```

$$x^4 - 2\,x^3 + x^2$$

In contrast consider a random polynomial, and determine what its linear recurrence relation would be.

```
> \mapleinline{active}{1d}{poly := randpoly(x);}{%
> }
```

$$poly := -55\,x^5 - 37\,x^4 - 35\,x^3 + 97\,x^2 + 50\,x + 79$$

```
> \mapleinline{active}{1d}{convert_rec(poly,b,x);}{%
> }
```

$$\mathrm{b}(x) = -\frac{37}{55}\,\mathrm{b}(x-1) - \frac{7}{11}\,\mathrm{b}(x-2) + \frac{97}{55}\,\mathrm{b}(x-3) + \frac{10}{11}\,\mathrm{b}(x-4) + \frac{79}{55}\,\mathrm{b}(x-5)$$

The recurrence polynomial $P^s(x)$ is defined in this way so that it will contain information about when a linear recurrence relation is valid. This construction was suggested by my supervisor, Jon Borwein partly because a useful corollary follows from this definition as a result.

Corollary 1 *If $s(x) \in \mathcal{P}$, $s(x) = \sum_{i=1}^{n} p_i(x) e^{\lambda_i x}$, with n distinct λ_i, then:*

$$\deg(P^s(x)) = \sum_{i=1}^{n} (\deg(p_i(x)) + 1).$$

Later, it will be show that this corollary also follows from Lemma 2.5 and is related to the definition of $deg^P(s(x))$ as given in Definition 2.7.

Let $s(x) \in \mathcal{P}$, $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$. It is possible to find more than one linear recurrence relation for the b_i. For example $b_i = b_{i-1} + b_{i-2}$ and $b_i = 2b_{i-2} + b_{i-3}$ are both valid linear recurrence relations for the Fibonacci numbers. Next it is shown how to avoid the ambiguity of which recurrence polynomial or linear recurrence relation to use.

Define the "*length*" of a linear recurrence relation to be the degree of the recurrence polynomial associated with it. (Later it is shown that this is equivalent to the metric deg^P.) Consider the

minimal integer $n \geq 0$ such that there is a linear recurrence relation of length n; this gives a unique lower bound to the length of a linear recurrence relation. From this it can be shown that this minimal linear recurrence relation is unique, for if there were two different linear recurrence relations of length N,

$$\begin{aligned} b_i &= \alpha_1 b_{i-1} + ... + \alpha_N b_{i-N} \\ \text{and } b_i &= \beta_1 b_{i-1} + ... + \beta_N b_{i-N}, \end{aligned}$$

then

$$0 = (\alpha_1 - \beta_1) b_{i-1} + ... + (\alpha_N - \beta_N) b_{i-N},$$

which has non-zero terms, hence is a smaller linear recurrence relation, which is a contradiction.

Therefore from the comments above, and the results of Corollary 1, assume that $P^s(x)$ is the unique smallest polynomial associated with the unique linear recurrence relation of minimal length associated with $s(x) \in \mathcal{P}$.

If $P(x)$ and $Q(x)$ are two recurrence polynomials associated with the linear recurrence relation of $s(x) \in \mathcal{P}$ (not necessarily minimal) then $\gcd(P(x), Q(x))$ is also associated with $s(x)$. In fact, any polynomial $P(x)$ such that $P^s(x) | P(x)$ will yield a linear recurrence relation for $s(x)$, albeit not one of minimal length.

2.4 The structure of $\mathcal{P}$.

As yet, $\mathcal{P}$ has only been looked at as a collection of functions. However $\mathcal{P}$ has an internal structure. The main result of this section is to show that $\mathcal{P}$ is a ring. As well, some subrings of $\mathcal{P}$ are examined. Some of the consequences of this are re-examined in Section 4.6 in which calculations over different subrings of $\mathcal{P}$ and $\mathcal{R}$ (to be defined in Chapter 3) are made.

To the best of my knowledge, the subrings of $\mathcal{P}$ in this section have never been examined before, and the results in this section are new.

Definition 2.3 ($\mathcal{P}_{R_1,R_2}$.) *Let R_1 and R_2 be subrings of $\mathbb{C}$. Define*

$$\mathcal{P}_{R_1,R_2} = \{s(x) \in \mathcal{P} : s(x) = \sum_{i=1}^{n} p_i(x) e^{\lambda_i x}, \lambda_i \in R_1, p_i(x) \in R_2[x]\}.$$

Definition 2.4 ($\mathcal{P}^{R_1,R_2}$.) *Let R_1 and R_2 be subrings of $\mathbb{C}$. Define*

$$\mathcal{P}^{R_1,R_2} = \{s(x) \in \mathcal{P} : s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}, P^s(x) \text{ factors in } R_1[x], b_i \in R_2\}.$$

The main result of this section is to show that $\mathcal{P}_{R_1,R_2}$ and $\mathcal{P}^{R_1,R_2}$ are both rings. First some preliminary definitions are made to help discuss multisectioning. The process of multisectioning has had a long history, including Ramanujan, Lehmer, Glaisher [14, 19, 22]. For a more detailed describe of the history, see Section 1.1.

Definition 2.5 *Define* $\omega_m = e^{\frac{2\pi i}{m}}$.

Definition 2.6 (Multisectioning.) *Let* $f(x)$ *be a function acting on a subset of* $\mathbb{C}$. *Define* $f_m^q(x) = \frac{1}{m}\sum_{i=0}^{m-1} \omega_m^{-iq}\ f(\omega_m^i x)$.

The term "*multisectioning*" is used to describe this process [24]. To say a function $s(x)$ is "*multisectioned by* m" means that $s_m^q(x)$ is being discussed for some q. To say a function $s(x)$ is "*multisectioned by* m *at* q" means that the function $s_m^q(x)$ is being discussed. The term "*lacunary recurrence relation*" is used to describe the linear recurrence relation of a poly-exponential function that has been multisectioned [24].

If $s(x) \in \mathcal{P}$, then it follows that $s_m^q(x) \in \mathcal{P}$. Let $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$, then:

$$\begin{aligned} s_m^q(x) &= \frac{1}{m}\sum_{k=0}^{m-1} \omega_m^{-kq} s(\omega_m^k x) = \frac{1}{m}\sum_{k=0}^{m-1} \omega_m^{-kq} \sum_{i=0}^{\infty} b_i \frac{x^i \omega_m^{ki}}{i!} = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}\frac{1}{m}\sum_{k=0}^{m-1} \omega_m^{-kq}\omega_m^{ki} \\ &= \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}\frac{1}{m}\sum_{k=0}^{m-1} \omega_m^{-kq+ki}. \end{aligned}$$

By noticing that $\frac{1}{m}\sum_{k=0}^{m-1} \omega_m^{-kq+ki}$ is equal to 1 if and only if $q \equiv i \pmod m$ and 0 otherwise, this simplifies to

$$s_m^q(x) = \sum_{i=0}^{\infty} b_{mi+q} \frac{x^{mi+q}}{(mi+q)!}.$$

So the process of multisectioning will isolate certain terms within the power series.

Consider a poly-exponential functions, say $t(x) = \sum_{i=1}^{n} p_i(x) e^{\lambda_i x}$, then a simple calculation shows that $t_m^q(x)$ has the form:

$$t_m^q(x) = \frac{1}{m}\sum_{j=0}^{m-1}\sum_{i=0}^{n} \omega_m^{-jq} p_i(x\omega_m^{-j}) e^{\lambda_i x \omega_m^{-j}}.$$

Rewriting this as $t_m^q(x) = \sum_{j=1}^{\bar{n}} \bar{p}_j(x) e^{\mu_j x}$, shows that, the recurrence polynomial of $t_m^q(x)$ is:

$$P^{t_m^q(x)}(x) = \prod_{j=1}^{\bar{n}} (x-\mu_j)^{\deg(\bar{p}_j(x))}.$$

The set $\{u_j : j = 1...n\}$ is independant of q, (they will run through $\lambda_i \omega_m^j$). By multisectioning, it is possible that the roots will be of a different multiplicity, hence giving a different recurrence polynomial

(as shown in the example below). But to do this, a sub-component of the poly-exponential function needs to have a symmetry when shifting by ω_m around the origin, which would result in a different degree for some $\bar{p}_i(x)$. (For example $e^x - e^{-x}$ has a symmetry which shifting by $\omega_2 = -1$ about the origin, as $e^x - e^{-x} = -1(e^{-1x} - e^{-1\times -x})$. For a more detailed discussions of symmetries, see Section 5.4.) For example when multisectioning by two, then the function would need to have an even component or an odd component. The probability of this happening is not very great (measure zero) so, as long as something is known about $s(x)$, then the fact that the recurrence for $s^q_m(x)$ is likely the same as $s^{\bar{q}}_m(x)$ can be taken advantage of; by simplifying the calculation of the lacunary recurrence relation of $s^{\bar{q}}_m(x)$ to checking if the lacunary recurrence relation of $s^q_m(x)$ is valid for the first few initial values.

Example 4 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

This is an example of a poly-exponential function, which when multisectioned by 2 will give a different linear recurrence relation if it is multisectioned at 0 or at 1. Consider the function $\mathrm{s}(x) = e^x + e^{(-x)} + e^{(2\,x)} - e^{(-2\,x)}$.

```
> \mapleinline{active}{1d}{s := exp(x)+exp(-x)+exp(2*x)-exp(-2*x);}{%
> }
```

$$\mathrm{s} := e^x + e^{(-x)} + e^{(2\,x)} - e^{(-2\,x)}$$

```
> \mapleinline{active}{1d}{`pe/ms`(s, f, x, 2, 0);}{%
> }
```

$$\mathrm{f}(x) = \mathrm{f}(x-2),\ f,\ x,\ [\mathrm{f}(0) = 2, \mathrm{f}(1) = 0]$$

```
> \mapleinline{active}{1d}{`pe/ms`(s, f, x, 2, 1);}{%
> }
```

$$\mathrm{f}(x) = 4\,\mathrm{f}(x-2),\ f,\ x,\ [\mathrm{f}(0) = 0, \mathrm{f}(1) = 4]$$

In the first case the linear recurrence relation is $\mathrm{f}(x) = \mathrm{f}(x-2)$ *and in the second* $\mathrm{f}(x) = 4\,\mathrm{f}(x-2)$.

The notation of Herstein [18] is used with respect to rings and subrings. Let A be a subset of $\mathbb{C}$. Then $\langle A\rangle$ is the smallest subring of $\mathbb{C}$ that contains A. Denote $A^{-1} = \{a^{-1} : a \in A\}$. Let R_1 and R_2 be subrings of $\mathbb{C}$. Denote $R_1R_2 = \{a_1a_2 : a_1 \in R_1 \text{ and } a_2 \in R_2\}$.

Next some closure properties for $\mathcal{P}^{R_1,R_2}$ and $\mathcal{P}_{R_1,R_2}$ are collected.

Lemma 2.2 *Let* R_1, R_2, R_3, R_4 *and* R_5 *be subrings of* $\mathbb{C}$. *If* $s(x) \in \mathcal{P}_{R_1,R_2}$, $t(x) \in \mathcal{P}_{R_3,R_4}$ *and* $\alpha \in R_5$, *then:*

1. $s(x)t(x) \in \mathcal{P}_{\langle R1,R3\rangle,R_2R_4}$,
2. $s(x)+t(x) \in \mathcal{P}_{\langle R_1,R_3\rangle,\langle R_2,R_4\rangle}$,
3. $s'(x) \in \mathcal{P}_{R_1,\langle R_1,R_2\rangle}$,
4. $\int_0^x s(y)dy \in \mathcal{P}_{R_1,\langle \mathbb{Q}R_2,R_1^{-1}R_2\rangle}$,
5. $s(\alpha x) \in \mathcal{P}_{R_1R_5,\langle R_2,R_2R_5\rangle}$,
6. $s_m^q(x) \in \mathcal{P}_{\langle \omega_m\rangle R_1,\langle \frac{1}{m}\rangle\langle \omega_m\rangle R_2}$.

Proof: Assume that $s(x)=\sum_{i=1}^n p_i(x)e^{\lambda_i x}$, and $t(x)=\sum_{j=1}^m q_j(x)e^{\mu_j x}$ throughout this proof.

1. Observe that:
$$s(x)t(x) = \sum_{i=1}^n p_i(x)e^{\lambda_i x}\sum_{j=1}^m q_j(x)e^{\mu_j x} = \sum_{i=1,j=1}^{i=n,j=m} p_i(x)q_j(x)e^{(\lambda_i+\mu_j)x}.$$
Then $p_i(x)q_j(x) \in R_2R_4[x]$, and $\lambda_i+\mu_j \in \langle R_1,R_3\rangle$. So $s(x)t(x) \in \mathcal{P}_{\langle R_1,R_3\rangle,R_2R_4}$.

2. Observe that:
$$s(x)+t(x) = \sum_{i=1}^n p_i(x)e^{\lambda_i x} + \sum_{j=1}^m q_j(x)e^{\mu_j x}.$$
Both $p_i(x)$ and $q_j(x)$ are in $\langle R_2,R_4\rangle[x]$ and further $\lambda_i,\mu_j \in \langle R_1,R_3\rangle$. Thus $s(x)+t(x) \in \mathcal{P}_{\langle R_1,R_3\rangle,\langle R_2,R_4\rangle}$.

3. Observe that:
$$s'(x) = \sum_{i=1}^n \lambda_i p_i(x)e^{\lambda_i x} + \sum_{i=1}^n p_i'(x)e^{\lambda_i x}.$$
Consequently $p_i'(x), \lambda_i p_i(x) \in \langle R_2, R_1R_2\rangle[x]$ and that $\lambda_i \in R_1$. Thus $s'(x) \in \mathcal{P}_{R_1,\langle R_2,R_1R_2\rangle}$.

4. Re-index the function $s(x)$ so that $s(x)=\sum_{i=1,\lambda_i\neq 0}^n \alpha_i x^{r_i}e^{\lambda_i x} + \sum_{i=0}^m \beta_i x^i$, where $\lambda_i \in R_1$, α_i, $\beta_i \in R_2$ and $r_i \in \mathbb{Z}$, $r_i \geq 0$. Then:
$$\begin{aligned}
\int_0^x s(y)dy &= \int_0^x \sum_{i=1,\lambda_i\neq 0}^n \alpha_i y^{r_i}e^{\lambda_i y} + \sum_{i=0}^m \beta_i y^i dy \\
&= \sum_{i=1,\lambda_i\neq 0}^n \int_0^x \alpha_i y^{r_i}e^{\lambda_i y}dy + \sum_{i=0}^m \int_0^x \beta_i y^i dy \\
&= \sum_{i=1,\lambda_i\neq 0}^n \alpha_i \sum_{j=0}^{r_i} \frac{r_i! x^{r_i-j}e^{\lambda_i x}(-1)^j}{\lambda_i^{j+1}(j-1)!} + \sum_{i=0}^m \frac{\beta_i x^{i+1}}{i+1} \\
&= \sum_{i=1,\lambda_i\neq 0}^n \alpha_i e^{\lambda_i x} \sum_{j=0}^{r_i} \frac{r_i! x^{r_i-j}(-1)^j}{\lambda_i^{j+1}(j-1)!} + \sum_{i=0}^m \frac{\beta_i x^{i+1}}{i+1}.
\end{aligned}$$

The case $\lambda_i \neq 0$ gives that the coefficients are contained in the subring $\langle R_2 R_1^{-1} \rangle$. In the case $\lambda_i = 0$, the coefficients are contained in $\mathbb{Q}R_2$. The λ_i are still in R_1. Therefore $\int_0^x s(y)dy \in \mathcal{P}_{R_1, \langle \mathbb{Q}R_2, R_2 R_1^{-1} \rangle}$.

5. Notice that $s(\alpha x) = \sum_{i=1}^n p_i(\alpha x) e^{\alpha \lambda_i x}$. So $p_i(\alpha x) \in \langle R_2, R_2 R_5 \rangle$. Further $\alpha \lambda_i \in R_1 R_5$, so $s(\alpha x) \in \mathcal{P}_{R_1 R_5, \langle R_2, R_2 R_5 \rangle}$.

6. By combining part 2 and part 5 of this lemma $s_m^q(x)$ can be written as:

$$\frac{1}{m} \sum_{i=1}^m \omega_m^{-iq} s(\omega_m^i x) \in \mathcal{P}_{\langle \langle \omega_m \rangle R_1, \langle \omega_m^2 \rangle R_1, \ldots \langle \omega_m^m \rangle R_1 \rangle, \langle \langle \frac{1}{m} \omega_m \rangle R_2, \langle \frac{1}{m} \omega_m^2 \rangle R_2, \ldots \langle \frac{1}{m} \omega_m^m \rangle R_2 \rangle}.$$

This simplifies to $\mathcal{P}_{\langle \omega_m \rangle R_1, \langle \frac{1}{m} \rangle \langle \omega_m \rangle R_2}$.

■

Lemma 2.3 *Let R_1, R_2, R_3, R_4 and R_5 be subrings of $\mathbb{C}$. If $s(x) \in \mathcal{P}^{R_1, R_2}$, $t(x) \in \mathcal{P}^{R_3, R_4}$, and $\alpha \in R_5$ then:*

1. $s(x)t(x) \in \mathcal{P}^{\langle R_1, R_3 \rangle, R_2 R_4}$,
2. $s(x) + t(x) \in \mathcal{P}^{\langle R_1, R_3 \rangle, \langle R_2, R_4 \rangle}$,
3. $s'(x) \in \mathcal{P}^{R_1, R_2}$,
4. $\int_0^x s(y)dy \in \mathcal{P}^{R_1, R_2}$,
5. $s(\alpha x) \in \mathcal{P}^{R_1 R3, \langle R_2, R_2 R_3 \rangle}$,
6. $s_m^q(x) \in \mathcal{P}^{R_1 \langle \omega_m \rangle, R_2}$.

Proof: Again, assume that $s(x) = \sum_{i=0}^\infty b_i \frac{x^i}{i!} = \sum_{i=1}^n p_i(x) e^{\lambda_i x}$, and $t(x) = \sum_{j=0}^\infty d_j \frac{x^j}{j!} = \sum_{j=1}^m q_j(x) e^{\mu_j x}$ throughout this proof.

1. Consider:

$$s(x)t(x) = \sum_{i=1}^n p_i(x) e^{\lambda_i x} \sum_{j=1}^m q_j(x) e^{\mu_j x} = \sum_{i=1 j=1}^{i=n, j=m} p_i(x) q_j(x) e^{(\lambda_i + \mu_j) x}.$$

From this $\prod_{i=1, j=1}^{i=n, j=m} (x - \lambda_i - \mu_j)^{\deg(p_i(x)) + \deg(q_i(x))}$ is a recurrence polynomial (not necessarily minimal) for $s(x)t(x)$. Hence:

$$P^{st}(x) | \prod_{i=1, j=1}^{i=n, j=m} (x - \lambda_i - \mu_j)^{\deg(p_i(x)) + \deg(q_i(x))}.$$

This splits in $\langle R_1, R_3\rangle$. Further,

$$s(x)t(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} \sum_{j=0}^{\infty} d_j \frac{x^j}{j!} = \sum_{j=0}^{\infty} \sum_{i=0}^{j} \frac{b_{j-i} d_i}{(j-i)!i!} x^j = \sum_{j=0}^{\infty} \sum_{i=0}^{j} b_{j-i} d_i \binom{j}{i} \frac{x^j}{j!}.$$

Therefore the coefficients are in $R_2 R_4$. Thus $s(x)t(x) \in \mathcal{P}^{\langle R_1, R_3\rangle, R_2 R_4}$.

2. Consider:

$$s(x) + t(x) = \sum_{i=1}^{n} p_i(x) e^{\lambda_i x} + \sum_{j=1}^{m} q_j(x) e^{\mu_j x}.$$

The polynomial $\prod_{i=1}^{n} (x-\lambda_i)^{\deg(p_i(x))} \prod_{j=1}^{m} (x-\mu)^{\deg(q_i(x))}$ is a recurrence polynomial for $s(x)+t(x)$ (not necessarily minimal). Hence $P^{s+t}(x) | P^s(x) P^t(x)$. Thus $P^{s+t}(x)$ will split in $\langle R_1, R_3\rangle$. Further,

$$s(x) + t(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} + \sum_{j=0}^{\infty} d_j \frac{x^j}{j!} = \sum_{i=0}^{\infty} (b_i + d_i) \frac{x^i}{i!}.$$

Where the $b_i + d_i$ are in $\langle R_2, R_4\rangle$. Hence $s(x) + t(x) \in \mathcal{P}^{\langle R_1, R_3\rangle, \langle R_2, R_4\rangle}$.

3. If $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$. then

$$s'(x) = \sum_{i=1}^{\infty} b_i \frac{x_{i-1}}{(i-1)!} = \sum_{i=0}^{\infty} b_{i+1} \frac{x_i}{i!}.$$

Hence the coefficients are in the same ring as before, hence in R_2.

Now consider $s(x) = \sum_{i=1}^{n} p_i(x) e^{\lambda_i x}$. This implies that:

$$s'(x) = \sum_{i=1}^{n} q_i(x) e^{\lambda_i x},$$

where $\deg(p_i(x)) = \deg(q_i(x))$ if $\lambda_i \neq 0$ and $\deg(p_i(x)) = \deg(q_i(x)) + 1$ if $\lambda_i = 0$. Thus $P^{s'}(x) = \prod_{i=1}^{n} (x-\lambda_i)^{\deg(q_i(x))}$. Therefore if there exists a λ_i that is equal to 0, then $P^{s'}(x) = P^s(x)x$, and otherwise $P^{s'}(x) = P^s(x)$. So $P^{s'}(x)$ splits over the same field as $P^s(x)$. Hence $s'(x) \in \mathcal{P}^{R_1, R_2}$.

4. By observing that

$$\int_0^x s(y)dy = \int_0^x \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} = \sum_{i=0}^{\infty} \int_0^x \frac{b_i}{i!} y^i dy = \sum_{i=0}^{\infty} \frac{b_i}{(i+1)!} y^{i+1} |_0^x = \sum_{i=1}^{\infty} \frac{b_{i-1}}{i!} x^i,$$

it follows that all the coefficients of $\int_0^x s(y)dy$ are in R_2

Now consider $s(x) = \sum_{i=1}^{n} p_i(x) e^{\lambda_i x}$. This implies that:

$$\int_0^x s(y)dy = \sum_{i=1}^{n} q_i(x) e^{\lambda_i x},$$

where $\deg(p_i(x)) = \deg(q_i(x))$ if $\lambda_i \neq 0$ and $\deg(p_i(x)) + 1 = \deg(q_i(x))$ if $\lambda_i = 0$. From this $P^{s'}(x) = \prod_{i=1}^{n}(x-\lambda_i)^{\deg(q_i(x))}$. Consequently if there exists a λ_i that is equal to 0, then $P^{\int_0^x s(y)dy}(x) = P^s(x)$, and otherwise $P^{\int_0^x s(y)dy}(x) = P^s(x)$. So $P^{\int_0^x s(y)dy}(x)$ splits over the same field as $P^s(x)$. Thus $\int_0^x s(y)dy \in \mathcal{P}^{R_1,R_2}$.

5. It can be seen that
$$s(\alpha x) = \sum_{i=0}^{\infty} b_i \frac{\alpha^i x^i}{i!},$$
Consequently all of the $b_i\alpha^i \in \langle R_2, R_2R_5\rangle$.

The next aim is to find a linear recurrence relation for the $b_i\alpha^i$. Now if:
$$P^s(x) = x^n - \beta_1 x^{n-1} - ... - \beta_n = \prod_{i=1}^{n}(x-\lambda_i)^{\deg(p_i(x))},$$
and letting $c_i = b_i\alpha^i$ then:
$$\frac{c_i}{\alpha^i} = \beta_1 \frac{c_{i-1}}{\alpha^{i-1}} + ... + \beta_n \frac{c_{i-n}}{\alpha^{m-i}}.$$
Multiplying through by α^i gives:
$$c_i = \alpha\beta_1 c_{i-1} + ... + \alpha^n \beta_n c_{i-n},$$
which gives:
$$P^{s(\alpha x)}(x) = x^n - \beta_1\alpha x^{n-1} - ...\alpha^n\beta_n,$$
this factors as:
$$P^{s(\alpha x)}(x) = \prod_{i=1}^{n}(x-\lambda_i\alpha)^{\deg(p_i(x))}.$$
Therefore $P^{s(\alpha x)}(x)$ splits over R_1R_5. So $s(\alpha x) \in \mathcal{P}^{R_1R_5,\langle R_2,R_2R_5\rangle}$.

6. By combining part 2 and part 5 of this lemma $s_m^q(x)$ can be written as:
$$\frac{1}{m}\sum_{i=1}^{m} \omega_m^{-i*q} s(\omega_m^i x) \in \mathcal{P}^{\langle\langle\omega_m\rangle R_1,\langle\omega_m^2\rangle R_1,...\langle\omega_m^m\rangle R_1\rangle,\langle\frac{1}{m}\langle\omega_m\rangle R_2,\langle\frac{1}{m}\omega_m^2\rangle R_2,...\langle\frac{1}{m}\omega_m^m\rangle R_2\rangle}.$$
But this will simplify to $\mathcal{P}^{\langle\omega_m\rangle R_1,\langle\frac{1}{m}\rangle\langle\omega_m\rangle R_2}$.

An even tighter bound on the coefficients can be seen by noticing that:
$$s_m^q(x) = \sum_{i=0}^{\infty} b_{mi+q}\frac{x^{mi+q}}{(mi+q)!}.$$
From this all the coefficients in the resulting formula are still contained within the ring R_2. Hence $s_m^q(x) \in \mathcal{P}^{R_1\langle\omega_m\rangle,R_2}$.

■

In the proof of Lemma 2.3, some intermediate results were obtained, which are summarized below:

Corollary 2 *Let R_1, R_2 and R_3 be subrings of $\mathbb{C}$. If $s(x)$, $t(x) \in \mathcal{P}$ such that $P^s(x) \in R_1[x]$, $P^t(x) \in R_2[x]$ and $\alpha \in R_3$. Then:*

1. $P^{st}(x) \in R_1R_2[x]$,
2. $P^{s+t}(x) \in R_1R_2[x]$,
3. $P^{s'}(x) \in R_1[x]$ *(infact* $P^s(x) = P^{s'}(x)$ *or* $P^s(x) = xP^{s'}(x)$*)*,
4. $P^{\int_0^x s(y)dy}(x) \in R_1[x]$ *(infact* $P^{\int_0^x s(y)dy}(x) = P^s(x)$ *or* $P^{\int_0^x s(y)dy}(x) = xP^s(x)$*)*,
5. $P^{s(\alpha x)}(x) \in \langle R_1, R_3 \rangle[x]$,
6. $P^{s^q_m}(x) \in R_1[x]$.

These results will be useful later in Chapters 4 and 5. These results imply that the calculations can normally be assumed to be over "nice" rings such as the integers or rationals.

Corollary 3 *Let R_1 and R_2 be subrings of $\mathbb{C}$. Then $\mathcal{P}_{R_1,R_2}$ and $\mathcal{P}^{R_1,R_2}$ are both rings.*

Example 5 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the function $\mathrm{s}(x) = \sum_{i=0}^{\infty} \frac{b_i\, x^i}{i!}$, *where* $b_i = b_{i-1} + b_{i-2}$ *and* $b_0 = 2$, $b_1 = 1$. *These are the Lucas numbers as defined by Graham, Knuth and Patashnik, [16, 24]. To avoid confusion with the Lucas numbers as defined by Lehmer, call these the "*Lucas numbers, type I*" . Now multisection s(x) by 4 at 1.*

```
> \mapleinline{active}{1d}{s := b(x) = b(x-1) + b(x-2), b, x, [b(0) =
> 2, b(1) = 1];}{%
> }
```

$$\mathrm{s} := \mathrm{b}(x) = \mathrm{b}(x-1) + \mathrm{b}(x-2),\ b,\ x,\ [\mathrm{b}(0) = 2,\ \mathrm{b}(1) = 1]$$

First convert this to poly-exponential form:

```
> \mapleinline{active}{1d}{pe := convert_pe(s)[1];}{%
> }
```

$$pe := e^{(x\,(1/2-1/2\,\sqrt{5}))} + e^{(x\,(1/2+1/2\,\sqrt{5}))}$$

Now multisection the poly-exponential function using the formula as given in Definition 2.6.

```
> \mapleinline{active}{1d}{ms  := 1/4*sum(subs(x=x*exp(2*Pi*I/4*i),
> pe)*exp(-2*Pi*I/4*i), i=0..3);}{%
> }
```

$$ms := \frac{1}{4}\,e^{(x\,\%2)} + \frac{1}{4}\,e^{(x\,\%1)} - \frac{1}{4}\,I\,(e^{(I\,x\,\%2)} + e^{(I\,x\,\%1)}) - \frac{1}{4}\,e^{(-x\,\%2)} - \frac{1}{4}\,e^{(-x\,\%1)} + \frac{1}{4}\,I\,(e^{(-I\,x\,\%2)} + e^{(-I\,x\,\%1)})$$

$$\%1 := \frac{1}{2} + \frac{1}{2}\,\sqrt{5}$$

$$\%2 := \frac{1}{2} - \frac{1}{2}\,\sqrt{5}$$

Now convert this back into an exponential generating function.

```
> \mapleinline{active}{1d}{convert_egf(ms, b, x);}{%
> }
```

$$\mathrm{b}(x) = -\mathrm{b}(x-8) + 7\,\mathrm{b}(x-4),\ b,\ x,$$
$$[\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 1,\ \mathrm{b}(2) = 0,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = 0,\ \mathrm{b}(5) = 11,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = 0]$$

From this it follows that $s_4^1(x) = \sum_{i=0}^{\infty} \frac{b_i\,x^i}{i!}$, *where* $b_i = 7\,b_{i-4} - b_{i-8}$ *and* $b_1 = 1$, $b_5 = 11$ *and* $b_i = 0$ *if* $i \neq 1 \bmod 4$. *So* $s_4^1(x) = \sum_{i=0}^{\infty} \frac{b_{4\,i+1}\,x^{(4\,i+1)}}{(4\,i+1)!}$.

Alternatively there is automated code to achieve the same result, using this naive method.

```
> \mapleinline{active}{1d}{`egf/ms/naive`(s,4,1);}{%
> }
```

$$\mathrm{b}(x) = -\mathrm{b}(x-8) + 7\,\mathrm{b}(x-4),\ b,\ x,$$
$$[\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 1,\ \mathrm{b}(2) = 0,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = 0,\ \mathrm{b}(5) = 11,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = 0]$$

This is a relationship for the Lucas numbers, type I that is only concerned with $b_1, b_5, b_9, \ldots$

Automating the process of multisectioning is covered in Chapter 4.

2.5 Hierarchy of $\mathcal{P}$.

While many results for both $\mathcal{P}^{R_1,R_2}$ and $\mathcal{P}_{R_1,R_2}$ have been obtained, it is not yet clear as to how these two rings relate to each other. This section shows that they are in fact different sets of rings. Further an inclusion relationship between the rings is shown.

Theorem 2.2 *Let R_1 and R_2 be subrings of $\mathbb{C}$. Then the following inclusion relationships of the subrings of $\mathcal{P}$ hold.*

1. $\mathcal{P}_{R_1,R_2} \subseteq \mathcal{P}^{R_1,\langle R_1R_2,R_2\rangle}$,
2. $\mathcal{P}^{R_1,R_2} \subseteq \mathcal{P}_{R_1,R_2\langle R_1^{-1},R_1\rangle}$.

Proof:

1. Let $s(x) \in \mathcal{P}_{R_1,R_2}$, $s(x) = \sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$, $p_i(x) \in R_2[x]$, $\lambda_i \in R_1$. Noticing that $P^s(x) = \prod_i^n (x-\lambda_i)^{\deg(p_i(x))}$, demonstrates that $P^s(x)$ splits in $R_1[x]$. Now notice that $b_i = s^{(i)}(0)$, the i-th derivative of $s(x)$. But $s^{(i)}(x) \in \mathcal{P}_{R_1,\langle R_1R_2,R_2\rangle}$ by Lemma 2.2 part 3. Evaluating at 0 gives $b_i \in \langle R_1R_2, R_2\rangle$, hence $\mathcal{P}_{R_1,R_2} \subseteq \mathcal{P}^{R_1,\langle R_1R_2,R_2\rangle}$.

2. Let $s(x) \in \mathcal{P}^{R_1,R_2}$, $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$. By definition $P^s(x)$ splits in R_1. Lemma 2.1 implies that if $s(x) = \sum_i^n \alpha_i x^{(r_i)} e^{\lambda_i x}$ then all the λ_i are in R_1.

   Again from Lemma 2.1 it follows that:

$$b_j = \sum_{i=1}^{n} j^{(r_i)} \lambda_i^{j-r_i} \alpha_{r_i},$$

   where $\lambda_i \in R_1$, $j^{(r_i)} \in \mathbb{Z}$ and $b_j \in R_2$, and $j^{(r)} = (j)(j-1)...(j-r+1)$. A solution to these equations using Gaussian elimination requires only addition, subtraction, multiplication, and division of elements in R_1. Thus $\alpha_{r_i} \in R_2\langle R_1^{-1}, R_1\rangle$. Hence $\mathcal{P}^{R_1,R_2} \subseteq \mathcal{P}_{R_1,R_2\langle R_1^{-1},R_1\rangle}$.

■

Consider the following examples, which shows that the two rings are distinct.

Example 6 *Consider $s(x) = e^{\sqrt{2}x} \in \mathcal{P}_{\mathbb{Q}(\sqrt{2}),\mathbb{Z}}$. Now $s'(x) = \sqrt{2}e^{\sqrt{2}x}$, and $\sqrt{2} \notin \mathbb{Z}$ implies that $s'(x) \notin \mathcal{P}_{\mathbb{Q}(\sqrt{2}),\mathbb{Z}}$. But all rings of the form $\mathcal{P}^{R_1,R_2}$ are closed under differentiation (Lemma 2.3). Hence there do not exist rings R_1, R_2 such that $\mathcal{P}_{\mathbb{Q}(\sqrt{2}),\mathbb{Z}} = \mathcal{P}^{R_1,R_2}$.*

Example 7 *Consider $\mathcal{P}^{\mathbb{Z},\mathbb{Z}}$. The goal here is to show that there do not exist rings R_1, R_2 such that $\mathcal{P}^{\mathbb{Z},\mathbb{Z}} = \mathcal{P}_{R_1,R_2}$. Consider the exponential generating function $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ with the linear relation $b_i = 3cb_{i-1} - 2c^2b_{i-2}$, for $c \in \mathbb{Z}$. If b_0, $b_1 \in \mathbb{Z}$, then $s(x) \in \mathcal{P}^{\mathbb{Z},\mathbb{Z}}$. But this is equivalent to $s(x) = \alpha_1 e^{cx} + \alpha_2 e^{2cx}$, where $\alpha_1 = 2b_0 - \frac{b_1}{c}$ and $\alpha_2 = -b_0 + \frac{b_1}{c}$. Hence α_1 can be any arbitrary rational in $\mathbb{Q}$, say $\frac{p}{q}$, by picking $b_0 = 0$, $b_1 = -p$ and $c = q$. Thus if $\mathcal{P}^{\mathbb{Z},\mathbb{Z}} \subseteq \mathcal{P}_{R_1,R_2}$, then $\mathbb{Z} \subseteq R_1$ (as R_1 must contain arbitrary c, where $c \in \mathbb{Z}$) and $\mathbb{Q} \subseteq R_2$. Now $\mathcal{P}^{\mathbb{Z},\mathbb{Z}}$ is a strict subset of $\mathcal{P}_{\mathbb{Z},\mathbb{Q}}$, as $\frac{1}{2} \in \mathcal{P}_{\mathbb{Z},\mathbb{Q}}$ and $\frac{1}{2} \notin \mathcal{P}^{\mathbb{Z},\mathbb{Z}}$. Hence there do not exist rings R_1 and R_2, such that $\mathcal{P}^{\mathbb{Z},\mathbb{Z}} = \mathcal{P}_{R_1,R_2}$.*

Corollary 4 *If F_1 is a subfield of $\mathbb{C}$, and $R_1 \subseteq F_1$ is a subring of $\mathbb{C}$ then $\mathcal{P}_{R_1,F_1} = \mathcal{P}^{R_1,F_1}$.*

2.6 Some complexity bounds.

Understanding the complexity of the functions being manipulated is useful for doing computations on $s(x) \in \mathcal{P}$. To this end some metrics of complexity are defined. These metrics have been looked at in the past but not in such a generalized fashion. Typically they would be applied to a particular problem, such as the Bernoulli numbers [9].

Definition 2.7 *Let $s(x) \in \mathcal{P}$, where $s(x) = \sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$. Define the following metrics:*

1. $deg^d(s(x)) = \max(\deg(p_i(x)))$,
2. $deg^P(s(x)) = \deg(P^s(x))$.

Example 8 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

This example determines what $deg^d(s(x))$ and $deg^P(s(x))$ are for various s(x). This example uses the automated code described in appendix A.

First consider the function from Example 1.

```
> \mapleinline{active}{1d}{s[1] := x + x * exp(x);}{%
> }
```

$$s_1 := x + x\,e^x$$

```
> \mapleinline{active}{1d}{`pe/metric/d`(s[1],x);}{%
> }
```

$$1$$

Recalls that $P^{s_1}(x) = x^4 - 2x^3 + x^2$.

```
> \mapleinline{active}{1d}{`pe/metric/P`(s[1],x);}{%
> }
```

$$4$$

Next, consider the Fibonacci numbers from Example 2.

```
> \mapleinline{active}{1d}{s[2] := b(x) = b(x-1)+b(x-2),b,x,
> [b(0)=0,b(1)=1];}{%
> }
```

$$s_2 := \mathrm{b}(x) = \mathrm{b}(x-1) + \mathrm{b}(x-2),\ b,\ x,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 1]$$

```
> \mapleinline{active}{1d}{`egf/metric/d`(s[2]);}{%
> }
```

$$0$$

```
> \mapleinline{active}{1d}{`egf/metric/P`(s[2]);}{%
> }
```

$$2$$

These metrics are of use later on in Chapter 4 and 5. In those two chapters, upper bounds for functions under different operations are required.

Lemma 2.4 *Let $s(x)$, $t(x) \in \mathcal{P}$, and $\alpha \neq 0$ a constant. Then:*

1. $deg^d(s(x)t(x)) = deg^d(s(x)) + deg^d(t(x))$,
2. $0 \leq deg^d(s(x) + t(x)) \leq \max(deg^d(s(x)), deg^d(t(x)))$,
3. $deg^d(s(x)) - 1 \leq deg^d(s'(x)) \leq deg^d(s(x))$,
4. $deg^d(s(x)) \leq deg^d(\int_0^x s(y)dy) = deg^d(s(x)) + 1$,
5. $deg^d(s(\alpha x)) = deg^d(s(x))$,
6. $0 \leq deg^d(s_m^q(x)) \leq deg^d(s(x))$.

Proof: Write $s(x) = \sum_{i=1}^n p_i(x)e^{\lambda_i x}$ and $t(x) = \sum_{j=1}^m q_j(x)e^{\mu_j x}$ for the remainder of this proof.

1. Notice that:
$$deg^d(s(x)t(x)) = deg^d(\sum_{i=1,j=1}^{i=n,j=m} p_i(x)q_j(x)e^{(\lambda_i+\mu_j)x}).$$
Denote $I = \{i : \deg(p_i(x)) = deg^d(s(x))\}$ and $J = \{j : \deg(q_i(x)) = deg^d(t(x))\}$. Pick $\lambda = \max_{i\in I}(\lambda_i)$ and $\mu = \max_{j\in J}(\lambda_i)$. (The maximum is taken lexigraphically, for example, for two complex numbers α and β, α is greater than β if the real component of α is greater than that of β, or if the real component of α and β are equal, and the imaginary component of α is greater than that of β.)
Consequently the polynomial associated with $\lambda + \mu$ is of degree $deg^d(s(x)) + deg^d(t(x))$. Thus $deg^d(s(x)t(x)) = deg^d(s(x)) + deg^d(t(x))$.
2. The upper bound is clear, and taking $s(x) = -t(x)$ gives the lower bound.

3. Notice that:

$$deg^d(s'(x)) \quad = \quad deg^d(\frac{d}{dx}\sum_{i=1}^{n} p_i(x)e^{\lambda_i x}) = deg^d(\sum_{i=1}^{n}(\lambda_i p_i(x) + p_i'(x))e^{\lambda_i x}).$$

Notice that $\deg(p_i(x)\lambda_i + p_i'(x)) = \deg(p_i(x))$ if $\lambda_i \neq 0$, and is equal to $\deg(p_i(x)) - 1$ if $\lambda_i = 0$. Hence $deg^d(s'(x)) = deg^d(s(x))$ or $deg^d(s(x)) - 1$.

4. Part 4 of Lemma 2.2 shows that:

$$deg^d(\int_0^x s(y)dy) \quad = \quad deg^d(\int_0^x \sum_{i=1}^{n} p_i(y)e^{\lambda_i y}dy) = deg^d(\sum_{i=1}^{n} q_i(x)e^{\lambda_i x}).$$

Where $\deg(q_i(x)) = \deg(p_i(x))$ if $\lambda_i \neq 0$ and $\deg(q_i(x)) = \deg(p_i(x)) + 1$ if $\lambda_i = 0$. Thus $deg^d(\int_0^x s(y)dy) = deg^d(s(x))$ or $deg^d(s(x)) + 1$.

5. Observe that:

$$deg^d(s(\alpha x)) = deg^d(\sum_{i=1}^{n} p_i(\alpha x)e^{\lambda_i \alpha x}).$$

As $\deg(p_i(\alpha x)) = \deg(p_i)$ it follows that $deg^d(s(\alpha x)) = deg^d(s)$.

6. Part 2 and part 5 of this lemma, in combination shows that $deg^d(s_m^q(x)) \leq deg^d(s(x))$. If $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ and $b_i = 0$ whenever $i \equiv q \pmod m$, then $s_m^q(x) = 0$. Hence $deg^d(s_m^q(x)) = 0$ in this case.

■

Lemma 2.5 *Let $s(x), t(x) \in \mathcal{P}$, and α a constant. Then:*

1. $deg^P(s(x)t(x)) \leq deg^P(s(x))deg^P(t(x))$,
2. $0 \leq deg^P(s(x) + t(x)) \leq deg^P(s(x)) + deg^P(t(x))$,
3. $deg^P(s(x)) - 1 \leq deg^P(s'(x)) = deg^P(s(x))$,
4. $deg^P(s(x)) \leq deg^P(\int_0^x s(y)dy) = deg^P(s(x)) + 1$,
5. $deg^P(s(\alpha x)) = deg^P(s(x))$,
6. $0 \leq deg^P(s_m^q(x)) \leq m \times deg^P(s(x))$.

Proof: Write $s(x) = \sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$ and $t(x) = \sum_{j=1}^{m} q_j(x)e^{\mu_j x}$ for the remainder of this proof.

1. Noticing that $P^{st}(x) | \prod_{i=1,j=1}^{i=n,j=m}(x - \lambda_i - \mu_j)^{\deg(p_i(x))+\deg(q_i(x))}$ as shown in Lemma 2.3 gives $deg^P(s(x)t(x)) \leq deg^P(s(x))deg^P(t(x))$.

2. Observing that $P^{s+t}(x)|P^s(x)P^t(x)$, as shown in Lemma 2.3 gives $deg^P(s(x)+t(x)) \leq deg^P(s(x))+deg^P(t(x))$. The lower bound comes from taking $s(x) = -t(x)$.

3. In Lemma 2.3 it was shown that $P^{s'}(x) = P^s(x)$ or $xP^{s'}(x) = P^s(x)$. Hence $deg^P(s'(x)) = deg^P(s(x))$ or $deg^P(s(x)) - 1$.

4. In Lemma 2.3 it was shown that $P^{\int_0^x s(y)dy}(x) = P^s(x)$ or $P^{\int_0^x s(y)dy}(x) = xP^s(x)$. Hence $deg^P(\int_0^x s(y)dy) = deg^P(s(x))$ or $deg^P(s(x)) + 1$.

5. If $P^s(x) = \prod_{i=1}^n (x-\lambda_i)^{\deg(p_i(x))}$ then $P^{s(\alpha x)}(x) = \prod_{i=1}^n (x-\alpha\lambda_i)^{\deg(p_i(x))}$, which has the same degree. Hence $deg^P(s(\alpha x)) = deg^P(s(x))$.

6. Part 2 and part 5 of this lemma, in combination shows that $deg^P(\sum_{k=1}^m s(\omega_m^k x)\omega_m^{-qk}) \leq \sum_{k=1}^m deg^P(s(\omega_m^k x)) = m \times deg^P(s(x))$. The lower bound follows by considering the same example as is found in Lemma 2.4 part 6.

■

Chapters 4 and 5 typically work with the recurrences instead of with the poly-exponential function directly. These results are useful as they give bounds for the linear recurrence relations. The bound given by the metric deg^P is obvious, and the metric deg^d gives a bound to the multiplicity of roots in the recurrence polynomial.

Now the relationship between the metrics is examined.

Lemma 2.6 *Let $s(x) \in \mathcal{P}$. Then $1 + deg^d(s(x)) \leq deg^P(s(x))$.*

Proof: Write $s(x) = \sum_{i=1}^n p_i(x)e^{\lambda_i x}$ for the remainder of this proof. By Corollary 1 it follows that:

$$1 + deg^d(s(x)) \quad = \quad \max_{i=1}^n(\deg(p_i(x)) + 1) \leq \sum_{i=1}^n (\deg(p_i(x)) + 1) = deg^P(s(x)).$$

Which gives the desired result.

■

2.7 Examples.

In this section, three detailed examples are worked out. That of $s(x) = \sum_{i=1}^n \alpha_i e^{\lambda_i(x)}$ and $t(x) = e^{\lambda x}p(x)$, and the Chebyshev T polynomials.

Example 9 *Consider* $s(x) = \sum_{i=1}^{n} \alpha_i \, e^{\lambda_i(x)}$. *Therefore the recurrence polynomial is* $P^s(x) = \prod_{i=1}^{n}(x-\lambda_i)$. *Denoting* $\beta_k = \sum_{J\subseteq\{\lambda_1,\ldots,\lambda_n\},|J|=k} \prod_{\lambda\in J} \lambda$ *to be the elementary symmetric polynomials of* N *variables gives* $P^s(x) = \sum_{k=0}^{N} x^{N-k}\beta_k(-1)^k$.

Writing $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ *gives a linear recurrence relation for the* b_i *namely* $b_i = \sum_{k=1}^{N} \beta_k \, b_{i-k}(-1)^k$.

The first N *values of the* b_i *must be determined. Note that* $b_i = s^{(i)}(0)$, *the* i*-th derivative of* $s(x)$. *Also* $s(x) = \sum_{i=1}^{n} \alpha_i e^{\lambda_i x}$, *so* $b_i = \sum_{k=1}^{n} \alpha_k \lambda_k^i$.

Example 10 *Consider* $t(x) = e^{\lambda x}p(x)$. *So the recurrence polynomial satisfies* $P^t(x) = (x-\lambda)^{\deg(p(x))}$. *Let* $N = \deg(p(x))$ *for convenience. Consequently* $P^t(x) = \sum_{k=0}^{N} \binom{N}{k} x^{N-k}(-\lambda)^k$. *Thus linear recurrence relation is simply:* $b_i = -\sum_{k=0}^{N} \binom{N}{k} b_{i-k}(-\lambda)^k$.

Now determine the first N *values of the* b_i. *Observe that* $b_i = t^{(i)}(0)$, *the* i*-th derivative of* $t(x)$. *Further, observe that* $t(x) = e^{\lambda x}p(x)$. *So* $t^{(0)}(0)$ *is simply* $p(0)$. *Next* $t^{(1)}(0) = \lambda p(0) + p'(0)$. *Next* $t^{(2)}(0) = \lambda^2 p(0) + 2\lambda p'(0) + p''(0)$. *In general* $t^{(k)}(0) = \sum_{i=0}^{k} \binom{k}{i} \lambda^{k-i} p^{(i)}(0)$. *If* $p(x) = a_N x^N + \ldots + a_0$, *then this formula for the* b_i *will simplify to* $t^{(k)}(0) = \sum_{i=0}^{k} \binom{k}{i} \lambda^{k-i} i! a_i$.

Thus the linear recurrence relation is $b_i = -\sum_{k=0}^{N} b_{i-k}(-\lambda)^k$ *and where for* $k < N$, $b_i = \sum_{k=0}^{i} \binom{i}{k} \lambda^{i-k} k! a_k$.

Example 11 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

*This example will demonstrate that process of multisectioning can be used where the recurrence has symbolic values rather than simply numeric ones. Consider the "*Chebyshev T polynomials*", as polynomials in* t, *with the recurrence* $T_n = 2\,t\,T_{n-1} - T_{n-2}$ *with initial polynomials* $T_0 = 1$ *and* $T_1 = t$ *[2]. Consider multisectioning this by 5 at 1, to get a recurrence for* T_1, T_6, T_{11}, T_{16}, *...*

```
> \mapleinline{active}{1d}{egf := f(x) = 2*t*f(x-1)-f(x-2),f,x,[f(0)=1,
> f(1)=t];}{%
> }
```

$$egf := \mathrm{f}(x) = 2\,t\,\mathrm{f}(x-1) - \mathrm{f}(x-2),\ f,\ x,\ [\mathrm{f}(0) = 1,\ \mathrm{f}(1) = t]$$

```
> \mapleinline{active}{1d}{`egf/ms`(egf,5,1);}{%
> }
```

$$\begin{aligned}\mathrm{f}(x) = {} & -\mathrm{f}(x-10) + (32\,t^5 - 40\,t^3 + 10\,t)\,\mathrm{f}(x-5),\ f,\ x, [\mathrm{f}(0) = 0,\ \mathrm{f}(1) = t,\ \mathrm{f}(2) = 0,\\ & \mathrm{f}(3) = 0,\ \mathrm{f}(4) = 0,\ \mathrm{f}(5) = 0, \mathrm{f}(6) = \\ & 2\,t\,(2\,t\,(2\,t\,(2\,t^2-1) - t) - 2\,t^2 + 1) - 2\,t\,(2\,t^2 - 1) + t)\\ & - 2\,t\,(2\,t\,(2\,t^2-1) - t) + 2\,t^2 - 1,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 0,\ \mathrm{f}(9) = 0]\end{aligned}$$

```
> \maple inline{active}{1d}{expand([%]);}{%
> }
```

$$[\mathrm{f}(x) = -\mathrm{f}(x-10) + 32\,\mathrm{f}(x-5)\,t^5 - 40\,\mathrm{f}(x-5)\,t^3 + 10\,\mathrm{f}(x-5)\,t,\ f,\ x, [\mathrm{f}(0) = 0,\ \mathrm{f}(1) = t,$$
$$\mathrm{f}(2) = 0,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = 0,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 32\,t^6 - 48\,t^4 + 18\,t^2 - 1,\ \mathrm{f}(7) = 0,$$
$$\mathrm{f}(8) = 0,\ \mathrm{f}(9) = 0]]$$

This example is interesting in that it shows that the λ_i used in the definition of poly-exponential functions, (Definition 2.1) can be symbolic values in the complex numbers, as opposed to just the numeric values.

2.8 Conclusions.

By combining the results of Theorem 2.1, Lemmas 2.3, 2.5 and Corollary 2 the following results are true.

Theorem 2.3 *Let $s(x) \in \mathcal{P}$.*

1. *Then there exists a lacunary recurrence relation for the $mi+q$-th coefficient of $s(x)$'s exponential generating function in terms of the $mj+q$-th coefficient $j = i-N, ..., i-1$, where N is bounded above by $deg^P(s(x))$.*
2. *Moreover if the linear recurrence relation associated with $s(x)$ is such that the associated recurrence polynomial is in $R_1[x]$, then the recurrence polynomial of the new lacunary recurrence relation will also be in $R_1[x]$.*
3. *Furthermore if the linear recurrence relation associated with $s(x)$ is of length N, then the new lacunary recurrence relation will be of length less than or equal to mN, where only $\frac{1}{m}$-th of the terms are non-zero.*

The following corollory was know in [16], but its proof was specific to either the Fibonacci or Lucas type I numbers, and was not the consequence of a more general theorem.

Corollary 5 *The $mi+q$ term of the Fibonacci and Lucas type I numbers can be computed in terms of $mj+q$ term for $j = i-2, i-1$ via a lacunary recurrence relation. Moreover the lacunary recurrence relation will be over $\mathbb{Z}$. Lastly, the lacunary recurrence relation will be of length $2m$ with 2 non-zero terms.*

Chapter 3

Rational poly-exponential functions.

3.1 Rational poly-exponential function.

Some techniques for poly-exponential functions where developed in Chapter 2. This chapter expands the scope of the study to a more general setting; that of ratios of poly-exponential functions. To that end, define:

Definition 3.1 (Rational poly-exponential function.) *Let $s(x)$, $t(x) \in \mathcal{P}$ and $t(x) \neq 0$. Then*

$$\frac{s(x)}{t(x)},$$

*is a "*rational poly-exponential function*". Denote the set of all such functions by $\mathcal{R}$.*

This definition was suggested by my supervisor, Jon Borwein, as a generalization of the Bernoulli numbers. All of the methods Lehmer, or Glaisher [19, 14] to multisectioning the Bernoulli numbers relied only upon the fact that these numbers had "nice" linear recurrence relation to describe the exponential generating function of the numerator and denominator. Definition 3.1 maintains this property, but expands the scope of the results to a much large class of functions. To the best of my knowledge, the results in this chapter are new, in the sense that they have not been done in this degree of generality before.

Section 3.2 shows how to calculate the coefficients of the exponential generating function of functions in $\mathcal{R}$ by use of recursion formulae. Section 3.3 will demonstrate the effects of multisectioning

on functions in $\mathcal{R}$. The structure of $\mathcal{R}$ is studied in Section 3.4, examining different rings, subrings, fields and subfields of $\mathcal{R}$, along with some closure properties. In Section 3.5 the examination of subfields of $\mathcal{R}$ is continued, by exploring how these subfields relate to each other. Some metrics of complexity for functions in $\mathcal{R}$ are investigated in Section 3.6. Section 3.7 contains three worked out examples. The last section, Section 3.8 summarizes the main points of this chapter into a final theorem.

3.2 Recursion formula for functions in $\mathcal{R}$.

The study of rational poly-exponential functions begins by looking at an example of how to calculate the coefficients of the exponential generating function of $\frac{x}{e^x-1}$. These are the "*Bernoulli numbers*" (in even suffix notation) [2].

Example 12 *Define*

$$\sum_{k=0}^{\infty} c_k \frac{x^k}{k!} = \frac{x}{e^x - 1} = \frac{\sum_{i=0}^{\infty} b_i \frac{x^i}{i!}}{\sum_{j=0}^{\infty} d_j \frac{x^j}{j!}}.$$

Then the c_k are the Bernoulli numbers. A simple calculation shows that $b_i = 1$ if $i = 1$ and 0 otherwise. Further $d_j = 0$ if $j = 0$ and 1 otherwise.

Now:

$$\begin{aligned}
\sum_{k=0}^{\infty} c_k \frac{x^k}{k!} &= \frac{\sum_{i=0}^{\infty} b_i \frac{x^i}{i!}}{\sum_{j=0}^{\infty} d_j \frac{x^j}{j!}} \\
\sum_{j=0}^{\infty} d_j \frac{x^j}{j!} \sum_{k=0}^{\infty} c_k \frac{x^k}{k!} &= \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} \\
\sum_{k=0}^{\infty} \sum_{j=0}^{k} d_j \frac{x^j}{j!} c_{k-j} \frac{x^{k-j}}{(k-j)!} &= \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} \\
\sum_{k=0}^{\infty} \sum_{j=0}^{k} \binom{k}{j} d_j c_{k-j} \frac{x^k}{k!} &= \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} \\
\sum_{j=0}^{k} \binom{k}{j} d_j c_{k-j} &= b_k.
\end{aligned}$$

From this a recursion formula for the Bernoulli numbers is derived that, for $k > 2$ gives:

$$\sum_{j=1}^{k} \binom{k}{j} c_{k-j} = 0$$

$$c_{k-1} = \frac{-1}{k}\sum_{j=0}^{k-2}\binom{n}{j}c_j.$$

This is the standard recursion formula used for the Bernoulli numbers, as would be found in [2, 10, 16].

Note 3.1 *It is important to note that the term "linear recurrence relation" is different than that of "recursion formula". A recursion formula is a formula where the n-th term depends on the previous $n-1$ terms, where as a linear recurrence relation only requires the previous N terms of which a linear combination is used to determine the n-th term. Examples 12 gives a recursion formula for the Bernoulli numbers.*

It is not always possible to write $f(x) \in \mathcal{R}$ as $\sum_{i=0}^{\infty} c_i \frac{x^i}{i!}$. In particular if $f(x)$ has a pole at 0, this will not be possible (i.e. $\frac{1}{x}$). The restriction to $f(x) \in \mathcal{R}$ which do not have poles at 0, is closed under addition, differentiation, multiplication, $f(x) \to f(\alpha x)$ and multisectioning, by simply looking at the Taylor series under these operations. Denote this set as $\hat{\mathcal{R}}$ to get this definition:

Definition 3.2 ($\hat{\mathcal{R}}$.) *Define*

$$\hat{\mathcal{R}} = \{f(x) : \lim_{x\to 0}\frac{1}{f(x)} \neq 0, f(x) \in \mathcal{R}\}.$$

3.3 Multisectioning.

This section explores the effects of multisectioning on rational poly-exponential functions. The main result of this section allows for the improvement in the efficiency of calculating the coefficients of exponential generating functions for functions in $\mathcal{R}$.

Lemma 3.1 *If $h(x) \in \mathcal{R}$ then $h_m^q(x)$ can be written as $\frac{s_m^q(x)}{t_m^0(x)}$ where $s(x)$, $t(x) \in \mathcal{P}$.*

Proof: Write $h(x) = \frac{s_h(x)}{t_h(x)}$. Thus:

$$\begin{aligned} h_m^q(x) &= \frac{1}{m}\sum_{i=0}^{m-1}\frac{\omega_m^{-iq}s_h(x\omega_m^i)}{t_h(x\omega_m^i)} = \frac{1}{m}\sum_{i=1}^{m-1}\frac{\omega_m^{-iq}s_h(x\omega_m^i)\prod_{j=1}^{m-1}t_h(x\omega_m^{j+i})}{\prod_{j=0}^{m-1}t_h(x\omega_m^j)} \\ &= \frac{\frac{1}{m}\sum_{i=1}^{m-1}\omega_m^{-iq}s_h(x\omega_m^i)\prod_{j=1}^{m-1}t_h(x\omega_m^{j+i})}{\prod_{j=0}^{m-1}t_h(x\omega_m^j)} = \frac{(s_h(x)\prod_{j=1}^{m-1}t_h(x\omega_m^j))_m^q}{(\prod_{j=0}^{m-1}t_h(x\omega_m^j))_m^0}. \end{aligned}$$

Picking $s(x) = s_h(x)\prod_{i=1}^{m-1}t_h(x\omega_m^i)$ and $t(x) = \prod_{i=0}^{m-1}t_h(x\omega_m^i)$ gives the desired result. It is also worthwhile to note that $t_m^0(x) = t(x)$.

■

Theorem 3.1 *Given a function $f(x) \in \hat{\mathcal{R}}$, $m, q \in \mathbb{Z}$, $0 \leq q < m$, a recursion formula can be found for the $mi+q$-th coefficient of the exponential generating function of $f(x)$ that depends only on the $mj+q$-th coefficient, for $j < i$, and two lacunary recurrence relations.*

Later, in Section 3.8, by combining this theorem, Theorem 3.1, with some later results, Lemmas 3.3, 3.4 and 3.6, an even tighter result will be given, the lengths of these lacunary recurrence relations, will be determined, and the ring that their coefficients will lie will be known.

Proof: Let

$$f(x) = \sum_{i=0}^{\infty} c_i \frac{x^i}{i!} = \frac{s_f(x)}{t_f(x)} = \frac{\sum_{i=0}^{\infty} b_i \frac{x^i}{i!}}{\sum_{j=0}^{\infty} d_j \frac{x^j}{j!}},$$

where $s(x)$, $t(x) \in \mathcal{P}$. Lemma 3.1 gives

$$f_m^q(x) = \sum_{i=0}^{\infty} c_{mi+q} \frac{x^{mi+q}}{(mi+q)!} = \frac{s_m^q(x)}{t_m^0(x)} = \frac{\sum_{i=0}^{\infty} \bar{b}_{mi+q} \frac{x^{mi+q}}{(mi+q)!}}{\sum_{j=0}^{\infty} \bar{d}_{mj} \frac{x^{mj}}{(mj)!}},$$

where s_m^q, $t_m^0 \in \mathcal{P}$, and the $\bar{b}_i$ and the $\bar{d}_j$ satisfy lacunary recurrence relations.

A simple calculation shows that

$$\begin{aligned}
\sum_{i=0}^{\infty} c_{mi+q} \frac{x^{mi+q}}{(mi+q)!} &= \frac{\sum_{i=0}^{\infty} \bar{b}_{mi+q} \frac{x^{mi+q}}{(mi+q)!}}{\sum_{j=0}^{\infty} \bar{d}_{mj} \frac{x^{mj}}{(mj)!}} \\
\sum_{j=0}^{\infty} \bar{d}_{mj} \frac{x^{mj}}{(mj)!} \sum_{i=0}^{\infty} c_{mi+q} \frac{x^{mi+q}}{(mi+q)!} &= \sum_{i=0}^{\infty} \bar{b}_{mi+q} \frac{x^{mi+q}}{(mi+q)!} \\
\sum_{i=0}^{\infty} \sum_{j=0}^{i} \binom{mi+q}{mj} \bar{d}_{mj} c_{m(i-j)+q} \frac{x^{mi+q}}{(mi+q)!} &= \sum_{i=0}^{\infty} \bar{b}_{mi+q} \frac{x^{mi+q}}{(mi+q)!} \\
\sum_{j=0}^{i} \binom{mi+q}{mj} \bar{d}_{mj} c_{m(i-j)+q} &= \bar{b}_{mi+q}.
\end{aligned}$$

Picking $s = \min\{j : d_{mj} \neq 0\}$ gives:

$$\begin{aligned}
\sum_{j=s}^{i} \binom{mi+q}{mj} \bar{d}_{mj} c_{m(i-j)+q} &= \bar{b}_{mi+q} \\
\binom{mi+q}{ms} \bar{d}_{ms} c_{m(i-s)+q} &= \bar{b}_{mi+q} - \sum_{j=s+1}^{i} \binom{mi+q}{mj} \bar{d}_{mj} c_{m(i-j)+q} \\
c_{m(i-s)+q} &= \frac{1}{\binom{mi+q}{ms} \bar{d}_{ms}} (\bar{b}_{mi+q} - \sum_{j=s+1}^{i} \binom{mi+q}{mj} \bar{d}_{mj} c_{m(i-j)+q}).
\end{aligned}$$

Let $k = i - s$, to get

$$\begin{aligned} c_{mk+q} &= \frac{1}{\binom{m(s+k)+q}{ms}\bar{d}_{ms}}(\bar{b}_{m(k+s)+q} - \sum_{j=s+1}^{k+s}\binom{m(k+s)+q}{mj}\bar{d}_{mj}c_{m((k+s)-j)+q}) \\ &= \frac{1}{\binom{m(s+k)+q}{ms}\bar{d}_{ms}}(\bar{b}_{m(k+s)+q} - \sum_{j=1}^{k}\binom{m(k+s)+q}{m(j+s)}\bar{d}_{m(j+s)}c_{m(k-j)+q}). \end{aligned}$$

This is a recursion formula for the c_{mk+q} based on the previous c_{mj+q} with $j < k$ and two lacunary recurrence relations for the $\bar{b}_{mi+q}$ and $\bar{d}_{mi}$.

■

The recursion formula associated with $f_m^q(x)$ is called the "*lacunary recursion formula*" [8, 14].

Example 13 *Consider the following example in Maple. For more information about the Maple code, see Appendix A. For the Maple code see Appendix E. The Maple code and help files (including information about syntax) are available on the web at [1].*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider again the Bernoulli numbers $\frac{x}{e^x-1} = \frac{\sum_{i=0}^{\infty}\frac{b_i x^i}{i!}}{\sum_{j=0}^{\infty}\frac{d_j x^j}{j!}}$. *Multisection this by 3 at 1, using the formula, as given in Lemma 3.1. After this, this example will calculate the 1-st, 4-th 7-th and 10-th Bernoulli number, using the formula given in Theorem 3.1.*

Let $s_h(x) = x$ *and* $t_h(x) = e^x - 1$, *and solve for* $\mathrm{s}(x)$ *and* $\mathrm{t}(x)$ *in the theorem.*

```
> \mapleinline{active}{1d}{s[h] := x -> x;}{%
> }
```

$$s_h := x \to x$$

```
> \mapleinline{active}{1d}{t[h] := (x) -> exp(x)-1;}{%
> }
```

$$t_h := x \to e^x - 1$$

```
> \mapleinline{active}{1d}{omega[3] := exp(2*Pi*I/3);}{%
> }
```

$$\omega_3 := -\frac{1}{2} + \frac{1}{2}\,I\,\sqrt{3}$$

From Lemma 3.1 $\mathrm{s}(x) = s_h(x)\,(\prod_{i=1}^{m-1} t_h\,{\omega_m}^i)$, *and* $\mathrm{t}(x) = \prod_{i=0}^{m-1} t_h(x\,{\omega_m}^i)$, *which, for this particular case is:*

```
> \mapleinline{active}{1d}{S := s[h](x) * t[h](x*omega[3]) *
> t[h](x*omega[3]^2);}{%
> }
```

$$S := x\,(e^{(x\,(-1/2+1/2\,I\,\sqrt{3}))} - 1)\,(e^{(x\,(-1/2+1/2\,I\,\sqrt{3})^2)} - 1)$$

```
> \mapleinline{active}{1d}{T :=
> t[h](x)*t[h](x*omega[3])*t[h](x*omega[3]^2);}{%
> }
```

$$T := (e^{x} - 1)\,(e^{(x\,(-1/2+1/2\,I\,\sqrt{3}))} - 1)\,(e^{(x\,(-1/2+1/2\,I\,\sqrt{3})^2)} - 1)$$

Now, determine what the linear recurrence relation for this would be.

```
> \mapleinline{active}{1d}{`pe/ms`(S,b,x,3,1);}{%
> }
```

$$\mathrm{b}(x) = -\mathrm{b}(x-12) + 2\,\mathrm{b}(x-6),\ b,\ x, [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 0,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = -12,\ \mathrm{b}(5) = 0,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = -7,\ \mathrm{b}(8) = 0,\ \mathrm{b}(9) = 0,\ \mathrm{b}(10) = -30,\ \mathrm{b}(11) = 0,\ \mathrm{b}(12) = 0,\ \mathrm{b}(13) = -13]$$

So $s_3^1(x) = \sum_{i=0}^{\infty} \frac{b_i\,x^i}{i!}$, *where* $b_i = b_{i-12} + 2\,b_{i-6}$, *with initial values of* $b_4 = -12$, $b_7 = -7$, $b_{10} = -30$ *and* $b_{13} = -13$.

```
> \mapleinline{active}{1d}{convert_egf(T,d,x);}{%
> }
```

$$\mathrm{d}(x) = \mathrm{d}(x-6),\ d,\ x,\ [\mathrm{d}(0) = 0,\ \mathrm{d}(1) = 0,\ \mathrm{d}(2) = 0,\ \mathrm{d}(3) = 6,\ \mathrm{d}(4) = 0,\ \mathrm{d}(5) = 0]$$

So the bottom linear recurrence relation $t_3^0(x) = \sum_{j=0}^{\infty} \frac{d_j\,x^i}{j!}$, *where* $d_j = d_{j-6}$, *and the initial values are* $d_3 = 6$.

Equally easy the two built-in commands could have been used to do this in the naive fashion.

```
> \mapleinline{active}{1d}{top := `top/ms/naive`(x,exp(x)-1,b,x,3,1);}{%
> }
```

$$top := \mathrm{b}(x) = -\mathrm{b}(x-12) + 2\,\mathrm{b}(x-6),\ b,\ x, [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 0,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = -12,\ \mathrm{b}(5) = 0,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = -7,\ \mathrm{b}(8) = 0,\ \mathrm{b}(9) = 0,\ \mathrm{b}(10) = -30,\ \mathrm{b}(11) = 0,\ \mathrm{b}(12) = 0,\ \mathrm{b}(13) = -13]$$

```
> \mapleinline{active}{1d}{bot := `bottom/ms/naive`(exp(x)-1,d,x,3);}{%
> }
```

$$bot := \mathrm{d}(x) = \mathrm{d}(x-6),\ d,\ x,\ [\mathrm{d}(0) = 0,\ \mathrm{d}(1) = 0,\ \mathrm{d}(2) = 0,\ \mathrm{d}(3) = 6,\ \mathrm{d}(4) = 0,\ \mathrm{d}(5) = 0]$$

Now, to calculate the first few Bernoulli numbers, use the formula as given in Theorem 3.1, first noting that s is equal to 1.

```
> \mapleinline{active}{1d}{Top := `egf/makeproc`(top):}{%
> }

> \mapleinline{active}{1d}{Bot := `egf/makeproc`(bot):}{%
> }

> \mapleinline{active}{1d}{s := 1:}{%
> }

> \mapleinline{active}{1d}{m := 3:}{%
> }

> \mapleinline{active}{1d}{k := 0:}{%
> }

> \mapleinline{active}{1d}{q := 1:}{%
> }

> \mapleinline{active}{1d}{Bernoulli[m * k + q] := 1/binomial(m*(s + k)
> + q, m * s) / Bot(m * s) * }{%
> }

> \mapleinline{active}{1d}{                        (Top(m *(k + s) + q)
> - add(binomial (m *(k + s) + q, }{%
> }

> \mapleinline{active}{1d}{                        m *(j + s)) * Bot(m *
> j) * Bernoulli[m * (k+s-j) + q], }{%
> }

> \mapleinline{active}{1d}{                        j = 1+s .. k+s));}{%
> }
```

$$Bernoulli_1 := \frac{-1}{2}$$

```
> \mapleinline{active}{1d}{k := 1:}{%
> }

> \mapleinline{active}{1d}{Bernoulli[m * k + q] := 1/binomial(m*(s + k)
> + q, m * s) / Bot(m * s) * }{%
> }

> \mapleinline{active}{1d}{                        (Top(m *(k + s) + q)
> - add(binomial (m *(k + s) + q, }{%
> }

> \mapleinline{active}{1d}{                        m *(j + s)) * Bot(m *
> (j + s)) * Bernoulli[m * (k-j) + q], }{%
> }

> \mapleinline{active}{1d}{                        j = 1 .. k));}{%
```

```
> }
```

$$Bernoulli_4 := \frac{-1}{30}$$

```
> \mapleinline{active}{1d}{k := 2:}{%
> }

> \mapleinline{active}{1d}{Bernoulli[m * k + q] := 1/binomial(m*(s + k)
> + q, m * s) / Bot(m * s) * }{%
> }

> \mapleinline{active}{1d}{                          (Top(m *(k + s) + q)
> - add(binomial (m *(k + s) + q, }{%
> }

> \mapleinline{active}{1d}{                          m *(j + s)) * Bot(m *
> (j + s)) * Bernoulli[m * (k-j) + q], }{%
> }

> \mapleinline{active}{1d}{                          j = 1 .. k));}{%
> }
```

$$Bernoulli_7 := 0$$

```
> \mapleinline{active}{1d}{k := 3:}{%
> }

> \mapleinline{active}{1d}{Bernoulli[m * k + q] := 1/binomial(m*(s + k)
> + q, m * s) / Bot(m * s) * }{%
> }

> \mapleinline{active}{1d}{                          (Top(m *(k + s) + q)
> - add(binomial (m* (k + s) + q, }{%
> }

> \mapleinline{active}{1d}{                          m *(j + s)) * Bot(m *
> (j + s)) * Bernoulli[m * (k-j) + q], }{%
> }

> \mapleinline{active}{1d}{                          j = 1 .. k));}{%
> }
```

$$Bernoulli_{10} := \frac{5}{66}$$

There is automated code to get the same result.

```
> \mapleinline{active}{1d}{A := `calcul/normal`(10, Top, Bot, 3, 1):}{%
> }

> \mapleinline{active}{1d}{seq(A[3 * i + 1], i = 0 ..3);}{%
> }
```

$$\frac{-1}{2}, \frac{-1}{30}, 0, \frac{5}{66}$$

3.4 The structure of $\mathcal{R}$.

Like $\mathcal{P}$, this section will show that $\mathcal{R}$ has a rich structure. To explore this structure, this section first makes some definitions for subsets of $\mathcal{R}$ analogous to the Definitions 2.3 and 2.4 for $\mathcal{P}$.

Definition 3.3 ($\mathcal{R}^{R_1,R_2}, \mathcal{R}_{R_1,R_2}$.) *Let R_1 and R_2 be subrings of $\mathbb{C}$. Denote $\mathcal{R}^{R_1,R_2}$ ($\mathcal{R}_{R_1,R_2}$) to be the subset of $\mathcal{R}$, such that all elements can be written in for the form $\frac{s(x)}{t(x)}$ with $s(x)$, $t(x) \in \mathcal{P}^{R_1,R_2}$ ($s(x)$, $t(x) \in \mathcal{P}_{R_1,R_2}$).*

Definition 3.4 ($\hat{\mathcal{R}}^{R_1,R_2}, \hat{\mathcal{R}}_{R_1,R_2}$.) *Let R_1 and R_2 be subrings of $\mathbb{C}$. Define $\hat{\mathcal{R}}^{R_1,R_2} = \mathcal{R}^{R_1,R_2} \cap \hat{\mathcal{R}}$ and $\hat{\mathcal{R}}_{R_1,R_2} = \mathcal{R}_{R_1,R_2} \cap \hat{\mathcal{R}}$.*

First collect some closure properties for $\mathcal{R}$.

Lemma 3.2 *Let R_1, R_2, R_3, and R_4 be subrings of $\mathbb{C}$ and let $h(x) \in \mathcal{R}_{R_1,R_2}$ and $g(x) \in \mathcal{R}_{R_3,R_4}$ then:*

1. $g(x)h(x) \in \mathcal{R}_{\langle R_1,R_3\rangle,R_2R_4}$,
2. $g(x) + h(x) \in \mathcal{R}_{\langle R_1,R_3\rangle,R_2R_4}$,
3. $h'(x) \in \mathcal{R}_{R_1,\langle R_1,R_2\rangle}$,
4. $h_m^q(x) \in \mathcal{R}_{R_1\langle\omega_m\rangle,R_2\langle\omega_m\rangle}$.

Proof: For convenience, write $h(x) = \frac{s_h(x)}{t_h(x)}$, with $s_h(x)$, $t_h(x) \in \mathcal{P}_{R_1,R_2}$, and $g(x) = \frac{s_g(x)}{t_g(x)}$, with $s_g(x)$, $t_g(x) \in \mathcal{P}_{R_3,R_4}$.

1. Now $g(x)h(x) = \frac{s_g(x)s_h(x)}{t_g(x)t_h(x)}$, so by Lemma 2.2 it follows that $s_g(x)s_h(x) \in \mathcal{P}_{\langle R_1,R_3\rangle,R_2R_4}$, and $t_g(x)t_h(x) \in \mathcal{P}_{\langle R_1,R_3\rangle,R_2R_4}$. Consequently $g(x)h(x) \in \mathcal{R}_{\langle R_1,R_3\rangle,R_2R_4}$.
2. Observe that $g(x) + h(x) = \frac{s_h(x)t_g(x)+s_g(x)t_h(x)}{t_g(x)t_h(x)}$. From Lemma 2.2 $s_g(x)t_h(x) + t_g(x)s_h(x) \in \mathcal{P}_{\langle R_1,R_3\rangle,R_2R_4}$, and $t_g(x)t_h(x) \in \mathcal{P}_{\langle R_1,R_3\rangle,R_2R_4}$. Hence $g(x) + h(x) \in \mathcal{R}_{\langle R_1,R_3\rangle,R_2R_4}$.
3. By considering $h'(x) = \frac{s_h'(x)t_h(x)-s_h(x)t_h'(x)}{t_h^2(x)}$, and Lemma 2.2 it is seen that $s_h'(x)t_h(x) - t_h'(x)s_h(x) \in \mathcal{P}_{R_1,\langle R_1,R_2\rangle}$ and $t_h^2(x) \in \mathcal{P}_{R_1,R_2}$. Thus $h'(x) \in \mathcal{R}_{R_1,\langle R_1,R_2\rangle}$.
4. Now $h_m^q(x) = \frac{(s_h(x)\prod_{i=1}^{m-1} t_h(x\omega_m^i))_m^q}{(\prod_{i=0}^{m-1} t_h(x\omega_m^i))_m^0}$ (Lemma 3.1). From Lemma 2.2 the numerator and the denominator are both in $\mathcal{P}_{R_1\langle\omega_m\rangle,R_2\langle\omega_m\rangle}$. This gives $h_m^q(x) \in \mathcal{R}_{R_1\langle\omega_m\rangle,R_2\langle\omega_m\rangle}$.

■

Lemma 3.3 *Let R_1, R_2, R_3, and R_4 be subrings of $\mathbb{C}$ and let $h(x) \in \mathcal{R}^{R_1,R_2}$ and $g(x) \in \mathcal{R}^{R_3,R_4}$ then:*

1. $g(x)h(x) \in \mathcal{R}^{\langle R_1,R_3\rangle,R_2R_4}$,
2. $g(x) + h(x) \in \mathcal{R}^{\langle R_1,R_3\rangle,R_2R_4}$,
3. $h'(x) \in \mathcal{R}^{R_1,R_2}$,
4. $h_m^q(x) \in \mathcal{R}^{R_1\langle\omega_m\rangle,R_2}$.

Proof: For convenience, write $h(x) = \frac{s_h(x)}{t_h(x)}$, with $s_h(x)$, $t_h(x) \in \mathcal{P}^{R_1,R_2}$, and $g(x) = \frac{s_g(x)}{t_g(x)}$, with $s_g(x)$, $t_g(x) \in \mathcal{P}^{R_3,R_4}$.

1. As $g(x)h(x) = \frac{s_g(x)s_h(x)}{t_g(x)t_h(x)}$ and Lemma 2.3 it follows that $s_g(x)s_h(x) \in \mathcal{P}^{\langle R_1,R_3\rangle,R_2R_4}$, and $t_g(x)t_h(x) \in \mathcal{P}^{\langle R_1,R_3\rangle,R_2R_4}$. Consequently $g(x)h(x) \in \mathcal{R}^{\langle R_1,R_3\rangle,R_2R_4}$.
2. Observing that $g(x)+h(x) = \frac{s_h(x)t_g(x)+s_g(x)t_h(x)}{t_g(x)t_h(x)}$, and appealing to Lemma 2.3 gives $s_g(x)t_h(x)$ $+ t_g(x)s_h(x) \in \mathcal{P}^{\langle R_1,R_3\rangle,R_2R_4}$ and $t_g(x)t_h(x) \in \mathcal{P}^{\langle R_1,R_3\rangle,R_2R_4}$. Hence $h(x)+g(x) \in \mathcal{R}^{\langle R_1,R_3\rangle,R_2R_4}$.
3. Now $h'(x) = \frac{s_h'(x)t_h(x)-s_h(x)t_h'(x)}{t_h^2(x)}$. So from Lemma 2.3 it follows that $s_h'(x)t_h(x) - t_h'(x)s_h(x) \in \mathcal{P}^{R_1,R_2}$ and $t_h^2(x) \in \mathcal{P}^{R_1,R_2}$. Thus $h'(x) \in \mathcal{R}^{R_1,R_2}$.
4. As a result of $h_m^q(x) = \frac{(s_h(x)\prod_{i=1}^{m-1} t_h(x\omega_m^i))_m^q}{(\prod_{i=0}^{m-1} t_h(x\omega_m^i))_m^0}$ (Lemma 3.1), and Lemma 2.3 it follows that both the numerator and the denominator are in $\mathcal{P}^{\langle\omega_m\rangle R_1,\langle\omega_m\rangle R_2}$. A tighter bound on the denominator $\prod_{i=0}^{m-1} t_h(x\omega_m^i)$ and numerator $(s_h(x)\prod_{i=1}^{m-1} t_h(x\omega_m^i))_m^q$, by noticing that they are fixed by automorphism of the number field $\langle\omega\rangle$ and hence are in $\mathcal{P}^{\langle\omega_m\rangle R_1,R_2}$.

■

Corollary 6 *Let R_1 and R_2 be subrings of $\mathbb{C}$. Then $\mathcal{R}^{R_1,R_2}$ and $\mathcal{R}_{R_1,R_2}$ are both fields, more over $\mathcal{R}^{R_1,R_2}$ is closed under differentiation.*

Corollary 7 *Let R_1 and R_2 be subrings of $\mathbb{C}$. Then $\hat{\mathcal{R}}^{R_1,R_2}$ and $\hat{\mathcal{R}}_{R_1,R_2}$ are both rings, more over $\hat{\mathcal{R}}^{R_1,R_2}$ is closed under differentiation.*

Now examine some closure properties of the recurrence polynomial.

Lemma 3.4 *Assume that $h(x)$, $g(x) \in \mathcal{R}$, where $h(x) = \frac{s_h(x)}{t_h(x)}$ and $g(x) = \frac{s_g(x)}{t_g(x)}$ with $s_h(x)$, $t_h(x)$, $s_g(x)$, $t_g(x) \in \mathcal{P}$. Let R_1, R_2, R_3 and R_4 be subrings of $\mathbb{C}$ and assume that $P^{s_h}(x) \in R_1[x]$, $P^{t_h}(x) \in R_2[x]$, $P^{s_g}(x) \in R_3[x]$ and $P^{t_g}(x) \in R_4[x]$.*

1. *Then* $g(x)h(x) = \frac{s_{gh}(x)}{t_{gh}(x)}$, *where* $P^{s_{gh}}(x) \in \langle R_1, R_3\rangle[x]$ *and* $P^{t_{gh}}(x) \in \langle R_2, R_4\rangle[x]$.

2. *Then* $g(x) + h(x) = \frac{s_{g+h}(x)}{t_{g+h}(x)}$, *where* $P^{s_{g+h}}(x) \in \langle R_1, R_2, R_3, R_4\rangle[x]$ *and* $P^{t_{g+h}}(x) \in \langle R_2, R_4\rangle[x]$.

3. *Then* $h'(x) = \frac{s_{h'}(x)}{t_{h'}(x)}$, *where* $P^{s_{h'}}(x) \in \langle R_1, R_2\rangle[x]$ *and* $P^{t_{h'}}(x) \in R_2[x]$.

4. *Then* $h_m^q(x) = \frac{s_{h_m^q}(x)}{t_{h_m^q}(x)}$, *where* $P^{s_{h_m^q}}(x) \in \langle R_1, R_2\rangle[x]$ *and* $P^{t_{h_m^q}}(x) \in R_2[x]$.

Proof:

1. By letting $s_{gh}(x) = s_g(x)s_h(x)$ and $t_{gh}(x) = t_g(x)t_h(x)$ the result follows from Corollary 2.

2. By letting $s_{g+h}(x) = s_g(x)t_h(x) + s_h(x)t_g(x)$ and $t_{g+h}(x) = t_g(x)t_h(x)$ the result follows from Corollary 2.

3. By letting $s_{h'}(x) = s_h'(x)t_h(x) - s_h(x)t_h'(x)$ and $t_{h'}(x) = t_h^2(x)$ the result follows from Corollary 2.

4. By letting $s_{h_m^q}(x) = (s_h(x)\prod_{i=1}^{m-1} t_h(x\omega_m^i))_m^q$ and $t_{h_m^q}(x) = (\prod_{i=0}^{m-1} t_h(x\omega_m^i))_m^0$ the result follows from Corollary 2.

■

These results are useful, as they allow the assumption to be made that certain calculations will always be over nice rings, (for example, the lacunary recurrence relation for the Euler numbers will be over the integers).

3.5 Hierarchy of $\mathcal{R}$.

As with $\mathcal{P}$, there is an interrelationship between the different subfields and subrings of $\mathcal{R}$, and a hierarchy of the different subfields.

Theorem 3.2 (Hierarchy.) *If* R_1 *and* R_2 *are subrings of* $\mathbb{C}$ *then the following subset relationships hold:*

1. $\hat{\mathcal{R}}_{R_1,R_2} \subsetneq \mathcal{R}_{R_1,R_2} \subseteq \mathcal{R}^{R_1,R_1R_2}$,

2. $\hat{\mathcal{R}}^{R_1,R_2} \subsetneq \mathcal{R}^{R_1,R_2} \subseteq \mathcal{R}_{R_1,R_1R_2}$.

Proof:

1. If $f(x) \in \mathcal{R}_{R_1,R_2}$, then $f(x) = \frac{s_f(x)}{t_f(x)}$, where $s_f(x),\ t_f(x) \in \mathcal{P}_{R_1,R_2}$, then $s_f(x),\ t_f(x) \in \mathcal{P}^{R_1,\langle R_1,R_1R_2\rangle}$. Take any non-zero element of R_2, say β, and notice that $\beta s_f(x),\ \beta t_f(x) \in \mathcal{P}^{R_1,R_1R_2}$, thus $f(x) = \frac{\beta s_f(x)}{\beta t_f(x)} \in \mathcal{R}^{R_1,R_1R_2}$ as required.

   Noticing that $\hat{\mathcal{R}}_{R_1,R_2} \subsetneq \mathcal{R}_{R_1,R_2}$ follows from noticing that $\hat{\mathcal{R}}_{R_1,R_2}$ is not closed under division.

2. If $f(x) \in \mathcal{R}^{R_1,R_2}$, where $f(x) = \frac{s_f(x)}{t_f(x)}$, with $s_f(x),\ t_s(x) \in \mathcal{P}^{R_1,R_2}$, then $s_f(x),\ t_f(x) \in \mathcal{P}_{R_1,R_2\langle R_1,R_1^{-1}\rangle}$ by Theorem 2.2. Say $s_f(x) = \sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$, and $t_f(x) = \sum_{j=1}^{m} q_j(x)e^{\mu_j x}$, with $p_i(x), q_i(x) \in R_2\langle R_1, R_1^{-1}\rangle$. For each coefficient of $p_i(x)$ and $q_i(x)$, multiply the coefficient by some $\alpha_i \in R_1$ (dependent on $p_i(x)$) so that the resulting coefficients are in R_1R_2. Now taking the least common multiple of all these α_i, gives some $\beta \in R_1$ such that $\beta p_i(x), \beta q_i(x) \in R_1R_2[x]$ for all i. Then write this as $f(x) = \frac{s_f(x)}{t_f(x)} = \frac{\beta s_f(x)}{\beta t_f(x)}$, where $\beta s_f(x), \beta t_f(x) \in \mathcal{P}_{R_1,R_1R_2}$. Hence $f(x) \in \mathcal{R}_{R_1,R_1R_2}$.

   Noticing that $\hat{\mathcal{R}}^{R_1,R_2} \subsetneq \mathcal{R}^{R_1,R_2}$ follows from noticing that $\hat{\mathcal{R}}^{R_1,R_2}$ is not closed under inversion.

■

Corollary 8 *Let R_1 and R_2 be subrings of $\mathbb{C}$. If $1 \in R_1 \subseteq R_2$ then $\mathcal{R}^{R_1,R_2} = \mathcal{R}_{R_1,R_2}$.*

The next two examples show that the set of rings $\mathcal{R}^{R_1,R_2}$ and that of $\mathcal{R}_{R_1,R_2}$ share neither a superset nor a subset relationship with each other. These examples are such that $\mathcal{R}^{R_1,R_2}$ for particular R_1 and R_2 that cannot be written as $\mathcal{R}_{R_3,R_4}$ for any R_3 and R_4 and vice-versa.

Example 14 *Let $f(x) = e^{\sqrt{2}x} \in \mathcal{R}_{\mathbb{Q}(\sqrt{2}),\mathbb{Q}}$. Notice that $f'(x) = \sqrt{2}e^{\sqrt{2}x} \notin \mathcal{R}_{\mathbb{Q}(\sqrt{2}),\mathbb{Q}}$. But $\mathcal{R}^{R_1,R_2}$ is closed under differentiation. Consequently there do not exist rings R_1, R_2 such that $\mathcal{R}_{\mathbb{Q}(\sqrt{2}),\mathbb{Q}} = \mathcal{R}^{R_1,R_2}$.*

Example 15 *The goal here is to show that there do not exist subrings R_1 and R_2 of $\mathbb{C}$ such that $\mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}} = \mathcal{R}_{R_1,R_2}$. Consider $s_c(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ where b_i satisfies $b_i = 2c^2 b_{i-2}$ for $c \in \mathbb{Z}$, with $b_0, b_1 \in \mathbb{Z}$. Then $s_c(x) \in \mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}}$. Further this is equivalent to*

$$s_c(x) = \alpha_1 e^{c\sqrt{2}x} + \alpha 2e^{-c\sqrt{2}x},$$

where $\alpha_1 = \frac{b_0}{2} + \frac{b_1}{2\sqrt{2}c}$ and $\alpha_2 = \frac{b_0}{2} - \frac{b_1}{2\sqrt{2}c}$. From this conclude that if $\mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}} = \mathcal{R}_{R_1,R_2}$, then $\mathcal{R}_{\mathbb{Z}[\sqrt{2}],\mathbb{Q}[\sqrt{2}]} \subseteq \mathcal{R}_{R_1,R_2}$.

Observing that $\mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}[\sqrt{2}]} = \mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}[\sqrt{2}]} \neq \mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}}$, as $\sqrt{2} \in \mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}[\sqrt{2}]}$ and $\sqrt{2} \notin \mathcal{R}^{\mathbb{Z}[\sqrt{2}],\mathbb{Q}}$, gives that $\mathcal{R}^{\mathbb{Z}\langle\sqrt{2}\rangle,\mathbb{Q}} \neq \mathcal{R}_{\mathbb{Z}[\sqrt{2}],\mathbb{Q}[\sqrt{2}]}$ from which the desired result follows.

3.6 Some complexity bounds.

This section determines some metrics of complexity of functions in $\mathcal{R}$, as was done earlier for functions in $\mathcal{P}$ (Section 2.6). This section uses the metrics from Definition 2.7 on the numerator and denominator of functions in $\mathcal{R}$ to get the following lemmas:

Lemma 3.5 (deg^d.) *Let $h(x) = \frac{s_h(x)}{t_h(x)}$, $g(x) = \frac{s_g(x)}{t_g(x)} \in \mathcal{R}$, such that $s_h(x)$, $t_h(x)$, $s_g(x)$, $t_g(x) \in \mathcal{P}$. Then:*

1. *Then $f(x) = \frac{s_f(x)}{t_f(x)} = g(x)h(x)$, where $1 \leq deg^d(s_f(x)) \leq deg^d(s_g(x)) + deg^d(s_h(x))$ and $1 \leq deg^d(t_f(x)) \leq deg^d(t_g(x)) + deg^d(t_h(x))$.*
2. *Then $f(x) = \frac{s_f(x)}{t_f(x)} = g(x) + h(x)$, where $0 \leq deg^d(s_f(x)) \leq \max(deg^d(s_g(x)) + deg^d(t_h(x)), deg^d(s_h(x)) + deg^d(t_g(x)))$ and $0 \leq deg^d(t_f(x)) \leq deg^d(t_g(x)) + deg^d(t_h(x))$.*
3. *Then $f(x) = \frac{s_f(x)}{t_f(x)} = g'(x)$, where $deg^d(s_f(x)) \leq deg^d(s_g(x)) + deg^d(t_g(x))$ and $deg^d(t_f(x)) \leq 2deg^d(t_g(x))$.*
4. *Then $f(x) = \frac{s_f(x)}{t_f(x)} = g^q_m(x)$, where $deg^d(s_f(x)) \leq deg^d(s_g(x)) + (m-1)deg^d(t_g(x))$ and $deg^d(t_f(x)) \leq m \times deg^d(t_g(x)))$.*

Proof:

1. By letting $s_f(x) = s_g(x)s_h(x)$ and $t_f(x) := t_g(x)t_h(x)$ the upper bounds follows from Lemma 2.4. The lower bounds follow by taking $f(x) = \frac{1}{g(x)}$.
2. By letting $s_f(x) = s_g(x)t_h(x) + s_h(x)t_g(x)$ and $t_f(x) = t_g(x)t_h(x)$ the upper bounds follow from Lemma 2.4. The lower bounds follow by taking $f(x) = -g(x)$.
3. By letting $s_f(x) = s'_g(x)t_g(x) - s_g(x)t'_g(x)$ and $t_f(x) = t^2_g(x)$ the upper bounds follow from Lemma 2.4.
4. By letting $s_f(x) = (s_g(x)\prod_{i=1}^{m-1} t_g(x\omega^i_m))^q_m$ and $t_f(x) = (\prod_{i=0}^{m-1} t_g(x\omega^i_m))^0_m$ the upper bounds follow from Lemma 2.4.

■

Lemma 3.6 (deg^P.) *Let $h(x) = \frac{s_h(x)}{t_h(x)}$, $g(x) = \frac{s_g(x)}{t_g(x)} \in \mathcal{R}$, such that $s_h(x), t_h(x), s_g(x), t_g(x) \in \mathcal{P}$.*

1. *Then $f(x) = \frac{s_f(x)}{t_f(x)} = g(x)h(x)$, where $1 \leq deg^P(s_f(x)) \leq deg^P(s_g(x))deg^P(s_h(x))$ and $1 \leq deg^P(t_f(x)) \leq deg^P(t_g(x))deg^P(t_h(x))$.*

2. *Then* $f(x) = \frac{s_f(x)}{t_f(x)} = g(x) + h(x)$, *where* $0 \leq deg^P(s_f(x)) \leq deg^P(s_g(x))deg^P(t_h(x)) + deg^P(s_h(x))\ deg^P(t_g(x)))$ *and* $1 \leq deg^P(t_f(x)) \leq deg^P(t_g(x))deg^P(t_h(x))$.

3. *Then* $f(x) = \frac{s_f(x)}{t_f(x)} = g'(x)$, *where* $deg^P(s_f(x)) \leq 2deg^P(s_g(x))deg^P(t_g(x))$ *and* $deg^P(t_f(x)) \leq deg^P(t_g(x))^2$.

4. *Then* $f(x) = \frac{s_f(x)}{t_f(x)} = g^q_m(x)$, *where* $deg^P(s_f(x)) \leq m \times deg^P(s_g(x))deg^P(t_g(x))^{m-1}$ *and also that* $deg^P(t_f(x)) \leq deg^P(t_g(x))^m$.

Proof:

1. By letting $s_f(x) = s_g(x)s_h(x)$ and $t_f(x) = t_g(x)t_h(x)$ the upper bounds follows from Lemma 2.5. The lower bounds follow by taking $f(x) = \frac{1}{g(x)}$.

2. By letting $s_f(x) = s_g(x)t_h(x) + s_h(x)t_g(x)$ and $t_f(x) = t_g(x)t_h(x)$ the upper bounds follow from Lemma 2.5. The lower bounds follow by taking $f(x) = -g(x)$.

3. By letting $s_f(x) = s'_g(x)t_g(x) - s_g(x)t'_g(x)$ and $t_f(x) = t^2_g(x)$ the upper bounds follow from Lemma 2.5.

4. By letting $s_f(x) = (s_g(x)\prod_{i=1}^{m-1} t_g(x\omega^i_m))^q_m$ and $t_f(x) = (\prod_{i=0}^{m-1} t_g(x\omega^i_m))^0_m$ the upper bounds follow from Lemma 2.5.

■

Note 3.2 *It is worth noting that the metrics under the operations of* $f(x) \to f(\alpha x)$ *was not examined as nothing interesting happens, and integration of functions in* $\mathcal{R}$ *was not examined as* $\mathcal{R}$ *is not closed under integration.*

These bounds will be used later in Chapter 5, as many methods to determine lacunary recurrence relations require bounds on the , size of these lacunary recurrence relations and also bounds on the multiplicity of the roots associated with their recurrence polynomials.

3.7 Examples.

This section does three detailed examples. That of $f(x) = \frac{1}{p(x)} \in \hat{\mathcal{R}}$ with $p(x)$ a polynomial, of $g(x) = \frac{1}{\sum_{i=1}^n \alpha_i e^{\lambda_i x}} \in \hat{\mathcal{R}}$, and lastly the Bernoulli polynomials.

Example 16 *Consider* $f(x) = \frac{1}{p(x)} \in \hat{\mathcal{R}}$. *Let* $p(x) = \alpha_n x^n + ... + \alpha_0$. *As* $f(x) \in \hat{\mathcal{R}}$, *notice that* $a_0 \neq 0$.

Then:

$$\begin{aligned}\sum_{k=0}^{\infty} c_k \frac{x^k}{k!} &= \frac{1}{\alpha_n x^n + ... + \alpha_0} \\ \sum_{i=0}^{n} \alpha_i i! \frac{x^i}{i!} \sum_{k=0}^{\infty} c_k \frac{x^k}{k!} &= 1 \\ \sum_{k=0}^{\infty} \sum_{i=0}^{n} \binom{k}{i} c_{k-i} \alpha_i i! \frac{x^k}{k!} &= 1.\end{aligned}$$

Considering $k = 0$ gives $c_0 = \frac{1}{\alpha_0}$, and considering $k > 0$ demonstrates that:

$$\begin{aligned}\sum_{i=0}^{n} \binom{k}{i} c_{k-i} \alpha_i i! &= 0 \\ c_k = \frac{-1}{\alpha_0} \sum_{i=1}^{n} \binom{k}{i} c_{k-i} \alpha_i i! &= 0.\end{aligned}$$

So a recursion formula for c_k was derived that only requires the previous $n-1$ terms.

Example 17 *Consider $g(x) \in \hat{\mathcal{R}}$ where $g(x) = \frac{1}{\sum_{i=1}^{n} \alpha_i e^{\lambda_i x}}$. A simple calculation gives $s(x) = \frac{1}{\sum_{i=0}^{\infty} b_i \frac{x^i}{i!}}$, where the $b_j = \sum_{i=1}^{n} \alpha_i \lambda_i^j$. This example will use this knowledge throughout.*

Hence:

$$\begin{aligned}\sum_{k=0}^{\infty} c_k \frac{x^k}{k!} &= \frac{1}{\sum_{j=0}^{\infty} \sum_{i=1}^{n} \alpha_i \lambda_i^j \frac{x^j}{j!}} \\ \sum_{j=0}^{\infty} \sum_{i=1}^{n} \alpha_i \lambda_i^j \frac{x^j}{j!} \sum_{k=0}^{\infty} c_k \frac{x^k}{k!} &= 1 \\ \sum_{j=0}^{\infty} \sum_{k=0} j \binom{j}{k} c_k \sum_{i=1}^{n} \alpha_i \lambda_i^{j-k} \frac{x^j}{j!} &= 1.\end{aligned}$$

Considering $k = 0$ shows that $c_0 = \frac{1}{\sum_{i=1}^{n} a_i}$. As $g(x) \in \hat{\mathcal{R}}$ it follows that $c_0 \neq 0$. Considering $k > 0$ gives:

$$c_k = \frac{1}{\sum_{i=1}^{n} a_i} (- \sum_{j=1}^{k} \binom{k}{i} \sum_{i=1}^{n} \alpha_i \lambda_i^j c_{m-j}).$$

Example 18 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

This example will demonstrate how the methods of multisectioning can be applied to functions with symbolic parameters for parameters of the exponentials of rational poly-exponential functions. Define

*the "*Bernoulli polynomials*" to be the coefficients of the exponential generating function of* $\frac{x\,e^{(t\,x)}}{e^x-1}$ *in* x*. The denominator and numerator of this function have very complicated lacunary recurrence relations, even when multisectioning by a small value such as 3 (at 0).*

```
> \mapleinline{active}{1d}{top := x* exp(t*x):}{%
> }
```

```
> \mapleinline{active}{1d}{bot := exp(x)-1:}{%
> }
```

```
> \mapleinline{active}{1d}{botlrr := `bottom/ms`(bot, f, x, 3);}{%
> }
```

$$botlrr := \mathrm{f}(x) = \mathrm{f}(x-6),\ f,\ x,\ [\mathrm{f}(0)=0,\ \mathrm{f}(1)=0,\ \mathrm{f}(2)=0,\ \mathrm{f}(3)=6,\ \mathrm{f}(4)=0,\ \mathrm{f}(5)=0]$$

```
> \mapleinline{active}{1d}{toplrr :=
> collect([`top/ms/linalg/sym`(top,bot, f, x, 3, 0)],f);}{%
> }
```

$$\begin{aligned}
toplrr := [\mathrm{f}(x) = {} & (-7152\,t^{14} - 7152\,t^{16} + 1932\,t^{11} - 3599\,t^{18} - 840\,t^{20} - t^6 + 5544\,t^{17} \\
& + 7780\,t^{15} + 286\,t^{21} + 5544\,t^{13} + 12\,t^7 - 72\,t^{22} - t^{24} - 72\,t^8 + 1932\,t^{19} \\
& + 12\,t^{23} - 840\,t^{10} + 286\,t^9 - 3599\,t^{12})\mathrm{f}(x-24) + 2(4\,t^{18} - 42\,t^{17} + 216\,t^{16} \\
& - 722\,t^{15} + 1764\,t^{14} - 3366\,t^{13} + 5244\,t^{12} - 6894\,t^{11} + 7836\,t^{10} - 7813\,t^9 \\
& + 6852\,t^8 - 5238\,t^7 + 3427\,t^6 - 1872\,t^5 + 828\,t^4 - 285\,t^3 + 72\,t^2 - 12\,t + 1) \\
& t^3\,\mathrm{f}(x-21) + (-28\,t^{18} + 252\,t^{17} - 1080\,t^{16} + 2928\,t^{15} - 5688\,t^{14} + 8568\,t^{13} \\
& - 10578\,t^{12} + 11052\,t^{11} - 9960\,t^{10} + 7978\,t^9 - 5976\,t^8 + 4320\,t^7 - 2910\,t^6 \\
& + 1692\,t^5 - 792\,t^4 + 282\,t^3 - 72\,t^2 + 12\,t - 1)\mathrm{f}(x-18) + (56\,t^{15} - 420\,t^{14} \\
& + 1440\,t^{13} - 2990\,t^{12} + 4272\,t^{11} - 4620\,t^{10} + 4066\,t^9 - 2952\,t^8 + 1536\,t^7 \\
& - 202\,t^6 - 552\,t^5 + 612\,t^4 - 346\,t^3 + 120\,t^2 - 24\,t + 2)\mathrm{f}(x-15) + (-70\,t^{12} \\
& + 420\,t^{11} - 1080\,t^{10} + 1550\,t^9 - 1368\,t^8 + 792\,t^7 - 354\,t^6 + 180\,t^5 - 120\,t^4 \\
& + 74\,t^3 - 24\,t^2 + 1)\mathrm{f}(x-12) + \\
& (56\,t^9 - 252\,t^8 + 432\,t^7 - 336\,t^6 + 72\,t^5 + 72\,t^4 - 24\,t^3 - 36\,t^2 + 24\,t - 4) \\
& \mathrm{f}(x-9) + (-28\,t^6 + 84\,t^5 - 72\,t^4 + 4\,t^3 + 24\,t^2 - 12\,t + 1)\,\mathrm{f}(x-6) \\
& + (8\,t^3 - 12\,t^2 + 2)\,\mathrm{f}(x-3),\ f,\ x, [\mathrm{f}(0)=0,\ \mathrm{f}(1)=0,\ \mathrm{f}(2)=0,\ \mathrm{f}(3)=6,\ \mathrm{f}(4)=0, \\
& \mathrm{f}(5)=0,\ \mathrm{f}(6)=60\,t + 120\,t^3 - 180\,t^2,\ \mathrm{f}(7)=0,\ \mathrm{f}(8)=0, \\
& \mathrm{f}(9)=18 - 252\,t^2 + 1260\,t^4 + 504\,t^6 - 1512\,t^5,\ \mathrm{f}(10)=0,\ \mathrm{f}(11)=0, \\
& \mathrm{f}(12)=264\,t + 3960\,t^3 - 1980\,t^2 + 7920\,t^7 - 5940\,t^8 - 5544\,t^5 + 1320\,t^9, \\
& \mathrm{f}(13)=0,\ \mathrm{f}(14)=0, \mathrm{f}(15)=30 - 1365\,t^2 + 30030\,t^4 - 16380\,t^{11} + 90090\,t^6 \\
& - 45045\,t^8 + 30030\,t^{10} - 90090\,t^5 + 2730\,t^{12},\ \mathrm{f}(16)=0,\ \mathrm{f}(17)=0, \mathrm{f}(18)= \\
& 612\,t - 36720\,t^{14} + 24480\,t^3 - 7344\,t^2 - 222768\,t^{11} + 4896\,t^{15} + 85680\,t^{13} \\
& + 700128\,t^7 - 1312740\,t^8 - 111384\,t^5 + 875160\,t^9,\ \mathrm{f}(19)=0,\ \mathrm{f}(20)=0, \\
& \mathrm{f}(21)=42 - 813960\,t^{14} - 3990\,t^2 + 203490\,t^4 + 203490\,t^{16} - 10581480\,t^{11}
\end{aligned}$$

$$+ 7980\,t^{18} + 1627920\,t^{6} - 71820\,t^{17} - 2645370\,t^{8} + 7759752\,t^{10}$$
$$- 976752\,t^{5} + 5290740\,t^{12}, \mathrm{f}(22) = 0, \mathrm{f}(23) = 0]]$$

Now, if $t = 0$ this will reduce to the situation of looking at the normal Bernoulli numbers.

```
> \mapleinline{active}{1d}{subs(t=0,[toplrr]);}{%
> }
```

$$[[\mathrm{f}(x) = -\mathrm{f}(x-18) + 2\,\mathrm{f}(x-15) + \mathrm{f}(x-12) - 4\,\mathrm{f}(x-9) + \mathrm{f}(x-6) + 2\,\mathrm{f}(x-3),\ f,\ x, [$$
$$\mathrm{f}(0) = 0, \mathrm{f}(1) = 0, \mathrm{f}(2) = 0, \mathrm{f}(3) = 6, \mathrm{f}(4) = 0, \mathrm{f}(5) = 0, \mathrm{f}(6) = 0, \mathrm{f}(7) = 0, \mathrm{f}(8) = 0,$$
$$\mathrm{f}(9) = 18, \mathrm{f}(10) = 0, \mathrm{f}(11) = 0, \mathrm{f}(12) = 0, \mathrm{f}(13) = 0, \mathrm{f}(14) = 0, \mathrm{f}(15) = 30,$$
$$\mathrm{f}(16) = 0, \mathrm{f}(17) = 0, \mathrm{f}(18) = 0, \mathrm{f}(19) = 0, \mathrm{f}(20) = 0, \mathrm{f}(21) = 42, \mathrm{f}(22) = 0,$$
$$\mathrm{f}(23) = 0]]]$$

This example is interesting because it demonstrates how large and complicated the results get when done symbolically, but still shows that feasibility of doing these calculations.

3.8 Conclusion.

By combining Theorem 3.1, Lemmas 3.3, 3.4 and 3.6 the follow results follow: Although some corollaries of this result are know, (for examples, for the particular cases of the Bernoulli, Euler, Genocchi, or Lucas type II numbers), to the best of my knowledge, they have not been done to this degree of generality before

Theorem 3.3 *Let $f(x) \in \hat{\mathcal{R}}$, $m, q \in \mathbb{Z}$, $0 \leq q < m$.*

1. *Then a lacunary recursion formula can be found for the $mi+q$-th coefficient of the exponential generating function of $f(x)$ that depends only on the $mj+q$-th coefficient, for $j = 0, 1, ..., i-1$, and two lacunary recurrence relations.*

2. *Moreover, if $f(x) = \frac{s(x)}{t(x)}$ then upper bounds on the length of the two lacunary recurrence relations are $m \times deg^P(s(x))deg^P(t(x))^{m-1}$ for the numerator and $deg^P(t(x))^m$ for the denominator.*

3. *Furthermore if $f(x) \in \hat{\mathcal{R}}^{R_1,R_2}$, then the two lacunary recurrence relations are both in $\mathcal{P}^{R_1\langle\omega_m^i\rangle,R_2}$.*

4. *Lastly, if the recurrence polynomials of $s(x)$ and $t(x)$ are in $R_3[x]$, then the recurrence polynomials of the two lacunary recurrence relations are in $R_3[x]$.*

Corollary 9 *A lacunary recursion formula can be found for the $(mi+q)$-th Bernoulli number that depends only on the $(mj+q)$-th Bernoulli number, for $j = 0, 1, ..., i-1i$, and two lacunary recurrence relations, with upper bounds on their sizes of $m2^m$ and 2^m respectively, where all the terms of the lacunary recurrence relations and the recurrence polynomials themselves are in $\mathbb{Z}$.*

Note 3.3 *Tighter upper bounds for the sizes of the lacunary recurrence relations were determined by Chellali [9] for the Bernoulli numbers. This was*

$$\sum_{d|m,\mathrm{odd}} \mu(d)2^{m/d}/2m$$

for the lacunary recurrence relation that is derived from the denominators and twice this for that of the numerator, when multisectioning by m. Here μ is the Mobius function, as defined in [2]. This result requires specialized techniques and does not follow directly follow from any of the results in this thesis.

Chapter 4

Calculations of recurrences for $\mathcal{P}$.

In the previous chapters a very naive approach was used to calculate the lacunary recurrence relations that would be needed for the calculation of the coefficients to the exponential generating functions of the functions in both $\mathcal{P}$ and $\mathcal{R}$. The function's representation as polynomials and exponential functions, was naively multisectioned using the formula in Definition 2.6. After this, the multisectioned function was converted to a formula where the lacunary recurrence relation could be observed. The goal of the next two chapters is to show some other, more efficient ways, by which these lacunary recurrence relations and lacunary recursion formulae can be computed.

In this chapter, different methods to multisection functions in $\mathcal{P}$ are examined, and Chapter 5 examines different methods for those functions in $\mathcal{R}$.

Section 4.1 looks at how to use recurrence polynomials to multisection poly-exponential functions. This method takes advantage of the factorization of m, the quantity by which the poly-exponential function is multisectioned. Section 4.2 looks at how to use recurrence polynomials and resultants to multisection poly-exponential functions. Using linear algebra to find the new lacunary recurrence relations of a poly-exponential functions that are multisectioned, as well as how to use symbolic differentiation with linear algebra is looked at in Section 4.3 and 4.4. Section 4.5 looks at how to take advantage of the factorization of m, by iteratively compressing the results. Section 4.6 and 4.7 looks at two theories where by the problem being studied can be simplified. The last section, Section 4.8, makes some conclusions based on empirical evidence as to which methods are best.

4.1 Multisectioning the recurrence polynomial.

Recall that if $s(x) \in \mathcal{P}$ then $P^s(x)$ is the recurrence polynomial associated with $s(x)$ (Definition 2.2). The first thing needed was shown in Corollary 2 and Lemma 2.5 which is reiterated here:

Lemma 4.1 *If $s(x)$, $t(x) \in \mathcal{P}$, $\alpha \neq 0$ where $s(x) = \sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$ and where $t(x) = \sum_{j=1}^{m} q_j(x)e^{\mu_j x}$ then:*

1. $P^{st}(x) | \prod_{i=1,j=1}^{i=n,j=m} (x - \lambda_i - \mu_j)^{\deg(p_i(x))+\deg(q_i(x))}$,
2. $P^{s+t}(x) | P^s(x)P^t(x)$,
3. $P^{s(\alpha x)}(x) = P^s(\alpha x)$,
4. $P^{\alpha s}(x) = P^s(x)$.

By using this information, the linear recurrence relation for a poly-exponential function may be multisectioned by only looking at the recurrence polynomial.

Lemma 4.2 *If $s(x) \in \mathcal{P}$ then*

$$P^{s^q_m(x)}(x) | \prod_{i=0}^{m-1} P^s(x\omega_m^i).$$

Proof: By noticing that $P^{s+t}(x) | P^s(x)P^t(x)$, and $P^{s(\alpha x)}(x) = P^s(\alpha x)$ from Lemma 4.1, it follows that:

$$P^{s^q_m(x)}(x) \quad = \quad P^{\frac{1}{m}\sum_{i=0}^{m-1} \omega_m^{-qi} s(x\omega_m^i)}(x) | \prod_{i=0}^{m-1} P^{s(x\omega_m^i)}(x) = \prod_{i=0}^{m-1} P^s(x\omega_m^i).$$

■

By recalling that any polynomial which the recurrence polynomial divides is a valid recurrence polynomial (Section 2.3), the above product $\prod_{i=0}^{m-1} P^s(x\omega_m^i)$ will give a valid lacunary recurrence relation for $s^q_m(x)$. Further it is fairly easy to do this computationally. With the additional information of $deg^d(s(x))$, an even better recurrence polynomial can be found, as $deg^d(s^q_m(x)) = deg^d(s(x))$ (Lemma 2.4). Hence this shows that the recurrence polynomial can have no roots of multiplicity greater than $deg^d(s(x)) + 1$ (Corollary 1).

From a computational point of view, the order in which the $P^s(x\omega_m^i)$ for $0 \leq i \leq m-1$ are multiplied together is important. For example if $m = 2^k$ and $P^s(x) \in \mathbb{Z}[x]$ then: $P^s(x), P^s(-x) \in \mathbb{Z}[x]$, and further that $P^s(x)P^s(-x) \in \mathbb{Z}[x^2]$. It follows that $P^s(ix)P^s(-ix) \in \mathbb{Z}[x^2]$ and hence $P^s(x)P^s(-x)P^s(ix)P^s(-ix) \in \mathbb{Z}[x^4]$. Etc.

In general, if $m = d_1 d_2 ... d_k$, for $d_i \in \mathbb{Z}$ where $2 \leq d_i$, then this computation is best done as:

$$\prod_{i_k=0}^{d_k-1} ... \prod_{i_2=0}^{d_2-1} \prod_{i_1=0}^{d_1-1} P^s(x\omega_{d_1}^{i_1}\omega_{d_1 d_2}^{i_2} ... \omega_{d_1 d_2 .. d_k}^{i_k}),$$

performing the computation at the inner levels first, and using scaling to perform the next level out.

As a result of implementing this, a bug in Maple was found, which made the original method to scaling very inefficient. See Appendix D Section D.1 for more information about this.

Example 19 *Consider the following example in Maple. For more information about the Maple code, see Appendix A. For the Maple code see Appendix E. The Maple code and help files (including information about syntax) are available on the web at [1].*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the exponential generating function $\mathrm{s}(x) = \sum_{i=0}^{\infty} \frac{b_i x^i}{i!}$ *with a linear recurrence relation* $b_i = b_{i-1} - b_{i-2} + b_{i-3}$*, with initial values of* $b_0 = 1$, $b_1 = 1$ *and* $b_2 = 1$*. This example multisections this linear recurrence relation by 16 at 0, using the methods described in this section. First determine the value of* $deg^d(s(x))$.

```
> \mapleinline{active}{1d}{s := b(x) = b(x-1)-b(x-2)+b(x-3), b,
> x, [b(0) = 1, b(1) = 1, b(2) = 1];}{%
> }
```

$$\mathrm{s} := \mathrm{b}(x) = \mathrm{b}(x-1) - \mathrm{b}(x-2) + \mathrm{b}(x-3),\ b,\ x,\ [\mathrm{b}(0) = 1,\ \mathrm{b}(1) = 1,\ \mathrm{b}(2) = 1]$$

```
> \mapleinline{active}{1d}{`egf/metric/d`(s);}{%
> }
```

$$0$$

From this it follows that the multisectioned recurrence polynomial can have no multiple roots.

So now determine the recurrence polynomial.

```
> \mapleinline{active}{1d}{P := convert_poly(s);}{%
> }
```

$$P := x^3 - x^2 + x - 1$$

Now multiply $\mathrm{P}(x)$ *by* $\mathrm{P}(-x)$ *and expand.*

```
> \mapleinline{active}{1d}{P2 := expand(subs(x=-x,P)*P);}{%
> }
```

$$P2 := -x^6 - x^4 + x^2 + 1$$

Now this polynomial should have no multiple roots, so get rid of the multiple roots.

```
> \mapleinline{active}{1d}{P2p := quo(P2,gcd(P2, diff(P2,x)),x);}{%
> }
```

$$P2p := -x^4 + 1$$

Now multiply $\mathrm{P2p}(x)$ *by* $\mathrm{P2p}(x\,I)$*, and expand. This gives a recurrence polynomial that divides* $\mathrm{P}(x)\,\mathrm{P}(-x)\,\mathrm{P}(I\,x)\,\mathrm{P}(-I\,x)$ *and has no multiple roots.*

```
> \mapleinline{active}{1d}{P4 := expand(subs(x=x*I,P2p)*P2p);}{%
> }
```

$$P4 := x^8 - 2\,x^4 + 1$$

Again, get rid of the multiple roots.

```
> \mapleinline{active}{1d}{P4p := quo(P4, gcd(P4, diff(P4,x)),x);}{%
> }
```

$$P4p := x^4 - 1$$

Lastly, multiply $\mathrm{P4p}(x)$ *by* $\mathrm{P4p}(x\,\sqrt{I})$ *and expand. This gives a recurrence polynomial that divides* $\mathrm{P}(x)\,\mathrm{P}(-x)\,\mathrm{P}(I\,x)\,\mathrm{P}(-I\,x)\,\mathrm{P}(\sqrt{I}\,x)\,\mathrm{P}(-\sqrt{I}\,x)\,\mathrm{P}(I\,\sqrt{I}\,x)\,\mathrm{P}(-I\,\sqrt{I}\,x)$ *and has no multiple roots.*

```
> \mapleinline{active}{1d}{P8 := expand(subs(x=x*sqrt(I),P4p)*P4p);}{%
> }
```

$$P8 := -x^8 + 1$$

Again, get rid of the multiple roots.

```
> \mapleinline{active}{1d}{P8p := quo(P8, gcd(P8, diff(P8,x)),x);}{%
> }
```

$$P8p := -x^8 + 1$$

So converting back gives a linear recurrence relation of:

```
> \mapleinline{active}{1d}{convert_rec(P8p,b,x);}{%
> }
```

$$\mathrm{b}(x) = \mathrm{b}(x-8)$$

This is the same linear recurrence relation that is derived using the naive technique discussed in Example 13.

```
> \mapleinline{active}{1d}{`egf/ms/naive`(s,8,0);}{%
> }
```

$$\mathrm{b}(x) = \mathrm{b}(x-8),\ b,\ x,$$
$$[\mathrm{b}(0) = 1,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 0,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = 0,\ \mathrm{b}(5) = 0,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = 0]$$

This has been automated as the Maple command 'egf/ms/rec'.

```
> \mapleinline{active}{1d}{`egf/ms/rec`(s,8,0);}{%
> }
```

$$\mathrm{b}(x) = \mathrm{b}(x-8),\ b,\ x,$$
$$[\mathrm{b}(0) = 1,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 0,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = 0,\ \mathrm{b}(5) = 0,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = 0]$$

4.2 Multisectioning via resultants.

In the previous section, the recurrence polynomials of $s(x) \in \mathcal{P}$, say $P^s(x)$, was multisectioned by computing $\prod_{i=0}^{m-1} P^s(x\omega_m^i)$ in a naive fashion, and then getting rid of root with too high of an order. This section again computes $\prod_{i=0}^{m-1} P^s(x\omega_m^i)$ but in a more sophisticated manner; by using resultants [20].

Definition 4.1 *Let* $p(x) = a\prod_{i=1}^{n}(x-\lambda_i)$ *and* $q(x) = b\prod_{j=1}^{m}(x-\mu_j)$. *The "*resultant*", denoted* $\mathrm{Res}_x(p(x), q(x))$ *is defined as:*

$$\mathrm{Res}_x(p(x), q(x)) = a^m b^n \prod_{i=1,j=1}^{i=n,j=m} (\lambda_i - \mu_j).$$

This next theorem follows from the definition of the resultant.

Theorem 4.1 *Let* $s(x) \in \mathcal{P}$, *and* $P^s(x)$ *be the recurrence polynomial for* $s(x)$ *and* $P^{s_m^q(x)}(x)$ *the recurrence polynomial for* $s_m^q(x)$. *Then:*

$$P^{s_m^q(x)}(x) | \mathrm{Res}_y(y^m - x^m, P^s(y))$$

Proof: Write $P^s(y) = \prod_{i=1}^{n}(y-\lambda_i)$. Notice that $y^m - x^m = \prod_{i=1}^{m}(y-\omega_m^i x)$. Thus from Lemma 4.2 it follows that $P^{s_m^q(x)}(x) | \prod_{j=0}^{m-1} P^s(x\omega_m^j)$. Further:

$$\prod_{j=0}^{m-1} P^s(x\omega_m^j) = \prod_{j=0}^{m-1}\prod_{i=1}^{n}(\omega_m^j x - \lambda_i) = \mathrm{Res}_y(y^m - x^m, P^s(y)).$$

Which is the desired result.

■

There are many good methods for computing resultants efficiently, in a symbolic setting. See, for example [12, 13].

Example 20 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the example of the Padovan numbers defined in [28] . Let $\mathrm{s}(x) = \sum_{i=0}^{\infty} \frac{b_i\, x^i}{i!}$*, where* $b_i = b_{i-2} + b_{i-3}$ *and* $b_0 = 1$*,* $b_1 = 0$*, and* $b_2 = 1$ *. Consider multisectioning this by 17 at 0. This example will do this by computing the resultant of* $\mathrm{P}^{\mathrm{s}}(y)$ *with* $y^{17} - x^{17}$*.*

```
> \mapleinline{active}{1d}{s := b(y) = b(y-2) + b(y-3), b, y, [b(0) =
> 1, b(1) = 0, b(2) = 1];}{%
> }
```

$$\mathrm{s} := \mathrm{b}(y) = \mathrm{b}(y-2) + \mathrm{b}(y-3),\, b,\, y,\, [\mathrm{b}(0) = 1,\, \mathrm{b}(1) = 0,\, \mathrm{b}(2) = 1]$$

```
> \mapleinline{active}{1d}{poly := convert_poly(s);}{%
> }
```

$$poly := y^3 - y - 1$$

```
> \mapleinline{active}{1d}{poly := resultant(y^17-x^17,poly,y);}{%
> }
```

$$poly := -18\, x^{17} - 1 - 119\, x^{34} + x^{51}$$

```
> \mapleinline{active}{1d}{convert_rec(poly, f, x);}{%
> }
```

$$\mathrm{f}(x) = 18\,\mathrm{f}(x-34) + \mathrm{f}(x-51) + 119\,\mathrm{f}(x-17)$$

There is a command in Maple to do this called 'egf/ms/result'.

```
> \mapleinline{active}{1d}{`egf/ms/result`(s,17,0);}{%
> }
```

$$\begin{aligned}\mathrm{b}(y) = 18\,\mathrm{b}(y-34) + \mathrm{b}(y-51) + 119\,\mathrm{b}(y-17),\, b,\, y,\, [&\mathrm{b}(0) = 1,\, \mathrm{b}(1) = 0,\, \mathrm{b}(2) = 0,\\ &\mathrm{b}(3) = 0,\, \mathrm{b}(4) = 0,\, \mathrm{b}(5) = 0,\, \mathrm{b}(6) = 0,\, \mathrm{b}(7) = 0,\, \mathrm{b}(8) = 0,\, \mathrm{b}(9) = 0,\, \mathrm{b}(10) = 0,\\ &\mathrm{b}(11) = 0,\, \mathrm{b}(12) = 0,\, \mathrm{b}(13) = 0,\, \mathrm{b}(14) = 0,\, \mathrm{b}(15) = 0,\, \mathrm{b}(16) = 0,\, \mathrm{b}(17) = 49,\\ &\mathrm{b}(18) = 0,\, \mathrm{b}(19) = 0,\, \mathrm{b}(20) = 0,\, \mathrm{b}(21) = 0,\, \mathrm{b}(22) = 0,\, \mathrm{b}(23) = 0,\, \mathrm{b}(24) = 0,\\ &\mathrm{b}(25) = 0,\, \mathrm{b}(26) = 0,\, \mathrm{b}(27) = 0,\, \mathrm{b}(28) = 0,\, \mathrm{b}(29) = 0,\, \mathrm{b}(30) = 0,\, \mathrm{b}(31) = 0,\\ &\mathrm{b}(32) = 0,\, \mathrm{b}(33) = 0,\, \mathrm{b}(34) = 5842,\, \mathrm{b}(35) = 0,\, \mathrm{b}(36) = 0,\, \mathrm{b}(37) = 0,\, \mathrm{b}(38) = 0,\\ &\mathrm{b}(39) = 0,\, \mathrm{b}(40) = 0,\, \mathrm{b}(41) = 0,\, \mathrm{b}(42) = 0,\, \mathrm{b}(43) = 0,\, \mathrm{b}(44) = 0,\, \mathrm{b}(45) = 0,\\ &\mathrm{b}(46) = 0,\, \mathrm{b}(47) = 0,\, \mathrm{b}(48) = 0,\, \mathrm{b}(49) = 0,\, \mathrm{b}(50) = 0]\end{aligned}$$

This gives the same result.

4.3 Using linear algebra on $\mathcal{P}$.

If $s(x) \in \mathcal{P}$, and an upper bound on the size of the linear recurrence relation is known, then this linear recurrence relation can be determined by the early cases.

This can be written concisely as:

Lemma 4.3 *If $s(x) \in \mathcal{P}$ and $deg^P(s(x)) \leq N$ and $deg^d(s(x)) = k$, then $P^s(x)$ can be calculated by the first $2N + k$ values.*

This result is fairly well know, and can be found in a number of difference linear algebra text books as an application of linear algebra. It is included here for completeness sake.

Proof: If b_{k+1}, b_{k+2}, ..., b_{k+2N} are the initial values of some linear recurrence relation, then this leads to the following system of N linear equations:

$$\begin{aligned} a_N b_{k+1} + a_{N-1} b_{k+2} + ... + a_1 b_{k+N} &= b_{k+N+1} \\ a_N b_{k+2} + a_{N-1} b_{k+3} + ... + a_1 b_{k+N+1} &= b_{k+N+2} \\ &\vdots \\ a_N b_{k+N} + a_{N-1} b_{k+N+1} + ... + a_1 b_{k+2N-1} &= b_{k+2N}. \end{aligned}$$

There are N linear equations, and N unknowns $(a_1, ..., a_N)$, hence a solution exists. To rewrite this in the language of linear algebra, find the values $a_1, ..., a_N$ so that they satisfy the equation:

$$\begin{bmatrix} b_{k+1} & b_{k+2} & \cdots & b_{k+N} \\ b_{k+2} & b_{k+3} & \cdots & b_{k+N+1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k+N} & b_{k+N+1} & \cdots & b_{k+2N-1} \end{bmatrix} \begin{bmatrix} a_N \\ a_{N-1} \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} b_{k+N+1} \\ b_{k+N+2} \\ \vdots \\ b_{k+2N} \end{bmatrix}.$$

■

If when solving for the a_1, ..., a_N above, a unique solution is not found, set a_N to zero, and see if that gives a unique solution. If not, set a_{N-1} to 0, and see if that gives a unique solution. Continue in this manner. In this way when a unique solution is found, it will be of the shortest possible length.

It is also worth noting that if the order of all the columns is reversed then the resulting matrix is a Toeplitz matrix (this would mean that the expected solution is also reversed). This is nice, because there is an $\mathcal{O}(n^2)$ algorithm for solving $n \times n$ Toeplitz matrix [15].

This algorithm was not implemented with the Maple package included with this thesis, as most of the problems would still finish in a reasonable amount of time with Maple's less efficient linear algebra package.

This lemma is of great use for the computation of Bernoulli numbers, as an upper bound for $\prod_{i=0}^{m-1}(e^{\omega_m^i x}-1)$ is determined in a paper by Chellali [9], as being:

$$\sum_{d|m,\text{odd}} \mu(d)2^{m/d}/2m. \tag{4.1}$$

Here μ is the Mobius function, as defined in [2]. Later in Section 5.2 of Chapter 5, it will be seen how to use this.

Example 21 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the example of the Fibonacci numbers. Let $\mathrm{s}(x)=\sum_{i=0}^{\infty}\frac{b_i\,x^i}{i!}$*, where* $b_0=0$ *and* $b_1=1$*. Consider multisectioning this by 17 at 0. From Lemma 2.5, the size of the new linear recurrence relation will be at most 17 times* $deg^P(s(x))=2$*. Further* $deg^d(s(x))=0$ *so it follows that the values* b_1*,* b_2*, ...* $b_{17\times 2\times 2}$ *are needed. All but* b_{17}*,* b_{34}*,* b_{51}*, and* b_{68} *will be zero, so only these four values are needed to determine the linear recurrence relation.*

```
> \mapleinline{active}{1d}{s := b(i) = b(i-1) + b(i-2), b, i, [b(0) =
> 0, b(1) = 1];}{%
> }
```

$$\mathrm{s} := \mathrm{b}(i) = \mathrm{b}(i-1)+\mathrm{b}(i-2),\ b,\ i,\ [\mathrm{b}(0)=0,\ \mathrm{b}(1)=1]$$

```
> \mapleinline{active}{1d}{`egf/metric/P`(s);}{%
> }
```

$$2$$

```
> \mapleinline{active}{1d}{`egf/metric/d`(s);}{%
> }
```

$$0$$

```
> \mapleinline{active}{1d}{Fib := `egf/makeproc`(s):}{%
> }
```

So this gives the following two linear equations:

```
> \mapleinline{active}{1d}{eqn1 := a[1] * Fib(17) + a[2] * Fib(34) =
> Fib(51);}{%
> }
```

$$eqn1 := 1597\,a_1 + 5702887\,a_2 = 20365011074$$

```
> \mapleinline{active}{1d}{eqn2 := a[1] * Fib(34) + a[2] * Fib(51) =
> Fib(68);}{%
> }
```

$$eqn2 := 5702887\,a_1 + 20365011074\,a_2 = 72723460248141$$

Solving these two equations gives a_1 *and* a_2.

```
> \mapleinline{active}{1d}{solve(\{eqn1, eqn2\});}{%
> }
```

$$\{a_1 = 1,\, a_2 = 3571\}$$

So this gives the linear recurrence relation $b_i = 3571\,b_{i-17} + b_{i-28}$. *This could have also been solved by using the linear algebra package in Maple in the following way.*

```
> \mapleinline{active}{1d}{C = matrix(2,2,[Fib(17), Fib(34), Fib(34),
> Fib(51)]);}{%
> }
```

$$C := \begin{bmatrix} 1597 & 5702887 \\ 5702887 & 20365011074 \end{bmatrix}$$

```
> \mapleinline{active}{1d}{B = vector(2, [Fib(51), Fib(68)]);}{%
> }
```

$$B := [20365011074,\, 72723460248141]$$

```
> \mapleinline{active}{1d}{linsolve(C,B);}{%
> }
```

$$[1,\, 3571]$$

There is also a command in Maple to do this called 'egf/ms/linalg'.

```
> \mapleinline{active}{1d}{`egf/ms/linalg`(s,17,0);}{%
> }
```

$$\mathrm{b}(i) = \mathrm{b}(i-34) + 3571\,\mathrm{b}(i-17),\, b,\, i, [\mathrm{b}(0) = 0,\, \mathrm{b}(1) = 1,\, \mathrm{b}(2) = 0,\, \mathrm{b}(3) = 0,\, \mathrm{b}(4) = 0,$$
$$\mathrm{b}(5) = 0,\, \mathrm{b}(6) = 0,\, \mathrm{b}(7) = 0,\, \mathrm{b}(8) = 0,\, \mathrm{b}(9) = 0,\, \mathrm{b}(10) = 0,\, \mathrm{b}(11) = 0,\, \mathrm{b}(12) = 0,$$
$$\mathrm{b}(13) = 0,\, \mathrm{b}(14) = 0,\, \mathrm{b}(15) = 0,\, \mathrm{b}(16) = 0,\, \mathrm{b}(17) = 0,\, \mathrm{b}(18) = 2584,\, \mathrm{b}(19) = 0,$$
$$\mathrm{b}(20) = 0,\, \mathrm{b}(21) = 0,\, \mathrm{b}(22) = 0,\, \mathrm{b}(23) = 0,\, \mathrm{b}(24) = 0,\, \mathrm{b}(25) = 0,\, \mathrm{b}(26) = 0,$$
$$\mathrm{b}(27) = 0,\, \mathrm{b}(28) = 0,\, \mathrm{b}(29) = 0,\, \mathrm{b}(30) = 0,\, \mathrm{b}(31) = 0,\, \mathrm{b}(32) = 0,\, \mathrm{b}(33) = 0]$$

So this again gives the same result.

4.4 Using symbolic differentiation with linear algebra.

Section 4.3 used knowledge about what the linear recurrence relation to determine the first $2N + k$ cases, (N and k defined as before). If $s(x)$ is function instead in poly-exponential form, then symbolic differentiation can be used to find the first $2N + k$ cases.

Example 22 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):with(linalg):}{%
> }
```

Consider the poly-exponential function $\mathrm{s}(x) = e^{(2\,x)}\,x^3 + e^{(3\,x)}$. *Notice that* $deg^P(s(x)) = 5$ *and* $deg^d(s(x)) = 3$. *Hence to multisection by 7 at 4, we need only look at the values for* b_4, b_{11}, b_{18}..., b_{74}.

```
> \mapleinline{active}{1d}{s := exp(2*x)*x^3 + exp(3*x);}{%
> }
```

$$\mathrm{s} := e^{(2\,x)}\,x^3 + e^{(3\,x)}$$

```
> \mapleinline{active}{1d}{`pe/metric/P`(s,x);}{%
> }
```

$$5$$

```
> \mapleinline{active}{1d}{`pe/metric/d`(s,x);}{%
> }
```

$$3$$

```
> \mapleinline{active}{1d}{for i from 4 to 74 by 7 do}{%
> }
> \mapleinline{active}{1d}{    b[i] := eval(diff(s,x$i),x=0);}{%
> }
> \mapleinline{active}{1d}{od;}{%
> }
```

$$b_4 := 129$$

$$b_{11} := 430587$$

$$b_{18} := 547852617$$

$$b_{25} := 905170004643$$

$$b_{32} := 1868997467192961$$

$$b_{39} := 4056323316806318091$$

$$b_{46} := 8863739267804963800569$$

$$b_{53} := 193834039196673260686556 67$$

$$b_{60} := 42391187864946619249022072241$$

$$b_{67} := 92709468450045486192098346397467$$

$$b_{74} := 20275559682282062436318697 4870842281$$

Set the matrix C equal to $\begin{bmatrix} b_{11} & b_{18} & b_{25} & b_{32} & b_{39} \\ b_{18} & b_{25} & b_{32} & b_{39} & b_{46} \\ b_{25} & b_{32} & b_{39} & b_{46} & b_{53} \\ b_{32} & b_{39} & b_{46} & b_{53} & b_{60} \\ b_{39} & b_{46} & b_{53} & b_{60} & b_{67} \end{bmatrix}$.

```
> \mapleinline{active}{1d}{C :=
> matrix(5,5,[seq(seq(b[4+7*(i+j-1)],i=1..5),j=1..5)]):}{%
> }
```

Set the vector v equal to $[b_{44}, b_{51}, b_{58}, b_{65}, b_{72}]$.

```
> \mapleinline{active}{1d}{v := vector(5, [seq(b[4+7*i+35],i=1..5)]):}{%
> }
```

Now solve.

```
> \mapleinline{active}{1d}{linsolve(C,v);}{%
> }
```

$$[587068342272, -18614321152, 223379456, -1218048, 2699]$$

This gives a linear recurrence relation of $d_i = 587068342272 d_{i-35} - 18614321152 b_{i-28} + 223379456\, b_{i-21} - 1218048 b_{i-14} + 2699 b_{i-7}$.

This could have also been done by the Maple function 'pe/ms/linalg/sym'.

```
> \mapleinline{active}{1d}{`pe/ms/linalg/sym`(s,f, x,7,2);}{%
> }
```

$$\begin{aligned} \mathrm{f}(x) = {} & 587068342272\,\mathrm{f}(x-35) - 18614321152\,\mathrm{f}(x-28) + 223379456\,\mathrm{f}(x-21) \\ & - 1218048\,\mathrm{f}(x-14) + 2699\,\mathrm{f}(x-7),\ f,\ x, [\mathrm{f}(0) = 0,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = 9,\ \mathrm{f}(3) = 0, \\ & \mathrm{f}(4) = 0,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 0,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 0,\ \mathrm{f}(9) = 51939,\ \mathrm{f}(10) = 0, \\ & \mathrm{f}(11) = 0,\ \mathrm{f}(12) = 0,\ \mathrm{f}(13) = 0,\ \mathrm{f}(14) = 0,\ \mathrm{f}(15) = 0,\ \mathrm{f}(16) = 70571841, \end{aligned}$$

f(17) = 0, f(18) = 0, f(19) = 0, f(20) = 0, f(21) = 0, f(22) = 0,
f(23) = 105285347403, f(24) = 0, f(25) = 0, f(26) = 0, f(27) = 0, f(28) = 0,
f(29) = 0, f(30) = 209160675948729, f(31) = 0, f(32) = 0, f(33) = 0,
f(34) = 0]

Which is the same result.

4.5 Using compression.

In most situations, the main interest is the lacunary recurrence relations not the poly-exponential functions themselves. Define a new operation that will maintain the useful information of a lacunary recurrence relation such that the function under this operation will have a smaller recurrence polynomial.

Definition 4.2 (C_m^q.) *Define C_m^q that acts on $\sum_{i=0}^{\infty} b_{mi+q}\frac{x^{mi+q}}{(mi+q)!}$ by $C_m^q(\sum_{i=0}^{\infty} b_{mi+q}\frac{x^{mi+q}}{(mi+q)!}) = \sum_{i=0}^{\infty} b_{im+q}\frac{x^i}{i!}$.*

The term "*compressing*" will be used to describe this process. When saying a function $s(x)$ is "*compressed by m*", $C_m^q(s(x))$ is being looked at for some q. When saying a function $s(x)$ is "*compressed by m at q*", then $C_m^q(s(x))$ is being studied.

Methods similiar to those that arrive via compressing can be found for Fibonacci or Lucas numbers [16]. To the best of my knowledge, the definition, or consequences of compressing have not been written in this way before.

Some properties of compression are enumerated below.

Lemma 4.4 *Let $s(x) \in \mathcal{P}$ and let R_1, R_2 be subrings of $\mathbb{C}$, then:*

1. *If $s_m^q(x) \in \mathcal{P}^{R_1,R_2}$ then $C_m^q(s_m^q(x)) \in \mathcal{P}^{R_1,R_2}$.*
2. *If $s_m^q(x) \in \mathcal{P}_{R_1,R_2}$ then $C_m^q(s_m^q(x)) \in \mathcal{P}_{R_1,R_2\langle R_1,R_1^{-1}\rangle}$.*
3. *If $P^{s_m^q(x)}(x) \in R_1[x]$ then $P^{C_m^q(s_m^q(x))}(x) \in R_1[x]$.*
4. *Then $deg^d(s_m^q(x)) \geq deg^d(C_m^q(s_m^q(x)))$.*
5. *Then $deg^P(s_m^q(x)) = m \times deg^P(C_m^q(s_m^q(x)))$.*

Proof:

1. If $P^{s^q_m(x)}(x) = \prod_{i=1}^{n}(x^m - \lambda_i)$ then $P^{C^q_m(s^q_m(x)}(x) = \prod_{i=1}^{n}(x - \lambda_i)$, hence the recurrence polynomial for $C^q_m(s^q_m(x))$ splits in R_1. The coefficients of the exponential generating function are still in R_2, as they haven't changed value, only positions within the exponential generating function.

2. This follows from the hierarchy theorem (Theorem 2.2) as if $s^q_m(x) \in \mathcal{P}_{R_1,R_2}$ then $s^q_m(x) \in \mathcal{P}^{R_1,\langle R_1R_2,R_2\rangle}$. Hence from part 1 of this lemma, as $(s^q_m(x)) \in \mathcal{P}^{R_1,\langle R_1R_2,R_2\rangle}$ then $C^q_m(s^q_m(x)) \in \mathcal{P}^{R_1,\langle R_1R_2,R_2\rangle}$. This again from Theorem 2.2 gives that $C^q_m(s^q_m(x)) \in \mathcal{P}_{R_1,\langle R_1R_2,R_2\rangle\langle R_1,R_1^{-1}\rangle}$ which is equal to $\mathcal{P}_{R_1,R_2\langle R_1,R_1^{-1}\rangle}$.

3. If $P^{s^q_m(x)}(x) = x^{mn} + a_{n-1}x^{m(n-1)} + ...a_0$, then $P^{C^q_m(s^q_m(x))} = x^n + a_{n-1}x^{n-1} + ...a_0$. From this coefficients of $P^{C^q_m(s^q_m(x))}$ are still in R_1.

4. The recurrence polynomial of $s^q_m(x)$ can be written as a polynomial in x^m, say $\prod_{i=1}^{n}(x^m - \lambda_i)$. After the compression, the recurrence polynomial will be written as a polynomial in x, namely $\prod_{i=1}^{n}(x - \lambda_i)$. If some λ_i has multiplicity $deg^d(C^q_m(s^q_m(x)))$ in $\prod_{i=1}^{n}(x - \lambda_i)$, then λ_i will also appear with that multiplicity in $\prod_{i=1}^{n}(x^m - \lambda_i)$. From this $deg^d(s^q_m(x)) \geq deg^d(C^q_m(s^q_m(x)))$.

5. The recurrence polynomial of $s^q_m(x)$ can be written as a polynomial in x^m say $x^{mn} + a_{n-1}x^{m(n-1)} + ... + a_0$. After the compression, it will be written as a polynomial in x, namely $x^n + a_{n-1}x^{n-1} + ...a_0$, in x^m. This is a polynomial with the same coefficients, but with $\frac{1}{m}$-th the degree. Thus $deg^P(s^q_m(x)) = m \times deg^P(C^q_m(s^q_m(x)))$.

■

Theorem 4.2 *Let $s(x) \in \mathcal{P}$, with $m = d_1...d_n$, and $q = a_1(d_2...d_n) + a_2(d_3...d_n) + ... + a_n$ where $0 \leq a_i < d_i$. Consequently:*

$$C^q_m(s^q_m(x)) = C^{a_1}_{d_1}((C^{a_2}_{d_2}((...C^{a_n}_{d_n}(s^{a_n}_{d_n}(x)))^{a_{n-1}}_{d_{n-1}}...)^{a_1}_{d_1}).$$

Proof: Show that if $m = d_1d_2$ and $q = a_2d_1 + a_1$ for $d_i \in \mathbb{Z}$ where $2 \leq d_i$, and $0 \leq a_i < d_i$ then:

$$C^q_m(s^q_m(x)) = C^{a_1}_{d_1}((C^{a_2}_{d_2}(s^{a_2}_{d_2}(x)))^{a_1}_{d_1}).$$

and then the result will follow by induction.

Assume that $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$. Then:

$$\begin{aligned}
C^{a_1}_{d_1}((C^{a_2}_{d_2}(s^{a_2}_{d_2}(x)))^{a_1}_{d_1}) &= C^{a_1}_{d_1}((C^{a_2}_{d_2}((\sum_{i=0}^{\infty} b_i \frac{x^i}{i!})^{a_2}_{d_2}))^{a_1}_{d_1}) = C^{a_1}_{d_1}((C^{a_2}_{d_2}(\sum_{i=0}^{\infty} b_{d_2i+a_2}\frac{x^{d_2i+a_2}}{(d_2i+a_1)!}))^{a_1}_{d_1}) \\
&= C^{a_1}_{d_1}((\sum_{i=0}^{\infty} b_{d_2i+a_2}\frac{x^i}{i!})^{a_1}_{d_1}) = C^{a_1}_{d_1}(\sum_{i=0}^{\infty} b_{d_1(d_2i+a_2)+a_1}\frac{x^{d_1i+a_1}}{(d_1i+a_1)!}) \\
&= \sum_{i=0}^{\infty} b_{d_1(d_2i+a_2)+a_1}\frac{x^i}{i!} = \sum_{i=0}^{\infty} b_{d_1d_2i+d_1a_2+a_1}\frac{x^i}{i!} = \sum_{i=0}^{\infty} b_{mi+q}\frac{x^i}{i!}.
\end{aligned}$$

But this is precisely $C_m^q(s_m^q(x))$, hence the result follows by induction.

■

This is of great value as $C_{d_1}^{a_1}((C_{d_2}^{a_2}((...C_{d_n}^{a_n}(s_{d_n}^{a_n}(x)))_{d_{n-1}}^{a_{n-1}}...)_{d_1}^{a_1})$ is much easier to compute than is $C_m^q(s_m^q(x))$. This method of iteratively multisectioning requires less memory and time than doing the multisectioning process all in one calculation.

To see this, first let $f(m)$ be the complexity of the underlying algorithm that a poly-exponential function $s(x)$ is being multisectioned, when multisectioned by m. (This is something roughly linear for a fixed $s(x)$ but the exact order is not relevant to this argument.) Consider multisectioning by $m = p_1 p_2 ... p_n$, where p_i is a non-decreasing sequence of primes (not necessarily distinct). Then to iteratively perform this multisectioning by m requires $\mathcal{O}(f(p_1) + f(p_2) + ... f(p_n)) \leq \mathcal{O}(mf(p_n))$. Thus even if $f(n) \geq n$ (i.e. $f(n)$ is worse than linear), and to multisection by a power of a prime p, say $m = p^n$, then the running time is logarithmic in m (regardless of the running time of the actual algorithm). (This ignores some of the problems associated with large integers, but is essentially correct.)

Example 23 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

This example looks at the Lucas numbers type I. Consider the linear recurrence relation $b_i = b_{i-1} + b_{i-2}$ where $b_0 = 2$ and $b_1 = 1$. Multisection this by 8 at 2. Notice that $8 = 2^3$ and further that 2 = 0 (4) + 1 (2) + 0. Any method can be used to compute the intermediate multisectioning. For this example the naive method is used.

So the first step is to calculate $s_2^0(x)$, where $\mathrm{s}(x) = \sum_{i=0}^{\infty} \frac{b_i x^i}{i!}$ with the b_is defined as above.

```
> \mapleinline{active}{1d}{s := b(i) = b(i-1) + b(i-2) , b, i, [b(0) =
> 2, b(1) = 1];}{%
> }
```

$$s := \mathrm{b}(i) = \mathrm{b}(i-1) + \mathrm{b}(i-2),\ b,\ i,\ [\mathrm{b}(0) = 2,\ \mathrm{b}(1) = 1]$$

```
> \mapleinline{active}{1d}{t := `egf/ms/naive`(s,2,0);}{%
> }
```

$$t := \mathrm{b}(i) = 3\,\mathrm{b}(i-2) - \mathrm{b}(i-4),\ b,\ i,\ [\mathrm{b}(0) = 2,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 3,\ \mathrm{b}(3) = 0]$$

Now compress this result.

```
> \mapleinline{active}{1d}{s2 := readlib(`egf/compress`)(t, 2, 0);}{%
> }
```

$$s2 := \mathrm{b}(i) = 3\,\mathrm{b}(i-1) - \mathrm{b}(i-2),\ b,\ i,\ [\mathrm{b}(0) = 2,\ \mathrm{b}(1) = 3]$$

The second step is to calculate the multisectioning of the above function s2 by 2 at 1.

```
> \mapleinline{active}{1d}{t2 := `egf/ms/naive`(s2, 2, 1);}{%
> }
```

$$t2 := \mathrm{b}(i) = 7\,\mathrm{b}(i-2) - \mathrm{b}(i-4),\ b,\ i,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 3,\ \mathrm{b}(2) = 0,\ \mathrm{b}(3) = 18]$$

Now compress the result.

```
> \mapleinline{active}{1d}{s3 := `egf/compress`(t2, 2, 1);}{%
> }
```

$$s3 := \mathrm{b}(i) = 7\,\mathrm{b}(i-1) - \mathrm{b}(i-2),\ b,\ i,\ [\mathrm{b}(0) = 3,\ \mathrm{b}(1) = 18]$$

Now the last step is to multisection the above function s3 by 2 at 0.

```
> \mapleinline{active}{1d}{t3 := `egf/ms/naive`(s3, 2, 0);}{%
> }
```

$$t3 := \mathrm{b}(i) = 47\,\mathrm{b}(i-2) - \mathrm{b}(i-4),\ b,\ i,\ [\mathrm{b}(0) = 3,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 123,\ \mathrm{b}(3) = 0]$$

By compressing this result, a linear recurrence relation for the Lucas numbers type I is found using only every 8-th term.

```
> \mapleinline{active}{1d}{s4 := `egf/compress`(t3, 2, 0);}{%
> }
```

$$s4 := \mathrm{b}(i) = 47\,\mathrm{b}(i-1) - \mathrm{b}(i-2),\ b,\ i,\ [\mathrm{b}(0) = 3,\ \mathrm{b}(1) = 123]$$

Uncompress this result to get the answer, as expected from the other commands.

```
> \mapleinline{active}{1d}{readlib(`egf/uncompress`)(s4, 8, 2);}{%
> }
```

$$\mathrm{b}(i) = 47\,\mathrm{b}(i-8) - \mathrm{b}(i-16),\ b,\ i,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 3,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = 0,\ \mathrm{b}(5) = 0,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = 0,\ \mathrm{b}(8) = 0,\ \mathrm{b}(9) = 0,\ \mathrm{b}(10) = 123,\ \mathrm{b}(11) = 0,\ \mathrm{b}(12) = 0,\ \mathrm{b}(13) = 0,\ \mathrm{b}(14) = 0,\ \mathrm{b}(15) = 0]$$

Notice that using the naive method directly to multisection by 8 at 2 gives the same result, but the method takes much longer to work.

```
> \mapleinline{active}{1d}{`egf/ms/naive`(s,8,2);}{%
> }
```

$$\mathrm{b}(i) = 47\,\mathrm{b}(i-8) - \mathrm{b}(i-16),\ b,\ i,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 3,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = 0,\ \mathrm{b}(5) = 0,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = 0,\ \mathrm{b}(8) = 0,\ \mathrm{b}(9) = 0,\ \mathrm{b}(10) = 123,\ \mathrm{b}(11) = 0,\ \mathrm{b}(12) = 0,\ \mathrm{b}(13) = 0,\ \mathrm{b}(14) = 0,\ \mathrm{b}(15) = 0]$$

This process has been automated with the Maple command 'egf/ms/compress'. The last option of the command specifies to use the naive method to do the underlying computation.

```
> \mapleinline{active}{1d}{`egf/ms/compress`(s, 8, 2, naive);}{%
> }
```

$$\mathrm{b}(i) = 47\,\mathrm{b}(i-8) - \mathrm{b}(i-16),\ b,\ i,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 0,\ \mathrm{b}(2) = 3,\ \mathrm{b}(3) = 0,\ \mathrm{b}(4) = 0,\ \mathrm{b}(5) = 0,\ \mathrm{b}(6) = 0,\ \mathrm{b}(7) = 0,\ \mathrm{b}(8) = 0,\ \mathrm{b}(9) = 0,\ \mathrm{b}(10) = 123,\ \mathrm{b}(11) = 0,\ \mathrm{b}(12) = 0,\ \mathrm{b}(13) = 0,\ \mathrm{b}(14) = 0,\ \mathrm{b}(15) = 0]$$

Which gives the same results.

4.6 Computing over the integers.

Doing calculations over the rationals is always expensive. This is because of the inherent problem of rational numbers of computing the greatest common divisor with every addition or multiplication. As well, memory requirements double for each addition of comparable sized rationals. For a more detailed description of these problems see Graham, Knuth and Patashnik's book *Concrete Mathematics* [16].

For this reason, it is desirable to perform the calculations over the integers if possible. Below are some conditions and techniques to get the computations to work for the integers.

Lemma 4.5 *If $s(x) \in \mathcal{P}^{\mathbb{C},\mathbb{Q}}$ say $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$, where $P^s(x) \in \mathbb{Q}[x]$, then all calculations can be performed for the b_i over the integers.*

Proof: To do this, make two observations.

The first observation is that if:

$$b_i = \frac{a_1}{c_1} b_{i-1} + \ldots + \frac{a_m}{c_m} b_{i-m},$$

with $a_i,\ c_i \in \mathbb{Z}$, then:

$$d^i b_i \quad = \quad \frac{a_1}{c_1} d^i b_{i-1} + \ldots + \frac{a_m}{c_m} d^i b_{i-m} = \frac{a_1 d}{c_1} d^{i-1} b_{i-1} + \ldots + \frac{a_m d^m}{c_m} d^{i-m} b_{i-m}.$$

So choose d such that $\frac{a_1 d}{c_1},\ \ldots,\ \frac{a_m d^m}{c_m} \in \mathbb{Z}$. This will give the relation:

$$\bar{b}_i = \bar{a}_1 \bar{b}_{i-1} + \ldots + \bar{a}_m \bar{b}_{i-m},$$

with $\bar{b}_i = b_i d^i$, and $\bar{a}_i = \frac{a_i d^i}{c_i} \in \mathbb{Z}$.

Notice that the initial values are changed to $\bar{b}_0 = b_0 d^0$, ..., $\bar{b}_m = b_0 d^m$.

The second observations is that if $\bar{b}_0 = \frac{e_0}{f_0}$, ..., $\bar{b}_m = \frac{e_m}{f_m}$, e_i, $f_i \in \mathbb{Z}$ are the initial conditions for the linear recurrence relation then by letting $\bar{d} = \mathrm{lcm}(f_0, ..., f_m)$, the linear recurrence relation:

$$\bar{d}\bar{b}_i = \bar{d}\bar{a}_1\bar{b}_{i-1} + ... + \bar{d}\bar{a}_m\bar{b}_{i-m},$$

is a calculation made completely over the integers.

■

Example 24 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the exponential generating function $\mathrm{s}(x) = \sum_{i=0}^{\infty} \frac{b_i x^i}{i!}$, *where* b_i *satisfy the linear recurrence relation* $b_i = \frac{b_{i-1}}{2} + \frac{b_{i-2}}{4}$, *with initial conditions of* $b_0 = 0$, $b_1 = \frac{1}{3}$. *Notice that the computation* $bp_i = 2^i b_i$ *using the linear recurrence relation* $2^i b_i = 2^{(i-1)} b_{i-1} + 2^{(i-2)} b_{i-2}$, *or equivalently* $bp_i = bp_{i-1} + bp_{i-2}$ *gives the same result. Remember that now the initial values are* $bp_0 = 0$ *and* $bp_1 = \frac{2}{3}$. *Now notice that if instead* $bpp_i = 3\, bp_i$ *is computed then the computation is wholly within the integers, as are the initial values. So from this it follows that* $b_i = \frac{bpp_i}{2^i 3}$. *Check this by computing the first few terms of both* $\frac{bpp_i}{2^i 3}$ *and* b_i.

```
> \mapleinline{active}{1d}{Bpp := `egf/makeproc`(bpp(i) = bpp(i-1) +
> bpp(i-2), bpp, i, }{%
> }

> \mapleinline{active}{1d}{    [bpp(0)= 0, bpp(1) = 2]):}{%
> }

> \mapleinline{active}{1d}{seq(1/3*(1/2)^i*Bpp(i),i=0..10);}{%
> }
```

$$0, \frac{1}{3}, \frac{1}{6}, \frac{1}{6}, \frac{1}{8}, \frac{5}{48}, \frac{1}{12}, \frac{13}{192}, \frac{7}{128}, \frac{17}{384}, \frac{55}{1536}$$

```
> \mapleinline{active}{1d}{B := `egf/makeproc`(b(i) = b(i-1)/2+b(i-2)/4,
> b, i, [b(0) = 0, b(1) = 1/3]):}{%
> }

> \mapleinline{active}{1d}{seq(B(i),i=0..10);}{%
> }
```

$$0, \frac{1}{3}, \frac{1}{6}, \frac{1}{6}, \frac{1}{8}, \frac{5}{48}, \frac{1}{12}, \frac{13}{192}, \frac{7}{128}, \frac{17}{384}, \frac{55}{1536}$$

4.7 Techniques for smaller recurrences.

This section is interested in methods to speed up the calculation of the coefficients of poly-exponential functions. One way, that was suggested by Wilf [30], is to do a calculation of a simpler linear recurrence relation, and then use a non-linear (yet simple) means to get the desired sequence.

This is stated formally as:

Theorem 4.3 *Let* $t(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!} \in \mathcal{P}$ *have an* N*-term linear recurrence relation* $b_i = \alpha_1 b_{i-1} + ...\alpha_N b_{i-N}$. *Let* $p(x) = \beta_n x^n + ... + \beta_0$ *be some polynomial in* $\mathbb{C}[x]$. *Then* $p(x)t(x) = \sum_{j=0}^{\infty} d_j \frac{x^j}{j!}$, *where* $d_i = \beta_n i^{(n)} b_{i-n} + \beta_{n-1} i^{(n-1)} b_{i-n+1} + ...\beta_0 b_i$.

Proof: Then:

$$\begin{aligned}\sum_{j=0}^{\infty} d_j \frac{x^j}{j!} &= p(x)t(x) = p(x)\sum_{i=0}^{\infty} b_i \frac{x^i}{i!} = \sum_{i=0}^{\infty} (\beta_n x^n + ... + \beta_0) b_i \frac{x^i}{i!} \\ &= \sum_{i=0}^{\infty} \beta_n b_i \frac{x^{i+n}(i+n)^{(n)}}{(i+n)!} + ... + \beta_0 b_i \frac{x^i}{(i)!} = \sum_{i=0}^{\infty} \beta_n b_{i-n} i^{(n)} \frac{x^i}{i!} + ...\beta_0 b_i \frac{x^i}{i!}.\end{aligned}$$

■

Example 25 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the function $\mathrm{s}(x) = (x^2+1)\left(\sum_{i=0}^{\infty} \frac{b_i x^i}{i!}\right)$, *where the* b_i*s are the Fibonacci numbers satisfing* $b_i = b_{i-1} + b_{i-2}$ *with initial values of* $b_0 = 0$ *and* $b_1 = 1$. *This example shows how to determine the linear recurrence relation for* $\mathrm{s}(x) = \sum_{i=0}^{\infty} \frac{d_i x^i}{i!}$, *where the* d_i *are to be written as functions of the* b_i. *But this can just be rewritten as* $\left(\sum_{i=0}^{\infty} \frac{b_i x^{(i+2)}}{i!}\right) + \left(\sum_{i=0}^{\infty} \frac{b_i x^i}{i!}\right)$, *which is just* $\left(\sum_{i=0}^{\infty} \frac{b_i (i+2)(i+1) x^{(i+2)}}{(i+2)!}\right) + \left(\sum_{i=0}^{\infty} \frac{b_i x^i}{i!}\right)$, *or in other words* $\sum_{i=0}^{\infty} \frac{(b_{i-2} i (i-1) + b_i) x^i}{i!}$. *There is a facility in Maple to make procedures with this additional information of the* $p(x)$ *in Theorem 4.3, (in this case* x^2+1*).*

```
> \mapleinline{active}{1d}{t := b(i) = b(i-1) + b(i-2), b, i,
> [b(0)=0,b(1)=1];}{%
> }
```

$$\mathrm{t} := \mathrm{b}(i) = \mathrm{b}(i-1) + \mathrm{b}(i-2),\ b,\ i,\ [\mathrm{b}(0) = 0,\ \mathrm{b}(1) = 1]$$

```
> \mapleinline{active}{1d}{T := `egf/makeproc`(t):}{%
> }

> \mapleinline{active}{1d}{S := `egf/makeproc`(t,i^2+1):}{%
> }
```

Check the first few cases to see if it is correct.

```
> \mapleinline{active}{1d}{seq(i*(i-1)*T(i-2)+T(i),i=0..10);}{%
> }
```

$$0, 1, 1, 8, 15, 45, 98, 223, 469, 970, 1945$$

```
> \mapleinline{active}{1d}{seq(S(i),i=0..10);}{%
> }
```

$$0, 1, 1, 8, 15, 45, 98, 223, 469, 970, 1945$$

4.8 Conclusions.

The conclusion that are listed in this section are conclusions as to which implemenations are faster, the conclusions are not for which methods are faster. This is because Maple combines a relatively sophisticate code to deal with certain problems, and some very naive methods for others. Hence the implementation of any method in this chapter can be greatly impacted on by the underlying methods used by Maple for certain problems, (for examples, solving linear systems of equations, how it performs resultants, etc).

The different methods that are possible (in combination or otherwise) are:

1. naive method (Chapter 2 Definition 2.6),
2. multiplying recurrence polynomial (Section 4.1),
3. using resultants on recurrence polynomial (Section 4.2),
4. linear algebra, (Section 4.3),
5. linear algebra with symbolic differentiation, (Section 4.4),
6. compression with any of the above methods, (Section 4.5),
7. working over the integers with any of the above methods, (Section 4.6),
8. factoring out a polynomial to reduce the size of the recurrence polynomial with any of the above methods, (Section 4.7).

- Of the first five, methods 4 and 5 are the most efficient. Multisectioning by m for $m > 7000$ are very doable problems.
- The naive method (method 1) is slow, and works poorly for $m > 14$.
- The recurrence polynomial method (method 2) works well for m that is a product of a large number of small primes. In general though, it does not work for large prime values; for primes $m > 43$, it is not really a feasible method.
- The resultant method (method 3), although not as bad as method 1 or 2 is noticeably slower than method 4 or 5. (For the situation of multisectioning the Fibonacci numbers by 1000, method 4 is faster than method 3 by a factor of 20.)
- The compression techniques (method 6) will improve the efficiency of methods 1, 3, 4, or 5, but do little for method 2, (as this method already takes into account the factorization of m). Here it is easy to do problems on the order of 10^5 (when used in combination with method 4).
- Functions rarely meet the criteria for methods 7 and 8 to be used, so they are not of interest.

Chapter 5

Calculations of recurrences for $\mathcal{R}$.

The previous chapter studied methods to determine the lacunary recurrence relations for multisectioned functions in $\mathcal{P}$. This chapter examines techniques for functions in $\mathcal{R}$.

Section 5.1 of this chapter deals with how to multisection the bottom of a rational poly-exponential function, (i.e. perform the necessary multiplication of poly-exponential functions) by looking at the recurrence polynomial and resultants. Section 5.2 looks at two different related methods to perform the multiplication for the bottom linear recurrence relation using fast Fourier transforms and linear algebra. These methods are also extended to determine the top recurrence. How to determine the top linear recurrence relation by using the knowledge about the bottom and about the numbers themselves is examined in Section 5.3. Section 5.4 investigates how symmetries in a poly-exponential function can simplify the calculation of the bottom lacunary recurrence relation. Sections 5.5 and 5.6 investigates two different methods to simplify the problem, by making sure that the work is always done over the integers, or by factoring out polynomials. The last section, Section 5.7 makes some conclusions about which methods are best for which problems.

5.1 Multisectioning recurrence polynomials by resultants.

Given $s(x)$, $t(x) \in \mathcal{P}$, with recurrence polynomials $P^s(x)$, P^t(x), it is difficult to calculate $P^{st}(x)$, the recurrence polynomial of $s(x)t(x)$. This section will demonstrate a method using resultants to perform this calculation.

Combining the results in Lemma 4.1 with the resultant (Definition 4.1) gives:

Lemma 5.1 *Let $s(x)$ and $t(x) \in \mathcal{P}$, where $s(x) = \sum_{i=1}^{n} p_i(x)e^{\lambda_i x}$ and $t(x) = \sum_{j=1}^{m} q_j(x)e^{\mu_j x}$. Then*

$P^{st}(x)|\prod_{i=1,j=1}^{i=n,j=m}(x-\lambda_i-\mu_j)^{\deg(p_i(x))+\deg(q_i(x))} = \mathrm{Res}_y(P^s(x-y),P^t(y))$.

Recall in Section 4.1 that the order in which the calculations were done made a difference in the efficiency of the computation. Here too, the same order is desirable for calculating the linear recurrence relation of $\prod_{i=0}^{m-1} t(x\omega_m^i)$.

Example 26 *Consider the following example in Maple. For more information about the Maple code, see Appendix A. For the Maple code see Appendix E. The Maple code and help files (including information about syntax) are available on the web at [1].*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the Genocchi numbers, as defined by Lehmer [19] having an exponential generating function of $\frac{2x}{e^x+1}$. *The calculation of* $\prod_{i=0}^{m-1}(e^{(x\,\omega_m{}^i)}+1)$, *where* ω_m *is* $e^{(\frac{2\pi I}{m})}$ *is of interest to compute the recurrence of the denominator. Set* $t(x)=e^x+1$ *and* $s(x)=2x$. *Assume that this function is to be multisectioned by 4. Then to do this with recurrence polynomials, first find the recurrence polynomial of* $\mathrm{t}(x)=e^x+1$. *Notice* $deg^d(t(x))=0$ *hence* $deg^d(\prod_{i=0}^{m-1}\mathrm{t}(x\,\omega_m{}^i))=0$. *This means that the resulting recurrence polynomial may have no multiple roots.*

```
> \mapleinline{active}{1d}{t := exp(x)+1;}{%
> }
```

$$t := e^x+1$$

```
> \mapleinline{active}{1d}{poly := convert_poly(convert_egf(t,f,x));}{%
> }
```

$$poly := x^2-x$$

Scale this to get the recurrence polynomial of $\mathrm{t}(-x)$ *and then use the resultant to get the result of multiplying the two poly-exponential functions together.*

```
> \mapleinline{active}{1d}{poly2 := subs(x=-x,poly);}{%
> }
```

$$poly2 := x^2+x$$

```
> \mapleinline{active}{1d}{poly3 :=
> resultant(subs(x=x-y,poly),subs(x=y,poly2),y);}{%
> }
```

$$poly3 := (x^2-x)\,(x^2+x)$$

There are no multiple roots, so factor out multiple root, and factor out the leading coefficient.

```
> \mapleinline{active}{1d}{gcd(poly3, diff(poly3,x), 'poly4'): poly4 :=
> expand(poly4/lcoeff(poly4,x));}{%
> }
```

$$poly4 := x^3 - x$$

Scale this again, to get the recurrence polynomial for $\mathrm{t}(I\,x)\,\mathrm{t}(-I\,x)$*, and then use the resultant to get the result of multiplying the two poly-exponential functions together.*

```
> \mapleinline{active}{1d}{poly5 := subs(x=I*x,poly4);}{%
> }
```

$$poly5 := -I\,x^3 - I\,x$$

```
> \mapleinline{active}{1d}{poly6 :=
> resultant(subs(x=x-y,poly4),subs(x=y,poly5),y);}{%
> }
```

$$poly6 := I\,(x^3 - x)\,(-x^4 - 4\,x^2 - 4 - x^6)$$

There will be no multiple roots, so factor out spurious multiple roots, and factor out the leading coefficient..

```
> \mapleinline{active}{1d}{gcd(poly6, diff(poly6,x), 'poly7'): poly7 :=
> expand(poly7/lcoeff(poly7,x));}{%
> }
```

$$poly7 := 3\,x^5 + x^9 - 4\,x$$

Now determine the linear recurrence relation.

```
> \mapleinline{active}{1d}{convert_rec(poly7,f,x);}{%
> }
```

$$\mathrm{f}(x) = -3\,\mathrm{f}(x-4) + 4\,\mathrm{f}(x-8)$$

Alternatively, the automated function in Maple could have been used.

```
> \mapleinline{active}{1d}{`bottom/ms/result`(t,f,x,4);}{%
> }
```

$$\mathrm{f}(x) = -3\,\mathrm{f}(x-4) + 4\,\mathrm{f}(x-8),\ f,\ x,$$
$$[\mathrm{f}(0) = 16,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = 0,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = -8,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 0,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 72]$$

Which gives the same result.

This example demonstrates how the order in which the resultants are taken is important. Also shown is how the use of the metric $deg^d(t(x))$ can be used to simplify the computation.

5.2 Fast Fourier transforms and linear algebra.

The methods of linear algebra from Section 4.3 needed to know the first $2N + k$ values, where N is the length of the recurrence polynomial, and k is a bound on the multiplicity of the roots. In a practical situation, the calculation of $\prod_{i=1}^{m} t(\omega_m^i x)$ is of interest where $f(x) = \frac{s(x)}{t(x)}$, $s(x)$, $t(x) \in \mathcal{P}$. If $t(x)$ is easy to approximate as a polynomials, then $t(x\omega_m^i)$ is also easy to approximate as a polynomial, via scaling.

Multiplying polynomials can be done quickly via the "*fast Fourier transform*". Maple uses a "*divide and conquer*" method instead of fast Fourier transform, which is still asymptotically better that the naive polynomial multiplication. All of these algorithms can use fast Fourier transform as the basis of polynomial multiplication, but it was deemed beyond the scope of this thesis to implement this method within Maple. See [12] for a proper definition of the divide and conquer and of fast Fourier transform.

Recall in Section 4.1 that the order in which the calculations were done made a difference in the efficiency of the computation. Here too, the same order is desirable for calculating the linear recurrence relation for $\prod_{i=0}^{m-1} t(x\omega_m^i)$. To determine the top linear recurrence relation, the order is not useful, and the polynomials can only be multiplied together in a naive fashion.

The calculation of multisectioning by m, where $m = d_1 d_2 ... d_k$ with $d_i \in \mathbb{Z}$ where $d_i \geq 2$, where an upper bound for $deg^P(\prod_{i=0}^{m-1}(t(x\omega_m^i))$ (from Lemma 2.5), say N and an upper bound for $deg^d(\prod_{i=0}^{m-1} t(x\omega_m^i))$ (from Lemma 2.4), say k, can use two different approaches to determine the new linear recurrence relation.

5.2.1 Fast Fourier transform method 1.

Calculate a polynomial approximation of $t(x)$ to degree $2N + k$, call this $p(x)$. Then iteratively perform:

$$\prod_{i_k=0}^{d_k-1} ... \prod_{i_2=0}^{d_2-1} \prod_{i_1=0}^{d_1-1} p(x\omega_{d_1}^{i_1}\omega_{d_1 d_2}^{i_2} ... \omega_{d_1 d_2 .. d_k}^{i_k}),$$

by the fast Fourier transform, doing the inner multiplication first, and using scaling for the next level out, etc. Each time a multiplication is done, truncate the polynomial to degree $2N + k$ as any component of the polynomial past that point is not of interest. After this, use linear algebra on the coefficients, to determine what the linear recurrence relation would be. Scaling by a factor of $(2N + k)!$ avoids using rationals in these calculations (assuming $t(x) \in \mathcal{P}^{\mathbb{C},\mathbb{Z}}$).

The problem with this is that the first few multiplications are expensive, as these are dense polynomials of typically large degree.

As a result of implementing this, a bug in Maple was found, which made the original method to scaling very inefficient. This bug had to do with inefficient powering of roots of unity. See Appendix D Section D.1 for more information about this.

Example 27 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

*When looking at the "*Euler numbers*" [2], generated by* $\frac{2}{e^x+e^{(-x)}}$*, the calculation of* $\prod_{i=0}^{m-1}(e^{(x\,{\omega_m}^i)}+e^{(-x\,{\omega_m}^i)})$*, where* ω_m *is* $e^{(\frac{2\,\pi\,I}{m})}$ *is of interest. Set* $\mathrm{t}(x)=e^x+e^{(-x)}$ *and* $\mathrm{s}(x)=2$*. This example will multisection by 4. An upper bound on the size of the linear recurrence relation is 16 from Lemma 2.5. Also* $deg^d(t(x))=0$*, and hence* $deg^d(\prod_{i=0}^{m-1}\mathrm{t}(x\,{\omega_m}^i))=0$*. So polynomials of degree 32 needs to be calculated, and then linear algebra is used to determine the result. So first calculate the Taylor series approximation for* $32!\,\mathrm{t}(x)$*, call this* $\mathrm{T}(x)$ *(scaling by* $32!$ *will mean that the calculation will avoid working over the rationals).*

```
> \mapleinline{active}{1d}{t := exp(x)+exp(-x);}{%
> }
```

$$\mathrm{t} := e^x + e^{(-x)}$$

```
> \mapleinline{active}{1d}{T :=
> convert(taylor(t,x=0,33),polynom)*32!;}{%
> }
```

$$\begin{aligned}\mathrm{T} := {} & 526261673867387060334436024320000000 \\ & + 263130836933693530167218012160000000\,x^2 \\ & + 21927569744474460847268167680000000\,x^4 \\ & + 730918991482482028242272256000000\,x^6 \\ & + 13052124847901464790040576000000\,x^8 \\ & + 145023609421127386556006400000\,x^{10} \\ & + 1098663707735813534515200000\,x^{12} \\ & + 6036613778768206233600000\,x^{14} + 2515255741153419264000\,x^{16} \\ & + 82197900037693440000\,x^{18} + 21631026325 7088000\,x^{20} \\ & + 468204033024000\,x^{22} + 848195712000\,x^{24} + 1304916480\,x^{26} \\ & + 1726080\,x^{28} + 1984\,x^{30} + 2\,x^{32}\end{aligned}$$

Now multiply $\mathrm{T}(x)$ *by* $\mathrm{T}(-x)$ *and divide by* $32!$*.*

```
> \mapleinline{active}{1d}{T2 := convert(series(expand(T *
> subs(x=-x, T)),x,33),polynom)/32!;}{%
> }
```

$$\begin{aligned}\mathrm{T2} := {} & 10525233477347741206688720486400000000 \\ & + 10525233477347741206688720486400000000\,x^2 \\ & + 3508411159115913735562906828800000000\,x^4 \\ & + 467788154548788498075054243840000000\,x^6 \\ & + 33413439610627749862503874560000000\,x^8 \\ & + 1485041760472344438333505536000000\,x^{10} \\ & + 45001265468858922373742592000000\,x^{12} \\ & + 989038801513382909313024000000\,x^{14} \\ & + 16483980025223048488550400000\,x^{16} \\ & + 215476863074811091353600000\,x^{18} + 22681775060506430668800\,x^{20} \\ & + 1963790048528695296000\,x^{22} + 14230362670497792000\,x^{24} \\ & + 87571462587678720\,x^{26} + 463341071892480\,x^{28} + 2130303778816\,x^{30} \\ & + 8589934592\,x^{32}\end{aligned}$$

Now scale this by I, so that the product will give an approximation for $\frac{\mathrm{T}(x)\,\mathrm{T}(-x)\,\mathrm{T}(I\,x)\,\mathrm{T}(-I\,x)}{32!}$.

```
> \mapleinline{active}{1d}{T3 := convert(series(expand(T2 *
> subs(x=I*x, T2)),x,33),polynom)/32!;}{%
> }
```

$$\begin{aligned}\mathrm{T3} := {} & 42100933909390964826754881945600000000 \\ & - 1403364463646365494225162731520000000\,x^4 \\ & + 120288382598259899505013948416000000\,x^8 \\ & - 5580156918138506374344081408000000\,x^{12} \\ & + 850573369301509302009200640000\,x^{16} \\ & - 46361548223675144287027200\,x^{20} + 1166320524473999032320000\,x^{24} \\ & - 15180906879485214720\,x^{28} + 1125934266580992\,x^{32}\end{aligned}$$

Now collect the coefficients of importance (the non-zero ones).

```
> \mapleinline{active}{1d}{for i from 0 to 32 by 4 do}{%
> }
```

```
> \mapleinline{active}{1d}{    b[i/4] := coeff(T3,x,i)*i!/32!;}{%
> }
```

```
> \mapleinline{active}{1d}{od;}{%
> }
```

$$b_0 := 16$$

$$b_1 := -128$$

$$b_2 := 18432$$

$$b_3 := -1015808$$

$$b_4 := 67633152$$

$$b_5 := -4286578688$$

$$b_6 := 275012124672$$

$$b_7 := -17590038560768$$

$$b_8 := 1125934266580992$$

Now use linear algebra to solve the linear recurrence relation.

```
> \mapleinline{active}{1d}{`recurrence/solve/linalg`(b, f, x, 4);}{%
> }
```

$$\mathrm{f}(x) = 1024\,\mathrm{f}(x-8) - 48\,\mathrm{f}(x-4)$$

This could also have be done by using the Maple function for this technique

```
> \mapleinline{active}{1d}{`bottom/ms/linalg/fft`(t,f,x,4);}{%
> }
```

$$\mathrm{f}(x) = 1024\,\mathrm{f}(x-8) - 48\,\mathrm{f}(x-4),\ f,\ x, [\mathrm{f}(0) = 16,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = 0,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = -128,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 0,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 18432]$$

Which is the same result.

5.2.2 Fast Fourier transform method 2.

Again, the calculation of interest is

$$\prod_{i_k=0}^{d_k-1} \cdots \prod_{i_2=0}^{d_2-1} \prod_{i_1=0}^{d_1-1} t(x\omega_{d_1}^{i_1}\omega_{d_1 d_2}^{i_2} \ldots \omega_{d_1 d_2 \ldots d_k}^{i_k})$$

with $t(x) \in \mathcal{P}$. Recall that method 1 (Subsection 5.2.1) performed all of these calculations with a large degree polynomial, performing the inner calculations first, and then the next level out, etc. This method differs in that the inner computation is done with a small degree polynomial, the linear recurrence relation for the inner multiplication is then determined with linear algebra, after which the large degree polynomial needed for the next computation is constructed. By scaling out a factor of $(2N+k)!$ each time, (for the various N and k as they apply to each step), can avoid using rationals in these calculations (assuming $t(x) \in \mathcal{P}^{\mathbb{C},\mathbb{Z}}$).

The advantage to this over method 1 is that the polynomials are of small degree near the beginning of the calculation when they are densest. The disadvantage is that linear algebra is repeatedly used.

As a result of implementing this, a bug in Maple was found, which made the original method to scaling very inefficient. This bug had to do with inefficient powering of roots of unity. See Appendix D Section D.1 for more information about this.

As a result of testing this on large examples, some inefficiencies with the factorial function in Maple were discovered. For more information about this, see Appendix D Section D.6.

Example 28 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the Lucas numbers as defined by Lehmer, [19]. To avoid confusion with the Lucas numbers defined in Graham, Knuth and Patashnik, [16] we will call these Lucas numbers, "Lucas numbers type II". When looking at the Lucas numbers type II generated by $\frac{x\,e^x}{e^{(2\,x)}-1}$*, the calculation of interest is* $\prod_{i=0}^{m-1}\left(e^{(2\,x\,{\omega_m}^i)}-1\right)$*, where* ω_m *is* $e^{(\frac{2\,\pi\,I}{m})}$*. Set* $\mathrm{t}(x)=e^{(2\,x)}-1$ *and* $\mathrm{s}(x)=x\,e^x$*. Assume that the function is being multisectioned by 4. Notice* $deg^d(t(x))=0$*, and hence that* $deg^d(\prod_{i=0}^{m-1}\mathrm{t}(x\,{\omega_m}^i))$ *= 0. Notice that* $deg^P(\mathrm{t}(x))=2$*, hence* $deg^P(\mathrm{t}(x)\,\mathrm{t}(-x))$ *is at most 4. So for the first step only a linear recurrence relation to degree 8 is needed. So first calculate the taylor series approximation for* $\mathrm{t}(x)$*, call this* $8!\,\mathrm{T}(x)$ *(scale by* $8!$ *to avoid having to work over the rationals).*

```
> \mapleinline{active}{1d}{t := exp(2*x)-1;}{%
> }
```

$$\mathrm{t} := e^{(2\,x)}-1$$

```
> \mapleinline{active}{1d}{T :=
> convert(taylor(t,x=0,9),polynom)*8!;}{%
> }
```

$$\mathrm{T} := 80640\,x+80640\,x^2+53760\,x^3+26880\,x^4+10752\,x^5+3584\,x^6+1024\,x^7 + 256\,x^8$$

Now multiply $\mathrm{T}(x)$ *by* $\mathrm{T}(-x)$ *and divide by 8!.*

```
> \mapleinline{active}{1d}{T2 := convert(series(expand(T *
> subs(x=-x,}{%
> }

> \mapleinline{active}{1d}{T)),x,9),polynom)/8!;}{%
> }
```

$$\mathrm{T2} := -161280\,x^2-53760\,x^4-7168\,x^6-512\,x^8$$

Determine the interesting (non-zero) values.

```
> \mapleinline{active}{1d}{for i from 0 to 4 do }{%
> }
```

```
> \mapleinline{active}{1d}{    b[i] := coeff(T2,x,2*i)*(2*i)!/8!; }{%
> }
```

```
> \mapleinline{active}{1d}{od;}{%
> }
```

$$b_0 := 0$$

$$b_1 := -8$$

$$b_2 := -32$$

$$b_3 := -128$$

$$b_4 := -512$$

Solve this linear recurrence relation.

```
> \mapleinline{active}{1d}{rec := `recurrence/solve/linalg`(b, f, x,
> 2);}{%
> }
```

$$rec := \mathrm{f}(x) = 4\,\mathrm{f}(x-2)$$

```
> \mapleinline{active}{1d}{t2 := rec, f, x, [f(0) = b[0], f(1) = 0,
> }{%
> }
```

```
> \mapleinline{active}{1d}{f(2) = b[1], f(3) = 0, f(4) = b[2], f(5) = 0,
> }{%
> }
```

```
> \mapleinline{active}{1d}{f(6) = b[3], f(7) = 0, f(8) = b[4]];}{%
> }
```

$$\mathrm{t2} := \mathrm{f}(x) = 4\,\mathrm{f}(x-2),\ f,\ x, [\mathrm{f}(0) = 0, \mathrm{f}(1) = 0, \mathrm{f}(2) = -8, \mathrm{f}(3) = 0, \mathrm{f}(4) = -32, \mathrm{f}(5) = 0, \mathrm{f}(6) = -128, \mathrm{f}(7) = 0, \mathrm{f}(8) = -512]$$

Now determine what $deg^P(T2(x))$ and $deg^d(T2(x))$ are, as these will be useful in the calculation.

```
> \mapleinline{active}{1d}{`egf/metric/P`(t2);}{%
> }
```

```
> \mapleinline{active}{1d}{`egf/metric/d`(t2);}{%
> }
```

$$0$$

Notice that $deg^P(t2(x)) = 3$, *and hence* $deg^P(t2(x) * t2(I * x))$ *is at most 9. Thus only the first 18 terms of the polynomial approximation needs to be calculated, say* $18!\,\mathrm{t2}(x)$ *(scale by* $18!$ *to avoid having to work over the rationals). Call this T2(x).*

```
> \mapleinline{active}{1d}{Fun := `egf/makeproc`(t2):}{%
> }

> \mapleinline{active}{1d}{T2 := add (Fun(i)*x^i/i!,i=0..18)*18!;}{%
> }
```

$$\begin{aligned}\mathrm{T2} := &-25609494822912000\,x^2 - 8536498274304000\,x^4 \\ &- 1138199769907200\,x^6 - 81299983564800\,x^8 - 3613332602880\,x^{10} \\ &- 109494927360\,x^{12} - 2406481920\,x^{14} - 40108032\,x^{16} - 524288\,x^{18}\end{aligned}$$

So now multiply $\mathrm{T2}(x)$ *by* $\mathrm{T2}(I\,x)$ *and divide by 18!.*

```
> \mapleinline{active}{1d}{T3 := convert(series(expand(T2 *
> subs(x=I*x, T2)),x,19),polynom)/18!;}{%
> }
```

$$\begin{aligned}\mathrm{T3} := &-102437979291648000\,x^4 + 2276399539814400\,x^8 \\ &- 14453330411520\,x^{12} + 20374880256\,x^{16}\end{aligned}$$

Collect the interesting (non-zero) terms.

```
> \mapleinline{active}{1d}{for i from 0 to 4 do }{%
> }

> \mapleinline{active}{1d}{    b[i] := coeff(T3,x,4*i)*(4*i)!/18!;
> }{%
> }

> \mapleinline{active}{1d}{od;}{%
> }
```

$$b_0 := 0$$

$$b_1 := -384$$

$$b_2 := 14336$$

$$b_3 := -1081344$$

$$b_4 := 66584576$$

Solve this linear recurrence relation.

```
> \mapleinline{active}{1d}{rec := `recurrence/solve/linalg`(b, f, x,
> 4);}{%
> }
```

$$rec := \mathrm{f}(x) = 1024\,\mathrm{f}(x-8) - 48\,\mathrm{f}(x-4)$$

This also could have been done by using the Maple function for this technique

```
> \mapleinline{active}{1d}{`bottom/ms/linalg/fft2`(t,f,x,4);}{%
> }
```

$$\mathrm{f}(x) = 1024\,\mathrm{f}(x-8) - 48\,\mathrm{f}(x-4),\ f,\ x,\ [$$
$$\mathrm{f}(0) = 0,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = 0,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = -384,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 0,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 14336]$$

Which is the same result.

It is worth pointing out in this example that fewer terms of the polynomial needed to be worked out. This was because a better bound for $deg^P(\prod_{i=0}^{m-1} t(x\omega_m^i)$ was known as a result of the iteratively calculating $t(x)t(-x)$ and then $t(x)t(-x)t(Ix)t(-Ix)$.

5.3 Using the bottom linear recurrence relation.

The method described in Section 4.3 is easy if $s(x) \in \mathcal{P}$ is known in poly-exponential function form. But there are situations when to explicitly calculate what $s(x)$ is in poly-exponential function form is space consuming and undesirable. For example when trying to determine the top linear recurrence relation of a rational poly-exponential function.

Consider a rational poly-exponential function $\frac{s(x)}{t(x)}$ where $s(x),\ t(x) \in \mathcal{P}$ with $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ and $t(x) = \sum_{j=0}^{\infty} d_j \frac{x^j}{j!}$. Further assume $\frac{s(x)}{t(x)} = \sum_{i=0}^{\infty} c_i \frac{x^i}{i!}$. This gives

$$\sum_{j=s}^{i} \binom{i}{j} d_j c_{i-j} \quad = \quad b_i \tag{5.1}$$

Then if a simple formulae for the d_is and c_is are known, then the b_i can be determined using Equation 5.1. If a bound on the size of the linear recurrence relation for the b_i is known, say N, and a bound for the metric deg^d on the linear recurrence relation for the b_i is known, say k, then only the first $2N + k$ values of b_i need be calculated to determine the linear recurrence relation for the b_i.

Recall from Section 2.4 that typically the linear recurrence relation for multisectioning some q will be the same regardless of the value of q. This can be utilized here by using the process above

for the top when multisectioned by m at 0, and then assume that the linear recurrence relation will be the same when multisectioning at other values of q. Hence linear algebra need not be used to determine the linear recurrence relation but instead simply reuse the linear recurrence relation from the first calculation, thus simplifying future calculations immensely.

Example 29 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

This example tries to find the linear recurrence relation for the top of the Euler numbers $\mathrm{f}(x) = \frac{2}{e^x+e^{(-x)}} = \sum_{i=0}^{\infty} \frac{c_i\, x^i}{i!} = \frac{\sum_{i=0}^{\infty} \frac{b_i\, x^i}{i!}}{\sum_{j=0}^{\infty} \frac{d_j\, x^j}{j!}}$ *given the bottom linear recurrence relation, when multisectioning by 4 at 0. As the function is being multisectioned by 4 at 0, then only those* b_i *where* $i = 0 \bmod 4$ *are needed.*

```
> \mapleinline{active}{1d}{bot :=
> `bottom/ms/linalg/fft2`(exp(x)+exp(-x),f,x,4);}{%
> }
```

$$bot := \mathrm{f}(x) = 1024\,\mathrm{f}(x-8) - 48\,\mathrm{f}(x-4),\ f,\ x, [\mathrm{f}(0) = 16,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = 0,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = -128,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 0,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 18432]$$

```
> \mapleinline{active}{1d}{Bot := `egf/makeproc`(bot):}{%
> }
```

Now $b_i = \sum_{j=0}^{i} \mathrm{binomial}(i,\, j)\, c_{i-j}\, d_j$ *from Equation 5.1. An upper bound of the number of* b_i *needed as* $4\,2^3\,2\,2 + 2 = 130$ *by Lemma 2.5.*

```
> \mapleinline{active}{1d}{F := i ->
> add(binomial(i,j)*euler(i-j)*`Bot`(j),j=0..i);}{%
> }
```

```
Warning, `j` in call to `add` is not local
```

$$F := i \to \mathrm{add}(\mathrm{binomial}(i,\, j)\, \mathrm{euler}(i-j)\, \mathrm{Bot}(j),\ j = 0..i)$$

```
> \mapleinline{active}{1d}{for i from 4 to 130 by 4 do}{%
> }
> \mapleinline{active}{1d}{    b[i/4] := F(i):}{%
> }
> \mapleinline{active}{1d}{od:}{%
> }
> \mapleinline{active}{1d}{rec := `recurrence/solve/linalg`(b,f,x,4);}{%
```

```
> }
```

$$rec := \mathrm{f}(x) = 625\,\mathrm{f}(x-12) - 611\,\mathrm{f}(x-8) - 13\,\mathrm{f}(x-4)$$

This could have also been discovered by using some of the other built in functions.

```
>
> \mapleinline{active}{1d}{`top/ms/linalg/fft`(2,exp(x)+exp(-x),f,x,4,2);
> }{%
> }
```

$$\mathrm{f}(x) = 625\,\mathrm{f}(x-12) - 611\,\mathrm{f}(x-8) - 13\,\mathrm{f}(x-4),\ f,\ x, [\mathrm{f}(0) = 0,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = -16,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = 0,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 944,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 0,\ \mathrm{f}(9) = 0,\ \mathrm{f}(10) = 1904,\ \mathrm{f}(11) = 0]$$

```
>
> \mapleinline{active}{1d}{`top/ms/linalg/sym`(2,exp(x)+exp(-x),f,x,4,2);
> }{%
> }
```

$$\mathrm{f}(x) = 625\,\mathrm{f}(x-12) - 611\,\mathrm{f}(x-8) - 13\,\mathrm{f}(x-4),\ f,\ x, [\mathrm{f}(0) = 0,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = -16,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = 0,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 944,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 0,\ \mathrm{f}(9) = 0,\ \mathrm{f}(10) = 1904,\ \mathrm{f}(11) = 0]$$

This method is automated with the given function below.

```
> \mapleinline{active}{1d}{`top/ms/linalg/know`(Bot, euler, f, x, 4, 2,
> 16, 2);}{%
> }
```

$$\mathrm{f}(x) = 625\,\mathrm{f}(x-12) - 611\,\mathrm{f}(x-8) - 13\,\mathrm{f}(x-4),\ f,\ x, [\mathrm{f}(0) = 0,\ \mathrm{f}(1) = 0,\ \mathrm{f}(2) = -16,\ \mathrm{f}(3) = 0,\ \mathrm{f}(4) = 0,\ \mathrm{f}(5) = 0,\ \mathrm{f}(6) = 944,\ \mathrm{f}(7) = 0,\ \mathrm{f}(8) = 0,\ \mathrm{f}(9) = 0,\ \mathrm{f}(10) = 1904,\ \mathrm{f}(11) = 0]$$

Which all give the same result.

Now determine the linear recurrence relation multisectioned by 4 at 2. Taking advantage of the fact of what the linear recurrence relation most likely is, all that really needs to be done is to determine the initial values, and see if the linear recurrence relation is correct. By looking at the recurrence that for the top multisectioned by 4 at 0 that there are only about 12 terms needed. Calculate the first 32 terms for when the function is multisectioned by 4 at 2, and see if this linear recurrence relation holds.

```
> \mapleinline{active}{1d}{initial :=
> [seq( op([ f(4*i) = F(4*i), f(4*i+1) = 0, f(4*i+2) = 0, f(4*i+3) \newline
> = 0 ]), i=0..8)];}{
> %
> }
```

$$\begin{aligned} \mathit{initial} := [&\mathrm{f}(0) = 16, \mathrm{f}(1) = 0, \mathrm{f}(2) = 0, \mathrm{f}(3) = 0, \mathrm{f}(4) = -48, \mathrm{f}(5) = 0, \mathrm{f}(6) = 0, \\ &\mathrm{f}(7) = 0, \mathrm{f}(8) = -4208, \mathrm{f}(9) = 0, \mathrm{f}(10) = 0, \mathrm{f}(11) = 0, \mathrm{f}(12) = 94032, \\ &\mathrm{f}(13) = 0, \mathrm{f}(14) = 0, \mathrm{f}(15) = 0, \mathrm{f}(16) = 1318672, \mathrm{f}(17) = 0, \mathrm{f}(18) = 0, \\ &\mathrm{f}(19) = 0, \mathrm{f}(20) = -77226288, \mathrm{f}(21) = 0, \mathrm{f}(22) = 0, \mathrm{f}(23) = 0, \\ &\mathrm{f}(24) = 257003152, \mathrm{f}(25) = 0, \mathrm{f}(26) = 0, \mathrm{f}(27) = 0, \mathrm{f}(28) = 44668390992, \\ &\mathrm{f}(29) = 0, \mathrm{f}(30) = 0, \mathrm{f}(31) = 0] \end{aligned}$$

```
> \mapleinline{active}{1d}{`egf/clean`(rec, f, x, initial);}{%
> }
```

$$\begin{aligned} &\mathrm{f}(x) = 625\,\mathrm{f}(x-12) - 611\,\mathrm{f}(x-8) - 13\,\mathrm{f}(x-4),\ f,\ x, [\mathrm{f}(0) = 16, \mathrm{f}(1) = 0, \mathrm{f}(2) = 0, \\ &\quad \mathrm{f}(3) = 0, \mathrm{f}(4) = -48, \mathrm{f}(5) = 0, \mathrm{f}(6) = 0, \mathrm{f}(7) = 0, \mathrm{f}(8) = -4208, \mathrm{f}(9) = 0, \\ &\quad \mathrm{f}(10) = 0, \mathrm{f}(11) = 0] \end{aligned}$$

When cleaning up all of the terms, (getting rid of the terms that can be calculated based on the linear recurrence relation) then fewer than the 32 terms are left. Hence, this linear recurrence relation is most probably correct.

This could have done this with the automated function.

```
> \mapleinline{active}{1d}{`top/ms/know`(rec, Bot, euler, f, x, 4, 0,
> 130);}{%
> }
```

$$\begin{aligned} &\mathrm{f}(x) = 625\,\mathrm{f}(x-12) - 611\,\mathrm{f}(x-8) - 13\,\mathrm{f}(x-4),\ f,\ x, [\mathrm{f}(0) = 16, \mathrm{f}(1) = 0, \mathrm{f}(2) = 0, \\ &\quad \mathrm{f}(3) = 0, \mathrm{f}(4) = -48, \mathrm{f}(5) = 0, \mathrm{f}(6) = 0, \mathrm{f}(7) = 0, \mathrm{f}(8) = -4208, \mathrm{f}(9) = 0, \\ &\quad \mathrm{f}(10) = 0, \mathrm{f}(11) = 0] \end{aligned}$$

Which gives the same result.

As a result of working on this example, a bug in the help for the Euler function in Maple was found. For more information see Appendix D Section D.2.

5.4 Symmetries.

Recall Lemma 3.1 showed that when multisectioning a rational poly-exponential function $\frac{s(x)}{t(x)}$ by m at q then the bottom poly-exponential function could be written as $\prod_{i=0}^{m-1} t(x\omega_m^i)$ and the top as $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$. Doing this made the simplifying assumption that there were no common factors among the $t(x\omega_m^i)$, as $0 \leq i \leq m-1$. For numerous examples of functions, such as the Bernoulli, Euler, Genocchi and Lucas type II numbers, this assumption is not true. (Some rewriting of the Bernoulli and Genocchi functions are needed for this.) This section explores a small subset of the possible situations where this assumption is not valid, and how, by looking at these common factors, the size of the linear recurrence relation can be reduced for the bottom.

These properties have been exploited before in the standard papers on Bernoulli and Euler numbers [9, 19], but, to the best of my knowledge, have not been written in this type of generality before, nor has there been a formal theory behind what is being done.

To this end, define a symmetry.

Definition 5.1 (Symmetry.) *A poly-exponential function, $s(x)$ has a "*symmetry of order p*" if*

$$s(x\omega_p) = \omega_p^k s(x)$$

for some integer k.

Example 30 *The denominator of the Euler numbers $e^x + e^{-x}$ has a symmetry of order 2.*

Note 5.1 *If $s(x)$ has a symmetry of order p, say $s(x\omega_p) = \omega_p^k s(x)$, then $s(x) = s_p^k(x)$.*

If a symmetry of a function is known, then it can be taken advantage of to find a smaller form for the linear recurrence relation of the denominator of a multisectioned rational poly-exponential function.

Theorem 5.1 *Let $f(x) = \frac{s(x)}{t(x)}$, where $s(x)$, $t(x) \in \mathcal{P}$, and let $t(x)$ have a symmetry of order p, say $t(x\omega_p) = \omega_p^k t(x)$. Further, let $p|m$. Then a recursion formula can be found for the coefficients of x^{mi+q} of the exponential generating function of $f(x)$ that depends only on the coefficients of x^{mj+q}, for $j < i$, and two lacunary recurrence relations, where the lacunary recurrence relation for the denominator has a smaller upper bound on its length than that of Theorem 3.3.*

Proof: Now

$$f_m^q(x) = \frac{1}{m}\sum_{i=0}^{m-1} \frac{\omega_m^{-iq} s(x\omega_m^i)}{t(x\omega_m^i)} = \frac{1}{m}\sum_{i=0}^{m/p-1}\sum_{j=0}^{p-1} \frac{\omega_m^{-(i+j(m/p))q} s(x\omega_m^{i+j(m/p)})}{t(x\omega_m^{i+j(m/p)})}$$

$$
\begin{aligned}
&= \frac{1}{m}\sum_{i=0}^{m/p-1}\sum_{j=0}^{p-1}\frac{\omega_m^{-iq}\omega_p^{-jq}s(x\omega_m^i\omega_p^j)}{t(x\omega_m^i\omega_p^j)} = \frac{1}{m}\sum_{i=0}^{m/p-1}\sum_{j=0}^{p-1}\frac{\omega_m^{-iq}\omega_p^{-jq}s(x\omega_m^i\omega_p^j)}{\omega_p^{jk}t(x\omega_m^i)}\\
&= \frac{1}{m}\sum_{i=0}^{m/p-1}\sum_{j=0}^{p-1}\frac{\omega_m^{-iq}\omega_p^{-jq-jk}s(x\omega_m^i\omega_p^j)}{t(x\omega_m^i)}\\
&= \frac{1}{m}\sum_{j=0}^{p-1}\sum_{i=0}^{m/p-1}\frac{\omega_m^{-iq}\omega_p^{-jq-jk}s(x\omega_m^i\omega_p^j)\prod_{l=1}^{m/p-1}t(x\omega_m^l)}{\prod_{l=0}^{m/p-1}t(x\omega_m^l)}\\
&= \frac{\frac{1}{m}\sum_{j=0}^{p-1}(\sum_{i=0}^{m/p-1}\omega_m^{-iq}\omega_p^{-jq-jk}s(x\omega_m^i\omega_p^j)\prod_{l=1}^{m/p-1}t(x\omega_m^l))}{\prod_{l=0}^{m/p-1}t(x\omega_m^l)}
\end{aligned}
$$

By observing that $t(x) = t_p^k(x)$ a careful analysis shows that $\prod_{l=0}^{m/p-1}t(x\omega_m^l) = (\prod_{l=0}^{m/p-1}\ t(x\ \omega_m^l))_m^{km/p}$. Denote this $r_m^{km/p}(x)$. Further, letting $r_m^{km/p}(x) = \sum_{j=0}^{\infty}d_j\frac{x^j}{j!}$, $f(x) = \sum_{i=0}^{\infty}c_i\frac{x^i}{i!}$ and the numerator as $\sum_{i=0}^{\infty}b_i\frac{x^i}{i!}$ gives, from Equation 5.1 that $b_i = 0$ unless $i \equiv q + m/pk \pmod m$. So both the numerator and the denominator are lacunary recurrence relation.

Further, from Lemma 2.5 the denominator $r_m^{mk/p}(x)$ has the property that $deg^P(r_m^{km/p}(x)) \leq deg^P(t(x))^{m/p}$, which is better than the upper bound in Theorem 3.3 of $deg^P(t(x))^m$.

■

Example 31 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

Consider the example of the Euler numbers, given by the exponential generating function of $\frac{2}{e^x+e^{(-x)}}$. *The denominator of this has a symmetry of order 2. Below are two methods to compute the recurrence for the denominator, when multisectioned by 8. The first method does not take into account the symmetry, where as the second does. Also demonstrated in this section is the code 'egf/strip', which will strip away the useless zeros.*

```
> \mapleinline{active}{1d}{botNoSym :=
> `egf/strip`(`bottom/ms/linalg/fft2`(exp(x)+exp(-x),f,x,8,[2,2,2]), 8,
> 0);}{%
> }
```

$$
\begin{aligned}
botNoSym := \mathrm{f}(x) = &-831705558809741310321986973047193 6\,\mathrm{f}(x-80)\\
&+372330027815123875790980360154644 48\,\mathrm{f}(x-72)\\
&+116678803396213749326868515074867 2\,\mathrm{f}(x-64)\\
&-285993709711940870227856719872 0\,\mathrm{f}(x-56)\\
&-946119117903717195314311987 2\,\mathrm{f}(x-48)
\end{aligned}
$$

$$
\begin{aligned}
&+ 2389168763320088873926656\,\mathrm{f}(x-40)\\
&+ 543960885098446848\,\mathrm{f}(x-32) + 3635955734937600\,\mathrm{f}(x-24)\\
&+ 158590697472\,\mathrm{f}(x-16) - 283392\,\mathrm{f}(x-8),\ f,\ x, [\mathrm{f}(0) = 256,\\
&\mathrm{f}(8) = -557056,\ \mathrm{f}(16) = 3901315088384,\ \mathrm{f}(24) = -968280866994257920,\\
&\mathrm{f}(32) = 88960303500317006653030 4,\\
&\mathrm{f}(40) = -391268789233378370377876504576,\\
&\mathrm{f}(48) = 24844419386894193060128270311227392 0,\\
&\mathrm{f}(56) = -12921533069165612319408971748216588048 7936,\\
&\mathrm{f}(64) = 74595026599387417869017590514149872898213412864,\\
&\mathrm{f}(72) = -40726729378210421739875778036712241401761762629386240,\\
&\mathrm{f}(80) =\\
&22901077288442548007301641325421696523514722946588788916224]
\end{aligned}
$$

```
> \mapleinline{active}{1d}{botSym :=
> `egf/strip`(`bottom/ms/linalg/fft2`(exp(x)+exp(-x),f,x,8,[2,2,2],2),
> 8, 0);}{%
> }
```

$$botSym := \mathrm{f}(x) = -4096\,\mathrm{f}(x-16) - 2176\,\mathrm{f}(x-8),\ f,\ x,\ [\mathrm{f}(0) = 16,\ \mathrm{f}(8) = -17408]$$

```
> \mapleinline{active}{1d}{BotNoSym := `egf/makeproc`(botNoSym):}{%
> }
```

```
> \mapleinline{active}{1d}{BotSym := `egf/makeproc`(botSym):}{%
> }
```

Next consider the top recurrence, determined by the bottom recurrence and the definition of the Euler numbers, when multisectioning by 8 at 0. Again, the first method does not take into account symmetries, where as the second does.

```
> \mapleinline{active}{1d}{topNoSym :=
> `egf/strip`(`top/ms/linalg/know`(BotNoSym, euler, f, x, 8, 0, 30,
> 2),8,0);}{%
> }
```

$$
\begin{aligned}
topNoSym := \mathrm{f}(x) = &-43920259282210583351533605075940239625113 46\backslash\\
&4868397821096440262 0\mathrm{f}(x-112) + 16393772837213378973317\backslash\\
&9388046674695228076550941155503547265532249311 3\mathrm{f}(x-128) -\\
&6793561703202246662336295977172054217035178878235409832186 0\\
&\mathrm{f}(x-104) +\\
&2876648532964249458940710162842517424309120851325640780\\
&\mathrm{f}(x-96) +\\
&1231735568549238110339881112892338931107684228529815 1
\end{aligned}
$$

$$
\begin{aligned}
&\mathrm{f}(x-88)+6558320624493722290765710044172635933557278001\backslash\\
&3113106869531093995 6\mathrm{f}(x-120)+2993228897753578954485\backslash\\
&4647110094907946387589481698636512048824915244243092 0\backslash\\
&7\mathrm{f}(x-144)-21560794660949732482905702\,\mathrm{f}(x-40)\\
&-1147574017569591751566\,\mathrm{f}(x-32)+40165361247172240331\backslash\\
&3427114798146852727676823111131311669932336239343894 1\\
&\mathrm{f}(x-136)+7853492007095947684783471067820024453438489 1\backslash\\
&8616770779592494739390443923558731\mathrm{f}(x-152)-131973\,\mathrm{f}(x-8)\\
&-134667150111\,\mathrm{f}(x-16)\\
&-95174145854476520680346374020 58\,\mathrm{f}(x-48)\\
&-92512594457554741735374579001443560538 03\,\mathrm{f}(x-64)\\
&+84498622102085814949560058480710284331283721\,\mathrm{f}(x-72)\\
&-1133062245492 7027\,\mathrm{f}(x-24)\\
&+2385705997943699776309273668532297345747765388163\,\mathrm{f}(x-80)\\
&+8130258237574025533842932844632118 06\,\mathrm{f}(x-56)-534357\backslash\\
&7892540217404359358265212387701756550406444636204178 2\backslash\\
&9564549499574770921434174 1\mathrm{f}(x-192)+48814666275054200\backslash\\
&1959884721019894101698944109780514309347770217348049 6\backslash\\
&780911470929727\mathrm{f}(x-200)-16533988700569217371853899 90\backslash\\
&8884152597118757650819502217162561473623123323028529 0\backslash\\
&971\mathrm{f}(x-208)+2326130891384570590380157721546063909849\backslash\\
&3030873003515999259565279964744546250390625\mathrm{f}(x-216)+\\
&6386324510731326302710718030142279040608032820925582 8\backslash\\
&9220903993807817345738772746958\mathrm{f}(x-184)-320988767861\backslash\\
&0118163845765170772200606809423123800354392414411198 5\backslash\\
&7085740733979542 66\mathrm{f}(x-176)-2210912755112381032331783\backslash\\
&3789252547717842872308441294637880749455926631158983 9\backslash\\
&3462\mathrm{f}(x-168)-1336067517229985307750611681782270299 48\backslash\\
&2572698765737852524999381919199459906027 18\mathrm{f}(x-160),\ f,\ x,[\\
&\mathrm{f}(0)=256,\,\mathrm{f}(8)=-202496,\,\mathrm{f}(16)=-1063953149696,\\
&\mathrm{f}(24)=64570730111514880,\,\mathrm{f}(32)=11475408412808238521 5744,\\
&\mathrm{f}(40)=-12617880498158977441699755776,\\
&\mathrm{f}(48)=-13558757497291064142754260447399680,\\
&\mathrm{f}(56)=21706198058970921333822210605329178851 84,\\
&\mathrm{f}(64)=15589104696765723271933888452504847360386173 44,\\
&\mathrm{f}(72)=-33388357131041594090540148156576875911690148418944 0,\\
&\mathrm{f}(80)=\\
&-17566284064452068398517686175037197689304053697426459417 6,
\end{aligned}
$$

$$\mathrm{f}(88) = 484965667430663900125702569069025106198846760656\backslash$$
$$73716295825664, \mathrm{f}(96) = 193208469295406084354751329180571\backslash$$
$$49348363091224927642362608361529600, \mathrm{f}(104) = -6772366215\backslash$$
$$58780337958692538975970599262543691319285386324099989\backslash$$
$$2141422336, \mathrm{f}(112) = -20617726717245267743646468252135139\backslash$$
$$63598057011996874738073663893749583225474816, \mathrm{f}(120) = 91\backslash$$
$$77542784877584642796201877918487801614158162503036098\backslash$$
$$669031044388172468649666214 40, \mathrm{f}(128) = 21143779189020025\backslash$$
$$19981142759968877814678251583679412389531928218427761\backslash$$
$$85067654949642600704, \mathrm{f}(136) = -12138020124754595767708 70\backslash$$
$$37339337942042861219535152681693440017948231511735765\backslash$$
$$368016547875428096, \mathrm{f}(144) = -20484343364395697460494343 2\backslash$$
$$60014391790909623367791573352316771152068070097942288\backslash$$
$$026991331458997169920, \mathrm{f}(152) = 15724546218391933128930 23\backslash$$
$$47331717651025235741794364808372617318943520862719524\backslash$$
$$52387887218455502637116274944, \mathrm{f}(160) = 18101793798981768\backslash$$
$$97468850458775460758110185354473802874927366725986819\backslash$$
$$47631352273021636968066240080652770942182 4, \mathrm{f}(168) = -199\backslash$$
$$99867933843650702413266982315969889081786544044255134\backslash$$
$$29193831155787180531188119439676780471386768670887694\backslash$$
$$728820480, \mathrm{f}(176) = -13311961863312070190185051208577110 8\backslash$$
$$32431364491287555186444541707366739308588765758549330\backslash$$
$$57365948078468045702926795507996 16, \mathrm{f}(184) = 250105285632\backslash$$
$$03785116103490520672684517978976645058273583097955243\backslash$$
$$88734189966221418884084211412008912595964318134910812\backslash$$
$$70300530944, \mathrm{f}(192) = 52588307706405763257356025072347343\backslash$$
$$60049528472499715635754362341679820173967167656920852\backslash$$
$$96048886590010893764649480773349501941237376 0, \mathrm{f}(200) = -\backslash$$
$$30775725978976969510046824935955938885269811664308452\backslash$$
$$69243135131736053830822282944758255863223017288525489\backslash$$
$$52056605785222164092004512130229 76, \mathrm{f}(208) = 711146486248\backslash$$
$$35393457944380270059054429282827263572950086035928327\backslash$$
$$17654936312173082292376933076484311275742660417555667\backslash$$
$$52813061990395310739755264]$$

```
> \mapleinline{active}{1d}{topSym :=
> `egf/strip`(`top/ms/linalg/know`(BotSym, euler, f, x, 8, 0, 30,
> 2),8,0);}{%
> }
```

$topSym :=$
$\mathrm{f}(x) = -6561\,\mathrm{f}(x-32) + 7571428\,\mathrm{f}(x-24) - 45798\,\mathrm{f}(x-16) + 1188\,\mathrm{f}(x-8),$
$f,\ x,\ [\mathrm{f}(0) = 16,\ \mathrm{f}(8) = 4752,\ \mathrm{f}(16) = 5278992,\ \mathrm{f}(24) = 6144667536]$

So both the top and the bottom recurrences are smaller when the symmetries of the denominator are taken into account.

5.5 Computing over the integers.

Recall in Section 4.6 that all of the calculations of the coefficients of the exponential generating function of a poly-exponential function can be calculated over the integers if certain criteria are met. Here, a similar result holds, given certain criteria all of the calculations of the coefficients of the exponential generating function of a rational poly-exponential function can be done over the integers.

Consider the equations in Theorem 3.1 again. This gives the following lemma.

Lemma 5.2 *If* $f(x) = \frac{s(x)}{t(x)} = \sum_{i=0}^{\infty} c_i \frac{x^i}{i!}$ *where* $s(x),\ t(x) \in \mathcal{P}$, *with* $s(x) = \sum_{i=0}^{\infty} b_i \frac{x^i}{i!}$ *and* $t(x) = \sum_{j=0}^{\infty} d_j \frac{x^j}{j!}$ *such that* $d_0 \neq 0$, *where* $d_i,\ b_i \in \mathbb{Q}$, *and* $P^s(x),\ P^t(x) \in \mathbb{Q}[x]$ *then all of the calculations of the* c_i *can be done over the integers.*

Proof: A few observations are needed to see this.

Without loss of generality, let $m = 1$ and $q = 0$. Based on the equation of Theorem 3.1 the following equation holds:

$$c_{k-s} = \frac{1}{\binom{k}{s} d_s}(b_k - \sum_{j=s+1}^{k} \binom{i}{j} d_j c_{i-j})$$

Hence, if $b_i,\ d_i \in \mathbb{Z}$ for all i, $s = 0$, and $d_s = \pm 1$ then $c_i \in \mathbb{Z}$. (This is in fact the case with the Euler numbers.)

Now if $s = 0$, and $d_0 \neq \pm 1$ and $d_0 \in \mathbb{Z}$, then instead calculate $c_i^* = c_i d_0^i$. Notice that:

$$\begin{aligned} d_0^i c_i &= \frac{d_0^i}{d_0}(b_i - \sum_{j=1}^{i} \binom{i}{j} d_j c_{i-j}) \\ c_i^* &= (d_0^{i-1} b_i - \sum_{j=1}^{i} \binom{i}{j} d_0^{i-1} d_j c_{i-j}) \\ c_i^* &= (d_0^{i-1} b_i - \sum_{j=1}^{i} \binom{i}{j} d_0^{j-1} d_j (d_0^{i-j} c_{i-j})) \end{aligned}$$

$$c_i^* = (d_0^{i-1} b_i - \sum_{j=1}^{i} \binom{i}{j} d_0^j d_j c_{i-j}^*).$$

which will remain in the integers.

Further, if b_i and d_i come from functions $s(x)$ and $t(x)$, both of which satisfy all of the conditions of Lemma 4.5, namely that $P^s(x)$, $P^t(x) \in \mathbb{Q}[x]$, where $s(x)$, $t(x) \in \mathcal{P}^{\mathbb{C},\mathbb{Q}}$, then by the c_n^* can be altered so that all the calculations are still done over the integers.

Here take e_b and f_b as the d and c in the proof of Lemma 4.5, as it applies to b_i, and set $\bar{e}_i = b_i e_b^i f_b$. Similarly set $\bar{d}_i = d_i e_d^i f_d$, where e_d and f_d have similar definitions. Further assume that $f_d = 1$.

So now consider calculating $\bar{c}_i = c_i^* \mathrm{lcm}(e_b, e_d)^n \mathrm{lcm}(f_b, f_d)$ For ease of notation, denote $e = \mathrm{lcm}(e_b, e_d)$ and f similarly. For ease of notation, denote $\bar{e}_b = \frac{e}{e_d}$, and define $\bar{e}_d$, $\bar{f}_b$ and $\bar{f}_d$ similarly.

Then:

$$\begin{aligned}
e^i f c_i^* &= e^i f (d_0^i b_i - \sum_{j=1}^{i} \binom{i}{j} d_0^j d_j c_{i-j}^*) \\
\bar{c}_i &= (d_0^i e^i f b_i - \sum_{j=1}^{i} \binom{i}{j} d_0^j e^i f d_j c_{i-j}^*) \\
\bar{c}_i &= (d_0^i (\bar{e}_b)^i \bar{f}_b \bar{b}_i - \sum_{j=1}^{i} \binom{i}{j} d_0^j e^j d_j f e^{i-j} c_{i-j}^*) \\
\bar{c}_i &= (d_0^i (\bar{e}_b)^i \bar{f}_b \bar{b}_i - \sum_{j=1}^{i} \binom{i}{j} d_0^j (\bar{e}_d)^j \bar{d}_j \bar{c}_{i-j}).
\end{aligned}$$

Where finally everything is calculated over the integers.

■

Corollary 10 *The Euler numbers and the Genocchi numbers are integers. Moreover the recursion formula and lacunary recursion formula used to compute the Euler and Genocchi numbers are also over the integers.*

5.6 Techniques for smaller linear recurrence relations.

As before, in Section 4.7, polynomials can be factored from a poly-exponential function, to make the linear recurrence relations easier to solve. Write $t(x) = p(x)\bar{t}(x)$, the denominator of some

rational poly-exponential function, for $t(x)$, $\bar{t}(x) \in \mathcal{P}$ and $p(x)$ a polynomials. Then notice, that for calculating the denominator, then a factor of $\prod_{i=0}^{m-1} p(x\omega_m^i)$ can be pulled out.

A similar process for the top linear recurrence relation can be done, but some extra care need be taken.

Example 32 *Consider the following example in Maple.*

```
> \mapleinline{active}{1d}{with(MS):}{%
> }
```

This example looks at the Bernoulli numbers. But for this example, modify the equation, so that it can be demonstrated how common factors of polynomials can be taken out. So examine $\frac{x^2+x}{x\,e^x-x+e^x-1} = \frac{\sum_{i=0}^{\infty} \frac{b_i\,x^i}{i!}}{\sum_{j=0}^{\infty} \frac{d_j\,x^j}{j!}}$. *Now multisection this by 4 at 2.*

So the bottom can be

$\prod_{i=0}^{3} (x\,{\omega_4}^i + 1)\,(e^{(x\,{\omega_4}^i)} - 1) = (\prod_{i=0}^{3} (x\,{\omega_4}^i - 1))\,(\prod_{i=0}^{3} (e^{(x\,{\omega_4}^i)} - 1))$. *So there is a polynomial that can be factored out. After this simply work out the normal linear recurrence relation for the bottom. This could have done automatically by:*

```
> \mapleinline{active}{1d}{`bottom/ms/factor`((x+1)*(exp(x)-1),f,x,4);}{
> %
> }
```

$$\mathrm{f}(x) = 4\,\mathrm{f}(x-8) - 3\,\mathrm{f}(x-4),\ f,\ x,\ [\mathrm{f}(0) = 0, \mathrm{f}(1) = 0, \mathrm{f}(2) = 0, \mathrm{f}(3) = 0, \mathrm{f}(4) = -24, \mathrm{f}(5) = 0, \mathrm{f}(6) = 0, \mathrm{f}(7) = 0, \mathrm{f}(8) = 56],\ -x^4 + 1$$

Where the last value is the polynomial that is pulled out.

The top can be similarly manipulated so as to get the common polynomial to be pulled out.

```
> \mapleinline{active}{1d}{`top/ms/factor`(x^2+x,
> (x+1)*(exp(x)-1),f,x,4,2);}{%
> }
```

$$\mathrm{f}(x) = 4\,\mathrm{f}(x-2) - 3\,\mathrm{f}(x-1),\ f,\ x,\ [\mathrm{f}(0) = 0, \mathrm{f}(1) = 0, \mathrm{f}(2) = 0, \mathrm{f}(3) = 0, \mathrm{f}(4) = 0, \mathrm{f}(5) = -10, \mathrm{f}(6) = 0, \mathrm{f}(7) = 0, \mathrm{f}(8) = 0, \mathrm{f}(9) = 30, \mathrm{f}(10) = 0, \mathrm{f}(11) = 0, \mathrm{f}(12) = 0, \mathrm{f}(13) = -130, \mathrm{f}(14) = 0, \mathrm{f}(15) = 0, \mathrm{f}(16) = 0, \mathrm{f}(17) = 510, \mathrm{f}(18) = 0, \mathrm{f}(19) = 0, \mathrm{f}(20) = 0, \mathrm{f}(21) = -2050, \mathrm{f}(22) = 0, \mathrm{f}(23) = 0, \mathrm{f}(24) = 0, \mathrm{f}(25) = 8190, \mathrm{f}(26) = 0, \mathrm{f}(27) = 0, \mathrm{f}(28) = 0, \mathrm{f}(29) = -32770, \mathrm{f}(30) = 0, \mathrm{f}(31) = 0],\ (x - I)\,(x - 1)\,(x + I)\,x\,(x + 1)$$

5.7 Conclusions.

The conclusion that are listed in this section are conclusions as to which implemenations are faster, the conclusions are not for which methods are faster. This is because Maple combines a relatively sophisticate code to deal with certain problems, and some very naive methods for others. Hence the implementation of any method in this chapter can be greatly impacted on by the underlying methods used by Maple for certain problems, (for examples, solving linear systems of equations, how it performs resultants, etc).

5.7.1 Denominator.

The different methods that are possible for determining the bottom linear recurrence relation of a multisectioned rational poly-exponential function are:

1. naive method, (Chapter 3, Lemma 3.1),
2. the recurrence polynomial with resultants (Section 5.1),
3. linear algebra, with symbolic differentiation (Chapter 4, Section 4.3),
4. linear algebra, fast Fourier transform method 1, (Subsection 5.2.1),
5. linear algebra, fast Fourier transform method 2, (Subsection 5.2.2),
6. looking at symmetries of the denominator, (Section 5.4),
7. computing over the integers, (Section 5.5),
8. factoring polynomials out, in combination with any of the above, (Section 5.6).

- Here, the use of some knowledge (of how large the linear recurrence relation will be) is of great use to method 3 and 4. For example, without this knowledge, trying to determine the bottom linear recurrence relation of the Euler numbers when multisectioned by 8 takes over 60 seconds and 10.65 for methods 3 and 4 respectively, where as with this knowledge this take 4.58 and 3.86 seconds.
- The naive method, method 1, although the easiest to implement, is not very efficient taking 11 seconds to do this problem, whereas method 2 and 5 take 2.72 seconds and 1.42 seconds respectively.
- If the same problem is looked at, but multisectioning by 9 instead of by 8, then of all the methods from 1 to 5, with the exception of method 5, take too long to be practical (even with knowledge).

- Method 5 takes about 126.9 seconds.
- By taking into account a symmetry (method 6) of order p, the existing methods can be expected to be able to multisection by a factor of p more. For example, with the Euler numbers, instead of having a upper bound of 12 for multisectioning, an upper bound of about 24 is achieved. (The Euler numbers have a symmetry of order 2 in the denominator.)
- Methods 7 and 8 are of little interest, as rarely do functions meet the criteria that would be required for these methods to be of use.
- (These times were done on "bb" (2 180 MHZ IP27 Processors, Main memory size, 256 Mbytes), using the Maple interpretation of a CPU second.)

5.7.2 Numerator.

The different methods that are possible for determining the top linear recurrence relation of a multisectioned rational poly-exponential function are:

1. naive method, (Chapter 3, Lemma 3.1),
2. the recurrence polynomial and resultants (Section 5.1),
3. linear algebra with symbolic differentiation, (Chapter 4, Section 4.3),
4. linear algebra, fast Fourier transform, (Subsection 5.2),
5. factoring polynomials out, in combination with any of the above, (Section 5.6),
6. using information about the bottom linear recurrence relation. (Section 5.3).

- Again the problem of the Euler numbers was looked at - trying to determine the top linear recurrence relation.
- An examination of the times gives that method 6 is by far the best.
- When multisectioning by 8 at 2, the other methods, in order take;
  - with method 1, 201.733 seconds,
  - with method 2, over 1000 seconds,
  - with method 3, over 1000 seconds,
  - with method 3, 55.62 (with knowledge),
  - with method 4, over 1000 seconds, and

  - with method 4, 494.15 (with knowledge).

- This is in comparison to method 6, which took only 30.467 seconds.
- If the denominator had a symmetry of order p, then it becomes possible to multisection by a factor of p more. For example, instead of having an upper bound of multisectioning by 12 for the Euler numbers, the upper bound becomes 24. (The Euler numbers have a symmetry of order 2 in the denominator.)
- (These times were done on "bb" (2 180 MHZ IP27 Processors, Main memory size, 256 Mbytes), using the Maple interpretation of a CPU second.)

Chapter 6

Doing the calculation.

When doing calculations, there are numerous things that can be done at the programming level to speed up the calculations. The first two sections, Sections 6.1 and 6.2 talk about methods where concurrence is exploited. The third section, Section 6.3 discusses the largest problems at the time of submission of this thesis that these techniques have been used for. The last section, Section 6.4 discusses some methods of validating the correctness of the results.

The methods in this thesis so far have allowed the calculation of terms of rational poly-exponential functions to be run on m different machines by multisectioning by m. After the problem is divided up by multisectioning, to m different computers, no communication is needed between these computers. The method of multisectioning is limited by the size m, as multisectioning by large m quickly becomes impractical. After multisectioning by m, the computation can only be done on at most m different machines.

This does not mean though that only m different processors can be used. By allowing communication between processors, the problem can be broken up further. The basis of this idea is that to calculate the k-th number, the previous $k-1$ numbers are needed, but not all of them need to be known when the computation is started. When calculating the k-th number, have n other processors working out the $k-1$, $k-2$, ..., $k-n$ numbers. So long as this information is available by the end of the computation there is no problem. Many of the techniques for concurrency used here are described in Snow, [27].

There are two different techniques described here. The first as described in Section 6.1 is in the case with n processors, where all the processors are the same speed (i.e. a dedicated multi-processor machine). This type of problem does not need to worry about load balancing.

The second case, as described in Section 6.2 is that with multiple CPU's, not all of which are

the same speed (i.e. a cluster of PCs with different clock speeds). To properly take advantage of the CPUs to their maximum efficiency, more complicated code need be written that will attempt to balance the load. Failing to do this will lead to a computation on n CPU's that is only n times faster than the slowest processor.

6.1 Load balanced code.

6.1.1 Overview.

Assume there are n processors, all of which are the same speed, and the calculations are of well-distributed difficulty (as is the case with rational poly-exponential function), then give every n-th problem to each CPU. At the end of each calculation, the results are communicated to the other processors.

For this problem, the master/slave paradigm is used, as it reduces the number of communication channels that are required. The "*process*" package in Maple was used, which utilized the Unix commands of fork, pipe, wait, block, etc. As a result in implementing this, and preparing the worksheets, numerous bugs in the "*process*" package in Maple were found. For more information see Appendix D Sections D.3, D.4, and D.5.

6.1.2 Details of algorithm.

Assume the program is run with n slaves. Using the master/slave paradigm, have the master tell the slave which calculation to start with, and how large an increment to use. So slave 1 is told to calculate b_1, b_{1+n}, b_{1+2n}, ..., up to some maximum, slave 2 will calculate b_2, b_{2+n}, ..., etc. The slave, when it has done a calculation will tell the master. The master then passes this information on to all of the other $n-1$ slaves.

When the slave needs information, it simply waits for the master to provide this information. This is one of the reasons why in this model it is very important that the slaves are the same speed. If one slaves is slower than the other slaves, then all of these slaves will constantly be waiting for this one slave to complete its calculation before they can continue.

This is summarized below in Figure 6.1.

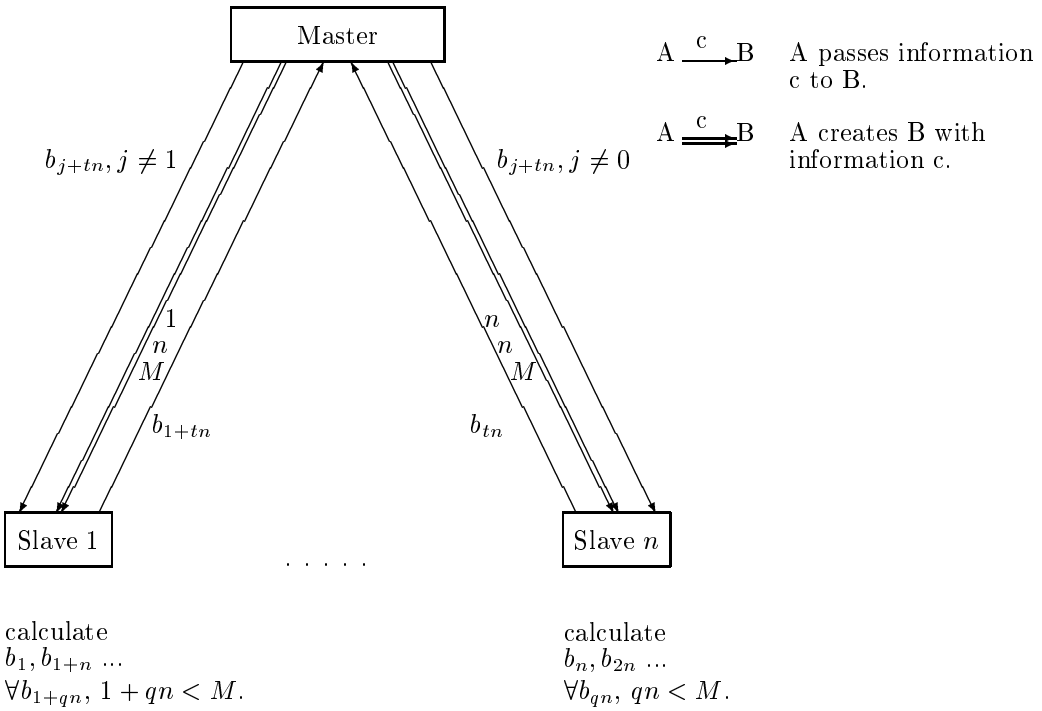


Figure 6.1: Load balanced master/slave diagram.

For more information, see Appendix A, Subsection A.7.2.

Example 33 *Consider the problem of calculating the Genocchi numbers, defined by the exponential generating function* $\frac{2x}{e^x+1}$. *For more information about the Maple code, see Appendix A. For the Maple code see Appendix E. The Maple code and help files (including information about syntax) are available on the web at [1]. For this, consider the calculation given that the recursion formula is multisectioned by 2 at 0. Further assume that there are two slaves (i.e. a 2 CPU machine).*

```
    |\^/|     Maple V Release 5 (Simon Fraser University)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> with(MS): with(process): readlib(‘calcul/balanced/worker‘):
>
> bot := ‘bottom/ms/linalg/fft2‘(exp(x)+1,f,x,2);
```

```
bytes used=1007116, alloc=851812, time=0.24
          bot := f(x) = f(x - 2), f, x, [f(0) = 4, f(1) = 0, f(2) = 2]

> Bot := `egf/makeproc`(bot):
> top := `top/ms/linalg/fft`(2*x, exp(x)+1, f, x, 2, 0);
top := f(x) = -f(x - 4) + 2 f(x - 2), f, x,

    [f(0) = 0, f(1) = 0, f(2) = -4, f(3) = 0]

> Top := `egf/makeproc`(top):
>
# Increase the information presented, so as to demonstrate how
# the slaves and the master interact with each other.
>
> infolevel[MS] := 4;
                                infolevel[MS] := 4

>
> B := `calcul/balanced`(2, 10, Top, Bot, 2, 0): seq(B[2*i], i=0..5);
calcul/balanced:   "Starting up slave"   0
calcul/balanced/worker:   "Slave"   0   "working on problem"   0
calcul/balanced/worker:   "Slave"   0   "getting needed info from Master"
calcul/balanced/worker:   "Slave"   0   "finishing calculation"
calcul/balanced:   "Starting up slave"   2
calcul/balanced/worker:   "Slave"   0   "Reporting to Master"
calcul/balanced/worker:   "Slave"   2   "working on problem"   2
calcul/balanced/worker:   "Slave"   2   "getting needed info from Master"
calcul/balanced:   "Getting information from slave"   0
calcul/balanced/worker:   "Slave"   0   "working on problem"   4
calcul/balanced/worker:   "Slave"   0   "getting needed info from Master"
calcul/balanced:   "Sending info to slave"   2
calcul/balanced:   "Getting information from slave"   2
calcul/balanced/worker:   "Slave"   2   "finishing calculation"
calcul/balanced/worker:   "Slave"   2   "Reporting to Master"
calcul/balanced/worker:   "Slave"   2   "working on problem"   6
calcul/balanced/worker:   "Slave"   2   "getting needed info from Master"
calcul/balanced:   "Sending info to slave"   0
```

```
calcul/balanced:   "Getting information from slave"   0
calcul/balanced/worker:   "Slave"   0   "finishing calculation"
calcul/balanced/worker:   "Slave"   0   "Reporting to Master"
calcul/balanced/worker:   "Slave"   0   "working on problem"   8
calcul/balanced:   "Sending info to slave"   2
calcul/balanced/worker:   "Slave"   0   "getting needed info from Master"
calcul/balanced/worker:   "Slave"   2   "finishing calculation"
calcul/balanced/worker:   "Slave"   2   "Reporting to Master"
calcul/balanced/worker:   "Slave"   2   "working on problem"   10
calcul/balanced/worker:   "Slave"   2   "getting needed info from Master"
calcul/balanced:   "Getting information from slave"   2
calcul/balanced:   "Sending info to slave"   0
calcul/balanced:   "Getting information from slave"   0
calcul/balanced/worker:   "Slave"   0   "finishing calculation"
calcul/balanced/worker:   "Slave"   0   "Reporting to Master"
calcul/balanced:   "Sending info to slave"   2
calcul/balanced/worker:   "Slave"   2   "finishing calculation"
calcul/balanced/worker:   "Slave"   2   "Reporting to Master"
calcul/balanced:   "Getting information from slave"   2
calcul/balanced:   "Sending info to slave"   0
calcul/balanced:   "Stopping slave"   0
bytes used=1964100, alloc=1441528, time=0.01
calcul/balanced:   "Stopping slave"   2
bytes used=1966624, alloc=1441528, time=0.02
                            0, -1, 1, -3, 17, -155

> quit
bytes used=1969268, alloc=1441528, time=0.51
```

6.2 Load balancing code.

6.2.1 Overview.

If the system does not have balanced CPU power, then the code must balance the load.

Again this method uses the master/slave paradigm, although refinements to this have been made which will be discussed later. Say at some time in the calculation there are k processes running to

calculate b_n, b_{n+1}, ..., b_{n+k}. If on the computation $n+s$, $(1 \leq s \leq k)$, the processor can do no more calculations until the information of the value of b_n is provided to it. Instead of waiting (as would have been done in Section 6.1), this process will ask for more work. It will then start calculating b_{n+k+1}, and will get back to the calculations of b_{n+s} when the necessary information is available.

For technical reasons it was decided to have an intermediate process, the overseer, between the master and the slave. This overseer's job is to provide communication between the master and the slave, as well as deciding when a slave can no longer continue working (as the information needed is not available yet), and start a new calculation.

6.2.2 Details of algorithm.

There is one overseer per machine, and one master.

The master will wait until it receives a "need work" message from an overseer. At this point, the master will send the overseer an index of something to be computed.

The overseer will first delegate the work to some slave (if creating the slave, the overseer will also tell the slave everything that the overseer knows).

The slave upon creation/call will start its calculation of the index i given to it. If the slave gets to a point where it needs more information, it will ask the overseer. Upon completion, it will send back the calculation to the overseer and await new work.

The overseer, when it gets a request for information from a slave, will send the information, if it is known. If the information is not known then the overseer will send a message to the master asking for more work. The overseer will keep track that this slave is waiting for this information, and when the overseer acquires this information, it will provide this information to the slave. When the overseer receives the result of a calculation, it will send the result of this calculation to the master. The overseer will ask for work if it has no slaves working (slaves get in each other's way).

The overseer will constantly be waiting for information from the master. The master, when it has a new calculation, will send the information to the other overseers.

This is summarized below in Figure 6.2.

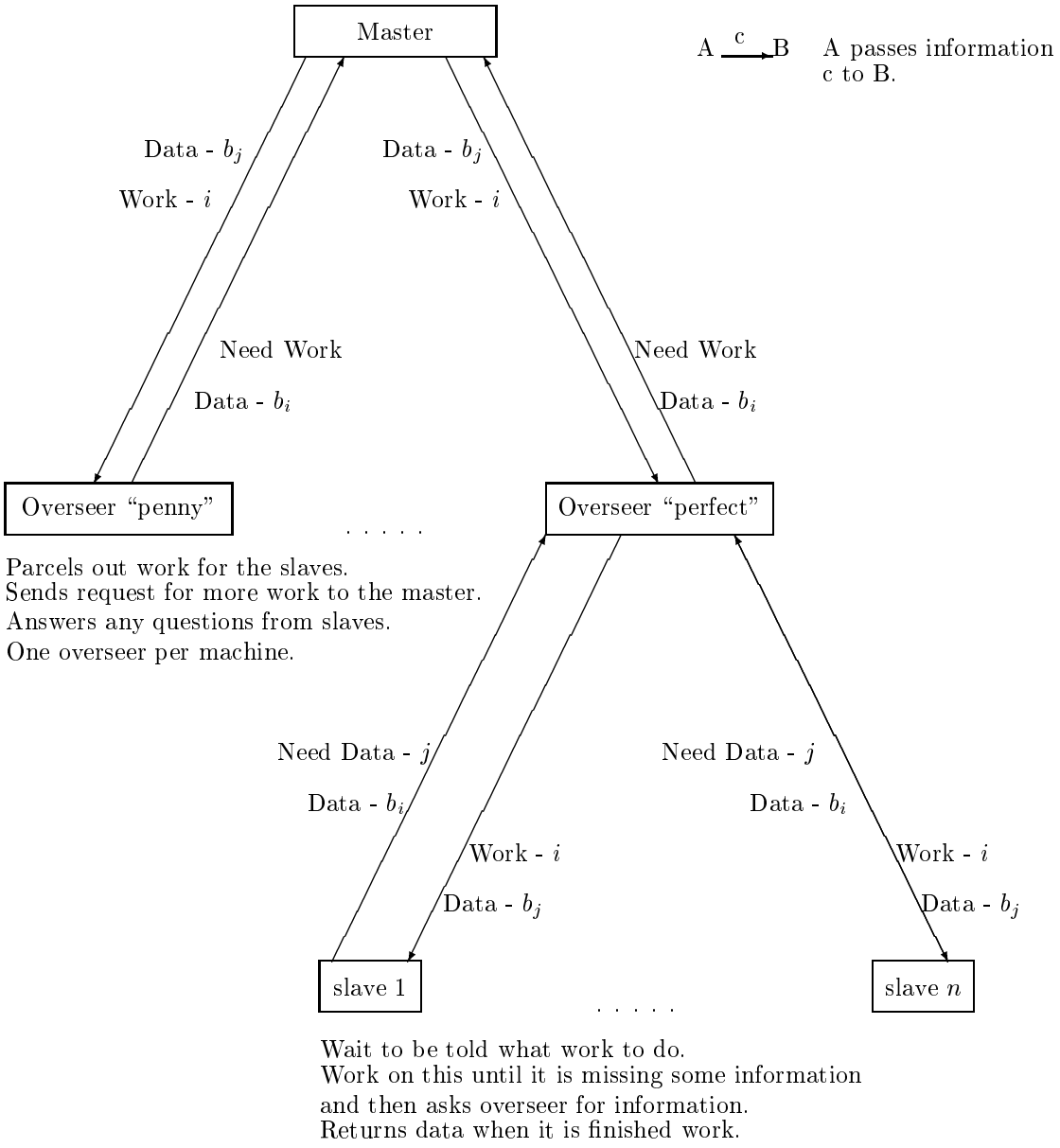


Figure 6.2: Load balancing master/overseer/slave diagram.

Example 34 *Consider the following example. The first part is the master, which shows what the*

master is asking the overseer to do. The second and third parts are the two overseers, which demonstrates their side of the conversation.

1. *The master,*

```
    |\^/|     Maple V Release 5 (Simon Fraser University)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> with(MS): with(process):
> Info[0] := 1:
> infolevel[MS] := 2:
> A := `calcul/balancing/master`(bb, [perfect, penny], 10, 2, 2,
>           Euler, 125, Info):
calcul/balancing/master:   "Working on requested for work from perfect"
calcul/balancing/master:   "Tell perfect to work on the value of 2"
calcul/balancing/master:   "Working on requested for work from penny"
calcul/balancing/master:   "Tell penny to work on the value of 4"
calcul/balancing/master:   "Working on requested for work from perfect"
calcul/balancing/master:   "Tell perfect to work on the value of 6"
calcul/balancing/master:   "Working on requested for work from penny"
calcul/balancing/master:   "Tell penny to work on the value of 8"
calcul/balancing/master:   "Got some data for the value of 2 from perfect"
calcul/balancing/master:   "Got some data for the value of 8 from penny"
calcul/balancing/master:   "Working on requested for work from perfect"
calcul/balancing/master:   "Tell perfect to work on the value of 10"
calcul/balancing/master:   "Working on requested for work from penny"
calcul/balancing/master:   "Tell penny to quit"
calcul/balancing/master:   "Got some data for the value of 4 from penny"
calcul/balancing/master:   "Working on requested for work from penny"
calcul/balancing/master:   "Tell penny to quit"
calcul/balancing/master:   "Got some data for the value of 6 from perfect"
calcul/balancing/master:   "Got some data for the value of 10 from perfect"
calcul/balancing/master:   "Telling perfect to quit"
calcul/balancing/master:   "Telling penny to quit"
>
> seq(A[i],i=0..10);
```

```
          1, A[1], -1, A[3], 5, A[5], -61, A[7], 1385, A[9], -50521

> quit
bytes used=420460, alloc=393144, time=0.12
```

2. *Overseer perfect,*

```
    |\^/|     Maple V Release 5 (Simon Fraser University)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> with(MS): with(process): readlib(`process/block`):
> readlib(`calcul/writepipe`):
>
> Info[0] := 1:
> Top := `egf/makeproc`(`top/ms/linalg/fft`(2,exp(x)+exp(-x),f,x,2,0)):
> Bot := `egf/makeproc`(`bottom/ms/linalg/fft2`(exp(x)+exp(-x),f,x,2)):
bytes used=1292572, alloc=1048384, time=0.35
>
> infolevel[MS] := 4:
>
> `calcul/balancing/overseer`(bb, perfect, Top, Bot, 2, 0, Info, 1, 1);
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 0 slaves 0 running 0 waiting and the message is Work"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Told to do work on 2 from 0"
calcul/balancing/slave:   "Slave 1 is waiting for instructions"
calcul/balancing/slave:   "Slave 1 is working on determining the value for 2"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Telling the overseer about the new value for 2"
calcul/balancing/slave:   "Slave 1 is waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Work"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Told to do work on 6 from 0"
calcul/balancing/overseer:   "Waiting for instructions"
```

```
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Given some new data 2 from 1"
calcul/balancing/overseer:   "Slave"   1   "is no longer working, "
"so give it outstanding work"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Slave 1 is working on determining the value for 6"
calcul/balancing/slave:   "Asking for data of "   2
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Need Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Asked for data"   2   "from"   1
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Got some data 2 from 1"
calcul/balancing/slave:   "Asking for data of "   4
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Need Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Asked for data"   4   "from"   1
calcul/balancing/overseer:   "Doesn't know the info"   4   "for"   1
calcul/balancing/overseer:   "Waiting for instructions"
bytes used=2293024, alloc=1703624, time=1.04
calcul/balancing/overseer:
"Has 1 slaves 1 running 1 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Given some new data 8 from 0"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 1 running 1 waiting and the message is Work"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Told to do work on 10 from 0"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 1 running 1 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Given some new data 4 from 0"
```

```
calcul/balancing/overseer:   "Telling waiting slave 1 about this data"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Got some data 4 from 1"
calcul/balancing/slave:   "Telling the overseer about the new value for 6"
calcul/balancing/slave:   "Slave 1 is waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Given some new data 6 from 1"
calcul/balancing/overseer:   "Slave"   1   "is no longer working, "
"so give it outstanding work"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Slave 1 is working on determining the value for
10"
calcul/balancing/slave:   "Asking for data of "   6
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Need Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Asked for data"   6   "from"   1
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Got some data 6 from 1"
calcul/balancing/slave:   "Asking for data of "   8
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Need Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Asked for data"   8   "from"   1
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Got some data 8 from 1"
calcul/balancing/slave:   "Telling the overseer about the new value for 10"
calcul/balancing/slave:   "Slave 1 is waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Given some new data 10 from 1"
calcul/balancing/overseer:   "Slave 1 is no longer working"
calcul/balancing/overseer:   "Ask for more work"
calcul/balancing/overseer:   "Waiting for instructions"
```

```
calcul/balancing/overseer:
"Has 1 slaves 0 running 0 waiting and the message is Quit"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Telling the 1th slaves to quit"
calcul/balancing/slave:   "Slave Quitting"   1
bytes used=2248948, alloc=1703624, time=0.02
calcul/balancing/overseer:   "The 1th slave has quit"
calcul/balancing/overseer:   "Everyones quit, time to go home"
> quit
bytes used=2600132, alloc=1703624, time=1.30
```

3. *Overseer penny,*

```
    |\^/|     Maple V Release 5 (Simon Fraser University)
._|\|   |/|_. Copyright (c) 1981-1997 by Waterloo Maple Inc. All rights
 \  MAPLE  /  reserved. Maple and Maple V are registered trademarks of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> with(MS): with(process): readlib(`process/block`):
> readlib(`calcul/writepipe`):
>
> Info[0] := 1:
> Top := `egf/makeproc`(`top/ms/linalg/fft`(2,exp(x)+exp(-x),f,x,2,0)):
> Bot := `egf/makeproc`(`bottom/ms/linalg/fft2`(exp(x)+exp(-x),f,x,2)):
bytes used=1292572, alloc=1048384, time=0.33
>
> infolevel[MS] := 4:
>
> `calcul/balancing/overseer`(bb, penny, Top, Bot, 2, 0, Info, 1, 1);
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 0 slaves 0 running 0 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Given some new data 8 from 0"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 0 slaves 0 running 0 waiting and the message is Work"
calcul/balancing/overseer:   "Got info from slave/master 0"
```

```
calcul/balancing/overseer:   "Told to do work on 4 from 0"
calcul/balancing/slave:   "Slave 1 is waiting for instructions"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Slave 1 is working on determining the value for 4"
calcul/balancing/slave:   "Asking for data of "   2
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Need Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Asked for data"   2   "from"   1
calcul/balancing/overseer:   "Doesn't know the info"   2   "for"   1
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 1 running 1 waiting and the message is Work"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Told to do work on 8 from 0"
calcul/balancing/overseer:   "Already know the info"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 1 running 1 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 0"
calcul/balancing/overseer:   "Given some new data 2 from 0"
calcul/balancing/overseer:   "Telling waiting slave 1 about this data"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/slave:   "Got some data 2 from 1"
calcul/balancing/slave:   "Telling the overseer about the new value for 4"
calcul/balancing/slave:   "Slave 1 is waiting for instructions"
bytes used=2292796, alloc=1572576, time=0.84
calcul/balancing/overseer:
"Has 1 slaves 1 running 0 waiting and the message is Data"
calcul/balancing/overseer:   "Got info from slave/master 1"
calcul/balancing/overseer:   "Given some new data 4 from 1"
calcul/balancing/overseer:   "Slave 1 is no longer working"
calcul/balancing/overseer:   "Ask for more work"
calcul/balancing/overseer:   "Waiting for instructions"
calcul/balancing/overseer:
"Has 1 slaves 0 running 0 waiting and the message is Quit"
calcul/balancing/overseer:   "Got info from slave/master 0"
```

```
calcul/balancing/overseer:   "Telling the 1th slaves to quit"
calcul/balancing/slave:   "Slave Quitting"   1
bytes used=2210520, alloc=1507052, time=0.01
calcul/balancing/overseer:   "The 1th slave has quit"
calcul/balancing/overseer:   "Everyones quit, time to go home"
> quit
bytes used=2346676, alloc=1572576, time=0.89
```

6.3 A large calculation.

As of submitting this thesis, the following upper bounds of calculations have been completed, as shown in the Table 6.1. These calculations are available on the web at [1].

	Bernoulli numbers	Euler numbers	Genocchi numbers	Lucas numbers type II
Bottom recurrence	20	24	20	20
Top recurrence	18	16	20	14
Largest number	35 298	8 500	8 700	5 404

Table 6.1: Upper bounds of completed calculations.

The typical bottle neck for a calculation is with the linear algebra. If a proper Toeplitz matrix solver were used, one would predict that the time to perform a calculation would be much improved. For example, to calculate the denominator of the Bernoulli numbers, multisectioned by 20, it requires only 7 minutes 15 seconds to determine the underlying matrix; the rest of the 2.6 days is to find the solution associated with this 90×90 matrix. (The time here represents a CPU second as measured by Maple on "penny", CPU: MIPS R10000 Processor Chip Revision: 2.7.)

Similarly, when multisectioning the numerator of the Bernoulli numbers by 18 it takes 69.6 seconds to determine the underlying 24×24 matrix and the remained of the 116.35 seconds to solve this linear algebra problem. (The time here represents a CPU second as measured by Maple on "pecos", CPU: MIPS R10000 Processor Chip Revision: 2.7.)

Next consider a large calculation of the Bernoulli numbers, say the first 1 800 Bernoulli numbers. It takes 30.56 seconds to perform this calculation, using recurrences that have been multisectioned by 18. (Hence only $\frac{1}{9}$-th of the information is calculated.) In contrast, the normal recurrence (which by the nature of the Bernoulli numbers is multisectioned by 2) takes 527.61 seconds. Thus there is a speed up of a factor of $\frac{527.61}{30.56 \times 9} = 1.92$ by multisectioning by 18. (Here, the extra factor of 9 comes

in because one would have to perform 9 different calculations to get all of the information using the multisectioned method.) This demonstrates that these multisectioned recursion formulae, even when used in serial environment upon a single computer, represent a significant speed up over the traditional recursion formula.

If the multi-processor method described in Section 6.1 is used, with 5 slaves, with the recurrences that has been multisectioned by 18, then it takes on average 6.20 seconds for each slave. (The master takes an insignificant amount of processor time; taking less than half a second.) So the total processor time is bounded above by 31.5 seconds. This indicates that about 3% of the processors time, when using a multi-processor method, goes towards the overhead of communication. (In actual fact, this is too high an estimate when doing a large calculation, but relatively little numerical data is available at this time.) So these calculations can advantageously exploit parallel computing techniques. (The time here represents a CPU second as measured by Maple on "manyjars", 8 250 MHZ IP27 Processors CPU: MIPS R10000 Processor Chip Revision: 3.4.)

6.4 Validating results.

When doing large calculations such as these, some methods to test if the calculations are done correctly are needed, both for confidence and as a useful aid to debugging.

6.4.1 Validating the Bernoulli numbers.

To test if the calculation for the Bernoulli numbers is done correctly, the following theorem of von Staudt [17] is used.

Theorem 6.1 (Clausen - von Staudt Theorem) *Let B_{2k} be the $2k$-th Bernoulli number. If $k \geq 1$, then*

$$(-1)^k B_{2k} \equiv \sum \frac{1}{p} \pmod 1$$

the summation being extended over the primes p such that $(p-1)|2k$.

From which it follows that:

Corollary 11 *If $k \geq 1$, then the denominator of $(-1)^k B_{2k}$, where B_{2k} is the $2k$-th Bernoulli number is equal to the denominator of $\sum \frac{1}{p}$ the summation being extended over the primes p such that $(p-1)|2k$.*

Example 35 *Thus, to test if the 10 008-th Bernoulli number, calculated as*

$$\frac{N}{3262901044146573454170}$$

where N is a 27716 digit number, is correct, the denominator need only be checked.

Calculate $(-1)^k \sum \frac{1}{p}$ for $(p-1)|2k$ where $2k = 10008$ yields:

$$\frac{4402843531608629672099}{3262901044146573454170}$$

Noticing that the denominator of these two numbers is the same is a good indication that the calculation was done correctly.

6.4.2 Validating the Euler numbers.

To test if the calculation for the Euler numbers is done correctly, the following theorem of Glaisher [14] is used.

Theorem 6.2 *Let E_{2k} be the $2k$-th Euler number. For $k > 0$, and any $r > 0$:*

$$E_{2k} \equiv (-1)^k 2[1^{2k} - 3^{2k} + 5^{2k} - .... + (-1)^{1/2(r-2)}(r-2)^{2k}] \pmod{r}.$$

Combining this with Fermat's little theorem gives that:

Theorem 6.3 *Let p be prime. If $2k \equiv 2j \pmod{p-1}$ and E_{2k}, E_{2j} the $2k$-th and $2j$-th Euler numbers respectively then*

$$E_{2k} \equiv E_{2j} \pmod{p}.$$

Example 36 *Thus, to test if the 8 000-th Euler number, calculated as N where N is a 26 184 digit number, is correct, look at N modulo a number of small primes.*

$$\begin{aligned} N &\equiv 2 \pmod{3}, \\ N &\equiv 0 \pmod{5}, \\ N &\equiv 6 \pmod{7}, \\ N &\equiv 2 \pmod{11}, \\ N &\equiv 7 \pmod{13}, \\ N &\equiv 0 \pmod{17}. \end{aligned}$$

Notice that

$$\begin{aligned}
8000 &\equiv 2 \pmod 2 \text{ and } E_2 \equiv 2 \pmod 3,\\
8000 &\equiv 4 \pmod 4 \text{ and } E_4 \equiv 0 \pmod 5,\\
8000 &\equiv 2 \pmod 6 \text{ and } E_2 \equiv 6 \pmod 7,\\
8000 &\equiv 10 \pmod{10} \text{ and } E_{10} \equiv 2 \pmod{11},\\
8000 &\equiv 8 \pmod{12} \text{ and } E_8 \equiv 7 \pmod{13},\\
8000 &\equiv 16 \pmod{16} \text{ and } E_{16} \equiv 0 \pmod{17}.
\end{aligned}$$

Thus N has the correct residues to be the 8 000-th Euler number, and it passes the test.

Chapter 7

Conclusion.

This thesis highlights the complex issues that arise when working in an environment, such as Maple, where the code is not all written by the principle author, or to an agreed standard. One problem in such a system is the necessary reliance on a mixture of code, some of which is very sophisticated, some of which is more naive, some of which is written for a very general problem, and some of which has been tailored to a specific problem. Hence the caveat in Sections 4.8 and 5.7 that the conclusions therein were as to which implementation was fastest, and not to which method was fastest. Another problem is in the debugging of code, where the underlying problem being tracked down in the debugging process might not be within the code written, but instead in the system being used. This could be either an incompatibility of the different functions within the system, a misuse of an algorithm being offered by the system, or an actual problem with the algorithm within the system. Hence the inclusion of Appendix D for bugs or weakness found in Maple.

Some of the achievements of this thesis include implementations of algorithms to multisection rational poly-exponential functions. The new recursion formulae, that these algorithms yield, represent an improvement over the traditional methods of computing Bernoulli numbers, Euler numbers, and other rational poly-exponential functions. Traditionally multisectioning has been looked at in the narrow setting to its use in calculating Bernoulli numbers and Euler numbers. Here, the investigation was done in a more general setting; allowing a wider applicability of the multisectioning process.

Appendix A

Outline of code.

This code can be found on my homepage [1]. It can also be found in Appendix E.

The appendix is laid into five sections. The first section will look at code for manipulating poly-exponential functions. Section A.2 will look at code for manipulating exponential generating functions. Section A.3 looks at the code to determine the metrics of different poly-exponential functions. Section A.4 looks at the code to convert poly-exponential functions to exponential generating functions and back, as well as code to convert linear recurrence relation to the recurrence polynomial and back. Then Section A.5 will look at code for manipulating the bottom linear recurrence relation of a rational poly-exponential function. After which Section A.6 will look at code for manipulating the top linear recurrence relation of a rational poly-exponential function. Lastly Section A.7 will deal with code to do the calculation, after the linear recurrence relations are know.

Within each section, a brief description of a piece of code, the command name, file where it can be found, which example in the thesis demonstrates how it is used with a page reference, the expected input and output of the command, and a reference to which theorems or definitions it automates.

A.1 Code for poly-exponential functions.

A.1.1 Naive method.

This will take a poly-exponential function and multisection it using the naive method, using the definition of multisectioning as given in Definition 2.6.

- file: Pe,

- command: ‘pe/ms/naive‘,
- examples: Example 5 pp. 17,
- input: exponential generating function, m,
- output: exponential generating function multisectioned by m,
- reference: Lemma 2.1, Definition 2.6 and Theorem 2.1.

A.1.2 Linear algebra and symbolic differentiation method.

This method will take a poly-exponential function and multisection it by using symbolic differentiation after which point the method will use linear algebra.

- file: Pe,
- command: ‘pe/ms/linalg/sym‘,
- examples: Example 22 pp. 53,
- input: exponential generating function, $(M, opt), m, q$,
- output: exponential generating function of the poly-exponential function multisectioned by m at q,
- reference: Section 4.3.

A.2 Code for exponential generating functions.

A.2.1 Making procedure from an exponential generating function.

This will turn a linear recurrence relation into a procedure, which will calculate any particular value of the linear recurrence relation.

- file: Egf,
- command: ‘egf/makeproc‘,
- examples: Example 21 pp. 51, Example 24 pp. 60, Example 25 pp. 61, Example 28 pp. 71, Example 29 pp. 75, Example 33 pp. 91, and Example 34 pp. 95,
- input: exponential generating function,
- output: Function.

A.2.2 Stripping zeros from exponential generating function.

This will take a multisectioned exponential generating function, and strip out the terms that are known to be zero.

- file: Egf,
- command: 'egf/strip'
- examples: Example 31 pp. 79,
- input: exponential generating function, m, q,
- output: exponential generating function.

A.2.3 Naive method to multisection.

This will take an exponential generating function and multisection it using the naive method as given in Definition 2.6.

- file: Egf,
- command: 'egf/ms/naive',
- examples: Example 5 pp. 17,
- input: exponential generating function, m, q
- output: exponential generating function multisectioned by m, at q,
- reference: Lemma 2.1, Definition 2.6 and Theorem 2.1.

A.2.4 Recurrence polynomial method.

This will take an exponential generating function and multisection it by multiplication of its recurrence polynomial.

- file: Egf,
- command: 'egf/ms/rec',
- examples: Example 19 pp. 46,

- input: exponential generating function, m, q,
- output: exponential generating function multisectioned by m, at q,
- reference: Section 4.1.

A.2.5 Recurrence polynomial via resultants method.

This will take an exponential generating function and multisection it by using resultants.

- file: Egf,
- command: 'egf/ms/result',
- examples: Example 20 pp. 49,
- input: exponential generating function, m, q,
- output: exponential generating function multisectioned by m, at q,
- reference: Section 4.2.

A.2.6 Linear algebra method.

This will take the exponential generating function and use linear algebra to multisection the linear recurrence relation.

- file: Egf,
- command: 'egf/ms/linalg',
- examples: Example 21 pp. 51,
- input: exponential generating function, M, m, q,
- output: exponential generating function multisectioned by m, at q,
- reference: Section 4.3.

A.2.7 Compression method.

This will use compression techniques to multisection the linear recurrence relation of an exponential generating function.

- file: Egf,
- command: 'egf/ms/compress',
- examples: Example 23 pp. 57,
- input: exponential generating function, m, q,
- output: exponential generating function multisectioned by m, at q,
- reference: Section 4.5.

A.3 Metrics.

A.3.1 Metric deg^d.

This is the code that will return $deg^d(s(x))$ given input $s(x)$.

- file: Metric,
- command: 'egf/metric/d', 'pe/metric/d',
- examples: Example 8 pp. 20,
- input: exponential generating function or poly-exponential function,
- output: $deg^d(s(x))$,
- reference: Definition 2.7.

A.3.2 Metric deg^P.

This is the code that will return $deg^P(s(x))$ given input $s(x)$.

- file: Metric,
- command: 'egf/metric/P', 'pe/metric/P',

- examples: Example 8 pp. 20,
- input: exponential generating function or poly-exponential function,
- output: $deg^P(s(x))$,
- reference: Definition 2.7.

A.4 Conversions.

A.4.1 Convert to the recurrence polynomial.

This will convert a linear recurrence relation to a recurrence polynomial.

- file: Convert,
- command: 'convert_poly',
- examples: Example 3 pp. 8,
- input: linear recurrence relation,
- output: recurrence polynomial,
- reference: Definition 2.2.

A.4.2 Convert to the linear recurrence relation.

This will convert a recurrence polynomial to a linear recurrence relation.

- file: Convert,
- command: 'convert_rec',
- examples: Example 3 pp. 8,
- input: recurrence polynomial,
- output: linear recurrence relation,
- reference: Definition 2.2.

A.4.3 Convert to the exponential generating function.

This will convert a poly-exponential function into an exponential generating function so that the linear recurrence relation is easily read.

- file: Convert,
- command: 'convert_egf',
- examples: Example 1 pp. 7,
- input: poly-exponential function,
- output: exponential generating function,
- reference: Lemma 2.1 and Theorem 2.1.

A.4.4 Convert to the exponential generating function.

This will convert an exponential generating function where the linear recurrence relation is easily readable into a poly-exponential function.

- file: Convert,
- command: 'convert_pe',
- examples: Example 2 pp. 8,
- input: exponential generating function,
- output: poly-exponential function,
- reference: Theorem 2.1.

A.5 Bottom linear recurrence relation.

A.5.1 Naive method.

This code will naively use the formula in Lemma 3.1 to determine the bottom linear recurrence relation.

- file: Bottom,

- command: 'bottom/ms/naive',
- examples: Example 13 pp. 30,
- input: poly-exponential function $t(x)$, m,
- output: exponential generating function of $\prod_{i=1}^{m} t(x\omega_m^i)$,
- reference: Lemma 3.1.

A.5.2 Fast Fourier transform and linear algebra.

Uses a combination of linear algebra and fast polynomial multiplication to determine the bottom linear recurrence relation.

- file: Bottom,
- command: 'bottom/ms/linalg/fft', 'bottom/ms/linalg/fft2',
- examples: Example 27 pp. 68 and Example 28 pp. 71,
- input: exponential generating function $t(x)$, M, m,
- output: exponential generating function of $\prod_{i=1}^{m} t(x\omega_m^i)$,
- reference: Section 5.2.

A.5.3 Symbolic differentiation and linear algebra.

This method uses a combination of symbolic differentiation and linear algebra.

- file: Bottom,
- command: 'bottom/ms/linalg/sym',
- examples: Example 22 pp. 53,
- input: poly-exponential function $t(x)$, $2M$, m,
- output: exponential generating function of $\prod_{i=1}^{m} t(x\omega_m^i)$,
- reference: Section 4.3.

A.5.4 Using the recurrence polynomial and resultants.

This will use the resultant to determine the linear recurrence relation.

- file: Bottom,
- command: 'bottom/ms/result',
- examples: Example 26 pp. 65,
- input: exponential generating function $t(x)$, m,
- output: exponential generating function of $\prod_{i=1}^{m} t(x\omega_m^i)$,
- reference: Section 5.1.

A.5.5 Factoring out common polynomials.

This factors out common polynomials to simplify the problem. This can be used in combination with any of the other methods.

- file: Bottom,
- command: 'bottom/ms/factor',
- examples: Example 32 pp. 85,
- input: poly-exponential function $t(x)$, m,
- output: exponential generating function of $(\prod_{i=1}^{m} t(x\omega_m^i))$,
- reference: Section 4.7.

A.6 Top linear recurrence relation.

A.6.1 Naive method.

This code will naively use the formula in Lemma 3.1 to determine the top linear recurrence relation.

- file: Top,
- command: 'top/ms/naive',

- examples: Example 13 pp. 30,
- input: poly-exponential functions $t(x)$, $s(x)$, m, q,
- output: exponential generating function of $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$,
- reference: Lemma 3.1.

A.6.2 Fast Fourier transform and linear algebra method.

This will use a combination of fast polynomial multiplication and linear algebra to solve the problem.

- file: Top,
- command: 'top/ms/linalg/fft',
- examples: Example 27 pp. 68,
- input: exponential generating function $t(x)$, $s(x)$, M, m, q,
- output: exponential generating function of $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$,
- reference: Section 5.2.

A.6.3 Symbolic differentiation and linear algebra.

This uses a combination of symbolic differentiation and linear algebra.

- file: Top,
- command: 'top/ms/linalg/sym',
- examples: Example 22 pp. 53,
- input: exponential generating function of $s(x), t(x)$, $\prod t(x\omega_m^i)$, m, q,
- output: exponential generating function of $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$,
- reference: Section 4.3.

A.6.4 Computing top linear recurrence relation with bottom.

This computes the top linear recurrence relation given the bottom linear recurrence relation.

- file: Top,
- command: 'top/ms/linalg/know',
- examples: Example 29 pp. 75,
- input: exponential generating function of $s(x)$, $t(x)$, $\prod t(x\omega_m^i)$, m, q,
- output: exponential generating function of $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$,
- reference: Section 5.3.

A.6.5 Knowing probably linear recurrence relation.

This computes the initial values given the top linear recurrence relation, the bottom linear recurrence relation and the recursion formula.

- file: Top,
- command: 'top/ms/know',
- examples: Example 29 pp. 75,
- input: exponential generating function of $s(x)$, $t(x)$, $\prod t(x\omega_m^i)$, m, q,
- output: exponential generating function of $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$,
- reference: Section 5.3.

A.6.6 Computing new recurrence polynomial using resultants.

This computes the new recurrence polynomial by using resultants.

- file: Top,
- command: 'top/ms/result',
- examples: Example 26 pp. 65,
- input: exponential generating function $s(x)$, $t(x)$, m, q,

- output: exponential generating function of $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$,
- reference: Section 5.1.

A.6.7 Factoring out common polynomials.

This method will factor out common polynomials to simplify the problem. This can be used in combination with any of the other methods.

- file: Top,
- command: 'top/ms/factor',
- examples: Example 32 pp. 85,
- input: poly-exponential function $s(x)$, $t(x)$,
- output: exponential generating function of $(s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$,
- reference: Section 4.7.

A.7 Doing the calculation.

A.7.1 Normal method.

This is just the normal method, using only one processor.

- file: Normal,
- command: 'calcul/normal',
- examples: Example 13 pp. 30,
- input: linear recurrence relations, m, q, and how far to calculate.,
- output: the $mi+q$-th values ,
- reference: Theorem 3.1.

A.7.2 Multiprocessor, even load-balance method.

This will assume multiple, evenly balanced processors, which this algorithm will take advantage of with communication.

- file: Multi,
- command: ‘calcul/balanced‘,
- examples: Example 33 pp. 91,
- input: linear recurrence relations, m, q, and how far to calculate,
- output: the $mi + q$-th values,
- reference: Section 6.1.

A.7.3 Multiprocessor, uneven load-balance method.

This will assume multiple, unevenly balanced processors. This algorithm will balance, and utilize these processors with communication to perform calculations.

- file: Multi,
- command: ‘calcul/balancing‘,
- examples: Example 34 pp. 95,
- input: linear recurrence relation, m, q, and how far to calculate,
- output: the $mi + q$-th values,
- reference: Section 6.2.

Appendix B

Notation.

Symbol,	Meaning,	Page,
α, β,	elements of $\mathbb{C}$,	
γ,	Euler gamma function,	1,
λ, μ,	elements of $\mathbb{C}$ as $e^{\lambda x}$ or $(x-\lambda)$,	
τ,	golden ratio,	1,
ω_m,	root of unity,	11,
$\zeta(n)$,	Riemann zeta function.	1,
a_i, b_i, d_i,	variables in a linear recurrence relation,	6,
c_i,	variables in a recursion formula,	28,
$deg^d(f(x))$,		20,
$deg^P(f(x))$,		20,
$f(x)$, $g(x)$, $h(x)$,	functions in $\mathcal{R}$,	26,
$f_m^q(x)$,	multisectioned function,	11,
i, j, k,	indexes for sums, or products,	
$j^{(r)}$,	$j(j-1)(j-2)...(j-r+1)$	
m,	by what a function is multisectioned,	11,
n,	a fixed integer,	
$p_i(x)$, $q_i(x)$	polynomials in x,	
q	to what a function is multisectioned,	11,
r_i	an unrelated set of integers,	
$r(x)$, $s(x)$, $t(x)$	functions in $\mathcal{P}$,	5,

x	variable,	
y	variable of integration or resultant,	
$\mathbb{C}$	Complex numbers,	
$C_m^q(f_m^q(x))$,	Compression,	55.
G,	Catalan's constant,	1,
N	size of the linear recurrence relation in $\mathcal{P}$,	
$P^f(x)$,	Recurrence polynomial	8,
$\mathcal{P}$,	Poly-exponential functions,	5,
$\mathcal{P}_{R_1,R_2}$,		10,
$\mathcal{P}^{R_1,R_2}$,		10,
$\mathbb{Q}$	Rationals,	
R_i,	subrings of $\mathbb{C}$,	
$\mathcal{R}$,	Rational poly-exponential functions,	26,
$\hat{\mathcal{R}}$,		28,
$\mathcal{R}^{R_1,R_2}$,		34,
$\mathcal{R}_{R_1,R_2}$,		34,
$\hat{\mathcal{R}}^{R_1,R_2}$,		34,
$\hat{\mathcal{R}}_{R_1,R_2}$,		34,
$\mathrm{Res}_x(p(x),q(x))$,	Resultant,	48,
$\mathbb{Z}$	Integers,	

Appendix C

Definitions.

Definition,	Symbol,	Page,
Bernoulli numbers,	$\frac{x}{e^x-1}$	27,
Bernoulli polynomials,	$\frac{xe^{tx}}{e^x-1}$	41,
Catalan's constant,	G,	1,
Chebyshev T polynomials,		24,
Compression,	C_m^q (for some q and m),	55,
Compression by m,	C_m^q (for some q),	55,
Compression by m at q,	C_m^q,	55,
Divide and conquer,		67,
Euler gamma function,	γ,	1,
Euler numbers	$\frac{2}{e^x+e^{-x}}$	68,
Fast Fourier transform		67,
Fibonacci numbers,		8,
Genocchi numbers,	$\frac{2x}{e^x+1}$	65,
Golden mean,	τ,	1,
Lacunary recurrence relation,		11,
Lacunary recursion formula,		30,
Linear recurrence relation,		6,
Lucas numbers type I,		17,
Lucas numbers type II,	$\frac{x}{e^x-e^{-x}}$	71,
Multisection,	$f_m^q(x)$ (for some q and m),	11,

Appendix D

Maple bugs and weaknesses.

This appendix includes some email corresponding between myself and Maple Software concerning bugs and weaknesses in their product. Some editing has been done on the letters for brevity as well as grammatical and spelling corrections.

D.1 Bug 7345 - expand/bigpow and roots of unity.

```
From kghare Thu Nov 26 17:14:46 1998
Subject: expand/bigpow
To: mapledev@daisy.uwaterloo.ca

Why is `expand/bigpow` being called in the second case?  It is noticeable
slower.

Kevin
-------

kernelopts(printbytes=false);
Poly := convert(taylor(exp(x)-1,x=0,73)*72!,polynom):

readlib(profile);
readlib(`expand/bigpow`):

profile(`expand/bigpow`):
```

```
tt := time():
poly[2] := expand(subs(x=x*exp(4*Pi*I/5),Poly)):
time() - tt;
showprofile(‘expand/bigpow‘);

tt := time();
poly[3] := expand(subs(x=x*exp(6*Pi*I/5),Poly)):
time() - tt;
showprofile(‘expand/bigpow‘);
```

```
> Poly := convert(taylor(exp(x)-1,x=0,73)*72!,polynom):
>
> readlib(profile);
                              proc()  ...  end

> readlib(‘expand/bigpow‘):
>
> profile(‘expand/bigpow‘):
>
> tt := time():
> poly[2] := expand(subs(x=x*exp(4*Pi*I/5),Poly)):
> time() - tt;
                                    .054

> showprofile(‘expand/bigpow‘);
function           depth    calls     time    time%         bytes   bytes%
-------------------------------------------------------------------------
expand/bigpow          0        0    0.000     0.00             0     0.00
-------------------------------------------------------------------------
total:                 0        0    0.000     0.00             0     0.00

>
> tt := time();
                                 tt := .122
```

```
> poly[3] := expand(subs(x=x*exp(6*Pi*I/5),Poly)):
> time() - tt;
                                  12.906

> showprofile(‘expand/bigpow‘);
function           depth    calls     time    time%         bytes   bytes%
-------------------------------------------------------------------------
expand/bigpow          2     1917   12.877   100.00      37245244   100.00
-------------------------------------------------------------------------
total:                 2     1917   12.877   100.00      37245244   100.00



From kghare Mon Nov 30 15:14:08 1998
Subject: Re: expand/bigpow
To: mapledev@daisy.uwaterloo.ca

I found an easier example demonstrating that something is
wrong.  Noticed, I only changed which 5th root of unity
I was looking at.

> exp(2*Pi*I*2/5)^500;
                                          500
                              exp(4/5 I Pi)

> expand(%);
                                    1

> time();
                                  .079

> exp(2*Pi*I*3/5)^500;
                                           500
                             exp(- 4/5 I Pi)

> expand(%);
```

```
bytes used=1000132, alloc=786288, time=0.19
bytes used=2000888, alloc=1179432, time=0.40
bytes used=3001084, alloc=1441528, time=0.68
bytes used=4001256, alloc=1769148, time=1.00
<SNIP>
bytes used=115099316, alloc=18477768, time=87.06
bytes used=116099516, alloc=18543292, time=88.17
bytes used=117099900, alloc=18739864, time=89.30
bytes used=119406568, alloc=19984820, time=90.15
                                        1

This amount of time, (and for that matter, memory requirements)
doesn't seem reasonable for a problem such as this.

Kevin
```

D.2 Bug 7357 - help for Euler.

Help for the Euler function was wrong.

```
From kghare Tue Dec  8 14:34:42 1998
Subject: Help page for Euler
To: mapledev@daisy.uwaterloo.ca

From the help page for the Euler function we have:

>euler - Euler numbers and polynomials
>
>Calling Sequence:
>     euler(n)
>     euler(n, x)
>
>Parameters:
>     n - a non-negative integer
>     x - an expression
>
>Description:
```

```
>- The function euler computes the nth Euler number, or the nth Euler
>  polynomial in x. The nth Euler number E(n) is defined by the exponential
>  generating function:
>
>
>        2/(exp(t)+exp(-t)) = sum(exp(n)/n!*t^n, n = 0..infinity)
```

This line should read

```
        2/(exp(t)+exp(-t)) = sum(E(n)/n!*t^n, n = 0..infinity)
```

and there should be some description of what E(n,x) is, the nth Euler polynomial.

Kevin

D.3 Bug 7497 - the "process" package.

```
From kghare Thu Oct 15 13:20:35 1998
Subject: Process Package in maple
To: mapledev@daisy.uwaterloo.ca

To: Stefan Vorkoetter;
cc: Maple Dev

I am currently trying to use the "process" package in Maple R5.
For some reason, the new forked processes are having problems
reading the library.

I get the error messages:

Error, (in DoWork) `/maple/mapleR5/lib/process/block.m` is an incorrect or ou\
tdated .m file (rFfn)
> quit
bytes used=239656, alloc=262096, time=0.01
Error, (in DoWork) `/maple/mapleR5/lib/process/block.m` is an incorrect or ou\
```

```
tdated .m file (ot3d)
> Error, (in Multi2) invalid subscript selector

This appears to be true on both the CECM machines at Simon Fraser University,
and daisy, at SCG.  If you want to see a copy of the code,
it can be found in my daisy account at
~kghare/Multi2

If you don't have access to daisy, and are interested in seeing
the code, just contact me, and I will mail it to you.  (approx 236 lines)

If I have in my program,
unprotect(block);
block := ....
and simply copy the code in, then everything works fine.
Except that it is an ever-growing list of files that I need to
do this to. (binomial, convert/string, type/odd, fprintf, close, readline, ...)

Any suggestions as to what I might be doing wrong would be appreciated.
I am too unfamiliar with the package to decide if it is a bug, or
I am just using it wrong.

Thanks

Kevin

From kghare Tue Nov 10 17:17:53 1998
Subject: Process Package
To: mapledev@daisy.uwaterloo.ca

When using the "process" package in maple, there is something
strange going on with the libraries and/or kernel after a fork
command.  The child process does not seem to be able to access anything
in the library properly, and I get errors such as:

    |\^/|     Maple V Release 5 (Simon Fraser University)
```

```
> read Multi;
> Multi(3,6);
Error, (in DoWork) could not find `process/block` in the library
Error, (in DoWork) could not find `binomial` in the library
> quit
bytes used=227108, alloc=262096, time=Error, exponent too large
maple: unexpected end of input
> quit
bytes used=227208, alloc=262096, time=Error, exponent too large
maple: unexpected end of input
Error, (in Multi) could not find `process/block` in the library

This is making the code very annoying to use, as I have to use
work-arounds to get around this bug.  (I predefine anything that
the child process will need, so that the child process will not need to
access the library.)  This is in the released version of maple, so it
is not simply a problem of rmaple being a bit out of sync.  Further
it occurs both on the CECM machines (in particular "bb"), and on
the SCG machine (daisy), so it is not a problem with any particular
maple installation.

It would be nice if a patch or fix could be found for this, as I am
using this functionality in my research.

read Multi;
Multi(3,100);

Thanks

Kevin Hare
```

D.4 Bug with "process package" and bytes used message.

```
Subject: process[fork] and bytes used message
To: mapledev@daisy.uwaterloo.ca
Date: Wed, 27 Jan 1999 16:19:28 -0800 (PST)
```

When the process[fork] command is called, the options about printing bytes, or not printing bytes is ignored by either the child of the parent. (Probably the child.) Also, the printbytes message is not able to figure out the time, and returns an error message. This was done with the following scripts.

```
kernelopts(printbytes=false);
with(process):

A := proc()
    local pid;
    kernelopts(printbytes=false);

    pid := fork();

    if pid = 0 then # This is the child
        print("The child has run");
        quit;
    else # This is the parent
        print("The parent has run");
    fi;
    RETURN():
end;

A();
```

```
----------------------------------------------------------------------
    |\^/|     Maple V Release 5 (Simon Fraser University)

> kernelopts(printbytes=false);
                                     true

> with(process):
> A := proc()
>     local pid;
>     kernelopts(printbytes=false);
```

```
>     pid := fork();
>     if pid = 0 then # This is the child
>         print("The child has run");
>         quit;
>     else # This is the parent
>         print("The parent has run");
>     fi;
>     RETURN():
> end;
A := proc()
local pid;
    kernelopts(printbytes = false);
    pid := fork();
    if pid = 0 then print("The child has run"); quit
    else print("The parent has run")
    fi;
    RETURN()
end

> A();
                              "The child has run"

                             "The parent has run"

bytes used=209100, alloc=196572, time=Error, (in A) exponent too large
> quit
> bytes used=209612, alloc=196572, time=Error, exponent too large
maple: unexpected end of input

> quit
bytes used=209184, alloc=196572, time=0.05
```

D.5 Bug with "process" package on xMaple.

```
Subject: process[fork] and xmaple interface
```

```
To: mapledev@daisy.uwaterloo.ca

When using the process[fork] command, I get more than one
thread of execution running.  As is standard, I must "quit"
all but one of these threads before returning control to
the command prompt level.  Unfortunately, if I am using
xmaple, any quit command, from either the child, or the parent
will result in the worksheet exiting.  Hence the following
procedure:

with(process);

A := proc()
    local pid;

    pid := fork();

    if pid = 0 then # This is the child
        print("The child has run");
        quit;
    else # This is the parent
        print("The parent has run");
    fi;
    RETURN():
end;

A();

This will run almost properly on the text based version (modulo the other
bug I just reported), but will terminate the worksheet if it is run under
xmaple (occasionally).

Kevin
```

D.6 Bug 7552 - factorial.

```
Subject: Kernel level factorial is slow, inefficient, and forgetful
To: bugkeeper@maplesoft.com

Below is a very rough version of a factorial function.  It is
written using interpreted maple, where as the built-in versions
is kernel level.  Despite the difference in speed of interpreted
code versus kernel level code, the interpreted version is considerably
faster.

    |\^/|     Maple V Release 5 (Simon Fraser University)

> Fac1 := proc(n)
>     local A;
>     if n < 100 then RETURN (n!)
>     else
>          A := ((n^10-45*n^9+870*n^8-9450*n^7+63273*n^6-269325*n^5+
>              723680*n^4-1172700*n^3+1026576*n^2-362880*n)*`procname`(n-10));
>         RETURN(A);
>     fi;
> end:
>
> tt := time(): Fac1(10000): time() - tt;
bytes used=1005196, alloc=982860, time=0.19
<SNIP>
bytes used=18202916, alloc=4259060, time=3.97
                                     4.013

> tt := time(): 10000!: time() - tt;
                                    11.516

Next, if we add some sort of memory to this function (for example,
here I remember every 100 th value), then the speed is greatly increased
for doing multiple calculations, (yet the memory requirements still
remain low).
```

```
> Fac2 := proc(n)
>     local A;
>     if n < 100 then RETURN (n!)
>     elif (n = 0) mod 10 then
>         A := ((n^10-45*n^9+870*n^8-9450*n^7+63273*n^6-269325*n^5+
>              723680*n^4-1172700*n^3+1026576*n^2-362880*n)*`procname`(n-10));
>         if (n=0) mod 100 then
>             `procname`(n) := A;
>         fi:
>         RETURN(A);
>     else
>         RETURN(`procname`(n-1)*n);
>     fi;
> end:
>
> tt := time():
> for i from 1 to 10000 by 19 do
> Fac2(i):
> od:
<SNIP>
bytes used=100348464, alloc=6945544, time=16.19
> time() - tt;
                                    16.263

>
> tt := time():
> for i from 1 to 10000 by 19 do
> i!:
> od:
bytes used=101348908, alloc=6945544, time=22.80
bytes used=102359904, alloc=6945544, time=115.56
bytes used=103367344, alloc=6945544, time=262.03
```

Killed as I didn't have the patients to wait. But it is clear that it is going to take more than 10 times the amount of time to finish. (I estimated the time that it would take at around 3000 seconds, but I don't know exactly.)

```
Kevin
```

D.7 Bug 5793 - Multi-argument forget does not work.

```
Subject: Forget forgets more than it should.

According to the help page for forget:

Calling Sequence:
     forget(f,...)
     forget(f,a,b,c,...)

Parameters:
     f      - any name assigned to a Maple procedure
     a, b, c, ... - (optional) specific argument sequence for the function f
     ...    - options

<SNIP>

- forget(f,a,b,c,...) causes the value of f(a,b,c,..) to be ``forgotten''. As
  with the one-argument case, the entry for the argument list a,b,c,... is
  removed from the remember table for f and also from the remember table for
  all functions whose names begin with f/.

<SNIP>

Yet this doesn't even work with the example given in the help page.

> f(x) := 456:
> f(y) := 12:
> f(x),f(y);
                                    456, 12

> forget(f,x);
> f(x),f(y);
```

```
f(x), f(y)
```

```
It is forgetting too much.
```

```
Kevin
```

Appendix E

Code

E.1 Conversions.

File name: Convert.

```
## Notation:
## m.s.   = multisection
## r.p.e. = rational poly-exponential function
## p.e.   = poly-exponential function
## e.g.f. = exponential generating function

macro(`egf/clean` = readlib(`egf/clean`));

# convert_pe
#    This will convert an e.g.f. to a p.e.
# Input: e.g.f.
# Output: p.e.
# References: Theorem 2.1.
`convert_pe` := proc(recur, f, var, init)
    local poly, lambda, n, alpha, Pe, i, deg, Ped, eq, soln, Pez, Eq;

    poly := convert_poly(recur, f, var, init);

    lambda := [solve(poly, var)];
    if has(lambda, RootOf) then
        lambda := map(allvalues, lambda):
    fi:

    n := nops(lambda):

    Pe := 0:
    for i from 1 to n do
        deg := degree(coeff(Pe, exp(var*lambda[i])), var):
        if deg = -infinity then
            deg := 0:
        else
            deg := deg + 1:
        fi:

        Pe := a[i] *exp(lambda[i]*var)*var^deg + Pe:
    od:

    Ped := Pe:

    for i from 0 to nops(init) -1 do
        Pez := subs(var=0,Ped):
        Ped := diff(Pe, var):
        eq[i] := subs(init,f(i)=Pez):
    od:

    Eq := {seq(eq[i],i=0..nops(init)-1)};
    Eq := simplify(Eq):
    soln := solve(Eq);

    Pe := subs(soln, Pe):

    RETURN(Pe, var):
end:

# pe/comb
#    Will take a sequence of p.e. components, and combine
#    ones with common lambda.
# Input: seqn of p.e.
# Output: seqn of p.e.
`pe/comb` := proc(seqn)
    local seqn2, lambda, temp, i;
    userinfo(5,'MS',"Combining lambdas together");
    lambda := {};
    seqn2 := {};
    for i in seqn do
        if member(i[2], lambda) then
            temp := select(proc(x,y)
                if evalb(x[2] = y) then RETURN(true) fi; RETURN(false) end,
                seqn2, i[2]);
            seqn2 := seqn2 minus temp;
            temp := op(temp);
            temp[1] := radnormal(temp[1] + i[1]);
            seqn2 := seqn2 union {[temp[1],temp[2]]};
        else
            lambda := lambda union {i[2]};
            seqn2 := seqn2 union {[i[1],i[2]]};
        fi;
    od;

    RETURN(seqn2);
end:

# convert_egf
#     Takes a p.e. and converts it to an e.g.f.
# Input: p.e.
# Output: e.g.f.
# Reference: Theorem 2.1.
`convert_egf` := proc(seqn, f, var)
    local temp, poly, y, seqn2, size, i, init;

    seqn2 := readlib(`pe/convert`)(seqn,var);

    userinfo(3,'MS',"Combining lambdas");
    seqn2 := readlib(`pe/comb`)(seqn2);

    userinfo(3,'MS',"Creating polynomial");
    poly := mul((var-y[2])^(degree(y[1],var)+1),y=seqn2);

    size := degree(poly,var);
    userinfo(3,'MS',"Expanding polynomial of degree", size);
    poly := radnormal(expand(poly * var));
```

```
    userinfo(3,'MS',"Creating Recurrence relation");
    poly := convert_rec(poly,f,var);

    userinfo(3,'MS',"Finding taylor series (to deal with lambda 0)");
    temp := add(y[1]*exp(var*y[2]),y=seqn2);

    init := []:
    for i from 0 to size-1 do
        init := [op(init),f(i)=simplify(subs(var=0,temp))];
      if (i mod 10) = 0 then
        userinfo(3,'MS',`Working on coeff`,i);
      fi;
        temp := expand(diff(temp,var));
    od;

    RETURN(`egf/clean`(poly, f, var, map(radnormal,init,expanded)));
end:

# pe/convert
#     Converts a p.e. to a seqence of p.e.'s.
# Input: p.e.
# Output: sequence of p.e.'s.
`pe/convert` := proc(f,var)
    option remember, system;
    local func, t, combo, p, lambda, alpha, tt, counter;

    userinfo(3, 'MS', "Working on poly-exponential function");
    func := convert(f,exp);
    func := expand(func);
    func := convert(func, exp);
    func := combine(func, exp);
    func := convert(func, exp);

    userinfo(3, 'MS', "Combining exp");
    func := combine(func, exp);

    if type(func, `+`) then
        func := [op(func)];
    else
        func := [f];
    fi;

    userinfo(3, 'MS', "Converting the", nops(func),
        "terms to the correct type");

    counter := 0;
    combo := {};
    for tt in func do
      counter := counter + 1;
      if (counter mod 10) = 0 then
          userinfo(3,'MS',"Working on number", counter);
      fi;

        t := combine(tt,exp);
        p := frontend(degree,[t,var]);
        t := t / var^p;
        t := simplify(convert(t,exp));

        if not has(t, var) then
            alpha := t;
            lambda := 0;
        elif type(t,`*`) then
            lambda := select(has, t, var);
            alpha := t/lambda;
            lambda := op(1,lambda);
            lambda := lambda/var;
        else
            alpha := 1;
            lambda := op(1,t);
            lambda := lambda/var;
        fi;
        combo := [op(combo), [alpha * var^p, lambda]];
    od;
    combo := readlib(`pe/comb`)(combo);
    RETURN(combo);
end:

# convert_poly
#     Converts the e.g.f. to its associate recurrence polynomial
# Input: e.g.f.
# Output: Recurrence polynomial
# Reference: Section 2.3, Definition 2.2.
`convert_poly` := proc(recur, f, var, init)
    local size, temp, poly, i, temp1, VAR, k, egf;

    userinfo(3,'MS',"Converting to polynomial");
    temp1 := expand(rhs(recur));

    temp := {};

    if type(temp1,`+`) then
        for i in temp1 do
            if type(i,`*`) then
                temp := {select(has,i,f)} union temp;
            else
                temp := {i} union temp;
            fi;
        od;
    else
        if type(temp1,`*`) then
            temp := {select(has,temp1,f)} union temp;
        else
            temp := {temp1} union op(temp);
        fi;
    fi;
    temp := map2(op,1, temp);
    temp := subs(var=0,temp);
    temp := min(op(temp));
    size := -temp;

    poly := rhs(recur);

    userinfo(3,'MS',"Creating Recurrence polynomial");
    for i from size+1 by -1 to 1 do
       poly := subs( {f(var-i) = VAR^(size-i)},poly);
    od;
    poly := expand(var^(size+1) - subs (VAR=var,poly)*var);

    userinfo(3,'MS',"Determine size of k");
    if type(init,list) then
        egf := `egf/clean`(recur, f, var, init);
        k := nops(egf[4])-size - 1;
        k := max(k,-1);
    else
        k := -1;
    fi;

    poly := expand(poly * var^k);

    RETURN(poly);
end:

# convert_rec
#     Converts the recurrence polynomial to the recurrence of some e.g.f.
# Input: Recurrence polynomial
# Output: Recurrence
# Reference: Section 2.3, Definition 2.2.
`convert_rec` := proc(Poly, f, var)
    local size, VAR, poly, i;

    poly := Poly;
    size := degree(poly,var);
    userinfo(3,'MS',"Expanding polynomial of degree", size);
    poly := expand(poly * var);
    poly := expand(poly/lcoeff(poly));
```

```
    userinfo(3,'MS',"Creating recurrence relation");
    for i from size+1 by -1 to 1 do
       poly := subs( {var^i=f(VAR-size+i-1)},poly);
    od;
    poly := subs (VAR=var,poly);
    poly := f(var) = solve(poly,f(var));

    RETURN(poly);
end:

savelib(`convert_pe`, `convert_pe.m`);
savelib(`convert_egf`, `convert_egf.m`);
savelib(`pe/convert`, `pe/convert.m`);
savelib(`convert_poly`, `convert_poly.m`);
savelib(`convert_rec`, `convert_rec.m`);
savelib(`pe/comb`, `pe/comb.m`);
```

E.2 Metrics.

File name: Metric.

```
## Notation:
## m.s.   = multisection
## r.p.e. = rational poly-exponential function
## p.e.   = poly-exponential function
## e.g.f. = exponential generating function

macro(`pe/convert` = readlib(`pe/convert`),
      `pe/comb`    = readlib(`pe/comb`)):

# pe/metric/d
#     Takes a p.e. $s$ and computes $deg^d(s)$
# Input: p.e.
# Output: $deg^d(p.e.)$
# Reference: Definition 2.7.
`pe/metric/d` := proc(pe, var)
    local seqn;
    userinfo(5,'MS',"Determining the maximal degree polynomial of the".
        " poly-exponential function.");
    seqn := [op(expand(pe))];
    seqn := subs(exp=1,seqn);
    seqn := simplify(seqn);
    seqn := map(degree, seqn, var);
    RETURN(max(op(seqn)));
end:

# pe/metric/P
#     Takes a p.e. $s$ and computes $deg^P(s)$
# Input: p.e.
# Output: $deg^P(p.e.)
# Reference: Definition 2.7
`pe/metric/P` := proc(pe, var)
    local seqn, i, P, x;
    userinfo(5,'MS',"Determining the size of the recurrence relationship of ".
                    "the poly-exponential function");
    seqn := `pe/convert`(pe, var);
    seqn := `pe/comb`(seqn);
    seqn := [op(seqn)];
    seqn := map(proc(x, var) RETURN(degree(x[1],var)) end, seqn,var);
    P := 1;
    for i in seqn do
        P := P + (i+1);
    od;
    RETURN(P-1);
end:

# egf/metric/d
#     Takes a e.g.f. $s$ and computes $deg^d(s)$
# Input: e.g.f.
# Output: $deg^d(e.g.f.)
# Reference: Definition 2.7
`egf/metric/d` := proc(recur, f, var, init)
    local poly, poly2, i, g;
    userinfo(5,'MS',"Determining the maximal degree polynomial of the ".
        "exponential generating function");
    poly := convert_poly(recur, f, var, init);
    i := 0;
    poly2 := diff(poly,var);
    g := gcd(poly, poly2);
    while g <> 1 do
        i := i + 1;
        poly := g;
        poly2 := diff(poly,var);
        g := gcd(poly, poly2);
    od;
    RETURN(i);
end:

# egf/metric/P
#     Takes a e.g.f. $s$ and computes $deg^P(s)$
# Input: e.g.f.
# Output: $deg^P(e.g.f.)
# Reference: Definition 2.7
`egf/metric/P` := proc(recur, f, var, init)
    local poly;
    userinfo(5,'MS',"Determining the size of the recurrence relationship of ".
                    "the exponential generating function");
    poly := convert_poly(recur, f, var, init);
    RETURN(degree(poly,var));
end:

savelib(`egf/metric/d`, `egf/metric/d.m`);
savelib(`egf/metric/P`, `egf/metric/P.m`);
savelib(`pe/metric/d`, `pe/metric/d.m`);
savelib(`pe/metric/P`, `pe/metric/P.m`);
```

E.3 Poly-exponenial function.

File name: Pe.

```
## Notation:
## m.s.   = multisection
## r.p.e. = rational poly-exponential function
## p.e.   = poly-exponential function
## e.g.f. = exponential generating function

macro(`egf/clean` = readlib(`egf/clean`));

# pe/ms/naive
#     M.s. the p.e. by $m$ at $q$ using the naive approach.
# Input: p.e., m, q
# Output: e.g.f.
# Reference: Definiton 2.6.
#            Appendix A.1.1.
`pe/ms/naive` := proc(func, f, var, m, q)
    local pe, egf, k;

    pe := func;

    # Ref: Definition 2.6.
    userinfo(1,'MS',"Multisectioning poly-exponential function");
    pe := 1/m*sum(subs(var=var*(-1)^(2*k/m),pe)*(-1)^(-2*k*q/m),k=1..m);
    userinfo(1,'MS',"Convering multisectioned poly-exponential function to".
        " an exponential generating function.");
    egf := convert_egf(pe, f, var);
```

```
    RETURN(`egf/clean`(egf));
end:

# pe/ms/linalg/sym
#     Here we determine the first $2 M m$ initial values (via
#     symbolic differentiation), and then use linear algebra
#     to solve the recurrence relationship.
# Reference: Section 4.3.
#            Appendix A.1.2.
`pe/ms/linalg/sym` := proc(func, f, var, m, q, N)
    local C, MM, Ff, rec, i, initial, FF, Zero, B;

    Zero := `pe/metric/d`(func, var);
    if nargs = 6 then
        MM := N;
    else
        MM := `pe/metric/P`(func,var);
    fi;

    userinfo(1, 'MS', "Taking derivatives to determine taylor-series coeff");
    if q <> 0 then
        Ff := combine(expand(diff(func,[var$q])),exp);
    else
        Ff := combine(func,exp);
    fi;
    C[0] := eval(Ff,var=0);
    for i from 1 to 2 * MM do
        Ff := combine(expand(diff(Ff,[var$m])),exp);
        C[i] := radnormal(eval(Ff,var=0));
    od;

    B := [seq(C[i],i=ceil((Zero-q)/m)..2*MM)];

    userinfo(1, 'MS', "Using linear algebra to determine recurrence of size",
            2*MM);
    rec := `recurrence/solve/linalg`(B, f, var, m);

    FF := proc(i, m, q, C)
        if (i = q) mod m then
            RETURN(C[(i-q)/m]);
        else
            RETURN(0);
        fi;
    end;

    initial := [seq(f(i)=FF(i,m,q, C), i=0..MM * m - 1)];

    RETURN(`egf/clean`(rec, f, var, initial));
end:

# pe/ms
#     M.s. the p.e. by $m$ at $q$.
# Input: p.e., m, q, method[methodarg]
# Output: e.g.f.
`pe/ms` := proc(pe, f, var, m, q, opt)
    local i, method, methodarg, egf;

    userinfo(1, 'MS', "Multisectioning the poly-exponential function.");

    if nargs = 6 then
        if type(opt, indexed) then
            method := `pe/ms/`.(op(0, opt));
            methodarg := op(1, opt);
        else
            method := `pe/ms/`.opt;
        fi;
    else
        method := `pe/ms/linalg/sym`;
    fi;

    if assigned(methodarg) then
        egf := method(pe,f,var,m,q,methodarg);
    else
        egf := method(pe,f,var,m,q);
    fi;

    RETURN(`egf/clean`(egf));
end:

#libname := libname[3], libname[1..2]:

savelib(`pe/ms`, `pe/ms.m`);
savelib(`pe/ms/linalg/sym`, `pe/ms/linalg/sym.m`);
savelib(`pe/ms/naive`, `pe/ms/naive.m`);
```

E.4 Exponential generating function.

File name: Egf.

```
## Notation:
## m.s.   = multisection
## r.p.e. = rational poly-exponential function
## p.e.   = poly-exponential function
## e.g.f. = exponential generating function

macro (clean      = readlib(`egf/clean`),
       ifactors   = readlib(ifactors),
       forget     = readlib(forget),
       compress   = readlib(`egf/compress`),
       y          = `egf/ms/variable/y`,
       nn         = `egf/makeproc/variable/nn`,
       uncompress = readlib(`egf/uncompress`));

# egf/ms/naive
#     M.s. the e.g.f. using the naive method of converting it
#     to a p.e., and then m.s.'ing that using the definition of
#     m.s.
# Input: e.g.f., m , q
# Output: e.g.f.
# Reference: Definition 2.6.
#            Appendix A.2.3.
`egf/ms/naive` := proc(recur, f, var, init, m, q)
    local pe, egf;

    userinfo(1,'MS',"Convering the exponential generating function".
        " to a poly-exponential function".
        " and multisection it");
    pe := convert_pe(recur, f, var, init)[1];
    egf := `pe/ms/naive`(pe, f, var, m, q);

    RETURN(clean(egf));
end:

# egf/ms/result
#     M.s. the e.g.f by looking at the recurrence polynomial, and
#     using resultants.
# Input: e.g.f, m, q
# Output: e.g.f.
# Reference: Section 4.2.
#            Appendix A.2.5.
`egf/ms/result` := proc(recur, f, var, init, m, q)

    local poly, rep, size;

    size := `egf/metric/P`(recur,f,var,init);

    # The maximum number of repeated roots.
    rep  := `egf/metric/d`(recur,f,var,init);

    # Ref Lemma 2.5.
    userinfo(1,'MS', "Creating recurrence polynomial");
```

```
    poly := convert_poly(recur,f,var,init);
    size := size * m;

    # Section 4.2.
    userinfo(1,'MS', "Using resultants with the polynomial");
    poly := resultant(subs(var=y,poly), y^m - var^m, y);

    userinfo(1,'MS', "Creating recurrence equation");
    poly := convert_rec(poly,f,var);
    poly := simplify(poly);

    RETURN(clean(poly,f,var,readlib(`egf/init`)(recur,f,var,init,size/m,m,q)));
end:

# egf/ms/rec
#     M.s. the e.g.f. by looking at the recurrence polynomial, and
#     dealing with it in an approriate manner.
# Input: e.g.f., m, q
# Output: e.g.f.
# Reference: Section 4.1.
#            Appendix A.2.4.
`egf/ms/rec` := proc(recur, f, var, init, m, q)

    local poly, size, rep;

    size := `egf/metric/P`(recur,f,var,init);

    # The maximum number of repeated roots.
    rep  := `egf/metric/d`(recur,f,var,init);

    # Ref Lemma 2.5.
    userinfo(1,'MS', "Creating recurrence polynomial");
    poly := convert_poly(recur,f,var,init);
    size := size * m;

    # Section 4.1.
    userinfo(1,'MS', "Multisection recurrence polynomial");
    poly := readlib(`egf/ms/rec/multi`)(poly, var, m, 1, rep);

    userinfo(1,'MS', "Creating recurrence equation");
    poly := convert_rec(poly,f,var);
    poly := simplify(poly);

    RETURN(clean(poly,f,var,readlib(`egf/init`)(recur,f,var,init,size/m,m,q)));
end:

# egf/ms/rec/multi
#     M.s. the recurrence polynomial
# Input: poly, m
# Output: poly
# Reference: Section 4.1.
`egf/ms/rec/multi` := proc(f, x, m, d, rep)
    local p, F, i, F2, G;

    userinfo(3, 'MS', "Using multiplication of recurrence to get ".
                      "the new multisectioned recurrence", d);
    F := 1;
    # Ref: Section 4.1.
    if isprime(m/d) then
       for i from 0 to m/d-1 do
           F := expand(F * subs(x=x*(-1)^(2*i*d/m),f));
       od;
    else
       p  := ifactors(m/d)[2][1][1];

       if nargs = 5 then
           F2 := `procname`(f,x,m,d*p, rep);
       else
           F2 := `procname`(f,x,m,d*p);
       fi;
       for i from 0 to p-1 do
           F := expand(F * subs(x=x*(-1)^(2*i*d/m),F2));
       od;
    fi;

    if nargs = 5 then
        G := F;
        for i from 0 to rep do
            G := gcd(diff(G,x), G);
        od;
        F := quo(F, G, x);
    fi;

    F := expand(F / lcoeff(F, x)):

    RETURN(radnormal(F));
end:

# egf/ms/compress
#     m.s. the e.g.f. by repeated m.s.'ing by prime factor,
#     compressing that result, and m.s.'ing again.  method
#     used to m.s. the e.g.f. will default to linalg, but
#     can be choosed to be something else.
# Input: e.g.f., m, q, (optional) method
# Output: e.g.f.
# Reference: Section 4.5.
#            Appendix A.2.7.
`egf/ms/compress` := proc(recur, f, var, init, m, q, opt, opt2)
    local method, d, p, q1, q2, egf;

    if nargs >= 7 then
        method := `egf/ms/`.opt;
    else
        method := `egf/ms/linalg`;
    fi;


    userinfo(1, 'MS', "Multisection the exponential generating function".
        " using compression techniques and", method);

    egf := recur, f, var, init;

    d := 1;
    q1 := 0;
    q2 := 0;
    p := 1:

    # Ref: Section 4.5.
    while d <> m do
        p := ifactors(m/d)[2][1][1];
        userinfo(2, 'MS', "Calculating multisectioning by", d, "at", q2);
        d := d * p;
        q1 := ((q mod d)-q2)/d*p;
        q2 := q2 + d * q1/p;

        egf := method(egf,p,q1);
        if d = m then break; fi;
        egf := compress(egf, p, q1);
    od;
    if nargs = 8 and opt2 = "Leave Compressed" then
        RETURN(clean(compress(egf,p,q1))):
    fi;

    if m <> p then
        egf := uncompress(egf, m/p, q-m/p*q1);
    fi;

    RETURN(clean(egf));
end:

# egf/ms/linalg
#     M.s. the e.g.f. determining how large the recurrence polynomial
#     is and then calculating even $m$th term and using
#     linear algebra to determine the new recurrence
# Input: e.g.f., m, q
# Output: e.g.f.
# Reference: Section 4.3.
#            Appendix A.2.6.
```

```
`egf/ms/linalg` := proc(recur, f, var, init, m, q)
    local C, MM, Ff, rec, i, initial, FF, Zero;

    MM := `egf/metric/P`(recur, f, var, init);
    Zero := `egf/metric/d`(recur, f, var, init);

    userinfo(1,'MS',"Make the procedure for the egf");
    Ff := `egf/makeproc`(recur, f, var, init);

    for i from Zero to 2 * MM do
        C[i] := Ff(m*i+q);
    od;

    C := convert(C, list):

    userinfo(1,'MS',"Solve new recurrence using linear algebra");
    rec := `recurrence/solve/linalg`(C, f, var, m);

    FF := proc(i, m, q, Ff)
        if (i = q) mod m then
            RETURN(Ff(i));
        else
            RETURN(0);
        fi;
    end;

    initial := [seq(f(i)=FF(i, m, q, Ff), i=0..MM * m - 1)];

    RETURN(clean(rec, f, var, initial));
end:

# egf/ms
#     M.s. the e.g.f. by $m$ at $q$.
# Input: e.g.f., m, q, method[methodarg]
# Output: e.g.f.
`egf/ms` := proc(recur, f, var, init, m, q, opt)
    local i, method, methodarg, egf;

    userinfo(1, 'MS', "Multisectioning the egf");

    if nargs = 7 then
        if type(opt, indexed) then
            method := `egf/ms/`.(op(0, opt));
            methodarg := op(1, opt);
        else
            method := `egf/ms/`.opt;
        fi;
    else
        method := `egf/ms/linalg`;
    fi;

    if assigned(methodarg) then
        egf := method(recur, f, var, init, m, q, methodarg);
    else
        egf := method(recur, f, var, init, m, q);
    fi;

    RETURN(clean(egf));
end:


# egf/clean
#     Will look at the initial conditions and get rid of terms at the
#     end which are not required.
# Input: e.g.f.
# Output: e.g.f.
# Reference: NONE
`egf/clean` := proc(recur, f, var, init)
    local Init, k, Recur, Value;
    option system, remember;
    userinfo(5,'MS',"Getting rid of useless initial values");
    Init := init;
    k := nops(Init);
    do
        Recur := subs(var=k-1, recur);
        Value := subs(init, Recur);
        Value := simplify(lhs(Value)-rhs(Value));
        if evalb(Value=0) then
            Init := Init[1..-2];
            k := k-1;
        else
            break;
        fi;
    od;
    RETURN(recur, f, var, Init, args[5..nargs]);
end:

# egf/makeproc
#     This, given and e.g.f. and a function name, will return a recurrsive
#     function using the recurrence relationship of the e.g.f. and
#     the initial values given.
# Input: e.g.f.
# Output: procedure
# Reference: Appendix A.2.1.
`egf/makeproc` := proc(recur, f, var, init, scale)

    local maxinit, P, Rec, Procname, T, m, n;

    userinfo(1,'MS',"Making the procedure to calculate a recurrence");

    maxinit := map(lhs,init);
    maxinit := map2(op,1,maxinit);
    maxinit := max(op(maxinit));

    Rec := rhs(recur);
    if Rec = NULL then Rec := 0; fi;
    Rec := subs({var=nn, f=Procname}, Rec);
    P := subs({REC=Rec,Init=init,MaxInit=maxinit, F= f},
        (proc(`egf/makeproc/variable/nn`)
            option remember, system;
            if `egf/makeproc/variable/nn` < 0 then
                RETURN(0);
            elif `egf/makeproc/variable/nn` <= MaxInit then
                RETURN(subs(Init,F(`egf/makeproc/variable/nn`)));
            else
                RETURN(REC);
            fi;
        end));

    # This is a hack suggested by Greg Fee to allow me
    # to get the key word "procname" substituted into the
    # procedure, as uneval quotes won't work.
    P := subs(Procname=procna.me,op(P));

    if nargs = 4 then
        RETURN(op(P));
    else
        T := add(coeff(scale,var,m)*expand(i!/(i-m)!)*'P'(i-m),
            m=0..degree(scale,var));
        RETURN(unapply(T,i));
    fi;
end:

# egf/makeproc2
#     This, given and e.g.f. and a function name, will return a recurrsive
#     function using the recurrence relationship of the e.g.f. and
#     the initial values given.
# Input: e.g.f.
# Output: procedure
# Reference: NONE (Yet)
`egf/makeproc2` := proc(recur, f, var, init, After, PROCNAME)

    local maxinit, P, Rec, Procname, T, m;

    userinfo(1,'MS',"Making the procedure to calculate a recurrence");

    maxinit := map(lhs,init);
    maxinit := map2(op,1,maxinit);
```

```
    maxinit := max(op(maxinit));

    Rec := rhs(recur);
    if Rec = NULL then Rec := 0; fi;
    Rec := subs({var=nn, f=Procname}, Rec);
    P := subs({REC=Rec,Init=init,MaxInit=maxinit, F= f, after=After,P=PROCNAME},
        (proc(`egf/makeproc/variable/nn`)
            option system, remember;
            if `egf/makeproc/variable/nn` < 0 then
                RETURN(0);
            elif `egf/makeproc/variable/nn` <= MaxInit then
                RETURN(subs(Init,F(`egf/makeproc/variable/nn`)));
            else
                forget(P, `egf/makeproc/variable/nn`-after);
                RETURN(REC);
            fi;
        end));

    # This is a hack suggested by Greg Fee to allow me
    # to get the key word "procname" substituted into the
    # procedure, as uneval quotes won't work.
    P := subs(Procname=procna.me,op(P));

    RETURN(op(P));
end:

# egf/scale
#     Scale an e.g.f. by lambda
# Input: e.g.f., lambda
# Output: e.g.f.
# Reference: NONE
`egf/scale` := proc(recur, f, x, init, lambda)
    local poly, Recur, Init, i;

    userinfo(5,'MS', "Finding P^{f(lambda x)} given P^f and P^g");
    poly := convert_poly(recur,f,x,init);
    poly := subs(x=x/lambda,poly);

    Recur := simplify(expand(convert_rec(poly,f,x)));

    userinfo(5,'MS', "Finding inital values for P^{f(lambda x)} ".
                     "given P^f and P^g");
    Init := [];
    for i in init do
        Init := [op(Init), op(1,i) = expand(op(2,i)*lambda^op([1,1],i))];
    od;
    Init := (expand(radnormal(Init)));

    # Note, do not "clean" these results.
    RETURN(Recur, f, x, Init);
end:


# egf/compress
#     Compress an e.g.f. by $m$ at $q$
# Input: e.g.f., $m$, $q$
# Output: e.g.f.
# Reference: Section 4.5.
`egf/compress` := proc(recur, f, x, init, m, q)

    local Recur, Init, i, F;

    userinfo(3, 'MS', "Working on compressing recurrence");
    Recur := subs([seq(f(x-m*i)=F(x-i),i=0..nops(rhs(recur)))],recur);
    Recur := subs(f = 0 ,Recur);
    Recur := subs(F = f ,Recur);

    Init := map(proc(x, mm, q, init) local i;
        subs([seq(i=(i-q)/mm, i=0..nops(init))],lhs(x)) = rhs(x);
        end, init, m, q, init);
    Init := simplify(Init);
    Init := select(proc(eq) type(op([1,1],eq), integer) end, Init);

    RETURN(clean(Recur, f, x, Init));
end:

# egf/uncompress
#     Uncompress an e.g.f. by $m$ at $q$
# Input: e.g.f., $m$, $q$
# Output: e.g.f.
# Reference: Section 4.5.
`egf/uncompress` := proc(recur, f, var, init, m, q)

    local i, egf, Init, F, j;

    egf := [clean(recur,f,var,init)];

    userinfo(3, 'MS', "Working on uncompressing recurrence");
    egf[1] := subs([seq(var-i=var-m*i,i=1..`egf/metric/P`(op(egf)))],egf[1]);

    Init := [];
    for j from 0 to nops(egf[4])-1 do
        Init := [op(Init),seq(F(i+j*m)=0,i=0..q-1), F(q+j*m)=f(j),
            seq(F(i+j*m)=0,i=q+1..m-1)]
    od;
    Init := subs(egf[4],Init);
    Init := subs(F=f,Init);

    RETURN(clean(egf[1], egf[2], egf[3], Init));
end:

# egf/init
#    Determine the first values up to $N$ of the
#    function for every $m$th value starting at $q$.
# Input: e.g.f., N, m, q
# Output: list
# Reference: NONE
`egf/init` := proc(recur, f, var, init, N, m, q)
    local b, Init, i, s;
    userinfo(4,'MS',"Find initial values for a recurrence");

    if not type(init[1],`=`) then RETURN(init); fi;

    b := `egf/makeproc`(recur, f, var, init);

    if nargs > 5 then
        Init := [seq(seq(f(m*i+s)=Heaviside(s-q+1/2)*
                 Heaviside(q-s+1/2)*b(m*i+s),s=0..m-1),i=0..N)];
    else
        Init := [seq(f(i)=b(i),i=0..N)];
    fi;

    RETURN(expand(radnormal(Init)));
end:

# egf/result
#    Determine the resultant of two e.g.f.'s.
# Input: e.g.f. 1, e.g.f. 2
# Output: e.g.f.
# Reference: NONE
`egf/result` := proc(recur1, f1, x1, init1, recur2, f2, x2, init2)
    local poly1, poly2, y, poly, rec, init, Init, init3, i, InitT, j, g;

    userinfo(5,'MS', "Finding Recurrence for P^{f g} given P^f and P^g");

    poly1 := convert_poly(recur1, f1, x1, init1);
    poly2 := convert_poly(recur2, f2, x2, init2);

    y := `egf/result/variablename/y`;
    poly := resultant(subs(x1=x1-y,poly1),subs(x2=y,poly2),y);
    poly := expand(poly);
    poly := radnormal(poly);
    poly := expand(poly);
    rec := convert_rec(poly, f1, x1);

    userinfo(5,'MS', "Finding inital values for P^{f g} given P^f and P^g");
    g := `egf/result/procname/g`:
    init3 := subs(f2=g, init2);
```

```
    Init := [];
    for i from 0 to min(nops(init1),nops(init2))-1 do
        InitT :=  add(f1(j)*g(i-j)*binomial(i,j),j=0..i):
        Init := [op(Init), f1(i) = expand(subs([op(init1), op(init3)], InitT))];
    od;
    init := (expand(radnormal(Init)));

    RETURN(clean(rec, f1, x1, init));
end:

# egf/strip
#    Remove extrenous zeros from e.g.f.
# Input: e.g.f.,
# Output: e.g.f.,
# Reference: Appendix A.2.2.
`egf/strip` := proc(rec, f, x, init, m, q)
    local Init, i;

    Init := NULL;
    for i in init do
        if (op([1,1], i) = q) mod m then
            Init := Init, i;
        fi;
    od;

    Init := [Init];
    RETURN(rec, f, x, Init);
end:

savelib(`egf/ms`, `egf/ms.m`);
savelib(`egf/ms/result`, `egf/ms/result.m`);
savelib(`egf/ms/rec`, `egf/ms/rec.m`);
savelib(`egf/ms/rec/multi`, `egf/ms/rec/multi.m`);
savelib(`egf/ms/linalg`, `egf/ms/linalg.m`);
savelib(`egf/ms/compress`, `egf/ms/compress.m`);
savelib(`egf/ms/linalg`, `egf/ms/linalg.m`);
savelib(`egf/ms/naive`, `egf/ms/naive.m`);
savelib(`egf/clean`, `egf/clean.m`);
savelib(`egf/strip`, `egf/strip.m`);
savelib(`egf/makeproc`, `egf/makeproc.m`);
savelib(`egf/makeproc2`, `egf/makeproc2.m`);
savelib(`egf/scale`, `egf/scale.m`);
savelib(`egf/compress`, `egf/compress.m`);
savelib(`egf/uncompress`, `egf/uncompress.m`);
savelib(`egf/init`, `egf/init.m`);
savelib(`egf/result`, `egf/result.m`);
```

E.5 Denominator.

File name: Bottom.

```
## Notation:
## m.s.   = multisection
## r.p.e. = rational poly-exponential function
## p.e.   = poly-exponential function
## e.g.f. = exponential generating function

macro(`Fac` = readlib(``bottom/ms/linalg/fft2/factorial``),
      ifactors = readlib(ifactors),
      `Expand` = readlib(`bottom/ms/linalg/fft2/expand`),
      `egf/clean` = readlib(`egf/clean`),
      `egf/init` = readlib(`egf/init`),
      `egf/result` = readlib(`egf/result`),
      `egf/ms/rec/multi` = readlib(`egf/ms/rec/multi`),
      `egf/scale` = readlib(`egf/scale`));

# bottom/ms/naive
#     M.s. the bottom of a r.p.e. using the naive method
#     of using the product as given in Lemma 3.1.
# Input: p.e., m
# Output: e.g.f.
# Reference: Lemma 3.1.
# Description Appendix A.5.1.

`bottom/ms/naive` := proc(pe, f, var, m)
    local omega, egf, pe_m, k;

    userinfo(1, 'MS', "Using naive method to find exponential generating".
        " function");
    omega := (k,m) -> exp(2*Pi*I*k/m);

    # Ref Lemma 3.1.
    pe_m := (product(subs(var=var*omega(k,m),pe),k=1..m));
    egf := convert_egf(pe_m, f, var);
    RETURN(`egf/clean`(egf));
end:

# bottom/ms/linalg/fft
#     M.s. the bottom of a r.p.e. using a combination of
#     linear algebra and the \fft\ method of fast
#     polynomial multiplication. N is the size of
#     the recurrence (less gaps). So (exp(x)-1), x, 8
#     would use an N of 10.
# Input: p.e., m, (optional) N
# Output: e.g.f.
# Reference: Subsection 5.2.1
# Description Appendix A.5.2.
`bottom/ms/linalg/fft` := proc(pe, f, var, m, N)
    local p, d, Poly, poly, FF, initial, i, rec, C, M, Zero;

    # Ref Lemma 2.5
    if nargs = 5 then
        M := N*m;
    else
        M := `pe/metric/P`(pe,var)^m+(m-1)*(`pe/metric/d`(pe,var)+1);
    fi;

    userinfo(1, 'MS', "Finding polynomial approximation for the
        poly-exponential function of degree", 2*M+1);
    Poly := (2*M)!*(convert(taylor(pe,var=0,2*M+1),polynom));

    d := 1;

    # Ref: Subsection 5.2.1.
    userinfo(1, 'MS', "Using fft to find a poly approx for the ".
        "bottom for the given poly-exponential function");
    while m <> d do
        p := ifactors(m/d)[2][1][1];
        d := d * p;

        userinfo(2, 'MS', "Dealing with primative", d, "roots of unity");
        for i from 0 to p-1 do
            poly[i] := subs(var=var*(-1)^(2*i/d),Poly);
        od;
        Poly := poly[0];
        for i from 1 to p-1 do
            if M > 250 then
                Poly := Expand(Poly, poly[i], var, m, 2*M+1)/(2*M)!;
            else
                Poly := convert(series(expand(Poly* poly[i]),var,2*M+1),
                    polynom)/(2*M)!: fi;
        od;
        Poly := radnormal(Poly);
    od;

    Poly := Poly /(2*M)!;

    Zero := `pe/metric/d`(pe, var)+1;
    for i from m*ceil(Zero*p/m) to 2*M by m do
        C[i/m-ceil(Zero*p/m)+1] := coeff(Poly,var,i)*i!;
    od;

    userinfo(1, 'MS', "Using linear algebra to determine recurrence");
```

```
    # Ref: Section 4.3.
    rec := `recurrence/solve/linalg`(C, f, var, m);

    FF := proc(i, m, q, Poly)
        if (i = q) mod m then
            RETURN(coeff(Poly,var,i)*i!);
        else
            RETURN(0);
        fi;
    end;

    initial := [seq(f(i)=FF(i, m, 0, Poly), i=0..M - 1)];

    RETURN(`egf/clean`(rec, f, var, initial));
end:

# bottom/ms/linalg/sym
#     M.s. the bottom of a r.p.e. using a combination of
#     linear algebra and symbolic differentiation.
#     N is the size of the recurrence (less gaps).
#     So (exp(x)-1), x, 8 would use an N of 10.
# Input: p.e., m, (optional) N
# Output: e.g.f.
# Reference: Section 4.4.
# Description Appendix A.5.3.

`bottom/ms/linalg/sym` := proc(pe, f, var, m, N)
    local i, egf, Pe, NN;

    # Ref Lemma 3.1.
    userinfo(1,'MS',"Taking the product of the poly-exponential function".
        " symbolically");
    Pe := expand(product(subs(var=var*exp(2*Pi*I*i/m), pe),i=1..m));

    if nargs = 5 then
        egf := `pe/ms/linalg/sym`(Pe, f, var, m, 0, N);
    else
        NN := `pe/metric/P`(Pe, var);
        egf := `pe/ms/linalg/sym`(Pe, f, var, m, 0, ceil(NN/m));
    fi;
    RETURN(`egf/clean`(egf));
end:

# bottom/ms/result
#     M.s. the bottom of a r.p.e. using a resultant
#     methods on the recurrence polynomial
#     This will give a valid recurrence relation,
#     although not necessarily minimal
# Input: p.e., m
# Output: e.g.f.
# Reference: Section 5.1.
# Description Appendix A.5.4.
`bottom/ms/result` := proc(pe, f, var, m)
    local Recur, recur, p, d, init, i, Init, size, egf, degr;

    d := 1;

    userinfo(1, 'MS', "Finding recurrision of the poly-exponential function");

    egf := [convert_egf(pe, f, var)];
    Recur := egf[1];
    Init := egf[4];


    # Ref: Section 5.1.
    userinfo(1, 'MS', "Using resultant to find a recursion for the ".
                      "bottom for the given poly-exponential function");

    while m <> d do
        p := ifactors(m/d)[2][1][1];
        d := d * p;

        userinfo(2, 'MS', "Dealing with primative", d, "roots of unity");
        size := `egf/metric/P`(Recur, f, var, Init);
        Init := `egf/init`(Recur, f, var, Init, size * m , 1, 0);

        for i from 0 to p-1 do
            recur[i] := `egf/scale`(Recur, f, var, Init, (-1)^(2*i/d));
            init[i] := recur[i][4];
            recur[i] := recur[i][1];
        od;
        Recur := recur[0];
        Init := init[0];
        for i from 1 to p-1 do
            Recur := `egf/result`(Recur, f, var, Init,
                    recur[i], f, var, init[i]);
            Init := Recur[4];
            Recur := Recur[1];
            userinfo(3,'MS',`Recur & Init are`, Recur, Init, i);
        od;
    od;
    size := `egf/metric/P`(Recur, f, var, Init);
    Init := `egf/init`(Recur, f, var, Init, size, 1, 0);
    egf := Recur, f, var, Init;

    RETURN(`egf/clean`(egf));
end:

# bottom/ms/linalg/fft2/factorial
#     This will compute the factorial of a value in a recurrrse manner.
#     It will compute this faster than the kernel level factorial in
#     maple, (which is a major bug in maple).
#     To do this, it will store every 100th value, as computed, (so
#     1% of the information calculated is remember, we don't want much
#     more than this for memory reasons.)
#     It will act recurrsively, with jumps of either 1 or 10, as required.
# Input: n
# Output: n!
`bottom/ms/linalg/fft2/factorial` := proc(n)
    option system;
    local A;
    if n < 100 then RETURN (n!) elif (n = 0) mod 10 then
        A := ((n^10-45*n^9+870*n^8-9450*n^7+63273*n^6-269325*n^5+
             723680*n^4-1172700*n^3+1026576*n^2-362880*n)*`procname`(n-10));
        if (n=0) mod 100 then
            `procname`(n) := A;
        fi:
        RETURN(A);
    else
        RETURN(`procname`(n-1)*n);
    fi;
end:

# bottom/ms/linalg/fft2
#     M.s. the bottom of a r.p.e. using a combination of
#     linear algebra and the \fft\ method of fast
#     polynomial multiplication. After the multiplication
#     to get $\prod f(x \omega_m^-d i})$, we use linalg
#     to determine the new recurrence, and then recompute
#     the new polynomial to the required length.
#     This will cut down on the initial polynomial size.
# Input: p.e., m
# Output: e.g.f.
# Reference: Subsection 5.2.2.
# Description Appendix A.5.2.
`bottom/ms/linalg/fft2` := proc(pe, f, var, m, Factors, Sym, Deg)
    local p, d, Poly, poly, i, rec, C, M, T, egf, size, Zero, MM, MMM, Poly2,
        deg, sym, sym2, fact;

    egf := convert_egf(pe, f, var): size := `egf/metric/P`(egf);

    if nargs >= 6 then
        sym := Sym;
    else
        sym := 1;
    fi;
```

```
if nargs >= 7 then
    deg := copy(Deg);
fi;

if nargs >= 5 then
    fact := Factors;
else
    fact := ifactors(m);
    fact := fact[2];
    fact := map(x->(x[1]$(x[2])),fact);
fi;

userinfo(1, 'MS', "Using fft to find a poly approx for the ".
    "bottom for the given poly-exponential function");

# Ref: Subsection 5.2.2.
d := 1;
sym2 := 1;
while m <> d do
    p := fact[1];
    fact := fact[2..-1];
    d := d * p;

    if (sym = 0) mod p then
        userinfo(2, 'MS', "Skipping primative ". d. "th roots of unity".
            " cause of symmetry");
        sym := sym / p;
        sym2 := sym2 * p;
        next;
    fi;

    userinfo(2, 'MS', "Dealing with primative ". d. "th roots of unity");

    # Ref: Lemma 2.5.
    if nargs >= 7 then
        M := deg[1];
        deg := deg[2..-1];
    else
        M := (size^p) + p*(`egf/metric/d`(egf)+1);
    fi;
    T := `egf/makeproc`(egf);

    userinfo(3, 'MS', "Determining polynomial to degree", 2*M,
        "Every", d/p, "term is present");


    Poly := 0: MM := Fac(2*M):
    MMM := MM:
    for i from 0 to floor(2*M/d*sym2*p) do
        Poly := Poly + T(d*i/p/sym2)*var^(d*i/p/sym2)*MM;
        MM := MM/product(d/p/sym2*i+j,j=1..d/p/sym2);
        if (i = 0) mod 10 then
            userinfo(6, 'MS', "Determined ", i*d/p/sym2, "term.");
        fi;
    od:

    userinfo(5, 'MS', "Scaling polynomials");
    # for i from 0 to p-1 do
    # poly[i] := subs(var=var*(-1)^(2*i/d),Poly);
    # od;

    userinfo(5, 'MS', "Multiplying the polynomials together");
    Poly2 := subs(var=var*(-1)^(2*(p-1)/d),Poly):
    for i from p-2 to 0 by -1 do

        userinfo(5, 'MS', "Scaling polynomials");
        poly := subs(var=var*(-1)^(2*i/d),Poly);

        if M > 250 then
            Poly2 := Expand(Poly2, poly, var, m*d/p, 2*M+1)/MMM;
        else
            Poly2 := convert(series(expand(Poly2 * poly),var,2*M+1),
                polynom)/MMM;
        fi;

        userinfo(6, 'MS', "Multiplied the ".i."th polynomial in");
        Poly2 := radnormal(Poly2):
        userinfo(6, 'MS', "Normalized the polynomial");
    od;
    Poly := Poly2/MMM;

    Poly2 := `Poly2`:
    Poly := radnormal(Poly);

    userinfo(3, 'MS', "Determining coefficents from polynomial");
    Zero := `egf/metric/d`(egf)+1;
    for i from d/sym2*ceil(Zero*p/d) to 2*M by d/sym2 do
        C[i/d*sym2-ceil(Zero*p/d)+1] := coeff(Poly,var,i)*Fac(i);
    od;

    userinfo(3, 'MS', "Determining recurrence for polynomial with linalg");
    rec := `recurrence/solve/linalg`(C, f, var, d/sym2);#, "toeplitz");


    egf := rec, f, var, [seq(f(i)=coeff(Poly,var,i)*Fac(i), i=0..M - 1)];
    size := `egf/metric/P`(egf): C := 'C';
od;

RETURN(`egf/clean`(rec, f, var,
    [seq(f(i)=coeff(Poly,var,i)*Fac(i), i=0..size - 1)]));
end:

# bottom/ms/factor
#     M.s. the bottom using any method mentioned, but factors out
#     any polynomials first, which it returns as a last argument
# Input: p.e., m, method[methodarg]
# Output: e.g.f., scale
`bottom/ms/factor` := proc(pe, f, var, m, opt)
    local i, method, methodarg, egf, Pe, Poly, j;

    userinfo(1, 'MS', "Removing common polynomials before determining".
        " exponential generating function");
    if nargs = 5 then
        if type(opt, indexed) then
            method := `bottom/ms/`.(op(0, opt));
            methodarg := op(1, opt);
        else
            method := `bottom/ms/`.opt;
        fi;
    else
        method := `bottom/ms/linalg/fft2`;
    fi;

    Pe := factor(pe);
    if type(Pe,`*`) then
        Poly := select(x->(type(x,polynom(anything,var))),[op(Pe)]);
        Pe := select(x->(not type(x,polynom(anything,var))),[op(Pe)]);
        Poly := mul(j,j=Poly);
        Pe := mul(j,j=Pe);
    else
        if type(Pe,polynom(anything,var)) then
            Poly := Pe;
        else
            Poly := 1;
        fi;
    fi;

    if assigned(methodarg) then
        egf := method(Pe,f,var,m,methodarg);
    else
        egf := method(Pe,f,var,m);
    fi;

    Poly := `egf/ms/rec/multi`(Poly,var,m,1):
    RETURN(`egf/clean`(egf), Poly);
end:
```

```
# bottom/ms
#     M.s. the bottom of the r.p.e. with a p.e. bottom by m
# Input: p.e., m, method[methodarg]
# Output: e.g.f.
`bottom/ms` := proc(pe, f, var, m, opt)
    local i, method, methodarg, egf;

    userinfo(1, 'MS', "Dealing with the bottom of the r.p.e.");
    if nargs = 5 then
        if type(opt, indexed) then
            method := `bottom/ms/`.(op(0, opt));
            methodarg := op(1, opt);
        else
            method := `bottom/ms/`.opt;
        fi;
    else
        method := `bottom/ms/linalg/fft2`;
    fi;

    if assigned(methodarg) then
        egf := method(pe,f,var,m,methodarg);
    else
        egf := method(pe,f,var,m);
    fi;

    RETURN(`egf/clean`(egf));
end:

# bottom/ms/linalg/fft2/expand
# Expands the product of two polynomials. Attempts to use
# less memory than the maple kernal equivalent.
# It will look at the different components of the polynomial,
# where the degree falls into different residuals modulo omega.

# Input: poly1, poly2, var, omega, cutoff
# Output: poly1*poly2
`bottom/ms/linalg/fft2/expand` := proc(poly1, poly2, var, omega, cutoff)
    local p1, p2, y, i, j, p, Poly, A, T;

    for i from 0 to omega-1 do
        p1[i mod omega] := 0:
        p2[i mod omega] := 0: i
    od:

    for i from 0 to omega - 1 do
        userinfo(6,'MS',"Got information for omega ".i.".");
        p1[i mod omega] := add(var^(omega*j + i)*coeff(poly1, var, omega*j+i),
            j= 0...ceil(cutoff/omega)+1);
        p2[i mod omega] := add(var^(omega*j + i)*coeff(poly2, var, omega*j+i),
            j=0...ceil(cutoff/omega)+1);
    od;

    for i from 0 to omega - 1 do
        p[i] := 0:
    od:

    for i from 0 to omega - 1 do
        for j from 0 to omega - 1 do
            userinfo(6,'MS',"Dealing with p1[".i."], and p2[".j."]");
            if nargs = 5 then
                p[(i+j) mod omega] :=
                    p[(i+j) mod omega] +
                    convert(series(expand(p1[i]*p2[j]),var,cutoff+1),polynom);
            else
                 p[(i+j) mod omega] :=
                    p[(i+j) mod omega] + expand(p1[i]*p2[j]);
            fi;
        od;
    od;

    userinfo(6,'MS',"Adding back together"):
    Poly := add(p[i],i=0..omega-1);

    RETURN(Poly):
end:

#libname := libname[3], libname[1..2]:
savelib(`bottom/ms/naive`, `bottom/ms/naive.m`);
savelib(`bottom/ms/linalg/fft`, `bottom/ms/linalg/fft.m`);
savelib(`bottom/ms/linalg/sym`, `bottom/ms/linalg/sym.m`);
savelib(`bottom/ms/result`, `bottom/ms/result.m`);
savelib(`bottom/ms/linalg/fft2`, `bottom/ms/linalg/fft2.m`);
savelib(`bottom/ms/linalg/fft2/expand`, `bottom/ms/linalg/fft2/expand.m`);
savelib(`bottom/ms/factor`,`bottom/ms/factor.m`);
savelib(`bottom/ms`,`bottom/ms.m`);
savelib(`bottom/ms/linalg/fft2/factorial`,`bottom/ms/linalg/fft2/factorial.m`);
```

E.6 Numerator.

File name: Top.

```
## Notation:
## m.s.   = multisection
## r.p.e. = rational poly-exponential function
## p.e.   = poly-exponential function
## e.g.f. = exponential generating function

macro(`egf/clean` = readlib(`egf/clean`),
      `egf/result` = readlib(`egf/result`),
      `egf/scale` = readlib(`egf/scale`),
      `egf/init` = readlib(`egf/init`),
      `egf/ms/rec/multi` = readlib(`egf/ms/rec/multi`));

# top/ms/naive
#     M.s. the top of the r.p.e. using the naive method.
# Input: p.e. (top), p.e. (bottom), m, q
# Output: e.g.f.
# References: Lemma 3.1.
#             Appendix A.6.1.
`top/ms/naive` := proc(top, bot, f, var, m, q)
    local omega, egf, pe_2, k;

    userinfo(1,'MS',"Using naive method to find exponential ".
       "generating function");

    # Ref Lemma 3.1.
    pe_2 := (top*product(subs(var=var*(-1)^(2*k/m),bot),k=1..m-1));
    egf := `pe/ms/naive`(pe_2, f, var, m, q);
    RETURN(`egf/clean`(egf));
end:

# top/ms/linalg/fft
#     M.s. the top of a r.p.e. using a combination of
#     linear algebra and the \fft\ method of fast
#     polynomial multiplication.  N is the size of
#     the recurrence (less gaps).  So (exp(x)-1), x, x, 8
#     would use an N of 20.
# Input: p.e. (top), p.e. (bottom), m, (optional) N
# Output: e.g.f.
# Reference: Section 5.2.
#            Appendix A.6.2.
`top/ms/linalg/fft` := proc(top, bot, f, var, m, q, N)

    local Poly, poly, FF, initial, i, rec, C, M, Zero;

    # Ref Lemma 3.6.
    Zero := `pe/metric/d`(top,var)+`pe/metric/d`(bot,var)*(m-1)+1;
    if nargs = 7 then
        M := N*m;
    else
        M := m*(`pe/metric/P`(top,var)+1)*(`pe/metric/P`(bot,var)+1)^(m-1)+Zero;
    fi;

    userinfo(1, 'MS', "Finding polynomial approximation for the pe of size",
```

```
        2*M+Zero);
    Poly := (2*M+Zero)!*(convert(taylor(bot,var=0,2*M+Zero+1),polynom));

    poly := (2*M+Zero)!*convert(taylor(top,var=0,2*M+Zero+1),polynom);

    # Ref: Section 5.2.
    userinfo(1, 'MS', "Using fft to find a poly approx for the ".
                      "top for the given pe");
    for i from 1 to m-1 do
        poly := convert(series(expand(poly *
            subs(var=var*(-1)^(2*i/m),Poly)),var,2*M+Zero),polynom)/(2*M+Zero)!;
#        poly := convert(series(expand(poly *
#            subs(var=var*exp(2*Pi*I*i/m),Poly)),var,2*M), polynom)/(2*M)!;
    od;

    poly := radnormal(poly / (2*M+Zero)!);

    for i from q+m*ceil(Zero/m) to 2*M by m do
        C[i/m-ceil(Zero/m)-q/m+1] := coeff(poly,var,i)*i!;
    od;
#    for i from Zero to 2*M by m do
#        C[i-Zero+1] := coeff(poly,var,i)*i!;
#    od;

    userinfo(1, 'MS', "Using linear algebra to determine recurrence");
    rec := `recurrence/solve/linalg`(C, f, var, m);

    FF := proc(i, m, q, poly)
        if (i = q) mod m then
            RETURN(coeff(poly,var,i)*i!)
        else
            RETURN(0);
        fi;
    end;

    initial := [seq(f(i)=FF(i, m, q, poly), i=0..M - 1 + q + Zero)];

    RETURN(`egf/clean`(rec, f, var, initial));
end:


# top/ms/linalg/sym
#     M.s. the top of a r.p.e. using a combination of
#     linear algebra and symbolic differentiation.
#     N is the size of the recurrence (less gaps).
#     So (exp(x)-1), x, x, 8 would use an N of 20.
# Input: p.e., m, (optional) N
# Output: e.g.f.
# Reference: Section 4.3.
#            Appendix A.6.3.
`top/ms/linalg/sym` := proc(top, bot, f, var, m, q, N)
    local i, egf, Pe;

    # Ref: Lemma 3.1.
    userinfo(1,'MS',"Taking the product of the poly-".
        "exponential functions symbolically");
    Pe := expand(product(subs(var=var*exp(2*Pi*I*i/m), bot),i=1..m-1)*top);
#    Pe := expand(product(subs(var=var*(-1)^(2*i/m), bot),i=1..m-1)*top);

    if nargs = 7 then
        egf := `pe/ms/linalg/sym`(Pe, f, var, m, q, N);
    else
        egf := `pe/ms/linalg/sym`(Pe, f, var, m, q);
    fi;
    RETURN(`egf/clean`(egf));
end:

# top/ms/result
#     M.s. the top of a r.p.e. using a resultant
#     methods on the recurrence polynomial
#     This will give a valid recurrence relation,
#     although not necessarily minimal
# Input: p.e. (top), p.e. (bottom), m
# Output: e.g.f.
# Reference: Section 5.1.
#            Appendix 6.6.
`top/ms/result` := proc(top, bot, f, var, m, q)

    local RecurB, recur,
        p, d, poly, FF, init, i, rec, C, InitB, size, egf, egfB,
        recurB, initB, Size;

    d := 1;

    userinfo(1, 'MS', "Finding recurrision of the top and bottom");
    egfB := [convert_egf(bot, f, var)];
    egf := [convert_egf(top, f, var)];
    recur := egf[1];
    init := egf[4];
    RecurB := egfB[1];
    InitB := egfB[4];
    Size := `egf/metric/P`(op(egfB));

    # Ref: Section 5.1.
    userinfo(1, 'MS', "Using resultant to find a recursion for the ".
                      "top for the given poly-exponential functions");
    for d from 1 to m-1 do
        size := `egf/metric/P`(recur, f, var, init) * Size;
        init := `egf/init`(recur, f, var, init, size, 1, 0);
        recurB := `egf/scale`(RecurB, f, var, InitB, (-1)^(2*d/m));
        initB := recurB[4];
        recurB := recurB[1];
        initB  := `egf/init`(recurB, f, var, initB, size, 1, 0);
        initB  := radnormal(initB);
        recur := `egf/result`(recurB, f, var, initB, recur, f, var, init);
        init := recur[4];
        recur := recur[1];
        init := map(radnormal,init);
    od;

    size := `egf/metric/P`(recur, f, var, init);
    init := `egf/init`(recur, f, var, init, size, 1, 0);

    egf := `egf/ms/rec`(recur, f, var, init, m, q);

    egf := op(radnormal([egf]));

    RETURN(`egf/clean`(egf));
end:

# top/ms/linalg/know
#     M.s. the top of a r.p.e. using a combination of
#     linear algebra and knowledge about the bottom, and actual
#     recurrence
#     N is the size of the recurrence (less gaps).
#     zero is the number of bad initial values to skip (defaults to 2)
# Input: proc (bot), proc (actual), m, N, (optional) zero
# Output: e.g.f.
# Reference: Section 5.3.
#            Appendix A.6.5.
`top/ms/linalg/know` := proc(botP, actP, f, var, m, q, N, zero, shift)
    local i, egf, Pe, Zero, j, temp, C, rec, initial, Shift;

    if nargs >= 9 then
        Shift := shift;
    else
        Shift := 0;
    fi;

    if nargs >= 8 then
        Zero := zero;
    else
        Zero := 2;
    fi;

    initial := [seq(f(i)=0,i=0..Shift-1)];
    userinfo(1, 'MS', "Determining top values");
    for i from Shift to 2 * N *m + Zero do
```

```
        j := 'j':
        if (i = q+Shift) mod m then
            temp := add(binomial(i, q+j*m)*actP(m*j+q)*botP(i-q-j*m),
                j=0..(i-q)/m);
        else
            temp := 0;
        fi;
        if (i = 0) mod 10 then
            userinfo(2, 'MS', "Determining value ".i);
        fi;
        if i > Zero and (i = q+Shift) mod m then
            C[(i-q-Shift-ceil((Zero-q-Shift+1)/m)*m)/m+1] := temp;
        fi;
        initial := [op(initial),f(i)=temp];
    od;

    userinfo(1, 'MS', "Using linear algebra to determine recurrence");
    rec := `recurrence/solve/linalg`(C, f, var, m);#, "toeplitzf");

    egf := rec, f, var, initial;

    RETURN(`egf/clean`(egf));
end:

# top/ms/factor
#     M.s. the top using any method mentioned, but factors out
#     any polynomials first, which it returns as a last argument
# Input: p.e. (top), p.e. (bot), m, q, method[methodarg]
# Output: e.g.f., scale
`top/ms/factor` := proc(top, bot, f, var, m, q, opt)
    local i, method, methodarg, egf, Pe, Poly, j, Top, PolyT, Bot,
        PolyB, T, g, B, newq;

    userinfo(1,'MS',"Removing common polynomials before determining ".
        "exponential generating function");
    if nargs = 7 then
        if type(opt, indexed) then
            method := `top/ms/`.(op(0, opt));
            methodarg := op(1, opt);
        else
            method := `top/ms/`.opt;
        fi;
    else
        method := `top/ms/linalg/fft`;
    fi;

    Top := factor(top);
    if type(Top,`*`) then
        PolyT := select(x->(type(x,polynom(anything,var))),[op(Top)]);
        PolyT := mul(j,j=PolyT);
    else
        if type(Top,polynom(anything,var)) then
            PolyT := Top;
        else
            PolyT := 1;
        fi;
    fi;

    Bot := factor(bot);

    if type(Bot,`*`) then
        PolyB := select(x->(type(x,polynom(anything,var))),[op(Bot)]);
        PolyB := mul(j,j=PolyB);
    else
        if type(Bot,polynom(anything,var)) then
            PolyB := Bot;
        else
            PolyB := 1;
        fi;
    fi;

    T := product(subs(var=var*(-1)^(2*i/m),PolyB),i=1..(m-1))*PolyT;
    T := simplify(T):
    PolyT := simplify(PolyT):
    PolyB := simplify(PolyB):
    g := T;
    for i from 1 to m-1 do
        g := gcd(g, simplify(subs(var=var*(-1)^(2*i/m), T)));
        g := simplify(g):
        if degree(g,var) = 0 then
            g := 1;
            break;
        fi;
    od;

    PolyT := gcd(PolyT, g);

    T := `egf/ms/rec/multi`(PolyB,var,m,1);
    T := gcd(T,g):

    PolyB := quo(T, simplify(g/PolyT), var):

    Bot := Bot/PolyB;
    Top := Top/PolyT;

    if type(g, `+`) then
        if nops({op(map(x->x mod m, map(degree,[op(randpoly(x))])))}) = 1 then
            newq := (q-degree(g,var)) mod m;
        else
            newq := "all";
        fi;
    else
        newq := (q-degree(g,var)) mod m;
    fi;

    if assigned(methodarg) then
        egf := method(Top,Bot,f,var,m,newq,methodarg);
    else
        egf := method(Top,Bot,f,var,m,newq);
    fi;

    RETURN(`egf/clean`(egf), g);
end:

# top/ms
#     M.s. the top of the r.p.e. by m
# Input: p.e. (top), p.e. (bot) m, method[methodarg]
# Output: e.g.f.
`top/ms` := proc(top, bot, f, var, m, q, opt)
    local i, method, methodarg, egf;

    userinfo(1, 'MS', "Dealing with the bottom of the rational ".
        "poly-exponential function");

    if nargs = 7 then
        if type(opt, indexed) then
            method := `top/ms/`.(op(0, opt));
            methodarg := op(1, opt);
        else
            method := `top/ms/`.opt;
        fi;
    else
        method := `top/ms/linalg/fft`;
    fi;

    if assigned(methodarg) then
        egf := method(top, bot, f, var, m, q, methodarg);
    else
        egf := method(top, bot, f, var, m, q);
    fi;

    RETURN(`egf/clean`(egf));
end:

# top/ms/know
#     M.s. the top of a r.p.e. using knowledge about the bottom, and actual
#     values, and the recurrence
#     N is the size of the recurrence (less gaps).
```

```
# Input: recurrence, proc (bot), proc (actual), m, N
# Output: e.g.f.
# Reference: Section 5.3.
#            Appendix A.6.6.
`top/ms/know` := proc(rec, botP, actP, f, var, m, q, N)
    local C, init, egf, i, mi, qi, j:

    C := (i, mi, qi) -> add(binomial(i, qi+j*mi)*actP(mi*j+qi)*botP(i-qi-j*mi),
                  j=0..(i-qi)/mi);

    userinfo(2, 'MS', "Getting initial values");
    init := [seq(f(i) = C(i, m, q), i = 0 .. N*m)];

    egf := `egf/clean`(rec, f, var, init);

    RETURN(egf);
end:

#libname := libname[3], libname[1..2]:
savelib(`top/ms/naive`, `top/ms/naive.m`);
savelib(`top/ms/linalg/fft`, `top/ms/linalg/fft.m`);
savelib(`top/ms/linalg/sym`, `top/ms/linalg/sym.m`);
savelib(`top/ms/result`, `top/ms/result.m`);
savelib(`top/ms/linalg/know`, `top/ms/linalg/know.m`);
savelib(`top/ms/factor`, `top/ms/factor.m`);
savelib(`top/ms`, `top/ms.m`);
savelib(`top/ms/know`, `top/ms/know.m`):
```

E.7 Linear Algebra.

File name: Linalg.

```
macro(linsolve = readlib(linalg)[linsolve],
      rDot     = readlib(`recurrence/solve/toeplitz/rdot`),
      HankelSolver = readlib(`recurrence/solve/hankel/solver`),
      Rev      = readlib(`recurrence/solve/toeplitz/rev`));

# recurrence/solve/linalg
#     Solves the recurrence relationship given the first
#     few initial values.  The recurrence relationship returned
#     will be using the function and variable given.
# Input: Value, fun, var, m
# Output: Recurrence relationship
# References: Section 4.3
`recurrence/solve/linalg` := proc(Value, fun, var, m, toe)
    local i, j, N, C, b, ans, rec;

    save Value, "Value".m."Problem";

    if true then #nargs=5 and toe = "hankel" then
        RETURN(readlib(`recurrence/solve/hankel`)(Value, fun, var, m));
    elif nargs=5 and toe = "toeplitz" then
        RETURN(readlib(`recurrence/solve/toeplitz`)(Value, fun, var, m));
    elif nargs=5 and toe = "toeplitzf" then
        RETURN(readlib(`recurrence/solve/toeplitzf`)(Value, fun, var, m));
    fi;

    userinfo(3, 'MS', "Using linear algebra to determine the recurrence");
    if type(Value,table) then
        N := floor(nops(op([1,2],Value))/2);
    elif type(Value,list) then
        N := floor(nops(Value)/2);
    fi;

    userinfo(4, 'MS', "Finding matrix of size ". N. " X ". N."");
    C := matrix(N,N):

    for i from 1 to N do
        for j from 1 to N do
            C[i,j] := Value[i+j-1];
        od;
    od;

    userinfo(4, 'MS', "Finding vector of size ". N."");
    b := vector([seq(Value[i+N],i=1..N)]);

    ans := linsolve(C,b);
    ans := convert(ans,list);

    i := 1;
    do
        if has(ans, _t[i]) then
            for j from 1 to N do
                if has(ans[j] , _t[i]) then
                    ans := subs(_t[i] = solve(ans[j], _t[i]),ans);
                    break;
                fi;
            od;
        else
            break;
        fi;
        i := i + 1;
    od;

    rec := fun(var) = add(ans[i]*fun(var-(N+1)*m+i*m),i=1..N);

    userinfo(5, 'MS', "Returing recurrsion"):
    RETURN(rec);
end:


`recurrence/solve/hankel` := proc(Value, fun, var, m)
    local N, H, X, i, rec;
    userinfo(3, 'MS', "Using George's methods algebra to".
        " determine the recurrence");
    if type(Value,table) then
        N := floor((nops(op([1,2],Value))-1)/2);
    elif type(Value,list) then
        N := floor((nops(Value)-1)/2);
    fi;

    H := matrix(N,N+1,[seq(seq(Value[i+j],i=1..N+1),j=1..N)]):

    userinfo(4, 'MS', "Finding matrix of size ". N. " X ". (N+1)."");
    X := HankelSolver(H):

    if abs(X[N+1,1]) <> 1 then print("Something is horribly wrong".
        " 2*N needs to be bigger than ". (2*N));
        RETURN("ERROR");
    fi;

    rec := fun(var) = add(-X[N+1,1]*X[i,1]*fun(var-(N+1)*m+i*m),i=1..N);

    userinfo(5, 'MS', "Returing recurrsion"):
    RETURN(rec);
end:

`recurrence/solve/hankel/solver` := proc(A)
    local i, z, C, F, n;

    n := linalg[rowdim](A);

    C :=series(add(A[1,i]*z^(i-1),i=1..n)+add(A[n,i]*z^(n+i-2),i=2..n+1),z,
        2*n+1);

    F := denom( convert( C, ratpoly, n-1,n ));

    matrix(n+1,1,[seq(coeff(F,z,n-i),i=0..n)]);

end:


# Examples which I ran it on just as a check:
```

```
`recurrence/solve/toeplitz/rdot` := proc(a,b)
    local i, ans, n;
    if a = 0 then RETURN(0); fi;
    n := nops(a);
    ans := 0;
    for i from 1 to nops(a) do
        ans := a[i] * b[1+n-i] + ans;
    od;
end:

`recurrence/solve/toeplitz/rev` := proc(a)
    local i, n, ans;
    if a = 0 then RETURN(0); fi;
    ans := [seq(a[nops(a)+1-i],i=1..nops(a))];
    RETURN(ans);
end:

`recurrence/solve/toeplitz` := proc(Value, fun, var, m)
    local r, s, y, f, g, delta, gamma, N, rp, sp, C, i, j, t, OldN, OldN2,
          ans, rec, Vvalue;

#    save Value, ToeplitzValue.m;

    if type(Value,table) then
        N := nops(op([1,2],Value));
        Vvalue := NULL;
        for i from 1 to N do
            Vvalue := Vvalue, Value[i];
        od;
        Vvalue := [Vvalue];
#        Vvalue := convert(Value, list);
    fi;

    N := floor(nops(Vvalue)/2);
#print("Original N", N);

    OldN2 := N;

    while Vvalue[N] = 0 do N := N-1 od;

    OldN := N:

    #B := matrix(N,N,[seq(seq(A(j-i+N-1),i=0..N-1),j=0..N-1)]);

    t[0] := Vvalue[N];

    userinfo(3, 'MS', "Using toeplitz method to determine the recurrence");
    for j from 1 to N-1 do
        userinfo(4, 'MS', "Setting up ".j."-th term of ".(N-1).".");
        r[(N-j)] := Rev(Vvalue[j .. N-1]);
        s[(N-j)] := Vvalue[N+1 .. 2*N-j];
        rp[j] := Vvalue[N-j];
        sp[j] := Vvalue[N+j];
    od:

    y[0] := 1/t[0];
    f[0] := 0;
    g[0] := 0;

    for i from 0 to N-2 do
        userinfo(4, 'MS', "Solving up ".i."-th problem of ".(N-2).".");
        gamma[i] := y[i] * rp[i+1] + rDot(f[i], r[i]);
        delta[i] := y[i] * sp[i+1] + rDot(g[i], s[i]);
        if (delta[i] * gamma[i] = 1) then
            N := i + 1;
            break;
        fi;
        y[i+1] := y[i] / (1-delta[i] * gamma[i]);
        if i = 0 then
            f[i+1] := y[i+1]/y[i] * [-gamma[i] * y[i]];
            g[i+1] := y[i+1]/y[i] * [-delta[i] * y[i]];
        else
            f[i+1] := y[i+1]/y[i] * [op(f[i] - gamma[i] * Rev(g[i])),
                -gamma[i] * y[i]];
            g[i+1] := y[i+1]/y[i] * [op(g[i] - delta[i] * Rev(f[i])),
                -delta[i] * y[i]];
        fi;
    od:

    C := matrix(N,N);
    C[1,1] := y[N-1];
    for i from 1 to N-1 do
        C[1,i+1] := f[N-1][i];
        C[i+1,1] := g[N-1][i];
    od:
    for i from 1 to (N-2) do
        C[N,i+1] := g[N-1][N-1-i];
        C[i+1,N] := f[N-1][N-1-i];
    od:
    C[N,N] := y[N-1]:
#    print(C);
    for i from 1 to N-2 do
        for j from 1 to N-2 do
            userinfo(4, 'MS', "Finding value for (".i.",".j.")-th entry");
            C[i+1,j+1] := C[i,j] + 1/C[1,1] * (C[i+1,1]*C[1,j+1] -
                C[1,N-i+1] * C[N-j+1,1]);
        od;
    od;

    i := 'i';
#    print(matrix(N,1,[seq(Vvalue[OldN+i],i=1..N)]));
    ans := evalm(C &* matrix(N,1,[seq(Vvalue[OldN2+i],i=1..N)]));
#print("N, OldN, OldN2", N, OldN, OldN2, "ans", ans);

    rec := fun(var) = add(ans[N+1-i,1]*fun(var-((OldN2-OldN)+N+1)*m+i*m),
        i=1..N);
    RETURN(rec);
end:

`recurrence/solve/toeplitzf` := proc(Value, fun, var, m)
    local r, s, y, f, g, delta, gamma, N, rp, sp, C, i, j, t, OldN,
          ans, rec, Vvalue;

#    save Value, ToeplitzfValue.m;

    if type(Value,table) then
        N := nops(op([1,2],Value));
        Vvalue := NULL;
        for i from 1 to N do
            Vvalue := Vvalue, Value[i];
        od;
        Vvalue := [Vvalue];
    fi;
    N := floor(nops(Vvalue)/2);

    OldN := N;

    while Vvalue[N] = 0 do N := N-1 od;


    Digits := ceil(sqrt(N)*max(op(map(x->log[10](abs(x)), Vvalue))));

    Vvalue := map(evalf, Vvalue);


    #B := matrix(N,N,[seq(seq(A(j-i+N-1),i=0..N-1),j=0..N-1)]);

    t[0] := Vvalue[N];

    userinfo(3, 'MS', "Using toeplitz method to determine the recurrence,".
        " with ".Digits." digits accuracy.");
    for j from 1 to N-1 do
        userinfo(4, 'MS', "Setting up ".j."-th term of ".(N-1).".");
        r[(N-j)] := Rev(Vvalue[j .. N-1]);
        s[(N-j)] := Vvalue[N+1 .. 2*N-j];
        rp[j] := Vvalue[N-j];
        sp[j] := Vvalue[N+j];
    od:
```

```
    y[0] := 1/t[0];
    f[0] := 0;
    g[0] := 0;

    for i from 0 to N-2 do
        userinfo(4, 'MS', "Solving up ".i."-th problem of ".(N-2).".");
        gamma[i] := y[i] * rp[i+1] + rDot(f[i], r[i]);
        delta[i] := y[i] * sp[i+1] + rDot(g[i], s[i]);
#print(evalf(delta[i]*gamma[i], 100));
        if (evalf(delta[i] * gamma[i], ceil(Digits/sqrt(N))) = 1.0) then
            N := i + 1;
            break;
        fi;
        y[i+1] := y[i] / (1-delta[i] * gamma[i]);
        if i = 0 then
            f[i+1] := y[i+1]/y[i] * [-gamma[i] * y[i]];
            g[i+1] := y[i+1]/y[i] * [-delta[i] * y[i]];
        else
            f[i+1] := y[i+1]/y[i] * [op(f[i] - gamma[i] * Rev(g[i])),
                -gamma[i] * y[i]];
            g[i+1] := y[i+1]/y[i] * [op(g[i] - delta[i] * Rev(f[i])),
                -delta[i] * y[i]];
        fi;
    od:

    C := matrix(N,N);
    C[1,1] := y[N-1];
    for i from 1 to N-1 do
        C[1,i+1] := f[N-1][i];
        C[i+1,1] := g[N-1][i];
    od:
    for i from 1 to (N-2) do
        C[N,i+1] := g[N-1][N-i-1];
        C[i+1,N] := f[N-1][N-i-1];
    od:
    C[N,N] := y[N-1]:
#    print(C);
    for i from 1 to N-2 do
        for j from 1 to N-2 do
            userinfo(4, 'MS', "Finding value for (".i.",".j.")-th entry");
            C[i+1,j+1] := C[i,j] + 1/C[1,1] * (C[i+1,1]*C[1,j+1] -
                C[1,N-i+1] * C[N-j+1,1]);
        od;
    od;

    i := 'i';
#    print(matrix(N,1,[seq(Vvalue[OldN+i],i=1..N)]));
    ans := evalm(C &* matrix(N,1,[seq(Vvalue[OldN+i],i=1..N)]));

#print(ans);
    ans := map(round,ans);
#print(ans);

    rec := fun(var) = add(ans[N+1-i,1]*fun(var-(N+1)*m+i*m),i=1..N);
    RETURN(rec);
end:

savelib(`recurrence/solve/linalg`, `recurrence/solve/linalg.m`);
savelib(`recurrence/solve/toeplitz/rev`, `recurrence/solve/toeplitz/rev.m`);
savelib(`recurrence/solve/toeplitz/rdot`, `recurrence/solve/toeplitz/rdot.m`);
savelib(`recurrence/solve/toeplitz`, `recurrence/solve/toeplitz.m`);
savelib(`recurrence/solve/hankel`, `recurrence/solve/hankel.m`);
savelib(`recurrence/solve/hankel/solver`, `recurrence/solve/hankel/solver.m`);
savelib(`recurrence/solve/toeplitzf`, `recurrence/solve/toeplitzf.m`);
```

E.8 Performing the calculations.

File name: Normal.

```
# calcul/normal
#     Perform the calculation using normal methods
# Input: Recurrence
# Output: Values
% Refence: Theorem 3.1.
`calcul/normal` := proc(Largest, Top, Bot, m, q, feq, File, Info)
    local i, B, info, Value, j, s, work;

    if nargs = 8 then
        B := copy(Info);
        for i from q to Largest by m do
            if has(B[i] , B) then
                work := i;
                break;
            fi;
        od;
    else
        work := q:
    fi;

    for i from 0 to infinity do
        if Bot(i) <> 0 then
            s := i;
            break;
        fi;
    od;

    for i from work to Largest by m do

        if not has(B[i] , B) then
            userinfo(3, 'MS', "Knew the ". i. "th value already.");
            next;
        fi;

        Value := Top(i+s);

        userinfo(2, 'MS', "Working on problem", i);
        for j from q to i-m by m do
            Value := Value - Bot(s+i-j)*B[j]*binomial(i+s,j);
        od;

        Value := Value / binomial(i+s,s)/Bot(s);

        userinfo(3, 'MS', "Determined ". i. "th value.");
        B[i] := Value;

        if nargs >= 7 then
            if (i = 0) mod feq then
                save B, File.i.`.m`;
            fi;
        fi;
    od;

    RETURN(copy(B));
end:

#libname := libname[3], libname[1..2]:
savelib(`calcul/normal`, `calcul/normal.m`);
```

File name: Multi.

```
macro(binomial = readlib(binomial),
      readpipe = readlib(`calcul/readpipe`),
      writepipe = readlib(`calcul/writepipe`));
```

```
# calcul/balanced/worker
#     The slave that does all the work
# Input: Recurrences
# Output: NOTHING
# Reads: Values of other calculations.
# Writes: Value to calculations performed
# Reference: Section 6.2.
`calcul/balanced/worker` :=
        proc(Largest, N, work, ReadPipe, WritePipe, Top, Bot, m, q, Info)
    local i, B, info, Value, j, s, start, tt;

    tt := time():
    B := copy(Info);

    for i from work to Largest by m*N do
        if has(B[i], B) then
            start := i;
            break;
        fi;
    od;

    for i from 0 to infinity do
        if Bot(i) <> 0 then
            s := i;
            break;
        fi;
    od;

    for i from start to Largest by N*m do

        Value := Top(i+s);
        userinfo(2, 'MS', "Slave", work, "working on problem", i);

        for j from q to max(q-m,i - N*m) by m do
            Value := Value - Bot(s+i-j)*B[j]*binomial(i+s,j);
        od;

        for j from 0 to min(i-m, m*N-2*m) by m do
            userinfo(3, 'MS', "Slave", work, "getting needed info from Master");
            info := NULL:
            while info = NULL do
                info := readpipe(ReadPipe[work]);
            od;
            B[info[1]] := info[2];
        od;

        userinfo(3, 'MS', "Slave", work, "finishing calculation");
        for j from max(q,i - N*m+m) to i-m by m do
            Value := Value - Bot(s+i-j)*B[j]*binomial(i+s,j);
        od;

        Value := Value / binomial(i+s,s)/Bot(s);

        userinfo(3, 'MS', "Slave", work, "Reporting to Master");
        writepipe(WritePipe[work],[i,Value]);
        B[i] := Value;
    od;

    print("Slave ".work." took",(time() - tt), "seconds."):
    RETURN();
end:

# calcul/balanced
#     The form of communication between the workers.
# Input: Recurrences
# Output: Values
# Reads: Values of calculations.
# Writes: Value to calculations.
`calcul/balanced` := proc(N, Largest, Top, Bot, m, q, feq, File, Info)
    local Slaves, Master, i, j, pid, work, info, l, B, start, i2, k;

    if nargs = 9 then
        B := copy(Info);
    for i from q to Largest by m do
        if has(B[i], B) then
            start := i;
            break;
        fi;
    od;
else
    start := q;
fi;

work := q;
for i from q to (N-1)*m+q by m do
    Slaves[i] := pipe();
    Master[i] := pipe();
od;

for i from 1 to N do

    pid := fork();
    if pid = 0 then # Slaves
        userinfo(1, 'MS', "Starting up slave", work);
        readlib(`calcul/balanced/worker`)
                (Largest, N, work, Slaves, Master, Top, Bot, m, q, B);
        system("sleep 1");

        userinfo(1, 'MS', "Stopping slave", work);
        quit;
    elif i = N then # Master
        if start <> q then
            k := 1;
             i := start mod N*m;
#            for i from (start mod N*m) to
#                    (start mod N*m) + (N-1)*m by m do
                for j from i - (N-1)*m to i - m*k by m do
                    userinfo(3, 'MS', "Sending info to slave", i);

#                    info := convert([j,B[j]],string);
                    writepipe(Slaves[(i mod N*m)],[j, B[j]]);
                od;
#                k := k + 1;
#            od;
        fi;

        for j from start to Largest by m do

            ## Get the info from the slaves.
            userinfo(3, 'MS', "Getting information from slave",
                (j) mod N*m );
            info := NULL;
            while info = NULL do
                info := readpipe(Master[(j) mod N*m ]);
            od;
            B[info[1]] := info[2];
#            info := convert(info,string);

            # Send info to next slaves.
            if (j+m <= Largest) then
                for i2 from (j-(N-2)*m) to j by m do
                    if i2 < 0 then next; fi ;
                    userinfo(3, 'MS', "Sending info to slave",(j+m)
                            mod N*m);
#                    info := convert([i2, B[i2]],string);
                    writepipe(Slaves[(j+m) mod N*m],[i2, B[i2]]);
                od;
            fi;

            if nargs >= 7 and ((j = 0) mod feq) then
                userinfo(3, 'MS', "Saving results so far");
                save B, File.j.`.m`;
            fi;

        od;
```

```
        fi;

        work := work + m;
    od;

    ## Wait for all the slaves to finish
    for i from 1 to N do
        wait();
    od;

    for i from q to (m-1)*N+q by m do
        close(Slaves[i][1]);
        close(Slaves[i][2]);
        close(Master[i][1]);
        close(Master[i][2]);
    od;

    RETURN(copy(B));
end:

savelib(‘calcul/balanced/worker‘, ‘calcul/balanced/worker.m‘);
savelib(‘calcul/balanced‘, ‘calcul/balanced.m‘);
```

File name: Multi2.

```
macro(binomial  = readlib(binomial),
      ceil      = readlib(ceil),
      frac      = readlib(frac),
      printf    = readlib(printf),
      readpipe  = readlib(‘calcul/readpipe‘),
      writepipe = readlib(‘calcul/writepipe‘),
      readfile  = readlib(‘calcul/readfile‘),
      writefile = readlib(‘calcul/writefile‘));

# calcul/readpipe
#     Performs the reading of information from pipe
# Input: pipe
# Output: informaton read
# Read: Informaiton
‘calcul/readpipe‘ := proc(pipeName, tries)
    local info, check;

    userinfo(5, ’MS’, "Reading information from pipe", pipeName);
    if nargs = 2 then
        userinfo(6, ’MS’, "Waiting", tries, "for pipe");
        if FAIL = block(tries, pipeName[1]) then
            userinfo(5,’MS’,"Failed to read from pipe");
            RETURN();
        fi;
    else
        userinfo(6, ’MS’, "Waiting forever for pipe", pipeName);
        if FAIL = block(5,pipeName[1]) then
            userinfo(5,’MS’,"Failed to read from pipe", pipeName);
            RETURN();
        fi;
    fi;
    userinfo(6, ’MS’, "Actually getting around to reading from pipe");
    info := readline(pipeName[1]);
    do
        check := traperror(parse(info));
        if check = lasterror then
            info := cat(info, readline(pipeName[1]));
        else
            break;
        fi;
    od;
    info := check;
    RETURN(info);
end:

# calcul/writepipe
#     Performs the writing of information to pipe
# Input: pipe, information
# Output: Error messages
# Write: Information
‘calcul/writepipe‘ := proc(pipeName, info)
    local LineToWrite, Length, SubLine, LARGE, k, t;
    userinfo(5, ’MS’, "Writing information to pipe", pipeName);
    LARGE := 10^8:
    LineToWrite := convert(info,string);
    Length := length(LineToWrite);
    for k from 1 to ceil(Length/LARGE) do
        SubLine := cat(LineToWrite[((k-1)*LARGE+1) ..
                min(Length,k*LARGE)], "\n");
        if FAIL = block(10,pipeName[2]) then
            print("Couldn’t write to pipe");
            RETURN(-1);
        fi;
        t := fprintf(pipeName[2],SubLine);
    od;
    RETURN(t);
end:

# calcul/readpipe
#     Performs the reading of information from pipe
# Input: pipe
# Output: informaton read
# Read: Information
‘calcul/readfile‘ := proc(fileName, tries)
    local info, check, fd, maxTries, good, i, ll;

    good := false;

    if nargs = 2 then maxTries := tries else maxTries := infinity fi;

    userinfo(5, ’MS’, "Reading information from file", fileName);
    for i from 1 to maxTries do
        fd := traperror(open(fileName,READ));

        if fd = lasterror then
            traperror(close(fileName));
            next;
        fi;

        info := traperror(readline(fd));
        if info = lasterror then
            next;
        fi;

        check := traperror(parse(info));
        if check = lasterror then
            next;
        fi;

        ll := traperror(close(fd));

        do
            ll := system("rm -f ".fileName);
            if ll = -1 then
                print("It is not removing ".fileName." properly");
                print("Giving up");
                quit;
            fi;
            break;
        od;

        good := true;

        break;
    od;
    if good then
        info := check;
        RETURN(info);
    else
        RETURN(NULL);
```

```
    fi;
end:

# calcul/writefile
#     Performs the writing of information to file
# Input: file, information
# Output: Error messages
# Write: Information
`calcul/writefile` := proc(fileName, info, tries)
    local fd, t, maxTries, i;
    if nargs = 3 then
        maxTries := tries;
    else
        maxTries := infinity;
    fi;

    t := -1;
    userinfo(5, 'MS', "Writing information to file", fileName);
    for i from 1 to maxTries do
        fd := traperror(open(fileName,WRITE));
        if fd = lasterror then
            userinfo(5,'MS',fd);
            traperror(close(fileName));
#            if maxTries <> i then system("sleep 1"); fi;
            userinfo(6, 'MS', "Trying to write again");
            next;
        fi;

        t := writeline(fd, convert(info,string));
        traperror(close(fd));
        break;
    od;
    userinfo(6, 'MS', "Finished writing information to file", fileName);

    RETURN(t);
end:

# calcul/balancing/slave
#     The slave that does all the work
# Input: Recurrences
# Output:  -
# Read: What work to do, and other infomration
# Write: Information discovered, and what info is needed.
`calcul/balancing/slave` := proc(Known, readPipe, writePipe, Top, Bot, m, Q,
        slaveNumber)
    local Info, info, largest, j, i, s, Value, q;

    q := Q mod m;

    Info := copy(Known);

    userinfo(5,'MS',"Figuring out how much info is known", slaveNumber);
    for i from q to infinity by m do
        if has(Info[i], `Info`) then break; fi;
    od;
    largest := i - m;

    userinfo(5,'MS',"Knows info", seq(Info[m*i+q],i=0..(largest-q)/m));
    userinfo(5,'MS',"Largest known is", largest, slaveNumber);

    userinfo(5,'MS',"Figuring out s value");
    for i from 0 to infinity do
        if Bot(i) <> 0 then
            s := i;
            break;
        fi;
    od;

    do
        userinfo(3,'MS',"Slave ".slaveNumber." is waiting for instructions");
        do
            info := readpipe(readPipe);
            if info <> NULL then break; fi;
        od;
userinfo(5,'MS',"Got ", info, "from pipe");

# If has some info.  Now it has to figure out what it means

# If it is a calculation request.
if info[1] = "Work" then
    userinfo(1,'MS',"Slave ".slaveNumber." is working on determining".
        " the value for ". (info[2]));

    j := info[2];

    Value := Top(j+s):

    for i from q to largest by m do
        Value := Value - Bot(s+j-i)*Info[i]*binomial(j+s,i);
    od;

    userinfo(5,'MS',"Value, before asking master for help", Value);

    while largest+m < j do

        userinfo(3,'MS',"Asking for data of ", largest+m);
        writepipe(writePipe,["Need Data", largest+m]);

        do
            info := readpipe(readPipe);
            if info <> NULL then break; fi;
        od;

        if info[1] = "Data" then
            userinfo(3,'MS',"Got some data ".(info[2])." from "
                    .slaveNumber);

            userinfo(5,'MS',"Using this new data");
            Info[info[2]] := info[3];
            largest := info[2];
            Value := Value - Bot(s+j-largest)*Info[largest]*
                    binomial(j+s,largest);
            userinfo(5,'MS',"Value, after asking master for help",
                    Value);

        # Don't know what the hell it is doing.
        else
            print("What the hell is going on, waiting for data", info);
            quit;
        fi;

    od;

    Value := Value / binomial(j+s,s)/Bot(s);

    userinfo(2,'MS',"Telling the overseer about the new value for ". j);
    writepipe(writePipe,["Data", j, Value]);

elif info[1] = "Data" then
    userinfo(5,'MS',"Got new data", slaveNumber);

    Info[info[2]] := info[3];
    largest := info[2];

# Just quit
elif info[1] = "Quit" then
    userinfo(2, 'MS', "Slave Quitting", slaveNumber);
    close(readPipe[1]);
    close(readPipe[2]);
    close(writePipe[1]);
    close(writePipe[2]);
    RETURN();

# Don't know what the hell it is doing.
else
   print("What the hell is going on got", info, slaveNumber);
   quit;
```

```
        fi;
    od;
end:

# calcul/balancing/overseerer
#     The communication on one machine
# Input: Recurrences
# Output:  -
# Read: What work to do, and other information
# Write: Information discovered, and what info is needed.
`calcul/balancing/overseer` := proc(Host, Me, Top, Bot, m, q, Known,
        numProcs, maxPipes)
    local readPipe, writePipe, info, numSlave, Info, slaveWait, slaveWork,
        Quit, slaveQuit, pid, i, j, workOn, messageSender, numProc, maxPipe, ll;

    workOn := []:
    if nargs >= 7 then
        Info := copy(Known);
    fi;
    if nargs = 9 then
        maxPipe := maxPipes;
    else
        maxPipe := 6;
    fi;
    if nargs >= 8 then
        numProc := numProcs;
    else
        numProc := 1;
    fi;

    numSlave := 0:

    writefile(cat(Me,2,Host), ["Need Work"]);

    do
        userinfo(3,'MS', "Waiting for instructions");
        info := NULL;
        do
            messageSender := 0:
            info := readfile(cat(Host,2,Me),1);
            if info <> NULL then break; fi;
            for messageSender from 1 to numSlave do
                info := readpipe(readPipe[messageSender],0);
                if info <> NULL then break; fi;
            od;
            if info <> NULL then break; fi;
        od;

        userinfo(1, 'MS', "Has ". numSlave. " slaves ".
                (numboccur([seq(slaveWork[i],i=1..numSlave)],true))
                ." running ". (numSlave - numboccur([seq(slaveWait[i],
                i=1..numSlave)],false)) ." waiting and the message is ".
                (info[1]));
        userinfo(5, 'MS', "Got info", info, "from ", messageSender);
        userinfo(3, 'MS', "Got info from slave/master ". messageSender);
        # Need to figure out what the message is

        # Find or create somebody to do the work
        if info[1] = "Work" then
            userinfo(1,'MS',"Told to do work on ".(info[2])." from ".
                    messageSender);

            if not has(Info[info[2]],'Info') then
                userinfo(2, 'MS', "Already know the info");
                writefile(cat(Me,2,Host), ["Data",info[2], Info[info[2]]]);
                Top(info[2]);
                Bot(info[2]);
                if workOn = [] then
                    writefile(cat(Me,2,Host), ["Need Work"]);
                fi;
                next;
            fi;

            for i from 1 to numSlave do
                if slaveWork[i] = false then break; fi;
            od;

            if i > maxPipe then
                workOn := [op(workOn),info[2]];

            # Create a new slave
            elif i > numSlave then
                userinfo(5,'MS',"Creating new slave",i,"to work on ",info[2]);
                numSlave := i;
                slaveWork[i] := true;
                slaveQuit[i] := false;
                slaveWait[i] := false;
                readPipe[i]  := pipe();
                writePipe[i] := pipe();
                Top(info[2]);
                Bot(info[2]);
                pid := fork();

                # The Slave
                if pid = 0 then
                    `calcul/balancing/slave`(Info, writePipe[i], readPipe[i],
                            Top, Bot, m, q, i);
                    quit;
                fi;

                writepipe(writePipe[i],info);

            # Use an old slave
            else
                userinfo(5,'MS',"Telling old slave ". i. " to work on ".
                        info[2]);
                slaveWork[i] := true;
                writepipe(writePipe[i],info);
            fi;

        # Check to see if the data is known
        # If it is, return it to the slave
        # If it isn't, put that slave in pending, and send off a need work
        elif info[1] = "Need Data" then

            userinfo(1,'MS', "Asked for data", info[2], "from", messageSender);

            # Doesn't know the information
            if has(Info[info[2]],Info) then
                userinfo(1,'MS',"Doesn't know the info", info[2], "for",
                        messageSender);
                slaveWait[messageSender] := info[2];

                if (numboccur([seq(slaveWork[l],l=1..numSlave)],true) -
                        (numSlave - numboccur([seq(slaveWait[l],
                        l=1..numSlave)], false))) < numProc and workOn = []
                        then
                    writefile(cat(Me,2,Host),["Need Work"]);
                    system("./sleepsm");
                fi;

            # It knows the information
            else
                userinfo(5,'MS',"Does know the info");
                writepipe(writePipe[messageSender],
                        ["Data",info[2],Info[info[2]]]);
            fi;

        # Deal with the data give overseer
        # Check to see if any slaves are waiting on it
        # If they are, make sure they get the information
        elif info[1] = "Data" then

            userinfo(1,'MS', "Given some new data ".(info[2])." from ".
                    messageSender);
            Info[info[2]] := info[3];
            for j from 1 to numSlave do
                if slaveWait[j] = info[2] then
```

```
                userinfo(3, 'MS', "Telling waiting slave ". j. " about ".
                        "this data");

                ll := writepipe(writePipe[j],
                        ["Data",info[2],Info[info[2]]]);
                slaveWait[j] := false:
            fi;
        od;

        # If this data came from a slave, then we might need more
        # work for the slave to do, and tell the master.
        if messageSender <> 0 then
            slaveWork[messageSender] := false;
            writefile(cat(Me,2,Host),["Data",info[2],info[3]]);
            if workOn = [] then
                userinfo(2,'MS',"Slave ". messageSender.
                        " is no longer working");
                if (numboccur([seq(slaveWork[l],l=1..numSlave)], true) -
                        (numSlave - numboccur([seq(slaveWait[l],
                        l=1..numSlave)], false))) <
                        numProc then
                    userinfo(2,'MS',"Ask for more work");
                    writefile(cat(Me,2,Host),["Need Work"]);
                fi;
            else
                userinfo(2,'MS',"Slave", messageSender,
                        "is no longer working, ",
                        "so give it outstanding work");
                writepipe(writePipe[messageSender],
                        ["Work",workOn[1]]);
                workOn := workOn[2..-1];
                slaveWork[messageSender] := true;
            fi;
        fi;

    # Doesn't want to give any more work.
    elif info[1] = "Quit" then
        for i from 1 to numSlave do
            if slaveWork[i] = false and slaveQuit[i] = false then
                userinfo(2,'MS',"Telling the ".i."th slaves to quit");
                slaveQuit[i] := true;
                writepipe(writePipe[i],["Quit"]);
            fi;
        od;
        for i from 1 to numSlave do
            if slaveQuit[i] = false then break; fi;
            userinfo(2,'MS',"The ".i."th slave has quit");
        od;
        if i > numSlave then
            userinfo(1,'MS',"Everyones quit, time to go home");
            for i from 1 to numSlave do
                close(writePipe[i][1]);
                close(writePipe[i][2]);
                close(readPipe[i][1]);
                close(readPipe[i][2]);
            od;
            RETURN();
        fi;

    # Don't know what the hell happened
    else
        RETURN("What the hell just happened");
        quit;
    fi;
od;
end:

# calcul/balancing/master
#     The main controller of all things good.
# Input: Nothing of importance
# Output:  -
# Read: Just about everything (the master knows all)
# Write: Just about anything (the master can order around all)
# Reference: Section 6.1.
```

```
`calcul/balancing/master` := proc(Host, Mach, Largest, m, q, fileName,
    interval, Known)

    local Info, info, i, j, k, maxKnown, needToWrite, writeThis, mach, l, fn,
            pid;

    Info := copy(Known);
    maxKnown := -1;

    mach := Mach;

    for i in Mach do
        needToWrite[i] := []:
    od;

    i := q;

    while Largest > maxKnown do
        info := NULL;
        userinfo(3,'MS',"Waiting for instructions");
        do
            for l from 1 to nops(mach) do
                j := mach[l];
                info := readfile(cat(j,2,Host), 1);

                userinfo(4,'MS',"Checking to see if there are outstanding ".
                       "messages for", j);
                if needToWrite[j] <> [] then
                    writeThis := needToWrite[j][1];
                    userinfo(5,'MS',"Sending information ".
                            "again to", j);
                    if (writefile(cat(Host,2,j),
                            writeThis, 1) <> -1) then
                        needToWrite[j] := needToWrite[j][2..-1];
                    fi;
                fi;
                if info <> NULL then
                    mach := [seq(mach[k],k=l+1..nops(mach)),
                            seq(mach[k],k=1..l)];
                    break;
                fi;
            od;
            if info <> NULL then break; fi;
            system("./sleepsm");
        od;

        userinfo(5,'MS', "Got information", info, "from", j);
        # We have info from one of the over seers, we have to
        # now figure out what it is.

        # Check to see if it is a request for work
        if info[1] = "Need Work" then
            userinfo(1,'MS', "Working on requested for work from ". j);

            if i > Largest then
                userinfo(2,'MS', "Tell ".j." to quit");
                if writefile(cat(Host,2,j),["Quit"],1) = -1 then
                    needToWrite[j] := [op(needToWrite[j]),["Quit"]]:
                fi;
            else
                userinfo(2,'MS', "Tell ".j." to work on the value of ". i);
                # Here I HAVE to make sure that they have had all
                # previous messages first.
                if needToWrite[j] = [] then
                    if writefile(cat(Host,2,j),["Work",i],1) = -1 then
                        needToWrite[j] := [op(needToWrite[j]),["Work",i]]:
#                       system("sleep 1");
                    fi;
                else
                    needToWrite[j] := [op(needToWrite[j]),["Work",i]]:
                fi;
            fi;
            i := i + m;
```

```
        # Check to see if it is new info
        elif info[1] = "Data" then
            userinfo(1,'MS', "Got some data for the value of ".(info[2]).
                    " from ". j);
            Info[info[2]] := info[3];

            maxKnown := max(maxKnown, info[2]);

            for k in Mach do
                if j = k then next; fi;
                userinfo(3,'MS', "Telling ". k. " about information");
                if writefile(cat(Host,2,k),
                        ["Data",info[2],info[3]],1) = -1 then
                    needToWrite[k] := [op(needToWrite[k]),
                            ["Data",info[2],info[3]]];
#                    system("sleep 1");
                fi;
            od;

            if (info[2] = 0) mod interval then
                fn := fileName.(info[2]).`.m`;
                pid := fork();
                if pid = 0 then
                    save Info, fn;
                    quit;
                fi
            fi;

        # Don't know what it is, make an error
        else
            print("What the hell is going on II got", info);
            quit;
        fi;
    od;
    for k in Mach do
        userinfo(1,'MS', "Telling ". k. " to quit");
        writefile(cat(Host,2,k),["Quit"],1);
    od;

    RETURN(op(Info));
    # Need to tell people to quit still.
end:

#libname := libname[3], libname[1..2]:
savelib(`calcul/readpipe`,  `calcul/readpipe.m`);
savelib(`calcul/writepipe`, `calcul/writepipe.m`);
savelib(`calcul/readfile`, `calcul/readfile.m`);
savelib(`calcul/writefile`, `calcul/writefile.m`);
savelib(`calcul/balancing/slave`,  `calcul/balancing/slave.m`);
savelib(`calcul/balancing/overseer`, `calcul/balancing/overseer.m`);
savelib(`calcul/balancing/master`,  `calcul/balancing/master.m`);
```

Bibliography

[1] *Cecm research projects*, http://www.cecm.sfu.ca/projects, 1999.

[2] Milton Abramowitz and Irene A. Stegun, *Handbook of mathematical functions*, 9th ed., Dover Publications, Inc, New York, 1992.

[3] J. L. Adams, *Conceptual blockbusting: A guide to better ideas*, Freeman, San Francisco, 1974.

[4] Bruce C. Berndt, *Ramanujan's notebooks*, Springer-Verlag, New York, 1994.

[5] Jonathan Borwein, Peter Borwein, and Lennart Berggren, *Pi: A source book*, Springer, New York, 1997.

[6] Jonathan M. Borwein, David M. Bradley, and Richard E. Crandall, *Computational strategies for the Riemann zeta function*, unpublished, 1996.

[7] Carl B. Boyer, *A history of mathematics*, John Wiley & Sons, Inc., 1968.

[8] L Carlitz, *Some arithmetic properties of the oliver functions.*, Mathematische Annalen **128** (1955), 412 – 419.

[9] Mustapha Chellali, *Accélération de calcul de nombres de Bernoulli*, Journal of Number Theory (1988), 347–362.

[10] Louis Comtet, *Advanced combinatorics, the art of finite and infinite expansions*, D. Reidel Publishing Company, Boston, 1974.

[11] F. N. David, M. G. Kendall, and D. E. Barton, *Symmetric function and allied tables*, Cambridge, Cambridge, 1966.

[12] K.O. Geddes, S.R. Czapor, and G. Labahn, *Algorithms for computer algebra*, Kluwer Academic Publishers, 1996.

[13] K.O. Geddes, G. Labahn, M. B. Monagan, and S. Vorketter, *The maple programming guide*, Springer-Verlag, New York, 1996.

[14] J. W. L. Glaisher, *On Eulerian numbers*, Quarterly Journal of Mathematics **45** (1914).

[15] Gene H. Golub and Charles F. van Loan, *Matrix computations*, second ed., The Johns Hopkins University Press, Baltimore, 1989.

[16] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, *Concrete mathematics*, second ed., Addison-Wesley Publishing Company, Reading, MA, 1994, A foundation for computer science.

[17] G. H. Hardy and W. M Wright, *An introduction to the theory of numbers*, fourth ed., Clarendon Press, Oxford, 1960.

[18] I.N. Herstein, *Topics in algebra*, second ed., John Wiley & Sons, Toronto, 1975.

[19] D.H. Lehmer, *Lacunary recurrence formulas for the numbers of Bernoulli and Euler*, Annals of Mathematics **36** (1935), no. 3, 637–649.

[20] Maurice Mignotte, *Mathematics for computer algebra*, Springer-Verlag, New York, 1992, Translated from the French by Catherine Mignotte.

[21] J. Miller, N. J. A. Sloane, and N. E. Young, *A new operation on sequences: the boustrophedon transform*, J. Combn Theory **17A** (1996), 44–54.

[22] S Ramanujan, *Some properties of Bernoulli's numbers*, Indian Mathematical Journal (1911).

[23] J. Riordan, *An introduction to combinatorial analysis*, Wiley, 1958.

[24] John Riordan, *Combinatorial identities*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York, 1968.

[25] N. J. A. Sloane and Simon Plouffe, *The encyclopedia of integer sequences*, Academic Press, Toronto, 1995.

[26] Neil J. A. Sloane, *Sloane's on-line encyclopedia of integer sequences*, http://akpublic.research.att.com/~njas/sequences/index.html, 1998.

[27] C. R. Snow, *Concurrent programming*, Cambridge Computer Science Texts, no. 26, Cambridge University Press, New York, 1992.

[28] I Steward, *Math. rec.*, Scientific American (1996).

[29] W. A. Whitworth, *Dcc exercises in choice and chance*, Stechert, New York, 1945.

[30] Herbert S Wilf, *Generating functionology*, Academic Press, Inc., Toronto, 1990.

Experimental Mathematics via Inverse Symbolic Computation

David M. Bradley

Centre for Experimental and Constructive Mathematics
Simon Fraser University

http://www.cecm.sfu.ca/~dbradley/

June 12, 1997

Introduction

Is mathematics created or discovered? Or both?

Traditionally, mathematics has distinguished itself from the empirical sciences, in that the latter must fit their theories to experimental reality.

On the other hand, mathematicians, within certain constraints, are free to choose their own reality.

If mathematics is primarily created, rather than discovered, then experimentation should have little or no influence in mathematical development.

However, if discovery plays a significant role, then we should expect that experimentation should also.

In previous centuries, experimentation in mathematics was mostly limited to trial and error hand-calculuation. With the advent of digital (and now molecular!) computers, a revolution is taking place in how mathematicians go about their work.

In fact, the past 20 years has seen a dramatic reconcretization of mathematics, in which fields such as number theory, classical analysis, and special functions have received new infusions fueled by advances in hardware, software, and algorithms.

A whole new palette of tools is available to the researcher, many of which, with a little effort, can operate like extensions of the mind, multiplying our ability to generate examples, test hypotheses, and build intuition by unprecendented factors.

The current generation of computers can not only verify results, but *predict* them.

The sophistication of computational tools continues to improve, yielding more than just a quantitative increase in mathematical results; rather, a *qualitative* shift is taking place.

Probably the best kind of tool is the one that enables people to build better tools.

As tool is built upon tool, a peculiar meld of mind and machine emerges. The distinction between mind and machine is increasingly blurred. Many of my most impressive results of the past two years could not and would not have been discovered without the assistance of sophisticated inverse symbolic computational software.

"Computers are useless. They can only give you answers."

Pablo Picasso (1881-1973)

Picasso was wrong.

• Answers most definitely are useful.

• In the realm of experimental mathematics, computers generate far more questions than answers.

• Computers are now sophisticated enough to make conjectures of their own.

• Computer-generated conjectures have led to questions that suggest fundamental new avenues of research.

• In many cases, consideration of computational issues has led to significant paradigm shifts.

Fractals and Chaos

• The study of fractals and chaos was simply too difficult to undertake before the advent of high-speed digital computers.

• Simply stated, it was humanly impossible to generate any but the crudest of fractal images.

*****

But...

Just as the study of fractals unveiled fantastically baroque images of stunning beauty, massive computer searches and inverse symbolic techniques are revealing equally enticing analytic objects whose beauty lies in the more conceptual realms of infinite series, continued fractions, and other iterative expansions.

What is Inverse Symbolic Computation?

First, what is symbolic computation?

Partial Answer by Examples:

• Evaluate a sum, an integral, a product,...
• Compute a derivative, a determinant,...
• Verify a formula
• Given a question, find the answer

The inverse problem to

"given a question, find the right answer"

is

Given an answer,

Find the "Right" Question!

Hence, Inverse Symbolic Computation:

• Given a number, ask "What combination of constants and special function values most likely produced it?"

• Given a sequence, ask "What is the generating function?" or "Does the sequence have a name?" "Does it arise in other contexts?"

• Given a finite set of functions and operations, ask "Does there exist a formula which looks something like $\{\cdots\}$ and involves these functions and operations? (And if so, what is it?)"

Inverse Symbolic Computational Tools

Inverse Symbolic Calculator
- Input a number, and find out where it comes from.
- Combines table lookup with "smart lookup" to check whether the number is a simple combination of known constants.
- Author: Simon Plouffe, formely of CECM, now at Wolfram Research
- http://www.cecm.sfu.ca/projects/ISC/

Encyclopedia of Integer Sequences
- Does for sequences what the ISC does for real numbers.
- Expanded and updated version of Sloane's handbook
- Text by Neil Sloane & Simon Plouffe, Academic Press, 1995, ISBN: 0-12-558632-9
- http://www.research.att.com/~njas/sequences/

Maple's GFUN package
-Given a sequence of numbers, the package will try to guess a recurrence, find the ordinary/exponential generating function, solve the recurrence, etc.

Integer Relations Algorithms
- Lattice Basis Reduction (Ferguson & Forcade)
- "LLL" (Lenstra, Lenstra & Lovasz)
- "PSLQ", (Dave Bailey, NASA Ames & Ferguson)
- Given a vector of real numbers, output a vector of integers such that the dot product is zero to within working precision.

Inverse Symbolic Computation

• What is 1.1981402347355922074...?

If $a_0 = 1$, $b_0 = \sqrt{2}$, and

$$a_{n+1} = \frac{a_n + b_n}{2}, \quad b_{n+1} = \sqrt{a_n b_n},$$

then

$$\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = \frac{\pi/2}{\int_0^1 \frac{dt}{\sqrt{1-t^4}}}$$

$$= 1.1981402347355922074\ldots$$

The convergence is quadratic - 10 iterations already yields nearly 1400 digits of accuracy.

In 1799, Gauss observed this purely numerically, and wrote that this result

"will surely open a whole new field of analysis."

The Inverse Symbolic Calculator (ISC)

To what extent can we automate Gauss's incredible insight?

Obviously, Gauss was familiar with the initial digits of the decimal expansions of π, the complete elliptic integrals, and probably simple rational multiples of these as well.

So, it would make sense to compile a table of "famous" constants, and for a first crack, try a simple table-lookup.

Even better, one could try simple rational multiples of tabulated constants, and various linear combinations and products of these.

Simon Plouffe's "Inverse Symbolic Calculator" is a practical instantiation of this idea. Essentially, it is a calculator with a big screen and only one button, which answers the question "What is this number made of?"

The number $r := 1.6180339887498948482\ldots$ happens to be the unique positive real number satisfying

$$\int_0^\infty \frac{dx}{(1+x^r)^r} = 1.$$

So, what is r? Is it just the number that happens to work or is it special in some other contexts as well?

Ask the ISC, and we find that, most likely,

$$r = \frac{1+\sqrt{5}}{2},$$

which is indeed the case.

The so-called "golden ratio" can turn up in the most unexpected places.

Detecting Integer Relations

• Given a vector of real numbers (or their decimal approximations), output a vector of integers such that the dot product is zero to within working precision.

• Let $x = (x_1, x_2, \ldots, x_n)$ be a vector of real numbers. Then x is said to possess an integer relation if there exist integers a_j not all zero such that

$$\sum_{j=1}^{n} a_j x_j = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0.$$

Problem: Find the integers a_j if they exist. If they do not exist, then truncating the x_j to working precision, obtain lower bounds on the size of the a_j.

• Euclid's algorithm gives a solution for $n = 2$.

• Euler, Jacobi, Poincare, Minkowski, Perron, Brun, Bernstein and others tried to find a general algorithm for $n > 2$.

• The first general algorithm, *Lattice Basis Reduction,* was discovered in 1977 by Ferguson and Forcade, however it suffered from numerical stability.

• Improvements were given by Lenstra, Lentra and Lovasz (LLL) and Dave Bailey of NASA Ames (PSLQ).

Applications of Integer Relation Detection

• Suppose α can be computed to high precision. Form the vector $x = (1, \alpha, \alpha^2, \ldots, \alpha^n)$ and apply an integer relation detecting algorithm. If a relation is found, the integers a_j satisfy

$$\sum_{j=0}^{n} a_j \alpha^j = a_0 + a_1\alpha + \cdots a_n\alpha^n = 0,$$

i.e. α is algebraic of degree $\leq n$. If no relation is found, bounds are obtained within which no such annihilating polynomial can exist.

• One can also check whether α satisfies an identity of the form

$$\alpha^p = 2^a 3^b 5^c \pi^d [\zeta(3)]^m [\Gamma(1/4)]^k \cdots,$$

say, by taking logarithms.

Some Integer Relation Results

$$\int_0^{\pi/3} \log(\tan\theta)\, d\theta = \int_0^{\pi/6} \log(\tan\theta)\, d\theta,$$

$$2\int_0^{\pi/4} \log(\tan\theta)\, d\theta = 3\int_0^{\pi/12} \log(\tan\theta)\, d\theta,$$

$$\begin{aligned} 2\int_0^{\pi/4} \log(\tan\theta)\, d\theta & \\ = 5\int_0^{3\pi/20} \log(\tan\theta)\, d\theta & \\ -5\int_0^{\pi/20} \log(\tan\theta)\, d\theta, & \end{aligned}$$

and many, many more...

Series Acceleration Formulae for Catalan's Constant

Catalan's constant is

$$G := \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}.$$

Let $L(n)$ denote the nth Lucas number (M0155 in Sloan and Plouffe's Encyclopaedia). The Lucas numbers satisfy the recursion

$$L(n) = L(n-1) + L(n-2), \qquad n > 2,$$

with initial conditions $L(1) = 1$, $L(2) = 3$.

Theorem 1

$$G = \frac{\pi}{8}\log\left(\frac{10+\sqrt{50-22\sqrt{5}}}{10-\sqrt{50-22\sqrt{5}}}\right) + \frac{5}{8}\sum_{n=0}^{\infty} \frac{L(2n+1)}{(2n+1)^2\binom{2n}{n}}.$$

Proof. (Sketch) First, define

$$T(r) := \int_0^{\pi r} \log(\tan\theta)\, d\theta, \quad 0 \leq r \leq \tfrac{1}{2}.$$

Lemma 1 *The Fourier Series expansion*

$$T(r) = -\sum_{n=0}^{\infty} \frac{\sin((2n+1)2\pi r)}{(2n+1)^2}, \quad 0 \leq r \leq \tfrac{1}{2},$$

holds.

Proof. Let $z = e^{-2ix}$, $0 \leq x \leq \frac{1}{2}\pi$. By multisecting the power series for $\log(1+z)$, we have

$$\begin{aligned} 2\sum_{n=0}^{\infty} \frac{\cos(4n+2)x}{2n+1} &= 2\Re\sum_{n=0}^{\infty} \frac{e^{-(4n+2)ix}}{2n+1} \\ &= \Re\log\frac{1+e^{-2ix}}{1-e^{-2ix}} \\ &= -\log(\tan x). \end{aligned}$$

Lemma 1 now follows on integrating and setting $x = \pi r$.

Corollary 1

$$\begin{aligned} G &:= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2} \\ &= -T(\tfrac{1}{4}) \\ &= -\int_0^{\pi/4} \log(\tan\theta)\, d\theta. \end{aligned}$$

Proof. Put $r = \frac{1}{4}$ in the Fourier Series expansion of Lemma 1.

We seek series acceleration formulae for G. The idea is to employ integer relations between log(tan) integrals. Ideally, the relations should involve integrals with reduced integration ranges. When re-expanded into series, the reduced integration range is manifested as a continuous analog of bunching several terms together, yielding a more rapidly convergent series.

Continuing with the proof of Theorem 1, we write

$$\begin{aligned} -\frac{2}{5}G &= \frac{2}{5}\int_0^{\pi/4} \log(\tan\theta)\, d\theta \\ &= \int_0^{3\pi/20} \log(\tan\theta)\, d\theta - \int_0^{\pi/20} \log(\tan\theta)\, d\theta \\ &= \int_{\pi/20}^{3\pi/20} \log(\tan\theta)\, d\theta \\ &= \theta\log(\tan\theta)\Big|_{\pi/20}^{3\pi/20} - \int_{\pi/20}^{3\pi/20} \frac{\theta\sec^2\theta}{\tan\theta}\, d\theta \\ &= \frac{\pi}{20}\log\left(\frac{\tan^3(\frac{3\pi}{20})}{\tan(\frac{\pi}{20})}\right) - \int_{\pi/10}^{3\pi/10} \frac{\frac{1}{2}\theta\,\frac{1}{2}d\theta}{\sin\frac{1}{2}\theta\cos\frac{1}{2}\theta}. \end{aligned}$$

But,

$$\begin{aligned} -\int_{\pi/10}^{3\pi/10} \frac{\frac{1}{2}\theta\,\frac{1}{2}d\theta}{\sin\frac{1}{2}\theta\cos\frac{1}{2}\theta} &= -\frac{1}{2}\int_{\pi/10}^{3\pi/10} \frac{\theta}{\sin\theta}\, d\theta \\ &= -\int_{\sin\frac{\pi}{10}}^{\sin\frac{3\pi}{10}} \frac{\sin^{-1}x}{\sqrt{1-x^2}}\cdot\frac{dx}{2x} \\ &= -\int_{\sin\frac{\pi}{10}}^{\sin\frac{3\pi}{10}} \sum_{n=1}^{\infty} \frac{(2x)^{2n-2}}{n\binom{2n}{n}}\, dx \\ &= \frac{1}{4}\sum_{n=0}^{\infty} \frac{\phi^{2n+1}-\tau^{2n+1}}{(2n+1)^2\binom{2n}{n}}, \end{aligned}$$

where

$$\begin{aligned} \phi &:= 2\sin(\pi/10) = \frac{\sqrt{5}-1}{2}, \\ \tau &:= 2\sin(3\pi/10) = \frac{\sqrt{5}+1}{2}. \end{aligned}$$

The proof of Theorem 1 is complete once we

a) note that the Lucas numbers satisfy

$$L(n) = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n, \qquad n \geq 0,$$

and

b) verify the non-trivial algebraic denesting relationship

$$\frac{\tan^3(\frac{3\pi}{20})}{\tan(\frac{\pi}{20})} = \frac{10-\sqrt{50-22\sqrt{5}}}{10+\sqrt{50-22\sqrt{5}}}.$$

Success or Failure?

The simpler relationship

$$2\int_0^{\pi/4} \log(\tan\theta)\,d\theta = 3\int_0^{\pi/12} \log(\tan\theta)\,d\theta,$$

yields Ramanujan's result (which he proved by quite different methods):

$$G = \frac{\pi}{8}\log(2+\sqrt{3}) + \frac{3}{8}\sum_{n=0}^{\infty} \frac{1}{(2n+1)^2\binom{2n}{n}}.$$

But, recall the adage

"If at first you don't succeed ...

... redefine success."

We have

$$\sum_{n=0}^{\infty} \frac{L(2n+1)}{(2n+1)^2\binom{2n}{n}} = \frac{8}{5}G + \frac{\pi}{5}\log\left(\frac{10-\sqrt{50-22\sqrt{5}}}{10+\sqrt{50-22\sqrt{5}}}\right),$$

where

$$G := \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}$$

is Catalan's constant.

Some Much Deeper Results
(Born of LLL/PSLQ)

$$\zeta(7) := \sum_{k=1}^{\infty} \frac{1}{k^7} = \frac{5}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^7\binom{2k}{k}} + \frac{25}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3\binom{2k}{k}} \sum_{j=1}^{k-1} \frac{1}{j^4}.$$

- For all positive integers n,

$$\frac{5}{2}\sum_{k=1}^{n} \binom{2k}{k} \frac{n^2k^2}{4n^4+k^4} \prod_{j=1}^{k-1} \frac{n^4-j^4}{4n^4+j^4} = 1.$$

- For all positive integers n,

$$\frac{1}{\pi}\int_0^{\infty} \frac{dy}{1+y^2} \prod_{j=0}^{n-1} \frac{4y^2-(j/n)^4}{y^2+(j/n)^4} = \binom{2n}{n}.$$

- For all positive integers n,

$${}_6F_5\left(\begin{matrix} n+1,\ n+1,\ 2n\pm in,\ \pm in \\ n+1/2, n, 2n+1, n+1\pm in \end{matrix}\ \middle|\ -\frac{1}{4}\right) = \frac{2}{5}\binom{2n}{n} \prod_{j=1}^{n-1} \frac{n^4-j^4}{4n^4+j^4}.$$

- For all complex numbers z,

$$\sum_{k=1}^{\infty} \frac{1}{k^3(1-z^4/k^4)} = \frac{5}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^3\binom{2k}{k}} \frac{1}{1-z^4/k^4} \prod_{j=1}^{k-1} \frac{1+4z^4/j^4}{1-z^4/j^4}.$$

Young–Tablåer och mönsterundvikande

Sverker Lundin

September 2001

Young–Tablåer och mönsterundvikande

Examensarbete i matematik

Sverker Lundin

Examinator: Einar Steingrímsson

Matematik
Chalmers tekniska högskola och Göteborgs universitet

Göteborg september 2001

Sammanfattning

Vi undersöker ett antal kopplingar mellan Young–Tablåer, Ballot–tal och mönsterundvikande permutationer. Vi visar också att för vissa mönster p kan antalet involutioner som undviker p förklaras med hjälp av p:s cykelstruktur. För att göra det lättare att se cykelstrukturen hos ett mönster, tar vi fram ett sätt att visualisera permutationer med längd ≤ 4.

Abstract

In this paper, we study some connections between Young–Tableau, Ballot–numbers and pattern avoiding permutations. We also show that, for some patterns p, the number of involutions avoiding p can be explained in terms of the cycle–structure of p. To facilitate the interpretation of patterns in terms of their cycle–structures, we propose a technique for visualisation of permutations of length ≤ 4.

Innehåll

1 Inledning

En Young–Tablå[1] är en tabell med talen $1, \ldots, n$ ordnade enligt följande regler:

- Om talen a och b är på samma rad och a står till vänster om b, så måste $a < b$ och
- om c och d är i samma kolumn och c är över d så måste $c < d$.

Raderna i tabellen måste inte vara lika långa, men rader högre upp måste vara längre än eller lika långa som rader längre ner.

1	3	5	8
2	4		
6	7		
9			

är ett exempel på en Young–Tablå med talen $1, \ldots, 9$. Två andra exempel är

1
2
3

och

1	2	3	5
4			

.

2 Young–Tablåer med två rader

En erfaren kombinatoriker föreslog mig att jag skulle förklara varför antalet Young–Tablåer (som i fortsättning helt enkelt benämns "tablåer") av storlek $n + 1$ med exakt två rader är

$$C(n) = \frac{1}{n+1}\binom{2n}{n}. \tag{1}$$

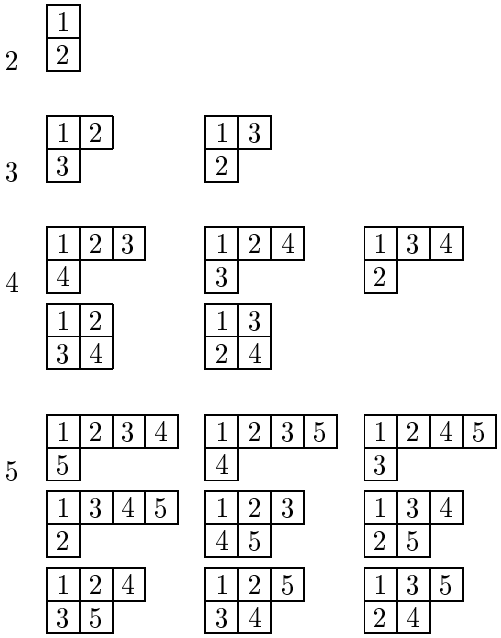
Talföljden som genereras av (1), och vars inledande termer är

$$1, 1, 2, 5, 14, 42 \ldots \tag{2}$$

[1] Efter den engelske gruppteoretikern Alfred Young (1873–1940).

kallas Catalantal[2], och dyker upp i en mängd kombinatoriska problem (i Richard Stanleys bok "Enumerative Combinatorics" [17] finns en övningsuppgift som innehåller över 40 olika problem där lösningarna involverar Catalantal).

För att testa hypotesen konstruerar vi för hand alla möjliga tablåer som har $2, 3, 4$ och 5 rutor:

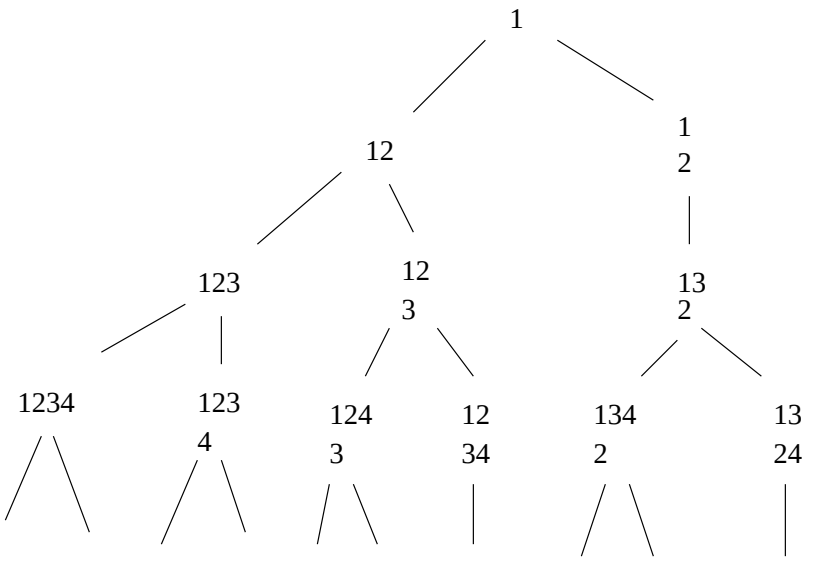


Räknar vi antalet tablåer av varje storlek får vi följande tabell, där vi kallar antalet tvåradiga tablåer med n rutor för $T'(n)$:

n	$C(n)$	$T'(n+1)$
1	1	1
2	2	2
3	5	5
4	14	9

Tydligen är lösningen *inte* Catalantalen, ty $T'(4+1) \neq C(4)$.

[2]Efter den franske matematikern Eugène Charles Catalan (1814–1894). Catalan arbetade med talteori och kedjebråk på École Polytechnique i Paris. Hans matematiska karriär försvårades för övrigt av hans starka sympatier för den politiska vänstern [13].



Figur 1: Konstruktions–algoritm för tvåradiga tablåer.

Hur gör man för att ta fram en formel för $T'(n)$?

De tal som ingår i dessa tablåer står antingen i den övre eller i den undre raden. Man kan bygga upp en tablå genom att placera ut talen $1, \ldots, n$ i nummerordning. Placera varje nytt tal längst till höger i antingen den övre eller den undre raden.

Det är dock inte alltid som man kan välja fritt om man vill placera ut talet i den övre eller den undre raden på grund av:

1 villkoret att den övre raden inte får vara kortare än den undre och

2 storleksvillkoret, som säger att varje tal i den undre raden måste ha ett mindre tal rakt ovanför sig.

Man kan visualisera konstruktionsprocessen med hjälp av ett träd, som i figur (1). Vi märker att trädet även innehåller tablåer med enbart en rad, och det verkar lättare att generellt beskriva antalet tablåer med en eller två rader, snarare än tablåer med exakt två rader. Vi kallar antalet tablåer med en eller två rader för $T(n)$.

De noder på trädet i figur (1) som representerar tablåer där båda raderna är lika långa skiljer sig från de övriga noderna, i det att de bara har ett "barn". När vi skall beskriva konstruktionsalgoritmen matematiskt måste vi därför ha en modell av tablåerna som beskriver tablåernas form. Inför vi beteckningen $B(a, b)$ för att beskriva antalet tablåer med a rutor i den övre raden och b rutor i den undre, kan vi teckna följande *rekursion*:

$$B(a,b) = B(a,b-1) \quad (a = b) \tag{3}$$

$$B(a,b) = B(a,b-1) + B(a-1,b) \quad (a > b) \tag{4}$$

Ekvation (3) säger att antalet tablåer med formen (n, n) det vill säga tablåer med två lika långa rader är lika stort som antalet tablåer med formen $(n, n-1)$. Orsaken är att tablåerna med formen (n, n) bara kan skapas på ett sätt, nämligen genom att lägga till en ruta i den undre raden på en tablå med formen $(n, n-1)$.

Ekvation (4) beskriver det faktum att övriga tablåer, dvs där $a > b$, kan skapas på två sätt; antingen genom att lägga till ett tal i den undre raden på en tablå med formen $(a, b-1)$ eller genom att lägga till ett tal i den övre raden på en tablå med formen $(a-1, b)$. Det totala antalet tablåer måste därför vara summan av antalet tablåer med dessa respektive former.

Ett steg mot att hitta en formel för $T(n)$ är att skapa en tabell med värden för $B(a, b)$. Detta kan vi göra med hjälp av ekvationerna (3) och (4) tillsammans med de värden vi fått genom att konstruera tablåer för hand. Vi börjar med att fylla i dessa värden:

$a \diagdown b$	0	1	2	3	4	5	6
0	?	?	?	?	?	?	?
1	1	1	?	?	?	?	?
2	1	2	2	?	?	?	?
3	1	3	5	?	?	?	?
4	1	?	?	?	?	?	?
5	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?

Eftersom vi inte tillåter tablåer där $a < b$, dvs där den undre raden har fler rutor, vet vi att $B(a, b) = 0$ för alla $a < b$:

$a \diagdown b$	0	1	2	3	4	5	6
0	?	0	0	0	0	0	0
1	1	1	0	0	0	0	0
2	1	2	2	0	0	0	0
3	1	3	5	?	0	0	0
4	1	?	?	?	?	0	0
5	?	?	?	?	?	?	0
6	?	?	?	?	?	?	?

Nu är det enkelt att fylla i resten av tabellen med hjälp av ekvationerna (3) och (4) eftersom de betyder att varje tal i tabellen är summan av talet till vänster och talet ovanför. Vi bestämmer oss också för att låta $B(0,0) = 1$:

$a \diagdown b$	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
2	1	2	2	0	0	0	0
3	1	3	5	5	0	0	0
4	1	4	9	14	14	0	0
5	1	5	14	28	42	42	0
6	1	6	20	48	90	132	132

Antalet tablåer med exakt två rader $T'(n)$, som vi sökte inledningsvis, kan vi nu få fram genom att summera $B(a,b)$ över alla former för vilka $a+b=n$

$$\sum_{a+b=n} B(a,b) - 1$$

(eftersom vi inte är intresserade av tablåerna med bara en rad) vilket ger talföljden:

$$1, 2, 5, 9, 19, 34 \ldots \tag{5}$$

detta är, kan man säga, en lösning till problemet eftersom vi, om det skulle behövas, kan göra tabellen större.

2.1 Försök till bijektivt bevis

Vi vill dock hitta ett enklare sätt att beskriva talföljden. En praktiskt metod för att komma vidare från detta läge är att skicka de tal man fått fram till

den stora databasen "Sloane's On–Line Encyclopedia of Integer Sequences" på Internet[3]. Vi får då reda på att vår talföljd heter **A014495** och beskrivs av $\binom{n}{\lceil n/2 \rceil} - 1$. Om vi struntar i att dra bort ett, och räknar tablåer en eller två rader får vi alltså att

$$T(n) = \binom{n}{\lceil n/2 \rceil}.$$

Ett sätt att visa, eller förklara, varför mängden av alla en– och tvåradiga tablåer är lika stor som mängden av alla uppdelningar av n element i två lika stora delmängder, är att skapa en bijektion. Detta innebär att vi skapar par; i vårt fall par av tablåer och uppdelningar av $[n]$ i två lika stora delmängder. Vi kallar mängden av alla uppdelningar av talen $1, \ldots, n$ för X och mängden av alla tablåer med en eller två rader för Y. Vi vill hitta på en algoritm – vi kallar den A – som tar ett element $x \in X$ och ger ett element $y \in Y$. Använder vi $n = 5$ som exempel vill vi alltså i följande tabell:

1 2 3 4 5	{1,2}
1 2 4 3 5	{1,3}
1 2 5 3 4	{1,4}
1 3 4 2 5	{1,5}
1 3 5 2 4	{2,3}
1 2 3 4 5	{2,4}
1 2 3 5 4	{2,5}
1 2 4 5 3	{3,4}
1 3 4 5 2	{3,5}
1 2 3 4 5	{4,5}

som innehåller både alla tablåer med 5 tal och alla sätt att välja ut 2 tal

[3] http://www.research.att.com/~njas/sequences/Seis.html

från $\{1, \ldots 5\}$, komma på ett sätt att koppla samman varje tablå till vänster med ett val av två tal till höger.

Vi försöker lösa detta problem genom att använda vår intuition om vad som kan tänkas leda till en rimlig generell algoritm för att skapa sådana par. Betraktar vi valet av talen $\{4, 5\}$, så kan man ju tänka sig att dessa skulle kopplas samman med tablån $\begin{array}{|c|c|c|}\hline 1&2&3\\\hline 4&5\\\cline{1-2}\end{array}$. Det samma gäller för de övriga tablåerna med två tal i den undre raden. Tar vi bort dessa får vi följande lite mer problematiska tabell:

$\begin{array}{|c|c|c|c|}\hline 1&2&3&4\\\hline 5\\\cline{1-1}\end{array}$	{1,2}
$\begin{array}{|c|c|c|c|}\hline 1&2&3&5\\\hline 4\\\cline{1-1}\end{array}$	{1,3}
$\begin{array}{|c|c|c|c|}\hline 1&2&4&5\\\hline 3\\\cline{1-1}\end{array}$	{1,4}
$\begin{array}{|c|c|c|c|}\hline 1&3&4&5\\\hline 2\\\cline{1-1}\end{array}$	{1,5}
$\begin{array}{|c|c|c|c|c|}\hline 1&2&3&4&5\\\hline\end{array}$	{2,3}

Tänker vi oss att de tal som är utvalda är tal som vi skulle vilja placera i undre raden (detta stämmer ju med vad vi gjort hittills) ser vi att alla val som återstår skulle leda till otillåtna tablåer. Fyran och femman förekommer bara i två av valen på högersidan, så det är inte speciellt långsökt att identifiera dessa med de respektive tablåer där fyran och femman är i den undre raden. Det enda som återstår är nu

$\begin{array}{|c|c|c|c|}\hline 1&2&4&5\\\hline 3\\\cline{1-1}\end{array}$	{1,2}
$\begin{array}{|c|c|c|c|}\hline 1&3&4&5\\\hline 2\\\cline{1-1}\end{array}$	{1,3}
$\begin{array}{|c|c|c|c|c|}\hline 1&2&3&4&5\\\hline\end{array}$	{2,3}

Det finns naturligtvis många alternativ, både i den här situationen och i det övriga resonemanget kring hur bijektionen skall utformas. Det känns dock naturligt att låta $\{1, 2\}$ kopplas samman med den enradiga tablån, $\{1, 3\}$ med tablån som har trean som enda tal i undre raden, och till sist $\{2, 3\}$ med tablån som har två som enda tal i undre raden.

Om vi med utgångspunkt från detta exempel kan ta fram en generell algoritm för att skapa en en– eller tvåradig tablå med utgångspunkt från en uppdelning av talen $1, \ldots, n$ i två lika stora delmängder, och varje uppdelning ger en specifik tablå, har vi hittat en bra förklaring till varför antalet en– eller tvåradiga tablåer med n tal är lika med antalet sätt att välja ut hälften av de n talen.

En generell algoritm som ger de kopplingar vi fått mellan tablåer och val av tal i exemplet är följande:

- Placera alla tal som inte är valda i den översta raden, och de valda talen i den undre raden, justerade till vänster. (Detta ger alltså en otillåten tablå med hälften, eller nästan hälften, av talen i den undre raden.)
- Titta på det tal i den undre raden som är längst till vänster. Det finns två möjligheter
  - Om talet ovanför är mindre så är dessa två tal en tillåten kombination. Gå då vidare med nästa tal i den undre raden.
  - Om talet ovanför är större så lägg det undre talet i den övre raden, närmast till vänster om det tal som är i den övre raden. Flytta sedan de övriga undre talen ett steg åt vänster så att det tal som står på tur hamnar under samma tal i övre raden som det vi just flyttade på hade ovan för sig innan vi flyttade på det.
- Fortsätt på detta sätt tills alla tal i den undre raden är beaktade.
- Vi har nu en tablå som kan se konstig ut, för talen i den undre raden är inte vänsterjusterade. Ordna detta genom att skjuta samman dessa tal längst till vänster.

För det första är det klart att det verkligen blir tillåtna tablåer, för vi låter bara tal vara kvar i den undre raden om de är tillåtna. När vi flyttar upp tal så har vi alltid ett större tal till höger, eftersom vi flyttar upp just om talet ovanför är större, och det är detta tal som sedan hamnar till höger. När vi slutligen flyttar vissa tal i den undre raden åt vänster så hamnar de under tal som är mindre än de tal de tidigare hade ovanför sig, så detta kan inte leda till att tablån blir otillåten.

Vi illustrerar med ett exempel: Antag att $n = 9$ och talen $\{1, 2, 5, 9\}$ är utvalda. Vi börjar då med den otillåtna tablån

3	4	6	7	8
1	2	5	9	

.

Eftersom $1 \not> 3$ så flyttar vi upp ettan till vänster om trean...

1	3	4	6	7	8
–	–	2	5	9	

Vi flyttar undre raden ett steg åt vänster, så att tvåan hamnar under det tal som ettan förut hade ovan för sig, dvs trean.

1	3	4	6	7	8
–	2	5	9		

Eftersom $2 \not> 3$ så flyttar vi även upp tvåan, som hamnar mellan ettan och trean...

1	2	3	4	6	7	8
–	–	–	5	9		

skifta undre raden ett steg åt vänster...

1	2	3	4	6	7	8
–	–	5	9			

vi har nu att både $5 > 3$ och $9 > 4$ så vi skall inte flytta upp några mer tal. Slutligen flyttar vi femman och nian åt vänster, så att vi får en riktig tablå...

1	2	3	4	6	7	8
5	9					

och vi är klara med exemplet. Den algoritm vi hittat på fungerar inte som bevis i matematisk bemärkelse. Det stora problemet är att vi inte övertygat oss om att alla möjliga tablåer verkligen kan skapas med hjälp av algoritmen. Vill vi vara matematiskt korrekta bör vi, för att försäkra oss om detta, hitta en algoritm B som tar ett $y \in Y$ och ger ett $x \in X$ och för vilken

$$B(A(x)) = x.$$

Kan vi bevisa att detta gäller för alla x så har vi bevisat att $|X| = |Y|$, det vill säga att mängderna har lika många element. Det var dock lite svårt att göra ett ordentligt bevis för detta, så vi nöjer oss här med att beskriva en algoritm som potentiellt uppfyller denna egenskap:

- Flytta det undre tal som står längst åt höger, längre åt höger tills dess att det, om vi flyttade det ett steg till, skulle få ett tal större än sig självt över sig.
- Upprepa för övriga tal, från höger till vänster. Det finns två orsaker till att ett tal inte kan flyttas längre till höger: antingen att talen i övre raden är för stora, eller att "flyttvägen" blockeras av ett annat redan flyttat tal i den undre raden.
- Fortsätt tills dess att alla tal i den undre raden är flyttade.
- Med början från vänster, flytta ner de tal i den övre raden som står över luckor i den undre raden, tills dess att hälften av talen står i den undre raden.

Med detta lämnar vi försöken att hitta en bijektion mellan Young–tablåer och val av tal. I kapitel (2.4) använder vi istället det förarbete som gjorts i kapitel (2) för att, med hjälp av Ballot–talen, matematiskt bevisa att antalet Young–tablåer av längd n med högst två rader är $\binom{n}{\lceil n/2 \rceil}$.

2.2 Ballot–tal

Vi har nu, om inte bevisat matematiskt, så åtminstone intuitivt gjort klart för oss att $T(n)$ faktiskt är $\binom{n}{n/2}$. Under lösningsprocessen skapade vi en tabell med värden för $B(a, b)$, och dessa värden vet vi ingenting om. Vi ser också att huvuddiagonalen i tabellen tycks vara Catalantalen, som uppenbarligen har något samband med detta problem. För att komma vidare använder vi samma konstgrepp som tidigare; databasen på Internet, men den här gången matar vi in tal från rad 5 (bara för att välja någon) i tabellen. Vi får då reda på att vår tabell faktiskt kallas Catalans triangel, att den heter **A009766**, och att den har anknytning till något som kallas Ballot–talen, vilka dyker upp som lösningen till ett problem som har stora likheter med vårt; ett problem vi skall återkomma till senare.

I en artikel av Ira Gessel [6] (som för övrigt studerat för Richard Stanley, som vi tidigare nämnt i samband med Catalantalen) finns en mycket elegant härledning av en sluten formel för Ballot–talen, som mycket enkelt kan fås att passa de definitioner vi tidigare gjort i samband med tvåradiga Young–Tablåer.

Gessel utgår ifrån en differensekvation snarlik våra ekvationer (3) och (4):

$$\begin{aligned} B(n,k) &= B(n-1,k) + B(n,k-1) && (6)\\ B(0,0) &= 0 && (7)\\ B(1,0) &= 1 && (8)\\ B(0,1) &= -1 && (9) \end{aligned}$$

Istället för att låta $B(0,1) = 0$ som vi hade (eftersom det inte finns några tablåer som bara har en ruta i den undre raden) gör han B antisymmetrisk, så att $B(n,k) = -B(k,n)$. Anledningen är att detta leder till en *mycket* enklare formel för $B(n,k)$, även för de värden på n och k där $n > k$. Utifrån dessa ekvationer konstruerar Gessel en genererande funktion.

2.3 Genererande funktioner

En genererande funktion är en kontinuerlig funktion vars serieutveckling har de (hel)tal man är intresserad av som koefficienter. Antag att vi är intresserade av följden $1, 1, 2, 3, 5, 8, \ldots$ vilken bestäms av differensekvationen $a_{k+1} = a_{k-1} + a_k$ samt begynnelsevillkoren $a_0 = 0$ och $a_1 = 1$ (det vill säga Fibonaccitalen). Den genererande funktionen $F(x)$ för denna talföljd är då, per definition,

$$\sum_{n=0}^{\infty} a_n x^n.$$

Den viktigaste poängen med genererande funktioner är deras användbarhet för att representera och få fram slutna uttryck för talföljder som är definierade i form av differensekvationer. Genererande funktioner används i samband med differensekvationer på samma sätt som Laplace–transformen i samband med differentialekvationer. För att demonstrera detta skall vi följa ett exempel ur Herbert Wilfs bok *generatingfunctionology* [19], där han tar fram en formel för Fibonaccitalen med hjälp av genererande funktioner. På samma sida som detta exempel finns en punktlista där Wilf redogör för den generella metod man använder när man löser problem med hjälp av genererande funktioner, en lista som är så användbar att den förtjänar att återges i sin helhet:

1. Försäkra dig om att de värden för den fria variabeln (säg n) där differensekvationen är uppfylld är klart avgränsade.

2. Ge ett namn åt den genererande funktionen som du letar efter, och skriv ut funktionen i termer av den okända talföljden (det vill säga, kalla den till exempel $A(x)$ och definiera den att vara $\sum_{n\geq 0} a_n x^n$).

3. Multiplicera båda sidorna av differensekvationen med x^n, och summera över alla värden av n där differensekvationen gäller.

4. Uttryck båda sidorna explicit i form av $A(x)$.

5. Lös ut den okända genererande funktionen ur de ekvationer som skapats.

6. Om du vill ha en exakt formel för talföljden som är definierad av differensekvationen, så försök att expandera $A(x)$ som en potensserie med hjälp av vilken metod som helst som du kan komma på. Speciellt, om $A(x)$ är en rationell funktion (en kvot av två polynom) så kommer detta att åstadkommas med hjälp av partialbråksuppdelning, följd av separat hantering av de respektive termerna.

Vi börjar alltså (punkt ett) med att konstatera att vår differensekvation

$$a_{n+1} = a_{n-1} + a_n$$

gäller för alla ($n \geq 1$). Vi kallar vår genererande funktion för $F(x)$, och definierar den som $\sum_{n\geq 1} a_n x^n$ (punkt två). Vi multiplicerar vänsterledet med x^n och summerar (punkt 3 och 4):

$$a_2 x + a_3 x^2 + a_4 x^3 + \ldots = \frac{F(x) - x}{x}.$$

Gör vi exakt samma sak med högerledet får vi:

$$(a_1 x + a_2 x^2 + a_3 x^3 + \ldots) + (a_0 x + a_1 x^2 + a_2 x^3 \ldots) = F(x) + xF(x)$$

Vi löser nu ut $F(x)$ (punkt 5):

$$F(x) = \frac{x}{1 - x - x^2}$$

Vi vill ha en explicit formel för fibonaccitalen, och då måste vi hitta potensserieutvecklingen av $\frac{x}{1-x-x^2}$. Det enklaste sättet att göra detta är att partialbråksuppdela, så vi slipper den kvadratiska termen i nämnaren. Wilf löser detta genom att först notera att

$$1 - x - x^2 = (1 - xr_+)(1 - xr_-) \quad (r_\pm = \frac{1 \pm \sqrt{5}}{2})$$

så vi kan skriva

$$\begin{aligned}\frac{x}{1-x-x^2} &= \frac{x}{(1-xr_+)(1-xr_-)} \\ &= \frac{1}{(r_+-r_-)}\left(\frac{1}{(1-xr_+)}-\frac{1}{(1-xr_-)}\right) \\ &= \frac{1}{\sqrt{5}}\left\{\sum_{j\geq 0} r_+^j x^j - \sum_{j\geq 0} r_-^j x^j\right\}\end{aligned}$$

Och enligt definitionen av vår genererande funktion så är alltså a_n koefficienten framför x^n i den serieutveckling vi fått fram, det vill säga

$$a_n = \frac{1}{\sqrt{5}}(r_+^n - r_-^n) \quad (n = 0, 1, 2, \ldots) \tag{10}$$

är en explicit formel för Fibonaccitalen. Det verkar smått osannolikt att uttrycket (10) skall vara heltaligt för alla n, men det är det.

2.4 Bevis med hjälp av Ballot–talen

Vi återgår nu till våra försök att med Ira Gessels hjälp ta fram en sluten formel för Ballot–talen, som alltså även beskriver antalet tvåradiga tablåer med en viss form. Gessel går direkt från ekvationerna (6)–(9) till uttrycket

$$(1-x-y)\sum_{m,n=0}^{\infty} B(m,n)x^m y^n = x - y$$

som faktiskt är ekvivalent med dessa ekvationer. I vänsterledet har vi motsvarande $B(m,n) - B(m-1,n) - B(m,n-1)$, och högerledet svarar mot begynnelsevillkoren. Vi kan nu lösa ut vår genererande funktion (som vi inte givit något namn):

$$\sum_{m,n=0}^{\infty} B(m,n)x^m y^n = \frac{x-y}{1-x-y}$$

och härifrån få de slutna uttryck för Ballot–talen som vi söker:

$$B(m,n) = \binom{m+n-1}{m-1} - \binom{m+n-1}{m} = \frac{m-n}{m+n}\binom{m+n}{m}. \tag{11}$$

Formel (11) svarar mot följande tabell:

$m \diagdown n$	0	1	2	3	4	5	6
0	0	-1	-1	-1	-1	-1	-1
1	1	0	-1	-2	-3	-4	-5
2	1	1	0	-2	-5	-9	-14
3	1	2	2	0	-5	-14	-28
4	1	3	5	5	0	-14	-42
5	1	4	9	14	14	0	-42
6	1	5	14	28	42	42	0

De negativa talen i tabellen har ingen egentlig kombinatorisk betydelse. Kombinatorikern Percy A. MacMahon, vars arbete i början av 1900–talet [11] haft stor betydelse för den senare kombinatoriska forskningen, kallade en sådan funktion för "redundant genererande funktion". De egentliga Ballot–talen är de tal i tabellen som är positiva.

Vi kan nu bevisa det vi kom fram till i det inledande kapitlet, nämligen att antalet Young–tablåer av storlek k med en eller två rader är $\binom{k}{\lceil k/2 \rceil}$. Vår algoritm för att konstruera tablåer ledde till ekvationerna (3) och (4). För $m > n$ är dessa ekvationer identiska med ekvationerna för Ballot–talen, och vi kan identifiera $B(m,n)$ som antalet tablåer med $m-1$ tal i den övre raden och n tal i den undre.

Antalet tablåer med k rader ges alltså som summan av de $B(m,n)$ för vilka $m+n = k+1$ och $m > n$. Vi har nu att

$$\begin{aligned}
\sum_{\substack{m>n \\ m+n=k+1}} B(m,n) &= \sum_{i=0}^{\lfloor k/2 \rfloor} B(k+1-i,i) \\
&= \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{k-i} - \binom{k}{k-i+1} \\
&= \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{k-i} - \sum_{j=-1}^{\lfloor k/2 \rfloor - 1} \binom{k}{k-j} \\
&= \binom{k}{k-(\lfloor k/2 \rfloor)} - \binom{k}{k+1} \\
&= \binom{k}{\lceil k/2 \rceil}
\end{aligned}$$

vilket visar påståendet.

2.5 Ballot–talens ursprung

Ballot–talen har fått sitt namn från följande problem: Antag att det varit val, och kandidaterna A och B fått a respektive b röster. Om $a > b$ hur stor är då sannolikheten att A haft fler röster än B under hela rösträkningen? Denna fråga ställdes för första gången i slutet av 1800-talet av fransmannen M. Bertrand, och fick sitt svar 1887 i en artikel av D. André [7].

Det finns en direkt koppling mellan tablåer med högst två rader och de röstningsförlopp vi är intresserade av här, genom att ett tal i den övre raden svarar mot en röst på kandidat A och ett tal i den undre raden svarar mot en röst på kandidat B.[4]

Det är lätt att inse att det totala antalet röstförlopp som kan inträffa är $\binom{a+b}{a}$. Om vi kallar antalet förlopp där A hela tiden haft fler röster än B för $B(a,b)$ (för att vara specifika får vi säga att $B(a,b)$ räknar antalet möjligheter efter det att a fått den första rösten, eftersom A inte har fler röster än B när båda har noll röster) så är den sannolikhet vi söker

$$\frac{B(a,b)}{\binom{a+b}{a}}. \tag{12}$$

Sätter vi in uttrycket för ballottalen (11) i (12) ser vi att lösningen till Ballotproblemet är $\frac{a-b}{a+b}$.

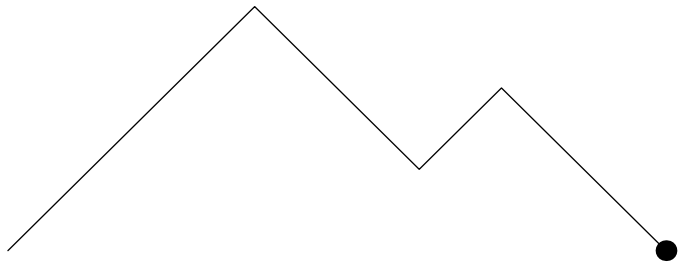
2.6 Dyckvägar

En mycket användbar teknik för att visualisera det fenomen som Ballot–talen och de tvåradiga tablåerna beskriver är Dyckvägar[5]. En Dyckväg är en

[4] Att kandidat A under hela förloppet måste *leda*, och sedan *vinna* svarar i tablåvärlden mot att vi bara är intresserade av tablåer där den övre raden är längre än den undre, samt den första ettan redan utplacerad, men detta är något vi inte skall fästa någon större vikt vid.

[5] Walther Franz Anton von Dyck (1856–1934) arbetade med gruppteori, bland annat för Felix Klein.

väg i ett heltalsgitter från punkten $(0,0)$ till en punkt $(2n,0)$ som: 1) bara innehåller steg av typen $(1,1)$, det vill säga snett upp åt höger, och $(1,-1)$, det vill säga snett ned åt höger och 2) aldrig går under x-axeln. Figur (2) visar en typisk Dyckväg. Antalet Dyckvägar av längd $2n$ räknas av Cata-



4	5	7	1	2	8	3	6
1	2	3	4	5	6	7	8

Figur 2: Bilden visar en typisk Dyckväg. Under Dyckvägen visas den motsvarande 3412-undvikande involutionen.

lantalen, och Dyckvägar som slutar i punkten (n,k) (som alltså egentligen inte är Dyckvägar, men ändå brukar kallas "Dyckvägar som slutar i punkten ...") räknas av Ballot–talen. Med detta avslutar vi undersökningen av de tvåradiga tablåerna.

3 Rekapitulation

När vi undersökte kombinatoriken kring Young–Tablåer med två rader fick vi användning av en rad olika kombinatoriska tekniker: En tabell med exempel ledde till förkastandet av vår första hypotes, att vårt problem beskrevs av Catalantalen. Genom att föreställa oss en stegvis konstruktion av våra kombinatoriska objekt (dvs de tvåradiga Young–Tablåerna) kunde vi formulera en differensekvation som, tillsammans med begynnelsevillkor, fullständigt beskrev problemet. Detta kunde vi betrakta som en första lösning på problemet, eftersom vi då relativt enkelt (speciellt genom att skriva ett litet datorprogram) kunde ta fram antalet tablåer för godtyckliga värden på n, antalet rutor i tablåerna. Med hjälp av vår differensekvation fick vi på köpet,

genom att göra en två–dimensionell tabell, värden för antalet tablåer med en specifik form, något vi inte frågade efter inledningvis. För att komma vidare utnyttjade vi en stor databas över heltalsföljder (The On–Line Encyclopedia of Integer Sequences), från vilken vi fick fram att de tal vi beräknade faktiskt hade en mycket enkel sluten formel i form av den centrala binomialkoefficienten $\binom{n}{\lceil n/2 \rceil}$. För att förklara detta faktum konstruerade vi sedan en bijektion mellan de kombinatoriska objekt som $\binom{n}{\lceil n/2 \rceil}$ beskriver, och de tvåradiga Young–Tablåerna.

Eftersom också de övriga talen i tabellen, och inte bara de diagonalsummor som beskriver antalet Young–Tablåer med två rader, väckt vår nyfikenhet skickade vi utdrag ur tabellen till databasen på Internet, och vi fick då kontakt med de så kallade Ballot–talen. Med Ira Gessel vid vår sida kunde vi sedan härleda en formel för hela tabellen; en process som gick via genererande funktioner, vilka vi också testade på ett lite enklare exempel hämtat ifrån en bok av Herbert Wilf. För att sluta säcken var vi tvungna att utföra några matematiska operationer på formeln för Ballot–talen så att vi fick fram den centrala binomialkoefficienten, vilken vi ursprungligen var ute efter.

Som en avslutning gick vi sedan till botten med det nära sambandet mellan det problem kring röstning som givit namnet åt Ballot–talen och våra tvåradiga Young–Tablåer. Detta med hjälp av en annan artikel (som vi hittat genom att söka efter information om Ballot–problemet) författad av av Peter Hilton och Jean Pedersen, där vi också fick bekanta oss med Dyckvägar.

4 Young–Tablåer

En *permutation* av talen $1, \ldots, n$ är ett sätt att flytta om ordningen på dessa tal. Så är till exempel 1423 och 3412 permutationer av 1234. En permutation kan ses som en funktion som går från permutationer till permutationer[6] (jag tar hjälp av en lärobok i diskret matematik skriven av Norman L. Biggs [1]). Om talet k står på plats i i en permutation σ betyder detta att det tal som står på plats i i den permutation som σ opererar på skall flyttas till plats k. Låter vi till exempel 4132 operera på 2341 skall vi

[6]Observera att detta kan vara förvirrande: permutationer är både funktioner, och det som funktionerna opererar på.

1. flytta tvåan till plats fyra,
2. flytta trean till plats ett,
3. flytta fyran till plats tre (den får alltså stå kvar på samma plats), och
4. flytta ettan till plats två.

Detta ger oss permutationen 3142. Mängden av alla permutationer av talen $1, \ldots, n$ (permutationer med längden n) kallas för S_n, och den innehåller $n!$ element.

En *involution* är en speciell permutation som bara tillåter att elementen i den permutation den opererar på antingen står kvar på samma plats, eller att två element byter plats. En konsekvens av detta är att om man låter en involution operera två gånger på samma permutation så får man tillbaka den permutation man hade från början. Permutationen 14523 är ett exempel på en involution. Vi kallar mängden av alla involutioner av längd n för I_n. Det finns ingen enkel formel för antalet involutioner av längd n, men den talföljd som beskriver antalet involutioner börjar

$$1, 1, 2, 4, 10, 26, 76, 232, 764, 2620 \ldots \tag{13}$$

Ett förbluffande faktum är att denna talföljd även beskriver antalet Young–Tablåer med n rutor. Vi skall med hjälp av ett avsnitt ut matematikern och datalogen Donald E. Knuths extremt betydelsefulla bok "The Art of Computer Programming" [8], försöka ge en antydan till förklaring av varför det förhåller sig på detta sätt.

Grunden i resonemanget är två algoritmer på en typ av tablåer som uppfyller alla krav som finns på Young–Tablåer förutom att de inte måste innehålla alla tal $1, \ldots, n$:

- En algoritm för att lägga till ett godtyckligt tal (som inte redan finns i tablån) till en tablå.
- En algoritm för att ta bort ett av talen från en tablå.

4.1 Algoritm I

Denna algoritm (och dess invers) kallas för *Robinson–Schenstedt–Knuth algoritmen* (förkortas RSK) efter de tre män som bidragit till dess slutgiltiga form. Låt P vara en tablå och x ett tal som inte finns i P. Vi sätter in x i P genom att rad för rad jämföra tal i P med x. Vi börjar i den översta raden.

Om x är större än alla tal i den rad vi undersöker, så placera x längst till höger i denna rad. Ta annars det tal y i denna rad som är närmast större än x och placera in x i dess ställe. Gå vidare och försök placera in y i nästa rad i tablån. Om det inte finns någon nästa rad, så placera y längst till vänster i en ny understa rad.

Exempel: Antag att

$$P = \begin{array}{|c|c|c|}\hline 1 & 4 & 5 \\ \hline 2 & 6 \\ \cline{1-2}\end{array}$$

och vi skall placera in $x = 3$. Vi börjar då med att jämföra 3 med talen 145 och konstaterar att 4 är det tal som är närmast större än 3. Alltså byter vi ut 4 mot 3, vilket ger tablån

$$\begin{array}{|c|c|c|}\hline 1 & 3 & 5 \\ \hline 2 & 6 \\ \cline{1-2}\end{array}$$

och vi skall nu placera in 4 i den andra raden. Eftersom 6 är det tal som är närmast större än 4 så byter vi ut 6 mot 4 vilket ger

$$\begin{array}{|c|c|c|}\hline 1 & 3 & 5 \\ \hline 2 & 4 \\ \cline{1-2}\end{array}$$

och vi placerar nu slutligen 6 underst i tablån:

$$\begin{array}{|c|c|c|}\hline 1 & 3 & 5 \\ \hline 2 & 4 \\ \cline{1-2} 6 \\ \cline{1-1}\end{array}$$

4.2 Algoritm D

Algoritm D är en "invers" av I på det sättet att om vi börjat med en tablå P, satt in talet y och detta resulterat i att det sista tal som sattes in i tablån

(insättningsalgoritmen innebär ju att man får en sekvens av olika tal som man placerar in i olika rader) hamnade på position (i, j) och vi därmed fått tablån P', får vi, om vi använder D för att ta bort det tal som står på (i, j) i P' tillbaka tablån P och talet x.

På samma sätt gäller att om vi använder algoritm D på tablån P och börjar med att ta bort talet som står på (i, j) och detta resulterar i tablån P' och talet y, så kan vi få tillbaka P genom att använda I på tablån P' och talet x.

Att göra I baklänges innebär att vi börjar på en position (i, j) i tablån (rad i, kolumn j) och plockar ut detta tal x. Vi går sedan till raden ovanför (om det finns någon sådan rad) och byter ut det tal y som är närmast mindre än x med x. På samma sätt fortsätter vi med alla rader.

Exempel: Antag att

$$P = \begin{array}{|c|c|c|}\hline 1 & 4 & 5 \\ \hline 2 \\ \cline{1-1} 3 \\ \cline{1-1} \end{array}$$

och vi skall ta bort talet på $(2, 1)$, det vill säga tvåan. Vi börjar då med att ta bort tvåan, vilket ger

$$\begin{array}{|c|c|c|}\hline 1 & 4 & 5 \\ \hline 3 \\ \cline{1-1} \end{array} \quad .$$

Vi tittar sedan på raden ovanför, som innehåller 145 och byter ut talet som är närmast mindre än 2, dvs 1 mot tvåan, vilket ger

$$P' = \begin{array}{|c|c|c|}\hline 2 & 4 & 5 \\ \hline 3 \\ \cline{1-1} \end{array} \quad .$$

Man kan se att om vi använder I för att placera in 1 i denna tablå, så får vi tillbaka P.

4.3 Young–Tablåer och involutioner

Med hjälp av algoritm I kan vi, genom att sätta in ett tal i taget, skapa en tablå P med hjälp av en permutation σ. Men tydligen är antalet tablåer mindre än antalet permutationer, så det måste gå att skapa samma tablå på många olika sätt. Man kan hålla ordning på konstruktionsprocessen med

hjälp av en andra tablå, där vi registrerar i vilken ordning elementen i tablån placerades in. Antag alltså att vi har en permutation, till exempel

$$\binom{51432}{12345}.$$

Vi bygger nu upp tablån P med hjälp av permutationen σ. Men om inplacerandet av talet x med ordningsnummer k i permutationen resulterade i att rutan (i, j) skapades i P, så placerar skapar vi en motsvarande ruta (i, j) i tablån Q och placerar där talet k. Med hjälp av Q kan vi nu, om vi vill, använda D upprepade gånger på P för att få tillbaka σ.

Permutationen 51432 leder till följande sekvens av tablåer

	P	Q
Sätt in 5	5	1
Sätt in 1	1 5	1 2
Sätt in 4	1 4 5	1 3 2
Sätt in 3	1 3 4 5	1 3 2 4
Sätt in 2	1 2 3 4 5	1 3 2 4 5

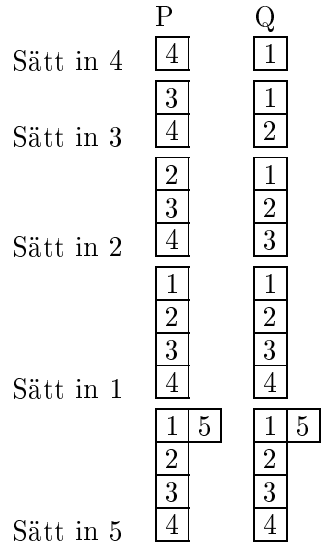
Detta resonemang ger oss en bijektion mellan *ordnade par av tablåer* (P, Q) av storlek n där båda tablåerna har samma form och permutationer av storlek n. Vi kan skriva detta som

$$\sum_{\lambda \vdash n} f_\lambda^2 = n!$$

där $\lambda \vdash n$ betyder att λ är en form för en tablå med n rutor, och f_λ är antalet tablåer som har formen λ.

Ett faktum som är ganska förvånande är att om permutationen $\binom{4132}{1234}$ ger tablåparet (P, Q) så ger den inversa permutationen (som kan fås genom att byta plats på de två raderna som beskriver permutationen, och sedan sortera kolumnerna så att den undre raden är sorterad) tablåparet (Q, P). Involutioner är, som vi konstaterade i avsnitt (4), sina egna inverser. Detta leder till

att om vi skapar tablåer med hjälp av en involution så måste $(P, Q) = (Q, P)$. Alltså måste $P = Q$. Eftersom varje involution ger en specifik tablå är antalet involutioner lika med antalet tablåer. För att försäkra oss om att vi verkligen bara får en tablå testar vi med involutionen $\binom{43215}{12345}$:



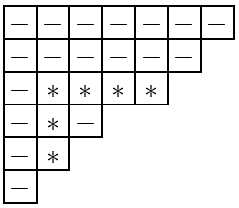
5 Young–Tablåer med fixerad form

Vi har tidigare studerat tablåer som fick ha högst två rader. Antag att vi bestämmer att tablån måste ha formen λ där λ är en lista med radlängder $\ell_1 > \ell_2 > \cdots > \ell_k$. Den fascinerande formeln för antalet tablåer är då [8]

$$f_\lambda = \frac{n!}{\prod_{(i,j)\in\lambda} h(i,j)} \tag{14}$$

där $h(i,j)$ är något som kallas "hook"–längden, fritt översatt till "kroklängden", för rutan (i,j). Kroklängden är det antal rutor som fås i den "krok" som bildas av rutorna till höger och rakt nedanför (i,j) i tablån, inklusive

rutan (i, j) själv. Alltså får vi i tablån



kroklängden 6 för rutan $(3, 2)$.

Fram tills 1997 fanns det inget enkelt bevis för sats (14), men då konstruerade Jean–Christophe Novelli, tillsammans med Igor Pak och Alexander V. Stoyanovskii ett bijektivt bevis [12] som anses vara beviset med stort B av denna sats [9]. Novelli m. fl. konstruerar algoritmer liknande algoritmerna I och D med vars hjälp de skapar en bijektion mellan de två mängderna

1. Tablåer med n rutor, med en given form λ som inte har några restriktioner vad gäller storleksförhållandet mellan rutorna. Det finns uppenbarligen $n!$ sådana tablåer.
2. Par (P, H) där P är en Young–Tablå och H är en tablå där varje ruta (i, j) innehåller ett tal mellan 1 och $h(i, j)$, något de kallar för en "krokfunktion".

Eftersom antalet krokfunktioner med form λ är $\prod_{(i,j)\in\lambda} h(i, j)$, så säger bijektionen att

$$f_\lambda \times \prod_{(i,j)\in\lambda} h(i, j) = n!$$

vilket är ekvivalent med ekvation (14).

5.1 Young–Tablåer och generaliserade Ballot–tal

Om vi tänker tillbaka på vad vi skrivit tidigare om Ballot–problemet inser vi att en tablå med en bestämd form motsvarar ett röstningsförlopp där antalet kandidater ges av antalet rader, och deras slutplaceringar av radernas respektive längd. Det var relativt enkelt att räkna antalet röstförlopp då

vi bara hade två kandidater, och detta resultat motsvarade exakt antalet tablåer med två rader med en viss form.

Antalet röstförlopp med k kandidater, där kandidaterna slutat på positioner $\ell_1 > \ell_2 > \cdots > \ell_k$ måste ges av kroklängdsformeln. Ballotproblemet för flera kandidater har studerats helt oberoende av teorin kring Young–Tablåer och har lösts genom att betrakta röstförloppet som en Dyckväg[7] i flera dimensioner.

Problemet, vars ursprung kan sträcka sig ändå tillbaka till en artikel av Abraham De Moivre (1667–1754) 1711, har sedan dess utvecklats och generaliserats, och när vi söker efter en formel för antalet röstförlopp, givet att k kandidater som slutligen fått $a_1 > a_2 > \cdots > a_k$ röster hittar vi följande formel

$$N = n! \det(\frac{1}{u(i,j)!}) \tag{15}$$

som kräver lite ytterligare förklaringar, i en artikel av Michael Filaseta [4].

Filaseta utgår från en generaliserad variant av Ballot–problemet där man kan bestämma *hur mycket* kandidaterna måste leda över varandra under röstförloppets gång. Detta specificeras av tal $t_1, \ldots, t_k$, som anger att

$$A_i(m) < A_{i-1}(m) + t_i \qquad (m = 1, \ldots, n\,, \quad i = 1, \ldots, k) \tag{16}$$

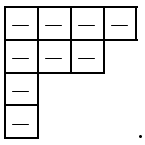
där $A_i(m)$ är antalet röster kandidat i har fått när m personer har röstat. Vi tillåter att kandidaterna har lika många röster, vilket i ekvation (16) motsvaras av att $t_1 = \cdots = t_k = 1$. (Det icke intuitiva sättet att formulera "$\leq$" beror på att vi följer Filasetas artikel.) Filaseta låter $u(i,j) = a_j + S(i,j)$, där $S(i,j) = \sum_{v=1}^{i} t_v - \sum_{v=1}^{j} t_v$, vilket för oss leder till att $u(i,j) = a_j + i - j$.

Vårt tidigare resonemang säger att uttrycket (15) skall vara ekvivalent med kroklängdsformeln (14), det vill säga:

$$n! \det(\frac{1}{(a_j + i - j)!}) \equiv \frac{n!}{\prod_{(i,j)\in\lambda} h(i,j)} \Leftrightarrow \det(\frac{1}{(a_j + i - j)!}) \equiv \frac{1}{\prod_{(i,j)\in\lambda} h(i,j)} \tag{17}$$

[7] Egentligen inte riktiga Dyckvägar, eftersom de avslutas i punkten $(\ell_1, \ldots, \ell_k)$ som inte nödvändigtvis ligger på x–axeln.

Vi testar på tablån



.

Kroklängdsformeln blir:

$$\frac{1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6\cdot 7\cdot 8\cdot 9}{7\cdot 5\cdot 4\cdot 3\cdot 2\cdot 2\cdot 1\cdot 1\cdot 1} = 216.$$

Matrisen vi får i Filasetas formel är

$$\begin{pmatrix} \frac{1}{24} & \frac{1}{2} & 0 & 0 \\ \frac{1}{120} & \frac{1}{6} & 1 & 0 \\ \frac{1}{720} & \frac{1}{24} & 1 & 1 \\ \frac{1}{5040} & \frac{1}{120} & \frac{1}{2} & 1 \end{pmatrix}$$

vars determinant är $\frac{1}{1680}$. Multiplicerar vi detta med $9!$ så får vi 216, samma sak som vi fick för kroklängdsformeln. Det tycks dock inte finnas något enkelt sätt att se att de två formlerna är ekvivalenta.

6 Young–Tablåer med högst n rader

Att räkna tablåer med begränsat antal rader större än två är ganska svårt. Amitai Regev [15] fick 1981 fram uttrycket

$$T_3(n) = \sum_{i=0}^{n/2} \binom{n}{2i} C_i \tag{18}$$

där C_i är det i:te Catalantalet, för antalet Young–Tablåer med högst tre rader. Dominique Gouyou–Beauchamps fick 1989 fram uttrycken

$$T_4(n) = C_{\lfloor (n+1)/2 \rfloor} C_{\lceil (n+1)/2 \rceil} \tag{19}$$

och

$$T_5(n) = 6 \sum_{i=0}^{n/2} \binom{n}{2i} C_i \frac{(2i+2)!}{(i+2)!(i+3)!} \tag{20}$$

för tablåer med högst fyra respektive fem rader. I en artikel [5] från 1990, som visat sig mycket användbar för den fortsatta forskningen kring Young–Tablåer, ger Ira Gessel en generell metod för att ta fram dessa uttryck med hjälp av symmetriska funktioner.

Symmetriska funktioner är en sorts genererande funktioner som har många fascinerande egenskaper. Även om själva idéen som ligger bakom de symmetriska funktionerna (som uttömmande redovisas i boken *Symmetric functions and Hall polynomials* av I. G. MacDonald [10]) inte är svår att förstå[8] krävs det en hel del ganska krånglig matematik för att nå fram till de resultat som används för att räkna tablåer med begränsat antal rader.

7 Mönsterundvikande

En konsekvens av algoritm I som vi inte nämnt är att om man skapar en tablå P med hjälp av en permutation π så bestäms antalet rader i P av den längsta avtagande sekvensen i π. En avtagande sekvens av längd k är en följd av tal $a_1 > \cdots > a_k$ som står i denna ordning i π. Begreppet *längsta avtagande sekvens* är ett specialfall av det mer generella begreppet *mönster* på permutationer. Vi hämtar nu, för att underlätta de fortsätta resonemangen, en del notation från Anders Claessons licenciatuppsats "Generalised Pattern Avoidance" [2].

Ett mönster är en permutation $\sigma \in S_k$. En permutation $\pi \in S_n$ *undviker* σ som det inte finns någon delföljd i π där talen[9] är i samma relativa ordning som i σ. Detta betyder till exempel att $\pi \in S_n$ undviker 132 om det inte finns några $1 < i < j < k \leq n$ sådana att[10] $\pi(i) < \pi(k) < \pi(j)$, det vill säga det största talet står i mitten och det minsta till vänster.

På ett liknande sätt som Claesson definierar vi den delmängd av S_n (permutationer av längd n) respektive I_n (involutioner av längd n) som undviker mönstret σ som $S_n(\sigma)$ respektive $I_n(\sigma)$. Vi kommer i fortsättningen även att intressera oss för mängden involutioner utan fixpunkter av längd $2n$ (det vill

[8] det handlar om funktioner i flera variabler, säg $f(x_1, \ldots, x_k)$ vars värde är oberoende av i vilken ordning man stoppar in variablerna.

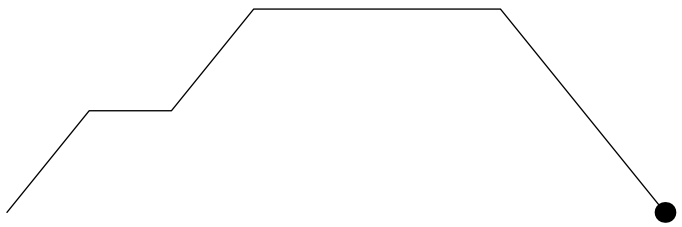
[9] Claesson arbetar med bokstäver snarare än tal.

[10] Här betyder $\pi(i)$ det tal som står på plats i i π.

säga involutioner som inte låter något av talen i permutationen den opererar på stå kvar på samma plats), vilken vi kallar för U_{2n}. I denna terminologi kan vi beskriva mängden av alla involutioner vars längsta avtagande sekvens är högst tre som $I_n(321)$, och antalet sådana involutioner som $|I_n(321)|$.

7.1 Motzkintal

Ekvation (18) som Regev [15] tagit fram för antalet tablåer med högst tre rader beskriver Motzkintalen[11] [3], vars enklaste tolkning förmodligen är i form av Motzkinvägar. Dessa är som Dyckvägar, förutom att det också



7	2	8	4	5	6	1	3
1	2	3	4	5	6	7	8

Figur 3: Bilden visar en typisk Motzkinväg. Under visas den motsvarande 4321–undvikande involutionen.

är tillåtet att gå steg av typen $(1, 0)$, det vill säga parallellt med x–axeln. Figur (3) visar en typisk Motzkinväg. De första Motzkintalen är

$$1, 2, 4, 9, 21, 51, 127, 323, \ldots \tag{21}$$

Algoritm I säger oss att Motzkintalen även räknar antalet involutioner vars längsta avtagande sekvens har längd tre. Vi har alltså tre till synes helt olika kombinatoriska objekt som, för varje storlek[12] på objekten, är lika många:

[11] Efter Theodor Motzkin, född 1918.

[12] I artiklar som behandlar kombinatoriska objekt används ofta termen "vikt" för att beskriva objektens storlek. Vi låter alltså vikten (storleken) av en Motzkinväg med n steg vara n, men det är även möjligt att "väga" Motzkinvägar på andra sätt.

1. Young–Tablåer med högst tre rader.

2. Involutioner vars längsta avtagande sekvens har längden tre.

3. Motzkinvägar.

Algoritmerna I och D ger oss en bijektion mellan tablåer och involutioner. Det vore fint om man kunde hitta någon motsvarande enkel koppling mellan antingen tablåer och Motzkinvägar, eller involutioner och Motzkinvägar. Vi skall strax undersöka detta närmare.

7.2 Mönsterundvikande och cykelstrukturer

Mönsterundvikande på permutationer, dvs bijektioner för identiteter $|S_n(\sigma)| \equiv |X_n|$ med varierande mönster σ och kombinatoriska objekt X är ett relativt nytt forskningsområde inom kombinatoriken. Med datorns hjälp[13] och Sloanes [16] databas på Internet kan man med hjälp av de inledande termerna i sekvensen, dvs $|S_1(\sigma)|, |S_2(\sigma)|, |S_3(\sigma)|, \ldots$ göra kvalificerade gissningar av vilken talföljd man har att göra med. Vi tar med hjälp av vårt datorprogram fram så många termer att ett av följande alternativ inträffar:

1. Sökningen i databasen säger att talföljden är okänd.

2. Vi hittar exakt *en* talföljd som inleds av de termer vi fått fram. Vi antar då att det vi studerar, till exempel $S_n(\sigma)$ för något σ, har något gemensamt med den kombinatorik som finns beskriven i databasen i anknytning till talföljden. Med detta som grund kan vi gå vidare och till exempel försöka hitta en bijektion mellan de objekt vi studerar, och något som finns beskrivet i databasen.

Vi tabellerar i kapitel (A), Tabeller och Bilder, resultatet av sökningar i Sloane för $S_n(\sigma)$, $I_n(\sigma)$, $U_{2n}(\sigma)$ (det vill säga permutationer, involutioner och involutioner utan fixpunkter) för alla möjliga $\sigma \in S_k$ $(k = 3, 4)$. Det visade sig att fem termer (som ibland kan vara ganska tidskrävande att ta fram) var tillräckligt för att få endast *en*, eller ingen, träff i Sloane. Observera att de namn som anges i tabellerna är just gissningar och antaganden om

[13] Vi har använt det utmärkta matematikprogrammet MATHEMATICA.

vilka talföljder vi har att göra med. I vissa fall, till exempel då vi har att göra med Catalantalen, finns det ingen större anledning att tvivla på att våra antaganden är riktiga. Tvärtom är det med antagandet att $|U_n(3142)|$ genererar, som det står i Sloane, "Number of 6–ary Lyndon words with trace 1 mod 6".

För att tydligare se eventuella samband i tabellerna, som är ganska många, kan det vara bra att visualisera mönstren. Det vi är ute efter är att se samband mellan mönster och de talföljder som genereras (speciellt om flera olika mönster tycks ge samma talföljd). För att underlätta detta sökande och upptäckande av samband visualiserar vi våra mönster med hjälp av små bilder som visar mönstrens cykelstruktur. Vi använder följande regler när vi konstruerar dessa bilder:

- En fixpunkt representeras av en punkt. Så skall till exempel $\cdot\cdot\cdot\cdot$ tolkas som mönstret (permutationen) 1234, som ju har fyra fixpunkter.
- En cykel av längden två representeras av en kurva eller en linje beroende på om cykeln gå mellan närliggande positioner i mönstret eller om den "innesluter" någonting. Bilden "–" skall tolkas som 21, medan mönstret 321, där alltså cykeln innesluter en fixpunkt, ritas $\frown$ där alltså cykeln blir en båge som går *ovanför* fixpunkten.
- En cykel med längd större än två ritas som en sluten kurva där
  - Ett hopp framåt till närmast framförliggande plats blir ett streck: "–".
  - Ett hopp framåt som innesluter någonting blir en båge *ovanför* det som innesluts.
  - Ett hopp bakåt ett steg blir ett streck: "–".
  - Ett hopp bakåt som innesluter någonting blir en båge *under* det som innesluts.

Tabellerna (1) och (2) som också hittas i kapitel (A), Tabeller och Bilder, visar hur mönster översätts till bilder enligt dessa regler.

Våra experiment (det är nödvändigt att nu, om inte förr, bläddra fram till tabellerna och titta på dem för att resten av texten i denna uppsats skall bli meningsfull) har onekligen gett oss en hel del ytterligare material att

arbeta med utöver sambandet mellan involutioner som undviker $\frown$ (4321) och Motzkinvägar. Med utgångspunkt från de tabeller vi tagit fram skall vi nu, helt i linje med den matematiska prosans praxis, göra några påståenden (som ibland knyter an till våra tidigare resonemang) och försöka bevisa dessa.

Ett centralt tema i det följande är förhållandet mellan mönster (i den vanliga enligt ovan definierade bemärkelsen), och cykelstrukturer. I många fall har dessa två sätt att beskriva en permutation inte någon uppenbar relation till varandra. Det visar sig dock att det åtminstone för involutioner och involutioner utan fixpunkter, finns intressanta kopplingar mellan dessa två begrepp. Ett exempel på en sådan koppling är att alla involutioner längd n som undviker *mönstret* 3412 även undviker den cykelstruktur som mönstret 3412 har (vilken vi alltså visualiserar med bilden $\frown\!\!\frown$). Mer förvånande är att även det omvända gäller. Vi skall strax definiera vad vi menar med att en permutation undviker en cykelstruktur.

För att tydliggöra skillnaden mellan mönster och cykelstrukturer definierar vi två operatorer $\otimes$ och $\odot$ vilka tar två permutationer av längd k och n och skapar en mängd med permutationer av längd $k+n$. Operatorerna beskriver två olika sätt att dela in en permutation i två disjunkta delar, den ena med avseende på *mönster*, den andra med avseende på *cykelstruktur*.

Givet två permutationer $p \in S_k$ och $\pi \in S_n$ säger vi att $\psi \in p \otimes \pi$ om det finns ett delord p' av ψ som i ψ har samma cykelstruktur som p, och π är den permutation man får om man tar bort delordet p' från ψ. Betydelsen av detta förstås enklast genom att tänka sig bilder av permutationerna p och π. Permutationerna i mängden $p \otimes \pi$ fås då genom att sträcka ut bilderna på längden, och lägga dem över varandra. Den kryssprodukt som är enklast att föreställa sig är produkten av en permutation p och identitetspermutationen (som består av enbart fixpunkter). Vi får då alla möjliga sätt att "skjuta in" fixpunkter i permutationen. Så är till exempel

$$21 \times 12 = \{\mathit{12}43, \mathit{13}2\mathit{4}, \mathit{14}32, 21\mathit{34}, 32\mathit{14}, 4\mathit{23}1\}$$

(siffror som hör till 12 är kursiverade) eller ekvivalent

$$- \times \cdot\cdot = \{\cdot\cdot -, \cdot - \cdot, \cdot \frown, - \cdot\cdot, \frown\!\!\cdot\cdot, \frown\!\!\cdot \cdot\}.$$

På liknande sätt som för $\otimes$ definierar vi nu $\odot$. Istället för att titta på cykelstrukturen i permutationerna intresserar vi oss nu för permutationerna som

mönster. Givet två permutationer $p \in S_k$ och $\pi \in S_n$ säger vi att $\psi \in p \odot \pi$ om det finns ett delord p' av ψ som motsvarar mönstret p och π är den permutation man får då man tar bort delordet p' från ψ.

Skillnaden mellan operatorerna $\otimes$ och $\odot$ ligger alltså i huruvida man ser en permutation p och ett delord p' (som ju inte är en permutation) som lika om de har samma cykelstruktur eller om man ser dem som lika om de motsvarar samma mönster. Om de motsvarar samma mönster kan man också säga att p är den permutation man får om man projicerar p' på mängden $\{1, 2, 3, \ldots\}$. Detta innebär i princip att man bara bevarar storleksförhållandet mellan talen i p'.

Som en generalisering av dessa definitioner skriver vi $\bigotimes p$ för unionen av de mängder man får då man applicerar $\otimes$ på samma permutation många gånger det vill säga $(p \otimes (p \otimes (\ldots \otimes p) \ldots))$. Vi skriver $\bigotimes^n p$ för $\otimes$ applicerad n gånger på p. Antag vidare att $A, B \subset S$. Vi låter då $A \otimes B$ vara unionen av de mängder som bildas för alla kombinationer av element mellan A och B.

För att exemplifiera hur våra definitioner kan användas har vi till exempel att $\bigodot^n 1 = S_n$, det vill säga mängden av alla permutationer av längd n. Om vi istället använder den andra operatorn får vi $\bigotimes^n 1 = 1234 \ldots n$. Eftersom 1 är en fixpunkt får vi mängden av alla permutationer av längd n som bara består av fixpunkter. Det finns bara en sådan permutation, och det är identitetspermutationen. Vi har också att

$$\bigotimes^n 21 = \{\text{involutioner av längd } 2n \text{ utan fixpunkter}\}.$$

Vi bevisar nu ett antal påståenden genom att se mönster som cykelstrukturer:

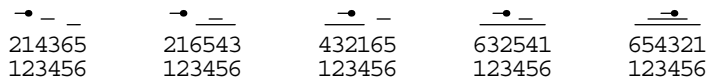
Påstående 1. *Antalet involutioner av längd $2n$ utan fixpunkter som undviker något av mönstren $\cdot-$, $-\cdot$, $\overset{\frown}{\cdot}$, $\overset{\frown}{\frown}$ och $\overset{\frown}{=}$ är C_n, det n:te Catalantalet.*

Bevis. Cykler av längd två kan parvis förhålla sig till varandra på tre olika sätt:

1. De kan ligga sida vid sida.
2. De kan överlappa varandra.

3. Den ena kan vara innesluten i den andra.

Involutioner som undviker $\frown$ har inga par av cykler av längd två som överlappar varandra. Figur (4) visar $U_6(\frown)$, där U_n står för mängden av alla

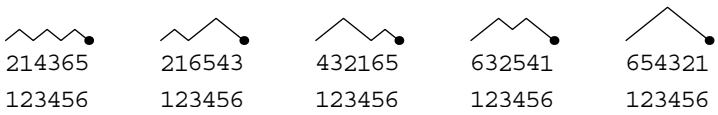


Figur 4: Visualisering av $U_6(\frown)$.

involutioner av längd n utan fixpunkter. En naturlig funktion från $U_n(\frown)$ till Dyckvägar (som ju räknas av Catalantalen) är att låta det första elementet i varje cykel bli ett $(1,1)$–steg och det andra elementet bli ett $(1,-1)$–steg i den motsvarande Dyckvägen. Att två cykler inte överlappar varandra motsvarar i Dyckvägen att;

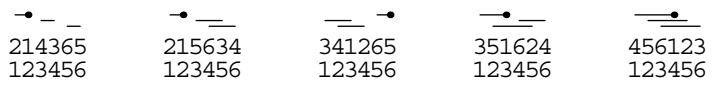
- varje cykel motsvarar ett (1,1) och ett (1,-1)–steg som ligger på samma höjd.
- de två steg som en cykel motsvarar aldrig har någonting mellan sig på samma eller lägre höjd.

Dessa två förhållanden gör att vi lätt kan bilda en motsvarande funktion från Dyckvägar till $U_n(\frown)$, genom att identifiera par av steg i Dyckvägen. Att samtidigt betrakta figurerna (4) och (5) skall räcka för att se hur bijektionen fungerar. Vi ser detta som ett bevis för att påståendet gäller för mönstret $\frown$.



Figur 5: De Dyckvägar som motsvarar $U_6(\frown)$. (Det vill säga alla Dyckvägar av längd 6.)

Involutioner som undviker $\frown$ har inga par av cykler av längd två där den ena cykeln är innesluten i den andra. Vi vill nu hitta en bijektion liknande den för mönstret $\frown$. Använder vi exakt samma algoritm, det vill säga omvandlar



Figur 6: Visualisering av $U_6(\frown)$.

det första elementet i varje cykel till ett $(1,1)$–steg och det andra till ett $(1,-1)$–steg, ser vi att detta faktiskt, för våra datorgenererade exempel, ger en bijektion. Hur skall vi tolka detta?

Vi kan konstruera en funktion från Dyckvägar av längd n till element i $U_n(\frown)$ genom att omvandla varje $(1,1)$–steg till första elementet i en cykel. Dessa cykler, vars andra element vi ännu inte vet, placerar vi i en "first in, first out" kö. Då vi i Dyckvägen, som vi traverserar från vänster till höger, kommer till ett $(1,-1)$–steg avslutar vi den cykel som står på tur i kön. Det inses lätt att detta verkligen ger involutioner som undviker $\frown$.

Mönstren $\cdot -$ och $- \cdot$ går båda att dela på mitten, utan att någon cykel går från den ena sidan till den andra. Antag att en permutation går att dela upp på detta sätt. Kan den då undvika något av mönstren? Ja, om den inte har någon cykel på någon av sidorna. Om vi begränsar oss till permutationer som består endast av cykler av längd två, det vill säga involutioner utan fixpunkter, kan vi konstatera att om de skall undvika något av dessa mönster, så får inga två cykler ligga bredvid varandra. En inte helt uppenbar konsekvens av detta är att de $n/2$ lägsta talen i permutationen ligger på de $n/2$ sista platserna. Vidare bestämmer dessa tal entydigt de $n/2$ största talens placering på de $n/2$ första platserna.

För att bevisa att antalet involutioner utan fixpunkter av längd $2n$ som undviker $\cdot -$ eller $- \cdot$ räknas av C_n kan vi först betrakta enbart de sista $n/2$ positionerna i involutionen. Bortser vi från första halvan av involutionen kan dessa tal placeras på C_n sätt, eftersom vi vet att antalet permutationer av längd n som undviker samma mönster räknas av C_n. Kan vi nu visa att denna placering av de $n/2$ sista talen alltid leder till att hela involutionen också undviker mönstret är vi klara.

Först kan vi konstatera att mönstret inte kan förekomma som en kombination av tal från båda sidor av mitten.

Det enklaste sättet att se att mönstret inte kan förekomma på den första

halvan är förmodligen med hjälp av ett motsägelsebevis. Antag därför att vi har en förekomst av mönstret $-\cdot$ på den första halvan. Vi skall nu visa att detta ger en förekomst av mönstret också på den andra halvan. Ty, det mellersta talet ligger längst till vänster. Detta motsvaras på den andra halvan av att det minsta talet hamnar i mitten. Det minsta talet ligger i mitten, vilket på den andra halvan gör att det mellersta talet hamnar längst till vänster. Slutligen är det största talet i mönstret längst till höger, vilket betyder att det största talet ligger lägst till höger även på den andra halvan. Eftersom ett mönster på första halvan alltså alltid leder till samma mönster på andra halvan, och andra halvan ju inte har några förekomster av mönstret, kan inte heller första halvan ha det.

Det samma gäller för mönstret $\cdot -$.

För mönstret $\overset{\cdot}{\frown}$ konstaterar vi att exakt samma involutioner utan fixpunkter undviker $\overset{\cdot}{\frown}$ som $\overset{-}{\frown}$. Detta beror på att $\overset{\cdot}{\frown}$ innebär att ingenting får vara inneslutet i en cykel av längd två, men eftersom vi inte tillåter några fixpunkter, så är det enda som kan vara inneslutet en cykel av längd två, vilket motsvarar mönstret $\overset{-}{\frown}$. □

Påstående 2. *Låt π vara en permutation i kryssprodukten $p_1 \otimes p_2$, där $p_1 \in U_n(\sigma)$ för $\sigma \in \{\overset{\cdot}{\frown}, \overset{-}{\frown}\}$ och $p_2 = e_m$ $(m > 0)$ (identitetspermutationen av längd m). Det vill säga π är en permutation som kan delas upp i två delar; en del som enbart innehåller cykler av längd två och som undviker mönstret σ, och en del som består av enbart fixpunkter.*

Då gäller att $\pi \in I_{n+m}(\sigma)$, det vill säga då undviker även π mönstret σ. Poängen är att vi kan "skjuta in" hur många fixpunkter vi vill i en involution utan fixpunkter som undviker σ, utan att för den skull få någon förekomst av detta mönster.

Bevis. Vi vill visa att om $\pi \notin I_n(\sigma)$, så finns det en permutation $p \in U_m$ sådan att $\pi \in p \otimes e_{n-m}$ och $p \notin U_n(\overset{\cdot}{\frown})$. Med andra ord betyder detta att om en eller flera fixpunkter i π ingår i mönstret $\overset{\cdot}{\frown}$ så finns det alltid ett antal cykler av längd två i π som också ger detta mönster.

Antag att vi har en permutation π i vilken det finns fyra element

$$\cdots a_1 \cdots a_2 \cdots a_3 \cdots a_4 \cdots$$

som motsvarar ett mönster σ. Vart och ett av dessa element är antingen en fixpunkt eller en del av en cykel av längd två. Man inser ganska lätt att inget av mönstren $\frown$ och $\smile$ går att skapa med hjälp av tre fixpunkter och en cykel av längd två. Figurerna (7), (8), (9) och (10) i appendix visar alla möjliga sätt att bilda de respektive mönstren med hjälp av en och två fixpunkter. Vi ser att alla dessa involutioner innehåller mönstret även i form av kombinationer av cykler av längd två, vilket bevisar påståendet. (Figur (11) visar alla sätt att bilda mönstret $--$. Här gäller inte det påstående vi visat.) □

Påstående 3. *Antalet involutioner som undviker något av mönstren* $\frown$ *och* $\smile$ *är det n:te Motzkintalet* M_n.

Bevis. Påstående (1) och (2) säger oss att varje permutation $\pi \in I_n(\sigma)$, $\sigma \in \{\frown, \smile\}$, kan skrivas som en produkt $p_1 \otimes p_2$ där $p_1 \in U_{2i}(\sigma)$ och p_2 består av $n-2i$ fixpunkter. Vi kan bestämma antalet element i $p_1 \otimes p_2$ med hjälp av något man kallar *multimängder*, och den där till hörande "binomialkoefficienten" $\left\langle {n \atop k} \right\rangle$ [17]. En multimängd är en mängd där samma element får förekomma flera gånger. På ett ekvivalent sätt betyder $\left\langle {n \atop k} \right\rangle$ antalet sätt att välja ut k element från en mängd med n element då det är tillåtet att välja samma element flera gånger. Vi har att

$$\left\langle {n \atop k} \right\rangle = \binom{n+k-1}{k}.$$

Fixpunkterna kan "placeras ut" i permutationen utan fixpunkter på $2i+1$ ställen, och vi skall placera ut $2i-n$ fixpunkter[14]. Detta ger att

$$\left\langle {2i+1 \atop n-2i} \right\rangle = \binom{2i+1+n-2i-1}{n-2i} = \binom{n}{n-2i} = \binom{n}{2i}.$$

Varje permutation i $U_{2i}(\sigma)$ ger alltså $\binom{n}{2i}$ permutationer i $I_n(\sigma)$. Eftersom $|U_{2i}(\sigma)| = C_i$ kan vi få det totala antalet permutationer i $I_n(\sigma)$ genom att summera:

$$|I_n(\sigma)| = \sum_{i=0}^{n/2} \binom{n}{2i} C_i. \tag{22}$$

som ger Motzkintalen. □

[14] Eller ekvivalent: de platta stegen kan placeras ut i Dyckvägen på $2i+1$ ställen.

Förmodan 1. *Antalet involutioner av längd $2n$ utan fixpunkter som undviker något av mönstren $\cdot\smile$; $\frown$, $\smile\cdot$ och $\frown\cdot$ är det n:te Delannoytalet D_n (A001850).*

Delannoytalen har mycket stora likheter med de tidigare undersökta Ballot–talen, och har dessutom lånat drag från Motzkinvägar [18]. De vägar vi här har att göra med har steg av typen $(1,0)$, $(0,1)$ och $(1,1)$, och Delannoytalet $D(a,b)$ är antalet sådana vägar från $(0,0)$ till punkten (a,b). Vi har här inga restriktioner för vägen, som vi har för både Dyck– och Motzkinvägar. Den visualisering detta ger av Delannoyvägarna motsvarar de visualiseringar vi gjort av Dyck– och Motzkinvägar roterade 90° motsols[15].

De *centrala* Delannoytalen, $D(n,n)$, vars inledande termer är

$$1, 3, 13, 63, 321, \ldots \tag{23}$$

genereras av differensekvationen

$$D(a,b) = D(a-1,b) + D(a,b-1) + D(a-1,b-1) \quad (D(0,0) = 1\,, \quad a,b > 0) \tag{24}$$

som kan jämföras med ekvationerna (3) och (4) som ger Ballot–talen.

Det tycks inte finnas något enkelt sätt att övertyga sig om att detta påstående är sant. Studerar man vilka involutioner som är tillåtna ser man att mönstren parvis är ekvivalenta: $U_n(\smile\cdot) \equiv U_n(\frown\cdot)$ och $U_n(\cdot\smile) \equiv U_n(\cdot\frown)$. De 13 involutionerna i $U_6(\smile\cdot)$ visas i figur (12).

Påstående 4. *Antalet involutioner utan fixpunkter av längd $2n$ som undviker mönstret $--$ är lika med antalet permutationer av längd n som undviker samma mönster, vilka kallas "vexillära". (Observera att $--$ är ekvivalent med 2143.)*

I detta bevis är det nödvändigt att se mönstret både som bilden $--$ och som tal, det vill säga 2143.

Vi kan bevisa detta på samma sätt som vi visade att $|U_{2n}(-\cdot)| = C_n$ (se beviset av påstående (1)). Mönstret $--$ säger att två cykler av längd två aldrig får ligga bredvid varandra. Alltså måste de antingen vara inneslutna

[15] se http://forum.swarthmore.edu/advanced/robertd/delannoy.html

i varandra, eller överlappa varandra (se definitioner i beviset av påstående (1)).

Av detta inser man att de sista n platserna i en involution utan fixpunkter av längd $2n$ som undviker $--$ måste innehålla de n lägsta talen. Dessa tal måste naturligtvis undvika $--$, vilket ger oss att antalet involutioner utan fixpunkter som undviker $--$ säkert inte kan vara större än V_n.

Man inser lätt (genom att titta på mönstret i talform: 2143) att en eventuell förekomst av mönstret inte kan innehålla tal från både den första och den andra halvan i involutionen utan fixpunkter, eftersom alla tal i den första halvan ju är större än de i den andra halvan. Det som återstår att visa är därför att mönstret inte kan förekomma på den första halvan, om det inte också förekommer på den andra halvan.

Antag därför att talen $a_1a_2a_3a_4$ på den första halvan motsvarar mönstret 2143 och att dessa fyra tal bildar cykel–par med talen $b_1b_2b_3b_4$ på den andra halvan. Betrakta nu talet a_1. Det motsvarar tvåan, vilket säger att det är det näst minsta talet av talen $a_1 \cdots a_4$. Varje tal a_i "pekar" på ett av talen b_j. Storleken på a_i bestämmer positionen hos det motsvarande talet b_j. Storleken på talet b_j bestäms å andra sidan av positionen hos a_i. Konsekvensen av detta är att b_2 måste vara det minsta av talen $b_1 \cdots b_4$, eftersom a_1 står längst till vänster.

Går vi vidare med a_2 så ser vi att detta ger att b_1 blir det nästa minsta av talen $b_1 \cdots b_4$. På samma sätt fås att b_4 blir näst störst, och b_3 störst. Alltså motsvarar även $b_1b_2b_3b_4$ mönstret 2143.

Om mönstret inte förekommer på den andra halvan, kan det därför inte heller förekomma på den första, och därmed inte alls. Antalet involutioner utan fixpunkter av längd $2n$ som undviker mönstret $--$ (2143) är alltså lika med antalet permutationer av längd n som undviker detta mönster, vilket per definition är V_n.

Förmodan 2. $|P_{2n}(\sigma)| = F_n \ (n > 0) \quad \forall \ \sigma \in \{\cdot -, - \cdot, \frown\}$ *där* P_{2n} *är mängden av alla permutation av längd* $2n$ *utan fixpunkter (på engelska så kallade '*derangements*'), och* F_n *Fine–talen, A000957.*

Det är känt att Fine–talen räknar antalet Dyckvägar som inte har någon topp[16] på höjden ett [14]. Vi vet vidare att antalet permutationer av längd n som undviker ett mönster av längd tre är C_n, det vill säga antalet Dyckvägar av längd $2n$.

Catalantalen kan definieras med hjälp av rekursionen

$$C_n = \sum C_k C_{n-k-1}. \tag{25}$$

En tolkning av ekvation (25) är att det objekt som ett Catalantal motsvarar går att dela in i två partitioner på n sätt där varje partition är ett objekt av samma typ.

Visualiseringen av de bilder som ger Finetalen, tillsammans med det faktum att Finetalen räknar Dyckvägar som inte har någon topp på höjden ett säger oss att det måste finnas en metod för att partitionera de permutationer som undviker dessa mönster på ett sådant sätt att partitioneringspunkter motsvarar att Dyckvägen når x–axeln, och fixpunkter motsvarar just toppar på höjden ett.

[16] En topp är ett U–steg följt av ett N–steg.

A Tabeller och Bilder

mönster	bild
123	
132	
213	
231	
312	
321	

Tabell 1: Översättningstabell från mönster i S_3 till bilder.

mönster	bild
1234	
1243	
1324	
1342	
1423	
1432	
2134	
2143	
2314	
2341	
2413	
2431	
3124	
3142	
3214	
3241	
3412	
3421	
4123	
4132	
4213	
4231	
4312	
4321	

Tabell 2: Översättningstabell från mönster i S_4 till bilder.

p	$\lvert U_n(p)\rvert$	$\lvert I_n(p)\rvert$	$\lvert S_n(p)\rvert$
$\cdots$	$\binom{n-1}{n/2}$ (n even)	$\binom{n}{n/2}$	C_n
$\cdot-$	C_k	$\binom{n}{n/2}$	C_n
$-\cdot$	C_k	$\binom{n}{n/2}$	C_n
$\smile$	2^k	2^n	C_n
$\frown$	2^k	2^n	C_n
$\overset{\cdot}{\frown}$	C_k	$\binom{n}{n/2}$	C_n

Tabell 3: De talföljder som genereras av mönster i S_3. De talföljder som indexeras med k är noll för udda n. För jämna n är $2k = n$. Förklaringar i tabell (9).

$\cdots$	A_k
$\cdot-,-\cdot,\overset{\cdot}{\frown}$	C_k
$\smile, \frown$	2^k

Tabell 4: Mönster i S_3 grupperade efter vilken talföljd de genererar på $\lvert U_{2k}(\sigma)\rvert$. Förklaringar i tabell (9).

$\cdots,\cdot-,-\cdot,\overset{\cdot}{\frown}$	A_n
$\smile, \frown$	2^n

Tabell 5: Mönster i S_3 grupperade efter vilken talföljd de genererar på $\lvert I_n(\sigma)\rvert$. Förklaringar i tabell (9).

$\cdots,\cdot-,-\cdot,\overset{\cdot}{\frown},\smile, \frown$	C_n

Tabell 6: Alla mönster i S_3 ger Catalantalen på $\lvert S_n(\sigma)\rvert$.

$\cdot-,-\cdot,\overset{\cdot}{\frown}$	F_n
$\cdots$	—
$\smile,\frown$	—

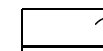
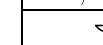
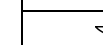
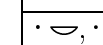
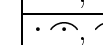
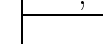
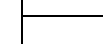
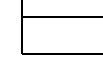
Tabell 7: Här har vi grupperat mönster i S_3 efter vilka talföljder de genererar på *permutationer* utan fixpunkter (på engelska: *derangements*), som vi kallar för P_n. Vi har med denna tabell för mönster i S_3 just på grund av att vi får Finetalen för mönstren $\cdot-$, $-\cdot$ och $\overset{\cdot}{\frown}$. Vi får inte träff i Sloane för något mönster i S_4 på P_n.

p	$\lvert U_n(p)\rvert$	$\lvert I_n(p)\rvert$	$\lvert S_n(p)\rvert$
	G_k	M_n	V_n
	—	M_n	V_n
	—	M_n	V_n
	D_k	—	W_n
	D_k	—	W_n
	—	M_n	V_n
	—	M_n	V_n
	V_k	M_n	V_n
	D_k	—	W_n
	R_k	—	V_n
	L_k	—	W_n
	—	—	W_n
	D_k	—	W_n
	L_k	—	W_n
	—	M_n	V_n
	—	—	W_n
	C_k	M_n	V_n
	B_n	—	V_n
	—	—	W_n
	—	—	W_n
	—	—	W_n
	—	—	W_n
	B_n	—	V_n
	C_k	M_n	V_n

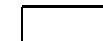
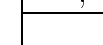
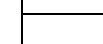
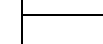
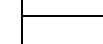
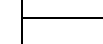
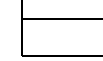
Tabell 8: De talföljder som genereras av mönster i S_4. De talföljder som indexeras med k är noll för udda n. För jämna n är $2k = n$. Förklaringar i tabell (9).

Beteckning	Namn	Nummer
A	centrala binomialtal	A001700
B	binomialsumma	A032443
C	Catalantal	A000108
D	centrala Delannoy	A001850
F	Finetal	A000957
G	generaliserade Catalantal	A006632
L	speciella Lyndonord	A054666
M	Motzkin	A001006
R	reguljärt uttryck	A052984
V	Vexillära	A005802
W	big Schröder	A022558

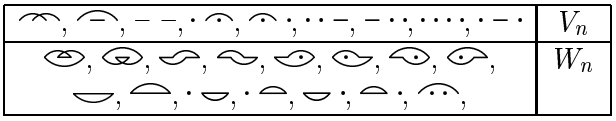
Tabell 9: Förklaring av beteckningar i tabeller.

	C_k
	V_k
	B_k
	—
	L_k
	—
	D_k
	—
	—
	—
	G_k

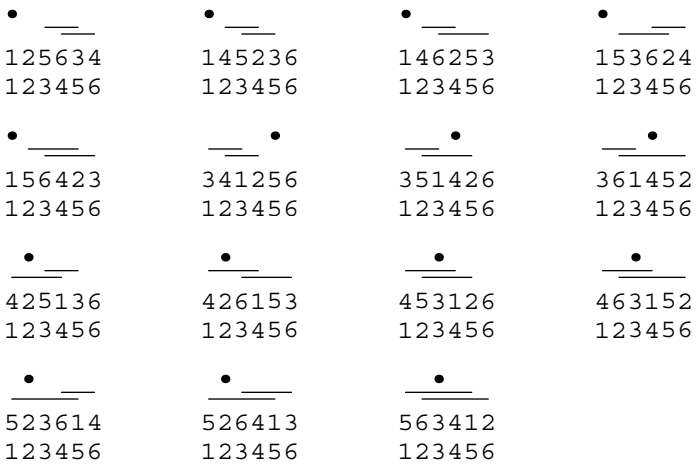
Tabell 10: Mönster i S_4 grupperade efter vilken talföljd de genererar på involutioner utan fixpunkter.

	—
	M_n
	—
	—
	—
	—
	—
	—

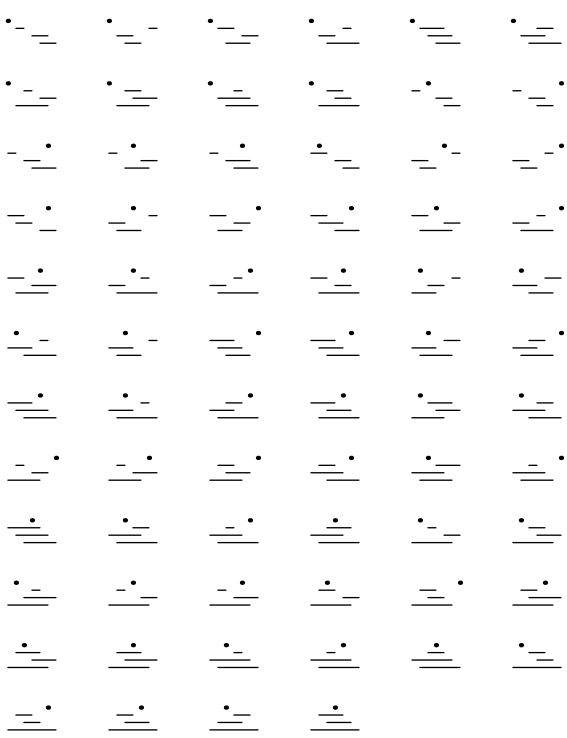
Tabell 11: Mönster i S_4 grupperade efter vilken talföljd de genererar på involutioner.



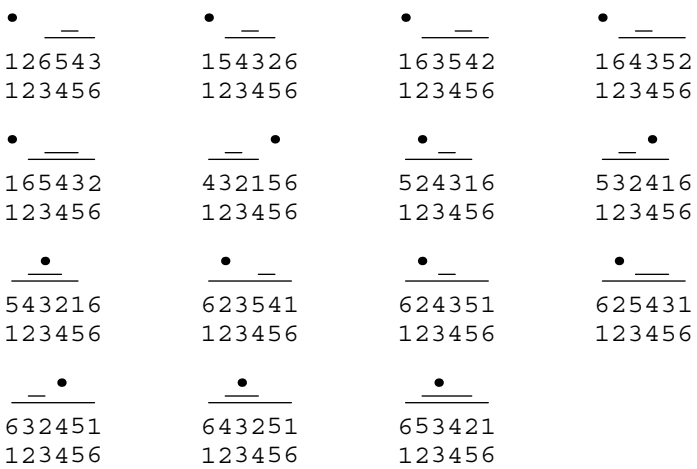
Tabell 12: Mönster i S_4 grupperade efter vilken talföljd de genererar på permutationer.



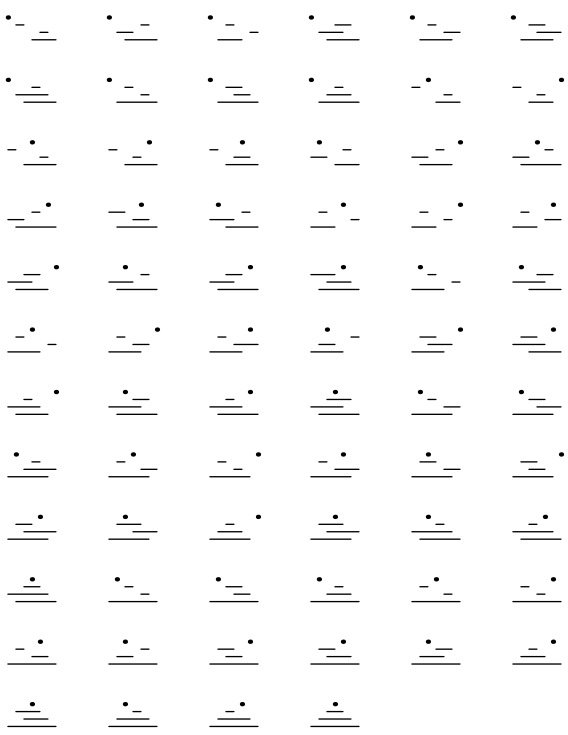
Figur 7: Alla involutioner av längd 6 med två fixpunkter som innehåller mönstret .



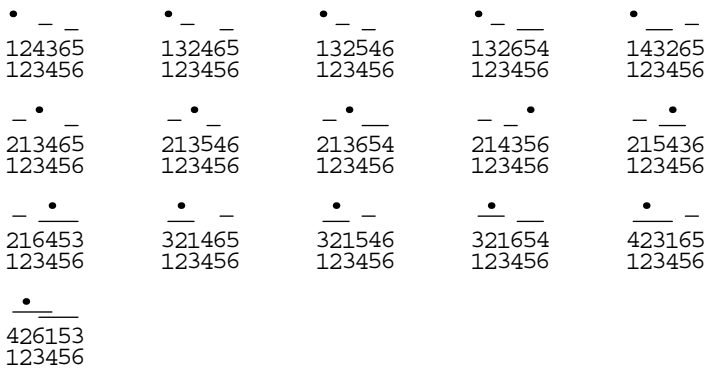
Figur 8: Alla involutioner av längd 7 med en fixpunkt som innehåller mönstret $\frown$.



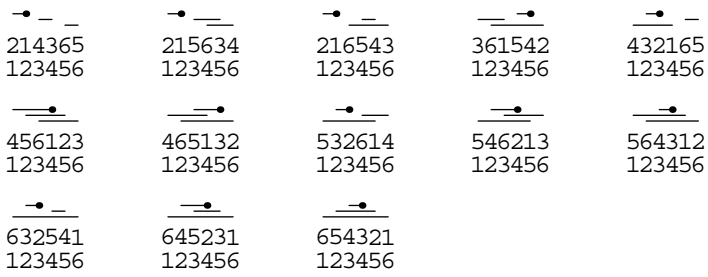
Figur 9: Alla involutioner av längd 6 med två fixpunkter som innehåller mönstret $\frown$.



Figur 10: Alla involutioner av längd 7 med en fixpunkt som innehåller mönstret $\frown\!\!\!\!\!-$.



Figur 11: Alla involutioner av längd 6 med två fixpunkter som innehåller mönstret $--$. Längst ner har vi en förekomst av mönstret, trots att de två cyklerna inte förhåller sig till varandra på ett förbjudet sätt.



Figur 12: De 13 elementen i $U_6(\smile \cdot)$.

Referenser

[1] N. L. Biggs, *Discrete Mathematics*, Oxford Science Publications, 1989. reprinted 1999.

[2] A. Claesson, *Generalised pattern avoidance*, European Journal of Combinatorics, (2001).

[3] R. Donaghey and L. W. Shapiro, *Motzkin numbers*, Journal of combinatorial theory, 23A (1977).

[4] M. Filaseta, *A new method for solving a class of ballot problems*, Journal of Combinatorial Theory, 39 (1985).

[5] I. M. Gessel, *Symmetric Functions and P-Recursiveness*, Journal of Combinatorial Theory, 53 (1990).

[6] I. M. Gessel and S. Ree, *Lattice Paths and Faber Polynomials*, Advances in combinatorial methods and applications to probability and statistics, (1996).

[7] P. Hilton and J. Pederson, *Catalan Numbers, Their Generalization, and Their Uses*, The Mathematical Intelligencer, 13(2) (1991), pp. 64–75.

[8] D. E. Knuth, *The art of computer programming (vol III)*, AddisonWesley, 1973.

[9] C. Krattenthaler, *An involution principle-free bijective proof of Stanley's hook-content formula*, Discrete Mathematics and Theoretical Computer Science, 3 (1998), pp. 11–32.

[10] I. G. Macdonald, *Symmetric Functions and Hall Polynomials*, Oxford University Press, 1995.

[11] P. A. MacMahon, *Combinatorial Analysis*, Cambridge University Press, 1915. reprented 1960.

[12] J.-C. Novelli, I. Pak, and A. V. Stoyanovskii, *A Direct Bijective Proof of the Hook-Length Formula*, Discrete Mathematics and Theoretical Computer Science, 1 (1997), pp. 53–67.

[13] J. J. O'Connor and E. F. Robertson, *Eugène charles catalan.*

[14] P. Peart and W.-J. Woan, *Dyck paths with no peaks at height k*, Journal of Integer sequences, 4 (2001).

[15] A. Regev, *Asymptotic Values for Degrees Associated with Strips of Young-Diagrams*, Advances in Mathematics, 41 (1981).

[16] N. J. A. Sloane, *The on-line encyclopedia of integer sequences*, 2001.

[17] R. P. Stanley, *Enumerative Combinatorics, vol 1*, Cambridge University Press, 1986. reprinted 1997.

[18] R. A. Sulanke, *Moments of Generalized Motzkin Paths*, Journal of Integer Sequences, 4 (2000).

[19] H. Wilf, *generatingfunctionology*, Academic Press, Inc., 1990. reprinted 1994.

OBJECT ORIENTED LINEAR ALGEBRA

A thesis submitted to the University of Manchester
for the degree of Master of Philosophy
in the Faculty of Science and Engineering

December 1999

By
Miguel Angel Luján Moreno (Mikel Luján)
Department of Computer Science

Contents

List of Tables

List of Figures

Abstract

The weak point of traditional Linear Algebra libraries is their intellectual distance from Linear Algebra. For one matrix calculation, such as multiplication of matrices, the library provides a large number of subroutines. Each of these subroutines is an optimal implementation for a specific situation (matrix properties and storage formats). Users are forced to analyse their problems in terms of the storage formats and matrix properties supported by the library in order to get good performance (or to use the library correctly).

At present, almost all Object Oriented Linear Algebra Libraries (OOLALs) offer a simpler interface. These OOLALs are equipped with a *rule based reasoning* system for certain matrix calculations. Thus, when a method is invoked the reasoning system decides which of the different implementations (with the same functionality) is appropriate for execution. The decision is based on those situations for which the library provides efficient subroutines. The matrix calculations, for which there is no reasoning system, are offered in different ways depending on the OOLAL.

An exception is the Matrix Template Library (MTL), which combines object oriented and generic programming to reduce the number of specialised implementations for each matrix calculations. It is based on the idea of *iterators*, which support transparent access to data structures without explicitly indications.

This thesis describes an object oriented analysis and design of linear algebra that establishes a context in which various OOLALs are evaluated. The Object Oriented Linear Algebra LibrAry (OoLaLa) is a new OOLAL which arises out of this analysis and design. OoLaLa specifies an interface suitable for both expert and non-expert users. This interface covers basic matrix operations (e.g. matrix addition), and the solution of matrix equations (e.g. system of linear equations, eigenproblem) with iterative and direct algorithms. None of the reviewed OOLALs addresses such a range of numerical linear algebra functionality.

In addition, OoLaLa's design enables libraries to change the storage format of a matrix in response to changes in its matrix properties. This is a novel functionality for linear algebra libraries. OoLaLa also illustrates how matrix calculations can be implemented at storage format (traditional libraries), at iterator level (MTL approach) and at matrix abstraction level (regardless of storage format, but explicitly indicating the position to be accessed) solely using object oriented programming.

Finally, linear algebra expressions are analysed. Some of these expressions are semantically equivalent but result in different programs delivering different execution times. These expressions constitute limitations that current linear algebra libraries cannot solve efficiently. Consequently, a Linear Algebra Problem Solving Environment is proposed in which compiler techniques and OoLaLa are integrated.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Copyright

To Agurtzane
and to my parents
Domingo and Juli

Acknowledgements

I would like to thank Professor John Gurd and Dr. Len Freeman for their support and guidance during this year. Dr. Len Freeman soon left its role of adviser to become a supervisor. This joint supervision has probed to be very helpful in this multidisciplinary thesis (mathematics and computer science). I am looking forward to continuing working towards the PhD with both of you.

During this year, I have enjoyed the company of the members of the Center for Novel Computing, specially in the tea breaks and in Rhodes. At different points, every one has provided his expertise and I want to thank you for this. In particular, Nicolas Fournier and Boby Cheng offered useful comments and discussions during the writing-up of this thesis.

This work has been supported by a research scholarship from the Department of Education, Universities and Research of the Basque Government.

Chapter 1

Introduction

1.1 Overview

Scientists and engineers describe physical phenomena in terms of mathematical models. These models are usually continuous and too complex to be solved analytically. In such cases, they are approximated with discrete mathematical models and solutions are obtained by applying numerical methods. A computer simulation of physical phenomena that follow a discrete mathematical model is called a scientific application. From a user point of view, a scientific application is a tool that enables scientists to experiment with physical phenomena and, in that way, increase their understanding. The advantage of scientific applications is that they have a limited cost and no risk. Real experiments consume products in each experiment and thus the cost is accumulative. In addition, real experiments, such as chemical reactions, can have high risk (e.g. explosions, environmental pollution) which do not exist in computer simulations. By contrast, a scientific application has a fixed cost, the software development cost, and enables scientists to experiment an unlimited number of times (assuming that the cost of running a scientific application on a computer is negligible compared with the development cost).

This thesis takes *numerical linear algebra* as an example family of scientific applications. Numerical linear algebra is a field lying between linear algebra and computer science, which generates computer programs to solve linear algebra problems.

This thesis analyses the software development process for sequential linear algebra applications in order to improve this process. The accepted development

process is based on using a library. This thesis follows the library approach, but instead of using traditional libraries, it focuses on object oriented libraries. The contributions can be summarised as follows:

1. a survey and classification of object oriented linear algebra libraries is presented;

2. a new design, for the Object Oriented Linear Algebra LibrAry (OoLaLa) is developed, which spans the functionality of both traditional and object oriented libraries; and finally

3. problems, or limitations, are identified which a library approach to the development of linear algebra programs cannot solve.

Metaphorically speaking, this thesis is an intellectual journey that begins with an unsatisfactory development process for linear algebra applications based on a library approach (Section 1.2). From this departure point, the journey builds on the valid library approach and mixes it with object oriented software construction techniques (Section 1.3). A new object oriented design increases the functionality of the existing object oriented linear algebra libraries. The journey arrives at the final station where a library approach can be dispensed with, and problems that any library cannot overcome are identified (Section 1.4). The journey returns to the original departure point, and concludes that a problem solving environment approach is needed to overcome these problems. In this problem solving environment the new object oriented design will be integrated with techniques developed in other areas of computer science.

1.2 Traditional Linear Algebra Libraries

Over the last 40 years the numerical linear algebra community has developed a large number of subroutines. These subroutines have been grouped into different libraries, each library targeting a set of linear algebra calculations. A major benefit of numerical libraries is that they are a means of reusing expert knowledge in the form of code. Ideally, a numerical linear algebra program would be the declaration of data structures used by the library and a succession of calls to library subroutines. However, sometimes users' requirements (e.g. multiplication of two banded matrices) go beyond the scope of traditional libraries; and the users then have to write code themselves.

A second benefit is portability. Libraries pass a standardisation process in which the functionality to be included, realised in the form of subroutine declarations and data structures (storage formats) in a specific programming language, is determined. The implementations are not standardised, although reference ones are distributed. This enables vendors to supply an implementation optimised to a specific architecture. In this way, not only is the library portable, since the programming language itself is portable, but also the performance can be ported from architecture to architecture.

The term *traditional libraries* is applied to the libraries developed by this research community using a top-down methodology and implemented in imperative languages. The predominant language in this field is Fortran 77 and examples of these libraries are LINPACK [DBMS79], EISPACK ([SBD+76],[GBDM77]) and more recently BLAS ([BLA99][1] [LHKK79], [DCHH88b], [DCHD90]) and LAPACK ([ABD+95]).

Given these traditional libraries, the development process of numerical linear algebra applications can be summarised as follows:

1. describe in terms of linear algebra calculations the problem to be solved;
2. select the numerical library (or libraries) which solves the problem;
3. translate the linear algebra problem so that it is defined in terms of the specific situations (storage formats and subroutines) supported by the library (or libraries).

The third step of this development process is non-trivial. A common characteristic of traditional libraries is that they provide many implementations for one mathematical operation. Knowing information about the matrices (matrix properties) involved in a matrix calculation has enabled the numerical linear algebra community to develop optimised implementations. This means that the number of combinations of different matrix properties supported by a library is the number of different implementations of each matrix calculation. Moreover, some traditional libraries provide the facility of storing matrices in different storage formats. Hence, the number of implementations of each matrix calculation is the number of combinations of the different matrix properties together with the possible storage formats supported by a library.

[1] Draft document under community revision that will substitute the other references of BLAS.

Certain matrix calculations can be implemented using different algorithms (not developed by exploiting matrix properties) and the numerical linear algebra community is not always able to identify the situations for which each algorithm is most appropriate. This is the case for iterative and direct algorithms applied to sparse systems of linear equations (see [BBC+94], [BDD+95], [DER86]).

Traditional libraries do not encapsulate or hide information; subroutine names and parameters reveal implementation details. Each subroutine name describes the basic type of the matrices, the properties of matrices, the storage format and the operation. The subroutine parameters are arrays that store matrices or vectors, integer values that declare the dimensions of matrices or vectors, and string values that declare more precisely the properties of matrices.

To sum up, the program development process requires:

- analysis of the properties of matrices,
- choice of the storage formats, and
- selection of the subroutines that will deliver the best performance.

To improve the process of developing linear algebra programs, the intellectual distance from a description of the problem in terms of linear algebra to a description in terms of traditional libraries must be reduced. Following the trend in other areas of computer science, object oriented linear algebra libraries (OOLALs) are a possible way of improving the software development process for linear algebra programs. OOLALs provide abstractions closer to linear algebra and, therefore, a reduced intellectual jump.

1.3 Object Oriented Linear Algebra Libraries

In contrast with traditional libraries, there is no consensus in the community about OOLALs, possibly due to their immature state. The first paper about object oriented linear algebra [McD89] only appeared in 1989 and the first international conference dedicated to object oriented numerical applications [Rog93] was not until 1993. Several OOLALs with different designs have been developed encapsulating matrices and vectors in classes. They differ in the sets of matrix properties for which they implement optimised versions and the storage formats for each matrix property. When there is only one storage format provided, “by

pure luck" users are relieved of managing the store format, but as result there is a loss of flexibility that might result in an excessive memory requirement. When there are many storage formats, users have to explicitly select the storage format.

The visible benefit for the user of OOLALs is a simpler interface than the interface of traditional libraries. The OOLALs provide for one matrix calculation one visible method. The different implementations are hidden behind the visible method. Each of these visible methods incorporates a set of rules that are able to decide the appropriated implementation. Obviously, in the cases where the numerical linear algebra community has not been able to identify which implementation is appropriate, the OOLALs have to give access to the different implementations.

The hidden implementations of matrix calculations access the representation of storage formats, as traditional libraries. This level of abstraction is referred to in this thesis as the *storage format abstraction level*.

A significantly different level of abstraction, called *iterator abstraction level* in this thesis, is used to implement the mathematical operations in the Matrix Template Library (MTL) ([SL98a], [SL98b], [SL98c], [SL99] [SLL99]). MTL combines object oriented and generic programming to reduce the number of implementations. The key for this change comes from the concept of an *iterator*. An iterator is a generic abstraction layer that provides a set of methods to traverse data structures. Each data structure implements the traversal methods in a different way, nevertheless these methods provide the same functionality. When applying iterators to linear algebra, the data structures are matrix properties with storage formats. The classes of MTL implement the iterator methods taking advantage of a given matrix property. The implementation of a matrix calculation changes from being written in terms of loop bounds to being an implementation written in terms of iterators.

Alternatively, a storage format can be considered as a mapping of element positions to memory positions. Given that every class representing a matrix implements (differently) the same methods to access (read and write) the matrix, an implementation of a matrix calculation can use these access methods and be independent of the storage formats. This level of abstraction is referred to in this thesis as the *matrix abstraction level*.

This thesis proposes a new design basis for the Object Oriented Linear Algebra LibrAry (OoLaLa). The novel characteristics of the design are the management

of storage formats and the propagation of matrix properties through matrix calculations.

The library is only active when one of its subroutines (or methods) is called (or invoked). OoLaLa checks the consistency between the storage formats and the matrix properties, which are the parameters of the method invoked. If necessary, OoLaLa will change the storage formats and properties to re-establish the consistency. It might not be obvious, but when OoLaLa checks the consistency of parameters, these include both input and output parameters. Therefore OoLaLa is able to propagate matrix properties to the output parameters from the input parameters. The idea of propagation of properties in not new ([Bik96], [Mar97]), but it is a novel functionality for a linear algebra library.

1.4 Limitations of a Library Approach

At this point, it is convenient to reconsider the intellectual distance between linear algebra and OoLaLa. The distance has been reduced, but the following tasks still remain:

1. analysis of the mathematical properties of the matrices that are the inputs of a linear algebra problem,

2. parsing of linear algebra expressions to the language defined by the visible methods of OoLaLa, and

3. selection of the appropriate method.

Bik and Wijshoff ([Bik96], [BW99]) have developed efficient algorithms to automatically analyse certain matrix properties. This analysis, when included in OoLaLa, could simplify the first task.

The limitations of the library approach are a consequence of its passive role. A library is only active when a subroutine (or method) is called (or invoked). At that moment, a library is not able to look ahead to subsequent computations, and therefore the library can only offer a correct solution at that point of the program.

The second remaining task can be seen as a compilation problem. The source language is defined by expressions accepted in linear algebra and the target language is the one defined by the visible methods of OoLaLa. The parser techniques need to have access to the whole program in order to generate efficient

code, but access to the whole program is incompatible with a library approach.

The third task remains an open problem. Rice and Boisvert [Ric96], among other ideas, propose expert systems or knowledge-based systems as a possible solution for this kind of problem [RB96]. They also remark that "the current state-of-the-art of knowledge-based frameworks is low-level and far from adequate for building Problem Solving Environments". A *problem solving environment* is a software system that integrates any computer science discipline in order to enable users to develop programs using the notation or language of their specific problem domain [GHR94]. The different tasks described for developing a linear algebra program constitute the description of a linear algebra problem solving environment.

1.5 Thesis Outline

The remainder of the thesis is organised as follows:

Chapter 2 introduces concepts of linear algebra and numerical linear algebra, and describes the BLAS and LAPACK designs. It is shown that the top-down design results in a complex interface. Matlab and a Sparse Compiler are introduced as alternative approaches.

Chapter 3 reviews object oriented software construction, and describes an object oriented analysis and design of linear algebra. This design is the basis of OoLaLa. Various object oriented models are proposed and used to classify several OOLALs. The design is balanced between the requirements of expert and non-experts users, and enables OoLaLa to manage the storage formats and to propagate matrix properties through matrix calculations; a novel functionality for a library. Iterator and matrix implementation abstraction levels are described as a way of reducing the number of implementations of matrix calculations.

Chapter 4 provides a high level description of the implementation issues of OoLaLa. The design of OoLaLa is adapted to the restrictions of the programming language Java. This chapter compares matrix calculations implemented at storage format, at iterator level and at matrix abstraction level.

Chapter 5 identifies limitations of a library approach in the context of linear algebra. Some of these limitations are due to in the difficulty for users to parse a linear algebra expression to an optimum set of calls to library subroutines.

Chapter 6 reviews the contributions of this thesis to the software development process of sequential linear algebra programs, and proposes future research directions.

Chapter 2

Numerical Linear Algebra

Since the mid 1950s, the numerical linear algebra community has been investigating the problem of how to write programs for matrix calculations so that the solutions are accurate and the execution times minimised. In this still open research area, numerical analysis and linear algebra are combined. The importance of numerical linear algebra resides in its applicability to important problems such as computational fluid dynamics, circuit simulations, data fitting, graph theory, etc. [AR94].

During the ensuing 40 years, important knowledge has been created in the form of algorithms and these have been made reusable as software libraries. To understand what functionality is provided, and why, as well as how the libraries are organised is the main objective of this chapter. The other important aspect is to analyse the influence on the user of the organisation and functionality of these libraries. Since the next chapter includes an object oriented analysis and design of linear algebra, this chapter can also be interpreted as a "requirements document" that summarises the domain.

The process of understanding begins with a review of the basic concepts of matrices and matrix calculations (Section 2.1). Then matrices are classified according to two criteria and the way a given matrix can be represented in different storage formats is examined (Section 2.2). The defined categories, or matrix properties, allow the creation of specialised algorithms which take advantage of certain specific matrix properties (Section 2.3). The algorithms and storage formats are combined to implement matrix calculations. Storage format abstraction level is the term used in this thesis to describe how libraries are traditionally implemented. This aspect is criticised in the next chapter by introducing another

two abstraction levels. Given this knowledge, the final step is to examine how BLAS and LAPACK are organised; two examples of libraries developed by community consensus (Section 2.4). These libraries are compared with two software environments; Matlab and the Sparse Compiler. Matlab and the Sparse Compiler represent alternatives to the libraries approach of linear algebra program construction, and permit examination of the difficulties, or steps to follow, in developing linear algebra programs.

2.1 Basic Background

Numerical linear algebra is primarily concerned with matrix calculations. These calculations can be subdivided into two groups. The first group consists of basic matrix operations (e.g. transpose, addition, ...), and the second group involves more complex matrix calculations. Systems of linear equations, eigenvalue and eigenvector problems, and least squares problems are the matrix calculations of this second group. It is out of the scope of this thesis to introduce and describe all the work and state-of-the-art of this research area. Nevertheless, it is the aim of this section to familiarise the reader with the necessary notation and definitions.

2.1.1 Matrix

A matrix is defined as a rectangular array of numbers.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix}$$

The size of a matrix is described in terms of the number of rows m and the number of columns n. When $m = n$, the matrix is called a *square matrix of order* n. When $m = 1$ or $n = 1$, the matrix is called a *row vector* or a *column vector*, respectively. The general case is called a *rectangular matrix of dimension* $m \times n$ (an $m \times n$ matrix). The numbers a_{ij} that constitute the matrix are called its *elements*.

Note that this is a mathematical definition and, therefore, "array" must not be taken in its computer science sense. For computer scientists, a suggested alternative is to substitute *rectangular array* with *two-dimensional container*.

The notation (followed throughout the thesis) is

- matrices are represented by upper case letters (A, B, C, ..., Z),
- column vectors are represented by lower case letters (a, b, ..., z),
- scalars are represented by lower case Greek letters (α, β, ..., ω).

The same letter that is used to represent the matrix, but in lower case and with two suffices represents the elements of a matrix. For example, a_{ij} represents the element which is situated in the i^{th} row and the j^{th} column of matrix A. The elements of a vector are represented with the same letter that is used to represent the vector with one suffix (e.g., x_i represents the i^{th} element of the x vector).

2.1.2 Matrix Calculations

Basic Matrix Operations

The basic matrix operations can be divided into two groups. There are operations that need only one matrix – (monadic) unary, while the others need two matrices – (dyadic) binary. This division is important when implementing the operations. Some definitions of basic matrix operations are presented in Table 2.1.

System of Linear Equations

A system of linear equations is a finite set of linear equations in the variables x_1, x_2, ..., x_n and can be expressed as:

$$\begin{array}{ccccccccc}
a_{11}x_1 + a_{12}x_2 + & \dots & +a_{1j}x_j + & \dots & +a_{1n}x_n & = & b_1 \\
a_{21}x_1 + a_{22}x_2 + & \dots & +a_{2j}x_j + & \dots & +a_{2n}x_n & = & b_2 \\
\vdots & \ddots & \vdots & \ddots & \vdots & = & \vdots \\
a_{i1}x_1 + a_{i2}x_2 + & \dots & +a_{ij}x_j + & \dots & +a_{in}x_n & = & b_i \\
\vdots & \ddots & \vdots & \ddots & \vdots & = & \vdots \\
a_{m1}x_1 + a_{m2}x_2 + & \dots & +a_{mj}x_j + & \dots & +a_{mn}x_n & = & b_m
\end{array} ,$$

Name	Notation	Definition
Vector Norms	$\|\|x\|\|_p$	$\alpha \leftarrow (\sum_i \|x_i\|)^{1/p}$
	$\|\|x\|\|_\infty$	$\alpha \leftarrow \max_i \|x_i\|$
Matrix Norms	$\|\|A\|\|_1$	$\alpha \leftarrow \max_j \sum_i \|a_{ij}\|$
	$\|\|A\|\|_\infty$	$\alpha \leftarrow \max_i \sum_j \|a_{ij}\|$
	$\|\|A\|\|_F$	$\alpha \leftarrow (\sum_{i,j} \|a_{ij}\|^2)^{1/2}$
Vector Transpose	x^T	
Matrix Transpose	A^T	
Matrix Inverse	A^{-1}	
Dot Product	$\alpha \leftarrow x^T y$	$\alpha \leftarrow \sum_i x_i y_i$
Vector Scale	$y \leftarrow \alpha x$	$y_i \leftarrow \alpha x_i$
Vector Addition	$z \leftarrow x + y$	$z_i \leftarrow x_i + y_i$
Matrix Vector Multiplication	$y \leftarrow Ax$	$y_i \leftarrow \sum_j a_{ij} x_j$
Matrix Scale	$C \leftarrow \alpha A$	$c_{ij} \leftarrow \alpha a_{ij}$
Matrix Addition	$C \leftarrow A + B$	$c_{ij} \leftarrow a_{ij} + b_{ij}$
Matrix Matrix Multiplication	$C \leftarrow AB$	$c_{ij} \leftarrow \sum_k a_{ik} b_{kj}$

Table 2.1: Definition of some basic matrix operations.

where a_{11}, a_{12}, ..., a_{mn}, b_1, b_2, ...,b_m are given constant numbers. The unknowns x_1, x_2, ..., x_n, occur linearly and do not appear as arguments for trigonometric, logarithmic or exponential functions.

The system of linear equations can be written more concisely in terms of the matrix A and the vectors x and b, as follows:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{pmatrix} \qquad Ax = b$$

Eigenvalues and Eigenvectors

Given an $n \times n$ matrix A, a vector x is called an *eigenvector* of A if Ax is a multiple of x and x has at least one nonzero element, i.e.

$$Ax = \lambda x$$

for some scalar λ. The scalar λ is an *eigenvalue* of A, and x is said to be the eigenvector of A corresponding to λ.

Least Square Problem

Given a linear system $Ax = b$ of m equations in n variables $n \leq m$, find a vector x that minimises

$$||Ax - b||_2.$$

2.2 Matrix Properties and their Storage Formats

2.2.1 Matrix Properties

Different criteria can be used to classify matrices. These criteria have been proposed because of the execution time benefit that results and because of their occurrence in real and important applications. Knowing the classification of a matrix, the implementation of a matrix calculation might take advantage and thereby reduce the execution time of the calculation. A second benefit might be a reduction in memory requirements for the computation. A third benefit might be that the accuracy of the results can be increased.

Two different criteria, and thereby two different classifications, are presented: nonzero elements structure and mathematical relations. The zero elements of matrices act in a particular way when added or multiplied ($a_{ij} + 0 = a_{ij}$ and $a_{ij} \times 0 = 0$). These properties enable implementations to avoid computations for which the result is already known.

The mathematical relations are relations independent of the zero elements and are expressed as operations of the matrix elements. For example, a matrix is symmetric if and only if $A = A^T$.

In general, the categories defined by these two criteria are not mutually exclusive, so that a matrix can have more than one category. For example, a matrix can be symmetric, positive definite and banded. The remainder of this section is dedicated to the definition of the categories.

From here on, the term *matrix properties* is used to refer to any category of mathematical relation or nonzero elements structure or combinations of these.

Nonzero Elements Structure Criteria

The nonzero elements structure criteria classifies matrices into *dense*, *banded*, *block* and *sparse*. Dense matrices are those matrices which have a majority of nonzero elements. At the other end of the spectrum, sparse matrices are those matrices which have a minority of nonzero elements (see Table 2.2 for dense and sparse matrix examples). A special sparse matrix is the zero matrix, $O_{m\times n}$, which has only zero elements. In the middle of the spectrum, banded and block matrices are matrices in which the nonzero elements have some structure. Both, banded and block matrices, have subcategories. Figure 2.1 presents an hierarchical view of different matrix properties derived from the nonzero elements structures.

Dense	Sparse
$\begin{pmatrix} \square & \square & \square & \square & \square & \square \\ \square & \square & & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \end{pmatrix}$ 6×6	$\begin{pmatrix} & & & \square & & \square \\ \square & \square & & & & \\ \square & & & \square & \square & \\ \square & & & \square & \square & \\ & & & & & \square \\ \square & & & & \square & \end{pmatrix}$ 6×6
$\begin{pmatrix} \square & \square & \square & \square & \square & \square \\ \square & \square & & \square & \square & \square \\ \square & \square & \square & \square & & \square \end{pmatrix}$ 3×6	$\begin{pmatrix} \square & & & \square & & \\ & & \square & \square & & \\ \square & & & & & \end{pmatrix}$ 3×6

Table 2.2: Examples of dense and sparse matrices – □'s represent nonzero elements and blanks represent 0.

A banded matrix is a matrix which has the nonzero elements grouped around the main diagonal. Formally, a $m \times n$ matrix A is banded if a lower bandwidth $b_l < m$ and upper bandwidth $b_u < n$ can be defined so that $a_{ij} \neq 0$ implies that $-b_u \leq i - j \leq b_l$. Different combinations of values for b_u and b_l yield different subcategories of banded matrices. For example, when $b_u = b_l = 0$ the matrix is *diagonal*. A special case of a diagonal matrix is the *identity matrix*, I_n, in which all the nonzero elements are 1. Table 2.3 presents graphical examples of banded matrices and some associated subcategories.

A matrix can be partitioned into sub-matrices A_{ij}. Since it is a partition, every element of A is in exactly one sub-matrix. Two sub-matrices which are in

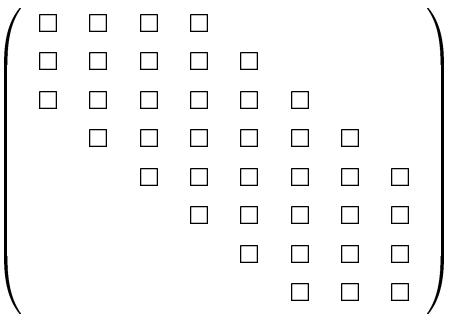
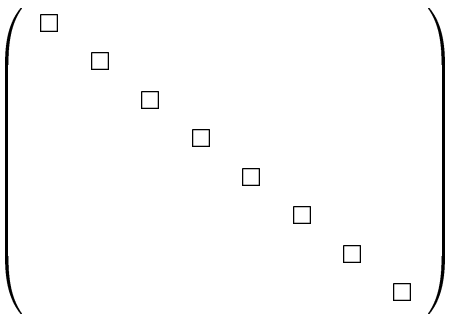
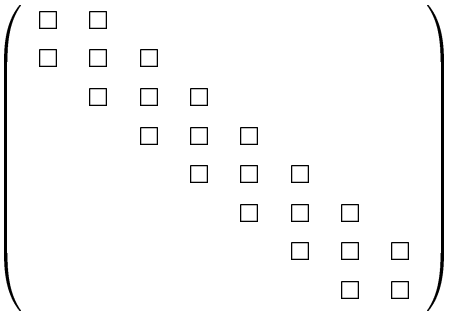
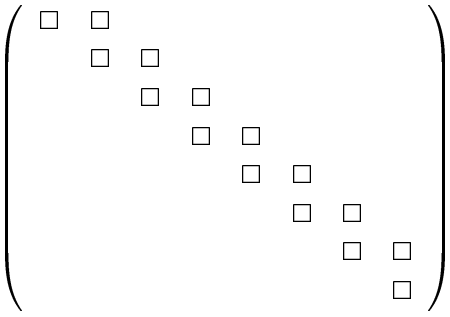
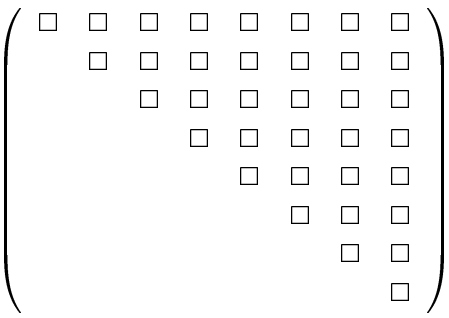
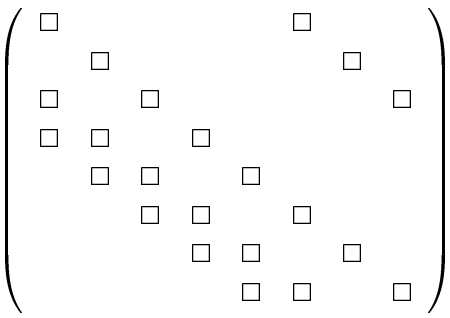
Matrix 8×8	Matrix 8×8
 banded $b_u = 3$, $b_l = 2$	 diagonal $b_u = 0$, $b_l = 0$
 tridiagonal $b_u = 1$, $b_l = 1$	 upper bidiagonal $b_u = 1$, $b_l = 0$
 upper triangular $b_u = 7$, $b_l = 0$	 multi-diagonal $b_u = 5$, $b_l = 3$

Table 2.3: Examples of banded matrices – □'s represent nonzero elements and blanks represent 0.

the same row (A_{ij} and $A_{i(j+1)}$) have the same number of rows. Two sub-matrices which are in the same column (A_{ij} and $A_{(i+1)j}$) have the same number of columns. Each sub-matrix can be classified as a zero matrix or a sparse matrix or a dense matrix or a banded matrix (and its subcategories).

$$A = \begin{pmatrix} A_{11} & \dots & A_{1q} \\ \vdots & \ddots & \vdots \\ A_{p1} & \dots & A_{pq} \end{pmatrix}$$

$$\left(\begin{array}{ccc|cccc|cc|c} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ a_{22} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ \hline a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \end{array}\right)$$

Having classified the sub-matrices for a given partition, a *block banded* matrix is defined as a *partitioned*, or *block*, matrix that has the nonzero sub-matrices grouped around the diagonal block (i.e. set of sub-matrices A_{ii}). Formally, a matrix A of dimension $m \times n$ and its partition in sub-matrices A_{11}, A_{12}, ..., A_{pq} are block banded if a lower bandwidth $B_l < p$ and upper bandwidth $B_u < q$ can be defined so that $A_{ij} \neq 0$ implies that $-B_u \leq i - j \leq B_l$. Different combinations of values for B_u and B_l yield different subcategories of block banded matrices. For example, when $B_u = B_l = 0$ the matrix is called *block diagonal*. Table 2.4 presents examples of block banded matrices and associated subcategories.

Comparing the subcategories of banded matrices with block banded matrices, a new subcategory is found, *bordered block banded* matrices (see Figure 2.1 and Table 2.4). Given a partition $A_{11}, A_{12}, \dots, A_{pp}$ of a matrix A, the set of sub-matrices A_{ip} are called the upper border sub-matrix and the set A_{pi} of sub-matrices are called the lower border sub-matrix. A bordered block banded matrix is a matrix whose off-border sub-matrices (i.e. A_{ij} with $i \neq p$ and $j \neq p$) form a block banded matrix, and the upper and the lower border sub-matrices are nonzero matrices.

Efficient algorithms for automatic detection of nonzero elements structures have been proposed by Bik and Wijshoff [BW99]. Other algorithms for reordering matrices (i.e. interchange columns or rows of a matrix) in order to create matrices that fall into some category are described in [DER86].

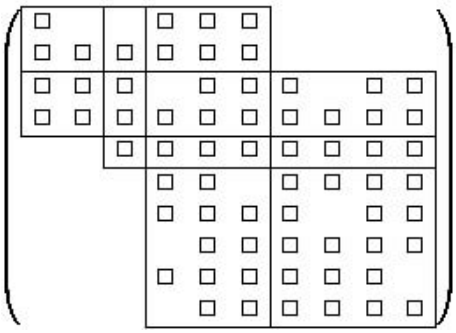
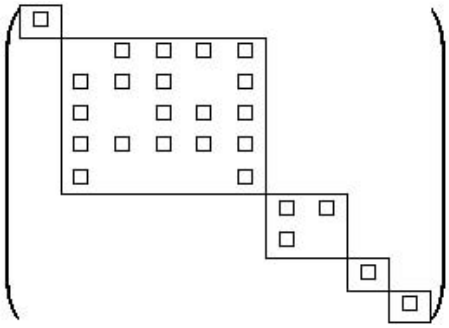
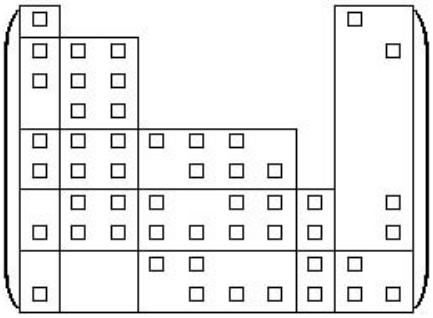
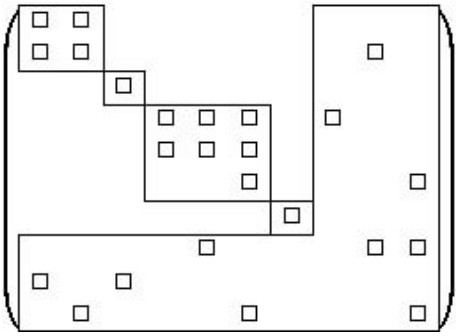
Matrix 10×10	Matrix 10×10
	
block banded $B_u = 2$, $B_l = 1$	block diagonal $B_u = 0$, $B_l = 0$
	
single bordered block lower triangular $B_u = 0$ $B_l = 4$	doubly bordered block diagonal $B_u = 0$ $B_l = 0$

Table 2.4: Examples of block matrices – □'s represent nonzero elements and blanks represent 0.

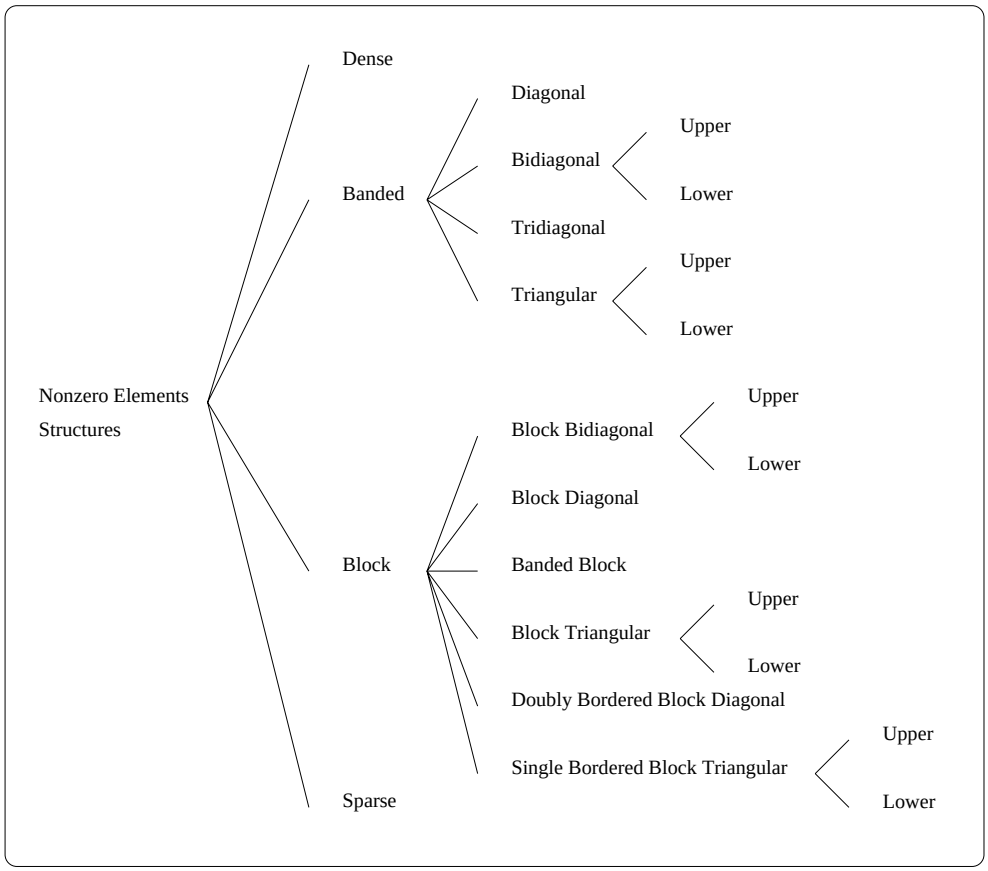


Figure 2.1: Hierarchical view of nonzero elements structures.

Mathematical Relation Criteria

The mathematical criteria, in contrast with nonzero elements structure, are not structural criteria. Loosely speaking, this means that the mathematical classification cannot be found simply by looking at the elements of the matrix. In order to verify if a matrix falls into a certain category, matrix calculations may be required. First, restrict consideration to square matrices; the following categories are used:

- *symmetric* – the matrix is equal to its transpose $A = A^T$,
- *orthogonal* – the inverse of the matrix is equal to its transpose $A^{-1} = A^T$ and therefore $AA^T = I$,
- *positive definite* – for all nonzero vectors x, $x^T Ax$ is positive, and
- *indefinite* - for some nonzero vectors x, $x^T Ax$ is positive, while for other nonzero vectors x it is negative or zero.

2.2.2 Storage Formats

Thus far, the matrices have not been represented by data structures; only mathematical notation has been used. The remainder of the section is dedicated to describing the most common data structures. The importance of this section is not simply to understand different storage formats (i.e. data structures to store matrices), but also to appreciate that a certain matrix with certain properties can be represented in a number of different storage formats.

At present, programming languages provide static and dynamic data structures. Static data structures have a predefined (compilation time) size and cannot be modified at run-time (e.g. arrays). On the other hand, a dynamic data structure can increase or decrease its size at run-time (e.g. lists, trees). Since Fortran 77 has been the dominant language for mathematicians and does not support dynamic data structures, the most commonly used storage formats are array-based. Dense, band and packed formats are presented in this section. Other storage formats for matrices can be found in [BBC+94] Section 4.3 and [DER86] Chapter 2.

Note that different memory layouts to store an array have been defined and are used. For example, a two-dimensional array in C is stored by rows, whereas

in Fortran it is stored by columns (see Figure 2.2). The storage formats presented in this section are organised by columns.

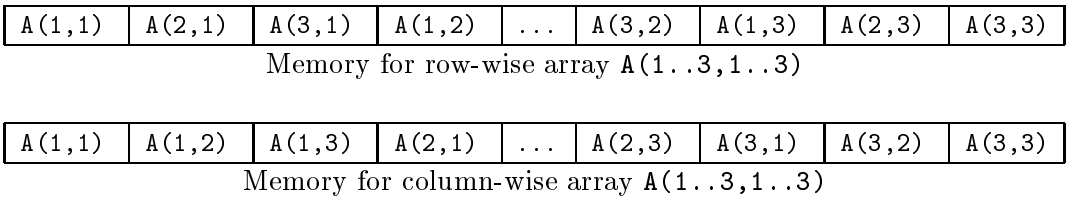


Figure 2.2: Row versus column-wise memory layout for arrays.

Dense Format

The most intuitive data structure to represent a matrix is a two-dimensional array. This is called *dense format*, or conventional format. The element a_{ij} of the matrix A is stored in `A(i,j)`. Figure 2.3 presents how different matrix properties can be stored in dense format. In fact, every matrix can be stored using this format.

Band Format

The *band format* uses a two-dimensional array to store the elements of a $n \times n$ banded matrix A. Given b_u and b_l as the upper and lower bandwidth of the matrix, the array `BAND` has $b_u + b_l + 1$ rows and n columns. The element a_{ij} is stored in `BAND(`$b_u + 1 + i - j$`,`j`)` if $-b_u \leq i - j \leq b_l$. Figure 2.4 presents examples of banded matrices represented in band format. Note that the first matrix is upper triangular and its array has the same size as its array when stored in dense format (see Figure 2.3). The drawback is that the cost for accessing an element is bigger (i.e. more operations need to be done in order to calculate the memory address). Band format reduces memory requirements when b_u and b_l are less than the matrix dimensions. Dense and triangular matrices should not use this format.

Packed Format

The *packed format* uses a one-dimensional array to store symmetric and triangular matrices. Given an $n \times n$ upper triangular matrix A, the array `PACK` is of size

Matrix	Data Structure
$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix}$	$\begin{array}{\|c\|c\|c\|c\|c\|} \hline a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \hline a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ \hline a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ \hline a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\ \hline \end{array}$
$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ & a_{22} & a_{23} & a_{24} & a_{25} \\ & & a_{33} & a_{34} & a_{35} \\ & & & a_{44} & a_{45} \\ & & & & a_{55} \end{pmatrix}$	$\begin{array}{\|c\|c\|c\|c\|c\|} \hline a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \hline & a_{22} & a_{23} & a_{24} & a_{25} \\ \hline & & a_{33} & a_{34} & a_{35} \\ \hline & & & a_{44} & a_{45} \\ \hline & & & & a_{55} \\ \hline \end{array}$
$\begin{pmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & a_{23} & & \\ & a_{32} & a_{33} & a_{34} & \\ & & a_{43} & a_{44} & a_{45} \\ & & & a_{54} & a_{55} \end{pmatrix}$	$\begin{array}{\|c\|c\|c\|c\|c\|} \hline a_{11} & a_{12} & & & \\ \hline a_{21} & a_{22} & a_{23} & & \\ \hline & a_{32} & a_{33} & a_{34} & \\ \hline & & a_{43} & a_{44} & a_{45} \\ \hline & & & a_{54} & a_{55} \\ \hline \end{array}$

Figure 2.3: Examples of matrices stored in dense format.

Matrix	Data Structure
$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ & a_{22} & a_{23} & a_{24} & a_{25} \\ & & a_{33} & a_{34} & a_{35} \\ & & & a_{44} & a_{45} \\ & & & & a_{55} \end{pmatrix}$ $b_u = 4,\ b_l = 0$	$\begin{array}{\|c\|c\|c\|c\|c\|} \hline & & & & a_{15} \\ \hline & & & a_{14} & a_{25} \\ \hline & & a_{13} & a_{24} & a_{35} \\ \hline & a_{12} & a_{23} & a_{34} & a_{45} \\ \hline a_{11} & a_{22} & a_{33} & a_{44} & a_{55} \\ \hline \end{array}$ `BAND(1..4+0+1,1..n)`
$\begin{pmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & a_{23} & & \\ & a_{32} & a_{33} & a_{34} & \\ & & a_{43} & a_{44} & a_{45} \\ & & & a_{54} & a_{55} \end{pmatrix}$ $b_u = 1,\ b_l = 1$	$\begin{array}{\|c\|c\|c\|c\|c\|} \hline & a_{12} & a_{23} & a_{34} & a_{45} \\ \hline a_{11} & a_{22} & a_{33} & a_{44} & a_{55} \\ \hline a_{21} & a_{32} & a_{43} & a_{54} & \\ \hline \end{array}$ `BAND(1..1+1+1,1..n)`

Figure 2.4: Examples of matrices stored in band format.

$\frac{1}{2}(n^2+n)$ and element a_{ij} is stored in `PACK`$(i+\frac{1}{2}j(j-1))$ when $i \leq j$; *upper packed format*. In the case where the matrix A is lower triangular, the array size is the same but element a_{ij} is stored in `PACK`$(i+\frac{1}{2}(2n-j)(j-1))$ when $j \leq i$; *lower packed format*. In both cases the zero elements are not stored. A symmetric matrix has the possibility to choose if the upper triangular or the lower triangular elements are stored. Figure 2.5 presents examples of matrices in this format.

Matrix	Data Structure
$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ & a_{22} & a_{23} \\ & & a_{33} \end{pmatrix}$	$\boxed{a_{11}}\boxed{a_{12}}\boxed{a_{22}}\boxed{a_{13}}\boxed{a_{23}}\boxed{a_{33}}$
$\begin{pmatrix} a_{11} & & \\ a_{21} & a_{22} & \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$	$\boxed{a_{11}}\boxed{a_{21}}\boxed{a_{31}}\boxed{a_{22}}\boxed{a_{32}}\boxed{a_{33}}$

Figure 2.5: Examples of matrices stored in packed format.

The final remark concerning matrices and storage formats comes in the form of an example. Given a matrix A which is symmetric banded, the matrix can be stored in 4 different ways. First, every matrix A can be stored in dense format. Second, a banded matrix A can be stored in band format. Third and fourth, as a symmetric matrix, A can be stored in packed format, either storing the upper triangular elements or the lower triangular elements.

2.3 Exploiting Matrix Properties

Two matrix calculations are used to illustrate their implementation in traditional libraries. The first calculation, matrix-matrix multiplication, is a basic binary matrix operation. However, this operation is enough to show that for one matrix operation many algorithms can be derived. Each algorithm is specialised for certain matrix properties, taking advantage of knowledge implied by the properties.

The second example is the solution of a system of linear equations. Two families of methods can be applied to solve systems of linear equations: *direct methods* and *iterative methods*. A direct method is an algorithm that calculates

```
for i = 1 to m
 for j = 1 to n
  for k = 1 to p
   c_ij ← c_ij + a_ik b_kj
  end for
 end for
end for
```

Figure 2.6: Algorithm for matrix-matrix multiplication $C \leftarrow AB$ with A and B dense matrices.

the solution in a known finite number of instructions. On the other hand, an iterative method is an algorithm that is executed repeatedly; each execution of the algorithm produces an approximate solution of the problem, and execution is stopped when the approximate solution is sufficiently accurate. The distinctive nature of the two families makes it clear that, in contrast with basic matrix operations algorithms, the different algorithms for systems of linear equations are not simple adaptations derived from the matrix properties.

The final subsection defines the storage format abstraction level; the abstraction level at which traditionally the matrix calculations are implemented. It is shown that, for each specialised algorithm when combined with storage formats for the matrix operands, different implementations are generated.

The terms *algorithm*, *storage format* and *implementation* are used in the computer science sense; i.e. that an implementation (program) is an algorithm plus storage format (data structure).

2.3.1 Matrix Matrix Multiplication

The product of a matrix A of dimension $m \times p$ with a matrix B of dimension $p \times n$ is another matrix C of dimension $m \times n$ with elements defined as

$$c_{ij} \leftarrow \sum_{k=1}^{p} a_{ik} b_{kj}.$$

When describing the algorithm, given by the above definition, three nested loops are necessary (see Figure 2.6). This algorithm assumes that both A and B are dense matrices.

The next algorithm is an example of matrix-matrix multiplication where one

```
for i = 1 to m
 for j = 1 to n
  for k = i to p
   c_ij ← c_ij + a_ik b_kj
  end for
 end for
end for
```

Figure 2.7: Algorithm for matrix-matrix multiplication $C \leftarrow AB$ with A upper triangular and B dense matrices.

```
for i = 1 to n
 for j = 1 to n
  for k = max(i, j) to n
   c_ij ← c_ij + a_ik b_kj
  end for
 end for
end for
```

Figure 2.8: Algorithm for matrix-matrix multiplication $C \leftarrow AB$ with A upper triangular and B lower triangular matrices.

of the matrices is not dense (see Figure 2.7). When A is upper triangular with dimension $m \times p$ and B is dense with dimensions $p \times n$ the algorithm can be modified (to shorten the k loop) so that the elements a_{ij} with $i \geq j$ are not used since they are known to be zero:

$$\left.\begin{array}{ll} a_{ik} \neq 0, & i \leq k \\ a_{ik} = 0, & i > k \end{array}\right\} \Rightarrow c_{ij} \leftarrow \sum_{k=i}^{p} a_{ik} b_{kj}.$$

Two more examples are given in which neither of the matrix operands is dense. For the first example, A is upper triangular and B is lower triangular, both of dimension $n \times n$. Having as a starting point the algorithm of Figure 2.7, the algorithm of Figure 2.8 is obtained. The k loop is further shortened exploiting the zeros in matrix B:

$$\left.\begin{array}{ll} b_{kj} \neq 0, & k \geq j \\ b_{kj} = 0, & k < j \end{array}\right\} \Rightarrow c_{ij} \leftarrow \sum_{k=max(i,j)}^{n} a_{ik} b_{kj}.$$

The final example multiplies two upper triangular matrices of dimension $n \times n$.

As with the previous example, the algorithm of Figure 2.7 is used as a starting point. Since B is upper triangular the elements b_{ij} with $i \leq j$ are zero:

$$\left.\begin{array}{ll} b_{kj} \neq 0, & k \leq j \\ b_{kj} = 0, & k > j \end{array}\right\} \Rightarrow c_{ij} \leftarrow \sum_{k=i}^{j} a_{ik} b_{kj}.$$

Note that for $i > j$ the elements c_{ij} are zero, i.e. C is also upper triangular. In this case it is possible to shorten the j loop (see Figure 2.9).

```
for i = 1 to n
 for j = i to n
  for k = i to j
   c_ij ← c_ij + a_ik b_kj
  end for
 end for
end for
```

Figure 2.9: Algorithm for matrix-matrix multiplication $C \leftarrow AB$ with A and B upper triangular matrices.

Generalising from these examples to all the basic matrix operations, it can be observed that for each basic matrix operation many algorithms can be derived. Each algorithm is derived by exploring the knowledge implied by the matrix properties. The number of algorithms that can be derived for a unary operation has a linear relation with the number of matrix properties. The number of algorithms that can be derived for a binary operation has a square relation with the number of matrix properties. Finally, as each specialised algorithm responds to certain matrix properties, a complete decision tree can be defined for each matrix operation. This takes as inputs the properties of the matrices and determines the specialised algorithm to be used and the properties of the solution matrix.

2.3.2 Systems of Linear Equations

Direct Methods

In the case of matrix-matrix multiplication the algorithms have been presented by refining the general algorithm for each special case. In the case of a system of linear equations $Ax = b$, the specialised algorithms are described first.

The first and simplest example is a diagonal matrix A. Remembering the

```
for i = 1 to n
  x_i ← b_i / a_ii
end for
```

Figure 2.10: Algorithm for a system of linear equations with A diagonal

definition of diagonal matrix, when $i \neq j$ the elements a_{ij} are zero. Therefore, the solution is obtained as follows:

$$\begin{pmatrix} a_{11} & & & & \\ & \ddots & & & \\ & & a_{ii} & & \\ & & & \ddots & \\ & & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{pmatrix} \Rightarrow \begin{cases} x_1 \leftarrow \frac{b_1}{a_{11}} \\ \vdots \\ x_i \leftarrow \frac{b_i}{a_{ii}} \\ \vdots \\ x_n \leftarrow \frac{b_n}{a_{nn}} \end{cases},$$

which is the basis of the algorithm of Figure 2.10.

In the second example, the $n \times n$ matrix A is lower triangular. This means that the elements a_{ij} with $i < j$ are zero. The solution is obtained as follows:

$$\begin{pmatrix} a_{11} & & & & & & \\ a_{21} & a_{22} & & & & & \\ a_{31} & a_{32} & a_{33} & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ a_{i1} & a_{i2} & a_{i3} & \dots & a_{ii} & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{ni} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{pmatrix} \Rightarrow \begin{cases} x_1 \leftarrow \frac{b_1}{a_{11}} \\ x_2 \leftarrow \frac{b_2 - a_{21}x_1}{a_{22}} \\ x_3 \leftarrow \frac{b_3 - (a_{31}x_1 + a_{32}x_2)}{a_{33}} \\ \vdots \\ x_i \leftarrow \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}} \\ \vdots \\ x_n \leftarrow \frac{b_n - \sum_{j=1}^{n-1} a_{nj}x_j}{a_{nn}} \end{cases},$$

which is the basis of the algorithm called *forward-substitution* and presented in Figure 2.11. In a similar way, the *back-substitution* algorithm to solve an upper triangular system of linear equations can be derived.

A direct method for the solution of a general system of linear is based on the factorisation of the matrix A. Since systems of linear equations with diagonal and triangular matrices have straightforward algorithms, the interesting factorisations are those which efficiently factorise matrices into the product of matrices with these properties. Taking LU-factorisation as an example, the matrix A is factorised as $A = LU$, where L is unit-diagonal ($l_{ii} = 1$) lower triangular, and

```
for i = 1 to n
 x_i ← b_i
 for j = 1 to i − 1
  x_i ← x_i − a_ij x_j
 end for
 x_i ← x_i / a_ii
end for
```

Figure 2.11: Forward-substitution algorithm for a system of linear equations with A lower triangular.

U is upper triangular. Given this factorisation, the system of linear equations $Ax = b$ can be rewritten as $LUx = b$. Thus the system $Ax = b$ can be solved, by forward-substitution for $Ly = b$ and back-substitution for $Ux = y$. Table 2.5 presents some other factorisations developed for systems of linear equations where matrix A has particular properties. Each of these factorisation algorithms can be specialised for nonzero structures.

Pivoting is a technique that is used within factorisations to keep the error of the solutions as low as possible. It is out of the scope of this thesis to present floating point arithmetic [Gol91], demonstrate the error bounds of solutions obtained by different factorisations with and without pivoting [Hig96] and therefore the need of pivoting.

When the coefficient matrix is sparse, a factorisation creates new nonzero elements in the factor matrices where zero elements were in the coefficient matrix. Each of these new nonzero element is called a *fill-in element*. Reordering the equations and the variables can reduce the number of fill-in elements. A reordering transforms the coefficient matrix by interchanging rows and columns. The execution time is reduced by reducing the number of fill-in elements since this preserves the sparsity of the coefficient matrix and so can be exploited.

The solution of sparse systems of linear equations is divided into reordering the coefficient matrix, factorisation and solve. An ordering implementation can take in account the numerical values or simply consider the position of the nonzero elements; *sparsity pattern*. A *numerical ordering*, first kind of ordering implementations, produces a reordering which includes the pivoting and performs a factorisation. The posterior factorisation may be only used by other systems of linear equations which have a similar sparsity pattern. A *symbolic ordering*, second kind of ordering implementations, produces a reordering which does not

When A dense or banded – LU-factorisation defined as $A = LU$ where L unit-diagonal lower triangular and U upper triangular
When A symmetric positive definite – Cholesky factorisation defined as $A = U^T U$ or $A = LL^T$ where L lower triangular and U upper triangular
When A symmetric positive definite tridiagonal – LDL^T-factorisation defined as $A = LDL^T$ or $A = UDU^T$ where L is unit-diagonal lower bidiagonal, U is unit-diagonal upper bidiagonal and D is diagonal
When A symmetric indefinite – Symmetric indefinite factorisation defined as $A = LDL^T$ or $A = UDU^T$ where L is unit-diagonal lower triangular, U is unit-diagonal upper triangular and D is block diagonal with blocks of order 1 or 2

Table 2.5: Recommended factorisations for systems of linear equations with dense and banded matrices.

include pivoting and is used by posterior factorisations. A numerical ordering uses dynamic data structures to store the coefficient matrix since the number of fill-in elements is not known until it is actually performed. Consequently, posterior factorisations can use a static storage format. A symbolic ordering also uses dynamic data structures, but its posterior factorisation uses dynamic data structures to account for the fill-in elements which are produced as a consequence of the pivoting.

An ordering implementation communicates the reordering to a factorisation. Some reorderings are represented as matrices known as permutation matrices. Other reorderings are represented as trees such as elimination trees [Liu90].

The numerical linear algebra community has not yet been able to determine the matrix properties for which each ordering algorithm is adequate.

For a more detailed approach to direct methods for linear systems of equations see [Ste73], [DER86], [GvL96], and [TI97].

Iterative Methods

The algorithms classified as iterative methods are mainly used with sparse matrices. The number of iterations necessary to achieve a sufficiently accurate solution defines the cost of these algorithms. This number depends on the characteristics of matrix A. For this reason, iterative algorithms usually involve the calculation of an extra matrix, a *preconditioner*, that transforms matrix A into one with more

favourable characteristics. The favourable characteristics can be seen as matrix properties but the cost of the algorithm to test these properties is comparable to the cost of solving the sparse system of equations. Thus, in practice, the choice of preconditioner and iterative algorithm cannot be determined as a function of matrix properties; it is a process determined by experimentation and testing of different combinations. For a more technical approach to iterative methods for linear systems of equations see [BBC+94], [Axe94] or [Saa96].

2.3.3 Storage Format Abstraction Level

The storage format abstraction level is defined as the level of abstraction of an implemented matrix operation that knows the representations of the matrix operands and accesses these directly.

As an example, take the matrix-matrix multiplication algorithm with A upper triangular and B dense (see Figure 2.7). An implementation of this algorithm using dense format for both A and B is presented in Figure 2.12. NumType is the data type of the matrix elements (real, complex, ...). Reading the code of this implementation, it can be seen that each matrix is stored in a two-dimensional array (i.e. dense format). This means that, if A instead is stored in packed format then the implementation is no longer valid. Figure 2.13 presents an implementation of the same algorithm, but with A stored in packed format (i.e. as a one-dimensional array).

```
NumType A(m,m)
NumType B(m,n)
NumType C(m,n)

do j=1,n
 do i=1,m
  temp = 0
  do k=i,m
   temp = temp + A(i,k)*B(k,j)
  end do
  C(i,j) = temp
 end do
end do
```

Figure 2.12: Implementation of matrix-matrix multiplication $C \leftarrow AB$ with A upper triangular and B dense, both stored in dense format.

```
NumType APACK(m*(m-1)/2+m)
NumType B(m,n)
NumType C(m,n)

do j=1,n
 do i=1,m
  temp = 0
  do k=i,m
   temp = temp + APACK(i+k*(k-1)/2)*B(k,j)
  end do
  C(i,j) = temp
 end do
end do
```

Figure 2.13: Implementation of matrix-matrix multiplication $C \leftarrow AB$ with A upper triangular stored in packed format and B dense stored in dense format.

Note that an implementation is at storage format abstraction level if changing the storage format implies changing the implementation. Traditional library implementations of the matrix calculations are implemented at this abstraction level.

To summarise the contents of this section, a given matrix calculation has many specialised algorithms. For each of these algorithms there can be many implementations corresponding to different storage formats for the matrix operands. Thus there is an explosion in the number of possible implementations. The developers of these libraries have to balance the number of implementations (i.e. algorithms and storage formats) that are supported with the effort of developing the code.

2.4 Developing Numerical Linear Algebra Programs

To review the contents of preceding sections:

- matrices can be classified by different criteria and each classification is known as a matrix property;
- a given matrix can have different storage formats;
- for each matrix calculation many algorithms that take particular advantage of the matrix properties can be derived;

- for each algorithm many implementations are necessary due to the different storage formats;
- for block banded, banded and dense matrices, the implementation to use for matrix calculation can be decided as a function of the matrix properties and their storage formats;
- for sparse systems of equations, either direct or iterative methods, it is not possible automatically to select the implementation (i.e ordering implementation or combination preconditioner iterative method).

The objective of this section is to understand how these concepts are organised in traditional linear algebra libraries. The term "traditional libraries" refers to the libraries developed, in this case by the numerical linear algebra community, using top-down methodology and implemented in imperative languages, predominantly Fortran, with no programmer-defined data types. BLAS [BLA99] and LAPACK [ABD+95] are chosen as examples of traditional libraries to be described. An important characteristic is the community consensus or *de facto* standardisation process which is behind their design. Other examples of libraries are LINPACK [DBMS79], EISPACK ([SBD+76], [GBDM77]), LAPACK [ABD+95], NAG[1], IMSL[2], SPARSPAK ([GL79], [GL81]), YSMP [EGSS82], MA28 [Duf77].

BLAS and LAPACK are compared with two alternative linear algebra environments: Matlab [Mat] and the Sparse Compiler ([Bik96], [BW96], [BW99], [BBKW98]). Rather than a theoretical discussion about the three possibilities, the matrix calculations introduced in Section 2.3 are used to illustrate the differences, advantages and disadvantages.

Matlab is a computing environment and programming language for numerical computations. Its main characteristic is that the programming language is matrix-based. Thus, a Matlab program for linear algebra resembles its mathematical form.

The Sparse Compiler parses a given dense Fortran 77 program into an equivalent sparse Fortran 77 program. A dense program means a linear algebra program that stores its matrices in dense format even if some of matrices have some nonzero elements structure. An equivalent sparse program means a linear algebra

[1] A commercial product of Numerical Algorithms Group Inc. http://www.nag.com

[2] International Mathematical and Statistical Libraries (IMSL) a commercial product of Visual Numerics Inc. http://www.vni.com

program that implements the same calculations but those matrices with nonzero elements structures are stored in advisable storage formats. The Sparse Compiler analyses the nonzero elements structure of matrices and transforms the parts of the dense program that define the matrices so detected to have certain nonzero elements structure, and the parts of the dense program that operate on these matrices. The dense program is transformed so that it uses the new storage formats selected by the compiler and exploits the nonzero elements structure of the matrices.

2.4.1 Using BLAS and LAPACK

BLAS (Basic Linear Algebra Subprograms) offers subroutines for basic matrix operation while LAPACK (Linear Algebra Package) offers subroutines for systems of linear equations, least square problems, and eigenvector and eigenvalue problems. Both libraries are implemented in Fortran 77 and are designed to provide high performance [DW95], i.e. to achieve maximum performance from a given computer.

The routines provided by the BLAS are divided into three groups:

- Level 1 BLAS – routines that require $O(n)$ floating point operations and involve $O(n)$ data items [LHKK79], e.g. dot product $x^T y$ or a vector norm $||x||_1$,
- Level 2 BLAS – routines that require $O(n^2)$ floating point operations and involve $O(n^2)$ data items ([DCHH88b], [DCHH88a]), e.g. matrix vector multiplication Ax, and
- Level 3 BLAS – routines that require $O(n^3)$ floating point operations and involve $O(n^2)$ data items ([DCHD90],[DCHD90]), e.g. matrix-matrix multiplication AB.

BLAS subroutines have been specified for dense, banded, sparse, symmetric, symmetric banded, upper and lower triangular, and upper and lower triangular banded matrices. Dense matrices are stored in dense format (GE). Banded matrices are stored in band format (GB). Symmetric matrices are stored in dense (SY) or packed format (SP). Triangular matrices are stored in dense (TR) or packed format (TP). Triangular band matrices are stored in band format (TB) and also symmetric banded (SB). Finally, sparse matrices (US) are stored in coordinate or

compressed sparse column or compressed sparse row or sparse diagonal or block coordinate or block compressed sparse column or block compressed sparse row or block sparse diagonal or variable block compressed sparse row format.

Based on the case of matrix-matrix multiplication (Section 2.3), the process of developing a linear algebra program with the BLAS is described below. Given the problem description $C \leftarrow AB$ where A and B are known to be dense, the first task is to find the correct BLAS subroutine. The subroutine names follow a strict naming scheme: the first letter of the name indicates the numerical data type (REAL, DOUBLE PRECISION, COMPLEX and DOUBLE COMPLEX) of the operands; the next two letters specify the matrix properties and the storage format (in the preceding paragraph, the pairs of letters between parenthesis show the different combinations and their meanings); the final three letters indicate the matrix operation. Table 2.6 includes the specification of the different subroutines for matrix-matrix multiplication. Apart from the number of subroutines, the long lists of parameters make for an unfriendly interface.

Following the naming scheme, the xGEMM subroutine is selected. The parameters pass information about the sizes of matrix operands, the representation of the three matrix storage formats, and flags to indicate if any of the matrices have to be transposed. The functionality of xGEMM implements four matrix operations: two matrix scalings, one matrix-matrix multiplication and one matrix-matrix addition ($C \leftarrow \alpha AB + \beta C$). The reason for these extensions to the basic matrix-matrix multiplication is that all the operations can be implemented within the three nested loops of matrix-matrix multiplication and it is, thus, more efficient than separating the operations.

For the case where A is upper triangular, the appropriate subroutines are xTRMM and xTPMM. The first subroutine implements the operation using dense format, while the second uses packed format. If the first subroutine is selected, memory space might be wasted, whereas if xTPMM is selected, the user must understand and create the representation (packed format) required by the subroutine.

For the case where A and B are both upper triangular, the appropriate subroutines are again xTRMM and xTPMM. BLAS subroutines have been developed in such a way that only one of the input matrices (for binary operations) is considered to have properties others than dense. Hence, the BLAS are not complete in the sense that not all of the possible implementations are included. In this

Subroutine Specification	Functionality
xGEMM(TRANSA, TRANSB, M, N, K, ALPHA, A, LDA, B, LDB, BETA, C, LDC)	$C \leftarrow \alpha op(A)op(B) + \beta C$
xGBMM(SIDE, TRANSA, TRANSB, M, N, K, KL, KU, ALPHA, A, LDA, B, LDB, BETA, C, LDC)	$C \leftarrow \alpha op(A)op(B) + \beta C$ or $C \leftarrow \alpha op(B)op(A) + \beta C$ where A is banded stored in band format
xSYMM(SIDE, UPLO, M, N, ALPHA, A, LDA, B, LDB, BETA, C, LDC)	$C \leftarrow \alpha AB + \beta C$ or $C \leftarrow \alpha BA + \beta C$ where A is symmetric
xSBMM(SIDE, UPLO, M, N, K, ALPHA, A, LDA, B, LDB, BETA, C, LDC)	$C \leftarrow \alpha AB + \beta C$ or $C \leftarrow \alpha BA + \beta C$ where A is symmetric banded stored in band format
xSPMM(SIDE, UPLO, M, N, ALPHA, AP, LDA, B, LDB, BETA, C, LDC)	$C \leftarrow \alpha AB + \beta C$ or $C \leftarrow \alpha BA + \beta C$ where A is symmetric stored in packed format
xTRMM(SIDE, UPLO, TRANSA, DIAG, M, N, ALPHA, A, LDA, B, LDB)	$B \leftarrow \alpha op(A)B$ or $B \leftarrow \alpha Bop(A)$ where A is unit-diagonal or not and upper or lower triangular
xTBMM(SIDE, UPLO, TRANSA, DIAG, M, N, K, ALPHA, A, LDA, B, LDB)	$B \leftarrow \alpha op(A)B$ or $B \leftarrow \alpha Bop(A)$ where A is unit-diagonal or not and upper or lower triangular banded stored in band format
xTPMM(SIDE, UPLO, TRANSA, DIAG, M, N, ALPHA, AP, LDA, B, LDB)	$B \leftarrow \alpha op(A)B + \beta C$ or $B \leftarrow \alpha Bop(A) + \beta C$ where A is unit-diagonal or not and upper or lower triangular stored in packed format
xUSMM(TRANSA, K, ALPHA, A, B, LDB, BETA, C, LDC)	$B \leftarrow \alpha op(A)B + \beta C$ where A is sparse stored in a sparse format

Table 2.6: BLAS subroutines for matrix-matrix multiplication – $op(A)$ represents A or A^T and, unless indicated, matrices are stored in dense format.

case, the waste of memory space is larger since the three matrices involved are all upper triangular, and only one matrix can be stored in packed format.

LAPACK subroutines are divided into those that solve standard problems, called driver subroutines, and presented in Section 2.1, and those which compute factorisations and other calculations used by the driver subroutines. Another long list of matrix properties and storage formats is supported and is organised following the naming scheme described with BLAS.

Figures 2.14, 2.15 and 2.16 present pseudo-Fortran programs to solve the system of linear equations $ABx = c$ where A and B are $n \times n$ matrices. The programs on the left hand side of these figures follow an algorithm which first performs the matrix-matrix multiplication and then solves the system of equations. Alternatively, the programs on the right hand side of these figures follow an algorithm which first solves the system of linear equations $Ay = c$ and then the system $Bx = y$. Both algorithms are semantically equivalent, i.e. they calculate the same result assuming perfect floating point arithmetic. Figure 2.14 presents programs to solve the system of linear equations $ABx = c$ where A and B are $n \times n$ dense matrices. Figures 2.15 and 2.16 presents programs to solve the same problem, but here A and B are upper triangular matrices. The first figure uses dense format while the second figure uses packed format, whenever possible.

```
NumType A(n,n)
NumType B(n,n)
NumType D(n,n)
NumType xc(n,1)
INTEGER IPIV(n), INFO

call initialise(A,B,xc)
C D=A*B

call XGEMM('N', 'N', n, n, n,
1.0, A, n, B, n, 0.0, D, n)
C solve system Dx=xc and leave x
in xc
call XGESV(n, 1, D, n, IPIV, xc,
1, INFO)
```

```
NumType A(n,n)
NumType B(n,n)

NumType xc(n,1)
INTEGER IPIV(n), INFO

call initialise(A,B,xc)
C solve system Ay=xc and leave y
in xc
call XGESV(n, 1, A, n, IPIV, xc,
n, INFO)
C solve system Bx=xc and leave x
in xc
call XGESV(n, 1, B, n, IPIV, xc,
1, INFO)
```

Figure 2.14: Programs using BLAS and LAPACK to solve the system of equations $ABx = c$ where A and B are $n \times n$ dense matrices.

```
NumType A(n,n)
NumType B(n,n)
NumType xc(n)

call initialise_tr(A,B,xc)
C B=A*B

call XTRMM('L', 'U', 'N', 'N',
n, n, 1.0, A, n, B, n)
C solve system Bx=xc and leave x
in xc
call XTRSV('U', 'N', 'N', n,
1.0, B, n, xc, 1)
```

```
NumType A(n,n)
NumType B(n,n)
NumType xc(n)

call initialise_tr(A,B,xc)
C solve system Ay=xc and leave y
in x
call XTRSV('U', 'N', 'N', n,
1.0, A, n, xc, 1)
C solve system Bx=xc and leave x
in xc
call XTRSV('U', 'N', 'N', n,
1.0, B, n, xc, 1)
```

Figure 2.15: Programs using BLAS and LAPACK to solve the system of equations $ABx = c$ where A and B are $n \times n$ upper triangular matrices stored in dense format.

```
NumType APACK(n,n)
NumType B(n,n)
NumType xc(n)

call initialise_tr(APACK,B,xc)
C B=APACK*B

call XTPMM('L', 'U', 'N', 'N',
n, n, 1.0, APACK, n, B, n)
C solve Bx=xc and leave x in xc

call XTRSV('U', 'N', 'N', n,
1.0, B, n, xc, 1)
```

```
NumType APACK(n*(n-1)/2+n)
NumType BPACK(n,n)
NumType xc(n)

call initialise_tr(APACK,BPACK,xc)
C solve APACKy=xc and leave y in
xc
call XTPSV('U', 'N', 'N', n,
1.0, APACK, xc, 1)
C solve BPACKx=xc and leave x in
xc
call XTPSV('U', 'N', 'N', n,
1.0, BPACK, xc, 1)
```

Figure 2.16: Programs using BLAS and LAPACK to solve the system of equations $ABx = c$ where A and B are $n \times n$ upper triangular matrices stored in packed format, whenever possible.

To summarise, this process can be generalised to describe the development of linear algebra programs with traditional libraries as:

- describe the problem in terms of matrix calculations,
- analyse the matrices to determine their properties,
- select the library or libraries which support the operations and properties,
- select the subroutines which best fit the matrix properties, and
- declare the variables conforming to the storage format that is supported by the selected subroutines.

2.4.2 Using Matlab

Matlab is not only an environment for numerical linear algebra; regressions, interpolation, numerical integration, graphs, visualisation of results, etc. are integrated. Its major characteristic is that the programming language is matrix based, i.e. every variable is a matrix. For example, the multiplication of two matrices $C \leftarrow AB$ is written as `C=A*B` and the solution of a system of linear equations $Ax = b$ can be written as `x=A\ b` or `x=inv(A)*b` where `inv(A)` performs A^{-1}.

Matlab does not always exploit the matrix properties that are supported in LAPACK and BLAS, and uses only dense and compressed sparse column format for sparse matrices [GMS92].

Matlab provides LU, Cholesky, QR, Eigenvalue and Singular value factorisations. Thus, for example, the solution of a system $Ax = b$ using LU-factorisation is written as `[L,U]= lu(A); y=L\ b; x=U\ y;`. The "\" operator follows the algorithm:

- if the matrix is not square then solve least squares problem,
- otherwise, if the matrix is triangular then use back or forward substitution,
- otherwise, if it is symmetric and the diagonal elements are positive real[3] then attempt to solve with Cholesky factorisation,
- otherwise (i.e. Cholesky factorisation fails or is not symmetric with positive diagonal elements), solve with LU-factorisation.

[3]Heuristic used by Matlab to test if a matrix could be positive definite.

Figure 2.17 presents Matlab programs to compute the system of linear equations $ABx = c$ where A and B are dense matrices. Figure 2.18 presents Matlab programs to compute the same problem except that A and B are upper triangular matrices. Note that both figures present identical programs, but for the initialisation. Although transparent for users, the "\" operator solves the system using LU factorisation for the first figure while for the second figure it uses back-substitution.

```
initialise(A,B,c)
D=A*B;
x=D\c;
```

```
initialise(A,B,c)
y=A\c;
x=B\y;
```

Figure 2.17: Matlab Programs to solve the system of equations $ABx = c$ where A and B are $n \times n$ dense matrices.

```
initialisetr(A,B,c)
D=A*B;
x=D\c;
```

```
initialisetr(A,B,c)
y=A\c;
x=B\y;
```

Figure 2.18: Matlab Programs to solve the system of equations $ABx = c$ where A and B are $n \times n$ upper triangular matrices.

The task of developing a linear algebra program with Matlab follows the steps:

- describe the problem in terms of matrix calculations,
- analyse the matrices to identify matrix properties, and
- map the problem into Matlab operators.

2.4.3 Using the Sparse Compiler

The Sparse Compiler is a source-to-source compiler that has as input dense Fortran 77 programs and as output sparse Fortran 77 programs. A dense program is a program which stores all the matrices in dense format (in Fortran 77 case `NumType A(n,m)`) and the matrix calculations are implemented using all the elements. A sparse program is a program that stores and implements the matrix calculations taking advantage of matrix properties. The compiler is divided into two phases: dense program analysis and sparse code generation.

The program analysis automatically detects the nonzero elements structure of matrices [BW99] and identifies the parts of the code that access zero elements. The user of the compiler can also provide information about the nonzero elements structure of the matrices through comments. Figure 2.19 presents the notation used in the comments to declare an upper triangular matrix.

The code generation phase takes into account the nonzero elements structure and how the matrices are accessed in order to select the storage format and automatically generate the sparse code ([BW96], [BBKW98]). In other words, the compiler changes the dense format declaration of some matrices by the declaration of the selected new storage format. It also eliminates the redundant instructions because of the nonzero elements structure found. Finally, it transforms those parts of the program that accessed matrices so that they align with the new storage formats.

The limitation of this work is that in some cases, specially hand optimised programs, the compiler fails to fully exploit the sparsity. Its second limitation is that reordering algorithms cannot be used, thus the fill-in effect, creation of nonzero elements where there were zero elements, cannot be avoided and usually the resultant sparse code could be significantly improved.

Figure 2.19 presents the Fortran 77 dense program commented for the sparse compiler to compute $ABx = c$ where A and B are upper triangular. Note that no support is provided by the compiler to develop the dense programs so usually the dense sub-set of BLAS or LAPACK would be used.

```
NumType A(n,n)
C_SPARSE(ARRAY(A), ZERO (I>J))
C_SPARSE(ARRAY(A), DENSE(I<=J)
NumType B(n,n)
C_SPARSE(ARRAY(B), ZERO (I>J))
C_SPARSE(ARRAY(B), DENSE(I<=J)
```

Figure 2.19: Sparse Compiler commented dense program to solve the system of equations $ABx = c$ where A and B are $n \times n$ upper triangular matrices.

The task of developing a linear algebra program with the sparse compiler follows the steps:

- describe the problem with matrix calculations,
- generate Fortran 77 dense program for the matrix calculations, and

- indicate the nonzero elements structure, or let the compiler give feedback on this.

2.4.4 Advantages and Disadvantages

From the user's point of view, Matlab provides the easiest way to generate a linear algebra program. The users do not need to know how the matrices are stored or how the operators are implemented. The mapping of the matrix calculation is straightforward, although it has been shown that a given matrix calculation can have different semantically equivalent programs. The main drawback is the execution time of the programs since the user does not provide information about matrix properties, and except in specific situations, the programs cannot take advantage of them.

The Sparse Compiler represents the next level of difficulty. The user has to write Fortran linear algebra programs and thereby has to know how the matrix calculations are implemented using dense format. However, the sparse compiler offers support to decide the nonzero elements structure and exploits any such structure that is found. Neither Matlab nor BLAS and LAPACK libraries provide such functionality.

BLAS and LAPACK represent the maximum level of difficulty. Matrices can be represented in different storage formats and the user has to know how to declare them. The selection of a subroutine is not a trivial process. The list of parameters is complicated and too long to remember, therefore difficult to use. The users have to know how to declare the different storage formats. The functionality is not complete, not all the possibilities of matrix properties and storage format operands are observed. On the other hand, BLAS and LAPACK subroutines deliver the minimum execution time as they utilise state-of-the-art implementations.

The user perceives the difficulty of developing a linear algebra program as the distance to jump from the problem defined in terms of matrix operations to the specific software environment expression (subroutines in traditional libraries, operators in Matlab and comments and dense program in the Sparse Compiler). This distance is represented by the tasks that need to be completed in order to develop the program. These tasks are:

- matrix properties analysis,

- selection of storage formats,
- and selection of specific environment expressions that align with the properties and storage formats.

2.5 Summary

Matrix calculations are the core of this chapter; beginning with their definitions, continuing with characterisation examples of how matrix calculations are implemented, and ending with how they are organised in libraries.

Matrix calculations have been divided into basic matrix operations and matrix equations. Due to certain matrix properties, the definition of a basic matrix operation can be specialised and thus different algorithms that exploit the matrix properties are created. Due to the different storage formats of a matrix, the set of algorithms are further extended into a set of implementations.

Matrix equations can be solved either with direct or iterative methods. Direct methods perform a factorisation and then solve the systems for the factored matrices. When the matrix equations are sparse, the matrix can be reordered to preserve the sparsity of the factored matrices. However, it is not possible to decide efficiently which of the different ordering algorithms is the adequate one. Iterative methods are usually combined with preconditioners. Some iterative algorithms are known to fail to converge to a solution for specific matrix properties. In practice, the appropriate combination of iterative method and preconditioner for a system of linear equations cannot be decided automatically.

Traditional libraries are organised by strict naming schemes. For each subroutine the naming scheme describes the matrix calculation, the matrix properties of the input matrices and their storage formats. The parameters describe how the storage formats are represented.

The comparison of the BLAS and LAPACK with Matlab and with the Sparse Compiler shows that when developing a linear algebra program the BLAS and LAPACK based programs constitute the maximum level of difficulty. The difficulty is summarised by the tasks to be completed:

- describe the problem in terms of matrix calculations,
- analyse matrices to determine their properties,

- select the library or libraries that support the matrix calculations and properties,
- select the subroutines which best fit the matrix properties, and
- declare the variables conforming to the storage format that is supported by the selected subroutines.

The information of this chapter is reused mainly to the next chapter, which reports an object oriented analysis and design of linear algebra.

Readers are referred to [Gan59a] and [Gan59b] for a more detailed, analytical, approach to Numerical Linear Algebra. Descriptions and analysis of algorithms for matrix calculations can be found in [Ste73], [GvL96] and [TI97]. Detailed study of accuracy and stability of these algorithms can be found in [Hig96].

Chapter 3

Object Oriented Linear Algebra

Traditional libraries of linear algebra present two weaknesses: complex interfaces and an explosion of implementations of matrix calculations. The first weakness affects users since they find it hard to develop linear algebra programs using these libraries. The second weakness affects library developers since they have to code the many different implementations.

This chapter focuses on the analysis and design of an object oriented linear algebra library in order to overcome or reduce the two weaknesses. Object oriented software construction offers the possibility to define and use abstractions from a problem domain, in this case linear algebra. The objective is to create an object oriented model of linear algebra that hides the implementation details.

Object oriented software construction is reviewed in order to be able to create object oriented models of linear algebra (Section 3.1). The object oriented models are displayed graphically using a subset of UML notation. This notation is also introduced in Section 3.1.

Different models are proposed and used to classify several existent object oriented linear algebra libraries (Section 3.2). The object oriented model created identifies that current models do not model fully Linear Algebra. The new model constitutes the design of the Object Oriented Linear Algebra LibrAry (OoLaLa). This model enables OoLaLa to automatically manage the storage formats of matrices and propagate the matrix properties through matrix calculations. In addition, two implementation abstraction levels are described and both reduce the explosion of implementations of matrix calculations.

3.1 Object Oriented Software Construction

The process of developing software, applications or libraries, is inherently a human activity. A group of human software developers analyses certain problem, creates a model of it, and develops an implementation of that model in a programming language. As with many other problems faced by humans, the model to solve a given problem is created by dividing the problem into sub-problems repeatedly so that they eventually become trivial to solve. The model for the problem is then created as the composition of the sub-models.

> "The technique of mastering complexity has been known since ancient times: *divide et impera* (divide and rule)." Dijkstra [Dij79]

Top-down methodology, or structured programming, used by traditional linear algebra libraries, divides problems using an algorithmic decomposition, i.e. expressing what has to be done in terms of basic control structures (loops, if-then, etc.) or basic algorithms (sort, search, etc.). The basic decomposition unit is the subroutine or procedure, and thus the model is a composition of subroutine calls.

On the other hand, bottom-up methodology searches for abstractions of the problem domain, and divides the problem into an appropriate set of these abstractions. An abstraction is a key concept of the problem domain with the operations or services provided within that domain. A model of the problem is the interaction of abstractions through their defined operations, or interfaces. Special importance is given to hiding details of how the operations of the abstractions are implemented; thereby emphasis is simply placed on using the operations. In the literature, the abstractions are known as *abstract data types*.

The main advantage for software developers is that abstractions are a normal human approach to decomposing problems whereas algorithmic decomposition is an influence of what is provided by the first programming languages, such as Fortran 66, Fortran 77 or C. Using an example, it is not attractive to pass as parameters the representation of an abstraction, instead of the abstraction itself. Nowadays, few software developers would operate on an array when they want to use a stack. They would use an abstraction of the stack, often provided by modern programming languages, and use the interface (push, pop, etc.) to operate on the stack, even if it is ultimately represented as an array. Traditional linear algebra libraries are implemented accessing directly (not using an interface) the representation of the matrix, and the explosion in the number of subroutines

that this provokes has been demonstrated in Chapter 2.

The objective of object oriented methodology is to propose an even more similar human approach to modelling complex problems. Object oriented methodology follows the bottom-up methodology and its basic concepts are explained in the next section. The motivation for object oriented methodology is to overcome the lack of abstraction which forces developers to always think about the problems in too much detail, thereby becoming error prone.

The remaining of the section is organised as follows. First, basic concepts (objects, classes, inheritance, client relation, etc.) of object oriented methodology (Section 3.1.1). These basic concepts are illustrated using examples from linear algebra. Some object oriented programming languages offer abstract classes and generic classes (Section 3.1.2). These are explained so as they are used in the posterior analysis and design. The next issue is to understand how the software development process is modified because of object orientation (Section 3.1.3). Finally, two design patterns and a short discussion about generic classes vs. inheritance (or how they can be simulated) are the suggested "tips" (Section 3.1.4).

3.1.1 Basic Concepts

Object oriented methodology is based on abstractions and information hiding, but includes another characteristic common of the human approach to decomposition of problems; classification. This new possibility enables software developers to create abstractions that are families of abstractions. Using object oriented terminology, the abstractions are now called *classes* and a specific member of an abstraction is called an *object*. Every object is said to be an *instance* of a class. For example, matrices might be an abstraction from the linear algebra problem domain and hence a class. A specific matrix would be an object of the class. Classes define common operations, such as "assign an element in certain position" or "access an element in certain position", and common characteristics that every object would have, such as the number of rows or number of columns. Classes have a static role since they are just definitions. Every object of a class conforms to the definitions described by the class and gives values to those definitions, also known as the state of an object. An object is dynamic since it is a run-time entity whose state can be modified.

Figure 3.1 presents a UML class diagram and object diagram of matrices. UML stands for Unified Modelling Language ([Rat97a], [Rat97b], [Mul97], [BRJ99]),

which is an industrial standard notation, used to document object oriented software development. Other object oriented and structured programming notations can be found in [Wie98]. Class diagrams are used to represent classes graphically using a rectangle divided into three sub-rectangles. The first sub-rectangle contains the *class name*, the second contains the characteristics called *attributes* and the third contains the operations called *methods*, or *operations*. An object diagram is used to represent objects and is similar to the class diagram. The first sub-rectangle contains the name of the object and its class, separated by a colon and underlined. The second sub-rectangle contains the attributes that define the state of the object, and the third one contains the methods. As can be seem in the class diagram (Figure 3.1), there is a method `create` which creates objects of class `Matrix`. This method is a class method and hence appears underlined. A class method is a method that cannot be invoked in an object; it is invoked in the class. For the sake of clarity objects and classes are often represented in class diagrams and object diagrams only by their first sub-rectangle, thus not repeating known information. UML specifies how attributes and methods have to be declared in the diagrams. This thesis does not follow this specification and uses a pseudo-code based on Java syntaxes.

Once some basic UML notation has been introduced, attention is directed again to the possibility of classifying classes. Humans create hierarchies of abstractions using criteria by which each classification adds new characteristics or re-adapts existing ones. In object oriented methodology, the classes can be organised into *inheritance* hierarchies. Each class is a classification, and traversing upwards in the class hierarchy means a more general class, whereas traversing down the class hierarchy means a more specialised class. Using the example of matrices, vectors can be considered a special class of matrices, since they are matrices with either only one column or one row. In a similar way, square matrices can be considered a special class of matrices since they are matrices whose number of columns and rows has to be equal. These examples should be taken as naïve examples to illustrate the concepts. A more complete object oriented analysis and design is described in Section 3.2. Figure 3.2 presents a UML class diagram showing the inheritance relation between the class `Matrix` and the classes `ColumnVector`, `SquareMatrix` and `RectangularMatrix`. The inheritance relation is represented by an arrow which begins in the specialised class and ends in a more general class. The class `Matrix` defines the attributes, the methods

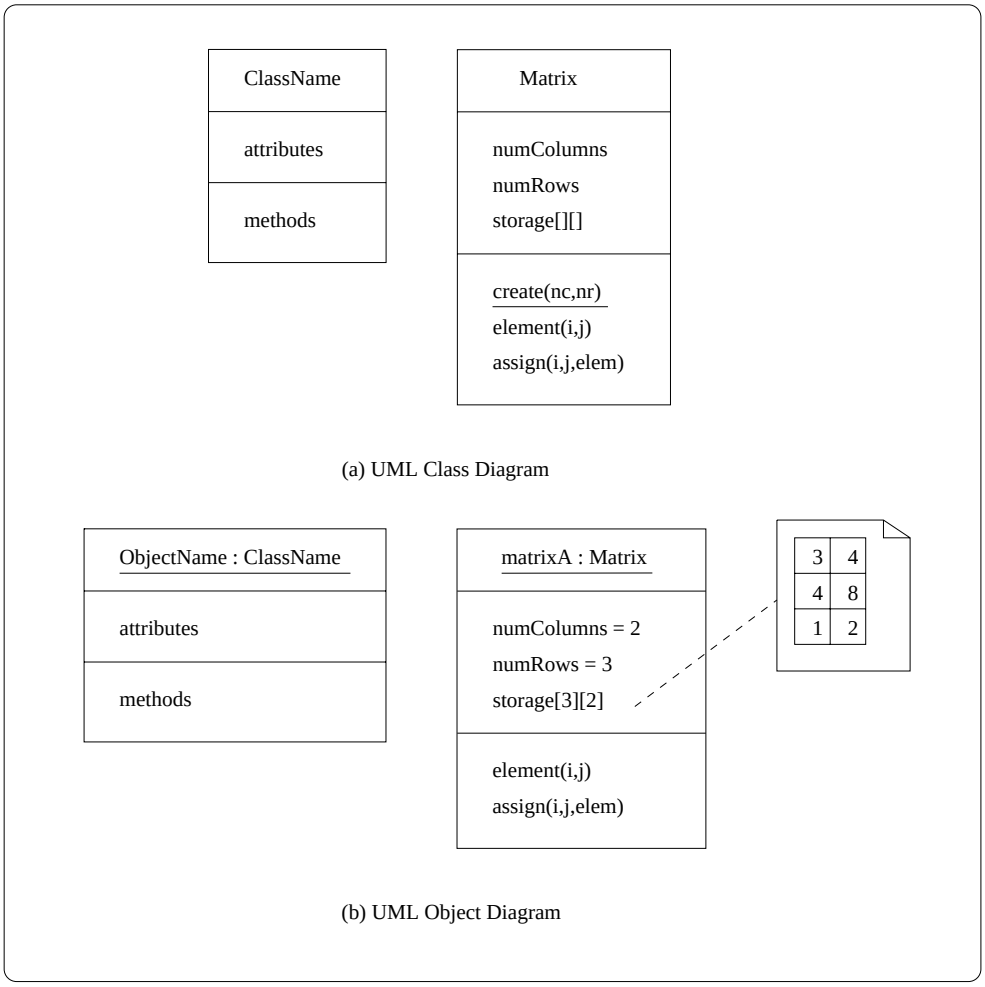


Figure 3.1: UML class diagram and object diagram for a naïve version of matrices.

and the implementations of the methods. All this is automatically inherited by the sub-classes `ColumnVector`, `SquareMatrix` and `RectangularMatrix`. A sub-class can add new methods or attributes, and also can adapt (re-implement) the methods inherited. In the class diagram, the method norm1 has been added to the class `Matrix` presented in Figure 3.1. The `norm1` method is implemented in this class following the definition for matrices ($max_j \sum_i |a_{ij}|$). However, in the class `ColumnVector` this method `norm1` is re-implemented efficiently for vectors ($\sum_i |x_i|$). The class `SquareMatrix` adds a new method `create`, which only needs one parameter for the number of rows and columns, and re-implements the inherited method `create` so that the parameters for the number of rows and columns are tested to be equal before an object is constructed.

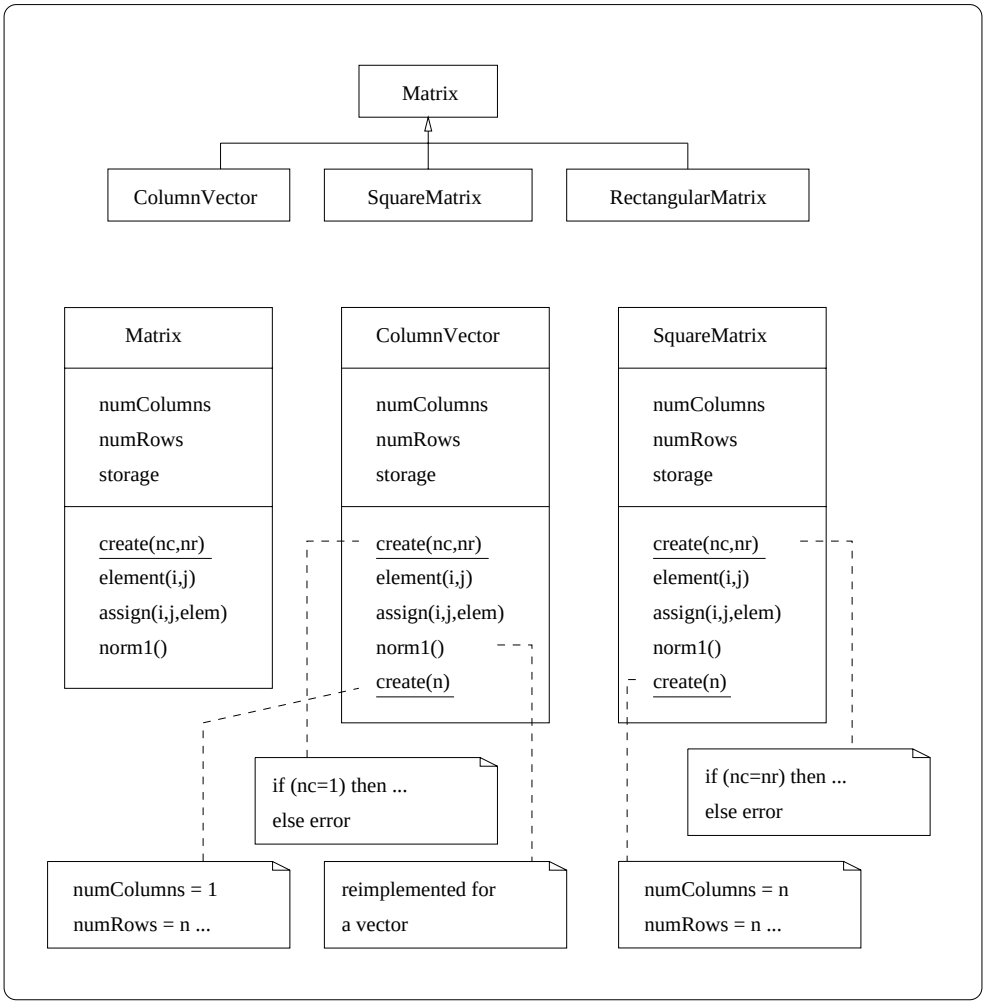


Figure 3.2: UML class diagram with a naïve inheritance hierarchy of matrices.

Classes can be seen as the data types defined by developers. The inheritance of a class `B` from a class `A` means that every object instance of class `B` is also an object of class `A`. In the case of matrices, every object of class `Vector` is always an object of class `Matrix`. A method that has as input parameter an object of class `A` accepts as valid all the objects of that class `A`. Apart from this and provided that class `B` inherits from `A`, every object instance of `B` is also an object of `A`. Hence, the method also accepts as valid the objects of `B`. In general, any object of a class that inherits directly or indirectly (i.e. inheritance through more than one class) from a class is a valid parameter. On the other hand, a second method that takes as input parameters of class `B` does not accept objects that are instances of class `A`. The feature that different objects of different classes are valid for a part of code is called *polymorphism*.

From the above paragraph and using the hierarchy introduced, every object of the classes `ColumnVector`, `SquareMatrix`, `RectangularMatrix` and `Matrix` is a valid parameter for methods that have as parameter an object of class `Matrix`. Suppose that one of these methods calls, or invokes, the method `norm1` in the parameter object of class `Matrix`. Note that the method `norm1` in the class `ColumnVector` is re-implemented while the classes `SquareMatrix` and `RectangularMatrix` inherit the implementation from the class `Matrix`. *Dynamic binding* is the mechanism which ensures that whatever valid object is passed as a parameter to the method, the correct `norm1` implementation would be executed. Dynamic binding identifies the exact class of the object and then checks if an implementation is provided in that class. Otherwise this mechanism traverses upwards through the class inheritance hierarchy, checking at each level whether or not an implementation of the method is provided. For example, when an object of class `ColumnVector` is passed as a parameter, the implementation provided in this class of `norm1` is executed. On the other hand, when an object of class `SquareMatrix` is passed as a parameter, the dynamic binding mechanism detects that its class `SquareMatrix` does not provide an implementation of `norm1`. Hence, it steps up one level to the class `Matrix` where the implementation is found and executed.

Apart from classifying, the inheritance relation between classes is a way of re-using code. Only the methods which need to be adapted to the characteristics of a more specialised class, and those methods specific to that class have to be implemented; the other implementations are simply inherited.

Multiple inheritance is a relation between one class which inherits from more

than one different classes. All the explanations for inheritance are applicable to multiple inheritance, although certain problems that are caused by multiple inheritance, and omitted in this thesis, can arise during the development of object oriented software ([Mey97] Chapter 15).

An *association* between classes represents *links* between objects of these classes. Different variants of associations are defined in UML, but since only the general case (notation defined in Figure 3.3) is necessary in this thesis, the other possibilities are not discussed. The number of objects linked by an association is determined by the *cardinality* of that association. The cardinality is represented by numbers and "*" in the class diagram.

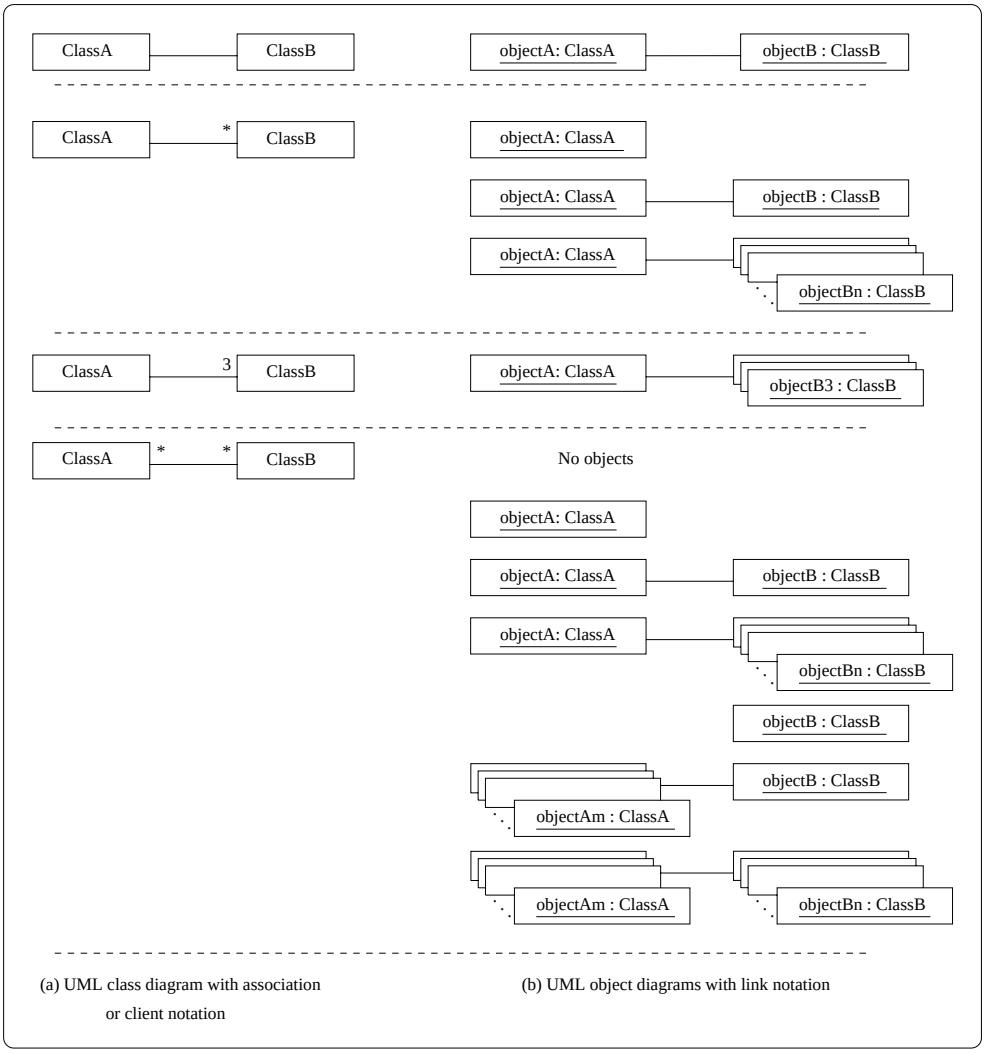


Figure 3.3: UML class and object diagrams with an association or client relation between two classes.

The association between classes represents a path through which methods are invoked by the objects so linked. This metaphoric path symbolises that a method is always invoked by an object in other object (although the other object might be itself). Meyer proposes the term *client relation* instead of association [Mey97]. The term comes from the fact that an object is using the interface of another object (the services provided by the other object) and thus they become client and supplier. The term client relation is used throughout this thesis, rather than association.

Compared with top-down decomposition, object oriented decomposition proposes a method closer to how humans approach problems. The decomposition is based on abstractions from the problem domain. These abstractions can be further abstracted creating hierarchical classifications of abstractions. The process of abstraction hides the details and enables developers to concentrate on how they interact together. The abstractions are called classes and individual members of a class are called objects. An object oriented model of a problem is a set of objects that, over a period of time, are created, destroyed and linked by client relations invoking operations (methods) from other objects. Each class knows the details of how it is implemented but does not know the details of the other's classes, just uses their services.

Object oriented concepts have been presented as an evolution towards a human-like approach to decomposition and composition of complex problems. From this perspective, it offers benefits to the developers of software. From a user perspective, the benefit depends mainly on whether the user is a user of software applications or a user of software libraries. Taking users of libraries, in particular the users of numerical linear algebra libraries, the interfaces would pass from being a list of subroutines with parameters showing the exact representation of the matrices, to operations (methods) between objects representing matrices where the representation and algorithm details are hidden from the user.

3.1.2 Implementation Related Concepts

Generic programming and *abstract classes* are two advanced concepts, which are sometimes supported by object oriented programming languages. These concepts are related to implementation aspects whereas those already explained are methodological.

In general, generic programming enables developers to write parts of programs

that have as a parameter the data type of some variables. Generic programming was proposed from the observation that some algorithms could be written independently of the data types. A typical example is a sorting algorithm. The implementation of a sorting algorithm could be the same as long as the data type of the elements to be sorted has defined the comparison functions "<", ">" and "=". An early version of the Z specification language ([Abr80], [ASM80]), CLU [LAB+81], and Ada [ANS83] are the first languages that supported generic programming.

In object oriented programming languages, a class can also be generic and, thus a *generic class* is a class that has as parameters the data types or classes of some of its attributes or parameters of its methods. Generic classes are also known as template classes in the context of C++. Generic classes cannot instantiate any object since they are not complete classes. In this context, the typical examples for generic classes are the containers of elements. Lists, stacks, trees, etc. are well documented container classes that benefit from generic programming (see the Standard Template Library [LS95], [MS95], [Aus98]). Using generic classes, the containers can be defined independently from the class of the elements they will hold at run-time. In the particular case of linear algebra, the `Matrix` class might be considered a generic class whose parameter is a numerical data type. Figure 3.4 introduces the UML class diagram notation for generic classes and presents the example of class `GenericMatrix`.

Abstract classes are classes which declare methods and attributes but do not implement all the methods. The implemented methods are allowed to invoke the non-implemented methods, called also *abstract methods*. Hence, an abstract class is a completely declared but partially implemented class. No object can be instantiated from an abstract class, and only those classes that inherit from an abstract class and provide implementation for all the inherited abstract methods are not abstract classes. Figure 3.5 presents the UML notation for abstract classes and describes a class diagram for the naïve class hierarchy described in Figure 3.2. In this case, class `Matrix` does not have any attributes since some attributes become redundant for some sub-classes.

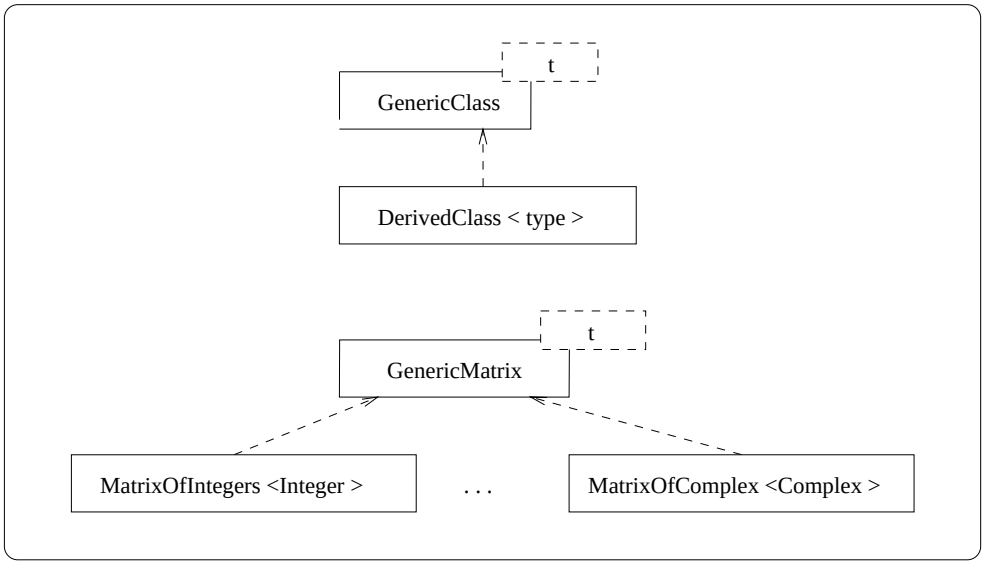


Figure 3.4: UML class diagram of a naïve generic class `GenericMatrix`.

3.1.3 The Software Development Process

Traditionally, the software development process has been divided into *analysis*, *design*, *implementation*, *testing* and *maintenance* phases using top-down decomposition. Each phase begins when the preceding phase has finished, and so the process can be seen as a linear execution. This life cycle is known as the linear sequential model, or *waterfall* model [Pre97].

Object oriented methodology does not change the abstract definition of the different phases. However, how they are carried out, and the products of each phase are different. The object oriented life-cycle is characterised by being iterative and incremental. At each iteration, object oriented analysis (OOA), object oriented design (OOD), object oriented implementation or programming (OOP) and object oriented testing are carried out increasing the part of the problem that is covered.

Of special interest for this thesis are OOA, OOD and OOP. OOA proposes classes, relations between classes, and the attributes and methods. The objective is to discover and understand the problem domain by modelling with objects and classes. OOD refines the classes by giving declarations to the classes and specification to the functionality of each method. At the same time, the model created by OOA is refined, adapting it to the restrictions of the application. The objective

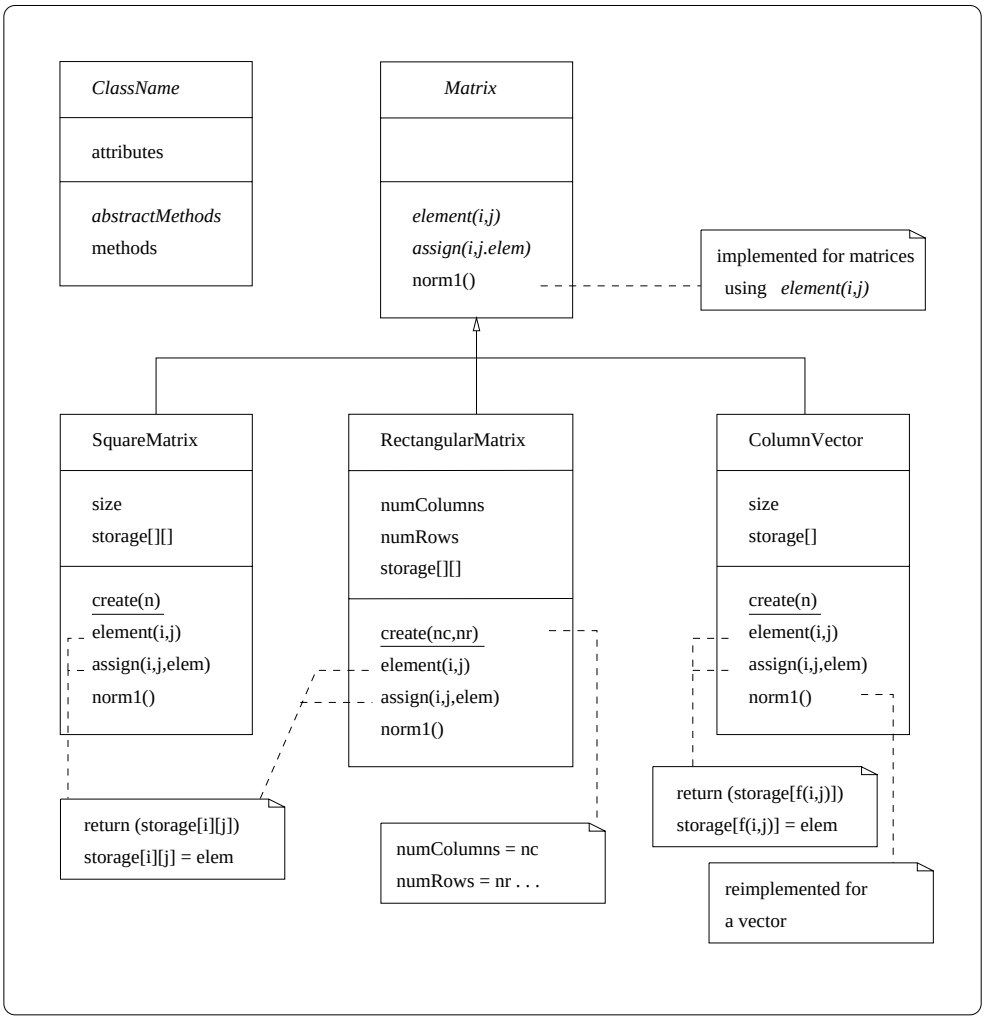


Figure 3.5: UML class diagram of a naïve abstract class `Matrix`.

is to plan how the model is going to be implemented. Finally, OOP is the implementation of the object oriented design in a given programming language. Ideally, the implementation should be made in an object oriented language; otherwise, the developers are forced to emulate the object oriented concepts. Guidelines for implementing object oriented models in non object oriented languages, such as Fortran 77 or C, are described by Meyer ([Mey97] Chapters 33 and 34) or by Decyk *et al.* ([DNS97a], [DNS97b], [DNS98]).

The division between OOA and OOD phases is fuzzy, although the focus and the products of both phases are clear. The analysis phase focuses on modelling the problem by proposing candidate classes and relations between the classes, evaluating them and rejecting the unsuitable proposals. Heuristics to find candidate classes are collected by Booch ([Boo94] Chapter 4) and Meyer ([Mey97] Chapter 22). Both authors identify as a source of candidate classes tangible things, roles, events, records of interactions, etc., from the problem domain ([SM88], [Ara89]). Also, both authors present a method based on studying a requirements document. The nouns and verbs expressing actions over them that are repeatedly used in this document become candidate classes and candidate methods [Abb83]. However, due to the complexity of natural language this approach has a limited success.

Booch and Meyer strongly disagree about the *use case* analysis formalised by Jacobson [JCJO92]. Use case analysis describes different scenarios, which are user-initiated transactions with the software. The scenarios represent the functions of the software. The analysis then takes each scenario, one-by-one, identifying possible classes and relations. In Booch's opinion, use case analysis provides an organised framework to discover the functionality required by an application and, from that, a good guide to follow. In Meyer's opinion, use case analysis is influenced by the users' vision about what the application has to do. This might lead non-expert object oriented developers to an algorithmic decomposition instead of an object oriented decomposition.

The OOD phase brings different requirements to the development process. Concurrency and synchronisation, mapping of the software onto the hardware (networks, modems, processors, etc.), and division of the object oriented model into packages, grouping related classes, are aspects that might be included during this phase [Kru95].

Following the above process, the OOA and part of the OOD for numerical

linear algebra is carried out in Section 3.2. The different proposed classes and relations are used to classify current object oriented numerical libraries. Section 4.1 refines the object oriented model proposed in this chapter to accommodate the restrictions associated with the implementation programming language (Java).

3.1.4 Some Tips

The "rules" given in the literature for deciding what are the relations between classes, can be considered more as heuristics; they always end with examples of "exceptions". *Design patterns* are class structures which model problems that repeatedly appear in almost every development of software. The definition of design patterns, and a collection of them is described by Gamma *et al.* [GHJV95]. Design patterns can be considered as the heuristics extracted from the experience of expert object oriented developers. Each design pattern describes the characteristics of a repeatedly faced problem for which an "elegant" and tested solution is known. Obviously, the description of the problem and solution are in abstract terms, but real examples of the successful application are presented.

Two design patterns, the *bridge* ([GHJV95] pages 151–162) and the *iterator* ([GHJV95] pages 257–272) patterns, and a comparison between generic classes and inheritance are the "tips" suggested. These are used in the object oriented analysis and design described in the next section.

Bridge Pattern

Normally, when deciding what is the relation between classes, the client relation does not offer problems. However, it is not trivial to decide when the inheritance relation should be applied. The client relation can be semantically interpreted as a "has-a"; class `A` is client of `B` means that `A` has-a `B`. Similarly, the inheritance can be semantically interpreted as an "is-a"; class `B` inherits from class `A` means that `B` is-a `A`. For example, the problem defined by the phrase – "a person has a car" – does not offer any doubt about a client relation between a class `Car` and a class `Person`. The models `Car` is-a `Person` or `Person` is-a `Car` do not make sense. However, when adding a new phrase – "a black car is a car" – it is suggested that there are two classes: a class `Car` and a class `BlackCar` that inherits from `Car`. It is also possible to model the phrase as an object class `Car` has-an object of class `Color` and its state indicates is black. The decision depends on the problem

domain and, without extra information, both models are valid.

In the case of linear algebra, the situation described in the last paragraph is repeated. The phrase to model is – "a matrix with some properties is a matrix". This phrase describing the problem suggests that class `MatrixWithProperties` is-a `Matrix`. It is also possible to model the phrase as class `Matrix` has-a `Property`. The decision and the arguments are presented in Section 3.2.1, although the bridge pattern, used to make the decision, is presented in the following paragraph.

The bridge pattern represents a problem where an abstraction can have different possibilities, only one possibility at each time, and during execution the possibility can change. The possibilities provide the same set of methods, but each possibility implements them differently. Figure 3.6 presents the class diagram of the proposed solution. A new abstract class named `Possibility` has been created where the common attributes and methods among the different possibilities is declared, but not implemented. Each possibility (`Possibility1`, ..., `PossibilityK`) is a class which inherits from the new abstract class `Possibility` and provides implementation for the inherited abstract methods. The abstraction becomes a class called `Abstraction` that is defined to be a client of the abstract class `Possibility`. This enables the client relation to be polymorphic. Figure 3.7 presents an example where the abstraction is a figure and the possibilities are circles and triangles.

Iterator Pattern

The iterator pattern presents a solution to traverse different kinds of containers with a unique interface. The iterator described by Gamma *et al.* [GHJV95] traverses and accesses the elements in sequential order and is presented in Figure 3.8. The methods `next` and `currentElement` advance one position in the container and return the current element, respectively. The method `begin` sets the iterator to the first position of the container, and the method `isFinished` tests if there are any more elements to be accessed in the container.

The Standard Template Library classifies the iterators, among others, into sequential and random access [LS95]. A random access iterator adds to class `Iterator` a new method `getElement` that returns the element in the position passed as a parameter.

The iterator pattern is used as a way to access the elements of matrices, thus enables linear algebra developers to adopt a different approach to the way that

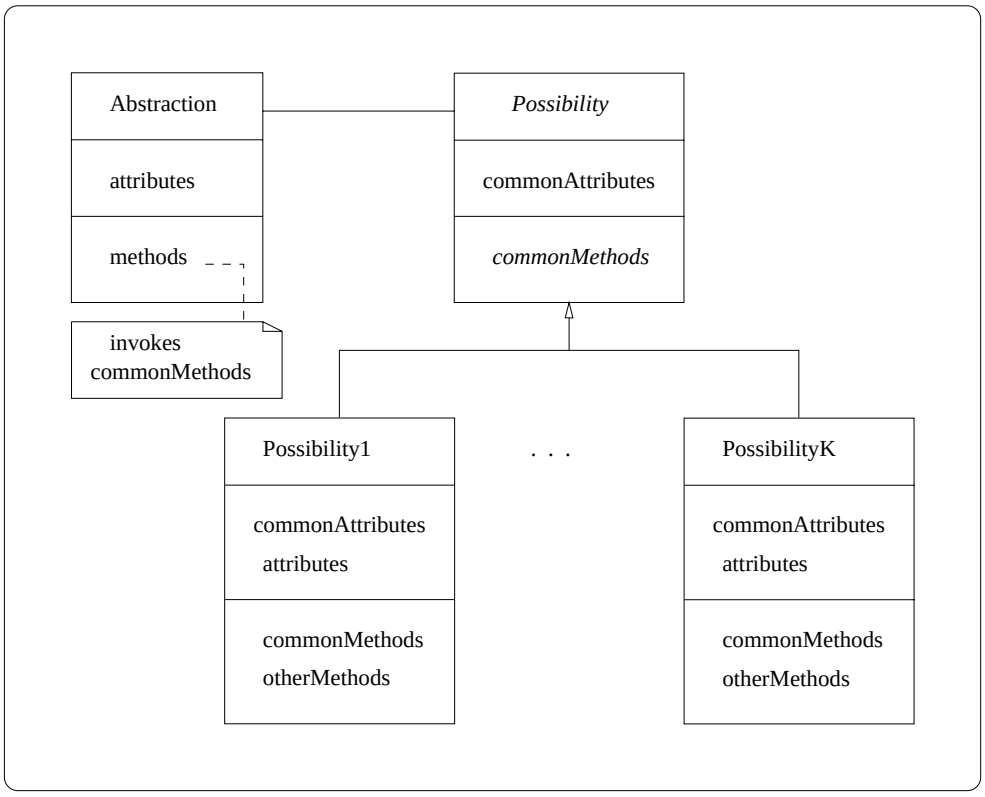


Figure 3.6: Class diagram of the bridge pattern.

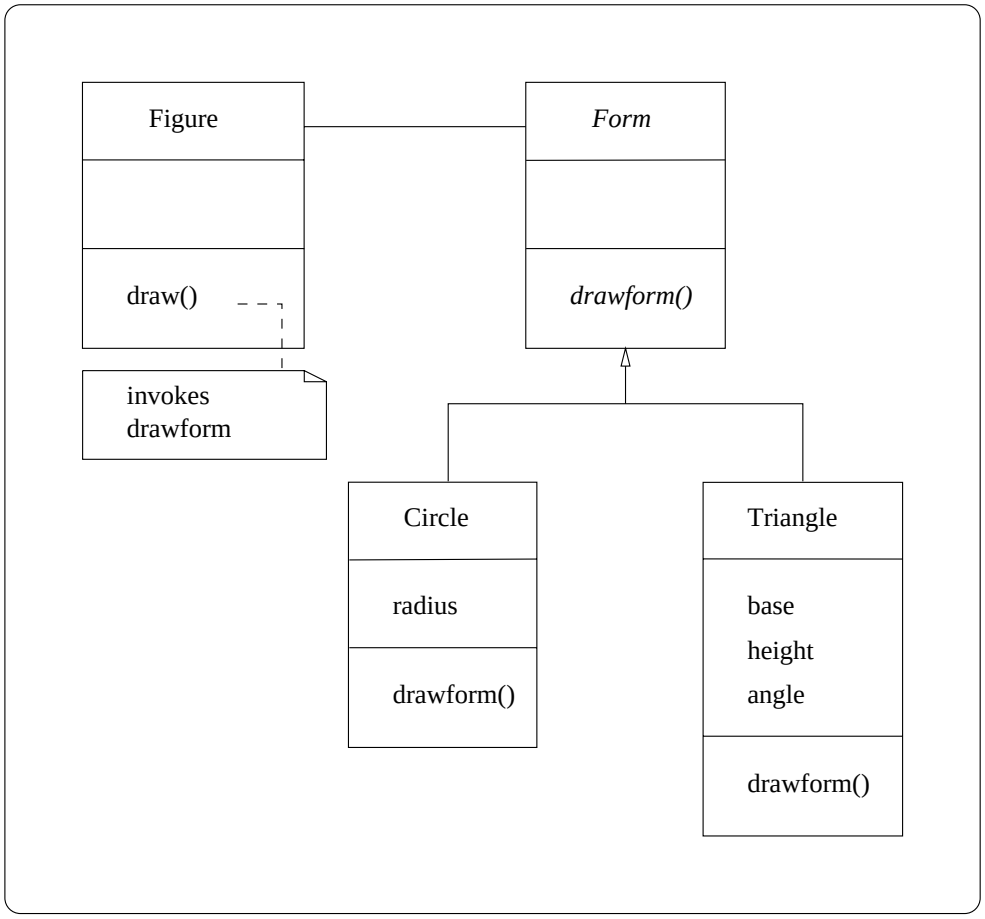


Figure 3.7: Class diagram of an application of the bridge pattern.

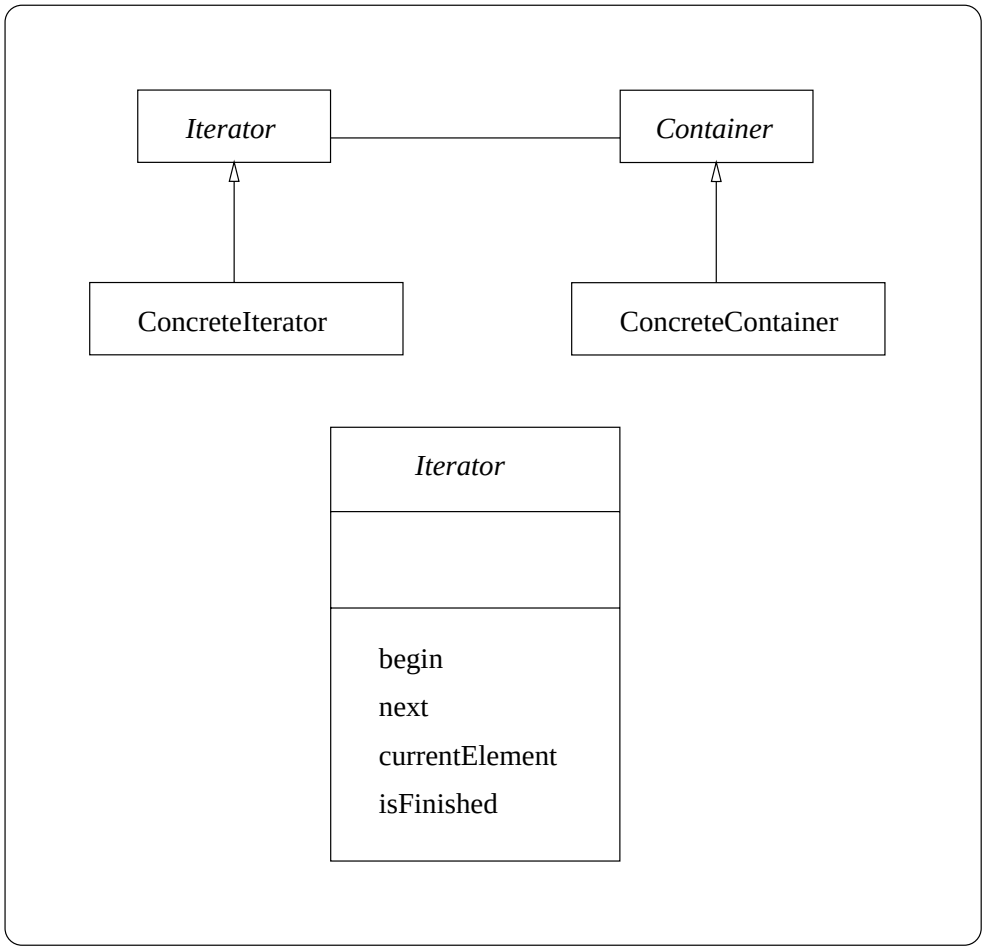


Figure 3.8: Class diagram of the iterator pattern.

matrix calculations can be implemented.

Simulation of Generic Classes

Generic classes are not supported by every object oriented language. In their absence, the developers may have to code, by hand, each of the different possible derived classes from the generic class. The number of classes that have to be written is linearly proportional to the number of different valid parameters of the generic class. Figure 3.9 presents a class diagram for a generic class `GenericMatrix` whose parameter is the class of the elements.

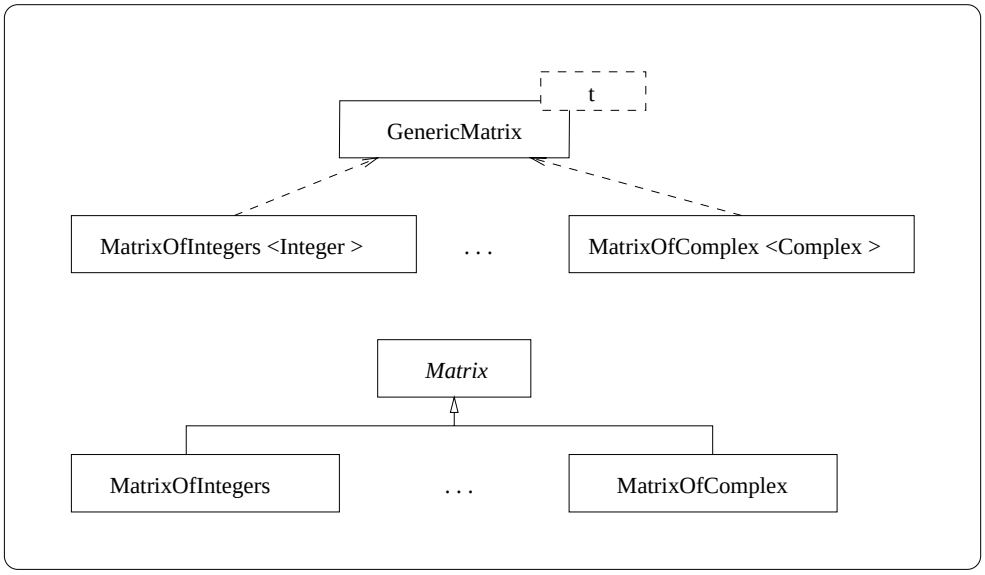


Figure 3.9: Class diagram emulating generic classes by hand code.

Alternatively, developers can simulate a generic class using a class with a polymorphic client relation. Each of the different valid parameters of generic class is made to inherit from a new abstract class. The class that simulates the generic class is a client of the new abstract class. Figure 3.10 presents the pertinent class diagram using the generic class `GenericMatrix`. Class `SimulatedGenericMatrix` simulates the class `GenericMatrix` by being a client of the abstract class `Number`.

`GenericMatrix` and `SimulatedGenericMatrix` class structures represent polymorphism. In the case of class `GenericMatrix`, the polymorphism is resolved at compile-time since its sub-classes resolve the polymorphism when choosing one class for the elements. In the other case, the polymorphism is resolved at run-time since every object of class `Number` or sub-classes might be assigned at any

time. The generic class creates an object matrix that only can store one class of objects. However, the class `Matrix` creates an object matrix that can store any object of the hierarchy `Number` (bridge pattern). Nevertheless, it is also possible that only objects of one class are stored and thus simulate the generic class.

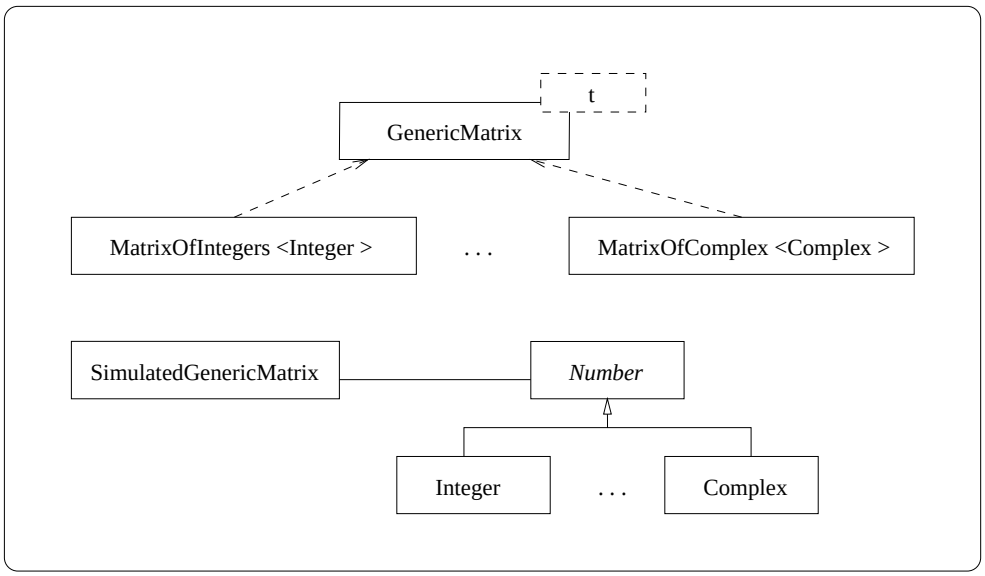


Figure 3.10: Class diagram of generic classes simulated by inheritance and client relation.

Developers simulating generic classes with polymorphism find, unless the compiler implements an aggressive algorithm, that generic classes are faster. In the case of generic classes, the dynamic binding mechanism is not necessary because the polymorphism has been resolved at compile-time. However, the emulation of generic classes needs the dynamic binding mechanism. In this case, an aggressive compiler would be able to resolve the polymorphism only if it can prove that only one class of objects is assigned.

3.2 Analysis and Design of OoLaLa

Object oriented analysis and design is the part of the software development process where an object oriented model of the problem to be solved is created. Key abstractions (classes) from the problem domain are identified and relations (client or inheritance) between these classes are proposed. The nature of this process is iterative and incremental; different models are created and evaluated against

parts of the problem domain until the parts are properly described, and then a new iteration begins, including new parts of the problem domain.

This section is dedicated to a review of different designs of object oriented linear algebra libraries. In order to present clear diagrams and discussions, the following aspects have been omitted: the class of the elements of matrices; methods that create objects, and methods that query the state (attributes).

An initial step is carried out modelling matrices, matrix properties and storage formats simply including the access methods of matrices (Section 3.2.1). This initial step provides the basic design which is extended, firstly, to allow sections of matrices to be matrices and matrices formed by merging other matrices (Section 3.2.2). Secondly, the iterator pattern is modified for the purpose of traversing linear algebra matrices (Section 3.2.3). Finally, basic matrix operations and matrix equations solved with direct or iterative algorithms are given a representation (Section 3.2.4). At each stage, different solutions are proposed. These are used to classify some object oriented linear algebra libraries (see Table 3.1). When selecting a solution, two user groups are kept in mind: numerical linear algebra experts and non-experts. The obvious differences between these two groups force the library to be as simple as possible for non-expert users, but also to provide as many tuning details as possible for expert users. However, these tuning details do not reveal how they are implemented.

3.2.1 Initial Analysis

A *matrix* is a two-dimensional container of numbers. The *dimensions* of a matrix are the number of rows (`numRows`) and number of columns (`numColumns`). The basic operations are to obtain an *element* of the matrix, a_{ij}, and to *assign* a value to an element of the matrix, e.g. $a_{ij} \leftarrow 32$. An element is determined by its (unique) position; number of row i and column j. Given two integers i and j, they determine an element if both are greater or equal than 1 and if they are less or equal than `numRows` and `numColumns`, respectively. In other words, every matrix has two methods to access the elements: `assign` and `element`. The `element` method needs two integers, i and j, and returns the element in the i^{th} row and j^{th} column, whereas `assign` needs the same two integers and a number to assign to the element in the i^{th} row and j^{th} column. Figure 3.11 presents a `Matrix` class according to the above description.

Library	References
LAPACK++	[DPW93a], [DPW93b], [DPW96], [LAP]
SparseLib++ and IML++	[DLN+94], [PRL96], [DLPR96], [Spa], [IML]
Paladin	[GJ95], [GJP96]
JLAPACK	[BC98], [BC99], [JLA]
OwlPack	[BKP98], [BK99b], [BK99a], [Owl]
MTL and ITL	[SL98b], [SL98c], [SLL99], [SL98a], [SL99], [MTL], [ITL]
PMLP	[BBV+99], [BPB+99], [PML]
Diffpack	[BL97], [Dif]
ISIS++	[ACMW99], [ISI]
Sparspak++ or Sparspak90	[GL99]
Oblio and Spindle	[DKP99], [DKP98], [KP98]
JAMA	[JAMb]
Jampack	[Ste99], [Jama]
BPKIT	[CH96], [CH98], [BPK]

Table 3.1: Object oriented linear algebra libraries.

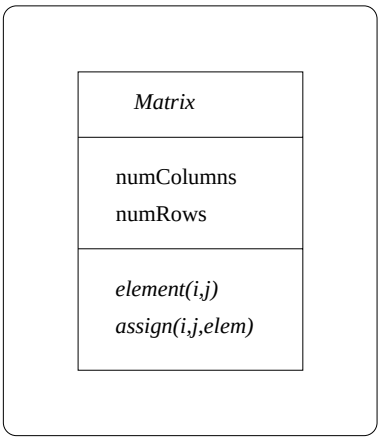


Figure 3.11: A simple `Matrix` class.

With this description of a matrix as a starting point, the discussion is organised around a set of different proposals. Each proposal differs in the organisation or relations between matrix, matrix properties and storage formats. For each proposal two class diagrams are presented. The first class diagram presents the general structure (generalised class diagram) without using real properties or storage formats. The second class diagram applies the generalised structure to dense, banded, symmetric, symmetric banded and symmetric positive definite matrix properties, and to dense and band storage formats (concrete class diagram).

Proposals

The first proposal, `Matrix` version 1 (see Figures 3.12 and 3.13), is based on the inheritance relation. The combinations of matrix properties and storage formats are considered to be sub-classes of `Matrix`. The class `Matrix` is on the first level of the inheritance hierarchy. On the second level, the `Matrix` class has been specialised by the matrix properties; a band matrix is always a matrix. The third level specialises the matrix properties by combining the properties of the second level and thus creating properties such as symmetric banded. The fourth level specialises the matrix properties by giving them a storage format. Only the fourth level classes are not abstract classes.

The second organisation, `Matrix` version 2 (see Figures 3.14 and 3.15), introduces the client relation between classes. A new abstract class called `StorageFormat` is created and every storage format inherits from it. The same two methods, `element` and `assign`, are included for the `StorageFormat` class, thereby creating a unified interface for all the storage formats. The class `Matrix` has a client relation with the class `StorageFormat`. The matrix properties classes inherit from the class `Matrix`, as in `Matrix` version 1, but they are not abstract classes any more.

The third organisation, `Matrix` version 3 (see Figures 3.16 and 3.17), introduces a new abstract class called `Property`. The matrix properties that can be represented in different storage formats inherit from `Property` while the other properties are attributes of `Property`. The class `Matrix` has a client relation with `Property`, which also has a client relation with `StorageFormat`.

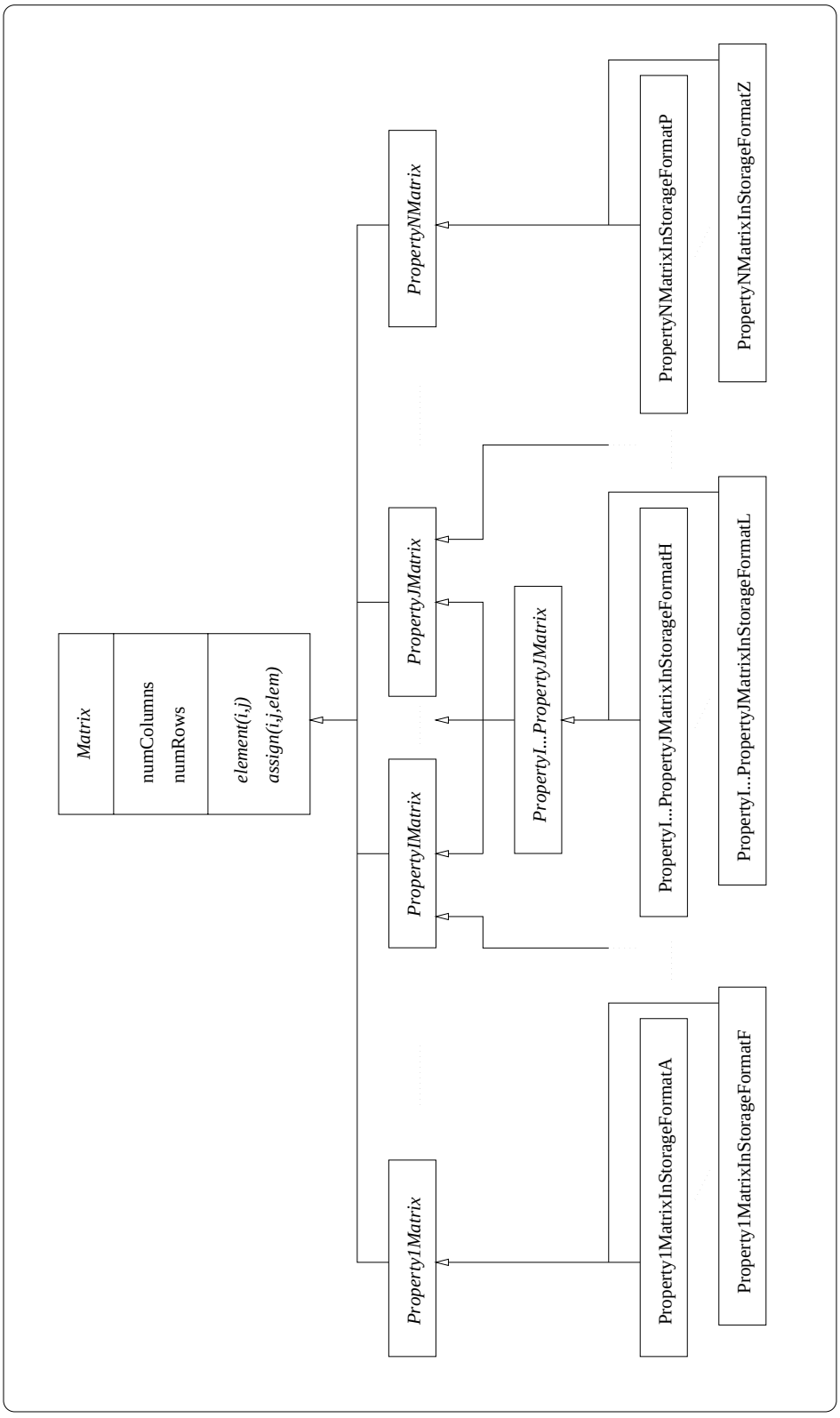


Figure 3.12: Generalised class diagram of `Matrix` version 1.

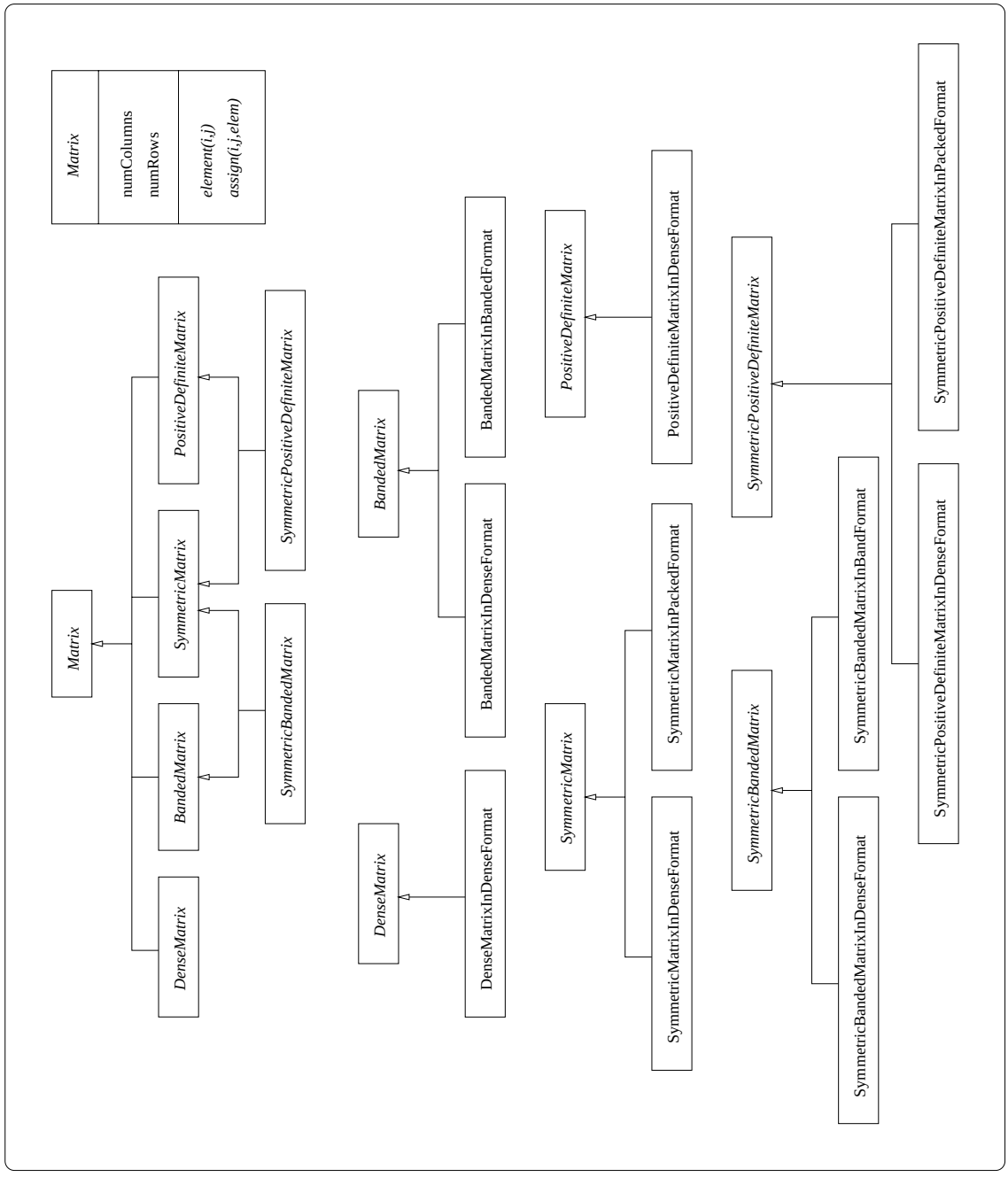


Figure 3.13: Concrete class diagram of `Matrix` version 1.

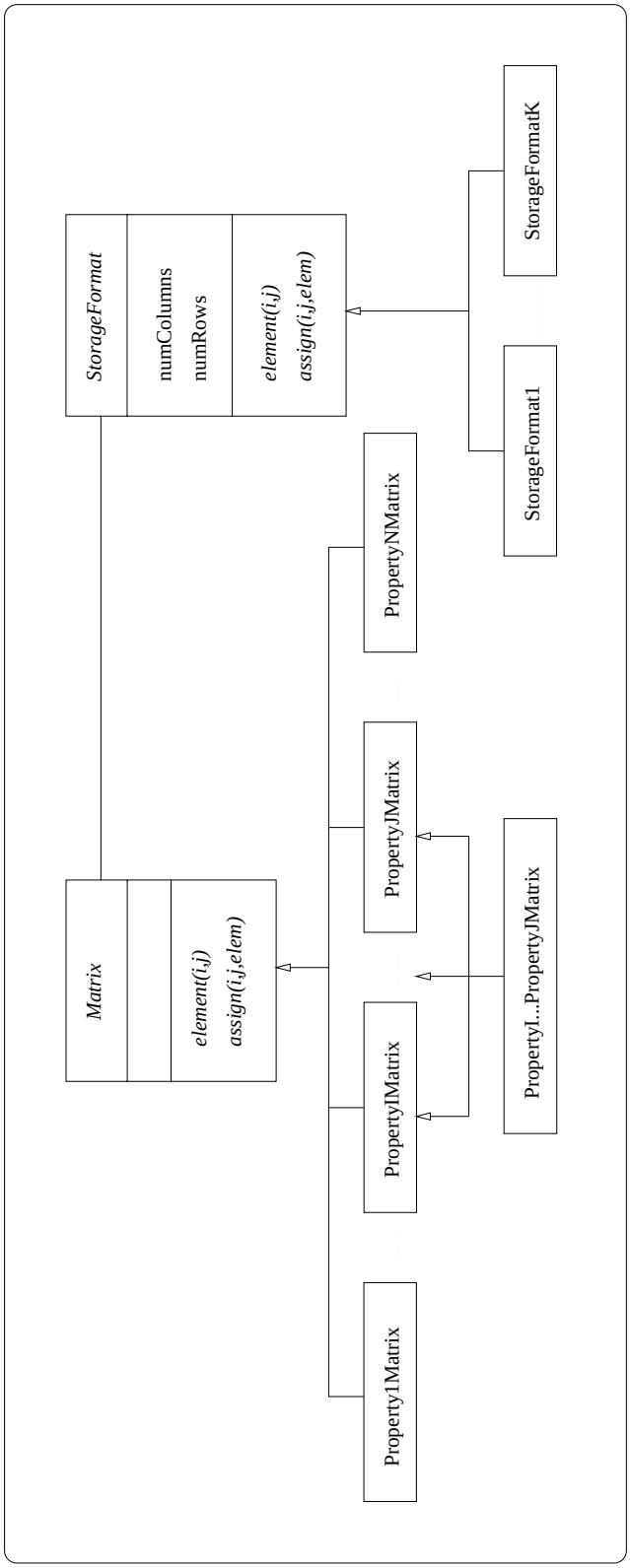


Figure 3.14: Generalised class diagram of `Matrix` version 2.

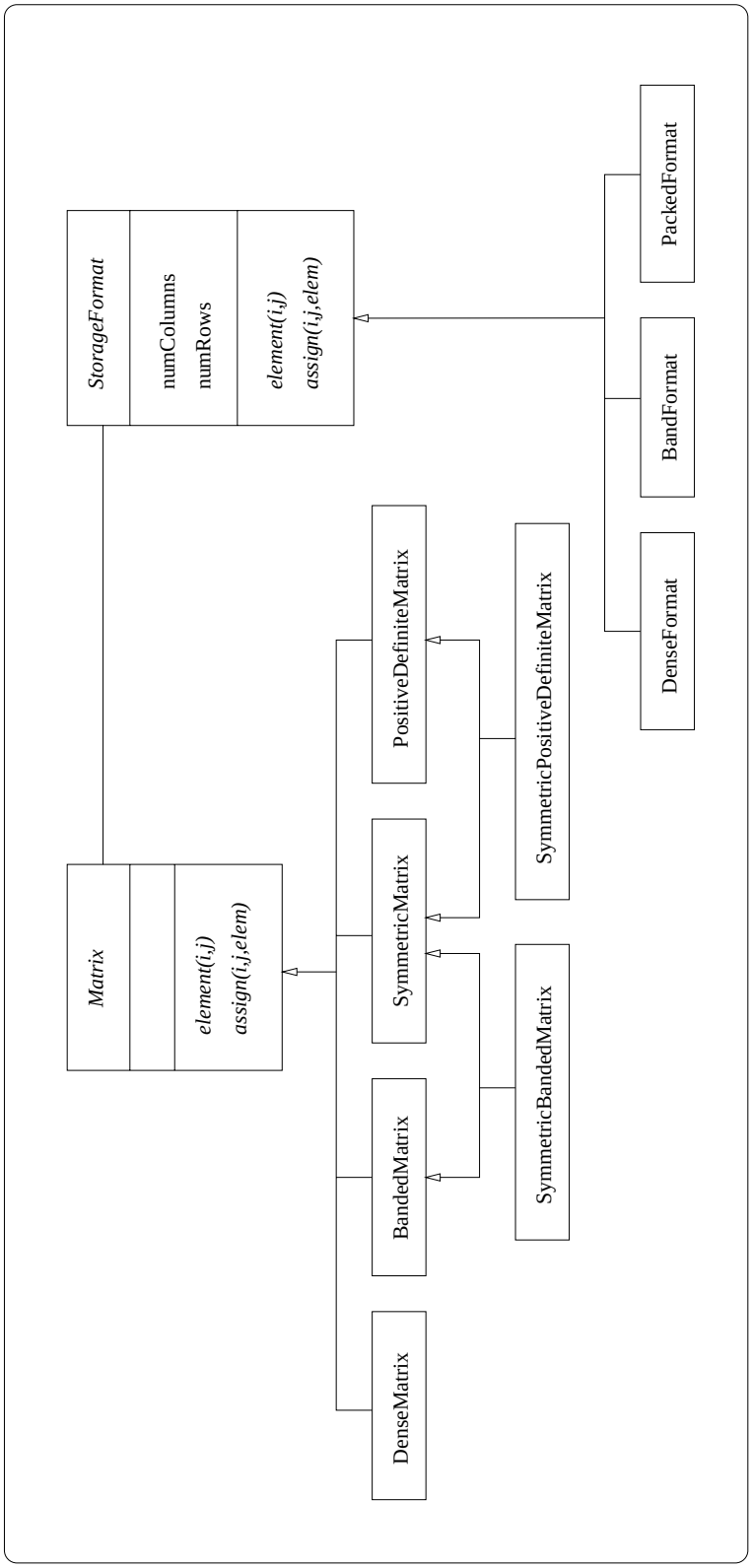


Figure 3.15: Concrete class diagram of `Matrix` version 2.

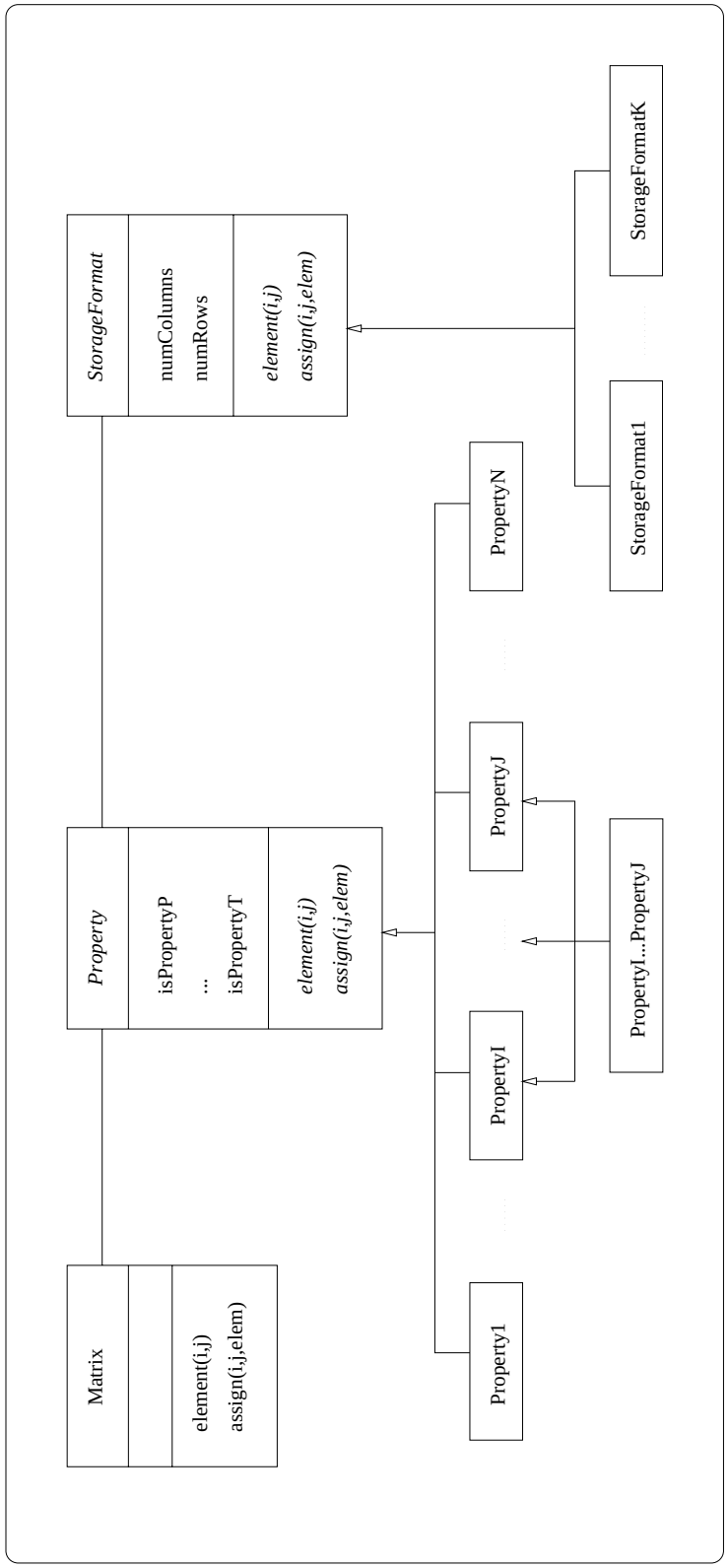


Figure 3.16: Generalised class diagram of `Matrix` version 3.

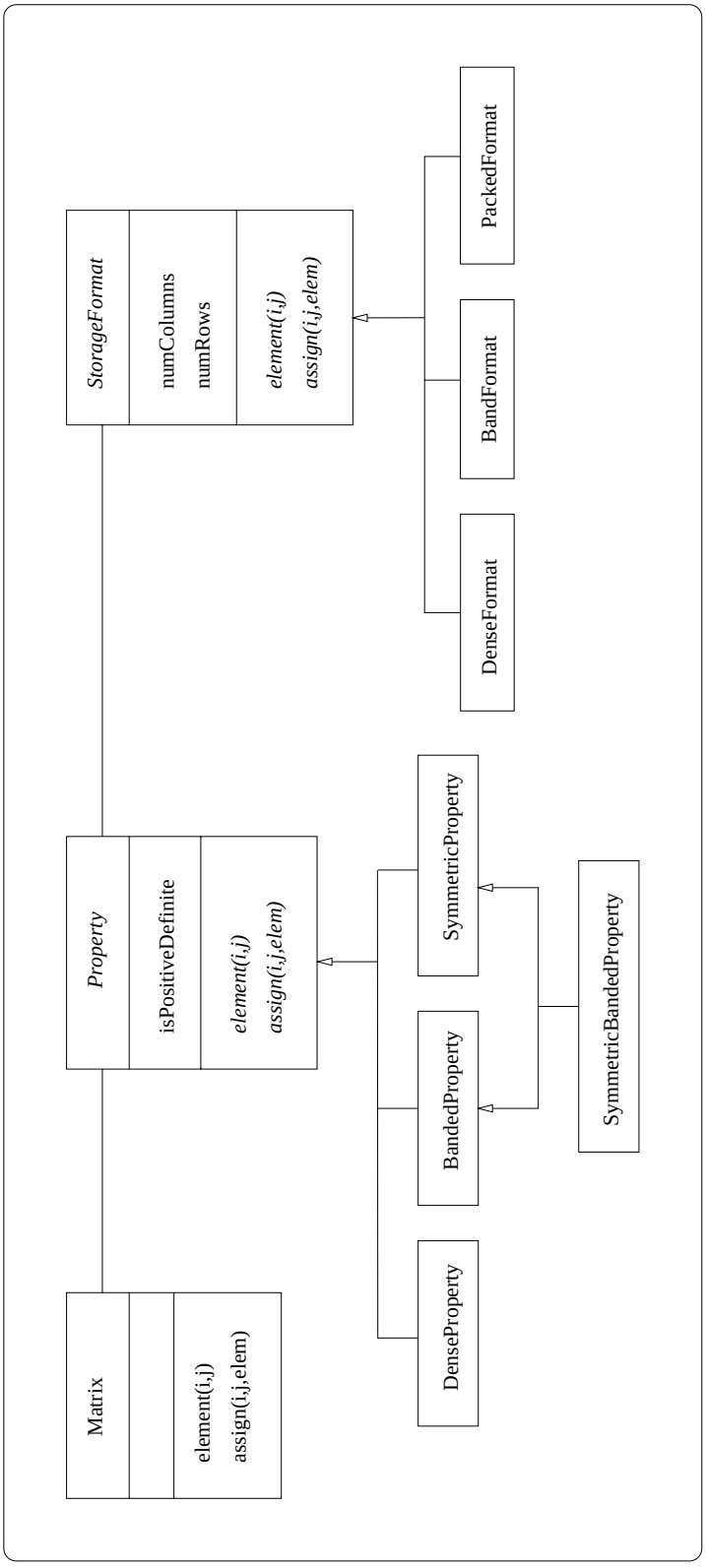


Figure 3.17: Concrete class diagram of `Matrix` version 3.

Discussion

In `Matrix` version 1, the classes at the bottom of the hierarchy can be seen as a possible combination of matrix properties and a storage format. Comparing these classes with the BLAS naming scheme, described in Section 2.4.1, for each two letters that represent matrix properties and a storage format (e.g. GE dense matrix in dense format or TP triangular matrix in pack format), a class is created.

LAPACK++, SparseLib++, Paladin, OwlPack, Diffpack, ISIS++, Spindle and Oblio, Jampack libraries (Table 3.1) are examples of `Matrix` version 1.

Since an object of any of the sub-classes of `Matrix` encapsulates the storage format, the number of rows and columns and the properties, a method `multiply`, with parameters of class `Matrix` can substitute for the BLAS subroutines `XGEMM`, `XGBMM`, etc. An implementation strategy for the method is to test the properties of the matrices and storage format and then decide which of the BLAS subroutines to call. The benefit for the user is that only one method, whenever possible, is offered for a matrix calculation. Section 3.2.4 returns to this point in more detail.

The benefit for the developer of the library is that a second implementation strategy is to use the unified access interface to every class in order to implement the methods. Hence, the number of implementations is reduced since the interface offers a way of accessing matrices that is independent of storage format. Figure 3.18 presents a naïve implementation of the method `element` for `DenseMatrixInDenseFormat`, `BandedMatrixInDenseFormat`, and `BandedMatrixInBandFormat`. Each implementation is adapted to the specific properties and storage format so that the correct element is returned. In a similar way, the method `assign` can be implemented and thus the unified access interface of every class is completed.

`Matrix` version 1 has a problem related to the number of classes that have to be implemented. For each matrix property, a matrix can be represented in many storage formats; therefore the number of required classes is of the order of the number of matrix properties multiplied by the number of storage formats.

`Matrix` version 2 uses the client relationship, or more precisely the bridge pattern, in order to reduce the number of classes of `Matrix` version 1. The class diagram can be read as "a matrix, with whatever properties, has a storage format". The storage format can be any of those in the hierarchy and can vary at run-time. The effect is that all the classes on the fourth level of the hierarchy of `Matrix` version 1 (Figure 3.12) are eliminated, and new ones encapsulating the storage format appear. The abstract class `StorageFormat` has the same two

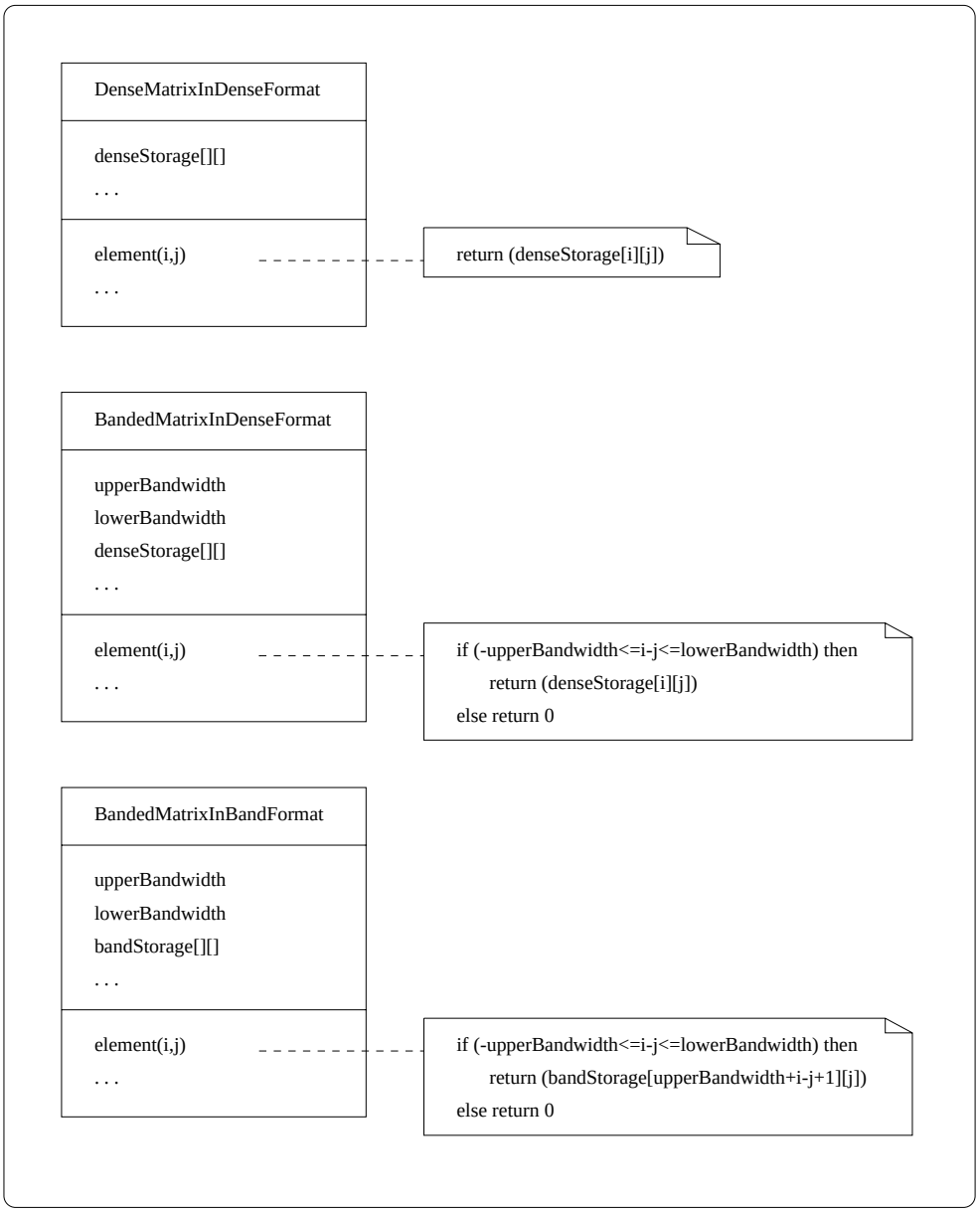


Figure 3.18: Implementation of the method `element` in `DenseMatrixInDenseFormat`, `BandedMatrixInBandFormat` and `BandedMatrixInDenseFormat` classes – `Matrix` version 1.

methods as `Matrix`; `element` and `assign`. This creates a unified access interface and, thus, the sub-classes of `Matrix` do not need to know in which storage format they are represented in order to access the storage format. Figure 3.19 presents naïve implementations of the method `element` for the `DenseFormat`, `BandFormat`, `DenseMatrix` and `BandedMatrix` classes. These implementation only access to the storage format when the element cannot be implied from the matrix property. Since storage formats are created omitting those elements that can be implied from the matrix properties, these implementations of `element` are independent of the storage format.

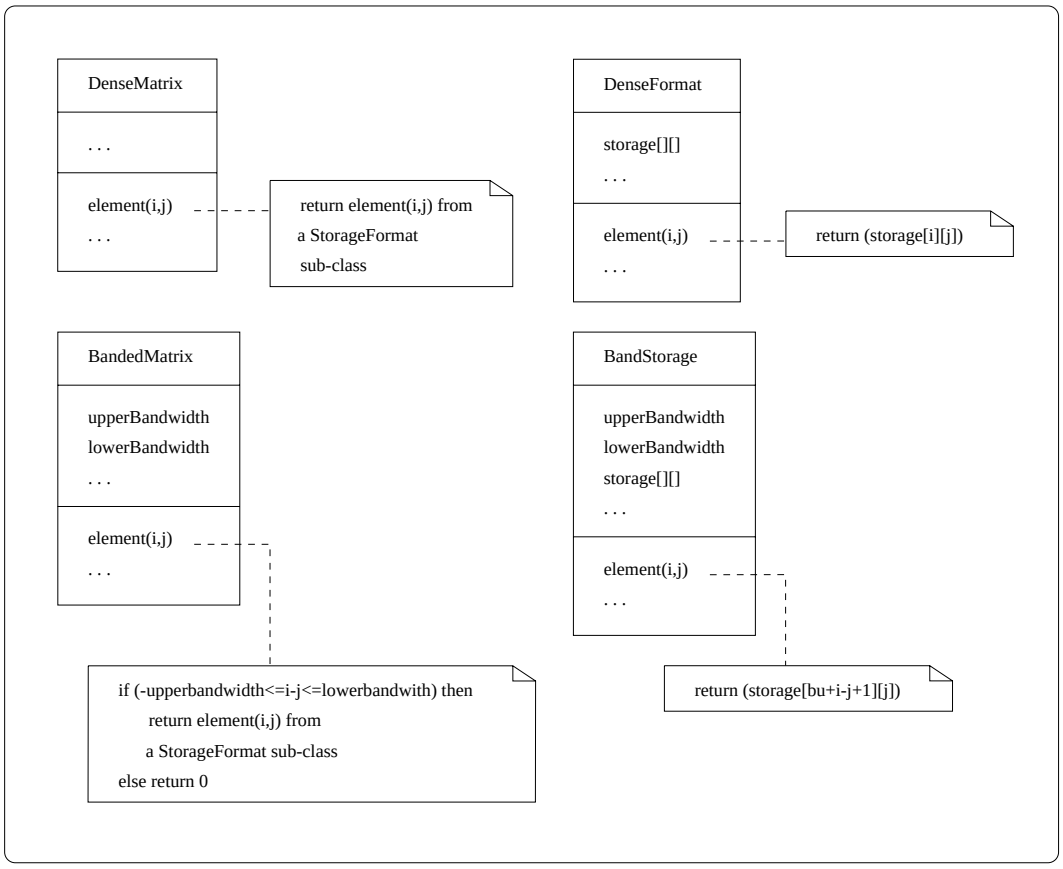


Figure 3.19: Naïve implementation of the method `element` in `DenseMatrix`, `BandedMatrix`, `DenseFormat` and `BandFormat` classes – `Matrix` version 2

Using generic programming, class `Matrix` can become a generic class with as its parameter a subclass of `StorageFormat`, and a similar model is obtained. PMLP and MTL (including other options as parameters, such as column-wise or row-wise arrays) libraries propose this variation to `Matrix` version 2.

PMLP and MTL face a common problem. Users do not need to know how a storage format is represented because it is encapsulated in the sub-classes of `StorageFormat`. The problem now is that users can create inadvisable combinations, such as a dense matrix stored in any sparse storage formats, or impossible combinations, such as a dense matrix stored in packed format.

Having identified this problem, and without a solution provided by any of the aforementioned libraries, an option is to hide the list of possible storage formats from users and rely on the library to decide which storage format to use. A second option is to allow users to define the storage format when an object of class `Matrix` is created and leave to the library to check the coherency between the matrix properties and the storage format specified.

The first option addresses the requirements of non-expert users of the library, who are relieved from having to know that a matrix can be represented in different storage formats and which one is advisable for their cases. However, this option is not satisfactory for expert users who wish to test different storage formats in order to determine the best for their needs (execution time, memory size, etc). The second option will satisfy the expert user as long as most storage formats are supported in the library. However, it has the disadvantage that adding a new storage format implies changes to the code that controls the coherence between storage format and matrix properties.

In the author's opinion, a combination of the two proposed options addresses the necessities of both user groups. The class organisation of `Matrix` version 2 does not need to be changed; the effect of accepting the two proposed options is reflected in the implementation of each subclass of `Matrix`.

The main change that the reader needs to understand is that the storage format, with any of the options for `Matrix` version 2, is a possible way to reduce execution time or memory requirements and not a restriction because the library does not support a combination of storage format for a determined matrix operation. This can be achieved because the implementations are able to use the unified access interface of `Matrix` inheritance hierarchy. Matrix calculations implemented using this interface are said to be implemented at *matrix abstraction level*. Nevertheless matrix operation can still be implemented at storage format abstraction level.

`Matrix` version 3 introduces the possibility that some matrix properties are not represented by classes. The positive definite property is a property that does

not give chances to represent a matrix in different storage formats; it is just a factor that influences the implementation of a matrix calculation. Furthermore, it can be combined with any other property, but the combinations do not change the advisable storage formats of the original properties. The positive definite property is represented as an attribute of `Property`, and is thus inherited by every sub-class producing all the combinations. The rule to apply in general to determine if a matrix property is represented as a class or as an attribute is whether or not the property enables a matrix to be represented in different storage formats.

The second modification introduced in version 3 is that class `Matrix` becomes a client of `Property` from which the different matrix properties inherit. The class `Property` follows the same unified access interface of `Matrix` and no changes are needed for the sub-classes representing matrix properties. The interface of class `Matrix` includes methods, `setProperty`, so that the properties of a matrix can be declared. The following example of a linear algebra calculation presents a situation that `Matrix` version 2 and version 1 both fail to model, and which motivates version 3. Suppose that $B \leftarrow AB$ is the desired linear algebra calculation, where A is a dense matrix and B is a banded matrix. Mathematically speaking, this calculation is correct as long as both A and B are square matrices of order n. In other words, a matrix calculation is correct as long as it conforms to its definition and the properties of the matrices do not interfere. However, using `Matrix` version 1 or version 2 this matrix calculation is not accepted or is performed incorrectly. A sensible program which uses version 2 creates an object `a` of class `DenseMatrix` and an object `b` of class `BandedMatrix`. After executing a method that assigns to `b` the product, two different problems may arise. The first problem is that, if the object `b` had-an object of class `BandFormat`, an exception should be raised informing that a dense matrix cannot be stored efficiently in band format; the solution to this problem is to allow the library to change the storage format. The second problem arises from the change of properties of `b`; although the library can change the storage format it is impossible for it to change `b` to be an object of class `DenseMatrix`, because `DenseMatrix` is not a sub-class of `BandedMatrix`. Consequently, `b` is an object of the class `BandedMatrix` that should be an object of class `DenseMatrix`. Access to elements outside the bandwidths would return zero, although the result is known to be different.

The characteristic of this example is that, during run-time, an object that

represents a matrix can vary its properties when it is operated upon. Using again the bridge pattern, the class `Matrix` has a client relation with `Property` under which the matrix property classes can be found. The class diagram of `Matrix` version 3 can be read as – "a given matrix can have different matrix properties and, in function of these properties, can be represented in different storage formats". The properties and storage formats are not fixed, this means that, when operated on, the properties and storage format of an object of class `Matrix` can be changed. The properties and storage format have to change in a way that the combination of both is advisable. The model created by `Matrix` version 3 allows to control these changes.

Through the different proposals, the functionality that the library provides has been increased. The interface has been adapted so that non-expert users can rely on the library to manage the properties and the storage formats for the matrices. The interface also offers expert users the possibility to create a matrix with a specific storage format supported. The library checks the coherency of the combinations determined by users and through calculations. The `StorageFormat` inheritance hierarchy unifies access to the different storage formats represented as its sub-classes. The `Property` inheritance hierarchy determines which elements are known owing to the properties. Otherwise, the storage format is accessed. The `Matrix` class is the user interface that encapsulates how properties and storage formats are implemented and enables the library to change them transparently for users.

None of the object oriented libraries reviewed in this section can be classified as `Matrix` version 3 (see Table 3.2); nor do they provide support for checking the coherency of matrix properties and storage formats. In order to provide this functionality, the library has to be able to propagate the properties of matrices from the operands to the results. A simple version of how to implement this new functionality is presented in Section 4.4. The `Matrix` version 3 and the functionality discussed above are the basis of a new library known as Object Oriented Linear Algebra LibrAry (OoLaLa). This design is refined in the next sections so as to enable users to use sections of matrices (rows, columns, sub-matrices) as if they were matrices, and to merge a set of matrices into one (Section 3.2.2). The second refinement includes the abstraction of *iterators* in order to traverse matrices and allowing a different abstraction level of implementation (Section 3.2.3). Finally, matrix calculations are included in the library (Section 3.2.4).

Library	Class Structure
LAPACK++	version 1
SparseLib++ and IML++	version 1
Paladin	version 1
JLAPACK	–
JAMA	–
Jampack	version 1
OwlPack	version 1
MTL and ITL	generic version 2
PMLP	generic version 2
Diffpack	version 1
ISIS++	version 1
Sparspak++ or Sparspak90	–
Oblio and Spindle	version 1
BPKIT	version 1

JLAPACK, JAMA and Sparspak++ or Sparspak90 do not offer enough information to be classified and "–" has been used to represent it.

Table 3.2: Class structure of various object oriented libraries.

3.2.2 Different Views of Matrices

Sometimes, applications need to work on sections of matrices as if they were matrices. For example, subroutines of LAPACK partition the matrices into blocks and work on these blocks independently. The transpose of a matrix can be treated as a section that is accessed by interchanging the indexes. An LU factorisation can store the L and U matrices in the matrix A, assuming that A is stored in dense format. This implementation of LU factorisation is called *in place factorisation*. The subsequent phase of solving the triangular systems with coefficient matrices U and L accesses only the upper triangular section or the lower triangular section. On other occasions, applications need to merge matrices to create a new matrix; for example, a block matrix can be created by merging its blocks.

Examples of matrix sections are a row or a column of an $m \times n$ matrix, which can be viewed as a row vector of size n or a column vector of size m, or three consecutive rows, which can be viewed as a $3 \times n$ matrix. A block lower triangular can be formed by merging its blocks and zero matrices. *View* is the term used to refer to either sections or merged matrices.

A simple solution for sections of matrices is to provide methods that create a new object of class `Matrix` with a corresponding new object of class `Property` and an advisable new object of class `StorageFormat`. The elements of the original matrix are copied into this new object of class `StorageFormat`. This solution does not modify the class structure of `Matrix` version 3. This is valid for applications that do not need to reflect in the original matrix the modifications made to the new section matrix. However, other applications need both matrices to reflect the modifications made to any of them. Hence, this solution becomes inefficient since applications need to copy back the elements in the original matrix (or section matrix) at the same time the section matrix is modified (or original matrix). A similar argument can be made for a matrix formed by merging other matrices.

In cases where the new section matrix and the original matrix need to keep a consistency (i.e. objects of class `Matrix` need to share an object of class `StorageFormat`) new classes have to be included. Among other solutions which share a common problem, Figure 3.20 presents a class diagram with one of these solutions. New abstract classes, called `Section` and `Merged`, are introduced. Their subclasses replicate those of the `Property` inheritance hierarchy and, thus, a view can also have properties. Since the position of an element of a view is based on the viewed matrices, this is reflected with `Section` and `Merged` being clients of `Matrix` and having as attributes information such as the row and column base or the list of merged matrices. The numbers in these client relations indicate that an object of class `Merged` merges at least two matrices. They also indicate that an object `Section` is a section of only one matrix. Their final indication is that an object of class `Matrix` can have as many sections, or be part of as many merged matrices, as wanted. The three hierarchies inherit from a new abstract class called `ViewOrProperty`. This solution is part of a family of solutions that has the major drawback that the `Property` inheritance hierarchy is triplicated.

Reviewing the role of the `Property` inheritance hierarchy, it is defined to determine whether an element is known independently of the way it is stored. In other words, the `Property` inheritance hierarchy is independent of elements being stored in sections of matrices or in sections of different matrices or in a storage format; its function is just to determine if an element is known. For example, when A is an upper triangular matrix the elements a_{ij} with $i > j$ are known to be zero elements independently of their storage format. Figure 3.21 presents a class diagram following this criterion. The classes `Section`, `StorageFormat` and `Merged`

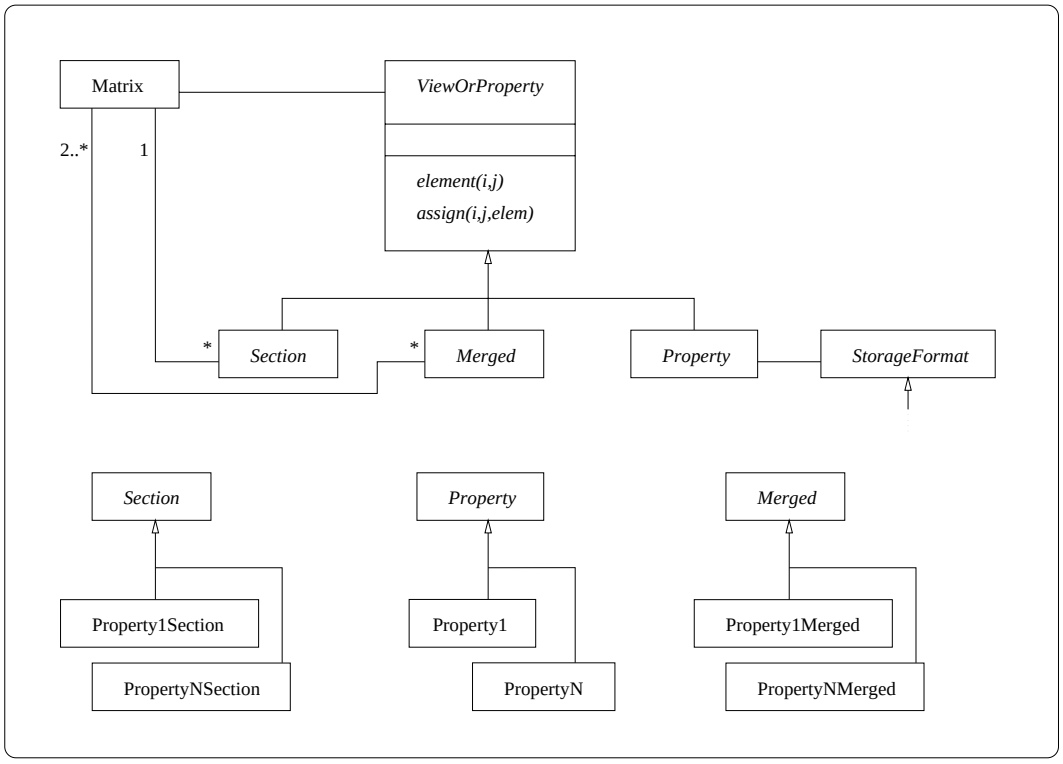


Figure 3.20: Class diagram of `Matrix` version 3 – first attempt to include different views of matrices.

are sub-classes of `ViewOrStorageFormat`. The class `Property` changes to have a client relation with `ViewOrStorageFormat` rather than with class `StorageFormat`. The classes `Section` and `Merged` keep their client relations with `Matrix`.

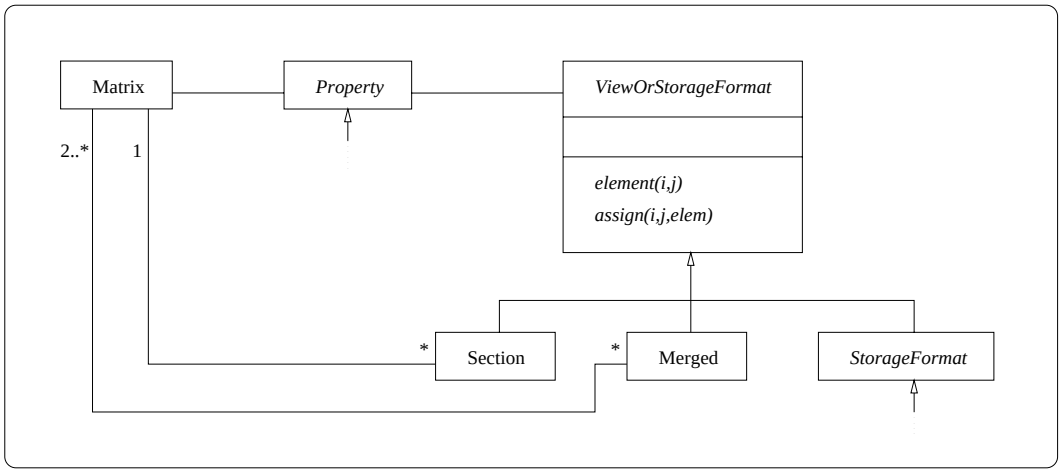


Figure 3.21: Class diagram of `Matrix` version 3 – second attempt to include different views of matrices.

Some current object oriented libraries provide views of matrices without replicating the elements. However, these libraries only allow the views to be dense matrices. Table 3.3 presents various object oriented libraries and how they support views of matrices.

3.2.3 Including Iterators

The iterator pattern, introduced in Section 3.1.4, is now extended to cover two-dimensional containers and, more specifically, linear algebra matrices. The iterator is redefined to traverse the elements of the matrices skipping those known to be zero.

Figure 3.22 presents the class `MatrixIterator`. The methods `setColumnWise` and `setRowWise` indicate how an object of class `MatrixIterator` traverses a matrix; column-wise or row-wise. The method `begin` places the object in the first column and first row. The method `beginAt` places the object in the position passed as a parameter. The class `MatrixIterator` considers a vector either as a column or as a row of the matrix and, thus, a matrix is traversed by passing through each vector of the matrix. The method `nextVector` increases an index of the current position and modifies the other index so that it points to the

Library	Sections	Merged Matrices
LAPACK++	(nce)	–
SparseLib++ and IML++	–	–
Paladin	(nce)	–
JLAPACK	(nce)	–
JAMA	(ce)	–
Jampack	(ce)	(ce)
OwlPack	–	–
MTL and ITL	(nce)	–
PMLP	–	–
Diffpack	–	–
ISIS++	–	–
Sparspak++ or Sparspak90	–	–
Oblio and Spindle	–	–
BPKIT	–	(nce)

The libraries that support views only allow them to be dense matrices or vectors. Only, BPKIT offers merged matrices whose blocks can be any kind of matrix. However, BPKIT's merged matrices are dense matrices. When a library does not support views it is represented as "–". When a library supports views copying elements it is represented as "(ce)". When a library supports views without copying elements it is represented as "(nce)".

Table 3.3: Support of views of matrices in various object oriented libraries.

first position. Which index is increased or modified depends on how the matrix is traversed. The method `isMatrixFinished` tests if there are more vectors to traverse in the matrix. A vector is traversed using the methods `nextElement` and `isVectorFinished`. The method `nextElement` searches for the next nonzero element within the vector while `isVectorFinished` tests if there are more nonzero elements in the vector. An element is accessed by the method `currentElement` that returns the current element and the row and column indexes.

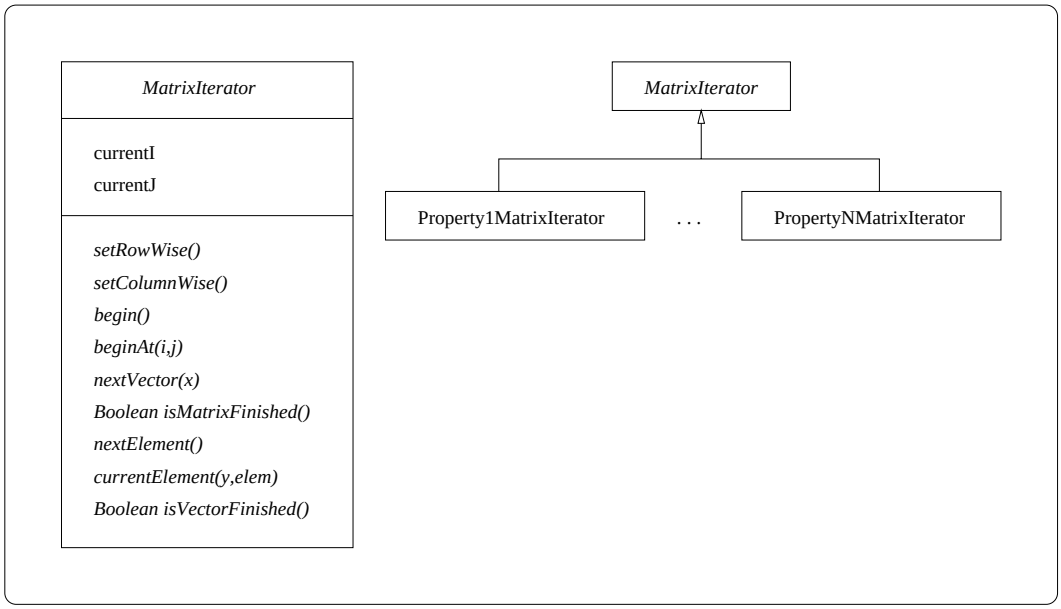


Figure 3.22: Class diagram of `MatrixIterator`.

Once the iterator pattern has been adapted to the requirements of linear algebra matrices, the next step is to integrate it with the class structure of OoLaLa. The `MatrixIterator` can be seen as an iterator for sequential access whereas `Matrix` and `Property` can be seen as iterators for direct access. The `MatrixIterator` interface can be integrated with the interface of `Matrix` and `Property` (see Figure 3.23) and, thus, the inheritance hierarchy of `Property` would not be replicated for `MatrixIterator`, if this class was included. The class structure is not modified with this integration.

Since an iterator traverses matrices skipping the zero elements, iterators constitute a new abstraction level at which matrix calculations can be implemented. Until now, an implementation of a matrix calculation was a set of nested loops defined in terms of explicit bounds which vary depending on the matrix properties.

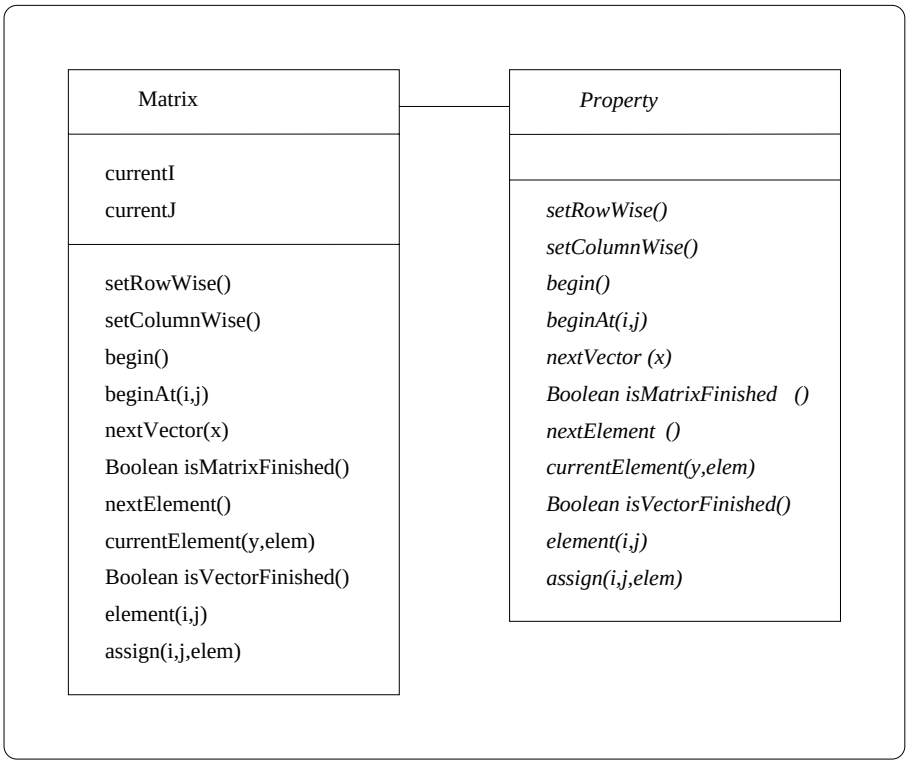


Figure 3.23: Class diagram of classes `Matrix` and `Property` including the methods of `MatrixIterator`.

In other words, an implementation of a matrix calculation traversed matrices by indicating explicitly which elements to access and by avoiding explicitly those elements that are known to be zero. On the other hand, an iterator expresses implicitly the elements to be accessed. An implementation of a matrix calculation using the interface of `MatrixIterator` implicitly changes the elements to be accessed when properties of the matrices are changed. This reduces the number of implementations of a matrix calculation.

MTL and PMLP use iterators, but with contradictory results. MTL has reported performance results on a Sun UltraSPARC for dense matrix-matrix multiplication and sparse matrix-vector multiplication, which are comparable to the highly optimised libraries ATLAS [WD98] and Sun Performance Library. On the other hand, PMLP declares [BBV+99]:

> "Iterators in PMLP provide a convenient means for users to iterate over elements in vectors and matrices, regardless of their internal data storage format. They also provide a storage format independent means for writing functions that access elements in objects using disparate storage formats. Since iterators are not an efficient mechanism for accessing elements in sparse matrices, much of the core functionality in PMLP is written using data access mechanisms specific to particular storage formats."

Note that no reference is given to justify their affirmation about the inefficiency of iterators or even a criterion to decide which functionality is written using what. The paper also does not reference MTL.

This thesis has introduced three implementation abstraction levels; storage format, matrix and iterator abstraction levels. The reader can either jump to Section 4.5 which compares the code of matrix calculations at the different abstraction levels for different matrix properties, or move to the next section that gives representations of matrix calculations in OoLaLa.

3.2.4 Including Matrix Calculations

The analysis has focused on modelling matrices, matrix properties and storage formats with respect to the access operations and matrix calculations. Access operations have been represented as methods (`assign` and `element`) of class `Matrix` while matrix calculations have been left without representations. The focus is now

directed towards the representation of matrix calculations in OoLaLa. From the design of other object oriented libraries, matrix calculations can be represented as follows:

(a) as methods of class `Matrix`, or an equivalent name in each library,

(b) as classes, sometimes grouped into inheritance hierarchies, with the parameters transformed into attributes and the operation performed through a method `execute`, or

(c) as methods of a *utility class*, where related operations are grouped together.

The following description makes the above representations concrete using the addition of matrices as an example. The first representation includes a method called `add` in class `Matrix`. This method takes as a parameter an object of class `Matrix` and returns a new object of class `Matrix`. This new object is the addition of the parameter object and the object in which the method `add` has been invoked.

The second representation creates a class `Add`. This class has three attributes of class `Matrix`, and, when the method `execute` is invoked, two of these attributes are added to form the third one. By using classes, related operations can be grouped into inheritance hierarchies, such as `MatrixOperation`; every matrix operation inherits from `MatrixOperation`.

Finally, the third representation includes a method called `add` in a utility class. A utility class is a class that, despite being fully defined and implemented, cannot be instantiated. A utility class is a similar concept to a library of subroutines. In this case, the utility class could be named `MatrixOperation`. The method `add` is declared to have three parameters of class `Matrix`; two inputs and one output. Figure 3.24 presents graphically each of the described representations.

The implementations of matrix calculations have different features. These features divide the calculations into basic matrix operations and solvers of matrix equations (direct and iterative). The remainder of this section examines the features of the calculations in order to decide which representation should be used. The objective is to provide a simple and consistent interface. At the same time, the interface has to satisfy the requirements of both user groups; experts and non-experts.

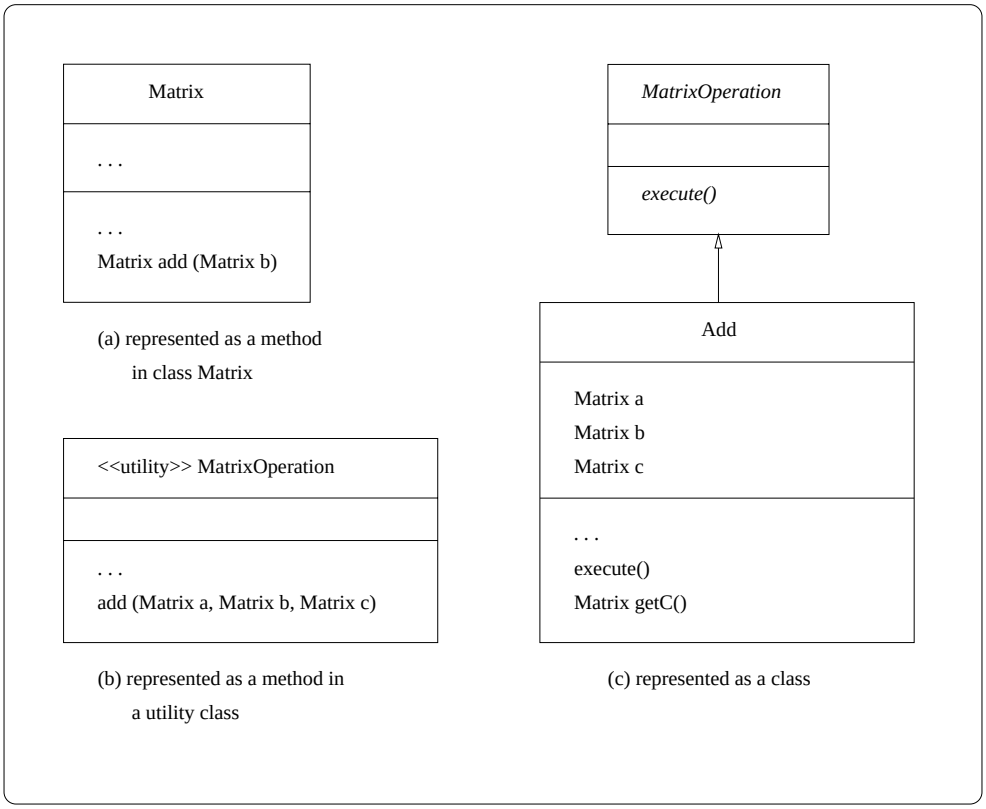


Figure 3.24: Different representations of matrix addition.

Basic Matrix Operations

The main feature of basic matrix operations is that, given the storage format and the matrix properties the implementation has already been decided. In other words, a set of "if-then" rules can be defined. These rules test the matrix properties and storage format of the operands and decide the corresponding implementation. The set of rules define a *rule based reasoning system*, or a *complete decision tree*.

Since an object of class `Matrix` encapsulates its matrix properties and its storage format, the reasoning system can be hidden behind the representation of each basic matrix operation. In this way, users have the impression that there is only one implementation of each basic matrix operation, although internally there may be multiple implementations. The interface is simplified in comparison with the BLAS because the number of visible subroutines for a matrix operation is reduced to only one visible representation. Moreover, the parameters of a basic matrix operation representation are no longer each detail of how the operands are stored, they are simply objects of class `Matrix`.

Due to the close relation between basic matrix operations and matrices, it is logical to represent them as methods of class `Matrix`. For example, the addition of matrices is an operation with domain and range matrices; it takes two matrices and produces a third matrix. On the other hand, to represent a basic matrix operation as a class is artificial, since such an operation is not an obvious abstraction from numerical linear algebra. Finally, a matrix operation can be represented as a method of a utility class. This utility class would resemble the BLAS and thereby users familiar with the BLAS would benefit. This benefit might be seen as an advantage over the first representation, but it is actually a signal expressing that this is not an object oriented form.

OoLaLa represents basic matrix operations as methods of class `Matrix`. In order to reduce execution time and memory requirements two syntaxes (or two methods) are discussed for a given basic matrix operation. For example, the matrix addition $C \leftarrow A + B$ can be represented by `c=a.add(b)` or `c.addInto(a,b)`, where `a`, `b` and `c` are objects of class `Matrix`. The method `Matrix add(Matrix b)` (`c=a.add(b)`) takes an object `b` as a parameter and performs the addition with the object in which `add` is invoked. This method returns a new object of class `Matrix`, i.e. also a new object `Property` and a new object `StorageFormat`. On the other hand, the method `void addInto(Matrix a, Matrix b)` performs

the same operation, but does not return anything. This method performs the addition in the object in which the method has been invoked. This enables the method to create new objects only if it is "strictly necessary". More details about how the storage formats and properties are managed in OoLaLa are described in Section 4.4.

Figure 3.25 presents the interface of class `Matrix` including a unary operation `norm1` ($||A||_1$) and two binary operations; `addInto` ($C \leftarrow A + B$) and `multiplyInto` ($C \leftarrow AB$). Table 3.4 offers a list of how matrix operations are represented in different object oriented linear algebra libraries.

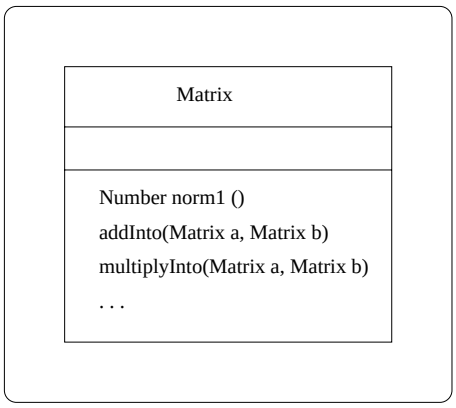


Figure 3.25: Class diagram of class `Matrix` including matrix operations as methods.

Solvers of Matrix Equations

In contrast with matrix operations, object oriented libraries disagree about how the operation of solving matrix equations should be represented. Some libraries represent these operations as methods (`solveLinearSystem`, `solveLeastSquares`, and `solveEigenproblem`) of class `Matrix` or as methods of a utility class. These methods have a parameter representing the solver as an object of class `LinearSystemSolver`, `LeastSquareSolver`, or `EigenProblemSolver`. Other libraries represent the matrix equation itself as a class (`LinearSystemEquation`, `LeastSquareEquation` or `EigenProblemEquation`) with attributes that are the matrices defining an equation, and the solver as another class (`LinearSystemSolver`, `LeastSquareSolver`, or `EigenProblemSolver`) with a client relation of class `LinearSystemEquation`, `LeastSquareEquation` or `EigenProblemEquation`. The

Library	Representation
LAPACK++	(a) and (b)
SparseLib++	(a) and (b)
IML++	(a)
Paladin	(a)
JLAPACK	(b)
JAMA	(a)
Jampack	(b) or (c)
OwlPack	(a)
MTL and ITL	(b)
PMLP	(a)
Diffpack	(a)
ISIS++	(a)
Sparspak++ or Sparspak90	–
Oblio and Spindle	–
BPKIT	(a)

Basic matrix operations represented as methods of a class `Matrix` are denoted with "(a)". Basic matrix operations represented as methods of a utility class are denoted with "(b)". Basic matrix operations represented as classes are denoted with "(c)". Basic matrix operations not supported by the library are denoted with "–". Note 1 – IML++ does not provide matrix operations, however it needs a library that provides them represented as (a). Note 2 – Jampack represents each matrix operation as a unique utility class and a method similar to `execute`. This method instead of using the attributes uses its parameters. Depends on the personal interpretation to decide between (b) or (c).

Table 3.4: Representation of basic matrix operations in various object oriented libraries.

operation of solving a matrix equation is represented by a method `solve` in the class representing the solvers. Finally, some other libraries have the same classes representing solvers but they do not have the classes representing the matrix equations.

Among these descriptions, the common point is that a solver is presented as a class. Each solver has different phases and for each phase different algorithms have been proposed by the numerical linear algebra community. The bridge pattern [1] can be applied again given the structure shown in Figure 3.26.

From an object oriented point of view, there is no argument against representing matrix equations as classes and the operation of solving a matrix equation as a method of these classes. Linear algebra defines matrix equations in terms of basic matrix operations. Hence, it is reasonable to represent them in a different way, as long as the model remains correct. However, from a consistency point of view, it can be argued that the operation of solving matrix equations should also be a method (`solveLinearSystem`, `solveLeastSquares`, and `solveEigenproblem`) of class `Matrix`. In order to keep the interface simple for non-expert users, these methods would have a solver as a parameter only if it is necessary. The solvers would be represented as a class inheriting from `MatrixEquationSolver`. OoLaLa represents the operation of solving a matrix equation in this way. Table 3.5 presents the representation of matrix equations and the operation of solving them in various object oriented libraries. Table 3.6 presents the matrix equations supported by these object oriented libraries.

Once it has been decided how the operation of solving a matrix equation is represented, the next requirement is to clarify when it is necessary to include a solver as a parameter, and to model the different kind of solvers: direct and iterative.

Direct Solvers of Matrix Equations

Direct solvers have different phases and characteristics depending on the properties of the coefficient matrix. In this discussion, structured matrices (dense, banded, block banded, block triangular) are distinct from sparse matrices.

A direct solver of a matrix equation with a structured coefficient matrix is composed of two phases. The first phase performs a factorisation of the coefficient

[1] The bridge pattern when applied to classes representing algorithms is known as the *strategy pattern*([GHJV95] pages 315–324).

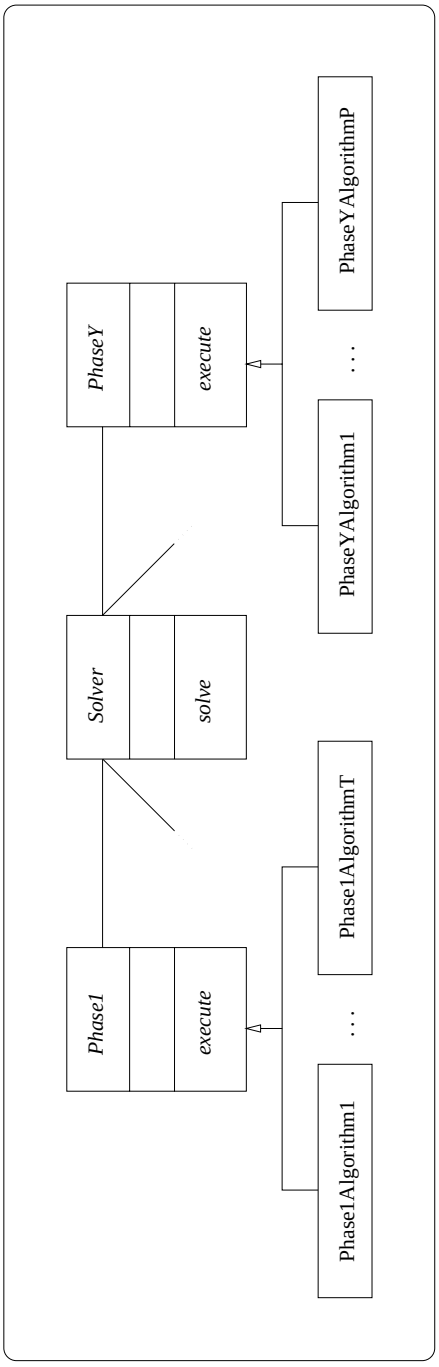


Figure 3.26: Class diagram of general `Solver` of matrix equations.

Library	Operation	Matrix Equation
LAPACK++	method in utility class	parameters of the method
SparseLib++ and IML++	method in utility class	parameters of the method
Paladin	method in `Matrix`	parameters of the method
JLAPACK	method in a utility class	parameters of the method
OwlPack	method in `Matrix`	parameters of the method
MTL and ITL	method in a utility class	parameters of the method
PMLP	method in `Solver`	attributes of `Solver`
Diffpack	method in `MatrixEquation`	class `MatrixEquation`
ISIS++	method in `MatrixEquation`	class `MatrixEquation`
Sparspak++ or Sparspak90	method in `Solver`	class `MatrixEquation`
Oblio and Spindle	method in `Solver`	attributes of `Solver`
JAMA	method in `Matrix` and `Solver`	parameters of the method or attributes of `Solver`
Jampack	method in utility class	parameters of the method
BPKIT	–	–

BPKIT provides block preconditioners and an interface to be used by iterative algorithms. However, BPKIT does not report how the iterative algorithms are represented.

Table 3.5: Representation of matrix equations and the operation of solving them in various object oriented libraries.

Library	Direct Solvers		Iterative Solvers
	Structured Matrix	Sparse Matrix	
LAPACK++	(a), (b) and (c)	–	–
SparseLib++ and IML++	–	–	(a)
Paladin	(a)	–	–
JLAPACK	(a)	–	–
OwlPack	–	–	–
MTL and ITL	(a)	–	(a)
PMLP	–	–	(a)
Diffpack	–	–	(a)
ISIS++	–	–	(a)
Sparspak++ or Sparspak90	–	(a)	–
Oblio and Spindle	–	(a)	–
JAMA	(a), (b) and (c)	–	–
Jampack	(a), (b) and (c)	–	–
BPKIT	–	–	see below

Systems of linear equations are represented as "(a)". Least square problems are represented as "(b)". Eigenproblems are represented as "(c)". Kinds of matrix equations that are not supported by the library are denoted by "–". BPKIT provides block preconditioners and an interfaces to be used by iterative algorithms. However, BPKIT does not report what iterative solvers are supported.

Table 3.6: Solvers of matrix equations provided by various object oriented libraries.

matrix, unless it is trivial and efficient to solve (e.g. diagonal matrix or triangular matrix). The second phase solves the matrix equation using the factorisation. According to the properties of the coefficient matrix and its storage format a factorisation and its specialised implementation can be selected. In other words, a set of "if-then" rules can be defined. These rules test the matrix properties and storage format of the operands and determine the corresponding implementation. This set of rules define another *rule based reasoning system*.

As with matrix operations, behind the methods `solveLinearSystem`, `solveLeastSquares`, and `solveEigenproblem` the existence of different factorisations and their specialised algorithms can be encapsulated. In this way, users have the impression that there is only one implementation, although internally there are multiple implementations.

In general, the factorisation phase can be characterised as pivoting or no-pivoting. This characteristic distinguishes between a factorisation that needs to check the stability or not. Hence, using method overloading, a method with different parameters but same name (`solveLinearSystem`, `solveLeastSquares`, and `solveEigenproblem`) is included. The parameters are the same, except for an object of class `MatrixEquationSolver` that will indicate the characteristic of pivoting or no-pivoting. Table 3.6 presents various object oriented libraries that provide direct solvers for structured matrix equations.

A direct solver of a linear system of equations with a sparse coefficient matrix has three different phases. The first phase produces a new ordering of the coefficient matrix in order to conserve the sparsity. The second phase factorises the re-ordered matrix, and then, the third phase solves the linear system.

The ordering phase can take into account the numerical values of the elements of a matrix and simulate a factorisation. The ordering algorithms that take into account the numerical values are called numerical ordering. Other ordering algorithms that take into account structure but not specific numerical values are called symbolic ordering. The factorisation phase after a numerical ordering does not perform pivoting since it has already been calculated. This factorisation phase knows exactly the fill-in elements and, therefore, the factorisation phase can use a static storage format. However, the factorisation phase after a symbolic ordering needs to perform pivoting, possibly creating an unknown number of new fill-in elements, and therefore a dynamic data structure is necessary.

Table 3.6 presents Spindle and Oblio, and Sparspak++ or Sparspak90 as

object oriented libraries that provide direct solvers for sparse systems of linear equations. Spindle and Oblio are two complementary libraries as Spindle provides ordering algorithms (minimum degree algorithms) and Oblio provides factorisations and for symmetric matrices. Sparspak++ and Sparspak90 are object oriented wrappers, C++ and Fortran 90 respectively, of the Sparspak library ([GL79], [GL81]).

Figures 3.27, 3.28, 3.29 and 3.30 present the classes `LinearSystemDirectSolver`, `GeneralFactorisation`, `KindOfPhase` and `Ordering`. `LinearSystemDirectSolver` has two sub-classes, `LinearSystemDirectSolverStructuredMatrix` and `LinearSystemDirectSolverSparseMatrix`, since the phases of solving a linear system are different for structured matrices and sparse matrices. `LinearSystemDirectSolverStructuredMatrix` is client of class `Factorisation` which represents the phases of solving a linear system with a structured matrix. `LinearSystemDirectSolverSparseMatrix` is a client of class `KindOfPhase`. `KindOfPhase` distinguishes between numerical ordering and factorisation represented as its sub-class `NumericalOrderingAndFactorisation`, and symbolic ordering and factorisation represented as `SymbolicOrderingAndFactorisation`. Since there is a dependence between the ordering phase and the factorisation, based on how the ordering is represented, `NumericalOrderingAndFactorisation` and `SymbolicOrderingAndFactorisation` are further specialised. Each of these classes is a client of two classes that represent the factorisation of a sparse matrix and the ordering. Class `Ordering` is specialised into `SymbolicOrdering` and `NumericalOrdering` and then further to take account of the data structure that represents the ordering. Class `GeneralFactorisation` is specialised into `Factorisation` and `SparseMatrixFactorisation`. `SparseMatrixFactorisation` is specialised for the structure in which the ordering is represented. Class `GeneralFactorisation` has as an attribute a boolean flag which indicates if pivoting is to be performed.

Iterative Solvers of Matrix Equations

An iterative solver of matrix equations comprises two phases that are repeatedly executed. The first phase is the algorithm itself, while the second phase is a termination test. The first phase usually requires preconditioning matrices. These matrices are created from the coefficient matrix in an attempt to make the algorithm converge in fewer iterations.

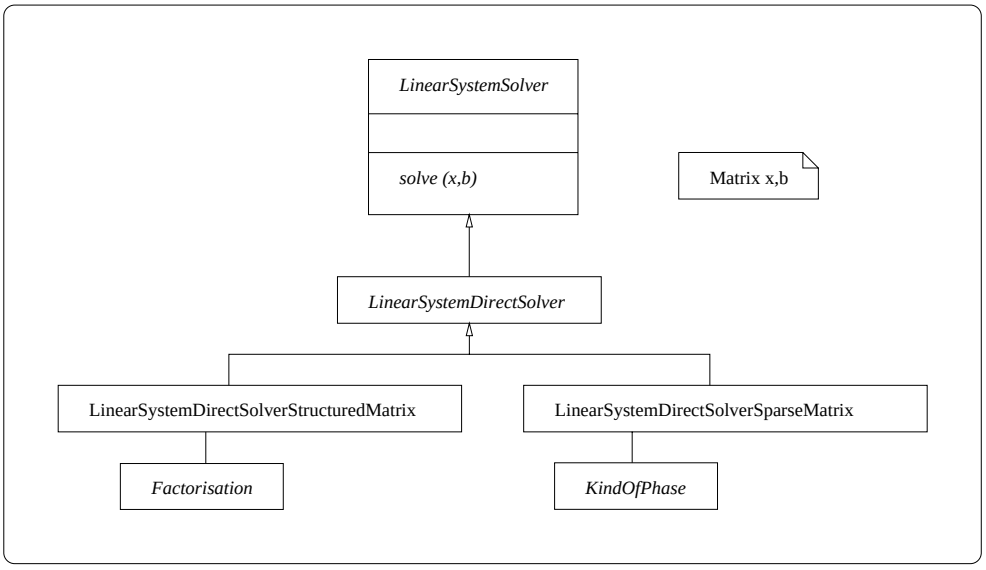


Figure 3.27: Class diagram of class `LinearSystemSolver` for direct solvers.

Some iterative algorithms are known not to converge for certain matrix properties. The best combination of a preconditioner and an iterative algorithm cannot be chosen, practically, only given the properties of the coefficient matrix. Users need to be able to select the iterative algorithm, the preconditioner, and the termination test to be used.

Figure 3.31 presents class `LinearSystemIterativeSolver`. A specific iterative algorithm is represented as a class inheriting from `LinearSystemIterativeSolver`. This class is a client of class `TerminationTest`. A termination test algorithm is represented as a class that inherits from `TerminationTest`. A method `test` that returns a `Boolean` is included in `TerminationTest`. The `create` method of sub-class of `LinearSystemIterativeSolver` takes as parameter the matrix defining the linear system of equations. When the algorithm can be preconditioned another method with the same name `create` but with other parameters the preconditioning matrices is included in the class.

A preconditioning matrix is the output of an operation that takes an input matrix and returns another matrix. Hence, a preconditioner operation is represented as a method in class `Matrix` having as parameter an object of class `Preconditioner`. Each kind of preconditioner operation is represented as a class inheriting from `Preconditioner`.

Table 3.6 presents various object oriented libraries that provide iterative solvers

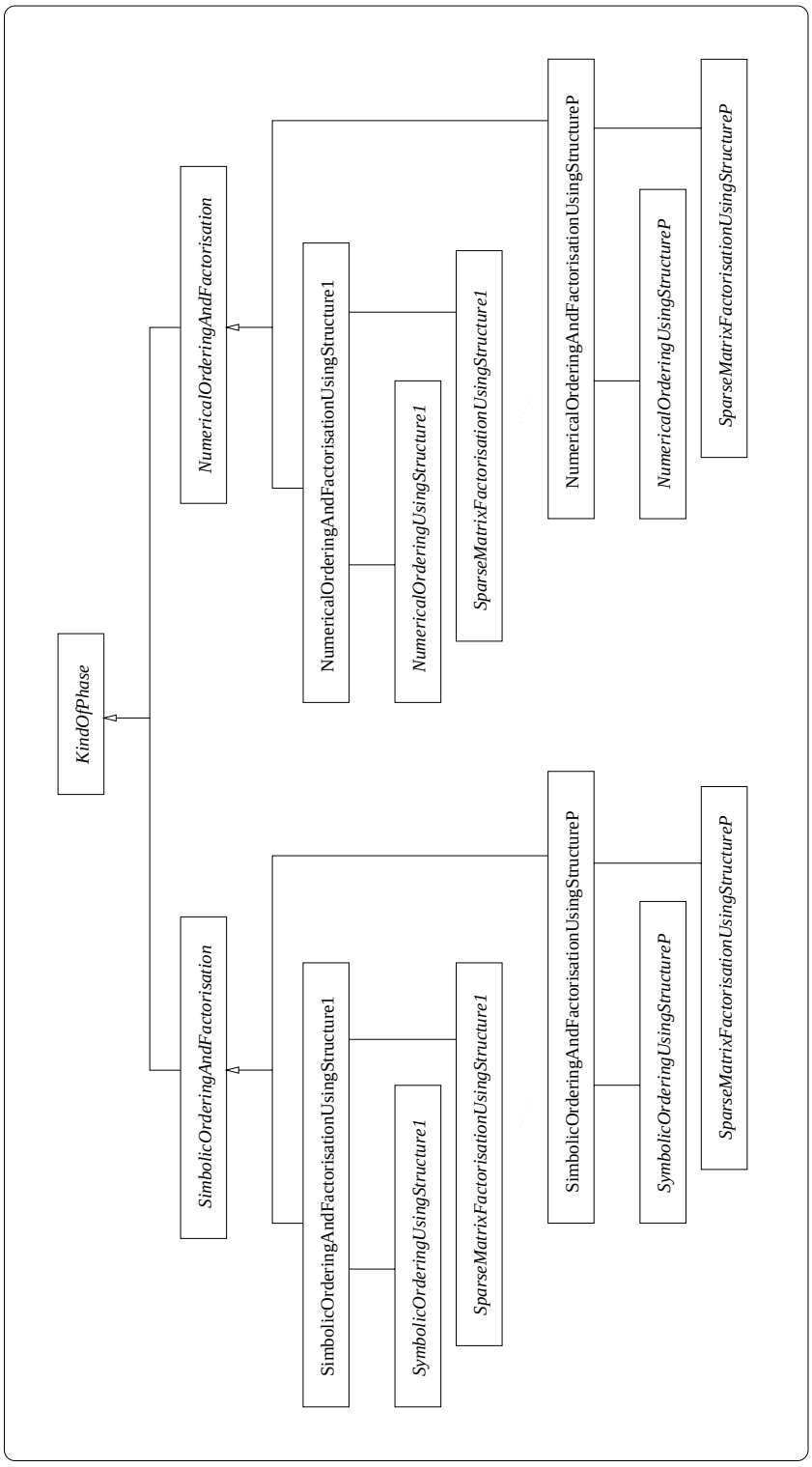


Figure 3.28: Class diagram of class `KindOfPhase` for direct solvers.

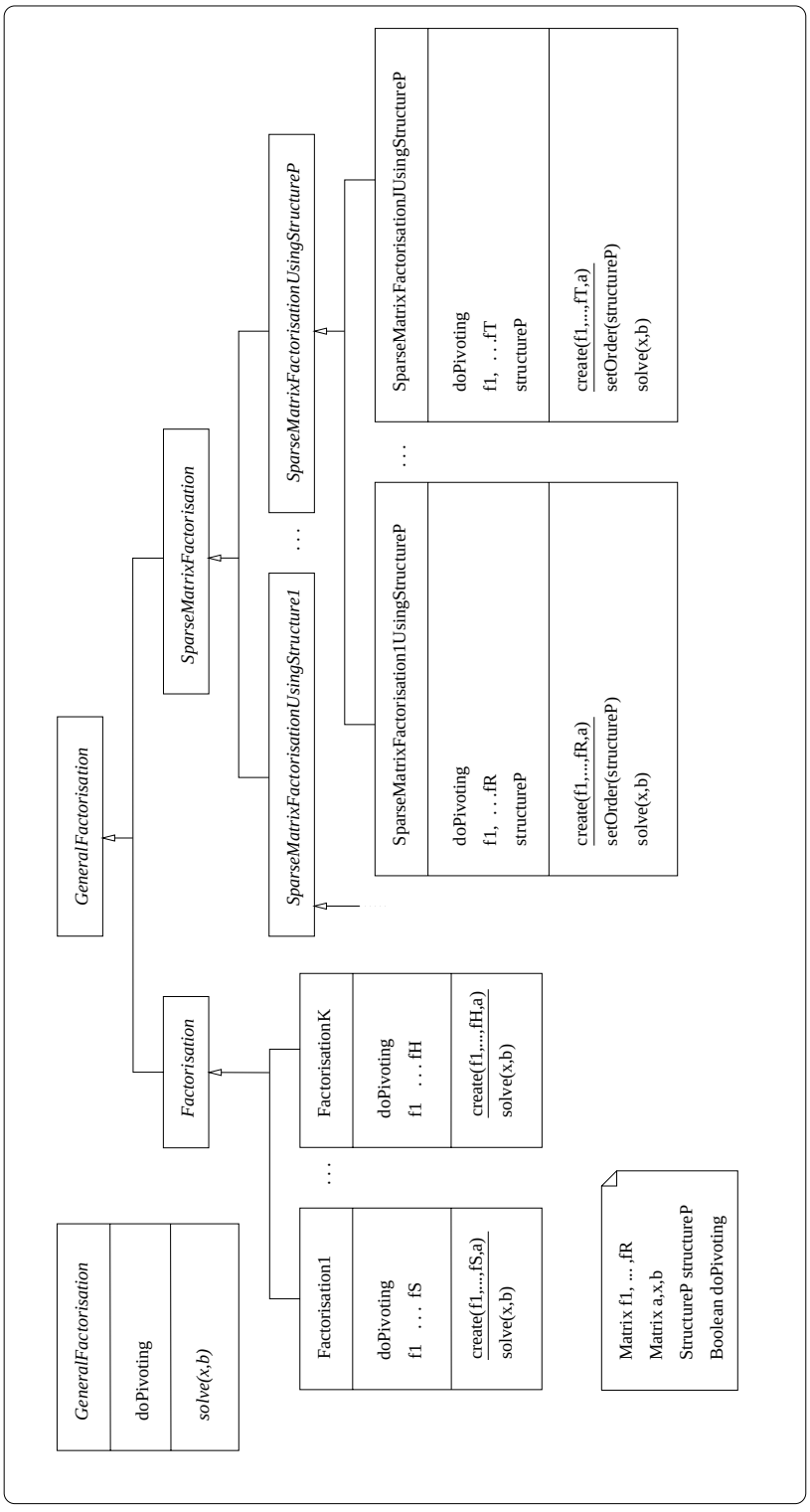


Figure 3.29: Class diagram of class `GeneralFactorisation` for direct solvers.

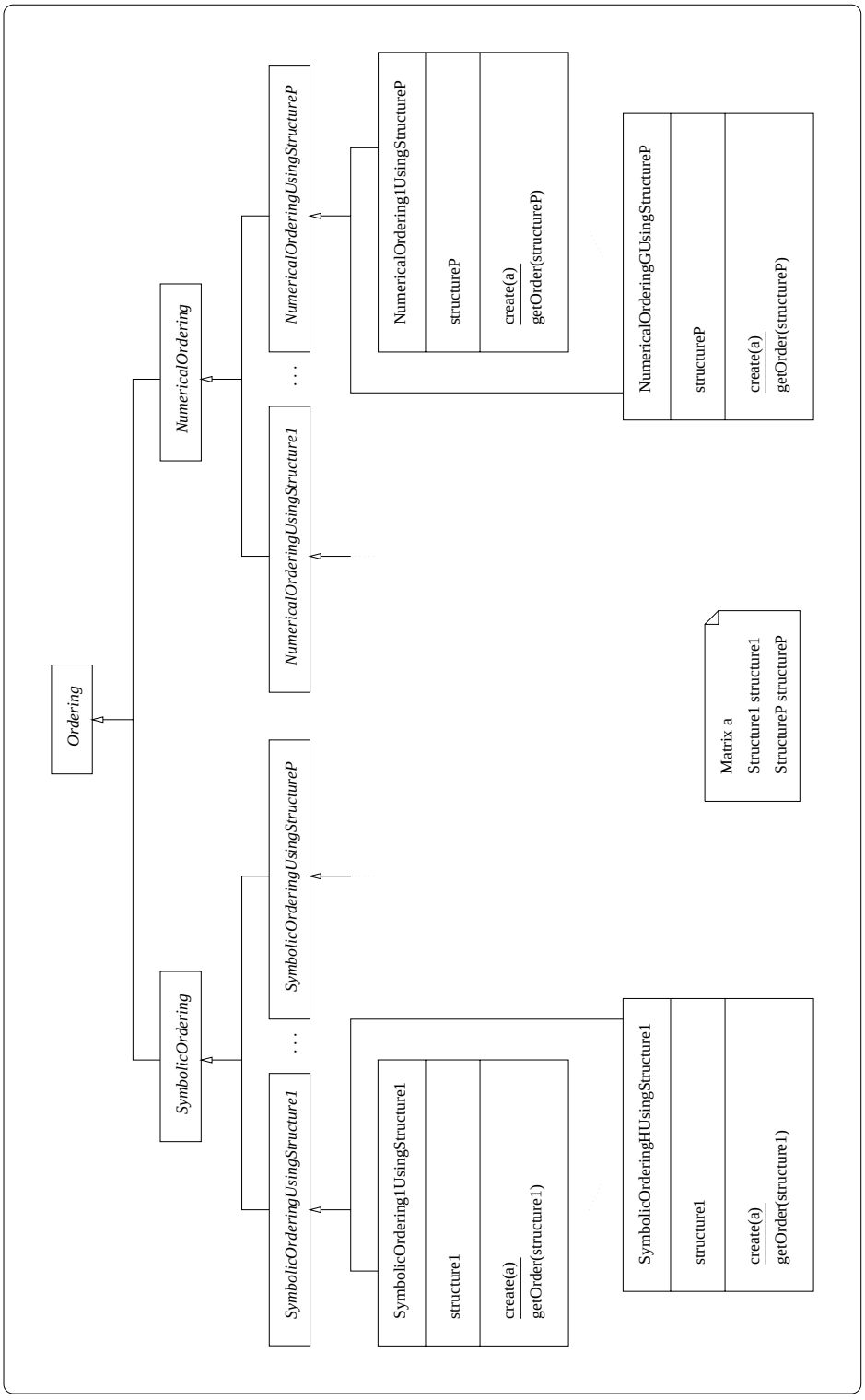


Figure 3.30: Class diagram of class `Ordering` for direct solvers.

for matrix equations.

3.3 Summary

The analysis and design of an object oriented linear algebra library is the core of this chapter. Object oriented software construction is proposed and reviewed as a way of improving the development of linear algebra programs.

The analysis and design has kept in mind the requirements of both users (experts and non-experts) and library developers. Traditional libraries provide users with complex interfaces, and library developers are faced with an explosion of matrix calculation implementations.

From the fact that matrix calculations are defined in terms of matrices and their dimensions and not in terms of matrix properties and storage formats, current object oriented libraries' designs (Tables 3.1 and 3.2, and Figures 3.12, 3.13, 3.14 and 3.15) do not fully model linear algebra. These libraries do not allow a matrix to vary its properties during execution time. Consider, for example, the $B \leftarrow A+B$ matrix calculation, where A and B are square matrices of order n, but A is a dense matrix while B is an upper triangular matrix. After the calculation is performed, B becomes a dense matrix. A new class structure (Figures 3.16 and 3.17) has been designed that enables a library to manage the storage formats and to propagate the matrix properties; this is a novel functionality for linear algebra libraries. In this way, matrices can vary their properties and storage formats transparently.

The class structure has been extended so that sections of matrices and matrices formed by merging other matrices can be created without the need to replicate matrix elements and can be used like any other matrix. Hence, the new matrices (sections and merged) can have any property. By contrast, the reviewed object oriented libraries (Table 3.1) consider these new matrices to be dense matrices (Table 3.3).

From the set of reviewed object oriented libraries (Tables 3.1, 3.4 and 3.5), and from the analysis and design reported in this chapter, the following guidelines support the creation of simpler interfaces:

- matrices are represented by classes that encapsulate the way they are stored;
- a matrix calculation is represented as a unique visible method, although

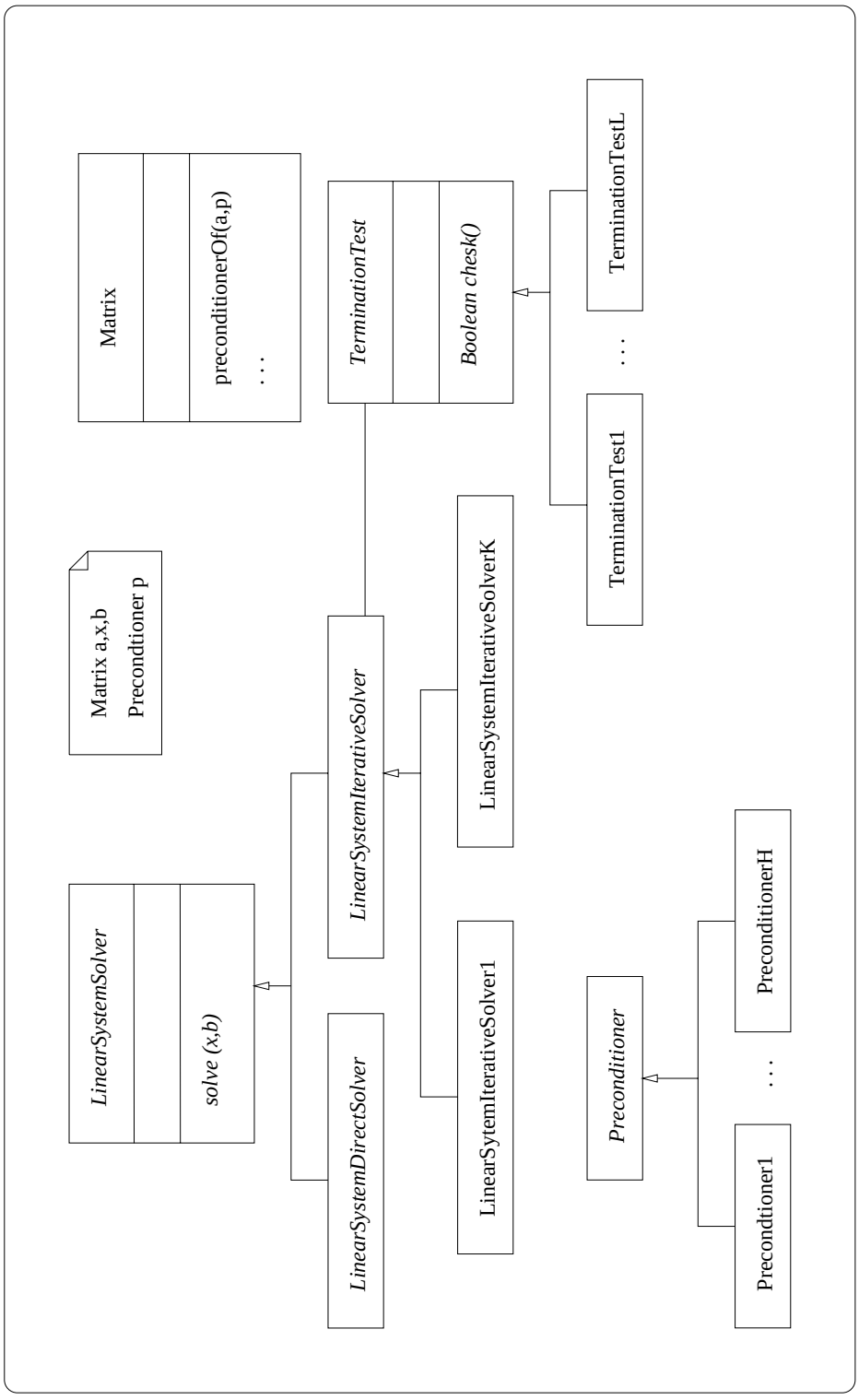


Figure 3.31: Class diagram of class `LinearSystemSolver` for iterative solvers.

different implementations and a rule based reasoning system that selects the adequate implementation are hidden behind the visible method; and

- when the reasoning system cannot be defined, the different algorithms, and not the implementations, are presented as classes and objects of these classes are passed as parameters.

The reviewed object oriented linear algebra libraries (Table 3.1) provide basic matrix operations, and solution of matrix equations with iterative and direct algorithms. However, none of them support all these matrix calculations (Tables 3.4 and 3.6). Following the above guidelines, a library interface that accounts for all these matrix calculations has been proposed. This class structure, the novel functionality and the proposed library interface constitute the design of a new library known as the Object Oriented Linear Algebra LibrAry (OoLaLa).

Developers of traditional libraries have benefited from two abstraction levels at which matrix calculations can be implemented. These abstraction levels reduce the number of implementations. Matrix abstraction level enables matrices to be represented and accessed independently of their storage formats. Iterator abstraction level is an implicit way of traversing matrices. That is, a matrix is traversed without explicitly expressing the positions of the elements that are accessed. A matrix iterator is defined so that it accesses only the elements that can be implied to be nonzero from the matrix properties.

The next chapter adapts OoLaLa to a specific object oriented programming language, Java. The implementation of the novel functionality and the implementation of matrix calculations using the abstraction levels are also illustrated. Chapter 5 describes the problems or limits in developing linear algebra programs based on libraries, either traditional or object oriented.

Readers interested in acquiring more background on object oriented software construction are recommended to look at [Mey97], [Boo94] and [GHJV95], and, as introductions to object oriented scientific programming, at [Dub97] and [Nor96]. Modelica ([MEO98], [FE98], [Mod]), and MathObject ([FVHF92], [FEV93], [FA93], [FVHF95], [AF95], [Obj]) offer an object oriented mathematical language that allows users to represent equation-based model directly. The projects Overture ([BHQ98], [BCHQ97], [BDH+98], [BHQ99], [OVE]), Pooma ([HKBR98], [HRC+98], [KCC+98], [CCH+99], [HC99], [POO]), Cogito ([Ran95], [Åhl95], [MOT97], [TMO+97], [Cog]), Diffpack ([BL97], [Lan99] , [Dif]) and PETSc ([BGMS97], [BGMS99], [PET]) have focused on object oriented partial differential equations.

Other object oriented linear algebra libraries, such as SLES a library of iterative solvers of systems of linear equations that is part of PETSc, SMOOTH ([AL96], [SMO]) an ordering library of sparse matrices, and SPOOLES ([AG99], [SPO]) a library of direct solvers for sparse linear equations, have not been reviewed since they are implemented in C (a non object oriented language) and, therefore, their designs are limited. Other object oriented linear algebra libraries have a different design objective. For example, LAKe ([NE99]) focuses on using the same code for sequential and parallel iterative solvers, and Cactus ([McD89]) focuses on finite dimensional vector spaces instead of matrix algebra. Other object oriented linear algebra libraries that have not been reviewed in this thesis include TNT ([Poz97], [TNT]) and some others listed at http://oonumerics.org/oon.

Chapter 4

Implementation of OoLaLa

The previous chapter has reported an analysis and design of an object oriented linear algebra library. The library, OoLaLa, has been designed independently of any programming language. OoLaLa offers a novel functionality for libraries: propagation of matrix properties and management of storage formats. OoLaLa also enables library developers to implement matrix calculations at two abstraction levels: matrix and iterator abstraction levels. These abstraction levels reduce the number of implementations of a given matrix calculation.

Matrix abstraction level is independent of the storage format in which matrices are represented. A given matrix element is mapped automatically to the position of its storage format. Iterator abstraction level, apart from also being independent of the storage format, traverses matrices without explicitly indicating the positions of the elements that are accessed. A matrix iterator is defined so that only the nonzero elements of matrices are accessed.

The objective of this chapter is to describe how OoLaLa is adapted and implemented in Java. At the same time, example programs are presented to show users how to develop programs using OoLaLa.

Firstly, OoLaLa is adapted to the specific characteristics of Java (Section 4.1). An example program that declares, creates and initialises matrices, illustrates how these are implemented using UML object diagrams (introduced in the previous chapter) and UML sequence diagrams (introduced in this chapter) (Section 4.2). Two more example programs show how views (i.e. sections of a matrix or matrices formed by merging other matrices) are created and how they are implemented (Section 4.3). The management of storage formats is presented in conjunction with the propagation of properties (Section 4.4). Finally, matrix

calculations are implemented at matrix and iterator abstraction levels (Section 4.5).

4.1 Adapting OoLaLa to Java

Java is a clean and strongly typed object oriented language. Unlike other languages (C++, Ada95, ...) which have evolved from their procedural subsets (C, Ada83, ...), Java was designed to be an object oriented language. Java offers built in parallelism, a powerful set of classes to develop graphical interfaces and makes network based applications easy to program.

Java programs are compiled into an intermediate language known as *bytecode*. A Java Virtual Machine (JVM) is an interpreter of bytecodes. The JVMs enable Java programs to be written once and run on any computer (as long as an implementation of a JVM exists for it). Both the language and the JVM have been fully specified, leaving no details to the discretion of compiler developers.

The Java Grande Forum[1] (JGF), an open forum to academia, industry or government, was formed under the belief that "Java has potential to be a better environment for Grande Applications development than any previous languages such as Fortran and C++" [Jav98]. The term Grande Application is also defined "as an application of large-scale nature, potentially requiring any combination of computers, networks, I/O, and memory". Numerical linear algebra libraries are the kernels of most of these applications.

However, Java has some poor characteristics for implementing OoLaLa:

- Java does not support multiple inheritance;
- Java does not support generic classes, nor complex numbers as a language data type, nor light-weight classes; and
- Java specifies a multidimensional array as an array of arrays.

The following paragraphs discuss the problems that these characteristics of Java cause and the decisions taken to overcome them.

Multiple inheritance has been used in the class structure of OoLaLa to model matrix properties that result from composing other matrix properties. For example, class `SymmetricBandedProperty` inherits from `SymmetricProperty` and

[1] Java Grande Forum web site at http://www.javagrande.org/

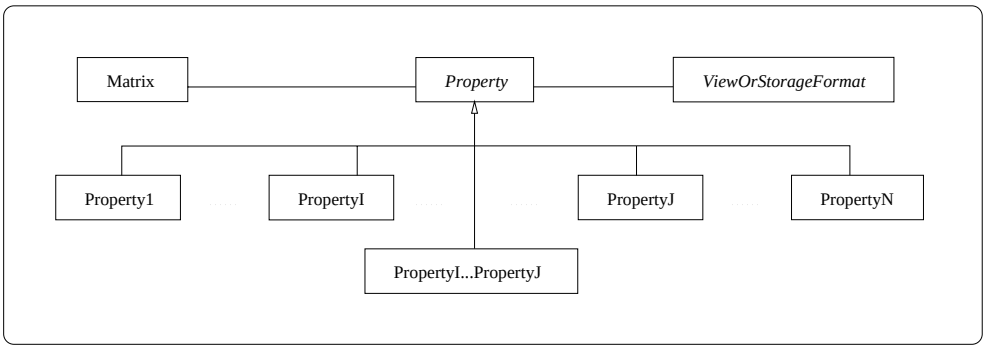


Figure 4.1: Class diagram of class `Property` and its sub-classes adapted to Java.

`BandedProperty`. Since multiple inheritance is not available, every class representing matrix properties simply inherits from the class `Property`. Figure 4.1 presents the changes to the `Property` class inheritance hierarchy.

Ideally, generic classes would be used to develop only one version of OoLaLa independent of the data type of the matrix elements. Users would choose the data type of the matrix elements and the compiler would generate automatically the version of OoLaLa. Section 3.1.4 described how generic classes can be emulated using inheritance and client relation. The OwlPack linear algebra library ([BK99b], [BK99a], [BKP98]) has been implemented emulating generic classes by an equivalent class `Matrix` having a client relation with an abstract class `Number` from which `Float`, ..., `Complex` classes inherit. OwlPack also has been implemented by writing one version of the library for each data type. It is reported that the version emulating generic classes is between 4 times and 100 times slower than writing one version for each data type depending on the benchmark. In order to close the gap, Budilimć and Kennedy ([BK99b], [BK99a]) propose interprocedural and interclass compiler optimisation, which are only possible if the compilation strategy of Java is changed. Currently, the compiler can only consider one class at the time. On the other hand, JGF proposes the inclusion of light-weight classes. A light-weight class is a class whose objects are treated by the compiler and the JVM as variables of a language data type. In response to this proposal, Sun (owner of Java specification) plans to write a proposal for light-weight classes [Jav99]. Other projects have experimented with generic classes in Java ([AFM97], [BML97], [OW97]). Given current circumstances, OoLaLa is implemented by developing a version for each data type.

Java multidimensional arrays are specified to be an object array that has objects array. This specification creates a very powerful data structure. Given a two-dimensional Java array, each of its one-dimensional arrays can be substituted with different arrays of different sizes. However, this structure does not ensure that the objects array are continuous in memory and this might result in a poor memory locality. This structure also needs to perform bound and null object checks for each dimension since both checks are compulsory in the Java language specification. This array structure and the precise exception model do not allow all the compiler optimisation techniques developed for fixed size multidimensional arrays. The Java exception model specifies that an exception must appear in strict program order. The above motivated the JGF Numerics Working Group to develop a Java package with multidimensional arrays mapped into one-dimensional Java arrays. This package was developed by the IBM's Ninja group[2] [MMG99], which at the same time developed compiler techniques to identify exception-free code sections. For these exception-free code sections, compiler optimisation techniques developed for Fortran and C can be applied [MMG98]. The experiments reported with the array package show an improvement of 15% in MFlops performance when multiplying two matrices using two-dimensional arrays from the package compared with two-dimensional language arrays [MMG99].

Blount and Chatterjee ([BC99],[BC98]) in their JLAPACK library also store matrices by mapping them into one-dimensional language arrays. They optimised this approach by noting that most matrix calculations are implemented with sequential, stride *inc*, access to the matrices. This enables the next position in a one-dimensional language array to be calculated as $lastposition + inc$ instead of as $i - 1 + (j - 1) * n$, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Note that a Java language array has its first element in index 0 and that two-dimensional arrays are mapped column-wise (as in Fortran). It is reported that LU factorisation on a Pentium II is around 1.5 times faster and on a Ultra Spark around 3 times faster than a version obtained using the f2j translator [DDS99] which uses language two-dimensional arrays ([BC98], [BC99]).

OoLaLa represents two-dimensional arrays by mapping them to one-dimensional language arrays in a column-wise form. In order to exploit Blount and Chatterjee's array access observation, new methods, `incIndexColumn` and `incIndexRow`,

[2]Ninja group address http://www.research.ibm.com/ninja

are included in `Matrix`. This constitutes the final design modification due to particular features of Java.

4.2 Declare and Access Matrices

The first step in writing a program is to declare variables. In numerical linear algebra the variables are mainly matrices. Using OoLaLa, users declare objects of class `Matrix`. These objects are then given dimensions and properties. The next step in writing a program is to initialise the variables. In OoLaLa, users access objects of class `Matrix` using mainly `element` and `assign`. Figure 4.2 gives an example program to declare and initialise matrices.

Figure 4.3 introduces UML sequence diagram notation. A sequence diagram is a way of representing the life (creation, invocations of methods, and destruction) of objects over time. Objects are represented by rectangles in which their names and class names are written underlined. A method invocation is represented as an arrow with solid head from the object that invokes the method to the object where the method is invoked. An object (in sequential execution) becomes active when a method is invoked in it. The time that an object is active is represented by a thin rectangle under the object. An object remains active while an invoked method remains unfinished. This does not mean that the flow of control is in this object. The flow of control is transferred to another object when a method is invoked in this other object. The flow of control returns when the method is finished. The arrows represent the transfer of control flow (in sequential execution).

Figure 4.4 presents the sequence diagram for the statements labelled as action 1 and action 2 in the example program of Figure 4.2. The first statement declares an object of class `Matrix` and the second statement sets the dimension and property. Users only perceive what is on the left of the object `a` in Figure 4.4; the methods invoked and objects on its right are not visible to users. Figure 4.5 presents the object diagram after every object `Matrix` has been declared and properties have been set. Note that only the object `e` has requested a specific storage format. The other storage formats have been selected automatically, see Section 4.4 for details.

Finally, Figure 4.6 presents the sequence diagram for the statements labelled as action 3 and action 4 in the example program of Figure 4.2. These are invocations to `assign` and `element` methods. Again, users only perceive what is on

```
class DeclareAndAccessMatrices
{
  public static void main(String args[])
  // how to declare and set properties
  {
    // begin declare matrices
    Matrix a = new Matrix(); // action 1
    Matrix b = new Matrix(); Matrix c = new Matrix();
    Matrix d = new Matrix(); Matrix e = new Matrix();
    // end declare matrices
    double temp;

    // begin set matrices properties
    a.setDenseMatrix(10,15); // action 2
       // numRows=10 and numColumns=15
    b.setBandedMatrix(20,30,2,1);
       // numRows=20, numColumns=30,
       // numUpperBandwidth=2 and numLowerBandwidth=1
    c.setSymmetricMatrix(15);
       // numRows=15 and numColumns=15
    d.setSymmetricBandedMatrix(15,3);
       // numRows=15, numColumns=15,
       // numUpperBandwidth=3 and numLowerBandwidth=3
    e.setBandedMatrix(100,100,50,65,OOLaLaStorageFormat.denseFormat());
       // numRows=100, numColumns=100
       // numUpperBandwidth=50 and numLowerBandwidth=65
       // requested dense format
    // end set matrices properties

    // begin access matrices
    a.assign(8,6,3.14159); // action 3
    temp=a.element(8,6); // action 4
    // end access matrices
  }// end main
}// end class DeclareAndAccessMatrices
```

Figure 4.2: Example program of how to declare and access matrices using OoLaLa.

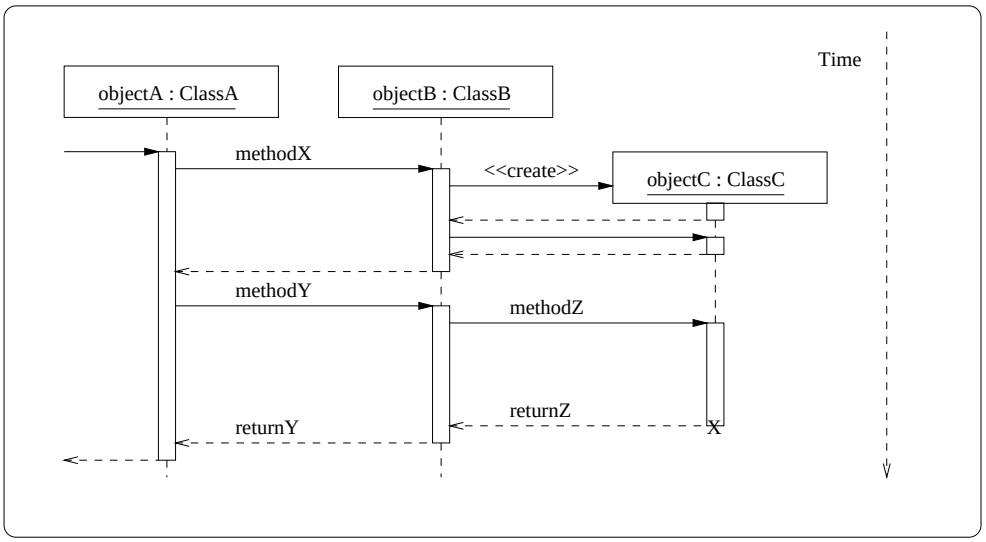


Figure 4.3: UML sequence diagram notation.

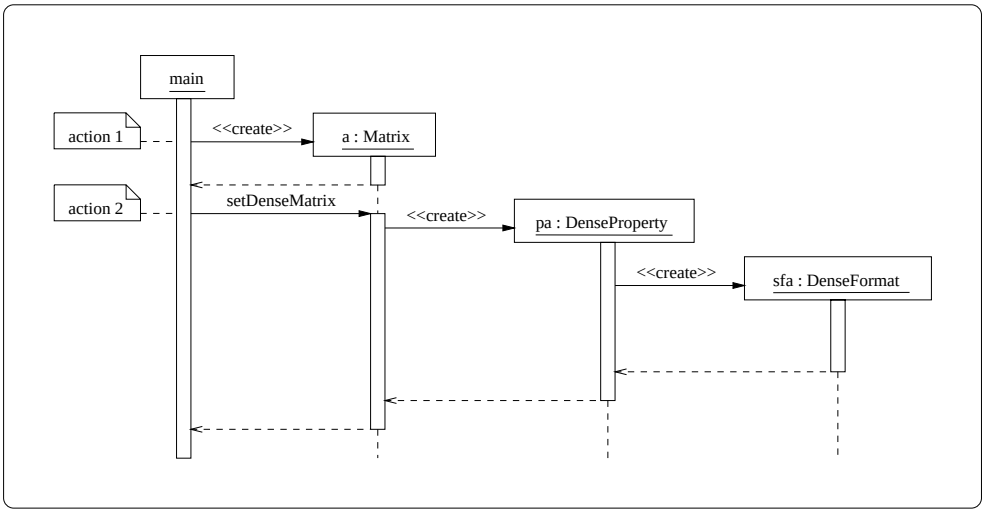


Figure 4.4: Sequence diagram for declaring a dense matrix using OoLaLa.

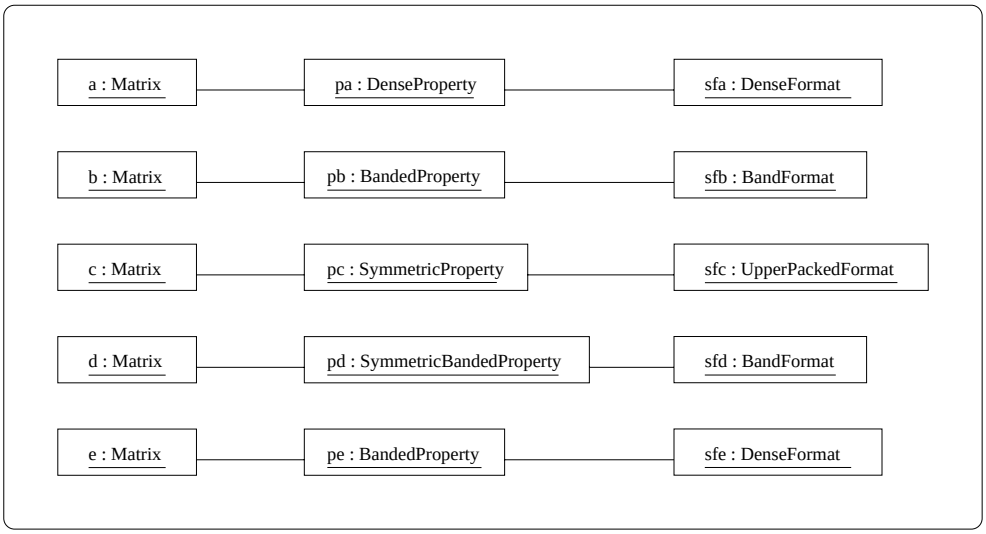


Figure 4.5: Object diagram after declaring and setting properties of matrices.

the left of object `a` in Figure 4.6, the rest occurs transparently.

4.3 Create Views

A view can be either a section of a matrix or a matrix formed by merging other matrices. Figure 4.7 presents an example program showing how different sections of matrices can be created. In this program, a 5×5 dense matrix A is represented by an object `a` of class `Matrix`. Three matrices represented by three objects (`section1`, `section2` and `section3`) of class `Matrix` are created as sections of `a`. These three matrices do not replicate the matrix elements. Figure 4.8 presents the sections of the matrix A for each object `Matrix`. Figure 4.9 presents the sequence diagram for the program and Figure 4.10 presents the object diagram after all the sections have been created. Each object of class `Matrix` has its own properties, but they share the object of class `DenseFormat`. This shared object stores the elements of the matrix A and, consequently, the elements of the defined sections of A.

A merged matrix is formed by merging other matrices. Figure 4.11 presents an example program which creates a 5×5 block diagonal matrix from its block sub-matrices. The objects `zero1_2`, `zero2_1` and `zero2_2` of class `Matrix` represent zero matrices with different dimensions. The objects `diag1`, `diag2` and `diag3` of class `Matrix` represent the block sub-matrices which are on the diagonal of

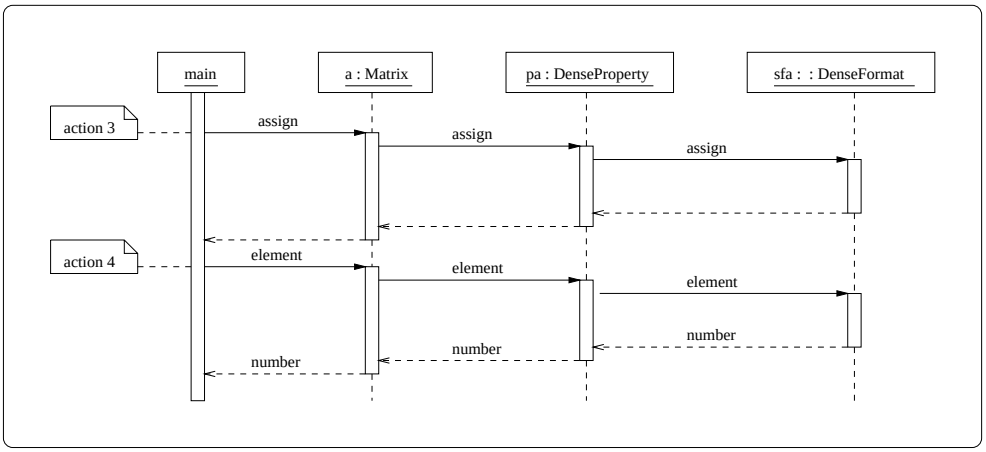
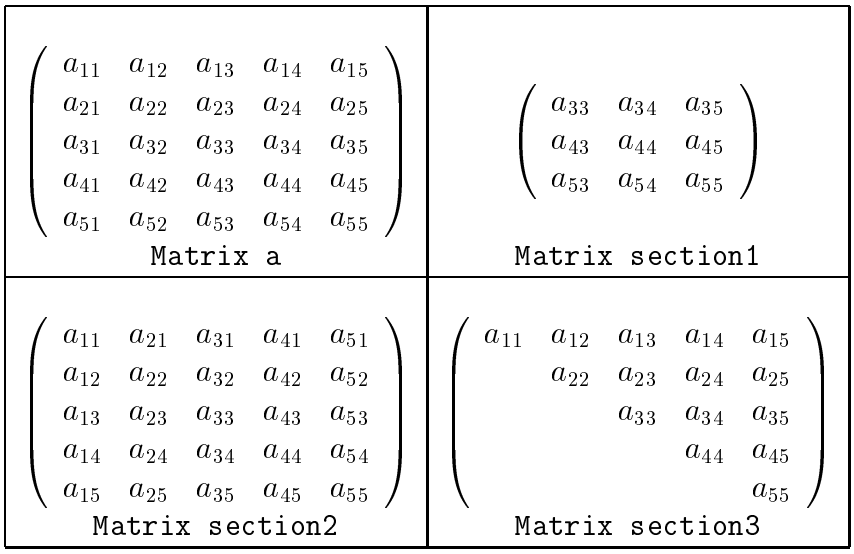


Figure 4.6: Sequence diagram for access methods.

```
class CreateSections
{
  public static void main (String args[])
  {
    // begin declare matrices
    Matrix a= new Matrix();
    Matrix section1= new Matrix();
    Matrix section2= new Matrix();
    Matrix section3= new Matrix();
    // end declare matrices

    a.setDenseMatrix(5,5); // set properties
    // begin create sections
    a.getSubMatrix(section1,3,5,3,5);
    a.getTranspose(section2);
    a.getUpperTriangularSection(section3);
    // end create sections
  }// end main
}// end CreateSections
```

Figure 4.7: Example program of how to create sections of matrices using OoLaLa.



Notation: blanks represent zero elements which cannot be modified.

Figure 4.8: Graphical representation of the sections of matrices and matrices created in Figure 4.7.

matrix A. Matrix A is represented by an object `a` of class `Matrix` formed after the execution of the statement labelled as action 1. Figure 4.12 describes the object structure after this statement has been executed. Figure 4.13 presents the matrices that each object of class `Matrix` represents.

The object `a` represents a block diagonal matrix. Looking at the object diagram (see Figure 4.12), the block diagonal matrix is stored as a set of objects of class `StorageFormat`. Each object is used for certain block sub-matrices of A. In general, any matrix can be partitioned into block sub-matrices. Each block can have different properties and therefore different advisable storage formats. The class structure of OoLaLa enables users to operate transparently with a matrix that is stored by its blocks, and each block is stored in any advisable storage format.

Moreover, since the object `a` is of class `Matrix`, sections of the matrix represented by `a` can be also created regardless of `a` being stored by its blocks. The statement labelled as action 2 in Figure 4.11 makes the object `section` a section of the matrix represented by `a`. The object `section` represents a diagonal matrix (see Figure 4.13). Hence, the object diagram in Figure 4.14, presents the object `section` linked to an object of class `DiagonalProperty`. Efficient algorithms for

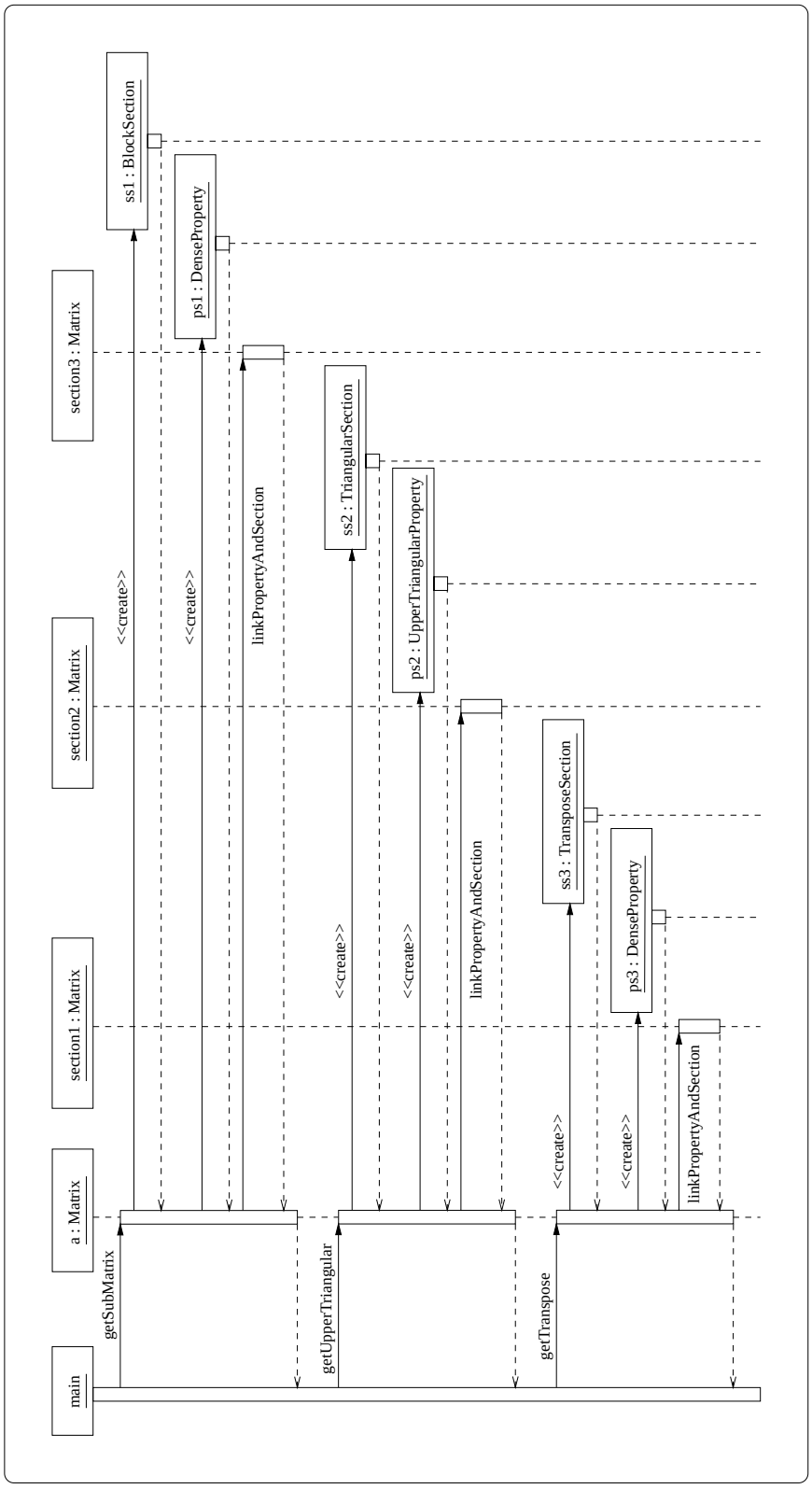


Figure 4.9: Sequence diagram for the sections created in Figure 4.7.

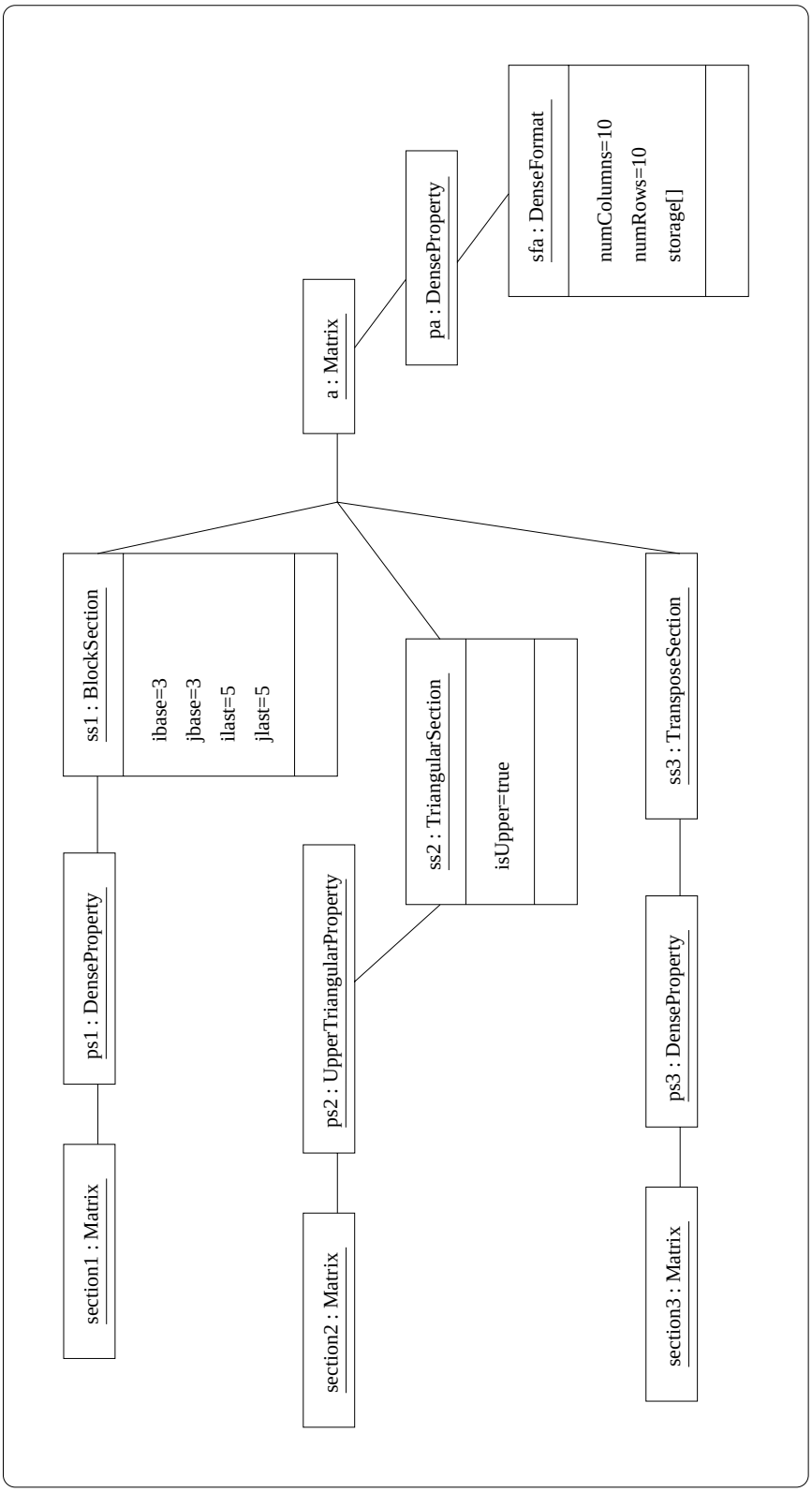


Figure 4.10: Object diagram after the sections have been created in Figure 4.7.

```
class CreateAMergedMatrix
{
  public static void main (String[] args)
  {
    //begin declare matrices
    Matrix a= new Matrix();
    Matrix zero1_2= new Matrix();
    Matrix zero2_1= new Matrix();
    Matrix zero2_2= new Matrix();
    Matrix diag1= new Matrix();
    Matrix diag2= new Matrix();
    Matrix diag3= new Matrix();
    Matrix section = new Matrix();
    //end declare matrices
    Matrix array={{diag1,zero2_1,zero2_2},
                  {zero1_2,diag2,zero1_2},
                  {zero2_2,zero2_1,diag3}};

    // begin set properties
    diag1.setDenseMatrix(2,2);
    diag2.setDenseMatrix(1,1);
    diag3.setLowerTriangularMatrix(2,2);
    zero1_2.setZeroMatrix(1,2);
    zero2_1.setZeroMatrix(2,1);
    zero2_2.setZeroMatrix(2,2);
    // end set properties

    // create a matrix by merging matrices
    a.merge(array); // action 1
    // create a section of matrix
    a.getSubMatrix(section,2,4,2,4); // action 2
  }// end main
}// end CreateAMergedMatrix
```

Figure 4.11: Example program of how to create a matrix by merging matrices using OoLaLa.

determining the nonzero elements structure, described in [BW99], enable the library to identify the matrix as being diagonal. In this way, a section of matrix A or of a set of matrices (block sub-matrices) can be created and used transparently as a matrix.

4.4 Management of Storage Formats

In all the examples that have been presented, the programs have not specified the class of storage format, except once (see Figures 4.2, 4.7, and 4.11). However, the

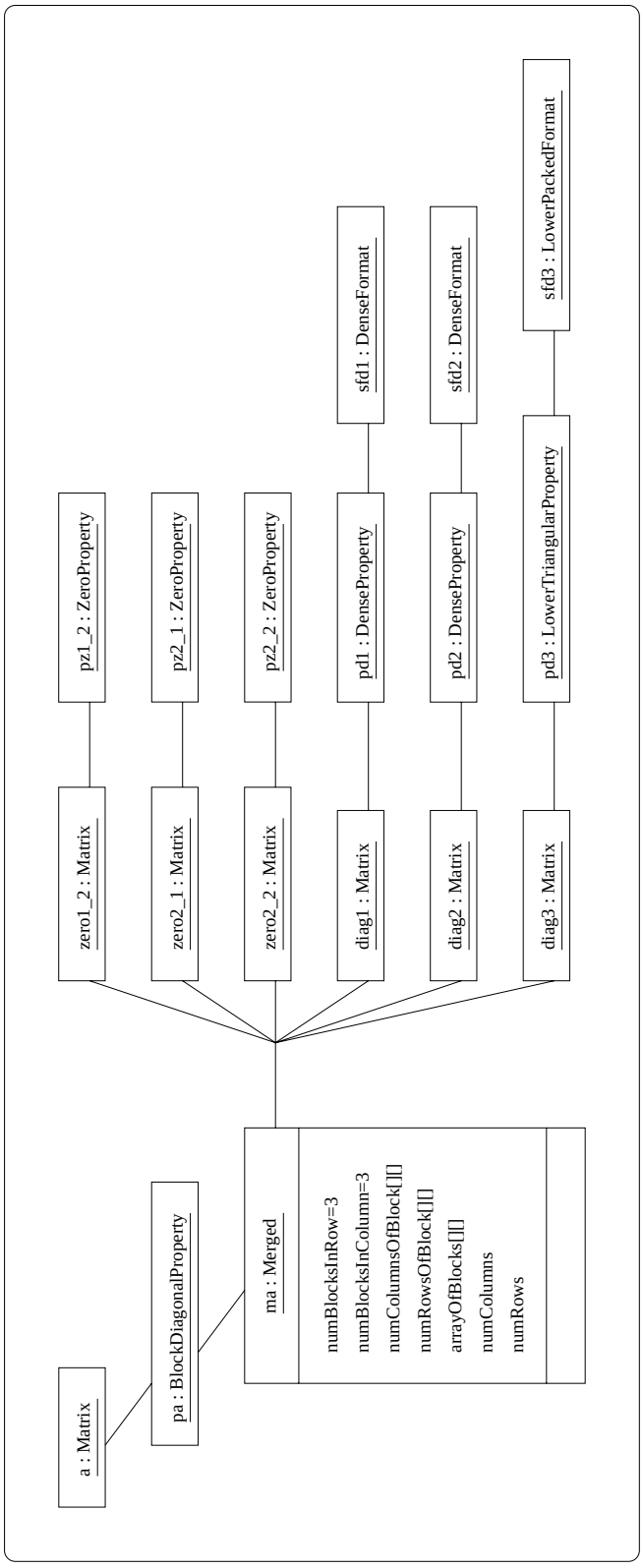


Figure 4.12: Object diagram after a matrix has been created by merging matrices from example program in Figure 4.11.

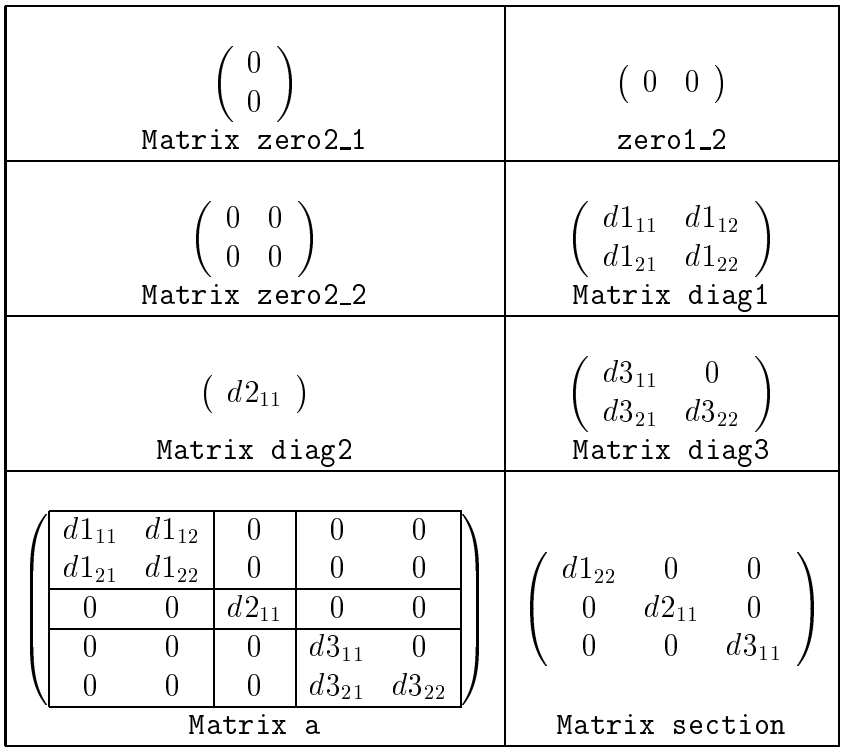


Figure 4.13: Graphical representation of the matrices created in Figure 4.11.

object diagrams have always presented objects of a sub-class of `StorageFormat` (see Figures 4.5, 4.10, and 4.12). OoLaLa chooses automatically and statically a storage format for every matrix. Before invoking any matrix calculation that changes the matrix elements, OoLaLa decides whether the storage format and properties need to be changed. Consider, for example, the addition $C \leftarrow A + B$ where A and B are tridiagonal matrices and C is a bidiagonal matrix. After performing the addition C also becomes tridiagonal. A program using OoLaLa would create objects `a`, `b`, `c` of class `Matrix` and set the correspondent properties of each matrix. The program would continue with the statement `c.addInto(a,b);`; this method invoked in `c`, would change its linked object of class `BidiagonalProperty` by one of class `TridiagonalProperty`. Depending on the class of the linked object that represents the storage format, different action could be taken. Suppose the actual storage format is large enough to store the extra elements that will be created and it is advisable to have the result matrix in this storage format, then no action is needed. This would be the case if `c` were linked with an object of class `DenseFormat`. Otherwise (either the storage format is not large enough or it is not advisable to store the result matrix in that storage

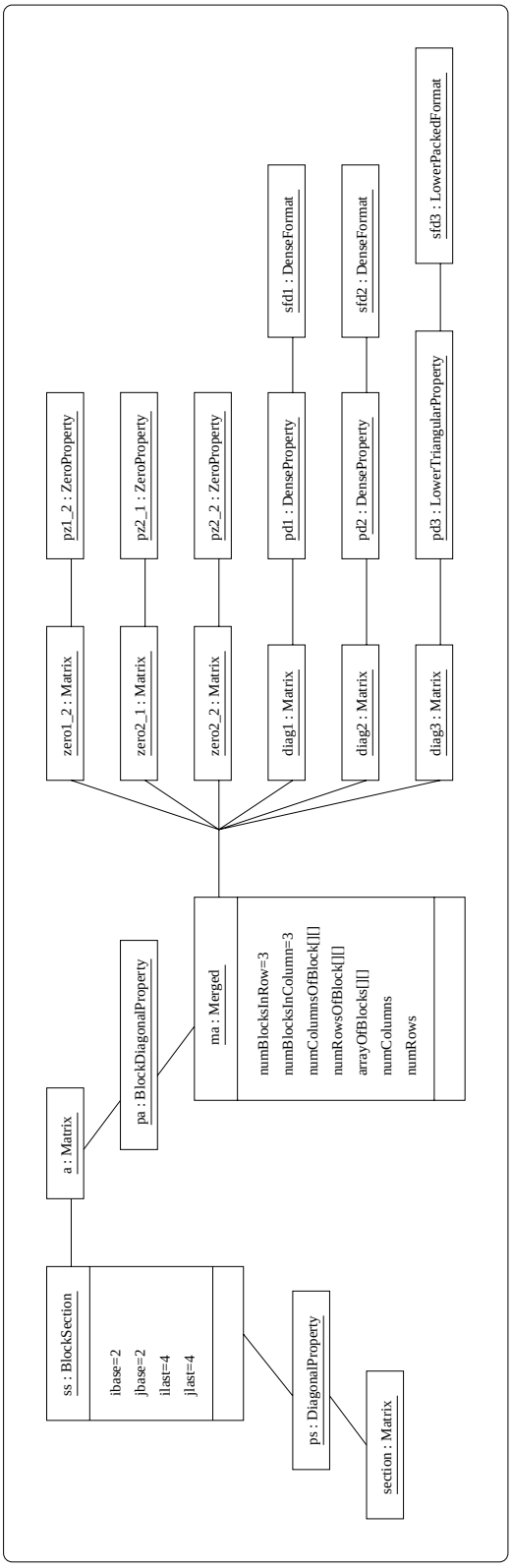


Figure 4.14: Object diagram after a section of matrix, which has been created by merging matrices, is created – example program in Figure 4.11.

Property	Storage Format
de	df
ba	df or bf
sy	upf
sb	upf or bf
ut	upf
lt	lpf
ub	upf or bf
lb	lpf or bf

Table 4.1: Storage format selected for each matrix property.

format), the linked object representing the storage format would be changed.

The following paragraphs explain how to select a storage format for a certain property, how to detect inconsistency between properties and storage formats, and how consistency is recovered. The description is limited to a set of properties (dense (de), banded (ba), symmetric (sy), symmetric banded (sb), upper triangular (ut), and lower triangular (lt) properties) and a set of storage formats (dense (df), band (bf), upper packed (upf) and lower packed (lpf) formats). These properties and storage formats are those supported in BLAS ([DCHH88b], [DCHD90]) and were described in Section 2.2.

The first question to answer is how OoLaLa chooses a storage format for a matrix with certain properties. Table 4.1 presents recommended storage formats for each matrix property as a set of static "if-then" rules. These rules do not have an explicit representation in OoLaLa; they are included in the code of each method `setPropertyMatrix`. For example, inside the code of `setDenseMatrix` there is a part that creates an object of class `DenseFormat`. The rules select, whenever possible, a storage format which uses the least memory space. Note that some rules can choose between band format and another format. The band format is selected when the upper bandwidth and lower bandwidth are less than half the number of columns and number of rows, respectively. This condition is an initial guess that needs validation with experiments.

The second question is how to detect inconsistency between matrix properties and storage formats. An inconsistent situation can only arise when a user sets a property and an inconsistent storage format, or when a matrix calculation results in a change of property. The first case is easier to solve. Table 4.2 presents

	df	bf	upf	lpf
de	√			
gb	√	√		
sy	√		√	
sb	√	√	√	
ut	√		√	
lt	√			√
ub	√	√	√	
lb	√	√		√

Table 4.2: Consistency between storage formats and matrix properties.

the advisable combinations. When a user sets a property and a storage format (e.g. `a.setDenseMatrix(10,10,OOLaLaStorageFormat.bandFormat());`) OoLaLa checks the combination against Table 4.2 and raises an exception of class `NonAdvisablePropertyAndStorageFormatCombination` when necessary.

The second case, when a matrix calculation results in a property change, requires the prediction of the new property of the matrix and the identification of the circumstances under which each matrix calculation triggers a property change. Note that a property is considered to be changed even if the property is the same but some characteristic of the property has been changed. For example, a banded matrix may remain a banded matrix but with a reduced or increased bandwidth.

Table 4.3 presents the matrix property of the result matrix from an analysis of the operand properties for matrix addition. Table 4.4 presents equivalent information for matrix-matrix multiplication. These tables are constructed assuming no knowledge of the numerical values of the elements apart from that implied by the properties of their matrices.

The prediction of matrix properties for matrix-matrix multiplication and matrix addition is fully determined; the tables represent static "if-then" rules. These rules use the properties of the operands to decide the property of the result matrix. OoLaLa implements them as internal tables that are consulted by the codes of `addInto` and `multiplyInto`.

Having explained how to detect inconsistent combinations of matrix properties and storage formats, it is now described how to recover consistency; i.e., the storage format needs to be changed in order to be consistent with the matrix property. Table 4.5 presents the selection of the new storage format. In the first

A \ B	de	ba	sy	sb	ut	lt	ub	lb
de	0	0	0	0	0	0	0	0
ba	0	1	0	1	1	1	1	1
sy	0	0	2	2	0	0	0	0
sb	0	1	2	3	1	1	1	1
ut	0	1	0	1	4	0	4	1
lt	0	1	0	1	0	5	1	5
ub	0	1	0	1	5	1	6	1
lb	0	1	0	1	1	4	1	7

```
0 →  c.setDenseMatrix(a.numRows(), a.numColumns())
1 →  if (Math.max(a.upperBandwidth(), b.upperBandwidth())
     == a.numColumns()-1 && Math.max(a.lowerBandwidth(),
     b.lowerBandwidth()) == a.numRows()-1)
     { c.setDenseMatrix(a.numRows(), a.numColumns()) }
     else { c.setBandedMatrix(a.numRows(), a.numColumns(),
     Math.max(a.upperBandwidth(), b.upperBandwidth()),
     Math.max(a.lowerBandwidth(), b.lowerBandwidth())) }
2 →  c.setSymmetricMatrix(a.numRows())
3 →  c.setSymmetricBandedMatrix(a.numRows(), a.upperBandwidth())
4 →  c.setUpperTriangularMatrix(a.numRows(), a.numColumns())
5 →  c.setLowerTriangularMatrix(a.numRows(), a.numColumns())
6 →  c.setSymmetricMatrix(a.numRows())
7 →  c.setUpperTriangularBandedMatrix(a.numRows(),
     a.numColumns(), Math.max(a.upperBandwidth(),
     b.upperBandwidth()), 0)
8 →  c.setLowerTriangularBandedMatrix(a.numRows(),
     a.numColumns(), 0, Math.max(a.lowerBandwidth(),
     b.lowerBandwidth()))
```

Table 4.3: Rules for determining the properties of the result matrix C for the addition of matrices $C \leftarrow A + B$.

A \ B	de	ba	sy	sb	ut	lt	ub	lb
de	0	0	0	0	0	0	1	1
ba	0	1	0	1	0	0	0	0
sy	0	0	0	0	0	0	0	0
sb	0	1	0	2	0	0	1	1
ut	0	0	0	0	3	0	3	1
lt	0	0	0	0	0	4	1	4
ub	0	1	0	1	3	1	5	1
lb	0	1	0	1	1	4	1	6

```
0 →  c.setDenseMatrix(a.numRows(), b.numColumns())
1 →  if (a.upperBandwidth() + b.upperBandwidth() >=
     b.nomColumns()-1 && a.lowerBandwidth() + b.lowerBandwidth()
     >= a.numRows()-1)
     { c.setDenseMatrix(a.numRows(), b.numColumns()) }
     else {c.setBandedMatrix(a.numRows(), b.numColumns(),
     Math.min(a.upperBandwidth() + b.upperBandwidth(),
     b.nomColumns()-1), Math.min(a.lowerBandwidth() +
     b.lowerBandwidth(), a.numRows()-1)) }
2 →  if (a.upperBandwidth()==0 && b.upperBandwidth()==0 &&
     a.lowerBandwidth()==0 && b.lowerBandwidth()==0)
     { c.setSymmetricBandedMatrix(a.numRows(), 0) }
     else { c.setBandedMatrix(a.numRows(), b.numColumns(),
     Math.min(a.upperBandwidth() + b.upperBandwidth(),
     b.nomColumns()-1), Math.min(a.lowerBandwidth() +
     b.lowerBandwidth(), a.numRows()-1)) }
3 →  c.setUpperTriangularMatrix(a.numRows(), b.numColumns())
4 →  c.setLowerTriangularMatrix(a.numRows(), b.numColumns())
5 →  c.setUpperTriangularBandedMatrix(a.numRows(),
     b.numColumns(), Math.min(a.upperBandwidth() +
     b.upperBandwidth(), b.nomColumns()-1))
6 →  c.setLowerTriangularBandedMatrix(a.numRows(),
     b.numColumns(),Math.min(a.lowerBandwidth() +
     b.lowerBandwidth(), a.numRows()-1))
```

Table 4.4: Rules for determining the properties of the resultant matrix C for the matrix-matrix multiplication $C \leftarrow AB$.

	df	bf	upf	lpf
de		df	df	df
ba		bf or df	bf or df	bf or df
sy		upf		upf
sb		bf or upf		bf or upf
utr		upf		upf
ltr		lpf	lpf	
utb		bf or upf		upf
ltb		bf or lpf	lpf	

Table 4.5: Storage formats transitions triggered by a new matrix property.

row the current storage format of the matrix is specified, while the new matrix properties are specified in the first column. In some cases, two different storage formats can be selected: band format or some other. As before, the band format is selected when the upper and lower bandwidths are less than half the number of columns and rows, respectively.

Since views of matrices do not have an explicit storage format, they are treated as special cases. Views are matrices that are sections of other matrices, or matrices formed by merging other matrices. When a section matrix is operated on and its property is changed, the property of the matrix of which it is a section might change and, consequently, its storage format also. These changes are performed when such situations arise. However, when the matrix that has a section is operated on, a lazy algorithm is implemented. This algorithm updates the matrix and leaves a signal for the section matrix that is not updated. Only when the section matrix is used again is its new property updated.

For a matrix formed by merging other matrices, either the matrices or the merged matrix can be operated on and their properties changed. When a merged matrix changes its properties, every matrix of which it is formed has to be adapted. However, when a matrix that forms part of the merged matrix changes its properties, a lazy implementation can again be used. The merged matrix is only updated when it is subsequently used.

Other matrix calculations, and techniques for dealing with sparse and block matrices, are implemented similarly, but following the structure predictions described in [Gil94] [Coh99].

4.5 Matrix Calculations

A matrix calculation is divided into four phases: select among the implementations with the appropriate functionality, check correctness of parameters, predict property of the result matrix, and the specialised implementation of the matrix calculation. In earlier chapters, the storage format (Section 2.3.3), matrix (Section 3.2.1) and iterator (Section 3.2.3) abstraction levels have been introduced. Matrix and iterator abstraction levels have been introduced as a way of traversing matrices, but examples of how to implement operations with them have yet to be presented. These abstraction levels (matrix and iterator) enable library developers to code fewer implementations, compared with the storage format abstraction level, but still deal with the same storage formats and matrix properties.

Matrix calculations are only defined for certain (conformable) matrices. For example, the addition of matrices is only defined for matrices that have the same numbers of rows and columns. Similarly, the matrix-matrix multiplication $C \leftarrow AB$ is only defined for A being a $n \times k$ matrix and B a $k \times m$ matrix. These tests are straightforward to implement and are, therefore, omitted in the following description. Simply note that the test is performed before any of the instructions of the matrix calculation implementation are executed. When the test is not successful an exception is raised and the matrices are left unmodified.

In traditional libraries, users have to select a subroutine that represents an implementation of the matrix calculation for matrices with certain properties and certain storage formats. Since OoLaLa encapsulates, whenever possible, the different implementations behind a unique method offered to its users, a selection algorithm is necessary. The selection algorithm varies with the abstraction level at which the matrix calculations are implemented.

4.5.1 Implementing at Different Abstraction Levels

Matrix-matrix multiplication is used to describe how matrix and iterator abstraction levels can reduce the number of implementations compared with storage format abstraction level. Among the different combinations of storage formats and matrix properties, the following combinations are selected to illustrate the abstraction levels for the operation $C \leftarrow AB$:

- A and B are both dense matrices stored in dense format,

- A is an upper triangular matrix and B is a dense matrix, both stored in dense format, and
- A is an upper triangular matrix stored in packed format and B is a dense matrix stored in dense format.

Figures 4.15 and 4.16 present implementations at storage format abstraction level. Note that these implementations use two-dimensional language arrays for the sake of clarity (on the right hand side of Figure 4.15 the same implementation is presented but the arrays are mapped into one-dimensional language arrays). Since implementation at this abstraction level requires access to the representation of the storage format, there is a different implementation for each storage format.

The matrix abstraction level is independent of the storage formats. Figure 4.17 presents the implementations for the three combinations. Only two implementations are necessary corresponding to A dense or A upper triangular, and the only difference between the implementations is the bound on the inner `i` loop.

The iterator abstraction level is also independent of the storage formats and of the nonzero element structures. Figure 4.18 presents an implementation in which the elements are accessed through the method `currentElement` and `nextElement`. Depending on the property of the matrix `nextElement` accesses different matrix positions.

4.5.2 Selecting an Implementation

The selection algorithm for implementations at storage format abstraction level checks the properties and storage formats of the matrices involved in an operation. The selection algorithm for implementations at matrix abstraction level simply checks the matrix properties. Finally, the selection algorithm for implementations at iterator abstraction level checks the mathematical relation matrix properties (symmetric, positive definite, etc.).

Recall that only certain matrix calculations have a complete decision tree and, therefore, not all matrix calculations have a selection algorithm implemented. In these cases, users pass the selection as a parameter (see direct and iterative solvers of matrix equations, in Section 3.2.4).

Some object oriented libraries follow the guidelines of BLAS and LAPACK and implement matrix calculations taking into account the properties and storage

```
double a[n][k];
double b[k][m];
double c[n][m];
double temp;
int j,l,i;

for (j=0; j!=m; j++)
{
  for (i=0; i!=n; i++)
  {
    c[i][j]=0.0;
  }// end for
}// end for
for (j=0; j!=m; j++)
{
  for (l=0; l!=k; l++)
  {
    temp=b[l][j];
    if (temp!=0.0)
    {
      for (i=0; i!=n; i++)
      {
        c[i][j]+=a[i][l]*temp;
      }// end for
    }// end if
  }// end for
}// end for
```

```
double a[n*k];
double b[k*m];
double c[n*m];
int j,l,i;
int column_c=0;
int column_a=0;
int ind_b=0;
int ind_a, ind_c;

for (ind_c=0; ind_c!=n*m; ind_c++)
{
  c[ind_c]=0.0;
}// end for
ind_c=0;
for (j=0; j!=m; j++)
{
  column_a=0;
  for (l=0; l!=k; l++)
  {
    temp=b[ind_b];
    if (temp!=0.0)
    {
      ind_a=column_a;
      ind_c=column_c;
      for (i=0; i!=n; i++)
      {
        c[ind_c]+=a[ind_c]*temp;
        ind_c++;
        ind_a++;
      }// end for
    }// end if
    column_a+=n;
    ind_b++;
  }// end for
  column_c+=n;
}// end for
```

Figure 4.15: Implementation of matrix-matrix multiplication $C \leftarrow AB$ at storage format abstraction level where A and B are dense matrices stored in dense format.

```
double a[n][k];
double b[k][m];
double c[n][m];
double temp;
int j,l,i;

for (j=0; j!=m; j++)
{
  for (i=0; i!=n; i++)
  {
    c[i][j]=0.0;
  }// end for
}// end for
for (j=0; j!=m; j++)
{
  for (l=0; l!=k; l++)
  {
    temp=b[l][j];
    if (temp!=0.0)
    {
      for (i=0; i!=l+1; i++)
      {
        c[i][j]+=a[i][l]*temp;
      }// end for
    }// end if
  }// end for
}// end for
```

```
double ap[(n*k)*(n*k)/2+k/2];
double b[k][m];
double c[n][m];
double temp;
int j,l,i;

for (j=0; j!=m; j++)
{
  for (i=0; i!=n; i++)
  {
    c[i][j]=0.0;
  }// end for
}// end for
for (j=0; j!=m; j++)
{
  for (l=0; l!=k; l++)
  {
    temp=b[l][j];
    if (temp!=0.0)
    {
      for (i=0; i!=l+1; i++)
      {
        c[i][j]+=ap[i+l*(l-1)/2]*temp;
      }// end for
    }// end if
  }// end for
}// end for
```

Figure 4.16: Implementations of matrix-matrix multiplication $C \leftarrow AB$ at storage format abstraction level where A is an upper triangular matrix stored in packed format (right) or dense format (left) and B is a dense matrix stored in dense format.

```
Matrix a=new Matrix();
Matrix b=new Matrix();
Matrix c=new Matrix();
double temp;
int j,l,i;

a.setDenseMatrix(n,k);
b.setDenseMatrix(k,m);
c.setDenseMatrix(n,k);

for (j=0; j!=m; j++)
{
  for (i=0; i!=n; i++)
  {
    c.assign(i,j,0.0);
  }// end for
}// end for
for (j=0; j!=m; j++)
{
  for (l=0; l!=k; l++)
  {
    temp=b.element(l,j);
    if (temp!=0.0)
    {
      for (i=0; i!=n; i++)
      {
        c.assign(i,j,c.element(i,j)
            +a.element(i,l)*temp);
      }// end for
    }// end if
  }// end for
}// end for
```

```
Matrix a=new Matrix();
Matrix b=new Matrix();
Matrix c=new Matrix();
double temp;
int j,l,i;

a.setUpperTriangularMatrix(n,k);
b.setDenseMatrix(k,m);
c.setDenseMatrix(n,k);

for (j=0; j!=m; j++)
{
  for (i=0; i!=n; i++)
  {
    c.assign(i,j,0.0);
  }// end for
}// end for
for (j=0; j!=m; j++)
{
  for (l=0; l!=k; l++)
  {
    temp=b.element(l,j);
    if (temp!=0.0)
    {
      for (i=0; i!=l+1; i++)
      {
        c.assign(i,j,c.element(i,j)
            +a.element(i,l)*temp);
      }// end for
    }// end if
  }// end for
}// end for
```

Figure 4.17: Implementations of matrix-matrix multiplication $C \leftarrow AB$ at matrix abstraction level where A is dense matrix (left) or upper triangular matrix (right) and B is a dense matrix.

```
Matrix a=new Matrix();
Matrix b=new Matrix();
Matrix c=new Matrix();
double atemp, btemp;
// set properties

b.setColumnWise();
a.setColumnWise();
b.begin();

while (!b.isMatrixFinished()) // for (j= ... )
{
  while (!b.isVectorFinished()) // for (l= ... )
  {
    b.nextElement();
    b.currentElement(l,btemp);
    if (btemp!=0.0)
    {
      a.beginAt(1,l);
      while (!a.isVectorFinished()) // for (i= ... )
      {
        a.nextElement();
        a.currentElement(i,atemp);
        c.assign(i,j,atemp*btemp);
      }// end while
    }// end if
  }// end while
  b.nextVector(j);
}// end while
```

Figure 4.18: Implementation of matrix-matrix multiplication $C \leftarrow AB$ at iterator abstraction level.

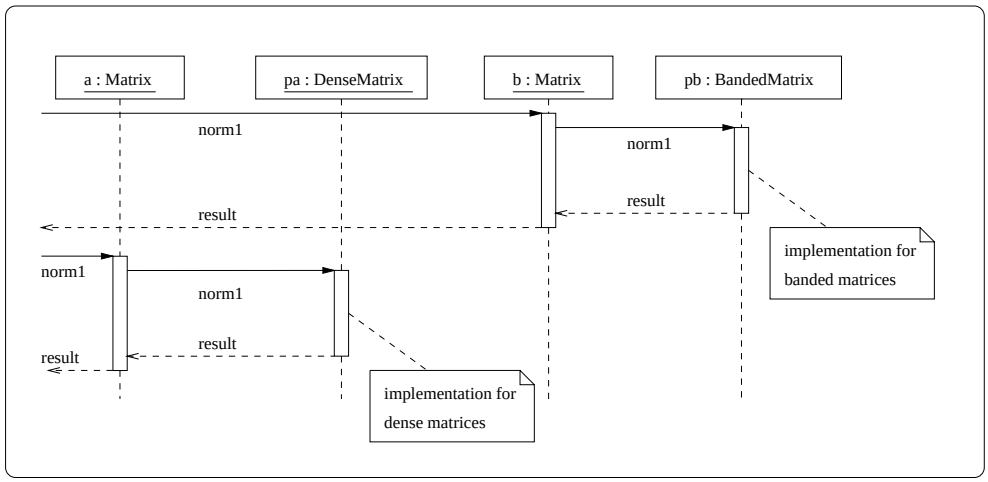


Figure 4.19: Sequence diagram of dynamic binding as a selection of `norm1` implementations.

format of only one matrix. These libraries can implement the selection algorithm implicitly using the dynamic binding mechanism provided by most object oriented languages. A simpler unary example $||A||_1$ is represented by `r=a.norm1();` where `a` is an object of class `Matrix`. The object `a` is linked with an object of a subclass of `Property`. The implementation of the selection algorithm is simply to invoke the method `norm1` in the linked object representing the property. The dynamic binding mechanism would check the class of this object and select the method that is implemented in this class. Figure 4.19 presents a sequence diagram where the method `norm1` is invoked in two matrices with different properties.

In a more general case, where binary operations are implemented using the properties and storage formats of both matrices, the selection algorithm has to be implemented explicitly.

Java offers the possibility of calling subroutines written in other languages using its *Java Native Interface*. OoLaLa can implement a selection algorithm that checks if a traditional library supports the combination of matrix properties and storage formats and then call the subroutine. Even when the combination of matrix properties and storage format is not supported it remains possible to always call traditional library subroutines. In this way, OoLaLa becomes simply a wrapper for traditional libraries; users of OoLaLa benefit from a simpler interface, and library developers can save their legacy code and concentrate on new functionality. Experiences with the Java Native Interface accessing Fortran

BLAS and LAPACK have reported similar performance to that achieved by the Fortran libraries ([BG97], [BC98], [BC99], [GFHM98], [GGMS99]).

4.6 Summary

The implementation of OoLaLa is the core of this chapter. The OoLaLa's design has been modified so that:

- the `Property` inheritance class hierarchy does not use multiple inheritance to model composed properties, such as symmetric banded;
- a version of OoLaLa is created for each numerical data type; and
- two-dimensional arrays are implemented by mapping them to one-dimensional Java language arrays.

Example programs have been presented showing how matrices and views are created and initialised. Users can specify the storage format for each matrix and can also rely on the library which can automatically select the storage format according to the matrix properties. UML object diagrams and sequence diagrams illustrate the implementations.

The management of storage formats requires the propagation of properties, and is implemented by checking the consistency between the property and storage format of a given matrix. Consistency is checked when matrices are created (users having specified a storage format) and when matrices are operated on as their properties may vary.

A matrix calculation in OoLaLa is divided into checking correctness of parameters, propagating the properties, selecting an implementation and implementing the matrix calculation. The implementation of the matrix calculation can be at storage format, matrix or iterator abstraction levels. Matrix abstraction level reduces the number of implementations, since this abstraction level is independent on storage formats. However, this abstraction level remains dependent on the matrix properties because the implementations indicate explicitly the matrix position of the elements to be accessed. The iterator abstraction level defines a matrix iterator that traverses a matrix (column-wise or row-wise) accessing the nonzero elements. A matrix iterator does not declare explicitly the elements to be accessed. Thus, for this abstraction level, the number of implementations is reduced to the number of mathematical matrix properties.

Obviously, the selection algorithm varies depending on the abstraction level at which matrix calculations are implemented. OoLaLa can become an object oriented interface, or a wrapper, of traditional libraries if the selection algorithm always selects subroutines of these libraries. OoLaLa can be also a hybrid library where some matrix calculations are implemented at iterator abstraction level while others are implemented at storage format abstraction level.

The next chapter analyses circumstances under which the propagation of properties and the management of storage formats can fail, or be inefficient. It also presents situations where users have to choose among semantically equivalent matrix calculations with different execution times. These situations are either not solved by libraries or it is unusual for libraries to address them.

Experiences of matrix properties propagation and of automatic storage format management by compilers have been reported in [Bik96], [BW96], [Mar97] and for Matlab in [GMS92]. Comparisons between implementations at iterator abstraction level and implementations at storage format abstraction level have been reported in [SL98b], [SL98c], [SL98a], [SL99], [SLL99] for MTL, which is written in C++.

Chapter 5

Limits of the Library Approach

This thesis has followed a library approach as the way of improving the development of linear algebra programs. An object oriented library that:

- encapsulates storage formats and matrices in classes,
- selects the appropriate implementation of certain matrix calculations given the properties and storage formats of the matrix operands, and
- is able to manage the storage formats and to propagate matrix properties (a novel functionality for libraries)

has been designed and described how the library could be implemented.

The objective of this chapter is to investigate the difficulties in developing the program with minimum execution time; linear algebra libraries, both traditional and object oriented, cannot solve this challenge.

The difficulties can be in one of two forms. Firstly, different semantically equivalent matrix expressions that can be implemented yielding different programs, and the execution times of these programs may be different. The term *semantically equivalent* is used since it is only when perfect floating point arithmetic is assumed that the programs are really equivalent. The equivalent expressions are obtained from the mathematical properties of the matrix operations. The commutative property of matrix addition (Section 5.1), the associative property of matrix multiplication (Section 5.2), and the distributive property of matrix multiplication are discussed.

The second difficulty is directly related to the novel functionality. Examples are presented to illustrate where a library approach cannot propagate efficiently

the properties through matrix calculations (Section 5.4). It is also described the problem of selecting the best storage format for each matrix of a program (Section 5.5).

Finally, the chapter provides an overview of a software environment for the development of linear algebra programs,i.e. a problem solving environment, that merges the library approach with techniques to address the difficulties identified (Section 5.6).

5.1 The Best Order Problem

The commutative property of matrix addition states that

$$A + B = B + A. \tag{5.1}$$

When adding 3 matrices, the commutative property yields the following identities:

$$\begin{aligned} A + B + C &= A + C + B \\ &= B + A + C \\ &= B + C + A \\ &= C + A + B \\ &= C + B + A \end{aligned} .$$

the number of different ways of representing the addition of 3 matrices is $3 \times 2 = 6$.

When the number of matrices is increased up to 4, the commutative property yields $4! = 24$ different representations (ordering of the additions). In general, when adding n matrices the commutative property yields $n!$ different representations. Users who want to develop a program that calculates the addition of n matrices can develop $n!$ different programs; each program corresponds to a different order of addition. For example, the addition $A+B+C$ can be programmed as `R=(A+B)+C` or `R=(B+C)+A` or `R=(C+B)+A` or ..., all being semantically equivalent programs. However, the execution time of each program varies depending on the order of addition and the properties of A, B and C.

For example, suppose that A and B are diagonal matrices and C is a dense matrix, and all of them are $m \times m$ matrices. A specialised program that implements `R=(A+B)+C` would use $2m$ floating point addition instructions, $3m + m^2$ memory read instructions and $2m^2$ memory write instructions. On the other

	$(A+B)+C$			$(C+A)+B$		
	`R=A+B`	`R=R+C`	Total	`R=C+A`	`R=R+B`	Total
	$\diagdown + \diagdown$	$\diagdown + \blacksquare$		$\blacksquare + \diagdown$	$\blacksquare + \diagdown$	
# add	m	m	$2m$	m	m	$2m$
# read	$2m$	m^2+m	m^2+3m	m^2+m	m^2+m	$2(m^2+m)$
# write	m^2	m^2	$2m^2$	m^2	m^2	$2m^2$

Table 5.1: Number of instructions for programs implementing $A+B+C$ and $C+A+B$, where A and B are $m \times m$ diagonal matrices ($\diagdown$) and C is a $m \times m$ dense matrix ($\blacksquare$).

hand, another specialised program which implements `R=(C+A)+B` would use the same number of instructions except that the number of memory read instructions becomes $2(m+m^2)$ (Table 5.1 shows how these counts are obtained). Assuming constant execution time for memory access, the program implemented as `R=(A+B)+C` would be faster as it executes m^2-m fewer memory read instructions.

The *best order problem* is defined as the search for the program that has minimum execution time to calculate an expression of n elements which are combined by the same commutative binary operation.

The addition of n matrices constitutes a *best order problem*, and so a search space of $n!$ possible solutions characterises the addition of n matrices.

In this case, the best order problem can be solved by first selecting the two matrices which, when added, produce a matrix with the minimum number of nonzero elements. When more than one pair of matrices produce a matrix with the minimum number of nonzero elements, the pair that collectively the smallest number of nonzero elements is selected. In this way the best order problem for n matrices is solved recursively in terms of the best order problem for $n-1$ matrices. The base case occurs when $n=2$.

This algorithm needs a mechanism to predict the number of nonzero elements for the result matrix. Table 4.3 presented the rules when dense and banded matrices are added. Different prediction algorithms can be used when sparse matrices are considered. The simplest algorithm makes the worst-case prediction, that the number of nonzero elements as the sum of the numbers of nonzero elements of the two added matrices. More sophisticated algorithms would need to exploit the specific structures of the matrices.

Note that, the best order problem cannot be solved by a library unless a subroutine (or method) is provided which implements the addition of n matrices.

This is not the usual case.

5.2 The Best Association Problem

The associative property of matrix multiplication states that

$$(AB)C = A(BC). \tag{5.2}$$

When 4 matrices are multiplied, the associative property yields

$$\begin{aligned} ((AB)C)D &= (A(BC))D \\ &= A(B(CD)) \\ &= A((BC)D) \\ &= (AB)(CD) \end{aligned} .$$

Each representation is formed dividing the 4 matrix multiplication into two subsets by introducing parenthesis (e.g. $(AB)(CD)$ or $(A)(BCD)$). When a subset has only one or two matrices, that subset is a base case. Otherwise, the subset is recursively subdivided until a base case subset is found.

Let $ANI(n)$ be the number of ways of representing the multiplication of n matrices (i.e., the association of the $n-1$ matrix multiplications). It is straightforward to show that $ANI(3) = 2$, $ANI(4) = 6$ and, in general, $ANI(n) = \sum_{i=1}^{n-1} ANI(i)ANI(n-i)$. $ANI(n)$ is known as the catalan number ([Slo73], [PB85]). Other examples of catalan numbers are $ANI(5) = 14$ and $ANI(15) = 2674440$.

Each representation is the basis of a different program, and all such programs are semantically equivalent. However, the execution time of these programs varies. The variation is due to matrix dimensions and matrix properties. For example, consider the matrix multiplication ABC where A and B are $n \times n$ dense matrices and C is a $n \times 1$ dense matrix. The association $(AB)C$ performs one matrix-matrix multiplication ($O(n^3)$ floating point operations) and one matrix-vector multiplication ($O(n^2)$ floating point operations). On the other hand, the association $A(BC)$ performs two matrix-vector multiplications ($2O(n^2)$ floating point operations).

The *best association problem*, also referred to as the chain multiplication

problem [God73], is defined as the search for the program to calculate an expression of n elements which are combined by the same binary associative and non-commutative operation.

The multiplication of n matrices constitutes a best association problem and so a search space of $ANI(n)$ (catalan numbers) possible solutions characterises the multiplication of n matrices. Algorithms to solve the best association problem can be found in [HS82] [HS84] [Coh99].

A library can only solve the best association problem if a subroutine (or method) is provided which implements the multiplication of n matrices. Again, this is not the usual case.

5.3 The Maximum Common Factor Problem

The distributive property of matrix multiplication states that

$$A(B + C) = AB + AC. \tag{5.3}$$

The right hand side of Equation 5.3 implies that the implementation would require two matrix multiplications and one addition. On the other hand, the left hand side of Equation 5.3 implies that the implementation would require one multiplication and one addition. The execution times would be significantly different and the left hand side of Equation 5.3 would be faster.

The distributive property can be generalised as

$$A(B_1 + B_2 + \cdots + B_h) = AB_1 + AB_2 + \cdots + AB_h,$$

where $A, B_1, B_2, \ldots, B_h$ are matrices or combinations of matrix calculations that produce a matrix. With this generalisation in mind, the *maximum common factor problem* is defined as finding the matrix A, so that the expression $A(B_1 + B_2 + \cdots + B_h)$ has no further common factors. That is, there is no matrix X, different from the identity matrix, such that $B_i = XY_i$ and $i = 1, 2, \ldots, h$.

Assuming a language that allows a matrix to be a variable, the maximum common factor problem can be solved applying standard compiler techniques. In the first phase, forward substitution is applied to replace variables by their current expression. This facilitates common subexpression elimination; the common expression is replaced by an appropriately initialised new variable. Finally,

```
A=C*D*H
B=C*D*J
R=A+B
```

(a) original code

```
R=C*D*H+C*D*J
```

(b) after forward substitution

```
TEMP=C*D
R=TEMP*H+TEMP*J
```

(c) after common subexpression elimination

```
R=TEMP*(H+J)
```

(d) after strength reduction

Figure 5.1: Example of applying standard compiler optimisations in order to solve the maximum common factor problem.

strength reduction optimisation exploits the distributive property of matrix multiplication to replace an expensive operation with an equivalent, but less expensive, operation. Figure 5.1 presents the effects of forward substitution, common subexpression elimination, and strength reduction in a program where the variables are matrices. The compiler optimisations above described are presented in more detail by Aho, Sethi and Ullman [ASU85].

A library can never solve the maximum common factor problem since its solution requires knowledge about the data flow in a program.

Similar situations arise when $AB^{-1}C$ or $A + B^{-1}C$ or $B^{-1}C$ need to be computed, where A, B and C are matrices or combinations of matrix calculations that produce matrices. Calculation of $B^{-1}C$ by forming the inverse matrix is known to be more time consuming than solving the system of linear equations $BX = C$ for X. The solution follows exactly the steps defined for solving the maximum common factor problem, except that the strength reduction rule is different.

A further example is the system of linear equations $A_1 A_2 \ldots A_p x = b$ where A_1, A_2, ..., A_p are square matrices. Instead of carrying out the chained matrix

multiplication, with a cost of $2n^3 + O(n^2)$ floating point operations for each multiplication, each matrix can be LU-factorised ($A_1 = L_i U_i$ and $i = 1, 2, \ldots, p$) at a cost less than or equal to $\frac{2}{3}n^3 + O(n^2)$ floating point operations per factorisation.

The work of Marsolf ([Mar97], [MGG97]) in the Falcon project uses transformation patterns for interactively restructuring Matlab's programs. Users define patterns to be found in a Matlab program and specify how the code matched with a pattern should be restructured ([Mar97] Chapter 4). These transformation patterns enable the Falcon environment to apply traditional restructuring compiler transformations ([Mar97] Chapter 5), such as loop unrolling [BGS94], and basic algebraic transformations ([Mar97] Chapter 6). Among other basic algebraic transformations, Marsolf presents a limited solution to the multiplication of n matrices (best association problem Section 5.2) and a solution to the example, where the inversion of a matrix is avoided by solving a system of linear equations, as presented above. Marsolf's solution to the multiplication of n matrices identifies the vectors and multiplies these first. However, transformation patterns cannot implement the algorithm presented in [Coh99] for the general best association problem. This algorithm uses information related to the number of nonzero elements in rows and columns and this information is not represented by the transformation patterns. Transformation patterns are able to perform the strength reductions presented in this section, but Marsolf does not show how forward substitution or common subexpression elimination can be implemented with the transformation patterns.

5.4 The Matrix Property Propagation Problem

OoLaLa is able to propagate the properties of a matrix through matrix calculations. However, a library cannot efficiently propagate matrix properties that are a consequence of the history of previous matrix calculations. For example, the matrix multiplication AB where A and B are symmetric is known to generate a dense unsymmetric result matrix. Similarly, the matrix multiplication BA also generates a dense unsymmetric matrix. Applying the rules of addition, $AB + BA$ is the addition of two dense matrices and generates a dense unsymmetric matrix. However, for A and B symmetric, $AB + BA$ is also a symmetric matrix ($AB + (AB)^T = AB + BA$).

In order to address this problem, a library would have to keep a history for

each matrix. This history would record the matrix calculations that have been carried out on each matrix and the parameters' matrix properties of those matrix operations. On the other hand, a compiler is able to identify these situations as long as they can be specified by a set of if-then rules. The implementation is similar to how a compiler checks the type of an expression; when it detects an incorrect type, it sends an error message. Similarly, the compiler is checking an expression of matrices and detects a special situation. Instead of sending an error message, the compiler changes the matrix properties of the expression. For a more technical approach to these compiler techniques consult [ASU85] Chapters 4 and 5.

Despite the fact that Marsolf's work ([Mar97], [MGG97]) in the Falcon environment and Bik and Wijshoff's Sparse Compiler ([Bik96], [BW96], [BBKW98], [BW99]) propagate matrix properties, they do not identify this problem or present any solution.

5.5 The Best Storage Format Problem

Matrices can be stored in different storage formats. Table 4.2 presents the advisable combinations of matrix properties and storage formats in the context of OoLaLa. The storage format influences the execution time of implementations of matrix operations and it determines the memory position where each element of a matrix is kept. An implementation of a matrix calculation determines a logical access pattern to the matrix elements, which is mapped to a physical access pattern to the memory. When the storage format is changed, the logical access pattern to the elements of a matrix remains unchanged, but the physical access pattern varies. Different physical access patterns have different rates of cache reuse. Consider, for example, the well-known case of arrays stored row-wise or column-wise. For this case, compiler optimisation techniques have been developed to modify the loops so that an array is traversed in the order it is stored in memory [BGS94].

OoLaLa enables users to abstract their programs from the storage formats and from how the matrix properties are propagated through matrix calculations. Hence, the structure of a linear algebra program is divided into two parts. The first part of the program declares the input matrices and their matrix properties (optionally their storage formats). In the second part, the matrices are operated

and auxiliary matrices are created to hold intermediate or final results. Before each matrix calculation is performed, the storage format of the associated matrices can be changed. These storage format changes could be represented by invocations of mapping methods. These methods would map from a current storage format of a matrix to a specified new storage format. These mapping methods can be inserted at any point of the program and the semantics of the program remains unchanged. The program produces the same result (assuming perfect arithmetic) independently of the number and the location in the program of the mapping methods. The visible effects of mapping methods are the execution time and memory requirement. The execution time decreases when the time of executing the mapping methods added to the time of executing the matrix calculations with the new storage formats is less than the time of executing the matrix operations with the previous storage formats; otherwise the execution time increases (or remains unchanged).

The *best storage format problem* is defined as the search for the linear algebra program with the minimum execution time among those programs with equivalent functionality but with different storage formats.

In general, the solution of the best storage format problem is computationally infeasible [Mac87]. Bik and Wijshoff have proposed an heuristic to automatically select the storage format [BW96]; this heuristic is integrated with their Sparse Compiler and, since it requires knowledge of the instruction flow, it cannot be included in any library.

5.6 Overview of a Linear Algebra Problem Solving Environment

Previous sections have presented problems or limits associated with linear algebra libraries. Some of these, for example the best order and the best association problems, can be solved within a library, but this is unusual. The other problems, the maximum common factor, the matrix properties propagation and the best storage format, can only be approached at compile time. These problems motivate a move from linear algebra libraries to *problem solving environments*. A problem solving environment is software, often with graphical user interfaces, which enables users to develop programs using as the programming language the problem domain language. A problem solving environment integrates domain

specific libraries, compiler techniques, artificial intelligence, visualisation and any other computer science discipline that may help users in developing their programs [GHR94].

A linear algebra problem solving environment should provide support for, and encapsulate, the different tasks that users have to perform when developing a linear algebra program, namely:

(a) describe the problem in terms of matrix calculations,

(b) analyse the matrices to determine their properties,

(c) select a library or libraries which support the calculations and properties,

(d) map the matrix calculations into the implementations provided by the library,

(e) analyse how the matrix properties are propagated through the matrix operations,

(f) declare the variables conforming to the storage format which is supported by the selected implementations,

(g) select the best combination of preconditioner and iterative solver for a given system of linear equations, and

(h) select the best ordering algorithm for a direct solver for a given sparse system of linear equations.

OoLaLa has encapsulated tasks (c) and (f), and, partially, tasks (d) and (e). This chapter has presented examples of how to help users to efficiently map their matrix calculations into matrix implementations provided by libraries, i.e. task (d). To this end, matrix operation properties have been presented as a way of describing different semantically equivalent programs but with different execution times. Solutions of the best association problem are proposed by Hu and Ching [HS82][HS84] and by Cohen [Coh99].

The basis for the solution of the maximum common factor problem is based on standard compiler optimisation techniques applied to variables of type matrix. Marsolf [Mar97] partially implements some of the solution of the maximum common factor problem together with solutions to other related problems' based on strength reduction.

The solution of propagating matrix properties through more than one operation at each time, i.e. task (e), is based on syntax directed translation [ASU85], a standard compiler technique to parse programming languages.

Automatic detection of nonzero structure, i.e. task (b), has been addressed by Bik and Wijshoff [BW99]. They have also proposed a heuristic for solving the best storage format problem [BW96].

The selection of the best combination of preconditioner and iterative solver, i.e. task (g), together with the best ordering algorithm, i.e. task (h), for sparse systems of linear equations, remain as open research problem.

Chapter 6

Conclusions

Object oriented linear algebra libraries are proposed as a way of improving the development process for linear algebra programs. Object oriented software construction offers linear algebra abstraction and encapsulation of implementation details. Designs for traditional linear algebra libraries are dominated by implementation details which are visible to the users. As a consequence, the intellectual distance between a linear algebra description of a problem and its description with traditional libraries is too large.

An object oriented analysis and design of a linear algebra library has been conducted, and, as a result, different object oriented models have been proposed. These models serve to classify a set of object oriented linear algebra libraries. The object oriented model accepted has features not found in other libraries, and it enables functionality previously reserved for compilers. Based on the reviewed object oriented libraries and on the conducted analysis and design, a library interface has been proposed for basic matrix operations and for the solution of matrix equations. The object oriented model, the increased functionality and the interface constitute the design of a new object oriented library.

Libraries offer limited help in developing a linear algebra program; they cannot identify a sequence of calls (or invocations) and match this with a different but semantically equivalent (assuming perfect floating point arithmetic) sequence of calls that could be less time consuming. This thesis has analysed and identified some of these limits.

The following section explains the above in more detail. The chapter ends with an evaluation of the limitations of the work presented (Section 6.2) and suggestions for future work (Section 6.3).

6.1 Summary

The numerical linear algebra community has analysed matrices and their calculations in order to find characteristics, i.e. matrix properties, which can be exploited by the implementations of the operations in order to reduce their execution times. Matrix properties are characteristics of matrix structures that arise repeatedly in linear algebra problems. Some of the matrix properties also enable matrices to be stored in compressed forms (e.g. for a sparse matrix that has 10% of nonzero elements). The algorithms that exploit the properties and use the data structures have been implemented in Fortran 77 as subroutines and these subroutines have been grouped into libraries, traditional libraries. For each algorithm there are as many different implementations as different combinations of advisable storage formats for the matrix parameters, and the number of algorithms is related to the number of combinations of properties for the matrix parameters. Thus, traditional libraries developers experience an explosion in the number of implementations and they have to choose which of the different possibilities are implemented.

The matrix calculations are divided into two groups: basic matrix operations and solution of matrix equations, some of which have rule based reasoning systems. These reasoning systems can be implemented as a set of "if-then" rules based on the properties and storage format of the matrices and decide the appropiate implementation for a matrix calculation.

When developing a linear algebra program with traditional libraries, the nontrivial tasks that have to be performed are:

- analysis of the properties of matrices,
- selection of the storage formats, and
- selection of the subroutines that deliver the minimum execution time.

Building on a review of existing object oriented linear algebra libraries a new class structure (see Figures 3.16 and 3.17) has been designed. This class structure enables a library to manage the storage formats and to propagate the matrix properties; a novel functionality for linear algebra libraries. In this way, matrices can transparently vary their properties and storage formats when they are operated on. This class structure and a proposed library interface constitute the

design of a new library known as the Object Oriented Linear Algebra LibrAry (OoLaLa).

Developers of traditional libraries have benefited from two abstraction levels at which matrix calculations can be implemented. These abstraction levels reduce the number of implementations. The matrix abstraction level enables matrices to be represented and accessed independently of their storage formats and the iterator abstraction level provides an implicit way of traversing matrices.

A matrix calculation in OoLaLa is divided into checking the correctness of the parameters, propagating the properties, selecting an implementation and implementing the matrix calculation. The implementation of the matrix calculation can be at storage format, matrix or iterator abstraction levels.

Obviously, the selection algorithm varies depending on the abstraction level at which matrix calculations are implemented. OoLaLa can become an object oriented interface, or a wrapper, of traditional libraries if the selection algorithm selects always subroutines of these libraries. OoLaLa can also be a hybrid library where some matrix calculations are implemented at iterator abstraction level while others are implemented at storage format abstraction level.

The thesis concludes by identifying difficulties in developing a linear algebra program with minimum execution time that linear algebra libraries, both traditional and object oriented, cannot solve, and suggest that a problem solving environment [GHR94] might overcome the difficulties.

6.2 Critique

The main omission from the thesis is that it has not been possible to address the question of how much performance is lost by implementing matrix calculations at matrix and iterator abstraction levels compared with traditional libraries' implementations at storage format abstraction level. However, the main objective was to create an object oriented design of linear algebra. Due to time constraints, it has not been possible to implement fully this design.

Further, object oriented libraries have been justified because they are easier to use than traditional libraries, and this has been supported by clear arguments. However, a more scientific approach would have used *metrics* defined by the software engineering community to justify this claim.

6.3 Future Work

The previous section summarises the immediate future work: an evaluation of the performance lost when implementing matrix calculations at matrix and iterator abstraction levels compared with traditional libraries' implementations at storage format abstraction level. Hybrid libraries, where some matrix calculations are implemented at storage format abstraction level and others at iterator abstraction level, pose the further question – which matrix calculations should be implemented at which abstraction level in order to minimise execution time.

Another performance question is whether block algorithms and recursive algorithms ([WD98], [AGK+99]) currently used in implementations at storage format abstraction level will reduce the execution time of implementations at iterator and matrix abstraction levels.

This thesis has concentrated on sequential linear algebra programs. A logical extension is to design and implement OoLaLa for parallel programs. Among others, the issues that need to be addressed are:

- threads sharing objects versus objects communicating and synchronising by remote method invocation,
- the way in which users take part in the parallelisation process of a linear algebra program,
- the performance comparison of parallel implementations at iterator and matrix abstraction levels with implementations at storage format abstraction level,
- the performance of compilers at parallelising implementations at iterator and matrix abstraction levels, ...

A long-term objective is the implementation of the outlined linear algebra problem solving environment for sequential and parallel programs is the objective. The implementation includes the improvement of current solutions to the tasks:

1. analysis of matrices to determine their properties ([Bik96], [BW99]);
2. mapping the matrix calculations into the implementations provided by libraries – ([Mar97], [MGG97]);

3. analysis of propagation of matrix properties through matrix operations, – ([Bik96], [BBKW98], [Mar97], [MGG97]);

4. selection of the best combination preconditioner and iterative solver for a given system of linear equations (open problem); and

5. selection of the best ordering algorithm for a direct solver for a given sparse system of linear equations (open problem).

Bibliography

[Abb83] Rossell J. Abbot. Program design by informal English descriptions. *Communications of the ACM*, 26(11):882–894, 1983.

[ABD+95] E. Anderson, Z. Bai, C. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostouchov, and S. Sorensen. *LAPACK User's Guide*. SIAM Press, 2^{th} edition, 1995.

[Abr80] Jean-Raymond Abrial. The specification language Z: Syntax and "semantics". Technical report, Oxford University Computing Laboratory, Programming Research Group, 1980.

[ACMW99] Benjamin A. Allan, Robert L. Clay, Kyran D. Mish, and Alan B. Williams. *ISIS++ Reference Guide: Iterative Scalable Implicit Solver in C++ version 1.1*. Sandia National Laboratories Livermore, 1999.

[AF95] Niclas Andersson and Peter Fritzson. Generating parallel code from object oriented mathematical models. In *Proceedings of the* 5^{th} *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 48–57, 1995.

[AFM97] O. Agesin, S. Freund, and J. Mitchell. Adding type parameterization to the Java language. In *Proceedings of the Symposium on Object Oriented Programming: Systems, Languages and Applications*, pages 49–65, 1997.

[AG99] Cleve Ashcraft and Roger Grimes. SPOOLES: An object-oriented sparse matrix library. In *SIAM Conference on Parallel Processing for Scientific Computing*, 1999.

[AGK+99] Bjarne Stig Andersen, Fred Gustavson, Alexander Karaivanov, Jerzy Wasniewski, and Plamen Y. Yalamov. Lawra – linear algebra with recursive algorithms. In *Proceedings of the Conference on Parallel Processing and Applied Mathematics*, 1999.

[Åhl95] Krister Åhlander. An object-oriented approach to construct pde solvers. Technical Report 197, Department of Scientific Computing, Uppsala University, 1995.

[AL96] Cleve Ashcraft and Joseph W. H. Liu. *SMOOTH: A Software Package For Ordering Sparse Matrices*, November 1996.

[ANS83] ANSI (American National Standards Institute) and US Goverment Deparment of Defense. *Ada Joint Program Office: Military Standard – Ada Programming Language*, 1983.

[AR94] Howard Anton and Chris Rorres. *Elementary Linear Algebra: Applications Versions.* John Wiley & Sons, 7^{th} edition, 1994.

[Ara89] G. Arango. Domain analysis: From art to engineering discipline. *SISOFT Engineering Notes*, 14(3), 1989.

[ASM80] Jean-Raymond Abrial, Stephen A. Schuman, and Bertran Meyer. A specification language. In R. McNaughten and R.C. Mckeag, editors, *On the Construction of Programs.* Cambridge University Press, 1980.

[ASU85] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers. Principles, Techniques and Tools.* Addison Wesley, 1985.

[Aus98] Matthew H. Austern. *Generic Programming and the STL: Using and Extending the C++ Standard Template Library.* Addison Wesley, 1998.

[Axe94] Owe Axelsson. *Iterative Solution Methods.* Cambridge University Press, 1994.

[BBC+94] Richard Barrett, Michael Berry, Tony Chan, James Demmel, June Donato, Jack J. Dongarra, Voctor Eijkhout, Roldan Pozo, Charles Romine, and Hank van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, 1994.

[BBKW98] Aart J. C. Bik, Peter J. H. Brinkhaus, Peter M. W. Knijnenburd, and Harry A.G. Wijshoff. The automatic generation of sparse primitives. *ACM Transactions on Mathematical Software*, 24(2):190–225, June 1998.

[BBV+99] Lubomir Birov, Yuri Bartenev, Anatoly Vargin, Avijit Purkayastha, Anthony Skjellum, Yoginder Dandass, and Purushotham Bangalore. The parallel mathematical libraries project (PMLP) – a next generation scalable sparse object oriented mathematical library suite. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, March 1999.

[BC98] Brian Blount and Siddhartha Chatterjee. An evaluation of Java for numerical computing. In Denis Caromel, Rodney R. Oldehoeft, and Marydell Tholburn, editors, *Computing in Object-Oriented Parallel Environments, Second International Symposium ISCOPE 98*, number 1505 in Lecture Notes in Computer Science, pages 35–46. Springer-Verlag, 1998.

[BC99] Brian Blount and Siddhartha Chatterjee. An evaluation of Java for numerical computing. *Scientific Programming*, 7(2):97–110, 1999. Special Issue: High Performance Java Compilation and Runtime Issues.

[BCHQ97] David L. Brown, Geoffrey S. Chesshire, William D. Henshaw, and Daniel J. Quinlan. OVERTURE: An object oriented software system for solving partial differential equations in serial and parallel environments. In *Proceedings of the Eigth SIAM Conference on Parallel Processing for Scientific Computing*, 1997.

[BDD+95] Zhaojun Bai, David Day, James Demmel, Jack Dongarra, Ming Gu, Axel Ruhe, and Henk van der Vorst. Templates for linear algebra problems. *Lecture Notes in Computer Science*, 1000:115–140, 1995.

[BDH+98] David L. Brown, Kei Davis, William D. Henshaw, Daniel J. Quinlan, and Kristi Brislawn. OVERTURE: Object-oriented parallel adaptive mesh refinement for serial and parallel environments. In S. Demeyer and J. Bosch, editors, *Object-Oriented Technology* –

ECOOP'98 Workshop Reader, volume 1543 of *Lecture Notes in Computer Science*, pages 446–447. Springer-Verlag, 1998. Workshop on Parallel Object-Oriented Scientific Computing.

[BG97] Aart J. C. Bik and Dennis B. Gannon. A note on native level 1 BLAS in Java. *Concurrency: Practice and Experience*, 9(11):1091–1099, 1997. Special Issue: Java for Computational Science and Engineering — Simulation and Modelling II.

[BGMS97] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhauser Press, 1997.

[BGMS99] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. PETSc 2.0 users manual. Technical Report ANL-95/11 - Revision 2.0.24, Argonne National Laboratory, 1999.

[BGS94] David F. Bacon, Susan L. Graham, and Oliver J. Sharp. Compiler transformations for high-performance computing. *Computing Surveys*, 26(4):345–420, 1994.

[BHQ98] David L. Brown, William D. Henshaw, and Daniel J. Quinlan. OVERTURE: An object-oriented framework for solving partial differential equations. In Yutaka Ishikawa, Rodney R. Oldehoeft, John V.W. Reynders, and Marydell Tholburn, editors, *Scientific Computing in Object-Oriented Parallel Environments, First International Conference ISCOPE 97*, volume 1343 of *Lecture Notes in Computer Science*, pages 177–184. Springer-Verlag, 1998.

[BHQ99] David L. Brown, William D. Henshaw, and Daniel J. Quinlan. OVERTURE: An object-oriented framework for solving partial differential equations on overlapping grids. In Michael E. Henderson, Christopher R. Anderson, and Stephen L. Lyons, editors, *Object Oriented Methods for Interoperable Scientific and Engineering Computing*, SIAM Proceedings in Applied Mathematics, 1999. Proceedings of SIAM Workshop on Object Oriented Methods for Interoperable Scientific and Engineering Computing, October 1998.

[Bik96] Aart J. C. Bik. *Compiler Support for Sparse Matrix Computations.* PhD thesis, Department of Computer Science, Leiden University, 1996.

[BK99a] Zoran Budimlić and Ken Kennedy. The cost of being object-oriented: A preliminary study. *Scientific Programming*, 7(2):87–96, 1999. Special Issue: High Performance Java Compilation and Runtime Issues.

[BK99b] Zoran Budimlić and Ken Kennedy. Prospects for scientific computing in polymorphic object-oriented style. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, March 1999.

[BKP98] Zoran Budimlić, Ken Kennedy, and Jeff Piper. The cost of being object-oriented: A preliminary study. In *Workshop for Java for High Performance Network Computing at EUROPAR'98*, 1998.

[BL97] Are Magnus Bruaset and Hans Petter Langatangen. Object-oriented design of preconditioned iterative methods in Diffpack. *ACM Transactions on Mathematical Software*, 23(1):50–80, 1997.

[BLA99] BLAS Technical Forum. *Document for the Basic Linear Algebra Subprograms Standard*, August 1999. Draft.

[BML97] Josehp A. Bandk, Andrew C. Myers, and Barbara Liskov. Parametized types for Java. In *Proceeding of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 132–145, 1997.

[Boo94] Grady Booch. *Object-oriented analysis and design with applications.* Benjamin Cummings, 1994.

[BPB+99] Lubomir Birov, Arkady Prokofiev, Yuri Bartenev, Anatoly Vargin, Avijit Purkayastha, Yoginder Dandass, Vladimir Erzunov, Elena Shanikova, Anthony Skjellum, Purushotham Bangalore, Eugeny Shuvalov, Vitaly Ovechkin, Nataly Frolova, Sergey Orlov, and Sergey Egorov. The parallel mathematical libraries project (PMLP): Overview, innovations and design issues. In V. Malyshkin, editor, *Fifth International Conference on Parallel Computing Technologies*

– *PaCT'99*, number 1662 in Lecture Notes in Computer Science. Springer-Verlag, 1999.

[BPK] Computer Science Deparment, University of Minnesota and Minnesota Supercomputer Institute, and Mathematical Algorithms and Scalable Computing Group, SGI/Cray Research, Inc. *Block Preconditioning ToolKit (BPKIT) web page.* http://www.cs.umn.edu/ chow/bpkit.html.

[BRJ99] Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide.* Addison-Wesley, 1999.

[BW96] Aart J. C. Bik and Harry A. G. Wijshoff. Automatic data structure selection and transformation for sparse matrix computations. *IEEE Transactions on Parallel and Distributed Systems*, 7(2):109–126, 1996.

[BW99] Aart J. C. Bik and Harry A. G. Wijshoff. Automatic nonzero structure analysis. *SIAM Journal of Computing*, 28(5):1576–1587, 1999.

[CCH+99] Julian C. Cummings, James A. Crotinger, Scott W. Haney, Willian F. Humphrey, Steve R. Karmesin, John V.W. Reynders, Stephen A. Simith, and Timothy J. Williams. Rapid application development and enhanced code interoperability using the POOMA framework. In Michael E. Henderson, Christopher R. Anderson, and Stephen L. Lyons, editors, *Object Oriented Methods for Interoperable Scientific and Engineering Computing*, SIAM Proceedings in Applied Mathematics, 1999. Proceedings of SIAM Workshop on Object Oriented Methods for Interoperable Scientific and Engineering Computing, October 1998.

[CH96] Edmond Chow and Michael A. Heroux. Block preconditioning toolkit reference manual. Technical Report UMSI 96/183, University of Minnesota Supercomputing Institute, September 1996.

[CH98] Edmond Chow and Michael A. Heroux. An object-oriented framework for block preconditioning. *ACM Transactions on Mathematical Software*, 24(2):159–183, 1998.

[Cog] Software tools for High-Performance Computing, Department of Scientific Computing, Uppsala University. *Cogito project web page.* http://www.tdb.uu.se/research/swtools/cogito.html.

[Coh99] Edith Cohen. Structure prediction and computation of sparse matrix products. *Journal of Combinatorial Optimization*, 2(4):307–332, 1999.

[DBMS79] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart. *LINPACK Users' Guide.* SIAM Press, 1979.

[DCHD90] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, and Iain Duff. A set of level 3 basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 16:1–17, 1990.

[DCHH88a] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, and Richard J. Hanson. Algorithm 656: An extended set of FORTRAN basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 14:18–32, 1988.

[DCHH88b] Jack J. Dongarra, Jeremy Du Croz, Sven Hammarling, and Richard J. Hanson. An extended set of FORTRAN basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 14:1–17, 1988.

[DDS99] David M. Dooling, Jack Dongarra, and Keith Seymour. JLAPACK – compiling LAPACK FORTRAN to Java. *Scientific Programming*, 7(2):111–138, 1999. Special Issue: High Performance Java Compilation and Runtime Issues.

[DER86] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices.* Oxford Science Publications, 1986.

[Dif] Numerical Objects A.S. *Diffpack web page.* http://www.nobjects.com.

[Dij79] E. Dijkstra. Programming as a human activity. In *Classics in Software Engineering.* Yourdon Press, 1979.

[DKP98] Florin Dobrian, Gary Kumfert, and Alex Pothen. Object-oriented design for sparse direct solvers. In Denis Caromel, Rodney R. Oldehoeft, and Marydell Tholburn, editors, *Computing in Object-Oriented Parallel Environments, Second International Symposium ISCOPE 98*, number 1505 in Lecture Notes in Computer Science, pages 207–214. Springer-Verlag, 1998.

[DKP99] Florin Dobrian, Gary Kumfert, and Alex Pothen. The design of sparse direct solvers using object-oriented techniques. In A. M. Bruaset, H. P. Langtangen, and E. Quak, editors, *Advances in Software Tools for Scientific Computing*, volume 10 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 1999.

[DLN+94] Jack Dongarra, Andrew Lumsdaine, Xinhui Niu, Roldan Pozo, and Karin Remington. Sparse matrix libraries in C++ for high performance architectures. In *Proceedings of the Conference on Object Oriented Numerics OON-SKI*, pages 122–138, 1994.

[DLPR96] Jack Dongarra, Andrew Lumsdaine, Roldan Pozo, and Karin A. Remington. *IML++ v. 1.2: Iterative Methods Library Reference Guide*, 1996.

[DNS97a] Viktor K. Decyk, Charles D. Norton, and Boleslaw K. Szymanski. Expressing object-oriented concepts in Fortran 90. *ACM Fortran Forum*, 16(1):13–18, 1997.

[DNS97b] Viktor K. Decyk, Charles D. Norton, and Boleslaw K. Szymanski. How to express C++ concepts in Fortran 90. *Scientific Programming*, 6(4):363–390, 1997.

[DNS98] Viktor K. Decyk, Charles D. Norton, and Boleslaw K. Szymanski. How to support inheritance and run-time polymorphism in Fortran 90. *Computer Physics Communications*, 115:9–17, 1998.

[DPW93a] Jack J. Dongarra, Roldan Pozo, and David W. Walker. LAPACK++: A design overview of object-oriented extensions for high performance linear algebra. In *Proceedings of Supercomputing '93*, pages 162–171. IEEE Computer Society Press, 1993.

[DPW93b] Jack J. Dongarra, Roldan Pozo, and David W. Walker. An object oriented design for high performance linear algebra on distributed memory architectures. In *Proceedings of the Conference on Object Oriented Numerics OON-SKI*, 1993.

[DPW96] Jack Dongarra, Roldan Pozo, and David Walker. *LAPACK++ v. 1.1: High Performance Linear Algebra Users' Guide*, April 1996.

[Dub97] Paul F. Dubois. *Object Technology for Scientific Computing.* Prentice-Hall, 1997.

[Duf77] Iain S. Duff. MA28 : A set of FORTRAN subroutines for sparse unsymmetric linear equations. Technical Report R-8730, HMSO, AERE Harwell Laboratory, 1977.

[DW95] Jack J. Dongarra and David W. Walker. Software libraries for linear algebra computations on high performance computers. *SIAM Review*, 37(2):151–180, June 1995.

[EGSS82] S. C. Eisenstat, M. C. Gursky, M. H. Schultz, and A. H. Sherman. Yale sparse matrix package. *International Journal of Numerical Methods for Engineering*, pages 1145–1151, 1982.

[FA93] Peter Fritzson and Niclas Andersson. Generating parallel code from equations in the objectmath programming environments. In Jens Volkert, editor, *Parallel Computation, Second International ACPC Conference*, volume 734 of *Lecture Notes in Computer Science*, pages 219–232. Springer-Verlag, 1993.

[FE98] Peter Fritzson and Vadim Engelson. Modelica - a unified object-oriented language for system modeling and simulation. In Eric Jul, editor, *ECOOP'98 – Object-Oriented Programming* 12^{th} *European Conference*, volume 1445 of *Lecture Notes in Computer Science*, pages 67–90. Springer-Verlag, 1998.

[FEV93] Peter Fritzson, Vadim Engelson, and Lars Viklund. Variant handling, inheritance and composition in the ObjectMath computer algebra environment. In Alfonso Miola, editor, *Design and Implementation of Symbolic Computation Systems*, volume 722 of *Lecture Notes in Computer Science*, pages 145–160. Springer-Verlag, 1993.

[FVHF92] Peter Fritzson, Lars Viklund, Johan Herber, and Dag Fritzson. Industrial application of object-oriented mathematical modeling and computer algebra in mechanical analysis. In Georg Heeg, Boris Magnusson, and Bertrand Meyer, editors, *Technology of Object-Oriented Languages and Systems – TOOLS 7*, pages 167–181. Prentice Hall, 1992.

[FVHF95] Peter Fritzson, Lars Viklund, Johan Herber, and Dag Fritzon. High-level mathematical modeling and programming. *IEEE Software*, 12(4):77–87, 1995.

[Gan59a] F. R. Gantmacher. *The Theory of Matrices Vol. 1.* Chelsea, 1959.

[Gan59b] F. R. Gantmacher. *The Theory of Matrices Vol. 2.* Chelsea, 1959.

[GBDM77] B. S. Garbow, J. M. Boyle, J. J. Dongarra, and C. B. Moler. *Matrix Eigensystem Routines: EISPACK Guide Extension*, volume 51 of *Lecture Notes in Computer Science*. Springer-Verlag, 1977.

[GFHM98] Vladimir Getov, Susan Flynn-Hummel, and Sava Mintchev. High-performance parallel programming in Java: Exploiting native libraries. *Concurrency: Practice and Experience*, 10(11):863–872, 1998.

[GGMS99] Vladimir Getov, Paul Gray, Sava Mintcheva, and Vaidy Sunderam. Multi-language programming environments for high performance Java computing. *Scientific Programming*, 7(2):139–146, 1999. Special Issue: High Performance Java Compilation and Runtime Issues.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johson, and John Vlissides. *Design Patterns: Elements of Reusable Object Oriented Software.* Addison Wesley, 1995.

[GHR94] Estratis Gallopoulos, Elias N. Houstis, and John R. Rice. Computer as thinker/doer: Problem solving environments for computational science. *IEEE Computational Science Engineering Magazine*, 1(2):11–23, 1994.

[Gil94] John R. Gilbert. Predicting structure in sparse matrix computations. *SIAM Journal of Matrix Analysis and Applications*, 15(1):62–79, 1994.

[GJ95] F. Guidec and J. M. Jézéquel. Polymorphic matrices in paladin. In *Workshop on Object-based Parallel and Distributed Computation OBPDC*, Lecture Notes in Computer Science. Springer-Verlag, 1995.

[GJP96] F. Guidec, J. M. Jézéquel, and J. L. Pacherie. An object-oriented framework for supercomputing. *Systems and Software*, June 1996. Special issue on Software Engineering for Distributed Computing.

[GL79] Alan George and Joseph W. H. Liu. The design of a user interface for a sparse matrix package. *ACM Transactions on Mathematical Software*, 5:134–162, 1979.

[GL81] Alan George and Joseph W. H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice Hall, 1981.

[GL99] Alan George and Joseph W. H. Liu. An object-oriented approach to the design of a user interface for a sparse matrix package. *SIAM Journal of Matrix Analisys and Applications*, 20(4):953–969, 1999.

[GMS92] John R. Gilbert, Cleve Moler, and Robert Schereiber. Sparse matrices in Matlab: Design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, 1992.

[God73] Sadashiva S. Godbole. On efficient computation of matrix chain products. *IEEE Transactions on Computer*, C-22(9):864–866, 1973.

[Gol91] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

[GvL96] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. John Hopkins University Press, 3^{th} edition, 1996.

[HC99] Scott Haney and James Crotinger. How templates enable high-performance scientific computing in C++. *IEEE Computing in Science and Engineering*, 1(4):66–72, 1999.

[Hig96] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM Publications, 1996.

[HKBR98] William Humphrey, Steve Karmesin, Federico Basseti, and John Reynders. Optimization of data-parallel field expressions in the POOMA framework. In Yutaka Ishikawa, Rodney R. Oldehoeft, John V.W. Reynders, and Marydell Tholburn, editors, *Scientific Computing in Object-Oriented Parallel Environments, First International Conference ISCOPE 97*, number 1343 in Lecture Notes in Computer Science, pages 184–194. Springer-Verlag, 1998.

[HRC+98] William Humphrey, Robert Ryne, Timothy Cleand, Julian Cummings, Salman Habib, Graham Mark, and Ji Qiang. Particle beam dynamics simulations using the POOMA framework. In Denis Caromel, Rodney R. Oldehoeft, and Marydell Tholburn, editors, *Computing in Object-Oriented Parallel Environments, Second International Symposium ISCOPE 98*, number 1505 in Lecture Notes in Computer Science, pages 25–34. Springer-Verlag, 1998.

[HS82] Te Chiang Hu and M. T. Shing. Computation of matrix chain products. Part I. *SIAM Journal on Computing*, 11(2):362–373, 1982.

[HS84] Te Chiang Hu and M. T. Shing. Computation of matrix chain products. Part II. *SIAM Journal on Computing*, 13(2):228–251, 1984.

[IML] *Iterative Methods Library (IML++) library web page.* http://math.nist.gob/iml++/.

[ISI] Distributed Applications Research Deparment, Sandia National Laboratories. *Iterative Scalable Implicit Solver in C++ (ISIS++) web page.* http://z.ca.sandia.gov/isis.

[ITL] Laboratory for Scientific Computing, University of Notre Dame. *Iterative Template Library (ITL) web page.* http://www.lsc.nd.edu/research/itl.

[Jama] Department of Computer Science, University of Maryland and Mathematical and Computations Sciences Division, NIST. *Jampack library web page.* ftp://math.nist.gov/pub/Jampack/Jampack.html.

[JAMb] Mathematical and Computations Sciences Division, NIST and The MathWorks. *JAMA library web page*. http://math.nist.gov/jama/.

[Jav98] Java Grande Forum. *Making Java Work for High-End Computing*, November 1998. available at http://www.javagrande.org/reports.htm.

[Jav99] Java Grande Forum. *Interim Java Grande Forum Report*, June 1999. available at http://www.javagrande.org/reports.htm.

[JCJO92] Ivar Jacobson, Magnus Christenson, Patrik Johnson, and Gunnar Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison Wesley, 1992.

[JLA] Department of Computer Science, University of North Carolina. *JLAPACK library web page*. http://www.cs.unc.edu/Research/HARPOON/jlapack.

[KCC+98] Steve Karmesin, James Crotinger, Julian Cummings, Scott Haney, William Humphrey, John Reynders, Stephen Smith, and Timothy Williams. Array design and expression evaluation in POOMA II. In Denis Caromel, Rodney R. Oldehoeft, and Marydell Tholburn, editors, *Computing in Object-Oriented Parallel Environments, Second International Symposium ISCOPE 98*, number 1505 in Lecture Notes in Computer Science, pages 231–238. Springer-Verlag, 1998.

[KP98] Gary Kumfert and Alex Pothen. An object-oriented collection of minimum degree algorithms. In Denis Caromel, Rodney R. Oldehoeft, and Marydell Tholburn, editors, *Computing in Object-Oriented Parallel Environments, Second International Symposium ISCOPE 98*, number 1505 in Lecture Notes in Computer Science, pages 95–106. Springer-Verlag, 1998.

[Kru95] Philippe Kruchten. The "4+1" view model of software architecture. *IEEE Software*, 12(6):42–50, November 1995.

[LAB+81] Barbara H. Liskov, Russel Atkinson, T. Bloom, E. Moss, J. Craig Schaffert, R. Scheiffer, and Alan Snyder. *CLU Reference Manual*. Springer-Verlag, 1981.

[Lan99] Hans Petter Langtangen. *Computational Partial Differential Equations, Numerical Methods and Diffpack Programming*, volume 2 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 1999.

[LAP] *LAPACK++ library web page*. http://math.nist.gov/lapack++/.

[LHKK79] C. L. Lawson, R. J. Hanson, D. Kincais, and F. T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Transactions on Mathematical Software*, 5:308–323, 1979.

[Liu90] Joseph W. H. Liu. The role of elimination trees in sparse factorization. *SIAM Journal of Matrix Analysis and Applications*, 11(1):134–172, 1990.

[LS95] Meng Lee and Alexander Stepanov. The Standard Template Library. Technical report, Hewlet Packard Laboratories, 1995.

[Mac87] Mary E. Mace. *Memory Storage Patterns in Parallel Processing*. Kluwer Academic Publishers, 1987.

[Mar97] Bret Andrew Marsolf. *Techniques for the Interactive Development of Numerical Linear Algebra Libraries for Scientific Computation*. PhD thesis, University of Illinois At Urbana-Champaign, 1997.

[Mat] The MathWorks. *PRO-MATLAB User's Guide*.

[McD89] John Alan McDonald. Object-oriented programming for linear algebra. In *OOPSLA'89 Conference Proceedings on Object-oriented Programming Systems, Languages and Applications*, pages 175–184, October 1989. Published in ACM SIGPLAN Notices, Vol. 24, No. 10.

[MEO98] Sven Erik Mattsson, Hilding Elmqvist, and Martin Otter. Physical system modeling with Modelica. *Control Engineering Practice*, 6(4):501–510, 1998.

[Mey97] Bertrand Meyer. *Object Oriented Software Construction*. Prentice Hall, 2^{th} edition, 1997.

[MGG97] Bret Andrew Marsolf, K. A. Gallivan, and E. Gallopoulos. On the use of algebraic and structural information in a library prototyping and development environment. In *Proceedings* 15^{th} *IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, pages 565–570, 1997.

[MMG98] José E. Moreira, Sam P. Midkiff, and Manish Gupta. From flop to megaflops: Java for technical computing. In *11th International Workshop on Languages and Compiler for Technical Computing*, 1998.

[MMG99] José E. Moreira, Samuel P. Midkiff, and Manish Gupta. A standard java array package for technical computing. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, March 1999.

[Mod] Modelica Design Group. *Modelica modeling language web page.* http://www.modelica.org/.

[MOT97] Eva Mossberg, Kurt Otto, and Michael Thuné. Object-oriented software tools for the construction of preconditioners. *Scientific Programming*, 6:285–295, 1997.

[MS95] David R. Musser and Atul Saini. *STL Tutorial and Reference Guide: C++ Programming with Standard Template Library.* Addison Wesley, 1995.

[MTL] Laboratory for Scientific Computing, University of Notre Dame. *Matrix Template Library (MTL) web page.* http://www.lsc.nd.edu/research/mtl.

[Mul97] Pierre-Alain Muller. *Instan UML.* Wrox, 1997.

[NE99] Eric Noulard and Nahid Emad. Object oriented design for reusable parallel linear algebra software. In Patrick Amestoy, Philippe Berger, Michael Daydé, Iain Duff, Valerie Fraysse Luc Giraud, and Daniel Ruiz, editors, *Proceedings Euro-Par'99 Parallel Processing – 5^{th} International Euro-Par Conference*, number 1685 in Lecture Notes in Computer Science. Springer-Verlag, 1999.

[Nor96] Charles D. Norton. *Object-Oriented Programming Paradigms in Scientific Computing*. PhD thesis, Department of Computer Science, Rensselaer Polytechnic Institute, New York, 1996.

[Obj] Programming Environments Laboratory, Department of Computer and Information Science, Linköping University. *ObjectMath programming environment web page*. http://www.ida.liu.se/labs/pelab/omath/.

[OVE] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory. *Overture framework web page*. http://www.llnl.gov/casc/Overture/.

[OW97] Martin Odersky and Philip Wadler. Pizza into java: Translating theory into practice. In *Proceeding of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 146–159, 1997.

[Owl] Department of Computer Science, University of Rice. *Objects within the Linear Algebra Package (OwlPack) web page*. http://www.cs.rice.edu/~budimlic/OwlPack.

[PB85] Paul W. Purdom and Cynthia A. Brown. *The Analysis of Algorithms*. Holt, Rinehart and Winston, 1985.

[PET] Mathematics and Computer Science Division at Argone National Laboratory. *Portable Extensible Toolkit for Scientific Computation (PETSc) web page*. http://www.mcs.anl.gov/petsc.

[PML] High Performance Computing Laboratory, Mississippi State University. *Parallel Mathematical library Project (PMLP) web page*. http://www.erc.msstate.edu/research/labs/hpcl/pmlp/.

[POO] Advanced Computing Laboratory, Los Alamos National Laboratory. *Parallel Object-Oriented Methods and Applications (POOMA) framework web page*. http://www.acl.lanl.gov/Pooma.

[Poz97] Roldan Pozo. Template numerical toolkit for linear algebra: High performance programming with C++ and the Standard Template

Library. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(3):251–263, 1997.

[Pre97] Roger S. Pressman. *Software Engineering: a Practioner's Approach.* McGraw Hill, 4^{th} edition, 1997.

[PRL96] Roldan Pozo, Karin A. Remington, and Andrew Lumsdaine. *SparseLib++ v. 1.5: Sparse Matrix Class Library Reference Guide*, April 1996.

[Ran95] Jarmo Rantakokko. Object-oriented software tools for composite-grid methods on parallel computers. Technical Report 165, Department of Scientific Computing, Uppsala University, 1995.

[Rat97a] Rational Software Corporation. *Unified Modeling Language: Notation Guide*, version 1.1 edition, 1997. Available at http://www.rational.com/uml/1.1/.

[Rat97b] Rational Software Corporation. *Unified Modeling Language: Semantics*, version 1.1 edition, 1997. Available at http://www.rational.com/uml/1.1/.

[RB96] John R. Rice and Ronald F. Boisvert. From scientific software libraries to problem solving environments. *IEEE Computational Science and Engineering Magazine*, pages 44–53, 1996.

[Ric96] John R. Rice. Scalable scientific software libraries and problem solving environments. Technical report, Computer Science, Purdue University, 1996. TR-96-001.

[Rog93] Rogue Wave Software Inc. *First Annual Object Oriented Numerics Conference*, 1993.

[Saa96] Youcef Saad. *Iterative Methods for Sparse Linear Systems.* PWS, 1996.

[SBD+76] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler. *Matrix eigensystem routines: EISPACK guide.*, volume 5 of *Lecture Notes in Computer Science.* Springer-Verlag, 2^{nd} edition, 1976.

[SL98a] Jeremy G. Siek and Andrew Lumsdaine. The matrix template library: A generic programming approach to high performance numerical linear algebra. In Denis Caromel, Rodney R. Oldehoeft, and Marydell Tholburn, editors, *Computing in Object-Oriented Parallel Environments, Second International Symposium ISCOPE 98*, number 1505 in Lecture Notes in Computer Science, pages 59–70. Springer-Verlag, 1998.

[SL98b] Jeremy G. Siek and Andrew Lumsdaine. The matrix template library: A unifying framework for numerical linear algebra. In S. Demeyer and J. Bosch, editors, *Object-Oriented Technology – ECOOP'98 Workshop Reader*, volume 1543 of *Lecture Notes in Computer Science*, pages 466–467. Springer-Verlag, 1998. Workshop on Parallel Object-Oriented Scientific Computing.

[SL98c] Jeremy G. Siek and Andrew Lumsdaine. A rational approach to portable high performance: The basic linear algebra instruction set (BLAIS) and the fixed algorithm size template (FAST) library. In S. Demeyer and J. Bosch, editors, *Object-Oriented Technology – ECOOP'98 Workshop Reader*, volume 1543 of *Lecture Notes in Computer Science*, pages 468–489. Springer-Verlag, 1998. Workshop on Parallel Object-Oriented Scientific Computing.

[SL99] Jeremy G. Siek and Andrew Lumsdaine. The matrix template library: Generic components for high-performance scientific computing. *IEEE Computing in Science and Engineering*, 1(6):70–78, 1999.

[SLL99] Jeremy G. Siek, Andrew Lumsdaine, and Lie-Quann Lee. Generic programming for high performance numerical linear algebra. In Michael E. Henderson, Christopher R. Anderson, and Stephen L. Lyons, editors, *Object Oriented Methods for Interoperable Scientific and Engineering Computing*, SIAM Proceedings in Applied Mathematics, 1999. Proceedings of SIAM Workshop on Object Oriented Methods for Interoperable Scientific and Engineering Computing, October 1998.

[Slo73] Neil J. A. Sloane. *A Handbook of Integer Sequences.* Academic Press, 1973.

[SM88] S. Shlaer and S. Mellor. *Object-Oriented Systems Analysis: Modeling the World in Data*. Yourdon Press, 1988.

[SMO] *Sparse Matrix Object-oriented Ordering methods (SMOOTH) web page*. http://www.cs.yorku.ca/ joseph/Smooth/SMOOTH.html.

[Spa] *SparseLib++ library web page*. http://math.nist.gov/sparselib++/.

[SPO] *Sparse Object Oriented Linear Equations Solver (SPOOLES) web page*. http://www.netlib.org/linalg/spooles/spooles.2.2.html.

[Ste73] George W. Stewart. *Introduction to Matrix Computations*. Academic Press, 1973.

[Ste99] George W. Stewart. *The Jampack Owner's Manual*, 1999. ftp://thales.cs.umd/pub/Jampack/AboutJampack.html.

[TI97] Lloyd N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM Press, 1997.

[TMO+97] Michael Thuné, Eva Mossberg, Peter Olsson, Jarmo Rantakokko, Krister Åhlander, and Kurt Otto. Object-oriented construction of parallel PDE solvers. In Erlend Arge, Are Magnus Bruaset, and Hans Petter Langtangen, editors, *Modern Software Tools for Scientific Computing*, pages 203–226. Birkhäuser, 1997.

[TNT] Mathematical and Computational Sciences Division, NIST. *Template Numerical Toolkit (TNT) web page*. http://math.nist.gov/tnt/.

[WD98] R. Clint Whaley and Jack J. Dongarra. Automatically tuned linear algebra software. In *Proceedings of Supercomputing '98*. IEEE Press, 1998.

[Wie98] Roel Wieringa. A survey of structured and object-oriented software specification methods and techniques. *ACM Computing Surveys*, 30(4):459–527, December 1998.

A Tight Lower Bound for Top-Down Skew Heaps*

Berry Schoenmakers†

January, 1997

Abstract

Previously, it was shown in a paper by Kaldewaij and Schoenmakers that for top-down skew heaps the amortized number of comparisons required for meld and delmin is upper bounded by $\log_\phi n$, where n is the total size of the inputs to these operations and $\phi = (\sqrt{5}+1)/2$ denotes the golden ratio. In this paper we present worst-case sequences of operations on top-down skew heaps in which each application of meld and delmin requires approximately $\log_\phi n$ comparisons. As the remaining heap operations require no comparisons, it then follows that the set of bounds is tight. The result relies on a particular class of self-recreating binary trees, which is related to a sequence known as Hofstadter's G-sequence.

1 Introduction

Top-down skew heaps are probably the simplest implementation of *mergeable* priority queues to date while still achieving good performance. As with other so-called self-adjusting data structures the catch is that the performance is merely good in the amortized sense, but in many applications this is perfectly acceptable. Figure 1 displays a purely functional version of top-down skew heaps, which is based on the original version of Sleator and Tarjan, the inventors of skew heaps [9]. Compared to the set of programs described in [9], however, we use operation single instead of an insert operation (note that insertion of a into heap x can be achieved as meld.(single.a).x).

The efficiency of skew heaps is entirely due to the particular way operation meld is defined. Informally the effect of meld.x.y can be described by two steps. First, the rightmost paths of trees x and y are merged, where the left subtrees of the nodes on the merge path stick to their nodes. Second, the left and right subtrees are swapped for every node on the merge path. Intuitively, the second step turns the potentially long merge path, which is a rightmost path, into a leftmost path of the resulting heap. In actual implementations it is worthwhile to program meld performing a single pass over the rightmost paths, while at the same time building up the leftmost path (see [9]). As shown in [3], it is then possible to get a simple implementation of mergeable priority queues that permits an interesting degree of concurrency.

In [4, 8] the following upper bounds have been proven for the amortized costs of the operations in terms of comparisons (each unfolding of meld.x.y requires one comparison if

*Appears in *Information Processing Letters* 61(5) 279–284, March 14, 1997

†DigiCash, Kruislaan 419, 1098 VA Amsterdam, Netherlands. berry@digicash.com

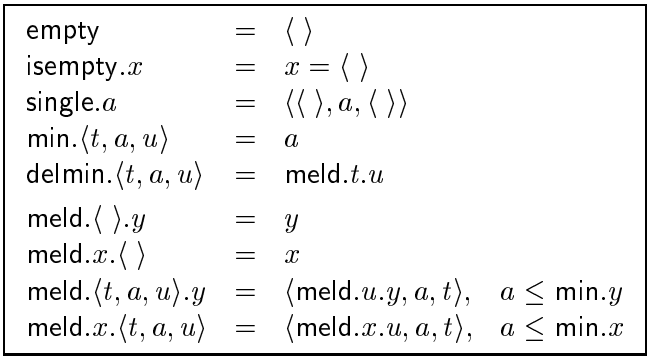


Figure 1: Purely functional top-down skew heaps, where $\langle\,\rangle$ denotes the empty tree and $\langle t, a, u\rangle$ denotes a tree with left subtree t, root a, and right subtree u.

x and y are both nonempty). These bounds improve upon the original bounds of Sleator and Tarjan [9] by more than a factor of two. As explained in [8, Chapter 5], these bounds do only hold for functional programs that restrict the use of skew heaps to "linear usage" as if operations delmin and meld were destructive (e.g., using both delmin.x and meld.$x.y$ in the same expression is not allowed). It is interesting to note that Okasaki has been able to remove this restriction of linear usage for many purely-functional data structures by making judicious use of lazy evaluation (see [6, 7]).

Theorem 1 *(cf. [8, Lemma 9.2]) There exists a potential function such that the amortized costs for top-down skew heaps satisfy (in terms of comparisons):* empty, isempty.x, *and* min.x *cost 0,* single.a *costs at most 1,* delmin.x *costs at most* $\log_\phi |x|$, *and* meld.$x.y$ *costs at most* $\log_\phi(|x| + |y|)$, *where* $\phi = (\sqrt{5}+1)/2$ *denotes the golden ratio.*

Here, we have used $|x|$ to denote the size of tree x, which is defined equal to one plus the number of nodes of x. A recursive definition is given by: $|\langle\,\rangle| = 1$ and $|\langle t, a, u\rangle| = |t| + |u|$.

In this paper we will show that the above set of bounds is in fact tight. We do so by presenting worst-case sequences of operations on top-down skew heaps in which the actual cost of each operation matches the alloted amortized cost of Theorem 1. Clearly, meld forms the central operation on skew heaps. In the next section we first consider a special version of meld that operates on *unlabelled* binary trees. This special version takes maximal time for a particular class of trees. In the subsequent section we then show that these cases actually arise in applications of skew heaps, which implies that the bounds of Theorem 1 are tight. Our methods resemble the methods used to obtain lower bounds on the amortized complexity of union-find data structures (see, e.g., [1, 11, 12]), in which finding a suitable class of self-recreating (or, self-reproducing) trees also constitutes an important part of the solution. Throughout the analysis we find it instrumental to use a functional notation.

2 Unlabelled case

We consider the following operation $\bowtie$ on unlabelled binary trees, which is strongly related to operation meld:

$$\begin{aligned} \langle\,\rangle \bowtie \langle\,\rangle &= \langle\,\rangle \\ \langle t, u\rangle \bowtie y &= \langle y \bowtie u, t\rangle. \end{aligned}$$

As part of operation $x \bowtie y$ the rightmost paths of x and y are traversed in alternating order starting with x. Note that for each application of $x \bowtie y$ we need that $x \neq \langle\,\rangle$ if $y \neq \langle\,\rangle$, which is ensured if $|x| \geq |y|$. This will be the case.

The goal of the analysis is to define a sequence of trees for which $\bowtie$ is expensive, while the resulting tree is again an element of the sequence. In light of Theorem 1 the cost of $x \bowtie y$ (which is defined as the sum of the lengths of the rightmost paths of x and y) should be close to $\log_\phi(|x| + |y|)$. It will be no surprise that the Fibonacci sequence plays an important role in our construction.

Define the Fibonacci sequence F_k, $k \geq 0$, as usual by $F_0 = 0$, $F_1 = 1$, and $F_k = F_{k-1} + F_{k-2}$, $k \geq 2$. Next define two related functions $L(n)$ and $R(n)$, $n \geq 0$, as follows:[1]

$$\begin{aligned} &L(0) = R(0) = 0 \\ &L(n) = F_{k-1} + L(n - F_k) \\ &R(n) = F_{k-2} + R(n - F_k), \quad n \geq 1, \end{aligned}$$

where k is uniquely determined by $F_k \leq n < F_{k+1}$. Hence $k \geq 2$. As an alternative characterisation of L and R we will often use the next two lemmas.

Lemma 1 *$L(F_k + a) = F_{k-1} + L(a)$, for $0 \leq a \leq F_{k-1}$ and $k \geq 1$.*

Lemma 2 *$R(F_k + a) = F_{k-2} + R(a)$, for $0 \leq a \leq F_{k-1}$ and $k \geq 2$.*

A few simple properties of L and R are given by the next lemma.

Lemma 3 *$L(n) + R(n) = n$, $L(n) \geq R(n)$, $L(n+1) \geq L(n)$, and $R(n+1) \geq R(n)$, for $n \geq 0$.*

Next we prove the three main lemmas we need.

Lemma 4 *$L(L(n-1)) = R(n)$, for $n \geq 1$.*

Proof By induction on n. If $n = 1$ both sides are zero. For $n \geq 2$, let k denote the unique integer satisfying $F_k < n \leq F_{k+1}$ (hence $k \geq 2$). Then we have:

$$\begin{aligned} & L(L(n-1)) \\ = & \quad \{ \text{ definition } L \,\} \\ & L(F_{k-1} + L(n - 1 - F_k)) \\ = & \quad \{ \text{ Lemma 1, using } 0 \leq L(n-1-F_k) \leq F_{k-2} \,\} \\ & F_{k-2} + L(L(n - 1 - F_k)) \\ = & \quad \{ \text{ induction hypothesis } \} \\ & F_{k-2} + R(n - F_k) \\ = & \quad \{ \text{ Lemma 2, using } 0 \leq n - F_k \leq F_{k-1} \,\} \\ & R(n). \end{aligned}$$

[1] Through the on-line version of Sloane's "Encyclopedia of Integer Sequences" [10], we found out that we had rediscovered Hofstadter's G-sequence [2, p.137], since function $L(n)$ satisfies $L(0) = 0$ and $L(n) = n - L(L(n-1))$, $n \geq 1$ (use Lemmas 3 and 4).

□

Lemma 5 *$L(L(n)-1) = R(n-1)$, for $n \geq 1$.*
Proof By induction on n. If $n = 1$ both sides are zero. For $n \geq 2$, let k denote the unique integer satisfying $F_k < n \leq F_{k+1}$ (hence $k \geq 2$). Then we have:

$$
\begin{aligned}
& L(L(n)-1) \\
= & \quad \{ \text{ Lemma 1, using } 1 \leq n - F_k \leq F_{k-1} \ \} \\
& L(F_{k-1} + L(n-F_k) - 1) \\
= & \quad \{ \text{ Lemma 1, using } 0 \leq L(n-F_k) - 1 \leq F_{k-2} \ \} \\
& F_{k-2} + L(L(n-F_k) - 1) \\
= & \quad \{ \text{ induction hypothesis } \} \\
& F_{k-2} + R(n - F_k - 1) \\
= & \quad \{ \text{ definition } R \ \} \\
& R(n-1).
\end{aligned}
$$

□

Lemma 6 *$R(L(n)-1) = R(L(n-1))$, for $n \geq 1$.*
Proof For any $n \geq 1$, we have:

$$
\begin{aligned}
& R(L(n)-1) - R(L(n-1)) \\
= & \quad \{ \text{ Lemma 3 (twice) } \} \\
& L(n) - 1 - L(L(n)-1) - L(n-1) + L(L(n-1)) \\
= & \quad \{ \text{ Lemmas 5, 4, resp. } \} \\
& L(n) - 1 - R(n-1) - L(n-1) + R(n) \\
= & \quad \{ \text{ Lemma 3 (twice) } \} \\
& 0 \, .
\end{aligned}
$$

□

Given functions L and R we now define a sequence of unlabelled binary trees G_n, $n \geq 0$, as follows:

$$
\begin{aligned}
& G_0 = \langle \ \rangle \\
& G_n = \langle G_{L(n-1)}, G_{R(n-1)} \rangle, \quad n \geq 1.
\end{aligned}
$$

We dub these trees "golden trees" because the ratio of the sizes of the left subtrees to the right subtrees approaches the golden ratio ϕ (since $L(n)/R(n)$ approaches ϕ). The golden trees can be seen as a supersequence of the Fibonacci trees [5, p.414] in the sense that a golden tree G_n corresponds to the Fibonacci tree of order k whenever $n = F_{k+1} - 1$, $k \geq 0$. See Figure 2, trees G_0, G_1, G_2, G_4, G_7, and G_{12} correspond to Fibonacci trees.

The main lemma for golden trees is stated below. It says that if two golden trees of appropriate sizes are melded, then the result is a golden tree as well. In particular, if the left and right subtrees of the root of a golden tree G_n are melded, then the result is a G_{n-1} tree.

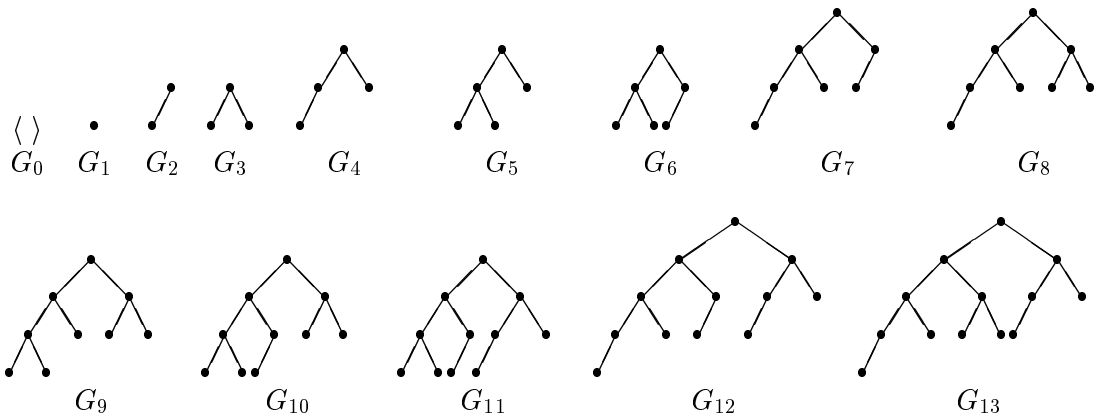


Figure 2: G_n trees for $n = 0, \ldots, 13$.

Lemma 7 $G_{L(n)} \bowtie G_{R(n)} = G_n$, $n \geq 0$.
Proof By induction on n. Clearly true for $n = 0$. For $n \geq 1$, we note:

$$
\begin{aligned}
& G_{L(n)} \bowtie G_{R(n)} \\
= & \quad \{ \text{ definition } G \ \} \\
& \langle G_{L(L(n)-1)}, G_{R(L(n)-1)} \rangle \bowtie G_{R(n)} \\
= & \quad \{ \text{ definition } \bowtie \ \} \\
& \langle G_{R(n)} \bowtie G_{R(L(n)-1)}, G_{L(L(n)-1)} \rangle \\
= & \quad \{ \text{ Lemmas 4, 6, and 5, resp. } \} \\
& \langle G_{L(L(n-1))} \bowtie G_{R(L(n-1))}, G_{R(n-1)} \rangle \\
= & \quad \{ \text{ induction hypothesis, using } L(n-1) < n \ \} \\
& \langle G_{L(n-1)}, G_{R(n-1)} \rangle \\
= & \quad \{ \text{ definition } G \ \} \\
& G_n .
\end{aligned}
$$

□

The cost of computing G_n from $G_{L(n)}$ and $G_{R(n)}$ can now be related to the size of G_n. Clearly, we have $|G_n| = n + 1$. As defined before, the cost of $x \bowtie y$ is equal to the sum of the lengths of the rightmost paths of x and y. Alternatively, the cost of $x \bowtie y$ is equal to the length of the leftmost path of $x \bowtie y$ (see, for example, Figure 3). Using $||x||$ to denote the length of the leftmost path of x, formally defined by $||\langle\ \rangle|| = 0$ and $||\langle t, u \rangle|| = ||t|| + 1$, we then have the following lemmas.

Lemma 8 $L(F_k - 2) = F_{k-1} - 1$ *and* $L(F_{k+1} - 3) = F_k - 2$ *, for* $k \geq 3$.
Proof (Only first part, second part is similar.) By induction on k. If $k = 3$ both sides are zero, and if $k = 4$ both sides are one. For $k \geq 5$, we have:

$$
\begin{aligned}
& L(F_k - 2) \\
= & \quad \{ \text{ definition } F \ \} \\
& L(F_{k-1} + F_{k-2} - 2)
\end{aligned}
$$

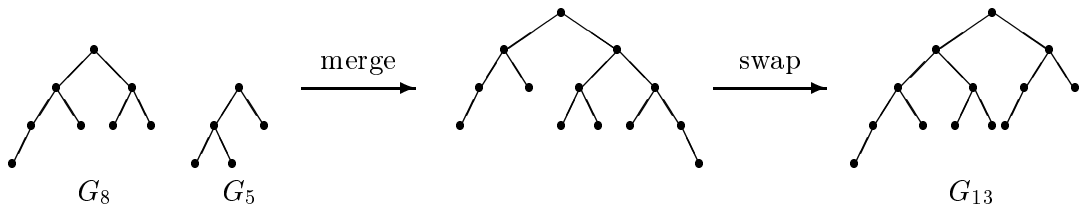


Figure 3: Operation $G_8 \bowtie G_5$, viewed as first merging the rightmost paths and then swapping the subtrees of all nodes on the rightmost path, resulting in G_{13}.

$$
\begin{array}{ll}
= & \{ \text{ Lemma 1, using } F_{k-2} \geq 2 \ \} \\
& F_{k-2} + L(F_{k-2} - 2) \\
= & \{ \text{ induction hypothesis } \} \\
& F_{k-2} + F_{k-3} - 1 \\
= & \{ \text{ definition } F \ \} \\
& F_{k-1} - 1.
\end{array}
$$

□

Lemma 9 *$\|G_n\| = k-2$, for $n \geq 0$, where k is the unique integer satisfying $F_k \leq |G_n| < F_{k+1}$.*
Proof By induction on n. Clearly true for $n = 0$ (and $k = 2$). For $n \geq 1$ (hence $k \geq 3$), we note that $\|G_n\| = \|G_{L(n-1)}\|+1$. From the induction hypothesis we get that $\|G_{L(n-1)}\| = k-3$, so the result follows provided $F_{k-1} \leq L(n-1) + 1 < F_k$ is implied by $F_k \leq n + 1 < F_{k+1}$. For the lower bound we note that $F_k - 2 \leq n - 1$ implies that $F_{k-1} - 1 \leq L(n-1)$, using Lemma 8. For the upper bound we note that $n - 1 \leq F_{k+1} - 3$ implies that $L(n-1) \leq F_k - 2$, again using Lemma 8. □

This leads to the following conclusion. Let $F_k \leq n + 1 < F_{k+1}$ and consider the computation of the meld of $G_{L(n)}$ and $G_{R(n)}$. On account of Lemma 9 the actual cost of this operation is $\|G_n\| = k - 2$. The upper bound (Theorem 1) for the amortized cost of this operation is $\log_\phi |G_n| = \log_\phi(n + 1)$, which is bounded by approximately $k - 0.67$, using that $F_{k+1} \approx \phi^{k+1}/\sqrt{5}$. So, the actual cost differs at most a small constant from the alloted amortized cost for such a meld operation.

3 Labelled case

The central properties achieved in the previous section are the facts that tree G_n can be written both as $G_n = G_{L(n)} \bowtie G_{R(n)}$ and as $G_n = \langle G_{L(n-1)}, G_{R(n-1)} \rangle$, $n \geq 1$, and that $\|G_n\|$ is approximately equal to $\log_\phi n$. In this section we use these results to construct a worst-case sequence of skew heap operations. The plan is to consider a particular sorting program and to see to it that each meld and delmin takes maximal time.

$$
\begin{array}{rcll}
sort.N & = & h.(g.N) & \\
g.0 & = & \mathsf{empty} & \\
g.1 & = & \mathsf{single}.some & \\
g.n & = & \mathsf{meld}.(g.L(n)).(g.R(n)), & n \geq 2 \\
h.x & = & [\,], & \mathsf{isempty}.x \\
h.x & = & \mathsf{min}.x \vdash h.(\mathsf{delmin}.x), & \neg\mathsf{isempty}.x
\end{array}
$$

Function g first builds a skew heap, where the elements for the singleton heaps are chosen appropriately. We will show that it is possible to choose each singleton element such that each $g.n$ heap is a G_n-tree, and moreover such that each tree to which h is applied, is a G_n-tree as well. It then follows from the results for the unlabelled case that the applications of meld and delmin all take maximal time.

For this to hold it suffices to show the existence of a labelling for which each application of meld and delmin simulates the behaviour of $\bowtie$. The next lemma captures the essence, where we limit ourselves to labellings without duplicates.

Lemma 10 *Consider any labelled G_n-heap x, $n \geq 0$. Then there exist a labelled $G_{L(n)}$-heap t and a labelled $G_{R(n)}$-heap u such that* meld.$t.u = x$.

Proof On account of Lemma 7, we know that $G_n = G_{L(n)} \bowtie G_{R(n)}$. This defines a one-to-one mapping between the nodes of G_n on the one hand and the nodes of $G_{L(n)}$ and $G_{R(n)}$ on the other hand. If we now copy the labels of x to trees t and u according to this mapping, we know that both t and u are heaps, and that indeed meld.$t.u = x$ provided that the labels of the rightmost paths of t and u alternate, starting with t. This is indeed true because the leftmost path of x forms an increasing list. □

We work backwards to show that the labels in the program *sort* (in single.*some*) can be picked such that each application of delmin and meld simulates $\bowtie$. First consider a labelled version of the G_n trees, such that delmin.$G_n = G_{n-1}$ and min.$G_n = -n$. The sequence is defined inductively by $G_0 = \langle\,\rangle$ and $G_n = \langle t, -n, u\rangle$, $n \geq 1$, where t is a labelled $G_{L(n-1)}$ tree and u is a labelled $G_{R(n-1)}$ tree such that meld.$t.u = G_{n-1}$. The existence of these labelled trees is guaranteed by Lemma 10. This fixes all the trees to which h is applied, hence tree $g.N$ as well. On account of Lemma 10 it follows that there exist labelled versions of $g.L(N)$ and $g.R(N)$ for which meld yields $g.N$. Repeating this argument it then follows that a (unique) assigment of the inputs to single exists that satisfies our requirements. As the argument holds for any N, $N \geq 0$, we have proved that:

Theorem 2 *There exist sequences of operations on top-down skew heaps such that the heaps may get arbitrarily large and for which the actual costs satisfy (in terms of comparisons):* empty, isempty.x, min.x, *and* single.a *cost 0,* delmin.x *costs at least* $\log_\phi |x| - c$, *and* meld.$x.y$ *costs at least* $\log_\phi(|x| + |y|) - c$, *where* $\phi = (\sqrt{5}+1)/2$ *denotes the golden ratio and c is a small constant, $c < 2$.*

4 Concluding remarks

As defined in Figure 1 operation meld.$x.y$ terminates as soon as the end of the rightmost path of either x or y is reached. It is also possible to define meld such that melding continues until both rightmost paths are completely traversed. The same bounds apply to this version of meld. (Actually, this version was analyzed in [4], and in [8] it was shown that the same upper bounds hold for both versions.)

It would be interesting to extend our results to bottom-up skew heaps as well. In [8] several sets of amortized bounds have been derived. Just as for top-down skew heaps, operations empty, isempty, single, and min all take $O(1)$ time. One set of bounds [8, Lemma 9.10] says that it is possible to amortize the costs (counting comparisons) such that the amortized costs are at most 3 for meld and at most $1 + 2\log_2 |x|$ for delmin.x. This improves upon the original

bounds by Sleator and Tarjan of [9] by a factor of two. A new parameterized set of bounds that is incomparable with previous bounds [8, Lemma 9.12] says that the amortized costs can be chosen at most $1+\varepsilon\log_\beta(|x|+|y|)$ for $\mathsf{meld}.x.y$ and at most $1+(\varepsilon+2)\log_\beta|x|$ for $\mathsf{delmin}.x$, where $\beta=\frac{(\varepsilon+1)^{\varepsilon+1}}{\varepsilon^\varepsilon}$, for any $\varepsilon>0$.

Picking $\varepsilon=\phi$ in the latter case yields as upper bounds $\frac{\phi}{\phi+2}\log_\phi(|x|+|y|)$ for $\mathsf{meld}.x.y$ and $\log_\phi|x|$ for $\mathsf{delmin}.x$. Since we now know that the bounds of Theorem 1 are tight, it follows that bottom-up skew heaps outperform top-down skew heaps, as the bound for meld is better. It is an interesting open problem whether the bounds for bottom-up skew heaps are tight as well.

References

[1] M.J. Fischer, Efficiency of equivalence algorithms, in: R.E. Miller and J.W. Thatcher, eds., *Complexity of Computer Computations* (Plenum Press, New York, 1972) 153–168.

[2] D.R. Hofstadter, *Gödel, Escher, Bach: an Eternal Golden Braid*, Basic Books (1979).

[3] D.W. Jones, Concurrent operations on priority queues, *Communications of the ACM* **32** (1989) 132–137.

[4] A. Kaldewaij and B. Schoenmakers, The derivation of a tighter bound for top-down skew heaps, *Information Processing Letters* **37** (1991) 265–271.

[5] D. Knuth, *The Art of Computing Programming*, Volume 3, Sorting and Searching, Addison Wesley (1975).

[6] C. Okasaki, Amortization, lazy evaluation, and persistence: Lists with catenation via lazy linking, in: *IEEE Symposium on Foundations of Computer Science* (October 1995) 646–654.

[7] C. Okasaki, The role of lazy evaluation in amortized data structures, in: *ACM SIGPLAN International Conference on Functional Porgramming* (May 1996) 62–72.

[8] B. Schoenmakers, *Data Structures and Amortized Complexity in a Functional Setting*, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (1992).

[9] D.D. Sleator and R.E. Tarjan, Self-adjusting heaps, *SIAM Journal on Computing* **15** (1986) 52–69.

[10] N.J.A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press (1995).

[11] R.E. Tarjan, Efficiency of a good but not linear set union algorithm, *Journal of the ACM* **22** (1975) 215–225.

[12] R.E. Tarjan and J. van Leeuwen, Worst-case analysis of set union algorithms, *Journal of the ACM* **31** (1984) 245–281.

,

Conjectures on the Size of Constellations Constructed from Direct Sums of PSK Kernels

Matthew G. Parker**

Department of Informatics, University of Bergen, N-5020 Bergen, Norway,
matthew@ii.uib.no

Abstract. A general equation is given for the size of complex constellations constructed from the direct sum of PSK-like constellation primitives. The equation uses a generating function whose numerator is a power of a 'coordination polynomial'. Conjectures are also given as to the form and value of these coordination polynomials for various PSK. The study has relevance to error-coding, polynomial residue number theory, and the analysis of random walks.

1 Introduction

Communications systems often transmit data by modulating using Binary or Quaternary Phase Shift Keyed (BPSK or QPSK) or Quadrature Amplitude Modulated (QAM) constellations in the complex plane. But larger constellations can be more bandwidth-efficient and lead to efficient hardware implementation of complex arithmetic and algorithms [1, 2]. This paper considers the problem of finding the size of constellations constructed from direct sums of {PSK plus the origin}, referred to here as 'PSK⊕' constellations. These constellations form lattices for 1,2,3, or 6 PSK primitives, but for any other PSK⊕ there will be residue 'folding' making the determination of constellation size more complicated. This problem can be recast, for mPSK⊕, as finding an expression for the number of non-identical polynomial residues resulting from the reduction, mod $\Phi_m(x)$, of polynomials in x of Coefficient Weight $\leq n$, (for some positive integer, n), and degree $< m$, where $\Phi_m(x)$ is the m^{th} cyclotomic polynomial in x. Although residue folding is, for many applications, undesirable, it is hoped that an algebraic understanding of PSK⊕ will help in the construction of constellations more suited to communications systems which use PSK⊕ as building blocks. Also, from an algebraic point of view, it is useful to be able to enumerate the residues of polynomials, mod $\Phi_m(x)$. The theorem and conjectures to be presented here are based on computational results. During the course of the work integer sequences, relating to the 8PSK⊕ and 16PSK⊕ constellations, were entered into Sloane's On-Line Encyclopedia of Integer Sequences [3] and were found to refer, in particular, to the paper by Conway and Sloane on Low Dimensional Lattices [4] which, in turn, references work by O'Keefe [5] and others [6].

** This work was funded by NFR Project Number 119390/431

Their results have applications to crystallography, and use generating functions which require the specification of a 'Coordination Sequence'. This paper conjectures a general solution to a related problem, although a general form for the Coordination Sequence (Polynomial) has yet to be found. The results could be used to help extend the scope of error coding strategies such as [7, 8], and may also be useful for the development of 'Random Walk' statistics.

2 Statement of the Problem

Define mPSK+ as the set of $m+1$ points in the complex plane given by,

$$m\text{PSK+} = \{0, 1, w, w^2, \ldots, w^{m-1}\}$$

where $w = e^{\frac{2\pi i}{m}}$, and $i^2 = -1$. Define mPSK$\oplus n$ as the direct sum of n copies of mPSK+, given by,

$$m\text{PSK} \oplus n = \sum_{k=0}^{n-1} \{0, 1, w, w^2, \ldots, w^{m-1}\}$$

We wish to find a formula for d_n as n varies over the positive integers, where d_n is the number of non-identical points in mPSK$\oplus n$, given by,

$$d_n = |\sum_{k=0}^{n-1} \{0, 1, w, w^2, \ldots, w^{m-1}\}|$$

For instance, let $m = 4$. The kernel constellation is $\{0, 1, w, w^2, w^3\}$, where $w = e^{\frac{2\pi i}{4}}$, and,

$$d_2 = |\sum_{k=0}^{1} \{0, 1, w, w^2, w^3\}| = |\{0, \pm 1, \pm w, \pm 1 \pm w, \pm 1 \mp w, \pm 2, \pm 2w\}| = 13$$

As another example, for $m = 6$ and $n = 2$,

$$d_2 = |\{0, \pm 1, \pm w, \pm w^2, \pm 2, \pm 2w, \pm 2w^2, \pm 1 \pm w, \pm 1 \mp w^2, \pm w \mp w^2\}| = 19$$

An algebraic description of the same problem is as follows.

Definition 1 *The 'Coefficient Weight', (cw), of a polynomial, $f(x)$, is the sum of it's coefficient values. In other words $cw(f(x)) = f(1)$.*

Let $g(x) = \sum_i g_i x^i$. Let,

$$\mathbf{G_{m,n}} = \{g(x) \mid 0 \leq \deg(g(x)) < m, g_i \geq 0 \ \ \forall i, 0 \leq \mathrm{cw}(g(x)) \leq n\}$$

where $\deg(a(x))$ is the degree of $a(x)$. Let $x = e^{\frac{2\pi i}{m}}$, where $i^2 = -1$. Then,

$$m\text{PSK} \oplus n = \{h(x) \mid h(x) = \langle g(x) \rangle_{\Phi_m(x)}, \forall g(x) \in \mathbf{G_{m,n}}\}$$

where $\langle a \rangle_b$ is the residue of a mod b, and $\Phi_m(x)$ is the m^{th} cyclotomic polynomial. Therefore,

$$d_n = |m\text{PSK} \oplus n|$$

as before.

3 Computational Results

Tables 1 and 2 show some computed values of d_n for various n and m. The number of Euclidean distances, D, refers to the size of the set of values for the absolute (straight-line) distance from each point in mPSK$\oplus n$ to the origin. The figures for D are not discussed further in this paper, but are included here for the reader's interest.

Table 1. Constellation and Euclidean Distance Enumerations for Various mPSK$\oplus n$

d_n-No of points in constellation. D-No of Euclidean distances.

n	1		2		3		4		5		6		7		8		9		10	
m	d_n	D	d_n	D	d_n	D	d_n	D	d_n	D	d_n	D	d_n	D	d_n	D	d_n	D	d_n	D
3	4	2	10	3	19	5	31	7	46	9	64	12	85	15	109	18	136	22	166	26
4	5	2	13	4	25	6	41	9	61	12	85	16	113	19	145	24				
5	6	2	21	5	56	8	126	17												
6	7	2	19	4	37	6	61	9	91	12	127	16	169	20						
7	8	2	36	6	120	14	330	30												
8	9	2	41	6	129	13	321	29	681	53	1289	96	2241		3649		5641		8361	
9	10	2	55	6	217	17	685	46	1837	99										
10	11	2	61	7	211	77	551	38	1201	72										
12	13	2	73	7	253	16	661	38	1441	72										
14	15	2	113	9	575	29	2171	96												
15	16	2	136	9	811	33	3751	132	14176	440										
16	17	2	145	10	833	35														
18	19	2	163	10	865	33	3313	114												
20	21	2	221	12	1521	46														
21	22	2	253	12	2017	59	12496	322	63946	1396										
22	23	2	265	13	2047	59	11969	310												
24	25	2	289	13	2089	54	10825	258												
25	26	2	351	15	3276	78														
27	28	2	406	15	4051	89	31213	4296												
30	31	2	451	16	3901	81	22831	425												
33	34	2	595	18	7129	125	65671	1072												
35	36	2	666	20	8436	138														
36	37	2	649	19	7237	118														
40	41	2	841	22	11441	161														
45	46	2	1081	24	17281	213														
48	49	2	1153	25																
49	50	2	1275	27																
50	51	2	1301	27	22051	246														
54	55	2	1459	28	24949	258														
60	61	2	1801	31	33901	310														
75	76	2	2926	39																
90	91	2	4051	46																

And here are a few more partial results for the case $m = 8$.

Table 2. Constellation Enumerations for More 8PSK$\oplus n$

n	11	12	13	14	15
m	d_n	d_n	d_n	d_n	d_n
8	11969	16641	22569	29961	39041

4 Some Conjectures

We shall form a generating function for the sequences, d_n, where d_n is different for every m. Thus define $d_m(x) = \sum_{n=0}^{\infty} d_n x^n$. The following conjecture satisfies all numerical results quoted above,

Conjecture 1

$$d_m(x) = \frac{c_h(x)^{\frac{m}{h}}}{(1-x)^{\phi(m)+1}}$$

where ϕ is Euler's Totient Function, h is the square free part of m, and $c_h(x)$ is referred to as the h^{th} coordination polynomial. $c_h(x)$ is palindromic and $\deg(c_h(x)) = \phi(h)$.

The above conjecture omits to specify exactly the form of $c_h(x)$. This is an area of further research. However the following theorem determines $c_h(x)$ where h is a prime, and two following conjectures satisfy the computational results for $h = 2p$, p an odd prime, and $h = 15$, respectively,

Theorem 1

$$c_p(x) = \Phi_p(x), \qquad p \text{ prime}$$

Theorem 1 was conjectured by the author based on numerical computation. A proof was found by T.Kløve and it is given in Appendix A.

Conjecture 2

$$c_{2p}(x) = \sum_{k=0}^{\frac{p-3}{2}} x^k + x^{p-1-k} \sum_{i=0}^{k} \binom{p}{i} + x^{\frac{p-1}{2}} \sum_{i=0}^{\frac{p-1}{2}} \binom{p}{i}, \qquad p \text{ an odd prime}$$

Conjecture 3

$$c_{15}(x) = (1+x^8) + 7(x+x^7) + 28(x^2+x^6) + 79(x^3+x^5) + 130x^4$$

The following observation was also made,

Conjecture 4

$$m \mid \left(\frac{m^{n+1}-1}{m-1} - d_n \right)$$

From the computational results values of $c_h(x)$ have also been partially ascertained for various h as shown in Table 3.

All preceding coordination polynomials were computed from the d_n sequences using the following strategy. For instance, for $m = 6$ the d_n sequence is computed to be 1,7,19,37,61,91,127,169,..... Thus $d_6(x) = 1+7x+19x^2+37x^3+61x^4+91x^5+127x^6+169x^7+\ldots$. Note that $\phi(6)+1 = 3$ so, from Conjecture 1, we multiply $d_6(x)$ (truncated to degree 7) by $(1-x)^3$ to get $c'_6(x) = e(x) + x^2 + 4x + 1$, where $e(x)$ is some error term due to having truncated $d_6(x)$ to degree 7. In this case $e(x) = -217x^8 + 380x^9 - 169x^{10}$, which is evidently an error term so $c_6(x) = x^2 + 4x + 1$. The same strategy can be used to compute $c_h(x)$ for all d_n sequences in the table, and hence arrive at the preceding Conjectures 2 - 3 on the form of $c_h(x)$.

Table 3. Incomplete Coordination Polynomials for Various Composite h

h	$c_h(x)$
21	$1 + 9x + 45x^2 + 158x^3 + 432x^4 + 909x^5 + \ldots$?
30	$1 + 22x + 208x^2 + 874x^3 + 1480x^4 + \ldots$?
33	$1 + 13x + 91x^2 + 444x^3 + 1677x^4 + \ldots$?
35	$1 + 22x + 208x^2 + 874x^3 + 1480x^4 + \ldots$?

5 Triangle Patterns

An examination of number triangles may give a clue as to how to extend the previous conjectures on coordination polynomials to the more general case. On page 14 of [4] it was observed that the coordination polynomials for the dual lattice, A_d^*, satisfy the following 'coordinator' triangle.

```
                                1
                              1   1
                            1   4   1
                          1   5   5   1
                        1   6  16   6   1
                      1   7  22  22   7   1
                    1   8  29  64  29   8   1
                  1   9  37  93  93  37   9   1
                1   10  46 130 256 130  46  10   1
              1   11  56 176 386 386 176  56  11   1
            1   12  67 232 562 1024 562 232  67  12   1
```

The p^{th} line of the above triangle, p prime, also provides the coordination polynomials, $c_{2p}(x)$, for Conjectures 1 and 2 of this paper.

In the same way we can construct a partial triangle for the $c_{3p}(x)$ case, using our previous computational results. Thus,

```
                                1
                              1   4   1
                            1   5   ?   5   1
                          1   6  21   ?  21   6   1
                        1   7  28  79 130  79  28   7   1
                      1   8  36 114 282   ?  282 114  36   8   1
                    1   9  45 158 432 909   ?  909 432 158  45   9   1
                  1  10  55 212 635 1499  ?   ?   ? 1499 635 212  55  10   1
                1  11  66 277 902 2346  ?   ?   ?   ?   ? 2346 902 277  66  11   1
              1  12  78 354 1245 3525  ?   ?   ?   ?   ?   ?   ? 3525 1245 354  78  12   1
            1  13  91 444 1677 5124  ?   ?   ?   ?   ?   ?   ?   ?   ? 5124 1677 444  91  13   1
```

where each entry apart from those of the middle three columns seems to be the sum of the three entries immediately above, e.g. $158 = 8 + 36 + 114$. **Note that the only triangle entries directly computed from computational results are the sequences, 1,4,1, and 1,7,28,79,130,79,28,7,1, and 1,9,45,158,432,909, and 1,13,91,444,1677. All other numbers in the above triangle are nominally filled in to fit the 'sum of three' conjecture.** The $c_{3p}(x)$ coordination polynomial can be read from the p^{th} line of the previous triangle for p prime. For instance, $c_{15}(x) = (1 + x^8) + 7(x + x^7) + 28(x^2 + x^6) + 79(x^3 + x^5) + 130x^4$. Although we do not currently have an equation for $c_{3p}(x)$ it is worth noting that the following triangle is similar to the previous triangle,

											1							
									1	4	1							
							1	5	15	5	1							
						1	6	21	52	21	6	1						
					1	7	28	79	175	79	28	7	1					
				1	8	36	114	282	576	282	114	36	8	1				
			1	9	45	158	432	972	1869	972	432	158	45	9	1			
		1	10	55	212	635	1562	3273	6000	3273	1562	635	212	55	10	1		
	1	11	66	277	902	2409	5470	10385	19107	10385	5470	2346	902	277	66	11	1	
1	12	78	354	1245	3588	8781	18714	35412	60460	35412	18714	8781	3588	1245	354	78	12	1
13	91	444	1677	5187	13614	31083	62907	114586	190333	114586	62907	31083	13614	5187	1677	444	91	13	1

and satisfies the equation,

$$P_r(x) = \sum_{k=0}^{r-1} (3x)^k (1 + x + x^2)^{r-k-1}$$

for each line of the triangle, r, thus providing a clue as to the true form of $c_{3p}(x)$.

Appendix B sketches out an alternative strategy for the rapid computation of the coefficients, d_n, of $d_m(x)$ for general m, and is a good starting place for further research. It is hoped that the strategy of Appendix B may lead to a proof of the remaining conjectures, in particular Conjecture 1, and may also lead to a theorem for the construction of $c_h(x)$ in the general case.

6 Conclusion

This paper has presented computational results relating to the size of constellations formed from direct sums of PSK-type constellations. A theorem and a number of conjectures have been offered, comprising formulae for the rapid computation of sizes of such 'direct-sum' constellations. These formulae have application to error-control coding, random-walk statistics, algebraic number representations, and (polynomial) residue number theory. It remains to verify the conjectures.

7 Appendix A- Proof of Theorem 1 by T.Kløve

Let $p_r(n)$ denote the number of ordered partitions of n into r parts, that is

$$p_r(n) = |\{(a_1, a_2, \ldots, a_r) \in N^r \mid a_1 + a_2 + \ldots + a_r = n\}|$$

where $N = \{0, 1, 2, \ldots\}$. If,

$$n = a_1 + a_2 + \ldots + a_{r-1} + a_r$$

and $a_i \geq 1$ for $i = 1, 2, \ldots, r-1$, then

$$n - (r-1) = (a_1 - 1) + (a_2 - 1) + \ldots + (a_{r-1} - 1) + a_r$$

and vice versa. Hence the number of ordered partitions of n into r parts such that the first $r-1$ parts are positive is $p_r(n-(r-1))$.

Lemma 1 *We have*

$$\sum_{n=0}^{\infty} p_r(n)x^n = \frac{1}{(1-x)^r} \qquad \textit{and} \qquad \sum_{n=r-1}^{\infty} p_r(n-(r-1))x^n = \frac{x^{r-1}}{(1-x)^r}$$

Proof of Lemma 1: These are standard results from the theory of partitions:

$$\sum_{n=0}^{\infty} p_r(n)x^n = (1+x+x^2+x^3+\ldots)^r = \frac{1}{(1-x)^r}$$

and

$\sum_{n=r-1}^{\infty} p_r(n-(r-1))x^n = x^{r-1}\sum_{n=r-1}^{\infty} p_r(n-(r-1))x^{n-(r-1)} = \frac{x^{r-1}}{(1-x)^r}$. ∎

Lemma 2 *Let m be an odd prime. Then $d_n = p_{m+1}(n) - p_{m+1}(n-m)$.*

Proof of Lemma 2: d_n counts the number of distinct sums

$$a_1 w + a_2 w^2 + \ldots + a_m w^m + a_{m+1} \cdot 0 \tag{1}$$

where $a_i \geq 0$ for $i = 1, 2, \ldots, m+1$ and $a_1 + a_2 + \ldots + a_{m+1} = n$. Noting that $w + w^2 + \ldots w^m = 0$ we get d_n by counting all sums (1), this number is $p_{m+1}(n)$, and subtracting the number of sums where $a_i \geq 1$ for $i = 1, 2, \ldots, m$, this number is $p_{m+1}(n-m)$ (as explained above). ∎

Theorem 1 now follows from the two lemmas:

$\sum_{n=0}^{\infty} d_n x^n = \sum_{n=0}^{\infty} p_{m+1}(n)x^n - \sum_{n=0}^{\infty} p_{m+1}(n-m)x^n = \frac{1-x^m}{(1-x)^{m+1}} = \frac{\Phi_m(x)}{(1-x)^m}$

since $\Phi_m(x) = x^m + x^{m-1} + \ldots + 1$. ∎

8 Appendix B - A General Strategy for Computing the Size of PSK⊕ Constellations

Here a technique is proposed for the fast computation of the coefficients of $d_m(x)$ in the general case. Hopefully this may lead to a general proof of the conjectures of this paper, and a fast way to construct $c_h(x)$ in the general case, at least for m up to some large value. The technique will be illustrated by looking at the case where $m = 6$. Note that $\Phi_6(x) = x^2 - x + 1$. The steps of the technique are the following subsection headings.

8.1 Find all Forbidden Binary Patterns

$\Phi_6(x)$ implies the following polynomial equivalences:

$$x^2 + 1 = x \qquad \text{pattern is } 101000$$

$$x^3 + 1 = 0 \qquad \text{pattern is } 100100$$

These are the two **binary** patterns (polynomials) which are 'forbidden' for $m = 6$. The forbidden polynomials are the set of polynomials which are equivalent, mod $\Phi_m(x)$, to polynomials of lower hamming weight. Note that, for example, x^2-x+1 is not included as a 'forbidden' polynomial as it includes the polynomial $x^2 + 1$ as a sub-polynomial. In general, for $m = 2p$, p prime, there are only two forbidden polynomials, namely, $x^{p-1} + x^{p-3} + x^{p-5} + \ldots + x^2 + 1$, and, x^p+1. More generally, for large, composite m, there may be non-binary forbidden polynomials.

8.2 Enumerate all Length m Binary Words Which Avoid the Forbidden Patterns

For $m = 6$, and for Hamming Weights (hw) 0-6 we have the following cyclically distinct **binary** strings which avoid the forbidden patterns or any cyclic shift of the forbidden patterns.

hw = 0 000000
hw = 1 100000
hw = 2 110000
hw = 3 none
hw = 4 none
hw = 5 none
hw = 6 none

Each string of non-zero Hamming Weight has cyclic shift order 6. We will refer to the set of length m strings which avoid the forbidden patterns as the 'foundation' polynomials. These 'foundation' polynomials form the set $\mathbf{E}$. For $m = 6$ $|\mathbf{E}| = 3$. We will define there to be $e_{\mathrm{hw},m}$ cyclically distinct length m binary words in $\mathbf{E}$, $0 \leq \mathrm{hw} \leq m$. For $m = 6$, $e_{0,6} = 1$, $e_{1,6} = 1$, $e_{2,6} = 1$, $e_{3,6} = 0$, $e_{4,6} = 0$, $e_{5,6} = 0$, $e_{6,6} = 0$. Note that $e_{0,m} = 1$ $\forall m$.

8.3 Use Each Member of E as a 'Foundation' for Building All Length m Inequivalent Polynomials of Coefficient Weight n, mod $\Phi_m(x)$

The '1' positions of the 'foundation' polynomials of $\mathbf{E}$ mark the positions where we are allowed to add 'coefficient weight' to construct our inequivalent polynomials. It therefore follows that the number of inequivalent polynomials, d_n, satisfies,

$$d_n = 1 + m \sum_{k=1}^{n} \sum_{\mathrm{hw}=1}^{m} \binom{k-1}{k-\mathrm{hw}} e_{\mathrm{hw},m} \tag{2}$$

For $m = 6$,

$d_0 = 1$
$d_1 = 1 + 6 = 7$
$d_2 = 1 + 6 + 6(1 + 1) = 19$
$d_3 = 1 + 6 + 6(1 + 1) + 6(1 + 2 + 0) = 37$
$d_4 = 1 + 6 + 6(1 + 1) + 6(1 + 2 + 0) + 6(1 + 3 + 0 + 0) = 61$
$d_5 = 1 + 6 + 6(1 + 1) + 6(1 + 2 + 0) + 6(1 + 3 + 0 + 0) + 6(1 + 4 + 0 + 0 + 0) = 91$
$\ldots$ etc

These numbers agree with those of Table 1. The number of r-way ordered partitions adding to n is $p_r(n)$, and

$$p_r(n) = \binom{n + r - 1}{n}$$

Therefore we can rewrite (2) in terms of partitions as,

$$d_n = 1 + m \sum_{k=1}^{n} \sum_{\mathrm{hw}=1}^{m} p_{\mathrm{hw}}(k - \mathrm{hw}) e_{\mathrm{hw},m} \tag{3}$$

8.4 Comments on the Technique

The technique assumes that all polynomials in **E** have cyclic order m. It seems likely that this is true in general as d_n appears to satisfy $m|(d_n - 1)$ for all cases computed in Tables 1 and 2. A proof of Conjecture 1, and a proof of the general form of $c_h(x)$ may well follow if one can do the following for a given m,

1. Derive an efficient method to compute the 'forbidden' polynomials.
2. Derive an efficient method to compute the elements $e_{\mathrm{hw},m}$ of **E** from the forbidden polynomials.

For large m (e.g. perhaps $m = 105$?) there may be non-binary forbidden polynomials for which the above technique must be modified as follows: Consider, as an example, a 'hypothetical' forbidden polynomial, $F(x)$, of the following form:

$$F(x) = x^5 + 3x^2 + x + 2$$

Then it has an associated binary forbidden polynomial, $f(x)$, where,

$$f(x) = x^5 + x^2 + x + 1$$

We wish to disallow all polynomials built from the foundation $F(x)$ not $f(x)$. Let the cyclic order (over m) of $F(x)$ and $f(x)$ be v. Then we should include γ_n polynomials in our count for d_n, where

$$\gamma_n = v(\sum_{k=1}^{n} p_4(k - 4) - \sum_{k=1}^{n-3} p_4(k - 4)) = v \sum_{k=n-2}^{n} p_4(k - 4)$$

where the '3' in the summation limit of the previous equation is the coefficient weight (cw) of $F(x)$ minus the hamming weight of $F(x)$. In general, for a given forbidden polynomial $F(x)$ we include γ_n in our count for d_n where γ_n satisfies,

$$\gamma_n = v \sum_{k=n+\mathrm{hw}(F(x))-\mathrm{cw}(F(x))+1}^{n} p_{\mathrm{hw}(F(x))}(k - \mathrm{hw}(F(x)))$$

In the case where the forbidden polynomial is a binary polynomial $\mathrm{hw}(F(x)) = \mathrm{cw}(F(x))$ and γ_n for $F(x)$ is 0, as expected. Things will be further complicated if the cyclic order of $F(x)$ is lower than that of $f(x)$.

9 Acknowledgements

The author thanks S.J.Shepherd and D.A.Gillies for helpful discussions, and D.A.Gillies for writing software which independently confirmed results for the $m = 8$ case, and provided extra data for this case.

References

1. Parker, M.G.: VLSI Algorithms and Architectures for the Implementation of Number-Theoretic Transforms, Residue and Polynomial Residue Number Systems. **PhD thesis, School of Eng, University of Huddersfield, March 1995**
2. Safer, T.: Polygonal Radix Representations of Complex Numbers. Theoretical Computer Science. **210**, (1999) 159–171
3. Sloane, N.J.A.: An On-Line Version of the Encyclopedia of Integer Sequences. *http://www.research.att.com/ njas/sequences/index.html*, The Electronic Journal of Combinatorics. **1**, (1994) 1–5
4. Conway, J.H., Sloane, N.J.A.: Low Dimensional Lattices VII: Coordination Sequences. Proc. Royal Soc. **A453** (1997) 2369–2389
5. O'Keeffe, M.: Coordination Sequences for Lattices. Zeit. f. Krist. **210**, (1995) 905–908
6. Grosse-Kunstleve, R.W., Brunner, G.O.: Algebraic Description of Coordination Sequences and Exact Topological Densities for Zeolites. Acta Crystallographica. Section A. **A52**, (1996) 879–889
7. Huber, K.: Codes Over Gaussian Integers. IEEE Trans. on Inf. Theory. **40**, No 1, Jan. (1994) 207–216
8. Huber, K.: Codes Over Eisenstein-Jacobi Integers. Contemporary Mathematics. **168**, (1994) 165–179

Dual form of combinatorial problems and Laplace techniques

Lorenz Halbeisen
Department of Mathematics
Evans Hall 938
University of California at Berkeley
Berkeley, CA 94720 (USA)
E-mail: halbeis@math.berkeley.edu

Norbert Hungerbühler
Department of Mathematics
University of Alabama at Birmingham
452 Campbell Hall, 1300 University Boulevard
Birmingham, AL 35294-1170 (USA)
E-mail: buhler@uab.edu

1 Introduction

One of the central tools in enumerative combinatorics is that of generating functions. Generating functions can e.g., be used to find the asymptotic behaviour of the enumerating sequence (e.g., the Hardy-Ramanujan estimate for the partition function $P(n)$, see [3]) or even may yield an explicit formula for the solution (e.g., Rademacher's famous explicit formula for $P(n)$, see [6]).

Given a combinatorial problem, there are numerous ways to find the corresponding generating function. One possibility is to start with a recurrence relation, as, e.g., the recurrence for the Fibonacci numbers $(a_n)_{n\in\mathbb{N}_0} = (0, 1, 1, 2, 3, 5, 8, \ldots)$, which we write in the following form:

$$\begin{aligned} a_n &= a_{n-2} + a_{n-1} + \delta_{1,n} \qquad && \forall n \in \mathbb{Z}, \\ a_n &= 0 && \forall n < 0. \end{aligned} \tag{1}$$

($\delta_{k,n}$ denotes the Kronecker symbol.) The z-transformation method requires to multiply (1) by z^n and to sum over n. This yields an algebraic equation for the generating function $f(z) = \sum_{n=0}^{\infty} a_n z^n$, namely

$$f(z) = z^2 f(z) + z f(z) + z,$$

which is easily solved, giving $f(z) = \frac{z}{1-z-z^2}$. The Taylor expansion of this function yields

$$f(z) = \frac{z}{1-z-z^2} = \sum_{n=0}^{\infty} \left(\frac{z}{2}\right)^n \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{\sqrt{5}},$$

2000 Mathematics Subject Classification: 11B39, 05A15

i.e., we obtain the explicit Euler-Binet[1] formula for the Fibonacci numbers

$$a_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right).$$

A second way to find a generating function is to use Polya's index theorem. For example, let M be the set of all syntactic bracket figures with index n equal to the number of bracket pairs. For $n = 3$ we have the set M_3 of three bracket pairs:

$$M_3 = \{[\,][\,][\,], [[\,]]\,], [[\,][\,]], [[\,]][\,], [\,][[\,]]\}.$$

By

$$\begin{array}{rcl} M & \to & M_1 \times M \times M \cup M_0 \\ [a]b & \mapsto & ([\,], a, b) \\ \varnothing & \mapsto & \varnothing \end{array}$$

we have a bijection between the sets M and $M_1 \times M \times M \cup M_0$ which is additive, that is, $\mathrm{ind}([a]b) = 1 + \mathrm{ind}(a) + \mathrm{ind}(b)$. Then, by Polya's theorem, the relation between the sets translates directly into a relation for the generating function for the numbers $c_n = \mathrm{card}(M_n)$, namely,

$$f(z) = zf^2(z) + 1.$$

Taylor expansion of the solution $f(z) = \frac{1}{2z}(1 - \sqrt{1-4z}) = \sum_{n=0}^{\infty} c_n z^n$ yields the Catalan numbers

$$c_n = \frac{1}{n+1}\binom{2n}{n}.$$

A third way is to use methods from the theory of difference equations, which reach from continued fractions to Laplace transformation. As an example, we mention a recent theorem of Oberschelp (see [5]) that allows to transform a difference equation into a differential equation for the exponential generating function by a formal procedure. For example, the sequence Sloane-Plouffe sequence M1497 in [7], f_n, which counts the number of ways to build a sequence without repetition with n variables satisfies the recurrence $f_{n+1} = (n+1)f_n + 1$. Oberschelp's theorem requires the exchange

$$\tbinom{n}{k} f_{n+s-k} \quad \longleftrightarrow \quad \frac{z^k}{k!} f^{(s)},$$

i.e., to replace f_{n+1} by f', nf_n by zf', f_n by f, and 1 by e^z. This procedure yields the ordinary differential equation $(1-z)f' - f = e^z$ with the solution $f(z) = \frac{e^z}{1-z}$

[1]This formula was derived by Jacques P.M. Binet in 1843, although the result was known to Euler and to Daniel Bernoulli more than a century earlier.

determined by $f(0) = 1$. Since $f(z)$ is the exponential generating function, we get in fact $f_n = n!(1 + \frac{1}{1!} + \ldots + \frac{1}{n!})$.

Experience shows that the situation becomes considerably more delicate as soon as the problem requires to solve partial difference equations. In this article we want to describe methods which allow us to calculate the generating function from a recurrence relation. The idea is to link the Laplace transform directly to generating functions by interpreting the Fourier formula for the inverse Laplace transform as a residual integral. The reader who is not familiar with the Laplace or Fourier transformation might consult [1] or [8]. The idea is certainly not new; however, we would like to show that it applies also to more complicated (e.g., non-local) partial difference equations.

2 Auxiliary Results

2.1 Laplace transformation

Let $(a_n)_{n\in\mathbb{Z}}$, $a_n = 0$ for $n < 0$, be a sequence of real numbers with generating function $f(z) = \sum_{n\in\mathbb{Z}} a_n z^n$. We call

$$A(z) := \sum_{n\in\mathbb{Z}} a_n \chi_{[n,n+1[}(z)$$

the associated step-function. Here, χ_I denotes the characteristic function of the set I. Then the following theorem holds.

Theorem 1 *If the Laplace transform $\mathscr{L}[A]$ of the associated step-function A exists; it is related to the generating function f by*

$$\mathscr{L}[A](s) = \frac{1}{s}(1 - e^{-s}) f(e^{-s}).$$

Proof. Since we assume A to have at most exponential growth, we may transform term by term and get

$$\mathscr{L}[A](s) = \sum_{n=0}^{\infty} a_n \mathscr{L}[\chi_{[n,n+1[}].$$

Writing $\chi_{[n,n+1[} = H(\cdot - n) - H(\cdot - (n+1))$, where $H = \chi_{[0,\infty[}$ denotes the Heaviside function, and using that $\mathscr{L}[H](s) = \frac{1}{s}$, we obtain, by applying the basic rules for the Laplace transformation,

$$\mathscr{L}[A](s) = \sum_{n=0}^{\infty} a_n \frac{1}{s} e^{-ns}(1 - e^{-s}),$$

which is what we claimed. □

The following calculation provides a useful variant of Theorem 1: If $\frac{1}{z}g(e^{-z})$ is the Laplace transform of a piecewise smooth function G, we have by Fourier's formula for the inverse Laplace transformation that, for every point $x \in \mathbb{R}_+$ where G is continuous,

$$G(x) = \frac{1}{2\pi i}\,\mathrm{pv}\int_\Gamma \frac{1}{z}\,g(e^{-z})e^{xz}dz.$$

Here, Γ is the curve $\Gamma : \mathbb{R} \to \mathbb{C}, t \mapsto s+it$, with $s \in \mathbb{R}$ large enough, and "pv" denotes the principal value. If we denote $\Gamma_n : [0, 2\pi[\to \mathbb{C}, t \mapsto z := s + i(t + 2n\pi)$, we have

$$G(x) = \frac{1}{2\pi i}\,\mathrm{pv}\sum_{n\in\mathbb{Z}}\int_{\Gamma_n} \frac{1}{z}\,g(e^{-z})e^{xz}dz. \tag{2}$$

Observe that, by Fourier-series expansion, we have, for $x \notin \mathbb{Z}$,

$$\sum_{n\in\mathbb{Z}} \frac{1}{s+i(t+2n\pi)}e^{x(s+i(t+2n\pi))} = \frac{e^{\lceil x\rceil(s+it)}}{e^{s+it}-1},$$

where $\lceil\cdot\rceil$ denotes the ceiling function, i.e., $\lceil x\rceil$ is the smallest integer larger than or equal to x. Hence, by substituting $u = e^{-z}$, we obtain from (2) with $n = \lfloor x\rfloor$,

$$G(x) = \frac{1}{2\pi i}\int_\gamma \frac{g(u)}{1-u}\,\frac{du}{u^{n+1}} \tag{3}$$

where $\gamma : [0, 2\pi[\to \mathbb{C}, t \mapsto e^{-s}e^{it}$, and where $\lfloor\cdot\rfloor$ denotes the floor function, i.e., $\lfloor x\rfloor$ is the largest integer smaller than or equal to x. Thus, if g is analytic in a neighborhood of 0, we may interpret the integral in (3) as the Cauchy residue integral for the nth Taylor coefficient of the function $\frac{g(u)}{1-u}$. Thus, we have the following corollary.

Corollary 1 *Assume f and g_n are analytic functions in a neighborhood of 0 and a_n is given by*

$$a_n = \frac{1}{2\pi i}\,\mathrm{pv}\int_\Gamma \frac{1}{z}\,g_n(e^{-z})e^{xz}dz \tag{4}$$

for some (and hence any) $x \in]n, n+1[$ and Γ as above. If $\lim_{z\to 0}\frac{f(z)-g_n(z)}{z^n} = 0$ for all $n \in \mathbb{N}_0$, then $\frac{f(z)}{1-z}$ is the generating function of the sequence a_n.

Let us briefly mention some advantages that the use of the Laplace transformation provides: Suppose we are given a generating function $f(u)$. Only in simple cases it is possible to use direct Taylor expansion to obtain a formula for the coefficient a_n of u^n. Also, the Cauchy residue $a_n = \mathrm{Res}_{u=0}\frac{f(u)}{u^{n+1}}$ or (in case of a meromorphic function

$f)$ $a_n = -\sum \mathrm{Res}_{u\neq 0} \frac{f(u)}{u^{n+1}}$ is often difficult to calculate. In such a situation, it may be helpful to split the residues via the Laplace transformation (as in the calculation preceding Corollary 1) in order to obtain an expansion (or at least an asymptotic formula) for the a_n. To illustrate this, let us consider the example of the generating function of the Bernoulli numbers

$$f(u) = u\cot u = 1 + \sum_{n=1}^{\infty} \frac{(-1)^n 2^{2n} B_{2n}}{(2n)!} u^n.$$

According to Theorem 1, the Laplace transform of the associated step-function G is

$$g(s) = \frac{1-e^{-s}}{s} f(e^{-s})$$

and we may use the Fourier formula to invert g: $\mathscr{L}^{-1}[g](t) = \sum \mathrm{Res}\, g(s)e^{ts}$. The singularities of $g(s)e^{ts}$ are located at $s_{k,m} = m\pi i - \log(k\pi)$, $k \in \mathbb{N}$, $m \in \mathbb{Z}$. For $t \in \mathbb{Z}$ we have

$$\mathrm{Res}_{s_{k,m}}\, g(s)e^{ts} = \begin{cases} -\frac{1-k\pi}{s_{k,m}(k\pi)^t} & \text{if } m \text{ is even,} \\ -\frac{1+k\pi}{s_{k,m}(-k\pi)^t} & \text{if } m \text{ is odd.} \end{cases}$$

Combining residues for m and $-m$, we can easily sum the residues for fixed k over all m and obtain

$$\mathscr{L}^{-1}[g](t) = -\sum_{k=1}^{\infty} \frac{1}{(k\pi)^{2\lfloor t/2 \rfloor}}.$$

(Notice that one obtains a formula for $\sum_{m=1}^{\infty} \frac{1}{a^2+m^2}$ by expanding e^{ax} on $]-\pi,\pi[$ in a Fourier series.) Since $t \in \mathbb{Z}$ (G jumps in $\mathbb{Z}$), we finally get the zeta-function formula for the Bernoulli numbers:

$$B_{2n} = (-1)^{n+1} \frac{2(2n)!}{(2\pi)^{2n}} \sum_{k=1}^{\infty} \frac{1}{k^{2n}}.$$

A second benefit of the Laplace transformation are the various rules. For example, by the rule $\mathscr{L}[f'](s) = s\,\mathscr{L}[f](s) - f(0)$, we have, for $f_z(t) := t^z$, that

$$\mathscr{L}[f_z'](s) = s\,\mathscr{L}[f_z](s) = z\,\mathscr{L}[f_{z-1}](s).$$

Hence, for fixed s, the analytic function

$$h_s(z) := \mathscr{L}[f_z](s) = \int_0^{\infty} t^z e^{-st} dt$$

solves the difference equation $s\,h_s(z) = z\,h_s(z-1)$. In particular, for $s = 1$, we obtain Euler's integral representation of the Gamma-function. It is a particular feature of

the Laplace-transformation method that it can be used to determine the analytic continuation of a discrete function. The Laplace transformation also yields a functional connection between the exponential generating function $e(x)$ and the ordinary generating function $f(x)$ of a sequence a_n. In fact, we have

$$\mathscr{L}[e](s) = \mathscr{L}\left[\sum_{n=0}^{\infty} \frac{a_n}{n!} x^n\right](s) = \sum_{n=0}^{\infty} \frac{a_n}{n!} \underbrace{\mathscr{L}[x^n](s)}_{\frac{n!}{s^{n+1}}} = \tfrac{1}{s} f\left(\tfrac{1}{s}\right).$$

The translation-rule $\mathscr{L}[f(t-c)](s) = e^{-sc}\mathscr{L}[f(t)](s)$ for $c \geq 0$ allows us to transform a (linear) difference equation into an algebraic equation for the transformed function (this feature is similar to the z-transformation). In particular, it is possible to reduce a linear partial difference equation with n variables to an equation with $n-1$ variables. For an example see Section 3.4 or 3.5.

Another virtue of the Laplace transformation appears when one looks for an asymptotic expansion of a sequence or (which is a similar thing) when one treats difference equations which show oscillation and damping effects. If one is only interested in the stationary state, one can already, at the level of the transformed function, identify terms which lead to exponentially decaying terms in the solution and drop them for the rest of the calculation.

2.2 The dual of a linear difference equation

Many combinatorial problems lead to partial difference equations. As a prototype example, we investigate the two dimensional case.

Let $X \subset \mathbb{Z}^2$. For a map $p : X \to \mathbb{R}$, we consider the linear equation

$$p(z) = \sum_{\{\zeta \in X : \zeta \in \operatorname{spt} a_z\}} a_z(\zeta) p(\zeta) \qquad (*)$$

where we assume that the cardinality of the support of a_z ($\operatorname{spt} a_z \subset X$) is finite for all $z \in X$, i.e., that the sum in $(*)$ is always finite. A set $A \subset X$ is called *stable* if for all maps $f : A \to \mathbb{R}$ there exists a unique solution p of $(*)$ such that $p|_A = f$. A triple $(X, A, *)$ is called *triangular* if X can be written as $X = (x_i)_{i \in \mathbb{N}}$ in such a way that, for all $i \in \mathbb{N}$, there holds $\operatorname{spt} a_{x_i} \subset A \cup \{x_1, \ldots, x_{i-1}\}$, and for all $z \in A$: $\operatorname{spt} a_z = \{z\}$ and $a_z(z) = 1$. In particular we have that, for a triangular triple $(X, A, *)$, the set A is stable.

Now, let $(X, A, *)$ be triangular and $f : A \to \mathbb{R}$ be given. Then, for any fixed $x = x_i \in X$, the solution p of $(*)$ in x is a finite linear combination of the values of f on A, i.e.,

$$p(x) = \sum_{\zeta \in A} \alpha_x(\zeta) f(\zeta).$$

In order to determine the weights $\alpha_x(\zeta)$, we proceed as follows:

(i) Put a red mark on x.

(ii) Replace each red mark on $y \in X \setminus A$ by a blue one on y and by $a_y(\zeta)$ many red marks on ζ for all $\zeta \in \operatorname{spt} a_y$.

(iii) Iterate (ii) until no more red marks on $X \setminus A$ exist.

If n denotes the maximum of the set $\{i\colon$ there is a red mark on $x_i\}$, then, in each iteration step, n decreases at least by one due to the triangular structure. Hence, the iteration process terminates. If we denote by $\tilde{q}(\zeta)$ the number of red marks on ζ, the quantity

$$\sum_{\zeta \in X} \tilde{q}(\zeta) p(\zeta)$$

is invariant during the iteration. Hence, we obtain the result that after the iteration is completed the number of (red) marks on $\zeta \in A$, i.e., $\tilde{q}(\zeta)$, equals the weight $\alpha_x(\zeta)$.

If we denote by $q(\zeta)$ the final number of marks (blue or red) on ζ (i.e., after termination of the iteration), the iteration process described above translates into a partial difference equation for the function q:

$$q(z) = \sum_{\{\zeta \in A_x : z \in \operatorname{spt} a_\zeta\}} a_\zeta(z) q(\zeta) \qquad (**)$$

with $q(x) = 1$ and with $A_x := \operatorname{tr} x \setminus A$, where $\operatorname{tr} x$ is the equivalence class of x with respect to the transitive hull of the relation $u \sim v :\iff u \in \operatorname{spt} a_v, v \notin A$. Notice that $(A_x, \{x\}, **)$ is triangular and finite. Let us summarize this result in a theorem.

Theorem 2 *If $(X, A, *)$ is triangular with prescribed values f on A, then the weights α_x in the solution formula $p(x) = \sum_{\zeta \in A} \alpha_x(\zeta) f(\zeta)$ can be determined by the iteration scheme (i)–(iii) or, equivalently, by solving the dual linear recursion $(**)$ with initial value $q(x) = 1$.*

Many transformation problems (for example the boustrophedon transformation in [4]) can be described as follows: Let $(X, A, *)$ be triangular; then we fix sets $A' = \{a_1, a_2, \ldots\} \subset A$ and $X' = \{b_1, b_2, \ldots\} \subset X$ and prescribe $f(a_i) = \phi_i$ and $f = 0$ on $A \setminus A'$. If we denote the solution $\psi_i = p(b_i)$, the mapping $\Psi_{X,X',A,A',*}\colon (\phi_i) \mapsto (\psi_i)$ is a linear transformation of sequences, the associated linear mapping (ALM). The problem to find its matrix (or the matrix of the inverse transformation) can often be solved by using the Laplace transformation technique for the partial difference equation for the *weights* $(**)$ even in cases where it is not possible to use directly the

Laplace transformation in the *original* partial difference equation $(*)$. We will see some examples in the following section.

Before we discuss the examples, we close this section by stating a simple path-counting lemma.

Lemma 1 *Suppose the coefficient functions a in $(*)$ satisfy the following invariance property for all $z = (n,k)$ and $z' = (n,k')$ in $X = \mathbb{Z}^2$:*

$$a_z(n+i, k+j) = a_{z'}(n+i, k'+j), \quad \forall i,j \in \mathbb{Z}. \tag{5}$$

Suppose, furthermore, that the column $\{(0,k)\colon k \in \mathbb{Z}\}$ is stable and that p denotes the solution of $(*)$ with prescribed values α_k on $(0,k)$. Then the column $\{(N,k)\colon k \in \mathbb{Z}\}$ is stable for

$$\tilde{p}(z) = \sum_{\{\zeta \in X\}} \bar{a}_z(\zeta)\tilde{p}(\zeta) \tag{$\dagger$}$$

where $\bar{a}_{u+v}(u) := a_u(u+\bar{v})$ and $\overline{(i,j)} := (i,-j)$. Finally, if we prescribe the values α_k on (N,k) for the equation $(\dagger)$, then $\tilde{p}(0,k) = p(N,k)$.

Proof of Lemma 1: We may interpret $(*)$ as a directed graph G with $a_z(\zeta)$ many edges from ζ to z. If we set $\alpha_k := \delta_{k,k_0}$, then $p(N,k)$ is the number of paths in G from $(0,k_0)$ to (N,k). If we flip the graph horizontally by $z \mapsto \bar{z}$ and invert the orientation of the edges, we obtain a graph G'. Now, $(\dagger)$ describes G' and $\tilde{p}(0,k)$ is the number of paths in G' from (N,k_0) to $(0,k)$ which equals, by construction, the number of paths in G from $(0,k_0)$ to (N,k).

For general (α_k) the claim follows by linearity. □

3 Examples and applications

3.1 The Fibonacci numbers and a variant of Faulhaber's formula

Let $X = \{(k,n) : n \geq k \geq 0\}$ and $A = \big\{(k,n) \in X\colon n \in \{k, k+1\}\big\}$. Further let

$$a_{(k,n)}(i,j) = \begin{cases} \delta_{k,i}\delta_{n-1,j} + \delta_{k+1,i}\delta_{n-1,j} & \text{for } (k,n) \notin A, \\ \delta_{k,i}\delta_{n,j} & \text{otherwise,} \end{cases}$$

in the equation $(*)$. This is easily seen to be triangular. For the sets $A' = \{(k,k+1) \in A\}$ and $X' = \{(0,n) \in X\colon n \geq 0\}$, we have that the ALM $\Psi_{X,X',A,A',*}$ applied to the

sequence $(1,1,\ldots)$ yields the Fibonacci sequence $\big(f(n)\big)_n$. Let us calculate the weights via $(**)$:

$$q(k,n) = q(k,n+1) + q(k-1,n+1)$$

with $q(0,l) = 1$. This is (up to renumbering) just the recursion for the binomial numbers, i.e., we get the "shallow diagonal" sum formula connecting Pascal's triangle to the Fibonacci numbers:

$$f(n+1) = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n-k}{n-2k}.$$

The binomial weights always occur for this type of equation: For another example, let $p(k,n) := \sum_{i=1}^{n} i^k$. Obviously, for fixed k, p is a polynomial in n of degree $k+1$. Faulhaber's famous formula expresses this polynomial in the basis $\{1, n, n^2, n^3, \ldots\}$, and the coefficients in this basis involve the Bernoulli numbers. Here, we want to express the polynomial in the basis $\left\{\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \binom{n}{3}, \ldots\right\}$. Consider again the "binomial" difference equation $f(k,n) = f(k,n-1) + f(k+1,n-1)$, this time on $X = \mathbb{N}_0^2$, with initial data $f(0,n) = p(k,n-1)$ for fixed k. The weights for the dual equation clearly are, as above, the binomial coefficients; hence, $p(k,n-1) = \sum_{i=1}^{n} \binom{n}{i} f(i,0)$, and it remains to find $f(i,0)$. Since $f(1,n) = n^k$, we use

$$\sum_{i=0}^{n} \binom{n}{i} i! S_2(k,i) = n^k, \tag{6}$$

where S_2 denotes the Stirling number of the second kind (see next section). Indeed, each term in the sum may be interpreted as the number of sequences in $\{1,\ldots,n\}^k$ with exactly i different numbers. Thus, $f(i+1,0) = i! S_2(k,i)$ and we recover the well known formula

$$p(k,n) = \sum_{i=1}^{n} \binom{n+1}{i+1} i! S_2(k,i),$$

which one also gets by summing (6).

3.2 The Stirling numbers

The Stirling numbers of the first kind $S_1(n,k)$ count the permutations of n distinct objects that can be written with exactly k disjoint cycles (cf. [2]). They can be computed recursively as follows:

$$S_1(n+1,k) \;:=\; n \cdot S_1(n,k) + S_1(n,k-1),$$

where $S_1(1,k) := \delta_{1,k}$.

Let $\tilde{S}_n(k) := S_1(n,k)$; then $\tilde{S}_n(k)$ satisfies the recurrence $\tilde{S}_{n+1}(k) = n\tilde{S}_n(k)+\tilde{S}_n(k-1)$. Let $L_n(s)$ denote the Laplace transform of the associated step-function of $\tilde{S}_n(k)$. Then we get

$$L_{n+1}(s) = nL_n(s) + e^{-s}L_n(s) = L_n(s)(n + e^{-s}),$$

with $L_1(s) = \frac{1}{s}(1 - e^{-s})$. Hence,

$$L_n(s) = \frac{1}{s}(1 - e^{-s})\prod_{j=1}^{n-1}(j + e^{-s}).$$

Thus, by Theorem 1 we find that

$$f_n(u) = \prod_{j=1}^{n-1}(j + u)$$

is the generating function for $\big(S_1(n,k)\big)_k$.

The Stirling numbers of the second kind $S_2(n,k)$ count the number of groupings of n distinct objects into k disjoint (nonempty) groups. They can be computed recursively as follows:

$$S_2(n+1,k) \;:=\; k \cdot S_2(n,k) + S_2(n,k-1),$$

where $S_2(1,k) := \delta_{1,k}$.

Let $\tilde{S}_k(n) := S_2(n,k)$; then $\tilde{S}_k(n)$ satisfies the recurrence $\tilde{S}_k(n) = k\tilde{S}_k(n-1) + \tilde{S}_{k-1}(n-1)$. Let $L_k(s)$ denote the Laplace transform of the associated step-function of $\tilde{S}_k(n)$. Then we obtain $L_k(s) = ke^{-s}L_k(s) + e^{-s}L_{k-1}(s)$. Therefore,

$$L_k(s) = L_{k-1}(s)\frac{e^{-s}}{1 - k\,e^{-s}} = L_1(s)\prod_{j=2}^{k}\frac{e^{-s}}{1 - j\,e^{-s}}$$

with $L_1(s) = \frac{1}{s}$. Thus, by Theorem 1, we get that

$$f_k(u) = \prod_{j=1}^{k}\frac{u}{1 - j\,u}$$

is the generating function for $\big(S_2(n,k)\big)_n$.

It is well known that the matrix of the Stirling numbers of the first and second kind are inverse in the sense that

$$f(n) = \sum_{i=1}^{n} S_1(n,i)e(i)$$

if and only if

$$e(n) = \sum_{i=1}^{n} (-1)^{n-i} S_2(n,i) f(i).$$

Instead of proving this rather special formula, we now investigate more general conditions which still imply an inversion formula of the above type.

3.3 An inversion formula

We consider the following situation: Given a linear equation $(*)$ with $X = \mathbb{N}_0 \times \mathbb{Z}$, which satisfies the invariance property (5), we suppose that with $A := \{(0,k) : k \in \mathbb{Z}\}$ the triple $(X, A, *)$ is triangular. We set $A' := \{(0,k) : k \in \mathbb{N}_0\}$ and $X' := \{(n,0) : n \in \mathbb{N}_0\}$ and consider the mapping $\Psi_{X,X',A,A',*} \colon (\phi_i) \mapsto (\psi_i)$. Notice that the equation $(**)$ for the weights inherits the invariance property (5), and hence we can apply Lemma 1 to $(**)$ and obtain

$$\tilde{p}(z) = \sum_{\{\zeta \in X\}} \bar{a}_z(\zeta) \tilde{p}(\zeta), \tag{$\dagger\dagger$}$$

with $\tilde{p}(n,0) = \delta_{n,0}$, where $\bar{a}_{u+v}(u) := a_{u+\bar{v}}(u)$. Then we have

$$\psi_n = \sum_{i=0}^{\infty} \tilde{p}(n,i) \phi_i. \tag{7}$$

Now we invert the previous equation: Let $Y := \mathbb{N}_0 \times \mathbb{N}_0$ and $Y' := \{(0,k) : k \in \mathbb{N}_0\}$. For any fixed $z \in X$, we can replace $(*)$ equivalently by the equation

$$p(\zeta_0) = \frac{1}{a_z(\zeta_0)} p(z) - \sum_{\{\zeta \in \operatorname{spt} a_z \setminus \{\zeta_0\}\}} \frac{a_z(\zeta)}{a_z(\zeta_0)} p(\zeta) =: \sum_{\{\zeta \in \operatorname{spt} a'_{\zeta_0}\}} a'_{\zeta_0}(\zeta) p(\zeta) \tag{$*'$}$$

for arbitrary $\zeta_0 \in \operatorname{spt} a_z$. Assume that for any $z \in X$ we can—by choosing a suitable ζ_0—replace $(*)$ by $(*')$ in such a way that

- the coefficients a'_z respect the invariance relation (5),
- the triple $(Y, Y', *')$ is triangular.

The equation for the weights for $(*')$ is

$$q(z) = \sum_{\{\zeta \in A_{(0,0)} : z \in \operatorname{spt} a'_\zeta\}} a'_\zeta(z) q(\zeta), \tag{$**'$}$$

with initial condition $q(0,0) = 1$ (because $(**')$ satisfies (5)). Then we have

$$\phi_n = \sum_{i=0}^{\infty} q(i,-n)\psi_i. \tag{8}$$

Hence, in view of (8) and (7), q and $\tilde{p}$ are inverse matrices, where q and $\tilde{p}$ satisfy certain difference equations which are related in the described manner. Notice also that, by choosing ζ_0 (see above), there is a certain freedom in the coefficients a' which can be useful sometimes.

As an example of the previous result we investigate a generalization of the Stirling numbers.

Let us define $a_{(n,k)}(i,j) := c(i)\delta_{i,n-1}\delta_{j,k} + d(i)\delta_{i,n-1}\delta_{j,k+1}$, where c and d are non-vanishing functions. Then the procedure described above yields the following proposition.

Proposition 1 *The numbers $s_1(n,k)$, $s_2(n,k)$ for $(n,k) \in \mathbb{Z} \times \mathbb{Z}$, defined by*

$$s_1(n,k) = c(n-1)s_1(n-1,k) + d(n-1)s_1(n-1,k-1)$$

and

$$s_2(n,k) = -\frac{c(n)}{d(n)}s_2(n,k-1) + \frac{1}{d(n-1)}s_2(n-1,k-1)$$

with $s_1(0,m) = s_2(m,0) = \delta_{m,0}$ are inverse in the sense that

$$\psi_n = \sum_{i=0}^{\infty} s_1(n,i)\phi_i \iff \phi_n = \sum_{i=0}^{\infty} s_2(i,n)\psi_i\,.$$

For special choices of the functions c and d, one easily gets e.g., the inversion formulas for the Stirling numbers ($c(n) = n$, $d(n) = 1$), the binomial numbers ($c(n) = 1$, $d(n) = 1$), or the numbers $Q_l(n) := \binom{n}{l} l!$ counting the number of ways to build sequences of length l with n objects without repetitions ($c(l) = -\frac{1}{l}$, $d(l) = \frac{1}{l}$)—guess what the inverse numbers are!

3.4 The partition numbers

As a further example, we consider the number $p(n,k)$ of partitions of an integer n into parts larger than or equal to k. This leads to the (non-local) partial difference equation

$$p(n,k) = p(n-k,k) + p(n,k+1), \tag{9}$$

with $p(n,k) = 0$ for $k > n > 0$ and $p(n,n) = 1$. In the above setting, the problem reads as follows: $X = \mathbb{N}^2$, $A = \{(n,k) : k \geq n\}$, $A' = \{(n,n) : n \in \mathbb{N}\}$ and $X' = \{(n,1) : n \in \mathbb{N}\}$, and for $(n,k) \in X \setminus A$ we have

$$p(n,k) = \sum_{i,j \in \mathbb{N}} (\delta_{i,n-k}\delta_{j,k} + \delta_{i,n}\delta_{j,k+1})p(i,j)\,. \tag{10}$$

The ALM $\Psi_{X,X',A,A',(10)}$ maps the sequence $(1,1,\ldots)$ into the sequence $p(n,1) = P(n)$ of the partition numbers. The equation for the weights is given by

$$q(n,k) = q(n,k-1) + q(n+k,k)$$

with initial conditions $q(n,1) = 1$ for $n \leq N$ and $q(n,k) = 0$ for $n > N$. Then we have $P(N) = \sum_{i=1}^{N} q(i,i)$. By renumbering, this is equivalent to saying

$$\tilde{q}(n,k) = \tilde{q}(n,k-1) + \tilde{q}(n-k,k) \tag{11}$$

with $\tilde{q}(n,1) = 1$ for all n, $\tilde{q}(n,k) = 0$ for $n \leq 0$, and $P(N) = \sum_{i=1}^{N} \tilde{q}(i, N-i+1)$. Note that $\tilde{q}(n,k)$ no longer depends on N. Laplace transformation of (11) with respect to the first variable with k fixed yields

$$r_k(s) = \frac{1}{1-e^{-sk}} r_{k-1}(s)$$

with initial value $r_1(s) = \frac{1}{s}$ (since $\tilde{q}(1,k) = 1$ for $k \in \mathbb{N}$). Thus, we have

$$r_k(s) = \frac{1}{s} \prod_{j=2}^{k} \frac{1}{1-e^{-js}}$$

and, by Theorem 1, the generating function $g_k(u)$ of $\big(\tilde{r}_k(n)\big)_n$ is given by

$$g_k(u) = \prod_{j=1}^{k} \frac{1}{1-u^j}\,.$$

From this, it is easy to derive Euler's classical generating function $E(u)$ of the partition numbers $P(N)$. But, by interpreting $\tilde{q}(n,k)$ as the number of partitions of $n-1$ into k or less parts (and hence $P(n-1) = \tilde{q}(n,n-1) = \tilde{q}(n,n)$), we immediately get from the above calculation together with Corollary 1 that

$$E(u) = \prod_{j=1}^{\infty} \frac{1}{1-u^j}\,. \tag{12}$$

Also, if $f(s)$ denotes the Laplace transform of E, it follows from (12) that

$$\frac{1}{s}(1-e^{-s})\prod_{j=1}^{\infty}(1-e^{-js}) = f(s)\sum_{j=1}^{\infty}(-1)^{\lfloor\frac{j}{2}\rfloor}e^{-st_j},$$

where $t_j = 0, 1, 2, 5, 7, \ldots$ are the pentagonal numbers. Laplace inversion of the last equation yields Euler's formula $\sum_{j=1}^{\infty}(-1)^{\lfloor\frac{j}{2}\rfloor}P(n-t_j) = \delta_{n,0}$.

What about counting weighted partitions? Let $f\colon \mathbb{N}\to\mathbb{R}$ be a weight function with the meaning that we count partitions into i parts $f(i)$ many times, or—what is the same thing by considering Ferrers diagram—count partitions which largest part of size i, $f(i)$ many times. Then the calculation above gives the generating function for this problem:

$$\sum_{i=1}^{\infty}\frac{f(i)u^i}{\prod_{j=1}^{i}1-u^j}.$$

So, choosing, e.g., f as the characteristic function of the even numbers, we compute $\big(e(n)\big)_n = (0, 1, 1, 3, 3, 6, 7, 12, 14, \ldots)$.

To conclude this section let us compute the inverse of the ALM $\Psi_{X,X',A,A',(10)}$. Let us put a red mark on (L, L). In view of (10) we can replace a red mark on (n, k) (for $n \geq k > 1$) by a red mark on $(n, k-1)$, a negative red mark on $(n-k+1, k-1)$ and a blue mark on (n, k). This game terminates when all red marks are in $A \setminus A'$ (these marks are multiplied by 0) or in X' (where a mark on $(i, 1)$ is multiplied by ψ_i). Hence, $\phi_L = \sum_{n=1}^{L}\psi_n\,\omega(L, n)$, where $\omega(L, n)$ denotes the number of red marks on $(n, 1)$.

To compute $\omega(L, n)$, we consider the directed, finite graph G_L with vertices $\{(n, k) : L \geq n \geq k \geq 1\}$ and an edge from (n, k) to (n', k') if $k' = k-1$ and $n' = n$ (this edges are called v-edges) or if $k' = k$ and $n' = n-k$ (this edges are called h-edges of length k). Now let $W_L(n)$ be the number of paths through the graph G_L from the vertex (L, L) to $(n, 1)$, such that all h-edges have different length and each path is weighted by $+1$ if the number of h-edges contained in the path is even, otherwise it is weighted by -1. It is easy to see that $W_L(n) = \omega(L, n)$. To compute $W_L(n)$, let us first define the function $w(m, l, s)$, which is the number of weighted paths from (m, m) to $(m-l, 1)$, such that the maximum of the lengths of h-edges contained in the path equals s (where $s = 0$ means that the path contains no h-edge). For the function $w(m, l, s)$, we have

$$w(m,l,s) = \begin{cases} 1 & \text{if } l = s = 0, \\ 0 & \text{if } s > l \text{ or } s > \lfloor\frac{m}{2}\rfloor, \\ -\sum_{j=1}^{s} w(m-s, l-s, s-j) & \text{otherwise.} \end{cases}$$

Now, by construction, we obtain

$$W_L(n) = \sum_{s=0}^{\lfloor \frac{L}{2} \rfloor} w(L, L-n, s).$$

For example, for $L = 12$, we get $\big(W_{12}(n)\big)_n = (1, -1, -2, 0, 2, 0, 1, 0, 0, -1, -1, 1)$ and, in fact, $P(12) - P(11) - P(10) + P(7) + 2P(5) - 2P(3) - P(2) + P(1) = 77 - 56 - 42 + 15 + 2 \cdot 7 - 2 \cdot 3 - 2 + 1 = 1$.

3.5 A path counting problem

We consider paths in a three-dimensional lattice: Starting point of the paths is a point $(x, 0, 0)$, $x \in \mathbb{N}_0$, on the x-axis. If (x, y, z) is a point on the path, then a unit step in positive y or z direction is allowed or a step of length $y + z + 1$ in negative x direction. We want to count the number $H_M(x)$ of allowed paths starting in $(x, 0, 0)$ which end in a given set $M \subset \mathbb{Z}^3$.

The dual of this problem is given by the non-local linear difference equation

$$q_{z,y}(x) = q_{z-1,y}(x) + q_{z,y-1}(x) + q_{z,y}(x - y - z - 1) \tag{13}$$

with $q_{z,y}(x) := 0$ if one of the numbers x, y, or z is negative and $q_{0,0}(0) := 1$. We already used an index notation because we want to Laplace-transform equation (13) with respect to the variable x. First, we have $Q_{0,0}(s) = \frac{1}{s}$, since $q_{0,0}(x) = 1$ for $x \geq 0$. Laplace transformation of (13) yields

$$Q_{z,y}(s) = Q_{z-1,y}(s) + Q_{z,y-1}(s) + e^{-s(y+z+1)} Q_{z,y}(s).$$

Considering s as a parameter, the solution of this difference equation in y and z is given by

$$Q_{z,y}(s) = \frac{1}{s} \binom{z+y}{z} \frac{1}{\prod_{j=2}^{z+y+1} (1 - e^{-js})}.$$

Thus, the generating function of $q_{z,y}(x)$ is

$$f_{z,y}(u) = \binom{z+y}{z} \prod_{j=1}^{z+y+1} \frac{1}{1-u^j}.$$

Hence, using the notation of Section 3.4,

$$q_{z,y}(x) = \tilde{r}_{z+y+1}(x) \binom{z+y}{z}.$$

Finally, the solution to our path counting problem is given by the formula

$$H_M(\xi) = \sum_{(\xi-x,y,z)\in M} \tilde{r}_{z+y+1}(x)\binom{z+y}{z}.$$

For example, let us count the paths starting in $(\xi,0,0)$ with at most h unit steps in z direction and such that the total number of unit steps in negative x and in positive y direction equals ξ. This corresponds to the set $M = \{(x,y,z) \in \mathbb{Z}^3 : x = y,\, z \le h\}$, and the solution formula yields

$$H_M(\xi) = \sum_{z\le h, x\le \xi} \tilde{r}_{z+\xi-x+1}(x)\binom{z+\xi-x}{z}.$$

3.6 Local linear difference equations

For $X = \{(k,l) : 0 \le k \le l\}$ and $A = \{(k,l) : l \in \{k, k+1, k+2\}\}$, we consider the model equation

$$z(k,l) = a_1 z(k,l-1) + a_2 z(k+1,l-1) + a_3 z(k+2,l-1). \tag{14}$$

$(X, A, (14))$ is triangular and, for $X' = \{(0,l) : l \ge 3\}$, the equation for the weights is

$$q(k,l) = a_1 q(k,l+1) + a_2 q(k-1,l+1) + a_3 q(k-2,l+1) \tag{15}$$

with initial condition $q(k,L) = \delta_{k,0}$ for a fixed $L \ge 0$. Laplace transformation of (15) with respect to the variable k with l fixed gives $Q_l(s) = Q_{l+1}(s)(a_1 + a_2 e^{-s} + a_3 e^{-2s})$ with initial condition $Q_L(s) = \frac{1}{s}(1 - e^{-s})$. The solution is

$$Q_l(s) = \frac{1}{s}(1 - e^{-s})(a_1 + a_2 e^{-s} + a_3 e^{-2s})^{L-l},$$

and Theorem 1 gives, for the generating function of the sequence $\big(q(k,l)\big)_k$, the function $(a_1 + a_2 u + a_3 u^2)^{L-l}$. Multinomial expansion yields

$$q(k,l) = \sum_{k_2+2k_3=k} \binom{L-l}{L-l-k_2-k_3, k_2, k_3} a_1^{L-l-k_2-k_3} a_2^{k_2} a_3^{k_3}.$$

Since (15) does not stop the iteration when a mark lies on A, we have to compensate by setting $\tilde{q}(k,k+2) = q(k,k+2)$, $\tilde{q}(k,k+1) = q(k,k+1) - a_1 q(k,k+2)$, and $\tilde{q}(k,k) = q(k,k) - a_1 q(k,k+1) - a_2 q(k-1,k+1)$. Then, if α_z is given on $z \in A$ as initial data for (14), we get the solution

$$z(0,l) = \sum_{i=2}^{l} \sum_{j=0}^{2} \alpha_{(i-j,i)} \tilde{q}(i-j,i). \tag{16}$$

In particular, if $\alpha_{(k+j,k)} = x_j$ (for $j = 0,1,2$), $z(0,l)$ is the solution of $x_n = a_1 x_{n-1} + a_2 x_{n-2} + a_3 x_{n-k}$ with initial values x_0, x_1, x_2 and (16) is a root-free representation of the solution.

References

[1] R. Beals: "Advanced mathematical analysis: periodic functions and distributions, complex analysis, Laplace transform and applications." Springer, New York 1973.

[2] J. H. Conway and R. K. Guy: "The book of numbers." Copernicus, New York 1996.

[3] G. H. Hardy and S. Ramanujan: Asymptotic formulae in combinatory analysis. *Proc. London Math. Soc.* (2) **17**(1918), 75–115.

[4] J. Millar, N. J. A. Sloane and N. E. Young: A new operation on sequences: the boustrophedon transform. *J. Comb. Theory* **A 76**(1996), 44–54.

[5] W. Oberschelp: Solving linear recurrences from differential equations in the exponential manner and vice versa, *in* "Applications of Fibonacci numbers, Vol. 6," (G. E. Bergum, A. N. Philippou and A. F. Horadam, Ed.), pp. 365–380, Kluwer Acad. Publ., Dordrecht 1996.

[6] H. Rademacher: On the expansion of the partition function in a series. *Ann. of Math.* **44**(1943), 416–422.

[7] N. J. A. Sloane, S. Plouffe: "The encyclopedia of integer sequences." Academic Press, San Diego 1995.

[8] A. Zygmund: "Trigonometric series." Cambridge University Press, Cambridge 1977.

Generalized Multipartitioning *

Alain Darte†
LIP, ENS-Lyon, 46, Allée d'Italie, 69007 Lyon, France.
Alain.Darte@ens-lyon.fr

Daniel Chavarría-Miranda Robert Fowler John Mellor-Crummey
Dept. of Computer Science MS-132, Rice University, 6100 Main, Houston, TX USA
{danich,johnmc,rjf}@cs.rice.edu

August 27, 2001

Abstract

Multipartitioning is a strategy for partitioning multi-dimensional arrays among a collection of processors. With multipartitioning, computations that require solving one-dimensional recurrences along each dimension of a multi-dimensional array can be parallelized effectively. Previous techniques for multipartitioning yield efficient parallelizations over three-dimensional domains only when the number of processors is a perfect square. This paper considers the general problem of computing optimal multipartitionings for d-dimensional data volumes on an arbitrary number of processors. We describe an algorithm that computes an optimal multipartitioning for this general case, which enables multipartitioning to be used for performing efficient parallelizations of line-sweep computations under arbitrary conditions.

Finally, we describe a prototype implementation of generalized multipartitioning in the Rice dHPF compiler and performance results obtained when using it to parallelize a line sweep computation for different numbers of processors.

1 Introduction

Line sweeps are used to solve one-dimensional recurrences along each dimension of a multi-dimensional discretized domain. This computational method is the basis for Alternating Direction Implicit (ADI) integration — a widely-used numerical technique for solving partial differential equations such as the Navier-Stokes equation [4, 13, 15] — and is also at the heart of a variety of other numerical methods and solution techniques [15]. Parallelizing computations based on line sweeps is important because these computations address important classes of problems and they are computationally intensive.

Recurrences along a dimension that line sweeps are used solve, serialize computation of each line along that dimension. If a dimension with such recurrences is partitioned, it induces serialization between computations on different processors. Using standard block uni-partitionings, in which each processor is assigned a single hyper-rectangular block of data, there are two classes of alternative partitionings. *Static block unipartitionings* involve partitioning some set of dimensions of the data domain, and assigning each processor one contiguous hyper-rectangular volume. To achieve significant parallelism for a line sweep computation with this type of partitionings requires exploiting wavefront parallelism within each sweep. In wavefront computations, there is a tension between using small messages to maximize parallelism by minimizing the length of pipeline fill and drain phases, and using larger messages to minimize communication overhead in the computation's steady state when the pipeline is full. *Dynamic block unipartitionings* involve partitioning a single data dimension, performing line sweeps in all unpartitioned data dimensions locally, transposing the data to localize the data along the previously partitioned dimension, and then performing the remaining sweep locally. While dynamic block unipartitionings achieve better efficiency during a (local) sweep over a single dimension compared to a (wavefront) sweep using static block unipartitionings, they require transposing *all* of the data to per-

*This research was supported in part by the Los Alamos National Laboratory Computer Science Institute (LACSI) through LANL contract number 03891-99-23 as part of the prime contract (W-7405-ENG-36) between the DOE and the Regents of the University of California.

†This work performed while a visiting scholar at Rice University.

form a complete set of sweeps, whereas static block unipartitionings communicate only data at partition boundaries.

To support better parallelization of line sweep computations, a third sophisticated strategy for partitioning data and computation known as *multipartitioning* was developed [4, 13, 15]. Multipartitioning distributes arrays of two or more dimensions among a set of processors so that for computations performing a directional sweep along any one of the array's data dimensions, (1) all processors are active in each step of the computation, (2) load-balance is nearly perfect, and (3) only a modest amount of coarse-grain communication is needed. These properties are achieved by carefully assigning each processor a balanced number of tiles between each pair of adjacent hyperplanes that are defined by the cuts along any partitioned data dimension. We describe multipartitionings in detail in Section 2. A study by van der Wijngaart [18] of implementation strategies for hand-coded parallelizations of ADI Integration found that 3D multipartitionings yield better performance than both static block unipartitionings and dynamic block unipartitionings.

All of the multipartitionings described in the literature to date consider only one tile per processor per hyperplane of a multipartitioning. The most general class of multipartitionings described in the literature is known as *diagonal multipartitionings*. While diagonal multipartitionings are optimal in two dimensions, for three dimensions diagonal multipartitionings are optimal only when the number of processors is a prime or a perfect square. This paper considers the general problem of computing optimal multipartitionings for d-dimensional data volumes on an arbitrary number of processors. We describe an algorithm that computes an optimal multipartitioning for this general case, which enables multipartitioning to be used for performing efficient parallelizations of line-sweep computations under arbitrary conditions.

In the next section, we describe prior work in multipartitioning. Then, we present our strategy for computing generalized multipartitionings. This has three parts: an objective function for computing the cost of a line sweep computation for a given multipartitioning, a cost-model-driven algorithm for computing the dimensionality and tile size of the best multipartitioning, and an algorithm for computing a mapping of tiles to processors. Finally, we describe a prototype implementation of generalized multipartitioning in the Rice dHPF compiler for High Performance Fortran. We report preliminary performance results obtained using it to parallelize a computational fluid dynamics benchmark.

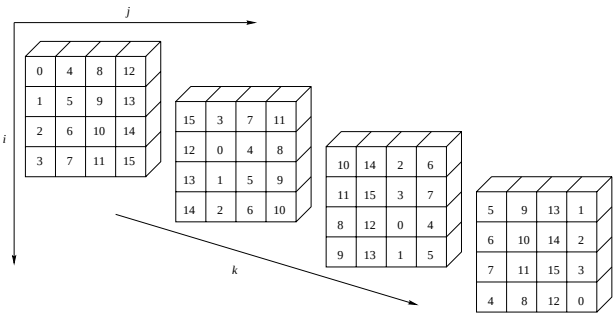


Figure 1: 3D Multipartitioning on 16 processors.

2 Background

Johnsson *et al.* [13] describe a two-dimensional domain decomposition strategy, now known as a multipartitioning, for parallel implementation of ADI integration on a multiprocessor ring. They partition both dimensions of a two-dimensional domain to form a $p \times p$ grid of tiles. They use a tile-to-processor mapping $\theta(i, j) = (i - j) \bmod p$, where $0 \leq i, j < p$. Using this mapping for an ADI computation requires each processor to exchange data with only its two neighbors in a linear ordering of the processors, which maps nicely to a ring.

Bruno and Cappello [4] devised a three-dimensional partitioning for parallelizing three-dimensional ADI integration computations on a hypercube architecture. They describe how to map a three-dimensional domain cut into $2^d \times 2^d \times 2^d$ tiles on to 2^{2d} processors. They use a tile to processor mapping $\theta(i, j, k)$ based on Gray codes. A Gray code $g_s(r)$ denotes a one-to-one function defined for all integers r and s where $0 \leq r < 2^s$, that has the property that $g_s(r)$ and $g_s((r + 1) \bmod 2^s)$ differ in exactly one bit position. They define $\theta(i, j, k) = g_d((j + k) \bmod 2^d) \cdot g_d((i + k) \bmod 2^d)$, where $0 \leq i, j, k < 2^d$ and $\cdot$ denotes bitwise concatenation. This θ maps tiles adjacent along the i or j dimension to adjacent processors in the hypercube, whereas tiles adjacent along the k dimension map to processors that are exactly two hops distant. They also show that no hypercube embedding is possible in which adjacent tiles always map to adjacent processors.

Naik *et al.* [15] describe *diagonal multipartitionings* for two and three dimensional problems. Diagonal multipartitionings are a generalization of Johnsson *et al*'s two dimensional partitioning strategy. This

class of multipartitionings is also more broadly applicable than the Gray code based mapping described by Bruno and Cappello. The three-dimensional diagonal multipartitionings described by Naik *et al.* partition data into $p^{\frac{3}{2}}$ tiles arranged along diagonals through each of the partitioned dimensions. Figure 1 shows a three-dimensional multipartitioning of this style for 16 processors; the number in each tile indicates the processor that owns the block. In three dimensions, a diagonal multipartitioning is specified by the tile to processor mapping $\theta(i,j,k) = ((i-k) \bmod \sqrt{p})\sqrt{p} + ((j-k) \bmod \sqrt{p})$ for a domain of $\sqrt{p} \times \sqrt{p} \times \sqrt{p}$ tiles where $0 \leq i,j,k < \sqrt{p}$.

More generally, we observe that diagonal multipartitionings can be applied to partition d-dimensional data onto an arbitrary number of processors p by cutting the data into an array of p^d tiles. For two dimensions, this yields a unique optimal multipartitioning (equivalent to the class of partitionings described by Johnsson *et al.* [13]). However, for $d > 2$, cutting data into so many tiles yields inefficient partitionings with excess communication. For three or more dimensions, diagonal multipartitioning is optimal only when $p^{\frac{1}{d-1}}$ is integral.

3 General Multipartitioning

Bruno and Cappello noted that multipartitionings need not be restricted to having only one tile per processor per hyperplane of a multipartitioning [4]. How general can multipartitioning mappings be? A sufficient condition to support load-balanced line-sweep computation is that in any hyperplane of the partitioning, each processor must have the same number of tiles. We call any hyperplane in which each processor has the same number of tiles *balanced*. This raises the question: can we find a way to partition a d-dimensional array into tiles and assign the tiles to processors so that each hyperplane is balanced? The answer is yes. However, such an assignment is possible if and only if the number of tiles in each hyperplane along any dimension is a multiple of p. We describe a "regular" solution (regular to be defined) to this general problem that enables us to guarantee that the neighboring tiles of a processor's tiles along a direction of a data dimension all belong to a single processor — an important property for efficient computation on a multipartitioned distribution.

In Section 4, we define an objective function that represents the execution time of a line-sweep computation over a multipartitioned array. In Section 5, we present an algorithm that computes a partitioning of a multidimensional array into tiles that is optimal with respect to this objective. In Section 6, we develop a general theory of modular mappings for multipartitioning. We apply this theory to define a mapping of tiles to processors so that each line sweep is perfectly balanced over the processors.

We use the following notations in the subsequent sections:

- p denotes the number of processors. We write $p = \prod_{j=1}^{s} \alpha_j^{r_j}$, to represent the decomposition of p into prime factors.
- d is the number of dimensions of the array to be partitioned. The array is of size $n_1, \ldots, n_d$. The total number of array elements $n = \prod_{i=1}^{d} n_i$.
- γ_i, for $1 \leq i \leq d$, is the number of tiles into which the array is cut along its i-th dimension. We consider the d-dimensional array as a $\gamma_1 \times \ldots \times \gamma_d$ array of tiles. In our analysis, we assume γ_i divides n_i evenly and do not consider alignment or boundary problems that must be handled when applying our mappings in practice if this assumption is not valid.

To ensure each hyperplane is balanced, the number of tiles it contains must be a multiple of p; namely, for each $1 \leq i \leq d$, p should divide $\prod_{j \neq i} \gamma_j$.

4 Objective Function

We consider the cost of performing a line sweep computation along each dimension of a multipartitioned array. The total computation cost is proportional to the number of elements in the array, n. A sweep along the i-th dimension consists of a sequence of γ_i computation phases (one for each hyperplane of tiles along dimension i), separated by $\gamma_i - 1$ communication phases. The work in each hyperplane is perfectly balanced, with each processor performing the computation for its own tiles. The total computational work for each processor is roughly $\frac{1}{p}$ of the total work in the sequential computation. The communication overhead is a function of the number of communication phases and the communication volume. Between two computation phases, a hyperplane of array elements is transmitted – the boundary layer for all tiles computed in first phase. The total communication volume for a phase communicated along dimension i is $\prod_{j \neq i} n_j$ elements, i.e., $\frac{n}{n_i}$. Therefore, the total execution time for a sweep along dimension i can be approximated by the following formula:

$$T_i(p) = K_1 \frac{n}{p} + (\gamma_i - 1)(K_2 + K_3 \frac{n}{n_i})$$

where K_1 is a constant that depends on the sequential computation time, K_2 is a constant that depends on the cost of initiating one communication phase (start-up), and K_3 is a constant that depends of the cost of transmitting one array element. Define $\lambda_i = K_2 + K_3 \frac{n}{n_i}$, λ_i depends on the domain size, number of processors and machine's communication parameters. The total cost of the algorithm, sweeping in all dimensions, is thus

$$T(p) = d\left(K_1 \frac{n}{p} - K_2 - K_3 \sum_{i=1}^{d} \frac{n}{n_i}\right) + \sum_{i=1}^{d} \gamma_i \lambda_i$$

Remark: if all communications are performed with perfect parallelism, with no overhead, then the term with K_3 is actually divided by p. We assume here that, in general, the cost of one communication phase is an affine function of the volume of transmitted data.

Assuming that p, n, and the n_i's are given, what we can try to minimize is $\sum_{i=1}^{d} \gamma_i \lambda_i$.

There are several cases to consider. If the number of phases is the critical term, the objective function can be simplified to $\sum_i \gamma_i$. If the volume of communications is the critical term, the objective function can be simplified to $\sum_i \frac{\gamma_i}{n_i}$, which means it is preferable to partition dimensions that are larger into relatively more pieces. For example, in 3D, even for a square number of processors (e.g., $p = 4$), if the data domain has one very small dimension, then it is preferable to use a 2D partitioning with the two larger ones rather than a 3D partitioning. Indeed, if n_1 and n_2 are at least 4 times larger than n_3, then cutting each of the first two dimensions into 4 pieces ($\gamma_1 = \gamma_2 = 4$, $\gamma_3 = 1$) leads to a smaller volume of communication than a "classical" 3D partitioning in which each dimension is cut into 2 pieces ($\gamma_1 = \gamma_2 = \gamma_3 = 2$). The extra communication while sweeping along the first two dimensions is offset by the absence of communication in the local sweep along the last dimension.

5 Finding the Partitioning

In this section, we address the problem of minimizing $\sum_i \gamma_i \lambda_i$ for general λ_i's, with the constraint that, for any fixed i, p divides the product of the γ_j's excluding γ_i. We give a practical algorithm, based on an exhaustive search, exponential in s (the number of factors) and the r_i's (see the decomposition of p into prime factors), but whose complexity in p grows slowly.

From a theoretical point of view, we do not know whether this minimization problem is NP-complete, even for a fixed dimension $d \geq 3$, even if all λ_i are equal to 1, or if there is an algorithm polynomial in $\log p$ or even in $\log s$ and the $\log r_i$'s. We suspect that our problem is strongly NP-complete, even if the input is s and the r_i's, instead of p. If p has only one prime factor, we point out that a greedy approach leads to a polynomial (i.e., polynomial in $\log r$) algorithm (see [10]). However, we do not know if an extension of this greedy approach can lead to a polynomial algorithm for an optimal solution in the general case.

5.1 Properties of Potentially Optimal Partitionings

We say that $(\gamma_i)_{1 \leq i \leq d}$ – or (γ_i) for short – is a **valid solution** if, for each $1 \leq i \leq d$, p divides $\prod_{j \neq i} \gamma_j$. Furthermore, if $\sum_i \gamma_i \lambda_i$ is minimized, we say that (γ_i) is an **optimal solution**. We start with some basic properties of valid and optimal solutions.

Lemma 1 *Let (γ_i) be given. Then, (γ_i) is a valid solution if and only if, for each factor α of p, appearing r_α times in the decomposition of p, the total number of occurrences of α in all γ_i is at least $r_\alpha + m_\alpha$, where m_α is the maximum number of occurrences of α in any γ_i.*

Proof: Suppose that (γ_i) is a valid solution. Let α be a factor of p appearing r_α times in the decomposition of p, let m_α be the maximum number of occurrences of α in any γ_i, and let i_0 be such that α appears m_α times in γ_{i_0}. Since p divides the product of all γ_i excluding γ_{i_0}, α appears at least r_α times in this product. The total number of occurrences of α in all of the γ_i is thus at least $r_\alpha + m_\alpha$. Conversely, if this property is true for any factor α, then for any product of $(d-1)$ different γ_i's, the number of occurrences of α is at least $r_\alpha + m_\alpha$ minus the number of occurrences in the γ_i that is not part of the product, and thus must be at least r_α. Therefore, p divides this product and (γ_i) is a valid solution. ■

Thanks to Lemma 1, we can interpret (and manipulate) a valid solution (γ_i) as a distribution of the factors of p into d bins. If a factor α appears r_α times in p, it must appear $(r_\alpha + m_\alpha)$ times in the d bins, where m_α is the maximal number of occurrences of α in a bin. As far as the minimization of $\sum_i \lambda_i \gamma_i$ is concerned, no other prime number can appear in the γ_i without increasing the objective function. The following lemma refines the result of Lemma 1 for a potentially optimal solution.

Lemma 2 *Let (γ_i) be an optimal solution. Then, each factor α of p, appearing r_α times in the decomposition of p, appears exactly $(r_\alpha + m_\alpha)$ times in (γ_i), where m_α is the maximum number of occurrences of α in any γ_i. Furthermore, the number of occurrences of α is m_α in at least two γ_i's.*

Proof: Let (γ_i) be an optimal solution. By Lemma 1, each factor α, $0 \leq j < s$, that appears r_α times in p, appears at least $(r_\alpha + m_\alpha)$ times in (γ_i). The following arguments hold independently for each factor α.

Suppose m_α occurrences of α appear in some γ_{i_0} and no other γ_i. Remove one α from γ_{i_0}. Now, the maximum number of occurrences of α in any γ_i is $m_\alpha - 1$ and we have $(r_\alpha + m_\alpha) - 1 = r_\alpha + (m_\alpha - 1)$ occurrences of α. By Lemma 1, we still have a valid solution, and with a smaller cost. This contradicts the optimality of (γ_i). Thus, there are at least two bins with m_α occurrences of α.

If c, the number of occurrences of α in (γ_i), is such that $c > r_\alpha + m_\alpha$, then we can remove one α from any nonempty bin, containing fewer than m_α occurrences. We now have $c-1 \geq r_\alpha + m_\alpha$ occurrences of α and the maximum is still m_α (since at least two bins had m_α occurrences of α). Therefore, according to Lemma 1, we still have a valid solution, and with smaller cost, again a contradiction. ■

We can now give some upper and lower bounds for the maximal number of occurrences of a given factor in any bin.

Lemma 3 *In any optimal solution, for any factor α appearing r_α times in the decomposition of p, we have $\lceil \frac{r_\alpha}{d-1} \rceil \leq m_\alpha \leq r_\alpha \leq (d-1)m_\alpha$ where m_α is the maximal number of occurrences of α in any bin and d is the number of bins.*

Proof: By Lemma 2, we know that the number of occurrences of α is exactly $r_\alpha + m_\alpha$, and at least two bins contain m_α elements. Thus, $r_\alpha + m_\alpha = 2 * m_\alpha + e$ where e is the total number of elements in $(d-2)$ bins, excluding two bins of maximal size m_α. Since $0 \leq e \leq (d-2)m_\alpha$, then $m_\alpha \leq r_\alpha \leq (d-1)m_\alpha$. Finally, any valid solution requires that p divides the product of all of the factor instances in each group of $d-1$ bins. Thus, there must be r_α instances of α in $d-1$ bins, and thus $m_\alpha \geq \lceil \frac{r_\alpha}{d-1} \rceil$. ■

5.2 Exhaustive Enumeration of Potentially Optimal Partitionings

We now give an algorithm that finds an optimal solution by generating all possible partitionings (γ_i) that satisfy the necessary optimality conditions given by Lemma 2, and determining which one yields the lowest cost partitioning. We also evaluate how many candidate partitions there are and present the complexity of our algorithm. For the complexity, we are not interested in the exact number of solutions that respect the conditions of Lemma 2, but in the order of magnitude, especially when the number of bins d is fixed (and small, equal to 3, 4, or 5), but when p can be large (up to 1000 for example), since this is the situation we expect to encounter in practice when computing multipartitionings.

The C program of Figure 2 generates, in linear time, all possible distributions into d bins, satisfying the $(r+m)$ optimality condition of Lemma 2, of a given factor appearing r times in the decomposition of p. It is inspired by a program [16] for generating all partitions of a number, which is a well-studied problem (see [17]) since the mathematical work of Euler and Ramanujam. The procedure `Partitions` first selects the maximal number m in a bin, and uses the recursive procedure `P(n,m,c,t,d)` that generates all distributions of n elements in $(d-t+1)$ bins (from index t to index d), where each bin can have at most m elements and at least c bins should have m elements. Therefore the initial call is `P(r+m,m,2,1,d)`.

We now prove the correctness of the program. The procedure `P` selects a number of elements for the bin number t and makes a recursive call with parameter $t+1$ for the selection in the next bin. It is thus clear that all generated solutions are different since each iteration of a loop selects a different number of elements for each bin. It remains to prove that all solutions generated by `P` are valid (the total number of elements should be $r+m$, each bin should have less than m elements, and there should be at least c bins with m elements), and that all solutions are generated. For that we prove that `P(n,m,c,t,d)` is always called with parameters for which there exists at least a valid solution, that all possible numbers of elements are selected and only those.

Let us first consider the loop in function `Partitions`. Thanks to Lemma 3, we know that the maximal number of elements in a bin is between $\lceil \frac{r}{d-1} \rceil$ and r. Furthermore, for each such m, there is indeed at least one valid solution with $(r+m)$ elements and two maxima equal to m (if $d \geq 2$), for example the solution where the first two bins have m elements and the $(d-2)$ other bins contain a total

```
// Precondition: d >= 2
void Partitions(int r, int d) {
  int m;
  for (m = (r+d-2)/(d-1); m <= r; m++)
    P(r+m,m,2,1,d);
}

void P(int n, int m, int c, int t, int d) {
  int i;
  if (t==d)
    bin[t] = n;
  else {
    for (i=max(0,n-(d-t)*m);
              i<=min(m-1,n-c*m); i++) {
      bin[t] = i;
      P(n-i,m,c,t+1,d);
    }
    if (n>=m) {
      bin[t] = m;
      P(n-m,m,max(0,c-1),t+1,d);
    }
  }
}
```

Figure 2: Program for generating all possible distributions for one factor.

of $(r-m)$ elements, one possibility being with the $r-m$ elements distributed so that $q = \lfloor \frac{r-m}{m} \rfloor$ bins contain m elements and one contains $(r-m-mq)$ elements. Therefore, if the function `P` is correct, the function `Partitions` is also correct.

To prove the correctness of the function `P` we prove by induction on $d-t+1$ (the number of bins) that there is at least one valid solution if and only if $c \leq d-t+1$ and $cm \leq n \leq (d-t+1)m$ and that `P` generates all of them if these conditions are satisfied. These conditions are simple to understand: we need at least cm elements (so that at least c bins have m elements) and at most $(d-t+1)m$ elements, otherwise at least one bin will contain more than m elements.

The terminal case is clear: if we have only one bin and n elements to distribute, the bin should contain n elements. Furthermore, if there is a solution, we should have $c \leq 1$ and $n = m$ if $c = 1$, i.e., $c \leq d-t+1$ and $cm \leq n \leq (d-t+1)m$.

The general case is more tricky. We first select the number of elements i in the bin number t and recursively call `P` for the remaining bins. If we select strictly less than m elements (this selection is in the loop), we will still have to select c bins with m elements for the remaining $(d-t)$ bins, with $(n-i)$ elements. Therefore, the number i that we select should not be too small, nor too large, and we should have $cm \leq n-i \leq m(d-t)$, i.e., $n-(d-t)m \leq i \leq n-cm$. Furthermore, i should be strictly less than m, nonnegative, and less than n. Since c is always positive, the constraint $i \leq n-cm$ ensures $i \leq n$. If the parameters are correct for the bin number t, we also have $c \leq d-t+1$ and if $c = d-t+1$, then the loop has no iteration, thus for an i selected in the loop, we have $c \leq d-t$. Therefore the recursive call `P(n-i,m,c,t+1,d)` has correct parameters. Finally, if we select m elements for the bin t (after the loop), this is possible only if m is less than n of course, and then it remains to put $(n-m)$ elements into $(d-t)$ bins, with a maximum of m, and at least $\max(0, c-1)$ maxima. Again, the recursive call has correct parameters since we decreased both c and $(d-t)$ and removed m elements.

5.3 Complexity of the Exhaustive Enumeration

For generating all optimal solutions to our minimization problem, we first decompose p into prime factors (complexity $O(\sqrt{p})$ by a standard algorithm, but could be less), we then generate all potentially optimal solutions that satisfy Lemma 2 for each factor (with the function `Partitions`), and we combine them while keeping track of the best overall solution. For evaluating each solution, we need to build the corresponding (γ_i)'s and add them. Each γ_i is at most p and is obtained by at most $\sum_i r_i \leq \log_2 p$ multiplications of numbers less than p. Therefore, building each γ_i costs at most $(\log_2 p)^3$. The overall complexity (excluding the cost of the decomposition of p into prime factors) is thus the product of the complexity of the function `Partitions` (which is the number of solutions generated by the algorithm) times $(\log_2 p)^3$. Therefore, it remains to evaluate the number of solutions generated by the function `Partitions`.

Consider first the case of a number p, product of simple prime factors, in particular the product of the first s prime numbers: $p = \prod_{i=1}^{s} \pi_i$ where π_i is the i-th prime number. For each factor, there are $\frac{d(d-1)}{2}$ possible distributions (picking two bins where to put one copy of each element), so the total number of solutions is $\left(\frac{d(d-1)}{2}\right)^s$. Now, the i-th prime number is approximated by $i \log i$ (see for example the Prime Pages [5]). Therefore, when p grows, we have

$$\begin{aligned} \log p &= \sum_{i=1}^{s} \log \pi_i \sim \sum_{i=1}^{s} \log(i \log i) \\ &\sim \sum_{i=1}^{s} \log i \sim \int_1^s \log x \, \mathrm{d}x \sim s \log s \end{aligned}$$

since divergent series with equivalent nonnegative terms are equivalent. Therefore $\log p \sim s \log s$ and $\frac{\log p}{\log\log p} \sim s$. The total number of solutions for p is thus $\left(\frac{d(d-1)}{2}\right)^{\frac{\log p}{\log\log p}(1+o(1))}$, thus at least of order $p^{\frac{f(d)(1+o(1))}{\log\log p}}$, for a small function $f(d)$ of d. We can prove that this situation (when p is the product of single prime factors) is actually representative of the worst case (in order of magnitude). The proof is too long to be provided here but is available in the extended version of this paper [10].

Theorem 1 *When p grows, the total number of generated solutions is less than $p^{\frac{f(d)(1+o(1))}{\log\log p}}$ where $f(d)$ is a small function of d.*

6 Finding the Mapping

In Section 5, we determined a particular way of cutting the array so as to optimize communications: after partitioning, we get an array (of tiles) whose size is (γ_i) for which the objective is minimized. But until now, we made the assumption that we will be indeed able to assign tiles to processors so that each slice of the array contains exactly the same number of tiles per processor (load-balancing property). This is not certain yet.

The only property we have until now is that the (γ_i) form is a **valid solution**: for each $1 \leq i \leq d$, p divides $\prod_{j \neq i} \gamma_j$, the defining property of a completely balanced multipartitioning. Our main result is that this condition is sufficient to guarantee a mapping of processors to tiles. Our proof is constructive. For any valid solution (γ_i), optimal or not, with or without the additional property of Lemma 2, we give an automatic way to assign a processor number to each tile so that the load-balancing property is satisfied. This assignment is done through the use of modular mappings, defined below. The proof of our construction is much too long to be given here. We refer the reader to the extended version of this paper [10] for details of the proof and interesting properties of modular mappings.

The solution we build is one particular assignment, out of a set of legal mappings. It is not unique, and more experiments might show that they are not all equivalent in terms of execution time, for example because of communication patterns. But, currently, with our objective function (Section 4), the network topology is not taken into account yet and all valid mappings are considered equally good.

6.1 Modular Mappings

Consider the assignment in Figure 1. Can we give a formula that describes it? There are 16 processors that can be represented as a 2-dimensional grid of size 4×4. For example the processor number $7 = 4 + 3$ can be represented as the vector $(3, 1)$, in general (r, q) where r and q are the remainder and the quotient of the Euclidean division by 16. The assignment in the figure corresponds to the assignment $(i - k \bmod 4, j - k \bmod 4)$, which is what we call a **multi-dimensional modular mapping**.

Definition 1 *A mapping $M_m : \mathbb{Z}^d \longrightarrow \mathbb{Z}^{d'}$ defined by $M_m(\vec{i}) = (M\vec{i}) \bmod \vec{m}$ where M is an integral $d \times d'$ matrix and $\vec{m}$ is an integral positive vector of dimension d' is a* **modular mapping**.

With a multi-dimensional mapping, each tile is assigned to a "processor number" in the form of a vector. The product of the components of $\vec{m}$ is equal to the number of processors. It then remains to define a one-to-one mapping from the hyper-rectangle $\{\vec{j} \in \mathbb{Z}^{d'} \mid \vec{0} \leq \vec{j} < \vec{m}\}$ (inequalities component-wise) onto the processor numbers. This can be done by viewing the processors as a virtual grid of dimension d' of size $\vec{m}$. The mapping $M_{\vec{m}}$ is then an assignment of each tile (described by its coordinates in the d-dimensional array of tiles) to a processor (described by its coordinates in the d'-dimensional virtual grid). (Note: in our construction, we will need only the case $d' = d - 1$.)

The following definitions summarize the notions of modular mappings and of modular mappings that satisfy the load-balancing property.

Definition 2 *Given a positive integral vector $\vec{b}$, the* **rectangular index set** *defined by $\vec{b}$ is the set $\mathcal{I}_b = \{\vec{i} \in \mathbb{Z}^n \mid 0 \leq \vec{i} < \vec{b}\}$ (component-wise) where n is the dimension of $\vec{b}$.*

Definition 3 *Given a rectangular index set $\mathcal{I}_b$, a* **slice** *$\mathcal{I}_b(i, k_i)$ of $\mathcal{I}_b$ is defined as the set of all elements of $\mathcal{I}$ whose i-th component is equal to k_i (an integer between 0 and $b_i - 1$).*

Definition 4 *Given an hyper-rectangle (or any more general set) $\mathcal{I}_b$, a modular mapping M_m is a* **one-to-one** *mapping* **from** *$\mathcal{I}_b$* **onto** *$\mathcal{I}_m$ if and only if for each $\vec{j} \in \mathcal{I}_m$ there is one and only one $\vec{i} \in \mathcal{I}_b$ such that $M_m(\vec{i}) = \vec{j}$.*

Definition 5 *Given an hyper-rectangle (or any more general set) $\mathcal{I}_b$, a modular mapping M_m is a* **many-to-one** *modular mapping* **from** *$\mathcal{I}_b$* **onto** *$\mathcal{I}_m$ if and only if the number of $\vec{i} \in \mathcal{I}_b$ such that $M_m(\vec{i}) = \vec{j}$ does not depend on $\vec{j}$.*

Definition 6 *Given a rectangular index set $\mathcal{I}_b$, a modular mapping M_m has the* **load-balancing property** *for $\mathcal{I}_b$ if and only if for any slice $\mathcal{I}_b(i, k_i)$, the restriction of M_m to $\mathcal{I}_b(i, k_i)$ is a many-to-one mapping onto $\mathcal{I}_m$.*

Because a modular mapping is linear, it is easy to see that the load-balancing property can be checked only for the slices that contain 0 (the slices $\mathcal{I}_b(i, 0)$). Furthermore, if $\vec{b}[i]$ denotes the vector obtained from $\vec{b}$ by removing the i-th component and $M[i]$ denotes the matrix obtained from M by removing the i-th column, then the images of $\mathcal{I}_b(i, 0)$ under M_m are the images of $\mathcal{I}_{b[i]}$ under the modular mapping $M[i]_m$. We therefore have the following property.

Lemma 4 *Given an hyper-rectangle $\mathcal{I}_b$, a modular mapping M_m has the load-balancing property for $\mathcal{I}_b$ if and only if each mapping $M[i]_m$ is a many-to-one modular mapping from $\mathcal{I}_{b[i]}$ to $\mathcal{I}_m$.*

We also have the following straightforward result.

Lemma 5 *If M_m is a one-to-one modular mapping from $\mathcal{I}_{b'}$ onto $\mathcal{I}_m$, then M_m is a many-to-one modular mapping from any multiple $\mathcal{I}_b$ of $\mathcal{I}_{b'}$ onto $\mathcal{I}_m$.*

Lemmas 4 and 5 explain why we focus on one-to-one modular mappings first, then on many-to-one modular mappings, and finally on modular mappings with the load-balancing property. In the extended version of this paper [10], we explore the properties of such modular mappings, in order to define a provably adequate matrix M and shape $\vec{m}$ for the virtual grid of processors. Our results are linked to previous works by Lee and Fortes [14] and Darte, Dion, and Robert [9] to the case of one-to-one modular mappings. As in [9], the theory we developed is linked to a famous (in covering/packing theory) theorem due to Hajos [12]. Our results are also connected (through the use of Hajos' theorem) to scheduling techniques used in systolic-like array design (see [8] and [11]) for generating "juggling schedules". However, unlike these two works, which are "one-to-one"-like problems, many questions remain open in the many-to-one case because the extension of Hajos' theorem to a similar "many-to-one" case is true only up to dimension 3 included. Also, while it is easy to build a one-to-one mapping (just take $\vec{m} = \vec{b}$ and the identity matrix!), here we need a much more constrained matrix, such that any submatrix obtained by removing one column is many-to-one for the corresponding $\vec{b}$ and $\vec{m}$. In other words, to use the terminology [11], we need to juggle simultaneously in all dimensions!

We just give here the steps of our construction. We build a modular mapping M_m with the load-balancing property for an index set $\mathcal{I}_b$ (which is given, $\vec{b}$ is the vector whose components are the γ_i's of Section 5). The freedom we have is that we can choose the matrix M and the modulo vector $\vec{m}$, but with the constraint that the cardinality of $\mathcal{I}_m$ (the product of the components of $\vec{m}$) is also given, (equal to the number of processors p). The only property of $\vec{b}$ we exploit is that $\vec{b}$ is a valid solution (with the meaning of Section 5), which means that the product of any $(d-1)$ components of $\vec{b}$ is a multiple of p.

We choose the matrix M with the following form:

$$M = \begin{pmatrix} N & 0 \\ \vec{\lambda} & 1 \end{pmatrix}$$

where N will be computed by induction. Therefore, finally, M will be even triangular, with 1's on the diagonal. We have the following preliminary result.

Lemma 6 *Suppose that m_d divides b_d, and that the modular mapping $N_{m'}$ – in dimension $(d-1)$ – defined by N and $\vec{m}'$ has the load-balancing property for $I_{b'}$, where $\vec{b}'$ and $\vec{m}'$ are the vectors defined by the $(d-1)$ first components of $\vec{b}$ and $\vec{m}$. Then, the modular mapping M_m defined by M and $\vec{m}$ has the load-balancing property for $\mathcal{I}_b$ if it is many-to-one from the last slice $\mathcal{I}_b(0, d)$ onto $\mathcal{I}_m$.*

Proof: In order to check that the mapping defined by M and $\vec{m}$ has the load-balancing property for the rectangular index set $\mathcal{I}_b$, we have to make sure that it is many-to-one for all slices $\mathcal{I}_b(0, i)$, $1 \leq i \leq d$ (Lemma 4). To prove this lemma, we only have to prove that this is true for the slices $\mathcal{I}_b(0, i)$, $i < d$ if N has the properties stated.

Without loss of generality, let us consider the first dimension, i.e., the first slice $\mathcal{I}_b(0, 1)$. Given $\vec{j} \in \mathbb{Z}^d/\vec{m}\mathbb{Z}$, let us count the number of vectors $\vec{i} \in \mathcal{I}_b$, such that $M\vec{i} = \vec{j} \bmod \vec{m}$ and $i_1 = 0$. Now $(M\vec{i} = \vec{j} \bmod \vec{m}) \Leftrightarrow (N\vec{i}' = \vec{j}' \bmod \vec{m}'$ and $\vec{\lambda}.\vec{i}' + i_d = j_d \bmod m_d)$, where $\vec{i}'$ and $\vec{j}'$ are defined the same way as $\vec{b}'$ and $\vec{m}'$, and $\vec{\lambda}$ is the row vector formed by the first $(d-1)$ component of the last row of M. Now, because of the load-balancing property of $N_{m'}$, there are exactly n vectors $\vec{i}' \in \mathcal{I}_{b'}$ such that $i_1 = 0$ and $N\vec{i}' = \vec{j}' \bmod \vec{m}'$, where n is a positive integer that does not depend on $\vec{j}'$. It remains to count the number of values i_d, between 0 and $b_d - 1$, such that $i_d = j_d - \vec{\lambda}.\vec{i}' \bmod m_d$. Since m_d divides b_d, there are exactly b_d/m_d such values, whatever the value $x = (j_d - \vec{\lambda}.\vec{i}' \bmod m_d)$. These are the values $x + km_d$,

with $0 \leq k < b_d/m_d$. Therefore, $\vec{j}$ has $(nb_d)/m_d$ preimages in $\mathcal{I}_b$ and this number does not depend on $\vec{j}$. ■

We define the vector $\vec{m}$ according to the following formula:

$$\forall i,\ 1 \leq i \leq d,\ m_i = \frac{\gcd\left(p, \prod_{j=i}^{d} b_j\right)}{\gcd\left(p, \prod_{j=i+1}^{d} b_j\right)} \qquad (1)$$

(By convention, an "empty" product is equal to 1). The vector $\vec{m}$ defined this way has several properties that will make a recursive construction of M possible (see [10] again).

Because $m_1 = 1$, we will be able to drop, at the end of the construction, the first component of the mapping, and end up with a mapping from $\mathbb{Z}^d$ into a subgroup of $\mathbb{Z}^{d-1}$ (or of smaller dimension if some other components of m are equal to 1). Once N is built, we write:

$$M = \begin{pmatrix} N & 0 \\ \vec{\lambda} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \vec{u} & T & 0 \\ \rho & \vec{z} & 1 \end{pmatrix}$$

and we define ρ and $\vec{z}$ such that $\vec{z} = -\vec{t}T$ and $\rho = 1 - \vec{t}.\vec{u}$, where the row vector $\vec{t}$, with $(d-2)$ components, is defined by the following (decreasing) recurrence:

- $r_{d-1} = m_d$,
- for $1 \leq i \leq d-2$, $t_i = \frac{r_{i+1}}{\gcd(b_{i+1}, r_{i+1})}$ and $r_i = \gcd(t_i m_{i+1}, r_{i+1})$.

This schema corresponds to the C program of Figure 3 (where the matrix M has rows and columns from 1 to d as in the presentation of this paper). In our current implementation, we of course take the final matrix modulo the corresponding values of $\vec{m}$. We also play some tricks, variants of the previous program (alternating signs of t for example, or prepermuting the components of $\vec{b}$) to make coefficients smaller. We also use Theorem 3 in [9] (injectivity of $M_{\lambda m}$ for $\mathcal{I}_{\lambda b}$) to reduce the components of M, dividing the components of $\vec{b}$ by their gcd. But the basic kernel is the one presented in Figure 3.

```
// Precondition: d >= 2
void ModularMapping(int d) {

  for (i=1; i<=d; i++)
    for (j=1; j<=d; j++)
      if ((i==1) || (i==j)) M[i][j] = 1;
      else M[i][j] = 0;

  for (i=2; i<=d; i++) {
    r = m[i];
    for (j=i-1; j>=2; j--) {
      t = r/gcd(r, b[j]);
      for (k=1; k<=i-1; k++) {
        M[i][k] -= t*M[j][k];
      }
      r = gcd(t*m[j],r);
    }
  }
}
```

Figure 3: Program for generating a mapping with the load-balancing property.

7 Multipartitionings in dHPF

We have implemented preliminary support for *generalized* multipartitionings in the Rice dHPF compiler for High Performance Fortran.

Multipartitioning within the dHPF compiler is implemented as a generalization of `BLOCK`-style HPF partitioning [6, 7]. The partitioned dimensions of the template are distributed onto a virtual array of processors that has the correct size for the rank of the multipartitioning. Internally, the compiler analyzes communication and loop bounds reduction as if the multipartitioned template was a standard `BLOCK` partitioned template onto a larger array of processors. The main difference comes in the interpretation that the compiler gives to the `PROCESSORS` directive. For a `BLOCK` partitioned template, the number of processors onto which each dimension is partitioned determines the data sizes of the tiles. The number of processors may be different for each dimension (i.e. `processors p(2, 3); distribute t(block, block) onto p`).

In the case of multipartitionings, the number of processors cannot be specified on a per dimension basis. All multipartitioned dimensions are distributed onto the number of processors corresponding to the leftmost dimension of the `PROCESSORS` directive. The tiles are partitioned according to the rank of the multipartitioning and then assigned in a skewed-cyclic fashion to the processors (as presented in section 2). Figure 1 illustrates a 3D diagonal multipartitioning on 16 processors.

There are several important issues for correctly generating efficient code for diagonal multipartitioned distributions:

- **Tile Iteration Order**: The order in which a processor's tiles are enumerated has to satisfy any loop-carried dependences present in the orig-

inal loop from which the multipartitioned loop has been generated. If the tiles are not enumerated in the order indicated by the loop-carried dependences, then it is possible to execute the loop correctly, but in a serialized manner induced by data exchange-related synchronization.

- **Inter-loop nest Communication Aggregation**: Communication, which has effectively been vectorized out of a loop nest, should not be performed on a tile-by-tile basis, but instead should be executed once for all of a processor's tiles. This is possible because multipartitioning guarantees that the neighboring tiles for a particular processor will be the same for all of its owned tiles.

In the case of generalized multipartitionings, we might have distributions in which we have more than one tile per processor on a single hyperplane. In order to generate high-performance code, we had to address these challenges:

- **Extended Tile Iteration Order**: For a single hyperplane, a processor may need to enumerate several tiles. The enumeration order does not have any bearing on correctness because dependences are being carried across hyperplanes instead of within a single hyperplane.

- **Intra-loop nest Communication Aggregation**: Communication caused by a loop-carried dependence may require several of a processor's tiles on a single hyperplane to send or receive data. We desire that this communication event should be executed as a single unit, instead of once per tile. This is possible because generalized multipartitionings provide the same neighborhood guarantee as simpler, diagonal multipartitionings.

8 Preliminary Results

Our implementation of multipartitioning in dHPF currently supports generalized multipartitionings. By using a multipartitioned data distribution in conjunction with sophisticated data-parallel compiler optimizations, we are closing the performance gap between compiler-generated and hand-coded implementations of line-sweep computations. Earlier results and details about dHPF's compilation techniques can be found elsewhere [7, 6, 1, 2]. Here we present some preliminary results applying generalized multipartitioning in a compiler-based parallelization of the NAS SP application benchmark [3, 7], a computational fluid dynamics code.

# CPUs	hand-coded	dHPF	% diff.
1	0.80	0.87	-8.30
2		1.30	
4	2.86	2.60	10.16
6		4.14	
8		6.35	
9	7.74	6.98	10.84
12		9.72	
16	13.00	13.97	-6.87
18		15.84	
20		16.44	
25	22.15	21.32	3.87
32		27.84	
36	36.51	32.38	12.79
49	51.78	41.32	25.32
50		38.88	
64	74.95	51.43	13.44

Table 1: Comparison of hand-coded and dHPF speedups for NAS SP (class B).

The most important analysis and code generation techniques used to obtain high-performance multipartitioned applications by the dHPF compiler are:

- partial replication of computation to reduce communication frequency and volume,
- communication vectorization,
- aggressive communication placement, and
- intra-variable and inter-variable communication aggregation.

We performed these experiments on a SGI Origin 2000 with 128 250MHz R10000 CPUs, each CPU has 32KB of L1 instruction cache, 32KB of L1 data cache and an unified, two-way set associative L2 cache of 4MB.

Table 1 shows the speedups obtained for both the dHPF-generated and hand-coded versions of the NAS SP benchmark using the class 'B' problem size (102^3). The hand-coded version implements three-dimensional diagonal multipartitionings, thus its results are only available for numbers of processors which are perfect squares. The compiler-generated version uses generalized multipartitioning to execute on other numbers of processors. The table presents the speedups for the hand-coded version (where available), the dHPF version and the differences between

them. All speedups presented are relative to the sequential version of NAS SP. Overall, the performance of the compiler-generated code is similar to that of the hand-coded versions with the exception of the gap between the versions for a 49 processor execution, which is wider for reasons that are currently unknown.

The performance differences observed between the hand-coded and compiler-generated versions are due in large part to a difference how off-processor values are stored and accessed in the two versions. In the dHPF-generated code, each data tile is extended with overlap areas (ghost regions around the tile's boundary) into which off-processor data is unpacked. Overlap areas enable a loop operating on the tile to reference all data uniformly without having to distinguish between local and off-processor data. The hand-coded version uses a clever buffering scheme in which iterations of a loop that need off-processer data are peeled off the main body of the loop. Then, in the peeled loop references to off-processor data read their values directly out of a message buffer without having to unpack it. In the dHPF-generated code, the use of extra data space for overlap areas degrades data cache efficiency, which appears to account for most of the observed performance differences.

One other factor that effects the execution efficiency of the dHPF-generated code when the number of tiles per hyperplane of a multipartitioning is greater than one (e.g., when the number of processors in a 3D partitioning is not a perfect square) is that the dHPF-generated code fails to effectively exploit reuse of data tiles across multiple loop nests. Currently, for a sequence of loop nests, dHPF-generated code executes one loop nest for each of the data tiles in a hyperplane of the data and then advances to the next loop nest. For a sequence of loop nests with compatible tile enumeration order, the tile enumeration loops could be fused so that all of the compatible loop nests in the sequence are performed on one tile before advancing to the next tile. When data tiles are small enough to fit into one or more caches, this strategy this would improve cache utilization by facilitating reuse of tile data among multiple loop nests.

9 Conclusions

The paper describes an algorithm for computing multipartitioned data distributions. These distributions are important because they support fully parallel execution of line-sweep computations. For arrays of two or more dimensions, our algorithm will compute an optimal multipartitioning that minimizes cost according to an objective function that measures communication in line sweep computations. Previously, optimal multipartitionings could be computed for d dimensional data only when $p^{\frac{1}{d-1}}$ is integral. Our extensions enable optimal multipartitionings to be computed for d dimensions.

We have shown that, having a partitioning in which the number of tiles in each slice is a multiple of the number of processors — an obvious necessary condition — is also a sufficient condition for a balanced mapping of tiles to processors. We also give a constructive method for building this mapping using new techniques based on modular mappings. This method assigns the tiles defined by the partitioning algorithm to the physical processors that should compute upon them.

One currently unresolved issue is that when we compute a multipartitioning for p processors, we force all processors to participate in the computation. In some cases, it might be more efficient to simply drop back to the nearest perfect square number of processors and let others sit idle. The extra communication overhead incurred by including them might dominate benefit of computation they could perform.

We have constructed a prototype code generator that exploits generalized multipartitionings in the Rice dHPF compiler; however, these partitionings could be exploited by hand-coded implementations as well. Preliminary performance results for generalized multipartitioning code generated by dHPF show encouraging scalability for small numbers of processors.

References

[1] V. Adve, G. Jin, J. Mellor-Crummey, and Q. Yi. High Performance Fortran Compilation Techniques for Parallelizing Scientific Codes. In *Proceedings of SC98: High Performance Computing and Networking*, Orlando, FL, Nov 1998.

[2] V. Adve and J. Mellor-Crummey. Using Integer Sets for Data-Parallel Program Analysis and Optimization. In *Proceedings of the SIGPLAN '98 Conference on Programming Language Design and Implementation*, Montreal, Canada, June 1998.

[3] D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow. The NAS parallel benchmarks 2.0. Technical Report NAS-95-020, NASA Ames Research Center, Dec. 1995.

[4] J. Bruno and P. Cappello. Implementing the beam and warming method on the hypercube. In *Proceedings of 3rd Conference on Hypercube Concurrent Computers and Applications*, pages 1073–1087, Pasadena, CA, Jan. 1988.

[5] C. Caldwell. The prime pages. `http://www.utm.edu/research/primes`, 2001.

[6] D. Chavarría-Miranda and J. Mellor-Crummey. Towards compiler support for scalable parallelism. In *Proceedings of the Fifth Workshop on Languages, Compilers, and Runtime Systems for Scalable Computers*, Lecture Notes in Computer Science 1915, pages 272–284, Rochester, NY, May 2000. Springer-Verlag.

[7] D. Chavarría-Miranda, J. Mellor-Crummey, and T. Sarang. Data-parallel compiler support for multipartitioning. In *European Conference on Parallel Computing (Euro-Par)*, Manchester, United Kingdom, Aug. 2001.

[8] A. Darte. Regular partitioning for synthesizing fixed-size systolic arrays. *INTEGRATION, The VLSI Journal*, pages 293–304, 1991.

[9] A. Darte, M. Dion, and Y. Robert. A characterization of one-to-one modular mappings. *Parallel Processing Letters*, 5(1):145–157, 1996.

[10] A. Darte, J. Mellor-Crummey, R. Fowler, and D. Chavarría. On efficient parallelization of line-sweep computations. Technical Report CS-TR01-377, Dept. of Computer Science, Rice University, Apr. 2001.

[11] A. Darte, R. Schreiber, B. R. Rau, and F. Vivien. A constructive solution to the juggling problem in systolic array synthesis. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'00)*, pages 815–821, Cancun, Mexico, May 2000.

[12] G. Hajós. Über einfache und mehrfache Bedeckung des n-dimensionalen Raumes mit einen Würfelgitter. *Math. Zschrift*, 47:427–467, 1942.

[13] S. L. Johnsson, Y. Saad, and M. H. Schultz. Alternating direction methods on multiprocessors. *SIAM Journal of Scientific and Statistical Computing*, 8(5):686–700, 1987.

[14] H. J. Lee and J. A. Fortes. On the injectivity of modular mappings. In P. Cappello, R. M. Owens, J. Earl E. Swartzlander, and B. W. Wah, editors, *Application Specific Array Processors*, pages 237–247, San Francisco, California, Aug. 1994. IEEE Computer Society Press.

[15] N. Naik, V. Naik, and M. Nicoules. Parallelization of a class of implicit finite-difference schemes in computational fluid dynamics. *International Journal of High Speed Computing*, 5(1):1–50, 1993.

[16] J. Sawada. C program for computing all numerical partitions of n whose largest part is k. Information on Numerical Partitions, Combinatorial Object Server, University of Victoria, `http://www.theory.csc.uvic.ca/~cos/inf/nump/NumPartition.html`, 1997.

[17] N. J. A. Sloane. The on-line encyclopedia of integer sequences. `http://www.research.att.com/~njas/sequences`, 2001.

[18] R. F. Van der Wijngaart. Efficient implementation of a 3-dimensional ADI method on the iPSC/860. In *Proceedings of Supercomputing 1993*, pages 102–111. IEEE Computer Society Press, 1993.

The Ring of k-regular Sequences, II

Jean-Paul Allouche

CNRS
Laboratoire de Recherche en Informatique
Bâtiment 490
F-91405 Orsay Cedex France

Jeffrey Shallit [1]

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Abstract

In this paper, we continue our study of k-regular sequences begun in 1992. We prove some new results, give many new examples from the literature, and state some open problems.

Key words: k-regular sequences, finite automata

1 Introduction

A sequence $(a(n))_{n\geq 0}$ over a finite alphabet Δ is said to be *k-automatic* if there exists a finite automaton with output $M = (Q, \Sigma_k, \delta, q_0, \Delta, \tau)$ such that $a(n) = \tau(\delta(q_0, (n)_k))$ for all $n \geq 0$. Here

- Q is a finite nonempty set of states;
- $\Sigma_k = \{0, 1, \ldots, k-1\}$;

Email addresses: Jean-Paul.Allouche@lri.fr (Jean-Paul Allouche), shallit@graceland.uwaterloo.ca (Jeffrey Shallit).

URLs: http://www.lri.fr/~allouche (Jean-Paul Allouche), http://www.math.uwaterloo.ca/~shallit (Jeffrey Shallit).

[1] Supported in part by a grant from NSERC.

- $\delta : Q \times \Sigma_k \to Q$ is the transition function;
- q_0 is the initial state;
- $(n)_k$ is the canonical base-k representation of n, starting with the most significant digit; and
- $\tau : Q \to \Delta$ is the output mapping.

For example, the famous Thue-Morse sequence

$$(t(n))_{n\geq 0} = \texttt{0110100110010110}\cdots$$

counts the number of 1's (mod 2) in the base-2 representation of n, and is generated by the automaton in Figure 1.

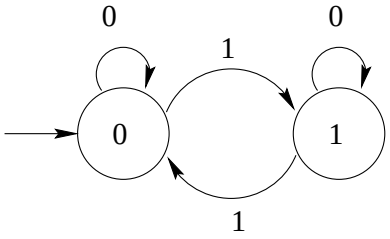


Fig. 1. Automaton generating the Thue-Morse sequence

Cobham [16] was the first to systematically study k-automatic sequences. They were then popularized and further studied by Mendès France, the authors, and others. They are extremely useful, with a well-developed theory; see, for example, [4]. However, they are somewhat restricted because of the requirement that they be defined over a finite alphabet. In [2], we gave a generalization that preserved the flavor of automatic sequences, but is defined over an infinite alphabet.

Our approach was through the k-kernel. The *k-kernel* of a sequence $(a(n))_{n\geq 0}$ is the set of subsequences

$$\{(a(k^e n + r))_{n\geq 0} \ : \ e \geq 0, \ 0 \leq r < k^e\}.$$

Then we have the following theorem of Eilenberg [19, Prop. V.3.3]:

Theorem 1 *A sequence* $\mathbf{a} = (a(n))_{n\geq 0}$ *is k-automatic if and only if the k-kernel of* $\mathbf{a}$ *is finite.*

Example 1. The Thue-Morse sequence. Consider the sequence $(t(n))_{n\geq 0}$. Then clearly

$$t(2^e n + r) \equiv t(n) + t(r) \pmod 2$$

so every sequence in the k-kernel is either $(t(n))_{n\geq 0}$ or $(t(2n+1))_{n\geq 0}$. This celebrated sequence occurs in many different areas of mathematics [3].

We now are ready to generalize k-automatic sequences. Instead of demanding that the k-kernel be finite, we instead ask that the set of sequences generated by the k-kernel be *finitely generated*. If it is, we call such a sequence *k-regular*.

More precisely, let R be a Noetherian ring. A sequence is (R,k)-regular if the R-module generated by its k-kernel is finitely generated. We usually take $R = \mathbb{Z}$, and in this case, we omit mention of R.

Example 2. Sums of digits. Consider the sequence $(s_2(n))_{n\geq 0}$, where $s_2(n)$ is the sum of the bits in the base-2 representation of n. Then

$$s_2(2^e n + r) = s_2(n) + s_2(r),$$

so every sequence in the 2-kernel is a $\mathbb{Z}$-linear combination of the sequence $(s_2(n))_{n\geq 0}$ and the constant sequence 1. Thus the 2-kernel is finitely generated, and so $(s_2(n))_{n\geq 0}$ is a 2-regular sequence.

The k-regular sequences satisfy a variety of useful properties. We recall a few results from [2]:

Theorem 2 *A sequence is k-regular and takes finitely many values if and only if it is k-automatic.*

Theorem 3 *If $(a(n))_{n\geq 0}$ and $(b(n))_{n\geq 0}$ are k-regular sequences, then so are $(a(n) + b(n))_{n\geq 0}$, $(a(n)b(n))_{n\geq 0}$, and $(ca(n))_{n\geq 0}$ for any c.*

Theorem 4 *Let $c, d \geq 0$ be integers. If $(a(n))_{n\geq 0}$ is k-regular, then so is the subsequence $(a(cn + d))_{n\geq 0}$.*

Theorem 5 *The sequence $(a(n))_{n\geq 0}$ is k-regular if and only if it is k^e-regular for any $e \geq 1$.*

2 Miscellaneous new theorems on k-regular sequences

In this section we prove a selection of new theorems about k-regular sequences.

Some sequences $\mathbf{u} = (u_n)_{n\geq 0}$ are known to satisfy recurrence relations that link subsequences such as $(u_{d^j n+i})_{n\geq 0}$ to subsequences of the same kind and to shifted sequences $(u_{n+p})_{n\geq 0}$, in a linear fashion. An example is given by the recurrence relations for the block-complexity of fixed points of uniform primitive morphisms [32]. We prove below that such sequences are d-regular. (Note that the proof of [32, Théorème 4] contains a particular case of our theorem

below.) Another application of the theorem below is the computation of the palindrome complexity of fixed points of certain uniform primitive morphisms [1].

Theorem 6 *Let $\mathbf{u} = (u_n)_{n\geq 0}$ be a sequence with values in a Noetherian ring R. Suppose there exist integers $d \geq 2$, $t, r, n_0 \geq 0$ such that each sequence $(u_{d^{t+1}n+e})_{n\geq n_0}$ for $0 \leq e < d^{t+1}$ is a linear combination of the sequences $(u_{d^j n+i})_{n\geq n_0}$ with $0 \leq j \leq t$, $0 \leq i < d^j$ and the sequences $(u_{n+p})_{n\geq n_0}$ with $0 \leq p \leq r$. Then the sequence u is d-regular.*

Proof. First for $\ell \geq 0$ define the sequences $(\lambda_n^\ell)_{n\geq 0}$ by $\lambda_n^\ell = 1$ if $n = \ell$, and $\lambda_n^\ell = 0$ otherwise. It is clear that, for $0 \leq e < d^{t+1}$, the sequences $(u_{d^{t+1}n+e})_{n\geq 0}$ are linear combinations of the sequences $(u_{d^j n+i})_{n\geq 0}$ with $0 \leq j \leq t$, $0 \leq i < d^j$, the sequences $(u_{n+p})_{n\geq 0}$ with $0 \leq p \leq r$, and the sequences $(\lambda_n^\ell)_{n\geq 0}$ for $0 \leq \ell < n_0$.

Define the sequences $(v_n^{j,i,q})_{n\geq 0}$ by, for each $n \geq 0$, $v_n^{j,i,q} := u_{d^j(n+q)+i}$, and let $\mathcal{F}$ be the R-module generated by the finite set of sequences

$$\{(v_n^{j,i,q})_{n\geq 0},\ 0 \leq j \leq t,\ 0 \leq i < d^j,\ 0 \leq q \leq Q\} \cup \{(\lambda_n^\ell)_{n\geq 0},\ 0 \leq n < n_0\},$$

where Q is fixed such that $Q \geq \frac{d(1+r)}{d-1}$.

Since the R-module $\mathcal{F}$ is finitely generated and contains the sequence $\mathbf{u}$, for proving the d-regularity of $\mathbf{u}$ it suffices to prove that $\mathcal{F}$ is stable by the maps $(w_n)_{n\geq 0} \to (w_{dn+k})_{n\geq 0}$, where $0 \leq k < d$. It even suffices to take $(w_n)_{n\geq 0}$ to be one of the generators of $\mathcal{F}$.

Let us begin with the sequences $(v_{dn+k}^{j,i,q})_{n\geq 0}$, where $0 \leq j \leq t$, $0 \leq i < d^j$, $0 \leq q \leq Q$, and $0 \leq k < d$. Let $d^j(k+q)+i = d^{j+1}x + y$, where $0 \leq y < d^{j+1}$.

Note that $d^{j+1}x \leq d^j(k+q)+i < d^j(d-1+Q)+d^j = d^{j+1}+d^jQ$. Hence $x \leq 1 + Q/d$.

We have $v_{dn+k}^{j,i,q} = u_{d^j(dn+k+q)+i} = u_{d^{j+1}(n+x)+y}$. We distinguish two cases:

- Case 1: $j < t$. Then $u_{d^{j+1}(n+x)+y} = v_n^{j+1,y,x}$, where $j+1 \leq t$, $0 \leq y < d^{j+1}$, and $0 \leq x \leq 1 + Q/d \leq Q$ since $Q \geq d(1+r)/(d-1) \geq d/(d-1)$. Hence $(v_{dn+k}^{j,i,q})_{n\geq 0}$ is one of the generators of $\mathcal{F}$.
- Case 2: $j = t$. Then $u_{d^{j+1}(n+x)+y} = u_{d^{t+1}(n+x)+y}$. Hence, from the remark at the beginning of this proof (replacing n by $n+x$), the sequence $(v_{dn+k}^{j,i,q})_{n\geq 0} = (v_{dn+k}^{t,i,q})_{n\geq 0}$ is a linear combination of the sequences $(u_{d^\ell(n+x)+i})_{n\geq 0}$ with $0 \leq \ell \leq t$, $0 \leq i < d^\ell$, the sequences $(u_{n+x+p})_{n\geq 0}$ with $p \leq r$, and the sequences $(\lambda_{n+x}^\ell)_{n\geq 0}$ for $0 \leq \ell < n_0$. We saw in the first case that $x \leq 1 + Q/d \leq Q$.

Hence the sequences $(u_{d^\ell(n+x)+i})_{n\geq 0}$ with $0 \leq \ell \leq t$, $0 \leq i < d^\ell$ are in the set of generators of $\mathcal{F}$.

On the other hand the sequences $(u_{n+x+p})_{n\geq 0}$ are also in this set of generators, since $x+p \leq x+r \leq 1+Q/d+r \leq Q$ from the choice of Q. Finally the sequences $(\lambda^\ell_{n+x})_{n\geq 0}$ satisfy:

- either $\ell \geq x$ and $(\lambda^\ell_{n+x})_{n\geq 0} = (\lambda^{\ell-x}_n)_{n\geq 0}$. Then $\ell - x \leq \ell < n_0$ and the sequence $(\lambda^\ell_{n+x})_{n\geq 0}$ belongs to the set of generators of $\mathcal{F}$;
- or $\ell < x$ and $(\lambda^\ell_{n+x})_{n\geq 0}$ is the constant sequence 0.

Hence all the sequences $(v^{j,i,q}_{dn+k})_{n\geq 0}$, where $0 \leq j \leq t$, $0 \leq i < d^j$, $0 \leq q \leq Q$, and $0 \leq k < d$, belong to the R-module $\mathcal{F}$.

Let us look now at the sequences $(\lambda^\ell_{dn+k})_{n\geq 0}$ for $0 \leq \ell < n_0$, and $0 \leq k < d$.

- If $\ell \neq k \bmod d$, then clearly the sequence $(\lambda^\ell_{dn+k})_{n\geq 0}$ is the constant sequence 0.
- If $\ell = k \bmod d$, say $\ell = dz + k$ with $z \in \mathbb{Z}$, then:
  - either $z \geq 0$, hence $(\lambda^\ell_{dn+k})_{n\geq 0} = (\lambda^z_n)_{n\geq 0}$, which belongs to the R-module $\mathcal{F}$ since $z \leq \ell < n_0$;
  - or $z < 0$, hence the sequence $(\lambda^\ell_{dn+k})_{n\geq 0}$ is the constant sequence 0.

Hence all the sequences $(\lambda^\ell_{dn+k})_{n\geq 0}$ for $0 \leq \ell < n_0$ and $0 \leq k < d$ belong to the R-module $\mathcal{F}$. ■

A celebrated theorem of Cobham says that if a sequence is both k- and ℓ-automatic, with k, ℓ multiplicatively independent, then it is ultimately periodic. The analogous conjecture about k-regular sequences is still open [2, p. 195]. But we do have the following result:

Theorem 7 *Let k and ℓ be integers ≥ 2. Let $\mathbf{x}$ be a sequence that is both k-regular and ℓ-regular. Then $\mathbf{x}$ is $k\ell$-regular.*

Proof. Suppose $\mathbf{x} = (x_n)_{n\geq 0}$ is both k-regular and ℓ-regular, for $k, \ell \geq 2$. Since $\mathbf{x}$ is k-regular, we know there exist sequences $(x^{(1)}_n)_{n\geq 0}$, ..., $(x^{(d)}_n)_{n\geq 0}$, each of the form $(x_{k^\alpha n+\beta})_{n\geq 0}$ for some $\alpha \geq 0$ and $\beta < k^\alpha$, such that $(x^{(1)}_n)_{n\geq 0} = (x_n)_{n\geq 0}$, and any sequence $(x_{k^\gamma n+\delta})_{n\geq 0}$ with $\gamma \geq 0$ and $\delta < k^\gamma$ is a $\mathbb{Z}$-linear combination of the sequences $(x^{(i)n}_{n\geq 0})$, for $i = 1, 2, \ldots, d$.

Since the sequence $(x_n)_{n\geq 0}$ is ℓ-regular, it follows from Theorem 2.6 of [2] that the sequences $(x^{(i)}_{n\geq 0})$ are also ℓ-regular, for each of them is of the form $(x_{k^\alpha n+\beta})_{n\geq 0}$. Hence for each $i = 1, 2, \ldots, d$ there exist sequences $(x^{(i,1)}_n)_{n\geq 0}$, $(x^{(i,2)}_n)_{n\geq 0}$, ..., $(x^{(i,e_i)}_n)_{n\geq 0}$, each of the form $(x^{(i)}_{\ell^\alpha n+\beta})_{n\geq 0}$ for some $\alpha \geq 0$ and

$\beta < \ell^\alpha$, such that $(x_n^{(i,1)})_{n\ge 0} = (x_n^{(i)})_{n\ge 0}$, and any sequence $(x^{(i)}_{\ell^\gamma n+\delta})_{n\ge 0}$ with $\gamma \ge 0$ and $\delta < \ell^\gamma$ is a linear combination of the sequences $(x_n^{(i,j)})_{n\ge 0}$, for $j = 1, 2, \ldots, e_i$.

Now let $\alpha \ge 0$ and $\beta < (k\ell)^\alpha$, and consider the sequence $(x_{(k\ell)^\alpha n+\beta})_{n\ge 0}$. Let $\beta = k^\alpha q + r$, with $q \ge 0$ and $0 \le r < k^\alpha$. Then $k^\alpha q \le k^\alpha q + r = \beta < (k\ell)^\alpha$. Hence $q < \ell^\alpha$.

We then have $x_{(k\ell)^\alpha n+\beta} = x_{k^\alpha(\ell^\alpha n+q)+r}$. The sequence $(x_{k^\alpha n+r})_{n\ge 0}$ is a $\mathbb{Z}$-linear combination of the sequences $(x_n^{(i)})_{n\ge 0}$, with $i = 1, 2, \ldots, d$. Hence the sequence $(x_{(k\ell)^\alpha n+\beta})_{n\ge 0}$ is the same linear combination of the sequences $(x^{(i)}_{\ell^\alpha n+q})$, and hence, since $q < \ell^\alpha$, a linear combination of the sequences $(x_n^{(i,j)})_{n\ge 0}$, with $i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, e_j$. ■

Theorem 8 *Let α, β be real numbers, and let k be an integer ≥ 2. The sequence $(\lfloor n\alpha + \beta\rfloor)_{n\ge 0}$ is k-regular if and only if α is rational.*

Proof. Suppose $\mathbf{a} := (\lfloor n\alpha + \beta\rfloor)_{n\ge 0}$ is k-regular. Then by Theorem 3 the sequence $\mathbf{b} := (\lfloor (n+1)\alpha + \beta\rfloor - \lfloor n\alpha + \beta\rfloor - \lfloor \alpha\rfloor)_{n\ge 0}$ is also k-regular. But $\mathbf{b}$ takes its values in $\{0, 1\}$, and hence by Theorem 2 is k-automatic. But then, by a classic theorem about automatic sequences, the limiting frequency of the symbol 1, if it exists, must be rational [16]. But it is easy to see that 1 occurs with frequency α mod 1. Hence α mod 1 is rational, and so α is rational.

On the other hand, if α is rational, then it is easy to see that $\mathbf{c} := (\lfloor (n+1)\alpha + \beta\rfloor - \lfloor n\alpha + \beta\rfloor)_{n\ge 0}$ is periodic and hence k-regular. Then by [2, Theorem 3.1], the running sum sequence $\mathbf{d} = (\sum_{0\le i<n} c_i)_{n\ge 0}$ of $\mathbf{c} = (c_n)_{n\ge 0}$ is also k-regular. But $\mathbf{d} = \mathbf{a}$. ■

In our previous paper [2] we remarked that if p is a prime number, the sequence $(\nu_p(n!))_{n\ge 0}$ is p-regular. Here $\nu_q(m)$ denotes the exponent of the highest power of q dividing m. Note that

$$\nu_p(n!) = \left\lfloor \frac{n}{p} \right\rfloor + \left\lfloor \frac{n}{p^2} \right\rfloor + \left\lfloor \frac{n}{p^3} \right\rfloor + \cdots .$$

In our last theorem of this section, we generalize this result as follows:

Theorem 9 *Let k be an integer ≥ 2, and let $(c_n)_{n\ge 0}$ be a k-regular sequence with $c(0) = 0$. Then the sequence $(b_n)_{n\ge 0}$ defined by*

$$b_n = c_n + c_{\lfloor n/k\rfloor} + c_{\lfloor n/k^2\rfloor} + \cdots$$

is also k-regular.

Proof. We have $b_n = c_n + b_{\lfloor n/k \rfloor}$. Since $(c_n)_{n\geq 0}$ is k-regular, the $\mathbb{Z}$-module generated by its k-kernel is finitely generated, say by $T = \{(c_{k^{e_i}n+f_i})_{n\geq 0} : 1 \leq i \leq t\}$. Let $e = \max_{1\leq i\leq t} e_i$. Let $\mathcal{F}$ be the R-module generated by

$$T \cup \{(b_{k^d n+a})_{n\geq 0} : d \leq e \text{ and } 0 \leq a < k^d\}.$$

Clearly $(b_n)_{n\geq 0} \in \mathcal{F}$. Further $\mathcal{F}$ is stable under each of the maps $(w_n)_{n\geq 0} \to (w_{kn+a})_{n\geq 0}$, $0 \leq a < k$. ■

The result about the k-regularity of $\nu_p(n!)$ follows by setting $c_n = n$. Then $\nu_p(n!) = b_n - n$.

3 k-regular sequences and arithmetic fractals

In this section we explore a connection between k-regular sequences and the "arithmetic fractals" of Morton and Mourant [30,31].

Let G be an abelian group, written additively, and let $\mathbf{a} = (a(n))_{n\geq 0}$ be a sequence of elements of G. For $n, q \geq 0$ we define a subword of $\mathbf{a}$ of length k^q as follows:

$$X_n^q = (a(nk^q), a(nk^q+1), \ldots, a(nk^q+k^q-1)).$$

For a subword $v = b_1 b_2 \cdots b_t$ over the set G, and $c \in G$, we define $v - c = d_1 d_2 \cdots d_t$, where $d_i = b_i - c$ for $1 \leq i \leq t$. Morton and Mourant defined the group $\Gamma_k(G)$ to be the set of all sequences $\mathbf{a}$ for which the sequence of blocks $(X_n^q - a(n))_{n\geq 0}$ is purely periodic, for all $q \geq 0$. In other words, a sequence $\mathbf{a}$ is in $\Gamma_k(G)$ if there exists an integer $M \geq 1$ such that for all $q \geq 0$ we have $X_m^q - a(m) = X_n^q - a(n)$ if $m \equiv n \pmod{M}$.

We now prove a simple characterization of $\Gamma_k(\mathbb{Z})$.

Theorem 10 *Let $\mathbf{a} = (a(n))_{n\geq 0}$ be a sequence of integers. Then $\mathbf{a} \in \Gamma_k(\mathbb{Z})$ if and only if the sequence $(a(n) - a(\lfloor n/k \rfloor))_{n\geq 0}$ is purely periodic.*

Proof. Suppose $\mathbf{a} \in \Gamma_k(\mathbb{Z})$. Then by taking $q = 1$ in the definition of $\Gamma_k(\mathbb{Z})$, we see there exists $M \geq 1$ such that $X_m^1 - a(m) = X_n^1 - a(m)$ if $m \equiv n \pmod{M}$. In other words, if $m \equiv n \pmod{M}$, then

$$a(km+i) - a(m) = a(kn+i) - a(n) \tag{1}$$

for all i, $0 \leq i < k$. We now show that $(a(n) - a(\lfloor n/k \rfloor))_{n\geq 0}$ is purely periodic with period Mk. Suppose $m' \equiv n' \pmod{Mk}$. Then $m' \equiv n' \pmod{k}$, so that we can write $m' = mk + i$, $n' = nk + i$ for some m, n with $0 \leq m, n < M$.

Then by Eq. (1), we have

$$a(m') - a(\lfloor m'/k \rfloor) = a(n') - a(\lfloor n'/k \rfloor),$$

which is the desired conclusion.

For the other direction, suppose that $(a(n) - a(\lfloor n/k \rfloor))_{n \geq 0}$ is purely periodic. In other words, suppose that if $m \equiv n \pmod{M}$, then

$$a(m) - a(\lfloor m/k \rfloor) = a(n) - a(\lfloor n/k \rfloor). \tag{2}$$

Now suppose $m \equiv n \pmod{M}$. Fix a $q \geq 0$ and i_0 with $0 \leq i_0 < k^q$. Define

$$\begin{aligned} m_0 &:= mk^q + i_0 \\ n_0 &:= nk^q + i_0 \end{aligned}$$

Then $m_0 \equiv n_0 \pmod{M}$, so by Eq. (2) we have

$$a(m_0) - a(\lfloor m_0/k \rfloor) = a(n_0) - a(\lfloor n_0/k \rfloor). \tag{3}$$

But $\lfloor m_0/k \rfloor = mk^{q-1} + \lfloor i_0/k \rfloor$ and similarly $\lfloor n_0/k \rfloor = nk^{q-1} + \lfloor i_0/k \rfloor$. Hence, defining

$$\begin{aligned} m_1 &:= mk^{q-1} + i_1 \\ n_1 &:= mk^{q-1} + i_1 \end{aligned}$$

where $i_1 := \lfloor i_0/k \rfloor$ we get

$$a(m_0) - a(m_1) = a(n_0) - a(n_1). \tag{4}$$

Note that $0 \leq i_1 < k^{q-1}$ and $m_1 \equiv n_1 \pmod{M}$, so we can repeat this procedure. This gives us a system of equalities

$$a(m_j) - a(m_{j+1}) = a(n_j) - a(n_{j+1}) \tag{5}$$

for $0 \leq j < q$ where

$$\begin{aligned} m_j &:= mk^{q-j} + i_j \\ n_j &:= nk^{q-j} + i_j \end{aligned}$$

for $0 \leq j \leq q$ and

$$i_{j+1} := \lfloor i_j/k \rfloor$$

for $0 \leq j < q$. Furthermore $0 \leq i_j < k^{q-j}$. Now add all of the equalities (5) together. Telescoping cancellation gives

$$a(m_0) - a(m_q) = a(n_0) - a(n_q).$$

Since $m_q = m$ and $n_q = n$, we get and hence

$$a(mk^q + i_0) - a(m) = a(nk^q + i_0) - a(n),$$

provided $m \equiv n \pmod{M}$. It follows that $X^q_m - a(m) = X^q_n - a(n)$ if $m \equiv n \pmod{M}$. Hence $\mathbf{a} \in \Gamma_k(\mathbb{Z})$. ■

Now we consider the more general case of $\Gamma_k(G)$, where G is an abelian group. We will show

Theorem 11 *Let* $\mathbf{a}$ *be a sequence in* $\Gamma_k(G)$, *i.e., if* $m \equiv n \pmod{M}$ *then* $X^q_m - a(m) = X^q_n - a(n)$ *for all* $q \geq 0$. *Then there exist a* k*-uniform morphism* $\varphi : (G \times \mathbb{Z}/M\mathbb{Z}) \to (G \times \mathbb{Z}/M\mathbb{Z})^k$ *and a coding* $\tau : G \times \mathbb{Z}/M\mathbb{Z} \to G$ *such that* $\mathbf{a} = \tau(\varphi^\omega([a(0), 0]))$.

Proof. Define

$$\varphi([g, i]) = [g_0, ki][g_1, ki + 1] \cdots [g_{k-1}, ki + k - 1]$$

where (of course) the second entries are computed mod M, and $g + X^1_i - a(i) = (g_0, g_1, \ldots, g_{k-1})$. Note that this map is well-defined, since if $i' \equiv i \pmod{M}$ then

$$\varphi([g, i']) = [g'_0, ki'] \cdots [g'_{k-1}, ki' + k - 1]$$

where $(g'_0, \ldots, g'_{k-1}) = g + X^1_{i'} - a(i')$ and by definition we have $X^1_{i'} - a(i') = X^1_i - a(i)$ if $i' \equiv i \pmod{M}$. Also define $\tau([g, i]) = g$. We now claim that

$$\varphi([a(i), i]) = [a(ki), ki] \cdots [a(ki + k - 1), ki + k - 1]. \tag{6}$$

To see this, note that

$$\varphi([a(i), i]) = [g_0, ki] \cdots [g_{k-1}, ki + k - 1]$$

where

$$\begin{aligned}(g_0, \ldots, g_{k-1}) &= a(i) + X^1_i - a(i) \\ &= X^1_i \\ &= (a(ki), \ldots, a(ki + k - 1)).\end{aligned}$$

Now a simple induction on j, together with Eq. (6), proves that

$$\varphi^j([a(0), 0]) = [a(0), 0][a(1), 1] \cdots [a(k^j - 1), k^j - 1].$$

From this the desired result follows. ■

As a corollary, we get a result of Morton and Mourant [31]:

Corollary 12 *If G is finite, the sequence* **a** *is k-automatic.*

Proof. From Theorem 11, the sequence **a** is the image, under a coding, of a fixed point of a k-uniform morphism over a finite alphabet (with $M|G|$ letters). Hence, by a well-known theorem [16], **a** is k-automatic. ■

4 New examples of k-regular sequences

In our 1992 paper [2] we gave about 30 examples of k-regular sequences from the literature. Now we give about twenty more. For more information about the sequences we discuss, the reader can consult Sloane and Plouffe's *Encyclopedia of Integer Sequences* [45], or the newer on-line version of this reference work [44]. Sequences from the former work are given in the form M`xxxx` and from the latter work in the form A`xxxxxx`.

Although we believe this catalogue is interesting in its own right, we observe that by [2, Corollary 4.6], it follows that each of these sequences can be computed in polynomial time. In some cases (e.g., Example 17), this is not at all obvious.

Example 3. Families of Separating Subsets. Consider a set S containing n elements. If a family $F = \{A_1, A_2, \ldots, A_k\}$ of subsets of S has the property that for every pair (x, y) of distinct elements of S, we can find indices $1 \leq i, j \leq k$ such that

(i) $A_i \cap A_j = \emptyset$; and
(ii) $x \in A_i$ and $y \in A_j$,

then we call F a *separating family* for S. Let $f(n)$ denote the minimum possible cardinality of F.

For example, the letters of the alphabet can be separated by only 9 subsets:

$$\{a,b,c,d,e,f,g,h,i\} \quad \{j,k,l,m,n,o,p,q,r\}$$
$$\{s,t,u,v,w,x,y,z\} \quad \{a,b,c,j,k,l,s,t,u\}$$
$$\{d,e,f,m,n,o,v,w,x\} \quad \{g,h,i,p,q,r,y,z\}$$
$$\{a,d,g,j,m,p,s,v,y\} \quad \{b,e,h,k,n,q,t,w,z\}$$
$$\{c,f,i,l,o,r,u,x\}$$

Cai Mao-Cheng showed (see [23, Chapter 18]) that

$$f(n) = \min_{0 \le i \le 2} f_i(n),$$

where

$$f_i(n) = 2i + 3\lceil \log_3 n/2^i \rceil.$$

The first few terms of this sequence are given in the following table:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$f(n)$	0	2	3	4	5	5	6	6	6	7	7	7	8	8

It is Sloane and Plouffe's sequence $M0456$ and Sloane's sequence $A007600$.

A priori, it is not clear that f is 3-regular, since the minimum of two k-regular sequences is not necessarily k-regular. However, in this case it is possible to prove the following characterization:

Theorem 13 *Let j be an integer such that $3^j < n \le 3^{j+1}$, i.e., $j = \lceil \log_3 n \rceil - 1$. Then*

$$f(n) = \begin{cases} 3j+1, & \text{if } 3^j < n \le 4 \cdot 3^{j-1}; \\ 3j+2, & \text{if } 4 \cdot 3^{j-1} < n \le 2 \cdot 3^j; \\ 3j+3, & \text{if } 2 \cdot 3^j < n \le 3^{j+1}. \end{cases}$$

Proof. First, note that $f_i(n) \ne f_j(n)$ for $i \ne j$, since if $i \ne j$, then $f_i(n)$ and $f_j(n)$ are in different residue classes, modulo 3.

Next, note that $f_0(n) < f_1(n)$ if and only if $3\lceil \log_3 n \rceil < 2 + 3\lceil \log_3 n/2 \rceil$, if and only if $\lceil \log_3 n \rceil \le \lceil \log_3 n/2 \rceil$, if and only if there exists j such that $2 \cdot 3^j < n \le 3^{j+1}$.

Similarly, $f_0(n) < f_2(n)$ if and only if there exists j such that $4 \cdot 3^j < n \le 3^{j+1}$, and $f_1(n) < f_2(n)$ if and only if there exists j such that $4 \cdot 3^j < n \le 2 \cdot 3^{j+1}$.

It follows that $f_0(n) \le \min(f_1(n), f_2(n))$ if and only if $2 \cdot 3^j < n \le 3^{j+1}$. In this case, $f_0(n) = 3j + 3$. Similar reasoning suffices for the other two cases. ∎

From Theorem 13, it easily follows that $f(n)$ is 3-regular. In fact, if we define $g(n+1) = f(n) - 3j$, where $j = \lceil \log_3 n \rceil - 1$, then it is easy to prove that $(g(n))_{n \ge 0}$ is actually a 3-automatic sequence which is the image under τ of the fixed point of φ, where $\tau(abcde) = 32312$, and φ sends $a \to abc$, $b \to dee$, $c \to ccc$, $d \to ddd$, and $e \to eee$.

Example 4. The sequences of Mallows and Propp. C. Mallows observed that there is a unique monotone sequence $(a(n))_{n\geq 0}$ of non-negative integers such that $a(a(n)) = 2n$ for $n \neq 1$. Here are the first few terms of this sequence:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$a(n)$	0	1	3	4	6	7	8	10	12	13	14	15	16	18	20	22	24

It can be shown that $a(2^i+j) = 3\cdot 2^{i-1}+j$ for $0 \leq j < 2^{i-1}$, and $a(3\cdot 2^{i-1}+j) = 2^{i+1} + 2j$ for $0 \leq j < 2^{i-1}$. This sequence is also 2-regular. It is Sloane and Plouffe's sequence $M2317$ and Sloane's sequence $A007378$. We have

$$\begin{aligned}
a(4n) &= 2a(2n);\\
a(4n+1) &= a(2n) + a(2n+1);\\
a(4n+3) &= -2a(n) + a(2n+1) + a(4n+2);\\
a(8n+2) &= 2a(2n) + a(4n+2);\\
a(8n+6) &= -4a(n) + 2a(2n+1) + 2a(4n+2).
\end{aligned}$$

Let P be a string of 0's and 1's that starts with a 1. We define the *pattern function* $e_P(n)$ to be the number of (possibly overlapping) occurrences of P in the binary representation of n. For example, $e_{101}(21) = 2$. The *pattern sequence* is the sequence $(e_P(n))_{n\geq 0}$.

We observe that the sequence $a(n+1) - a(n)$ has the following interesting representation as a sum of pattern sequences:

$$a(n+1) - a(n) = 1 + e_1(n) - e_{10}(n) + \left(\sum_{i\geq 0} e_{10^i10}(n)\right) - \left(\sum_{i\geq 0} e_{10^i1}(n)\right).$$

The proof is left to the reader.

J. Propp [37] introduced the sequence $(s(n))_{n\geq 0}$, defined to be the unique monotone sequence such that $s(s(n)) = 3n$. The table below gives the first few terms:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$s(n)$	0	2	3	6	7	8	9	12	15	18	19	20	21	22	23	24	25

It is Sloane and Plouffe's sequence $M0747$ and Sloane's sequence $A003605$. Patruno [33] showed that

$$s(n) = \begin{cases} n + 3^k, & \text{if } 3^k \le n < 2 \cdot 3^k; \\ 3(n - 3^k), & \text{if } 2 \cdot 3^k \le n < 3^{k+1}. \end{cases}$$

This sequence is 3-regular, and satisfies the recurrence

$$\begin{aligned} s(3n) &= 3s(n); \\ s(9n+1) &= 6s(n) + s(3n+1); \\ s(9n+2) &= 6s(n) + s(3n+2); \\ s(9n+4) &= 2s(3n+1) + s(3n+2); \\ s(9n+5) &= s(3n+1) + s(3n+2); \\ s(9n+7) &= -6s(n) + 3s(3n+1) + 2s(3n+2); \\ s(9n+8) &= -12s(n) + 6s(3n+1) + s(3n+2). \end{aligned}$$

More generally, one can consider solutions to the equation $a(a(n)) = dn$ for a fixed integer $d \ge 4$. Elsewhere we show that the lexicographically least monotone increasing solution to $a(a(n)) = dn$ is d-regular, and its first differences are d-automatic [5].

Example 5. The Hurwitz-Radon function. Every $n \ge 1$ can be uniquely expressed as $n = 2^{4a+b}u$, where u is odd, and $0 \le b \le 3$. In [27, p. 131], T. Y. Lam discusses the function $\rho_F(n) = 8a + 2^b$ in connection with real periodicity and Clifford modules. The table below gives the first few terms of this sequence:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\rho_F(n)$	1	2	1	4	1	2	1	8	1	2	1	4	1	2	1	9

Also see [43]. This function is 2-regular. It is Sloane and Plouffe's sequence $M0161$ and Sloane's sequence $A003484$.

Example 6. The Stanton-Kocay-Dirksen sequence. In [46], Stanton, Kocay, and Dirksen studied the sequence $f(n)$, defined to be the cardinality of a certain set. More precisely, define the product of sets C, D to be $CD = \{cd \ : c \in C, d \in D\}$. Let $A(n) = \{1, 2, \ldots, n\}$, and $B = \{2^i \ : \ i \ge 0\}$; then we define $f(n) := |A(n) \setminus (A(\lfloor n/2 \rfloor)B)|$.

They showed that $f(0) = 0$, and $f(n) = f(n-1) + (-1)^{\nu_2(n)}$ for $n \geq 1$. They also introduced the functions $F(n) = \sum_{1 \leq k \leq n} f(k)$ and $R(n) = \sum_{1 \leq k \leq n} h(k)$, where $h(k)$ is the alternating sum of the binary digits of i. More precisely, write $k = \sum_{i \geq 0} a_i(k) 2^i$, where each a_i is either 0 or 1; then $h(k) := \sum_{i \geq 0} (-1)^i a_i$.

Here is a brief table of these sequences:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$f(n)$	0	1	0	1	2	3	2	3	2	3	2	3	4	5
$F(n)$	0	1	1	2	4	7	9	12	14	17	19	22	26	31
$R(n)$	0	1	0	0	1	3	3	4	3	3	1	0	0	1
$h(n)$	0	1	-1	0	1	2	0	1	-1	0	-2	-1	0	1

All of these sequences are 2-regular. For example, it is easy to see that the following relations hold:

$$\begin{aligned} f(4n) &= f(4n+2) = 2f(n) + f(2n); \\ f(4n+1) &= f(4n+3) = 2f(n) + f(2n+1); \\ h(2n) &= -h(n); \\ h(2n+1) &= 1 - h(n). \end{aligned}$$

The sequence $R(n)$ is Sloane and Plouffe's sequence $M2274$ and Sloane's sequence $A005536$. The sequence $h(n)$ is Sloane's sequence $A065359$.

From the recursion relations for $h(n)$, the following relations follow easily:

$$\begin{aligned} R(2n+1) &= n + 1 - R(n) \quad (n \geq 0); \\ R(4n) &= R(n) + 3R(n-1) \quad (n \geq 1); \\ R(4n+2) &= 3R(n) + R(n-1) \quad (n \geq 1). \end{aligned}$$

Combining these with the identity $R(2n) = n - R(n) - R(n-1)$ for $n \geq 1$ demonstrates that $(R(n))_{n \geq 0}$ is 2-regular.

Furthermore, $R(n)$ is always non-negative, although it is not immediately obvious from the preceding identities. To prove this fact, first observe that

$$R(n) = \sum_{0 \leq k \leq n} \sum_{j \geq 0} (a_{2j}(k) - a_{2j+1}(k))$$

for $n \geq 0$. To prove $R(n)$ non-negative, it suffices to show that each of the sequences $\sum_{0 \leq k \leq n} (a_{2j}(k) - a_{2j+1}(k))$ is non-negative.

To see this, observe that

$$(a_{2j}(i))_{i\geq 0} = (0^r, 1^r)^\omega, \tag{7}$$

where $r = 2^{2j}$, the exponents in the right-hand-side of Eq. (7) denote repetitions, and comma denotes concatenation. Similarly,

$$(a_{2j+1}(i))_{i\geq 0} = (0^{2r}, 1^{2r})^\omega.$$

It follows that

$$(a_{2j}(i) - a_{2j+1}(i))_{i\geq 0} = (0^r, 1^r, (-1)^r, 0^r)^\omega.$$

Hence

$$\left(\sum_{0\leq i\leq n} (a_{2j}(i) - a_{2j+1}(i))\right)_{n\geq 0} = (0^r, 1, 2, \ldots, r, r-1, r-2, \ldots, 2, 1, 0, 0^r)^\omega.$$

The result now follows.

Example 7. Length of subgroup chains. Let G be a finite group. A *subgroup chain of length m* is a strictly descending chain of the form

$$G = G_0 > G_1 > \cdots > G_m = 1.$$

Let $\ell(G)$ be the maximum possible chain length for G. In [7], L. Babai investigated $\ell(G)$ for $G = S_n$, the symmetric group on n letters. He conjectured that $\ell(S_n) = \lceil 3n/2 \rceil - s_2(n) - 1$, where $s_2(n)$ denotes the number of 1's in the binary expansion of n. This conjecture was proved by Cameron, Solomon, and Turull [9]. The sequence $\ell(S_n)$ is 2-regular, as it is the sum of three sequences, each of which is 2-regular.

Example 8. The "odious" numbers. Consider the set

$$B = \{1, 2, 4, 7, 8, 11, 13, 14, 16, \ldots\}$$

of integers containing an odd number of 1's in their base-2 expansion. Let $b_0 = 1$ and in general, let b_i be the i'th smallest number in this set. See [38, p. 22]; [28]. For sets $S \subseteq \mathbb{N}$ define a function $f : S \times \mathbb{N} \to \mathbb{N}$ as follows: $f_S : n \to$ number of solutions to $x + y = n$, with $x, y \in S$. Then Lambek and Moser [28] showed that the sets B and $A = \mathbb{N} \setminus B$ have the property that $f_A = f_B$, and furthermore these are the only two complementary sets with this property.

The sequence $(b_i)_{i\geq 0}$ is 2-regular, as it satisfies the following recurrence relations:

$$
\begin{aligned}
b_{4n} &= -2b_n + 3b_{2n};\\
b_{4n+1} &= -2b_n + 2b_{2n} + b_{2n+1};\\
b_{4n+2} &= \frac{2}{3}b_n + \frac{5}{3}b_{2n+1};\\
b_{4n+3} &= 6b_n - 3b_{2n} + 2b_{2n+1}.
\end{aligned}
$$

Example 9. The Josephus problem. The *Josephus problem* is as follows: the numbers from 1 to n are written in a circle. Starting our count with the number 1, every 2nd number that remains is crossed off until only one is left. For example, if $n = 7$, then we cross off successively $2, 4, 6, 1, 5, 3$ and we are left with 7. The "survivor" is denoted $J(n)$. The first few values of $J(n)$ are as follows:

$$1, 1, 3, 1, 3, 5, 7, 1, 3, 5, 7, 9, 11, 13, 15, \cdots .$$

It is Sloane and Plouffe's sequence $M2216$ and Sloane's sequence $A006257$.

This problem was discussed by Graham, Knuth, and Patashnik [21, pp. 8–16] who observed that $J(2n) = 2J(n) - 1$ and $J(2n + 1) = 2J(n) + 1$ for $n \geq 1$. It follows that $J(n)$ is 2–regular.

The same problem, where 2 is replaced by k and the result is the first uncrossed-off number encountered when there are only $k-1$ numbers left, does not appear to be k-regular in general. See [21, pp. 79–81].

We are grateful to P. Dumas for pointing out this example.

Example 10. Kimberling's paraphrases. Kimberling [24] introduced the sequence $(c(n))_{n\geq 1}$

$$1, 1, 2, 1, 3, 2, 4, 1, 5, 3, 6, 2, 7, 4, 8, \ldots$$

which arose in his study of numeration systems. It follows from his paper that this sequence is $(n/2^{\nu_2(n)} + 1)/2$, which is a 2-regular sequence. It is Sloane and Plouffe's sequence $M0145$ and Sloane's sequence $A003602$. It has the pleasant property that deleting the first occurrence of each positive integer in the sequence leaves the sequence unchanged. In fact, it is easily verified that $c(2n) = c(n)$ and $c(2n - 1) = n$ for $n \geq 1$.

Example 11. The Arkin-Arney-Dewald-Ebel sequences. In [6], the sequence

$$1, 1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 8, 8, 8, 8, 8, 9, 10, \ldots$$

is studied. This sequence $(F(n))_{n\geq 1}$ satisfies the recurrence $F(2n) = 2F(n)$; $F(2n + 1) = F(n) + F(n + 1)$. It is 2-regular, and is Sloane and Plouffe's sequence $M0277$ and Sloane' sequence $A006165$.

The same paper also discusses the sequence

$$(G(n))_{n\geq 1} = 1, 1, 3, 3, 3, 3, 5, 7, 9, 9, 9, 9, 9, \ldots$$

which satisfies the recurrence $G(3n) = 3G(n)$; $G(3n+1) = G(n+1)+2G(n)$; $G(3n+2) = 2G(n+1)+G(n)$. This is 3-regular, and is Sloane and Plouffe's sequence $M2270$ and Sloane's sequence $A006166$.

Example 12. Cost of grid communications on the Connection Machine. In [49], A. Weitzman gave the following formula for $F(n)$, the optimal cost of grid communication between two processors of distance n on the Connection Machine:

$$F(j) = \begin{cases} 0, & \text{if } j = 0; \\ 1, & \text{if } j \text{ is a power of 2;} \\ 1 + \min(F(n-2^k), F(2^{k+1}-n)), & \text{if } 2^k < n < 2^{k+1}. \end{cases}$$

It is Sloane and Plouffe's sequence $M0103$ and Sloane's sequence $A007302$.

Here is a brief table of this sequence:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$F(n)$	0	1	1	2	1	2	2	2	1	2	2	3	2	3	2	2	1

This sequence has the following beautiful expansion as a sum of pattern sequences (see Example 4 for definition):

$$F(n) = e_1(n) - \sum_{i\geq 0} e_{11(01)^i 1}(n).$$

There is also a connection between $F(n)$ and representation in base-2 using the digits $0, 1, \bar{1}$, where $\bar{1} = -1$. Define the weight of such a representation to be the number of non-zero terms. For example, $10\bar{1}01$ is a representation for 13 of weight 3. Then it can be shown that $F(n)$ is the minimum weight over all such representations for n.

Example 13. Sums of squares of digits. Porges [36] has discussed properties of the sequence $b_k(n)$, defined as follows: let $n = \sum_{i\geq 0} a_i k^i$, where $0 \leq a_i < k$, and set $b_k(n) = \sum_{i\geq 0} a_i^2$. We have $b_k(kn+a) = b_k(n) + a^2$ for $0 \leq a < k$, and so it follows that this sequence is k-regular for all $k \geq 2$.

Also see [47].

Example 14. The correlation of the Thue-Morse sequence. Define the *correlation* $\gamma_f(h)$ of a sequence $(f_n)_{n\geq 0}$ of complex numbers as follows:

$$\gamma_f(h) = \lim_{N\to\infty} \frac{1}{N} \sum_{0\leq n<N} \overline{f(n)} f(n+h).$$

if this limit exists. In the case where $f_n = (-1)^{s_2(n)}$, the Thue-Morse sequence on symbols 1 and -1, Mahler [29] proved that $\gamma_f(0) = 1$, $\gamma_f(2k) = \gamma_f(k)$, and $\gamma_f(2k+1) = -\frac{1}{2}(\gamma_f(k) + \gamma_f(k+1))$. It follows that the sequence $\gamma_f(k)$ is 2-regular (with respect to $\mathbb{Q}$ and not $\mathbb{Z}$). We are grateful to Michel Mendès France for pointing out this example.

Example 15. Kuczma's sequences. Define $f(0) = 0$, $f(2n+1) = 2f(n)$ for $n \geq 0$, and $f(2n) = 2f(n)+1$ for $n \geq 1$. Then f is 2-regular, and it is easy to show [25] that for $n \geq 1$ we have $f(n) = 2^{\lfloor \log_2 n \rfloor + 1} - n - 1$. In fact, f maps n to the number whose base-2 representation is obtained by changing every 1 to 0, and vice-versa, in the binary representation of n. It is Sloane's sequence $A035327$.

Similarly, if we define define $g(n) = f(f(n))$, then we have

$$\begin{aligned}
g(2n) &= g(n);\\
g(4n+3) &= -2g(n) + 3g(2n+1);\\
g(8n+1) &= 4g(n) + g(4n+1);\\
g(16n+5) &= -4g(2n+1) + 4g(4n+1) + g(8n+5);\\
g(16n+13) &= -8g(n) + 8g(2n+1) + g(8n+5);
\end{aligned}$$

and so $(g(n))_{n\geq 0}$ is also 2-regular.

Finally, Kuczma defined $r(n)$ to be the least non-negative integer i such that $f^i(n) = 0$, where f^i denotes the i-fold composition of f with itself. In other words, $r(n)$ is the number of blocks of adjacent identical symbols in the base-2 representation of n. It is easy to see that

$$\begin{aligned}
r(4n) &= r(2n);\\
r(4n+1) &= r(2n) + 1;\\
r(4n+2) &= r(2n+1) + 1;\\
r(4n+3) &= r(2n+1);
\end{aligned}$$

and so $(r(n))_{n\geq 0}$ is also 2-regular. It is Sloane and Plouffe's sequence $M0110$ and Sloane's sequence $A005811$.

Here are the first few values of these sequences.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f(n)$	0	0	1	0	3	2	1	0	7	6	5	4	3	2	1	0	15
$g(n)$	0	0	0	0	0	1	0	0	0	1	2	3	0	1	0	0	0
$r(n)$	0	1	2	1	2	3	2	1	2	3	4	3	2	3	2	1	2

The sequence $r(n)$ has the following simple expansion as a sum of pattern sequences (see Example 4 for definition):

$$r(n) = e_1(n) + e_{10}(n) - e_{11}(n).$$

Example 16. The sparse space for Grundy's game. In [22], the sequence $(r_n)_{n\geq 0}$ is discussed:

$$0, 1, 6, 7, 10, 11, 12, 13, 18, 19, 20, 21, 24, \ldots$$

These values are $\{n \ : \ s_2(\lfloor n/2 \rfloor) \equiv 0 \pmod 2\}$. This sequence is 2-regular, and is Sloane and Plouffe's sequence $M4060$ and Sloane's sequence $A006364$. The complementary sequence $(c_n)_{n\geq 0}$:

$$2, 3, 4, 5, 8, 9, 14, 15, 16, 17, 22, 23, \ldots$$

is also 2-regular.

Example 17. Counting overlap-free words over a binary alphabet. A word w is said to be *overlap-free* if it contains no subword of the form $axaxa$, where a is a single letter and x is a (possibly empty) word. For example, `alfalfa` is not overlap-free (take $a =$ `a`, $x =$ `lf`), but `overlap` is. Cassaigne [13] has shown that the sequence counting the number of overlap-free binary words is a 2-regular sequence.

Example 18. Bottomley's sequence. Henry Bottomley proposed a sequence, Sloane's $A055562$, [44], defined as follows: define $a(0) = 2$ and $a(n)$ to be the least integer $> a(n-1)$ such that $a(n) \neq a(k) + a(k-1)$ for all k with $1 \leq k < n$.

An easy induction shows that $a(2n) = 3n + 1 + (\lfloor \log_2 n \rfloor \bmod 2)$ for $n \geq 1$ and $a(2n+1) = 3n + 3$ for $n \geq 0$. It follows that $(a(n))_{n\geq 0}$ is 2-regular.

There is an interesting connection between this sequence and a sequence of Kimberling (Sloane's sequence A022441). Let $b(n) = a(n) + a(n-1)$ for $n \geq 1$.

Then $b(n) = 3n + 2 - (\lfloor \log_2 n \rfloor \bmod 2)$ for $n \geq 1$. Furthermore, the sequences $(a(n))_{n\geq 0}$ and $(b(n))_{n\geq 1}$ are complementary, in the sense that $\{a(n) : n \geq 0\} \cup \{b(n) : n \geq 1\} = \{2, 3, 4, \ldots\}$.

Example 19. A counterexample sequence.

Many k-regular sequences have the property that their first differences (or, more generally, tth order differences for some $t \geq 0$) are k-automatic. It might be thought the class of such sequences exhausts the k-regular sequences. But this is not the case. For example, consider the sequence $\mathbf{u} = (u_n)_{n\geq 0}$ defined by

$$u_n = \begin{cases} n, & \text{if } n \text{ is a power of 2;} \\ 0, & \text{otherwise.} \end{cases}$$

This sequence is k-regular since it satisfies the recurrence

$$\begin{aligned} a_{2n} &= 2a_n \\ a_{4n+1} &= a_{2n+1} \\ a_{4n+3} &= 0. \end{aligned}$$

Hence all its tth order differences $\Delta^t \mathbf{u}$ are also k-regular. But $\Delta^t \mathbf{u}$ is unbounded for all t and hence never k-automatic.

Example 20. Chang and Tsai's recurrence.

Chang and Tsai [14] proved an interesting theorem on recurrences, which we reformulate as follows: Let $a \geq b \geq 0$ be integers. Then the solution to the recurrence

$$S_n = \min_{1\leq k\leq n/2} (aS_{n-k} + bS_k)$$

for $n \geq 2$ is

$$S_n = S_1 + (a + b - 1)S_1 \sum_{1\leq i\leq n-1} a^{e_0(i)} b^{e_1(i)-1}.$$

Here $e_P(n)$ is the pattern function introduced above in Example 4. Then $(S_n)_{n\geq 1}$ is 2-regular for all integers $a \geq b \geq 0$.

Example 21. A recurrence from automata theory. In a recent paper [20] the second author and co-authors were led to study the recurrence given by $V(1) = 1$ and

$$V(n) = s\Big(V(\lfloor n/2 \rfloor) + V(\lceil n/2 \rceil)\Big)$$

for $n \geq 2$.

They proved that if $n = 2^a + b$ with $0 \le b < 2^a$, then $V(n) = 2bs^{a+1} + (2^a - b)s^a$. It follows that $(V(n))_{n\ge 0}$ is 2-regular. In fact, if $V(0) = 0$, this sequence satisfies the following relations for $n \ge 0$:

$$\begin{aligned}
V(2n) &= 2sV(n) \\
V(4n+3) &= (4s^3 - 2s^2)V(n) + (2s^3 + 3s)V(2n+1) - 2sV(4n+1) \\
V(8n+1) &= (4s^3 - 2s^2)V(n) - sV(2n+1) + (s+1)V(4n+1) \\
V(8n+5) &= (4s^4 - 2s^3)V(n) + (2s^3 + 5s^2)V(2n+1) - 2s^2V(4n+1).
\end{aligned}$$

In the case $s = 2$, this is Sloane's sequence $A073121$.

Example 22. Carlitz's sequences related to Stirling numbers.

Let $\left\{ {n \atop r} \right\}$ denote the Stirling numbers of the second kind, i.e.,

$$\left\{ {n \atop r} \right\} = \frac{1}{r!} \sum_{0\le j\le r} (-1)^{r-j} \binom{r}{j} j^n.$$

Carlitz [10,11] defined

$$\theta_0(n) := \text{Card}\{r \ : \ \left\{ {n \atop 2r} \right\} \text{ is odd and } 0 \le 2r \le n\}$$

and later [12] $\omega_0(n) := \theta_0(n+2)$. Here is a brief table of this function:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\omega_0(n)$	1	1	2	1	3	2	3	1	4	3	5	2	5	3	4	1	5

(Carlitz's table [11] contains an error for $n = 14$.)

Carlitz showed that [10]

$$\sum_{n\ge 0} \omega_0(n) X^n = \prod_{n\ge 0} (1 + X^{2^n} + X^{2^{n+1}}).$$

It now follows that $(\omega_0(n))_{n\ge 0}$ is 2-regular. We have the relations

$$\begin{aligned}
\omega_0(2n+1) &= \omega_0(n) \\
\omega_0(4n) &= -\omega_0(n) + 2\omega_0(2n) \\
\omega_0(4n+2) &= \omega_0(n) + \omega_0(2n).
\end{aligned}$$

In fact, $\omega_0(n)$ is just the famous Stern-Brocot sequence shifted by 1; it is Sloane and Plouffe's sequence $M0141$ and Sloane's sequence $A002487$.

More generally, for a prime p define $\theta_j(n)$ to be the number of Stirling numbers $\left\{ {n \atop k} \right\}$, $0 \le k \le n$, that are relatively prime to p and such that $k \equiv j \pmod{p}$. Define $\omega_j(n) = \theta_j(n+p)$. Carlitz [12] showed that there exists a polynomial f of degree $p(p-1)$ such that if $W_0(X) := \sum_{n \ge 0} \omega_0(n) X^n$, then

$$W_0(X) = \prod_{n \ge 0} f(X^{p^n}).$$

It now follows from a result of Dumas [18, Thm. 24, p. 131] that $(\omega_0(n))_{n \ge 0}$ is a p-regular sequence.

Example 23. The "infinity series" of composer Per Nørgård.

Danish composer Per Nørgård (1932–) used a particular mathematical sequence $(c_n)_{n \ge 0}$, called by some commentators the "infinity series", in many of his music compositions. Here $(c_n)_{n \ge 0}$ is defined by $c_0 = 0$, and for $n \ge 0$ we have $c_{2n} = -c_n$, and $c_{2n+1} = c_n + 1$. The first few values of this sequence are given in the following table.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c_n	0	1	-1	2	1	0	-2	3	-1	2	0	1	2	-1	-3	4	1
m_n	G	A♭	F♯	A	A♭	G	F	B♭	F♯	A	G	A♭	A	F♯	E	B	A♭

This is Sloane's sequence $A004718$.

For example, the first 1024 notes of the second movement of his symphony *Voyage into the Golden Screen* (1968) are defined as follows: the n'th note of the composition m_n is the note offset by c_n halftones of the chromatic scale from G (sol). The sequence $(c_n)_{n \ge 0}$ is 2-regular.

For further details, see [26] and the following web pages about Per Nørgård:

`http://www.pernoergaard.dk/indexeng.html`

`http://www.pernoergaard.dk/eng/strukturer/uendelig/uintro.html`

`http://www.pernoergaard.dk/eng/strukturer/uendelig/ukonstruktion.html`

`http://www.pernoergaard.dk/eng/strukturer/uendelig/uhierarki.html`

The second movement of *Voyage into the Golden Screen* is discussed at

http://www.pernoergaard.dk/eng/udvalgte/111b.html

with music available at

http://www.pernoergaard.dk/ress/musexx/m1110356.mp3

The first 128 notes of the infinity series are available at

http://www.pernoergaard.dk/ress/musexx/mu01.mp3 .

Another 2-regular sequence appears in the so-called "rhythmic infinity system" of Nørgård [26].

Let the Fibonacci numbers $(F_n)_{n\geq 0}$ be defined as usual by $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$. Starting with the pair $(c_0, c_1) = (F_{2n}, F_{2n+1})$, perform the following operation $n-2$ times:

- If a number F_i appears in an even-indexed position, replace it with (F_{i-2}, F_{i-1})
- If a number F_i appears in an odd-indexed position, replace it with (F_{i-1}, F_{i-2})

Kullberg illustrates this procedure in the case $n = 5$, as follows:

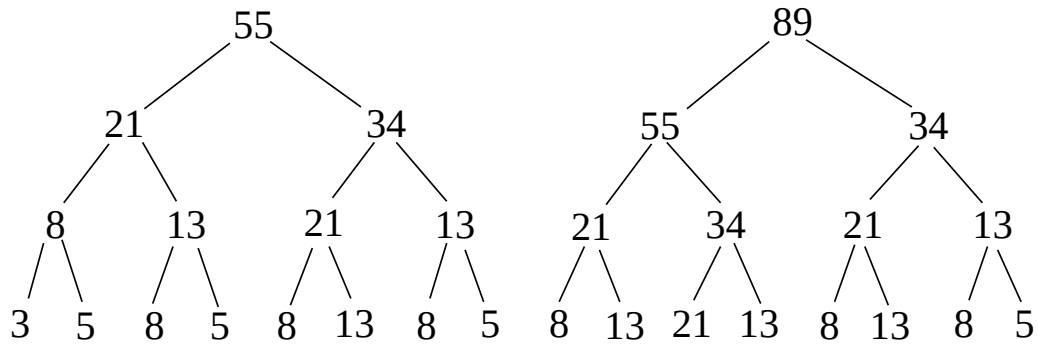


Fig. 2. Generating the rhythmic infinity series

The resulting sequence is of length 2^{n-1}. As $n \to \infty$ we get a limiting sequence $(a_i)_{i\geq 0}$:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	$\cdots$
a_i	3	5	8	5	8	13	8	5	8	13	21	13	8	13	8	5	8	13	$\cdots$

It can be shown that this sequence is 2-regular; see [42] for the details. It is Sloane's sequence *A*073334.

5 k-regular two-dimensional arrays

It is also possible to define the notion of k-regular two-dimensional array. To do so, we need a generalize the notion of k-kernel which was provided by Salon [41]. The k-kernel of a two-dimensional array $\mathbf{a} = (a(m,n))_{m,n\geq 0}$ is defined to be the set

$$\{(a(k^e m + a, k^e n + b))_{m,n\geq 0} \ : \ e \geq 0, \ 0 \leq a, b < k^e\}.$$

Again, we say $\mathbf{a}$ is k-regular if the module generated by the k-kernel is finitely generated.

One way to generate k-regular two-dimensional arrays is suggested by the following theorem:

Theorem 14 *Suppose* $\mathbf{a} = (a_m)_{m\geq 0}$ *and* $\mathbf{b} = (b_n)_{n\geq 0}$ *are* k*-regular sequences. Then* $(a_m + b_n)_{m,n\geq 0}$ *and* $(a_m b_n)_{m,n\geq 0}$ *are* k*-regular two-dimensional arrays.*

Proof. By [2, Thm. 2.2] we have that the module generated by the k-kernel of $\mathbf{a}$ (resp. $\mathbf{b}$) is generated by $\{(a_{k^i m+r})_{m\geq 0} \ : \ (i,r) \in S\}$ (resp. $\{(b_{k^j n+s})_{n\geq 0} \ : \ (j,s) \in T\}$) for some finite set S (resp. T). Then the k-kernel of $(a_m + b_n)_{m,n\geq 0}$ is a subset of

$$\{(a_{k^i m+r} + b_{k^j n+s})_{m,n\geq 0} \ : \ (i,r) \in S, \ (j,s) \in T\}.$$

Similarly, the k-kernel of $(a_m b_n)_{m,n\geq 0}$ is a subset of

$$\{(a_{k^i m+r} b_{k^j n+s})_{m,n\geq 0} \ : \ (i,r) \in S, \ (j,s) \in T\}.$$

■

We now give two examples of k-regular two-dimensional arrays.

Example 24. Let r, s be non-negative integers with base-2 representation given by $\sum_{0\leq i<t} c_i 2^i$ and $\sum_{0\leq i<t} d_i 2^i$, respectively. Define the *Nim-sum* of two integers, $r \oplus s$, to be the integer given by $\sum_{0\leq i<t}((c_i + d_i) \bmod 2)2^i$ ([50, p. 19], [17, Chapter 6], [8]). Consider the two-dimensional array $\mathbf{N} = (m \oplus n)_{m,n\geq 0}$. Here are the first few rows and columns of this array:

$\oplus$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	0	3	2	5	4	7	6	9	8
2	2	3	0	1	6	7	4	5	10	11
3	3	2	1	9	7	6	5	4	11	10
4	4	5	6	7	0	1	2	3	12	13
5	5	4	7	6	1	0	3	2	13	12
6	6	7	4	5	2	3	0	1	14	15
7	7	6	5	4	3	2	1	0	15	14
8	8	9	10	11	12	13	14	15	0	1
9	9	8	11	10	13	12	15	14	1	0

It is easily seen that $\mathbf{N}$ is 2-regular, as we find

$$\begin{aligned}\mathbf{N}[2i, 2j] &= \mathbf{N}[2i+1, 2j+1] = 2\mathbf{N}[i, j] \\ \mathbf{N}[2i+1, 2j] &= \mathbf{N}[2i, 2j+1] = 2\mathbf{N}[i, j] + 1.\end{aligned}$$

Example 25. Let F be a field with characteristic $\neq 2$. Define $D_F(n) = \{a \in F^* \ : \ a \text{ is a sum of } n \text{ squares in } F\}$. Pfister proved that there is a binary operation on $\mathbb{N} \times \mathbb{N}$ ($\mathbb{N} = \{1, 2, 3, \ldots\}$) such that $D_F(r)D_F(s) = D_F(r \circ s)$; see [43, pp. 250–252].

Here is a short table of the function $\circ$:

$\circ$	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	2	4	4	6	6	8	8	10	10
3	3	4	4	4	7	8	8	8	11	12
4	4	4	4	4	8	8	8	8	12	12
5	5	6	7	8	8	8	8	7	13	14
6	6	6	8	8	8	8	8	8	14	14
7	8	8	8	8	8	8	8	8	15	16
8	8	8	8	8	8	8	8	8	16	16
9	9	10	11	12	13	14	15	16	16	16
10	10	10	12	12	14	14	16	16	16	16

It can be shown using results of Pfister [34,35] that the operation $\circ$ satisfies the following identities:

$$\begin{aligned} 2m \circ 2n &= 2(m \circ n); \\ (2m-1) \circ 2n &= 2(m \circ n); \\ 2m \circ (2n-1) &= 2(m \circ n); \\ (2m-1) \circ (2n-1) &= 2(m \circ n) - (\binom{m+n-2}{m-1} \bmod 2). \end{aligned}$$

It follows that the infinite array $(m \circ n)_{m,n \geq 1}$ is 2-regular.

6 Recognizing a k-regular sequence

Some tips for recognizing an unknown sequence are given by Sloane and Plouffe [45]. The k-regular sequences form another easy-to-recognize class.

Given a sequence $\mathbf{s} = (s_n)_{n \geq 0}$, how can we determine if it is k-regular? Evidently no finite examination of the values of $\mathbf{s}$ will suffice. But in practice the following procedure often succeeds in deducing the k-regular relations for $\mathbf{s}$ from knowledge of the first few values. The basic idea is to construct a matrix in which the rows represent truncated versions of elements of the k-kernel, together with row reduction.

(1) Initialize the matrix to have as its single row the elements s_n, $0 \le n \le M$, where M is an arbitrarily chosen upper limit. In practice one may start with $M = 100$.
(2) Now repeat the following step: if the subsequence $(s(k^j n + c))_{n \ge 0}$ is not linearly dependent on the previous sequences, add the $(s(k^j(kn + a) + c))_{n \ge 0}$ for $0 \le a < k$ as rows, using as many terms as you have. Decrease the number of columns of the matrix appropriately.
(3) As elements further out in the k-kernel are examined, the number of columns of the matrix that are known in all entries decreases. If rows that are previously linearly independent suddenly become dependent with the elimination of terms further out in the sequence, then no relation can be accurately deduced; stop and retry after computing more terms.
(4) When no more linearly independent sequences can be found, you have found hypothetical relations for the sequence. These can then be verified through induction or other means.

There are some variations possible on this procedure. For example, we may initialize our matrix to have two or more rows, in which all but the last correspond to known sequences such as the constant sequence 1; the last is reserved for the sequence you wish to analyze. This frequently results in much smaller lists of relations.

This procedure has been implemented in APL, and has discovered many sequences known or conjectured to be k-regular.

As an example, let us consider a sequence due to N. Strauss [48]. Define

$$r(n) = \sum_{0 \le i < n} \binom{2i}{i},$$

and, as above, let $f(n) = \nu_3(r(n+1))$ be the exponent of the highest power of 3 that divides $r(n+1)$. The following table gives the first few values of f:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$f(n)$	0	1	2	0	2	3	1	2	4	0	1	2	0	3	4	2	3	5	1	2

Our k-regular sequence recognizer easily produced the following conjectured relations:

$$f(3n+2) = f(n) + 2;$$
$$f(9n) = f(9n+3) = f(3n);$$
$$f(9n+1) = f(3n) + 1;$$
$$f(9n+4) = f(9n+7) = f(3n+1) + 1;$$
$$f(9n+6) = f(3n+1).$$

With a little more work, one arrives at the conjecture

$$\nu_3(r(n)) = \nu_3(n^2 \binom{2n}{n}),$$

which we proved in 1989.

A beautiful proof of this identity using 3-adic analysis was later given by Don Zagier [51]. Zagier showed that if we set

$$F(n) = \frac{\sum_{0 \le k < n} \binom{2k}{k}}{n^2 \binom{2n}{n}},$$

then $F(n)$ extends to a 3-adic analytic function from $\mathbb{Z}_3$ to $-1 + 3\mathbb{Z}_3$, and can be evaluated at the negative integers as follows:

$$F(-n) = -\frac{(2n-1)!}{(n!)^2} \sum_{0 \le k < n} \frac{(k!)^2}{(k-1)!}$$

for $n \ge 0$.

A heuristic k-regular sequence recognizer can produce many interesting conjectures. For example, let

$$a(n) = \sum_{0 \le k \le n} \binom{n}{k} \binom{n+k}{k}.$$

Let $b(n) = \nu_3(a(n))$. Then computer experiments strongly suggest:

$$b(n) = \begin{cases} b(\lfloor n/3 \rfloor) + (\lfloor n/3 \rfloor \bmod 2), & \text{if } n \equiv 0, 2 \pmod 3; \\ b(\lfloor n/9 \rfloor) + 1, & \text{if } n \equiv 1 \pmod 3. \end{cases}$$

This recurrence has been verified for $0 \le n \le 10,000$, but no proof of the conjecture is currently known.

7 Open Problems

In this section we list some open problems about k-regular sequences.

1. Prove or disprove: $(\lfloor \frac{1}{2} + \log_2 n \rfloor)_{n \geq 1}$ is not a 2-regular sequence.

Comment. Suppose $a(n) = \lfloor \frac{1}{2} + \log_2 n \rfloor$ is 2-regular. Define $b(n) := a(n+1) - a(n)$ for $n \geq 1$. Then $(b(n))_{n \geq 0}$ would be 2-automatic, and is over the alphabet $\{0, 1\}$. The 1's in b are in positions $c_1 = 1$, $c_2 = 2$, $c_3 = 5$, $c_4 = 11$, $c_5 = 22$, $c_6 = 45$, $c_7 = 90$, etc. Then $c_{i+1} - 2c_i$ is the i'th bit in the binary expansion of $\sqrt{2}$.

2. Suppose S and T are k–regular sequences and $T(n) \neq 0$ for all n. Prove or disprove: if $S(n)/T(n)$ is always an integer, then $S(n)/T(n)$ is k–regular.

Comment. This is an analogue of van der Poorten's Hadamard quotient theorem [39,40].

3. Prove or disprove: if a sequence $(a(n))_{n \geq 0}$ is simultaneously k- and l-regular, where k and l are multiplicatively independent, then $(a(n))_{n \geq 0}$ satisfies a linear recurrence.

Comment. This is an analogue of a famous theorem of Cobham [15] for automatic sequences.

4. Prove or disprove: if q is a polynomial taking integer values and p is a prime, then $(\nu_p(q(n)))_{n \geq 0}$ is either ultimately periodic or not p-regular.

Comment. If we understood, for example, the sequence $\nu_5(n^2 + 1)$, then we would understand the 5-adic expansion of $\sqrt{-1}$.

5. Show that if $S(n)$ is k–regular and unbounded, then it takes on infinitely many composite values.

6. Suppose $(a(n))_{n \geq 0}$ is a k-regular sequence of integers such that $a(n)$ is a perfect square for all $n \geq 0$. Prove or disprove: $(a(n)^{1/2})_{n \geq 0}$ is a k-regular sequence.

8 Acknowledgments.

We are grateful to N. J. A. Sloane, who in 1995 permitted us to use an electronic copy of his book with Plouffe [45]. Thanks to L. Babai for telling us about the paper of Cameron, Solomon, and Turull [9]. Thanks to Doug Morton for help with citation searches.

We are very grateful to Per Bak and Mikael Christensen for sharing their knowledge about Per Nørgård with us.

We thank G. Skordev for having suggested Example 22, and we thank the referee for helpful comments.

References

[1] J.-P. Allouche, M. Baake, J. Cassaigne, and D. Damanik. Palindrome complexity. To appear, *Theoret. Comput. Sci.*, 2002.

[2] J.-P. Allouche and J. O. Shallit. The ring of k–regular sequences. *Theoret. Comput. Sci.* **98** (1992), 163–197.

[3] J.-P. Allouche and J. O. Shallit. The ubiquitous Prouhet-Thue-Morse sequence. In C. Ding, T. Helleseth, and H. Niederreiter, editors, *Sequences and Their Applications, Proceedings of SETA '98*, pp. 1–16. Springer-Verlag, 1999.

[4] J.-P. Allouche and J. O. Shallit. *Automatic Sequences: Theory, Applications, Generalizations*. Cambridge University Press, 2003. In press.

[5] J.-P. Allouche, N. Rampersad, and J. O. Shallit. On integer sequences whose first iterates are linear. Manuscript in preparation, November 2002.

[6] J. Arkin, D. C. Arney, L. S. Dewald, and W. E. Ebel, Jr. Families of recursive sequences. *J. Recreational Math.* **22** (1990), 85–94.

[7] L. Babai. On the length of subgroup chains in the symmetric group. *Commun. Algebra* **14** (1986), 1729–1736.

[8] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for Your Mathematical Plays*, Vol. 1. Academic Press, Toronto, 1982.

[9] P. J. Cameron, R. Solomon, and A. Turull. Chains of subgroups in symmetric groups. *J. Algebra* **127** (1989), 340–352.

[10] L. Carlitz. Single variable Bell polynomials. *Collect. Math.* **14** (1962), 13–25.

[11] L. Carlitz. A problem in partitions related to the Stirling numbers. *Bull. Amer. Math. Soc.* **70** (1964), 275–278.

[12] L. Carlitz. Some partition problems related to the Stirling numbers of the second kind. *Acta Arith.* **10** (1965), 409–422.

[13] J. Cassaigne. Counting overlap-free binary words. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, *STACS 93, Proc. 10th Symp. Theoretical Aspects of Comp. Sci.*, Vol. 665 of *Lecture Notes in Computer Science*, pp. 216–225. Springer-Verlag, 1993.

[14] K.-N. Chang and S.-C. Tsai. Exact solution of a minimal recurrence. *Inform. Process. Lett.* **75** (2000), 61–64.

[15] A. Cobham. On the base-dependence of sets of numbers recognizable by finite automata. *Math. Systems Theory* **3** (1969), 186–192.

[16] A. Cobham. Uniform tag sequences. *Math. Systems Theory* **6** (1972), 164–192.

[17] J. H. Conway. *On Numbers and Games*. Academic Press, 1976.

[18] P. Dumas. Récurrences Mahlériennes, Suites Automatiques, Études Asymptotiques. Thèse de Doctorat, Université Bordeaux I, 1993. INRIA Rapport 952, available from `ftp://ftp.inria.fr/INRIA/publication/Theses/TU-0252/` .

[19] S. Eilenberg. *Automata, Languages, and Machines*, Vol. A. Academic Press, 1974.

[20] K. B. Ellul, J. Shallit, and M. w. Wang. Regular expressions: new results and open problems. In J. Dassow, M. Hoeberechts, H. Jürgensen, and D. Wotschke, editors, *Descriptional Complexity of Formal Systems (DCFS), Pre-Proceedings*, pp. 17–34. 2002. Technical Report No. 586, University of Western Ontario.

[21] R. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 1989.

[22] R. K. Guy. Impartial games. In R. K. Guy, editor, *Combinatorial Games*, Vol. 43 of *Proc. Symp. Appl. Math.*, pp. 35–55. Amer. Math. Soc., 1991.

[23] R. Honsberger. *Mathematical Gems III*. Mathematical Association of America, 1985.

[24] C. Kimberling. Numeration systems and fractal sequences. *Acta Arith.* **73** (1995), 103–117.

[25] M. E. Kuczma. Problem 1922. *Crux Math.* **20** (1994), 74. Solution in **21** (1995), 62–64.

[26] E. Kullberg. Beyond infinity: On the infinity series — the DNA of hierarchical music. In A. Beyer, editor, *The Music of Per Nørgård: Fourteen Intepretive Essays*, pp. 71–93. Scolar Press, 1996.

[27] T. Y. Lam. *The Algebraic Theory of Quadratic Forms*. Benjamin, 1973.

[28] J. Lambek and L. Moser. On some two way classifications of integers. *Canad. Math. Bull.* **2** (1959), 85–89.

[29] K. Mahler. The spectrum of an array and its application to the study of the translation properties of a simple class of arithmetic functions. Part Two: On the translation properties of a simple class of arithmetic functions. *Journal of Mathematics and Physics* **6** (1927), 158–163.

[30] P. Morton and W. Mourant. Paper folding, digit patterns, and groups of arithmetic fractals. *Proc. Lond. Math. Soc.* **59** (1989), 253–293.

[31] P. Morton and W. J. Mourant. Digit patterns and transcendental numbers. *J. Austral. Math. Soc. Ser. A* **51** (1991), 216–236.

[32] B. Mossé. Reconnaissabilité des substitutions et complexité des suites automatiques. *Bull. Soc. Math. France* **124** (1996), 329–346.

[33] G. Patruno. Solution to problem proposal 474. *Crux Math.* **6** (1980), 198.

[34] A. Pfister. Zur Darstellung von -1 als Summe von Quadraten in einem Körper. *J. London Math. Soc.* **40** (1965), 159–165.

[35] A. Pfister. Quadratische Formen in beliebigen Körpern. *Inventiones Math.* **1** (1966), 116–132.

[36] A. Porges. A set of eight numbers. *Amer. Math. Monthly* **52** (1945), 379–382.

[37] J. Propp. Problem proposal 474. *Crux Math.* **5** (1979), 229.

[38] J. Roberts. *Lure of the Integers*. Mathematical Association of America, 1992.

[39] R. Rumely. Notes on van der Poorten's proof of the Hadamard quotient theorem: part I. In *Séminaire de Théorie des Nombres, Paris 1986–87*, Vol. 75 of *Progress in Mathematics*, pp. 349–382. Birkhäuser Boston, 1989.

[40] R. Rumely. Notes on van der Poorten's proof of the Hadamard quotient theorem: part II. In *Séminaire de Théorie des Nombres, Paris 1986–87*, Vol. 75 of *Progress in Mathematics*, pp. 383–409. Birkhäuser Boston, 1989.

[41] O. Salon. Propriétés arithmétiques des automates multidimensionnels, 1989. Thèse, Université Bordeaux I.

[42] J. Shallit. The mathematics of Per Nørgård's rhythmic infinity system. Submitted for publication, November 2002.

[43] D. B. Shapiro. Products of sums of squares. *Exposition. Math.* **2** (1984), 235–261.

[44] N. J. A. Sloane. The on-line encyclopedia of integer sequences, 2002. `http://www.research.att.com/~njas/sequences/`.

[45] N. J. A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences*. Academic Press, 1995.

[46] R. G. Stanton, W. L. Kocay, and P. H. Dirksen. Computation of a combinatorial function. In C. St. J. A. Nash-Williams and J. Sheehan, editors, *Proc. 5th British Combinatorial Conf.*, pp. 569–578. 1975. (= *Congressus Numerantium* **15**).

[47] C. Stewart. Sums of functions of digits. *Canad. J. Math.* **12** (1960), 374–389.

[48] N. Strauss and J. Shallit. Advanced problem 6625. *Amer. Math. Monthly* **97** (1990), 252.

[49] A. Weitzman. Transformation of parallel programs guided by micro-analysis. In B. Salvy, editor, *Algorithms Seminar, 1992–1993*, pp. 155–159. Institut National de Recherche en Informatique et en Automatique, France, December 1993. Rapport de Recherche, No. 2130.

[50] A. M. Yaglom and I. M. Yaglom. *Challenging Mathematical Problems with Elementary Solutions*, Vol. II. Holden-Day, 1967. Translated by J. McCawley.

[51] D. Zagier. Solution to advanced problem 6625. *Amer. Math. Monthly* **99** (1992), 66–69.

KOMBINATORICKÉ POČÍTÁNÍ 1999

MARTIN KLAZAR

Tento text je poměrně věrným zápisem přednášek, které jsem konal v letním semestru v r. 1999 na MFF UK v Praze. Na řadě příkladů ilustruje použití metody generujících funkcí (GF) v kombinatorické enumeraci. K jeho vylepšení přispěli cennými připomínkami a opravami Jakub Černý, Jan Foniok, Pavel Podbrdský, Pavel Příhoda a Patricie Rexová. Patří jim můj dík. S díky rovněž zmiňuji podporu grantu GAUK 158/99.

Martin Klazar

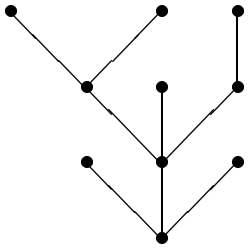
OBSAH

1. přednáška 3.3.1999

I. ÚVOD ANEB CATALANOVA ČÍSLA

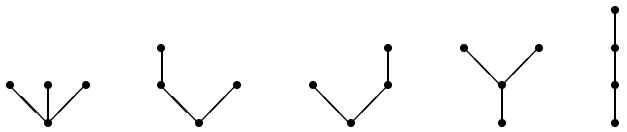
Na příkladu Catalanových čísel si předvedeme hlavní rysy enumerativní kombinatoriky.

1. KOMBINATORICKÁ STRUKTURA. Na počátku stojí enumerativní problém — třída kombinatorických struktur, které chceme spočítat. My se nejprve podíváme na *zakořeněné rovinné stromy* (stručně *zr stromy* nebo jen *stromy*). Zr strom je konečný strom s vytčeným vrcholem, kterému říkáme *kořen*, a lineárním uspořádáním na každé *množině dětí*. (Při orientaci hran směrem od kořene množina dětí sestává z konců hran vycházejících z jednoho vrcholu.) Stromy znázorňujeme obrázkem:

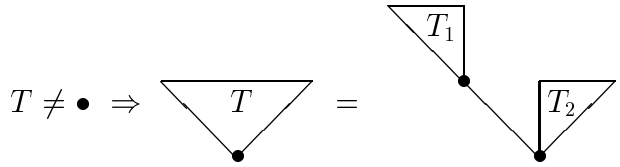


Kořen je nejníže, orientace hran souhlasí se směrem nahoru a lineární uspořádání jsou zachycena směrem zleva doprava. To například znamená, že následující dva stromy jsou různé:

Množinu všech neprázdných stromů označíme $\mathcal{T}$ a pro $n \in \mathbf{N}$, kde $\mathbf{N} = \{1, 2, \ldots\}$, definujeme $\mathcal{T}(n) = \{T \in \mathcal{T} : v(T) = n\}$, kde $v(T)$ je počet vrcholů T. Chceme spočítat čísla $c_n = |\mathcal{T}(n)|$. Například $c_1 = c_2 = 1, c_3 = 2$ a $c_4 = 5$:



2. KOMBINATORICKÝ ROZKLAD. Strom T následujícím způsobem rozložíme:



Tedy $T = (T_1, T_2)$, kde T_1 je podstrom zakořeněný v prvním dítěti kořene T a T_2 je zbytek T. Stromy T_i jsou vždy neprázdné a $v(T) = v(T_1) + v(T_2)$. Dostáváme bijekci

$$f : \mathcal{T} \setminus \{\bullet\} \to \mathcal{T} \times \mathcal{T},$$

$f(T) = (T_1, T_2)$, $v(T) = v(T_1) + v(T_2)$.

3. REKURENCE. Předchozí rozklad dává rekurenci

$$c_1 = 1 \quad \text{a} \quad c_n = \sum_{i=1}^{n-1} c_i c_{n-i} \quad \text{pro } n > 1.$$

Takže $c_5 = c_1c_4 + c_2c_3 + c_3c_2 + c_4c_1 = 2 \cdot 5 + 2 \cdot 2 = 14$.

4. GENERUJÍCÍ (VYTVOŘUJÍCÍ) FUNKCE. Posloupnost $\{c_n\}_{n\geq 1}$ zakódujeme do koeficientů mocninné řady

$$C(x) = \sum_{n=1}^{\infty} c_n x^n = \sum_{T \in \mathcal{T}} x^{v(T)}.$$

Říká se jí (obyčejná) *generující funkce* posloupnosti $\{c_n\}_{n\geq 1}$. Místo výrazu „generující funkce“ budeme pro jednotné i množné číslo psát zkratku GF. Z 2, popř. 3 plyne — toto je klíčový krok kombinatorické enumerace — relace

$$C(x) - x = C(x) \cdot C(x).$$

Dospíváme k rovnici pro GF.

4a. Rovnice pro GF. Sice, označíme-li $C(x)$ jako C,

$$C^2 - C + x = 0.$$

Dostali jsme kvadratickou rovnici. Jindy to může být algebraická rovnice vyššího stupně nebo diferenciální či funkcionální rovnice. Pro GF s více proměnnými dostaneme soustavu rovnic. Naši rovnici umíme vyřešit; často ale takové štěstí nemáme.

4b. Explicitní formule pro GF. Podle středoškolské algebry

$$C = \tfrac{1}{2}(1 - \sqrt{1-4x}).$$

Ze znamének $\pm$ jsme zvolili $-$, protože C má nulový absolutní člen. Nás však zajímá hlavně číslo $c_n = [x^n]C$; symbolem $[x^n]$ se označuje koeficient u x^n v následné mocninné řadě. Počítáme dále.

4c. Explicitní formule pro c_n. Binomická věta praví, že

$$(1+y)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} y^k,$$

kde $\alpha \in \mathbf{R}$ je pevné a $\binom{\alpha}{k} = \frac{1}{k!}\alpha(\alpha-1)\cdots(\alpha-k+1)$. Proto pro $n > 0$ máme

$$c_n = [x^n](-\tfrac{1}{2}(1-4x)^{1/2}) = -\tfrac{1}{2}(-4)^n \binom{1/2}{n}.$$

A to se rovná

$$(-1)^{n+1} \cdot \tfrac{1}{2} \cdot 4^n \cdot \frac{\frac{1}{2}\cdot\frac{-1}{2}\cdot\frac{-3}{2}\cdot\ldots\cdot\frac{-(2n-3)}{2}}{n!} = \frac{2^{n-1}\cdot 1\cdot 3\cdot\ldots\cdot(2n-3)}{n!}.$$

Tudíž (zlomek rozšíříme $(n-1)!$)

$$c_n = \frac{(2n-2)!}{(n-1)!n!} = \frac{1}{n}\binom{2n-2}{n-1}.$$

Čísla c_n jsou tzv. *Catalanova čísla*.

Ne vždy se nám v kombinatorické enumeraci podaří dopočítat se až k výsledku typu 4c, mnohdy uvízneme ve stadiu 4b nebo už ve stadiu 4a. To

však není žádné neštěstí, i tehdy se dá o hledaných počtech zjistit mnoho zajímavých věcí.

5. Nová rekurence. Pomocí GF nyní odvodíme rekurenci pro c_n, která je praktičtější než ta v 3. Ze vzorce pro $C = C(x)$ v 4b plyne, že

$$C' = \frac{1}{\sqrt{1-4x}} \quad \text{a tedy} \quad (1-4x)C' = -2C + 1.$$

To jest,

$$2C + (1-4x)C' - 1 = 0.$$

Vlevo máme mocninnou řadu, která má všechny koeficienty nulové a současně je vyjádřena pomocí C. Pro každé $n > 0$ tak dostáváme rovnici

$$2c_n + (n+1)c_{n+1} - 4nc_n = 0.$$

Takže

$$(2-4n)c_n + (n+1)c_{n+1} = 0 \quad \text{a} \quad c_{n+1} = \frac{4n-2}{n+1} \cdot c_n.$$

To se samozřejmě dostane snadno i ze vzorce pro c_n v 4c, ale právě předvedený postup funguje, i když takový vzorec nemáme k dispozici. Nepotřebujeme vlastně ani 4b a vystačíme si s 4a! Rovnici $C^2 - C + x = 0$ derivujeme podle x a vyjádříme C',

$$2C \cdot C' - C' + 1 = 0 \quad \text{a} \quad C' = \frac{1}{1-2C},$$

a výsledek zjednodušíme,

$$C' = \frac{C-1/2}{(1-2C)(C-1/2)} = \frac{C-1/2}{-2C^2+2C-1/2} = \frac{C-1/2}{2x-1/2}.$$

Při racionalizaci zlomku jsme využili, že C splňuje kvadratickou rovnici. Dospíváme opět k diferenciální rovnici $(1-4x)C' = 1-2C$.

Trocha terminologie. Posloupnost komplexních čísel $A = \{a_n\}_{n\geq 0}$ se nazývá *P-rekurzivní*, existuje-li číslo $m \in \mathbf{N}$ a celočíselné polynomy $p_0, p_1, \ldots, p_m$ (ne všechny nulové) takové, že pro každé $n \in \mathbf{N}$ platí

$$p_0(n)a_n + p_1(n)a_{n+1} + \cdots + p_m(n)a_{n+m} = 0.$$

Pokud $m = 1$, je A *hypergeometrická* posloupnost. Podíl sousedních členů pak splňuje $a_{n+1}/a_n \in \mathbf{Z}(n)$, tj. je racionální funkcí v n. (Racionální funkce

je podíl dvou polynomů.) Catalanova čísla jsou příkladem P-rekurzivní a dokonce hypergeometrické posloupnosti.

Novou rekurencí se Catalanova čísla dobře počítají:

$$c_6 = \frac{18}{6} \cdot 14 = 42, c_7 = \frac{22}{7} \cdot 42 = 132, c_8 = \frac{26}{8} \cdot 132 = 429, \ldots$$

Nenapadá vás něco při pohledu na paritu c_n?

6. KONGRUENČNÍ VLASTNOSTI. Kdy je c_n liché? Odpověď: tehdy a jen tehdy, je-li n mocnina 2. Teď výborně poslouží posmívaná rekurence 3, naopak 5 je pro tuto úlohu příliš těžkopádná. Modulo 2 máme $c_1 \equiv 1$ a, pro $n > 1$,

$$\begin{aligned} c_n = \sum_{i=1}^{n-1} c_i c_{n-i} &\equiv 0 \quad \text{pro } n = 2m+1 \text{ a} \\ &\equiv c_m^2 \quad \text{pro } n = 2m. \end{aligned}$$

Indukcí podle n plyne okamžitě, že $c_n \equiv 1$, právě když $n = 2^m$.

7. JAK RYCHLE c_n ROSTOU? Odhadneme je *Stirlingovou formulí*

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n, \text{ tj. } \frac{n!}{\sqrt{2\pi n}(n/\mathrm{e})^n} \to 1 \text{ pro } n \to \infty.$$

Formule zahrnuje dvě nejdůležitější matematické konstanty, Eulerovo číslo $\mathrm{e} = 2.71828\ldots$ a Ludolfovo číslo $\pi = 3.14159\ldots$, a dokážeme ji později. Protože

$$c_n = \frac{1}{n}\binom{2n-2}{n-1} = \frac{1}{n}\binom{2n}{n}\frac{n^2}{2n(2n-1)} \sim \frac{1}{4n}\binom{2n}{n},$$

máme asymptotiku

$$c_n \sim \frac{1}{4n} \cdot \frac{\sqrt{2 \cdot 2\pi n}(2n/\mathrm{e})^{2n}}{(\sqrt{2\pi n}(n/\mathrm{e})^n)^2} = \frac{n^{-3/2}}{4\sqrt{\pi}} \cdot 4^n.$$

Vyšli jsme z 4c, metoda GF však umí odvodit asymptotiku, i když známe jen vztah typu 4a.

2. přednáška 10.3.1999

8. JAK ROZUMĚT POJMU GF? Dvěma vzájemně se doplňujícími způsoby.

8A. FORMÁLNĚ ČILI ALGEBRAICKY. $C(x)$ nebo jinou GF chápeme jako prvek $\mathbf{C}[[x]]$, okruhu mocninných řad v jedné proměnné s komplexními koeficienty. Jeho prvky jsou nekonečné posloupnosti $A = \{a_n\} = \{a_n\}_{n\geq 0}$ komplexních čísel, s nimiž počítáme formálně podle „zřejmých“ pravidel. Pro $A = \{a_n\}$ a $B = \{b_n\}$ z $\mathbf{C}[[x]]$ mají součet $A + B$, součin AB, derivace A a integrál A n-tou složku postupně $a_n + b_n$, $\sum_{i=0}^{n} a_i b_{n-i}$, $(n+1)a_{n+1}$ a $\frac{1}{n}a_{n-1}$ (0-tá složka se zde definuje jako 0). Prakticky používáme samozřejmě raději zápis $A = \sum_{n\geq 0} a_n x^n$.

Posloupnost mocninných řad $A_1, A_2, \ldots$ *formálně konverguje* k A, jeli pro každé pevné $n \in \mathbf{N}_0 = \{0, 1, \ldots\}$ posloupnost komplexních čísel $[x^n]A_1, [x^n]A_2, \ldots$ až na konečně mnoho členů rovna $[x^n]A$. Jinými slovy, posloupnost koeficientů n-té mocniny x se stabilizuje po konečně mnoha krocích na $[x^n]A$. Nekonečné součty a součiny se v $\mathbf{C}[[x]]$ definují jako formální limity posloupností částečných součtů a součinů.

Důležitou operací je *substituce* neboli dosazení jedné mocninné řady do druhé. Pro $A = \sum_{n\geq 0} a_n x^n$ a $B = \sum_{n\geq 0} b_n x^n$ položíme

$$A(B) = \sum_{n\geq 0} a_n B^n.$$

Pro $b_0 = 0$ tento nekonečný součet formálně konverguje a $A(B)$ je definována. Pro $b_0 \neq 0$ nemá obecně formální smysl (může mít smysl analyticky). Jeli jen konečně mnoho koeficientů a_n nenulových, je $A(B)$ definována pro každé B. Takže mocninná řada $\mathrm{e}^{\mathrm{e}^x - 1}$ je dobře definována, ale výrazu $\mathrm{e}^{\mathrm{e}^x}$ z formálního hlediska nerozumíme. Shrnuto: dosadit můžeme jen mocninnou řadu s nulovým absolutním členem.

Pokud $a_0 \neq 0$, definujeme *multiplikativní inverz* A pomocí substituce jako

$$\begin{aligned} A^{-1} = \frac{1}{A} &= \frac{1}{a_0} \cdot \frac{1}{1 + ((a_1/a_0)x + (a_2/a_0)x^2 + \cdots)} \\ &= \frac{1}{a_0} \sum_{n\geq 0} (-1)^n ((a_1/a_0)x + (a_2/a_0)x^2 + \cdots)^n. \end{aligned}$$

Dělit v $\mathbf{C}[[x]]$ tedy můžeme jen řadami s nenulovým absolutním členem.

Podobně, je-li $a_0 = 0$ a $a_1 \neq 0$, existuje jednoznačný *funkcionální inverz* A značený $A^{\langle -1\rangle}$ a je to řada B splňující vztah

$$A(B) = x.$$

Odtud pro koeficienty $B = A^{\langle -1\rangle}$ dostáváme rovnice vyjadřující jednoznačně $\{b_n\}$ z $\{a_n\}$. Například, podle 4a, $(x - x^2)^{\langle -1\rangle} = C(x)$. K tomuto příkladu se vrátíme v přednášce o Lagrangeově inverzní formuli.

Popsané operace splňují spoustu identit známých z analýzy. Platí například Leibnizova formule pro derivaci součinu nebo identita $\log(\mathrm{e}^x) = \log(1 + (\mathrm{e}^x - 1)) = x$. (Formálně $\mathrm{e}^x = \sum_{n\geq 0} x^n/n!$ a $\log(1+x) = \sum_{n\geq 1}(-1)^{n-1}x^n/n$; $\mathrm{e}^{\log x}$ nemá smysl, protože funkce $\log x$ není definována jako mocninná řada.) Nebudeme se jimi podrobně zabývat. Formální hledisko je podrobně popsáno v úvodu Gouldena a Jacksona [7] a trochu v Stanleym [21]. Algebře mocninných řad se věnuje Ruizova kniha [18].

8b. Analyticky, to jest komplexně analyticky. Prvky $\mathbf{C}[[x]]$ se pak chápou ve smyslu komplexní analýzy. V pár odstavcích nelze pochopitelně ani zhruba přiblížit tuto rozsáhlou a podivuhodnou disciplínu. Chceme spíše čtenářku i čtenáře nasměrovat a motivovat k jejímu studiu. Význam komplexní analýzy pro odvozování asymptotik v enumeraci je nezastupitelný a její moc je občas skoro zázračná.

Jak plyne z 7, řada

$$C(x) = \sum_{n\geq 1} c_n x^n = \sum_{n\geq 1} \frac{1}{n}\binom{2n-2}{n-1} x^n$$

konverguje všude uvnitř kruhu $|x| \leq 1/4$ (i na hranici) a diverguje všude mimo něj, proto má poloměr konvergence $1/4$. Podle jedné ze základních vět komplexní analýzy splňuje poloměr konvergence R řady $\sum_{n\geq 0} a_n x^n$ vztah

$$\frac{1}{R} = \limsup_{n\to\infty} |a_n|^{1/n}.$$

Koeficienty tedy rostou velmi zhruba jako $(1/R)^n$. Na druhé straně je R roven absolutní hodnotě *dominantní* singularity funkce definované příslušnou řadou. Dominantní singularita je singularita nejbližší počátku. Rychlost růstu absolutních hodnot koeficientů je určena chováním funkce v okolí dominantních singularit. A toto chování se pozná již z výsledků typu 4b nebo 4a. To je v kostce princip analytických metod v kombinatorické enumeraci.

Například $\sum_{n\geq 1} c_n x^n$ definuje funkci

$$C(x) = \tfrac{1}{2}(1 - \sqrt{1-4x}),$$

která je holomorfní v kruhu $|x| < 1/4$, ale v $1/4$ má singularitu ($\sqrt{0}$). Proto $R = 1/4$ a i bez 4c je nabíledni, že ze všech exponenciel popisuje rychlost růstu čísel c_n nejlépe 4^n.

V kombinatorické enumeraci a_n ovšem nejsou obecná komplexní čísla, ale nezáporná celá čísla. Podle Pringsheimovy věty má řada s *nezápornými reálnými* koeficienty v poloměru konvergence vždy singularitu. Mezi dominantními singularitami se proto vždy musí vyskytovat kladné reálné číslo. V enumeraci nám proto jako GF nikdy nemůže vyjít třeba funkce

$$\frac{1}{x^4 - 3x + 3},$$

protože ta nemá dokonce vůbec žádnou reálnou singularitu.

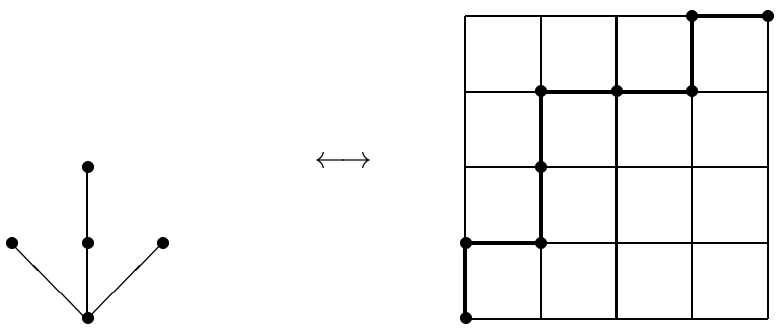
9. Nedá se vzorec pro Catalanova čísla odvodit bez GF? Existuje řada takových důkazů. Ukážeme si dva.

9a. Důkaz pomocí mřížových cest. Jako $\mathbf{Z}^2$ označíme množinu mřížových bodů $\{(a,b) : \ a,b \in \mathbf{Z}\}$, přičemž $\mathbf{Z} = \{\ldots,-1,0,1,\ldots\}$. *Cestou* budeme rozumět posloupnost $v = v_0 v_1 \ldots v_n$ bodů z $\mathbf{Z}^2$, kde $v_{i+1} - v_i$ je $(0,1)$ (krok na sever) nebo $(1,0)$ (krok na východ); cesta tedy sebe samu nikdy neprotne.

Nechť $\mathcal{B}(n)$ je množina všech cest z $(0,0)$ do (n,n) a $\mathcal{A}(n) \subset \mathcal{B}(n)$ podmnožina těch z nich, které se nikdy nedostanou pod diagonálu $y = x$.

Pozorování. Máme bijekci mezi $\mathcal{T}(n)$ a $\mathcal{A}(n-1)$.

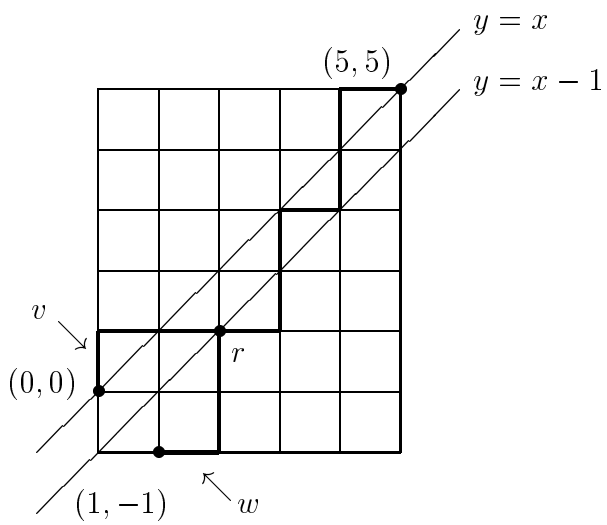
Důkaz. Obraz stromu $T \in \mathcal{T}(n)$ získáme tak, že T obcházíme dokola ve směru hodinových ručiček. Začneme od kořene a za krok vzhůru (dolů) uděláme v cestě krok na sever (východ). Inverzní zobrazení postupuje stejně opačným směrem. Příklad:



Nyní je pozorování zřejmé. □

Takže $|\mathcal{T}(n)| = |\mathcal{A}(n-1)|$. Potřebovali bychom spočítat $|\mathcal{C}(n)|$, kde $\mathcal{C}(n)$ je množina cest z $(0,0)$ do (n,n), které se pod diagonálu dostanou. Pak už $|\mathcal{A}(n)|$ spočteme snadno: $|\mathcal{A}(n)| = |\mathcal{B}(n)| - |\mathcal{C}(n)|$ a $|\mathcal{B}(n)| = \binom{2n}{n}$. (Cesty v $\mathcal{B}(n)$ odpovídají n-prvkovým podmnožinám $2n$-prvkové množiny.)

Každá cesta $v \in \mathcal{C}(n)$ má alespoň jeden společný bod s přímkou $y = x-1$. První z nich buď r. Počáteční úsek v od $(0,0)$ do r zobrazíme zrcadlením $(x,y) \to (y+1, x-1)$ podle přímky $y = x-1$, zbytek v ponecháme nezměněný. Dostaneme cestu w z $(1,-1)$ do (n,n). Příklad:



Pozorování. Zobrazení $v \to w$ je bijekce mezi $\mathcal{C}(n)$ a množinou všech cest z $(1,-1)$ do (n,n).

Důkaz. Je zřejmé, že jde o zobrazení do a že je prosté. Každá cesta w z druhé množiny má vzor v $\mathcal{C}(n)$: počátek a konec w leží na různých stranách od $y = x-1$, w tudíž protíná $y = x-1$ a vzor se dostane zrcadlením počátečního úseku. □

Druhá množina cest z předešlého pozorování má zjevně $\binom{2n}{n-1}$ prvků ($n+1$

kroků na sever a $n-1$ kroků na východ). Podle hořejší diskuse dostáváme

$$c_n = |\mathcal{T}(n)| = |\mathcal{B}(n-1)| - |\mathcal{C}(n-1)| = \binom{2n-2}{n-1} - \binom{2n-2}{n-2},$$

což je

$$\frac{(2n-2)! - (2n-2)!\frac{n-1}{n}}{(n-1)!(n-1)!} = \frac{1}{n}\binom{2n-2}{n-1}.$$

3. přednáška 17.3.1999

Trik se zrcadlením se nazývá *Andrého princip odrazu*. Pochází od D. Andrého (1840–1917) ([1]), který ho použil při řešení následujícího *hlasovacího problému* (ballot problem). Ve volbách soupeří dva kandidáti. Kandidát P dostal celkem p hlasů a kandidát Q celkem q hlasů. Platí $q > p$, takže Q vyhrál a porazil P. Jaká je pravděpodobnost, že Q neustále vedl před P v každém okamžiku hlasování? Všechny průběhy hlasování považujeme za stejně pravděpodobné. Zkuste si to spočítat jako DOM CV.

9B. Důkaz pomocí uzávorkování. Druhý důkaz publikoval Rubenstein v r. 1994 v [17]. *Uzávorkováním* rozumíme posloupnost $u = a_1, a_2, \ldots, a_{2n}$, kde $a_i = [$ nebo $a_i =]$ a máme celkem n levých a n pravých závorek. *Dobré uzávorkování* je uzávorkování, jehož každý počáteční úsek obsahuje alespoň tolik levých závorek jako pravých. Nechť $\mathcal{B}(n)$ je množina všech uzávorkování s $2n$ závorkami a $\mathcal{A}(n) \subset \mathcal{B}(n)$ je podmnožina dobrých uzávorkování.

Pozorování. Máme bijekci mezi $\mathcal{T}(n)$ a $\mathcal{A}(n-1)$.

Důkaz. Prakticky stejný jako pro mřížové cesty. Krok vzhůru (dolů) při obcházení stromu nyní odpovídá levé (pravé) závorce. □

Zatím to jsou mřížové cesty v trochu jiném hávu. Využijeme vlastnosti uzávorkování, která nám je důvěrně známa již ze základní školy.

Pozorování. Uzávorkování $u = a_1, a_2, \ldots, a_{2n}$ je dobré, právě když můžeme $\{1, 2, \ldots, 2n\}$ spárovat do n dvojic $i_1 < j_1, \ldots, i_n < j_n$ tak, že $a_{i_k} = [$, $a_{j_k} =]$ a nikdy nenastane $i_a < i_b < j_a < j_b$. Toto spárování je navíc jednoznačné. Dvojicím a_{i_k} a a_{j_k} budeme říkat *páry* (závorek).

Důkaz. Není-li u dobré, pak jeho některý počáteční úsek obsahuje více] než [. Pak ale popsané spárování zjevně nemůže existovat. Naopak, nechť u je dobré. Indukcí podle n dokážeme existenci a jednoznačnost spárování. Zřejmě $a_i = [$ a $a_{i+1} =]$ pro některé i. Pak i a $i+1$ musejí být spolu v každém spárování. Z u vyhodíme a_i a a_{i+1}, čímž dostaneme zase dobré uzávorkování. Nyní užijeme indukční předpoklad. □

Tvrzení. Pro obecné uzávorkování u máme rozklad

$$u = u_1]u_2]\ldots]u_k]v[v_k[\ldots[v_2[v_1,$$

kde $k \geq 0$ a u_i, v_i a v jsou dobrá uzávorkování. Tento rozklad je jednoznačný.

Důkaz. Jednoznačnost plyne hned z definice dobrého uzávorkování: počáteční dobré úseky u_1 a u_1' musejí být v obou případných rozkladech stejné, takže $u_1 = u_1'$, a stejně tak dále. Oba rozklady splývají.

Ukážeme existenci. Pro dobré u rozklad triviálně existuje ($k = 0$ a $u = v$). Nechť u není dobré. Pak můžeme psát $u = u_1]r$, kde u_1 je dobré (a r není vůbec uzávorkování). Ze symetrie máme, že též $u = s[v_1$, kde v_1 je dobré (a s není uzávorkování). Počáteční úsek $u_1[$ a koncový úsek $]v_1$ se nepřekrývají, jinak by totiž celé u bylo dobré. Takže máme rozklad $u = u_1]v[v_1$, kde u_1 a v_1 jsou dobrá a v nyní je uzávorkování (možná prázdné). Na v užijeme indukční předpoklad a máme hledaný rozklad. □

Uvažme nyní následující zobrazení $F : \mathcal{B}(n) \to \mathcal{A}(n)$. Dané $u \in \mathcal{B}(n)$ rozložíme jako $u = u_1]u_2]\ldots]u_k]v[v_k[\ldots[v_2[v_1$ podle posledního tvrzení a definujeme

$$F(u) = u_1[u_2[\ldots[u_k[v]v_k]\ldots]v_2]v_1,$$

tj. otočíme nespárované závorky. Je jasné, že $F(u)$ je dobré a že otočené závorky se spárují spolu: závorka $u_1[u_2$ se závorkou $v_2]v_1$ atd.

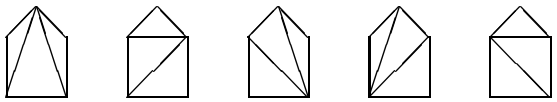
Tvrzení. V zobrazení F má každé $w \in \mathcal{A}(n)$ právě $n+1$ vzorů.

Důkaz. Nechť $w \in \mathcal{A}(n)$. *Hnízdo* ve w je takový systém do sebe vnořených párů $-[-[\ldots[-]\ldots]-]-$ ze spárování w, že sousední páry $\ldots[-[\ldots]-]\ldots$ už neodděluje žádný jiný pár w a že úplně vnější pár není obsažen v žádném páru w (úplně vnitřní pár může obsahovat jiné páry w). Snadno se vidí, že $F(u) = w$, právě když u vznikne z w otočením závorek v některém (i případně prázdném) hnízdě w. Po chvilkové meditaci je stejně tak jasné, že w obsahuje přesně $n+1$ hnízd: každý pár w je úplně vnitřním párem právě jednoho hnízda a pak máme ještě prázdné hnízdo. □

Zřejmě $|\mathcal{B}(n)| = \binom{2n}{n}$ a podle posledního tvrzení $|\mathcal{A}(n)| = \frac{1}{n+1}|\mathcal{B}(n)| = \frac{1}{n+1}\binom{2n}{n} = c_{n+1}$. Podle hořejší bijekce $|\mathcal{T}(n)| = |\mathcal{A}(n-1)| = c_n$.

10. Další kombinatorické struktury počítané Catalanovými čísly. Stanley [22] jich uvádí 66. My jich uvedeme jen pár. Struktury ilustrujeme vždy příkladem pro $c_4 = 5$.

Triangulace n-úhelníku:



Posloupnosti přirozených čísel $1 \le a_1 \le a_2 \le \cdots \le a_n$ splňující $a_i \le i$:

$$111, 112, 113, 122, 123.$$

Posloupnosti přirozených čísel $1 \le a_1 < a_2 < \cdots < a_n$ splňující $a_i \le 2i$:

$$12, 13, 14, 23, 24.$$

Permutace množiny $\{1, 2, \ldots, n\}$ neobsahující rostoucí podposloupnost délky 3:

$$132, 213, 231, 312, 321.$$

(Jen jedna permutace je zakázána.)

Permutace množiny $\{1, 2, \ldots, n\}$ neobsahující podposloupnost typu 312, tj. ty $a_1 a_2 \ldots a_n$, že neexistují tři indexy $i_1 < i_2 < i_3$, že $a_{i_1} > a_{i_3} > a_{i_2}$:

$$132, 213, 231, 123, 321.$$

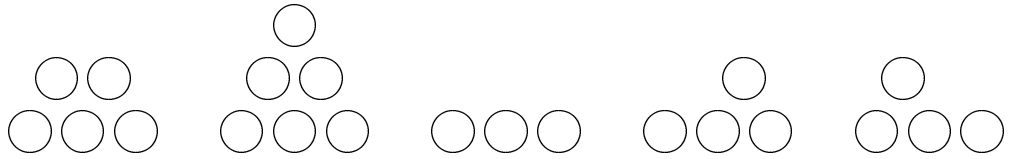
(I zde je zakázána jen jedna permutace.)

Nekřížící se rozklady $\{1, 2, \ldots, n\}$. To jest rozklady $\{1, 2, \ldots, n\} = P_1 \cup P_2 \cup \cdots P_k$, pro něž neexistují $1 \le a < b < c < d \le n$ tak, že $a, c \in P_i$ a $b, d \in P_j$ pro nějaké $i \ne j$. Pro $n = 3$ jich máme pět:

$$1|2|3,\ 12|3,\ 13|2,\ 23|1, 123.$$

Je to ovšem trochu nejapný příklad.

Hromádky mincí v rovině s n mincemi v dolní řadě:



Nakonec zmíníme zajímavý výsledek kolegy P. Valtra ([24] a [25]). Nějakých n bodů v rovině tvoří *konvexní řetězec*, pokud — uspořádáme-li je podle vzrůstajících hodnot x-ových souřadnic jako $b_1, b_2, \ldots, b_n$ — jsou směrové vektory $b_{i+1} - b_i$ uspořádány monotóně proti směru hodinových ručiček. Nechť A_n je jev, že n bodů vybraných náhodně a nezávisle v jednotkovém čtverci tvoří konvexní řetězec. Nechť B_n je jev, že tyto body tvoří vrcholy konvexního n-úhelníku. Je jasné, že když nastane A_n, nastane i B_n: $\Pr(A_n B_n) = \Pr(A_n)$.

Platí ([25]):

$$\Pr(A_n|B_n) = \frac{\Pr(A_n B_n)}{\Pr(B_n)} = \frac{\Pr(A_n)}{\Pr(B_n)} = \frac{1}{c_n}.$$

Pravděpodobnost jevu A_n podmíněná jevem B_n je rovna převrácené hodnotě Catalanova čísla!

11. Zjemnění Catalanových čísel. Nechť $n(a,b)$ je počet stromů s a vrcholy a b listy, kde list je vrchol bez dítěte. Položíme $n(1,1) = 1$. Například $n(4,2) = 3$ a $n(4,1) = n(4,3) = 1$. Zřejmě $n(a,b) > 0$, právě když $1 \le b \le a-1$ (kromě $a = 1$). Dále je jasné, že

$$\sum_{b=1}^{a-1} n(a,b) = c_a = \frac{1}{a}\binom{2a-2}{a-1}.$$

Co se dá o číslech $n(a,b)$ říci? Nasadíme na ně GF a uvidíme. Připomínáme, že pro $T \in \mathcal{T}$ počítá $v(T)$ počet vrcholů stromu T. Zavedeme ještě funkci $l(T)$, která počítá počet listů T. Definujeme GF o dvou proměnných

$$C(x,y) = \sum_{T \in \mathcal{T}} x^{v(T)} y^{l(T)} = \sum_{a,b \ge 1} n(a,b) x^a y^b.$$

4. přednáška 24.3.1999

Kombinatorický rozklad 2 z 1. přednášky nám dává vztah

$$C(x,y) - xy = C(x,y) \cdot (C(x,y) - xy + x).$$

Nezapomněli jsme na to, že jednovrcholový T_2 nepřispívá žádným listem. Takže, označíme-li na chvíli $C(x,y)$ jako C,

$$C^2 - C \cdot (1 - x + xy) + xy = 0$$

a

$$C = \tfrac{1}{2}(1 - x + xy - \sqrt{(1 - x + xy)^2 - 4xy}\,).$$

Je jasné, že substituce $y = 1$ (kterou můžeme provést, protože $C(x,y)$ je mocninná řada v x, jejímiž koeficienty jsou polynomy v y, nikoli mocninné řady v y) vymazává informaci o listech. To jest, $C(x,1) = C(x)$ a substituce $y = 1$ v posledním vzorci dává formuli 4b z 1. přednášky.

Čísla $n(a,b)$ jsou tzv. *Narayanova čísla*. Platí pro ně vzorec

$$n(a,b) = \frac{1}{a-1}\binom{a-1}{b}\binom{a-1}{b-1}.$$

Máme $n(a,b) = [x^a y^b]C(x,y)$, ale neznám žádný efektní výpočet, kterým bych vám poslední vzorec odvodil z formule pro $C(x,y)$. Pomocí Lagrangeovy inverzní formule (LIF) se dá odvodit z kvadratické rovnice pro $C(x,y)$, ale k LIF se dostaneme až později.

Symetrie binomických koeficientů $\binom{n}{m} = \binom{n}{n-m}$ je již téměř naší druhou přirozeností. Je pozoruhodné, že Narayanova čísla mají tuto vlastnost také:

$$n(a,b) = n(a, a-b).$$

Můžeme to dokázat z explicitního vzorce pro $n(a,b)$, kombinatoricky pomocí bijekce nebo pomocí GF. Ovšemže volíme poslední možnost. Substituce $x := xy, y := 1/y$ převádí $C(x,y)$ na mocninnou řadu $D = D(x,y) = C(xy, 1/y)$, přičemž pro $a > 1$ platí $[x^a y^b]D = [x^a y^{a-b}]C$. Stačí tedy ukázat, že $D - x = C - xy$. To je ale téměř očividné, protože pro D dostáváme vzorec

$$D = \tfrac{1}{2}(1 - xy + x - \sqrt{(1 - xy + x)^2 - 4x}\,)$$

a, jak se snadno přesvědčíme,

$$(1 - x + xy)^2 - 4xy = (1 - xy + x)^2 - 4x.$$

II. STIRLINGOVA FORMULE

Nyní slíbený důkaz Stirlingovy formule. Pro $n \to \infty$ platí

$$n! = (1 + O(n^{-1}))\sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n.$$

Lemma. Pro $m \to \infty$ platí

$$\log m = \int_{m-1/2}^{m+1/2} \log x \; dx + O(m^{-2}).$$

Důkaz. Jak známo, $\int \log x = x \log x - x$ a $\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots$. Takže

$$\begin{aligned}
(x \log x - x)|_{m-1/2}^{m+1/2} &= m \log \frac{m+1/2}{m-1/2} + \frac{\log(m^2 - 1/4)}{2} - 1 \\
&= m \log\left(1 + \frac{1}{m-1/2}\right) + \frac{\log(1 - 1/(4m^2))}{2} \\
&\quad + \log m - 1 \\
&= \frac{m}{m-1/2} - \frac{m}{2(m-1/2)^2} - 1 + O(m^{-2}) + \log m \\
&= \frac{-1}{2(m-1/2)^2} + O(m^{-2}) + \log m \\
&= \log m + O(m^{-2}).
\end{aligned}$$

□

S použitím lemmatu a Taylorova rozvoje pro logaritmus máme, že

$$\begin{aligned}
\log n! = \sum_{m=2}^{n} \log m &= \sum_{m=2}^{n} \left(\int_{m-1/2}^{m+1/2} \log x \cdot dx + O(m^{-2}) \right) \\
&= c_1 + O(n^{-1}) + \int_{3/2}^{n+1/2} \log x \cdot dx \\
&= (n+1/2)\log(n+1/2) - (n+1/2) + c_2 + O(n^{-1}) \\
&= n \log n - n + \tfrac{1}{2} \log n + c + O(n^{-1}).
\end{aligned}$$

Odlogaritmování nám dává vztah

$$n! = (1 + O(n^{-1}))\sqrt{dn}\left(\frac{n}{e}\right)^n,$$

kde $d > 0$ je neznámá konstanta. Spočítáme, že $d = 2\pi$. (Postupujeme podle cvičení v Tenenbaumovi [23].)

Použijeme tzv. *Wallisův integrál*

$$W_n = \int_0^{\pi/2} (\cos x)^n dx.$$

Je nabíledni, že $W_0 = \pi/2$ a $W_1 = 1$. Integrace per partes dává

$$\begin{aligned} W_n &= \sin x \cdot (\cos x)^{n-1}|_0^{\pi/2} + (n-1)\int_0^{\pi/2} \sin^2 x \cdot (\cos x)^{n-2} \cdot dx. \\ &= 0 + (n-1)(W_{n-2} - W_n). \end{aligned}$$

(Použili jsme rovnost $\sin^2 x = 1 - \cos^2 x$.) Takže, pro $n > 1$,

$$W_n = \frac{n-1}{n} \cdot W_{n-2}.$$

Pomocí této rekurence a již dokázané neúplné Stirlingovy formule dostáváme

$$W_{2n} = \frac{(2n-1)(2n-3)\cdot\ldots\cdot 1}{2n(2n-2)\cdot\ldots\cdot 2} \cdot \frac{\pi}{2} = \frac{(2n)!}{(2^n n!)^2} \cdot \frac{\pi}{2} \sim \frac{\pi}{2} \cdot \sqrt{\frac{2}{dn}}.$$

Podobně

$$W_{2n+1} = \frac{2n(2n-2)\cdot\ldots\cdot 2}{(2n+1)(2n-1)\cdot\ldots\cdot 1} \cdot 1 = \frac{(2^n n!)^2}{(2n+1)!} \sim \sqrt{\frac{d}{8n}}.$$

Z definice W_n plyne bez trápení, že

$$W_n < W_{n-1} < W_{n-2}.$$

Proto, podle rekurence,

$$1 < \frac{W_{n-1}}{W_n} < \frac{W_{n-2}}{W_n} = 1 + \frac{1}{n-1}.$$

Tedy $W_{n-1}/W_n \to 1$ pro $n \to \infty$. Nutně

$$\frac{(\pi/2)\cdot\sqrt{2/dn}}{\sqrt{d/8n}} \to 1.$$

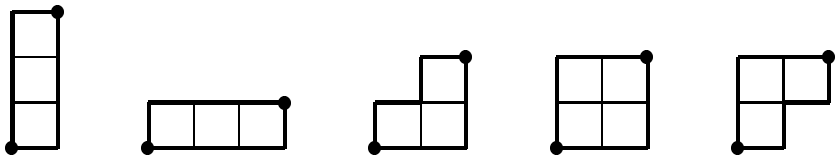
To je ovšem možné, jen když $d = 2\pi$.

III. DVAKRÁT O ZÁHADNÉ MOCNINĚ 4^n

Obrazce. Postupujeme podle článku Woana, Shapira a Rogerse [28]. *Obrazec velikosti n* je dvojice mřížových cest (typu sever-východ, známe je z druhé přednášky) délky n, které mají společný počáteční a koncový bod, ale jinak se neprotínají. Množinu obrazců velikosti n označíme jako $\mathcal{A}(n)$.

Větička. Pro každé n platí $|\mathcal{A}(n)| = c_n = \frac{1}{n}\binom{2n-2}{n-1}$.

Poprvé to dokázal v r. 1959 Levine [11] a pak o deset let později jinak Pólya [15]. Jako příklad uvádíme prvky množiny $\mathcal{A}(4)$:



Nás ale více zajímá

Věta. Pro každé n platí

$$P_n := \sum_{X\in\mathcal{A}(n)} \text{Plocha}(X) = 4^{n-2}.$$

To objevil a dokázal někdy před rokem 1985 Schwarzler. Například v posledním obrázku máme čtyři obrazce s plochou 3 a jeden obrazec s plochou 4, což dohromady dává 16.

Woan, Shapiro a Rogers uvádějí, že není známo, zda se pro každé n dá beze zbytku a bez překrývání pomocí c_n obrazců velikosti n (můžeme je otáčet) pokrýt šachovnice $2^{n-2} \times 2^{n-2}$. Případ $n = 5$ se čtrnácti obrazci a standardní šachovnicí 8×8 je prý „amusing puzzle“. Vyzkoušejte si puzzle za DOM CV.

5. přednáška 31.3.1999

Důkaz větičky. Odvodíme explicitní vzorec pro obecnější veličinu $b(n,k)$ rovnající se počtu *k-otevřených obrazců velikosti n*, jimiž rozumíme dvojice mřížových cest délky n se společným počátečním vrcholem, které se jinak neprotínají a jejichž koncové vrcholy mají vzdálenost $k\sqrt{2}$. Platí rekurence ($k \geq 1$, $b(n,0) = 0$)

$$b(n,k) = b(n-1,k-1) + 2b(n-1,k) + b(n-1,k+1).$$

Jsou totiž dvě možnosti, jak obě cesty po jednom kroku zůstanou stejně daleko, ale jen jedna možnost, jak se po jednom kroku rozejdou nebo sejdou o $\sqrt{2}$.

Indukcí podle n dokážeme vzorec

$$b(n,k) = \frac{k}{n}\binom{2n}{n-k}.$$

Pro $n = 1$ platí: $b(1,0) = 0$ a $b(1,1) = 1$. Dále jde jen o manipulaci s binomickými koeficienty. Protože $2k = (k-1)+(k+1)$, můžeme po dosazení vzorce do pravé strany rekurence zjednodušovat:

$$\begin{aligned} & b(n,k-1) + 2b(n,k) + b(n,k+1) \\ = \; & \frac{k-1}{n}\binom{2n}{n-k+1} + \frac{2k}{n}\binom{2n}{n-k} + \frac{k+1}{n}\binom{2n}{n-k-1} \\ = \; & \frac{k-1}{n}\binom{2n+1}{n-k+1} + \frac{k+1}{n}\binom{2n+1}{n-k}. \end{aligned}$$

(Použili jsme základní rekurenci $\binom{a}{b} = \binom{a-1}{b} + \binom{a-1}{b-1}$.) Rozepíšeme-li $\frac{k-1}{n} = \frac{k-1}{n} - \frac{k}{n+1} + \frac{k}{n+1}$ a stejně tak i $\frac{k+1}{n}$, dostaneme, znovu s použitím základní binomické rekurence, že se poslední výraz rovná

$$\begin{aligned} & \frac{k}{n+1}\binom{2n+2}{n-k+1} \\ & + \frac{1}{n(n+1)}\left((n+k+1)\binom{2n+1}{n-k} - (n+1-k)\binom{2n+1}{n-k+1}\right). \end{aligned}$$

Rozdíl v závorce je roven nule a tak

$$b(n+1,k) = b(n,k-1) + 2b(n,k) + b(n,k+1) = \frac{k}{n+1}\binom{2n+2}{n-k+1}.$$

Tím jsme vyřídili i $|\mathcal{A}(n)|$, neboť

$$|\mathcal{A}(n)| = b(n-1,1) = \frac{1}{n-1}\binom{2n-2}{n-2} = \frac{1}{n}\binom{2n-2}{n-1} = c_n.$$

□

Důkaz věty. GF pro čísla $b(n,1)$ již máme:

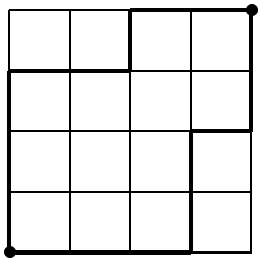
$$D(x) := \sum_{n\geq 1} b(n,1)x^n = \sum_{n\geq 1} c_{n+1}x^n = \frac{C(x)}{x} - 1.$$

($C(x)$ je GF Catalanových čísel.) Tvrdíme, že GF pro čísla $b(n,k)$ je rovna

$$\sum_{n\geq 1} b(n,k)x^n = \left(\sum_{n\geq 1} b(n,1)x^n)\right)^k = D(x)^k.$$

Abychom to nahlédli, v k-otevřeném obrazci velikosti n rozdělíme obě cesty na úseky délky $l_1, l_2, \ldots, l_k$, kde l_1 je jednoznačně určený počet kroků od počátku do okamžiku, kdy jsou obě cesty *naposledy* daleko $\sqrt{2}$, l_2 je jednoznačně určený počet kroků od této chvíle do okamžiku, kdy jsou obě cesty *naposledy* daleko $2\sqrt{2}$ a tak dále. Jistě $l_1 + l_2 + \cdots + l_k = n$. Počet možností pro i-tý úsek obou cest je ale stále $b(l_i, 1)$, protože šikmým posunutím úseku dolní cesty o $(i-1)\sqrt{2}$ ztotožníme počáteční vrcholy a obdržíme 1-otevřený obrazec velikosti l_i. Tudíž $b(n,k) = \sum b(l_1,1)b(l_2,1)\cdots b(l_k,1)$, kde sčítáme přes všechny k-tice přirozených čísel l_i splňující $l_1 + l_2 + \cdots + l_k = n$. To je přesně koeficient u x^n v $D(x)^k$.

Pro přirozené m rozumíme *m-diagonálou* diagonální (severozápadní směr) posloupnost m jednotkových čtverečků. Dále, $B(n,m)$ označuje celkový počet všech m-diagonál ve všech obrazcích $X \in \mathcal{A}(n)$. Například obrazec



přispívá 3 do $B(8,1)$, 3 do $B(8,2)$, 1 do $B(8,3)$ a 0 do $B(8,m)$ pro $m > 3$.

Je jasné, že hledaná plocha P_n se rovná

$$\sum_{m\geq 1} mB(n,m).$$

Tvrdíme, že GF pro čísla $B(n,m)$ je

$$B_m(x) := \sum_{n\geq 1} B(n,m)x^n = \left(\sum_{n\geq 1} b(n,m)x^n\right)^2 = D(x)^{2m}.$$

Vskutku, m-diagonála rozděluje obrazec $X \in \mathcal{A}(n)$ na dva m-otevřené obrazce velikosti l_1 a l_2, kde $l_1 + l_2 = n$. Proto $B(n,m) = \sum b(l_1,m)b(l_2,m)$, kde sčítáme přes dvojice přirozených čísel splňující $l_1 + l_2 = n$.

Pro GF celkové plochy dostáváme formuli

$$\begin{aligned} P(x) &:= \sum_{n\geq 1} P_n x^n = \sum_{n\geq 1} x^n \sum_{m\geq 1} mB(n,m) \\ &= \sum_{m\geq 1} m \sum_{n\geq 1} B(n,m)x^n = \sum_{m\geq 1} mB_m(x) \\ &= \sum_{m\geq 1} mD(x)^{2m}. \end{aligned}$$

Binomická věta pro exponent -2 praví, že

$$x + 2x^2 + 3x^3 + \cdots = \frac{x}{(1-x)^2}.$$

Proto

$$P(x) = \frac{D(x)^2}{(1-D(x)^2)^2}.$$

Protože $D(x) = C(x)/x - 1$ a $C(x)^2 - C(x) + x = 0$, máme $D = C^2/x$. Takže

$$\begin{aligned} P(x) &= \frac{C^4/x^2}{(1-D)^2(1+D)^2} = \frac{C^4/x^2}{(2-C/x)^2C^2/x^2} = \frac{C^2}{(2-C/x)^2} \\ &= \frac{x^2C^2}{(2x-C)^2} = \frac{x^2C^2}{4x^2-4xC+C^2} = \frac{x^2C^2}{4x^2-4xC+C-x} \\ &= \frac{x^2C^2}{(1-4x)(C-x)} = \frac{x^2}{1-4x}. \end{aligned}$$

Tudíž $P_n = [x^n]P(x) = [x^n]x^2/(1-4x) = 4^{n-2}$. □

STROMY. Náš druhý příklad pracuje se (zakořeněnými a rovinnými) stromy. Postupujeme podle článku Klazara [9]. Je-li $v \in V(T)$ vrchol stromu $T \in \mathcal{T}$, označuje $d(T,v)$ počet jeho dětí.

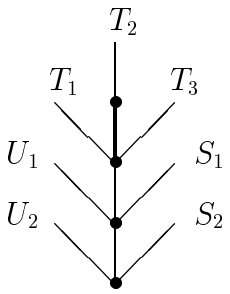
Věta. Platí

$$k_n := \sum_{T\in\mathcal{T}(n)} \sum_{v\in V(T)} 2^{d(T,v)} = \frac{1}{2}\left(4^{n-1} + \binom{2n-2}{n-1}\right).$$

Důkaz. Nejprve k_n vyložíme trochu jinak. *Košťata* jsou tyto stromy:

(Stromy výšky 1.) Koště K je *obsaženo* ve stromu T tehdy, když se K objevuje v T jako orientovaný podgraf. Očividně k_n počítá všechna košťata obsažená ve všech stromech $T \in \mathcal{T}(n)$, protože $2^{d(T,v)}$ je počet těch košťat obsažených v T, jejichž kořen splývá s v.

Na záležitost s košťaty se nyní podíváme z jejich hlediska. Zřejmě je k_n rovno také počtu rozšíření nějakého koštěte K na nějaký strom $T \in \mathcal{T}(n)$. Generické rozšíření K na T vypadá takto:



K je vyznačeno tučně, v našem příkladu má $k = 2$ vrcholy. Do mezer mezi sousedními hranami K vkládáme libovolně a nezávisle $2k-1$ stromů $T_i \in \mathcal{T}$. Z kořene K spustíme cestu s $l \geq 0$ vrcholy (v našem příkladu je $l = 2$), do nichž zakořeníme libovolně a nezávisle $2l$ stromů U_i a S_i. Jediné omezení je, že celkový počet vrcholů musí být n.

Po chvilkové meditaci nad obrázkem vidíme rovnost

$$K(x) := \sum_{n\geq 1} k_n x^n = \sum_{l\geq 0} \left(\frac{C(x)^2}{x}\right)^l \cdot \sum_{k\geq 1} x^k \left(\frac{C(x)}{x}\right)^{2k-1}.$$

Takže

$$\begin{aligned} K(x) &= \frac{1}{1-C^2/x} \cdot \frac{x}{C} \cdot \frac{C^2/x}{1-C^2/x} = \frac{x^2 C}{(x-C^2)^2} \\ &= \frac{x^2 C}{(2x-C)^2} = \frac{x^2 C}{4x^2 - 4xC + C - x} = \frac{x^2 C}{(1-4x)(C-x)} \\ &= \frac{x}{1-4x} \cdot \frac{x}{C} = \frac{x}{1-4x} \cdot \frac{1+\sqrt{1-4x}}{2} \\ &= \frac{1}{2}\left(\frac{x}{1-4x} + \frac{x}{\sqrt{1-4x}}\right). \end{aligned}$$

Pomocí binomické formule s exponentem $-1/2$ dostáváme

$$\begin{aligned} k_n &= [x^n]K(x) = \tfrac{1}{2}[x^{n-1}]((1-4x)^{-1} + (1-4x)^{-1/2}) \\ &= \frac{1}{2}\left(4^{n-1} + \binom{2n-2}{n-1}\right). \end{aligned}$$

□

6. přednáška 7.4.1999

V celém počítání jsme ale o košťatech potřebovali vědět jen to, že pro každé $n \in \mathbf{N}$ máme právě jedno koště s n vrcholy. (Tomu odpovídá koeficient 1 u x^k v první formuli pro $K(x)$.) Dál již nic nezávisí na tvaru koštěte. Dokázali jsme více:

Pozorování. Nechť $\mathcal{S}, \mathcal{S} \subset \mathcal{T}$ je třída stromů, v níž je pro každé $n \in \mathbf{N}$ právě jeden strom s n vrcholy. Nechť, pro $T \in \mathcal{T}$, funkce $w_{\mathcal{S}}(T)$ počítá celkový počet způsobů, jak se nějaký strom z $\mathcal{S}$ objevuje ve stromu T jako orientovaný podgraf. Pak

$$w_{\mathcal{S}}(n) := \sum_{T\in\mathcal{T}(n)} w_{\mathcal{S}}(T) = \frac{1}{2}\left(4^{n-1} + \binom{2n-2}{n-1}\right).$$

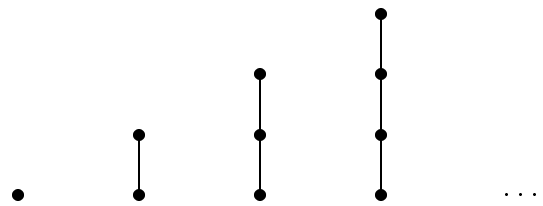
Důkaz. Stejný jako pro třídu košťat. □

Pomocí pozorování odvodíme další výsledek, v němž se „záhadně“ objevuje mocnina čtyř. Připomínáme, že každý strom $T \in \mathcal{T}$ je též částečně uspořádaná množina: pro dva vrcholy u a v položíme $u \leq_T v$, právě když u leží na cestě spojující kořen a v. Řekneme, že vrcholy u a v jsou ve stromu T *porovnatelné*, pokud $u \leq_T v$ nebo $v \leq_T u$.

Věta. Platí

$$\sum_{T \in \mathcal{T}(n)} \#\{(u,v) \in V(T) \times V(T) : \ u \text{ a } v \text{ jsou v } T \text{ porovnatelné}\} = 4^{n-1}.$$

Důkaz. Pozorování použijeme pro třídu stromů $\mathcal{S}$ rovnou cestám:



Nyní $w_{\mathcal{S}}(n)$ počítá všechny dvojice vrcholů (u,v) ve všech stromech $T \in \mathcal{T}(n)$, že $u \leq_T v$. Hledaná hodnota se proto rovná dvojnásobku $w_{\mathcal{S}}(n)$ minus počet diagonálních dvojic (u,u) (bez odečtení bychom je započetli dvakrát):

$$2w_{\mathcal{S}}(n) - \sum_{T \in \mathcal{T}(n)} \sum_{u \in V(T)} 1 = 4^{n-1} + \binom{2n-2}{n-1} - n \cdot c_n = 4^{n-1}.$$

□

IV. LAGRANGEOVA INVERZNÍ FORMULE

Lagrangeova inverzní formule (LIF) je velmi užitečný klasický výsledek o mocninných řadách.

Věta (LIF). Nechť $\varphi(u) \in \mathbf{C}[[u]]$ je mocninná řada splňující $\varphi(0) = 1$ a $w = w(u) \in \mathbf{C}[[u]]$ je jednoznačně určené řešení funkcionální rovnice

$$w = u \cdot \varphi(w).$$

Pak

$$[u^n]w = \tfrac{1}{n}[u^{n-1}]\varphi(u)^n.$$

Je-li $f(u) \in \mathbf{C}[[u]]$ další mocninná řada, platí obecněji

$$[u^n]f(w) = \tfrac{1}{n}[u^{n-1}]f'(u)\varphi(u)^n.$$

Důkaz. Později. □

Ekvivalentní důsledek. Nechť $\varphi(u) \in \mathbf{C}[[u]]$, přičemž $[u^0]\varphi = 0$ a $[u^1]\varphi = 1$. Pak

$$[u^n]\varphi^{\langle -1\rangle} = \tfrac{1}{n}[u^{n-1}]\left(\frac{u}{\varphi(u)}\right)^n.$$

Důkaz. Řada $w = w(u) = \varphi^{\langle -1\rangle}(u)$ je řešením rovnice $u = \varphi(w)$, kterou přepíšeme jako $w = u \cdot (w/\varphi(w))$. Zbytek plyne pomocí LIF. Obdobně se naopak odvodí LIF z výsledku o funkcionálním inverzu. □

Předvedeme si několik klasických použití LIF. Nejprve spočítáme stromy $T \in \mathcal{T}$ s daným počtem vrcholů, jejichž každý vrchol má stupeň (tj. počet dětí) rovný 0 nebo 3. Hledaný počet označíme jako a_n — například $a_1 = 1, a_2 = 0$ a $a_7 = 3$ — a definujeme GF

$$A = A(x) = \sum_{n\geq 1} a_n x^n.$$

Má-li strom T více vrcholů než jeden, má jeho kořen tři děti a v nich jsou zakořeněné tři stromy téhož typu. Dostáváme rovnici

$$A = x + xA^3 = x(1 + A^3),$$

která je šitá na míru LIF. Podle ní

$$\begin{aligned} a_n &= [x^n]A = \tfrac{1}{n}[x^{n-1}](1+x^3)^n \\ &= \tfrac{1}{n}[x^{n-1}]\sum_{k=0}^{n}\binom{n}{k}x^{3k} \\ &= \frac{1}{n}\binom{n}{(n-1)/3} \end{aligned}$$

pro $n-1$ dělitelné třemi a $a_n = 0$ jinak.

Úplně stejně se spočítají stromy, jejichž vrcholy mají stupeň rovný 0 nebo k. I vzorec pro Catalanova čísla vyplyne pomocí LIF: rovnice $C^2 - C + x = 0$ se přepíše jako

$$C = \frac{x}{1 - C}$$

a podle LIF

$$\begin{aligned} c_n &= [x^n]C = \tfrac{1}{n}[x^{n-1}](1-x)^{-n} \\ &= \tfrac{1}{n}[x^{n-1}] \sum_{k \geq 0} \binom{-n}{k} (-x)^k = \frac{1}{n}\binom{-n}{n-1}(-1)^{n-1} \\ &= \frac{1}{n} \cdot \frac{(-n) \cdot (-n-1) \cdot \ldots \cdot (-2n+2)}{(n-1)!} \cdot (-1)^{n-1} \\ &= \frac{1}{n} \cdot \frac{n \cdot (n+1) \cdot \ldots \cdot (2n-2)}{(n-1)!} = \frac{1}{n}\binom{2n-2}{n-1}. \end{aligned}$$

Podívejme se, co nám LIF řekne o řešení $w = w(t) \in \mathbf{C}[[t]]$ rovnice

$$w = t \cdot \mathrm{e}^w.$$

Říká, že

$$[t^n]w = \tfrac{1}{n}[t^{n-1}]\mathrm{e}^{nt} = \frac{n^{n-1}}{n!}.$$

Takže

$$w(t) = \sum_{n \geq 1} \frac{n^{n-1} t^n}{n!}.$$

Podobnost s Cayleyovou formulí n^{n-2} pro počet všech označených (ne zakořeněných rovinných) stromů na množině $\{1, 2, \ldots, n\}$ není náhodná, takto ji později odvodíme. Obecná verze LIF s (například) $f(t) = t^2$ nám dá

$$[t^n]w(t)^2 = \tfrac{1}{n}[t^{n-1}]2t\mathrm{e}^{nt} = \frac{2}{n} \cdot \frac{n^{n-2}}{(n-2)!}.$$

Koeficient $[t^n]w(t)^2$ lze však spočítat přímo z definice, vyjde jistá suma. Porovnáním tak jako vedlejší produkt LIF dostáváme identitu

$$\sum_{i=1}^{n-1} \binom{n}{i} i^{i-1}(n-i)^{n-i-1} = 2(n-1)n^{n-2}.$$

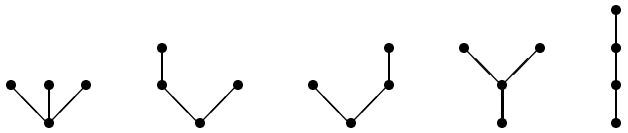
(Vše jsme vynásobili $n!$.) Jde o speciální případ Abelova zobecnění binomické formule.

Nezávislé množiny. Jako poslední příklad užitečnosti LIF nalezneme celkový počet všech nezávislých množin ve všech stromech $T \in \mathcal{T}(n)$. Postupujeme podle Klazara [9]. Množina $X \subset V(T)$ je *nezávislá*, nejsou-li žádné její dva vrcholy spojené hranou. Počet všech (včetně $\emptyset$) nezávislých podmnožin v T označíme $w(T)$ a počet těch z nich (opět včetně $\emptyset$), které neobsahují kořen stromu T, jako $z(T)$. Zajímají nás veličiny

$$w(n) := \sum_{T \in \mathcal{T}(n)} w(T) \quad \text{a} \quad z(n) := \sum_{T \in \mathcal{T}(n)} z(T).$$

Pro ilustraci uvádíme hodnoty funkcí w a z na čtyřvrcholových stromech:

$z(T) =$	8	6	6	5	5
$w(T) =$	9	8	8	9	8



Věta. Platí

$$w(n) = \frac{1}{n-1}\binom{3n-3}{n} \quad \text{a} \quad z(n) = \frac{1}{n}\binom{3n-2}{n-1}.$$

Důkaz. Jak počítat $w(T)$ a $z(T)$ pro daný strom T? Je-li T jednovrcholový, máme $z(T) = 1$ a $w(T) = 2$. Má-li T více vrcholů, označíme jako $T_1, T_2, \ldots, T_k$ podstromy zakořeněné v dětech kořene T. Lehce se nahlédnou rekurence

$$z(T) = \prod_{i=1}^{k} w(T_i) \quad \text{a} \quad w(T) = \prod_{i=1}^{k} w(T_i) + \prod_{i=1}^{k} z(T_i).$$

První je jasná, nezávislou množinu ve stromu T neobsahující kořen dostaneme tak, že v každém podstromu T_i zvolíme libovolně nezávislou množinu. V druhé rekurenci ještě připočteme nezávislé množiny, které kořen obsahují.

Definujeme GF

$$F(x) = \sum_{n\geq 1} w(n)x^n = \sum_{T\in\mathcal{T}} w(T)x^{v(T)} \quad \text{a} \quad G(x) = \sum_{n\geq 1} z(n)x^n = \sum_{T\in\mathcal{T}} z(T)x^{v(T)}$$

($v(T)$ počítá vrcholy stromu T). Rekurence se překládají do rovnic

$$\begin{aligned} G(x) &= x\sum_{k\geq 0} F(x)^k = \frac{x}{1-F(x)} \\ F(x) &= x\sum_{k\geq 0} G(x)^k + x\sum_{k\geq 0} F(x)^k = \frac{x}{1-G(x)} + \frac{x}{1-F(x)}. \end{aligned}$$

Dostali jsme soustavu

$$F = \frac{x}{1-G} + \frac{x}{1-F} \quad \text{a} \quad G = \frac{x}{1-F}.$$

7. přednáška 14.4.1999

Eliminujeme-li ze soustavy G (G v první rovnici nahradíme $x/(1-F)$), obdržíme po úpravách vztah $F^3 - 2F^2 + (1+2x)F + x^2 - 2x = 0$. To moc nadějně nevypadá. Eliminujeme-li F (z druhé rovnice vyjádříme F jako $F = 1 - x/G$ a dosadíme do první rovnice $F = x/(1-G) + G$), obdržíme po úpravách vztah $G^3 - 2G^2 + G - x = 0$. Jinými slovy,

$$G = \frac{x}{(1-G)^2}.$$

A to je jiná káva. Podle LIF

$$\begin{aligned} z(n) &= [x^n]G(x) = \tfrac{1}{n}[x^{n-1}](1-x)^{-2n} \\ &= \tfrac{1}{n}[x^{n-1}]\sum_{k\geq 0}\binom{-2n}{k}(-x)^k \\ &= \ldots = \frac{1}{n}\binom{3n-2}{n-1}. \end{aligned}$$

Jak ale dopočítat $w(n)$? Ukážeme, že je splněna lineární diferenciální rovnice

$$3xF' - 4xG' - 2(F-G) = 0.$$

Ta pro koeficienty dává relaci $(3n-2)w(n) = (4n-2)z(n)$. Vzorec pro $z(n)$ tak po lehkých úpravách poskytne i vzorec pro $w(n)$:

$$w(n) = \frac{1}{n-1}\binom{3n-3}{n}.$$

Zbývá dokázat onu relaci mezi xF', xG' a $F-G$. Z výchozí soustavy pro F a G je jasné, že

$$F - G = \frac{x}{1-G}.$$

Zderivujeme-li podle x kubickou rovnici pro G a vyjádříme-li G', dostaneme $G' = 1/(3G^2 - 4G + 1)$. Takže

$$xG' = \frac{x}{3G^2 - 4G + 1} = \frac{1}{1-3G} \cdot \frac{x}{1-G}.$$

Konečně, zderivujeme-li podle x rovnici $F = x/(1-G) + G$, dostaneme $F' = G' + 1/(1-G) + xG'/(1-G)^2$. Díky $x/(1-G)^2 = G$ máme $F' = G' + 1/(1-G) + GG'$. Takže

$$xF' = \frac{x}{1-G} + (1+G)xG' = \frac{2-2G}{1-3G} \cdot \frac{x}{1-G}.$$

Z těchto vyjádření xF', xG' a $F-G$ vyplývá, že jejich lineární kombinace s koeficienty $3, -4$ a -2 je identicky nulová. □

Důkaz LIF. Postupujeme podle knihy Gouldena a Jacksona [7]. Od $\mathbf{C}[[x]]$ přejdeme k obecnější struktuře

$$\mathbf{C}((x)) = \{\textstyle\sum_{n\geq k} a_n x^n : \ a_n \in \mathbf{C}, k \in \mathbf{Z}\},$$

to jest k rozvojům s konečným počtem mocnin se záporným exponentem. Říká se jim *Laurentovy řady*. $(\mathbf{C}((x)), +, \cdot)$ je těleso: označíme-li pro $f \in \mathbf{C}((x))$ jako $\mathrm{val}(f)$ nejmenší $k \in \mathbf{Z}$ takové, že $[x^k]f \neq 0$, máme $f = x^{\mathrm{val}(f)}g$, kde $g \in \mathbf{C}[[x]]$ a $[x^0]g \neq 0$, a $1/f = x^{-\mathrm{val}(f)}g^{-1}$ (jak víme, g^{-1} v $\mathbf{C}[[x]]$ existuje).

Reziduem $f \in \mathbf{C}((x))$ rozumíme koeficient $[x^{-1}]f$. Jeho výsadní postavení plyne ze skutečnosti, že x^{-1} jako jediná celočíselná mocnina $x^n, n \in \mathbf{Z}$ není derivací žádné řady $g \in \mathbf{C}((x))$.

Pozorování. Pro každé dvě Laurentovy řady $f, g \in \mathbf{C}((x))$ platí identity

$$[x^{-1}]f' = 0 \quad \text{a} \quad [x^{-1}]f'g = -[x^{-1}]fg'.$$

Důkaz. První rovnost je ona základní vlastnost rezidua. Druhá plyne z ní a z Leibnizovy formule $(fg)' = f'g + fg'$. □

Tvrzení (reziduum a substituce). Nechť $f, r \in \mathbf{C}((x))$ jsou Laurentovy řady, přičemž $k = \mathrm{val}(r) > 0$ (aby substituce $f(r(x))$ byla definovaná). Pak

$$k[x^{-1}]f(x) = [x^{-1}]f(r(x))r'(x).$$

Důkaz. Nejprve ověříme speciální případ $f(x) = x^n, n \in \mathbf{Z}$. Pro $n \neq -1$

$$[x^{-1}]r(x)^n r'(x) = \tfrac{1}{n+1}[x^{-1}](r(x)^{n+1})' = 0,$$

podle základní vlastnosti rezidua. Vlevo máme též nulu: $k[x^{-1}]x^n = 0$.

Nechť $n = -1$. Máme $r(x) = bx^k h(x)$, kde $b \neq 0$ a $h \in \mathbf{C}[[x]]$ splňuje $h(0) = 1$. Tedy existují mocninné řady $1/h$ a $\log(h)$. (Jak víme, $\log(h) = \log(1 + (h-1)) = \sum_{n\geq 1}(-1)^n(h-1)^n/n$.) Podle základní vlastnosti rezidua

$$\begin{aligned}
[x^{-1}]r(x)^{-1}r'(x) &= [x^{-1}]b^{-1}x^{-k}h(x)^{-1} \cdot (bkx^{k-1}h(x) + bx^k h'(x)) \\
&= [x^{-1}](kx^{-1} + h'(x)/h(x)) = k + [x^{-1}](\log h(x))' \\
&= k.
\end{aligned}$$

Vlevo máme taky k: $[x^{-1}]kx^{-1} = k$.

Pro obecnou Laurentovu řadu $f(x) = \sum_{n\geq l} a_n x^n$ se $k[x^{-1}]f$ rovná ka_{-1}. Což se rovná pravé straně, podle předchozí úvahy totiž

$$[x^{-1}]\sum_{n\geq l} a_n r(x)^n r'(x) = [x^{-1}]a_{-1}r(x)^{-1}r'(x) = a_{-1}k.$$

□

Vlastní důkaz LIF. Dokazujeme větu ze strany 26. Řada $w = w(x) \in \mathbf{C}[[x]]$ je řešením rovnice $w = x \cdot \varphi(w)$. Položíme $\Phi(x) = x/\varphi(x)$. Pak, podle předpokladu o $\varphi(x)$, $\mathrm{val}(\Phi) = 1$. Dále $\Phi(w(x)) = x$, a tak $w(x) = \Phi(x)^{\langle -1\rangle}$. Pro libovolnou mocninnou řadu f a číslo $n \in \mathbf{N}$ dostáváme

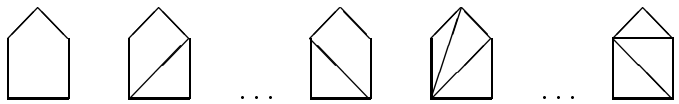
$$\begin{aligned}
[x^n]f(w(x)) &= [x^{-1}]x^{-(n+1)}f(\Phi(x)^{\langle -1\rangle}) \\
&= [x^{-1}]\Phi(x)^{-(n+1)}f(x)\Phi'(x) \\
&= -\tfrac{1}{n}[x^{-1}]f(x)(\Phi(x)^{-n})' \\
&= \tfrac{1}{n}[x^{-1}]f'(x)\Phi(x)^{-n} \\
&= \tfrac{1}{n}[x^{n-1}]f'(x)\varphi(x)^n.
\end{aligned}$$

Na druhý řádek jsme přešli pomocí substituce $x := \Phi(x)$ a posledního tvrzení. Na čtvrtý jsme se dostali pomocí druhé rovnosti z posledního pozorování. Při přechodu na pátý jsme $\Phi(x)$ nahradili $x/\varphi(x)$. □

V. SCHRÖDEROVA A MOTZKINOVA ČÍSLA

Jsou to blízcí příbuzní Catalanových čísel, protože jejich GF splňují kvadratické rovnice.

SCHRÖDEROVA ČÍSLA. P buď konvexní n-úhelník s vrcholy očíslovanými $1, 2, \ldots, n$ proti směru hodinových ručiček. *Rozřezáním* P rozumíme jakýkoli (i prázdný) systém úhlopříček v P, v němž se žádné dvě úhlopříčky nekříží. Například pro $n = 5$ máme těchto 11 rozřezání:



Jako a_n označíme počet všech rozřezání P a jako b_n počet těch, v nichž z 1 nevychází žádná úhlopříčka. Takže $a_1 = a_2 = b_1 = b_2 = 0$, $a_3 = b_3 = 1$, $a_4 = 3$, $b_4 = 2$ atd. Nalezneme GF

$$F = F(x) = \sum_{n\geq 1} a_n x^n \quad \text{a} \quad G = G(x) = \sum_{n\geq 1} b_n x^n.$$

Uvážíme rozřezání P, v nichž z 1 vychází alespoň jedna úhlopříčka. Rozříznutím P podle nejlevější z těchto úhlopříček dostaneme rozřezání P_1 a P_2, přičemž v prvním z 1 nevychází úhlopříčka, druhé je obecné a mnohoúhelníky P_1 a P_2 mají celkem $n + 2$ vrcholů. Z tohoto rozkladu plyne rovnice

$$F = G + \frac{GF}{x^2}.$$

8. přednáška 21.4.1999

Druhou rovnici

$$G = 2xF + x^3$$

dostaneme tak, že rozřezání P ($n > 3$), v nichž z 1 nevychází žádná úhlopříčka rozdělíme na dvě skupiny podle toho, zda vrcholy n a 2 jsou nebo nejsou spojeny. Oba případy se lehce převedou na obecné rozřezání $(n-1)$-úhelníka, a tak vidíme, že v obou skupinách máme a_{n-1} rozřezání.

Eliminujeme-li ze soustavy G, dostaneme pro F rovnici

$$2F^2 + (3x^2 - x)F + x^4 = 0.$$

Pro F tak máme formuli

$$F = F(x) = \tfrac{1}{4}x(1 - 3x - \sqrt{x^2 - 6x + 1}).$$

Zvolili jsme řešení se znaménkem minus, protože $F(x) = x^3 + \cdots$. Víme, že $a_3 = 1, a_4 = 3, a_5 = 11$. Odvodíme rekurenci pro počítání a_n. Místo F budeme pracovat s $H = \frac{F}{x} = \frac{1}{4}(1 - 3x - \sqrt{1 - 6x + x^2}) = \sum_{n\geq 3} a_n x^{n-1}$. Protože

$$\begin{aligned} (x-3)\cdot H &= \tfrac{1}{4}(-3 + 10x - 3x^2 - (x-3)\sqrt{\ldots}) \\ (1 - 6x + x^2)\cdot H' &= \tfrac{1}{4}(-3 + 18x - 3x^2 - (x-3)\sqrt{\ldots}), \end{aligned}$$

splňuje H diferenciální rovnici

$$(1 - 6x + x^2)H' - (x-3)H = 2x.$$

Pro $n > 1$ je tedy koeficient u x^n v mocninné řadě vlevo roven nule, což je vyjádřeno vztahem $a_{n+2}(n+1) - a_{n+1}(6n-3) + a_n(n-2) = 0$. Takže

$$a_{n+2} = \frac{(6n-3)a_{n+1} - (n-2)a_n}{n+1}.$$

Dostáváme hodnoty

$$a_6 = \frac{21\cdot 11 - 2\cdot 3}{5} = 45,\ a_7 = \frac{27\cdot 45 - 3\cdot 11}{6} = 197, \ldots$$

Posloupnost

$$\{s_n\}_{n\geq 1} = \{a_n\}_{n\geq 3} = \{1, 3, 11, 45, 197, 903, 4279, 20793, \ldots\}$$

se nazývá posloupností *Schröderových čísel*. Je pojmenována podle E. Schrödera, který ji zavedl v roce 1870 v [19].

Uvedeme si tři explicitní vzorce pro s_n vyskytující se v literatuře.

$$\begin{aligned} s_n &= \frac{1}{2}\sum_{j=0}^{n}\frac{1}{j+1}\binom{2j}{j}\binom{j+n}{2j} \\ s_n &= \frac{1}{n+1}\sum_{j=0}^{n}(-1)^j 2^{n-j}\binom{n+1}{j}\binom{2n-j}{n} \\ s_n &= \sum_{j=0}^{\lfloor (n+1)/2\rfloor}(-1)^j\frac{(2n-2j-1)!!}{j!(n+1-2j)!}\cdot\frac{3^{n+1-2j}}{2^{2+j}} \quad (n>1). \end{aligned}$$

V posledním vzorci $(2m+1)!!$ označuje *lichý faktoriál*, součin $1\cdot 3\cdot 5\cdot\ldots\cdot(2m+1)$.

Za DOM CV uhodněte, která ze tří formulí se dá odvodit LIFou, a odvoďte ji tak.

Singularita funkce $\sqrt{1-6x+x^2}$ nejbližší k počátku se dostane z kvadratické rovnice $x^2-6x+1=0$, jejíž řešení jsou $3\pm 2\sqrt{2}$. Schröderova čísla tedy rostou zhruba jako $(3-2\sqrt{2})^{-n}=(3+2\sqrt{2})^n=(5.828\ldots)^n$.

Další struktura počítaná Schröderovými čísly. Pro pevné $n\in\mathbf{N}$ uvažme rozklad množiny $[l]=\{1,2,\ldots,l\}$ (l může být libovolné) na n bloků, přičemž (i) žádná dvě čísla m a $m+1$ nejsou v témže bloku, (ii) jde o nekřížící se rozklad a (iii) 1 a l jsou v tomtéž bloku. Počet takových rozkladů označíme jako r_n. Připomínáme, že nekřížící se rozklad je ten, pro nějž neexistují čtyři čísla $1\le a<b<c<d\le l$ a dva různé bloky A a B tak, že $a,c\in A$ a $b,d\in B$.

Tyto struktury se dají přehledněji reprezentovat pomocí posloupností. Místo rozkladu $[l]$ na n bloků vezmeme posloupnost $u=a_1a_2\ldots a_l$, kde $a_i=a_j$ tehdy a jen tehdy, když i a j jsou v témže bloku rozkladu. Přidáme ještě *normalizační požadavek*: $\{a_1,a_2,\ldots,a_l\}=\{1,2,\ldots,n\}$ a $1\le i<j\le n$ implikuje, že první výskyt i v u přechází první výskyt j. Posloupnost u je pak pro daný rozklad určena jednoznačně. Je jasné, jak z ní rozklad zpětně vyčteme. Hořejší podmínka (i) říká, že $a_j\ne a_{j+1}$ pro každé j. Podmínka (ii) zakazuje výskyt podposloupnosti typu *abab*. Podmínka (iii) chce, aby $a_l=1$ (vždy $a_1=1$). Například $r_3=3$, jak dosvědčují rozklady

$$1231,\ 12321 \text{ a } 12131.$$

Tvrzení. Posloupnost $\{r_n\}_{n\ge 2}$ je posloupnost Schröderových čísel.

Důkaz. Odvodíme rovnici pro GF

$$F = \sum_{n\geq 1} r_n x^n = x + x^2 + 3x^3 + \cdots.$$

Vezmeme libovolný rozklad $[l]$ na n bloků vyhovující podmínkám (i)–(iii) a reprezentujeme ho posloupností u. Výskyty jedničky rozdělují u na úseky: $u = 1u_1 1u_2 1 \ldots 1u_k 1$. Úseky u_i se mohou nezávisle na sobě (vzhledem k podmínce (ii) nesdílejí symboly) volit jako *neprázdné* rozklady splňující podmínky (i) a (ii). Podmínku (iii) obecně nesplňují (u_i nesplňují ani normalizační požadavek, to je ale jen formální závada). Nečiní to velký problém, snadno se totiž vidí, že počet rozkladů majících m bloků a splňujících (i) a (ii) je pro $m > 1$ roven $2r_m$ (chybějící koncovou jedničku lze vždy přidat); pro $m = 1$ máme jen jeden takový rozklad. Dostáváme rovnici

$$F = x \sum_{k\geq 0} (2F - x)^k = \frac{x}{1 + x - 2F},$$

to jest kvadratickou rovnici

$$2F^2 - (1 + x)F + x = 0.$$

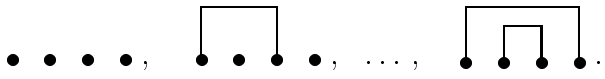
Jejím řešením je mocninná řada

$$F = F(x) = \frac{1 + x - \sqrt{1 - 6x + x^2}}{4}.$$

Což je, až na nepodstatné odchylky, vzorec pro GF Schröderových čísel. □

Motzkinova čísla. Zavedl je Th. Motzkin v roce 1948 v [13], když zkoumal následující problém. Na kružnici je umístěno n bodů. Kolika způsoby se dají spojit vzájemně disjunktními tětivami? Žádné dvě tětivy nemají společný bod a jejich počet je libovolný, mezi 0 a $n/2$. Počet možností označíme jako m_n.

Úlohu mírně přeformulujeme. Máme dáno n bodů nakreslených ve vodorovné řadě. Kolika způsoby je (ne nutně všechny) můžeme spojit oblouky, které všechny leží nad touto řadou a jsou vzájemně disjunktní? Pro $n = 4$ to lze 9 způsoby:



Rovnice pro GF

$$M = \sum_{n\geq 0} m_n x^n = 1 + x + 2x^2 + \cdots$$

se odvodí lehoulinoučce. První z bodů buď není nebo je koncovým bodem oblouku. Není-li, můžeme zbylých $n-1$ bodů propojovat oblouky libovolně. Je-li, určuje druhý konec oblouku dvě skupiny bodů, které mají dohromady $n-2$ členů, a na každé z nich můžeme libovolně a nezávisle na druhé kreslit oblouky (mezi skupinami oblouk vést nemůže). Stručně řečeno,

$$M = 1 + xM + x^2M^2.$$

Kvadratická rovnice $x^2M^2 + (x-1)M + 1 = 0$ dává vzorec

$$M = M(x) = \frac{1 - x - \sqrt{1 - 2x - 3x^2}}{2x^2} = \frac{1 - x - \sqrt{(1-3x)(1+x)}}{2x^2}.$$

Posloupnost

$$\{m_n\}_{n\geq 1} = \{1, 2, 4, 9, 21, 51, 127, 323, 835, \ldots\}$$

je posloupnost *Motzkinových čísel*. Je jasné, že rostou zhruba jako 3^n.

Rekurence pro m_n se odvodí podobně jako pro Schröderova čísla. Ponecháváme to za DOM CV.

Explicitní formule? Jedna plyne přímo ze samé definice:

$$\begin{aligned} m_n &= \sum_{k\geq 0} \binom{n}{2k} \cdot \#\text{ dobrých uzávorkování s } k \text{ dvojicemi závorek} \\ &= \sum_{k\geq 0} \frac{1}{k+1} \binom{2k}{k} \binom{n}{2k}. \end{aligned}$$

Oblouky totiž vytvářejí dobrá uzávorkování, která jsme spočetli ve třetí přednášce. Jiná možnost je využít binomickou větu a rozvinout podle ní $(1-3x)^{1/2}(1+x)^{1/2}$ v hořejším vzorci. To po menším výpočtu dá vztah

$$m_n = \frac{1}{2 \cdot 4^{n+1}} \sum_{i=0}^{n+2} (-1)^{n+1+i} 3^i c_i c_{n+2-i},$$

kde $c_n = \binom{2n-2}{n-1}/n$ jsou Catalanova čísla a $c_0 = -1/2$.

Další struktura počítaná Motzkinovými čísly. Jsou jí zakořeněné rovinné stromy s n vrcholy, v nichž žádný vrchol s případnou výjimkou kořene nemá jen jedno dítě. (Řadu dalších struktur počítaných m_n uvádějí Donaghey a Shapiro v [4].) Počet těchto stromů označíme a_n. Kupříkladu $a_5 = 4$:

Ukážeme, že a_n jsou až na posun indexu Motzkinova čísla. Pomocná GF

$$B = \sum_{n\geq 0} b_n x^n = x + x^3 + \cdots$$

počítá stromy, v nichž vůbec žádný vrchol nemá jen jedno dítě. Pro ni máme — z rozkladu na podstromy zakořeněné v dětech kořene — vztah

$$B = x(1 + B^2 + B^3 + \cdots) = x\left(\frac{1}{1-B} - B\right).$$

Takže

$$(1+x)B = \frac{x}{1-B}.$$

Celkem dostáváme pro B rovnici $(1+x)B^2 - (1+x)B + x = 0$. Hledaná GF $A = \sum_{n\geq 0} a_n x^n$ splňuje

$$A = x(1 + B + B^2 + \cdots) = \frac{x}{1-B} = (1+x)B.$$

Z rovnice pro B tak dostaneme hned rovnici pro A, totiž

$$A^2 - (1+x)A + x(1+x) = 0.$$

Jejím vyřešením dostáváme motzkinovský vzorec

$$A = A(x) = \tfrac{1}{2}(1 + x - \sqrt{1 - 2x - 3x^2})$$

a vše je jasné.

9. přednáška 5.5.1999

Literatura a informace o kombinatorické enumeraci. **1. Knihy.** Comtet: Advanced Combinatorics [2], Goulden a Jackson: Combinatorial Enumeration [7], kapitoly ve van Lintovi a Wilsonovi: A Course in Combinatorics [26], Wilf: Generatingfunctionology [27], kapitola v Lovászovi: Combinatorial Problems and Exercises [12], Stanley: Enumerative Combinatorics Vol I [21] a právě vyšlý Vol II [22], pasáže v Knuthovi: The Art of Computer Programming [10], dvě kapitoly v Handbook of Combinatorics [8] a sice 21. Gessel a Stanley: Algebraic Enumeration a hlavně 22. Odlyzko: Asymptotic Enumeration Methods. Většina toho ovšem v knihovně v Karlíně není. **2. Časopisy.** Například Journal of Combinatorial Theory A, Discrete Mathematics, European Journal of Combinatorics, ... **3. Internet.** Electronic Journal of Combinatorics (EJC) [5], Discrete Mathematics & Theoretical Computer Science [3], aj. Stránky hyperaktivních kombinatoriků: Flajolet, Gessel, Knuth, Odlyzko, Sloane, Zeilberger, ... (další odkazy viz www stránka EJC). Sloane: Handbook of Integer Sequences na Sloanově www stránce, který vyšel nejprve knižně [20]. Umožňuje testovat, zda vaše oblíbená posloupnost čísel je přítomna v rozsáhlé databázi (tj. zda se tímto nebo ekvivalentním problémem už někdo zabýval), popř. zda tam je přítomna modifikace vaší posloupnosti.

VI. POUŽITÍ GF V TEORII PRAVDĚPODOBNOSTI

Větvící se náhodný proces. Jedinec zplodí k dětí s pravděpodobností p_k, kde $k = 0, 1, 2, \ldots$, a umírá. Jeho děti mají opět děti se stejnými pravděpodobnostmi, umírají a vše pokračuje stejně dále. Zajímají nás počty jedinců v n-té generaci, zejména, co se dá říci o pravděpodobnosti q_n, že v n-té generaci (tím pádem i v dalších) všichni vymřeli.

Je možný i technicko-budovatelský pohled Rényiho [16], podle jehož knihy zde postupujeme. Na první z mnoha stínítek dopadne elektron, při srážce zanikne, ale vytvoří k nových elektronů s pravděpodobností p_k. Vzniklé elektrony dopadnou na druhé stínítko a každý z nich vytvoří podle téhož zákona další elektrony atd. Rozumí se, že události vzniku elektronů v jednotlivých srážkách jsou vzájemně nezávislé.

Důležitým parametrem popisujícím náš proces je

$$M = \sum_{k \geq 0} k p_k,$$

střední (očekávaná) hodnota počtu dětí jedince, popř. počtu elektronů zrozených ve srážce.

Větička. Existuje limita q pravděpodobností vymření q_n,

$$\lim_{n\to\infty} q_n = q, \text{ a } q \begin{cases} = 1 & \text{pro } M \le 1 \text{ a} \\ < 1 & \text{pro } M > 1. \end{cases}$$

Důkaz. Vyloučíme degenerované případy $p_0 = 0, 1$ a předpokládáme, že $0 < p_0 < 1$. Zavedeme GF

$$G(z) = \sum_{k\ge 0} p_k z^k,$$

GF rozdělení pravděpodobnosti, nástroj v teorii pravděpodobnosti hojně užívaný. Podobně definujeme

$$G_n(z) = \sum_{k\ge 0} p_{n,k} z^k,$$

kde $p_{n,k}$ udává pravděpodobnost, že n-tá generace obsahuje k jedinců. Klademe $G_1 = G$. Protože $\sum_{k=0}^{\infty} p_{n,k} = 1$, máme $G_n(1) = 1$. Je rovněž očividné, že funkce G_n jsou definovány pro každé $|z| < 1$. Platí kruciální vztah

$$G_{n+1}(z) = G_n(G(z)),$$

protože

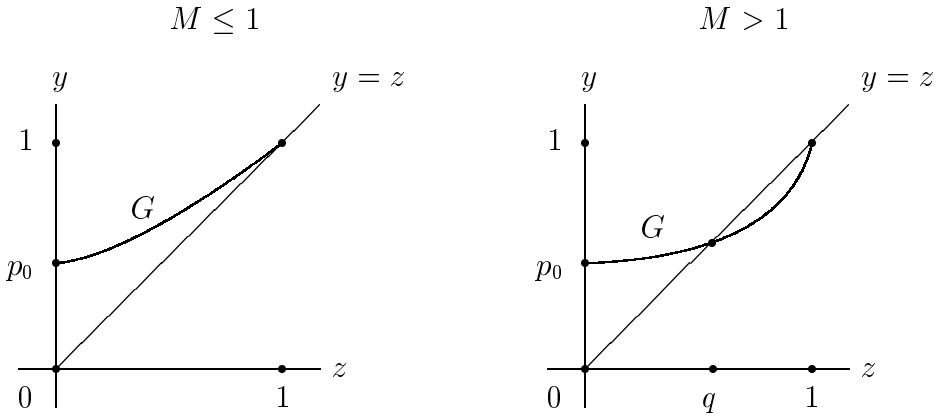
$$[z^l]G_{n+1}(z) = p_{n+1,l} = \sum_{k\ge 0} p_{n,k} \sum_{\dots} p_{l_1} p_{l_2} \cdots p_{l_k},$$

kde sčítáme přes všechny k-tice celých čísel $0 \le l_1, l_2, \dots, l_k$ splňující $l_1 + l_2 + \cdots + l_k = l$, a to je přesně $[z^l]G_n(G(z))$. GF $G_n(z)$ je tedy n-násobnou složeninou $G(G(\cdots G(z)\cdots))$.

Patrně $q_n = p_{n,0} = G_n(0)$. Pravděpodobnosti q_n tvoří rostoucí posloupnost: $q_{n-1} = G_{n-1}(0) < G_{n-1}(G(0)) = G_n(0) = q_n$. Limita $q = \lim q_n$ tedy existuje. Protože též $q_n = G(G_{n-1}(0)) = G(q_{n-1})$, limitní přechod vede na rovnici

$$q = G(q).$$

Limita q je pevným bodem funkce G. Jistě jím je 1, neboť $G(1) = \sum p_k = 1$. Ukážeme, že pro $M \le 1$ jiný pevný bod v $[0, 1]$ není a pro $M > 1$ je právě jeden další.



G má mocninný rozvoj s nezápornými koeficienty. Je proto, stejně jako její všechny derivace, v intervalu $[0,1]$ nezáporná. G je rostoucí a konvexní. Protože $G(0) > 0$ a $M = G'(1)$, pro $M \le 1$ se její graf přibližuje k přímce $y = z$ shora a protne ji až v $z = 1$. Tudíž $q_n = G(G(\cdots G(0)\cdots)) \to 1$.

Pro $M > 1$ se v levém okolí 1 přibližuje graf G k přímce $y = z$ zdola a někdy předtím ji v $z = q$ musí protnout. Průsečík je zjevně jen jeden. Z $z < q$ plyne $G(z) < G(q) = q$. Tudíž $q_n = G(G(\cdots G(0)\cdots)) \to q < 1$. □

Střední hodnota $M_n = \sum_{k=0}^{\infty} k p_{n,k}$ počtu potomků v n-té generaci splňuje $M_n = M^n$, neboť $M_n = G_n'(1)$ a $G_n'(1) = G_{n-1}'(G(1)) \cdot G'(1) = G_{n-1}'(1) \cdot G'(1) = M_{n-1} \cdot M$ a $M_1 = M$. Shrňme na závěr, co se děje pro jednotlivá M.

Nechť $M > 1$. Pak $q_n \to q < 1$ a M_n roste exponenciálně k nekonečnu. Protože $G_n(z) \to q$ pro každé $z \in [0,1)$, máme $p_{n,k} \to 0$ pro každé pevné $k > 0$ (to platí pro každé M), a tak

$$\Pr(\#\text{ potomků v } n\text{-té generaci je } > k \mid \text{ještě nevymřeli}) \to 1$$

pro každé pevné k a $n \to \infty$.

Pro $M < 1$ máme $q_n \to 1$ a M_n jde exponenciálně k nule. Pro $M = 1$ jde pravděpodobnost vymření rovněž k 1, ale střední hodnota počtu potomků je v každé generaci 1.

HÁZENÍ MINCÍ A ČEKÁNÍ NA SLOVO. Postupujeme dle Odlyzka v [8]. $A = a_1 a_2 \ldots a_k \in \{P,O\}^k$ je slovo délky $k, k > 0$, nad abecedou $\{P,O\}$ („panna nebo orel“). Házíme poctivou mincí (P i O padají s pravděpodobností 1/2) a zajímá nás, jak dlouho musíme v průměru čekat, než se v posloupnosti výsledků objeví A jako souvislé podslovo. Odpověď nalezneme pomocí GF.

Nechť

$$F_A(z) = \sum_{n\geq 0} f_A(n) z^n = 1 + \cdots,$$

kde $f_A(n)$ je počet slov z $\{P,O\}^n$ neobsahujících A, a

$$G_A(z) = \sum_{n\geq 1} g_A(n) z^n,$$

kde $g_A(n)$ počítá slova v $\{P,O\}^n$, která obsahují A na začátku, ale nikde jinde.

Veličinu $c_A(j)$ definujeme jako 1, pokud se počáteční úsek A délky $k-j$ shoduje s koncovým úsekem téže délky, a jako 0 jinak. Takže vždy $c_A(0) = 1$. *Korelační polynom* $C_A(z)$ je definován jako

$$C_A(z) = \sum_{j=0}^{k-1} c_A(j) z^j.$$

Například pro $A_0 = POPOPPOP$ máme $C_{A_0}(z) = 1 + z^5 + z^7$.

Odvodíme, že obě GF splňují soustavu

$$2zF_A = F_A - 1 + G_A \quad \text{a} \quad z^k F_A = C_A G_A.$$

Slovo $v \in \{O,P\}^n, A \not\subset v$, buď libovolné. První rovnice vyplývá z faktu, že po přidání O nebo P před v se A může vytvořit, ale jen na začátku. Druhá rovnice se dostane uvážením slov tvaru Av. Každé z nich má jednoznačný rozklad $Av = BAw$, kde Aw obsahuje A jen na začátku. (Vezmeme poslední výskyt A v Av.) Patrně je B počátečním úsekem A, $A = BC$, a tedy i $A = CD$. C je tedy shodným počátečním i koncovým úsekem A. Sumací přes C dostáváme druhou rovnici.

Soustava má řešení

$$F_A(z) = \frac{C_A(z)}{z^k + (1-2z)C_A(z)} \quad \text{a} \quad G_A(z) = \frac{z^k}{z^k + (1-2z)C_A(z)}.$$

Tvrzení. Střední čekací doba na A je $2^k C_A(1/2)$.

Důkaz. Jako p_n označíme pravděpodobnost, že se A objeví poprvé po n hodech, a q_n pravděpodobnost, že se A během n hodů neobjeví. Zřejmě

$p_n = q_{n-1} - q_n$ a $q_n = f_A(n)(\frac{1}{2})^n$. Proto

$$\begin{aligned} E(\#\text{ hodů, než se } A \text{ objeví}) &= \sum_{n\geq 1} np_n \\ &= \sum_{n\geq 1} n(q_{n-1} - q_n) = \sum_{n\geq 0} q_n \\ &= \sum_{n\geq 0} f_A(n)(1/2)^n = F_A(1/2). \\ &= 2^k C_A(1/2). \end{aligned}$$

□

Na $A_0 = POPOPPOP$ nutno v průměru čekat $2^8(1 + (\frac{1}{2})^5 + (\frac{1}{2})^7) = 266$ hodů.

10. přednáška 12.5.1999

VII. POUŽITÍ GF V TEORII ČÍSEL

V následujících pěti úlohách postupujeme podle Newmanovy knihy [14].

ROZKLAD NA ARITMETICKÉ POSLOUPNOSTI. $X \subset \mathbf{N}$ je (nekonečná) *aritmetická posloupnost*, má-li tvar $X = \{a, a+d, a+2d, a+3d, \ldots\}$, kde $a, d \in \mathbf{N}$. Konstanta $d > 0$ se nazývá *diference* X.

Tvrzení. Množinu přirozených čísel $\mathbf{N} = \{1, 2, \ldots\}$ nelze rozložit na disjunktní sjednocení (alespoň dvou, ale konečně mnoha) aritmetických posloupností se vzájemně různými diferencemi.

Důkaz. Řekněme, že to možné je. Tedy

$$\mathbf{N} = S_1 \cup S_2 \cup \cdots \cup S_l,$$

kde $S_i = \{a_i, a_i + d_i, a_i + 2d_i, \ldots\}$, $d_i \neq d_j$ a $S_i \cap S_j = \emptyset$ pro $i \neq j$ a $l \geq 2$. Řečeno GF,

$$\begin{aligned} \sum_{k\geq 1} z^k &= \sum_{k\in S_1} z^k + \sum_{k\in S_2} z^k + \cdots + \sum_{k\in S_l} z^k \\ \frac{z}{1-z} &= \frac{z^{a_1}}{1-z^{d_1}} + \frac{z^{a_2}}{1-z^{d_2}} + \cdots + \frac{z^{a_l}}{1-z^{d_l}}. \end{aligned}$$

Což je sporná rovnost: je-li d největší d_i, jde pro $z \to \mathrm{e}^{2\pi i/d}$ právě jeden její člen (v absolutní hodnotě) do nekonečna a ostatní mají konečné limity. □

DOKONALÉ PRAVÍTKO. Dokonalé pravítko je n-tice $0 \le a_1 < a_2 < \cdots < a_n$ celých čísel vyznačující se tím, že rozdíly $a_i - a_j$, $i > j$, probíhají všech $N = \binom{n}{2}$ hodnot $1, 2, \ldots, N$. Příkladem dokonalého pravítka je čtveřice $(0, 1, 4, 6)$ — na odměření vzdáleností $1, 2, \ldots, 6$ nám stačí jen uvedené čtyři rysky.

Tvrzení. Pro $n > 4$ dokonalé pravítko neexistuje.

Důkaz. Nechť $0 \le a_1 < a_2 < \cdots < a_n$ je dokonalé pravítko a $n > 4$. Pak GF

$$A(z) = \sum_{k=1}^{n} z^{a_k}$$

splňuje rovnici

$$A(z)A(1/z) = \sum_{k=-N}^{N} z^k + n - 1.$$

(Rozdíly $a_i - a_j$ dávají jednou každé z čísel $\pm 1, \pm 2, \ldots, \pm N$ a n krát číslo 0.) Protože

$$z^{-N} + z^{-N+1} + \cdots + z^N = \frac{z^{-N}(z^{2N+1} - 1)}{z - 1} = \frac{z^{N+1/2} - z^{-(N+1/2)}}{z^{1/2} - z^{-1/2}},$$

$\mathrm{e}^{i\varphi} = \cos\varphi + i \sin\varphi$ a $A(\mathrm{e}^{-i\theta})$ je číslo komplexně sdružené k $A(\mathrm{e}^{i\theta})$, dostáváme po dosazení $z = \mathrm{e}^{i\theta}$ do hořejší rovnosti nerovnost

$$0 \le |A(\mathrm{e}^{i\theta})|^2 = A(\mathrm{e}^{i\theta})A(\mathrm{e}^{-i\theta}) = \frac{\sin(N + 1/2)\theta}{\sin\theta/2} + n - 1.$$

Pro spor stačí nalézt θ tak, že poslední zlomek je menší než $-(n-1)$. Položíme $\theta = \frac{3\pi}{n^2-n+1}$. Pak $\sin(N + 1/2)\theta = \sin\frac{n^2-n+1}{2}\theta = -1$, $0 < \sin\theta/2 < \theta/2$ a pro $n \ge 5$ opravdu

$$\frac{\sin(N + 1/2)\theta}{\sin\theta/2} < -\frac{2}{\theta} = -\frac{2n^2 - 2n + 2}{3\pi} < -(n - 1),$$

protože $2n^2 - 2n + 2 - 3\pi(n-1) > 2n^2 - 2n + 2 - 10(n-1) = 2(n-3)^2 - 6 \ge 2 > 0$. □

Eulerova identita. *Číselným rozkladem* $n \in \mathbf{N}$ rozumíme rozklad n na součet přirozených sčítanců, přičemž na pořadí nezáleží. Například číslo 6 má celkem jedenáct rozkladů:

$$\begin{array}{llll} 6 & = 6 & = 3+3 & = 2+2+1+1 \\ & = 5+1 & = 3+2+1 & = 2+1+1+1+1 \\ & = 4+2 & = 3+1+1+1 & = 1+1+1+1+1+1. \\ & = 4+1+1 & = 2+2+2 & \end{array}$$

Z nich čtyři používají vzájemně různé sčítance ($6, 5+1, 4+2$ a $3+2+1$) a rovněž čtyři používají jen liché sčítance ($5+1, 3+3, 3+1+1+1$ a $1+1+1+1+1+1$). Euler dokázal, že to není náhoda.

Tvrzení. Pro každé $n \in \mathbf{N}$ se počet rozkladů r_n čísla n na různé sčítance rovná počtu rozkladů l_n čísla n na liché sčítance.

Důkaz. Dokážeme, že se GF

$$R = \sum_{n\geq 0} r_n x^n = 1 + x + \cdots \quad \text{a} \quad L = \sum_{n\geq 0} l_n x^n = 1 + x + \cdots$$

rovnají. Není složité si uvědomit, že

$$R = (1+x^1)(1+x^2)(1+x^3)\cdots \quad \text{a} \quad L = \frac{1}{(1-x^1)(1-x^3)(1-x^5)\cdots}.$$

Ovšem $1 + x^i = (1-x^{2i})/(1-x^i)$, a tak opravdu

$$R = \frac{\cancel{(1-x^2)}\cancel{(1-x^4)}\cancel{(1-x^6)}\cancel{(1-x^8)}\cdots}{(1-x^1)\cancel{(1-x^2)}(1-x^3)\cancel{(1-x^4)}\cdots} = L.$$

□

Rozměňování bankovky. Kolika způsoby se dá rozměnit bankovka hodnoty n korun na jedno-, dvou- a tříkorunové mince? Je-li a_n počet všech možných rozměnění, je GF čísel a_n dána formulí

$$\sum_{n\geq 0} a_n x^n = \frac{1}{(1-x)(1-x^2)(1-x^3)}.$$

Po chvilce počítání ověříme identitu

$$\frac{1}{(1-x)(1-x^2)(1-x^3)} = \frac{1/6}{(1-x)^3} + \frac{1/4}{(1-x)^2} + \frac{1/4}{1-x^2} + \frac{1/3}{1-x^3}.$$

Ale

$$\begin{aligned}\frac{1}{(1-x)^2} &= \frac{d}{dx}\left(\frac{1}{1-x}\right) = \sum_{n\geq 0}(n+1)x^n \\ \frac{1}{(1-x)^3} &= \frac{1}{2}\cdot\frac{d^2}{dx^2}\left(\frac{1}{1-x}\right) = \sum_{n\geq 0}\frac{(n+2)(n+1)}{2}x^n\end{aligned}$$

a zbývající dva členy jsou geometrické řady. Získáváme formuli

$$a_n = \frac{(n+2)(n+1)}{12} + \frac{n+1}{4} + \begin{cases} 1/4 & \text{pro } n \text{ sudé} \\ 1/3 & \text{pro } n \text{ dělitelné třemi,}\end{cases}$$

která se kompaktně zapíše jako

$$a_n = \left\lfloor \frac{n^2}{12} + \frac{n}{2} + 1 \right\rfloor.$$

Sčítací funkce není skoro konstantní. Pro $A \subset \mathbf{N}$ definujeme *sčítací funkci* $r_A(n)$ jako počet řešení rovnice

$$n = a + a',\ a \leq a',\ a, a' \in A.$$

Funkce definovaná na $\mathbf{N}$ je skoro konstantní, pokud je konstantní od určitého n_0 dále.

Tvrzení. Pro žádnou nekonečnou A není $r_A(n)$ skoro konstantní.

Důkaz. Sporem pomocí GF. Z GF

$$A(z) = \sum_{a\in A} z^a$$

množiny A snadno odvodíme GF sčítací funkce:

$$\sum_{n\geq 1} r_A(n) z^n = \tfrac{1}{2}(A(z)^2 + A(z^2)).$$

Byla-li by $r_A(n)$ skoro konstantní, měli bychom pro nějaké $c \in \mathbf{N}$ a polynom P s celočíselnými koeficienty rovnici

$$\tfrac{1}{2}(A(z)^2 + A(z^2)) = P(z) + \frac{c}{1-z}.$$

Ta je pro $z \to -1^+$ sporná. $A(z^2) \to \infty$, $A(z)^2 \geq 0$ a levá strana jde do nekonečna. Pravá však jde ke konečné limitě $P(-1) + c/2$. □

11. přednáška 19.5.1999

VIII. EXPONENCIÁLNÍ GF

V této přednášce postupujeme volně podle Stanleyho [22]. *Exponenciální GF* posloupnosti $\{a_n\}_{n\geq 0}$ je definována jako mocninná řada

$$\sum_{n\geq 0} \frac{a_n x^n}{n!}.$$

Proč jsou EGF užitečné? Protože se dobře chovají ke kombinatorickým konstrukcím.

SOUČINOVÁ FORMULE. Mějme dva typy struktur, $\mathcal{F}$ a $\mathcal{G}$, které jsou definovány na množině $[n] = \{1, 2, \ldots, n\}$. Jejich počty označíme jako f_n a g_n. Na $[n]$ definujeme novou strukturu $\mathcal{H}$: vezmeme uspořádanou dvojici množin (A, B), kde $A \cap B = \emptyset$ a $A \cup B = [n]$, na A definujeme $\mathcal{F}$-strukturu a na B (nezávisle na předchozí volbě) $\mathcal{G}$-strukturu. Počet těchto složených struktur označíme jako h_n. Nechť

$$F(x) = \sum_{n\geq 0} \frac{f_n x^n}{n!},\ G(x) = \sum_{n\geq 0} \frac{g_n x^n}{n!} \quad \text{a} \quad H(x) = \sum_{n\geq 0} \frac{h_n x^n}{n!}$$

jsou příslušné EGF. Pak platí *součinová formule*

$$H(x) = F(x)G(x).$$

Důkaz není složitý. Zřejmě

$$h_n = \sum_{k=0}^{n} \binom{n}{k} f_k g_{n-k},$$

protože $\binom{n}{k}$ je počet voleb dvojic (A, B) s $|A| = k$ a $f_k g_{n-k}$ je počet voleb $\mathcal{F}$-struktur a $\mathcal{G}$-struktur pro dané (A, B). Rovnici vydělíme $n!$ a máme

$$[x^n]H = \frac{h_n}{n!} = \sum_{k=0}^{n} \frac{f_k}{k!} \cdot \frac{g_{n-k}}{(n-k)!} = [x^n]FG.$$

KOMPOZIČNÍ FORMULE. $\mathcal{F}$, $\mathcal{G}$, f_n a g_n buďte jako výše. Z $\mathcal{F}$ a $\mathcal{G}$ opět budujeme složenou strukturu $\mathcal{H}$: vezmeme neuspořádaný rozklad $\{A_1, A_2, \ldots, A_k\}$ množiny $[n]$, to jest $A_i \neq \emptyset$, $A_i \cap A_j = \emptyset$ a $A_1 \cup A_2 \cup \cdots \cup A_k = [n]$, na $\{1, 2, \ldots, k\}$ definujeme $\mathcal{F}$-strukturu a na každé množině A_i definujeme $\mathcal{G}$-strukturu (všechny volby jsou nezávislé). Pomocí h_n opět označíme počet složených struktur. Jsou-li $F(x), G(x)$ a $H(x)$ příslušné EGF, platí *kompoziční formule*

$$H(x) = F(G(x)).$$

Pozor: nyní nutně $G(0) = g_0 = 0$. Důkaz je zase přímočarý. Zřejmě

$$h_n = \sum_{k=1}^{\infty} \frac{f_k}{k!} \sum_{\ldots} \binom{n}{m_1\ m_2\ \ldots\ m_k} g_{m_1} g_{m_2} \ldots g_{m_k},$$

kde ve vnitřní sumě sčítáme přes všechny k-tice přirozených čísel $(m_1, m_2, \ldots, m_k)$ splňující $m_1 + m_2 + \cdots + m_k = n$. Multinomický koeficient udává počet uspořádaných rozkladů $(A_1, A_2, \ldots, A_k)$ množiny $[n]$ na části s předepsanými mohutnostmi $|A_i| = m_i$. Musíme ho ještě vydělit $k!$, abychom dostali počet neuspořádaných rozkladů. Zbylé členy f_k a $g_{m_1} g_{m_2} \ldots g_{m_k}$ udávají počty voleb $\mathcal{F}$-struktur a $\mathcal{G}$-struktur. Po vydělení $n!$ máme

$$[x^n]H = \frac{h_n}{n!} = \sum_{k=1}^{\infty} \frac{f_k}{k!} \sum_{\ldots} \frac{g_{m_1}}{m_1!} \cdot \ldots \cdot \frac{g_{m_k}}{m_k!} = [x^n]F(G(x)).$$

BELLOVA ČÍSLA. Kolik je všech neuspořádaných rozkladů $[n]$ na neprázdné podmnožiny? Nechť jich je b_n. Například $b_3 = 5$:

$$123,\ 1|23,\ 2|13,\ 3|12 \text{ a } 1|2|3.$$

Pro nalezení EGF čísel b_n provedeme triviální kompoziční konstrukci. $\mathcal{F}$-strukturu definujeme jako „být množinou“, EGF pak je $\sum_{n\geq 0} 1x^n/n! = \mathrm{e}^x$ a $\mathcal{G}$-strukturu jako „být neprázdnou množinou“, její EGF je $\sum_{n\geq 1} 1x^n/n! = \mathrm{e}^x - 1$. Složená $\mathcal{H}$-struktura je zjevně strukturou rozkladů na neprázdné části. Podle kompoziční formule mají b_n EGF

$$\sum_{n\geq 0} \frac{b_n x^n}{n!} = \mathrm{e}^{\mathrm{e}^x - 1}.$$

Čísla

$$\{b_n\}_{n\geq 1} = \{1, 2, 5, 15, 52, 203, 877, 4140, 21147, \ldots\}$$

se nazývají *Bellovými čísly*. Byla pojmenována podle E. T. Bella (1883–1960), který napsal známý soubor medailonů matematiků *Men of Mathematics*. Pod pseudonymem psal též sci-fi novely.

Cayleyho formule. Známý kombinatorický drahokam: počet t_n označených (tj. izomorfismus nebereme v úvahu) stromů na množině $[n]$ se rovná n^{n-2}. Nyní se jedná nikoli o zakořeněné rovinné stromy, ale o všechny neorientované stromy. Místo t_n nalezneme počet z_n *zakořeněných stromů*, to jest stromů s jedním vyznačeným vrcholem; jejich strukturu označíme jako $\mathcal{Z}$. To stačí, neboť $z_n = nt_n$. Odvodíme rovnici pro EGF

$$Z(x) = \sum_{n\geq 1} \frac{z_n x^n}{n!}.$$

Vyhozením kořene se $\mathcal{Z}$-struktura rozpadne znovu na několik $\mathcal{Z}$-struktur (jejich kořeny jsou sousedé zmizelého kořene). Obecná $\mathcal{Z}$-struktura na $[n]$ se tedy dostane následující rekurzivní konstrukcí: nejprve vezmeme uspořádaný rozklad (A, B) množiny $[n]$. Na A zvolíme strukturu „být jednoprvkovou množinou“ (jejíž EGF je zjevně x) a na B provedeme kompoziční konstrukci s vnější strukturou „být množinou“ (EGF je e^x) a vnitřní strukturou rovnou $\mathcal{Z}$ (EGF je $Z(x)$). Podle součinové a kompoziční formule obdržíme rovnici

$$Z(x) = x\mathrm{e}^{Z(x)}.$$

Podle LIF

$$\frac{z_n}{n!} = [x^n]Z(x) = \tfrac{1}{n}[x^{n-1}](\mathrm{e}^x)^n = \frac{1}{n}\cdot\frac{n^{n-1}}{(n-1)!}$$

a $z_n = n^{n-1}$. Takže $t_n = n^{n-2}$.

2-regulární grafy. Označme d_n počet všech označených 2-regulárních grafů na $[n]$, to jest neorientovaných grafů bez smyček a paralelních hran, jejichž každý vrchol má stupeň 2; izomorfismus nebereme v úvahu (jakou úlohu dostaneme v opačném případě?). Pro kontrolu, $d_1 = d_2 = 0$, $d_3 = 1$ a $d_4 = 3$. EGF pro d_n získáme pomocí kompoziční formule. Vnější struktura je struktura „být množinou“ s EGF e^x a vnitřní struktura je struktura *souvislých* 2-regulárních grafů čili cyklů. Má EGF

$$G(x) = \sum_{n\geq 3} \frac{(n-1)!}{2}\cdot\frac{x^n}{n!} = \frac{1}{2}\sum_{n\geq 3}\frac{x^n}{n} = \frac{1}{2}\left(\log\frac{1}{1-x} - x - \frac{x^2}{2}\right),$$

neboť z $n!$ permutací $a_1a_2\dots a_n$ množiny $[n]$ jich vždy $2n$ určuje týž cyklus. (Liší-li se jen cyklickým pořadím nebo obrácením.) Dostáváme vzorec

$$\sum_{n\geq 3}\frac{d_n x^n}{n!} = \exp(G(x)) = \frac{\mathrm{e}^{-x/2-x^2/4}}{\sqrt{1-x}}.$$

Odtud se dá získat asymptotika. Za DOM CV odtud odvoďte rekurenci pro d_n. Použijte logaritmickou derivaci.

SOUVISLÉ GRAFY. Jako a_n označíme počet souvislých označených grafů na množině $[n]$. Například $a_1 = a_2 = 1$ a $a_3 = 4$:

Čísla a_n neznáme, ale známe počty b_n úplně všech gafů na $[n]$: $b_n = 2^{\binom{n}{2}}$. (Odpovídají podmnožinám množiny $\{E : E \subset [n] \ \&\ |E| = 2\}$.) Podle kompoziční formule je mezi EGF

$$A(x) = \sum_{n\geq 1}\frac{a_n x^n}{n!} \quad \text{a} \quad B(x) = \sum_{n\geq 1}\frac{b_n x^n}{n!}$$

vztah

$$B(x) = \exp(A(x)).$$

Aplikujeme-li na obě strany operátor $x\frac{d}{dx}\log$ (tj. logaritmická derivace násobená x), dostaneme vztah

$$xB' = xA'B.$$

Odtud dostáváme po úpravách rekurenci pro a_n:

$$a_n = 2^{\binom{n}{2}} - \frac{1}{n}\sum_{k=1}^{n-1} k a_k \binom{n}{k} 2^{\binom{n-k}{2}}.$$

Takže třeba

$$a_4 = 64 - \tfrac{1}{4}(1\cdot 1\cdot 4\cdot 8 + 2\cdot 1\cdot 6\cdot 2 + 3\cdot 4\cdot 4\cdot 1) = 38.$$

V MATEŘSKÉ ŠKOLCE. Děti, kterých je n, se rozdělí do skupinek. V každé z nich se všechny kromě jednoho vezmou za ruce a postaví do kroužku kolem

zbylého dítěte. Kroužek se může skládat i jen z jednoho děcka. Kolika způsoby se to dá udělat?

Kroužek s dítětem uprostřed se ze skupinky i dětí dá vytvořit $i(i-2)!$ způsoby (proč?). Pro EGF hledaných počtů a_n dostáváme podle kompoziční formule vzorec

$$\begin{aligned}\sum_{n\geq 2}\frac{a_n x^n}{n!} &= \exp\Big(\sum_{i\geq 2}\frac{i(i-2)!}{i!}x^i\Big)\\ &= \exp\Big(\sum_{i\geq 2}\frac{x^i}{i-1}\Big) = \exp(x\log 1/(1-x))\\ &= \Big(\frac{1}{1-x}\Big)^x.\end{aligned}$$

12. přednáška 26.5.1999

Hádanka: čemu se rovná

$$\sum_{n\geq 0}\frac{x^n}{n!}?$$

Každý ví, že e^x. A čemu se rovná

$$\sum_{n\geq 0}\frac{n^x}{n!}?$$

Pro $x\in\mathbf{N}$ se rovná $\mathrm{e}b_x$. Pro Bellova čísla tak platí

$$b_n = \frac{1}{\mathrm{e}}\sum_{m\geq 0}\frac{m^n}{m!}.$$

Tuto tzv. Dobinského formuli ponecháme bez důkazu. Lze jej nalézt například v Lovászově cvičebnici [12].

Jiná zajímavá reprezentace b_n pochází od Flajoleta [6]:

$$\sum_{n=0}^{\infty} b_n x^n = \cfrac{1}{1-x-\cfrac{x^2}{1-2x-\cfrac{2x^2}{1-3x-\cfrac{3x^2}{\cdots}}}}.$$

Ani toto vyjádření GF Bellových čísel řetězovým zlomkem nebudeme dokazovat.

Není však těžké nahlédnout rekurenci ($n > 0$)

$$b_n = \sum_{k=0}^{n-1} \binom{n-1}{k} b_{n-1-k} = \sum_{k=0}^{n-1} \binom{n-1}{n-k-1} b_{n-1-k} = \sum_{k=0}^{n-1} \binom{n-1}{k} b_k.$$

V rozkladech $[n]$ uvážíme blok X obsahující číslo 1. Množinu $X\backslash\{1\}$ s $|X| = k+1$ můžeme volit $\binom{n-1}{k}$ způsoby a nezávisle na nich b_{n-1-k} způsoby rozklad $\{2,3,\ldots,n\}\backslash X$.

Větička. GF Bellových čísel splňuje vztahy

$$\begin{aligned} B(x) = \sum_{n\geq 0} b_n x^n &= 1 + \tfrac{x}{1-x} B(\tfrac{x}{1-x}) \\ &= \sum_{k\geq 0} \frac{x^k}{(1-x)(1-2x)\cdots(1-kx)}. \end{aligned}$$

Důkaz. Obě formulace jsou jednoduše ekvivalentní. Sčítanec

$$\frac{x^k}{(1-x)(1-2x)\cdots(1-kx)}$$

totiž substitucí $x := x/(1-x)$ přejde na $x^k/(1-2x)(1-3x)\cdots(1-(k+1)x)$. Substitucí, vynásobením $x/(1-x)$ a přičtením 1 se proto nekonečný součet nemění a $B(x)$ splňuje funkcionální rovnici. Na druhou stranu jejím iterováním dostáváme pro $B(x)$ náš nekonečný součet.

1. Důkaz pomocí GF. Využijeme poslední rekurenci a binomickou větu pro záporný celočíselný exponent.

$$\begin{aligned} B(x) - 1 = \sum_{n\geq 1} b_n x^n &= \sum_{n\geq 1} x^n \sum_{k=0}^{n-1} \binom{n-1}{k} b_k \\ &= \sum_{k\geq 0} b_k \sum_{n>k} \binom{n-1}{k} x^n \\ &= \sum_{k\geq 0} b_k x^{k+1} \sum_{n>k} \binom{n-1}{n-k-1} x^{n-k-1} \\ &= \sum_{k\geq 0} b_k x^{k+1} \sum_{m\geq 0} \binom{k+1+m-1}{m} x^m \end{aligned}$$

$$
\begin{aligned}
&= \sum_{k\geq 0} b_k x^{k+1}(1-x)^{-k-1} \\
&= \frac{x}{1-x}\sum_{k\geq 0} b_k \left(\frac{x}{1-x}\right)^k \\
&= \tfrac{x}{1-x}B(\tfrac{x}{1-x}).
\end{aligned}
$$

2. Bijektivní důkaz. Písmenem $\mathcal{N}$ označíme množinu všech *normálních* slov, to jest konečných slov u nad abecedou $\mathbf{N} = \{1, 2, \ldots\}$, která mají tyto dvě vlastnosti : (i) v u jsou použita právě čísla $1, 2, \ldots, n$ pro nějaké $n \in \mathbf{N}$ a (ii) pro každá dvě čísla $1 \leq i < j \leq n$ první výskyt i v u předchází první výskyt j. (Normální slova by nám měla být povědomá z přednášky o Schröderových číslech.) Množina všech rozkladů $[l]$, kde l probíhá $\mathbf{N}$, je zjevně v bijekci s $\mathcal{N}$. Normální slovo $u = a_1 a_2 \ldots a_l$ kóduje rozklad $[l]/\sim$, kde relace ekvivalence $\sim$ je dána vztahem $i \sim j \Leftrightarrow a_i = a_j$, a každý rozklad $[l]$ je kódován právě jedním normálním slovem délky l. Proto, označuje-li $|u|$ délku u, platí

$$B(x) = \sum_{n\geq 0} b_n x^n = \sum_{u\in\mathcal{N}} x^{|u|}.$$

Je-li $u = a_1 a_2 \ldots a_l$ normální slovo, *nafouknutím* u rozumíme každé slovo tvaru

$$00\ldots0a_1 0\ldots0a_2 0\ldots0a_3\ldots a_l 0\ldots0,$$

kde první úsek nul před a_1 obsahuje alespoň jednu nulu a ostatní úseky nul mohou být i prázdné. Slova, která takto z u vzniknou, počítá podle délek GF

$$\frac{x}{1-x}\cdot\left(\frac{1}{1-x}\right)^l \cdot x^l.$$

Pro množinu $\mathcal{N}'$ všech nafouknutí všech slov $u \in \mathcal{N}$ tak platí

$$\sum_{u\in\mathcal{N}'} x^{|u|} = \frac{x}{1-x}\sum_{u\in\mathcal{N}}\left(\frac{x}{1-x}\right)^{|u|} = \tfrac{x}{1-x}B(\tfrac{x}{1-x}).$$

Na druhou stranu se ale nafouknutím vlastně skoro nic nezměnilo,

$$\sum_{u\in\mathcal{N}'} x^{|u|} = \sum_{u\in\mathcal{N}} x^{|u|} - 1 = B(x) - 1,$$

protože $\mathcal{N}'$ se skládá z neprázdných normálních slov nad abecedou $\{0, 1, 2, \ldots\}$. Takže $B(x) - 1 = \frac{x}{1-x}B(\frac{x}{1-x})$. □

Nechť $S(n,k)$ označuje počet rozkladů $[n]$ na právě k neprázdných bloků. Číslům $S(n,k)$ se říká *Stirlingova čísla druhého druhu*. (Stirlingova čísla prvního druhu střídají znaménko a počítají permutace $[n]$ s daným počtem cyklů.)

Pozorování.

$$\sum_{n\geq 0} S(n,k)x^n = \frac{x^k}{(1-x)(1-2x)\cdots(1-kx)}.$$

Důkaz. Plyne po chvilce zamyšlení nad strukturou normálních slov. $S(n,k)$ je právě počet $u \in \mathcal{N}$ délky n s k symboly. Takové u se dá rozložit na

$$u = 1u_1 2u_2 \ldots k u_k,$$

kde u_i je slovo (i prázdné a ne nutně normální) nad abecedou $\{1,2,\ldots,i\}$. Slova u_i můžeme takto volit libovolně a nezávisle na sobě. Počet u_i délky l je i^l. Podle délek je počítá GF

$$\frac{1}{1-ix}.$$

Normální slova s k symboly tak podle délek počítá GF

$$x^k \prod_{i=1}^{k} \frac{1}{1-ix} = \frac{x^k}{(1-x)(1-2x)\cdots(1-kx)}.$$

□

Formule vyjadřující $B(x)$ nekonečným součtem tak pouze zachycuje rozdělení třídy všech rozkladů na podtřídy podle počtu bloků.

Přestože má $B(x)$ nulový poloměr konvergence a je pouze mocninnou řadou, leckdy se hodí. Ukážeme si dvě její použití.

ŘÍDKÉ ROZKLADY. *Řídkými* rozklady $[n]$ rozumíme rozklady, v nichž žádná dvě po sobě jdoucí čísla $i, i+1$ neleží ve stejném bloku. Jejich počet označíme r_n. Například $r_4 = 5$:

$$1|2|3|4,\ 13|24,\ 1|3|24,\ 13|2|4 \text{ a } 14|2|3.$$

Následující výsledek (i zesílení, které neuvádíme) dokázal Yang [29].

Větička. Pro každé $n \in \mathbf{N}$ platí $r_n = b_{n-1}$.

Důkaz. Uvažme množinu $\mathcal{R} \subset \mathcal{N}$ normálních slov kódujících řídké rozklady. Normální slovo u padne do $\mathcal{R}$, právě když v něm není žádné bezprostřední opakování. *Nadutím* $u = a_1 a_2 \ldots a_l \in \mathcal{R}$ rozumíme každé slovo tvaru

$$a_1 a_1 \ldots a_1 a_2 a_2 \ldots a_2 \ldots a_l a_l \ldots a_l,$$

kde $a_i a_i \ldots a_i$ je libovolné slovo nad $\{a_i\}$ délky alespoň jedna. Množina všech nadutí všech $u \in \mathcal{R}$ je právě $\mathcal{N}$. Slova vzniklá nadutím pevného $u \in \mathcal{R}$ délky l jsou podle délek počítána GF

$$\left(\frac{x}{1-x}\right)^l.$$

Takže

$$B(x) = \sum_{u \in \mathcal{N}} x^{|u|} = \sum_{u \in \mathcal{R}} \left(\frac{x}{1-x}\right)^{|u|} = \sum_{n \geq 0} r_n \left(\frac{x}{1-x}\right)^n.$$

Označíme-li GF čísel r_n jako $R(x)$, dostáváme odtud s pomocí identity pro $B(x)$ vztah

$$1 + \tfrac{x}{1-x} B(\tfrac{x}{1-x}) = B(x) = R(\tfrac{x}{1-x}).$$

Substitucí $x := x/(1+x))$ ho převedeme na $1 + xB(x) = R(x)$. Opravdu $r_n = b_{n-1}$. □

Periodičnost zbytků Bellových čísel. V první přednášce jsme prozkoumali paritu Catalanových čísel a zjistili jsme, že jejich zbytky modulo 2 netvoří periodickou posloupnost. (Periodickou posloupností rozumíme posloupnost, v níž se od jistého členu opakuje jeden konečný úsek.) Bellova čísla se na rozdíl od Catalanových chovají v tomto ohledu spořádaně.

Tvrzení. Pro každé $m \in \mathbf{N}$ je posloupnost $\{b_n \bmod m\}_{n \geq 0}$ periodická.

Důkaz. Pro $m \in \mathbf{N}$ a mocninné řady $S, T \in \mathbf{Z}[[x]]$ pomocí $S \equiv T \bmod m$ označíme kongruenci po koeficientech, to jest $[x^n]S \equiv [x^n]T \bmod m$ pro každé $n \in \mathbf{N}_0$. Je očividné, že z $S_1 \equiv T_1$ a $S_2 \equiv T_2$ plyne $S_1 S_2 \equiv T_1 T_2$ a $S_1 + S_2 \equiv T_1 + T_2$. Takže (kongruence jsou vždy modulo m) se $B(x)$ rovná

$$\begin{aligned} & \sum_{k \geq 0} \frac{x^k}{(1-x)(1-2x)\cdots(1-kx)} \\ \equiv \; & \sum_{l=0}^{m-1} \frac{x^l}{(1-x)(1-2x)\cdots(1-lx)} \sum_{r \geq 0} \left(\frac{x^m}{(1-x)(1-2x)\cdots(1-mx)}\right)^r \end{aligned}$$

$$= \frac{1}{1-x^m/Q(x)} \sum_{l=0}^{m-1} \frac{1}{Q_l(x)},$$

kde $Q_l(x) = (1-x)(1-2x)\cdots(1-lx)$ a $Q(x) = (1-x)(1-2x)\cdots(1-mx)$. Celkově

$$B(x) \equiv \frac{S(x)}{T(x)},$$

kde $S(x), T(x) \in \mathbf{Z}[x]$ a $T(0) = 1$. Existuje tedy posloupnost celých čísel $\{v_n\}_{n\geq 0}$ taková, že $b_n \equiv v_n$ a $\{v_n\}_{n\geq 0}$ má racionální GF $\frac{S(x)}{T(x)}$. Nechť $T(x) = t_k x^k + t_{k-1}x^{k-1} + \cdots + 1$, $t_i \in \mathbf{Z}$. Ze vztahu

$$T(x) \sum_{n\geq 0} v_n x^n = S(x)$$

plyne pro $n > \deg S(x)$ rekurence

$$v_n + t_1 v_{n-1} + \cdots + t_k v_{n-k} = 0.$$

Čísla v_n splňují lineární rekurenci s konstantními koeficienty a posloupnost jejich zbytků modulo m (vlastně modulo jakékoli číslo) je nutně periodická (proč přesně?). Díky $b_n \equiv v_n$ totéž platí i pro Bellova čísla. □

Reference

[1] D. André, Solution directe de problème résolu par M. Bertrand, *C. R. Acad. Sci. Paris* **105** (1887), 436–437.

[2] L. Comtet, *Advanced Combinatorics*, D. Reidel, Dordrecht, 1974.

[3] Discrete Mathematics & Theoretical Computer Science, `http://dmtcs.loria.fr/`

[4] R. Donaghey and L.W. Shapiro, Motzkin numbers, *Journal of Combinatorial Theory A* **23** (1977), 291–301.

[5] The Elektronic Journal of Combinatorics, `http://www.combinatorics.org/ejc-wce.html`

[6] P. Flajolet, Combinatorial aspects of continued fractions, *Discrete Mathematics* **32** (1980), 125–161.

[7] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, John Wiley & Sons, New York, 1983.

[8] R. L. Graham, M. Grötschel, and L. Lovász (eds.), *Handbook of Combinatorics*, Elsevier, Amsterdam, 1995.

[9] M. Klazar, Twelve countings with rooted plane trees, *European Journal of Combinatorics* **18** (1997), 195–210.

[10] D. E. Knuth, *The Art of Computer Programming, Volumes 1–3*, Addison-Wesley, Reading, 1997, 1998, 1998. [Poslední vydání.]

[11] J. Levine, Note on the number of pairs of non-intersecting routes, *Scripta Mathematica* **24** (1959), 335–338.

[12] L. Lovász, *Combinatorial Problems and Exercises*, North Holland, Amsterdam, 1993. [Poslední vydání.]

[13] Th. Motzkin, Relations between hypersurfaces cross ratios, and a combinatorial formula for partitions of a polygon, for a permanent preponderance and for nonassociative products, *Bulletin American Mathematical Society* **54** (1948), 352–360.

[14] D. J. Newman, *Analytic Number Theory*, Springer Verlag, New York, 1998.

[15] G. Pólya, On the number of certain lattice polygons, *Journal of Combinatorial Theory* **6** (1969), 102–105.

[16] A. Rényi, *Teorie pravděpodobnosti*, Academia, Praha, 1972.

[17] D. Rubenstein, Catalan numbers revisited, *Journal of Combinatorial Theory A* **68** (1994), 486–490.

[18] J. M. Ruiz, *The Basic Theory of Power Series*, Friedr. Viewegh & Sohn, Braunschweig, 1993.

[19] E. Schröder, Vier combinatorische Probleme, *Zeitschrift für Mathematik und Physik* **15** (1870), 361–376.

[20] N. J. A. Sloane, *A Handbook of Integer Sequences*, Academic Press, New York and London, 1973.

[21] R. P. Stanley, *Enumerative Combinatorics, Volume 1*, Wadsworth & Brooks/Cole, Monterey CA, 1986.

[22] R. P. Stanley, *Enumerative Combinatorics, Volume 2*, Cambridge University Press, Cambridge UK, 1999.

[23] G. Tenenbaum, *Introduction to Analytic and Probabilistic Number Theory*, Cambridge University Press, Cambridge UK, 1995.

[24] P. Valtr, Probability that n random points are in a convex position, *Discrete and Computational Geometry* **13** (1995), 637–643.

[25] P. Valtr, Catalan numbers via random planar point sets, 441–443. In: I. Bárány (ed.), *Intuitive Geometry*, Bolyai Society Mathematical Studies 6 (1997).

[26] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, Cambridge University Press, Cambridge UK, 1992.

[27] H. Wilf, *Generatingfunctionology*, Academic Press, San Diego CA, 1994.

[28] W.-J. Woan, L. Shapiro and D. G. Rogers, The Catalan numbers, the Lebesgue integral, and 4^{n-2}, *American Mathematical Monthly* **104** (1997), 926–931.

[29] W. Yang, Bell numbers and k-trees, *Discrete Mathematics* **156** (1996), 247–252.

The Mathematical Knight

Noam D. Elkies
Richard P. Stanley

Introduction

Much has been said of the affinity between mathematics and chess: two domains of human thought where very limited sets of rules yield inexhaustible depths, challenges, frustrations and beauty. Both fields support a venerable and burgeoning technical literature and attract much more than their share of child prodigies. For all that, the intersection of the two domains is not large. While chess and mathematics may favor similar mindsets, there are few places where a chess player or analyst can benefit from a specific mathematical idea, such as the symmetry of the board and of most pieces' moves (see for instance [24]) or the combinatorial game theory of Berlekamp, Conway, and Guy (as in [4]). Still, when mathematics does find applications in chess, striking and instructive results often arise.

This two-part article shows several such applications that feature the knight and its characteristic $(2,1)$ leap. It is based on portions of a book tentatively entitled *Chess and Mathematics*, currently in preparation by the two authors of this article, that will cover all aspects of the interactions between chess and mathematics. Mathematically, the choice of $(2,1)$ and of the 8×8 board may seem to be a special case of no particular interest, and indeed we shall on occasion indicate variations and generalizations involving other leap parameters and board sizes. But long experience points to the standard knight's move and chessboard size as felicitous choices not only for the game of chess but also for puzzles and problems involving the board and pieces, including several of our examples.

This first part concentrates on puzzles such as the knight's tour. Many of these are clearly mathematical problems in a very thin disguise (for instance, a closed knight's tour is a Hamiltonian circuit on a certain graph $\mathcal{G}$), and can be solved or at least better understood using the terminology and techniques of combinatorics. We also relate a few of these ideas with practical endgame technique (see Diagrams 1ff., 10, 11). The second part shows some remarkable chess problems featuring the knight or knights. Most "practical" chess players have little patience for the art of chess problems, which has evolved a long way from its origins in instructive exercises. But the same formal concerns that may deter the over-the-board player give some problems a particular appeal to mathematicians. For instance, we will exhibit a position, constructed by P. O'Shea and published in 1989, where White, with only king and knight, has just one way to force mate in 48 (the current record). We also show the longest known legal game of chess that is determined completely by its last move (discovered by Rösler in 1994) — which happens to be checkmate by promotion to a knight.

Algebraic notation.
We assume that the reader is familiar with the rules of chess, but require very little knowledge of chess strategy. (The reader who knows, or is willing to accept as intuitively obvious, that king and queen win against king or even king and knight if there is no immediate draw, will have no difficulty following the analysis.) The reader will, however, have to follow the notation for chess moves, either by visualizing the moves on the diagram or by setting up the position on the board. Several notation systems have been used; the most common one nowadays, and the one we use here, is "algebraic notation", so called because of the coordinate system used to name the squares of the board. In the remaining paragraphs of this introductory section we outline this notation system. Readers already fluent in algebraic notation may safely skip ahead to Section 1.

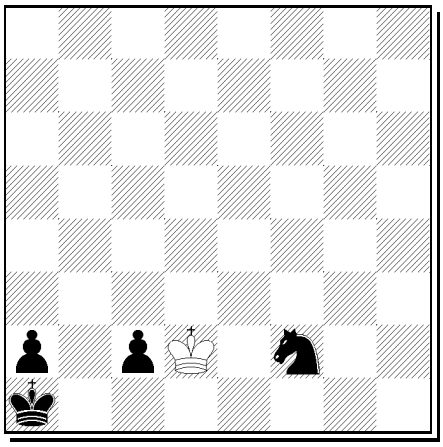
Each square on the 8×8 board is uniquely determined by its row and column, called "rank" and "file" respectively. The ranks are numbered from 1 to 8, the files named by letters a through h. In the initial array, ranks 1 and 2 are occupied by White's pieces and pawns, ranks 8 and 7 by Black's, both queens are on the d-file, and both kings on the e-file. Thus, viewed from White's side of the board (as are all the diagrams in this article), the ranks are numbered from bottom to top, the files from left to right. We name a square by its column followed by the row; for instance, the White king in Diagram 1 below is at d2. Each of the six kinds of chessmen is referred to by a single letter, usually its initial: K, Q, R, B, P are king, queen, rook, bishop, and pawn (often lower-case p is seen for pawn). We cannot use the initial letter for the knight because K is already the king, so we use its phonetic initial, N for kNight. For instance, Diagram 1 can be described as: White Kd2, Black Ka1, Nf2, Pa2, Pc2. To notate a chess move we name the piece and its destination square, interpolating "×" if the move is a capture. For pawn moves the P is usually suppressed; for pawn captures, it is replaced by the pawn's file. Thus in Diagram 11, Black's pawn moves are notated a2 and a×b2 rather than Pa2 and P×b2. We follow a move by "+" if it gives check, and by "!" or "?" if we regard it as particularly strong or weak. In some cases "!" is used to indicate a thematic move, i.e., a move that is essential to the "theme" or main point of the problem. As an aid to following the analysis, moves are numbered consecutively, from the start of the game or from the diagram. For instance, we shall begin the discussion of Diagram 1 by considering the possibility "1.K×c2 Nd3!". Here "1" indicates that these are White's and Black's first moves from the diagram; "K×c2" means that the White king captures the unit on c2; and "Nd3!" means that the Black knight moves to the unoccupied square d3, and that this is regarded as a strong move (the point here being that Black prevents 2.Kc1 even at the cost of letting White capture the knight). When analysis begins with a Black move, we use "..." to represent the previous White move; thus "1 ... Nd3!" is the same first Black move.

A few further refinements are needed to subsume promotion and castling, and to ensure that every move is uniquely specified by its notation. For instance, if Black were to move first in Diagram 1 and promoted his c2-pawn to a queen (giving check), we would write this as 1 ... c1Q+, or more likely 1 ... c1Q+?, because we shall see that after 2.K×c1 White can draw. Short and long castling are notated 0-0 and 0-0-0 respectively. If the piece and destination square do not specify the move uniquely, we also give the departure square's file, rank, or both. An extreme example: Starting from Diagram 9, "Nb1" uniquely specifies a move of the c3 knight. But to move it to d5 we would write "Ncd5" (because other knights on the b- and f-files could also reach d5); to a4, "N3a4" (not "Nca4" because of the knight on c5); and to e4, "Nc3e4" (why?).

1 A chess endgame

We begin by analyzing a relatively simple chess position (Diagram 1 below). This may look like an endgame from actual play, but is a composed position — an "endgame study" — created (by NDE) to bring the key point into sharper focus.

Diagram 1



White to move

White, reduced to bare king, can do no better than draw, and even that with difficulty: Black will surely win if either pawn safely promotes to a queen. A natural try is 1.K×c2, eliminating one pawn and imprisoning two of Black's remaining three men in the corner. But 1 . . . Nd3! breaks the blockade (Diagram 2a). Black threatens nothing but controls the key square c1. The rules of chess do not allow White to pass the move; unable to go to c1, the king must move elsewhere and release Black's men. After 2.K×d3 (or any other move) Kb1 followed by 3 . . . a1Q, Black wins easily.

Diagram 2a

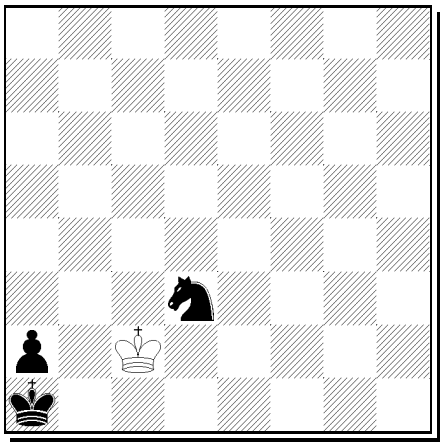
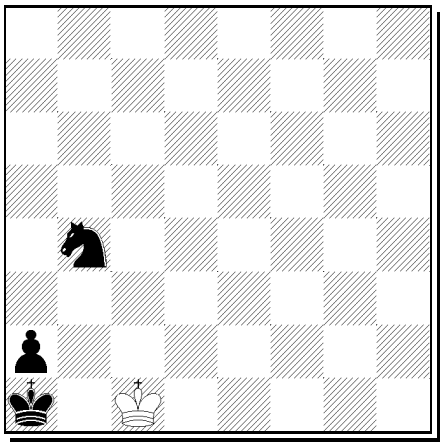


Diagram 2b



Returning to Diagram 1, let us try instead 1.Kc1! This still locks in the Black Ka1 and Pa2, and prepares to capture the Pc2 next move, for instance 1 . . . Nd3+ 2.K×c2, arriving at Diagram 2a with Black to move. White has in effect succeeded in passing the move to Black by taking a detour from d2 to c2. Now it is Black who cannot pass, and any move restores the White king's access to c1. For instance, play may continue 2 . . . Nb4+ 3.Kc1, reaching Diagram 2b. Black is still bottled up. If it were White to move in Diagram 2b, White would have to release Black with Kd1 or Kd2 and lose; but again Black must move and allow White back to c2, for instance 3 . . . Nd3+ 4.Kc2 and we are back at Diagram 2a.

So White does draw — at least if Black obligingly shuttles the knight between d3 and b4 to match the White king's oscillations between c1 and c2. But what if Black tries to improve on this? While the king is limited to those two squares, the knight can roam over almost the entire board. For instance, from Diagram 2a Black might bring the knight to the far corner in m moves, reaching a position such as Diagram 3a, and then back to d3 in n moves. If $m + n$ is odd, then Black will win since it will be White's turn to move. Instead of d3, Black can aim for b3 or e2, which also control c1; but each of these is two knight moves away from d3, so we get an equivalent parity condition. Alternatively, Black might try to reach b4 from d3 in an *even* number of moves, to reach Diagram 2b with White to move; and again Black could aim for another square that controls c2. But each of these squares is one or three knight moves away from d3, so again would yield a closed path of odd length through d3.

Can Black thus pass the move back to White? For that matter, what should White do in Diagram 3b? Does either Kc1 or K×c2 draw, or is White lost regardless of this choice?

Diagram 3a

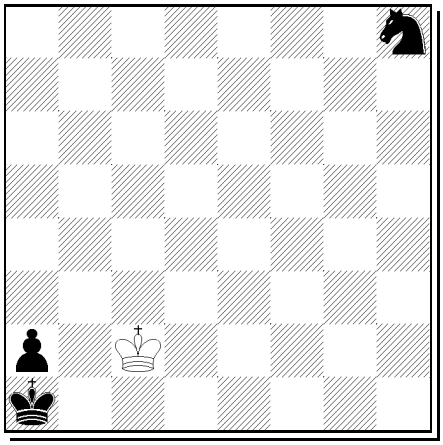
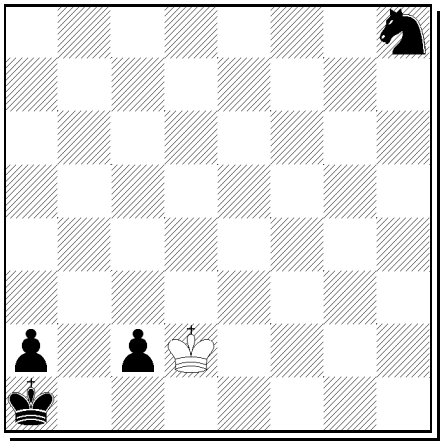


Diagram 3b



White to move

The outcome of Diagram 2a thus hinges on the answer to the following problem in graph theory:

Let $\mathcal{G} = \mathcal{G}_{8,8}$ be the graph whose vertices are the 64 squares of the 8×8 chessboard and whose edges are the pairs of squares joined by a knight's move. Does $\mathcal{G}$ have a cycle of odd length through d3?

Likewise White's initial move in Diagram 3b and the outcome of this endgame comes down to the related question concerning the same graph $\mathcal{G}$:

What are the possible parities of lengths of paths on $\mathcal{G}$ from h8 to c1 or c2?

The answers result from the following basic properties of $\mathcal{G}$:

Lemma. *(i) The graph $\mathcal{G}$ is connected. (ii) The graph is bipartite, the two parts comprising the 32 light squares and 32 dark squares of the chessboard.*

Proof: Part (i) is just the familiar fact that a knight can get from any square on the chessboard to any other square. Part (ii) amounts to the observation that every knight move connects a light and a dark square.

Corollaries. 1) There are no knight cycles of odd length on the chessboard. 2) Two squares

of the same color are connected by knight-move paths of even length but not of odd length; two square of opposite color are connected by knight-move paths of odd length but not of even length.

We thus answer our chess questions: White draws both Diagram 1 and Diagram 3b by starting with Kc1. More generally, for any initial position of the Black knight, White chooses between c1 and c2 by moving to the square of the same color as the one occupied by the knight.

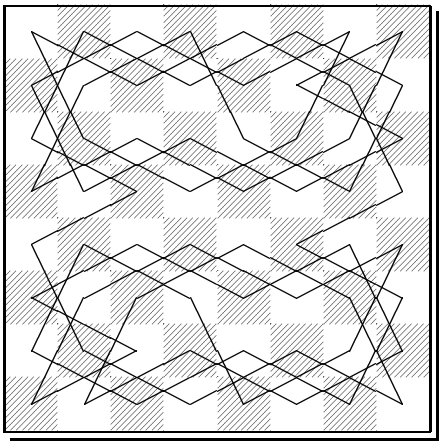
REMARK. Our analysis would reach the same conclusions if the Black pawn on c2 were removed from Diagrams 1 and 3b; we included this superfluous pawn only as bait to make the wrong choice of c2 more tempting.

Puzzle 1. For which rectangular boards (if any) does part (i) or (ii) of the Lemma fail? That is, which $\mathcal{G}_{m,n}$ are not connected, or not bipartite? (All puzzles and all diagrams not explicated in the text have solutions at the end of this article.)

Knight's tours and the Thirty-Two Knights

The graph $\mathcal{G}$ arises often in problems and puzzles involving knights. For instance, the perennial knight's tour puzzle asks in effect for a Hamiltonian path on $\mathcal{G}$; a "re-entrant" or "closed" knight's tour is just a Hamiltonian circuit. The existence of such tours is classical — even Euler spent some time constructing them, finding among others the following elegant centrally symmetric tour (from [9, p. 191]):

Diagram 4



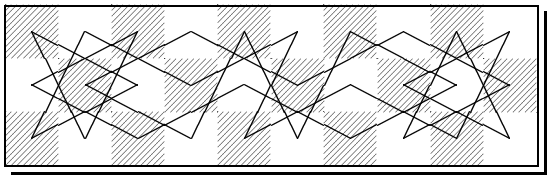
a closed knight's tour constructed by Euler

The extensive literature on knight's tours includes many examples, which, when numbered along the path from 1 to 64, yield semi-magic squares (all row and column sums equal 260), sometimes with further "magic" properties, but it is not yet known whether a fully magic knight's tour (one with major diagonals as well as rows and columns summing to 260), either open or closed, can exist.

More generally, we may ask for Hamiltonian circuits on $\mathcal{G}_{m,n}$ for other m, n; that is, for closed knight's tours on other rectangular chessboards. A necessary condition is that $\mathcal{G}_{m,n}$ be a connected graph with an even number of vertices. Hence we must have $2|mn$ and both m, n at least 3 (cf. Puzzle 1). But not all $\mathcal{G}_{m,n}$ satisfying this condition admit Hamiltonian circuits. For instance, one easily checks that $\mathcal{G}_{3,4}$ is not Hamiltonian. Nor are $\mathcal{G}_{3,6}$ and $\mathcal{G}_{3,8}$,

but $\mathcal{G}_{3,10}$ has a Hamiltonian circuit, as does $G_{3,n}$ for each even $n > 10$. For instance, the next diagram shows a closed knight's tour on the 3×10 board:

Diagram 5



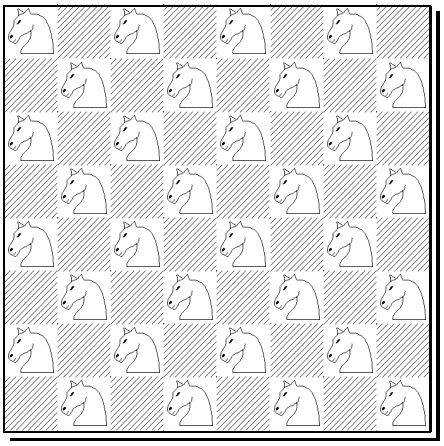
a closed knight's tour on the 3×10 board

There are sixteen such tours (ignoring the board symmetries). More generally, enumerating the closed knight's tours on a $3 \times (8 + 2n)$ board yields a sequence 16, 176, 1536, 15424, ... satisfying a constant linear recursion of degree 21 that was obtained independently by Knuth and NDE in April, 1994. See [23, Sequence A070030]. In 1997, Brendan McKay first computed that there are 13267364410532 (more than 1.3×10^{13}) closed knight's tours on the 8×8 board ([19]; see also [23, Sequence A001230],[26]).

We return now from enumeration to existence. After $\mathcal{G}_{3,n}$ the next case is $\mathcal{G}_{4,n}$. This is trickier: the reader might try to construct a closed knight's tour on a 4×11 board, or to prove that none exists. We answer this question later.

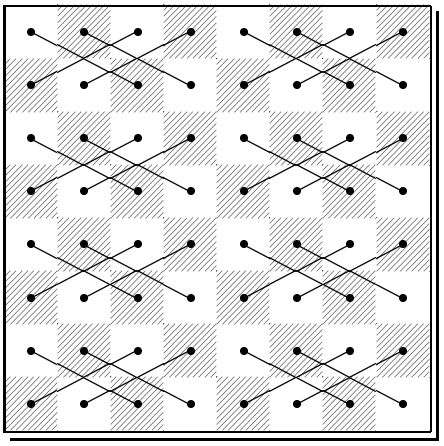
What of maximal cliques and cocliques on $\mathcal{G}$? A clique is just a collection of pairwise defending (or attacking) knights. Clearly there can be no more than two knights, again because $\mathcal{G}$ is bipartite: two squares of the same color cannot be a knight's move apart, and any set of more at least three squares must include two of the same color. Cocliques are more interesting: how many pairwise *non*attacking knights can the chessboard accommodate?[1] We follow Golomb ([21], via M. Gardner [9, p. 193]). Again the fact that $\mathcal{G}$ is bipartite suggests the answer (Diagram 6):

Diagram 6



32 mutually nonattacking knights

Diagram 7



A one-factor in $\mathcal{G}$

It is not hard to see that we cannot do better: the 64 squares may be partitioned into 32 pairs each related by a knight move, and then at most one square from each pair can be

[1] Burt Hochberg jokes (in [11, p. 5], concerning the analogous problem for queens) that the answer is 64, all White pieces or all Black: pieces of the same color cannot attack each other! Of course this joke, and similar jokes such as crowding several pieces on a single square, are extraneous to our analysis.

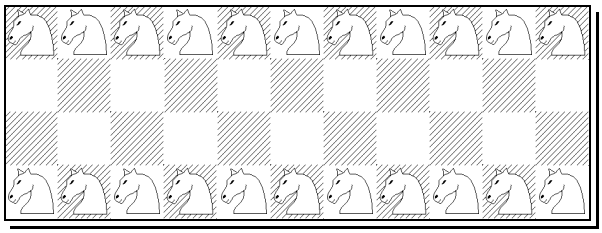
used. See Diagram 7. This is Patenaude's solution in [21]. Such a pairing of $\mathcal{G}$ is called a "one-factor" in graph theory. Similar one-factors exist on all $\mathcal{G}_{m,n}$ when $2|mn$ and m, n both exceed 2; they can be used to show that in general a knight coclique on an $m \times n$ board has size at most $mn/2$ for such m, n.

Puzzle 2. What happens if m, n are both odd, or if $m \leq 2$ or $n \leq 2$?

Are Diagram 6 and its complement the only maximal cocliques? Yes, but this is harder to show. One elegant proof, given by Greenberg in [21], invokes the existence of a closed knight's tour, such as Euler's Diagram 4. In general, on a circuit of length $2M$ the only sets of M pairwise nonadjacent vertices are the set of even-numbered vertices and the set of odd-numbered ones on the circuit. Here $M = 32$, and the knight's tour in effect embeds that circuit into $\mathcal{G}$, so *a fortiori* there can be at most two cocliques of size M on $\mathcal{G}$ — and we have already found them both!

Of course this proof applies equally to any board with a closed knight's tour: on any such board the light- and dark-squared subsets are the only maximal cocliques. Conversely, a board for which there are further maximal cocliques cannot support a closed knight's tour. For example, any $4 \times n$ board has a mixed-color maximal coclique, as illustrated for $n = 11$ in the next diagram:

Diagram 8



a third maximal knight coclique on the 4×11 board

This yields possibly the cleanest proof that *there is no closed knight's tour on a $4 \times n$ board for any n.* (According to Jelliss [14], this fact was known to Euler and first proved by C. Flye Sainte-Marie in 1877; Jelliss attributes the above clean proof to Louis Posa.)

Warning: the existence of a closed knight's tour is a sufficient but not necessary condition for the existence of only two maximal knight cocliques. It is known that an $m \times n$ board supports a closed tour if and only if its area mn is an even integer > 24 and neither m nor n is 1, 2, or 4. In particular, as noted above there are no closed knight's tours on the 3×6 and 3×8 boards, though as it happens on each of these boards the only maximal knight cocliques are the two obvious monochromatic ones.

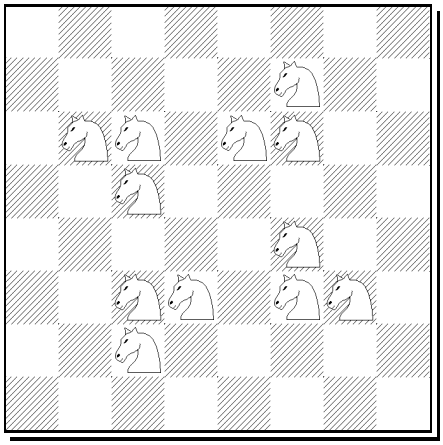
More about $\mathcal{G}$: Domination number, girth, and the knight metric

Another classic puzzle asks: how many knights does it take to either occupy or defend every square on the board? In graph theory parlance this asks for the "domination number" of $\mathcal{G}$.[2]

[2] This terminology is not entirely foreign to the chess literature: A piece is said to be "dominated" when it can move to many squares but will be lost on any of them. (The meaning of "many" in this definition is not precise because domination is an artistic concept, not a mathematical one.) The introduction of this term into the chess lexicon is attributed to Henri Rinck ([12, p. 93], [16, p. 151]). The task of constructing economical domination positions, where a few chessmen cover many squares, has a pronounced combinatorial flavor; the great composer of endgame studies G.M. Kasparyan devoted an entire book to the subject, *Domination in 2545 Endgame Studies*, Progress Publishers, Moscow, 1980.

For the standard 8×8 board, the following symmetrical solution with 12 knights has long been known:

Diagram 9



All unoccupied squares controlled

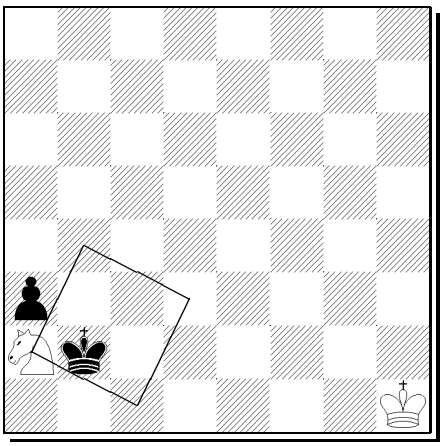
Puzzle 3. Prove that this solution is unique up to reflection.

The knight domination number for chessboards of arbitrary size is not known, not even asymptotically. See [9, Ch.14] for results known at the time for square boards of order up to 15, most dating back to 1918 [1, Vol.2, p. 359]. If we ask instead that every square, occupied or not, be defended, then the 8×8 chessboard requires 14 knights. On an $m \times n$ board, at least $mn/8$ knights are needed since a knight defends at most 8 squares.

Puzzle 4. Prove that $mn/8 + O(m+n)$ knights suffice. HINT: treat the light and dark squares separately.

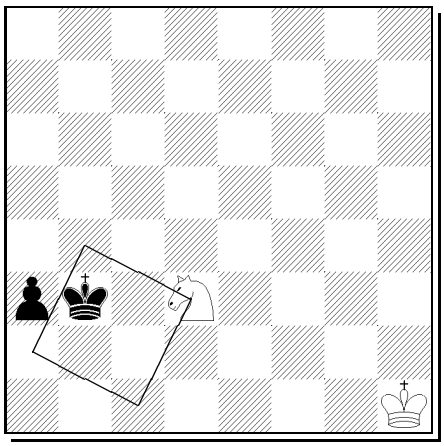
We already noted that $\mathcal{G}$, being bipartite, has no cycles of odd length. (We also encountered the non-existence of 3-cycles as "$\mathcal{G}$ has no cliques of size 3".) Thus the girth (minimal cycle length) of $\mathcal{G}$ is at least 4. In fact the girth is exactly 4, as shown for instance in Diagram 10.

Diagram 10



White to move draws

Diagram 10a



After 2 Nd3!

This square cycle is important to endgame theory: a White knight traveling on the cycle can

prevent the promotion of the Black pawn on a3 supported by its king. To draw this position White must either block the pawn or capture it, even at the cost of the knight. The point is seen after 1.Nb4 Kb3 2.Nd3! (reaching Diagram 10a) a2 3.Nc1+!, "forking" king and pawn and giving White time for 4.N×a2 and a draw. On other Black moves from Diagram 10a White resumes control of a2 with 3.Nc1 or 3.Nb4; for instance 2 . . . Kc2 3.Nb4+ or 2 . . . Kc3 3.Nc1 Kb2 (else Na2+) 4.Nd3+! etc. Note that the White king was not needed. [3]

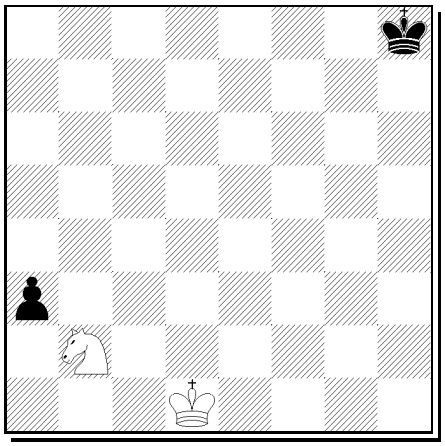
Puzzle 5. Construct a position where this Nd5 resource is White's only way to draw.

Warning: this puzzle is hard, and requires considerably more chess background than anything else in this article. The construction requires some delicacy: is not enough to simply stalemate the White king, since then White can play 2.Na2 with impunity; on the other hand if the White king is put in Zugzwang (so that it has some legal moves, but all of them lose), then the direct 1 . . . a2 2.N×a2 K×a2 wins for Black.

Even more important for the practical chessplayer is the distance function on $\mathcal{G}$, which encodes the number of moves a knight needs to get from any square to any other. The diameter (maximal distance) on $\mathcal{G}$ is 6, which is attained only by diagonally opposite corners. This is to be expected, but shorter distances bring some surprises. The following table shows the distance from each vertex of $\mathcal{G}$ to a corner square:

5	4	5	4	5	4	5	6
4	3	4	3	4	5	4	5
3	4	3	4	3	4	5	4
2	3	2	3	4	3	4	5
3	2	3	2	3	4	3	4
2	1	4	3	2	3	4	5
3	4*	1	2	3	4	3	4
♘	3	2	3	2	3	4	5

Diagram 11



White loses

The starred entry is due to the board edges: a knight can travel from any square to any diagonally adjacent square in two moves except when one of them is a corner square. But the other irregularities of the table at short distances do not depend on edge effects. Anywhere on the board, it takes the otherwise agile knight three moves to reach an orthogonally adjacent square, and four moves to travel two squares diagonally. This peculiarity must be absorbed by any chessplayer who would learn to play with or against knights. One consequence, known to endgame theory, is Diagram 11, which exploits both the generic irregularity and the special corner case. Even with White to move, this position is a win for Black, who will play . . . a2 and . . . a1Q. One might expect that the knight is close enough to stop this, but in fact it would take it three moves to reach a2 and four to reach a1, in each case one too many. In

[3] Note to more advanced chessplayers: it might seem that the knight does need a bit of help after 1.Nb4 Kb1!?, when either 2.Na2? or 2.Nd3? loses (in the latter case to 2 . . . a2) but Black has no threat so White can simply make a random ("waiting") king move. But this is not necessary, as White could also draw by thinking (and playing) out of the a2-b4-d3-c1-a2 box: 1.Nb4 Kb1 2.Nd5! If now 2 . . . a2 then 3.Nc3+ is a new drawing fork, and otherwise White plays 3.Nb4 and resumes the square dance.

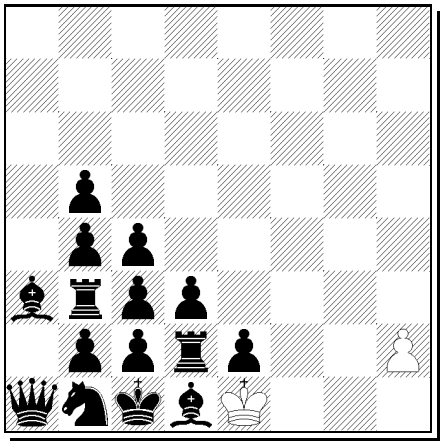
fact this knight helps Black by blocking the White king's approach to a1!

Puzzle 6. Determine the knight distance from $(0,0)$ to (m,n) on an infinite board as a function of the integers m, n.

Further puzzles

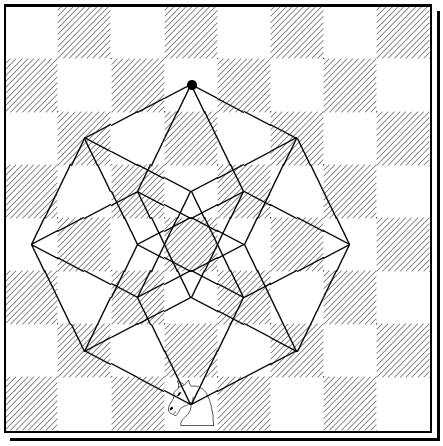
We conclude the first part with several more puzzles that exploit or extend our discussion:

Diagram 12



White to play and mate as quickly as possible

Diagram 13



the 4! shortest knight paths from d1 to d7

Puzzle 7. How does White play in Diagram 12 to force checkmate as quickly as possible against any Black defense?

Yes, it's White who wins, despite having only king and pawn against 15 Black men. But these men are almost paralyzed, with only the queen able to move in its corner prison. White must keep it that way: if he ever moves his king, Black will sacrifice his e2-pawn by promoting it, bring the Black army to life and soon overwhelm White. So White must move only the pawn, and the piece that it will promote to. That's good enough for a draw, but how to actually win?

Puzzle 8. (See Diagram 13.) There are exactly $24 = 4!$ paths that a knight on d1 can take to reach d7 in four moves; plotting these paths on the chessboard yields a beautiful projection of (the 1-skeleton of) the 4-dimensional hypercube! Explain.

Puzzle 9. We saw that there is an essentally unique maximal configuration of 32 mutually non-defending knights on the 8×8 board.
i) Suppose we allow each knight to be defended at most once. How many more knights can the board then accommodate?
ii) Now suppose we require each knight to be defended *exactly* once. What is the largest number of knights on the 8×8 board satisfying this constraint, and what are all the maximal configurations?

Puzzle 10. A "camel" is a $(3,1)$ leaper, that is, an unorthodox chess piece that moves from (x,y) to one of the squares $(x \pm 3, y \pm 1)$ or $(x \pm 1, y \pm 3)$. (A knight is a $(2,1)$ leaper.) Since there are eight such squares, it takes at least $mn/8$ camels to defend every square, occupied or not, on an $m \times n$ board. Are $mn/8 + O(m+n)$ sufficient, as in Puzzle 4?

Synthetic games

The remainder of this article will be devoted to composed chess problems featuring knights. A *synthetic game* [13] is a chess game composed (rather than played) in order to achieve some objective, usually in a minimal number of moves. Ideally the solution should be unique, but this is very rare. Failing this, we can hope for an "almost unique" solution, e.g., one where the final position is unique though not the move order. For instance, the shortest game ending in checkmate by a knight is 3.0 moves: 1.e3 Nc6 2.Ne2 Nd4 3.g3 Nf3 mate. White can vary the order of his moves and can play e4 and/or g4 instead of e3 and g3. The Black knight has two paths to f3. The biggest flaw, however, is that White could play c3/c4 instead of g3/g4, and Black could mate at d3. At least all 72 solutions share the central feature that White incarcerates his king at its home square. A better synthetic game involving a knight is the following.

Puzzle 11. Construct a game of chess in which Black checkmates White on Black's fifth move by promoting a pawn to a knight.

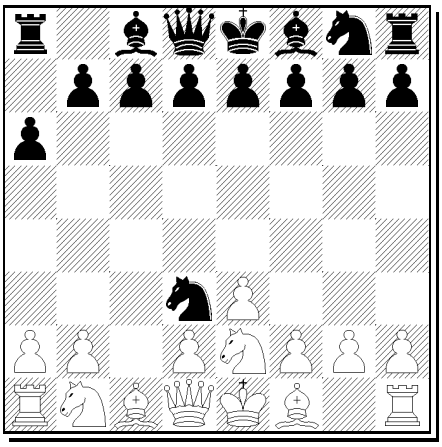
Proof games

A very successful variation of synthetic games that allows unique solutions are *proof games*, for which the length n of the game and the final position P are specified. In order for the condition (P, n) to be considered a sound problem, there should be a *unique* game in n moves ending in P. (Sometimes there will be more than one solution, but they should be related in some thematic way. Here we will only consider conditions (P, n) that are uniquely realizable, with the exception of Diagram 17.)

The earliest proof games were composed by the famous "Puzzle King" Sam Loyd in the 1890's but did not have unique solutions; the earliest sound (by today's standards) proof game seems to have been composed by T. R. Dawson in 1913. Although some interesting proof games were composed in subsequent years, the vast potential of the subject was not suspected until the fantastic pioneering efforts of Michel Caillaud in the early 1980's. A close to complete collection of all proof games published up to 1991 (around 160 problems) appears in [28].

Let us consider some proof games related to knights. We mentioned above that the shortest game ending in mate by knight has length 3.0 moves. None of the 72 solutions yield proof games with unique solutions, i.e., every terminal position has more than one way of reaching it in 3.0 moves. It is therefore natural to ask for the least number n (either an integer or half-integer) for which there exists a *uniquely realizable* game of chess in n moves ending with checkmate by knight, i.e., given the final position, there is a unique game that reaches it in n moves. Such a game was found independently by the two authors of this article in 1996 for $n = 4.0$, which is surely the minimum. The final position is shown in Diagram 14.

Diagram 14



Position after Black's 4th move. How did the game go?

Five other proof game problems involving knights are the following. The minimum known number of moves for achieving the game is given in parentheses. (We repeat that the game must be uniquely realizable from the number of moves and final position.)

Puzzle 12. Construct a proof game without any captures that ends with mate by a knight (4.5).

Puzzle 13. Construct a proof game ending with mate by a knight making a capture (5.5)

Puzzle 14. Construct a proof game ending with mate by a pawn promoting to a knight (5.5).

Puzzle 15. Construct a proof game ending with mate by a pawn promoting to a knight without a capture on the mating move (6.0).

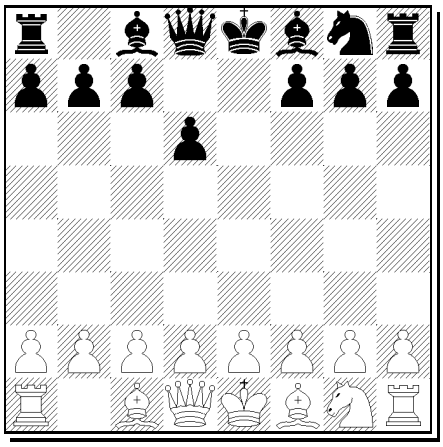
Puzzle 16. Construct a proof game ending with mate by a pawn promoting to a knight with no captures by the mating side throughout the game (7.0).

There is a remarkable variant of Puzzle 14. Rather than having the game determined by its final position and number of moves, it is instead completely determined by its last move (including the move number)! This is the longest known game with this property.

Puzzle 14′. Construct a game of chess with last move 6.g×f8N mate.

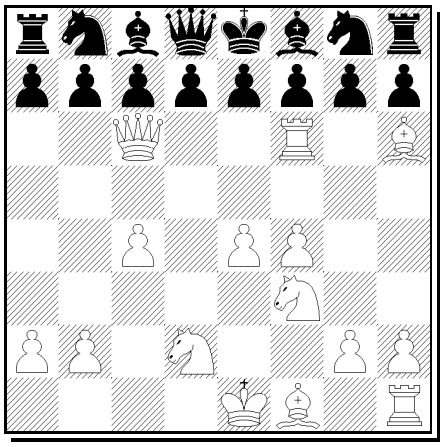
The above proof games focused on achieving some objective in the minimum number of moves. Many other proof games in which knights play a key role have been composed, of which we give a sample of five problems. Diagrams 15, 16, and 17 feature "impostors"—some piece(s) are not what they seem. The first of these (Diagram 15) is a classic problem that is one of the earliest of all proof games, while Diagram 16 is considerably more challenging. Diagram 17 features a different kind of impostor. Note that it has two solutions; it is remarkable how each solution has a different impostor. The complex and difficult Diagram 18 illustrates the *Frolkin theme*: the multiple capture of promoted pieces. Diagram 19 shows, in the words of Wilts and Frolkin [28, p. 53], that "the seemingly indisputable fact that a knight cannot lose a tempo is not quite unambiguous."

Diagram 15



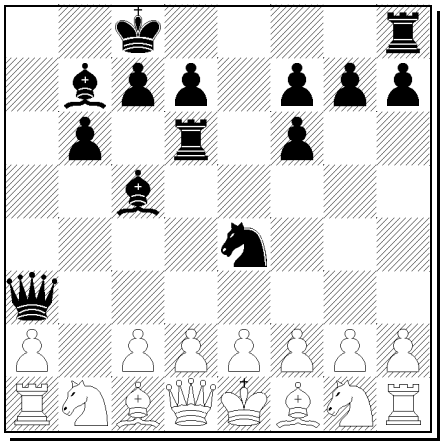
After Black's 4th. How did the game go?

Diagram 16



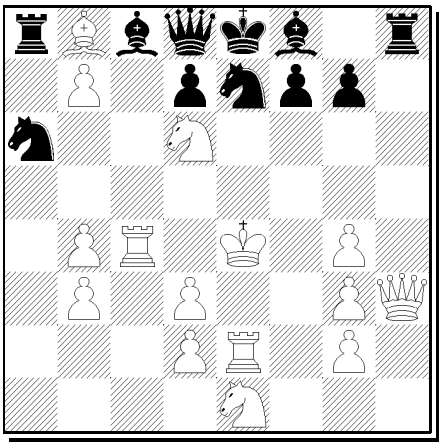
After Black's 12th. How did the game go?

Diagram 17



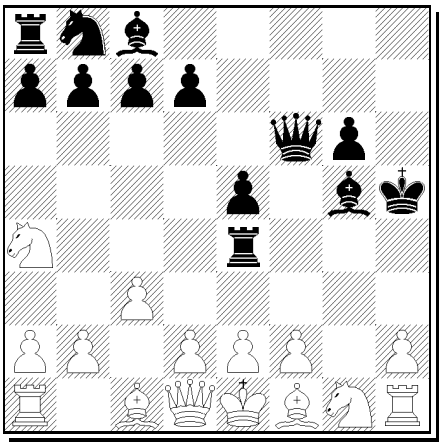
After White's 13th. How did the game go? Two solutions!

Diagram 18



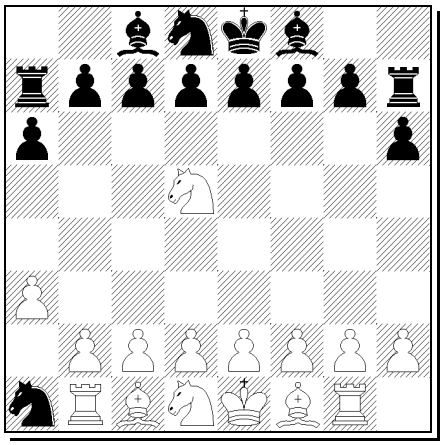
After White's 27th. How did the game go?

Diagram 19



After Black's 10th move. How did the game go?

Diagram 20



Mate in one

Retrograde analysis

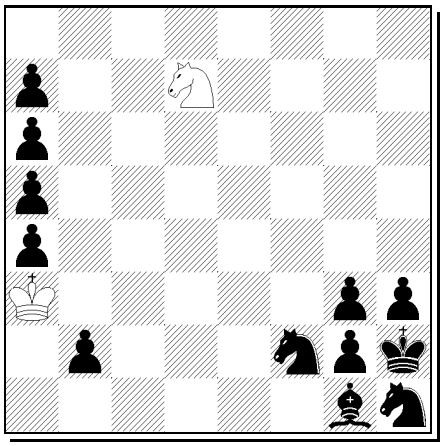
In retrograde analysis problems (called retro problems for short), it is necessary to deduce information from the current position concerning the prior history of the game. It is only assumed that the prior play is legal; no assumption is made that the play is "sensible." Proof games are a special class of retro problems. We will give only one illustration here of a retro problem that is not a proof game. It is based on considerations of parity, a common theme whenever knights are involved. Diagram 20 is a *mate in one*. A chess problem with this stipulation almost invariably involves an element of retrograde analysis, such as determining who has the move.[4]

Length records

[4] In a problem with the stipulation "Mate in n," it is assumed that White moves first unless it can be proved that Black has the move in order for the position to be legal.

Here one tries to construct a position that maximizes the number of moves which must elapse before a certain objective is satisfied. The most obvious and most-studied objective is checkmate. In other words, how large can n be in a problem with the objective "mate in n" (i.e., White to play and checkmate Black in n moves)? Chess problem standards demand that the solution should be unique if at all possible. It is too much to expect, especially for long-range problems, that White has a unique response to *every* Black move in order for White to achieve his objective. In other words, it is possible for Black to defend poorly and allow White to achieve his objective in more than one way, or even achieve it earlier than specified. The correct uniqueness condition is that the problem should be *dual-free*, which means that Black has at least one method of defending which forces each White move uniquely if White is to achieve his objective. The objective of checkmate can be combined with other conditions, such as White having only one unit besides his king. The ingenious Diagram 21 shows the current record for a "knight minimal," i.e., White's only unit besides his king is a knight. For other length records, as well as many other tasks and records, see [20].

Diagram 21



Mate in 48

Paradox

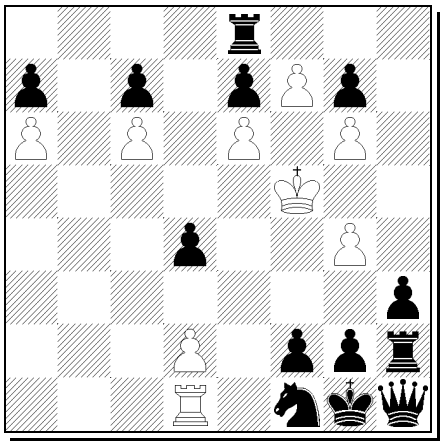
The term "paradox" has several meanings in both mathematics and ordinary discourse. We will regard a feature of a chess problem (or chess game) as paradoxical if it is seemingly opposed to common sense. For instance, common sense tells us that a material advantage is beneficial in winning a chess game or mating quickly. Thus *sacrifice* in an orthodox chess problem (i.e., a direct mate or study) is paradoxical. Of course it is just this paradoxical element that explains the appeal of a sacrifice. Another common paradoxical theme is underpromotion. Why not promote to the strongest possible piece, namely, the queen? This theme is related to that of sacrifice, since in each case the player is forgoing material. To be sure, underpromotion to knight in order to win, draw, or checkmate quickly is not so surprising (and has even occurred a fair number of times in games) since a knight can make moves forbidden to a queen. Tim Krabbé thus remarks in [15] that knighting hardly counts as a true "underpromotion."[5] Nevertheless, knight promotions can be used for surprising purposes that heighten the paradoxical effect.

Diagram 22 shows four knight sacrifices, all promoted pawns, with a total of five promotions

[5]More paradoxical are underpromotions to rooks and bishops, but we will not be concerned with them here.

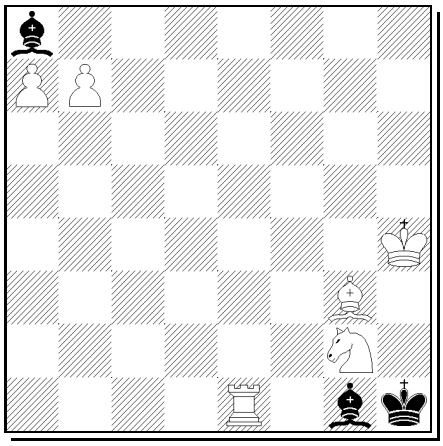
to knight. Diagram 23 shows a celebrated problem composed by Sam Loyd where a pawn promotes to a knight that threatens no pieces or checks and is hopelessly out of play. For some interesting comments by Loyd on this problem, see [27, p. 403].

Diagram 22



White to play and win

Diagram 23



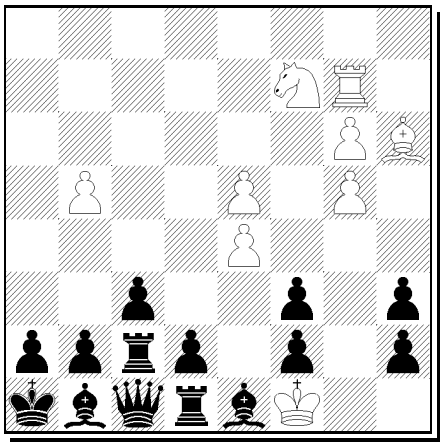
Mate in 3

Note that the impostors of Figures 15–17 may also be regarded as paradoxical, since we're trying to reach the position as quickly as possible, and it seems a waste of time to move knights into the original square(s) of other knights. Similarly the time-wasting 5.h×g8N 6.Nh6 7.N×f7 of Diagram 19 seems paradoxical—why not save a move by 5.h×g8B and 6.B×f7+?

Helpmate

In a *helpmate in n moves*, Black moves first and *cooperates* with White so that White mates Black on White's nth move. If the number of solutions of a helpmate is not specified, then there should be a unique solution. For a long time it was thought impossible to construct a sound helpmate with the theme of Diagram 24, featuring knight promotions. Note that the first obstacle to overcome is the avoidance of checkmating White or stalemating Black. The composer of this brilliant problem, Gabor Cseh, was tragically killed in an accident in 2001 at the age of 26.

Diagram 24

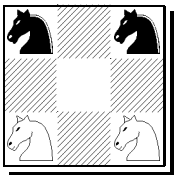


Helpmate in 10

Piece shuffle

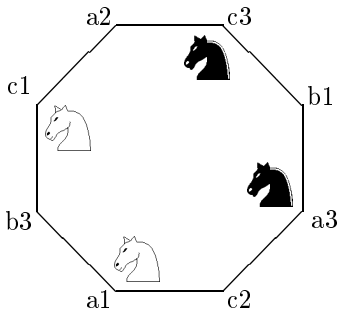
In *piece shuffles* or *permutation tasks*, a rearrangement of pieces is to be achieved in a minimum number of moves, sometimes subject to special conditions. They may be regarded as special cases of "moving counter problems" such as given in [2, pp. 769–777] or [3, pp. 58–68]. A classic example involving knights, going back to Guarini in 1512, is shown in Diagram 25. The knights are to exchange places in the minimum number of moves. (Each White knight ends up where a Black knight begins, and *vice versa*.) The systematic method for doing such problems, first enunciated by Dudeney [3, solution to #341] and called the method of "buttons and strings," is to form a graph whose vertices are the squares of the board, with an edge between two vertices if the problem piece (here a knight) can move from one vertex to the other. For Diagram 25 the graph is just an eight-cycle (with an irrelevant isolated vertex corresponding to the center square of the board). See Diagram 26. This representation of the problem makes it quite easy to see that the minimum number of moves is sixteen (eight by each color), achieved for instance by cyclically moving each knight four steps clockwise around the eight-cycle. If a White knight is added at b1 and a Black knight at b3, then somewhat paradoxically the minimum number of moves is reduced to eight! A variation of the stipulation of Diagram 25 is the following problem, whose solution is a bit tricky and essentially unique.

Diagram 25



Exchange the knights
in a minimum number of moves

Diagram 26



The graph corresponding
to Diagram 25

Puzzle 17 In Diagram 25 exchange the knights in a minimum number of move sequences, where a "move sequence" is an unlimited number of consecutive moves by the same knight.

For some more sophisticated problems similar to Diagram 25, see [10, pp. 114–124]. The most interesting piece shuffle problems connected with the game of chess (though not focusing on knights) are due to G. Foster [5, 6, 7, 8], created with the help of his computer program WOMBAT (Work Out Matrix By Algorithmic Techniques).

Puzzle answers, hints, and solutions

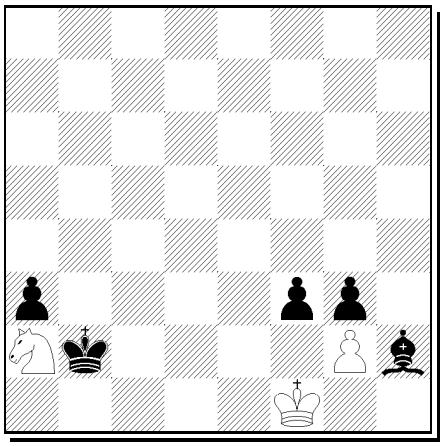
1 The graph $\mathcal{G}_{m,n}$ is connected for $m = n = 1$ (only one vertex) and not connected for $m = n = 3$ (the central square is an isolated vertex). With those two exceptions, $\mathcal{G}_{m,n}$ is connected if and only if $m > 2$ and $n > 2$. Every $G_{m,n}$ is bipartite, except $G_{1,1}$ (empty parts not allowed); each non-connected graph $G_{m,n}$ is bipartite in several ways except for $\mathcal{G}_{1,2} = \mathcal{G}_{2,1}$.

2 If $m = 1$ or $n = 1$ then $\mathcal{G}_{m,n}$ is disconnected, so the maximal coclique is the set of all mn vertices. The graph $\mathcal{G}_{2,n}$ (or $\mathcal{G}_{n,2}$) decomposes into two paths of length $\lfloor n/2 \rfloor$ and two of length $\lceil n/2 \rceil$. It thus has a one-factor if an only if $4|n$, and otherwise has cocliques of size $> n$; the maximal coclique size is $n+\delta$ where $\delta \in \{0, 1, 2\}$ and $n \equiv \pm\delta \bmod 4$. If m and n are odd integers greater than 1 then the maximal coclique size of $\mathcal{G}_{m,n}$ is $(mn+1)/2$, attained by placing a knight on each square of the same parity as a corner square of an $m \times n$ board. One can prove that this is maximal by deleting one of these squares and constructing a one-factor on the remaining $mn - 1$ vertices of $\mathcal{G}_{m,n}$.

3 Each of the four 2×2 corner subboards requires at least three knights, and no single knight may occupy or defend squares in two different subboards. Hence at least $4 \cdot 3 = 12$ knights are needed. For three knights to cover the $\{\mathrm{a1}, \mathrm{b1}, \mathrm{a2}, \mathrm{b2}\}$ subboard, one of them must be on c3; likewise f3, f6, c6 must be occupied if 12 knights are to suffice. It is now easy to verify that Diagram 9 and its reflection are the only ways to place the remaining 8 knights so as to cover the entire chessboard.

4 ([3, #319, p. 127]) On an infinite chessboard, each square of odd parity is a knight-move away from exactly one of the squares with coordinates $(2x, 2y)$ with $x \equiv y \bmod 4$. Intersecting this lattice with an $m \times n$ chessboard yields $mn/16 + O(m + n)$ knights that cover all odd squares at distance at least 3 from the nearest edge. Thus an extra $O(m+n)$ knights defend all the odd squares on the board. The same construction for the even squares yields a total of $mn/8 + O(m + n)$.

Diagram 27



White to move draws

5 One such position is Diagram 27 above. Once the a-pawn is gone, the position is a theoretical draw whether Black plays f×g2+ (Black can do no better than stalemate against K×g2, Kh1, Kg2 etc.) or f2 (ditto after Ke2, Kf1, etc.), or lets White play g×f3 and Kg2 and then jettison the f-pawn to reach the same draw that follows f×g2+. But as long as Black's a-pawn is on the board, White can move only the knight since g×f3 would liberate Black's bishop which could then force White's knight away (for instance 1.Nb4 Kb1 2.g×f3? g2+! 3.K×g2 Bd6 4.Nd5 Kb2) and safely promote the a-pawn. Black's pawn on f3 could also be on h3 with the same effect.

6 The distance is an integer, congruent to $m+n$ mod 2, that equals or exceeds each of $|m|/2$, $|n|/2$, and $(|m|+|n|)/3$. It is the smallest such integer except when in the cases already noted of $(m,n) = (0,\pm1)$, $(\pm1,0)$, or $(\pm2,\pm2)$, when the distance exceeds the above lower bound by 2.

7 (adapted from Gorgiev) To win, White must promote the pawn to a knight, capture the pawns on b5 and c4, and then mate with N×b3 when the Black queen is on a1. Thus N×b3 must be an odd-numbered move. Therefore 1.h4, 2.h5, 3.h6, 4.h7, 5.h8N does not work because all knight paths from h8 to b3 have odd length. Since the knight cannot "lose the move", the pawn must do so on its initial move: 1.h3!, followed by 6.h8N!, 7.Nf7, 8.Nd6, 9.N×b5, 10.Nd6, 11.N×c4, 12.Na5. At this point the Black queen is on a2, having made 11 moves from the initial position; whence the conclusion: 12 . . . Qa1 13.N×b3 mate. (We omitted from Gorgiev's original problem the initial move 1.Kf2×Ne1 Qa2-a1, which only served to give Black his entire army in the initial position and thus maximize the material disparity; and moved a Black pawn from c5 to b5 to make the solution unique, at some cost in strategic interest.)

8 Recall that a knight's move joins squares differing by one of the eight vectors $(\pm1,\pm2)$ or $(\pm2,\pm1)$, and check that to get some four of those to add to $(0,6)$ we must use the four vectors with a positive ordinate in some order. Thus, to reach d7 from d1 (or, more generally, to travel six squares north with no obstruction from the edges of the board) in four moves, the knight must move once in each of its four north-going directions. Therefore a path corresponds to a permutation of the four vectors $(\pm1,2)$ and $(\pm2,1)$. The number of paths is thus $4! = 24$, and drawing them all yields the image of the 4-cube under a projection taking the unit vectors to $(\pm1,2)$ and $(\pm2,1)$. Instead of d1 and d7 we could also draw the

24 paths from a4 to g4 in four moves to get the same picture. Not b2 and f6, though: besides the 24 paths of Diagram 13 there are other four-move journeys, for instance b2-d3-f4-h5-f6.

9 (i) The maximum is still 32 (though there are many more configurations that attain this maximum). To show this, it is enough to prove that at most 8 knights can fit on a 4×4 board if each is to be defended at most once. This in turn can be seen by decomposing $\mathcal{G}_{4,4}$ as a union of four 4-cycles (Diagram 28), and noting that only two knights can fit on each 4-cycle.

(ii) Once again, the maximum is 32, this time with a new configuration (Diagram 29) unique up to reflection! (But note that this configuration has a cyclic group of 4 symmetries, unlike the elementary abelian 2-group of symmetries of the maximal coclique (Diagram 6).) That this is maximal follows from the first part of this puzzle. For uniqueness, our proof is too long to reproduce here in full; it proceeds as follows. In any 32-knight configuration, each of the four 4×4 corner subboards must contain 8 knights, two on each of its four 4-cycles. We analyze cases to show that it is impossible for two knights in different subboards to defend each other. We then show that Diagram 29 and its reflection are the only ways to fit four 8-knight configurations into an 8×8 board under this constraint.

10 Yes, $mn/8+O(m+n)$ camels suffice. The camel always stays on squares of the same color. The squares of one color may be regarded on a chessboard in its own right, tilted 45° and magnified by a factor of $\sqrt{2}$ — in other words, multiplied by the complex number $1+i$. On this board, the camel's move amounts to the ordinary knight's move since $3+i = (2-i)(1+i)$. We can thus adapt our solution of Puzzle 4. Explicitly, on an infinite chessboard each square with both coordinates odd is a camel's move away from exactly one square of the form $(4x, 8y)$. Thus camels at $(4x + a, 8x + b)$ $(a, b \in \{0, 1\})$ cover the entire board without duplication, and the intersection of this configuration with an $m \times n$ board covers all but $O(m + n)$ of its squares.

Diagram 28

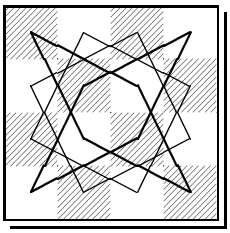
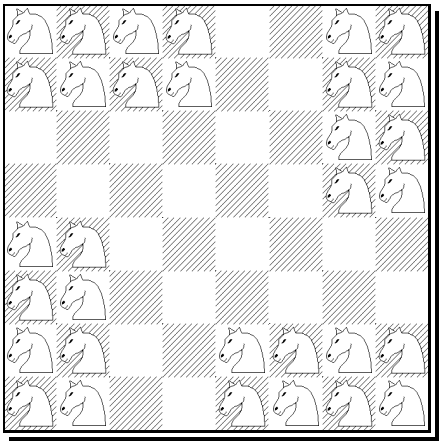


Diagram 29



Solution of Puzzle 8(ii)

11 1.d3 e5 2.Kd2 e4 3.Kc3 e×d3 4.b3 d×e2 5.Kb2 e×d1N mate. White can play d4 instead of d3 (so Black plays e×d4) and can vary his move order, but the final position is believed to be unique. This game first appeared in [17].

12 (G. Forslund, Retros Mailing List, June 1996) 1.e3 f5 2.Qf3 Kf7 3.Bc4+ Kf6 4.Qc6+ Ke5 5.Nf3 mate.

13 (G. Wicklund, Retros Mailing List, October 1996) 1.Nf3 e6 2.Ne5 Ne7 3.N×d7 e5 4.N×f8 Bd7 5.Ne6 Rf8 6.N×g7 mate.

14 (P. Rössler, *Problemkiste*, August 1994 (version)) 1.h4 d5 2.h5 Nd7 3.h6 Ndf6 4.h×g7 Kd7 5.Rh6 Ne8 6.g×f8N mate.

15 (G. Donati, Retros Mailing List, June 1996) 1.h4 g6 2.Rh3 g5 3.Re3 g×h4 4.f3 h3 5.Kf2 h2 6.Qe1 h1N mate.

16 (O. Heimo, Retros Mailing List, June 1996) 1.d4 e5 2.d×e5 d5 3.Qd4 Be6 4.Qb6 d4 5.Kd2 d3 6.Kc3 d2 7.a3 d1N mate.

14′ See solution to Puzzle 14.

17 a1-c2, c1-b3-a1, c3-a2-c1-b3, a3-b1-c3-a2-c1, c2-a3-b1-c3, a1-c2-a3, b3-c1. Seven move sequences.

Diagram solutions

Diagram 14. (N. Elkies, R. Stanley, 1996) 1.c4 Na6 2.c5 N×c5 3.e3 a6 4.Ne2 Nd3 mate.

Diagram 15. (G. Schweig, *Tukon*, 1938) 1.Nc3 d6 2.Nd5 Nd7 3.N×e7 Ndf6 4.N×g8 N×g8. The impostor is the knight at g8, which actually started out at b8.

Diagram 16. (U. Heinonen, *The Problemist* 1991) 1.c4 Nf6 2.Qa4 Ne4 3.Qc6 N×d2 4.e4 Nb3 5.Bh6 Na6! 6.Nd2 Nb4 7.Rc1 Nd5 8.Rc3 Nf6 9.Rf3 Ng8 10.Rf6 Nc5 11.f4 Na6 12.Ngf3 Nb8. Here both Black knights are impostors, as they have exchanged places! For a detailed analysis of this problem, see [16, pp. 207–209].

Diagram 17 (D. Pronkin, *Die Schwalbe*, 1985, 1st prize) 1.b4 Nf6 2.Bb2 Ne4 3.Bf6 e×f6 4.b5 Qe7 5.b6 Qa3 6.b×a7 Bc5 7.a×b8B Ra6 8.Ba7 Rd6 9.Bb6 Kd8 10.Ba5 b6 11.Bc3 Bb7 12.Bb2 Kc8 13.Bc1.

1.Nc3 Nf6 2.Nd5 Ne4 3.Nf6+ e×f6 4.b4 Qe7 5.b5 Qa3 6.b6 Bc5 7.b×a7 b6 8.a×b8N Bb7 9.Na6 0-0-0 10.Nb4 Rde8 11.Nd5 Re6 12.Nc3 Rd6 13.Nb1. This problem illustrates the *Phoenix theme*: a piece leaves its original square to be sacrificed somewhere else, then a pawn promotes to exactly the same piece which returns to the original square to replace the sacrificed piece. In the first solution the bishop at c1 is phoenix, while in the second it is the knight at b1! As if this weren't spectacular enough, Black castles in the second solution but not the first.

Diagram 18. (M. Caillaud, *Thèmes-64*, 1982, 1st prize) 1.a4 c5 2.a5 c4 3.a6 c3 4.a×b7 a5 5.Ra4 Na6 6.Rc4 a4 7.b4 a3 8.Bb2 a2 9.Na3 a1N! 10.Nb5 Nb3 11.c×b3 c2 12.Be5 c1N! 13.Bb8 Nd3+ 14.e×d3 e5 15.Qg4 e4 16.Ke2 e3 17.Kf3 e2 18.Ke4 e×f1N! 19.Nf3 Ng3+ 20.h×g3 h5 21.Re1 h4 22.Re2 h3 23.Ne1 h2 24.Qh3 h1N! 25.g4 Ng3+ 26.f×g3 Ne7 27.Nd6 mate. An amazing four promotions by Black to knight, all captured!

Diagram 19. (A. Frolkin, *Shortest Proof Games*, 1991) 1.g4 e5 2.g5 Be7 3.g6 Bg5 4.g×h7 Qf6 5.h×g8N! Rh4 6.Nh6 Re4 7.N×f7 K×f7 8.Nc3 Kg6 9.Na4 Kh5 10.c3 g6. If 5.h×g8B? Rh4 6.B×f7+ K×f7 7.Nc3 Re4 8.Na4 Kg6 9.c3 Kh5, then White must disturb his position before 10... g6. A knight is able to "lose a tempo" by taking two moves to get from g8 to f7, while a bishop must take one or at least three moves.

Diagram 20. (V. A. Korolikov, *Schach*, 1957) White's knights are on squares of the same color and hence have made an odd number of moves in all. Each White rook and the White king have made an even number of moves, and White has made one pawn move. No other

White unit (i.e., the queen and bishops) have moved. Hence White has made an even number of moves in all. Similarly Black has made an odd number of moves. Since White moved first it is currently Black's move, so Black mates in one with 1...N×c2 mate.

Diagram 21. (P. O'Shea, *The Problemist*, 1989, 1st prize) 1.Ne5 b1N+ (the only defense to 2.Nf3 mate) 2.Ka2 Nd2 3.Ka1 Nb3+ 4.Kb1 Nd2+ 5.Ka2. If Black moves either knight then checkmate is immediate, so 5... a3 is forced. Now White and Black repeat the maneuver Ka1, Nb3+, Kb1, Nd2+, Ka2 (any pawn moves by Black would just hasten the end): 8.Ka2 a4 11.Ka2 a5 14.Ka2 a6. Then 15.Ka1 Nb3+ 16.Kb1 a2+ 17.K×a2 Nd2. This manuever gets repeated until all Black's a-pawns are captured: 44.K×a2 Nd2 45.Ka1 Nb3+ 46.Kb1 Nd2+ 47.Ka2. Finally Black must allow 48.Nf3 mate or 48.Ng4 mate!

Diagram 22. (H. M. Lommer, *Szachy*, 1965) White cannot allow Black's rook at e8 to stay on the board, but how does White prevent Black from being stalemated without releasing the sleeping units in the h1 corner? 1.f×e8N d3 2.Nf6 (not 2.Nd6? e×d6, and stalemate cannot be prevented without releasing the h1 corner) g×f6 (capturing with the other pawn merely hastens the end) 3.g5 f×g5 4.g7 g4 5.g8N g3 6.Nf6 e×f6 7.Kg6 f5 8.e7 f4 9.e8N f3 10.Nd6 c×d6 11.c7 d5 12.c8N d4 13.Nb6 a×b6 14.a7 b5 15.a8N and wins, as White can play 19 N×f3 mate just after 18... b1Q. For the history of this problem, see [25, pp. xxi–xxii].

Diagram 23. (S. Loyd, *Holyoke Transcript*, 1876) 1.b×a8N! K×g2 2.Nb6, followed by 3.a8Q (or B) mate. Note that a knight is needed to prevent 2... B×a7. A queen or bishop promotion at move one would be stalemate, and a rook promotion leads nowhere. Normally a key move of capturing a piece is considered a serious flaw since it reduces Black's strength. Here, however, the capture seems to accomplish nothing so it is acceptable. Loyd himself says "[i]f the capture seems a hopeless move ...then it is obviously well concealed, and the most difficult key-move that could be selected" [18, p. 156]. For further problems by Loyd featuring distant knight promotion, see [27, pp. 402–403].

Diagram 24. (G. Cseh, *StrateGems*, 2000, 1st prize) 1.h1N! Nd6 2.h2 Nf5 3.Ng3+ N×g3 4.h1N! Ne2 5.f×e2+ Kg2! (not 5... K×e2?, since Black's tenth move would then check White) 6.f1N! Rc7 7.Bg3 R×c3 8.B×e5 R×c2 9.Bg7 R×c1 10.b×c1N! B×g7 mate. Four promotions to knight by Black.

References

[1] W. Ahrens, *Mathematische Unterhaltungen und Spiele*, Teubner, Leipzig, 1910 (Vol. 1) and 1918 (Vol. 2).

[2] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning Ways*, vol. 2, Academic Press, London/New York, 1982.

[3] H. E. Dudeney, *Amusements in Mathematics*, Dover, New ork, 1958, 1970 (reprint of Nelson, 1917).

[4] N. D. Elkies, On numbers and endgames: Combinatorial game theory in chess endgames, in [22], pp. 135–150.

[5] G. Foster, Sliding-block problems, Part 1, *The Problemist Supplement* **49** (November, 2000), 405–407.

[6] G. Foster, Sliding-block problems, Part 2, *The Problemist Supplement* **51** (March, 2001), 430–432.

[7] G. Foster, Sliding-block problems, Part 3, *The Problemist Supplement* **54** (September, 2001), 454.

[8] G. Foster, Sliding-block problems, Part 4, *The Problemist Supplement* **55** (November, 2001), 463–464.

[9] M. Gardner, *Mathematical Magic Show*, Vintage Books, New York, 1978.

[10] J. Gik, *Schach und Mathematik*, MIR, Moscow, and Urania-Verlag, Leipzig/Jena/Berlin, 1986; translated from the Russian original published in 1983.

[11] B. Hochberg, *Chess Braintwisters*, Sterling, New York, 1999.

[12] D. Hooper and K. Whyld, *The Oxford Companion to Chess*, Oxford University Press, 1984.

[13] G. P. Jelliss, *Synthetic Games*, September 1998, 22 pp.

[14] G. P. Jelliss, *Knight's Tour Notes*: Knight's Tours of Four-Rank Boards (Note 4a, 30 November 2001), `http://home.freeuk.net/ktn/4a.htm`

[15] T. Krabbé, *Chess Curiosities*, George Allen & Unwin Ltd., London, 1985.

[16] J. Levitt and D. Friedgood: *Secrets of Spectacular Chess*, Batsford, London, 1995.

[17] C. D. Locock, *Manchester Weekly Times*, December 28, 1912.

[18] S. Loyd, *Strategy*, 1881.

[19] B. McKay, Comments on: Martin Loebbing and Ingo Wegener, The Number of Knight's Tours Equals 33,439,123,484,294 — Counting with Binary Decision Diagrams, *Electronic J. Combinatorics*, `http://www.combinatorics.org/Volume_3/Comments/v3i1r5.html`.

[20] J. Morse, *Chess Problems: Tasks and Records*, Faber and Faber, 1995; second ed., 2001.

[21] I. Newman, problem E 1585 ("What is the maximum number of knights which can be placed on a chessboard in such a way that no knight attacks any other?"), with solutions by R. Patenaude and R. Greenberg, *Amer. Math. Monthly* **71** #2 (Feb. 1964), 210–211.

[22] R. J. Nowakowski, ed., *Games of No Chance*, MSRI Publ. #29 (proceedings of the 7/94 MSRI conference on combinatorial games), Cambridge Univ. Press, 1996.

[23] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, on the Web at `http://www.research.att.com/~njas/sequences`.

[24] L. Stiller, Multilinear Algebra and Chess Endgames, in [22], pp. 151–192.

[25] M. A. Sutherland and H. M. Lommer, *1234 Modern End-Game Studies*, Dover, New York, 1968.

[26] G. Törnberg, "Knight's Tour", `http://w1.859.telia.com/~u85905224/knight/eknight.htm`.

[27] A. C. White, *Sam Loyd and His Chess Problems*, Whitehead and Miller, 1913; reprinted (with corrections) by Dover, New York, 1962.

[28] G. Wilts and A. Frolkin, *Shortest Proof Games*, Gerd Wilts, Karlsruhe, 1991.

ON MULTI-AVOIDANCE OF GENERALIZED PATTERNS

SERGEY KITAEV AND TOUFIK MANSOUR

Abstract. In [Kit1] Kitaev discussed simultaneous avoidance of two 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. In three essentially different cases, the numbers of such n-permutations are 2^{n-1}, the number of involutions in $\mathcal{S}_n$, and $2E_n$, where E_n is the n-th Euler number. In this paper we give recurrence relations for the remaining three essentially different cases.

To complete the descriptions in [Kit3] and [KitMans], we consider avoidance of a pattern of the form x-y-z (a classical 3-pattern) and beginning or ending with an increasing or decreasing pattern. Moreover, we generalize this problem: we demand that a permutation must avoid a 3-pattern, begin with a certain pattern and end with a certain pattern simultaneously. We find the number of such permutations in case of avoiding an arbitrary generalized 3-pattern and beginning and ending with increasing or decreasing patterns.

1. Introduction and Background

Permutation patterns: All permutations in this paper are written as words $\pi = a_1 a_2 \dots a_n$, where the a_i consist of all the integers $1, 2, \dots, n$. Let $\alpha \in S_n$ and $\tau \in S_k$ be two permutations. We say that α *contains* τ if there exists a subsequence $1 \le i_1 < i_2 < \cdots < i_k \le n$ such that $(\alpha_{i_1}, \dots, \alpha_{i_k})$ is order-isomorphic to τ; in such a context τ is usually called a *pattern*. We say that α *avoids* τ, or is τ-*avoiding*, if such a subsequence does not exist. The set of all τ-avoiding permutations in S_n is denoted by $S_n(\tau)$. For an arbitrary finite collection of patterns T, we say that α avoids T if α avoids any $\tau \in T$; the corresponding subset of S_n is denoted by $S_n(T)$.

While the case of permutations avoiding a single pattern has attracted much attention, the case of multiple pattern avoidance remains less investigated. In particular, it is natural, as the next step, to consider permutations avoiding pairs of patterns τ_1, τ_2. This problem was solved completely for $\tau_1, \tau_2 \in S_3$ (see [SchSim]), for $\tau_1 \in S_3$ and $\tau_2 \in S_4$ (see [W]), and for $\tau_1, \tau_2 \in S_4$ (see [B, K] and references therein). Several recent papers [CW, MV1, Kr, MV3, MV2] deal with the case $\tau_1 \in S_3$, $\tau_2 \in S_k$ for various pairs τ_1, τ_2.

Generalized permutation patterns: In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns* (*GPs*) where two adjacent letters in a pattern may be required to be adjacent in the permutation. Such an adjacency requirement is indicated by the absence of a dash between the corresponding letters in the pattern. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, but the pattern 2-3-1 occurs also in the subwords 562 and 563. Note that a classical pattern should, in our notation, have dashes at the beginning and end. Since most of the patterns considered in this paper satisfy this, we suppress these dashes from the notation. Thus, a pattern with no dashes corresponds to a contiguous subword anywhere in a permutation. The motivation for introducing these patterns was the study of Mahonian statistics. A number of results on GPs were obtained by Claesson, Kitaev and Mansour. See for example [Claes], [Kit1, Kit2, Kit3] and [Mans1, Mans2, Mans3].

As in [SchSim], dealing with the classical patterns, one can consider the case when permutations have to avoid two or more generalized patterns simultaneously. A complete solution for the number of permutations avoiding a pair of 3-patterns of type (1,2) or (2,1), that is the patterns having one internal dash, is given in [ClaesMans1]. In [Kit1] Kitaev discussed simultaneous avoidance of two 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. In three essentially different cases, the numbers of such n-permutations are 2^{n-1}, the number of involutions in $\mathcal{S}_n$, and $2E_n$, where E_n is the n-th Euler number. The remaining cases are avoidance of 123 and 231, 213 and 231, 132 and 213. In Section 3 we give recurrence relations for these cases.

In Section 4, we consider avoidance of a pattern x-y-z, and beginning or ending with increasing or decreasing pattern. This completes the results made in [KitMans], which concerns the number of permutations that avoid a generalized 3-pattern and begin or end with an increasing or decreasing pattern.

In Sections 5–8, we give enumeration for the number of permutations that avoid a generalized 3-pattern, begin *and* end with increasing or decreasing patterns. We record our results in terms of either *generating functions*, or *exponential generating functions*, or formulas for the numbers which appear.

In Section 9, we discuss possible directions of generalization of the results from Sections 5–8.

2. Preliminaries

The *reverse* $R(\pi)$ of a permutation $\pi = a_1 a_2 \dots a_n$ is the permutation $a_n \dots a_2 a_1$. The *complement* $C(\pi)$ is the permutation $b_1 b_2 \dots b_n$ where $b_i = n + 1 - a_i$. Also, $R \circ C$ is the composition of R and C. For example,

$R(13254) = 45231$, $C(13254) = 53412$ and $R \circ C(13254) = 21435$. We call these bijections of S_n to itself *trivial*, and it is easy to see that for any pattern p the number $A_p(n)$ of permutations avoiding the pattern p is the same as for the patterns $R(p)$, $C(p)$ and $R \circ C(p)$. For example, the number of permutations that avoid the pattern 132 is the same as the number of permutations that avoid the pattern 231. This property holds for sets of patterns as well. If we apply one of the trivial bijections to all patterns of a set G, then we get a set G' for which $A_{G'}(n)$ is equal to $A_G(n)$. For example, the number of permutations avoiding $\{123, 132\}$ equals the number of those avoiding $\{321, 312\}$ because the second set is obtained from the first one by complementing each pattern.

In this paper we denote the nth Catalan number by C_n; the generating function for these numbers by $C(x)$; the nth Bell number by B_n.

Also, $N_p^q(n)$ denotes the number of permutations that avoid the pattern p and begin with the pattern q; $G_p^q(x)$ (respectively, $E_p^q(x)$) denotes the ordinary (respectively, exponential) generating function for the number of such permutations. Besides, $N_p^{q,r}(n)$ denotes the number of permutations that avoid the pattern p, begin with the pattern q and end with the pattern r; $G_p^{q,r}(x)$ (respectively, $E_p^{q,r}(x)$) denotes the ordinary (respectively, exponential) generating function for the number of such permutations.

Recall the following properties of $C(x)$:

$$C(x) = \frac{1 - \sqrt{1-4x}}{2x} = \frac{1}{1 - xC(x)}. \tag{1}$$

3. Simultaneous avoidance of two 3-patterns with no dashes

3.1. **Avoidance of patterns 123 and 231 simultaneously.** We first consider the avoidance of the patterns 123 and 231 simultaneously. Let $a(n; i_1, i_2, \ldots, i_m)$ denote the number of permutations $\pi \in S_n(123, 231)$ such that $\pi_1\pi_2 \ldots \pi_m = i_1 i_2 \ldots i_m$ and let $a(n) = |S_n(123, 231)|$. By the definitions, we get that $a(n) = \sum_{j=1}^{n} a(n; j)$ and $a(n; n) = a(n-1)$. Hence

$$a(n) = a(n-1) + a(n;1) + a(n;2) + \cdots + a(n; n-1). \tag{2}$$

Also, by the definitions, for all $1 \le i \le n-1$, we get

$$a(n; i) = \sum_{j=1}^{i-1} a(n; i, j) + \sum_{j=i+1}^{n} a(n; i, j). \tag{3}$$

Suppose $\pi \in S_n(123, 231)$ is such that $\pi_1 = i$ and $\pi_2 = j$. If $i > j$ then there is no occurrence of the pattern 123 or 231 that contains π_1, so $a(n; i, j) = a(n-1; j)$. If $i < j$ then since π avoids 123 and 231, we get that $i < \pi_3 < j$, and thus in this case $a(n; i, j) = a(n-2; i) + a(n-2; i+1) + \cdots + a(n-2; j-2)$. Hence, using (2) and (3), we get the following theorem.

Proposition 1. *For all* $n \geq 3$,

$$a(n) = a(n-1) + a(n;1) + a(n;2) + \cdots + a(n;n-1),$$

where for all $1 \leq i \leq n$,

$$a(n;i) = \sum_{j=1}^{i-1} a(n-1;j) + \sum_{j=i}^{n-2} (n-1-j)a(n-2;j),$$

and $a(3;1) = 1$, $a(3;2) = 1$, $a(3;3) = 2$.

Using this theorem, we get quickly the first values of the sequence $a(n)$ for $n = 0, 1, 2, \ldots, 10$:

n	0	1	2	3	4	5	6	7	8	9	10
$a(n)$	1	1	2	4	11	39	161	784	4368	27260	189540

3.2. **Avoidance of patterns 132 and 213 simultaneously.** We consider avoidance of the patterns 132 and 213 simultaneously. Let $b(n; i_1, i_2, \ldots, i_m)$ denote the number of permutations $\pi \in S_n(132, 213)$ such that $\pi_1\pi_2 \ldots \pi_m = i_1 i_2 \ldots i_m$ and let $b(n) = |S_n(132, 213)|$. Suppose $\pi \in S_n(132, 213)$ is such that $\pi_1 = i$ and $\pi_2 = j$. If $i > j$ then, since π avoids 213, we get $\pi_3 \leq i - 1$. Thus

$$b(n;i,j) = \sum_{k=1,\ k\neq j}^{i-1} b(n-1;j,k). \tag{4}$$

If $i < j$ then, since π avoids 132, we get $\pi_3 \leq i - 1$ or $\pi_3 \geq j + 1$. Thus

$$b(n;i,j) = \sum_{k=1}^{i-1} b(n-1;j-1,k) + \sum_{k=j}^{n-1} b(n-1;j-1,k). \tag{5}$$

Using (4) and (5), we get the following theorem.

Proposition 2. *We have* $b(n) = \sum_{i,j=1}^{n} b(n;i,j)$ *with*

$b(n;i,i) = 0$ *for all* $n, i \geq 1$;

$b(n;i,j) = \sum_{k=1}^{i-1} b(n-1;j,k)$ *if* $i > j$;

$b(n;i,j) = \sum_{k=1}^{i-1} b(n-1;j-1,k) + \sum_{k=j}^{n-1} b(n-1;j-1,k)$ *if* $i < j$;

and $b(2;1,2) = b(2;2,1) = 1$, $b(2;1,1) = b(2;1,1) = 0$.

Using this theorem, we get

n	0	1	2	3	4	5	6	7	8	9	10
$b(n)$	1	1	2	4	11	37	149	705	3814	23199	156940

3.3. **Avoidance of the patterns 213 and 231 simultaneously.** We now consider avoidance of the patterns 213 and 231 simultaneously. This case is equivalent to avoidance of the patterns 132 and 312 by applying the reverse operation. Let $c(n; i_1, i_2, \ldots, i_m)$ denote the number of permutations $\pi \in S_n(132, 312)$ such that $\pi_1\pi_2 \ldots \pi_m = i_1 i_2 \ldots i_m$ and let $c(n) = |S_n(132, 312)|$. We proceed as in the previous case. For $n \geq i > j \geq 1$, we have

$$c(n; i, j) = \sum_{k=1}^{j-1} c(n-1; j, k) + \sum_{k=i}^{n-1} c(n-1; j, k). \tag{6}$$

For $1 \leq i < j \leq n$, we have

$$c(n; i, j) = \sum_{k=1}^{i-1} c(n-1; j-1, k) + \sum_{k=j}^{n-1} c(n-1; j-1, k). \tag{7}$$

Using (6) and (7), we get the following theorem.

Proposition 3. *We have* $c(n) = \sum_{i,j=1}^{n} c(n; i, j)$ *with*

$c(n; i, i) = 0$ *for all* $n, i \geq 1$;

$c(n; i, j) = \sum_{k=1}^{j-1} c(n-1; j, k) + \sum_{k=i}^{n-1} c(n-1; j, k)$ *if* $i > j$;

$c(n; i, j) = \sum_{k=1}^{i-1} c(n-1; j-1, k) + \sum_{k=j}^{n-1} c(n-1; j-1, k)$ *if* $i < j$;

and $c(2; 1, 2) = c(2; 2, 1) = 1$, $c(2; 1, 1) = c(2; 1, 1) = 0$.

Using this theorem, we get

n	0	1	2	3	4	5	6	7	8	9	10
$c(n)$	1	1	2	4	10	30	108	454	2186	11840	71254

4. Avoiding a pattern x-y-z and beginning or ending with certain patterns

Recall that according to the definitions from Section 2, $N_p^{q,r}(n)$ denotes the number of permutations that avoid the pattern p, begin with the pattern q and end with the pattern r; $G_p^{q,r}(x)$ (respectively, $E_p^{q,r}(x)$) denotes the ordinary (respectively, exponential) generating function for the number of such permutations. Besides, C_n and $C(x)$ denote the n-th Catalan number and the ordinary generating function for the Catalan numbers.

Proposition 4. *We have*

$$G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C^2(x).$$

Proof. Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 < \pi_2 < \cdots < \pi_k$ and $\pi_j = n$. It is easy to see that π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \ldots, n-1$,

and $\pi'' \in S_{n-j}$(1-3-2). If we now consider two cases, namely $j = k$ and $j \geq k+1$, we get

$$G^{12\ldots k}_{1\text{-}3\text{-}2}(x) = x^k C(x) + xG^{12\ldots k}_{1\text{-}3\text{-}2}(x)C(x).$$

Thus, $G^{12\ldots k}_{1\text{-}3\text{-}2}(x) = x^k C(x)/(1 - xC(x))$ and, using (1), we get the desired result. □

Proposition 5. *We have*

$$G^{k(k-1)\ldots 1}_{1\text{-}3\text{-}2}(x) = x^k C^{k+1}(x).$$

Proof. Suppose $\pi = \pi' n \pi'' \in S_n$(1-3-2) is such that $\pi_1 > \pi_2 > \cdots > \pi_k$ and $\pi_j = n$. It is easy to see that π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \ldots, n-1$, and $\pi'' \in S_{n-j}$(1-3-2). If we consider separately the cases $j = 1$ and $j \geq 2$, we get

$$G^{k(k-1)\ldots 1}_{1\text{-}3\text{-}2}(x) = xG^{(k-1)(k-2)\ldots 1}_{1\text{-}3\text{-}2}(x) + xG^{k(k-1)\ldots 1}_{1\text{-}3\text{-}2}(x)C(x).$$

Hence,

$$G^{k(k-1)\ldots 1}_{1\text{-}3\text{-}2}(x) = xG^{(k-1)(k-2)\ldots 1}_{1\text{-}3\text{-}2}(x)/(1 - xC(x))$$

and, using (1), we get $G^{k(k-1)\ldots 1}_{1\text{-}3\text{-}2}(x) = xC(x)G^{(k-1)(k-2)\ldots 1}_{1\text{-}3\text{-}2}(x)$. By induction on k, using the fact that $G^{1}_{1\text{-}3\text{-}2}(x) = C(x) - 1 = xC^2(x)$, we get the desired result. □

Proposition 6. *We have*

$$G^{12\ldots k}_{2\text{-}1\text{-}3}(x) = x^k C^{k+1}(x).$$

Proof. One can use the same considerations as we have in the proof of Proposition 5, by considering a permutation $\pi = \pi' 1 \pi'' \in S_n$(2-1-3) such that $\pi_1 < \pi_2 < \cdots < \pi_k$ and $\pi_j = 1$. □

Proposition 7. *We have*

$$G^{k(k-1)\ldots 1}_{2\text{-}1\text{-}3}(x) = x^k C^2(x).$$

Proof. One can use the same considerations as we have in the proof of Proposition 4, by considering a permutation $\pi = \pi' 1 \pi'' \in S_n$(2-1-3) such that $\pi_1 > \pi_2 > \cdots > \pi_k$ and $\pi_j = 1$. □

Let $s_n(i_1, \ldots, i_m)$ denote the number of permutations $\pi \in S_n$(1-2-3) such that $\pi_1 \pi_2 \ldots \pi_m = i_1 i_2 \ldots i_m$. It is easy to see that

$$s_n(n) = s_n(n-1) = C_{n-1}, \tag{8}$$

and for $1 \leq t \leq n-2$,

$$s_n(t) = s_n(t, n) + \sum_{j=1}^{t-1} s_n(t, j) = s_{n-1}(t) + \sum_{j=1}^{t-1} s_{n-1}(j). \tag{9}$$

Now, (8) and (9) with induction on t give

$$s_n(n-t) = \sum_{j=0}^{t} (-1)^j \binom{t-j}{j} C_{n-j-1} \tag{10}$$

Let us prove the following proposition.

Proposition 8. *We have*

$$G^{12\ldots k}_{1\text{-}2\text{-}3}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ x^2C^2(x), & \text{if } k = 2, \\ xC^2(x), & \text{if } k = 1\ . \end{cases}$$

Proof. For $k \geq 3$, the statement is obviously true. If $k = 1$ then

$$G^{1}_{1\text{-}2\text{-}3}(x) = C(x) - 1 = xC^2(x).$$

Suppose now that $k = 2$. From the definitions, for all $n \geq 2$, we have

$$N^{12}_{1\text{-}2\text{-}3}(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s_n(i,j).$$

In this formula, j can only be equal to n, since otherwise we have an occurrence of the pattern 1-2-3. Using this fact with (8) and (9), we get for $n \geq 2$,

$$N^{12}_{1\text{-}2\text{-}3}(n) = \sum_{i=1}^{n-1} s_n(i,n) = \sum_{i=1}^{n-1} s_{n-1}(i) = C_{n-1}.$$

Hence, $G^{12}_{1\text{-}2\text{-}3}(x) = x(C(x)-1) = x^2C^2(x)$. □

Proposition 9. *We have*

$$N^{k(k-1)\ldots 1}_{1\text{-}2\text{-}3}(n) = \sum_{t=1}^{n+1-k} \binom{n-t}{k-1} \sum_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j-1}.$$

Proof. From the definitions, we have

$$N^{k(k-1)\ldots 1}_{1\text{-}2\text{-}3}(n) = \sum_{t=1}^{n+1-k} \binom{n-t}{k-1} s_n(t).$$

Using (10), we get

$$N^{k(k-1)\ldots 1}_{1\text{-}2\text{-}3}(n) = \sum_{t=1}^{n+1-k} \binom{n-t}{k-1} \sum_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j-1}.$$

□

5. Avoiding a pattern x-y-z, beginning and ending with certain patterns simultaneously

Recall that according to the definitions from Section 2, $N_p^{q,r}(n)$ denotes the number of permutations that avoid the pattern p, begin with the pattern q and end with the pattern r; $G_p^{q,r}(x)$ (respectively, $E_p^{q,r}(x)$) denotes the ordinary (respectively, exponential) generating function for the number of such permutations.

Proposition 10. *Let $k, \ell \geq 1$ and $m = max(k, \ell)$. We have*

(i) $G_{1\text{-}3\text{-}2}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) = x^{k+\ell-1}C^2(x)$.

(ii) $G_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots \ell}(x) = x^{k+\ell-1}C^{\ell+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(iii) $G_{1\text{-}3\text{-}2}^{k(k-1)\ldots 1,\ell(\ell-1)\ldots 1}(x) = x^{k+\ell-1}C^{k+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(iv) *the generating function* $G_{1\text{-}3\text{-}2}(x,y,z) = \sum_{k,\ell\geq 0} G_{1\text{-}3\text{-}2}^{k(k-1)\ldots 1,12\ldots \ell}(x) y^k z^\ell$

for the sequence $\{G_{1\text{-}3\text{-}2}^{k(k-1)\ldots 1,12\ldots \ell}(x)\}_{k,\ell\geq 0}$ *(where k and ℓ go through all natural numbers) is*

$$\frac{1}{1-x(y+z)}\left(x(y+z+yz) + \frac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right).$$

Proof.

(i) **Beginning with $12\ldots k$ and ending with $\ell(\ell-1)\ldots 1$:** Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 < \pi_2 < \cdots < \pi_k$, $\pi_n < \pi_{n-1} < \cdots < \pi_{n-\ell+1}$ and $\pi_j = n$. It is easy to see that π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \ldots, n-1$, and $\pi'' \in S_{n-j}(1\text{-}3\text{-}2)$. We now consider three cases, namely $j = k$, $k+1 \leq j \leq n-\ell$ and $j = n-\ell+1$. In terms of generating functions, we have

$$G_{1\text{-}3\text{-}2}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) = x^k G_{2\text{-}1\text{-}3}^{\ell(\ell-1)\ldots 1}(x) + xG_{1\text{-}3\text{-}2}^{12\ldots k}(x) G_{2\text{-}1\text{-}3}^{\ell(\ell-1)\ldots 1}(x) + x^\ell G_{1\text{-}3\text{-}2}^{12\ldots k}(x) + x^{k+\ell-1},$$

where we observed that to avoid 1-3-2 and end with $\ell(\ell-1)\ldots 1$ is the same as to avoid 2-1-3 and begin with $\ell(\ell-1)\ldots 1$ by applying the reverse and complement operations. Also, we added the term $x^{k+\ell-1}$, since when $j = k = n-\ell+1$, we have one "good" $(k+\ell-1)$-permutation, which is not counted by our three cases.

From Propositions 4 and 7, we have that

$$G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C^2(x) \text{ and } G_{2\text{-}1\text{-}3}^{\ell(\ell-1)\ldots 1}(x) = x^\ell C^2(x).$$

Thus, using the fact that $xC^2(x) = C(x) - 1$, we get

$$\begin{aligned} G_{1\text{-}3\text{-}2}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) &= x^{k+\ell}C^2(x)(2 + xC^2(x)) + x^{k+\ell-1} \\ &= x^{k+\ell-1}(C(x)-1)(C(x)+1) + x^{k+\ell-1} \\ &= x^{k+\ell-1}C^2(x). \end{aligned}$$

(ii) **Beginning with $12\dots k$ and ending with $12\dots\ell$:** Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 < \pi_2 < \cdots < \pi_k$, $\pi_n > \pi_{n-1} > \cdots > \pi_{n-\ell+1}$ and $\pi_j = n$. As above, π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \dots, n-1$, and $\pi'' \in S_{n-j}(1\text{-}3\text{-}2)$. We consider the cases $j = k$, $k+1 \le j \le n-\ell$ and $j = n$. In terms of generating functions, the first approximation for the function $G_{1\text{-}3\text{-}2}^{12\dots k,12\dots\ell}(x)$ is

$$G_{1\text{-}3\text{-}2}^{12\dots k,12\dots\ell}(x) \approx x^k G_{2\text{-}1\text{-}3}^{12\dots\ell}(x) + xG_{1\text{-}3\text{-}2}^{12\dots k}(x)G_{2\text{-}1\text{-}3}^{12\dots\ell}(x) + xG_{1\text{-}3\text{-}2}^{12\dots k,12\dots(\ell-1)}(x),$$

where we observed that to avoid 1-3-2 and end with $12\dots\ell$ is the same as to avoid 2-1-3 and begin with $12\dots\ell$ by applying the reverse and complement operations. We use the sign "$\approx$" because there are some "good" permutations, which are not counted by our considerations. We discuss them below.

From Propositions 4 and 6, we have that $G_{1\text{-}3\text{-}2}^{12\dots k}(x) = x^k C^2(x)$ and $G_{2\text{-}1\text{-}3}^{12\dots\ell}(x) = x^\ell C^{\ell+1}(x)$. Thus, using the fact that $xC^2(x) = C(x) - 1$ and $G_{1\text{-}3\text{-}2}^{12\dots k,1}(x) = G_{1\text{-}3\text{-}2}^{12\dots k}(x) = x^k C^2(x)$ (Proposition 4), we get

$$\begin{aligned}
&G_{1\text{-}3\text{-}2}^{12\dots k,12\dots\ell}(x)\\
&\approx x^{k+\ell}C^{\ell+1}(x) + x^{k+\ell+1}C^{\ell+3}(x) + xG_{1\text{-}3\text{-}2}^{12\dots k,12\dots(\ell-1)}(x)\\
&= x^{k+\ell}C^{\ell+2}(x) + xG_{1\text{-}3\text{-}2}^{12\dots k,12\dots(\ell-1)}(x)\\
&= x^{k+\ell}C^{\ell+2}(x) + x^{k+\ell}C^{\ell+1}(x) + x^2G_{1\text{-}3\text{-}2}^{12\dots k,12\dots(\ell-2)}(x)\\
&= \cdots = x^{k+\ell}C^4(x)(C^{\ell-2}(x) + C^{\ell-3}(x) + \cdots + 1) + x^{k+\ell-1}C^2(x)\\
&= x^{k+\ell-1}(C(x)-1)C^2(x)\tfrac{1-C^{\ell-1}(x)}{1-C(x)} + x^{k+\ell-1}C^2(x)\\
&= x^{k+\ell-1}C^{\ell+1}(x).
\end{aligned}$$

To complete the proof of this case, we observe that in our considerations above, we do not count increasing permutations of length $m = max(k,\ell)$, $m+1, \dots, k+\ell-2$, which satisfy all our restrictions. We did not count them because the k-beginning and ℓ-ending in these permutations overlap in more than one letter. So, to get the desired result, we need to add the term

$$x^m + x^{m+1} + \ldots + x^{k+\ell-2} = (x^m - x^{k+\ell-1})/(1-x)$$

to the approximate value of $G_{1\text{-}3\text{-}2}^{12\dots k,12\dots\ell}(x)$. For example, expanding the ordinary generating function $G_{1\text{-}3\text{-}2}^{12,123}(x)$, we have, in particular, that there are 2002 10-permutations that avoid 1-3-2, begin with the pattern 12 and end with the pattern 123.

(iii) **Beginning with $k(k-1)\dots1$ and ending with $\ell(\ell-1)\dots1$:** If $\ell = 1$ then, by Proposition 5, $G_{1\text{-}3\text{-}2}^{k(k-1)\dots1,1}(x) = x^k C^{k+1}(x)$. Suppose $\ell \geq 2$, and $\pi = \pi' 1 \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 > \pi_2 > \cdots > \pi_k$, $\pi_n < \pi_{n-1} < \cdots < \pi_{n-\ell+1}$ and $\pi_j = 1$. Obviously, π'' is the empty word, since otherwise we have an occurrence of the pattern 1-3-2 starting from the letter 1. Thus, the first approximation for the function $G_{1\text{-}3\text{-}2}^{k(k-1)\dots1,\ell(\ell-1)\dots1}$ is

$$G_{1\text{-}3\text{-}2}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x) \approx xG_{1\text{-}3\text{-}2}^{k(k-1)\dots1,(\ell-1)(\ell-2)\dots1}(x) = \cdots = x^{k+\ell-1}C^{k+1}(x).$$

Like in the previous case, we did not count decreasing permutations of length m, $m+1, \dots, k+\ell-2$, which satisfy all our restrictions. Thus, to get the desired result, we add the term $(x^m - x^{k+\ell-1})/(1-x)$ to the approximate value of $G_{1\text{-}3\text{-}2}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)$.

(iv) **Beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$:** Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$. Any letter of π' is greater than any letter of π'', since otherwise we have an occurrence of the pattern 1-3-2 in π containing the letter n which is forbidden. Also, π' and π'' avoid 1-3-2. If π begins with $k(k-1)\dots1$, ends with $12\dots\ell$ and π' and π'' are not empty, then π' must begin with $k(k-1)\dots1$ and π'' must end with $12\dots\ell$. If π' is empty then π'' must begin with $(k-1)(k-2)\dots1$ and end with $12\dots\ell$. If π'' is empty then π' must begin with $k(k-1)\dots1$ and end with $12\dots(\ell-1)$. In terms of generating functions, the discussion above leads to the following:

$$G_{1\text{-}3\text{-}2}^{k(k-1)\dots1,12\dots\ell}(x) \approx xG_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x)G_{2\text{-}1\text{-}3}^{12\dots\ell}(x) + xG_{1\text{-}3\text{-}2}^{(k-1)\dots1,12\dots\ell}(x) + xG_{1\text{-}3\text{-}2}^{k(k-1)\dots1,12\dots(\ell-1)}(x),$$

where we observed that to avoid 1-3-2 and end with $12\dots\ell$ is the same as to avoid 2-1-3 and begin with $12\dots\ell$. However, to put the sign "=" instead of "$\approx$", we have to correct the right-hand side of the recurrence relation by observing that when either $k = 1$ and $\ell = 0$, or $k = 0$ and $\ell = 1$, or $k = 1$ and $\ell = 1$, the formula does not count the permutation $\pi = 1$ which satisfies all the conditions needed. Thus, if we make correction of the right-hand side, then multiply both parts of the obtained equality by $x^k y^\ell$ and sum over all natural k and ℓ we get (recall the definition of $G_{1\text{-}3\text{-}2}(x,y,z)$ in the statement of the theorem):

$$G_{1\text{-}3\text{-}2}(x,y,z) = x\sum_{k,\ell\geq 0} G_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x)G_{2\text{-}1\text{-}3}^{12\dots\ell}(x)y^k z^\ell + x(y+z)G_{1\text{-}3\text{-}2}(x,y,z) + x(y+z+yz).$$

From Propositions 5 and 6, $G_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x)G_{2\text{-}1\text{-}3}^{12\dots\ell}(x) = x^{k+\ell}C^{k+\ell+2}(x)$, and thus

$$\begin{aligned}
&G_{1\text{-}3\text{-}2}(x,y,z)\\
&= \tfrac{1}{1-x(y+z)}\left(x(y+z+yz) + \sum_{k,\ell\geq 0} x^{k+\ell}C^{k+\ell+2}(x)y^k z^\ell\right)\\
&= \tfrac{1}{1-x(y+z)}\left(x(y+z+yz) + zC^2(z)\sum_{k\geq 0}(xyC(x))^k \sum_{\ell\geq 0}(xzC(x))^\ell\right)\\
&= \tfrac{1}{1-x(y+z)}\left(x(y+z+yz) + \tfrac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right),
\end{aligned}$$

where we used that $xC^2(x) = C(x) - 1$. □

Proposition 11. *Let $k, \ell \geq 1$ and $m = max(k,\ell)$. We have*

(i) $G_{2\text{-}1\text{-}3}^{12\dots k,12\dots\ell}(x) = x^{k+\ell-1}C^{k+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(ii) $G_{2-1-3}^{k(k-1)\dots1,12\dots\ell}(x) = x^{k+\ell-1}C^2(x)$.

(iii) $G_{2\text{-}1\text{-}3}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x) = x^{k+\ell-1}C^{\ell+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(iv) *the generating function* $G_{2\text{-}1\text{-}3}(x,y,z) = \sum_{k,\ell\geq 0} G_{2\text{-}1\text{-}3}^{12\dots k,\ell(\ell-1)\dots1}(x)y^k z^\ell$

for the sequence $\{G_{2\text{-}1\text{-}3}^{12\dots k,\ell(\ell-1)\dots1}(x)\}_{k,\ell\geq 0}$ *(where k and ℓ go through all natural numbers) is*

$$\frac{1}{1-x(y+z)}\left(x(y+z+yz) + \frac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right).$$

Proof. We apply the reverse and complement operations and then use the results of Proposition 10. For example, to avoid 2-1-3, begin with $12\dots k$ and end with $12\dots\ell$ is the same as to avoid 1-3-2, begin with $12\dots\ell$ and end with $12\dots k$. □

Let $h_n^{k,\ell}(t;s)$ denote the number of 1-2-3-avoiding n-permutations such that $\pi_k = t$, $\pi_{n-\ell+1} = s$, $\pi_1 > \pi_2 > \cdots > \pi_k$, and $\pi_{n-\ell+1} > \pi_{n-\ell+2} > \cdots > \pi_n$. Also, we define $g_n(i_1, i_2, \dots, i_m; b)$ to be the number of 1-2-3-avoiding n-permutations such that $\pi_1\pi_2\cdots\pi_m = i_1 i_2 \dots i_m$ and $\pi_n = b$. We need the following two lemmas to prove Proposition 14.

Lemma 12. *For all $n \geq 2$, $g_n(a;b)$ is given by*

$$\begin{cases}
0, & 2 \leq a+1 < b \leq n,\\
\binom{n-2}{a-1}, & 1 \leq a \leq n-1,\ b = a+1\\
\binom{n+a-b-3}{a-2} - \binom{n+a-b-3}{a-b-2}, & 2 \leq b < a \leq n.\\
\binom{n+a-5}{a-2} - \binom{n+a-5}{a-4}, & 1 \leq b < a \leq n.
\end{cases}$$

Proof. By definitions we have

(1) $g_n(a;b) = 0$ for all $2 \le a+1 < b \le n$;

(2) $g_n(a;a+1) = g_n(a,1;a+1) + \ldots + g_n(a,a-1;a+1) + g_n(a,a+2;a+1) + \ldots + g_n(a,n-1;a+1) + g_n(a,n;a+1)$. Using the fact that no there exists a permutation $\pi \in S_n$(1-2-3) such that $\pi_1 = a$, $\pi_2 \le a-2$, and $\pi_n = a+1$ we get

$$g_n(a;a+1) = g_n(a,a-1;a+1) + g_n(a,a+2;a+1) + \ldots + g_n(a,n;a+1).$$

Using the fact that no there exists a permutation $\pi \in S_n$(1-2-3) such that $\pi_1 = a$ and $a \le \pi_2 \le n-1$ we get $g_n(a;a+1) = g_n(a,a-1;a+1) + g_n(a,n;a+1)$. On the other hand, it is easy to see that $g_n(a,a-1;a+1) = g_{n-1}(a-1;a)$ and $g_n(a,n;a+1) = g_{n-1}(a;a+1)$. Hence,

$$g_n(a;a+1) = g_{n-1}(a-1;a) + g_{n-1}(a;a+1).$$

Using induction we get that $g_n(a;a+1) = \binom{n-2}{a-1}$ for all $n \ge 2$ and $1 \le a \le n-1$.

(3) Using Equation (10) we get

$$g_n(a;1) = g_n(a;2) = s_{n-1}(a-1) = \sum_{j=0}^{n-a} (-1)^j \binom{n-a-j}{j} C_{n-2-j}.$$

Similarly as (2) we have for all $a > b$,

$$g_n(a;b) = g_{n-1}(b-1;b) + g_{n-1}(b+1;b) + g_{n-1}(b+2;b) + \cdots + g_{n-1}(a;b).$$

Using the above equation together with induction on n, a, b, we get the desired result. □

Lemma 13. *The number $h_n^{k,\ell}(t;s)$ is given by*

$$\begin{cases} \binom{n-t}{k-1}\binom{s-1}{\ell-1} g_{n+2-k-\ell}(t-(\ell-1);s-(\ell-1)), & \text{if } 1 \le s < t \le n; \\ h_n^{k,\ell}(t+1;t), & \text{if } s = t+1; \\ h_{n-1}^{k,\ell-1}(t;s-1) + h_{n-1}^{k-1;\ell}(t;s-1), & \text{if } 2 \le t+1 < s \le n. \end{cases}$$

Proof. (1) Let $n \ge t > s \ge 1$; so by definitions we get

$$h_n^{k,\ell}(t;s) = \binom{n-t}{k-1}\binom{s-1}{\ell-1} g_{n-(k-1)-(\ell-1)}(t-(\ell-1);s-(\ell-1)).$$

(2) Let $s = t+1$; so it is easy to see $h_n^{k,\ell}(t;t+1) = h_n^{k,\ell}(t+1;t)$;

(3) Let $2 \le t+1 < s \le n$. Let π be any permutations in S_n(1-2-3) such that $\pi_k = t$ and $\pi_{n+1-\ell} = s$ where $\pi_1 > \cdots > \pi_k$ and $\pi_{n+1-\ell} > \cdots > \pi_n$; so there two possibilities either $\pi_{n+2-\ell} = s-1$ or $\pi_j = s-1$ where $j \le k-1$. In this first case we get that there exist $h_{n-1}^{k,\ell-1}(t;s-1)$ permutations, and in the second case we have that there exist $h_{n-1}^{k-1;\ell}(t;s-1)$ permutations. (We extend the number $h_n^{k,l}(a;b)$ as 0 for any $\ell \le 0$ or $k \le 0$). □

We recall that the Kronecker delta $\delta_{n,k}$ is defined to be

$$\delta_{n,k} = \begin{cases} 1, & \text{if } n = k, \\ 0, & \text{else.} \end{cases}$$

Proposition 14. *We have*

(i) $G_{1\text{-}2\text{-}3}^{12\ldots k,12\ldots\ell}(x) = \begin{cases} 0, & \text{if } k \geq 3 \text{ or } \ell \geq 3 \\ xC^2(x), & \text{if } k = 1 \text{ and } \ell = 1 \end{cases}$,

$N_{1\text{-}2\text{-}3}^{12,12}(n) = \begin{cases} 0, & \text{if } n = 3 \\ C_{n-2}, & \text{else} \end{cases}$, *and* $N_{1\text{-}2\text{-}3}^{12,1}(n) = N_{1-2-3}^{1,12}(n) = C_{n-1}$.

(ii) *The number* $N_{1\text{-}2\text{-}3}^{k(k-1)\ldots1,12\ldots\ell}(n)$ *is given by*

$$\begin{cases} 0, & \text{if } \ell \geq 3, \\ \sum\limits_{t=1}^{n-k} \binom{n-t-1}{k-1} \sum\limits_{j=0}^{n-t-1} (-1)^j \binom{n-t-j-1}{j} C_{n-t-j-1} + (k-1)\delta_{n,k+1}, & \text{if } \ell = 2, \\ \sum\limits_{t=1}^{n+1-k} \binom{n-t}{k-1} \sum\limits_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j-1}, & \text{if } \ell = 1. \end{cases}$$

(iii) *The number* $N_{1\text{-}2\text{-}3}^{12\ldots k,\ell(\ell-1)\ldots1}(n)$ *is given by*

$$\begin{cases} 0, & \text{if } k \geq 3, \\ \sum\limits_{t=1}^{n-\ell} \binom{n-t-1}{\ell-1} \sum\limits_{j=0}^{n-t-1} (-1)^j \binom{n-t-j-1}{j} C_{n-t-j-1} + (\ell-1)\delta_{n,\ell+1}, & \text{if } k = 2, \\ \sum\limits_{t=1}^{n+1-\ell} \binom{n-t}{\ell-1} \sum\limits_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j-1}, & \text{if } k = 1. \end{cases}$$

(iv) $N_{1-2-3}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) = \sum_{t=1}^{n-k+1} \sum_{s=\ell}^{n} h_n^{k,\ell}(t;s)$, *where* $h_n^{k,\ell}(t;s)$ *is given in Lemma 13.*

Proof.

(i) **Beginning with** $12\ldots k$ **and ending with** $12\ldots\ell$**:** If $k \geq 3$ or $\ell \geq 3$, the statement is obvious, since in this case $12\ldots k$ or $12\ldots\ell$ does not avoid the pattern 1-2-3. If $k = 1$ or $\ell = 1$, we get the statement from Proposition 8 (in the first of these cases we apply the reverse and complement operations). Suppose now that $k = 2$, $\ell = 2$, and an n-permutation π avoids 1-2-3, begins with the pattern 12 and ends with the pattern 12. The letter n must be next to the leftmost letter, since otherwise two leftmost letters and n form the pattern 1-2-3. Also, the letter 1 must be next to the rightmost letter, since otherwise 1 and the two rightmost letters form the pattern 1-2-3. It is easy to see now that there are C_{n-2} possibilities to choose π, since we can take any 1-2-3-avoiding permutation on the letters $\{2, 3, \ldots, n-1\}$ (there are C_{n-2} such permutations), then let the letters n and 1 be in the second and $(n-1)$-st positions respectively. These considerations only fail when $n = 3$, since in this case the second and $(n-1)$-st positions coincide. However, in this case we obviously have no permutations with the good properties.

(ii) **Beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$:** The statement is true for $\ell \geq 3$, since in this case $12\dots\ell$ does not avoid 1-2-3. For the case $\ell = 1$ we use Proposition 9. Suppose now that $\ell = 2$, and an n-permutation π avoids 1-2-3, begins with the pattern $k(k-1)\dots1$ and ends with the pattern 12. The letter 1 must be next to the rightmost letter, since otherwise 1 and two rightmost letters form the pattern 1-2-3. So, to form π we can take any $(n-1)$-permutation on the letters $\{2,3,\dots,n\}$ that avoids 1-2-3 and begins with the pattern $k(k-1)\dots1$ (the number of such permutations is given by Proposition 9), and then let the letter 1 be in the $(n-1)$-st position. Also, we observe that in the case $n = k+1$ we have $k-1$ extra permutations, which are obtained from the $(n-1)$-permutations having the $k-1$ leftmost letters in decreasing order and two rightmost letters in increasing order.

(iii) **Beginning with $12\dots k$ and ending with $\ell(\ell-1)\dots1$:** By the reverse and complement operations, to avoid 1-2-3, begin with the pattern $12\dots k$ and end with the pattern $\ell(\ell-1)\dots1$ is the same as to avoid 1-2-3, begin with the pattern $\ell(\ell-1)\dots1$ and end with the pattern $12\dots k$, so we can apply the results of the previous case.

(iv) **Beginning with $k(k-1)\dots1$ and ending with $\ell(\ell-1)\dots1$:** The statement is immediate from the definitions of $N_{1\text{-}2\text{-}3}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(n)$ and $h_n^{k,\ell}(t,s)$. □

6. Avoiding a pattern xyz, beginning and ending with certain patterns simultaneously

Recall that according to Section 2, $E_p^{q,r}(x)$ denotes the exponential generating function for the number of permutations that avoid the pattern p, begin with the pattern q and end with the pattern r.

Proposition 15. *We have*

(i)

$$E_{213}^{12\dots k,12\dots\ell}(x) = \begin{cases} E_{132}^{12\dots\ell}(x), & \text{if } k=1 \\ E_{213}^{12\dots k}(x), & \text{if } \ell=1 \end{cases},$$

where $E_{132}^{12\dots\ell}(x)$ and $E_{213}^{12\dots k}(x)$ are given in Table 1(K1-K3) and Table 1(K5) respectively. For $k,\ell \geq 2$, $E_{213}^{12\dots k,12\dots\ell}(x)$ satisfies

$$\frac{\partial}{\partial x}E_{213}^{12\dots k,12\dots\ell}(x) = E_{213}^{12\dots k,12\dots(\ell-1)}(x) + \left(E_{213}^{12\dots k,12}(x) + \frac{x^{k-1}}{(k-1)!}\right)E_{132}^{12\dots\ell}(x).$$

(ii)

$$E_{213}^{12\dots k,\ell(\ell-1)\dots1}(x) = \begin{cases} E_{132}^{\ell(\ell-1)\dots1}(x), & \text{if } k=1 \\ E_{213}^{12\dots k}(x), & \text{if } \ell=1 \end{cases},$$

where $E_{132}^{\ell(\ell-1)\dots1}(x)$ *and* $E_{213}^{12\dots k}(x)$ *are given in Table 1(K4) and (K5) respectively. For* $k,\ell\geq 2$, $E_{213}^{12\dots k,\ell(\ell-1)\dots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{213}^{12\dots k,\ell(\ell-1)\dots1}(x)=\frac{x^{\ell-1}}{(\ell-1)!}E_{213}^{12\dots k}(x)+\left(E_{213}^{12\dots k,12}(x)+\frac{x^{k-1}}{(k-1)!}\right)E_{132}^{\ell(\ell-1)\dots1}(x)+\binom{k+\ell-2}{k-1}\frac{x^{k+\ell-2}}{(k+\ell-2)!}.$$

(iii)

$$E_{213}^{k(k-1)\dots1,12\dots\ell}(x)=\begin{cases}E_{132}^{12\dots\ell}(x), & \text{if } k=1\\ E_{213}^{k(k-1)\dots1}(x), & \text{if } \ell=1\end{cases},$$

Formula	Eq.
$E_{132}^{1}(x)=\frac{1}{1-\int_0^x e^{-t^2/2}\,dt}$	K1
$E_{132}^{12}(x)=\frac{e^{-x^2/2}}{1-\int_0^x e^{-t^2/2}\,dt}-x-1$	K2
$E_{132}^{12\dots k}(x)=E_{132}^{1}(x)\cdot$ $\cdot\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_2}\left(e^{-t_1^2/2}-\frac{t_1+1}{E_{132}^{1}(t_1)}\right)dt_2\cdots dt_{k-2}dt_1$	K3
$E_{132}^{k(k-1)\dots1}(x)=\frac{E_{132}^{1}(x)}{(k-1)!}\int_0^x t^{k-1}e^{-t^2/2}\,dt.$	K4
$E_{213}^{12\dots k}(x)=\int_0^x\int_0^t\frac{s^{k-2}e^{T(t)-T(s)}}{(k-2)!(1-\int_0^t e^{-m^2/2}dm)}\,dsdt$ where $T(x)=-x^2/2+\int_0^x\frac{e^{-t^2/2}}{1-\int_0^t e^{-s^2/2}ds}\,dt$	K5
$E_{213}^{k(k-1)\dots1}(x)=-\frac{x^{k-1}}{(k-1)!}+\sum_{n=0}^{k-2}\int_0^x\int_0^{t_n}\cdots\int_0^{t_1}\frac{C_{k-n}(t)+\delta_{n,k-2}}{1-\int_0^t e^{-m^2/2}dm}dtdt_1\cdots dt_n$ where $C_k(x)=e^{T(x)}\cdot\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_1}e^{-T(t)}\left(\frac{e^{-t^2/2}}{1-\int_0^t e^{-m^2/2}dm}-t-1\right)dtdt_1\cdots dt_{k-2}$ with $T(x)$ which is given in K5.	K6

Table 1. [Kit3, Equation 12, Theorem 6(i),6(ii),7,10,11].

where $E_{132}^{12\dots\ell}(x)$ *and* $E_{213}^{k(k-1)\dots1}(x)$ *are given in Table 1(K1-K3) and (K6) respectively. For* $k,\ell\geq 2$, $E_{213}^{k(k-1)\dots1,12\dots\ell}(x)$ *satisfies*

$$\begin{aligned}\tfrac{\partial}{\partial x}E_{213}^{k(k-1)\dots1,12\dots\ell}(x) &= E_{213}^{(k-1)\dots1,12\dots\ell}(x)\\ &\quad +E_{213}^{k(k-1)\dots1,12}(x)E_{132}^{12\dots\ell}(x)+E_{213}^{k(k-1)\dots1,12\dots(\ell-1)}(x).\end{aligned}$$

(iv)

$$E_{213}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)=\begin{cases}E_{132}^{\ell(\ell-1)\dots1}(x), & \text{if } k=1\\ E_{213}^{k(k-1)\dots1}(x), & \text{if } \ell=1\end{cases},$$

where $E_{132}^{\ell(\ell-1)\dots1}(x)$ *and* $E_{213}^{k(k-1)\dots1}(x)$ *are given in Table 1(K4) and (K6) respectively. For* $k,\ell\geq 2$, $E_{213}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)$ *satisfies*

$$\begin{aligned}&\tfrac{\partial}{\partial x}E_{213}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)\\ &\quad=E_{213}^{(k-1)\dots1,\ell(\ell-1)\dots1}(x)+\left(E_{132}^{\ell(\ell-1)\dots1}(x)+\tfrac{x^{\ell-1}}{(\ell-1)!}\right)E_{213}^{k(k-1)\dots1,12}(x).\end{aligned}$$

Proof.

(ii) **Beginning with** $12\dots k$ **and ending with** $\ell(\ell-1)\dots1$**:** The statement is obviously true when $k=1$ and $\ell=1$. Suppose now that $k\geq 2$, $\ell\geq 2$ and an $(n+1)$-permutation π avoids 213, begins with the pattern $12\dots k$ and ends with the pattern $12\dots\ell$. The letter $(n+1)$ can only be in the position k, or in the position i, where $(k+1)\leq i\leq n-\ell+1$, or in the position $n-\ell+2$. In the first case, we choose the $(k-1)$ leftmost letters in $\binom{n}{k-1}$ ways, rearrange them into the increasing order, and observe, that the letters of π to the right of $(n+1)$ must form an $(n-k+1)$-permutation, that avoids 213 and ends with the pattern $\ell(\ell-1)\dots1$ (the number of such permutations, using the reverse and complement operation, is equal to the number of $(n-k+1)$-permutations that avoid 132 and begin with the pattern $\ell(\ell-1)\dots1$). In the third case, we choose the $(\ell-1)$ rightmost letters in $\binom{n}{\ell-1}$ ways, rearrange them into the decreasing order, and observe, that the letters of π to the left of $(n+1)$ must form an $(n-\ell+1)$-permutation, that avoids 213, begins with the pattern $12\dots k$, and ends with the pattern 12 (if it ends with the pattern 21, the letter $(n+1)$ and two letters immediately to the left of it form the pattern 213). In the second case, we choose the letters of π to the left of $(n+1)$ in $\binom{n}{i-1}$ ways and observe, that these letters must form a $(i-1)$-permutation that avoids 213, begins with the pattern $12\dots k$ and ends with the pattern 12. At the same time, the letters to the right of $(n+1)$ must form an $(n-i+2)$-permutation that avoids 213 and ends with the pattern $\ell(\ell-1)\dots1$. Besides, we observe that if $n=k+\ell-2$, that is $|\pi|=k+\ell-1$, and first k-letters of π are rearranged into the increasing order, whereas the last ℓ letters are rearranged in the decreasing order, we have a number of extra "good" permutations. The

number of such permutations is the number of ways of choosing the first $(k-1)$ letters, that is $\binom{k+\ell-2}{k-1}$. This discussion leads to the following:

$$\begin{aligned} N_{213}^{12\ldots k,\ell(\ell-1)\ldots 1}(n+1) \\ &= \tbinom{n}{k-1} N_{132}^{\ell(\ell-1)\ldots 1}(n-k+1) + \tbinom{n}{\ell-1} N_{213}^{12\ldots k}(n-\ell+1) \\ &+ \sum_{i=0}^{n} \binom{n}{i} N_{213}^{12\ldots k,12}(i) N_{132}^{\ell(\ell-1)\ldots 1}(n-i) + \binom{k+\ell-2}{k-1} \delta_{n,k+\ell-2}, \end{aligned}$$

where $\delta_{n,k+\ell-2}$ is the Kronecker delta. We get the desired result by multiplying both sides of the last equality by $x^n/n!$ and summing over n.

(i) **Beginning with $12\ldots k$ and ending with $12\ldots\ell$:** The statement is obviously true when $k=1$ and $\ell=1$. Suppose now that $k\geq 2$, $\ell\geq 2$ and an $(n+1)$-permutation π avoids 213, begins with the pattern $12\ldots k$ and ends with the pattern $12\ldots\ell$. The letter $(n+1)$ can only be in the position k, or in the position i, where $(k+1)\leq i\leq n-\ell$, or in the $(n+1)$-th position. In the last case, the number of such permutations is obviously $N_{213}^{12\ldots k,12\ldots\ell-1}(n)$. In the first case, we choose the $(k-1)$ leftmost letters in $\binom{n}{k-1}$ ways, rearrange them into increasing order, and observe, that the letters of π to the right of $(n+1)$ must form an $(n-k+1)$-permutation, that avoids 213 and ends with the pattern $12\ldots\ell$ (the number of such permutations, using the reverse and complement operation, is equal to the number of $(n-k+1)$-permutations that avoid 132 and begin with the pattern $12\ldots\ell$). In the second case, we choose the letters of π to the left of $(n+1)$ in $\binom{n}{i-1}$ ways and observe, that these letters must form a $(i-1)$-permutation that avoids 213, begins with the pattern $12\ldots k$ and ends with the pattern 12 (if it ends with the pattern 21, the letter $(n+1)$ and two letters immediately to the left of it form the pattern 213). At the same time, the letters to the right of $(n+1)$ must form an $(n-i+2)$-permutation that avoids 213 and ends with the pattern $12\ldots\ell$. This discussion leads to the following:

$$\begin{aligned} N_{213}^{12\ldots k,12\ldots\ell}(n+1) &= N_{213}^{12\ldots k,12\ldots\ell-1}(n) \\ &+ \sum_{i=0}^{n} \binom{n}{i} N_{213}^{12\ldots k,12}(i) N_{132}^{12\ldots\ell}(n-i) + \binom{n}{k-1} N_{132}^{12\ldots\ell}(n-k+1). \end{aligned}$$

We get the desired result by multiplying both sides of the last equality by $x^n/n!$ and summing over n.

(iii,iv) **Beginning with $k(k-1)\ldots 1$ and ending with $12\ldots\ell$ or with $\ell(\ell-1)\ldots 1$:** We proceed in the same way as we do under considering the previous case. □

We observe that the number of permutations that avoid the pattern 132, begin with the pattern p and end with the pattern r is equal to the number of permutations that avoid the pattern 213, begin with the pattern r' and

end with the pattern p', where p' and r' are obtained from p and r by applying the composition of the reverse and complement operations. Thus,

$$E_{132}^{p,r}(x) = E_{213}^{C\circ R(r),C\circ R(p)}(x).$$

Proposition 16. *We define* $\Theta_k(x)$ *to be*

$$\int_0^x \sec(\Psi_6(t))\left(\sin(\Psi_3(t)) - \frac{\sqrt{3}}{2}e^{-t/2}\right)\left(\Phi_k(t) + \frac{t^{k-1}}{(k-1)!}\right)\,dt,$$

where

$$\Phi_k(x) = \frac{e^{x/2}}{(k-1)!}\sec(\Psi_6(x))\int_0^x e^{-t/2}t^{\ell-1}\sin(\Psi_3(t))\,dt,$$

and $\Psi_k(x) = \frac{\sqrt{3}}{2}x + \frac{\pi}{k}$. *We have*

(i) $E_{123}^{12\ldots k,12\ldots\ell}(x) =$

$$= \begin{cases} 0, & \text{if } k \geq 3 \text{ or } \ell \geq 3,\\ x - \frac{1}{2} - \frac{\sqrt{3}}{2}\tan(\Psi_6(x)) + & \\ \quad \sec(\Psi_6(x))\left(\frac{\sqrt{3}}{2}\left(e^{x/2}+e^{-x/2}\right) - \sin(\Psi_3(x))\right), & \text{if } k = 2 \text{ and } \ell = 2,\\ \frac{\sqrt{3}}{2}e^{x/2}\sec(\Psi_6(x0) - 1, & \text{if } k = 1 \text{ and } \ell = 1,\\ \frac{\sqrt{3}}{2}e^{x/2}\sec(\Psi_6(x)) - \frac{1}{2} - \frac{\sqrt{3}}{2}\tan(\Psi_6(x)), & \text{otherwise;} \end{cases}$$

(ii) $E_{123}^{12\ldots k,\ell(\ell-1)\ldots1}(x) = \begin{cases} 0, & \text{if } k \geq 3,\\ \Phi_\ell(x), & \text{if } k = 1,\\ \Theta_\ell(x), & \text{if } k = 2; \end{cases}$

(iii) $E_{123}^{k(k-1)\ldots1,12\ldots\ell}(x) = \begin{cases} 0, & \text{if } \ell \geq 3,\\ \Phi_k(x), & \text{if } \ell = 1,\\ \Theta_k(x), & \text{if } \ell = 2; \end{cases}$

(iv) $E_{123}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$ *is given by*

$$\begin{cases} E_{123}^{\ell(\ell-1)\ldots1}(x), & \text{if } k = 1,\\ E_{123}^{k(k-1)\ldots1}(x), & \text{if } \ell = 1,\\ E_{123}^{k(k-1)\ldots1}(x) - E_{123}^{k(k-1)\ldots1,12}(x), & \text{if } \ell = 2; \end{cases}$$

For $k \geq 2$ *and* $\ell \geq 3$, $E_{123}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{123}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$$
$$= \left(E_{123}^{\ell(\ell-1)\ldots1}(x) + \frac{x^{\ell-1}}{(\ell-1)!}\right)E_{123}^{k(k-1)\ldots1,21}(x) + E_{123}^{(k-1)\ldots1,\ell(\ell-1)\ldots1}(x),$$

where $E_{123}^{k(k-1)\dots1}(x)$ *is given in* [KitMans, Theorem 2]*:*

$$E_{123}^{k(k-1)\dots1}(x) = \frac{e^{x/2}\int_0^x e^{-t/2}t^{k-1}\sin(\Psi_6(t))\ dt}{(k-1)!\cos(\Psi_6(x))},$$

and $E_{123}^{k(k-1)\dots1,12}$ *is given in this theorem above.*

Proof.

(iii) **Beginning with** $k(k-1)\dots1$ **and ending with** $12\dots\ell$**:** If $\ell \geq 3$ then the pattern $12\dots\ell$ does not avoid 123, thus the statement is true. If $\ell = 1$, the statement is true according to [Kit3, Theorem 8] and the observation that if $k = 1$ then this formula gives the expression

$$\frac{\sqrt{3}}{2}e^{x/2}\sec(\Psi_6(x)) - 1,$$

which is true according to [ElizNoy, Theorem 4.1] and the assumption that the empty permutation does not begin or end with the pattern $p = 1$. So, we need only to consider the case $\ell = 2$.

Let $P_k(n)$ denote the number of n-permutations that avoid the pattern 123, begin with a decreasing subword of length k and end with the pattern 12. Also, let $R_k(n)$ denote the number of n-permutations that avoid the pattern 123 and begin with a decreasing subword of length k. Let $\pi = \pi_1 1 \pi_2$ be an $(n+1)$-permutation that avoids the pattern 123, begins with the pattern $k(k-1)\dots1$ and ends with the pattern 12. We observe that π_1 avoids 123 and begins with $k(k-1)\dots1$; π_2 ends with the pattern 12 and $|\pi_2| > 0$ since otherwise π cannot end with the pattern 12; if $|\pi_2| > 1$ then π_2 must begin with the pattern 21 since otherwise we have an occurrence of the pattern 123 beginning from the letter 1. If $|\pi_1| = i$ then the letters of π_1 can be chosen in $\binom{n}{i}$ ways. So, there are at least

$$\sum_{i\geq 0}\binom{n}{i}R_k(i)P_2(n-i) + nR_k(n-1)$$

$(n+1)$-permutations with the good properties, where the first term corresponds to the case $|\pi_2| > 1$ and the second term to the case $|\pi_2| = 1$. By this formula, we do not count the permutations having $|\pi_1| = k-1$, although in this case π begins with the pattern $k(k-1)\dots1$. So, we can choose the letters of π_1 in $\binom{n}{k-1}$ ways, and according to whether $|\pi_2| \geq 1$ or $|\pi_2| = 1$, we have two terms:

$$\binom{n}{k-1}P_2(n-k+1) + k\delta_{n,k},$$

where $\delta_{n,k}$ is the Kronecker delta. Thus,

$$P_k(n+1) = \sum_{i\geq 0}\tbinom{n}{i}R_k(i)P_2(n-i) + nR_k(n-1) + \tbinom{n}{k-1}P_2(n-k+1) + k\delta_{n,k}.$$

After multiplying both sides of the last equality with $x^n/n!$ and summing over n, we have

$$\frac{d}{dx}E_{123}^{k(k-1)\dots1,12}(x) = (E_{123}^{21,12}(x)+x)\left(E_{123}^{k(k-1)\dots1}(x)+\frac{x^{k-1}}{(k-1)!}\right), \tag{11}$$

with the initial condition $E_{123}^{k(k-1)\dots1,12}(0)=0$. Since

$$\begin{aligned} E_{123}^{k(k-1)\dots1}(x) &= E_{123}^{k(k-1)\dots1,1}(x) \\ &= \tfrac{e^{x/2}}{(k-1)!}\sec(\Psi_6(x))\int_0^x e^{-t/2}t^{k-1}\sin(\Psi_3(t))\,dt, \end{aligned}$$

to solve (11), we only need to know $E_{123}^{21,12}(x)$. To find it, we set $k=2$ into (11) and solve this equation. For an example how to solve such an equation, we refer to Table 1(K1-K3). We get

$$E_{123}^{21,12}(x) = -x+\sec(\Psi_6(x))e^{-x/2}\int_0^x e^{t/2}\cos(\Psi_6(t))\,dt.$$

Now, we put the formula for $E_{123}^{21,12}(x)$ into (11) and solve this differential equation to get the desired result.

(ii) **Beginning with $12\dots k$ and ending with $\ell(\ell-1)\dots1$:** By the reverse and complement operations, to avoid 123, begin with the pattern $12\dots k$ and end with the pattern $\ell(\ell-1)\dots1$ is the same as to avoid 123, begin with the pattern $\ell(\ell-1)\dots1$ and end with the pattern $12\dots k$, so we can apply the results of the previous case.

(i) **Beginning with $12\dots k$ and ending with $12\dots\ell$:** The statement is obvious if $k\geq 3$ or $\ell\geq 3$. If $k=1$ and $\ell=1$ then the statement is true according to [ElizNoy, Theorem 4.1] (but we need to subtract 1, since by our assumption the empty permutation does not begin or end with the pattern $p=1$). If $\ell=1$ and $k=2$, the statement is true according [Kit3, Theorem 9]. If $k=1$ and $\ell=2$, we apply the reverse and complement operations, and use again [Kit3, Theorem 9]. So, we only need to consider the case $k=2$ and $\ell=2$. It is easy to see that

$$E_{123}^{12,12}(x) = E_{123}^{1,12}(x) - E_{123}^{21,12}(x),$$

and from the previous cases

$$E_{123}^{1,12}(x) = \frac{\sqrt{3}}{2}e^{x/2}\sec(\Psi_6(x)) - \frac{1}{2} - \frac{\sqrt{3}}{2}\tan(\Psi_6(x)),$$

and

$$E_{123}^{21,12}(x) = -x+\sec(\Psi_6(x))\left(\sin(\Psi_3(x)) - \frac{\sqrt{3}}{2}e^{-x/2}\right).$$

(iv) **Beginning with $k(k-1)\dots1$ and ending with $\ell(\ell-1)\dots1$:** If $\ell=1$, the statement is trivial. If $k=1$, we get the statement by using the reverse and complement operations. For the case $\ell=2$, we observe that the number of n-permutations that avoid the pattern 123, begin with the

pattern $k(k-1)\dots1$ and end with the pattern 21 is equal to the number of n-permutation that avoid 123 and begin with the pattern $k(k-1)\dots1$ minus the number of n-permutations that avoid the pattern 123, begin with the pattern $k(k-1)\dots1$ and end with the pattern 12. Suppose now that $k\geq 2$ and $\ell\geq 3$ and an $(n+1)$-permutation π avoids 123, begins with $k(k-1)\dots1$ and ends with $\ell(\ell-1)\dots1$. It is easy to see that the letter $(n+1)$ can be either in the first position, or in the position i, where $(k+1)\leq i\leq(n-\ell)$, or in the position $(n-\ell+1)$. In the first of these cases, obviously we have $N_{123}^{(k-1)\dots1,\ell(\ell-1)\dots1}(n)$ permutations. In the second case, we choose the letters of π to the left of $(n+1)$ in $\binom{n}{i-1}$ ways. These letters must form a permutation that avoids 123, begins with the pattern $k(k-1)\dots1$, and ends with the pattern 21 (if the last condition does not hold, the letter $(n+1)$ and two letters to the left of it form a 123-pattern. At the same time, the letters to the right of $(n+1)$ form a permutation that avoids 123 and ends with the pattern $\ell(\ell-1)\dots1$. In the third case, we can choose the letters to the right of $(n+1)$ in $\binom{n}{\ell-1}$ ways, rearrange them into the decreasing order, and form from the letters to the left of $(n+1)$ a permutation that avoids 123, begins with the pattern $k(k-1)\dots1$ and ends with the pattern 21 (by the same reasons as above) in $N_{123}^{k(k-1)\dots1,21}(n-\ell+1)$ ways. Thus,

$$\begin{aligned}&N_{123}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(n+1)\\&\quad=N_{123}^{(k-1)\dots1,\ell(\ell-1)\dots1}(n)+\sum_{i=0}^{n}\binom{n}{i}N_{123}^{k(k-1)\dots1,21}(i)N_{123}^{\ell(\ell-1)\dots1}(n-i)\\&\quad+\tbinom{n}{\ell-1}N_{123}^{k(k-1)\dots1,21}(n-\ell+1),\end{aligned}$$

where we observed, that to avoid 123 and end with $\ell(\ell-1)\dots1$ is the same as to avoid 123 and begin with $\ell(\ell-1)\dots1$ using the reverse and complement. Now, we multiply both sides of the equality by $x^n/n!$ and sum over n to get the desired result. □

7. Avoiding a pattern x-yz, beginning and ending with certain patterns simultaneously

Proposition 17. *We have*

$$\text{(i) } E_{1\text{-}32}^{12\dots k,1}(x)=E_{1\text{-}32}^{12\dots k}(x)=\begin{cases}e^{e^x}\int_0^x e^{-e^t}\sum\limits_{n\geq k-1}\dfrac{t^n}{n!}\,dt, & \text{if } k\geq 2\\ e^{e^x-1}, & \text{if } k=1\end{cases}.$$

For $\ell\geq 2$, $E_{1\text{-}32}^{12\dots k,12\dots\ell}(x)$ satisfies

$$\frac{\partial}{\partial x}E_{1\text{-}32}^{12\dots k,12\dots\ell}(x)=\left(e^x-\sum_{i=0}^{\ell-2}\frac{x^i}{i!}\right)E_{1\text{-}32}^{12\dots k}(x)+e^x x^{max(\ell,k)-1}.$$

(ii) $E_{1\text{-}32}^{12\dots k,\ell(\ell-1)\dots1}(x)$ *satisfies*

$$\frac{\partial^{\ell-1}}{\partial x^{\ell-1}}E_{1\text{-}32}^{12\dots k,\ell(\ell-1)\dots1}(x)=\begin{cases} e^{e^x}\int_0^x e^{-e^t}\sum\limits_{n\geq k-1}\dfrac{t^n}{n!}\,dt, & \text{if } k\geq 2,\\ e^{e^x-1}, & \text{if } k=1.\end{cases}$$

(iii) *the generating functions* $E_{1\text{-}32}^{k(k-1)\dots1,1}(x)$ *and* $E_{1\text{-}32}^{k(k-1)\dots1}(x)$ *are given by*

$$\begin{cases} (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k\geq 2\\ e^{e^x-1}, & \text{if } k=1\end{cases}.$$

For $\ell\geq 2$, $E_{1\text{-}32}^{k(k-1)\dots1,12\dots\ell}(x)$ *satisfies*

$$\begin{aligned}&\tfrac{\partial}{\partial x}E_{1\text{-}32}^{k(k-1)\dots1,12\dots\ell}(x)\\ &\qquad=\left(e^x-\sum_{i=0}^{\ell-2}\frac{x^i}{i!}\right)E_{1\text{-}32}^{k(k-1)\dots1}(x)+\left(e^x-\sum_{i=0}^{\ell-2}\frac{x^i}{i!}\right)\tfrac{x^{k-1}}{(k-1)!}.\end{aligned}$$

(iv) $E_{1\text{-}32}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)$ *satisfies*

$$\begin{aligned}&\tfrac{\partial^{\ell-1}}{\partial x^{\ell-1}}\left(E_{1\text{-}32}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)-\tfrac{x^{max(k,\ell)}-x^{k+\ell-1}}{1-x}\right)\\ &\qquad=\begin{cases}\frac{e^{e^x}}{(k-1)!}\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k\geq 2,\\ e^{e^x-1}, & \text{if } k=1.\end{cases}\end{aligned}$$

Proof.

(ii) **Beginning with** $12\dots k$ **and ending with** $\ell(\ell-1)\dots1$**:** If $\ell=1$ then the result follows from [KitMans, Proposition 5], since to avoid 1-32 and begin with $12\dots k$ is the same as to avoid 3-12 and begin with $k(k-1)\dots1$. Suppose now that $\ell\geq 2$ and a permutation π avoids the pattern 1-32, begins with the pattern $12\dots k$ and ends with the pattern $\ell(\ell-1)\dots1$. Since $\ell\geq 2$, we have that the letter 1 must be in the rightmost position since otherwise, this letter and two rightmost letters of π form the pattern 1-32, which is forbidden. Thus,

$$N_{1\text{-}32}^{12\dots k,\ell(\ell-1)\dots1}(n)=N_{1\text{-}32}^{12\dots k,(\ell-1)(\ell-2)\dots1}(n-1)=\cdots=N_{1\text{-}32}^{12\dots k,1}(n-\ell+1).$$

Multiplying both sides of the equality

$$N_{1\text{-}32}^{12\dots k,\ell(\ell-1)\dots1}(n)=N_{1\text{-}32}^{12\dots k,1}(n-\ell+1)$$

by $x^{n-\ell+1}/(n-\ell+1)!$ and summing over n, we get

$$\frac{\partial^{\ell-1}}{\partial x^{\ell-1}}E_{1\text{-}32}^{12\dots k,\ell(\ell-1)\dots1}(x)=E_{1\text{-}32}^{12\dots k}(x),$$

where $E_{1\text{-}32}^{12\dots k}(x)$ is given in [KitMans, Proposition 5], since to avoid 1-32 and begin with $12\dots k$ is the same as to avoid 3-12 and begin with $k(k-1)\dots1$.

(iv) **Beginning with $k(k-1)\dots1$ and ending with $\ell(\ell-1)\dots1$:** We use the same arguments as those given under consideration of the previous case, but instead of [KitMans, Proposition 5] we use [KitMans, Proposition 4]. However, we observe, that when we use the argument

$$N_{1\text{-}32}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(n) = N_{1\text{-}32}^{k(k-1)\dots1,(\ell-1)(\ell-2)\dots1}(n-1) = \dots = N_{1\text{-}32}^{k(k-1)\dots1,1}(n-\ell+1)$$

for $k,\ell \geq 2$, we do not count the decreasing permutations of length $max(k,\ell)$, $max(k,\ell)+1,\dots,k+\ell-2$, since in this case, the patterns $k(k-1)\dots1$ and $\ell(\ell-1)\dots1$ overlap in more than one letter, which causes the observation. So, we need to consider additionally the term

$$x^{max(k,\ell)} + x^{max(k,\ell)+1} + \dots + x^{k+\ell-2} = \frac{x^{max(k,\ell)} - x^{k+\ell-1}}{1-x},$$

which vanishes if $k=1$ or $\ell=1$.

(i) **Beginning with $12\dots k$ and ending with $12\dots\ell$:** The only interesting case here is the case $k\geq 2$ and $\ell\geq 2$. Using the reverse and complement, instead of considering avoiding 1-32, beginning with $12\dots k$ and ending with $12\dots\ell$, we consider avoiding 21-3, beginning with $12\dots\ell$ and ending with $12\dots k$. Suppose an n-permutation π satisfies all the conditions. We observe, that the letter n can be in the position i, where $\ell\leq i\leq n-k$. Also, n can be in the rightmost position if $n\geq max(\ell,k)$. In any case, the letters of π to the left of n must be in increasing order, since otherwise we have an occurrence of the pattern 21-3. This means that in the second case we have the only one permutation. In the first case, the letters of π to the right of n must avoid 21-3 and end with the pattern $12\dots k$. The number of such permutations, using the reverse and complement, is given by $N_{1\text{-}32}^{12\dots k}(n-i)$. Thus, for $n\geq max(\ell,k)$,

$$N_{21\text{-}3}^{12\dots\ell,12\dots k}(n) = \sum_{i=\ell}^{n-k}\binom{n-1}{i-1}N_{1\text{-}32}^{12\dots k}(n-i)+1.$$

This gives

$$N_{21\text{-}3}^{12\dots\ell,12\dots k}(n) = \sum_{i=1}^{n}\binom{n-1}{i-1}N_{1\text{-}32}^{12\dots k}(n-i) - \sum_{i=1}^{\ell-1}\binom{n-1}{i-1}N_{1\text{-}32}^{12\dots k}(n-i)+1,$$

which leads to the desired result after multiplying both sides of the last equality by $x^n/n!$ and summing over n.

(iii) **Beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$:** The only interesting case here is the case $k\geq 2$ and $\ell\geq 2$. Using the reverse and complement, instead of considering avoiding 1-32, beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$, we consider avoiding 21-3, beginning with $12\dots\ell$ and ending with $k(k-1)\dots1$. Suppose an n-permutation π satisfies all the conditions. We observe, that the letter n can only be in the position

i, where $\ell \le i \le n-k$, or in position $(n-k+1)$ (in the case $n \ge k+\ell-1$). In the first case, it is easy to see that the letters of π to the left of n must be in the increasing order, and the letters of π to the right of n must avoid 21-3 and end with the pattern $k(k-1)\dots 1$. Using the reverse and complement, the total number of permutations counted in the first case is $\sum_{i=\ell}^{n-k} \binom{n-1}{i-1} N_{1\text{-}32}^{k(k-1)\dots 1}(n-i)$. In the second case, the letters to the left of n are in increasing order, whereas the letters to the right of n are in decreasing order. The number of such permutations is $\binom{n-1}{k-1}$, which is the number of ways to choose the last $k-1$ letters. Thus,

$$N_{21\text{-}3}^{12\dots\ell,k(k-1)\dots 1}(n) = \sum_{i=\ell}^{n-k} \binom{n-1}{i-1} N_{1\text{-}32}^{k(k-1)\dots 1}(n-i) + \binom{n-1}{k-1}.$$

Multiplying both parts of the equality by $x^{n-1}/(n-1)!$ and summing over n, we get

$$\frac{\partial}{\partial x} E_{21\text{-}3}^{12\dots\ell,k(k-1)\dots 1}(x) = \sum_{n\ge k+\ell} \binom{n-1}{k-1} \frac{x^{n-1}}{(n-1)!} +$$
$$\sum_{n\ge 0} \left(\sum_{i=1}^{n-1} \binom{n-1}{i-1} N_{1\text{-}32}^{k(k-1)\dots 1}(n-i) - \sum_{i=1}^{\ell-1} \binom{n-1}{i-1} N_{1\text{-}32}^{k(k-1)\dots 1}(n-i) \right) \frac{x^{n-1}}{(n-1)!},$$

which leads to the desired result. □

Proposition 18. *Let $k, \ell \ge 1$ and $m = max(k,\ell)$. We have*

(i) $G_{2\text{-}13}^{12\dots k,12\dots\ell}(x) = x^{k+\ell-1}C^{k+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(ii) $G_{2\text{-}13}^{k(k-1)\dots 1,12\dots\ell}(x) = x^{k+\ell-1}C^2(x)$.

(iii) $G_{2\text{-}13}^{k(k-1)\dots 1,\ell(\ell-1)\dots 1}(x) = x^{k+\ell-1}C^{\ell+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(iv) *the generating function* $G_{2\text{-}13}(x,y,z) = \sum_{k,\ell\ge 0} G_{2\text{-}13}^{12\dots k,\ell(\ell-1)\dots 1}(x) y^k z^\ell$

for the sequence $\{G_{2\text{-}13}^{12\dots k,\ell(\ell-1)\dots 1}(x)\}_{k,\ell\ge 0}$ (*where k and ℓ go through all natural numbers*) *is*

$$\frac{1}{1-x(y+z)} \left(x(y+z+yz) + \frac{C(x)-1}{(1-xyC(x))(1-xzC(x))} \right).$$

Proof. By [Claes, Lemma 2], to avoid the pattern 2-13 is the same as to avoid the pattern 2-1-3. Thus we can apply the results of Proposition 11. □

Proposition 19. *We have*

(i) $E_{1\text{-}23}^{12\dots k,12\dots\ell}(x) = \begin{cases} 0, & \text{if } k \ge 3 \text{ or } \ell \ge 3, \\ E_{1\text{-}23}^{12\dots k}(x), & \text{if } \ell = 1, \\ E_{12\text{-}3}^{12\dots\ell}(x), & \text{if } k = 1, \\ \int_0^x t E_{12\text{-}3}^{12}(t)\, dt + \frac{x^2}{2!}, & \text{if } k = 2 \text{ and } \ell = 2, \end{cases}$

where $E_{12\text{-}3}^{12\dots k}(x)$ and $E_{1\text{-}23}^{12\dots k}(x)$ are given by [KitMans, Proposition 10] *and* [KitMans, Proposition 6] *respectively:*

$$E_{12\text{-}3}^{12\ldots k}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ x^2 \displaystyle\sum_{j=0}^{k} \frac{1}{1-jx} \sum_{d\geq 0} \frac{x^d}{\prod_{s=0}^{d}(1-sx)}, & \text{if } k = 2, \\ \displaystyle\sum_{d\geq 0} \frac{x^d}{\prod_{s=0}^{d}(1-sx)}, & \text{if } k = 1; \end{cases}$$

$$E_{1\text{-}23}^{12\ldots k}(x) = E_{3-21}^{k(k-1)\ldots 1}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ e^{e^x} \int_0^x e^{-e^t}(e^t-1)\, dt, & \text{if } k = 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

(ii) $N_{1\text{-}23}^{12\ldots k,\ell(\ell-1)\ldots 1}(n) =$

$$\begin{cases} 0, & \text{if } k \geq 3, \\ 0, & \text{if } k = 2 \text{ and } n \leq \ell, \\ 1 + N_{1\text{-}23}^{12,(\ell-1)(\ell-2)\ldots 1}(n-1) & \\ \qquad + \displaystyle\sum_{j=\ell+1}^{n-2} \binom{n-1}{j-1} N_{1\text{-}23}^{12}(n-j), & \text{if } k = 2 \text{ and } n \geq \ell+1, \\ N_{12\text{-}3}^{\ell(\ell-1)\ldots 1}(n), & \text{if } k = 1, \end{cases}$$

where the numbers $N_{12\text{-}3}^{\ell(\ell-1)\ldots 1}(n)$ *are given in* [KitMans, Proposition 9]*, and the numbers* $N_{1\text{-}23}^{12}(n)$ *are given by expanding the exponential generating functions in* [KitMans, Proposition 6].

(iii) *the exponential generating function* $E_{1\text{-}23}^{k(k-1)\ldots 1,12\ldots\ell}(x)$ *is given by*

$$\begin{cases} 0, & \text{if } \ell \geq 3 \\ \frac{1}{(k-1)!} \int_0^x \int_0^t t m^{k-1} e^{e^t - e^m + m}\, dm dt + \frac{k x^{k+1}}{(k+1)!}, & \text{if } \ell = 2 \\ (e^{e^x}/(k-1)!) \int_0^x t^{k-1} e^{-e^t+t}\, dt, & \text{if } \ell = 1 \end{cases},$$

where $E_{1\text{-}23}^{k(k-1)\ldots 1,1}(n) = E_{1\text{-}23}^{k(k-1)\ldots 1}(n)$ *is given by* [KitMans, Proposition 4]*, and* $N_{1\text{-}23}^{1,\ell(\ell-1)\ldots 1}(n) = N_{12\text{-}3}^{\ell(\ell-1)\ldots 1}(n)$ *is given by* [KitMans, Proposition 9]*;*

(iv) *For* $k \geq 2$ *and* $\ell \geq 2$, $E_{1\text{-}23}^{k(k-1)\ldots 1,\ell(\ell-1)\ldots 1}(x)$ *satisfies*

$$\frac{\partial}{\partial x} E_{1\text{-}23}^{k(k-1)\ldots 1,\ell(\ell-1)\ldots 1}(x) = E_{1\text{-}23}^{k(k-1)\ldots 1,(\ell-1)\ldots 1}(x) + \left(e^x - \sum_{i=0}^{\ell-1} \frac{x^i}{i!}\right)\left(E_{1\text{-}23}^{k(k-1)\ldots 1}(x) + \frac{x^k}{(k-1)!}\right).$$

Proof.

(iii) **Beginning with** $k(k-1)\ldots 1$ **and ending with** $12\ldots\ell$**:** If $\ell \geq 3$ then $E_{1\text{-}23}^{k(k-1)\ldots 1,12\ldots\ell}(x) = 0$, since in this case the pattern $12\ldots\ell$ does not avoid 1-23. If $\ell = 1$ then we use [KitMans, Proposition 4], since in this case the only restrictions to the permutations are avoiding 1-23 and beginning

with the pattern $k(k-1)\dots1$. Suppose now that $\ell = 2$ and an $(n+1)$-permutation π avoids 1-23, begins with $k(k-1)\dots1$ and ends with the pattern 12. The letter 1 must be in next to the rightmost position, since otherwise this letter and two rightmost letters form the pattern 1-23. We can choose the rightmost letter of π in n ways, and the letters to the left of 1 must form a 1-23-avoiding permutation that begins with $k(k-1)\dots1$. Besides, if $n = k$, and the $k-1$ letters to the left of 1 are in the decreasing order, we get n extra permutations that satisfy our restrictions. Thus,

$$N_{1\text{-}23}^{k(k-1)\dots1,12}(n+1) = nN_{1\text{-}23}^{k(k-1)\dots1}(n) + n\delta_{n,k},$$

where $\delta_{n,k}$ is the Kronecker delta. Multiplying both sides of the equality by $x^n/n!$ and summing over n we get

$$E_{1\text{-}23}^{k(k-1)\dots1,12}(x) = \int_0^x tE_{1\text{-}23}^{k(k-1)\dots1}(t)\ dt + \frac{kx^{k+1}}{(k+1)!}.$$

Using the formula for $E_{1\text{-}23}^{k(k-1)\dots1}(t)$ in [KitMans, Proposition 4], we get the desired result.

(i) **Beginning with $12\dots k$ and ending with $12\dots\ell$:** The first three cases are easy to prove in the same manner as we do in the proves of previous propositions. The only interesting case is when $k = 2$ and $\ell = 2$. Using the reverse and complement operations, instead of considering avoiding 1-23, beginning with 12 and ending with 12, we consider avoiding 12-3, beginning with 12 and ending with 12, which we find to be more easy. Suppose an $(n+1)$-permutation π satisfies all the restrictions. It is easy to see that $|\pi| \neq 1$ and $|\pi| \neq 3$, as well as if $|\pi| = 2$ (that is $n = 1$) then π must be 12. Suppose $|\pi| \geq 4$. Since π begins with the pattern 12, it is impossible for the letter $(n+1)$ to be somewhere to the right of the second letter of π or to be the leftmost letter. Thus, $(n+1)$ must be in the second position. We can choose the leftmost letter of π in n ways, since any choice of this letter will not lead to an occurrence of the pattern $12-3$ beginning with two leftmost letters. If $\pi = a(n+1)\pi'$ then π' must avoid 12-3 and end with the pattern 12. The number of such permutations, using the reverse and complement, is given by $N_{1\text{-}23}^{12}(n-1)$. Thus,

$$N_{12\text{-}3}^{12,12}(n+1) = nN_{1\text{-}23}^{12}(n-1).$$

Multiplying both sides of the equality by $x^n/n!$ and summing over all n, we get

$$(E_{12\text{-}3}^{12,12}(x))' = xE_{1\text{-}23}^{12}(x) + x,$$

where the term x corresponds to the permutation 12. We have the desired result by integrating both sides of the last equality.

(ii) **Beginning with $12\dots k$ and ending with $\ell(\ell-1)\dots1$:** All the cases but $k = 2$ and $n \geq \ell + 1$ are easy to prove. Let us consider this case. Using the reverse and complement operations, instead of considering

avoiding 1-23, beginning with 12 and ending with $\ell(\ell-1)\dots 1$, we consider avoiding 12-3, beginning with $\ell(\ell-1)\dots 1$ and ending with 12, which we find to be easier. Let an n-permutation π satisfy all the conditions. We observe, that the letter n is either in the first position, or in position j, where $k+1 \le j \le n-2$, or in the last position. Obviously, in the first of these cases the number of "good" permutations is given by $N_{12\text{-}3}^{(\ell-1)(\ell-2)\dots 1,12}(n-1)$, which is equivalent to $N_{1\text{-}23}^{12,(\ell-1)(\ell-2)\dots 1}(n-1)$ by using the reverse and complement. In the second case, we choose the letters to the left of n in $\binom{n-1}{j-1}$ ways, rearrange them to the decreasing order (we do it since otherwise we have an occurrence of the pattern 12-3 having the letter n). After that, the letters to the right of n must form a permutation that avoid 12-3 and end with the pattern 12. Using the reverse and complement, there are $N_{1\text{-}23}^{12}(n-j)$ such permutations. So, totally, in the second case there are $\sum_{j=\ell+1}^{n-2}\binom{n-1}{j-1}N_{1\text{-}23}^{12}(n-j)$ permutations. Finally, if n is at the last position, we have the only one such permutation, since the other letters must be in the decreasing order.

(iv) **Beginning with $k(k-1)\dots 1$ and ending with $\ell(\ell-1)\dots 1$:** The only interesting case here is the case $k \ge 2$ and $\ell \ge 2$. Using the reverse and complement operations, instead of considering avoiding 1-23, beginning with $k(k-1)\dots 1$ and ending with $\ell(\ell-1)\dots 1$, we consider avoiding 12-3, beginning with $\ell(\ell-1)\dots 1$ and ending with $k(k-1)\dots 1$, which we find to be more easy. Let an n-permutation π satisfy all the conditions. We observe, that the letter n is either in the first position, or in position j, where $\ell+1 \le j \le n-k$, or in the last position $n-k+1$. We proceed as in the previous case to get the following

$$N_{12\text{-}3}^{\ell(\ell-1)\dots 1,k(k-1)\dots 1} = N_{12\text{-}3}^{(\ell-1)\dots 1,k(k-1)\dots 1} + \sum_{i=\ell+1}^{n-k}\binom{n-1}{i-1}N_{1\text{-}23}^{k(k-1)\dots 1}(n-i) + \binom{n-1}{k-1},$$

where three terms in the right-hand side correspond to the three cases described above. We now multiply both sides of the equality by $x^n/n!$, sum over n and observe the following detail. We cannot write instead of $i=\ell+1$ (in the sum above) $i=1$ as we did in most of the cases above, since, for instance, the case $i=1$ do not necessarily make the term of summation equal 0 as it was before. Thus, instead of the factor e^x, we have the factor

$$\left(e^x - \sum_{i=0}^{\ell-1}\frac{x^i}{i!}\right).$$

□

8. Avoiding a pattern xy-z, beginning and ending with certain patterns simultaneously

To obtain results for the number of permutations that avoid the pattern $xy-z$, begin with the pattern p and end with the pattern r, one can apply the results from Section 7 and subsequently together the composition of the reverse and complement operations.

9. Further results

In this section, we propose two directions of generalization of the results from the previous sections. The first one is a consideration of avoiding more than one pattern, beginning with some pattern and ending with another pattern. For example, suppose that $v = 12\text{-}3$, $w = 21\text{-}3$, $p = 12\dots k$, $q = 12\dots\ell$, and $E_{v,w}^{p,q}(x)$ denotes the exponential generating function for the number of permutations that avoid the patterns v and w simultaneously, begin with the pattern p and end with the pattern q. It is easy to see that if $k \geq 3$ or $\ell \geq 3$ then $E_{12\text{-}3,21\text{-}3}^{12\dots k,12\dots\ell}(x) = 0$. For the other k and ℓ, one can prove the following theorem:

Theorem 20. *We have*

(i) $E_{12\text{-}3,21\text{-}3}^{1,1}(x) = e^{x+x^2/2} - 1$.

(ii) $E_{12\text{-}3,21\text{-}3}^{1,12}(x) = e^{x+x^2/2}\left(1 - \int_0^x e^{-t-t^2/2}dt\right) - 1$.

(iii) $E_{12\text{-}3,21\text{-}3}^{12,1}(x) = \int_0^x te^{t+t^2/2}dt$.

(iv) $E_{12\text{-}3,21\text{-}3}^{12,12}(x) = \frac{1}{2}x^2 + \int_0^x \left[e^{t+t^2/2}\left(1 - \int_0^t e^{-r-r^2/2}dr\right) - 1\right]dt$.

The second direction is a consideration of permutations in S_n containing a pattern v exactly r times, beginning with some pattern and ending with another pattern. For example, suppose that $v = 12\text{-}3$, $r = 1$, $p = 1\dots k$, $q = 1$, and $N_{v;r}^{p,q}(n)$ denotes the number of n-permutations that contain the pattern v exactly r times, begin with the pattern p, and end with the pattern q. It is easy to see that the only interesting case is $1 \leq k \leq 3$, since otherwise $N_{12\text{-}3;1}^{12\dots k,1}(n) = 0$. Moreover, one can prove the following theorem:

Theorem 21. *Let F_n denote the number of n-permutations containing* 12-3 *exactly once. Then, for all $n \geq 3$,*

$$N_{12\text{-}3;1}^{1,1}(n) = F_n N_{12\text{-}3;1}^{12,1}(n) = (n-1)F_{n-1} + (n-2)B_{n-2},$$
$$N_{12\text{-}3;1}^{123,1}(n) = (n-2)B_{n-3},$$

where B_n is the nth Bell number, and F_n is given by [ClaesMans2, Corollary 13].

Acknowledgments. The authors are grateful to the referees for the careful reading of the manuscript.

References

[BabStein] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, *Séminaire Lotharingien de Combinatoire*, B44b:18pp, 2000.

[Bon] M. Bóna: Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A* **80** (1997), no. 2, 257–272.

[B] M. Bóna: The permutation classes equinumerous to the smooth class. *Electron. J. Combin. 5* (1998), no. **1**, Research Paper 31, 12 pp. (electronic).

[CW] T. Chow and J. West: Forbidden subsequences and Chebyshev polynomials. *Discrete Math.* **204** (1999), no. 1-3, 119–128.

[Claes] A. Claesson: Generalised Pattern Avoidance, *European J. Combin.* **22** (2001), 961-971.

[ClaesMans1] A. Claesson and T. Mansour: Enumerating Permutations Avoiding a Pair of Babson-Steingrímsson Patterns, preprint CO/0107044.

[ClaesMans2] A. Claesson and T. Mansour, Counting Occurrences of a Pattern of Type (1,2) or (2,1) in Permutations, *Advances in Applied Mathematics*, to appear (2002).

[ElizNoy] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, *Proceedings of FPSAC 2001*.

[Ent] R. Entringer: A Combinatorial Interpretation of the Euler and Bernoulli Numbers, *Nieuw. Arch. Wisk.* **14** (1966), 241–246.

[Kit1] S. Kitaev: Multi-avoidance of generalised patterns, *Discrete Math.* **260** (2003), 89–100.

[Kit2] S. Kitaev: Partially ordered generalized patterns, *Discrete Math.*, to appear (2002).

[Kit3] S. Kitaev: Generalized pattern avoidance, *Séminaire Lotharingien de Combinatoire* **48** (2003), Article B48e, 19 pp.

[KitMans] S. Kitaev and T. Mansour: Simultaneous avoidance of generalized patterns, *Ars Combinatorica*, to appear, preprint math.CO/0205182.

[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Kr] C. Krattenthaler: Permutations with restricted patterns and Dyck paths, *Adv. in Appl. Math.* **27** (2001), 510–530.

[K] D. Kremer: Permutations with forbidden subsequnces and a generalized Schröder number, *Discrete Math.* **218** (2000), 121–130.

[Loth] M. Lothaire: *Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, **17**, Addison-Wesley Publishing Co., Reading, Mass. (1983).

[Mans1] T. Mansour: Continued fractions and generalized patterns, *European Journal of Combinatorics* **23:3** (2002), 329–344.

[Mans2] T. Mansour: Continued fractions, statistics, and generalized patterns, to appear in Ars Combinatorica (2002), preprint CO/0110040.

[Mans3] T. Mansour: Restricted 1-3-2 permutations and generalized patterns, *Annals of Combinatorics* **6** (2002), 65–76.

[MV1] T. Mansour and A. Vainshtein: Restricted permutations, continued fractions, and Chebyshev polynomials, *Electron. J. Combin.* **7** (2000) no. 1, Research Paper 17, 9 pp. (electronic).

[MV2] T. Mansour and A. Vainshtein: Restricted 132-avoiding permutations, *Adv. in Appl. Math.* **126** (2001), no. 3, 258–269.

[MV3] T. Mansour and A. Vainshtein: Layered restrictions and Chebyshev polynomials, *Annals of Combinatorics* **5** (2001), 451–458.

[MV4] T. Mansour and A. Vainshtein: Restricted permutations and Chebyshev polyonomials, *Séminaire Lotharingien de Combinatoire* **47** (2002), Article B47c.

[R] A. Robertson: Permutations containing and avoiding 123 and 132 patterns, *Discrete Math. Theor. Comput. Sci.* **3** (1999), no. 4, 151–154 (electronic).

[RWZ] A. Robertson, H. Wilf, and D. Zeilberger: Permutation patterns and continued fractions, *Electron. J. Combin.* **6** (1999), no. 1, Research Paper 38, 6 pp. (electronic).

[SloPlo] N. J. A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, (1995).

[Stan] R. Stanley: *Enumerative Combinatorics*, Vol. **1**, Cambridge University Press, (1997).

[SchSim] R. Simion, F. Schmidt: Restricted permutations, *European J. Combin.* **6** (1985), no. 4, 383–406.

[W] J. West: Generating trees and forbidden subsequences, *Discrete Math.* **157** (1996), 363–372.

Matematik, Chalmers tekniska högskola och Göteborgs universitet, 412 96 Göteborg, Sweden

E-mail address: kitaev@math.chalmers.se

Department of mathematics, Chalmers university of technology, 412 96 Göteborg, Sweden

E-mail address: toufik@math.chalmers.se

SIMULTANEOUS AVOIDANCE OF GENERALIZED PATTERNS

SERGEY KITAEV AND TOUFIK MANSOUR

Abstract. In [BabStein] Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. In [Kit1] Kitaev considered simultaneous avoidance (multi-avoidance) of two or more 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. There either an explicit or a recursive formula was given for all but one case of simultaneous avoidance of more than two patterns. In this paper we find the exponential generating function for the remaining case. Also we consider permutations that avoid a pattern of the form $x-yz$ or $xy-z$ and begin with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $23\dots k1$, $(k-1)(k-2)\dots 1k$ or end with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $1k(k-1)\dots 2$, $k12\dots(k-1)$. For each of these cases we find either the ordinary or exponential generating functions or a precise formula for the number of such permutations. Besides we generalize some of the obtained results as well as some of the results given in [Kit3]: we consider permutations avoiding certain generalized 3-patterns and beginning (ending) with an arbitrary pattern having either the greatest or the least letter as its rightmost (leftmost) letter.

1. Introduction and Background

Permutation patterns: All permutations in this paper are written as words $\pi = a_1 a_2 \dots a_n$, where the a_i consist of all the integers $1, 2, \dots, n$. Let $\alpha \in S_n$ and $\tau \in S_k$ be two permutations. We say that α *contains* τ if there exists a subsequence $1 \le i_1 < i_2 < \cdots < i_k \le n$ such that $(\alpha_{i_1}, \dots, \alpha_{i_k})$ is order-isomorphic to τ, that is, for all j and m, $\tau_j < \tau_m$ if and only if $a_{i_j} < a_{i_m}$; in such a context τ is usually called a *pattern*. We say that α *avoids* τ, or is τ-*avoiding*, if α does not contain τ. The set of all τ-avoiding permutations in S_n is denoted by $S_n(\tau)$. For an arbitrary finite collection of patterns T, we say that α avoids T if α avoids each $\tau \in T$; the corresponding subset of S_n is denoted by $S_n(T)$.

While the case of permutations avoiding a single pattern has attracted much attention, the case of multiple pattern avoidance remains less investigated. In particular, it is natural, as the next step, to consider permutations avoiding pairs of patterns τ_1, τ_2. This problem was solved completely

for $\tau_1, \tau_2 \in S_3$ (see [SchSim]), for $\tau_1 \in S_3$ and $\tau_2 \in S_4$ (see [W]), and for $\tau_1, \tau_2 \in S_4$ (see [B, K] and references therein). Several recent papers [CW, MV1, Kr, MV3, MV2] deal with the case $\tau_1 \in S_3$, $\tau_2 \in S_k$ for various pairs τ_1, τ_2.

Generalized permutation patterns: In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns (GPs)* where two adjacent letters in a pattern may be required to be adjacent in the permutation. Such an adjacency requirement is indicated by the absence of a dash between the corresponding letters in the pattern. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, but the pattern 2-3-1 occurs also in the subwords 562 and 563. Note that a classical pattern should, in our notation, have dashes at the beginning and end. Since most of the patterns considered in this paper satisfy this, we suppress these dashes from the notation. Thus, a pattern with no dashes corresponds to a contiguous subword anywhere in a permutation. The motivation for introducing these patterns was the study of Mahonian statistics. A number of results on GPs were obtained by Claesson, Kitaev and Mansour. See for example [Claes], [Kit1, Kit2, Kit3] and [Mans1, Mans2, Mans3].

As in [SchSim], dealing with the classical patterns, one can consider the case when permutations have to avoid two or more generalized patterns simultaneously. A complete solution for the number of permutations avoiding a pair of 3-patterns of type (1,2) or (2,1), that is, the patterns having one internal dash, is given in [ClaesMans1]. In [Kit1] Kitaev gives either an explicit or a recursive formula for all but one case of simultaneous avoidance of more than two patterns. This is the case of avoiding the GPs 123, 231 and 312 simultaneously. In Theorem 1 we find the exponential generating function (e.g.f.) for the number of such permutations.

As it was discussed in [Kit3], if a permutation begins (resp. ends) with the pattern $p = p_1 p_2 \dots p_k$, that is, the k leftmost (resp. rightmost) letters of the permutation form the pattern p, then this is the same as avoidance of $k! - 1$ patterns simultaneously. For example, beginning with the pattern 123 is equivalent to the simultaneous avoidance of the patterns (132], (213], (231], (312] and (321] in the Babson-Steingrímsson notation. Thus demanding that a permutation must begin or end with some pattern, in fact, we are talking about simultaneous avoidance of generalized patterns. The motivation for considering additional restrictions such as beginning or ending with some patterns is their connection to some classes of trees. An example of such a connection can be found in [Kit3, Theorem 5]. There it was shown that there is a bijection between n-permutations avoiding the pattern 132 and beginning with the pattern 12 and *increasing rooted trimmed trees* with $n + 1$ nodes. We recall that a trimmed tree is a tree where no node has a single leaf as a child (every leaf has a sibling) and in an increasing rooted

tree, nodes are numbered and the numbers increase as we move away from the root. The avoidance of a generalized 3-pattern p with no dashes and, at the same time, beginning or ending with an increasing or decreasing pattern was discussed in [Kit3]. Theorem 2 generalizes some of these results to the case of beginning (resp. ending) with an arbitrary pattern avoiding p and having the greatest or least letter as the rightmost (resp. leftmost) letter.

Propositions $4-15$ (resp. $16-27$) give a complete description for the number of permutations avoiding a pattern of the form $x-yz$ or $xy-z$ and beginning with one of the patterns $12\dots k$ or $k(k-1)\dots 1$ (resp. $23\dots k1$ or $(k-1)(k-2)\dots 1k$). For each of these cases we find either the ordinary or exponential generating functions or a precise formula for the number of such permutations. Theorem 27 generalizes some of these results. Besides, the results from Propositions 4–27 give a complete description for the number of permutations that avoid a pattern of the form $x-yz$ or $xy-z$ and end with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $1k(k-1)\dots 2$ and $k12\dots(k-1)$. To get the last one of these we only need to apply the reverse operation discussed in the next section. The results of Theorems 2 and 27 can also be used to get the case of ending with a pattern from the sets Δ_k^{min} or Δ_k^{max} introduced in the next section.

Except for the empty permutation, every permutation ends and begins with the pattern $p=1$. To simplify the discussion we assume that the empty permutation also begin with the pattern 1. This does not course any harm since, to count the generating functions in question for this, we need only subtract 1 from the generating functions obtained in this paper.

2. Preliminaries

The *reverse* $R(\pi)$ of a permutation $\pi = a_1a_2\dots a_n$ is the permutation $a_na_{n-1}\dots a_1$. The *complement* $C(\pi)$ is the permutation $b_1b_2\dots b_n$ where $b_i = n+1-a_i$. Also, $R\circ C$ is the composition of R and C. For example, $R(13254) = 45231$, $C(13254) = 53412$ and $R\circ C(13254) = 21435$. We call these bijections of S_n to itself *trivial*, and it is easy to see that for any pattern p the number $A_p(n)$ of permutations avoiding the pattern p is the same as for the patterns $R(p)$, $C(p)$ and $R\circ C(p)$. For example, the number of permutations that avoid the pattern 132 is the same as the number of permutations that avoid the pattern 231. This property holds for sets of patterns as well. If we apply one of the trivial bijections to all patterns of a set G, then we get a set G' for which $A_{G'}(n)$ is equal to $A_G(n)$. For example, the number of permutations avoiding $\{123, 132\}$ equals the number of those avoiding $\{321, 312\}$ because the second set is obtained from the first one by complementing each pattern.

In this paper we denote the nth Catalan number by C_n; the generating function for these numbers by $C(x)$; the nth Bell number by B_n.

Also, $N_q^p(n)$ denotes the number of permutations that avoid the pattern q and begin with the pattern p; $G_q^p(x)$ (resp. $E_q^p(x)$) denotes the ordinary (resp. exponential) generating function for the number of such permutations. Besides, Γ_k^{min} (resp. Γ_k^{max}) denotes the set of all k-patterns with no dashes such that the least (resp. greatest) letter of a pattern is the rightmost letter; Δ_k^{min} (resp. Δ_k^{max}) denotes the set of all k-patterns with no dashes such that the least (resp. greatest) letter of a pattern is the leftmost letter.

Recall the following properties of $C(x)$:

$$C(x) = \frac{1-\sqrt{1-4x}}{2x} = \frac{1}{1-xC(x)}. \tag{1}$$

3. Simultaneous avoidance of 123, 231 and 312

The *Entringer numbers* $E(n,k)$ (see [SloPlo, Seq. A000111]) are the number of permutations on $1,2,\ldots,n+1$, starting with $k+1$, which, after initially falling, alternately fall then rise. The Entringer numbers (see [Ent]) are given by

$$E(0,0)=1, \qquad E(n,0)=0,$$

together with the recurrence relation

$$E(n,k) = E(n,k-1) + E(n-1,n-k).$$

The numbers $E(n) = E(n,n)$, are the secant and tangent numbers given by the generating function

$$\sec x + \tan x.$$

The following theorem completes the consideration of multi-avoidance of more than two generalized 3-patterns with no dashes made in [Kit1].

Theorem 1. *Let $E(x)$ be the e.g.f. for the number of permutations that avoid* 123, 231 *and* 312 *simultaneously. Then*

$$E(x) = 1 + x(\sec(x) + \tan(x)).$$

Proof. Let $s(n; i_1, \ldots, i_m)$ denote the number of permutations π in the set $S_n(123, 231, 312)$ such that $\pi_1\pi_2\ldots\pi_m = i_1 i_2 \ldots i_m$ and $f : S_n \to S_n$ be a map defined by

$$f(\pi_1\pi_2\ldots\pi_n) = (\pi_1+1)(\pi_2+1)\ldots(\pi_n+1),$$

where the addition is modulo n. Using f one can see that for all $1 \le a \le n-1$,

$$s(n;a) = s(n;a+1). \tag{2}$$

Thus, $|S_n(123,231,312)| = ns(n;1)$ and we only need to prove that $s(n;1) = E_{n-1}$, where E_n is the nth Euler number (see [SloPlo, Seq. A000111]).

Suppose $\pi \in S_n(123, 231, 312)$ is an n-permutation such that $\pi_1 = 1$ and $\pi_2 = t$. Since π avoids 123, we get $\pi_3 \leq t-1$ and it is easy to see that

$$s(n;1,t) = \sum_{j=2}^{t-1} s(n;1,t,j) = \sum_{j=1}^{t-2} s(n-1;t-1,j),$$

so

$$s(n;1,t+1) = s(n;1,t) + \sum_{j=1}^{t-1} s(n-1;t,j) - \sum_{j=1}^{t-2} s(n-1;t-1,j).$$

Using the map f that proves (2) we get

$$\begin{aligned} s(n;1,t+1) = s(n;1,t) &+ s(n-1;t,1) \\ &+ \sum_{j=2}^{t-1} s(n-1;t-1,j-1) - \sum_{j=1}^{t-2} s(n-1;t-1,j), \end{aligned}$$

and by the map f again, we have for all $t = 2, 3, \ldots, n-1$,

$$s(n;1,t+1) = s(n;1,t) + s(n-1;1,n-t+1).$$

Besides, by the definition, it is easy to see that $s(n;1,2) = 0$ for all $n \geq 3$, hence using the definition of Entringer numbers [Ent] we get that $s(n;1) = \sum_{t=2}^{n} s(n;1,t) = E_{n-1}$, as required. □

4. Avoiding a 3-pattern with no dashes and beginning with a pattern whose rightmost letter is the greatest or smallest

The following theorem generalizes Theorems 7 and 8 in [Kit3]. Recall that according to Section 2, $E_q^p(x)$ denotes the exponential generating function for the number of permutations that avoid the pattern q and begin with the pattern p.

Theorem 2. *Suppose $p_1, p_2 \in \Gamma_k^{min}$ and $p_1 \in S_k(132)$, $p_2 \in S_k(123)$. Thus, the complements $C(p_1), C(p_2) \in \Gamma_k^{max}$ and $C(p_1) \in S_k(312)$, $C(p_2) \in S_k(321)$. Then, for $k \geq 2$,*

$$E_{132}^{p_1}(x) = E_{312}^{C(p_1)}(x) = \frac{\int_0^x t^{k-1} e^{-t^2/2}\, dt}{(k-1)!(1 - \int_0^x e^{-t^2/2}\, dt)}$$

and

$$E_{123}^{p_2}(x) = E_{321}^{C(p_2)}(x) = \frac{e^{x/2} \int_0^x e^{-t/2} t^{k-1} \sin(\frac{\sqrt{3}}{2}t + \frac{\pi}{6}))\, dt}{(k-1)! \cos(\frac{\sqrt{3}}{2}x + \frac{\pi}{6})}.$$

Proof. Let $p \in \{p_1, p_2\}$. To prove the theorem, it is enough to copy the proofs of Theorems 7 and 8 in [Kit3], since the fact that the first $k-1$ letters of p are possibly not in decreasing order is immaterial for the proofs of that theorems. Thus we can get the formula for $E_{132}^p(x)$ and $E_{123}^p(x)$, and

automatically, using properties of the complement, the formula for $E^{C(p)}_{312}(x)$ and $E^{C(p)}_{321}(x)$, directly from these theorems. However we give here a proof of the formula for $E^{p}_{132}(x)$ and refer to [Kit3, Theorem 8] for a proof of the formula for $E^{p}_{123}(x)$.

If $k = 1$, we have no additional restrictions, that is, we are dealing only with the avoidance of 132 and, according to [ElizNoy, Theorem 4.1] or [Kit2, Theorem 12],

$$E^{1}_{132}(x) = \frac{1}{1 - \int_0^x e^{-t^2/2}\ dt}.$$

Also, according to [Kit3, Theorem 6],

$$E^{12}_{132}(x) = \frac{e^{-x^2/2}}{1 - \int_0^x e^{-t^2/2}\ dt} - x - 1.$$

Let $R_{n,k}$ (resp. $F_{n,k}$) denote the number of n–permutations that avoid the pattern 132 and begin with a pattern of type p_1 (resp. of type $C(p_1)$) of length $k > 1$ and let π be such a permutation of length $n+1$. Suppose $\pi = \sigma 1 \tau$. If τ is empty then, obviously, there are $R_{n,k}$ ways to choose σ. If $|\tau| = 1$, that is, 1 is in the second position from the right, then there are n ways to choose the rightmost letter in π and we multiply this by $R_{k,n-1}$, which is the number of ways to choose σ. If $|\tau| > 1$ then τ must begin with the pattern 12, otherwise the letter 1 and the two leftmost letters of τ form the pattern 132, which is forbidden. So, in this case there are $\sum_{i\geq 0}\binom{n}{i} R_{i,k}F_{n-i,2}$ such permutations with the right properties, where i indicates the length of σ. In the last formula, of course, $R_{i,k} = 0$ if $i < k$. Finally we have to consider the situation when 1 is in the k-th position. In this case we can choose the letters of σ in $\binom{n}{k-1}$ ways, write them in decreasing order and then choose τ in $F_{n-k+1,2}$ ways. Thus

$$R_{n+1,k} = R_{n,k} + nR_{n-1,k} + \sum_{i\geq 0}\binom{n}{i} R_{i,k}F_{n-i,2} + \binom{n}{k-1}F_{n-k+1,2}. \tag{3}$$

We observe that (3) is not valid for $n = k-1$ and $n = k$. Indeed, if 1 is in the kth position in these cases, the term $\binom{n}{k-1}F_{n-k+1,2}$, which counts the number of such permutations, is zero, whereas there is one "good" $(n+1)$–permutation in case $n = k-1$ and n good $(n+1)$–permutations in case $n = k$. Multiplying both sides of the equality with $x^n/n!$, summing over n and using the observation above (which gives the term $x^{k-1}/(k-1)! + kx^k/k!$ in the right-hand side of equality (4)), we get

$$\frac{d}{dx}E^{p}_{132}(x) = (E^{12}_{132}(x) + x + 1)E^{p}_{132}(x) + (E^{12}_{132}(x) + x + 1)\frac{x^{k-1}}{(k-1)!}, \tag{4}$$

with the initial condition $E^p_{132}(0) = 0$. We solve this equation and get

$$E^p_{132}(x) = \frac{E^1_{132}(x)}{(k-1)!}\int_0^x \frac{(E^{12}_{132}(t)+t+1)t^{k-1}}{E^1_{132}(t)}\,dt = \frac{E^1_{132}(x)}{(k-1)!}\int_0^x t^{k-1}e^{-t^2/2}\,dt.$$

□

Remark 3. It is obvious that if in the previous theorem $p_1 \notin S_k(132)$ and $p_2 \notin S_k(123)$, then $E^{p_1}_{132}(x) = E^{p_2}_{123}(x) = 0$.

5. Avoiding a pattern x-yz and beginning with an increasing or decreasing pattern

In this section we consider avoidance of one of the patterns $1-23$, $1-32$, $2-31$, $2-13$, $3-12$ and $1-32$ and beginning with a decreasing pattern. We get all the other cases, that is, avoidance of one of these patterns and beginning with an increasing pattern, by the complement operation. For instance, we have $E^{k(k-1)\cdots1}_{1-23}(x) = E^{12\cdots k}_{3-21}(x)$.

Proposition 4. *The exponential generating functions* $E^{k(k-1)\cdots1}_{1-23}(x)$ *and* $E^{k(k-1)\cdots1}_{1-32}(x)$ *are given by*

$$\begin{cases} (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. We prove the statement for the pattern $1-23$. All the arguments we give for this pattern are valid for the pattern $1-32$. The only difference is that instead of decreasing order in τ (see below), we have increasing order.

Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n–permutations that avoid the pattern $1-23$ and begin with a decreasing subword of length k. Suppose $\pi = \sigma 1 \tau$ is one of such permutations of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of $1-23$ in π starting from the letter 1. If $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Besides, if $|\sigma| = k-1$ and letters of σ are in decreasing order, we get $\binom{n}{k-1}$ additional possibilities to choose π. Thus

$$B_{n+1,k} = \sum_{i\geq 0}\binom{n}{i}B_{i,k} + \binom{n}{k-1}.$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get the differential equation

$$\frac{d}{dx}E^{k(k-1)\cdots1}_{1-23}(x) = (E^{k(k-1)\cdots1}_{1-23}(x) + \frac{x^{k-1}}{(k-1)!})e^x$$

with the initial condition $E_{1-23}^{k(k-1)\dots1}(0) = 0$. The solution to this equation is given by

$$E_{1-23}^{k(k-1)\dots1}(x) = (e^{e^x}/(k-1)!) \int_0^x t^{k-1} e^{-e^t+t}\, dt. \tag{5}$$

If $k = 1$, then there is no additional restriction. According to [Claes, Prop. 2] (resp. [Claes, Prop. 3]), the number of n-permutations that avoid the pattern 1-23 (resp. 1-32) is the nth Bell number and the e.g.f. for the Bell numbers is e^{e^x-1}. However, all the arguments used for $k \geq 2$ remain the same for the case $k = 1$ except for the fact that we do not count the empty permutation, which, of course, avoids 1-23. So, if $k = 1$, we need to add 1 to the right-hand side of (5):

$$E_{1-23}^{1}(x) = e^{e^x} \int_0^x e^{-e^t+t}\, dt + 1 = e^{e^x-1}.$$

□

Proposition 5. *We have*

$$E_{3-12}^{k(k-1)\dots1}(x) = \begin{cases} e^{e^x} \int_0^x e^{-e^t} \sum\limits_{n\geq k-1} \dfrac{t^n}{n!}\, dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n–permutations that avoid the pattern $3-12$ and begin with a decreasing subword of length k. Suppose $\pi = \sigma(n+1)\tau$ is such a permutation of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of the pattern $3-12$ in π starting from the letter $(n+1)$. If $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Besides, if $n \geq k-1$, then π can be decreasing, that is, $(n+1)$ can be in the leftmost position. Thus

$$B_{n+1,k} = \sum_{i\geq 0} \binom{n}{i} B_{i,k} + \delta_{n,k},$$

where

$$\delta_{n,k} = \begin{cases} 1, & \text{if } n \geq k-1, \\ 0, & \text{else.} \end{cases}$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get the differential equation

$$\frac{d}{dx} E_{3-12}^{k(k-1)\dots1}(x) = e^x E_{3-12}^{k(k-1)\dots1}(x) + \sum_{n\geq k-1} \frac{x^n}{n!}$$

with the initial condition $E_{3-12}^{k(k-1)\dots1}(0) = 0$. The solution to this equation is given by

$$E_{3-12}^{k(k-1)\dots1}(x) = e^{e^x} \int_0^x e^{-e^t} \sum_{n\geq k-1} \frac{t^n}{n!}\, dt. \tag{6}$$

If $k = 1$, then there is no additional restriction. In [Claes, Prop. 3] it is shown that $E_{1-32}^1(x) = e^{e^x-1}$. Using the complement, the number of n-permutations that avoid $1-32$ is equal to the number of n-permutations that avoid $3-12$. We get that $E_{3-12}^1(x) = e^{e^x-1}$. However, all the arguments used for the case $k \geq 2$ remain the same for the case $k = 1$ except the fact that we do not count the empty permutation, which avoids 3-12. So, if $k = 1$, we need to add 1 to the right-hand side of (6):

$$E_{3-12}^1(x) = e^{e^x} \int_0^x e^{-e^t} e^t\, dt + 1 = e^{e^x-1}.$$

□

Proposition 6. *We have*

$$E_{3-21}^{k(k-1)\dots1}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ e^{e^x} \int_0^x e^{-e^t}(e^t - 1)\, dt, & \text{if } k = 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. For $k \geq 3$, the statement is obviously true. If $k = 1$, then the statement follows from [Claes, Prop. 2] and the fact that there are as many n-permutations avoiding the pattern $1-23$, as n-permutations avoiding the pattern $3-21$. For the case $k = 2$, we can use exactly the same arguments as those in the proof of Proposition 5 to get the same recurrence relation and thus the same formula, which, however, is valid only for $k = 2$. □

Recall that according to Section 2, N_q^p denotes the number of permutations that avoid the pattern q and begin with the pattern p.

Proposition 7. *We have*

$$N_{2-13}^{k(k-1)\dots1}(n) = \begin{cases} C_{n-k+1}, & \text{if } n \geq k, \\ 0, & \text{else.} \end{cases}$$

Proof. If $k = 1$, then the statement follows from [Claes, Prop. 7]. Suppose $k \geq 2$ and let $\pi = \sigma n \tau$ be an n-permutation avoiding $2-31$ and beginning with the pattern $k(k-1)\dots1$. Suppose, without loss of generality that σ consists of the letters $1, 2, \dots, \ell$. Now ℓ must be the rightmost letter of σ, since otherwise ℓ, the rightmost letter of σ and n form the pattern $2-13$. Also, the letter $(\ell-1)$ must be next to the rightmost letter of σ since otherwise the letter $(\ell-1)$, next to the rightmost letter of σ and the letter ℓ form the pattern $2-13$. And so on. Thus σ must be increasing,

which contradicts the fact that π must begin with a decreasing pattern of length greater than 1. So $|\sigma| = 0$ and τ must begin with the pattern $(k-1)(k-2)\dots 1$. Now, we can consider the letter $(n-1)$ and, by the same reasoning, get that it must be in the second position of π. Then we consider $(n-2)$, and so on up to the letter $(n-k+2)$. Finally, we get that $\pi = n(n-1)\dots(n-k+2)\pi'$, where π' must avoid the pattern $2-13$ and thus, there are C_{n-k+1} ways to choose π ([Claes, Prop. 7]). □

Recall that $C(x)$ is the generating function for the Catalan numbers. Also recall that according to Section 2, $G_q^p(x)$ denotes the ordinary generating function for the number of permutations that avoid the pattern q and begin with the pattern p.

Proposition 8. *We have*

$$G_{2-31}^{k(k-1)\dots 1}(x) = \begin{cases} x^{k-1}C^k(x), & \text{if } k \geq 2 \\ C(x), & \text{if } k = 1. \end{cases}$$

Proof. If $k = 1$, then there is no additional restriction, and thus $G_{2-31}^{1}(x) = C(x)$ (applying the complement operation to [Claes, Prop. 7]).

Suppose $k \geq 2$. Using the reverse, we see that beginning with $k(k-1)\dots 1$ and avoiding $2-31$ is equivalent to ending with $12\dots k$ and avoiding $13-2$, which by [Claes, Lemma 2] is equivalent to ending with $12\dots k$ and avoiding $1-3-2$.

Suppose $\pi = \pi' n \pi''$ ends with $12\dots k$ and avoids $1-3-2$. Each letter of π' must be greater than any letter of π'', since otherwise we have an occurrence of the pattern $1-3-2$ involving the letter n. Also, π' and π'' avoid the pattern $1-3-2$, and π'' ends with the pattern $12\dots k$. In terms of generating functions (the generating function for the number of permutations ending with $12\dots k$ and avoiding $1-3-2$ is, of course, $G_{2-31}^{k(k-1)\dots 1}(x)$) this means that

$$G_{2-31}^{k(k-1)\dots 1}(x) = xC(x)G_{2-31}^{k(k-1)\dots 1}(x) + xG_{2-31}^{(k-1)\dots 1}(x), \tag{7}$$

where the rightmost term corresponds to the case when π'' is empty. Now, (1) and (7) give $G_{2-31}^{k(k-1)\dots 1}(x) = x^{k-1}C(x)/(1-xC(x))^{k-1} = x^{k-1}C^k(x)$. □

6. Avoiding a pattern xy-z and beginning with an increasing or decreasing pattern

First of all we state the following well-known binomial identity

$$\sum_{i=1}^{n-m-k+1} \binom{n-m-i}{k-1}\binom{m+i-1}{m} = \binom{n}{m+k}. \tag{8}$$

Let $s_q(n)$ denote the cardinality of the set $S_n(q)$ and $s_q(n; i_1, i_2, \dots i_m)$ denote the number of permutations $\pi \in S_n(q)$ with $\pi_1\pi_2 \dots \pi_m = i_1 i_2 \dots i_m$.

In this section we consider avoidance of one of the patterns 12-3, 13-2 and 23-1 and beginning with an increasing or decreasing pattern. We get all the other cases, which are avoidance of one of the patterns 32-1, 31-2 and 21-3 and beginning with an increasing or decreasing pattern, by the complement operation. For instance, we have $N_{13-2}^{12\dots k}(n) = N_{31-2}^{k(k-1)\dots 1}(n)$.

6.1. **The pattern $12-3$.** We first consider beginning with the pattern $p = k \dots 21$. In [ClaesMans2, Lemma 9] it was proved that

$$s_{12-3}(n; i) = \sum_{j=0}^{i-1} \binom{i-1}{j} s_{12-3}(n-2-j),$$

together with $s_{12-3}(n; n) = s_{12-3}(n; n-1) = s_{12-3}(n-1)$.

On the other hand, from the definitions, it is easy to see that

$$N_{12-3}^{k(k-1)\dots 1}(n) = \sum_{i=1}^{n-k+1} \binom{n-i}{k-1} s_{12-3}(n-k+1; i).$$

Hence, using (8) and the fact shown in [Claes, Prop. 2] that $s_{12-3}(n)$ equals B_n, we get the following proposition.

Proposition 9. *For all $n \geq k+1$, we have*

$$N_{12-3}^{k(k-1)\dots 1}(n) = (k+1)B_{n-k} + \\ + \sum_{j=0}^{n-k-2} \left(\tbinom{n}{k+j} - k\tbinom{n-k-1}{j} - \tbinom{n-k}{j} \right) B_{n-k-1-j},$$

together with $N_{12-3}^{k(k-1)\dots 1}(k) = 1$ and $N_{12-3}^{k(k-1)\dots 1}(n) = 0$ for all $n \leq k-1$.

Now, let us consider beginning with the pattern $p = 12 \dots k$. From the definitions, it is easy to see that $N_{12-3}^{12\dots k}(n) = 0$ for all n, where $k \geq 3$, and $N_{12-3}^{1}(n) = s_{12-3}(n) = B_n$ (see [ClaesMans1, Prop. 3]). Thus, we only need to consider the case $k = 2$.

Suppose $\pi \in S_{12-3}(n)$ is a permutation with $\pi_1 < \pi_2$. It is easy to see that $\pi_2 = n$. Hence $N_{12-3}^{12}(n) = (n-1)s_{12-3}(n-2)$, for all $n \geq 2$, and by [ClaesMans1, Prop. 3], we get the truth of the following

Proposition 10. *The exponential generating function $E_{12-3}^{12\dots k}(x)$ is given by*

$$\begin{cases} 0, & \text{if } k \geq 3, \\ x^2 \displaystyle\sum_{j=0}^{k} (1-jx)^{-1} \sum_{d \geq 0} \frac{x^d}{(1-x)(1-2x)\dots(1-dx)}, & \text{if } k = 2, \\ \displaystyle\sum_{d \geq 0} \frac{x^d}{(1-x)(1-2x)\dots(1-dx)}, & \text{if } k = 1. \end{cases}$$

6.2. **The pattern** $13-2$. Now we find $G_{13-2}^{k(k-1)\dots1}(n)$.

Proposition 11. *For any* $k \geq 1$,

$$G_{13-2}^{k(k-1)\dots1}(x) = x^k C^{k+1}(x).$$

Proof. Claesson [Claes, Lemma 2] proved that the set of permutations that avoid the pattern $13-2$ is the same as the set of permutations that avoid the pattern $1-3-2$, hence

$$G_{13-2}^{k(k-1)\dots1}(x) = G_{1-3-2}^{k(k-1)\dots1}(x). \tag{9}$$

Let $\pi = (\pi', n, \pi'')$ avoids $1-3-2$ and beginning with $k(k-1)\dots1$. Since π avoids $1-3-2$, π' and π'' avoid $1-3-2$, and every letter in π' is greater than any letter in π''. We have two possibilities: either π' is empty or π' begins with $k(k-1)\dots1$. One can see that in the first (resp. second) case the generating function for the number of such permutations is $xG_{13-2}^{(k-1)(k-2)\dots1}(x)$ (resp. $xG_{13-2}^{k(k-1)\dots1}(x)C(x)$). Hence, for $k \geq 2$ we have

$$G_{13-2}^{k(k-1)\dots1}(x) = xG_{13-2}^{(k-1)(k-2)\dots1}(x) + xG_{13-2}^{k(k-1)\dots1}(x)C(x),$$

with $G_1^{1-3-2}(x) = C(x) - 1 = xC^2(x)$. The rest is given by induction on k. □

Now, let us consider the case of $N_{13-2}^{12\dots k}(n)$.

Proposition 12. *Let* $k \geq 1$. *For all* $n \geq k$, *we have*

$$N_{13-2}^{12\dots k}(n) = C_{n+1-k}.$$

Proof. Suppose $\pi = \pi' n \pi''$ is a permutation in $S_n(13-2) = S_n(1-3-2)$ (see (9)), such that $\pi_1 < \pi_2 < \cdots < \pi_k$. It is easy to see that there exists an m such that

$$\pi = (m+1)(m+2)\dots(m+k-1)\beta n \pi'',$$

where β is a $1-3-2$-avoiding permutation on the letters $m+k, m+k+1, \dots, n-1$, and $\pi'' \in S_m(1-3-2)$. Hence, in terms of generating functions, we get

$$\sum_{n\geq 0} N_{13-2}^{12\dots k}(n)x^n = x^k C^2(x).$$

The rest is easy to check using the identity $xC^2(x) = C(x) - 1$. □

6.3. **The pattern** $23-1$. We first consider beginning with the pattern $p = k(k-1)\dots1$.

Proposition 13. *For all* $k \geq 1$,

$$E_{23-1}^{k(k-1)\dots1}(x) = x^{k-1}\left(\sum_{d\geq 0} \frac{x^d}{(1-x)(1-2x)\cdots(1-dx)} - 1\right).$$

Proof. Let $\pi \in S_n(23-1)$ be a permutation such that $\pi_1 < \pi_2 < \cdots < \pi_k$. Since π avoids $23-1$, we have $\pi_j = j$, for each $j = 1, 2, \ldots, k-1$. Hence $\pi = 12\ldots(k-1)\pi'$, where π' is a non-empty $23-1$-avoiding permutation in S_{n+1-k}. The rest is easy to get by using [ClaesMans1, Prop. 17]. □

Now let us consider beginning with the pattern $p = 12\ldots k$.

Proposition 14. *Suppose $k \geq 1$. For all $n \geq k+1$,*

$$N_{23-1}^{12\ldots k}(n) = \left(1 + \tbinom{n-1}{k-1}\right) B_{n-k} + \sum_{j=0}^{n-k-2} \left[\tbinom{n-1}{k+j} - \tbinom{n-k-1}{j}\right] B_{n-k-1-j},$$

with $N_{23-1}^{12\ldots k}(k) = 1$.

Proof. In [ClaesMans2, Lemma 16] proved that for all $2 \leq i \leq n-1$,

$$s_{23-1}(n;i) = \sum_{j=0}^{i-2} \binom{i-2}{j} s_{23-1}(n-2-j),$$

together with $s_{23-1}(n;n) = s_{23-1}(n;1) = s_{23-1}(n-1) = B_{n-1}$.

On the other hand, by the definitions, it is easy to see that

$$N_{23-1}^{12\ldots k}(n) = \sum_{i=1}^{n-k+1} \binom{n-i}{k-1} s_{23-1}(n-k+1;i).$$

Hence, using (8) and the fact that [Claes, Prop. 4] $s_{23-1}(n)$ is given by B_n, we get the desired result. □

7. Avoiding a pattern xy-z and beginning with the pattern $(k-1)(k-2)\ldots 1k$ or $23\ldots k1$

In this section we consider avoidance of one of the patterns $12-3$, $13-2$, $23-1$, $21-3$, $31-2$ and $13-2$ and beginning with the pattern $(k-1)(k-2)\ldots 1k$. The case when a permutation begins with the pattern $23\ldots k1$ and avoids a pattern $xy-z$ can be obtained then by the complement operation.

7.1. Avoiding $12-3$ and beginning with $(k-1)(k-2)\ldots 1k$.

Proposition 15. *We have*

$$N_{12-3}^{(k-1)(k-2)\ldots 1k}(n) = \binom{n-1}{k-1} B_{n-k}.$$

Proof. Suppose $\pi = \pi' n \pi''$ avoids the pattern $12-3$ and begins with the pattern $(k-1)(k-2)\ldots 1k$. We have that π' must be decreasing, since otherwise we have an occurrence of the pattern $12-3$ involving the letter n, and π'' must avoid $12-3$. Also, since π begins with $(k-1)\ldots 21k$, the

length of π' is $k-1$. Hence, by [Claes, Prop. 2] (the number of permutations in $S_n(12-3)$ is given by B_n), we have

$$N_{12-3}^{(k-1)(k-2)\dots 1k}(n) = \binom{n-1}{k-1} B_{n-k}.$$

□

7.2. **Avoiding $13-2$ and beginning with $(k-1)(k-2)\dots 1k$.** By [Claes, Lemma 2], a permutation π avoids the pattern $13-2$ if and only if π avoids $1-3-2$.

Suppose $\pi = \pi' n \pi''$ is an n-permutation avoiding $1-3-2$ and beginning with $(k-1)(k-2)\dots 1k$. Obviously, π' and π'' avoid $1-3-2$ and each letter of π' is greater than any letter of π'', since otherwise we have an occurrence of the pattern $1-3-2$ involving the letter n. Also, π' begins with the pattern $(k-1)(k-2)\dots 1k$ or $\pi' = (k-1)(k-2)\dots 1$.

By [Knuth], the generating function for the number of permutations that avoid $1-3-2$ is $C(x)$, hence, using the considerations above,

$$G_{13-2}^{(k-1)(k-2)\dots 1k}(x) = xG_{13-2}^{(k-1)(k-2)\dots 1k}(x)C(x) + x^k C(x).$$

Therefore, by (1), we get the following.

Proposition 16. *We have*

$$G_{13-2}^{(k-1)(k-2)\dots 1k}(x) = x^k C^2(x).$$

Hence

$$N_{13-2}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} C_{n-(k-1)}, & \text{if } n \geq k \\ 0, & \text{else.} \end{cases}$$

7.3. **Avoiding $21-3$ and beginning with $(k-1)(k-2)\dots 1k$.** If $k \geq 3$ then, by the definitions, we have $N_{21-3}^{(k-1)(k-2)\dots 1k}(n) = 0$. If $k = 1$ then, by the definitions and [Claes, Prop. 4], we have $N_{21-3}^{1}(n) = B_n$. Suppose $k = 2$ and $\pi = \pi' n \pi''$ is an n-permutation avoiding the pattern $21-3$ and beginning with the pattern $(k-1)(k-2)\dots 1k = 12$. It is easy to see that π' must be increasing, and the length of π' is at least 1. Thus, using the fact that the number of permutations in $S_n(21-3)$ is given by B_n (see [Claes, Prop. 4]), we have

$$N_{21-3}^{(k-1)(k-2)\dots 1k}(n) = \sum_{j=1}^{n-1} \binom{n-1}{j} B_{n-1-j}. \tag{10}$$

Since $B_n = \sum_{j=0}^{n-1} \binom{n-1}{j} B_{n-1-j}$, equality (10) gives that

$$N_{21-3}^{(k-1)(k-2)\dots 1k}(n) = B_n - B_{n-1}.$$

Thus we have proved the following.

Proposition 17. *We have*

$$N_{21-3}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} 0, & \text{if } k \geq 3 \\ B_n - B_{n-1}, & \text{if } k = 2, \\ B_n, & \text{if } k = 1. \end{cases}$$

7.4. **Avoiding $23-1$ and beginning with $(k-1)(k-2)\dots 1k$.**

Proposition 18. *The number $N_{23-1}^{(k-1)(k-2)\dots 1k}(n)$ is given by*

$$\begin{cases} B_{n-k} + \displaystyle\sum_{t=2}^{n-k+2} \binom{t+k-3}{k-2} \sum_{j=0}^{t-2} \binom{t-2}{j} B_{n-k-1-j}, & \text{if } k \geq 3 \\ B_{n-1}, & \text{if } k = 2, \\ B_n, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k = 2$. We are interested in the permutations $\pi \in S_n(23-1)$ that begin with the pattern 12. It is easy to see that $\pi_1 = 1$, hence $B_{12}^{23-1}(n) = B_{n-1}$ for all $n \geq 2$.

Suppose $k \geq 3$. We recall that $s_{23-1}(n;t)$ is the number of permutations in $S_n(23-1)$ having t as the first letter. If now j denotes the number of letters between the letters t and 1, then by [ClaesMans1], $s(n;1) = B_{n-1}$ and for $t \geq 2$, we have

$$s_{23-1}(n;t) = \sum_{j=0}^{t-2} \binom{t-2}{j} B_{n-2-j}.$$

On the other hand, if a permutation $\pi = \pi' 1 t \pi''$ avoids $23-1$ and begins with the pattern $(k-1)(k-2)\dots 1k$, then π' is decreasing of length $k-2$, and using $s_{23-1}(n;t)$, we get

$$N_{23-1}^{(k-1)(k-2)\dots 1k}(n) = B_{n-k} + \sum_{t=2}^{n-k+1} \binom{t+k-3}{k-2} \sum_{j=0}^{t-2} \binom{t-2}{j} B_{n-k-1-j}.$$

□

7.5. **Avoiding $31-2$ and beginning with $(k-1)(k-2)\dots 1k$.** By [Claes, Lemma 2], a permutation π avoids the pattern $31-2$ if and only if π avoids the pattern $3-1-2$.

Suppose $\pi = \pi' 1 \pi''$ is an n-permutation avoiding $3-1-2$ and beginning with $(k-1)(k-2)\dots 1k$. Obviously, π' and π'' avoid $3-1-2$ and each letter of π' is smaller than any letter of π'', since otherwise we have an occurrence of the pattern $3-1-2$ involving the letter 1. Also, π' begins with the pattern $(k-1)(k-2)\dots 1k$ or $\pi' = (k-1)(k-2)\dots 2$ and π'' is not empty. So, using the generating function for the number of permutations avoiding the pattern $3-1-2$, which is $C(x)$ ([Knuth]), we get

$$G_{31-2}^{(k-1)(k-2)\dots 1k}(x) = xG_{31-2}^{(k-1)(k-2)\dots 1k}(x)C(x) + x^{k-1}(C(x)-1).$$

Therefore, using (1), we get the following.

Proposition 19. *We have*

$$G_{31-2}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} x^k C^3(x), & \text{if } k \geq 2, \\ C(x), & \text{if } k = 1. \end{cases}$$

Hence

$$N_{31-2}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} C_{n-k+2} - C_{n-k+1}, & \text{if } k \geq 2, \\ C_n, & \text{if } k = 1. \end{cases}$$

7.6. **Avoiding $32-1$ and beginning with $(k-1)(k-2)\dots 1k$.**

Proposition 20. *We have*

$$N_{32-1}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} 0, & \text{if } k \geq 4 \\ B_{n-1} - (n-2)B_{n-3}, & \text{if } k = 3 \text{ and } n \geq 3, \\ B_n - (n-1)B_{n-2}, & \text{if } k = 2 \text{ and } n \geq 2, \\ B_n, & \text{if } k = 1. \end{cases}$$

Proof. Using the definitions and [Claes, Prop. 2], it is easy to see that the statement is true for $k = 1, 2$ and $k \geq 4$.

Suppose now that $k = 3$ and $\pi = \pi' 1 \pi''$ is an n-permutation avoiding the pattern $32-1$ and beginning with the pattern $(k-1)(k-2)\dots 1k = 213$. We have that π' must be increasing, since otherwise we have an occurrence of the pattern $32-1$ involving the letter 1, and π'' must avoid $32-1$. Moreover, since π begins with 213, the length of π is 1 and the rightmost letter of π'' is greater than the letter of π'. Also, it is easy to see that the number of permutations in $S_{n-1}(32-1)$ beginning with the pattern 12 is the same as the number of permutations in $S_n(32-1)$ beginning with the pattern 213 (one can see it by placing 1 in the second position). Hence $N_{32-1}^{(k-1)\dots 21k}(n) = B_{n-1} - (n-2)B_{n-3}$ for all $n \geq 3$. □

8. Avoiding a pattern x-yz and beginning with the pattern $(k-1)(k-2)\dots 1k$ or $23\dots k1$

In this section we consider avoidance of one of the patterns $1-23$, $1-32$, $2-31$, $2-13$, $3-12$ and $1-32$ and beginning with the pattern $(k-1)(k-2)\dots 1k$. The case when a permutation begins with the pattern $23\dots k1$ and avoids a pattern $x-yz$ can be obtained by the complement operation.

Proposition 21. *We have*

$$E_{1-32}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} e^{e^x} \int_0^x e^{-e^t} \sum\limits_{n \geq k-1} \dfrac{t^n}{n!}\, dt, & \text{if } k \geq 2, \\ e^{e^x - 1}, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n–permutations that avoid the pattern $1-32$ and begin with the pattern $(k-1)(k-2)\dots 1k$. Suppose $\pi = \sigma 1 \tau$ is such a permutation of length $n+1$. Obviously, the letters of τ must be in increasing order, since otherwise we have an occurrence of the pattern $1-32$ in π starting from the letter 1. If $|\sigma| = i$, then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in increasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Also, if $n \geq k-1$, then 1 can be in the $(k-1)$th position, and in this case, since π begins with the pattern $(k-1)(k-2)\dots 1k$, it must be that $\pi = (k-1)(k-2)\dots 21k(k+1)\dots(n+1)$. Thus, in the last case we have only one permutation. This leads to the recurrence relation

$$B_{n+1,k} = \sum_{i \geq 0} \binom{n}{i} B_{i,k} + \delta_{n,k},$$

where

$$\delta_{n,k} = \begin{cases} 1, & \text{if } n \geq k-1, \\ 0, & \text{else.} \end{cases}$$

This recurrence relation is identical to the one given in the proof of Proposition 5, so using this proof we get the desired result. □

Proposition 22. *We have*

$$E_{1-23}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} e^{e^x} \int_0^x \int_0^t \frac{r^{k-2}}{(k-2)!} e^{r-e^t}\, dr dt, & \text{if } k \geq 2, \\ e^{e^x - 1}, & \text{if } k = 1. \end{cases}$$

Proof. If $k = 1$, then the statement is true due to Proposition 4.

Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n-permutations that avoid the pattern $1-23$ and begin with the pattern $(k-1)(k-2)\dots 1k$. Suppose $\pi = \sigma 1 \tau$ is such a permutation of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of the pattern $1-23$ in π starting from the letter 1. If $|\sigma| = i$, then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in the decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Besides, if $n \geq k-1$, then 1 can be in the $(k-1)$th position, and in this case, since π begins with the pattern $(k-1)(k-2)\dots 1k$ and τ is decreasing, it must be that the kth letter of π is $(n+1)$ and there are $\binom{n-1}{k-2}$ ways to choose the letters of σ and then write them in decreasing order. Thus,

$$B_{n+1,k} = \sum_{i \geq 0} \binom{n}{i} B_{i,k} + \binom{n-1}{k-2}.$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get the differential equation

$$\frac{d}{dx}E_{1-23}^{(k-1)(k-2)\dots 1k}(x) = E_{1-23}^{(k-1)(k-2)\dots 1k}e^x + \sum_{n\geq 0}\binom{n-1}{k-2}\frac{x^n}{n!},$$

with the initial condition $E_{1-23}^{(k-1)(k-2)\dots 1k}(0) = 0$. If $F(x)$ denotes the last term, then it is easy to see that $F'(x) = \frac{x^{k-2}}{(k-2)!}e^x$, and thus

$$F(x) = \int_0^x \frac{t^{k-2}}{(k-2)!}e^t\ dt.$$

Now, the solution to the equation above is given by

(11)
$$E_{1-23}^{(k-1)(k-2)\dots 1k}(x) = e^{e^x}\int_0^x e^{-e^t}F(t)\ dt = e^{e^x}\int_0^x\int_0^t \frac{r^{k-2}}{(k-2)!}e^{r-e^t}\ drdt.$$

For example, if $k = 2$, then $(k-1)(k-2)\dots 1k = 12$ and (11) gives

$$E_{1-23}^{12} = e^{e^x}\int_0^x e^{-e^t}(e^t-1)\ dt,$$

which is a particular case of Proposition 6, since the number of n-permutations that avoid the pattern $3-21$ and begin with the pattern 21 is equal to the number of n-permutations that avoid the pattern $1-23$ and begin with the pattern 12 by applying the complement. □

Proposition 23. *We have*

$$G_{2-13}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} 0, & \text{if } k\geq 3\\ x^2C^3(x), & \text{if } k=2\\ C(x), & \text{if } k=1.\end{cases}$$

Hence

$$N_{2-13}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} 0, & \text{if } k\geq 3\\ C_{n-1}-C_{n-2}, & \text{if } k=2\\ C_n, & \text{if } k=1.\end{cases}$$

Proof. For the case $k = 1$, see Proposition 7. If $k \geq 3$, then the statement is true, since in this case the pattern $(k-1)(k-2)\dots 1k$ does not avoid $2-13$.

Suppose now that $k = 2$. Using the reverse, we see that beginning with the pattern 12 and avoiding $2-13$ is equivalent to ending with the pattern 21 and avoiding $31-2$, which by [Claes, Lemma 2] is equivalent to ending with the pattern 21 and avoiding the pattern $3-1-2$.

Let $\pi = \pi'1\pi''$ be an n-permutation avoiding $3-1-2$ and ending with the pattern 21. Obviously, π' and π'' avoid $3-1-2$ and each letter of π' is less than any letter of π'', since otherwise we have an occurrence of $3-1-2$ involving the letter 1. Also, π'' ends with the pattern 21 or $|\pi''| = 1$.

So, using the generating function for the number of permutations avoiding $3-1-2$, which is $C(x)$ ([Knuth]), we have

$$G^{12}_{2-13}(x) = xG^{12}_{2-13}(x)C(x) + x(C(x)-1).$$

Therefore, using (1), we get the desired result. □

Proposition 24. *We have*

$$G^{(k-1)(k-2)\dots 1k}_{2-31}(x) = x^k C^2(x).$$

Hence

$$N^{(k-1)(k-2)\dots 1k}_{2-31}(n) = \begin{cases} C_{n-(k-1)}, & \text{if } n \geq k \\ 0, & \text{else.} \end{cases}$$

Proof. Using the reverse, we see that beginning with the pattern $(k-1)(k-2)\dots 1k$ and avoiding the pattern $2-31$ is equivalent to ending with the pattern $k12\dots(k-1)$ and avoiding the pattern $13-2$, which, by [Claes, Lemma 2], is equivalent to ending with the pattern $k12\dots(k-1)$ and avoiding the pattern $1-3-2$.

Let $\pi = \pi' n \pi''$ be an n-permutation avoiding the pattern $1-3-2$ and ending with the pattern $k12\dots(k-1)$. Obviously, π' and π'' avoid the pattern $1-3-2$ and each letter of π' is greater than any letter of π'', since otherwise we have an occurrence of the pattern $1-3-2$ involving the letter n. Also, π'' ends with the pattern $k12\dots(k-1)$ or $\pi'' = 12\dots(k-1)$.

Using the reverse operation, the generating function for the number of permutations ending with the pattern $k12\dots(k-1)$ and avoiding $1-3-2$ is equal to $G^{(k-1)(k-2)\dots 1k}_{2-31}(x)$. In terms of generating functions, the considerations above lead to

$$G^{(k-1)(k-2)\dots 1k}_{2-31}(x) = xG^{(k-1)(k-2)\dots 1k}_{2-31}(x)C(x) + x^k C(x).$$

Therefore, by (1), we get the desired result. □

Proposition 25. *We have*

$$E^{(k-1)(k-2)\dots 1k}_{3-12}(x) = \begin{cases} (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n-permutations that avoid the pattern $3-12$ and begin with a decreasing subword of length k. Let $\pi = \sigma(n+1)\tau$ be such a permutation of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of $3-12$ in π starting from the letter (n+1). If $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities

to choose σ. Also, if $|\sigma| = k-1$ and the letters of σ are in decreasing order, we get $\binom{n}{k-1}$ additional ways to choose π. Thus

$$B_{n+1,k} = \sum_{i\geq 0} \binom{n}{i} B_{i,k} + \binom{n}{k-1}.$$

This recurrence relation is identical to the one given in the proof of Proposition 4, and we get the desired result using that proof. □

Proposition 26. *We have*

$$E_{3-21}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} 0, & \text{if } k \geq 4 \\ (e^{e^x}/(k-1)!) \int_0^x t^{k-1} e^{-e^t+t}\, dt, & \text{if } k = 2, 3, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. If $k \geq 4$ then the statement is true, since in this case the pattern $(k-1)(k-2)\dots 1k$ does not avoid the pattern $3-21$. In the other cases, we use the same arguments as we have in the proof of Proposition 25. The only difference is that instead of decreasing order in τ, we have increasing order. □

9. Conclusions

The goal of our paper is to give a complete description for the numbers of permutations avoiding a pattern of the form $x - yz$ or $xy - z$ and either beginning with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $23\dots k1$, $(k-1)(k-2)\dots 1k$, or ending with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $1k(k-1)\dots 2$, $k12\dots(k-1)$. This description is given in Sections 5–8. However, some of our results can be generalized to beginning with a pattern belonging to Γ_k^{min} or Γ_k^{max}, and thus to the ending with a pattern belonging to Δ_k^{min} or Δ_k^{max} (see Section 2 for definitions). An example of such a generalisation is given in Theorem 27 below. This theorem generalizes Propositions 4 and 25 and can be proved by using the same considerations as we do in the proofs of these propositions.

Theorem 27. *Suppose $p_1, p_2 \in \Gamma_k^{min}$ and $p_1 \in S_k(1-23)$, $p_2 \in S_k(1-32)$. Thus, the complements $C(p_1), C(p_2) \in \Gamma_k^{max}$ and $C(p_1) \in S_k(1-23)$, $C(p_2) \in S_k(3-12)$. Then, we have*

$$E_{1-23}^{p_1}(x) = E_{3-21}^{C(p_1)}(x) = E_{1-32}^{p_2}(x) = E_{3-12}^{C(p_2)}(x) =$$

$$\begin{cases} (e^{e^x}/(k-1)!) \int_0^x t^{k-1} e^{-e^t+t}\, dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Acknowledgments. The authors are grateful to the referees for the careful reading of the manuscript. The final version of the paper was written

during the second author's (T.M.) stay at Chalmers University of Technology in Göteborg, Sweden. T.M. want to express his gratitude to Chalmers University of Technology for the support.

References

[BabStein] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, *Séminaire Lotharingien de Combinatoire*, B44b:18pp, 2000.

[Bon] M. Bóna: Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A* **80** (1997), no. 2, 257–272.

[B] M. Bóna: The permutation classes equinumerous to the smooth class. *Electron. J. Combin. 5* (1998), no. **1**, Research Paper 31, 12 pp. (electronic).

[CW] T. Chow and J. West: Forbidden subsequences and Chebyshev polynomials. *Discrete Math.* **204** (1999), no. 1-3, 119–128.

[Claes] A. Claesson: Generalised Pattern Avoidance, *European J. Combin.* **22** (2001), 961–971.

[ClaesMans1] A. Claesson and T. Mansour: Enumerating Permutations Avoiding a Pair of Babson-Steingrímsson Patterns, preprint CO/0107044.

[ClaesMans2] A. Claesson and T. Mansour: Counting Occurrences of a Pattern of Type (1,2) or (2,1) in Permutations, *Adv. App. Math.* **29** (2002), 293–310.

[ElizNoy] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, *Proceedings of FPSAC 2001.*

[Ent] R. Entringer: A Combinatorial Interpretation of the Euler and Bernoulli Numbers, *Nieuw. Arch. Wisk.* **14** (1966), 241–246.

[Kit1] S. Kitaev: Multi-avoidance of generalised patterns, *Discrete Math.* **260** (2003), 89–100.

[Kit2] S. Kitaev: Partially ordered generalized patterns, *Discrete Math.*, to appear (2002).

[Kit3] S. Kitaev: Generalized pattern avoidance, *Séminaire Lotharingien de Combinatoire* **48** (2003), Article B48e, 19 pp.

[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Kr] C. Krattenthaler: Permutations with restricted patterns and Dyck paths, *Adv. in Appl. Math.* **27** (2001), 510–530.

[K] D. Kremer: Permutations with forbidden subsequnces and a generalized Schröder number, *Discrete Math.* **218** (2000), 121–130.

[Loth] M. Lothaire: *Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, **17**, Addison-Wesley Publishing Co., Reading, Mass. (1983).

[Mans1] T. Mansour: Continued fractions and generalized patterns, *European J. Combin.* **23** (2002), no. 3, 329–344.

[Mans2] T. Mansour: Continued fractions, statistics, and generalized patterns, to appear in Ars Combinatorica (2002), preprint CO/0110040.

[Mans3] T. Mansour: Restricted 1-3-2 permutations and generalized patterns, *Annals of Combinatorics* **6** (2002), 1–12.

[MV1] T. Mansour and A. Vainshtein: Restricted permutations, continued fractions, and Chebyshev polynomials, *Electron. J. Combin.* **7** (2000) no. 1, Research Paper 17, 9 pp. (electronic).

[MV2] T. Mansour and A. Vainshtein: Restricted 132-avoiding permutations, *Adv. in Appl. Math.* **126** (2001), no. 3, 258–269.

[MV3] T. Mansour and A. Vainshtein: Layered restrictions and Chebyshev polynomials, *Annals of Combinatorics* **5** (2001), 451–458.

[MV4] T. Mansour and A. Vainshtein: Restricted permutations and Chebyshev polyonomials, *Séminaire Lotharingien de Combinatoire* **47** (2002), Article B47c.

[R] A. Robertson: Permutations containing and avoiding 123 and 132 patterns, *Discrete Math. Theor. Comput. Sci.* **3** (1999), no. 4, 151–154 (electronic).

[RWZ] A. Robertson, H. Wilf, and D. Zeilberger: Permutation patterns and continued fractions, *Electron. J. Combin.* **6** (1999), no. 1, Research Paper 38, 6 pp. (electronic).

[SloPlo] N. J. A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, (1995). `http://www.research.att.com/~njas/sequences/`

[Stan] R. Stanley: *Enumerative Combinatorics*, Vol. **1**, Cambridge University Press, (1997).

[SchSim] R. Simion, F. Schmidt: Restricted permutations, *European J. Combin.* **6** (1985), no. 4, 383–406.

[W] J. West: Generating trees and forbidden subsequences, *Discrete Math.* **157** (1996), 363–372.

Matematik, Chalmers tekniska högskola och Göteborgs universitet, 412 96 Göteborg, Sweden

E-mail address: `kitaev@math.chalmers.se`

LaBRI, Université Bordeaux 1, 351 cours de la Libération 33405 Talence Cedex, France

E-mail address: `toufik@labri.fr`

A set partition identity via trees

Martin Klazar[1] and Vít Novák

Department of Applied Mathematics of Charles University
Malostranské náměstí 25
118 00 Praha 1
Czech Republic
klazar@kam.ms.mff.cuni.cz
novakvt@fzu.cz

Abstract

We consider two kinds of partitions having n blocks and an initial segment of positive integers as a ground set. Pretty partition has all blocks of size at most 2, does not induce the pattern $\ldots a \ldots b \ldots b \ldots a \ldots$, and has no two consecutive numbers in the same block. Ugly partition differs only in that it does have some two consecutive numbers in the same block. Using rooted plane trees we construct, for any $n \geq 1$, a bijection matching pretty and ugly partitions.

1 Introduction

A *partition* u with n *blocks* is a set of n nonempty disjoint subsets of $X = \{1, 2, \ldots, l\}$ whose union is X. We say that u is *abba-free* if there are no four distinct numbers $1 \leq i_1 < \ldots < i_4 \leq l$ and no two distinct blocks A and B such that $i_1, i_4 \in A$ and $i_2, i_3 \in B$. Partitions having no two consecutive numbers in the same block are called *pretty*, otherwise they are *ugly*.

The purpose of this note is to prove bijectively the following identity.

Identity 1.1 *Among abba-free partitions with $n \geq 1$ blocks, each block of size 1 or 2, there is as many pretty partitions as ugly partitions.*

Any partition u can be written as a sequence $a_1 a_2 \ldots a_l$ of labels given to the blocks: a_i is the label of the block B, $i \in B$. The *canonical form* of u is obtained when the blocks are ordered by their least elements as $B_1, B_2, \ldots, B_n$ and B_i is labeled by i. We shall work with partitions in their sequential form.

[1]supported by grants GAČR 0194 and GAUK 194.

For instance, one way how to write $u = \{\{2,3,5\},\{1,6\},\{4\}\}$ as a sequence is $u = bccacb$ and the canonical form is $u = 122321$. For $n = 2$ the pretty and ugly partitions appearing in the identity are:

$$\{12, 121, 1212\} \text{ and } \{112, 122, 1122\}.$$

For $n = 3$ the two sets described in the identity have 11 elements.

The identity was discovered in [1] as a byproduct of formulae for generating functions enumerating *abba*-free partitions. In the next section we present a bijection proving the identity. Our main tool is an encoding of *abba*-free partitions by rooted plane trees.

2 The bijection

A *rooted plane tree* is a finite directed tree with all edges directed away from the distinguished vertex, called a *root*, and with a linear order on any set of children of a vertex. From now on we call them shortly *trees*.

We think of trees as plane pictures. We draw vertices as points, the root in the lowest position, and edges as straight segments directed up. The children of a vertex are drawn from left to right in accordance with the prescribed linear order. It is well known that there are $\binom{2n}{n}/(n+1)$ (Catalan number) different trees with n edges.

For $e = v_1v_2$ an edge in a tree T we refer to v_2 as to the *child* of v_1 and to v_1 as to the *parent* of v_2. A vertex with no child is called a *leaf*. A *layer* in T is the set of vertices with the same distance from the root. Suppose the vertices of T are ordered as $v_0, v_1, \ldots, v_n$ so that lower layers come first and in one layer left vertices come first. Hence v_0 is the root. Such an order is called *good ordering*. A vertex of T is called *solitary* (*young*) if it is the only vertex in its layer and its parent is the rightmost vertex in its layer (if it is a leaf whose parent is the root).

Let $\mathcal{S}(n)$ stand for the set of *abba*-free partitions with n blocks, each block of size 1 or 2. The subsets of pretty and ugly partitions are denoted by $\mathcal{P}(n)$ and $\mathcal{U}(n)$. The subset of partitions with two-element blocks only is $\mathcal{R}(n)$. The set of trees with n edges is denoted by $\mathcal{T}(n)$.

In the rest of the note we shall construct a bijection F between the sets $\mathcal{P}(n)$ and $\mathcal{U}(n)$. First we restate the identity in terms of tree structures called *gap trees*. In the second step we construct the desired bijection, working with gap trees rather than with partitions.

From partitions to gap trees

We start with a bijection G between $\mathcal{R}(n)$ and $\mathcal{T}(n)$. Suppose $u = a_1 a_2 \dots a_{2n} \in \mathcal{R}(n)$ is in the canonical form. The tree $T = G(u)$ is constructed by processing u from left to right. In the beginning $i = 1$, $T_0 = p$, and $v = p$ where p is a single unlabeled vertex. In the general step T_{i-1} is a tree with unlabeled root and all other vertices labeled by positive integers and v is a vertex of T_{i-1}. If $a_i \neq a_j$ for all $1 \leq j < i$ we derive T_i from T_{i-1} by adding a new child with the label a_i to the right of the children of v. Then we move to the next term of u, v remains the same. If a_i appears in u before we put T_i equal to T_{i-1}, v equal to the vertex labeled by a_i, and we move to the next term of u. The procedure terminates for $i = 2n$, we forget the labels and set $G(u) = T = T_{2n}$.

Lemma 2.1 *The mapping $G: \mathcal{R}(n) \to \mathcal{T}(n)$ is a bijection.*

Proof. The algorithm adds vertices in their good order and v traces T in the good order. Let us define the inverse of G. We take the vertices $(v_0, v_1, \dots, v_n)$ of $T \in \mathcal{T}(n)$ in their good order and write down for each v_i first the index i and then, left to right, the indices of its children. We set $G^{-1}(T)$ equal to the sequence obtained, the initial 0 deleted. Clearly, G and G^{-1} are inverses of one another. □

The mapping G corresponds to the bredth-first search in T. We remark that *abab*-free partitions (the avoidance of *abab* is defined in a way analogous to that of *abba*) with n blocks, each of size 2, can be put in a bijective correspondence with $\mathcal{T}(n)$ as well. These partitions are proper bracketings with n brackets. The correspondence matching them with trees is based on the depth-first search and is well known.

A *gap* in a finite sequence $u = a_1 a_2 \dots a_l$ is the space between two consecutive terms or the space before a_1 or the space after a_l. The set of gaps $g(u)$ has $l + 1$ elements. Suppose $u = a_1 \dots a_{2n} \in \mathcal{R}(n)$ and let $x = a_i = a_j$, $i < j$. The *first* (the *second*) *gap of* x is the gap following after a_i (after a_j). The *first gap of* u is the gap of u before a_1.

The *gaps of a vertex* v of a tree $T \in \mathcal{T}(n)$ are the wedge-shaped spaces into which the edges going up from v divide the neighborhood of v. A vertex with d children has $d + 1$ gaps. In particular, any leaf has exactly one gap. The set $g(T)$ of all gaps has $2n + 1$ elements. For $e = v_1 v_2$ an edge of T we call the leftmost gap of v_2 the *top gap of* e and the gap of v_1 to the right of e the *bottom gap of* e. The *first gap of* T is root's leftmost gap.

The mapping G induces a bijection $G^* : g(u) \to g(G(u))$. Suppose $u = a_1 a_2 \dots a_{2n} \in \mathcal{R}(n)$ is in the canonical form. The first gap of u is sent to the first gap of $T = G(u)$. The first (the second) gap of an integer x is sent to the top (to the bottom) gap of the edge whose endvertex is the xth one in the good order, we remind that the root is the 0th vertex.

A *gap tree* is a pair (T, s) where T is a tree and $s : g(T) \to \mathbf{N}_0$ is an integer mapping. Its *size* is $|E(T)| + \sum s(g)$ where we sum over $g(T)$. The set of gap trees of size n is denoted by $\mathcal{GT}(n)$. A vertex is *solitary* (*young*) in (T, s) if it is solitary (young) in T and $s(g) = 0$ for its leftmost gap (for its only gap).

Any sequence $u \in \mathcal{S}(n)$ can be encoded by a gap tree $H(u) = (T, s)$ of size n as follows. We decompose u into (u^*, t) where $u^* \in \mathcal{R}(m)$ is the subsequence of 2-element blocks and $t : g(u^*) \to \mathbf{N}_0$ counts the numbers of 1-element blocks in the gaps of u^*. We set $T = G(u^*)$ and $s(G^*(g)) = t(g)$ for any $g \in g(u^*)$. For an example illustrating H see Fig. 2.

Lemma 2.2 *The above mapping $H : \mathcal{S}(n) \to \mathcal{GT}(n)$ is a bijection. Moreover, it maps pretty partitions to those and only those gap trees which have no solitary vertex.*

Proof. Check the definitions. □

Thus the desired bijection $F : \mathcal{P}(n) \to \mathcal{U}(n)$ is constructed as soon as we exhibit a bijection matching gap trees of size n without solitary vertices with those having at least one solitary vertex. We denote the former set as $\mathcal{GT}_0(n)$ and the latter set as $\mathcal{GT}_1(n)$.

Bijections for gap trees

It is was not too obvious to us how to match the elements of $\mathcal{GT}_0(n)$ and $\mathcal{GT}_1(n)$. However, we could easily see the bijection between the sets $\mathcal{GT}^0(n)$ and $\mathcal{GT}^1(n)$. The former set consists of gap trees of size n with no young vertex and the latter set of gap trees of size n with at least one young vertex.

Lemma 2.3 *There is a bijection $I : \mathcal{GT}^1(n) \to \mathcal{GT}^0(n)$.*

Proof. Suppose (T, s) is a gap tree with young vertices, let v be the leftmost one. We transform (T, s) into a gap tree $I((T, s)) = (U, t)$ of the same size and with no young vertex. Let (T_0, s) be the gap subtree of (T, s) rooted in the root r which is lying to the right of v. There is to distinguish two cases.

1. If there is nothing to the right of v — (T_0, s) consists of r only and $s(g) = 0$ for g the rightmost gap of r in T — we delete v and put $t(h) = s(g') + 1$ where h is now the rightmost gap of r in U and g' was the second rightmost gap of r in T (h arises by merging g and g'). The values of t on other gaps equal to those of s.

2. If there is anything to the right of v — (T_0, s) has more than one vertex or $s(g) > 0$ — (T_0, s) is cut off from r (r gets duplicated for a while) and is glued to v. We set $t(h) = 0$, on other gaps t retains the values of s.

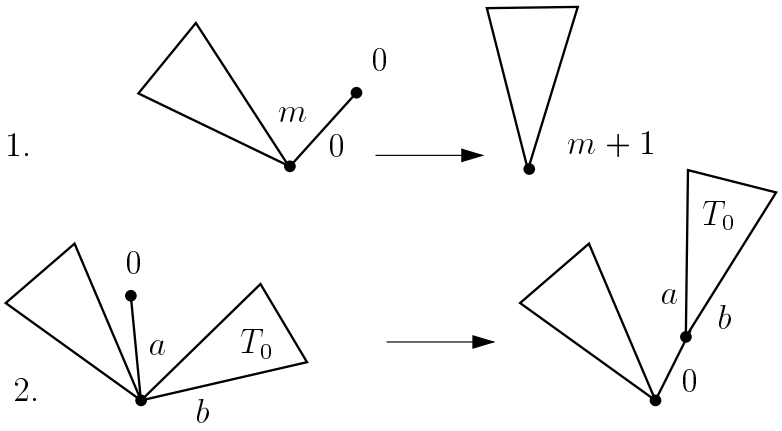


Figure 1: The bijection I.

The transformation is depicted schematicly on the above figure (0, a, b, and m stand for the values of s and t on the corresponding gaps). All young vertices are destroyed. To reconstruct (T, s) from (U, t) we check first whether $t(h) > 0$ for h the rightmost gap of the root of U. If yes we proceed backwards via (1) otherwise via (2). Hence I is a bijection. □

It remains to work out the bijections between $\mathcal{GT}_0(n)$ and $\mathcal{GT}^0(n)$, and $\mathcal{GT}_1(n)$ and $\mathcal{GT}^1(n)$. We prove more. We present a bijection $J : \mathcal{GT}(n) \to \mathcal{GT}(n)$ that maps a gap tree with k solitary vertices to a gap tree with k young vertices.

We need few definitions. For T a tree the rightmost branch $(x_1, x_2, \ldots, x_k)$, $x_1 = r$, the final leaf x_{k+1} is omitted, is called the *right side of* T. The *top side of* T $(y_1, y_2, \ldots, y_l)$ consists of the vertices $y_1, \ldots, y_m$ of the highest layer, ordered from right to left, and of the vertices $y_{m+1}, \ldots, y_l$ of the highest but one layer which lie to the right of y_1's parent, again taken from right to left. An *encoding sequence* is a sequence $((a_1, b_1), \ldots, (a_m, b_m))$ of pairs of positive integers satisfying

$$b_1 = 1 \text{ and } b_i \leq a_{i-1} + b_{i-1} - 1 \text{ for } i = 2, \ldots, m.$$

Its *size* is $a_1 + a_2 + \ldots a_m$. The one term sequence $((0,1))$ is defined to be an encoding sequence too.

Lemma 2.4 *There is a bijection $J : \mathcal{GT}(n) \to \mathcal{GT}(n)$ that maps a gap tree with k solitary vertices to a gap tree with k young vertices.*

Proof. Suppose $z = ((a_1, b_1), \ldots, (a_m, b_m))$ is an encoding sequence. We show two ways to decode it and to obtain a tree T with $a_1 + a_2 + \ldots + a_m$ edges.

The sequence $z = ((0,1))$ is decoded in both ways as the one vertex tree. We start the first decoding with drawing, from bottom to top, a path of a_1 edges. We denote this initial tree as T_1. In the general step, to derive T_{i+1} from T_i, we draw from bottom to top and to the right of T_i a path P of a_{i+1} edges starting in the b_{i+1}th vertex of the right side of T_i. P is clearly the final segment of the right side of T_{i+1}. On the end we set $T = T_m$. We denote this decoding as J_1. The order in which the edges of T are drawn is called the J_1-*order*.

The second decoding is a similar one, the difference being that T_1 is the broom of a_1 edges (the root has a_1 children, all of them are leaves) and that in the general step we join to the b_{i+1}th vertex of the top side of T_i a broom of a_{i+1} edges. Their endpoints become the initial segment of the top side of T_{i+1}. Each broom is drawn from right to left. This decoding is denoted as J_2, the J_2-*order* is defined analogously.

Both decodings are bijections from the set of encoding sequences of size n to $\mathcal{T}(n)$. Hence $J_3 = J_1 \circ J_2^{-1}$ is a bijection on $\mathcal{T}(n)$. Since solitary (young) vertices correspond in J_2 (in J_1) exactly to the terms $(1,1)$ of the encoding sequence, we conclude that J_3 has the property stated in the lemma. It remains to extend it to $\mathcal{GT}(n)$.

We define the bijection $J_3^* : g(T) \to g(J_3(T))$ as follows. Suppose g is the top (the bottom) gap of the mth edge, in the J_2-order, of T. We set $J_3^*(g)$ equal to the top (to the bottom) gap of the mth edge, in the J_1-order, of $J_3(T)$. The first gap of T is sent, of course, to the first gap of $J_3(T)$.

Finally, let $(T, s) \in \mathcal{GT}(n)$. We define $J((T,s)) = (J_3(T), t)$ where $t(J_3^*(g)) = s(g)$ for any $g \in g(T)$. Clearly g is the leftmost gap of a solitary vertex in T iff $J_3^*(g)$ is the gap of a young vertex in $J_3(T)$. Thus J has the property stated. $\square$

Our construction of the bijection $F : \mathcal{P}(n) \to \mathcal{U}(n)$ is complete: $F = H^{-1} \circ J^{-1} \circ I^{-1} \circ J \circ H$. We illustrate it for a specific partition on Fig. 2. In the top row the encoding sequence is $((2,1),(2,2),(1,2))$ and in the bottom row $((1,1),(1,1),(2,1),(1,1))$. In I^{-1} we proceed backwards via (2).

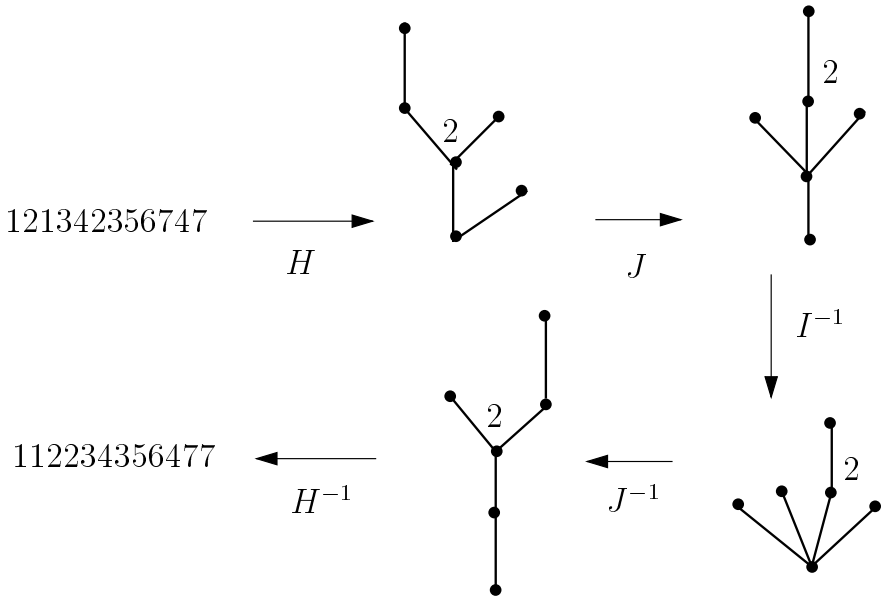


Figure 2: The bijection F.

3 Concluding remarks

The reader may wonder about the numbers $a_n = |\mathcal{P}(n)| = |\mathcal{U}(n)|$,

$$\{a_n\}_{n\geq 1} = \{1, 3, 11, 45, 197, 903, 4279, 20793, 103049, \ldots\}.$$

These are Schröder numbers [3], A1003 in [4], one of their explicit forms [2] is

$$a_n = \sum_{l=0}^{n-1} \frac{2^l}{n-l} \binom{n}{l+1} \binom{n-1}{l}.$$

The interested reader will find more references and expressions for Schröder numbers in [2] or in [4].

Our construction could be translated back to partitions but we prefer tree structures because they enable visual insight in the whole matter. We plan to prove along similar lines two other identities of [1] concerning *abba*-free and *abab*-free partitions.

References

[1] M. Klazar, On *abab*-free and *abba*-free set partitions, *Europ. J. of Combinatorics* **17** (1996), 53–68.

[2] M. Klazar, On numbers of Davenport-Schinzel sequences, submitted.

[3] E. Schröder, Vier combinatorische Probleme, *Zeitschrift für Mathematik und Physik* **15** (1870), 361–376.

[4] N. J. A. Sloane and collaborators, The On-Line Encyclopedia of Integer Sequences, `sequences@research.att.com`

PALINDROMIC PRIME PYRAMIDS

G. L. HONAKER, JR.
Bristol Virginia Public Schools
Bristol, VA 24201
sci-tchr@3wave.com

CHRIS K. CALDWELL
University of Tennessee at Martin
Martin, TN 38238
caldwell@utm.edu

Have you ever seen the great stone pyramids of ancient Egypt or Central America? For over 5000 years, mankind has been building, visiting, and even sleeping in pyramids. When Memphis, Tennessee, decided to build a new arena in 1991, they chose the shape of a pyramid—this time constructed of steel and glass rather than rock and rubble. In this paper we also build pyramids, ours built of the unbreakable stones of mathematics: the primes. But not just any primes, we have chosen the symmetry of nested palindromes as mortar. For example, beginning with the prime 2, we can build two pyramids of height five. (Unlike the ancients, we build our pyramids from the top down.)

2	2
929	929
39293	39293
7392937	3392933
373929373	733929337

Here each step is a palindromic prime with the previous step as its central digits. These two pyramids are the tallest that can be built beginning with the prime 2.

The tallest such pyramids that can be built from the other one-digit primes are as follows:

		5	5	5	
3	3	151	353	757	7
131	131	31513	33533	37573	373
11311	71317	3315133	1335331	9375739	93739

Like many that have come before us, we ask how can we build them higher? For example, if instead of just one-digit primes, we begin with larger palindromic primes, can they be taller? If instead of adding just one digit to each side, we allow two or more, how much taller can we get? Are these pyramids always finite? Join us on a quick tour as we seek answers to these questions, and pose others for our readers.

Simple Step Pyramids

Starting with a single digit prime and at each level adding just one digit to each side, we found the tallest possible prime-pyramids (using nested palindromes) had height five. This is because there are only four possible digits we can add at each step: 1, 3, 7, and 9. Starting with larger primes is unlikely to help much, but there are so many to choose from that we might get lucky. For example, Felice Russo [10, 11 seq. A046210] found the following truncated palindromic prime pyramid of height nine.

7159123219517
371591232195173
33715912321951733
7337159123219517337
973371591232195173379
39733715912321951733793
3397337159123219517337933
933973371591232195173379339
39339733715912321951733793393

However, if instead we add two digits on each side, there are forty pairs of digits we can add to each end (and still avoid our steps being divisible by 2 or 5). Starting with the prime 2, the tallest that can be built (with step two) has height 26. In fact, there are two pyramids of this height. One of these is shown in figure 1. The other is the pyramid ending in the following 101-digit prime:

Insert figure 1 on a page near here
---

1 3189272993 3733012747 5151938943 3901197127
2339635702 0753693327 2179110933 4983915157 4721033733 9927298131

How do we know these are the tallest? Using UBASIC [3] we started with 2 and built *every possible pyramid*--at each step discarding those for which the new number was not a Fermat probable-prime [7, pg. 140]. Then for those pyramids of maximum height, we used UBASIC's application program APRT-CL [5] to complete primality proofs for every step. We also applied this approach to pyramids starting with the other one-digit primes. There are three pyramids tied for tallest starting with the prime 3, each of height 28. There is one each starting with the primes 5 and 7, both of height 29. Further information is available on-line [4].

Surely increasing the step size to three (or more) should increase the height, but by how much? How many pyramids would we have to check for an exhaustive search? We address these questions in the next section.

Heights and Heuristics

First, let $l(n)$ be the number of digits (the length) of n. Let $f(n,h,d)$ be the number of palindromic primes pyramids with height h (not necessarily the maximal height), beginning with n and with step size d. For example, $f(2,1,d) = 1$ (there is only one pyramid starting with 2 and

height 1, that is just "2"). However, f(101,2,2) = 4 since there are four pyramids starting with 101 of height 2 and step 2:

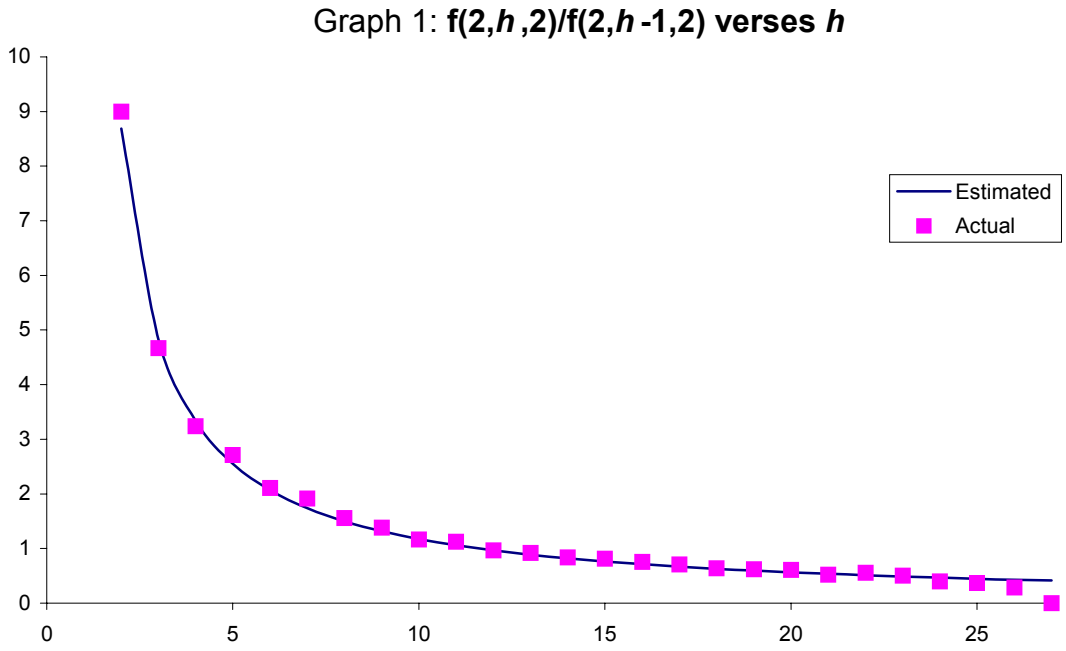
101	101	101	101
3310133	7310137	9110119	9610169

We will attempt to estimate f(n,h,d) and use this to estimate the maximum height.

Recall that the *prime number theorem* [7, pp. 225-227] states that the number of primes less than x is approximately $x/\ln x$. (Technically, the theorem says these quantities are asymptotic--so the larger x is, the better this estimate is). One interpretation of this theorem is that the probability of a random integer the size of the integer x being prime is about $1/\ln x$. When we move to the next step of a pyramid, there are 10^d integers to try, so if these new numbers behave as a random sample we would expect

$$\frac{f(n,h,d)}{f(n,h\text{-}1,d)} \approx \frac{10^d}{(l(n) + 2(h-1)d)\ln 10}. \tag{1}$$

In graph 1 we see this rough estimate (graphed as a solid curve) compared to the actual ratios (graphed as individual squares) for n=2 and d=2.



These match surprisingly well! The drop off at the end is caused by the low numbers—we might expect 40% of two numbers to be prime, but in actuality only 0, 1, or 2 of them can be, not 0.80 of them. This is typical since this type of heuristic estimate (educated guess) works best for large numbers (see, for example, [2] or [12]).

As h grows, the ratio in (1) soon becomes one and then decreases to zero, so we would expect the number of pyramids to start decreasing at that point and then drop off to zero. From this we make the following conjecture.

Conjecture: All palindromic prime pyramids with fixed step size are finite.

We can predict more with this heuristic model. Repeatedly using the estimate (1), we have

$$\mathrm{f}(n,h,d) \approx \left(\frac{10^d}{\ln 10}\right)^{h-1} \frac{\mathrm{f}(n,1,d)}{(l(n)+2d(h-1))\,(l(n)+2d(h-2))\cdot\ldots\cdot(l(n)+2d)}. \tag{2}$$

The denominator of the second term is Pochhammer's symbol and can be expressed via the gamma function[1] [1, eq. 6.1.22]. This yields the following.

$$\mathrm{f}(n,h,d) \approx \left(\frac{10^d}{2d\ln 10}\right)^{h-1} \frac{\Gamma\left(\frac{l(n)}{2d}+1\right)}{\Gamma\left(\frac{l(n)}{2d}+h\right)}. \tag{3}$$

This estimate is one when h=1 (the top of the pyramid). Just past where this estimate is one again (for some larger h), we would expect to have the pyramid with greatest height. Using a computer program such as Maple [13] it is easy to solve for this value. Sadly, we can only test our estimates for small values of d where we expect the greatest relative error, but the comparisons still are heartening—see Table 1.

Table 1: **The average maximum height of pyramids**

length		**step d = 1**		**step d = 2**		**step d=3**	**step d=4**
***l*(*n*)**	number*	predicted	**actual**	predicted	**actual**	predicted	predicted
1	4	3.55	**3.75**	26.8	**28.0**	193	1471
3	15	1.31	**2.53**	25.0	**25.8**	191	1469
5	93	1**	**2.10**	23.3	**24.3**	190	1467
7	668	1**	**1.79**	21.7	**22.1**	188	1466
9	5172	1**	**1.58**	20.1	**20.2**	186	1464

* The number of starting values (palindromic primes) n of the given length $l(n)$
** The estimate (3) is only one once for these values of $l(n)$ (when h=1)

We found the actual values in table one by exhaustive search: for each palindromic prime of the given length, we found the pyramid of maximum height (by finding all pyramids beginning with this prime). We then averaged over all the palindromic primes of this length.

Notice that even for d=3 and n=2, this exhaustive approach would be beyond the world's current computing ability because when the height was 73, (2) predicts there should be almost 10^{30} pyramids to deal with. However, we can still test this heuristic by keeping a fixed number

[1] $\Gamma(n)$ is the analytic continuation of the familiar factorial function: $\Gamma(n+1) = n!$ for positive integers n.

of pyramids at each step. In our test we kept a maximum of 160 pyramids at each height, so beginning at h=74 (when the ratio (2) is one) and continuing until we get a product less than one, we predict we should find a maximum height of about 103. Starting with the primes 2, 3, 5 and 7 we found maximal pyramids of heights 94, 101, 102 and 100 respectively. This is reasonable agreement for the relatively small number of pyramids (a maximum of 160) involved. (Again, these prime pyramids are available on the web [4])

Related Sequences

Keeping the step size fixed (apparently) forever binds our pyramids to a finite height. But suppose we instead allow any step size? An argument similar to the one above suggests that for any starting prime we should be able to build as high as we like, though the taller the pyramids get the larger our step size must be (on the average).

There is one case that is especially interesting: Suppose we ask that each row be the smallest prime that can be used. Then our pyramid would begin as follows:

2
727
37273
333727333
93337273339
309333727333903
1830933372733390381
9218309333727333903 8129
39218309333727333903 81293
133392183093337273339038 1293331
1813339218309333727333903 8129333181

When the first author built this pyramid, he was able to verify the primality of the first 33 rows.

This pyramid has also been presented as a sequence a_1, a_2, a_3, …. To do this let a_1=2 and then for each positive integer n, let a_{n+1} be the smallest palindromic prime with a_n as the central digits [11, seq. A053600]. We can condense this sequence by writing a_1, followed by the digits added on the left at each stage. Carlos Rivera [8] extended the first author's 33 terms using a probabilistic primality test and found the condensed sequence [11, seq. A052091] (most likely) begins:

2, 7, 3, 33, 9, 30, 18, 92, 3, 133, 18, 117, 17, 15, 346, 93, 33,
180, 120, 194, 126, 336, 331, 330, 95, 12, 118, 369, 39, 32, 165,
313, 165, 134, 13, 149, 195, 145, 158, 720, 18, 396, 193, 102,
737, 964, 722, 156, 106, 395, 945, 303, 310, 113, 150, 303, 715,
123

Finally, Russo took a different approach to palindromic prime pyramids, and asked what was the smallest palindromic prime a_n that generates a prime pyramid of maximum height n? This sequence [11, seq. A046210] begins 11, 131, 2, 929, 10301, 16361, 10281118201, 35605550653, 7159123219517…

Conclusion

As we look around in the world, we see many variations on the basic pyramids of Egypt. Above we have mentioned just a few of the variations on our pyramids that have appeared since the first author proposed the idea. For even more variations, look on-line [4, 6, 9, 11].

We have left many open questions and leave the reader with the most basic of challenges: build them higher! Perhaps you can develop a way of finding the tallest pyramids with fixed step sizes--something far better than exhaustive search. Or perhaps can you prove (rather than just heuristically suggest) that fixed step size pyramids are finite. We built pyramids in decimal (base 10), why not try another base (e.g., binary)?

In all cases, we would be glad to hear of your results.

References

1. M. Ambramowitz and I. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1974.
2. P. T. Bateman and R. A. Horn, "A heuristic asymptotic formula concerning the distribution of prime numbers," *Math. Comp.*, **16** (1962) 363-367.
3. C. Caldwell, "UBASIC," *J. Recreational Math.*, **25**:1 (1993) 47-54. (UBASIC is avaialable on-line at http://archives.math.utk.edu/software/msdos/number.theory/ubasic/.)
4. C. Caldwell, "Palindromic Prime Pyramids—on-line supplement," http://www.utm.edu/~caldwell/supplements.
5. H. Cohen and A. K. Lenstra, "Implementation of a new primality test," *Math. Comp.*, **48** (1987) 103-121.
6. P. De Geest, "Palindromic Numbers and Other Recreational Topics," http://www.ping.be/~ping6758/index.shtml.
7. P. Ribenboim, *The New Book of Prime Number Records,* 3rd ed., Springer-Verlag, New York, 1995.
8. C. Rivera, Private correspondence to De Geest and Honaker, 22 January 2000. (All 164 rows are available on the page http://www.ping.be/~ping6758/palprim3.htm.)
9. C. Rivera, "The prime puzzles & problems connection," http://www.primepuzzles.net/
10. F. Russo, Private correspondence to Honaker, 28 Jan 2000
11. N. J. A. Sloane, "The on-line encyclopedia of integer sequences," http://www.research.att.com/~njas/sequences/.
12. S. Wagstaff, "Divisors of Mersenne Numbers," *Math. Comp.*, **40**:161 (January 1983) 385-397.
13. "Waterloo Maple" (program), http://www.maplesoft.com/, Waterloo Maple Inc., Ontario Canada N2L 6C2

Figure 1: **A palindromic prime pyramid of step size two**

2
30203
903020309
3790302030973
98379030203097389
969837903020309738969
99969837903020309738969999
729996983790302030973896999927
99729996983790302030973896999279927
9099729996983790302030973896999279909
949099729996983790302030973896999279909049
77949099729996983790302030973896999279909497797
7797949099729996983790302030973896999279909497797
177977949099729996983790302030973896999279909497797771
75177977949099729996983790302030973896999279909497797157
7375177977949099729996983790302030973896999279909497797715737
727375177977949099729996983790302030973896999279909497797715737727
98727375177977949099729996983790302030973896999279909497797715737278789
309872737517797794909972999698379030203097389699927990949779771573727890307
70309872737517797794909972999698379030203097389699927990949779771573727890307
3970309872737517797794909972999698379030203097389699927990949779771573727890307793
353970309872737517797794909972999698379030203097389699927990949779771573727890307935353
36353970309872737517797794909972999698379030203097389699927990949779771573727890307935363
3336353970309872737517797794909972999698379030203097389699927990949779771573727890307935363333
343336353970309872737517797794909972999698379030203097389699927990949779771573727890307935363334343
99343336353970309872737517797794909972999698379030203097389699927990949779771573727890307935363334399

ЗАДАЧА ИОСИФА ФЛАВИЯ (JOSEPHUS PROBLEM)

М.А.Алексеев

Рассмотрим задачу, носящую имя известного историка первого века Иосифа Флавия. Ее формулировка связана с такой легендой [1]. Во время иудейской войны отряд Иосифа, состоящий из 41 человека, был окружен римлянами. Не желая попасть в плен, воины решили выстроиться в круг и убивать каждого третьего из живых до тех пор, пока не останется ни одного человека. Иосиф счел подобный конец бессмысленным и быстро вычислил для себя спасительное "последнее" место.

В более общей формулировке задача выглядит так: n человек выстраиваются по кругу и начинают счет по часовой стрелке, при котором каждый q-ый человек выбывает из круга. Необходимо определить, кто же останется последним.

Для упрощения дальнейшего повествования занумеруем людей в круге по часовой стрелке числами от 0 до $n-1$, причем номер 0 дадим человеку, с которого начинается счет. Номер искомого "последнего" человека обозначим через $J_q(n)$.

Эта задача не так проста, как кажется на первый взгляд, и в полном объеме не решена до сих пор. Под решением здесь понимается либо нахождение простой замкнутой формулы для $J_q(n)$, либо указание достаточно быстрого алгоритма для вычисления $J_q(n)$ на компьютере. Удовлетворительные результаты получены только в случае $n \gg q$. Как будет показано далее, в этом случае существует алгоритм для вычисления $J_q(n)$ требующий $O((q-1)\ln n)$ действий.

Сначала попробуем подсчитать $J_q(n)$ для малых значений n.

Случай $n = 1$ тривиален: у нас всего одно место — оно и является "последним". Поэтому $J_q(1) = 0$.

В случае $n = 2$ все решает четность числа q: если q четно, то выбывает человек по номером 1; если нечетно — по номером 0. Поэтому можно утверждать, что номер оставшегося $J_q(2) = q \bmod 2$.

Случай $n = 3$ сложнее. Первым из круга выбывает человек под номером $(q-1) \bmod 3$. В кругу останутся 2 человека — ситуация предыдущего случая с единственным отличием: счет начинается с человека под номером $q \bmod 3$. Но это расхождение легко устранить: "повернем" известный нам ответ $J_q(2)$ на $q \bmod 3$ номеров по часовой стрелке. Численно это выглядит так

$$J_q(3) = (J_q(2) + (q \bmod 3)) \bmod 3 = (J_q(2) + q) \bmod 3.$$

Аналогичные рассуждения позволяют установить общую рекуррентную формулу.

Лемма 1. *Справедлива рекуррентная формула*

$$J_q(n+1) = (J_q(n) + q) \bmod (n+1). \tag{1}$$

Заметим, что если $J_q(n) + q < n + 1$, то операция взятия по модулю в формуле (1) никак не влияет на результат, а потому $J_q(n+1) = J_q(n)+q$. Попробуем двинуться далее: если $J_q(n+1) + q < n + 2$, то $J_q(n+2) = J_q(n+1) + q = J_q(n) + 2q$.

Возникает вопрос: до каких пор мы можем получать такие относительно простые выражения? Ответ дает неравенство $J_q(n) + sq < n + s$. Пусть $s = s_0$ его максимальное решение, тогда для всех $t \leqslant s_0$ справедлива формула $J_q(n + t) = J_q(n) + tq$. Можно также сделать один шаг "за s_0" и получить для $t = s_0 + 1$ формулу $J_q(n + t) = (J_q(n) + tq) \bmod (n + t)$.

Лемма 2. *Для* $t = 0, 1, \ldots, \left\lceil \frac{n - J_q(n)}{q-1} \right\rceil$ *справедлива формула*

$$J_q(n + t) = (J_q(n) + tq) \bmod (n + t). \tag{2}$$

При $n < q$ эта формула не позволяет двигаться с шагом большим 1 и по сути ничем не отличается от (1).

Зато при $n \geqslant q$ формула (2) позволяет быстро вычислять $J_q(n)$ по заранее вычисленному значению $J_q(q - 1) = J_q(q)$. В частности поэтому особый интерес представляет вычисление "стартового" значения $J_q(q)$. На данный момент не известно алгоритма, позволяющего вычислять значение $J_q(q)$ быстрее чем за $O(q)$ операций, или, другими словами, не найдено формулы существенно лучшей формулы (1). Последовательность $\{J_q(q)\}_{q=1}^{\infty}$ достаточно известна и указана, например, в [2].

Идея упомянутого ускорения вычислений состоит в том, что с помощью формулы (2) из очередного вычисленного значения $J_q(m)$ мы сразу переходим к $J_q(m + \left\lceil \frac{m - J_q(m)}{q-1} \right\rceil)$, и такими "семимильными" шагами двигаемся от значения $J_q(q-1)$ к $J_q(n)$. Но прежде чем переходить к деталям данного процесса, давайте упростим формулу (2) при $t = \left\lceil \frac{n - J_q(n)}{q-1} \right\rceil$.

Теорема 3. *При* $n \geqslant q$ *и* $t = \left\lceil \frac{n - J_q(n)}{q-1} \right\rceil$ *справедлива формула*

$$J_q(n + t) = (J_q(n) - n) + t(q - 1).$$

Построим последовательность "опорных" значений

$$m_{k+1} = m_k + \left\lceil \frac{m_k - J_q(m_k)}{q-1} \right\rceil = \left\lceil \frac{qm_k - J_q(m_k)}{q-1} \right\rceil, \quad k = 0, 1, \ldots \tag{3}$$

Здесь у нас есть свобода в выборе начального значения m_0. Можно, например, считать $m_0 = q - 1$.

Формула (3) неудобна тем, что в нее входит неизвестное значение $J_q(m_k)$. Нашей ближайшей целью будет избавиться в рекуррентной формуле для m_k от зависимости от $J_q(m_k)$.

Теорема 3 позволяет получить рекуррентное соотношение для $J_q(m_k)$

$$J_q(m_{k+1}) = J_q(m_k) - m_k + (m_{k+1} - m_k)(q-1) = J_q(m_k) + (q-1)m_{k+1} - qm_k.$$

Просуммировав это равенство по k от 0 до $k-1$, получим

$$J_q(m_k) = J_q(m_0) + (q-1)m_k - \sum_{j=0}^{k-1} m_j - (q-1)m_0. \tag{4}$$

Подставляя это выражение в (3), мы получаем формулу

$$m_{k+1} = \left\lceil \frac{\sum_{j=0}^{k} m_j - J_q(m_0)}{q-1} \right\rceil + m_0,$$

Теперь последовательность $m_0 < m_1 < \ldots$ легко построить отталкиваясь только от $J_q(m_0)$. Значения J_q в точках этой последовательности также легко определяются по формуле (4).

Понятно, что для любого $n \geqslant m_0$ мы можем найти такое k, что $m_{k-1} \leqslant n < m_k$. Теорема 3 вкупе с формулой (3) позволяет утверждать

Следствие 4. *Пусть $m_{k-1} \leqslant n < m_k$. Тогда*

$$J_q(n) = J_q(m_0) + qn - \sum_{j=0}^{k-1} m_j - (q-1)m_0 = qn - (q-1)m_k + J_q(m_k). \tag{5}$$

Наша цель достигнута: по значению $J_q(m_0)$ мы научились быстро находить значение $J_q(n)$ для $n \geqslant m_0$. А вот, чтобы оценить быстроту описанного способа вычислений, лучше взглянуть на формулы немного с другой стороны.

Тождество (4) можно переписать в виде

$$(q-1)m_k - J_q(m_k) = ((q-1)m_0 - J_q(m_0)) + \sum_{j=0}^{k-1} m_j. \tag{6}$$

Введем обозначение $D_k^{(q)} \stackrel{\text{def}}{=} (q-1)m_k - J_q(m_k)$ соответствующее аналогичному (с точностью до сдвига индексов) в [1]. Тогда формулу (5) можно записать как

$$J_q(n) = nq - D_k^{(q)}. \tag{7}$$

При этом формулы (3) и (6) можно соответственно привести к виду

$$m_k = \left\lceil \frac{D_k^{(q)}}{q-1} \right\rceil \tag{8}$$

и

$$D_k^{(q)} = D_0^{(q)} + \sum_{j=0}^{k-1} m_j.$$

Преобразуем последнюю формулу

$$D_k^{(q)} = D_0^{(q)} + \sum_{j=0}^{k-1} m_j = D_{k-1}^{(q)} + m_{k-1} = D_{k-1}^{(q)} + \left\lceil \frac{D_{k-1}^{(q)}}{q-1} \right\rceil = \left\lceil \frac{q}{q-1} D_{k-1}^{(q)} \right\rceil. \qquad (9)$$

Из формулы (8) следует, что неравенство $m_{k-1} \leqslant n < m_k$ равносильно тому, что значение $D_k^{(q)}$ является минимальным большим чем $n(q-1)$. Отсюда и формулы (9) заключаем, что число шагов

$$k \quad \leqslant \quad \log_{\frac{q}{q-1}} \frac{n(q-1)}{D_0^{(q)}} \quad \overset{n \gg q}{\sim} \quad (q-1)\ln n.$$

В заключение докажем один интересный результат, по-видимому, впервые полученный в работе [3].

Теорема 5.

$$D_n^{(3)} = \left\lfloor \left(\frac{3}{2}\right)^n C \right\rfloor,$$

где C — конструктивно определяемая константа (зависящая от $D_0^{(3)}$).

Докажем его, используя технику, описанную в [4] (с.34).

Доказательство. Перепишем формулу (9) при $q = 3$ в виде

$$D_k^{(3)} = \frac{3}{2} D_{k-1}^{(3)} + \alpha_k, \qquad (10)$$

где α_k принимает значения 0 или $\frac{1}{2}$ в зависимости от того, является ли число $\frac{3}{2}D_{k-1}^{(3)}$ целым или нет.

"Развернем" рекуррентную формулу (10)

$$D_n^{(3)} = \tfrac{3}{2} D_{n-1}^{(3)} + \alpha_n = \tfrac{3}{2}\left(\tfrac{3}{2} D_{n-2}^{(3)} + \alpha_{n-1}\right) + \alpha_n = \left(\tfrac{3}{2}\right)^2 D_{n-2}^{(3)} + \tfrac{3}{2}\alpha_{n-1} + \alpha_n = \ldots =$$
$$= \left(\tfrac{3}{2}\right)^n D_0^{(3)} + \left(\tfrac{3}{2}\right)^{n-1} \alpha_1 + \left(\tfrac{3}{2}\right)^{n-2} \alpha_2 + \ldots + \alpha_n = \left(\tfrac{3}{2}\right)^n \left(D_0^{(3)} + \sum_{k=1}^{n} \left(\tfrac{2}{3}\right)^k \alpha_k \right).$$

Определим константу C формулой

$$C \overset{\text{def}}{=} D_0^{(3)} + \sum_{k=1}^{\infty} \left(\frac{2}{3}\right)^k \alpha_k = D_0^{(3)} + \frac{2}{3}\alpha_1 + \left(\frac{2}{3}\right)^2 \alpha_2 + \ldots \qquad (11)$$

Рассмотрим разность

$$\begin{aligned}\left(\tfrac{3}{2}\right)^n C - D_n^{(3)} &= \left(\tfrac{3}{2}\right)^n \left(D_0^{(3)} + \sum_{k=1}^{\infty} \left(\tfrac{2}{3}\right)^k \alpha_k\right) - \left(\tfrac{3}{2}\right)^n \left(D_0^{(3)} + \sum_{k=1}^{n} \left(\tfrac{2}{3}\right)^k \alpha_k\right) = \\ &= \left(\tfrac{3}{2}\right)^n \sum_{k=n+1}^{\infty} \left(\tfrac{2}{3}\right)^k \alpha_k = \sum_{k=n+1}^{\infty} \left(\tfrac{2}{3}\right)^{k-n} \alpha_k\end{aligned} \tag{12}$$

Для доказательства утверждения теоремы достаточно показать, что

$$0 \leqslant \left(\frac{3}{2}\right)^n C - D_n^{(3)} < 1. \tag{13}$$

Левая часть этого неравенства тривиальна ввиду формулы (12) и неравенства $\alpha_k \geqslant 0$.

Для доказательства правой части воспользуемся (12), неравенством $\alpha_k \leqslant \frac{1}{2}$ и формулой для суммы геометрической прогрессии

$$\left(\frac{3}{2}\right)^n C - D_n^{(3)} = \sum_{k=n+1}^{\infty} \left(\frac{2}{3}\right)^{k-n} \alpha_k \leqslant \sum_{k=n+1}^{\infty} \left(\frac{2}{3}\right)^{k-n} \frac{1}{2} = 1. \tag{14}$$

Мы почти у цели: осталось только показать, что разность $\left(\frac{3}{2}\right)^n C - D_n^{(3)}$ не может равняться 1 ни при каком n. Предположим противное, т.е. что $\left(\frac{3}{2}\right)^n C - D_n^{(3)} = 1$ при некотором n, но тогда из (14) получим, что $\alpha_k = \frac{1}{2}$ при всех $k > n$. Последнее равносильно тому, что при $k \geqslant n$ все числа $D_k^{(3)}$ нечетны. Пусть 2^t максимальная степень 2-ки, которая делит число $D_n^{(3)} + 1$. Из формулы

$$D_{n+1}^{(3)} + 1 = \left(\frac{3}{2} D_n^{(3)} + \frac{1}{2}\right) + 1 = 3\frac{D_n^{(3)} + 1}{2}$$

следует, что максимальной степенью 2-ки, которая делит $D_{n+1}^{(3)} + 1$, будет 2^{t-1}. Аналогично $D_{n+2}^{(3)} + 1$ будет делиться максимум на 2^{t-2} и т.д., $D_{n+t}^{(3)} + 1$ будет делиться максимум на $2^0 = 1$, что означает, что число $D_{n+t}^{(3)} + 1$ нечетное, а число $D_{n+t}^{(3)}$ соответственно четное. Полученное противоречие завершает доказательство неравенства (13). □

Пример 6. *Если, следуя [1], считать* $D_0^{(3)} = 1$, *то численные значения для* α_k *при* $k = 1, 2, \ldots, 40$ *будут следующими:*

$\alpha_1 = \frac{1}{2}$	$\alpha_{11} = \frac{1}{2}$	$\alpha_{21} = 0$	$\alpha_{31} = 0$
$\alpha_2 = 0$	$\alpha_{12} = 0$	$\alpha_{22} = \frac{1}{2}$	$\alpha_{32} = 0$
$\alpha_3 = \frac{1}{2}$	$\alpha_{13} = 0$	$\alpha_{23} = \frac{1}{2}$	$\alpha_{33} = 0$
$\alpha_4 = \frac{1}{2}$	$\alpha_{14} = \frac{1}{2}$	$\alpha_{24} = 0$	$\alpha_{34} = 0$
$\alpha_5 = 0$	$\alpha_{15} = \frac{1}{2}$	$\alpha_{25} = \frac{1}{2}$	$\alpha_{35} = 0$
$\alpha_6 = 0$	$\alpha_{16} = 0$	$\alpha_{26} = 0$	$\alpha_{36} = \frac{1}{2}$
$\alpha_7 = 0$	$\alpha_{17} = \frac{1}{2}$	$\alpha_{27} = 0$	$\alpha_{37} = \frac{1}{2}$
$\alpha_8 = \frac{1}{2}$	$\alpha_{18} = 0$	$\alpha_{28} = \frac{1}{2}$	$\alpha_{38} = 0$
$\alpha_9 = \frac{1}{2}$	$\alpha_{19} = \frac{1}{2}$	$\alpha_{29} = 0$	$\alpha_{39} = \frac{1}{2}$
$\alpha_{10} = 0$	$\alpha_{20} = 0$	$\alpha_{30} = \frac{1}{2}$	$\alpha_{40} = 0$

Они позволяют вычислить $C = 1.622270503\ldots$ *с точностью* 10^{-7}.

Литература

[1] Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: Пер. с англ. — М.: Мир, 1998.

[2] On-Line Encyclopedia of Integer Sequences `http://www.research.att.com/~njas/sequences`

[3] A.M.Odlyzko, H.S.Wilf "Functional iteration and the Josephus problem", Glasgow Mathematical Journal 33 (1991), 235–240.

[4] Грин Д., Кнут Д. Математические методы анализа алгоритмов. — М.: Мир, 1987.

An Introduction to Digit Product Sequences

Paul A. Loomis

Department of Mathematics, Computer Science, and Statistics, Bloomsburg University, Bloomsburg, PA 17815

A problem [1] and two articles [4,5] in this *Journal* have considered properties of digit sum sequences - sequences generated by iterating the function $g(n) = n + \Sigma(n)$, where $\Sigma(n)$ is the sum of the digits of a natural number n. None, though, have considered the similar idea of a digit product sequence.[1] Let n be a natural number, written in base 10, and let $\Pi(n)$ equal the product of the nonzero digits of n. Let $f(n) = n + \Pi(n)$ and define a sequence recursively by $a_{n,0} = n$, $a_{n,k+1} = f(a_{n,k})$ for $k \geq 0$. Following the notation of [5], we let $\overline{n}$ denote the sequence generated by n. The first such sequence is $\overline{1} = 1, 2, 4, 8, 16, 22, 26, 38, 62, \ldots$. The first natural number not in $\overline{1}$ is 3; it generates the sequence $\overline{3} = 3, 6, 12, 14, 18, 26, \ldots$. Similarily, $\overline{5} = 5, 10, 11, 12, \ldots$, and $\overline{7} = 7, 14, \ldots$. Each of these sequences joins a previously generated sequence, which in turn joins $\overline{1}$. This joining is illustrated by the tree in Fig. 1, in which $a \to b$ if $f(a) = b$. Thus a natural question arises: does every sequence join the main sequence $\overline{1}$?

Conjecture 1. For any natural number n, the sequence $\overline{n}$ merges with $\overline{1}$. That is, given n, there exist nonnegative integers i, j such that $f^i(1) = f^j(n)$.

In 1991 I wrote a C program to confirm this conjecture for $n \leq 1,000,000$. In 2002, Tim Smith, a Bloomsburg University undergraduate, wrote a C++ program that extends this to 10^8 and provides the other numerical evidence used in this paper. The first "stubborn" sequence is $\overline{63}$, which merges with $\overline{1}$ at 150,056, its 323rd term (and the 262nd of $\overline{1}$).

In [5] natural numbers that don't appear in any previously generated sequence are called *starters*; here, we call them *unattainables*, since they cannot be attained by applying f to any other number. (Unattainables are the ends of the branches in Fig. 1.) We also call n a *descendent* of m if n is in $\overline{m}$; that is, if $f^j(m) = n$ for some nonnegative integer j.

Theorem 1. *There are infinitely many unattainables.*

Proof. Suppose there are a finite number k of unattainables. Then let l be an integer so that $9^l > k$ and let $m = 9\ldots90\ldots0$ be the $2l$ digit number consisting of l 9's followed by l 0's. Now consider the $k+1$ integers $m, m+1, \ldots, m+k$. For any $0 \leq i \leq m$ and $j \geq 1$, $f^j(m+i) \geq f(m+i) \geq m+9^l > m+k$, so none

[1] The only previous appearance of these sequences has been in [6], submitted by the author and Neal Sloane.

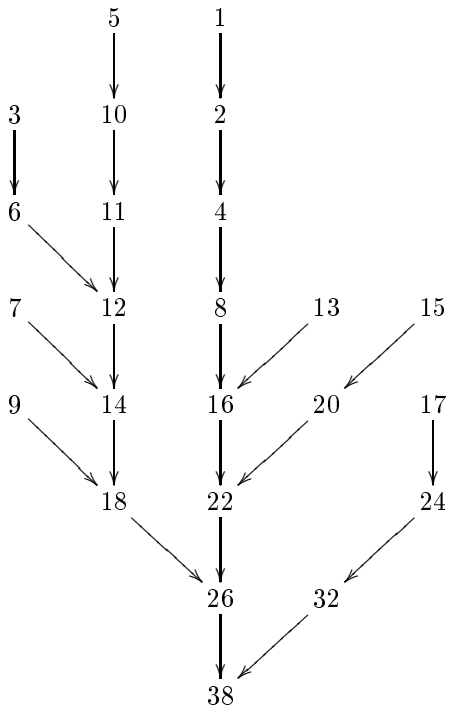


Fig. 1. Sequence tree in base 10

of these $k+1$ integers is a descendent of another. Clearly, every positive integer is either an unattainable or a descendent of an unattainable; thus these integers must correspond to $k+1$ distinct unattainables, a contradiction.

This is our only rigorous result. The rest is a series of questions, heuristic calculations, and numerical evidence.

Question 1. How common are the unattainables?

Let $u(n)$ = the number of unattainables $\leq n$. It is natural to ask if $u(n)/n$ tends to some finite limit. The following table suggests that this is the case.

n	$u(n)/n$
10	.5
100	.44
1000	.429
10^4	.4069
10^5	.39433
10^6	.388459
10^7	.3855173
10^8	.38374875

Question 2. What about bases other than 10?

In base 2, $\Pi(n) = 1$, so $f(n) = n + 1$ for any n. Thus 1 is the only unattainable and the tree has only a single branch. As the base increases, values for $\Pi(n)$ become larger, the sequences move more quickly, there are more unattainables, and the trees are more branched. One could eventually ask for a universal result on unattainables, which would find $\lim_{n\to\infty} u(n)/n$ as a function of the base b.

Question 3. How fast do the sequences grow?

We can approximate the growth of the main base 10 sequence $\{a(n)\} = \{f^{n-1}(1)\}$. On average, an n digit number has $\log_{10} n$ digits. Since a 0 is treated like a 1, the geometric mean of the 10 possible digits is $\sqrt[10]{9!} \approxeq 3.5973$. Thus

$$\Pi(n) \approxeq (\sqrt[10]{9!})^{\log_{10} n} = n^{\log_{10}(\sqrt[10]{9!})}.$$

Letting $k = \log_{10}(\sqrt[10]{9!})$, we can now write $a(n+1) \approxeq a(n) + a(n)^k$, or $a(n+1) - a(n) \approxeq a(n)^k$. This is a difference equation, which we can for the moment treat like a differential equation and write $\frac{da}{dn} = a(n)^k$. Integrating, we have $\int a^{-k} da = \int dn$, and hence

$$a(n) \approxeq [(1-k)(n+c)]^{\frac{1}{1-k}}.$$

Since $a(1) = 1$, solving for c we find $c = \frac{k}{k-1}$. Putting it all together, we have the following asymptotic approximation for $a(n)$:

$$a(n) \approxeq [(1-k)n - k]^{\frac{1}{1-k}} \text{ with } k = \log_{10} \sqrt[10]{9!} \approxeq .55598.$$

This approximation for $a(n)$ works reasonably well, as the following table shows.

n	actual $a(n)$	approx $a(n)$
100	21428	4987
10^4	5.06×10^8	1.64×10^8
10^6	1.02×10^{13}	5.23×10^{12}
10^8	1.02×10^{17}	1.67×10^{17}

It should be noted that under f even numbers tend to stay even - they can only become odd when the last digit is 0 and all others are odd - and odd numbers tend to become even. As a result, the assumption that all digits occur with equal frequency isn't entirely accurate, but as n increases the importance of the terminal digit decreases.

Question 4. How many preimages under f, or "immediate predecessors", can an integer have?

The number 12 is the least with more than one direct predecessor (6 and 11), while 102 has three: 66, 74, and 101; 116 has four: 68, 84, 108, and 116; and 1474 has five: 898, 1366, 1393, 1426, and 1442. The number 11474 will have 6 direct predecessors, 8786 and 5 others found by adding 10000 to the direct predecessors of 1474. The number $1,011,474$ has seven direct predecessors; it should be possible to construct a sequence of numbers with a strictly increasing number of direct predecessors.

Question 5. How many sequences will contain a given number?

Looking back at Fig. 1, note that the sequences beginning at 20 of the integers less than or equal to 26 pass through 26. If we let $j(n)$ be the proportion of numbers less than or equal to n that pass through n, then $j(26) = \frac{20}{26} \approxeq .769$. What happens to the maximum values of $j(n)$ as n increases? In other words, what is $\limsup_{n\to\infty} j(n)$? It would seem natural that $j(n)$ values would decrease, but does the limsup have a nonzero lower bound?

And, lastly,

Question 6. Do digit-product sequences occur in nature?

In 1989 I was an undergraduate at Wabash College trying to devise sequences with unpredictable growth properties. It was a calculation error that first brought the merging property to my attention. It is a pleasure to thank Bonnie Gold, Ganesh Sundaram, Bill Calhoun, John Riley, and Neal Sloane, who have listened, made comments, and shared ideas since then, Tim Smith, for the programming, Bob Montante, for making that connection, and Steve Krantz, who during a summer REU in 1991 helped get these sequences rolling.

References

1. F. Rubin, Problem 1078: Digit Sum Sequences, *Journal of Recreational Mathematics, 14:2*, pp. 141-142, 1981-82.
2. C. G. Feser, Solution to Problem 1078: Digit Sum Sequences, *Journal of Recreational Mathematics, 15:2*, pp. 155-156, 1982-83.
3. H. L. Nelson, Commentary to the Solution to Problem 1078, *Journal of Recreational Mathematics, 20:4*, pp. 304-305, 1988.
4. C. Long, Some Results on Digit Sum Sequences, *Journal of Recreational Mathematics, 23:4*, pp. 244-246, 1991.
5. G. E. Stevens and L. G. Hunsberger, A Result and a Conjecture on Digit Sum Sequences, *Journal of Recreational Mathematics, 27:4*, pp. 285-288, 1995.
6. N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences. http://www.research.att.com/~njas/sequences, sequences AO63108, AO63112, AO63113, AO63114, AO63425.

ACM International Symposium on Software Testing and Analysis (ISSTA), July 2002. Winner of an ***ACM SIGSOFT Distinguished Paper Award.***

Korat: Automated Testing Based on Java Predicates

Chandrasekhar Boyapati, Sarfraz Khurshid, and Darko Marinov
MIT Laboratory for Computer Science
200 Technology Square
Cambridge, MA 02139 USA
{chandra,khurshid,marinov}@lcs.mit.edu

ABSTRACT

This paper presents Korat, a novel framework for automated testing of Java programs. Given a formal specification for a method, Korat uses the method precondition to automatically generate all (nonisomorphic) test cases up to a given small size. Korat then executes the method on each test case, and uses the method postcondition as a test oracle to check the correctness of each output.

To generate test cases for a method, Korat constructs a Java predicate (i.e., a method that returns a boolean) from the method's precondition. The heart of Korat is a technique for automatic test case generation: given a predicate and a bound on the size of its inputs, Korat generates all (nonisomorphic) inputs for which the predicate returns true. Korat exhaustively explores the bounded input space of the predicate but does so efficiently by monitoring the predicate's executions and pruning large portions of the search space.

This paper illustrates the use of Korat for testing several data structures, including some from the Java Collections Framework. The experimental results show that it is feasible to generate test cases from Java predicates, even when the search space for inputs is very large. This paper also compares Korat with a testing framework based on declarative specifications. Contrary to our initial expectation, the experiments show that Korat generates test cases much faster than the declarative framework.

1. INTRODUCTION

Manual software testing, in general, and test data generation, in particular, are labor-intensive processes. Automated testing can significantly reduce the cost of software development and maintenance [4]. This paper presents Korat, a novel framework for automated testing of Java programs. Korat uses specification-based testing [5, 13, 15, 25]. Given a formal specification for a method, Korat uses the method precondition to automatically generate all nonisomorphic test cases up to a given small size. Korat then executes the method on each test case, and uses the method postcondition as a test oracle to check the correctness of each output.

To generate test cases for a method, Korat constructs a Java predicate (i.e., a method that returns a boolean) from the method's precondition. One of the key contributions of Korat is a technique for automatic test case generation: given a predicate, and a bound on the size of its inputs, Korat generates all nonisomorphic inputs for which the predicate returns `true`. Korat uses backtracking to systematically explore the bounded input space of the predicate. Korat generates *candidate* inputs and checks their validity by invoking the predicate on them. Korat monitors accesses that the predicate makes to all the fields of the candidate input. If the predicate returns without reading some fields of the candidate, then the validity of the candidate must be independent of the values of those fields—Korat uses this observation to prune large portions of the search space. Korat also uses an optimization to generate only nonisomorphic test cases. (Section 3.4 gives a precise definition of nonisomorphism.) This optimization reduces the search time without compromising the exhaustive nature of the search.

Korat lets programmers write specifications in any language as long as the specifications can be automatically translated into Java predicates. We have implemented a prototype of Korat that uses the Java Modeling Language (JML) [20] for specifications. Programmers can use JML to write method preconditions and postconditions, as well as class invariants. JML uses Java syntax and semantics for expressions, and contains some extensions such as quantifiers. A large subset of JML can be automatically translated into Java predicates. Programmers can thus use Korat without having to learn a specification language much different than Java. Moreover, since JML specifications can call Java methods, programmers can use the full expressiveness of the Java language to write specifications.

To see an illustration of the use of Korat, consider a method that removes the minimum element from a balanced binary tree. The (implicit) precondition for this method requires the input to satisfy its class invariant: the input must be a binary tree and the tree must be balanced. Korat uses the code that checks the class invariant as the predicate for generating all nonisomorphic balanced binary trees bounded by a given size. Good programming practice [21] suggests that implementations of abstract data types provide predicates (known as the `repOk` or `checkRep` methods) that check class invariants—Korat then generates test cases almost for free. Korat invokes the method on each of the generated trees and checks the postcondition in each case. If a method postcondition is not (explicitly) specified, Korat can still be used to test partial correctness of the method. In the binary tree example, Korat can be used to check the class invariant at the end of the remove method, to see that the tree remains a balanced binary tree after removing the minimum element from it.

ISSTA'02, July 22-24, 2002, Rome, Italy.

```
import java.util.*;
class BinaryTree {
    private Node root; // root node
    private int size;  // number of nodes in the tree
    static class Node {
        private Node left;  // left child
        private Node right; // right child

    }
    public  boolean repOk() {
        // checks that empty tree has size zero
        if (root == null) return size == 0;
        Set visited = new HashSet();
        visited.add(root);
        LinkedList workList = new LinkedList();
        workList.add(root);
        while (!workList.isEmpty()) {
            Node current = (Node)workList.removeFirst();
            if (current.left != null) {
                // checks that tree has no cycle
                if (!visited.add(current.left))
                    return false;
                workList.add(current.left);
            }
            if (current.right != null) {
                // checks that tree has no cycle
                if (!visited.add(current.right))
                    return false;
                workList.add(current.right);
            }
        }
        // checks that size is consistent
        if (visited.size() != size) return false;
        return true;
    }
}
```

Figure 1: `BinaryTree` example

We have used Korat to test several data structures, including some from the Java Collections Framework. The experimental results show that it is feasible to generate test cases from Java predicates, even when the search space for inputs is very large. In particular, our experiments indicate that it is practical to generate inputs to achieve complete statement coverage, even for intricate methods that manipulate complex data structures. This paper also compares Korat with the Alloy Analyzer [16], which can be used to generate test cases [22] from declarative predicates. Contrary to our initial expectation, the experiments show that Korat generates test cases much faster than the Alloy Analyzer.

The rest of this paper is organized as follows. Section 2 illustrates the use of Korat on two examples. Section 3 presents the algorithm that Korat uses to explore the search space. Section 4 describes how Korat checks method correctness. Section 5 presents the experimental results. Section 6 reviews related work, and Section 7 concludes.

2. EXAMPLES

This section presents two examples to illustrate how programmers can use Korat to test their programs. These examples, a binary tree data structure and a heap[1] data structure, illustrate methods that manipulate linked data structures and array-based data structures, respectively.

2.1 Binary tree

This section illustrates the generation and testing of linked data structures using simple binary trees. The Java code in Figure 1 declares a binary tree and defines its `repOk` method, i.e., a Java

[1] The term "heap" refers to the data structure (priority queues) and not to the garbage-collected memory.

```
public static Finitization finBinaryTree(int NUM_Node) {
  Finitization f = new Finitization(BinaryTree.class);
  ObjSet nodes = f.createObjects("Node", NUM_Node);
                                // #Node = NUM_Node
  nodes.add(null);
  f.set("root", nodes);         // root in null + Node
  f.set("size", NUM_Node);      // size = NUM_Node
  f.set("Node.left", nodes);  // Node.left in null + Node
  f.set("Node.right", nodes); // Node.right in null+ Node
  return f;
}
```

Figure 2: Finitization description for the `BinaryTree` example

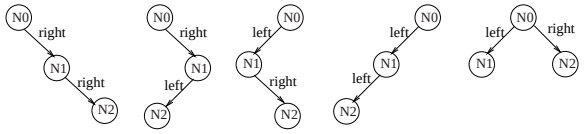


Figure 3: Trees generated for `finBinaryTree(3)`

predicate that checks the representation invariant (or class invariant) of the corresponding data structure [21]. In this case, `repOk` checks if the input is a tree with the correct `size`.

Each object of the class `BinaryTree` represents a tree. The `size` field contains the number of nodes in the tree. Objects of the inner class `Node` represent nodes of the trees. The method `repOk` first checks if the tree is empty. If not, `repOk` traverses all nodes reachable from `root`, keeping track of the visited nodes to detect cycles. (The method `add` from `java.util.Set` returns `false` if the argument already exists in the set.)

To generate trees that have a given number of nodes, the Korat search algorithm uses the *finitization* description shown in Figure 2. The statements in the finitization description specify bounds on the number of objects to be used to construct instances of the data structure, as well as possible values stored in the fields of those objects. Most of the finitization description shown in the figure is automatically generated from the type declarations in the Java code. In Figure 2, the parameter `NUM_Node` specifies the bound on number of nodes in the tree. Each reference field in the tree is either `null` or points to one of the `Node` objects. Note that the identity of these objects is irrelevant—two trees are *isomorphic* if they have the same branching structure, irrespective of the actual nodes in the trees.

Korat automatically generates all nonisomorphic trees with a given number of nodes. For example, for `finBinaryTree(3)`, Korat generates the five trees shown in Figure 3. As another example, for `finBinaryTree(7)`, Korat generates 429 trees in less than one second.

We next illustrate how programmers can use Korat to check correctness of methods. The JML annotations in Figure 4 specify partial correctness for the example `remove` method that removes from a `BinaryTree` a node that is in the tree. The `normal_behavior` annotation specifies that if the precondition (`requires`) is satisfied at the beginning of the method, then the postcondition (`ensures`) is satisfied at the end of the method and the method returns without throwing an exception. (The helper method `has` checks that the tree contains the given node.) Implicitly, the class `invariant` is added to the precondition and the postcondition. Korat uses the JML tool-set to translate annotations into runtime Java assertions.

```
//@ public invariant repOk(); // class invariant
                              //   for BinaryTree
/*@ public normal_behavior    // specification for remove
  @     requires has(n);      //     precondition
  @     ensures !has(n);      //     postcondition
  @*/
public void remove(Node n) {
    // ... method body
}
```

Figure 4: Partial specification for `BinaryTree.remove`

```
public class HeapArray {
    private int size; // number of elements in the heap
    private Comparable[] array; // heap elements
    //@ public invariant repOk();
    public boolean repOk() {
        // checks that array is non-null
        if (array == null) return false;
        // checks that size is within array bounds
        if (size < 0 || size > array.length)
            return false;
        for (int i = 0; i < size; i++) {
            // checks that elements are non-null
            if (array[i] == null) return false;
            // checks that array is heapified
            if (i > 0 &&
                array[i].compareTo(array[(i-1)/2]) > 0)
                return false;
        }
        // checks that non-heap elements are null
        for (int i = size; i < array.length; i++)
            if (array[i] != null) return false;
        return true;
    }
}
```

Figure 5: `HeapArray` example

To test a method, Korat first generates test inputs. For `remove`, each input is a pair of a tree and a node. The precondition defines valid inputs for the method: the tree must be valid and the node must be in the tree. Given a finitization for inputs (which can be written reusing the finitization description for trees presented in Figure 2), Korat generates all nonisomorphic inputs. For `remove`, the number of input pairs is the product of the number of trees and the number of nodes in the trees. After generating the inputs, Korat invokes the method (with runtime assertions for postconditions) on each input and reports a counterexample if the method fails to satisfy the correctness criteria.

2.2 Heap array

This section illustrates the generation and checking of array-based data structures, using the heap data structure [8]. The (binary) *heap* data structure can be viewed as a complete binary tree—the tree is completely filled on all levels except possibly the lowest, which is filled from the left up to some point. Heaps also satisfy the *heap property*—for every node n other than the root, the value of n's parent is greater than or equal to the value of n. The Java code in Figure 5 declares an array-based heap and defines the corresponding `repOk` method that checks if the input is a valid `HeapArray`.

The elements of the heap are stored in `array`. The elements implement the interface `Comparable`, providing the method `compareTo` for comparisons. The method `repOk` first checks for the special case when `array` is `null`. If not, `repOk` checks that the `size` of the heap is within the bounds of the `array`. Then, `repOk` checks that the array elements that belong to the heap are not `null` and that they satisfy the heap property. Finally, `repOk` checks that the array elements that do not belong to the heap are `null`.

```
public static Finitization finHeapArray(int MAX_size,
                                        int MAX_length,
                                        int MAX_elem) {
    Finitization f = new Finitization(HeapArray.class);
    // size in [0..MAX_size]
    f.set("size", new IntSet(0, MAX_size));
    f.set("array",
          // array.length in [0..MAX_length]
          new IntSet(0, MAX_length),
          // array[] in null + Integer([0..MAX_elem])
          new IntegerSet(0, MAX_elem).add(null));
    return f;
}
```

Figure 6: Finitization description for the `HeapArray` example

```
size = 0, array = []
size = 0, array = [null]
size = 1, array = [Integer(0)]
size = 1, array = [Integer(1)]
```

Figure 7: Heaps generated for `finHeapArray(1,1,1)`

To generate heaps, the Korat search algorithm uses the finitization description shown in Figure 6. Again, most of the finitization description shown in the figure is automatically generated from the type declarations in the Java code. In Figure 6, the parameters `MAX_size`, `MAX_length`, and `MAX_elem` bound the size of the heap, the length of the array, and the elements of the array, respectively. The elements of the array can either be `null` or contain `Integer` objects where the integers can range from 0 to `MAX_elem`.

Given values for the finitization parameters, Korat automatically generates all heaps. For example, for `finHeapArray(1,1,1)`, Korat generates the four heaps shown in Figure 7. As another example, in less than one second, for `finHeapArray(5,5,5)`, Korat generates 1919 heaps. Note that Korat requires only the `repOk` method (which can use the full Java language) and finitization to generate all heaps. Writing a dedicated generator for complex data structures [2] is much more involved than writing `repOk`.

We next illustrate how programmers can use Korat to check partial correctness of the `extractMax` method that removes and returns the largest element from a `HeapArray`. The JML annotations in Figure 8 specify partial correctness for the `extractMax` method. The `normal_behavior` specifies that if the input heap is valid and non-empty, then the method returns the largest element in the original heap and the resulting heap after execution of the method is valid. The JML keywords `\result` and `\old` denote, respectively, the object returned by the method and the expressions that should be evaluated in the pre-state. JML annotations can also express exceptional behavior of methods. The example `exceptional_behavior` specifies that if the input heap is empty, the method throws an `IllegalArgumentException`.

To check the method `extractMax`, Korat first uses a finitization to generate all nonisomorphic heaps that satisfy either the `normal_behavior` precondition or the `exceptional_behavior` precondition. Next, Korat invokes the method (with runtime assertions for postconditions) on each input and reports a counterexample if any invocation fails to satisfy the correctness criteria.

3. TEST CASE GENERATION

The heart of Korat is a technique for test case generation: given a Java predicate and a finitization for its input, Korat automatically generates all nonisomorphic inputs for which the predicate

```
/*@ public normal_behavior
  @     requires size > 0;
  @     ensures \result == \old(array[0]);
  @ also public exceptional_behavior
  @     requires size == 0;
  @     signals (IllegalArgumentException e) true;
  @*/
public Comparable extractMax() {
    // ... method body
}
```

Figure 8: Partial specification for `HeapArray.extractMax`

```
void koratSearch(Predicate p, Finitization f) {
    intialize(f);
    while (hasNextCandidate()) {
        Object candidate = nextCandidate();
        try {
            if (p.invoke(candidate))
                output(candidate);
        } catch (Throwable t) {}
        backtrack();
    }
}
```

Figure 9: Pseudo-code of the Korat search algorithm

returns `true`. Figure 9 gives an overview of the Korat search algorithm. The algorithm uses a *finitization* (described in Section 3.1) to bound the *state space* (Section 3.2) of predicate inputs. Korat uses backtracking (Section 3.3) to exhaustively explore the state space. Korat generates *candidate* inputs and checks their validity by invoking the predicate on them. Korat monitors accesses that the predicate makes to all the fields of the candidate input. To monitor the accesses, Korat instruments the predicate and all the methods that the predicate transitively invokes (Section 3.5). If the predicate returns without reading some fields of the candidate, the validity of the candidate must be independent of the values of those fields—Korat uses this observation to prune the search. Korat also uses an optimization that generates only nonisomorphic test cases (Section 3.4).

This section first illustrates how Korat generates valid inputs for predicate methods that take only the implicit `this` argument. Section 3.6 shows how Korat generates valid inputs for Java predicates that take multiple arguments.

3.1 Finitization

To generate a finite state space of a predicate's inputs, the search algorithm needs a *finitization*, i.e., a set of bounds that limits the size of the inputs. Since the inputs can consist of objects from several classes, the finitization specifies the number of objects for each of those classes. A set of objects from one class forms a *class domain*. The finitization also specifies for each field the set of classes whose objects the field can point to. The set of values a field can take forms its *field domain*. Note that a field domain is a union of some class domains.

In the spirit of using the implementation language (which programmers are familiar with) for specification and testing, Korat provides a `Finitization` class that allows finitizations to be written in Java.[2] Korat automatically generates a finitization *skeleton* from the type declarations in the Java code. For the `BinaryTree` example presented in Figure 1, Korat automatically generates the skeleton shown in Figure 10.

[2]The initial version of Korat provided a special-purpose language for more compact descriptions of finitizations, sketched in the comments in the examples in Figures 2 and 6.

```
public static Finitization finBinaryTree(int NUM_Node,
                                         int MIN_size,
                                         int MAX_size) {
  Finitization f = new Finitization(BinaryTree.class);
  ObjSet nodes = f.createObjects("Node", NUM_Node);
  nodes.add(null);
  f.set("root", nodes);
  f.set("size", new IntSet(MIN_size, MAX_size));
  f.set("Node.left", nodes);
  f.set("Node.right", nodes);
  return f;
}
```

Figure 10: Generated finitization description for `BinaryTree`

In Figure 10, the `createObjects` method specifies that the input contains at most `NUM_Node` objects from the `Node`. The `set` method specifies the field domain for each field. In the skeleton, the fields `root`, `left`, and `right` are specified to contain either `null` or a `Node` object. The `size` field is specified to range between `MIN_size` and `MAX_size` using the utility class `IntSet`. The Korat package provides several additional classes for easy construction of class domains and field domains.

Once Korat generates a finitization skeleton, programmers can further specialize or generalize it. For example, the skeleton shown in Figure 10 can be specialized by setting `MIN_size` to 0 and `MAX_size` to `NUM_Node`. We presented another specialized finitization in Figure 2. Note that programmers can use the full expressive power of the Java language for writing finitization descriptions.

3.2 State space

We continue with the `BinaryTree` example to illustrate how Korat constructs the state space for the input to `repOk` using the finitization presented in Figure 2. Consider the case when Korat is invoked for `finBinaryTree(3)`, i.e., `NUM_Node = 3`. Korat first allocates the specified objects: one `BinaryTree` object and three `Node` objects. The three `Node` objects form the `Node` class domain. Korat then assigns a field domain and a unique identifier to each field. The identifier is the index into the *candidate vector*. In this example, the vector has eight elements; there are total of eight fields: the single `BinaryTree` object has two fields, `root` and `size`, and the three `Node` objects have two fields each, `left` and `right`.

For this example, a *candidate* `BinaryTree` input is a sample valuation of those eight fields. The state space of inputs consists of all possible assignments to those fields, where each field gets a value from its corresponding field domain. Since the domain for fields `root`, `left`, and `right` has four elements (`null` and three `Node`s from the `Node` class domain), the state space has $4 * 1 * (4 * 4)^3 = 2^{14}$ potential candidates. For `NUM_Node`$= n$, the state space has $(n + 1)^{2n+1}$ potential candidates. Figure 11 shows an example candidate that is a valid binary tree on three nodes. Not all valuations are valid binary trees. Figure 12 shows an example candidate that is not a tree; `repOk` returns `false` for this input.

3.3 Search

To systematically explore the state space, Korat orders all the elements in every class domain and every field domain (which is a union of class domains). The ordering in each field domain is consistent with the orderings in the class domains, and all the values that belong to the same class domain occur consecutively in the ordering of each field domain.

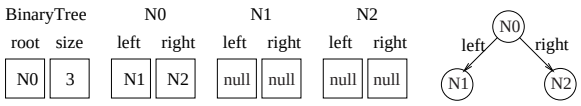


Figure 11: Candidate input that is a valid `BinaryTree`.

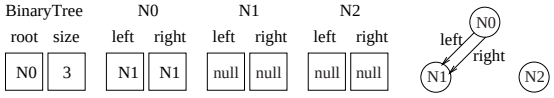


Figure 12: Candidate input that is not a valid `BinaryTree`.

Each candidate input is a vector of *field domain indices* into the corresponding field domains. For our running example with `NUM_Node = 3`, assume that the `Node` class domain is ordered as [N_0,N_1,N_2], and the field domains for `root`, `left`, and `right` are ordered as [`null`,N_0,N_1,N_2]. (`null` by itself forms a class domains.) The domain of the `size` field has a single element, 3. According to this ordering, the candidate inputs in Figures 11 and 12 have candidate vectors $[1, 0, 2, 3, 0, 0, 0, 0]$ and $[1, 0, 2, 2, 0, 0, 0, 0]$, respectively.

The search starts with the candidate vector set to all zeros. For each candidate, Korat sets fields in the objects according to the values in the vector. Korat then invokes `repOk` to check the validity of the current candidate. During the execution of `repOk`, Korat monitors the fields that `repOk` accesses. Specifically, Korat builds a *field-ordering*: a list of the field identifiers ordered by the first time `repOk` accesses the corresponding field. Consider the invocation of `repOk` from Figure 1 on the candidate shown in Figure 12. In this case, `repOk` accesses only the fields [`root`,N_0`.left`,N_0`.right`] (in that order) before returning `false`. Hence, the field-ordering that Korat builds is `[0,2,3]`.

After `repOk` returns, Korat generates the next candidate vector backtracking on the fields accessed by `repOk`. Korat first increments the field domain index for the field that is last in the field-ordering. If the domain index exceeds the domain size, Korat resets that index to zero, and increments the domain index of the previous field in the field-ordering, and so on. (The next section presents how Korat generates only nonisomorphic candidates by resetting a domain index for a field to zero even when the index does not exceed the size of the field domain.)

Continuing with our example, the next candidate takes the next value for N_0`.right`, which is N_2 by the above order, whereas the other fields do not change. This prunes from the search all 4^4 candidate vectors of the form `[1,_,2,2,_,_,_,_]` that have the (partial) valuation: `root=`N_0, N_0`.left=`N_1, N_0`.right=`N_1. This pruning does not rule out any valid data structure because `repOk` did not read the other fields, and it could have returned `false` irrespective of the values of those fields.

Continuing further with our example, the next candidate is the valid tree shown in Figure 11. Before executing `repOk` on this candidate, Korat also initializes the field-ordering to `[0,2,3]`. Note that, if `repOk` accesses fields in a deterministic order, this is consistent with the first three fields that `repOk` is going to access, because the values of the first two fields in the field-ordering were not changed when constructing this candidate from the previous candidate. When `repOk` executes on this candidate, `repOk` returns `true` and the field-ordering that Korat builds is `[0,2,3,4,5,6,7,1]`. If `repOk` returns `true`, Korat outputs all (nonisomorphic) candidates that have the same values for the accessed fields as the current candidate. (Note that `repOk` may not access all reachable fields before returning `true`.) The search then backtracks to the next candidate.

Recall that Korat orders the values in the class and field domains. Additionally, each execution of `repOk` on a candidate imposes an order on the fields in the field-ordering. Together, these orders induce a lexicographic order on the candidates. The search algorithm described here generates inputs in the lexicographical order. Moreover, for non-deterministic `repOk` methods, our algorithm provides the following guarantee: all candidates for which `repOk` always returns `true` are generated; candidates for which `repOk` always returns `false` are never generated; and candidates for which `repOk` sometimes returns `true` and sometimes `false` may or may not be generated.

In practice, our search algorithm prunes large portions of the search space, and thus enables Korat to explore very large state spaces. The efficiency of the pruning depends on the `repOk` method. An ill-written `repOk`, for example, might always read the entire input before returning, thereby forcing Korat to explore almost every candidate. However, our experience indicates that naturally written `repOk` methods, which return `false` as soon as the first invariant violation is detected, induce very effective pruning.

3.4 Nonisomorphism

To further optimize the search, Korat avoids generating multiple candidates that are isomorphic to one another. Our optimization is based on the following definition of isomorphism.

Definition: Let $O_1, \ldots, O_n$ be some sets of objects from n classes. Let $O = O_1 \cup \ldots \cup O_n$, and suppose that candidates consist only of objects from O. (Pointer fields of objects in O can either be `null` or point to other objects in O.) Let P be the set consisting of `null` and all values of primitive types (such as `int`) that the fields of objects in O can contain. Further, let $r \in O$ be a special root object, and let O_C be the set of all objects reachable from r in C. Two candidates, C and C', are *isomorphic* iff there is a permutation π on O, mapping objects from O_i to objects from O_i for all $1 \leq i \leq n$, such that:

$\forall o, o' \in O_C.\ \forall f \in fields(o).\ \forall p \in P.$
$o.f == o'$ in C iff $\pi(o).f == \pi(o')$ in C' and
$o.f == p$ in C iff $\pi(o).f == p$ in C'.

The operator `==` is Java's comparison by object identity. Note that isomorphism is defined with respect to a root object. Two candidates are defined to be isomorphic if the parts of their object graphs reachable from the root object are isomorphic. In case of `repOk`, the root object is the `this` object that is passed as an implicit argument to `repOk`.

Isomorphism between candidates partitions the state space into *isomorphism partitions*. Recall the lexicographic ordering induced by the ordering on the values in the field domains and the field-orderings built by `repOk` executions. For each isomorphism partition, Korat generates only the lexicographically smallest candidate in that partition.

Conceptually, Korat avoids generating multiple candidates from the same isomorphism partition by incrementing field domain indices

```
class SomeClass {
    boolean somePredicate(X x, Y y) {...}
    ...
}
```

Figure 13: Predicate method with multiple arguments

by more than one: while backtracking on a field f in the field-ordering, Korat checks for how much to increment the field domain index of f as follows. Suppose that f contains a pointer to an object o_f that belongs to a class domain c_f. Recall that all objects in a class domain are ordered. Let i_f be the index of o_f in c_f. For instance, in the example ordering used above for `finBinaryTree(3)`, field domain index `2` for `right` corresponds to the class domain `Node` and class domain index `1`.

Further, Korat finds all fields f' such that f' occurs before f in the field-ordering and f' contains a pointer to an object o'_f of the same class domain c_f. Let i'_f be the index of o'_f in c_f, and let m_f be the maximum of all such indices i'_f. (If there is no such field f' before f in the field-ordering, $m_f =$`-1`.) In the example candidate for Figure 12, backtracking on $f =$`N0.right` gives $m_f =$`1`.

Then, during backtracking on f, Korat checks if i_f is greater than m_f. If $i_f \leq m_f$, Korat increments the field domain index of f by one. If $i_f > m_f$, Korat increments the field domain index of f so that it contains a pointer to an object of the class domain after c_f. If no such domain exists, i.e., c_f is the last domain for the field f, Korat resets the field domain index of f to zero and continues backtracking on the previous field in the field-ordering. The actual Korat implementation uses caching to speed up the computation of m_f.

For example, Korat for `finBinaryTree(3)` generates only the five trees shown in Figure 3. Each tree is a representative from an isomorphism partition that has six distinct trees, one for each of $3!$ permutations of nodes.

3.5 Instrumentation

To monitor `repOk`'s executions, Korat instruments all classes whose objects appear in finitizations by doing a source to source translation. For each of the classes, Korat adds a special constructor. For each field of those classes, Korat adds an identifier field and special `get` and `set` methods. In the code for `repOk` and all the methods that `repOk` transitively invokes, Korat replaces each field access with an invocation of the corresponding `get` or `set` method. Arrays are similarly instrumented, essentially treating each array element as a field.

To monitor the field accesses and build a field-ordering, Korat uses an approach similar to the *observer* pattern [11]. Korat uses the special constructors to initialize all objects in a finitization with an observer. The search algorithm initializes each of the identifier fields to a unique index into the candidate vector. Special `get` and `set` methods first notify the observer of the field access using the field's identifier and then perform the field access (return the field's value or assign to the field).

3.6 Predicates with multiple arguments

The discussion so far described how Korat generates inputs that satisfy a `repOk` method. This section describes how Korat generalizes this technique to generate inputs that satisfy any Java predicate, including predicates that take multiple arguments. Figure 13 shows

```
class SomeClass_somePredicate {
    SomeClass This;
    X x;
    Y y;
    boolean repOk() {
        return This.somePredicate(x, y);
    }
}
```

Figure 14: Equivalent `repOk` method

a Java predicate that takes two arguments (besides `this`). In order to generate inputs for this predicate, Korat generates an equivalent `repOk` method shown in Figure 14. Korat then generates inputs to the `repOk` method using the technique described earlier.

4. TESTING METHODS

The previous section focused on automatic test case generation from a Java predicate and a finitization description. This section presents how Korat builds on this technique to check correctness of methods. Korat uses specification-based testing: to test a method, Korat first generates test inputs from the method's precondition, then invokes the method on each of those inputs, and finally checks the correctness of the output using the method's postcondition.

The current Korat implementation uses the Java Modeling Language (JML) [20] for specifications. Programmers can use JML annotations to express method preconditions and postconditions, as well as class invariants; these annotations use JML keywords `requires`, `ensures`, and `invariant`, respectively. Each annotation contains a boolean expression; JML uses Java syntax and semantics for expressions, and contains some extensions such as quantifiers. Korat uses a large subset of JML that can be automatically translated into Java predicates.

JML specifications can express several *normal* and *exceptional behaviors* for a method. Each behavior has a precondition and a postcondition: if the method is invoked with the precondition being satisfied, the behavior requires that the method terminate with the postcondition being satisfied. Additionally, normal behaviors require that the method return without an exception, whereas exceptional behaviors require that the method return with an exception. Korat generates inputs for all method behaviors using the *complete* method precondition that is a conjunction of: 1) the class invariant for all objects reachable from the input parameters and 2) a disjunction of the preconditions for all behaviors. In the text that follows, we refer to complete precondition simply as precondition.

4.1 Generating test cases

Valid test cases for a method must satisfy its precondition. To generate valid test cases, Korat uses a class that represents method's inputs. This class has one field for each parameter of the method (including the implicit `this` parameter) and a `repOk` predicate that uses the precondition to check the validity of method's inputs. Given a finitization, Korat then generates all inputs for which this `repOk` returns `true`; each of these inputs is a valid input to the original method.

We illustrate generation of test cases using the `remove` method for `BinaryTree` from Section 2. For this method, each input consists of a pair of `BinaryTree this` and a `Node n`, and the precondition is `this.has(n)`. Figure 15 shows the class that Korat uses for the method's inputs. For this class, Korat creates the finitization skeleton that reuses the finitization for `BinaryTree`, as shown in

```
class BinaryTree_remove {
    BinaryTree This;   // the implicit "this" parameter
    BinaryTree.Node n; // the Node parameter
    //@ invariant repOk();
    public boolean repOk() {
        return This.has(n);
    }
}
```

Figure 15: Class representing `BinaryTree.remove`

```
public static Finitization
        finBinaryTree_remove(int NUM_Node) {
    Finitization f =
        new Finitization(BinaryTree_remove.class);
    Finitization g = BinaryTree.finBinaryTree(NUM_Node);
    f.includeFinitization(g);
    f.set("This", g.getObjects(BinaryTree.class));
    f.set("n", /***/);
    return f;
}
```

Figure 16: Finitization skeleton for `BinaryTree_remove`

Figure 16. The comment `/***/` indicates that Korat cannot automatically determine an appropriate field domain for `n`.

To create finitization for `BinaryTree_remove`, the programmer modifies the skeleton, e.g., by replacing `/***/` with `g.get("root")` or `g.getObjects(BinaryTree.Node.class)` to set the domain for the parameter `n` to the domain for the field `root` or to the set of nodes from the finitization `g`, respectively. Given a value for `NUM_Node`, Korat then generates all valid test cases, each of which is a pair of a tree (with the given number of nodes) and a node from that tree.

4.1.1 Dependent and independent parameters

For the `remove` method, the precondition makes the parameters `This` and `n` explicitly dependent. When the parameters are independent, programmers can instruct Korat to generate all test cases by separately generating all possibilities for each parameter and creating all valid test cases as the Cartesian product of these possibilities.

We next compare Korat with another approach for generating all valid (nonisomorphic) test cases, which uses the Cartesian product even for dependent parameters. Consider a method `m`, with n parameters and precondition $\mathtt{m}_{pre}$. Suppose that a set of possibilities S_i, $1 \leq i \leq n$, is given for each of the parameters. All valid test cases from $S_1 \times \ldots \times S_n$ can be then generated by creating *all* n-tuples from the product, followed by filtering each of them through $\mathtt{m}_{pre}$. (This approach is used in the JML+JUnit testing framework [6] that combines JML [20] and JUnit [3].) Note that this approach requires manually constructing possibilities for all parameters, some of which can be complex data structures.

Korat, on the other hand, constructs data structures from a simple description of the fields in the structures. Further, in terms of Korat's search of `repOk`'s state space, the presented approach would correspond to the search that tries every candidate input. Korat improves on this approach by: 1) pruning the search based on the accessed fields and 2) generating only one representative from each isomorphism partition.

4.2 Checking correctness

To check a method, Korat first generates all valid inputs for the method using the process explained above. Korat then invokes the method on each of the inputs and checks each output with a *test oracle*. To check partial correctness of a method, a simple test oracle could just invoke `repOk` in the *post-state* (i.e., the state immediately after the method's invocation) to check if the method preserves its class invariant. If the result is `false`, the method under test is incorrect, and the input provides a concrete counterexample. Programmers could also manually develop more elaborate test oracles. Programmers can also check for properties that relate the post-state with the *pre-state* (i.e., the state just before the method's invocation).

	Testing framework		
testing activity	JUnit	JML+JUnit	Korat
generating test cases			√
generating test oracle		√	√
running tests	√	√	√

Table 1: Comparison of several testing frameworks for Java. Automated testing activities are indicated with "√".

The current Korat implementation uses the JML tool-set to automatically generate test oracles from method postconditions, as in the JML+JUnit framework [6]. The JML tool-set translates JML postconditions into runtime Java assertions. If an execution of a method violates such an assertion, an exception is thrown to indicate a violated postcondition. Test oracle catches these exceptions and reports correctness violations. These exceptions are different from the exceptions that the method specification allows, and Korat leverages on JML to check both normal and exceptional behavior of methods. More details of the JML tool-set and translation can be found in [20].

Korat also uses JML+JUnit to combine JML test oracles with JUnit [3], a popular framework for unit testing of Java modules. JUnit automates test execution and error reporting, but requires programmers to provide test inputs and test oracles. JML+JUnit, thus, automates both test execution and correctness checking. However, JML+JUnit requires programmers to provide sets of possibilities for all method parameters: it generates all valid inputs by generating the Cartesian product of possibilities and filtering the tuples using preconditions. Korat additionally automates generation of test cases, thus automating the entire testing process. Table 1 summarizes the comparison of these testing frameworks.

5. EXPERIMENTAL RESULTS

This section presents the performance results of the Korat prototype. We used Java to implement the search for valid nonisomorphic `repOk` inputs. For automatic instrumentation of `repOk` (and transitively invoked methods), we modified the sources of the Sun's `javac` compiler. We also modified `javac` to automatically generate finitization skeletons. For checking method correctness, we slightly modified the JML tool-set, building on the existing JML+JUnit framework [6].

We first present Korat's performance for test case generation, then compare Korat with the test generation that uses Alloy Analyzer [16], and finally present Korat's performance for checking method correctness. We performed all experiments on a Linux machine with a Pentium III 800 MHz processor using Sun's Java 2 SDK1.3.1 JVM.

benchmark	package	finitization parameters
BinaryTree	korat.examples	NUM_Node
HeapArray	korat.examples	MAX_size, MAX_length, MAX_elem
LinkedList	java.util	MIN_size, MAX_size, NUM_Entry, NUM_Object
TreeMap	java.util	MIN_size, NUM_Entry, MAX_key, MAX_value
HashSet	java.util	MAX_capacity, MAX_count, MAX_hash, loadFactor
AVTree	ins.namespace	NUM_AVPair, MAX_child, NUM_String

Table 2: Benchmarks and finitization parameters. Each benchmark is named after the class for which data structures are generated; the structures also contain objects from other classes.

5.1 Benchmarks

Table 2 lists the benchmarks for which we show Korat's performance. BinaryTree and HeapArray are presented in Section 2. (Additionally, HeapArrays are similar to array-based stacks and queues, as well as java.util.Vectors.) LinkedList is the implementation of linked lists in the Java Collections Framework, a part of the standard Java libraries. This implementation uses doubly-linked, circular lists that have a size field and a header node as a sentinel node. (Linked lists also provide methods that allow them to be used as stacks and queues.) TreeMap implements the Map interface using red-black trees [8]. This implementation uses binary trees with parent fields. Each node (implemented with inner class Entry) also has a key and a value. (Setting all value fields to null corresponds to the set implementation in java.util.TreeSet.) HashSet implements the Set interface, backed by a hash table [8]. This implementation builds collision lists for buckets with the same hash code. The loadFactor parameter determines when to increase the size of the hash table and rehash the elements.

AVTree implements the *intentional name* trees that describe properties of services in the Intentional Naming System (INS) [1], an architecture for service location in dynamic networks. Each node in an intentional name has an attribute, a value, and a set of child nodes. INS uses attributes and values to classify services based on their properties. The names of these properties are implemented with arbitrary Strings except that "*" is a wildcard that matches all other values. The finitization bounds the number of AVPair objects that implement nodes, the number of children for each node, and the total number of Strings (including the wildcard).

5.2 Korat's test case generation

Table 3 presents the results for generating valid structures with our Korat implementation. For each benchmark, all finitization parameters are set to the same (size) value (except the loadFactor parameter for HashSet, which is set to default 0.75). For a range of size values, we tabulate the time that Korat takes to generate all valid structures, the number of structures generated, the number of candidate structures checked by repOk, and the size of the state space.

Korat can generate all structures even for very large state spaces because the search pruning allows Korat to explore only a tiny fraction of the state space. The ratios of the number of candidate structures considered and the size of the state spaces show that the key to effective pruning is backtracking based on fields accessed during repOk's executions. Without backtracking, and even with isomorphism optimization, Korat would generate infeasibly many candidates. Isomorphism optimization further reduces the number of candidates, but it mainly reduces the number of valid structures.

benchmark	size	time (sec)	structures generated	candidates considered	state space
BinaryTree	8	1.53	1430	54418	2^{93}
	9	3.97	4862	210444	2^{63}
	10	14.41	16796	815100	2^{72}
	11	56.21	58786	3162018	2^{82}
	12	233.59	208012	12284830	2^{92}
HeapArray	6	1.21	13139	64533	2^{20}
	7	5.21	117562	519968	2^{25}
	8	42.61	1005075	5231385	2^{29}
LinkedList	8	1.32	4140	5455	2^{91}
	9	3.58	21147	26635	2^{105}
	10	16.73	115975	142646	2^{120}
	11	101.75	678570	821255	2^{135}
	12	690.00	4213597	5034894	2^{150}
TreeMap	7	8.81	35	256763	2^{92}
	8	90.93	64	2479398	2^{111}
	9	2148.50	122	50209400	2^{130}
HashSet	7	3.71	2386	193200	2^{119}
	8	16.68	9355	908568	2^{142}
	9	56.71	26687	3004597	2^{166}
	10	208.86	79451	10029045	2^{190}
	11	926.71	277387	39075006	2^{215}
AVTree	5	62.05	598358	1330628	2^{50}

Table 3: Korat's performance on several benchmarks. All finitization parameters are set to the size value. Time is the elapsed real time in seconds for the entire generation. State size is rounded to the nearest smaller exponent of two.

For BinaryTree, LinkedList, TreeMap, and HashSet (with the loadFactor parameter of 1), the numbers of nonisomorphic structures appear in the Sloane's On-Line Encyclopedia of Integer Sequences [30]. For all these benchmarks, Korat generates exactly the actual number of structures.

5.2.1 Comparison with Alloy Analyzer

We next compare Korat's test case generation with that of the Alloy Analyzer (AA) [16], an automatic tool for analyzing Alloy *models*. Alloy [17] is a first-order, declarative language based on relations. Alloy is suitable for modeling structural properties of software. Alloy models of several data structures can be found in [22]. These models specify class invariants in Alloy, which correspond to repOk methods in Korat, and also declare field types, which corresponds to setting field domains in Korat finitizations.

Given a model of a data structure and a *scope*—a bound on the number of atoms in the universe of discourse—AA can generate all (mostly nonisomorphic) *instances* of the model. An instance valuates the relations in the model such that all constraints of the model are satisfied. Setting the scope in Alloy corresponds to setting the finitization parameters in Korat. AA translates the input Alloy model into a boolean formula and uses an off-the-shelf SAT solver to find a satisfying assignment to the formula. Each such assignment is translated back to an instance of the input model. AA adds symmetry-breaking predicates [29] to the boolean formula so that different satisfying assignments to the formula represent (mostly) nonisomorphic instances of the input model.

benchmark	size	Korat			Alloy Analyzer		
		struc. gen.	total time	first struc.	inst. gen.	total time	first inst.
BinaryTree	3	5	0.56	0.62	6	2.63	2.63
	4	14	0.58	0.62	28	3.91	2.78
	5	42	0.69	0.67	127	24.42	4.21
	6	132	0.79	0.66	643	269.99	6.78
	7	429	0.97	0.62	3469	3322.13	12.86
HeapArray	3	66	0.53	0.58	78	11.99	6.20
	4	320	0.57	0.59	889	171.03	16.13
	5	1919	0.73	0.63	1919	473.51	39.58
LinkedList	3	5	0.58	0.60	10	2.61	2.39
	4	15	0.55	0.65	46	3.47	2.77
	5	52	0.57	0.65	324	14.09	3.51
	6	203	0.73	0.61	2777	148.73	5.74
	7	877	0.87	0.61	27719	2176.44	10.51
TreeMap	4	8	0.75	0.69	16	12.10	6.35
	5	14	0.87	0.88	42	98.09	18.08
	6	20	1.49	0.98	152	1351.50	50.87
AVTree	2	2	0.55	0.65	2	2.35	2.43
	3	84	0.65	0.61	132	4.25	2.76
	4	5923	1.41	0.61	20701	504.12	3.06

Table 4: Performance comparison. For each benchmark, performances of Korat and AA are compared for a range of finitization values. For values larger than presented, AA does not complete its generation within 1 hour. Korat's performance for larger values is given in Table 3.

Table 4 summarizes the performance comparison. Since AA cannot handle arbitrary arithmetic, we do not generate HashSets with AA. For all other benchmarks, we compare the total number of structures/instances and the time to generate them for a range of parameter values. We also compare the time to generate the first structure/instance.

Time presented is the total elapsed real time (in seconds) that each experiment took from the beginning to the end, including start-up.[3] Start-up time for Korat is approximately 0.5 sec. (That is why in some cases it seems that generating all structures is faster than generating the first structure or that generating all structures for a larger input is faster than generating all structures for a smaller input.) Start-up time for AA is somewhat higher, approximately 2 sec, as AA needs to translate the model and to start a SAT solver. AA uses precompiled binaries for SAT solvers.

In all cases, Korat outperforms AA; Korat is not only faster for smaller inputs, but it also completes generation for larger inputs than AA. There are two reasons that could account for this difference. Since AA translates Alloy models into boolean formulas, it could be that the current (implementation of the) translation generates unnecessarily large boolean formulas. Another reason is that often AA generates a much greater number of instances than Korat, which takes a greater amount of time by itself. One way to reduce the number of instances generated by AA is to add more symmetry-breaking predicates.

Our main argument for developing Korat was simple: for Java programmers not familiar with Alloy, it is easier to write a repOk method than an Alloy model. (From our experience, for researchers familiar with Alloy, it is sometimes easier to write an Alloy model than a repOk method.) Before conducting the above experiments, we expected that Korat would generate structures slower than AA.

[3] We include start-up time, because AA does not provide generation time only for generating all instances. We eliminate the effect of cold start by executing each test twice and taking the smaller time.

benchmark	method	max. size	test cases generated	gen. time	test time
BinaryTree	remove	3	15	0.64	0.73
HeapArray	extractMax	6	13139	0.87	1.39
LinkedList	reverse	2	8	0.67	0.76
TreeMap	put	8	19912	136.19	2.70
HashSet	add	7	13106	3.90	1.72
AVTree	lookup	4	27734	4.33	14.63

Table 5: Korat's performance on several methods. All upper-limiting finitization parameters for method inputs are set to the given maximum size. These sizes give complete statement coverage. Times are the elapsed real times in seconds for the entire generation of all valid test cases and testing of methods for all those inputs. These times include writing and reading of files with test cases.

Our intuition was that Korat depends on the executions of repOk to "learn" the invariants of the structures, whereas AA uses a SAT solver that can "inspect" the entire formula (representing invariants) to decide how to search for an assignment. The experimental results show that our assumption was incorrect—Korat generates structures much faster than AA. We are now exploring a translation of Alloy models into Java (or even C) and the use of Korat (or a similar search) to generate instances.

5.3 Checking correctness

Table 5 presents the results for checking methods with Korat. For each benchmark, a representative method is chosen; the results are similar for other methods. Methods remove and extractMax are presented in Section 2. Method reverse, from java.util.Collections, uses list iterators to reverse the order of list elements; this method is static. Method put, from java.util.TreeMap, inserts a key-value pair into the map; this method has three parameters (this, key, and value) and invokes several helper methods that rebalance the tree after insertion. Method add inserts an element into the set. Method lookup, from INS, searches a database of intentional names for a given query intentional name. The correctness specifications for all methods specify simple containment properties (beside preservation of class invariants).

For each method, the MIN finitization parameters are set to zero and the MAX and NUM parameters to the same size value. Thus, the methods are checked for all valid inputs up to the maximum size, not only for the maximum size. The results show that it is practical to use Korat to exhaustively check correctness of intricate methods that manipulate complex data structures.

AA can also be used to check correctness of Java methods by writing method specifications as Alloy models and defining appropriate translations between Alloy instances and Java objects, as demonstrated in the TestEra framework [22]. However, the large number of instances generated by AA makes TestEra less practical to use than Korat. For example, maximum sizes six and eight for extractMax and put methods, respectively, are the smallest that give complete statement coverage. As shown in Table 4, for these sizes, AA cannot in a reasonable time even generate data structures that are parts of the inputs for these methods.

6. RELATED WORK

6.1 Specification-based testing

There is a large body of research on specification-based testing. An early paper by Goodenough and Gerhart [13] emphasizes its impor-

tance. Many projects automate test case generation from specifications, such as Z specifications [15, 31], UML statecharts [25, 26], or ADL specifications [5, 28]. These specifications typically do not consider linked data structures, and the tools do not generate Java test cases.

The TestEra framework [22] generates Java test cases from Alloy [17] specifications of linked data structures. TestEra uses the Alloy Analyzer (AA) [16] to automatically generate method inputs and check correctness of outputs, but it requires programmers to learn a specification language much different than Java. Korat generates inputs directly from Java predicates and uses the Java Modeling Language (JML) [20] for specifications. The experimental results also show that Korat generates test cases faster and for larger scopes than AA.

Cheon and Leavens [6] describe automatic translation of JML specifications into test oracles for JUnit [3]. This framework automates execution and checking of methods. However, the burden of test case generation is still on programmers: they have to provide sets of possibilities for all method parameters. Korat builds on this framework by automating test case generation.

6.2 Static analysis

Several projects aim at developing static analyses for verifying program properties. The Extended Static Checker (ESC) [10] uses a theorem prover to verify partial correctness of classes annotated with JML specifications. ESC has been used to verify absence of such errors as null pointer dereferences, array bounds violations, and division by zero. However, tools like ESC cannot verify properties of complex linked data structures.

There are some recent research projects that attempt to address this issue. The Three-Valued-Logic Analyzer (TVLA) [27] is the first static analysis system to verify that the list structure is preserved in programs that perform list reversals via destructive updating of the input list. TVLA has been used to analyze programs that manipulate doubly linked lists and circular lists, as well as some sorting programs. The pointer assertion logic engine (PALE) [24] can verify a large class of data structures that can be represented by a spanning tree backbone, with possibly additional pointers that do not add extra information. These data structures include doubly linked lists, trees with parent pointers, and threaded trees. While TVLA and PALE are primarily intraprocedural, Role Analysis [19] supports compositional interprocedural analysis and verifies similar properties.

While static analysis of program properties is a promising approach for ensuring program correctness in the long run, the current static analysis techniques can only verify limited program properties. For example, none of the above techniques can verify correctness of implementations of balanced trees, such as red-black trees. Testing, on the other hand, is very general and can verify any decidable program property, but for inputs bounded by a given size.

Jackson and Vaziri propose an approach [18] for analyzing methods that manipulate linked data structures. Their approach is to first build an Alloy model of bounded initial segments of computation sequences and then check the model exhaustively with AA. This approach provides static analysis, but it is unsound with respect to both the size of input and the length of computation. Korat not only checks the entire computation, but also handles larger inputs and more complex data structures than those in [18]. Further, Korat does not require Alloy, but JML specifications, and more importantly, unlike [18], Korat does not require specifications for all (helper) methods.

6.3 Software model checking

There has been a lot of recent interest in applying model checking to software. JavaPathFinder [32] and VeriSoft [12] operate directly on a Java, respectively C, program and systematically explore its state to check correctness. Other projects, such as Bandera [7] and JCAT [9], translate Java programs into the input language of existing model checkers like SPIN [14] and SMV [23]. They handle a significant portion of Java, including dynamic allocation, object references, exceptions, inheritance, and threads. They also provide automated support for reducing program's state space through program slicing and data abstraction.

However, most of the work on applying model checking to software has focused on checking event sequences and not linked data structures. Where data structures have been considered, the purpose has been to reduce the state space to be explored and not to check the data structures themselves. Korat, on the other hand, checks correctness of methods that manipulate linked data structures.

7. CONCLUSIONS

This paper presented Korat, a novel framework for automated testing of Java programs. Given a formal specification for a method, Korat uses the method precondition to automatically generate all nonisomorphic test cases up to a given small size. Korat then executes the method on each test case, and uses the method postcondition as a test oracle to check the correctness of each output.

To generate test cases for a method, Korat constructs a Java predicate (i.e., a method that returns a boolean) from the method's precondition. The heart of Korat is a technique for automatic test case generation: given a predicate and a finitization for its inputs, Korat generates all nonisomorphic inputs for which the predicate returns `true`. Korat exhaustively explores the input space of the predicate, but does so efficiently by: 1) monitoring the predicate's executions to prune large portions of the search space and 2) generating only nonisomorphic inputs.

The Korat prototype uses the Java Modeling Language (JML) for specifications, i.e., class invariants and method preconditions and postconditions. Good programming practice suggests that implementations of abstract data types should already provide methods for checking class invariants—Korat then generates test cases almost for free.

This paper illustrated the use of Korat for testing several data structures, including some from the Java Collections Framework. The experimental results show that it is feasible to generate test cases from Java predicates, even when the search space for inputs is very large. This paper also compared Korat with the Alloy Analyzer, which can be used to generate test cases from declarative predicates. Contrary to our initial expectation, the experiments show that Korat generates test cases much faster than the Alloy Analyzer.

Acknowledgements

We would like to thank Michael Ernst, Daniel Jackson, Alexandru Sălcianu, and the anonymous referees for their comments on this paper. We are also grateful to Viktor Kuncak for helpful discussions on Korat and Alexandr Andoni for helping us with experiments. This work was funded in part by NSF grant CCR00-86154.

8. REFERENCES

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. The design and implementation of an intentional naming system. In *Proc. 17th ACM Symposium on Operating Systems (SOSP)*, Kiawah Island, Dec. 1999.

[2] T. Ball, D. Hoffman, F. Ruskey, R. Webber, and L. J. White. State generation and automated class testing. *Software Testing, Verification & Reliability*, 10(3):149–170, 2000.

[3] K. Bech and E. Gamma. Test infected: Programmers love writing tests. *Java Report*, 3(7), July 1998.

[4] B. Beizer. *Software Testing Techniques*. International Thomson Computer Press, 1990.

[5] J. Chang and D. J. Richardson. Structural specification-based testing: Automated support and experimental evaluation. In *Proc. 7th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, pages 285–302, Sept. 1999.

[6] Y. Cheon and G. T. Leavens. A simple and practical approach to unit testing: The JML and JUnit way. Technical Report 01-12, Department of Computer Science, Iowa State University, Nov. 2001.

[7] J. Corbett, M. Dwyer, J. Hatcliff, C. Pasareanu, Robby, S. Laubach, and H. Zheng. Bandera: Extracting finite-state models from Java source code. In *Proc. 22nd International Conference on Software Engineering (ICSE)*, June 2000.

[8] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 1990.

[9] C. Demartini, R. Iosif, and R. Sisto. A deadlock detection tool for concurrent Java programs. *Software - Practice and Experience*, July 1999.

[10] D. L. Detlefs, K. R. M. Leino, G. Nelson, and J. B. Saxe. Extended static checking. Research Report 159, Compaq Systems Research Center, 1998.

[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements od Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Addison-Wesley Publishing Company, New York, NY, 1995.

[12] P. Godefroid. Model checking for programming languages using VeriSoft. In *Proc. 24th Annual ACM Symposium on the Principles of Programming Languages (POPL)*, pages 174–186, Paris, France, Jan. 1997.

[13] J. Goodenough and S. Gerhart. Toward a theory of test data selection. *IEEE Transactions on Software Engineering*, June 1975.

[14] G. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23(5), May 1997.

[15] H.-M. Horcher. Improving software tests using Z specifications. In *Proc. 9th International Conference of Z Users, The Z Formal Specification Notation*, 1995.

[16] D. Jackson, I. Schechter, and I. Shlyakhter. ALCOA: The Alloy constraint analyzer. In *Proc. 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, June 2000.

[17] D. Jackson, I. Shlyakhter, and M. Sridharan. A micromodularity mechanism. In *Proc. 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, Vienna, Austria, Sept. 2001.

[18] D. Jackson and M. Vaziri. Finding bugs with a constraint solver. In *Proc. International Symposium on Software Testing and Analysis (ISSTA)*, Portland, OR, Aug. 2000.

[19] V. Kuncak, P. Lam, and M. Rinard. Role analysis. In *Proc. 29th Annual ACM Symposium on the Principles of Programming Languages (POPL)*, Portland, OR, Jan. 2002.

[20] G. T. Leavens, A. L. Baker, and C. Ruby. Preliminary design of JML: A behavioral interface specification language for Java. Technical Report TR 98-06i, Department of Computer Science, Iowa State University, June 1998. (last revision: Aug 2001).

[21] B. Liskov. *Program Development in Java: Abstraction, Specification, and Object-Oriented Design*. Addison-Wesley, 2000.

[22] D. Marinov and S. Khurshid. TestEra: A novel framework for automated testing of Java programs. In *Proc. 16th IEEE International Conference on Automated Software Engineering (ASE)*, San Diego, CA, Nov. 2001.

[23] K. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.

[24] A. Moeller and M. I. Schwartzbach. The pointer assertion logic engine. In *Proc. SIGPLAN Conference on Programming Languages Design and Implementation*, Snowbird, UT, June 2001.

[25] J. Offutt and A. Abdurazik. Generating tests from UML specifications. In *Proc. Second International Conference on the Unified Modeling Language*, Oct. 1999.

[26] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley Object Technology Series, 1998.

[27] M. Sagiv, T. Reps, and R. Wilhelm. Solving shape-analysis problems in languages with destructive updating. *ACM Trans. Prog. Lang. Syst.*, January 1998.

[28] S. Sankar and R. Hayes. Specifying and testing software components using ADL. Technical Report SMLI TR-94-23, Sun Microsystems Laboratories, Inc., Mountain View, CA, Apr. 1994.

[29] I. Shlyakhter. Generating effective symmetry-breaking predicates for search problems. In *Proc. Workshop on Theory and Applications of Satisfiability Testing*, June 2001.

[30] N. J. A. Sloane, S. Plouffe, J. M. Borwein, and R. M. Corless. The encyclopedia of integer sequences. *SIAM Review*, 38(2), 1996. `http://www.research.att.com/~njas/sequences/Seis.html`.

[31] J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall, second edition, 1992.

[32] W. Visser, K. Havelund, G. Brat, and S. Park. Model checking programs. In *Proc. 15th IEEE International Conference on Automated Software Engineering (ASE)*, Grenoble, France, 2000.

Generalized Multipartitioning for Multi-dimensional Arrays*

Alain Darte†
LIP, ENS-Lyon, 46, Allée d'Italie, 69007 Lyon, France.
Alain.Darte@ens-lyon.fr

Daniel Chavarría-Miranda Robert Fowler John Mellor-Crummey
C. S. Dept., MS-132, Rice University, 6100 Main St, Houston, TX USA
{danich,rjf,johnmc}@cs.rice.edu

Abstract

Multipartitioning is a strategy for parallelizing computations that require solving 1D recurrences along each dimension of a multi-dimensional array. Previous techniques for multipartitioning yield efficient parallelizations over 3D domains only when the number of processors is a perfect square. This paper considers the general problem of computing multipartitionings for d-dimensional data volumes on an arbitrary number of processors. We describe an algorithm that computes an optimal multipartitioning onto all of the processors for this general case. Finally, we describe how we extended the Rice dHPF compiler for High Performance Fortran to generate code that exploits generalized multipartitioning and show that the compiler's generated code for the NAS SP computational fluid dynamics benchmark achieves scalable high performance.

1. Introduction

Line sweeps are used to solve one-dimensional recurrences along each dimension of a multi-dimensional discretized domain. This computational method is the basis for Alternating Direction Implicit (ADI) integration – a widely-used numerical technique for solving partial differential equations such as the Navier-Stokes equation [4, 13, 15] – and is also at the heart of a variety of other numerical methods and solution techniques [15]. Parallelizing computations based on line sweeps is important because these computations address important classes of problems and they are computationally intensive.

However, parallelizing multi-dimensional line sweep computations is difficult because for each of multiple data dimensions, recurrences serialize computation along that dimension. Using standard block partitionings, which assign a single hyper-rectangular volume of data to each processor, there are two reasonable parallelization strategies. A **static block unipartitioning** partitions one of the array dimensions for the entire computation. To achieve significant parallelism with this type of partitioning, one must exploit wavefront parallelism within each sweep. In wavefront computations, there is a tension between using small messages to maximize parallelism by minimizing the length of pipeline fill and drain phases, and using larger messages to minimize communication overhead in the computation's steady state when the pipeline is full. A **dynamic block partitioning** involves partitioning some subset of the dimensions, performing line sweeps in all unpartitioned dimensions locally, and then transposing the data (when necessary) between sweeps so that each of the sweeps, in turn, can be performed locally. While a dynamic block partitioning achieves better efficiency during a (local) sweep over a single dimension compared to a (wavefront) sweep using a static block unipartitioning, the cost of its data transposes can be substantial.

To support better parallelization of line sweep computations, a third sophisticated strategy for partitioning data and computation known as **multipartitioning** was developed [4, 13, 15]. This strategy partitions arrays of $d \geq 2$ dimensions among a set of proces-

*This research was supported in part by the Los Alamos National Laboratory Computer Science Institute (LACSI) through LANL contract number 03891-99-23 as part of the prime contract (W-7405-ENG-36) between the DOE and the Regents of the University of California.

†This work performed while a visiting scholar at Rice University.

sors so that for a line sweep computation along any dimension of an array, all processors are active in each step of the computation, load-balance is nearly perfect, and only coarse-grain communication is needed. These properties are achieved by (1) assigning each processor a balanced number of tiles in each hyper-rectangular slab defined by a pair of adjacent cuts along a partitioned data dimension and (2) ensuring that for all tiles mapped to a processor, their immediate tile neighbors in any one coordinate direction are all mapped to some other single processor. We later refer to these two properties as the **balance** property, and the **neighbor** property respectively. A study by van der Wijngaart [18] of strategies for hand-coded parallelizations of ADI Integration found that 3D multipartitionings yield better performance than static block or dynamic block partitionings.

All of the multipartitionings described in the literature to date consider only one tile per processor per hyper-rectangular slab along a partitioned dimension. The most broadly applicable of the multipartitioning strategies in the literature is known as **diagonal multipartitioning**. In 2D, these partitionings can be performed on any number of processors, p; however, in 3D they are only useful if p is a perfect square. We consider the general problem of computing optimal multipartitionings for d-dimensional data volumes for an arbitrary number of processors.

In the next section, we describe prior work in multipartitioning. Then, we present our strategy for computing generalized multipartitionings. This has three parts: an objective function for computing the cost of a line sweep computation for a given multipartitioning, a cost-model-driven algorithm for computing the dimensionality and tile size of the best multipartitioning, and an algorithm for computing a mapping of tiles to processors. Finally, we describe an implementation of generalized multipartitioning in the Rice dHPF compiler for High Performance Fortran. We show that it yields scalable high performance when used to parallelize the NAS SP [3] computational fluid dynamics benchmark.

2. Background

Johnsson *et al.* [13] describe a 2D domain decomposition strategy, now known as a multipartitioning, for parallel implementation of ADI integration on a multiprocessor ring. They partition both dimensions of a 2D domain to form a $p \times p$ grid of tiles. They use a tile-to-processor mapping $\theta(i,j) \equiv (i-j) \bmod p$, $0 \le i,j < p$, to map from the $[i,j]$ coordinates of

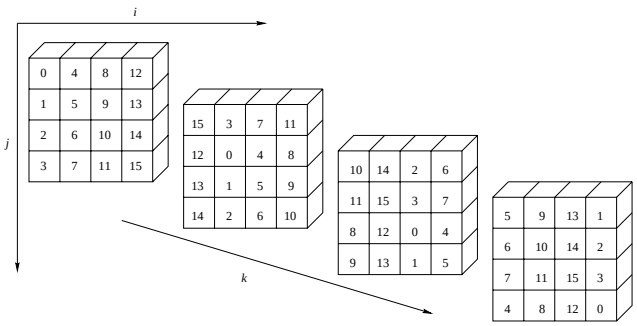


Figure 1. A 3D Multipartitioning.

each tile to its corresponding processor. This partitioning is an instance of a **latin square** [10]. Using this mapping for an ADI computation, each processor exchanges data with only its 2 neighbors in a linear ordering of the processors, which maps nicely to a ring.

Bruno and Cappello [4] devised a 3D partitioning for parallelizing 3D ADI integration computations on a hypercube architecture. They describe how to map a 3D domain cut into $2^d \times 2^d \times 2^d$ tiles on to 2^{2d} processors with a tile-to-processor mapping $\theta(i,j,k)$ based on Gray codes: θ maps tiles adjacent along the i or j dimension to adjacent processors in the hypercube, whereas tiles adjacent along the k dimension map to processors that are exactly two hops distant. They also show that no hypercube embedding is possible in which adjacent tiles always map to adjacent processors.

Naik *et al.* [15] describe **diagonal multipartitionings** for 2D or 3D problems. Diagonal multipartitionings are a generalization of Johnsson *et al.*'s 2D partitioning strategy that are more broadly applicable than the Gray code based mapping described by Bruno and Cappello. The 3D diagonal multipartitionings described by Naik *et al.* partition the data into $p^{\frac{3}{2}}$ tiles, with each processor's tiles arranged along wrapped diagonals through the 3D volume. Figure 1 shows a 3D multipartitioning of this style for 16 processors; the number in each tile indicates the processor that owns the block. This 3D diagonal multipartitioning (there are many) is specified by the tile to processor mapping $\theta(i,j,k) \equiv ((i-k) \bmod \sqrt{p})\sqrt{p} + ((j-k) \bmod \sqrt{p})$ for a domain of $\sqrt{p} \times \sqrt{p} \times \sqrt{p}$ tiles where $0 \le i,j,k < \sqrt{p}$, where $\sqrt{p} = 4$.

More generally, we observe that diagonal multipartitionings can be applied to partition d-dimensional data onto an arbitrary number of processors p by cutting the data into p slices in each dimension, *i.e.,* into an array of p^d tiles. In 2D, this yields an *optimal* multipartitioning (equivalent to those described by Johnsson *et al.*). We call a multipartitioning optimal for a particu-

lar number of processors if no other multipartitioning exists that has lower communication cost according to a cost model that considers both fixed overhead for communicating and overhead proportional to the size of the hyper-surfaces that must be communicated. For $d > 2$, diagonal multipartitionings are only optimal and efficient when $p^{\frac{1}{d-1}}$ is integral.

Bruno and Cappello noted that multipartitionings need not be restricted to having only one tile per processor per hyper-rectangular slab of a multipartitioning [4]. How general can multipartitioning mappings be? A necessary condition to support load-balanced line-sweep computation is that in any hyper-rectangular slab defined by adjacent cuts along a partitioned dimension, each processor must have the same number of tiles. We call any such slab in which each processor has the same number of tiles **balanced**. This raises the question: can we find a way to partition a d-dimensional array into tiles and assign the tiles to processors so that the mapping possesses the **balance** and **neighbor** properties of a multipartitioning? The answer is yes. We show that such an assignment is possible if and only if the number of tiles in each hyper-rectangular slab along any partitioned dimension is a multiple of p ("if" being the difficult part of the proof). We describe a "regular" solution (regular to be defined) that enables us to guarantee that the neighboring tiles along any one coordinate direction of all tiles mapped to a processor all belong to a single processor. This property of multipartitionings is essential for fully-vectorized, directional-shift communication to be efficient.

In Section 3.1, we define an objective function that represents the execution time of a line-sweep computation over a multipartitioned array, and in Section 3.3, we present an algorithm that computes a partitioning of a multi-dimensional array into tiles that is optimal with respect to this objective. In Section 4, we develop a general theory of modular mappings for multipartitioning. We apply this theory to define a mapping of tiles to processors so that each line sweep is perfectly balanced over the processors.

We use the following notation:

- p denotes the number of processors. We write $p = \prod_{j=1}^{s} \alpha_j^{r_j}$ to represent the decomposition of p into prime factors, α_j.
- d is the number of dimensions of the array to be partitioned. The array is of size $\eta_1, \ldots, \eta_d$. The total number of array elements $\eta = \prod_{i=1}^{d} \eta_i$.
- γ_i is the number of tiles into which the array is cut along its i-th dimension. We consider the array of elements as a $\gamma_1 \times \ldots \times \gamma_d$ array of tiles. In our analysis, we assume that γ_i divides η_i evenly and do not consider alignment or boundary problems that must be handled when applying our mappings in practice if this assumption is not valid.

To ensure that each slab is balanced, the number of tiles it contains must be a multiple of p; namely, for each $1 \leq i \leq d$, p should divide $\prod_{j \neq i} \gamma_j$. When this is true, we say that (γ_i) is a **valid partitioning**.

3. Finding the Partitioning

3.1. Objective Function

We consider the cost of performing a line sweep computation along each dimension of a multipartitioned array. The total computation cost is proportional to η, the number of elements in the array. A sweep along the i-th dimension consists of a sequence of γ_i computation phases (one for each hyper-rectangular slab of tiles along dimension i), separated by $\gamma_i - 1$ communication phases. The work in each slab is perfectly balanced, with each processor performing the computation for its own tiles. The total computational work for each processor is roughly $\frac{1}{p}$ of the total work in the sequential computation. The communication overhead is a function of the number of communication phases and the communication volume. Between two computation phases, a hyperplane of array elements is transmitted – the boundary layer for all tiles computed in first phase. The total communication volume for a phase communicated along dimension i is $\prod_{j \neq i} \eta_j$ elements, i.e., $\frac{\eta}{\eta_i}$, yielding a communication volume per processor of $\frac{\eta}{p\eta_i}$. The total execution time for a sweep along dimension i can be approximated by:

$$T_i(p) = K_1 \frac{\eta}{p} + (\gamma_i - 1)(K_2 + K_3(p)\frac{\eta}{\eta_i})$$

where K_1 is a constant that depends on the sequential computation time per data element, K_2 is a constant that depends on the cost of initiating one communication phase (start-up), and $K_3(p)$ is a function of p that reflects the bandwidth-sensitive communication cost per element of hyper-surface area along a cut in dimension i.[1] Define $\lambda_i = K_2 + K_3(p)\frac{\eta}{\eta_i}$; λ_i depends on the domain size, number of processors and machine's communication parameters. The total cost, sweeping

[1] On a parallel machine in which the network bandwidth available is directly proportional to the number of processors, $K_3(p)$ would be proportional to $\frac{1}{p}$, whereas on a bus-based system for which available bandwidth is fixed, $K_3(p)$ would be a constant.

in all dimensions, is thus

$$T(p) = d\left(K_1\frac{\eta}{p} - \sum_{i=1}^{d}\lambda_i\right) + \sum_{i=1}^{d}\gamma_i\lambda_i$$

Assuming that p, η, and the η_i's are given, the first term is a constant, and what we want to minimize is the second term $\sum_{i=1}^{d}\gamma_i\lambda_i$.

Remark: If the number of phases is the critical term, the objective function can be simplified to $\sum_i \gamma_i$. If the volume of communications is the critical term, the objective function can be simplified to $\sum_i \frac{\gamma_i}{\eta_i}$, which means it is preferable to partition dimensions that are larger into relatively more pieces. For example, in 3D, even for a square number of processors (e.g., $p = 4$), if the data domain has a short extent in one dimension, it is preferable to use a 2D partitioning of the other 2 dimensions rather than a 3D partitioning. Indeed, if η_1 and η_2 are at least 4 times larger than η_3, then cutting each of the first 2 dimensions into 4 pieces ($\gamma_1 = \gamma_2 = 4$, $\gamma_3 = 1$) leads to a smaller volume of communication than a "classical" 3D partitioning in which each dimension is cut into 2 pieces ($\forall i$, $\gamma_i = 2$). The extra communication while sweeping along the first 2 dimensions is offset by the absence of communication in the local sweep along the last one.

We now address the problem of minimizing $\sum_i \gamma_i\lambda_i$ with the constraint that, for any fixed i, p divides the product of the γ_j's, $j \neq i$. We give a practical algorithm, based on an (optimized) exhaustive search, exponential in s (the number of distinct factors) and the r_i's (see the decomposition of p into prime factors), but whose complexity in p grows slowly. From a theoretical point of view, we do not know whether this minimization problem is NP-complete, even for a fixed dimension $d \geq 3$, even if $\forall i$, $\lambda_i = 1$, or if there is an algorithm polynomial in $\log p$ or even in the s values $\log r_i$. If p has only one prime factor, a greedy approach leads to a polynomial (polynomial in $\log p$) algorithm (see [8]). However, we do not know if an extension of this greedy approach can lead to a polynomial algorithm for an optimal partitioning in the general case.

3.2. Elementary Partitionings

If (γ_i) is a valid partitioning such that $\sum_i \gamma_i\lambda_i$ is minimized, we say that (γ_i) is an **optimal partitioning**. Using the fact that for each $1 \leq i \leq d$, p divides $\prod_{j\neq i}\gamma_j$ and that the objective function increases when the γ_i increase (the λ_i are positive), we can show the following result. (The proof is not difficult, we omit it due to space constraints.)

Lemma 1 *Let (γ_i) be an optimal partitioning. Then, each factor α_j of p, appearing r_j times in the decomposition of p, appears exactly $(r_j + m_j)$ times in (γ_i), where m_j is the maximum number of occurrences of α_j in any γ_i. Furthermore, the number of occurrences of α_j is m_j in at least two γ_i's.*

We can thus restrict to **elementary partitionings**, those that satisfy the conditions of Lemma 1. We can interpret (and manipulate) an elementary partitioning as a distribution of the factors of p into d bins, satisfying a particular constraint on the number of occurrences. Elementary partitionings are those which are not a "multiple" of another possible size; in other words, these are the sizes for which a multipartitioning exists that cannot be obtained by composing it (by paving) from multiple instances of a smaller multipartitioning. For example, in 3D, with 8 processors, only the partitionings $4\times4\times2$, $8\times8\times1$, and their permutations are elementary. With $p = 5\times3\times2$, only the partitionings $10\times15\times6$, $15\times30\times2$, $10\times30\times3$, $5\times30\times6$, $30\times30\times1$ (and permutations) are elementary.

3.3. Exhaustive Enumeration

We now give an algorithm that finds an optimal partitioning by generating all possible elementary partitionings (γ_i), which satisfy the necessary optimality conditions given by Lemma 1, and determining which one yields the lowest cost partitioning. We also evaluate how many candidate partitions there are to give the complexity of our algorithm. For the complexity, we are not interested in the exact number of elementary partitionings, but in the order of magnitude, especially when the number of bins d is fixed (and small, equal to 3, 4, or 5), but when p can be large (up to 1000 for example), since this is the situation we expect to encounter in practice when computing multipartitionings.

The C program shown in Figure 2 generates, in linear time, all possible distributions of r_j instances of a factor α_j of p into d bins that satisfy the $(r_j + m_j)$ optimality condition of Lemma 1. This program is inspired by a program [16] for generating all partitions of a number, which is a well-studied problem (see [17]) since the mathematical work of Euler and Ramanujam. The procedure `Partitions` first selects the maximal multiplicity m of the factor under consideration that may appear in any bin, and uses the recursive procedure `P(n,m,c,t,d)` to generate all distributions of n elements in $(d - t + 1)$ bins (from index t to index d), where each bin can have at most m instances of the factor and at least c bins must have m instances of the factor. Therefore, the initial call is `P(r+m,m,2,1,d)`.

```
// Precondition: d >= 2
void Partitions(int r, int d) {
  int m;
  for (m = (r+d-2)/(d-1); m <= r; m++)
    P(r+m,m,2,1,d);
}

void P(int n, int m, int c, int t, int d) {
  int i;
  if (t==d)
    bin[t] = n;
  else {
    for (i=max(0,n-(d-t)*m);
              i<=min(m-1,n-c*m); i++) {
      bin[t] = i;
      P(n-i,m,c,t+1,d);
    }
    if (n>=m) {
      bin[t] = m;
      P(n-m,m,max(0,c-1),t+1,d);
    }
  }
}
```

Figure 2. Program for generating all possible distributions for one factor.

We now prove the correctness of the program. The procedure `P` selects a number of elements for the bin number t and makes a recursive call with parameter $t+1$ for the selection in the next bin. It is thus clear that all generated solutions are different since each iteration of the loop selects a different number of elements for the current bin. It remains to prove that all solutions generated by `P` are valid (the total number of elements should be $r+m$, each bin should have at most m elements, and there should be at least c bins with m elements), and that all solutions are generated. For that, we prove that `P(n,m,c,t,d)` is always called with parameters for which there exists at least one valid partitioning, that all possible numbers of elements are selected and only those.

Let us first consider the loop in function `Partitions`. Thanks to Lemma 1, it is easy to see that the maximal number of elements in a bin is between $\lceil \frac{r}{d-1} \rceil$ and r. Furthermore, for each such m, there is indeed at least one valid solution with $(r+m)$ elements and two maxima equal to m (if $d \geq 2$), for example the solution where the first two bins have m elements and the $(d-2)$ other bins contain a total of $(r-m)$ elements; for instance, the $r-m$ elements could be distributed so that $q = \lfloor \frac{r-m}{m} \rfloor$ bins contain m elements and one contains $(r-m-mq)$ elements. Thus, if the function `P` is correct, `Partitions` is also correct.

To prove the correctness of the function `P`, we prove by induction on $d-t+1$ (the number of bins) that there is at least one valid solution if and only if $c \leq d-t+1$ and $cm \leq n \leq (d-t+1)m$ and that `P` generates all of them if these conditions are satisfied. These conditions are simple to understand: we need at least cm elements (so that at least c bins have m elements) and at most $(d-t+1)m$ elements, otherwise at least one bin will contain more than m elements.

The terminal case is clear: if we have only one bin and n elements to distribute, the bin should contain n elements. Furthermore, if there is a solution, we should have $c \leq 1$ and $n = m$ if $c = 1$, i.e., $c \leq d-t+1$ and $cm \leq n \leq (d-t+1)m$.

The general case is more tricky. We first select the number of elements i in the bin number t and recursively call `P` for the remaining bins. If we select strictly less than m elements (this selection is in the loop), we will still have to select c bins with m elements for the remaining $(d-t)$ bins, with $(n-i)$ elements. Therefore, the number i that we select should not be too small, nor too large, and we should have $cm \leq n-i \leq (d-t)m$, i.e., $n-(d-t)m \leq i \leq n-cm$. Furthermore, i should be strictly less than m, nonnegative, and at most n. Since c is always positive, the constraint $i \leq n-cm$ ensures $i \leq n$. If the parameters are correct for the bin number t, we also have $c \leq d-t+1$ and if $c = d-t+1$, then the loop has no iteration, thus for an i selected in the loop, we have $c \leq d-t$. Therefore, the recursive call `P(n-i,m,c,t+1,d)` has correct parameters. Finally, if we select m elements for the bin t (after the loop), this is possible only if m is at most n of course, and then it remains to put $(n-m)$ elements into $(d-t)$ bins, with a maximum of m, and at least $\max(0, c-1)$ maxima. Again, the recursive call has correct parameters since we decreased both c and $(d-t)$ and removed m elements.

For generating all optimal solutions to our minimization problem, we first decompose p into prime factors (complexity $O(\sqrt{p})$ by a standard algorithm, but could be less), we then generate all elementary partitionings, which satisfy Lemma 1 for each factor, with the function `Partitions` and we combine them while keeping track of the best overall solution. The overall complexity (excluding the cost of the decomposition of p into prime factors) is the product of the complexity of the function `Partitions` (which is the number of solutions generated by the algorithm) times $(\log_2 p)^3$ (to built the γ_i's and evaluate them). We proved that the total number of generated solutions (i.e., the number of elementary partitionings) is $O\left(\left(\frac{d(d-1)}{2}\right)^{\frac{(1+o(1))\log p}{\log\log p}}\right)$

and that this bound is tight. (The proof is too long to be provided here but is available in the extended version of this paper [8].)

4. Finding the Mapping

In Section 3, we determined a particular way of cutting the array so as to optimize communications: after partitioning, we get an array (of tiles) whose size is (γ_i) for which the objective is minimized. Up to this point, we have assumed that we will be able to assign tiles to processors so that the assignment posesses the *balance* and *neighbor* properties of a multipartitioning. This has not yet been shown, and we need to prove it. We point out that an assignment with the *balance* property is a generalization of the notion of **latin square** that is known as as an **F-hyper-rectangle** [10, page 392]. However, despite this reference, we have not found any paper that gives a construction for such an assignment, or even an existence proof, for our general case. Furthermore, even if such a proof exists, which we are not aware of, our constructive proof is of interest because:

- its tile-to-processor mappings have the neighbor property,
- its tile-to-processor mappings are given by a simple formula, and conversely, for each processor, the list of tiles assigned to it can be easily formulated, which is handy for use in a run-time library,
- it gives a new insight to the properties of "modular" mappings (defined below).

Therefore, we make no further reference to latin squares and F-hyper-rectangles and proceed with a presentation of our proof.

The only property we know so far is that the (γ_i) is a valid partitioning, namely, for each i, p divides $\prod_{j\neq i}\gamma_j$. Our main result is that this condition is sufficient to guarantee a mapping of processors to tiles that possesses both the balance and neighbor properties. Our proof is constructive. For any valid partitioning (γ_i), optimal or not, with or without the additional property of Lemma 1, we give an automatic way to assign a processor number to each tile so that the properties are satisfied. This assignment is done through the use of modular mappings, defined below. The proof of our construction is much too long to be given here. We refer the reader to the extended version of this paper [8] for details of the proof and interesting properties of modular mappings.

The solution we build is one particular assignment, out of a set of legal mappings. It is not unique, and more experiments might show that they are not all equivalent in terms of execution time, for example because of communication patterns. But, currently, with our objective function (Section 3.1), the network topology is not taken into account yet and all valid mappings are considered equally good.

Consider the assignment in Figure 1. Can we give a formula that describes it? There are 16 processors that can be represented as a 2-dimensional grid of size 4×4. For example the processor number $7 = 4 + 3$ can be represented as the vector $(3, 1)$, in general (r, q) where r and q are the remainder and the quotient of the Euclidean division by 4. The assignment in the figure corresponds to $(i-k \bmod 4, j-k \bmod 4)$, which is what we call a **multi-dimensional modular mapping**, i.e., a mapping $M_{\vec{m}}$ from $\mathbb{Z}^d$ to $\mathbb{Z}^{d'}$ defined by an integral $d\times d'$ matrix M and an integral positive vector $\vec{m}$ of dimension d' with $M_{\vec{m}}(\vec{i}) = (M\vec{i}) \bmod \vec{m}$. With such a mapping, each tile is assigned to a "processor number" in the form of a vector. The product of the components of $\vec{m}$ is equal to the number of processors. It then remains to define a one-to-one mapping from the hyper-rectangle $\{\vec{j} \in \mathbb{Z}^{d'} \mid \vec{0} \leq \vec{j} < \vec{m}\}$ onto the processor numbers. This can be done by viewing the processors as a virtual grid of dimension d' of size $\vec{m}$. The mapping $M_{\vec{m}}$ is then an assignment of each tile (described by its coordinates in the d-dimensional array of tiles) to a processor (described by its coordinates in the d'-dimensional virtual grid). (Actually, we need only the case $d' = d - 1$.)

The following definitions summarize the notions of modular mappings and of modular mappings that satisfy the load-balancing property. Given $\vec{b} \in \mathbb{N}^n$, the **hyper-rectangle** defined by $\vec{b}$ is the set $\mathcal{I}_{\vec{b}} = \{\vec{i} \in \mathbb{Z}^n \mid \vec{0} \leq \vec{i} < \vec{b}\}$ (component-wise). A **slice** $\mathcal{I}_{\vec{b}}(i, k_i)$ of $\mathcal{I}_{\vec{b}}$ is defined as the set of all elements of $\mathcal{I}$ whose i-th component is equal to k_i (an integer between 0 and $b_i - 1$). Given a hyper-rectangle $\mathcal{I}_{\vec{b}}$ (or any more general set), a modular mapping $M_{\vec{m}}$ is **one-to-one from $\mathcal{I}_{\vec{b}}$ onto $\mathcal{I}_{\vec{m}}$** if and only if for each $\vec{j} \in \mathcal{I}_{\vec{m}}$ there is one and only one $\vec{i} \in \mathcal{I}_{\vec{b}}$ such that $M_{\vec{m}}(\vec{i}) = \vec{j}$. $M_{\vec{m}}$ is **equally-many-to-one from $\mathcal{I}_{\vec{b}}$ onto $\mathcal{I}_{\vec{m}}$** if and only if the number of $\vec{i} \in \mathcal{I}_{\vec{b}}$ such that $M_{\vec{m}}(\vec{i}) = \vec{j}$ does not depend on $\vec{j}$. Finally, $M_{\vec{m}}$ has the **load-balancing property** for $\mathcal{I}_{\vec{b}}$ if and only if for any slice $\mathcal{I}_{\vec{b}}(i, k_i)$, the restriction of $M_{\vec{m}}$ to $\mathcal{I}_{\vec{b}}(i, k_i)$ is equally-many-to-one onto $\mathcal{I}_{\vec{m}}$.

Because a modular mapping is linear, it is easy to see that the load-balancing property needs to be checked only for the slices that contain $\vec{0}$ (the slices $\mathcal{I}_{\vec{b}}(i, 0)$). Furthermore, if $\vec{b}[i]$ denotes the vector obtained from $\vec{b}$ by removing the i-th component and $M[i]$ denotes the

matrix obtained from M by removing the i-th column, then the images of $\mathcal{I}_{\vec{b}}(i,0)$ under $M_{\vec{m}}$ are the images of $\mathcal{I}_{\vec{b}[i]}$ under the modular mapping $M[i]_{\vec{m}}$. We therefore have the following properties.

Lemma 2 *Given an hyper-rectangle $\mathcal{I}_{\vec{b}}$, a modular mapping $M_{\vec{m}}$ has the load-balancing property for $\mathcal{I}_{\vec{b}}$ if and only if each mapping $M[i]_{\vec{m}}$ is equally-many-to-one from $\mathcal{I}_{\vec{b}[i]}$ to $\mathcal{I}_{\vec{m}}$.*

Lemma 3 *If $M_{\vec{m}}$ is a one-to-one modular mapping from $\mathcal{I}_{\vec{b}'}$ onto $\mathcal{I}_{\vec{m}}$, then $M_{\vec{m}}$ is an equally-many-to-one modular mapping from any multiple $\mathcal{I}_{\vec{b}}$ of $\mathcal{I}_{\vec{b}'}$ onto $\mathcal{I}_{\vec{m}}$.*

Lemmas 2 and 3 explain why we focus on one-to-one modular mappings first, then on equally-many-to-one modular mappings, and finally on modular mappings with the load-balancing property. In the extended version of this paper [8], we explore the properties of such modular mappings, in order to define a provably adequate matrix M and shape $\vec{m}$ for the virtual grid of processors. Our results are linked to previous works on one-to-one modular mappings by Lee and Fortes [14] and Darte, Dion, and Robert [7]. As in [7], the theory we developed is linked to a famous (in covering/packing theory) theorem due to Hajós [12], which has previously been used to generate "juggling schedules" for systolic-like array designs (see [9]). These earlier papers all consider "one-to-one"-like problems; however, many questions remain open in the equally-many-to-one case because the extension of Hajós' theorem to a similar "equally-many-to-one" case is true only up through 3 dimensions. Also, while it is easy to build a one-to-one mapping (just take $\vec{m} = \vec{b}$ and the identity matrix), here we need a more constrained matrix such that any submatrix obtained by removing one column is equally-many-to-one for the corresponding $\vec{b}$ and $\vec{m}$. In other words, to use the terminology in [9], we need to juggle simultaneously in all dimensions.

Here we present our construction of a modular mapping $M_{\vec{m}}$ with the load-balancing property for an index set $\mathcal{I}_{\vec{b}}$ (which is given, $\vec{b}$ is the vector whose components are the γ_i's found in Section 3.3). The freedom we have is that we can choose the matrix M and the modulo vector $\vec{m}$, but with the constraint that the cardinality of $\mathcal{I}_{\vec{m}}$ (the product of the components of $\vec{m}$) is also given (equal to the number of processors p). The only property of $\vec{b}$ we exploit is that $\vec{b}$ is a valid partitioning: the product of any $(d-1)$ components of $\vec{b}$ is a multiple of p. We choose the matrix M with the following form:

$$M = \begin{pmatrix} N & 0 \\ \vec{\lambda} & 1 \end{pmatrix}$$

where N will be computed by induction. Therefore, finally, M will be even triangular, with 1's on the diagonal. We have the following preliminary result.

Lemma 4 *Suppose that m_d divides b_d and that the modular mapping $N_{\vec{m}[d]}$ – in dimension $(d-1)$ – has the load-balancing property for $I_{\vec{b}[d]}$. Then, the modular mapping $M_{\vec{m}}$ – in dimension d – has the load-balancing property for $\mathcal{I}_{\vec{b}}$ if it is equally-many-to-one from the last slice $\mathcal{I}_{\vec{b}}(d,0)$ onto $\mathcal{I}_{\vec{m}}$.*

Proof: In order to check that the mapping defined by M and $\vec{m}$ has the load-balancing property for the rectangular index set $\mathcal{I}_{\vec{b}}$, we have to make sure that it is equally-many-to-one for all slices $\mathcal{I}_{\vec{b}}(i,0)$, $1 \leq i \leq d$ (Lemma 2). Since we assume that this is true for $i = d$, we only have to prove it for the slices $\mathcal{I}_{\vec{b}}(i,0)$ with $i < d$.

Without loss of generality, let us consider the first dimension, i.e., the first slice $\mathcal{I}_{\vec{b}}(1,0)$. Given $\vec{j} \in \mathcal{I}_{\vec{m}}$, let us count the number of vectors $\vec{i} \in \mathcal{I}_{\vec{b}}$ such that $M\vec{i} = \vec{j} \bmod \vec{m}$ and $i_1 = 0$. By definition of M and N, $(M\vec{i} = \vec{j} \bmod \vec{m}) \Leftrightarrow (N\vec{i}[d] = \vec{j}[d] \bmod \vec{m}[d]$ and $\vec{\lambda}.\vec{i}[d] + i_d = j_d \bmod m_d)$ where $\vec{\lambda}$ is the row vector formed by the first $(d-1)$ component of the last row of M. Because $N_{\vec{m}[d]}$ has the load-balancing property for $\mathcal{I}_{\vec{b}[d]}$, there are exactly n vectors $\vec{i}' \in \mathcal{I}_{\vec{b}[d]}$ such that $i'_1 = 0$ and $N\vec{i}' = \vec{j}[d] \bmod \vec{m}[d]$, where n is a positive integer that does not depend on $\vec{j}[d]$. It remains to count the number of values i_d, between 0 and $b_d - 1$, such that $i_d = j_d - \vec{\lambda}.\vec{i}' \bmod m_d$. Since m_d divides b_d, there are exactly b_d/m_d such values, whatever the value $x = (j_d - \vec{\lambda}.\vec{i}' \bmod m_d)$. These are the values $x + km_d$, with $0 \leq k < b_d/m_d$. Therefore, $\vec{j}$ has exactly $(nb_d)/m_d$ pre-images in $\mathcal{I}_{\vec{b}}(1,0)$ and this number does not depend on $\vec{j}$. ■

We define the vector $\vec{m}$ according to the following formula:

$$\forall i,\ 1 \leq i \leq d,\ m_i = \frac{\gcd\left(p, \prod_{j=i}^{d} b_j\right)}{\gcd\left(p, \prod_{j=i+1}^{d} b_j\right)}$$

(By convention, an "empty" product is equal to 1.) Thanks to the previous lemma and the properties of the vector $\vec{m}$ defined this way, we will be able to build M in a recursive manner (see [8]). Because $m_1 = 1$, we will be able to drop, at the end, the first component of the mapping and get a mapping from $\mathbb{Z}^d$ into a subgroup of $\mathbb{Z}^{d-1}$ (or of smaller dimension if some other components of $\vec{m}$ are equal to 1). Once N is built, we write:

$$M = \begin{pmatrix} N & 0 \\ \vec{\lambda} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \vec{u} & T & 0 \\ \rho & \vec{z} & 1 \end{pmatrix}$$

```
// Precondition: d >= 2
void ModularMapping(int d) {
  int i,j,r,t;
  for (i=1; i<=d; i++)
    for (j=1; j<=d; j++)
      if ((j==1) || (i==j)) M[i][j] = 1;
      else M[i][j] = 0;

  for (i=2; i<=d; i++) {
    r = m[i];
    for (j=i-1; j>=2; j--) {
      t = r/gcd(r, b[j]);
      for (k=1; k<=i-1; k++) {
        M[i][k] -= t*M[j][k];
      }
      r = gcd(t*m[j],r);
    }
  }
}
```

Figure 3. Program for generating a mapping with the load-balancing property.

and we define ρ and $\vec{z}$ (a row vector) such that $\vec{z} = -\vec{t}T$ and $\rho = 1 - \vec{t}\vec{u}$, where the row vector $\vec{t}$, with $(d-2)$ components, is defined by the following (decreasing) recurrence (with the help of an intermediate vector $\vec{r}$):

- $r_{d-1} = m_d$,
- for $1 \leq i \leq d-2$, $t_i = \frac{r_{i+1}}{\gcd(b_{i+1}, r_{i+1})}$ and $r_i = \gcd(t_i m_{i+1}, r_{i+1})$.

This recurrence is linked to the *symbolic* computation of some **Hermite form** that we use to be able to apply Lemma 4 and prove the validity of the recursive construction. See details in [8].

This schema is implemented by the C program shown in Figure 3 (rows and columns are from 1 to d). In our actual implementation of this algorithm, we augment the basic kernel presented to compute the final matrix modulo the corresponding values of $\vec{m}$ as well as apply some strategies (*e.g.,* alternating signs of $\vec{t}$, or pre-permuting the components of $\vec{b}$) to make coefficients smaller.

5. Experiments

We extended the Rice dHPF compiler for High Performance Fortran to generate code based on generalized multipartitionings.

Multipartitioning within the dHPF compiler is implemented as a generalization of `BLOCK`-style HPF partitionings [5, 6]. The dHPF compiler analyzes communication and reduces loop bounds as if a multipartitioned template is a standard `BLOCK` partitioned template mapped onto an array of processors of symbolic extent. The main difference comes in the interpretation that the compiler gives to the `PROCESSORS` directive. When using multipartitioning, the number of processors cannot be specified on a per dimension basis for dimensions of the template because each hyperplane defined by a partitioning along a multipartitioned template dimension is distributed among all processors. A multipartitioned template is partitioned into tiles according to the rank and extent of the virtual processor array. These tiles are then assigned in a skewed-cyclic fashion to the processors as described in previous sections.

There are several important issues for correctly generating efficient code for multipartitioned distributions. First, the order in which a processor's tiles are enumerated has to satisfy any loop-carried dependences present in the original loop from which the multipartitioned loop has been generated. Second, communication that has been fully vectorized out of a loop nest should not be performed on a tile-by-tile basis; instead it should be performed for all of a processor's tiles at once. Communication aggregation is more tricky than for diagonal multipartitionings since generalized multipartitionings have multiple tiles per hyperrectangular slab, but it is possible because generalized multipartitionings also possess the *neighbor* property described earlier in Section 1. Third, communication caused by loop-carried dependences should not be performed on a tile-by-tile basis either. Instead, communication should be vectorized for all tiles within a hyperrectangular slab along the partitioned dimension.

By using a multipartitioned data distribution in conjunction with sophisticated data-parallel compiler optimizations, we are closing the performance gap between compiler-generated and hand-coded implementations of line-sweep computations. Earlier results and details about dHPF's compilation techniques can be found elsewhere [6, 5, 1, 2]. Here we present results from applying generalized multipartitioning in the context of a compiler-based parallelization of the NAS SP computational fluid dynamics application benchmark [3, 6] for the "class B" problem size of 102^3.

The most important analysis and code generation techniques used to obtain high-performance multipartitioned applications by the dHPF compiler are: partial replication of computation to reduce communication frequency and volume, communication vectorization, aggressive communication placement, and communication aggregation to reduce the number of mes-

# CPUs	hand-coded	dHPF	% diff.
1	0.95	0.91	3.84
2		1.43	
4	2.96	2.93	1.00
6		5.06	
8		7.57	
9	7.95	8.04	-1.14
12		11.80	
16	16.64	16.25	2.34
18		18.54	
20		19.03	
24		22.25	
25	27.44	24.32	11.38
32		32.22	
36	38.46	38.83	-0.97
45		39.78	
49	48.37	51.49	-6.46
50		47.35	
64	76.74	59.84	22.02
72		66.96	
81	81.40	70.63	13.23

Table 1. Comparison of hand-coded and dHPF speedups for NAS SP (class B).

sages. In addition, we use an extended on-home directive (inspired by the HPF/JA `EXT_HOME` directive[11]) to partially replicate computation into a processor's shadow regions, and the HPF/JA `LOCAL` directive to eliminate unnecessary communication for values that were previously explicitly computed in a processor's shadow region.

We performed these experiments on a SGI Origin 2000 with 128 250MHz R10000 CPUs, each CPU has 32KB of L1 instruction cache, 32KB of L1 data cache and an unified, two-way set associative L2 cache of 4MB.

Table 1 compares the performance of a hand-coded MPI version of the SP benchmark developed at NASA Ames Research Center with an MPI version generated by the dHPF compiler.[2] The hand-coded version uses 3D diagonal multipartitioning and thus can only be run on a perfect square number of processors. The dHPF-generated code MPI uses generalized multipartitioning which enables the code to be run on arbitrary numbers of processors. As Table 1 shows, the performance of the dHPF-generated code is quite close to (and sometimes exceeds) the performance of the hand-coded MPI for numbers of processors that are perfect squares. When the number of processors is a perfect square, the generalized multipartitionings used by the dHPF-generated code are exactly diagonal multipartitionings. These measurements show that our implementation of generalized multipartitionings is efficient in the case of diagonal multipartitionings, in which each processor has one tile per hyperplane of the partitioning. Both the hand-coded and dHPF-generated versions of SP deliver roughly linear speedup on numbers of processors that are perfect squares.

In the measurements taken of the dHPF-generated code for numbers of processors that are not perfect squares, we see that generalized multipartitionings deliver near linear speedup in these cases as well. The cases we have measured exploiting generalized multipartitioning are ones in which the factors of the number of processors are small primes. Performance would be less for numbers of processors that are prime or have large prime factors because computation would be divided into a large number of phases and communication volume grows in proportion to the number of phases. Currently, the code generated by dHPF cannot exploit generalized multipartitionings when the block size on any processor falls below the shift width associated with communication operations, which happens when a dimension is partitioned many times (as occurs with large primes and prime factors). This limitation prevents experiments with generalized multipartitionings using the 102^3 problem size of the SP benchmark on numbers of processors that are large primes or have large prime factors.[3]

Overall, these preliminary experiments show that generalized multipartitionings are of practical as well as theoretical interest and can be used to efficiently parallelize applications using multipartitioning in a wider range of cases.

6. Conclusions

This paper describes an algorithm for computing an optimal multipartitioning of d-dimensional arrays, $d > 2$, onto an arbitrary number of processors, p. Our algorithm minimizes cost according to an objective function that measures communication in line sweep computations. Previously, optimal multipartitionings could be computed only when $p^{\frac{1}{d-1}}$ is integral. We show that a partitioning in which the number of tiles in each hyperrectangular slab is a multiple of the num-

[2] All speedups presented are relative to the original sequential version of the code.

[3] To be perfectly clear, this limitation applies only to code generated by the dHPF compiler; the *technique* of generalized multipartitioning itself is completely general.

ber of processors — an obvious necessary condition — is also a sufficient condition for a multipartitioned mapping of tiles to processors. We present a constructive method for building the mapping of tiles to processors using new techniques based on modular mappings and demonstrate experimentally that code using generalized multipartitionings is both scalable and efficient.

Currently, when we multipartition a d-dimensional array onto p processors, we force *all* processors to participate in the computation; however, this may lead to suboptimal performance. If the partitioning is not **compact**, *i.e.*, the number of tiles per processor is large relative to a diagonal multipartitioning (more precisely, when $\prod_{i=1}^{d} \gamma_i$ is large compared to $p^{\frac{d}{d-1}}$), and the cost of communicating at tile boundaries is not small compared to the cost of the computation on tile data (the relative cost of communication to computation is proportional to the surface to volume ratio in the partitioning: $\sum_{i=1,d} \frac{\gamma_i}{\eta_i}$), it will be faster to drop back to a nearby lower number of processors for which a compact partitioning exists. For example, table 1 shows that for the 102^3 problem size, a $5 \times 10 \times 10$ decomposition on 50 processors is slower than a $7 \times 7 \times 7$ decomposition on 49 processors for NAS SP. Given a cost function (see Section 3.1) that models the cost of computation as well as communication, our algorithm could be used to search for the most efficient partitioning, which will occur on some number of processors between $\lfloor p^{\frac{1}{d-1}} \rfloor^{d-1}$ (for which a diagonal multipartitioning is possible) and p as long as the communication term is not dominant.

Acknowledgments

The authors wish to gratefully acknowledge the anonymous reviewers for their thoughtful comments which helped us improve the presentation of this paper.

References

[1] V. Adve, G. Jin, J. Mellor-Crummey, and Q. Yi. High Performance Fortran compilation techniques for parallelizing scientific codes. In *SC'98: High Performance Computing and Networking*, Orlando, FL, Nov. 1998.

[2] V. Adve and J. Mellor-Crummey. Using integer sets for data-parallel program analysis and optimization. In *SIGPLAN'98 Conference on Programming Language Design and Implementation*, Montreal, Canada, Jun. 1998.

[3] D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow. The NAS parallel benchmarks 2.0. Technical Report NAS-95-020, NASA Ames Research Center, Dec. 1995.

[4] J. Bruno and P. Cappello. Implementing the beam and warming method on the hypercube. In *3rd Conference on Hypercube Concurrent Computers and Applications*, pages 1073–1087, Pasadena, CA, Jan. 1988.

[5] D. Chavarría-Miranda and J. Mellor-Crummey. Towards compiler support for scalable parallelism. In *5th Workshop on Languages, Compilers, and Runtime Systems for Scalable Computers*, LNCS 1915, pages 272–284, Rochester, NY, May 2000. Springer-Verlag.

[6] D. Chavarría-Miranda, J. Mellor-Crummey, and T. Sarang. Data-parallel compiler support for multipartitioning. In *European Conference on Parallel Computing (Euro-Par)*, Manchester, United Kingdom, Aug. 2001.

[7] A. Darte, M. Dion, and Y. Robert. A characterization of one-to-one modular mappings. *Parallel Processing Letters*, 5(1):145–157, 1996.

[8] A. Darte, J. Mellor-Crummey, R. Fowler, and D. Chavarría. On efficient parallelization of line-sweep computations. Research Report RR2001-45, LIP, ENS-Lyon, France, 2001.

[9] A. Darte, R. Schreiber, B. R. Rau, and F. Vivien. A constructive solution to the juggling problem in systolic array synthesis. In *International Parallel and Distributed Processing Symposium (IPDPS'00)*, pages 815–821, Cancun, Mexico, May 2000.

[10] J. Dénes and A. D. Keedwell. *Latin Squares: New Developments in the Theory and Applications*. North Holland, 1991.

[11] J. A. for High Performance Fortran. HPF/JA language specification (version 1.0). Available at URL `http://www.tokyo.rist.or.jp/jahpf/spec/index-e.html`, Jan. 1999.

[12] G. Hajós. Über einfache und mehrfache Bedeckung des n-dimensionalen Raumes mit einen Würfelgitter. *Math. Zschrift*, 47:427–467, 1942.

[13] S. L. Johnsson, Y. Saad, and M. H. Schultz. Alternating direction methods on multiprocessors. *SIAM Journal of Scientific and Statistical Computing*, 8(5):686–700, 1987.

[14] H. J. Lee and J. A. Fortes. On the injectivity of modular mappings. In *Application Specific Array Processors*, pages 237–247, San Francisco, California, Aug. 1994. IEEE Computer Society Press.

[15] N. Naik, V. Naik, and M. Nicoules. Parallelization of a class of implicit finite-difference schemes in computational fluid dynamics. *International Journal of High Speed Computing*, 5(1):1–50, 1993.

[16] J. Sawada. C program for computing all numerical partitions of n whose largest part is k. Information on Numerical Partitions, `http://www.theory.csc.uvic.ca/~cos/inf/nump/NumPartition.html`, 1997.

[17] N. J. A. Sloane. The on-line encyclopedia of integer sequences. `http://www.research.att.com/~njas/sequences`, 2001.

[18] R. F. Van der Wijngaart. Efficient implementation of a 3-dimensional ADI method on the iPSC/860. In *Supercomputing 1993*, pages 102–111. IEEE Computer Society Press, 1993.

Generating Indecomposable Permutations

Andrew King
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

Abstract

An indecomposable permutation π on $[n]$ is one such that $\pi([m]) = [m]$ for no $m < n$. We consider indecomposable permutations and give a new, inclusive enumerative recurrence for them. This recurrence allows us to generate all indecomposable permutations of length n in transposition Gray code order, in constant amortized time (CAT). We also present a CAT generation algorithm which is based on the Steinhaus-Johnson-Trotter algorithm for generating all permutations of length n. The question of whether or not there exists an adjacent transposition Gray code for indecomposable permutations remains open.

1 Introduction

A permutation π on the interval $[n]$ is indecomposable if and only if $\pi([m]) = [m]$ for no $m < n$. In other words, if and only if it has no proper prefix which is itself a permutation. It is easy to see that there is one indecomposable permutation of length 1, one such permutation of length 2, and three such permutations of length 3.

Indecomposable permutations (sometimes called irreducible permutations) were introduced by Comtet [1, 2], who enumerated them and discussed them in the more general context of permutations with a given number of components (see [2], Exercise 6.14). They have since been investigated in several

contexts, mostly combinatorial and algebraic. We seek to generate them quickly and in a meaningful order.

2 Combinatorial Issues

Let the set of indecomposable permutations of length n be denoted I_n. Comtet [1, 2] noted that $|I_n|$, the number of indecomposable permutations of length n, has the generating function

$$f(t) = 1 - \frac{1}{\sum_{n=1}^{\infty} n! t^n}. \tag{1}$$

The well-known recurrence for $|I_n|$ considers cases of permutations which are not indecomposable. Consider the number of decomposable permutations π on $[n]$ having i as the smallest integer such that $\pi([i]) = [i]$. It is easy to see that there are $|I_i|$ such prefixes, and the other $n-i$ elements can be permuted arbitrarily, therefore there are $|I_i|(n-i)!$ such sequences. The prefix length i lies between 1 and $n-1$, and there are $n!$ permutations in total. This yields the recurrence

$$|I_n| = n! - \sum_{i=1}^{n-1} |I_i|(n-i)! \tag{2}$$

This recurrence is simple, but not particularly useful, as it uses exclusion and is therefore unlikely to help us in finding a Gray code. The more useful recurrence follows:

Theorem 1.

$$|I_n| = \sum_{r=2}^{n} \sum_{j=0}^{r-2} |I_{n-j-1}| j! \tag{3}$$

$$= \sum_{j=0}^{n-2} (n-j-1) |I_{n-j-1}| j! \tag{4}$$

Proof. Consider the number of indecomposable permutations on $[n]$ with first element r. Remove the first element and subtract 1 from any element greater than r. The result is a permutation π on $[n-1]$. Consider the

j	$r=2$	$r=3$	$r=4$	$r=5$
3				**51234**
2			**41253**	51243
3				**51324**
1		**31452**	41352	51342
1		**31524**	41523	51423
1		**31542**	41532	51432
3				**52134**
2			**42153**	52143
3				**52314**
0	**23451**	32451	42351	52341
0	**23514**	32514	42513	52413
0	**23541**	32541	42531	52431
3				**53124**
0	**24153**	34152	43152	53142
3				**53214**
0	**24351**	34251	43251	53241
0	**24513**	34512	43512	53412
0	**24531**	34521	43521	53421
0	**25134**	35124	45123	54123
0	**25143**	35142	45132	54132
0	**25314**	35214	45213	54213
0	**25341**	35241	45231	54231
0	**25413**	35412	45312	54312
0	**25431**	35421	45321	54321

Table 1: Indecomposable permutations with parameters r and j for $n = 5$

largest $m < n$ such that $\pi([j]) = [j]$ (let $j = 0$ if none exists). Remove this prefix and subtract j from each element. The result is a permutation on $[n - j - 1]$, and this must be indecomposable since decomposability would imply that j was chosen incorrectly. So there are $|I_{n-j-1}|$ such suffixes, and for it there are $j!$ possible prefixes of length j. Since the original permutation is indecomposable, $0 \leq j \leq r - 2$. This yields the first identity. The second follows by simple arithmetic. See Table 1 as an example of the parameters r and j. □

This parameterization is essential to generating the permutations in Gray

code order in Section 4.

3 Generation in SJT Order

The Steinhaus-Johnson-Trotter (SJT) algorithm (see [5], p. 136) is a CAT generation algorithm for all permutations on $[n]$ in adjacent transposition Gray code order. Algorithm 1 generates all permutations in the same order as the SJT algorithm, put outputs only those which are indeomposable.

Algorithm 1 can be turned into the SJT algorithm by removing lines 9 to 16 and changing the condition on line 4 to be merely $m > n$. Initially, $spp[i] = 1$ and $p[i] = i$ for all i. As with the SJT algorithm, the initial call is Perm(1).

Algorithm 1 Generate indecomposable permutations in SJT order

```
 1: procedure Perm ( int m )
 2: local int i, j, t
 3: begin
 4:   if m > n and spp[n] = n then
 5:     Printit;
 6:   else
 7:     Perm(m + 1);
 8:     for i := 1 to m − 1 do
 9:       p[m] := p[m] + dir[m];
10:       for j := m to n do
11:         if p[j] ≤ s[j − 1] then
12:           spp[j] := j;
13:         else
14:           spp[j] := spp[j − 1];
15:         end if
16:       end for
17:       t := π^{-1}[m];   π[t] := π[t + dir[m]];   π[t + dir[m]] := n;
18:       π^{-1}[m] := t + dir[m];   π^{-1}[π[t]] := t;
19:       Perm(n + 1);
20:     end for
21:   end if
22:   dir[m] := −dir[m];
23: end
```

$p[i]$ represents the position that i would hold in the permutation if all numbers greater than i were to be removed. For example, in the permutation 635142, p would be [1, 2, 1, 3, 2, 1]. This explains line 9, because as a property of the SJT algorithm, m is always transposed with a smaller number, so $p[m]$ will always change by 1, and no other entry of p will change.

$spp[i]$ is the length of the smallest prefix which is itself a permutation, once all numbers greater than i have been removed. For example, in the permutation 635142, spp would be [1, 1, 3, 4, 5, 6]. Note that $spp[1] = 1$ always, and for $i > 1$, $spp[i] = i$ if and only if removing all numbers greater than i leaves an indecomposable permutation. In this example, 1, 312, 3142, 35142, and 635142 are all indecomposable.

Lemma 2. *Let π be a permutation on $[n]$. For $1 < i \leq n$,*

$$spp[i] = \begin{cases} i & \text{if } p[i] \leq spp[i-1] \\ spp[i-1] & \text{otherwise} \end{cases} \tag{5}$$

This follows from the fact that, when inserting i into a permutation on $[i-1]$, if i comes before the end of the smallest prefix which is itself a permutation, the result will be an indecomposable permutation. If i comes after the end of this prefix, then the existing prefix will remain the shortest such prefix.

Theorem 3. *Algorithm 1 generates all indecomposable permutations, and no others.*

Proof. It is clear that a permutation on $[n]$ is indecomposable if and only if $spp[n] = n$, so it remains only to show that lines 9 through 16 maintain p and spp correctly. We have already illustrated that line 9 increments or decrements $p[m]$ appropriately.

Lemma 2 establishes the appropriate value for $spp[n]$ when n is inserted into a permutation on $[n-1]$. Moving m in π in the algorithm will never change $spp[i]$ for $i < m$; that much is clear. However, it can change $spp[i]$ for $i > m$. By the claim, we can correctly maintain spp by recomputing $spp[m], spp[m+1], \ldots, spp[n]$ in that order. Hence lines 9 through 16 correctly maintain p and spp.

Therefore, adding to the SJT algorithm the specification that a permutation is printed if and only if $spp[n] = n$ ensures that the algorithm generates exactly all indecomposable permutations. □

Theorem 4. *Algorithm 1 runs in constant amortized time.*

Proof. We know from Comtet ([1]) that $\lim_{n\to\infty} |I_n|/(n!) = 1$, and we know further that $n!/|I_n| \leq 2$. Therefore it suffices to show that the algorithm's running time is bounded by a constant factor with respect to the running time of the SJT algorithm, since the SJT algorithm is itself CAT.

Modifying line 4 and adding line 9 clearly adhere to this constraint (because they do no more than add a constant amount of work to each node in the computation tree), so we need only be concerned about the **for** loop beginning at line 10. This loop does a constant amount of work $n - m$ times at each node at distance m from the root of the computation tree, for $m = 0, 1, 2, \ldots, n$. There are $m!$ nodes at distance m from the root, so let us bound the total amount of work done in the tree.

We take as granted that for $n > 0$, $\sum_{i=0}^{n} i! \leq 2 \cdot n!$. This can be proven trivially by induction. Let $W(n)$ be the number of times the inner **for** loop (line 10) is run through in total. Now consider the computation tree for $W(n+1)$. It is the same as the tree for $W(n)$ but with $n+1$ children added to each leaf, and with the inner **for** loop run through one extra time in every node at distance $\leq n$ from the root. There are $\sum_{i=0}^{n} i!$ such nodes, so $W(n+1) = W(n) + \sum_{i=0}^{n} i! \leq W(n) + 2 \cdot n!$. The sequence $\{W(n)\}_{n=1}^{\infty}$ begins $1, 3, 7, 17, 51, \ldots$, so we will show that for $n \geq 4$, $W(n) < n!$, using the basis $W(4) = 17$. Take $n \geq 4$ and suppose that $W(n) < n!$.

$$\begin{aligned} W(n+1) &\leq W(n) + 2 \cdot n! \\ &< 3 \cdot n! \\ &< (n+1)! \end{aligned} \tag{6}$$

So $W(n)$ is bounded by a constant times the number of nodes in the computation tree for $\mathsf{Perm}(n)$. Therefore Algorithm 1 runs in constant time amortized over the number of permutations output by the SJT algorithm, and therefore over the number of its own output permutations. □

4 Generation in Gray Code Order

In this section we first present the theory behind the Gray code, then present the Gray code generation algorithm.

4.1 Existence of a Gray Code

There is a transposition Gray code for indecomposable permutations which uses the partitioning of the set of indecomposable permutations induced jointly by the parameters r and j, discussed in Section 2. We must introduce several terms.

Terminology 1. *We shall denote the transposition Gray graph of indecomposable permutations G_n. Let the subgraph of G_n induced by those permutations which begin with r be denoted $G_{n,r}$. Let the subgraph of $G_{n,r}$ induced by those permutations with parameter j (as in Section 2) be denoted $G_{n,r,j}$. Let the vertex sets of these graphs be denoted I_n, $I_{n,r}$, and $I_{n,r,j}$ respectively.*

Terminology 2. *Consider two finite sets S_1, $S_2 \subseteq \mathbb{Z}^+$ such that $|S_1| = |S_2| = n > 0$. Now consider two bijections (permutations), $\pi_1 : S_1 \to [n]$ and $\pi_2 : S_2 \to [n]$. We say that π_1 and π_2 are* equivalent *permutations if and only if for any i and j such that $0 < i, j \leq n$, we have that $\pi_1^{-1}(i) < \pi_1^{-1}(j) \iff \pi_2^{-1}(i) < \pi_2^{-1}(j)$. When an underlying set S is implied or known, and π is a permutation on a set of size $|S|$, let $E(\pi)$ denote the permutation on S which is equivalent to π.*

Lemma 5. *Let j satisfy $0 \leq j \leq n-2$. Then for any two r_1 and r_2 satisfying $j+2 \leq r \leq n$, $G_{n,r_1,j} \cong G_{n,r_2,j}$.*

Proof. Consider the bijection $f : I_{n,r_1,j} \to I_{n,r_2,j}$ which maps $\pi_1 \in I_{n,r_1,j}$ to $\pi_2 \in I_{n,r_2,j}$ if and only if when the first elements of each permutation (i.e. the prefixes of length 1) are removed, the remaining permutations are equivalent. The image of π_1 is unique, as there can only be one permutation on $[n] \setminus r_2$ equivalent to a given permutation on $[n] \setminus r_1$.

Since r_1 and r_2 are both greater than $j+1$, it is easy to see that $f(\pi_1) \in I_{n,r_2,j}$. Now consider a permutation $\pi_1' \in I_{n,r_1,j}$ which is reached from π_1 by a single transposition. $f(\pi_1')$ is reached from π_2 by transposing the same two positions. By generality, the same rule applies in the other direction, so f is a graph isomorphism. □

At this point, we must define special vertices in the Gray graph.

Terminology 3. *Let the* top *vertices of $G_{n,r,j}$ and G_n be denoted* $\text{top}_{n,r,j}$ *and* top_n *respectively. Let the* bottom *vertices of $G_{n,r,j}$ and G_n be denoted*

$\mathrm{bot}_{n,r,j}$ *and* bot_n *respectively. Let them be defined as follows:*

$$\begin{aligned}
\mathrm{top}_n &= 2, 3, 4, \ldots, n, 1 \\
\mathrm{bot}_n &= n, 1, 2, 3, \ldots, n-1 \\
\mathrm{top}_{n,n,j} &= \begin{cases} n, 2, 3, \ldots, n-1, 1 & \text{if } j = 0 \\ n, 1, 3, 4, \ldots, n-1, 2 & \text{if } j = 1 \\ n, 1, 2, \ldots, j-2, j, j-1, n-1, j+1, j+2, \ldots, n-2 & \text{otherwise} \end{cases} \\
\mathrm{bot}_{n,n,j} &= n, 1, 2, \ldots, j-2, j-1, j, n-1, j+1, j+2, \ldots, n-2
\end{aligned}$$

For $r \neq n$, $\mathrm{top}_{n,r,j}$ *and* $\mathrm{bot}_{n,r,j}$ *are those vertices in* $G_{n,r,j}$ *which are isomorphic to* $\mathrm{top}_{n,n,j}$ *and* $\mathrm{bot}_{n,n,j}$ *respectively, under the isomorphism described in the proof of Lemma 5.*

The following facts are to be noted:

- $\mathrm{top}_n = \mathrm{top}_{n,2,0} = 2, E(\mathrm{top}_{n-1})$.
- $\mathrm{bot}_n = \mathrm{bot}_{n,n,n-2}$.
- For $j \geq 2$, $\mathrm{top}_{n,n,j} = n, 1, 2, \ldots, j-2, j, j-1, E(\mathrm{bot}_{n-j-1})$.
- $\mathrm{bot}_{n,n,j} = n, 1, 2, \ldots, j-2, j-1, j, E(\mathrm{bot}_{n-j-1})$.
- Transposing positions $j+2$ and $j+4$ in $\mathrm{bot}_{n,n,j}$ results in $\mathrm{top}_{n,n,j+2}$.
- Transposing positions 2 and n in $\mathrm{top}_{n,n,0}$ results in $\mathrm{top}_{n,n,1}$.
- Because of isomorphism, the above apply to all values of r, not just n.
- For $2 \leq r < n$, transposing elements (not positions) r and $r+1$ in $\mathrm{bot}_{n,r,j}$ results in $\mathrm{bot}_{n,r+1,j}$.

Let P_n be the transposition Gray graph of all permutations on $[n]$. Note, then, that $G_{n,r,j} \cong G_{n-j-1} \times P_j$. To see this, consider the explanation of Theorem 1.

Lemma 6. *To show that there is a Gray code for* I_n*, it suffices to show that for* $0 \leq j \leq n-2$*, there is a Gray code for* $I_{n,n,j}$ *that begins at* $\mathrm{top}_{n,n,j}$ *and ends at* $\mathrm{bot}_{n,n,j}$.

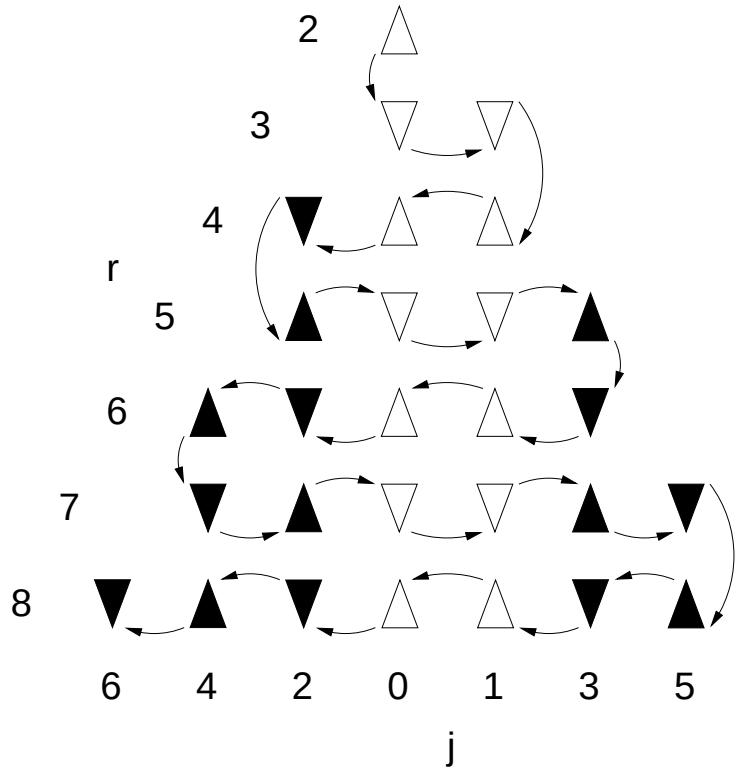


Figure 1: The traversals of $G_{n,r,j}$ graphs, combined to traverse G_n. Empty triangles indicate traversal from top to bottom, or bottom to top. Filled triangles indicate traversal from middle to bottom, or bottom to middle.

Proof. Suppose there are such Gray codes. Then by isomorphism, there is a top-to-bottom Gray code for $I_{n,2,0}$. We can traverse it, then make a transposition to reach $\text{bot}_{n,3,0}$. We can then traverse the previous Gray code in reverse to reach $\text{top}_{n,3,0}$ and make a transposition to reach $\text{top}_{n,3,1}$. Since there is a Gray code for $I_{n,n,1}$, we can traverse $G_{n,3,1}$ in Gray code order, then make a transposition to jump from $\text{bot}_{n,3,1}$ to $\text{bot}_{n,4,1}$.

Again, we can traverse from $\text{bot}_{n,4,1}$ to $\text{bot}_{n,4,2}$ by making the transpositions we used to traverse $r = 3$, only in reverse order. We can then jump from $\text{bot}_{n,4,0}$ to $\text{top}_{n,4,2}$. At this point, we can traverse $G_{n,4,2}$ from top to bottom, since there is a Gray code isomorphic to that for $I_{n,n,2}$.

In this fashion, we can continue jumping between values of r and j in the order shown in Figure 1 until we have traversed I_n completely. □

Theorem 7. *There is a transposition Gray code for I_n.*

Proof. We will use strong induction to prove the theorem.

BASIS: $n = 1$. $|I_1| = 1$, so there is a trivial Gray code from top to bottom.

INDUCTION: Assume that for all $0 < i < n$, there is a Gray code for I_i which runs from top_i to bot_i.

$\text{top}_{n,n,0} = n, E(\text{top}_{n-1})$. $\text{bot}_{n,n,0} = n, E(\text{bot}_{n-1})$. So by traversing the Gray code for I_{n-1} in the last $n-1$ positions of the permutation, we can traverse $G_{n,n,0}$ from top to bottom in Gray code order. Similarly, $\text{top}_{n,n,1} = n, 1, E(\text{top}_{n-2})$ and $\text{bot}_{n,n,1} = n, 1, E(\text{bot}_{n-2})$, so we can traverse $G_{n,n,1}$ from top to bottom in Gray code order using the Gray code for I_{n-2}.

We must now address the case where $j \leq 2$, i.e. the top to bottom traversals, using the fact that $G_{n,r,j} \cong G_{n-j-1} \times P_j$. First note that we have a transposition Gray code for P_j whose final permutation is the same as its initial permutation, but with the final two positions transposed: left-to-right SJT. This is the same as the Steinhaus-Johnson-Trotter algorithm, but the element in the leftmost position is moved first, not the element in the rightmost position. In order to traverse $G_{n,n,j}$, we must traverse G_{n-j-1} in the final $n-j-1$ positions once for every permutation of length j.

Recall that $\text{top}_{n,n,j} = n, 1, 2, \ldots, j-2, j, j-1, E(\text{bot}_{n-j-1})$, and $\text{bot}_{n,n,j} = n, 1, 2, \ldots, j-2, j-1, j, E(\text{bot}_{n-j-1})$, and note that we have an even number ($j!$) of permutations of length j. Therefore we can simply traverse G_{n-j-1} from bottom to top, advance the permutation of length j, traverse G_{n-j-1} from top to bottom, advance the permutation, etc., until every permutation

in $I_{n,n,j}$ has been exhausted. Because $j!$ is even and we are using left-to-right SJT, by beginning at $\text{mid}_{n,n,j}$ we ensure that we will end the traversal of $G_{n,n,j}$ at $\text{bot}_{n,n,j}$.

We have now established the necessary conditions given Lemma 6, so there is a Gray code for I_n. □

4.2 Generating the Gray Code

In this section we present a recursive algorithm for generating I_n in Gray code order. The initial call is PrintIt; Indec$(n, 0)$. RevIndec$(n, 0)$ generates I_n in reverse order (from bottom to top). To generate I_n with Indec, we must initialize the permutation p to top_n. Swap simply transposes the two positions, given as parameters, in p.

Indec and RevIndec call Permute, which in turn calls Indec and RevIndec. Every time Permute$(n, j, depth, m,$ true$)$ is called, we must allocate three vectors of length j, as in Algorithm 1: π, π^{-1}, and $dirArr$. To perform the left-to-right SJT traversal of permutations, we initialize π and π^{-1} to $j, j-1, j-2, \ldots, 1$, and we initialize $dirArr[i]$ to 1 for all i.

Indec jumps between the $I_{n,r,j}$ graphs in the order shown in Figure 1, traversing each subgraph by calling Permute. The algorithm uses the subgraph isomorphism described earlier in this section. NextJ and its inverse RevNextJ select the next value of j by simple case analysis. The values of j are changed in Indec, lines 18–24, and in RevIndec, lines 14–20, choosing the positions to transpose by the difference in j. The values of r are changed in Indec, lines 27–31, and in RevIndec, lines 23–27, this time choosing between two transpositions, depending on the structure of the current permutation. We will now prove that this algorithm is CAT, but first we will prove two technical lemmas.

Lemma 8. *Let $W_I(n)$ be the number of times a single call to Indec$(n, 0)$ results in a call to Indec$(i, 0)$ for $i \leq 2$. $W_I(n) = |I_n|$.*

Proof. $W_I(1) = 1$. For $n > 1$,

$$W_I(n) = \sum_{r=2}^{n} \sum_{j=0}^{r-2} W_I(n-j-1) j!. \tag{7}$$

This is because for a given value of j, Indec calls Indec$(n-j-1, depth)$ $j!$ times, and each of these calls contains $W_i(n-j-1)$ calls to Indec$(i, 0)$ for

Algorithm 2 Traverse G_n in Gray code order

```
 1: procedure Indec ( int n, depth )
 2: local int r, j, j'; boolean dir
 3: begin
 4:   if n > 2 then
 5:     for r := 2 to n do
 6:       if r > 2 then
 7:         j := r − 3;
 8:       else
 9:         j := 0;
10:       end if
11:       while true do
12:         dir := (r % 2 = j);
13:         Permute(n, j, depth, 1, dir, true);
14:         j' := NextJ(r, j);
15:         if j' > r − 2 then
16:           break;
17:         end if
18:         if j' = j + 2 then
19:           Swap(depth + j + 2, depth + j + 4); PrintIt;
20:         else if j' = j − 2 then
21:           Swap(depth + j, depth + j + 2); PrintIt;
22:         else
23:           Swap(depth + 2, depth + n); PrintIt;
24:         end if
25:         j := j';
26:       end while
27:       if r < n − 1 then
28:         Swap(depth + 1, depth + r + 2); PrintIt;
29:       else if r = n − 1 then
30:         Swap(depth + 1, depth + r); PrintIt;
31:       end if
32:     end for
33:   end if
34: end
```

Algorithm 3 Traverse G_n in reverse Gray code order

```
 1: procedure RevIndec ( int n, depth )
 2: local int r, j, j'; boolean dir
 3: begin
 4:   if n > 2 then
 5:     for r := n down to 2 do
 6:       j := r − 2;
 7:       while true do
 8:         dir := (r % 2 ≠ j);
 9:         Permute(n, j, depth, 1, dir, true);
10:         j' := RevNextJ(r, j);
11:         if j' > r − 2 then
12:           break;
13:         end if
14:         if j' = j + 2 then
15:           Swap(depth + j + 2, depth + j + 4); PrintIt;
16:         else if j' = j − 2 then
17:           Swap(depth + j, depth + j + 2); PrintIt;
18:         else
19:           Swap(depth + 2, depth + n); PrintIt;
20:         end if
21:         j := j';
22:       end while
23:       if r = n then
24:         Swap(depth + 1, depth + r − 1); PrintIt;
25:       else if r > 2 then
26:         Swap(depth + 1, depth + r + 1); PrintIt;
27:       end if
28:     end for
29:   end if
30: end
```

Algorithm 4 Traverse $G_{n,r,j}$

```
 1: procedure Permute ( int n, j, depth, m; boolean first )
 2: local int i, t
 3: begin
 4:   if m > j then
 5:     if ¬first then
 6:       Printit;
 7:     end if
 8:     if dir then
 9:       Indec(n − j − 1, depth + j + 1)
10:     else
11:       RevIndec(n − j − 1, depth + j + 1)
12:     end if
13:     dir := ¬dir;
14:   else
15:     Permute(n, j, depth, m + 1, first);
16:     first := false;
17:     for i := 1 to m − 1 do
18:       t := π^{-1}[m];   π[t] := π[t + dirArr[m]];   π[t + dirArr[m]] := n;
19:       π^{-1}[m] := t + dirArr[m];   π^{-1}[π[t]] := t;
20:       Permute(n, j, depth, m + 1, false);
21:     end for
22:   end if
23:   dirArr[m] := −dirArr[m];
24: end
```

Algorithm 5 Determine the next value of j to traverse

```
1:  procedure NextJ ( int r, j )
2:  begin
3:    if j = 0 and r is even then
4:      return j + 1;
5:    else if j = 1 and r is odd then
6:      return j − 1;
7:    else if j % 2 = r % 2 then
8:      return j + 2;
9:    else
10:     return j − 2;
11:   end if
12: end
13:
14: procedure RevNextJ ( int r, j )
15: begin
16:   if j = 0 and r is odd then
17:     return j + 1;
18:   else if j = 1 and r is even then
19:     return j − 1;
20:   else if j % 2 = r % 2 then
21:     return j − 2;
22:   else
23:     return j + 2;
24:   end if
25: end
```

$i \leq 2$. Recall that this is the same recurrence as in Equation 3, and since $W_I(1) = 1$, $W_I(n) = |I_n|$. □

Lemma 9. *Let $W_P(n)$ be the number of times a single call to* Indec$(n, 0)$ *results in a call to* Permute(n, j) *for $j \leq 1$. $W_P(n) \leq n!$.*

Proof. $W_I(1) = 0$ and $W_I(2) = 0$. For $n > 2$, Indec$(n, 0)$ calls Permute$(n, 0)$ $n - 1$ times, Permute$(n, 1)$ $n - 2$ times. These are the only such calls that happen in the calls at the top level. Beyond that, we have the familiar recurrence for calls to Permute(n, j) for $j \leq 1$ by recursive calls. This gives us

$$W_P(n) = 2n - 3 + \sum_{j=0}^{n-2} (n - j - 1) \cdot W_P(n - j - 1) \cdot j! \tag{8}$$

$$= 2n - 3 + \sum_{j=0}^{n-4} (n - j - 1) \cdot W_P(n - j - 1) \cdot j!, \tag{9}$$

the second formulation coming from the knowledge that $W_I(i) = 0$ for $i < 2$. The sequence $\{W_P(n)\}_{n=1}^{\infty}$ begins $0, 0, 3, 14, 72, 443, \ldots$. We claim that $W_P(n) \leq |I_n|$ for $n \geq 6$, and will prove it by induction.

Basis: $|I_6| = 461$, $W_P(6) = 443$.

INDUCTION: Let $n \geq 7$. Assume that all $6 \leq i \leq n-1$, $W_P(i) < |I_i|$.

$$\begin{aligned}
W_P(n) &= 2n - 3 + \sum_{j=0}^{n-7} ((n-j-1) \cdot W_P(n-j-1) \cdot j!) \\
&\quad + \sum_{j=n-6}^{n-2} ((n-j-1) \cdot W_P(n-j-1) \cdot j!) && (10) \\
&\leq 2n - 3 + \sum_{j=0}^{n-7} ((n-j-1) \cdot |I_{n-j-1}| \cdot j!) \\
&\quad + \sum_{j=n-6}^{n-2} ((n-j-1) \cdot W_P(n-j-1) \cdot j!) && (11) \\
&= 2n - 3 + |I_n| - \sum_{j=n-6}^{n-2} ((n-j-1) \cdot |I_{n-j-1}| \cdot j!) \\
&\quad + \sum_{j=n-6}^{n-2} ((n-j-1) \cdot W_P(n-j-1) \cdot j!) && (12) \\
&= |I_n| + (2n + 5(n-6)! + 4(n-5)!) \\
&\quad - (3 + 2(n-3)! + (n-2)!) && (13) \\
&\leq |I_n| + 2n - 43(n-6)! - 60(n-5)! && (14) \\
&\leq |I_n| && (15)
\end{aligned}$$

These steps are reasonably straightforward. (13) recalls Equation 4, and the steps that follow are arithmetical facts which rely on n being greater than 6. Since $|I_n| \leq n!$, the lemma is proved. □

These bounds on the call counts help us prove that the algorithm is efficient (CAT).

Theorem 10. *Indec*$(n, 0)$ *generates* I_n *in constant amortized time.*

Proof. Each call counted by $W_I(n)$ and $W_P(n)$ runs in constant time and outputs no permutations (though Permute will still call Indec once), so we can omit them from our analysis, since both $W_I(n)$ and $W_P(n)$ are bounded by $2|I_n|$; the total time is constant per indecomposable permutation generated by the main call, so it suffices to show that the rest of the algorithm is CAT.

Not counting the time taken during the recursive calls to Permute, each call to Indec$(n, depth)$ outputs $(n^2 - n - 2)/2$ permutations (one for each jump between a $G_{n,r,j}$ subgraph) and makes $(n^2 - n)/2$ calls to Permute (one for each $G_{n,r,j}$) in $O(n^2)$ time. Since we can assume $n \geq 3$, $(n^2 - n - 2)/2$ and $(n^2 - n)/2$ are each greater than $n^2/5$. Therefore Indec$(n, depth)$ itself does a constant amount of work per output and a constant amount of work per call to Permute.

Not counting the time taken during the recursive calls, each call to Permute(n, j) outputs $j! - 1$ permutations and makes $j!$ calls to Indec in $O(j!)$ time, which we can see because the SJT algorithm is CAT, and we have added only a constant amount of work per node. Because we can assume that $j \geq 2$, we know that $j! - 1 \geq j!/2$. Therefore we know that Permute(n, j) itself does a constant amount of work per output and a constant amount of work per recursive call made.

We have now shown that each of Indec and Permute does a constant amount of work per permutation output (recall that NextJ runs in constant time), so the entire algorithm is CAT. □

5 Conclusions

We have given two CAT algorithms for generating indecomposable permutations: one generates them by selecting them efficiently in the Steinhaus-Johnson-Trotter algorithm, and one generates them in transposition Gray code order. The Gray code was developed by using a new parameterization of indecomposable permutations. The question of whether or not there is an adjacent transposition Gray code for these permutations remains open.

6 Acknowledgements

I would like to thank Frank Ruskey for presenting the problem to me and helping me along the way, and Joe Sawada for his helpful comments on CAT algorithms.

References

[1] Comtet, Louis. *Sur les coefficients de l'inverse de la série formelle* $\sum n!t^n$. Comptes Rend. Acad. Sci. Paris, A 275, pp 569-572, 1972.

[2] Comtet, Louis. *Advanced Combinatorics*. Dordrecht, Holland: D. Reidel Publ. Co., 1974.

[3] Hertel, Alex and Philip. The stonecarver's Hamilton cycle algorithm. Private communication.

[4] King, Andrew D. *Transposition Gray codes for indecomposable permutations*, B.Sc. honours thesis, University of Victoria, Canada, 2002.

[5] Ruskey, Frank. *Combinatorial Generation, Working Version (1j)*. Victoria, Canada: University of Victoria, 2001.

[6] Sloane, N. J. A. The On-Line Encyclopedia of Integer Sequences. `http://www.research.att.com/~njas/sequences`.

[7] Wilf, Herbert S. *Generatingfunctionology*. San Diego: Academic Press, Inc., 1990.

The Sphere Packing Problem

N. J. A. Sloane

Abstract. A brief report on recent work on the sphere-packing problem.

1991 Mathematics Subject Classification: 52C17
Keywords and Phrases: Sphere packings; lattices; quadratic forms; geometry of numbers

1 Introduction

The sphere packing problem has its roots in geometry and number theory (it is part of Hilbert's 18th problem), but is also a fundamental question in information theory. The connection is via the sampling theorem. As Shannon observes in his classic 1948 paper [37] (which ushered in the age of digital communication), if f is a signal of bandwidth W hertz, with almost all its energy concentrated in an interval of T secs, then f is accurately represented by a vector of $2WT$ samples, which may be regarded as the coordinates of a single point in $\mathbb{R}^n$, $n = 2WT$. Nearly equal signals are represented by neighboring points, so to keep the signals distinct, Shannon represents them by n-dimensional 'billiard balls', and is therefore led to ask: what is the best way to pack 'billiard balls' in n dimensions?

This talk will report on a few selected developments that have taken place since the appearance of Rogers' 1964 book on the subject, proceeding upwards in dimension from 2 to 128. The reader is referred to [16] (especially the third edition, which has 800 references covering 1988-1998) for further information, definitions and references. See also the lattice data-base [31].

2 Dimension 2

The best packing in dimension 2 is the familiar 'hexagonal lattice' packing of circles, each touching six others. The centers are the points of the root lattice A_2. The *density* Δ of this packing is the fraction of the plane occupied by the spheres: $\pi/\sqrt{12} = 0.9069\ldots$.

In general we wish to find Δ_n, the highest possible density of a packing of equal nonoverlapping spheres in $\mathbb{R}^n$, or $\Delta_n^{(L)}$, the highest density of any packing in which the centers form a lattice. It is known (Fejes Tóth, 1940) that $\Delta_2 = \Delta_2^{(L)} = \pi/\sqrt{12}$. An n-dimensional lattice Λ of determinant d and minimal nonzero squared length (or *norm*) μ has packing radius $\rho = \sqrt{\mu}/2$ and density $\Delta = V_n\rho^n/\sqrt{\det\Lambda}$, where

$V_n = \pi^{n/2}/(n/2)!$ is the volume of a unit sphere. The *center density* of a packing is $\delta = \Delta/V_n$.

We are also interested in packing points on a sphere, and especially in the 'kissing number problem': find τ_n (resp. $\tau_n^{(L)}$), the maximal number of spheres that can touch an equal sphere in $\mathbb{R}^n$ (resp. in any lattice in $\mathbb{R}^n$). It is trivial that $\tau_2 = \tau_2^{(L)} = 6$.

3 Dimension 3

In spite of much recent work ([20], [21]) Δ_3 is still unknown; nor is Δ_n known in any dimension above 2. It is conjectured that $\Delta_3 = \pi/\sqrt{18} = 0.74048\ldots$, as in the face-centered cubic (f.c.c.) lattice A_3. Muder [28] has shown that $\Delta_3 \le 0.773055\ldots$. It is worth mentioning, however, that there are packings of congruent ellipsoids with density considerably greater than $\pi/\sqrt{18}$ [3].

In two dimensions the hexagonal lattice is (a) the densest lattice packing, (b) the least dense lattice covering, and (c) is geometrically similar to its dual lattice. There is a little-known three-dimensional lattice that is similar to its dual, and, among all lattices with this property, is both the densest packing and the least dense covering. This is the m.c.c. (or *mean-centered cuboidal*) lattice [11] with Gram matrix

$$\frac{1}{2}\begin{bmatrix} 1+\sqrt{2} & 1 & 1 \\ 1 & 1+\sqrt{2} & 1-\sqrt{2} \\ 1 & 1-\sqrt{2} & 1+\sqrt{2} \end{bmatrix} .$$

In a sense this lattice is the geometric mean of the f.c.c. lattice and its dual the body-centered cubic (b.c.c.) lattice. Consider the lattice generated by the vectors $(\pm u, \pm v, 0)$ and $(0, \pm u, \pm v)$ for real numbers u and v. If the ratio u/v is respectively 1, $2^{1/2}$ or $2^{1/4}$ we obtain the f.c.c., b.c.c. and m.c.c. lattices. The m.c.c. lattice also recently arose in a different context, as the lattice corresponding to the period matrix of the hyperelliptic Riemann surface $w^2 = z^8 - 1$

4 Dimensions 4–8

Table 1 summarizes what is presently known about the sphere packing and kissing number problems in dimensions ≤ 24. Entries enclosed inside a solid line are known to be optimal, those inside a dashed line optimal among lattices.

The large box in the 'density' column refers to Blichfeldt's 1935 result that the root lattices $\mathbb{Z} \simeq A_1$, A_2, $A_3 \simeq D_3$, D_4, D_5, E_6, E_7, E_8 achieve $\Delta_n^{(L)}$ for $n \le 8$. It is remarkable that more than 60 years later $\Delta_9^{(L)}$ is still unknown.

The large box in the right-hand column refers to Watson's 1963 result that the kissing numbers of the above lattices, together with that of the laminated lattice Λ_9, achieve $\tau_n^{(L)}$ for $n \le 9$. Odlyzko and I [16, Ch. 13] and independently Levenshtein determined τ_8 and τ_{24}. The packings achieving these two bounds are unique [16, Ch. 14].

Dim.	Densest packing	Highest kissing number	
1	$\mathbb{Z} \simeq \Lambda_1$	2	
2	$A_2 \simeq \Lambda_2$	6	
3	$A_3 \simeq D_3 \simeq \Lambda_3$	12	
4	$D_4 \simeq \Lambda_4$	24	
5	$D_5 \simeq \Lambda_5$	40	
6	$E_6 \simeq \Lambda_6$	72	
7	$E_7 \simeq \Lambda_7$	126	
8	$E_8 \simeq \Lambda_8$	240	
9	Λ_9	272	(306 from P_{9a})
10	Λ_{10} (P_{10c})	336	(500 from P_{10b})
12	K_{12}	756	(840 from P_{12a})
16	$BW_{16} \simeq \Lambda_{16}$	4320	
24	Leech $\simeq \Lambda_{24}$	196560	

Table 1: Densest packings and highest kissing numbers known in low dimensions. (Parenthesized entries are nonlattice arrangements that are better than any known lattice.)

THE 'LOW DIMENSIONAL LATTICES' PROJECT Some years ago Conway and I noticed that there were several places in the literature where the results could be simplified if they were described in terms of lattices rather than quadratic forms. (It seems clearer to say 'the lattice E_8' rather than 'the quadratic form $2x_1^2+2x_2^2+4x_3^2+4x_4^2+20x_5^2+12x_6^2+4x_7^2+2x_8^2+2x_1x_2+2x_2x_3+6x_3x_4+10x_4x_5+6x_5x_6+2x_6x_7+2x_7x_8$'.) This led to a series of papers [7], [10], [13].

Integral lattices of determinant $d = 1$ ('unimodular' lattices) have been classified in dimensions ≤ 25, dimensions 24, 25 being due to Borcherds. In [16, Ch. 15] and [7, (I)] we extended this to $d \leq 25$ for various ranges of dimension.

[7, (II)] is based on the work of Dade, Plesken, Pohst and others, and describes the lattices associated with the maximal irreducible subgroups of $GL(n, \mathbb{Z})$ for $n = 1, \ldots, 9, 11, 13, 17, 19, 23$. Nebe, and Nebe and Plesken (see [29], [32]) have recently completed the enumeration of the maximal finite irreducible subgroups of $GL(n, \mathbb{Q})$ for $n \leq 31$, together with the associated lattices.

[7, (IV)] gives an improved version of the mass formula for lattices, and [7, (V)] studies when an n-dimensional integral lattice can be represented as a sublattice of $\mathbb{Z}^m$ for some $m \geq n$, or failing that, by a sublattice of $s^{-1/2}\mathbb{Z}^m$ for some integer s. [10] describes the Voronoi and Delaunay cells of all the root lattices and their duals, and [7, (VI), (VIII)] discusses how the Voronoi cell of a 3- or 4-dimensional lattice changes as the lattice is continuously varied.

[7, (VII)] determines the 'coordination sequences' of various lattices. Consider E_8, for example, and let $S(k)$ denote the number of lattice points that are k steps from the origin, where a step is a move to an adjacent sphere ($S(1)$ is the kissing

number). Then $\sum_{k=0}^{\infty} S(k)x^k = f(x)/(1-x)^8$, where $f(x) = 1 + 232x + 7228x^2 + \ldots + x^8$. Thus the coordination sequence for E_8 begins $1, 240, 9120, \ldots$. For other examples see [39]

Perfect lattices One possible approach to the determination of the densest lattices in dimensions 7 to 9 is via Voronoi's theorem that the density of Λ is a local maximum if and only if Λ is perfect and eutactic [27].

In 1975 Stacey, extending the work of several earlier authors, published a list of 33 perfect lattices in dimension 7. Unfortunately one of the 33 was omitted from her papers and her dissertation. In [7, (III)] we reconstructed the missing lattice and 'beautified' all 33, computing their automorphism groups, etc. In 1991 Jaquet-Chiffelle [22] completed this work by showing that this is indeed the full list of perfect lattices in $\mathbb{R}^7$. This provides another proof that E_7 is the densest lattice in dimension 7.

Martinet, Bergé and their students are presently attempting to classify the eight-dimensional perfect lattices, and it appears that there will be roughly 10000 of them. Whether this approach can be used to determine $\Delta_9^{(L)}$ remains to be seen!

5 Dimension 9. Laminated lattices

There is a simple construction, the 'laminating' or 'greedy' construction, that produces many of the densest lattices in dimensions up to 26. Let Λ_1 denote the even integers in $\mathbb{R}^1$, and define the n-dimensional laminated lattices Λ_n recursively by: consider all lattices of minimal norm 4 that contain some Λ_{n-1} as a sublattice, and select those of greatest density. It had been known since the 1940's that this produces the densest lattices known for $n \leq 10$. In [6] we determined *all* inequivalent laminated lattices for $n \leq 25$, and found the density of Λ_n for $n \leq 48$ (Fig. 1). A key result needed for this was the determination of the covering radius of the Leech lattice and the enumeration of the deep holes in that lattice [16, Ch. 23].

What are all the best sphere packings in low dimensions? In [13] we describe what may be *all* the best packings in dimensions $n \leq 10$, where 'best' means both having the highest density and not permitting any local improvement. In particular, we conjecture that $\Delta_n^{(L)} = \Delta_n$ for $n \leq 9$. For example, it appears that the best five-dimensional sphere packings are parameterized by the 4-colorings of $\mathbb{Z}$. We also find what we believe to be the exact numbers of 'uniform' packings among these, those in which the automorphism group acts transitively. These assertions depend on certain plausible but as yet unproved postulates.

A remarkable property of 9-dimensional packings. We also show in [13] that the laminated lattice Λ_9 has the following astonishing property. Half the spheres can be moved bodily through arbitrarily large distances without overlapping the other half, only touching them at isolated instants, the density remaining

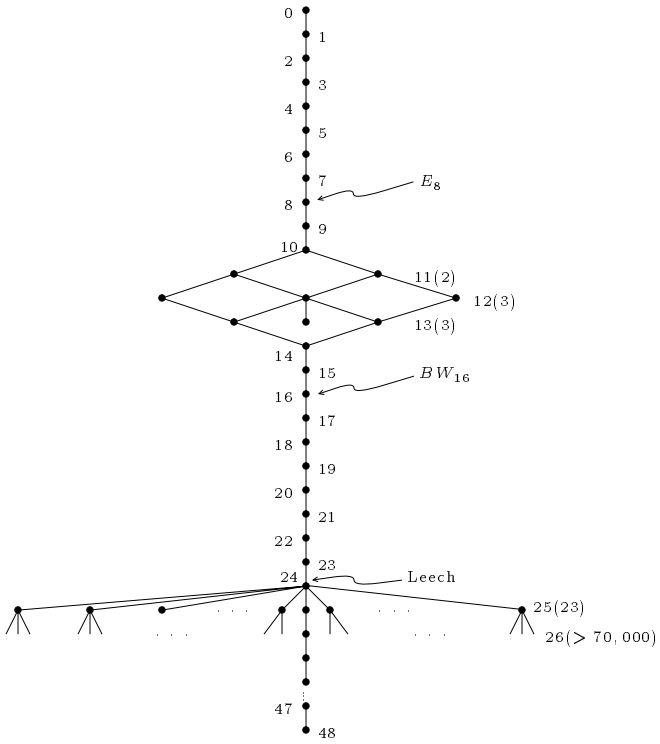


Figure 1: Inclusions among laminated lattices Λ_n.

the same at every instant. A typical packing in this family consists of the points of $D_9^{\theta+} = D_9 \cup D_9 + ((1/2)^8, \theta/2)$, for θ real. D_9^{0+} is Λ_9 and D_9^{1+} is D_9^+, the 9-dimensional diamond structure. All these packings have the same density, which we conjecture is the value of $\Delta_9 = \Delta_9^{(L)}$. Another result in [13] is that there are extraordinarily many 16-dimensional packings that are just as dense as the Barnes-Wall lattice $BW_{16} \simeq \Lambda_{16}$.

6 Dimension 10. Construction A.

In dimension 10 we encounter for the first time a nonlattice packing that is denser than all known lattices. This packing, and the nonlattice packing with the highest known kissing number in dimension 9, are easily obtained from 'Construction A' (cf. [24]). If $\mathcal{C}$ is a binary code of length n, the corresponding packing is $P(\mathcal{C}) = \{x \in \mathbb{Z}^n : x \pmod 2 \in \mathcal{C}\}$.

Consider the vectors $abcde \in (\mathbb{Z}/4\mathbb{Z})^5$ where $b, c, d \in \{+1, -1\}$, $a = c - d$, $e = b + c$, together with all their cyclic shifts, and apply the 'Gray map' $0 \to 00$, $1 \to 01$, $2 \to 11$, $3 \to 10$ to obtain a binary code $\mathcal{C}_{10}$ containing 40 vectors of length 10 and minimal distance 4. This is our description [12] of a code first

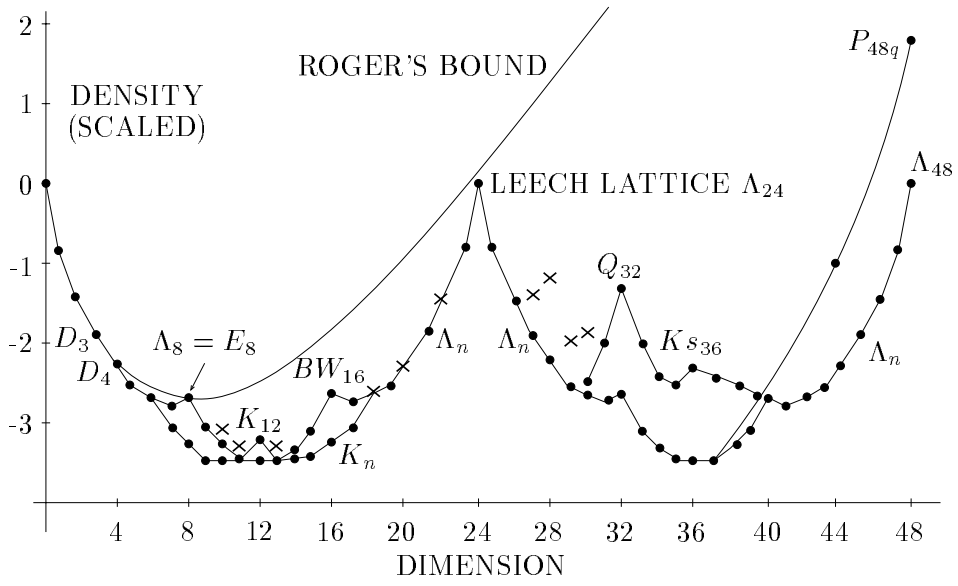


Figure 2: Densest sphere packings known in dimensions $n \leq 48$.

discovered by Best. The code is unique [25]. Then $P(\mathcal{C}_{10}) = P_{10c}$ is the record 10-dimensional packing.

Figure 2 shows the density of the best packings known up to dimension 48, rescaled to make them easier to read. The vertical axis gives $\log_2 \delta + n(24-n)/96$. The figure also shows the upper bounds of Muder (for $n = 3$) and Rogers ($n \geq 4$). Lattice packings are indicated by small circles, nonlattices by crosses (however, the locations of the lattices are only approximate). The figure is dominated by the two arcs of the graph of the laminated lattices Λ_n, which touch the zero ordinate at $n = 0, 24$ (the Leech lattice) and 48. K_{12} is the Coxeter-Todd lattice.

7 Dimensions 18–22

Record nonlattice packings in dimensions 18, 20 and 22 have recently been given in [4], [14], [40]. Vardy's construction [40], 'Construction B^*', also uses binary codes. Let $\mathcal{B}$ and $\mathcal{C}$ be codes of length n such that $c \cdot (\mathbf{1}+b) = 0$ for all $b \in \mathcal{B}$, $c \in \mathcal{C}$, and set $P^*(\mathcal{B}, \mathcal{C}) = \{\mathbf{0}+2b+4x, \mathbf{1}+2c+4y \ : \ b \in \mathcal{B}, c \in \mathcal{C}, x, y \in \mathbb{Z}^n, \sum x_i \text{ even}, \sum y_i \text{ odd}\}$. For example, by taking $\mathcal{B}$ to be the quadratic residue code of length 18 and $\mathcal{C}$ to be its dual, Bierbrauer and Edel [4] obtain a new record packing in $\mathbb{R}^{18}$.

8 Dimension 24. The Leech lattice

The Leech lattice Λ_{24} is a remarkably dense packing in $\mathbb{R}^{24}$ (as can be seen from Fig. 2). Here are four constructions. (i) As a laminated lattice: start in dimension 1 with the lattice $\Lambda_1 = \mathbb{Z}$ and apply the greedy algorithm (see Fig. 1). (ii) Apply

Construction A to the Golay code of length 24 to obtain a lattice L_{24}. Then Λ_{24} is spanned by $(-3/2, 1/2, \ldots, 1/2)$ and $\{x \in L_{24} : \sum x_i \equiv 0 \pmod 4\}$. (iii) Hensel lift the Golay code to an extended cyclic (and self-dual) code over $\mathbb{Z}/4\mathbb{Z}$ and apply 'Construction A mod 4' [5]. (iv) There is a unique unimodular even lattice $\mathrm{II}_{25,1}$ in Lorentzian space $\mathbb{R}^{25,1}$, consisting of the points $(x_0 x_1 \cdots x_{24}|x_{25})$ with all $x_i \in \mathbb{Z}$ or all $x_i \in \mathbb{Z}+1/2$ and satisfying $x_0+\cdots+x_{24}-x_{25} \in 2\mathbb{Z}$. Let $w = (0\ 1 \cdots 24|70)$, a vector of zero length. Then $(w^\perp$ in $\mathrm{II}_{25,1})/w$ is Λ_{24} [16, Ch. 26].

9 Dimensions 26–31

New packings in these dimensions have been discovered by Bacher, Borcherds, Conway, Vardy, Venkov — see [16] for details.

10 Dimension 32. Modular lattices

An *N-modular* lattice [34] is an integral lattice that is similar to its dual, under a similarity that multiplies norms by N. A unimodular lattice is 1-modular. The interest in this family arises because many of the densest known lattices are N-modular: $\mathbb{Z}$, A_2, D_4, E_8, K_{12}, BW_{16}, Λ_{24}, Q_{32}, P_{48q}, ...

Quebbemann's lattice Q_{32}, for example, is 2-modular, and can be constructed from a Reed-Solomon code of length 8 over $\mathbb{F}_9$ [33], [16, Ch. 8].

Shadow theory. The concept of the shadow of a lattice or code was introduced in [8], [9] (see also [15]) and has proved to be very useful ([9] has stimulated over 50 sequels in the coding literature).

Let Λ be an n-dimensional unimodular lattice. If Λ is even then the *shadow* $S(\Lambda) = \Lambda$, otherwise $S(\Lambda) = (\Lambda_0)^* \setminus \Lambda$, where the subscript 0 denotes even sublattice. The set $2S(\Lambda) = \{2s : s \in S(\Lambda)\}$ is precisely the set of *parity vectors* for Λ, i.e. the vectors $u \in \Lambda$ such that $u \cdot x \equiv x \cdot x \pmod 2$ for all $x \in \Lambda$. Such vectors have been studied by many authors from Braun (1940) onwards, but their application to obtaining bounds on lattices seems to have been overlooked.

If the theta series of Λ is $\Theta_\Lambda(z)$ then [8] the shadow has theta series

$$\left(\frac{e^{\pi i/4}}{\sqrt{z}}\right)^n \Theta_\Lambda\left(1-\frac{1}{z}\right) \ . \tag{1}$$

One of the most satisfying properties of integral lattices is the classical theorem that (a) if Λ is a unimodular lattice then Θ_Λ belongs to the graded ring $\mathbb{C}[\Theta_\mathbb{Z}, \Theta_{E_8}]$, and (b) if Λ is even then Θ belongs to $\mathbb{C}[\Theta_{E_8}, \Theta_{\Lambda_{24}}]$.

To illustrate the use of the shadow, let us prove there is no 9-dimensional unimodular lattice of minimal norm 2. If so then from (a) $\Theta_\Lambda = -\Theta_\mathbb{Z}/8 + 9\Theta_{E_8}/8 = 1 + 252q^2 + 456q^3 + \cdots$, where $q = e^{\pi i z}$. But then (1) implies $\Theta_{S(\Lambda)} = \frac{9}{4}q^{1/4} + \frac{1913}{4}q^{9/4} + \cdots$, a contradiction since $\Theta_{S(\Lambda)}$ must have integer coefficients.

In [26] we used (a), (b) to show that the minimal norm μ of an n-dimensional odd unimodular lattice satisfies

$$\mu \le \left[\frac{n}{8}\right] + 1 \ , \tag{2}$$

and for an even unimodular lattice

$$\mu \leq 2\left[\frac{n}{24}\right] + 2 \ . \tag{3}$$

In [36] we used shadow theory to strengthen (2) by showing that odd lattices satisfy

$$\mu \leq 2\left[\frac{n}{24}\right] + 2 \ , \tag{4}$$

except that $\mu \leq 3$ when $n = 23$. In view of the similarity between (3) and (4) we propose that a lattice satisfying either bound with equality be called *extremal* (the old definition of this term was based on (2) and (3)).

Quebbemann [35] has generalized (3) to certain families of even N-modular lattices, and analogous bounds for odd N-modular lattices (using an appropriate generalization of the shadow) were given in [36]. One can then define extremal N-modular lattices.

11 Higher dimensions

Space does not permit more than a mention of the following: Kschischang and Pasupathy's lattice Ks_{36} in $\mathbb{R}^{36}$ [23]; the three extremal unimodular lattices P_{48q}, P_{48p}, P_{48n} in $\mathbb{R}^{48}$, the latter being a recent discovery of Nebe [30]; Bachoc's extremal 2-modular lattice in $\mathbb{R}^{48}$ [1]; Nebe's extremal 3-modular lattice in $\mathbb{R}^{64}$ [30]; and Bachoc and Nebe's extremal unimodular lattice in $\mathbb{R}^{80}$ [2].

The existence of the following extremal lattices is an open question: 3-modular in $\mathbb{R}^{36}$ (determinant $d = 3^{18}$, minimal norm $\mu = 8$); 2-modular in $\mathbb{R}^{64}$ ($d = 2^{32}$, $\mu = 10$); unimodular in $\mathbb{R}^{72}$ ($d = 1$, $\mu = 8$).

From dimensions 80 to about 4096 the densest lattices known are the Mordell-Weil lattices discovered by Elkies [19], and Shioda [38]. But we know very little about this range, as evidenced by the recent construction of record kissing numbers in dimensions 32 to 128 [17] from binary codes. In dimension 128, for example, the Mordell-Weil lattice has kissing number 218044170240 [18], whereas in our construction (which admittedly is not a lattice) some spheres touch 8812505372416 others.

It would also be desirable to have better upper bounds, especially in low dimensions (see Fig. 2). The Kabatiansky-Levenshtein bound is asymptotically better than the Rogers' bound, but not until the dimension is above about 40. We know very little about these problems!

In short, many beautiful packings have been discovered, but there are few proofs that any of them are optimal.

References

[1] C. Bachoc, *Applications of coding theory to the construction of modular lattices*, J. Combin. Theory A 78 (1997), 92–119.

[2] C. Bachoc and G. Nebe, *Extremal lattices of minimum 8 related to the Mathieu group M_{22}*, J. reine angew. Math. 494 (1998), 155–171.

[3] A. Bezdek and W. Kuperberg, *Packing Euclidean space with congruent cylinders and with congruent ellipsoids*, in *Victor Klee Festschrift*, ed. P. Gritzmann et al., Amer. Math. Soc., 1991, pp. 71–80.

[4] J. Bierbrauer and Y. Edel, *Dense sphere packings from new codes*, preprint, 1998.

[5] A. Bonnecaze, A. R. Calderbank and P. Solé, *Quaternary quadratic residue codes and unimodular lattices*, IEEE Trans. Inform. Theory 41 (1995), 366–377.

[6] J. H. Conway and N. J. A. Sloane, *Laminated lattices*, Ann. Math. 116 (1982), 593–620.

[7] J. H. Conway and N. J. A. Sloane, *Low-dimensional lattices*: Proc. Royal Soc. Ser. A. I: 418 (1988), 17–41; II: 419 (1988), 29–68; III: 418 (1988), 43–80; IV: 419 (1988), 259–286; V: 426 (1989), 211–232; VI: 436 (1991), 55–68; VII: 453 (1997), 2369–2389; VIII (in preparation).

[8] J. H. Conway and N. J. A. Sloane, *A new upper bound for the minimum of an integral lattice of determinant one*, Bull. Am. Math. Soc. 23 (1990), 383–387; 24 (1991), 479.

[9] J. H. Conway and N. J. A. Sloane, *A new upper bound for the minimal distance of self-dual codes*, IEEE Trans. Inform. Theory 36 (1990), 1319–1333.

[10] J. H. Conway and N. J. A. Sloane, *The cell structures of certain lattices*, in *Miscellanea mathematica*, ed. P. Hilton et al., Springer-Verlag, NY, 1991, pp. 71–107.

[11] J. H. Conway and N. J. A. Sloane, *On lattices equivalent to their duals*, J. Number Theory 48 (1994), 373–382.

[12] J. H. Conway and N. J. A. Sloane, *Quaternary constructions for the binary single-error-correcting codes of Julin, Best and others*, Designs, Codes, Crypt. 4 (1994), 31–42.

[13] J. H. Conway and N. J. A. Sloane, *What are all the best sphere packings in low dimensions?*, Discrete Comput. Geom. 13 (1995), 383–403.

[14] J. H. Conway and N. J. A. Sloane, *The antipode construction for sphere packings*, Invent. math. 123 (1996), 309–313.

[15] J. H. Conway and N. J. A. Sloane, *A note on unimodular lattices*, J. Number Theory (to appear).

[16] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, NY, 3rd edition, 1998.

[17] Y. Edel, E. M. Rains and N. J. A. Sloane, *On kissing numbers in dimensions 32 to 128*, Electron. J. Combin. 5 (1) (1998), paper R22.

[18] N. D. Elkies, personal communication.

[19] N. D. Elkies, *Mordell-Weil lattices in characteristic 2: I. Construction and first properties*, Internat. Math. Res. Notices (No. 8, 1994), 353–361.

[20] T. C. Hales, *Sphere packings*, Discrete Comput. Geom. I: 17 (1997), 1–51; II: 18 (1997), 135–149; III: preprint.

[21] W.-Y. Hsiang, *On the sphere packing problem and the proof of Kepler's conjecture*, Internat. J. Math. 93 (1993), 739–831; but see the review by G. Fejes Tóth, Math. Review 95g #52032, 1995.

[22] D.-O. Jaquet-Chiffelle, *Enumération complète des classes de formes parfaites en dimension 7*, Ann. Inst. Fourier 43 (1993), 21–55.

[23] F. R. Kschischang and S. Pasupathy, *Some ternary and quaternary codes and associated sphere packings*, IEEE Trans. Inform. Theory 38 (1992) 227–246.

[24] J. Leech and N. J. A. Sloane, *Sphere packing and error-correcting codes*, Canad. J. Math. 23 (1971), 718–745.

[25] S. Litsyn and A. Vardy, *The uniqueness of the Best code*, IEEE Trans. Inform. Theory 40 (1994), 1693–1698.

[26] C. L. Mallows, A. M. Odlyzko and N. J. A. Sloane, *Upper bounds for modular forms, lattices and codes*, J. Alg. 36 (1975), 68–76.

[27] J. Martinet, *Les réseaux parfaits des espaces euclidiens*, Masson, Paris, 1996.

[28] D. J. Muder, *A new bound on the local density of sphere packings*, Discrete Comput. Geom. 10 (1993), 351–375.

[29] G. Nebe, *Finite subgroups of $GL_n(\mathbb{Q})$ for $25 \leq n \leq 31$*, Comm. Alg. 24 (1996), 2341–2397.

[30] G. Nebe, *Some cyclo-quaternionic lattices*, J. Alg. 199 (1998), 472–498.

[31] G. Nebe and N. J. A. Sloane, *A Catalogue of Lattices*, published electronically at http://www.research.att.com/~njas/lattices/.

[32] W. Plesken, *Finite rational matrix groups — a survey*, in *Proc. Conf. "The ATLAS: Ten Years After"*, to appear.

[33] H.-G. Quebbemann, *Lattices with theta-functions for $G(\sqrt{2})$ and linear codes*, J. Alg. 105 (1987), 443–450.

[34] H.-G. Quebbemann, *Modular lattices in Euclidean spaces*, J. Number Theory 54 (1995), 190–202.

[35] H.-G. Quebbemann, *Atkin-Lehner eigenforms and strongly modular lattices*, L'Enseign. Math. 43 (1997), 55–65.

[36] E. M. Rains and N. J. A. Sloane, *The shadow theory of modular and unimodular lattices*, J. Number Theory, to appear.

[37] C. E. Shannon, *A mathematical theory of communication*, Bell Syst. Tech. J. 27 (1948), 379–423 and 623–656.

[38] T. Shioda, *Mordell-Weil lattices and sphere packings*, Am. J. Math. 113 (1991), 931–948.

[39] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, published electronically at http://www.research.att.com/~njas/sequences/.

[40] A. Vardy, *A new sphere packing in 20 dimensions*, Invent. math. 121 (1995), 119–133.

N. J. A. Sloane
AT&T Labs-Research
180 Park Avenue
Florham Park NJ 07932-0971 USA
njas@research.att.com

Ulam Quaterly – Volume 3, Number 1, 1994

Applications of Modified Pell Numbers to Representations

A.F. Horadam

University of New England
Department of Mathematics
Armidale, Australia 2351

1 Background

Define the sequence $\{q_n\}$ for all integers n by the recurrence

$$q_{n+2} = 2q_{n+1} + q_n \qquad (q_0 = 1, q_1 = 1), \tag{1.1}$$

and the associated *Pell sequence* $\{P_n\}$ for all integers by the recurrence

$$P_{n+2} = 2P_{n+1} + P_n \qquad (P_0 = 0, P_1 = 1). \tag{1.2}$$

For a few basic relationships connecting P_n and q_n, see, for instance, [1] and [14].

Closely related to $\{q_n\}$ is the *Pell-Lucas sequence* $\{Q_n\}$ which has been extensively analyzed in a series of publications (e.g. [8], [11]), principally in relation to $\{P_n\}$, but also in its own right. In fact, $Q_n = 2q_n$. Consequently, the known properties of $\{Q_n\}$ are easily transferable to $\{q_n\}$.

Here, we are concerned only with $\{q_n\}$ and more especially, with the problem of representing any integer by sums of the numbers q_n.

For this purpose, we require the extension of the sequence $\{q_n\}$ to negative values of n. Inevitably, a study of $\{q_n\}$ involves familiarity with P_n. Using (1.1) and (1.2), we readily derive the following tabulation:

n	$\cdots$	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	$\cdots$
q_n	$\cdots$	99	-41	17	-7	3	-1	1	1	3	7	17	41	99	$\cdots$
P_n	$\cdots$	-70	29	-12	5	-2	1	0	1	2	5	12	29	70	$\cdots$

Fig. 1: Values of $q_n, P_n \quad (-6 \leq n \leq 6)$.

Clearly

$$q_{-n} = (-1)^n q_n \tag{1.3}$$

and

$$P_{-n} = (-1)^{n+1} P_n. \tag{1.4}$$

Also,

$$q_n = P_n + P_{n-1} = P_{n+1} - P_n = \frac{Pn + 1 + P_{n-1}}{2} \tag{1.5}$$

while

$$2P_n = q_n + q_{n-1} = q_{n+1} - q_n = \frac{q_{n+1} + q_{n-1}}{2}. \tag{1.6}$$

Explicit *Binet forms* for q_n and P_n are

$$q_n = \frac{\alpha^n + \beta^n}{2}, \tag{1.7}$$

and

$$P_n = \frac{\alpha^n + \beta^n}{\alpha - \beta} \tag{1.8}$$

where α,β are the roots of the characteristic equation $x^2 - 2x + 1 = 0$ of the recurrence relations (1.1) and 1.2), i.e.,

$$\begin{cases} \alpha = 1 + \sqrt{2} \\ \beta = 1 - \sqrt{2} \end{cases} \quad \text{so } \alpha + \beta = 2, \ \alpha - \beta = 2\sqrt{2} \ , \alpha\beta = -1. \tag{1.9}$$

With negative subscripts, (1.6) becomes

$$2P_{-n} = q_{-n} + q_{-n-1} = q_{-n+1} - q_{-n} = \frac{q_{-n+1} + q_{-n-1}}{2}. \tag{1.10}$$

The Name of the Sequence $\{q_n\}$ References to the numbers q_n in Sloane [13] are associated with Thébault [14] in 1949, and earlier in 1916 with an unspecified writer in [12]. Both P_n and q_n were designated *Eudoxus numbers* by Budden [2] in 1969, though I have no independent information of this claim.

Lucas [10] makes no specific reference to q_n so far as I am aware, but he does use the numbers $2q_n$ which he designates $V_n - a$ generic symbol of his, and couples them with P_n (his U_n) as *Suites de Pell.* Because of these historical origins, I have called $2q_n$, which I label Q_n, the *Pell-Lucas numbers.* (Perhaps, then, p_n might be named the "quasi Pell-Lucas numbers"?)

All things considered, I accept the nomenclature of Bruckman [1], who refers to q_n as *modified Pell numbers*, as suitably apt, and to this I have adhered.

2 Properties of $\{q_n\}$

Some properties of $\{q_n\}$ *per se* which are relevant to our study include the *Simson formula*

$$q_{n+1}q_{n-1} - q_n^2 = 2(-1)^{n+1} \tag{2.1}$$

and the summations

$$\sum_{i=1}^{n} q_{2i} = \frac{q_{2n+1} - 1}{2} \tag{2.2}$$

$$\sum_{i=1}^{n} q_{2i-1} = \frac{q_{2n} - 1}{2} \tag{2.3}$$

$$\sum_{i=1}^{n} q_i = \frac{q_{n+1} + q_n}{2} - 1 = P_{n+1} - 1 \quad \text{by (1.6)} \tag{2.4}$$

$$\sum_{i=0}^{n-1} (-1)^i q_{-i} = \frac{(-1)^n (q_{-n} - q_{-n+1})}{2} = (-1)^{n+1} P_{-n} \quad \text{by (1.10)} \tag{2.5}$$

$$\begin{aligned} q_n &= 2(q_{n-1} + q_{n-3} + \cdots + q_4 + q_2) + 1 \quad n \text{ odd } (q_0 = 1) \\ &= 2(q_{n-1} + q_{n-3} + \cdots + q_3 + q_1) + 1 \quad n \text{ even (c.f. (2.2), (2.3))} \end{aligned} \tag{2.6}$$

$$\sum_{i=1}^{n} P_i = \frac{-q_{n+1} - 1}{2} \tag{2.7}$$

Checking on the validity of these results is left to the vigilance of the reader. Moreover,

$$\sum_{i=0}^{n} q_{-2i} = \frac{-q_{-2n-1} + 1}{2} \tag{2.8}$$

$$\sum_{i=1}^{n} q_{-2i+1} = \frac{-q_{-2n} + 1}{2}. \tag{2.9}$$

Presence of 0 as the starting point in (2.8) is to be especially noted.

3 Representation of Positve Integers By $\{q_n\}$, $n \geq 0$

A. Minimal Representation

Theorem 3.1 *The representation of a positive integer $N > 0$ in the form*

$$N = \sum_{i=1}^{\infty} \alpha_i q_i \quad (\alpha_i = 0, 1 \text{ or } 2) \tag{3.1}$$

where

$$\alpha_i = 2 \Rightarrow \alpha_{i-1} = 0 \tag{3.2}$$

is unique *and* minimal.

Proof This is similar to that for $\{P_n\}$ in [7], with appropriate adjustments. E.g., use (2.6). Alternatively, consult the proof given in outline after Theorem 2.

By the phrase minimal representation we mean the representation with the least number of numbers q_i occurring in the sum (3.1), subject to the proviso (3.2). Such a representation may be called a *Zeckendorf representation* [7]. Figure 4 gives the minimal representations of N by $\{q_n\} : 1 \leq N \leq 50$. Absence of q_0 in (3.1) ought to be compared with the situation in (3.3) for Theorem 3.2.

A "Greedy" Algorithm. Remarks similar to those in [7] regarding a "greedy algorithm" for $\{P_n\}$ are also applicable in the case of $\{q_n\}$. As an illustration, from Figure 1 we have

$$350 - q_7 = 111,\ 111 - q_5 = 70,\ 70 - q_5 = 29,\ 29 - q_4 = 12,\ 12 - q_3 = 5,$$
$$5 - q_2 = 2,\ 2 - q_1 = 1,\ 1 - q_1 = 0, \text{ so that}$$
$$350 = q_7 + 2q_5 + q_4 + q_3 + q_2 + 2q_1.$$

B. Maximal Representation. By a *maximal representation*, we mean the greatest number of q_i occurring in the sum (3.3) below, in the context of the criteria (3.4).

Introduction of q_0. For maximality, we must introduce $q_0 = 1$. Otherwise 2, for example, could not be expressed maximally ($2 = 1.q_0 + 1.q_1$). Lucas numbers L_n similarly require the use of $L_0 = 2$ in the theory of maximal representations [15]. But, for uniqueness, we define $1 = q_i$. Furthermore, in the ensuing MinMax theory (section 4) we require $q_i = 1$ (not $q_0 = 1$) since q_0 is absent in considerations of minimality.

Pertinent to our usage is the fact that in section 4 $2q_0(= 2)$ is never used, only $1q_0$.

Thus, the special purpose of q_0 for maximality is to fill in the gap ($2 = 3 - 1$) between $q_1 = 1$ and $q_2 = 3$.

Theorem 3.2 *Every positive integer $N > 0$ has a unique representation in the form*

$$N = \sum_{i=0}^{m} \beta_i q_i \quad (\beta_i = 0, 1, \text{ or } 2) \tag{3.3}$$

where

$$\begin{cases} \beta_i = 0 & \Rightarrow \quad \beta_{i-1} = 2 \\ \beta_m = 1 & \text{or} \quad 2. \end{cases} \tag{3.4}$$

Proof of this Theorem has a number of lemma as a prologue:

First consider the sequence of coefficients of q_i in (3.3) of *length* $k \geq 1$, namely,

$$(\beta_0, \beta_1, \beta_2, \cdots, \beta_{k-1}) \tag{3.5}$$

subject to the criteria (3.4).

Write S_k	$\equiv$	the number of sequences (3.5) with (3.4) attached
r_k	$\equiv$	the range of values of N for S_k
N_k^{min}	$\equiv$	the minimum number of r_k
N_k^{max}	$\equiv$	the maximum number of r_k
I_k	$\equiv$	the number of integers in r_k.

Data relevant to these symbols for the number N in (3.3) are:

k	S_k	r_k	N_k^{min}	N_k^{max}	I_k
1	S_1	1			
2	S_2	$2, 3$	$2P_1$	$2P_2 - 1$	$2q_1$
3	S_3	$4, \cdots, 9$	$2P_2$	$2P_3 - 1$	$2q_2$
4	S_4	$10, \cdots, 23$	$2P_3$	$2P_4 - 1$	$2q_3$
5	S_5	$24, \cdots, 57$	$2P_4$	$2P_5 - 1$	$2q_4$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	S_k	$2P_{k-1} \cdots, 2P_k - 1$	$2P_{k-1}$	$2P_k - 1$	$2q_{k-1}$

Fig. 2: Basic Data for Theorem 2

Let us elaborate a little on this information.

Lemma 3.1 $2P_{k-1} \leq N \leq 2P_k - 1 \quad (k \geq 2)$.

Proof: For a sequence (3.5) of length $k \geq 2$, the maximum number N_k^{max} which it can represent is given by

$$\underbrace{(1,2,2,2,\cdots,2)}_{k \text{ digits}}$$

corresponding to

$$\begin{aligned} N_k^{max} &= 1.q_0 + 2\sum_{i=1}^{k-1} q_i \\ &= 1 + 2\left[\frac{q_k + q_{k-1}}{2} - 1\right] \quad \text{by (1.1), (2.4)} \\ &= q_k + q_{k-1} - 1 \\ &= 2P_k - 1 \qquad\qquad \text{by (1.6).} \end{aligned} \tag{3.6}$$

The minimum number N_k^{min} for a sequence of length k is obtained by the following reasoning:

After the number in (3.6), the next number in order is $(2P_k - 1) + 1 = 2P_k$ which occurs as the first number in the (next) sequence of length $k+1$. Accordingly, the minimum number in the sequence of length k is derived from $2P_k$ by replacing k by $k-1$, i.e., $N_k^{min} = 2P_{k-1}$. Hence

$$\underbrace{2P_{k-1} \leq N leq 2P_k - 1}_{r_k} \quad (k \geq 2).$$

Corollary 3.1 *When $k-1$, we are left merely with the number 1.*

Lemma 3.2 $S_k = 2q_{k-1} \quad (k \geq 2)$.

Proof From Lemma 3.1, the number of numbers I_k included in the range r_k, which is the same as the number of sequences S_k, is

$$\begin{aligned} S_k &= (2P_k - 1) - \{(2Pp_{k-1} - 1\} = I_k \\ &= 2(P_k - P_{k-1}) \\ &= 2q_{k-1} \qquad\qquad \text{by (1.5).} \end{aligned}$$

Lemma 3.3 *k is uniquely determined by $N(\beta_k \neq 0)$.*

This is obvious from Figure 2. As an example, consider

$$\begin{aligned} N &= 1000 \\ &\Rightarrow 816 \leq 1000 \leq 1969 \quad (= 1970 - 1) \\ &\Rightarrow 2P_8 \leq 1000 \leq 2P_9 - 1 \\ &\Rightarrow k = 9 \end{aligned}$$

(with $r_9 = 1969 - 815 = (1970 - 1) - (816 - 1) = 1154 = 2q_s$).

Lemma 3.4 *β_k ($\neq 0$) is uniquely determined by N.*

This is clearly so, since, from (3.3), $N - \beta_k q_k$ is a specific number.

For instance, $N = 50 \Rightarrow \begin{cases} N - 2q_4 &= 1 + 2 + 6 + 7 &= 16 & (\beta_4 = 2) \\ N - q_4 &= 1 + 2 + 6 + 7 + 17 &= 33 & (\beta_4 = 1). \end{cases}$

Proof of Theorem 3.1 Assembling all the evidence which has followed the enunciation of Theorem 3.1 (namely, Figure 2 and Lemmas 3.1-3.4), we are led to accept the validity of the Theorem.

Observation We remark that the numbers N_k^{max} are identical with the *subsidiary MinMax numbers* N_{k-1} ($k \geq 1$) for Pell numbers [5]. The reason for this is that, by (3.6), $N_k^{max} = 2P_k - 1 = N_{k-1}$ by [5].

Alternative Proof (Outline) of Theorem 3.1 This may be set out to parallel the treatment in Theorem 3.1 by using similar techniques. Firstly, consider the sequence of length k in (3.7)

$$(\alpha_1, \alpha_2, \cdots, \alpha_k).$$

The minimum number represented by this sequence is given by

$$\underbrace{(0, 0, 0, \cdots, 1)}_{k \text{ digits}}$$

i.e., q_k.

The maximum number is given by

$$(2, 0, 2, 0, \cdots, 2, 0) \text{ or } (0, 2, 0, 2, \cdots 0, 2)$$

depending on the parity of k, i.e., $q_{2k+1} - 1$ or $q_{2k} - 1$ from (2.2) and (2.3) respectively. Eventually, we may assert (replacing S_k in Lemma 3.2 by s_k):

Lemma 3.5 $q_k \leq N \leq q_{k+1} - 1$

Lemma 3.6 $s_k = 2P_k$.

Lemma 3.7 *k is uniquely determined by N ($\alpha_k \neq 0$).*

Lemma 3.8 α_k $(\neq 0)$ *is uniquely determined by* N.

Gathering together these results, we establish the validity of Theorem 3.1.

Remarks

(i) The ranges in the above treatment for Theorem 3.1 are:
$1,2;\ 3,\cdots,6;\ 7,\cdots,16;\ 17,\cdots,40;\ \cdots$.

(ii) Figure 5 gives representations of N for $1 \leq N \leq 50$.

(iii) In Figure 4, 2 is always preceded by 0 (except in the first column).
In Figure 5, 0 is always preceded by 2 (except in the last column).

That is, there is a type of **duality** in the enunciations of Theorems 3.1 and 3.2 -cf. the criteria (3.2) and (3.4). In a similar context for Fibonacci numbers, Brown [9] refers to a "Dual Zeckendorf Theorem".

4 The MinMax Sequence $\{Q_n\}$.

Comparing the data in Figures 4 and 5 we discern that, for certain values of N, the minimal and maximal representations are identical. These may be designated as the *MinMax numbers* for $\{q_n\}$. But what are these numbers? Inspection of Figures 4 and 5, buttressed by an argument paralleling that used in [5] relating to $\{P_n\}$, establishes that the *MinMax sequence* $\{Q_n\}$ consists of those numbers whose representations (3.1) and (3.3) have coefficients α_i and β_i of q_i which are all unity, i.e. for which $\alpha_i = \beta_i = 1$, where $i = 1, 2, 3, \cdots$.

Write

$$Q_n = \sum_{i=1}^{n} q_i. \tag{4.1}$$

Take

$$Q_0 = 0. \tag{4.2}$$

Then by (2.4) and (1.8),

$$Q_n = P_{n+1} - 1 \tag{4.3}$$

that is,

$$Q_n = \frac{\alpha^{n+1} - \beta^{n+1}}{\alpha - \beta} - 1 \quad (\textit{Binet} \text{ form}) \tag{4.4}$$

Assembling this information in order, we find that the first few members of the MinMax sequence $\{Q_n\}$ for $\{q_n\}$ are:

n	1	2	3	4	5	6	7	8	9	10	$\cdots$
Q_n	1	4	11	28	69	168	407	984	2377	5740	$\cdots$

(4.5)

Using (4.3) with (1.2), or (4.4), one discovers the recursion

$$Q_{n+2} = 2Q_n + Q_n + 2, \tag{4.6}$$

the *Simson formula* analogue for $\{Q_n\}$

$$Q_{n+1}Q_{n-1} - Q_n^2 = (-1)^{n+1} - 2P_n, \tag{4.7}$$

and

$$\sum_{i=1}^{n} Q_i = \frac{q_{n+2} - 1}{2} - (n+1). \tag{4.8}$$

Generating function for $\{Q_n\}$ is

$$(1+x)(1-3x+x^2+x^3)^{-1} = \sum_{n=1}^{\infty} Q_n x^{n-1}. \tag{4.9}$$

Other relationships of interest include

$$Q_n - Q_{n-1} = q_n \tag{4.10}$$

$$Q_n + Q_{n+1} = q_{n+2} - 2 \tag{4.11}$$

$$Q_n - Q_{n-2} = 2P_n \tag{4.12}$$

$$Q_n + Q_{n+2} = 2q_{n+2} - 2 \tag{4.13}$$

$$Q_n^2 - Q_{n-1}^2 = q_n(q_{n+1} - 2) \tag{4.14}$$

$$Q_n^2 - Q_{n-2}^2 = 4P_n(q_n - 1) \tag{4.15}$$

$$Q_n^2 + Q_{n+1}^2 = P_{2n+3} - 2q_{n+2} + 2. \tag{4.16}$$

Discoveries of further properties of $\{Q_n\}$, e.g. divisibility properties, may be unearthed *ad infinitum, ad nauseam* according to one's fortitude and motivation.

Worthy of recording is the following relationship, where $\{N_n\}$ is the subsidiary MinMax sequence of $\{M_n\}$ - see [5] -:

$$\begin{aligned} 2Q_n &= 2P_{n+1} - 2 \qquad \text{from (4.3)} \\ &= (2P_{n+1} - 1) - 1 \\ &= N_n - 1 \qquad \text{from [5]} \end{aligned}$$

5 The Subsidiary MinMax Sequence $\{R_n\}$

Suppose we introduce the *subsidiary MinMax sequence* $\{R_n\}$ of $\{Q_n\}$ defined recursively by

$$R_n = Q_{n+1} + Q_{n-1} \quad (R_0 = 0, Q_{-1} = -1). \tag{5.1}$$

Values of R_n are, from (4.5) thus:

$$\begin{array}{c|cccccccccc} n & = & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \cdots \\ \hline R_n & = & 4 & 12 & 32 & 80 & 196 & 476 & 1152 & 2784 & \cdots \\ & = & 4(1 & 3 & 8 & 20 & 49 & 119 & 288 & 696 & \cdots) \\ & = & 4M_n. & & & & & & & & \end{array} \tag{5.2}$$

where $\{M_n\}$ is the *MinMax sequence* for $\{P_n\}$ examined in [5].

No undue surprise should emanate from this fact, for, with the notation and results of [5],

$$\begin{aligned} R_n &= (M_{n+1} + M_n) + (M_{n-1} + M_{n-2}) = Q_{n+1} + Q_{n-1} \\ &= (2M_n + M_{n-1} + 1) + M_n + M_{n-1} + (M_n - 2M_{n-1} - 1) \\ &= 4M_n \end{aligned}$$

Properties of $\{M\}$ enumerated in [5] may accordingly be transferred to $\{R_n\}$, provided the appropriate modifications are made. Thus, for instance, we have the recurrence relation

$$R_{n+2} = 2R_{n+1} + R_n + 4 \quad (R_0 = 0), \tag{5.3}$$

$$4(1 - 3x + x^2 + x^3)^{-1} = \sum_{n=1}^{\infty} R_n x^{n-1} \quad \text{(generating function)}, \tag{5.4}$$

$$R_{n+1}R_{n-1} - R_n^2 = 8((-1)^n - q_n) \quad \text{(Simson's formula)}, \tag{5.5}$$

and

$$R_n = \alpha^{n+1} + \beta^{n+1} - 2 \quad \text{(Binetform)}. \tag{5.6}$$

Divisibility attributes of $\{M_n\}$ mentioned in [5] automatically carry over to $\{R_n\}$. Of course, the quality of primeness is absent.

Neither the sequence $\{R_n\}$ nor the sequence $\{q_n\}$ is listed in [13].

References to many seminal contributions to representations involving Fibonacci and Lucas numbers (e.g. these by Zeckendorf and Lekkerkerker) are to be found in [5].

6 Negatively Subscripted Q_n.

Though it is not meaningful in the context of representations to extend $\{Q_n\}$ through negative values of n, let us nonetheless complete the mathematical theory by considering the numbers Q_{-n}, $n > 0$. Imagine that the recursive statement (4.6) is applied to negative subscripts. Then the following table results:

n	$\cdots$	9	8	7	6	5	4	3	2	1	0	1	$\cdots$
Q_{-n}	$\cdots$	-409	170	-71	28	-13	4	-3	0	-1	0	1	$\cdots$

(6.7)

Inherent in (6.1) is the recursion

$$Q_{-n+2} = 2Q_{-n+1} + Q_{-n} + 2. \tag{6.8}$$

Many other characteristic features of $\{Q_{-n}\}$ are deducible, e.g. (cf. (4.3)),

$$Q_{-n} = P_{-n+1} - 1. \tag{6.9}$$

Replacing n by $-n$ in (4.4), (4.7), and (4.9) readily produces expressions for the Binet form, Simson's formula, and the generating function, respectively.

Similar avenues for development exist in the case of R_{-n}, $n > 0$.

Each of Q_{-n} and R_{-n}, where $n > 0$, opens up fertile new territory for exploration.

However, we must restrict over freedom of choice to our stated goal: the representations of the integers by q_n, where n may be positive or negative.

7 Representation of Any Integer by $\{q_{-n}\}$, $n > 0$.

To demonstrate the truth of Theorem 3.1 below, two options are available to us, namely,

(i) to follow the techniques for Fibonacci numbers used in [3], and

(ii) to modify the proof in [3] to suit our purposes.

Initially, (i) was attempted but its procedures seemed too intricate to pursue. Possibly it is still amenable to mathematical discipline. However, the treatment defined in (ii) appeared quicker and generally more desirable (cf. [6], [4]).

Heavy reliance will be placed on (2.5) in what follows.

But first we need to note the following comments.

Representation of Zero (0) Evidently the integer 0 is not representable by $\{q_n\}$ since no q_i, where $i = 0, 1, 2, \cdots$ is zero. To represent 0, we need to introduce a negatively subscripted q_n, via q_{-1}:

$$0 = 1.q_{-1} + 1.q_0. \tag{7.1}$$

Theorem 7.1 *The representation of any integer N as*

$$N = \sum_{i=0}^{\infty} a_i q_{-i} \quad (a_i = 0, 1, \textbf{or } 2) \tag{7.2}$$

where

$$a_i = 2 \Rightarrow a_{i+1} = 0, \tag{7.3}$$

is unique and minimal.

Proof Suppose there are two different representations

$$N = \sum_{i=0}^{h} a_i q_{-i} \quad a_k \neq 0, a_i = 2 \Rightarrow a_{i+1} = 0 \tag{7.4}$$

$$N = \sum_{i=0}^{m} b_i q_{-i} \quad b_m \neq 0, b_i = 2 \Rightarrow b_{i+1} = 0 \tag{7.5}$$

Case I Assume $h = m$. Conceivably, the numbers in (7.3) and (7.4) are the same but their coeffiecients a_i, b_i are generally different.

Write

$$c_i = a_i - b_i \quad (c_i = 1, \pm 1, \pm 2;\ i = 0, 1, 2, \cdots, m). \tag{7.6}$$

Subtract (7.4) from (7.3). After simplification using (7.5), we derive

$$c_m q_{-m} + \sum_{i=0}^{m-1} c_i q_{-i} = 0 \quad (m \geq 1). \tag{7.7}$$

Employing (2.5), we see that for a maximum or a minimum sum (7.6) i.e., $c_i = \pm 2$ where $i = 0, 1, 2, \cdots, m-1$, we must have

$$c_m q_{-m} + (-1)^m (q_{-m} - q{-m} + 1) = 0 \quad (m \geq 1) \tag{7.8}$$

in which the notation of (1.10) may alternatively be used. Concentrate now on $c_m q_{-m}$ because this term reigns supreme over the sums (7.6) and (7.7).

m **even** ($q_{-m} > 0$) Equation (7.7) now yields

$$(c_m + 1)q_{-m} - q_{-m+1} = 0 \quad (m \geq 2). \tag{7.9}$$

So, with $m \geq 2$,

$$\begin{array}{rclcrcl} c_m & = & 0 & \Rightarrow & q_{-m} & = & q_{-m+1} \\ c_m & = & 1 & \Rightarrow & q_{-m-1} & = & 0 \\ c_m & = & 2 & \Rightarrow & q_{-m} & = & q_{-m-1}. \end{array}$$

For m odd ($q_{-m} < 0$: Under these circumstances (7.7) becomes

$$(c_m - 1)q_{-m} + q_{-m+1} = 0 \quad (m \geq 1). \tag{7.10}$$

Then, for $m \geq 1$,

$$\begin{array}{rclcrcl} c_m & = & 0 & \Rightarrow & q_{-m} & = & q_{-m+1} \text{ as before} \\ c_m & = & 1 & \Rightarrow & q_{-m+1} & = & 0 \\ c_m & = & 2 & \Rightarrow & q_{-m} & = & -q_{-m+1}. \end{array}$$

None of these can possibly be valid, as a little checking discloses. Similar reasoning can be applied for $c_m = -1, -2$. Consequently, the assumption in Case I is untrue.

Summary of Case I conclusions If $h = m$, then $a_i = b_i$ where $i = 0, 1, 2, \cdots, m$. That is, (7.3) and (7.4) are identical, so the representation (7.1) with (7.2) is unique.

Case II Assume $h > m$. Consider the set of coefficients of q_{-i} of length $k + 1$ in (7.1), namely,

$$(a_0, a_1, a_2, \cdots, a_{k-1}, a_k). \tag{7.11}$$

For a minimum sum, we must have the arrangement

$$(0, 2, 0, 2, 0, 2, \cdots, 0, 2) \tag{7.12}$$

while for a maximum sum we have

$$(2, 0, 2, 0, 2, 0, \cdots, 2, 0) \tag{7.13}$$

Now replace the symbolism used in Figure 2 by the corresponding asterisked symbolism, e.g., r_k is replaced by r_k^*. Then the appropriate data may be

tabulated in this manner (cf. Figure 2 and the discussion germane to it) with the aid of (2.8) and (2.9):

k	r_k^*	N_k^{*min}	N_k^{*max}
$\left\{\begin{array}{l}0\\1\end{array}\right.$	$\left\{\begin{array}{l}\\-2,\cdots,2\end{array}\right.$	$-q_{-2}+1=-2$	$-q_{-1}+1=2$
$\left\{\begin{array}{l}1\\2\end{array}\right.$	$\left\{\begin{array}{l}\\-16,\cdots,8\end{array}\right.$	$-q_{-4}+1=-16$	$-q_{-3}+1=8$
$\left\{\begin{array}{l}2\\3\end{array}\right.$	$\left\{\begin{array}{l}\\-98,\cdots,42\end{array}\right.$	$-q_{-6}+1=-98$	$-q_{-7}+1=240$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\left\{\begin{array}{l}m\\m+1\end{array}\right.$	$\left\{\begin{array}{l}\\-x,\cdots,y\end{array}\right.$	$-q_{-2(m+1)}+1=-x$	$-q_{-2m-1}+1=y$

k	I_k^*
$\left\{\begin{array}{l}0\\1\end{array}\right.$	$2P_2+1=5$
$\left\{\begin{array}{l}1\\2\end{array}\right.$	$2P_4+1=25$
$\left\{\begin{array}{l}2\\3\end{array}\right.$	$2P_8+1=817$
$\vdots$	$\vdots$
$\left\{\begin{array}{l}m\\m+1\end{array}\right.$	$2P_{2(m+1)}+1=x+y+1$

Basic Data for Theorem 7.1

Appealing to (2.8) and (2.9), we may check this information thus:

$$\begin{aligned} I_k^* &= N_k^{*max}-N_k^{*min}+1 \quad \text{for the zero representation} \\ &= (-q_{-2m-1}+1)-(-q_{-2(m+1)}+1)+1 \\ &= q_{-2m-2}-q_{-2m-1}+1 \\ &= 2P_{2(m+1)}+1 \quad \text{by (1.4), (1.10).} \end{aligned}$$

Evidently, each number N, as it occurs for the first time in Figure 3, is represented uniquely and minimally, E.g.,

$$-10 = (1q_0 + 0q_{-1} + 1q_{-2} + 2q_{-3}) + 0q_{-4} + 0q_{-5} + 0q_{-6} + \cdots$$

has a unique minimal representation $1q_0 + 1q_{-2} + 2q_{-3}$, i.e., the sequence $(1, 1, 0, 2)$. We conclude that h not greater than m and similarly that h not less than m . Consequently, $h = m$. Hence, Case I and the Summary, are true.

Collecting together all the arguments above, we agree that the validity of the Theorem has been established. (Consult Figure 6 and 7 for details of the numerical mechanism of Theorem 7.1)

There can be no maximal representation of a number by means of negatively subscripted q_n. Arguments for this salient feature are analogous to those used in [6] for negatively subscripted P_n. Having asserted this, we may now mentally review our attainments. These confirm that our stated objectives have indeed been achieved.

References

[1] P.S. Bruckman. Solution to Advanced Problem H-361. *The Fibonacci Quarterly*, 23(1):95–96, 1985.

[2] F.J. Budden. *An Introduction to Number Scales and Computers*. Longmans, 1965.

[3] M.W. Bunder. Zeckendorf Representations Using Negative Fibonacci Numbers. *The Fibonacci Quarterly*, 30(2):111–115, 1992.

[4] A.F. Horadam. An Alternative Proof of a Unique Representation Theorem. Submitted.

[5] A.F. Horadam. MinMax Sequences for Pell Numbers . Submitted.

[6] A.F. Horadam. Unique Minimal representation of Integers by Negatively Subscripted Pell Numbers. In Press.

[7] A.F. Horadam. Zeckendorf Representations of Positive and Negative Integers by Pell Numbers. In Press.

[8] A.F. Horadam and Br. J.M. Mahon. Pell and Pell-Lucas Polynomials. *The Fibonacci Quarterly*, 23(1):7–20, 1985.

[9] J.L. Brown Jr. A New Characterization of the Fibonacci Numbers. *The Fibonacci Quarterly*, 3(1):1–8, 1965.

[10] E. Lucas. *Théorie des Nombres.* Blanchard, 1961.

[11] Br. J.M. Mahon. *M.A.(Hons.) Thesis.* PhD thesis, The University of New England, Armidale, Australia, 1984.

[12] W.J. Miller. Mathematical questions and solutions. *Educational Times*, 1:9, 1916.

[13] N.J.A. Sloane. *A Handbook of Integer Sequences.* Academic Press, 1973.

[14] V. Thébault. Concerning two classes of remarkable perfect square pairs. *American Mathematical Monthly*, 56:443–448, 1949.

[15] V.E. Hoggatt, Jr. *Fibonacci and Lucas Numbers.* Houghton Mifflin, 1969.

N^+	q_1	q_2	q_3	q_4	q_5	N^+	q_1	q_2	q_3	q_4	q_5
1	1					26	2		1	1	
2	2					27		1	1	1	
3		1				28	1	1	1	1	
4	1	1				29	2	1	1	1	
5	2	1				30		2	1	1	
6		2				31			2	1	
7			1			32	1		2	1	
8	1		1			33	2		2	1	
9	2		1			34				2	
10		1	1			35	1			2	
11	1	1	1			36	2			2	
12	2	1	1			37		1		2	
13		2	1			38	1	1		2	
14			2			39	2	1		2	
15	1		2			40		2		2	
16	2		2			41					1
17				1		42	1				1
18	1			1		43	2				1
19	2			1		44		1			1
20		1		1		45	1	1			1
21	1	1		1		46	2	1			1
22	2	1		1		47		2			1
23		2		1		48			1		1
24			1	1		49	1		1		1
25	1		1	1		50	2		1		1

Fig. 3: Minimal Representations of Positive Integers by Sums of the Numbers q_i where $i = 1, 2, 3....$

N^+	q_0	q_1	q_2	q_3	q_4	N^+	q_0	q_1	q_2	q_3	q_4
1		1				26	1	2	2		1
2	1	1				27	1	2		1	1
3	1	2				28		1	1	1	1
4		1	1			29	1	1	1	1	1
5	1	1	1			30	1	2	1	1	1
6	1	2	1			31		1	2	1	1
7		1	2			32	1	1	2	1	1
8	1	1	2			33	1	2	2	1	1
9	1	2	2			34	1	2		2	1
10	1	2		1		35		1	1	2	1
11		1	1	1		36	1	1	1	2	1
12	1	1	1	1		37	1	2	1	2	1
13	1	2	1	1		38		1	2	2	1
14		1	2	1		39	1	1	2	2	1
15	1	1	2	1		40	1	2	2	2	1
16	1	2	2	1		41		1	2		2
17	1	2		2		42	1	1	2		2
18		1	1	2		43	1	2	2		2
19	1	1	1	2		44	1	2		1	2
20	1	2	1	2		45		1	1	1	2
21		1	2	2		46	1	1	1	1	2
22	1	1	2	2		47	1	2	1	1	2
23	1	2	2	2		48		1	2	1	2
24		1	2		1	49	1	1	2	1	2
25	1	1	2		1	50	1	2	2	1	2

Fig. 4: Maximal Representations of Positive Intergers by Sums of the Numbers q_i, where $i = 0, -1, -2, ...$

0	1	1													
N^+	q_0	q_{-1}	q_{-2}	q_{-3}	q_{-4}	q_{-5}	q_{-6}	N^+	q_0	q_{-1}	q_{-2}	q_{-3}	q_{-4}	q_{-5}	q_{-6}
1	1							26		1		1	2		
2	2							27				1	2		
3		1						28	1			1	2		
4	1		1					29	2			1	2		
5	2		1					30			1	1	2		
6			2					31	1		1	1	2		
7	1		2					32		2			2		
8	2		2					33		1			2		
9		1		1	1			34					2		
10				1	1			35	1				2		
11	1		1	1				36	2				2		
12	2		1	1				37			1		2		
13			1	1	1			38	1		1		2		
14	1		1	1	1			39	2		1		2		
15		2			1			40			2		2		
16		1			1			41	1		2		2		
17					1			42	2		2		2		
18	1				1			43		1		2		1	1
19	2				1			44				2		1	1
20			1		1			45	1			2		1	1
21	1		1		1			46	2			2		1	1
22	2		1		1			47			1	2		1	1
23			2		1			48	1		1	2		1	1
24	1		2		1			49	2		1	2		1	1
25	2		2		1			50			2	2		1	1

Fig. 5: Minimal Representations of Positive Intergers by Sums of the Numbers q_i, where $i = 0, -1, -2, ...$

0	1	1											
N^+	q_0	q_{-1}	q_{-2}	q_{-3}	q_{-4}	q_{-5}	N^+	q_0	q_{-1}	q_{-2}	q_{-3}	q_{-4}	q_{-5}
-1		1					-26		2			1	1
-2		2					-27	1		1	1	1	1
-3	1		1	1			-28			1	1	1	1
-4			1	1			-29	2			1	1	1
-5	2			1			-30	1			1	1	1
-6	1			1			-31				1	1	1
-7				1			-32		1		1	1	1
-8		1		1			-33		2		1	1	1
-9		2		1			-34	1		2			1
-10	1		1	2			-35			2			1
-11			1	2			-36	2		1			1
-12	2			2			-37	1		1			1
-13	1			2			-38			1			1
-14				2			-39	2					1
-15		1		2			-40	1					1
-16		2		2			-41						1
-17	1		2		1	1	-42		1				1
-18			2		1	1	-43		2				1
-19	2		1		1	1	-44	1		1	1		1
-20	1		1		1	1	-45			1	1		1
-21			1		1	1	-46	2			1		1
-22	2				1	1	-47	1			1		1
-23	1				1	1	-48				1		1
-24					1	1	-49		1		1		1
-25		1			1	1	-50		2		1		1

Fig. 6: Minimal Representations of Negative Integers by Sums of the Numbers q_i, where $i = 0, -1, -2, ...$

Hipparchus, Plutarch, Schröder and Hough

Richard P. Stanley

1. Hipparchus and Plutarch. Plutarch was a Greek biographer and philosopher from Chaeronea, who was born before A.D. 50 and died after A.D. 120. He is best known for his *Parallel Lives*, which inspired such Renaissance writers as Montaigne, Shakespeare, Dryden, and Rousseau. His many other works have been gathered together under the name *Moralia*, "a collection of comparatively short treatises and dialogues which cover an immense range of subjects, literary, ethical, political, and scientific" [21, p. 8]. Part of the *Moralia* consists of the *Table-Talk*, "a collection of dialogues purporting to reproduce the after-dinner conversation of Plutarch and his friends and relatives on various occasions" [20, p. 2]. In the *Table-Talk* [20, VIII.9, 732] appears the following statement:

> Chrysippus says that the number of compound propositions that can be made from only ten simple propositions exceeds a million. (Hipparchus, to be sure, refuted this by showing that on the affirmative side there are 103,049 compound statements, and on the negative side 310,952.)

Chrysippus (*c.* 280–207 B.C.) came to Athens around 260 and became a leading Stoic philosopher. Hipparchus was a Greek astronomer (*c.* 190–after 127 B.C.) from Nicaea in Bithynia (now Iznik, Turkey) who spent much of his life at Rhodes. He was perhaps the greatest astronomer of antiquity. He is most famous for his discovery of the precession of the equinoxes, based on his own observations and those of Timocharis 160 years earlier. For further information on the work of Hipparchus, see [19, Book I, E][32]. Hipparchus was an excellent mathematician (though for a contrary view see [33, p. 211]); he was the first person to make systematic use of trigonometry, and he was probably the inventor of stereographic projection. However, for many centuries no one was able to make sense of the statement of Plutarch. For instance, T. L. Heath [12, vol. 2, p. 256], a standard older authority on Greek mathematics, says of Plutarch's statement that "it seems impossible to make anything of these figures," while the more recent authority O. Neugebauer [19, p. 338]

states that Plutarch's statement "[has], however, so far eluded a satisfactory explanation." Similarly W. and M. Kneale [16, p. 162], authorities on the history of logic, remark that "It is difficult to make any satisfactory sense of the passage." N. L. Biggs [2, p. 113] notes the paucity of combinatorial computations by the ancient Greeks and referring to Plutarch's passage says that "the most interesting of them is also the most mysterious." A number of eminent mathematicians and historians of mathematics, such as M. Cantor, J. Tropfke, S. Günther, and E. Artin, have attempted to understand Plutarch's statement without success. An attempt to reconstruct Hipparchus' procedure appears in [1], though it will be apparent from our discussion that this attempt is incorrect. Another incorrect speculation appears in [30, p. 63].

2. Schröder. Friedrich Wilhelm Karl Ernst Schröder was a German logician who was born in Mannheim on November 25, 1841, and died in Karlsruhe on June 16, 1902. He passed the doctoral exam at the University of Heidelberg in 1862 and had positions in Zurich (at the Eidgenössische Polytechnikum), Karlsruhe, Pforzheim, and Baden-Baden, before accepting a post as full professor at Karlsruhe in 1876. Schröder worked mainly on the foundations of mathematics, notably with combinatorics, the theory of functions of a real variable, and mathematical logic. He was one of the first persons to accept Cantor's ideas in set theory and was one of the developers of mathematical logic in the second half of the nineteenth century. Schröder is best known to combinatorialists for his paper [25], in which he discusses four "bracketing problems." The first two problems concern the bracketing or parenthesization of a string of letters that we may assume to be all identical, say the letter x. The second two problems are analogues of the first two where the string of letters is replaced by a set of elements. We will discuss only the first two problems here.

The formal definition of a bracketing is the following. First, x itself is considered to be a bracketing. Recursively define a bracketing to be a sequence $B = (B_1, \ldots, B_k)$, where $k \geq 2$ and each B_i is a bracketing. We represent the bracketing B as a parenthesized string of x's. Thus, think of B as a k-ary product $(B_1)(B_2)\cdots(B_k)$. If some B_i is the single letter x, then we remove the parentheses surrounding B_i for clarity of notation. Thus, for example, the bracketing

$$(xx)((xxxxx)x(xx))(xx(xx)) \tag{1}$$

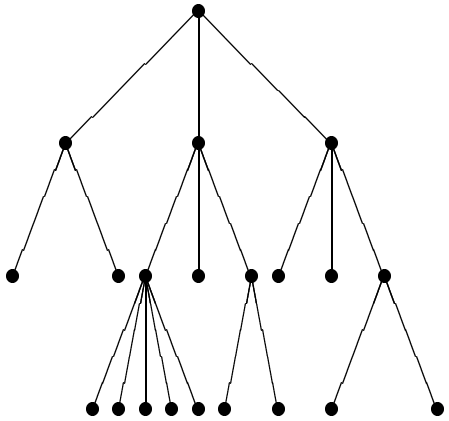


Figure 1: A plane tree.

represents a way of multiplying 14 x's whose last operation was a ternary operation $(B_1)(B_2)(B_3)$, where $B_1 = xx$, $B_2 = (xxxxx)x(xx)$, and $B_3 = xx(xx)$, and similarly for B_1, B_2, and B_3. There are exactly eleven bracketings of four letters, namely,

$$xxxx \quad (xx)xx \quad x(xx)x \quad xx(xx) \quad (xxx)x \quad x(xxx)$$
$$((xx)x)x \quad (x(xx))x \quad (xx)(xx) \quad x((xx)x) \quad x(x(xx)).$$

Note that the last five of these are built up entirely from *binary* operations and are therefore called *binary bracketings*.

There are three fundamental equivalent ways to represent a bracketing in addition to a parenthesized string discussed above: as *plane trees*, *polygon dissections*, and *Lukasiewicz words*. We now briefly describe these alternative representations. If B is a bracketing, then we first define the plane tree $\tau(B)$ corresponding to B. If B consists of a single letter, then $\tau(B)$ is a single root vertex. If $B = (B_1, \ldots, B_k)$ then $\tau(B)$ consists of a root vertex (drawn at the top), with subtrees $\tau(B_1), \ldots, \tau(B_k)$, drawn in that order from left to right. Thus, the key property defining a plane tree is that the subtrees of every vertex are linearly ordered. For instance, the plane tree corresponding to the bracketing of equation (1) is shown in Figure 1. Note that a binary bracketing corresponds to a *binary plane tree*, i.e., a plane tree for which every non-endpoint vertex has exactly two successors.

Next we consider polygon dissections. Let P be a convex polygon. A

dissection of P is obtained by drawing some diagonals that don't intersect in their interiors. Thus, P is divided up into regions that are themselves convex polygons. In particular, if P has m sides and we draw $m-3$ such diagonals (the maximum number possible), then we obtain a dissection for which every region is a triangle; such dissections are called *triangulations*. We now explain how to associate a plane tree $\tau(D)$ with a polygon dissection D. We associate with the "degenerate" polygon with just two vertices a single root vertex. Now fix once and for all an edge e of the polygon P, called the *root edge*. In a given dissecton D, the edge e is contained in a unique polygon Q which is a region of D. Let $k+1$ be the number of edges of Q. If we remove the edge e and the interior of Q from D, then we are left with dissections $D_1, D_2, \ldots, D_k$ of k polygons (some possibly with just two vertices), reading counterclockwise from e along the boundary of Q, such that D_i and D_{i+1} intersect at a single vertex for $1 \leq i \leq k-1$. Define recursively $\tau(D)$ to be the plane tree whose subtrees of the root are $\tau(D_1), \ldots, \tau(D_k)$ in that order. Note that if P has $n+1$ vertices, then $\tau(D)$ has n endpoints. Figure 2 shows the polygon dissection corresponding to the tree of Figure 1.

Finally we consider Łukasiewicz words. The letters of such words come from the alphabet $A = \{x_0, x_1, x_2, \ldots\}$. The *weight* $\delta(x_i)$ of a letter x_i is defined by $\delta(x_i) = i-1$. A word $y_1 y_2 \cdots y_m$ made of letters from A is said to be a *Łukasiewicz word* if $\delta(y_1) + \cdots + \delta(y_j) \geq 0$ for $1 \leq j \leq m-1$, and $\delta(y_1) + \cdots + \delta(y_m) = -1$. Thus, $y_m = x_0$. The set of all Łukasiewicz words is called the *Łukasiewicz language* [17, Ch. 11.3]. To obtain a Łukasiewicz word $\omega(\tau)$ from a plane tree τ, do a depth-first (preorder) search through the tree. By definition, this is a linear ordering $\delta(\tau) = v_1, v_2, \ldots, v_p$ of the vertex set of τ defined recursively by $\delta(\tau) = v, \delta(\tau_1), \ldots, \delta(\tau_k)$, where v is the root of τ, and $\tau_1, \ldots, \tau_k$ are the subtrees of v (in that order). Define

$$\omega(\tau) = x_{\deg(v_1)} x_{\deg(v_2)} \cdots x_{\deg(v_k)},$$

where $\deg(v_i)$ denotes the degree (number of successors or children) of vertex v_i. For instance, the Łukasiewicz word corresponding to the plane tree of Figure 1 is

$$x_3 x_2 x_0^2 x_3 x_5 x_0^6 x_2 x_0^2 x_3 x_0^2 x_2 x_0^2.$$

Note that since our bracketings B do not allow unary operations, the plane tree $\tau(B)$ has no vertices of degree one, and the corresponding Łukasiewicz word does not involve the letter x_1.

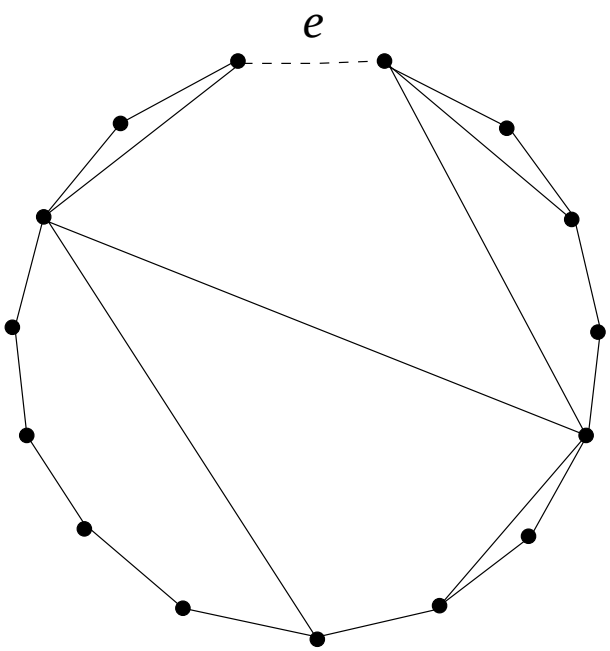


Figure 2: A polygon dissection.

The correspondences established above are easily seen to yield the following result.

Proposition. (a) *Let $s(n)$ denote the total number of bracketings of a string of n letters. Then $s(n)$ is also equal to (i) the number of plane trees with no vertex of degree one and with n endpoints, (ii) the number of dissections of a convex $(n+1)$-gon, and (iii) the number of Łukasiewicz words with no x_1's and with n x_0's.*

(b) *Let $b(n)$ denote the number of* binary *bracketings of a string of n letters. Then $b(n)$ is also equal to (i) the number of binary plane trees with n endpoints (and hence with $2n-1$ vertices), (ii) the number of triangulations of a convex $(n+1)$-gon, and (iii) the number of Łukasiewicz words with n x_0's and $n-1$ x_2's (and with no other letters); such words, usually with the last x_0 deleted, are sometimes called* Dyck words.

We are now ready to explain the contribution of Schröder to these bracketing problems. Schröder's first problem asks for the number $b(n)$ of binary bracketings of a string of n letters. Using a generating function argument, Schröder derives the formula (stated slightly differently)

$$b(n) = \frac{1}{n}\binom{2n-2}{n-1}.$$

Thus $b(n)$ is just the *Catalan number* C_{n-1}, for which an enormous literature exists. For some further information and references, see [11],[14]. A list of about fifty combinatorial interpretations of Catalan numbers will appear in [31, Exercise 6.17] and is available on the World Wide Web at http://www-math.mit.edu/~rstan/ec/ec.html.

Schröder's second problem asks for the total number $s(n)$ of bracketings of a string of n letters. Schröder's main result on his second problem is the generating function

$$\sum_{n\geq 1} s(n)x^n = \frac{1}{4}\left(1+x-\sqrt{1-6x+x^2}\right). \tag{2}$$

He also gives the values (with the typographical error 145 for $s(5) = 45$)

$$(s(1),\ldots,s(10)) = (1,1,3,11,45,197,903,4279,20793,103049). \tag{3}$$

Perhaps the quickest way to obtain equation (2) is the following. Let y denote the left-hand side. The recursive definition of bracketing is equivalent to the formula

$$y = x + y^2 + y^3 + y^4 + \cdots = x + \frac{y^2}{1-y}. \tag{4}$$

Multiplying by $1 - y$ yields the quadratic equation

$$2y^2 - (1+x)y + x = 0. \tag{5}$$

One of the solutions is spurious, and the other one is just the right-hand side of (2).

The numbers $s(n)$ are now called *Schröder numbers*. Schröder does not mention any other combinatorial interpretations of Schröder numbers, nor does he give a single outside reference. Let us point out some additional references. The problem of counting the triangulations of a convex polygon was raised by Segner [26] and solved (anonymously) by Euler [9]. The connection between bracketings and plane trees was known to Cayley [4]. The bijection between plane trees and polygon dissections appears in Etherington [8], with a sequel by Erdélyi and Etherington in [7]. The bijection between bracketings and Łukasiewicz works is essentially the "reverse Polish notation" or "parenthesis-free notation" developed by the Polish logician Jan Łukasiewicz (1878–1956). He came upon the idea of this notation in 1924 and first published it in 1929, as explained in [18, p. 180, footnote 3]. The connection between reverse Polish notation and enumerative combinatorics appears in a pioneering paper of George Raney [22].

There is now a considerable literature on Schröder numbers and related numbers. To get into this literature, see [3][15, p. 55][23][27][34]. Let us also mention that it is easy to obtain a simple recurrence relation [5][6, p. 57] for the Schröder numbers which allows them to be computed rapidly. Namely, differentiate (5) with respect to x and solve for y' to obtain

$$y' = \frac{y-1}{4y-1-x} = \frac{(x-3)y - x + 1}{x^2 - 6x + 1},$$

the latter equality a consequence of the quadratic equation (5). Hence

$$(x^2 - 6x + 1)y' - (x-3)y + x - 1 = 0.$$

Expanding the left-hand side in a power series in x and setting the coefficient of x^n equal to 0 yields

$$(n+2)s(n+2) - 3(2n+1)s(n+1) + (n-1)s(n) = 0, \quad n \geq 1. \tag{6}$$

No direct combinatorial proof of this formula was known until D. Foata and D. Zeilberger, after reading an earlier version of this paper, found such a proof [10].

3. Hough. The stage is now set for the *dénouement*. The astute reader may have already anticipated it by comparing Plutarch's cryptic statement with the values (3) of the Schröder numbers. In January 1994 David Hough (1949–), a graduate student at George Washington University (who decided only in 1992 that he would pursue a career in mathematics), noticed that the mysterious number 103,049 of Plutarch, i.e., the number of compound propositions that can be formed from ten simple propositions, is just the tenth Schröder number! Hough learned about Plutarch's statement from [30, Exercise 1.45]. Hough's discovery strongly suggests that Hipparchus was carrying out a calculation equivalent to the modern calculation of the number of bracketings of a string of ten letters. However, it remains to determine exactly what Hipparchus and Plutarch meant by a "compound proposition." In Stoic logic, compound propositions are built up from simple ones using such connectives as "and," "or," and "if ... then" [16, Ch. III.5]. This does not seem like enough information to pinpoint precisely what Hipparchus had in mind.

We can also ask how Hipparchus computed the number 103,049. As noted in [24, p. 101], this number is much too large to have been computed by a direct enumeration of all the cases. Moreover, it is highly unlikely that Hipparchus was aware of the sophisticated recurrence (6). More probable is that Hipparchus used the "obvious" recurrence (equivalent to equation (4))

$$s(n) = \sum_{i_1+\cdots+i_k=n} s(i_1)\cdots s(i_k), \quad n \geq 2, \tag{7}$$

where the sum ranges over all ways to write n as an (ordered) sum of $k \geq 2$ positive integers. The sum on the right-hand side of equation (7) in the case $n = 10$ has 511 terms. There are only 41 "essentially different" terms, corresponding to the 41 partitions of 10 into a least two parts, i.e., the 41

ways to write 10 as an *unordered* sum of at least two positive integers. If the terms of the sum are grouped according to the partition of 10 to which they correspond, it is still necessary to count the number of ways of ordering each partition. For instance, the partition $3+2+2+1+1+1$ has 60 orderings of its terms, thus contributing the amount $60s(3)s(2)^2s(1)^3$ to the sum (7). We cannot but admire Hipparchus' ability to compute the Schröder number $s(10)$ at a distant time when not even a remotely similar accurate computation is known. For further information about combinatorics in ancient times, see [2],[24].

The number 310,952 in Plutarch's statement, i.e., the number of compound propositions that can be formed from ten simple propositions "on the negative side," remains an enigma. Many possible variants of plane trees have been looked at without success. Moreover, Neil Sloane has verified that the numbers 310,952 and 103,049 + 310,952 = 414,001 do not appear anywhere in the valuable tables [28]. Thus the mystery of Plutarch's statement remains at most half solved.

ACKNOWLEDGMENT. The research was partially supported by NSF grant DMS-9500714. I am grateful to Judith Grabiner, Wilbur Knorr, and four anonymous referees for providing invaluable suggestions and references.

References

[1] K.-R. Biermann and J. Mau, Überprüfung einer frühen Anwendung der Kombinatorik in der Logik, *J. Symbolic Logic* **23** (1958), 129–132.

[2] N. L. Biggs, The roots of combinatorics, *Historia Mathematica* **6** (1979), 109–136.

[3] J. Bonin, L. W. Shapiro, and R. Simion, Some q-analogues of the Schröder numbers arising from combinatorial statistics on lattice paths, *J. Stat. Planning and Inference* **34** (1993), 35–55.

[4] A. Cayley, On the analytical form called trees, Part II, *Philos. Mag. (4)* **18** (1859), 374–378.

[5] L. Comtet, Calcul pratique des coefficients de Taylor d'une fonction algébrique, *Enseignement Math.* **10** (1964), 267–270.

[6] L. Comtet, *Advanced Combinatorics*, Reidel, Dordrecht/Boston, 1974.

[7] A. Erdélyi and I. M. H. Etherington, Some problems of non-associative combinations (2), *Edinburgh Math. Notes* **32** (1940), 7–12.

[8] I. M. H. Etherington, Some problems of non-associative combinations (1), *Edinburgh Math. Notes* **32** (1940), 1–6.

[9] L. Euler, Summarium, *Novi Commentarii academiae scientiarum Petropolitanae* **7** (1758/59), 13-15. Reprinted in *Opera Omnia (1)* **26** (1953), xvi–xviii.

[10] D. Foata and D. Zeilberger, A classic proof of a recurrence for a very classical sequence, preprint. Available from the website http://www.math.temple.edu/~zeilberg/mamarim/mamarimhtml/classic.html.

[11] M. Gardner, Catalan numbers, Mathematical Games, *Scientific American* **234** (June, 1976), pp. 120–125, and bibliography on p. 132. Reprinted (with an Addendum) in Chapter 20 of *Time Travel and Other Mathematical Bewilderments*, W. H. Freeman, New York, 1988.

[12] T. L. Heath, *A History of Greek Mathematics*, Oxford University Press, Oxford; 1921; reprinted by Dover, New York, 1981.

[13] T. L. Heath, *A Manual of Greek Mathematics*, Oxford University Press, Oxford, 1931; reprinted by Dover, New York, 1963.

[14] P. Hilton and J. Pedersen, Catalan numbers, their generalizations, and their uses, *Math. Intelligencer* **13** (Spring, 1991), 64–75.

[15] M. Klazar, On *abab*-free and *abba*-free set partitions, *Europ. J. Combinatorics* **17** (1996), 53–68.

[16] W. Kneale and M. Kneale, *The Development of Logic*, Oxford University Press, Oxford, 1962, 1971.

[17] M. Lothaire, *Combinatorics on Words*, Encyclopedia of Mathematics and Its Applications, vol. 17, Addison-Wesley, Reading, Massachusetts, 1983.

[18] J. Łukasiewicz, *Selected Works* (L. Borkowski, ed.), North-Holland, Amsterdam, 1970.

[19] O. Neugebauer, *A History of Ancient Mathematical Astronomy*, vol. 1, Springer-Verlag, Berlin/Heidelberg/New York, 1975.

[20] Plutarch, *Moralia*, vol. IX (introduction by E. L. Minar, Jr.), Loeb Classical Library, Harvard University Press, Cambridge, Massachusetts, 1961.

[21] Plutarch, *The Rise and Fall of Athens: Nine Greek Lives*, translated by I. Scott-Kilvert, Penguin, London, 1960.

[22] G. N. Raney, Functional composition patterns and power series reversion, *Trans. Amer. Math. Soc.* **94** (1960), 441–451.

[23] D. G. Rogers and L. W. Shapiro, Deques, trees and lattice paths, in *Combinatorial Mathematics VIII* (K. L. McAvaney, ed.), Lecture Notes in Mathematics, no. 884, Springer-Verlag, Berlin, pp. 293–303.

[24] A. Rome, Procédés anciens de calcul des combinaisons, *Ann. Soc. sci. Bruxelles*, ser. A **50** (1930), 97–104.

[25] E. Schröder, Vier combinatorische Probleme, *Z. für Math. Physik* **15** (1870), 361–376.

[26] A. de Segner, Enumeratio modorum, quibus figurae planae rectilineae per diagonales dividuntur in triangula, *Novi Commentarii academiae scientiarum Petropolitanae* **7** (1758/59), 203–209.

[27] L. W. Shapiro and A. B. Stephens, Bootstrap percolation, the Schröder numbers, and the n-kings problem, *SIAM J. Discrete Math.* **4** (1991), 275–280.

[28] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, 1995.

[29] R. Stanley, Differentiably finite power series, *European J. Combinatorics* **1** (1980), 175–188.

[30] R. Stanley, *Enumerative Combinatorics*, vol. 1, Wadsworth & Brooks/Cole, Belmont, CA, 1986; second printing, Cambridge University Press, 1996.

[31] R. Stanley, *Enumerative Combinatorics*, vol. 2, Cambridge University Press, Cambridge, in preparation.

[32] G. J. Toomer, Hipparchus, in *Dictionary of Scientific Biography* (C. C. Gillipsie, editor in chief), vol. XV, supplement I, Schribner's, New York, 1978, pp. 207–224.

[33] B. L. van der Waerden, *Geometry and Algebra in Ancient Civilizations*, Springer-Verlag, Berlin, 1983.

[34] J. West, Generating trees and the Catalan and Schröder numbers, *Discrete Math.* **146** (1995), 247–262.

Department of Mathematics 2-375
Massachusetts Institute of Technology
Cambridge, MA 02139
e-mail: rstan@math.mit.edu

Annals of Combinatorics **2** (1998) 197-210

Annals of Combinatorics

Heap Games, Numeration Systems and Sequences

Aviezri S. Fraenkel

Department of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel
fraenkel@wisdom.weizmann.ac.il http://www.wisdom.weizmann.ac.il/~fraenkel

Received June 3, 1998

AMS Subject Classification: 90D46, 05A99

Abstract. We propose and analyze a 2-parameter family of 2-player games on two heaps of tokens, and present a strategy based on a class of sequences. The strategy looks easy, but it is actually hard. A class of exotic numeration systems is then used, which enables us to decide whether the family has an efficient strategy or not. We introduce yet another class of sequences and demonstrate its equivalence with the class of sequences defined for the strategy of our games.

Keywords: heap games, numeration systems, sequences

1. Example

Given a 2-player game played on two heaps (piles) of finitely many tokens. There are two types of moves:

(I) Take any positive number of tokens from *one* heap, possibly the entire heap.

(II) Take from *both* heaps, k from one and l from the other, with, say, $k \le l$. Then the move is constrained by the condition $0 < k \le l < 2k+2$, which is equivalent to $0 \le l-k < k+2,\ k > 0$. The player making the last move (after which both heaps are empty) wins, and the opponent loses.

A position q in a game of this sort is called a *P*-position if the *P*revious player can win, i.e., the player who moved to q. It is an *N*-position if the *N*ext player can win, i.e., the player moving from q. The position $(0,0)$ (two empty heaps) is a *P*-position, since the first player cannot even make a move, so the second wins by default. The next *P*-position is $(1,4)$: if Jean takes an entire heap, then Gill takes the other and wins. If Jean takes any part of the larger heap, Gill can take the balance of both heaps. Lastly, Jean cannot remove both heaps, and if she takes from both heaps, then Gill takes the balance and wins.

Table 1 lists the first few *P*-positions. The reader will do well to try and construct the next few entries of the table before reading on.

Table 1: The first few P-positions.

n	A_n	B_n
0	0	0
1	1	4
2	2	8
3	3	12
4	5	18
5	6	22
6	7	26
7	9	32
8	10	36
9	11	40
10	13	46
11	14	50
12	15	54
13	16	58

If S is any finite subset of nonnegative integers, denote by mex S the least nonnegative integer in the complement of S, i.e., the least nonnegative integer not occurring in S. Note that the mex of the empty set is 0. The term mex, used in [1], stands for Minimum EXcluded value. The structure of Table 1 is made explicit by:

$$A_n = \text{mex}\{A_i, B_i : i < n\},\ B_n = 2(A_n + n) \quad (n \geq 0).$$

This is a special case of Theorem 2.1 below, in the proof of which we also see that if $A = \bigcup_{n=1}^{\infty} A_n$, $B = \bigcup_{n=1}^{\infty} B_n$, then A and B are *complementary*, i.e., $A \cup B =$ set of all positive integers, and $A \cap B = \emptyset$.

Given any two heaps of our game, containing x and y tokens with $x \leq y$. The complementarity of A and B implies that either $x = A_n$ or $x = B_n$ for some n. Hence, Table 1 has to be computed only up to the encounter of x. Moreover, it is not hard to see that $n \leq x$, and if $x = A_n$, then $x/2 < n$, so the table has to be computed up to at most $\Omega(x)$, which implies a strategy computation linear in x, which looks good.

The trouble with this strategy is the same as that of the simple-minded primality-testing algorithm for a given integer m: divide m by the integers $\leq \sqrt{m}$, and if none of them divides m, then m is prime. This algorithm is linear in m. Of course, the problem in both cases is that the input length is the log of the input numbers x, y and m, rather than x, y and m themselves.

The two algorithms mentioned above, that for the strategy computation and for primality testing, are thus actually exponential in the input length.

The central question we address here is whether games of the type considered above have a polynomial strategy, or whether their best strategies are necessarily exponential.

Before that we define precisely the family of games in Sect. 2, introduce a family of sequences, and formulate and prove the winning strategy in terms of these sequences.

In Sect. 3, we present an argument against polynomiality of the games, and in Sect. 4, we introduce a numeration system that turns out to be relevant to our games. The connection between the games and the numeration system is made explicit in Sect. 5. This enables us to decide the games' polynomiality question in Sect. 6. Yet another class of sequences is introduced in Sect. 7, where we prove equivalence between the two classes of sequences. In Sect. 8, we summarize our results, give motivation and present a few open problems.

2. A Family of Heap Games and Their Winning Strategies

Denote by $\mathbb{Z}^0$ and $\mathbb{Z}^+$ the set of nonnegative integers, and positive integers, respectively. Our family of heap games depends on two parameters $s,t \in \mathbb{Z}^+$. Given are two heaps of finitely many tokens. There are two types of moves:

(I) Take any positive number of tokens from a single heap, possibly the entire heap.

(II) Take $k > 0$ and $l > 0$ from the two heaps, say, $0 < k \le l$. This move is constrained by the condition

$$0 < k \le l < sk + t, \tag{2.1}$$

which is equivalent to $0 \le l - k < (s-1)k + t, \quad k \in \mathbb{Z}^+$.

The example presented in Sect. 1 is the special case $s = t = 2$. Denote by P the set of all P-positions.

Theorem 2.1. $P = \bigcup_{i=0}^{\infty}\{(A_i, B_i)\}$, *where*

$$A_n = \operatorname{mex}\{A_i, B_i : 0 \le i < n\},\ B_n = sA_n + tn \quad (n \in \mathbb{Z}^0). \tag{2.2}$$

Proof. Let $A = \bigcup_{n=1}^{\infty} A_n$, $B = \bigcup_{n=1}^{\infty} B_n$. Then A, B are complementary with respect to $\mathbb{Z}^+$: $A \cup B = \mathbb{Z}^+$ follows from the mex property. Suppose $A_m = B_n$. Then $m > n$ implies that A_m is the mex of a set containing $B_n = A_m$, a contradiction. If $m \le n$, then $B_n = sA_n + tn \ge sA_m + tm > A_m$, another contradiction. Thus, $A \cap B = \emptyset$. We will also need the fact that A_n and B_n are strictly increasing sequences, which is clear from their definition.

Let $W = \bigcup_{i=0}^{\infty}(A_i, B_i)$. It evidently suffices to show two things:

I. A player moving from some $(A_n, B_n) \in W$ lands in a position not in W.

II. Given any position (x,y) not in W, there is a move to some $(A_n, B_n) \in W$.

I. A move of the first type from $(A_n, B_n) \in W$ clearly leads to a position not in W, since A_n and B_n are strictly increasing, so they have no repeating terms. Suppose a move of the second type from $(A_n, B_n) \in W$ produces a position $(A_m, B_m) \in W$. Then $m < n$. For $k = A_n - A_m$, $l = B_n - B_m$, we have

$$l = sA_n + tn - sA_m - tm = s(A_n - A_m) + t(n - m) \ge sk + t,$$

which contradicts condition (2.1).

II. Let (x,y) with $x \le y$ be a position not in W. Since A and B are complementary, every positive integer appears exactly once in exactly one of A and B. Therefore, we have either $x = B_n$ or else $x = A_n$ for some $n \ge 0$.

Case (i). $x = B_n$. Then move $y \to A_n$.
Case (ii). $x = A_n$. If $y > B_n$, then move $y \to B_n$. So suppose $A_n \le y < B_n$. If $y < sA_n + t$, move $(x,y) \to (0,0)$, which satisfies (2.1) with $k = A_n$, $l = y$. So let $y \ge sA_n + t$. Put $m = \lfloor (y - sA_n)/t \rfloor$ and move $(x,y) \to (A_m, B_m)$, where $\lfloor x \rfloor$ denotes the largest integer $\le x$. This move is legal, since (a) $m < n$, (b) $y > B_m$, (c) $A_n - A_m \le y - B_n < s(A_n - A_m) + t$. Indeed,

(a) $y - sA_n < B_n - sA_n = tn$, so $m = \lfloor (y - sA_n)/t \rfloor \le (y - sA_n)/t < n$;
(b) $m \le (y - sA_n)/t$, so $y \ge tm + sA_n = B_m + s(A_n - A_m) > B_m$;
(c) $m > ((y - sA_n)/t) - 1$, so $y < tm + sA_n + t$; by (b), $y - B_m \ge A_n - A_m$,

hence,

$$A_n - A_m \le y - B_m < tm + sA_n + t - sA_m - tm = s(A_n - A_m) + t,$$

and (2.1) is satisfied. ∎

The *statement* of Theorem 2.1 enables one to decide whether any given position (x,y) is a P-position or N-position, and the *proof* clearly indicates a winning move from any N-position. These two parts together constitute a winning strategy for the game.

However, as was pointed out in Sect. 1 after Table 1, the strategy is exponential (the inequalities for x hold for all $s,t \in \mathbb{Z}^+$, not just for the special example considered there). But only the construction of the table needs exponential time and, in fact, exponential space. The rest of the algorithm is polynomial. A winning strategy is polynomial only if both of its parts are polynomial. Our central question is whether there is a polynomial strategy for this game.

3. An Argument Against Polynomiality

Suppose there exist real numbers α and β such that for A_n and B_n defined in Theorem 2.1, $A_n = \lfloor n\alpha \rfloor$ and $B_n = \lfloor n\beta \rfloor$ for all $n \in \mathbb{Z}^0$. A simple density argument then shows that α and β must satisfy $\alpha^{-1} + \beta^{-1} = 1$, hence, $1 < \alpha < 2 < \beta$ and α, β are in fact irrational.

A strategy based on this observation can be applied to any given game position (x,y). Since $\alpha > 1$,

$$x = \lfloor n\alpha \rfloor \iff x < n\alpha < x+1 \iff \frac{x}{\alpha} < n < \frac{x+1}{\alpha}$$
$$\iff n = \left\lfloor \frac{x+1}{\alpha} \right\rfloor = \left\lfloor \frac{x}{\alpha} \right\rfloor + 1.$$

Therefore, either $x = \lfloor n\alpha \rfloor = A_n$ where $n = \lfloor (x+1)/\alpha \rfloor$, or else, by complementarity, $x = \lfloor n\beta \rfloor = B_n$, where $n = \lfloor (x+1)/\beta \rfloor$. We have thus reduced the situation to that

considered in cases (i) and (ii) in the proof of Theorem 2.1, and hence, the strategy presented in that proof can be followed. For example, if $x = \lfloor n\alpha \rfloor = A_n$ and $s\lfloor n\alpha \rfloor + t \le y < s\lfloor n\alpha \rfloor + tn = \lfloor n\beta \rfloor$, then for $m = \lfloor (y - s\lfloor n\alpha \rfloor)/t \rfloor$, we move $(x,y) \to (\lfloor m\alpha \rfloor, \lfloor m\beta \rfloor) \in P$. For implementing this strategy, α has to be computed to a precision of $O(\log x)$ digits, and its storage requires $O(\log x)$ words, which is linear in the input size of x (given in binary, say). Thus, this strategy *is* polynomial. See also the remark at the end of the previous section.

Is it far-fetched to hope for the existence of such real numbers α and β? For the special case $s = t = 1$, our games are reduced to Wythoff's game, for which such real numbers indeed exist, namely, $\alpha = (1+\sqrt{5})/2$, $\beta = (3+\sqrt{5})/2$. This was already shown in [13] (see also [5, 14]). In [6], a generalization of Wythoff's game was proposed, namely, the case of any $t \in \mathbb{Z}^+$, but $s = 1$. Also for this case these numbers exist, namely, $\alpha = (2 - t + \sqrt{t^2+4})/2$, $\beta = \alpha + t$.

We now show, however, that for $s > 1$, such real numbers cannot exist!

Theorem 3.1. *For A_n, B_n as defined in Theorem 2.1, there exist real numbers α, γ, β, δ such that $A_n = \lfloor n\alpha + \gamma \rfloor$ and $B_n = \lfloor n\beta + \delta \rfloor$ for all $n \in \mathbb{Z}^0$, if and only if $s = 1$.*

Proof. Since the sequence $\{A_n\}$ is strictly increasing,

$$B_{n+1} - B_n = sA_{n+1} + t(n+1) - sA_n - tn = s(A_{n+1} - A_n) + t \ge s + t \ge 2.$$

Since $B_n = sA_n + tn$, the sequence $\{B_n\}$ is nonempty. Since A and B are complementary, we thus cannot have $A_{n+1} - A_n = 1$ for all n. Therefore there exists n such that $A_{n+1} - A_n \ge 2$. Hence, there is n for which $B_{n+1} - B_n \ge 2s + t \ge 3$. It follows that there is n for which $A_{n+1} - A_n = 1$. Since for all $n \in \mathbb{Z}^0$, we have $B_{n+1} - B_n \ge 2$; there can be no n for which $A_{n+1} - A_n > 2$. Hence, $A_{n+1} - A_n \in \{1,2\}$ and $B_{n+1} - B_n \in \{s+t, 2s+t\}$ for all $n \ge 0$.

Given a nondecreasing sequence of integers $S = a_1, a_2, \ldots$, the *spectrum* question is whether there exist real numbers α, γ such that $S = \lfloor n\alpha + \gamma \rfloor$. The spectrum terminology is used in [9], where it is shown, for the *homogeneous* case ($\gamma = 0$), that if the prefix M_r of length r of S is "nearly linear", then it is the beginning of a spectrum. If it is, we will say that M_r is *spectral*. In [2], necessary and sufficient conditions are given for M_r to be spectral in the (possibly) nonhomogeneous case (see also [3]).

Let

$$\underline{d}(M_r) = \max_{1 \le i < k \le r} \frac{a_k - a_{k-i} - 1}{i}, \quad \overline{d}(M_r) = \min_{1 \le i < k \le r} \frac{a_k - a_{k-i} + 1}{i}.$$

One of the necessary and sufficient conditions for M_r to be spectral given in [2] is that $\underline{d}(M_r) < \overline{d}(M_r)$.

Put $a_k = B_{n+1}$, $a_{k-1} = B_n$. For any portion of length r of the sequence $\{B_n\}$ for which both the difference $2s+t$ and $s+t$ occurs, we have

$$\underline{d}(M_r) \ge a_k - a_{k-1} - 1 = 2s + t - 1, \quad \overline{d}(M_r) \le a_k - a_{k-1} + 1 = s + t + 1$$

where we use the larger difference for $\underline{d}$ and the smaller for $\overline{d}$. So a necessary condition for M_r to be spectral is $2s + t - 1 < s + t + 1$, which holds if and only if $s < 2$, i.e., $s = 1$. For $s = 1$, the sequence $\{B_n\}$ is indeed a spectrum, as remarked above.

The structure of $\{B_n\}$ implies that in $\{A_n\}$ there are runs of 1's of length $s+t-2$ and $2s+t-2$. An argument analogous to the above then leads to the necessary condition $2s+t-3 < s+t-1$, which again leads to $s=1$, for which $\{A_n\}$ is indeed a spectrum. ∎

Thus, the question whether our heap games have a polynomial strategy or not is still open for all $s > 1$.

4. A Class of Exotic Numeration Systems

For u_{-1} a constant, $u_0 = 1$ and b_1, b_2 integers satisfying $b_1 \geq b_2 \geq 1$, consider the linear recurrence $u_n = b_1 u_{n-1} + b_2 u_{n-2}$ $(n \geq 1)$. We can consider $u_0, u_1, \ldots$ as bases of a numeration system with digits $d_i \in \{0, \ldots, b_1\}$. But then an integer such as u_n has two representations: u_n itself and $b_1 u_{n-1} + b_2 u_{n-2}$. Since we would like to have uniqueness of representation, it is natural to require that $d_i = b_1 \implies d_{i-1} < b_2$ $(i \geq 1)$. It turns out that under this condition, every positive integer m indeed has a unique representation. This is a special case of Theorem 3.1 in [7]. Moreover, the greedy algorithm of repeatedly dividing m or its remainder by the largest u_i not exceeding this remainder yields this unique representation. The case $b_1 = b_2 = 1$ gives a binary representation known as the Zeckendorf representation [15].

Example. We consider the case $u_{-1} = 1/2$, $(b_1, b_2) = (3, 2)$. Then $u_1 = 4$, $u_2 = 14$, $u_3 = 50$, $u_4 = 178$, $\ldots$. The representations of the integers 1 to 60 in this numeration system are displayed in Table 2.

A question we just might ask at this point is whether there is any connection between Tables 1 and 2. If we scan the first few entries of both, we may be tempted to conclude that all the entries under A_n in Table 1 have representations not ending in 0 in Table 2. But then 14 is a counterexample, whose representation ends in two 0's. Also it appears that the B_n all have representations ending in a single 0. But 50, with representation 1000 is a counterexample, in fact, the only counterexample in the range of the two tables.

However, there is no counterexample, as far as the two tables go, to the following two remarkable, aesthetically pleasing properties:

(a) All the A_n's have representations ending in an *even* number of 0's and all the B_n's have representations ending in an *odd* number of 0's.

(b) For every $(A_n, B_n) \in P$, the representation of B_n is the "left shift" of the representation of A_n.

Thus, $(1, 4)$ of Table 1 has representation $(1, 10)$, and $(6, 22)$ has representation $(12, 120)$: 10 is the "left shift" of 1, 120 the left shift of 12. We remark that the second part of (a) is not independent; it follows from its first part, since A and B are complementary.

In the next section, we state these properties in a precise manner and prove their validity.

5. Wedding Numeration Systems with Heap Games

For fixed $s,t \in \mathbb{Z}^+$, put $u_{-1} = 1/s$, $u_0 = 1$, and let $u_n = (s+t-1)u_{n-1} + su_{n-2}$ $(n \geq 1)$. Denote by U the numeration system with bases $u_0, u_1, \ldots$ and digits $d_i \in \{0,\ldots, s+t-1\}$ such that $d_{i+1} = s+t-1 \implies d_i < s$ $(i \geq 0)$. Every positive integer has a unique representation over U, as mentioned in the previous section.

Table 2: Representation of first few integers in $\mathbb{Z}^+$.

50	14	4	1	n	14	4	1	n
	2	0	3	31			1	1
	2	1	0	32			2	2
	2	1	1	33			3	3
	2	1	2	34		1	0	4
	2	1	3	35		1	1	5
	2	2	0	36		1	2	6
	2	2	1	37		1	3	7
	2	2	2	38		2	0	8
	2	2	3	39		2	1	9
	2	3	0	40		2	2	10
	2	3	1	41		2	3	11
	3	0	0	42		3	0	12
	3	0	1	43		3	1	13
	3	0	2	44	1	0	0	14
	3	0	3	45	1	0	1	15
	3	1	0	46	1	0	2	16
	3	1	1	47	1	0	3	17
	3	1	2	48	1	1	0	18
	3	1	3	49	1	1	1	19
1	0	0	0	50	1	1	2	20
1	0	0	1	51	1	1	3	21
1	0	0	2	52	1	2	0	22
1	0	0	3	53	1	2	1	23
1	0	1	0	54	1	2	2	24
1	0	1	1	55	1	2	3	25
1	0	1	2	56	1	3	0	26
1	0	1	3	57	1	3	1	27
1	0	2	0	58	2	0	0	28
1	0	2	1	59	2	0	1	29
1	0	2	2	60	2	0	2	30

Notations and Definitions.

(a) For every $m \in \mathbb{Z}^0$, write $R(m)$ for the representation of m over U.

(b) Denote by $LR(m)$ the "left shift" of $R(m)$, i.e., if $R(m) = \sum_{i=0}^{n} d_i u_i$, then $LR(m) = \sum_{i=0}^{n} d_i u_{i+1}$.

(c) A positive integer m is *EVen-taILed* (for short: *evil*) if $R(m)$ ends in an even (possibly 0) number of 0's. It is *Odd-taiLeD* (for short: *old*) if $R(m)$ ends in an odd number of 0's. It is convenient to let 0 be both evil and old.

(d) Put $q = s-1$ and $r = s+t-1$. Then the above recurrence has the form $u_{-1} = 1/s$, $u_0 = 1$, $u_n = ru_{n-1} + su_{n-2}$ $(n \geq 1)$; and the representation with digits $d_i \in \{0,\ldots,r\}$ satisfies $d_{i+1} = r \implies d_i \leq q$ $(i \geq 0)$.

We mention that in [1, Ch. 4], "evil number" is used for a number whose binary expansion contains an even number of 1's (*even weight* in coding theory language).

Lemma 5.1. *For $m \in \mathbb{Z}^+$, let $R(m) = \sum_{i=0}^{n} d_i u_i$.*

(i) *Suppose for some $k \in \mathbb{Z}^0$, the tail of $R(m)$ has digits*

$$d_{2k} d_{2k-1} d_{2k-2} \ldots d_3 d_2 d_1 d_0 = d_{2k} rq \ldots rqrq,$$

where $d_{2k} \in \{0,\ldots,q\}$ and $d_{2k} = q \implies d_{2k+1} < r$. Then $R(m+1) = (d_{2k}+1)u_{2k} + \sum_{i=2k+1}^{n} d_i u_i$, so $m+1$ is evil.

(ii) *Suppose for some $k \in \mathbb{Z}^0$, the tail of $R(m)$ has digits*

$$d_{2k+1} d_{2k} d_{2k-1} \ldots d_2 d_1 d_0 = d_{2k+1} rq \ldots rqr,$$

where $d_{2k+1} \in \{0,\ldots,q\}$ and $d_{2k+1} = q \implies d_{2k+2} < r$. Then $R(m+1) = (d_{2k+1}+1)u_{2k+1} + \sum_{i=2k+2}^{n} d_i u_i$, so $m+1$ is old.

Note. We point out the special case $k = 0$, where $d_0 \leq q$ and $d_0 = q \implies d_1 < r$ for (i) and $d_1 \leq q$ and $d_1 = q \implies d_2 < r$ for (ii).

Proof. We note that the hypothesis on d_{2k} for (i) implies that $(d_{2k}+1)u_{2k} + \sum_{i=2k+1}^{n} d_i u_i$ is a legal representation over U. Similarly for (ii).

(i) By adding and subtracting u_0, we obtain

$$\begin{aligned} m &= (qu_0 + ru_1) + (qu_2 + ru_3) + \cdots + (qu_{2k-2} + ru_{2k-1}) + d_{2k}u_{2k} + \sum_{i=2k+1}^{n} d_i u_i \\ &= (d_{2k}+1)u_{2k} + \sum_{i=2k+1}^{n} d_i u_i - 1. \end{aligned}$$

Thus, $m+1 = (d_{2k}+1)u_{2k} + \sum_{i=2k+1}^{n} d_i u_i = R(m+1)$.

(ii) Adding and subtracting $su_{-1} = 1$ gives

$$\begin{aligned} m &= ru_0 + (qu_1 + ru_2) + (qu_3 + ru_4) + \cdots + (qu_{2k-1} + ru_{2k}) \\ &\quad + d_{2k+1}u_{2k+1} + \sum_{i=2k+2}^{n} d_i u_i \\ &= (d_{2k+1} + 1)u_{2k+1} + \sum_{i=2k+2}^{n} d_i u_i - 1. \end{aligned}$$

Thus, $m+1 = (d_{2k+1}+1)u_{2k+1} + \sum_{i=2k+2}^{n} d_i u_i = R(m+1)$. ∎

Lemma 5.2. *Consider the set of pairs $\bigcup_{k=0}^{\infty}(V_k, W_k)$, where $0 = V_0 < V_1 < \cdots$ is the set of all evil numbers, and $R(W_k) = LR(V_k)$ for all k. Then $W_k - sV_k = tk$ for all k.*

Proof. Induction on k. The assertion holds trivially for $k = 0$. Suppose $W_k - sV_k = tk$ for arbitrary but fixed k. Let $R(V_k) = \sum_{i=0}^{n} d_i u_i$. Then

$$R(W_k) - sR(V_k) = LR(V_k) - sR(V_k) = \sum_{i=0}^{n} d_i(u_{i+1} - su_i),$$

so

$$tk = \sum_{i=0}^{n} d_i(u_{i+1} - su_i). \tag{5.1}$$

We consider three cases.

(I) The tail of $R(V_k)$ is as in Lemma 5.1(i). Then $V_k + 1$ is evil, so $V_{k+1} = V_k + 1$ and $R(V_{k+1}) = (d_{2k}+1)u_{2k} + \sum_{i=2k+1}^{n} d_i u_i$. Thus,

$$\begin{aligned} LR(V_{k+1}) - sR(V_{k+1}) &= (d_{2k}+1)(u_{2k+1} - su_{2k}) \\ &\quad + \sum_{i=2k+1}^{n} d_i(u_{i+1} - su_i) \\ &= u_{2k+1} - su_{2k} + \sum_{i=2k}^{n} d_i(u_{i+1} - su_i). \end{aligned} \tag{5.2}$$

For Lemma 5.1(i), we have by (5.1),

$$\begin{aligned} tk &= q(u_1 - su_0) + r(u_2 - su_1) + q(u_3 - su_2) + r(u_4 - su_3) \\ &\quad + \cdots + q(u_{2k-1} - su_{2k-2}) + r(u_{2k} - su_{2k-1}) + \sum_{i=2k}^{n} d_i(u_{i+1} - su_i). \end{aligned}$$

We sum together the positive terms, adding and subtracting u_1. Then $ru_2 + su_1 = u_3$ is added to $(s-1)u_3$, and so on, leading to $u_{2k+1} - u_1$. We then sum all the negative terms, subtracting and adding su_0, leading to $-su_{2k} + su_0$. Thus,

$$\begin{aligned} tk &= u_{2k+1} - u_1 - su_{2k} + su_0 + \sum_{i=2k}^{n} d_i(u_{i+1} - su_i) \\ &= u_{2k+1} - su_{2k} - t + \sum_{i=2k}^{n} d_i(u_{i+1} - su_i). \end{aligned}$$

Hence, by (5.2),

$$t(k+1) = u_{2k+1} - su_{2k} + \sum_{i=2k}^{n} d_i(u_{i+1} - su_i) = LR(V_{k+1}) - sR(V_{k+1}),$$

as was to be shown.

(II) The tail of $R(V_k)$ is as in Lemma 5.1(ii). Then V_k+1 is old, but V_k+2 is clearly evil, since $R(V_k+2)$ ends in 1, so $V_{k+1} = V_k+2$. Then $V_{k+1} = 1+(d_{2k+1}+1)u_{2k+1} + \sum_{i=2k+2}^{n} d_iu_i$, so $R(V_{k+1}) = u_0 + (d_{2k+1}+1)u_{2k+1} + \sum_{i=2k+2}^{n} d_iu_i$. Hence,

$$\begin{aligned} LR(V_{k+1}) - sR(V_{k+1}) &= (u_1 - su_0) + (d_{2k+1}+1)(u_{2k+2} - su_{2k+1}) \\ &\quad + \sum_{i=2k+2}^{n} d_i(u_{i+1} - su_i) \\ &= t + u_{2k+2} - su_{2k+1} + \sum_{i=2k+1}^{n} d_i(u_{i+1} - su_i). \end{aligned} \tag{5.3}$$

For Lemma 5.1(ii), we have by (5.1),

$$\begin{aligned} tk &= r(u_1 - su_0) + q(u_2 - su_1) + r(u_3 - su_2) + q(u_4 - su_3) \\ &\quad + \cdots + q(u_{2k} - su_{2k-1}) + r(u_{2k+1} - su_{2k}) + \sum_{i=2k+1}^{n} d_i(u_{i+1} - su_i). \end{aligned}$$

Summing the positive terms, adding and subtracting su_0, leads to $u_{2k+2} - s$. Summing the negative terms, subtracting and adding $s^2q_{-1} = s$, gives $-su_{2k+1} + s$. Thus, $tk = u_{2k+2} - su_{2k+1} + \sum_{i=2k+1}^{n} d_i(u_{i+1} - su_i)$.

Thus, by (5.3), $LR(V_{k+1}) - sR(V_{k+1}) = t(k+1)$, as required.

(III) The digit d_0 satisfies $q < d_0 < r$. Since clearly $d_1 < r$, we have $V_{k+1} = V_k + 1$, so by (5.1),

$$\begin{aligned} LR(V_{k+1}) - sR(V_{k+1}) &= (d_0+1)(u_1 - su_0) + \sum_{i=1}^{n} d_i(u_{i+1} - su_i) \\ &= t + \sum_{i=0}^{n} d_i(u_{i+1} - su_i) = t(k+1). \end{aligned}$$

Every evil number P is of one of the three forms considered above. Note that if P ends in an even number of 0's, it is of the form **(I)**. ∎

Lemma 5.2 enables us to prove our main result.

Theorem 5.1. *For all* $n \in \mathbb{Z}^0$, $(V_n, W_n) = (A_n, B_n)$.

Proof. Since $(V_0, W_0) = (A_0, B_0) = (0, 0)$, it suffices to show that for $n > 0$, the numbers V_n, W_n have the same inductive formation laws as the numbers A_n, B_n. By Lemma 5.2, $W_n = sV_n + tn$, the same as the formation rule for B_n given in (2.2). It remains only to show that $V_n = \text{mex}\, S$, where $S = \{V_i, W_i : i < n\}$. Suppose $\text{mex}\, S = W_j$. Clearly, $j \geq n$. But then $V_j \in S$, since $V_j < W_j$. Hence, $j < n$, a contradiction.

Now, $\bigcup_{i=1}^{\infty} V_i$ and $\bigcup_{i=1}^{\infty} W_i$ are complementary since every positive integer has precisely one representation in U, either ending in an even number of 0's, as the V_n's do, or in an odd number, as the W_n's do, since the W_n's are a left shift of the V_n's. Therefore, if $\text{mex}\, S \neq W_j$, we must have $\text{mex}\, S = V_n$, so the formation laws are the same. ∎

6. The End of the Games Story

Theorem 5.1 enables us to decide the main question: whether or not our heap games have a polynomial strategy. Given any game position (x,y) with $0 < x \le y$, compute $R(x)$ using the greedy algorithm mentioned in the first paragraph of Sect. 4. If $R(x)$ ends in an odd number of 0's, then $x = B_n$ for some $n > 0$. Then move $y \to A_n$, where $R(B_n) = LR(A_n)$. If $R(x)$ ends with an even number of 0's, then $x = A_n$ for some $n > 0$. We can also test the relative size of y and B_n, since $R(B_n) = LR(A_n)$. This information suffices for deciding the game, as indicated in the proof of Theorem 2.1 (and used again in Sect. 3). So the complexity of this computation, up to a multiplicative constant, is that of computing $R(x)$, which is linear in $\log x$.

The recurrence $u_n = ru_{n-1} + su_{n-2}$ has characteristic polynomial $x^2 - rx - s = 0$, with roots $\alpha = (r + \sqrt{r^2+4s})/2$, $\beta = (r - \sqrt{r^2+4s})/2$. Since $s > 0$, we have $\alpha > 1$. Since $s = r - t + 1 \le r$, we have $0 < -\beta \le (\sqrt{(r+2)^2 - 4} - r)/2 < 1$, so $|\beta| < 1$. Therefore, $u_n = E(c\alpha^n)$ for some constant $c > 0$, where $E(v)$ is the nearest integer to the real number v. It follows that the first $n = O(\log x)$ bases of U suffice for computing the strategy. Thus, this strategy is in fact *linear* in the input size.

7. Yet Another Class of Sequences

In addition to the class of sequences $\{A_n\}$ and $\{B_n\}$ defined in (2.2), we now define another class of three sequences, $Q = \{Q_n\}$, $\{A'_n\}$, $\{B'_n\}$ $(n \in \mathbb{Z}^0)$, also depending on positive integer parameters s, t.

(a) $Q_n = Q_m$ if $n = tQ_m + sm$ and Q_m has already occurred precisely once, or else

$$Q_n = \operatorname{mex}\{Q_m : 0 \le m < n\}. \tag{7.1}$$

Initial values: $Q_0 = 0$, $Q_1 = 1$.

(b) A'_n = smallest k such that $Q_k = n$.

(c) B'_n = largest k such that $Q_k = n$ and there is $j < k$ with $Q_j = Q_k$.

Our main purpose here is to show that, despite the different definitions of the sequences, we actually have $A'_n = A_n$ and $B'_n = B_n$ for all $n \in \mathbb{Z}^0$.

Partition Q into subsequences $Q^1 = \{Q^1_n\}$, $Q^2 = \{Q^2_n\}$, where Q^1 consists of all the terms $Q_n = Q_m$ with smallest m, and Q^2 consists of the same terms but with largest n.

Example. For $s = 2$, $t = 1$, Table 3 lists the first few terms of these sequences. It is convenient to precede the proof with two lemmas.

Lemma 7.1. (i) *Let $Q^2_i = r$, $Q^2_j = r+1$ be any two consecutive terms of Q^2. Then $j - i \ge 2$.*

(ii) *Let $Q^1_i = r$, $Q^1_j = r+1$ be any two consecutive terms of Q^1. Then $j - i \le 2$.*

Proof. (i) We have $Q^2_i = r$ if $i = tQ_m + sm$ for some $m < i$, and $Q^2_j = r+1$ if $j = tQ_n + sn$ for some $m < n < i+1$, and Q_m, Q_n occurred precisely once before. Then $j - i = t(Q_n - Q_m) + s(n-m)$. We clearly have $Q_n > Q_m$. Therefore, $j - i \ge t + s \ge 2$.

(ii) The mex property (7.1) implies $j = i+1$, unless $i+1 = tQ_m + sm$ for some $m < i+1$, where Q_m appeared precisely once before. In this case, $Q_{i+1} \in Q^2$. Part (i) implies that in this latter case, $Q_{i+2} \in Q^1$, so $j = i+2$. ∎

Table 3: The beginning terms of the five sequences for $s=2$, $t=1$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Q_n	0	1	2	1	3	4	2	5	6	7	8	3	9	10	4	11	12	13	14	5	15
Q_n^1	0	1	2		3	4		5	6	7	8		9	10		11	12	13	14		15
Q_n^2	0			1			2					3			4					5	
A'_n	0	1	2	4	5	7	8	9	10	12	13	15	16	17	18	20	21	23	24	26	
B'_n	0	3	6	11	14	19	22	25													

Lemma 7.2. *$A'_{r+1}-A'_r \in \{1,2\}$ for all $r \in \mathbb{Z}^+$.*

Proof. A'_r is the smallest i such that $Q_i = r$, and A'_{r+1} is the smallest j such that $Q_j = r+1$. The minimality of i and j means that $Q_i = Q_i^1$, $Q_j = Q_j^1$, and $Q_i^1 = r$, $Q_j^1 = r+1$ are consecutive. The result now follows from Lemma 7.1(ii). ∎

We are now ready to prove

Theorem 7.1. *For every $s,t \in \mathbb{Z}^+$ we have $A'_n = A_n$, $B'_n = B_n$ for all $n \in \mathbb{Z}^+$, where A_n, B_n are defined in* (2.2).

Proof. Induction on n. By (2.2), $A_1 = 1$. Also, $Q_1 = 1$ implies $A'_1 = 1$. Suppose we already showed that $A'_i = A_i$ for all $i \le n$.

By Lemma 7.2, $A'_{n+1} - A'_n \in \{1,2\}$. In the proof of Theorem 3.1 (Sect. 3), we also showed that $A_{n+1} - A_n \in \{1,2\}$ for all $n \in \mathbb{Z}^+$.

Case (i). $A_{n+1} = A_n + 1$. Then by (2.2), $A_n + 1 = sA_m + tm$ for no $m \in \mathbb{Z}^+$. Suppose $A'_{n+1} = A'_n + 2$. Let $A'_n = k$. Then $A'_{n+1} = k+2$, $Q_k = n$, $Q_{k+2} = n+1$; and $Q_{k+1} = Q_{A'_n+1}$ has the property that $A'_n + 1 = A_n + 1 = tQ_m + sm$, where Q_m has already occurred precisely once before. Thus, if $Q_m = r$, then $A'_r = m$. Thus, $A_n + 1 = tQ_{A'_n} + sA'_r = tr + sA'_r = tr + sA_r$ by the induction hypothesis, which is a contradiction.

Case (ii). $A_{n+1} = A_n + 2$. The argument is similar to that of Case (i), therefore, it is omitted. Thus, $A'_n = A_n$ for all $n \in \mathbb{Z}^+$.

Now, $B'_n = k$ if $Q_k = sn$ and Q_k occurred precisely once as some Q_j. It follows that for every $k \in \mathbb{Z}^+$, we have either $B'_n = k$ or $A'_n = k$, but not both. Hence, A', B' are complementary sets, where $A' = \bigcup_{n=1}^{\infty} A'_n$, $B' = \bigcup_{n=1}^{\infty} B'_n$. The same was shown for $A = \bigcup_{n=1}^{\infty} A_n$, $B = \bigcup_{n=1}^{\infty} B_n$ at the beginning of the proof of Theorem 2.1 (Sect. 2). It also follows that $B'_n = B_n$ for all $n \in \mathbb{Z}^+$. ∎

(1) The special case $s=2$, $t=1$ of the second class (without B'_n) is listed in Neil Sloane's database of sequences [12], having there sequence numbers 26366 (Q_n) and 26367 (A'_n), ascribed to Clark Kimberling. We have not found a reference to other sequences of these families in [12]. In the definition of sequence 26366, "$a(n) = a(m)$ if m has already occurred exactly once..."; m should presumably be replaced by $a(m)$.

(2) We have $Q_0 = 0$, $Q_1 = 1$ for all $s,t \in \mathbb{Z}^0$, and $Q_2 = 1$ for $s = t = 1$, $Q_2 = 2$ for all s,t with $s+t > 2$.

(3) The definition of the second class of sequences and the proof that both classes are identical, throws some light on the properties of both.

8. Epilogue

The heap games proposed and analyzed here belong to the family of *succinct* games, so named because their input size is succinct: $O(\log n)$ rather than $O(n)$. Usually, extra effort is required for showing that such games are polynomial, i.e., have a polynomial strategy, because not more than $O(\log n)$ computation steps can be used. Different families of succinct games seem to require different methods of strategy computations.

For example, in *octal* games, invented by Guy and Smith [10], a linearly ordered string of beads may be split and or reduced according to rules encoded in octal (see also [1, Ch. 4], [4, Ch. 11]). The standard method for showing that an octal game is polynomial is to demonstrate that its *Sprague–Grundy* function (the 0's of which constitute the set of *P*-positions) is periodic. Periodicity has been established for a number of octal games. Some of the periods and or preperiods may be very large (see [8]). Another way to establish polynomiality is to show that the Sprague–Grundy function values obey some other simple rule, such as forming an arithmetic sequence, as for Nim.

For the present class of heap games, polynomiality was established by a nonstandard method. An arithmetic procedure, based on a class of special numeration systems, was the key to polynomiality. It appears that at this stage in the development of combinatorial game theory, there is no unified method for establishing polynomiality. But this malady seems to be common to most of discrete mathematics. Some might not even call it a malady, but a feature inherent in the nature of mathematics.

In [14], the special case of the Zeckendorf numeration system [15] was used to give one of the characterizations of the *P*-positions of Wythoff's game $(s = t = 1)$. This method was extended in [6] for the generalized Wythoff game introduced there $(s = 1,\ t \geq 1)$. In both cases, the bases of the numeration system were the numerators of the simple continued expansion of α, where α is such that $A_n = \lfloor n\alpha \rfloor$ for all $n \geq 0$. The interesting aspect is that despite the fact that such α does not exist for $s > 1$ (Theorem 3.1, Sect. 3), the polynomial characterization based on special numeration systems nevertheless does. We also remark that it would be of interest to compute the Sprague–Grundy function for these heap games. For Wythoff's game, this seems to be quite difficult, but this fact says nothing about the case $s > 1$.

In [2], it is shown that a sequence $\{A_n\}$ is spectral (defined in the proof of Theorem 3.1), if and only if $|(A_{n+i} - A_n) - (A_{m+i} - A_m)| \leq 1$ for all $i, m, n \geq 1$. Another motivation for the present paper was to extend this condition, namely, to create and characterize sequences satisfying $|(A_{n+i} - A_n) - (A_{m+i} - A_m)| \leq 2$. Vera Sós told me that she has also been interested in this question. For the subfamily $s = t$ of the sequences $\{A_n\}$ defined in (2.2) (Sect. 2), we have perhaps $|(A_{n+i} - A_n) - (A_{m+i} - A_m)| \leq s$. If this is true, does the converse also hold, namely, does $|(A_{n+i} - A_n) - (A_{m+i} - A_m)| \leq s$ imply (2.2) with $s = t$? Investigation of the full family of these sequences (any $s, t \in \mathbb{Z}^+$) is of independent interest. In Sect. 7, we defined a class of sequences and demonstrated its equivalence with the class of sequences defined in (2.2).

A succinct game is not always more difficult than "its nonsuccinct version"! We illustrate this with the game *vertex Kayles*. Given a finite (undirected) graph G, a move is to label an as yet unlabeled vertex not adjacent to any labeled vertex. The first player unable to play loses and the opponent wins. A partizan variation is called *bigraph*

vertex Kayles. Both versions have been proved P-space hard in [11]. If G is a path, the resulting succinct game, known as *Kayles*, is actually polynomial! It is the octal game 0.137 (see [1, 4, 10]). Incidentally, there are "no-man's-land" games lying in between the polynomial 0.137 and the P-space hard vertex Kayles. It would be of interest to shrink the area of this no-man's-land.

Finally, the family of combinatorial games roughly consists of two-player games with perfect information (no hidden information as in some card games), no chance moves (no dice) and outcome restricted to win/lose. These games are *completely determined*, so one of their main mathematical interests is in bounding the complexity of their strategies. This explains why we have often mentioned efficiency of strategy computation in this paper.

Added in proof. Neil Sloane has recently put some of our (s,t)-sequences into his on-line database of sequences [12]: the prefixes of the A_n/B_n-sequences for $(s,t) \in \{(2,2),(2,3),(3,1),(3,2)\}$ have sequence numbers 45671/72, 45681/82, 45749/50, 45774/75 respectively.

References

1. E.R. Berlekamp, J.H. Conway and R.K. Guy, Winning Ways (two volumes), Academic Press, London, 1982.
2. M. Boshernitzan and A.S. Fraenkel, Nonhomogeneous spectra of numbers, Discrete Math. **34** (1981) 325–327.
3. M. Boshernitzan and A.S. Fraenkel, A linear algorithm for nonhomogeneous spectra of numbers, J. Algorithms **5** (1984) 187–198.
4. J.H. Conway, On Numbers and Games, Academic Press, London, 1976.
5. H.S.M. Coxeter, The golden section, phyllotaxis and Wythoff's game, Scripta Math. **19** (1953) 135–143.
6. A.S. Fraenkel, How to beat your Wythoff games' opponents on three fronts, Amer. Math. Monthly **89** (1982) 353–361.
7. A.S. Fraenkel, Systems of numeration, Amer. Math. Monthly **92** (1985) 105–114.
8. A. Gangolli and T. Plambeck, A note on periodicity in some octal games, Internat. J. Game Theory **18** (1989) 311–320.
9. R.L. Graham, S. Lin, and C.-S. Lin, Spectra of numbers, Math. Mag. **51** (1978) 174–176.
10. R.K. Guy and C.A.B. Smith, The *G*-values of various games, Proc. Camb. Phil. Soc. **52** (1956) 514–526.
11. T.J. Schaefer, On the complexity of some two-person perfect-information games, J. Comput. System Sci. **16** (1978) 185–225.
12. N.J.A. Sloane, Sloane's On-Line Encyclopedia of Integer Sequences, http://www.research.att.com/~njas/sequences/ , 1998.
13. W.A. Wythoff, A modification of the game of Nim, Nieuw Arch. Wisk. **7** (1907) 199–202.
14. A.M. Yaglom and I.M. Yaglom, Challenging Mathematical Problems with Elementary Solutions, J. McCawley, Jr., Transl., B. Gordon, Ed., Vol. II, Holden-Day, San Francisco, 1976.
15. E. Zeckendorf, Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas, Bull. Soc. Roy. Sci. Liège **41** (1972) 179–182.

On growth rates of hereditary permutation classes

Tomáš Kaiser[1,3] and Martin Klazar[2,3]

May 17, 2002

Abstract

A class of permutations Π is called hereditary if $\pi \subset \sigma \in \Pi$ implies $\pi \in \Pi$, where the relation $\subset$ is the natural containment of permutations. Let Π_n be the set of all permutations of $1, 2, \ldots, n$ belonging to Π. We investigate the counting functions $n \mapsto |\Pi_n|$ of hereditary classes. Our main result says that if $|\Pi_n| < 2^{n-1}$ for at least one $n \geq 1$, then there is a unique $k \geq 1$ such that $F_{n,k} \leq |\Pi_n| \leq F_{n,k} \cdot n^c$ holds for all $n \geq 1$ with a constant $c > 0$. Here $F_{n,k}$ are the generalized Fibonacci numbers which grow like powers of the largest positive root of $x^k - x^{k-1} - \cdots - 1$. We characterize also the constant and the polynomial growth of hereditary permutation classes and give two more results on these.

1 Introduction

A permutation $\sigma = (b_1, b_2, \ldots, b_n)$ of $[n] = \{1, 2, \ldots, n\}$ *contains* a permutation $\pi = (a_1, a_2, \ldots, a_k)$ of $[k]$, in symbols $\sigma \supset \pi$, if σ has a (not necessarily consecutive) subsequence of length k whose terms induce the same order pattern as π. For example, $(3, 5, 4, 2, 1, 7, 8, 6, 9)$ contains $(2, 1, 3)$, as $(\ldots, 5, \ldots, 1, \ldots, 6, \ldots)$ or as $(\ldots, 4, 2, \ldots, 9)$, but it does not contain $(3, 1, 2)$.

Let $f(n, \pi)$ be the number of the permutations of $[n]$ not containing π. The following conjecture was made by R. P. Stanley and H. S. Wilf (it appeared first in print in Bóna [10, 11, 12]).

The Stanley–Wilf conjecture. For every permutation π, there is a constant $c > 0$ such that $f(n, \pi) < c^n$ for all $n \geq 1$.

[1]Department of Mathematics, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic, e-mail: kaisert@kma.zcu.cz.

[2]Department of Applied Mathematics, Charles University, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic, e-mail: klazar@kam.mff.cuni.cz.

[3]Institute for Theoretical Computer Science (ITI), Charles University, Praha, Czech Republic. Supported by project LN00A056 of the Czech Ministery of Education.

It is known to hold for all π of length at most 4 (Bóna [12]), for all layered π (Bóna [13], see below for the definition of layered permutations), and for *all* π in a weaker form with an almost exponential upper bound (Alon and Friedgut [3]). A permutation π of $[n]$ is called *layered* if $[n]$ can be partitioned into intervals $I_1 < I_2 < \ldots < I_k$ so that every restriction $\pi|I_i$ is decreasing and $\pi(I_1) < \pi(I_2) < \ldots < \pi(I_k)$. (We call π layered also in the case when $\pi(I_1) > \pi(I_2) > \ldots > \pi(I_k)$ and the restrictions $\pi|I_i$ are increasing.) Equivalently, π is layered (in the former sense) if and only if it contains neither $(2,3,1)$ nor $(3,1,2)$. Other works dealing with the conjecture and/or the containment of permutations are, to name a few, Adin and Roichman [1], Albert et al. [2], Bóna [11], Klazar [18], Stankova-Frenkel and West [29], and West [33].

A class Π of permutations is *hereditary* if, for every π and σ, $\pi \subset \sigma \in \Pi$ implies $\pi \in \Pi$. The symbol Π_n, $n \in \mathbf{N} = \{1, 2, \ldots\}$, denotes the set of all permutations in Π of length n. The *counting function* of Π is the function $n \mapsto |\Pi_n|$ whose value at n is the number of permutations in Π of length n. For example, $n \mapsto 0$ is the counting function of the empty class $\Pi = \emptyset$, while the (hereditary) class of all permutations has the counting function $n \mapsto n!$. The Stanley–Wilf conjecture says, in effect, that except for the latter trivial example, there are no other superexponential counting functions.

A reformulation of the Stanley–Wilf conjecture. Let Π be any hereditary class of permutations different from the class of all permutations. Then $|\Pi_n| < c^n$ for all $n \geq 1$ and a constant $c > 0$.

Indeed, if Π is hereditary and $\pi \notin \Pi$, then $|\Pi_n| \leq f(n, \pi)$ for all $n \geq 1$. On the other hand, for every π the function $n \mapsto f(n, \pi)$ is the counting function of the hereditary class consisting of all permutations not containing π.

If one starts to investigate the realm of hereditary permutation classes from the top, one gets immediately stuck at the question whether every counting function different from the trivial $n \mapsto n!$ has to be at most exponential. In this article we take the other course and start from the bottom, at the empty class $\Pi = \emptyset$. We shall investigate the counting functions of 'small' hereditary permutation classes.

We summarize our results and give a few more definitions. Theorem 2.1 points out two simple set-theoretical facts about the set of all hereditary classes. Theorem 2.2, due to P. Valtr, gives a uniform lower bound on $\liminf_{n\to\infty} f(n,\pi)^{1/n}$. Sections 3 and 4 contain our main results. Theorem 3.4 shows that any counting function grows either at most polynomially or at least as the Fibonacci numbers F_n. Thus $n \mapsto F_n$ is the smallest superpolynomial counting function. Theorem 3.8 classifies the possible exponential growth rates below $n \mapsto 2^{n-1}$: Either $|\Pi_n| \geq 2^{n-1}$ for all $n \geq 1$, or there is a unique $k \geq 1$ such that $|\Pi_n|$ grows, up to a polynomial factor, as the generalized Fibonacci numbers $F_{n,k}$. Theorem 4.2 shows that any counting function is either eventually constant or grows at least as the identity function $n \mapsto n$. Thus $n \mapsto n$ is the smallest unbounded counting function. Theorem 4.4 shows that if the function $n \mapsto |\Pi_n|$ grows polynomially, then it is eventually an integral linear combination of the polynomials $\binom{n-i}{j}$. The concluding part (Section 5) contains some remarks and comments.

Recall that $\mathbf{N}$ denotes $\{1, 2, \ldots\}$, the set of positive integers, and $[n]$ denotes the set $\{1, 2, \ldots, n\}$. More generally, for $a, b \in \mathbf{N}$ and $a \leq b$, the interval $\{a, a+1, \ldots, b\}$ is denoted by $[a, b]$. If π is a permutation of $[n]$, we say that n is its *length* and write $|\pi| = n$. Let $A_1, A_2, B_1, B_2 \subset \mathbf{N}$ be four finite sets of the same cardinality. We call two bijections $f : A_1 \to A_2$ and $g : B_1 \to B_2$ *similar* iff $f(x) = j(g(h(x)))$ holds for every $x \in A_1$, where $h : A_1 \to B_1$ and $j : B_2 \to A_2$ are the unique increasing bijections. In other words, using only the order relation we cannot distinguish the graphs of f and g. Every bijection between two n-element subsets of $\mathbf{N}$ is similar to a unique permutation of $[n]$. For two permutations σ and π, σ contains π iff a subset of σ (regarded as a set of pairs) is similar to π. We take the *restriction* $\pi|X$ of a permutation π of $[n]$ to a subset $X \subset [n]$ to be the unique permutation similar to the usual restriction. For a set of permutations X we define

$$\mathrm{Forb}(X) = \{\pi : \ \pi \text{ contains no } \sigma \in X\}.$$

For any X, this is a hereditary class. Note that for every hereditary class Π there is exactly one set X of permutations pairwise incomparable by $\subset$ (that is, X is an *antichain*) such that $\Pi = \mathrm{Forb}(X)$; the set X consists of the minimal permutations not in Π. Thus the hereditary permutation classes correspond bijectively to antichains of permutations. A function $f : \mathbf{N} \to \mathbf{N}$ *eventually dominates* another function $g : \mathbf{N} \to \mathbf{N}$ iff $f(n) \geq g(n)$ for every $n \geq n_0$.

2 The number of hereditary classes and a lower bound on $f(n, \pi)$

If Π and Π' are hereditary classes of permutations and $\Pi \backslash \Pi'$ is finite then, trivially, $n \mapsto |\Pi'_n|$ eventually dominates $n \mapsto |\Pi_n|$. By the following theorem, there are uncountably many classes such that this trivial comparison does not apply for any two of them.

Theorem 2.1 *(1) There exist $2^{\aleph_0}$ (continuum many) distinct hereditary classes of permutations.*

(2) In fact, there exists a set S of $2^{\aleph_0}$ hereditary classes of permutations such that for every $\Pi, \Pi' \in S$, $\Pi \neq \Pi'$, both sets $\Pi \backslash \Pi'$ and $\Pi' \backslash \Pi$ are infinite.

Proof. (1) It is known (see, for example, Spielman and Bóna [28]) that there is an infinite antichain of permutations A. Then

$$\{\mathrm{Forb}(X) : \ X \subset A\}$$

is a set of $2^{\aleph_0}$ hereditary classes. Indeed, every $\mathrm{Forb}(X)$ is hereditary and it is easy to see that $X, Y \subset A$, $X \neq Y$ implies $\mathrm{Forb}(X) \neq \mathrm{Forb}(Y)$.

(2) In fact, if $X, Y \subset A$ and $\pi \in X \backslash Y$ then $\pi \in \mathrm{Forb}(Y) \backslash \mathrm{Forb}(X)$. It suffices to show that there is a system of $2^{\aleph_0}$ subsets of A such that the set difference of every two distinct members of the system is infinite. For the notational convenience we identify A with $\mathbf{N}$.

Recall that for $X \subset \mathbf{N} = A$, the upper and lower asymptotic densities of X are defined as

$$\overline{d}(X) = \limsup_{n\to\infty} \frac{|X \cap [n]|}{n} \quad \text{and} \quad \underline{d}(X) = \liminf_{n\to\infty} \frac{|X \cap [n]|}{n}.$$

For every real constant c, $0 < c < \frac{1}{2}$, we select a subset $X_c \subset \mathbf{N} = A$ such that $\overline{d}(X_c) = 1 - c$ and $\underline{d}(X_c) = c$. Then

$$S = \{\mathrm{Forb}(X_c) : \ 0 < c < \tfrac{1}{2}\}$$

is a set of $2^{\aleph_0}$ hereditary classes with the stated property. Indeed, for every two real constants $c, d \in (0, \frac{1}{2})$, $c \neq d$, the set $X_c \backslash X_d$ is infinite because for $X, Y \subset \mathbf{N}$ with $X \backslash Y$ finite one has $\underline{d}(X) \leq \underline{d}(Y)$ and $\overline{d}(X) \leq \overline{d}(Y)$. □

Of course, there is nothing special about permutations in the previous theorem. It holds for the hereditary classes in any countably infinite poset that has an infinite antichain. Do there exist two hereditary classes of permutations such that their counting functions are incomparable by the eventual dominance? Are there $2^{\aleph_0}$ such hereditary permutation classes?

We take the opportunity to include an unpublished lower bound on the size of a class characterized by a forbidden permutation. The following theorem and its proof are due to Pavel Valtr [32] and are reproduced here with his kind permission.

Theorem 2.2 *Let c be any constant such that $0 < c < \mathrm{e}^{-3} = 0.04978\ldots$ where $\mathrm{e} = 2.71828\ldots$ is Euler number. Then for any permutation π of length k, where $k > k_0 = k_0(c)$, we have*

$$\liminf_{n\to\infty} f(n, \pi)^{1/n} > ck^2.$$

Proof. Let π be a permutation of length k. A random permutation τ of length m contains π with probability

$$\Pr[\tau \supset \pi] \leq \frac{1}{k!}\binom{m}{k} < \frac{m^k}{(k!)^2}.$$

We set $m = \lfloor dk^2 \rfloor$ where $0 < d < \mathrm{e}^{-2}$ is a constant. Then, by the Stirling asymptotics, this probability goes to 0 with $k \to \infty$ and for all sufficiently large k we have

$$f(m, \pi) > \frac{m!}{2}.$$

We can assume that π cannot be split as $[k] = I \cup J$, $I < J$, where both I and J are nonempty and $\pi(I) < \pi(J)$. (Otherwise we replace π with the reversed permutation.) Let $n \in \mathbf{N}$ and $n = mt + u$, where $t \geq 0$ and $0 \leq u < m$ are integers. It follows that none of the $f(m, \pi)^t f(u, \pi)$ permutations

$$(b_1, \ldots, b_u, d_1 + a_1^1, \ldots, d_1 + a_m^1, d_2 + a_1^2, \ldots, d_2 + a_m^2, \ldots, d_t + a_1^t, \ldots, d_t + a_m^t)$$

of length n, where $d_i = u + (i-1)m$ and $(b_1, \ldots, b_u)$ and $(a_1^i, \ldots, a_m^i)$ are permutations not containing π, contains π. Since $m! > (m/\mathrm{e})^m$ for large k,

$$f(n,\pi)^{1/n} \geq f(m,\pi)^{t/n} > \left(\frac{m!}{2}\right)^{t/n} > \left(\frac{m!}{2}\right)^{1/m-1/n} > \frac{2^{1/n-1/m}}{(m!)^{1/n}} \cdot \frac{m}{\mathrm{e}}.$$

By the choice of m, for any $\varepsilon > 0$ and $k > k_0 = k_0(\varepsilon)$,

$$\liminf_{n\to\infty} f(n,\pi)^{1/n} > \frac{(1-\varepsilon)d}{\mathrm{e}} \cdot k^2.$$

□

Arratia [4] proved that $\lim_{n\to\infty} f(n,\pi)^{1/n}$ always exists, and therefore in the previous bound we can replace lim inf with lim. For a general permutation π of length k the bound is best possible, up to the constant c, because

$$f(n,(1,2,\ldots,k)) \leq \frac{1}{(k-1)!} \sum_{\substack{0\leq i_1,\ldots,i_{k-1} \\ i_1+\cdots+i_{k-1}=n}} \binom{n}{i_1,\ldots,i_{k-1}}^2 \leq \frac{(k-1)^{2n}}{(k-1)!}.$$

The first inequality follows from the fact that by Dilworth's theorem, every permutation with no increasing subsequence of length k can be partitioned in at most $k-1$ decreasing subsequences. The second inequality follows by the multinomial theorem. Thus $\lim_{n\to\infty} f(n,(1,2,\ldots,k))^{1/n} \leq (k-1)^2$. By the exact asymptotics found by Regev [25], $\lim_{n\to\infty} f(n,(1,2,\ldots,k))^{1/n} = (k-1)^2$.

3 Below $n \mapsto 2^{n-1}$ — the Fibonacci growths

In this section we prove Theorem 3.8 which characterizes the exponential growth rates possible for the hereditary permutation classes Π satisfying $|\Pi_n| < 2^{n-1}$ for at least one $n \geq 1$. For the proof of the following classic result see, for example, Lovász [22, Problem 14.25].

Theorem 3.1 (Erdős–Szekeres) *Every sequence of n integers has a monotone subsequence of length $\geq n^{1/2}$.*

A permutation π, $|\pi| = n$, has k *alternations* if there are $2k$ indices $1 \leq i_1 < j_1 < i_2 < j_2 < \ldots < i_k < j_k \leq n$ such that

$$\pi(\{i_1, i_2, \ldots, i_k\}) > \pi(\{j_1, j_2, \ldots, j_k\}).$$

A hereditary permutation class Π *unboundedly alternates* if for every $k \geq 1$ there is a $\pi \in \Pi$ such that π or π^{-1} has k alternations.

Lemma 3.2 *If a hereditary permutation class Π unboundedly alternates, then $|\Pi_n| \geq 2^{n-1}$ for every $n \geq 1$.*

Proof. We suppose that for any k there is a $\pi \in \Pi$ with k alternations; the case with π^{-1} is analogous. Using the hereditarity of Π and Theorem 3.1, we see that for every $n \geq 1$ there is a $\pi \in \Pi_{2n+1}$ such that the restriction $\pi|\{2i-1 : \ 1 \leq i \leq n+1\}$ is monotone, and $\pi(i) > \pi(j)$ whenever i is odd and j is even. We may assume that the restriction is increasing; the other case when it is decreasing is quite similar. By the hereditarity of Π, for every $X \subset [2,n]$ there is a $\pi_X \in \Pi_n$ such that $\pi_X(i) > \pi_X(1) \Leftrightarrow i \in X$. Distinct X give distinct permutations π_X and thus $|\Pi_n| \geq 2^{n-1}$. □

A word $u = u_1, u_2, \ldots, u_n$ *has no immediate repetitions* if $u_i \neq u_{i+1}$ for each $1 \leq i \leq n-1$. We say that u is *alternating* if $u = ababa\ldots$ for two distinct symbols a and b. For a word u we denote $\ell(u)$ the maximum length of an alternating subsequence of u. Let

$$W_{m,l,n} = \{u \in [m]^* : \ |u| = n \ \& \ \ell(u) \leq l\}$$

be the set of all words over the alphabet $[m]$ of length n which have no alternating subsequence of length $l+1$. Claim (1) of the following lemma is a result of Davenport and Schinzel [15].

Lemma 3.3 *(1) If $u = u_1, u_2, \ldots, u_n$ is a word over $[m]$ which has no immediate repetitions and satisfies $\ell(u) \leq l$, then*

$$n \leq \binom{m}{2}(l-1)+1.$$

(2) For every $m, l, n \geq 1$ we have

$$|W_{m,l,n}| \leq (m+1)^c \cdot n^c$$

where $c = \binom{m}{2}(l-1)+1$.

(3) Suppose that the alphabet $[m]$ is partitioned into r subalphabets $A_1, \ldots, A_r$ and u is a word over $[m]$ such that every subword v_i of u consisting of the occurrences of the letters in A_i satisfies $\ell(v_i) \leq l$. Then u can be split into t intervals $u = I_1 I_2 \ldots I_t$ such that every I_i uses at most one letter from every A_j, and

$$t \leq 2\binom{m}{2}(l-1)+2.$$

Proof. (1) Let $u = u_1, u_2, \ldots, u_n$ be over $[m]$, without immediate repetitions, and let $n \geq \binom{m}{2}(l-1)+2$. By the pigeon-hole principle, some l of the $n-1$ two-element sets $\{u_i, u_{i+1}\}$ must coincide. It is easy to see that the corresponding positions in u contain an $l+1$-element alternating subsequence.

(2) Every word $u \in W_{m,l,n}$ splits uniquely into intervals $u = I_1 I_2 \dots I_t$ such that $I_i = a_i a_i \dots a_i$ consists of repetitions of a single letter a_i and $a_i \neq a_{i+1}$ for $1 \leq i \leq t-1$. Contracting every I_i into one term, we obtain a word u^* over $[m]$, $|u^*| = t$, with $\ell(u^*) \leq l$ and no immediate repetitions. By (1), $t \leq \binom{m}{2}(l-1)+1 = c$. Clearly, u^* and the composition $|I_1|, |I_2|, \dots, |I_t|$ of n determine u uniquely. Thus

$$|W_{m,l,n}| \leq (\#u^*) \cdot n^c \leq (m+1)^c \cdot n^c.$$

(3) We consider the unique splitting $u = I_1 I_2 \dots I_t$, where I_1 is the longest initial interval of u using at most one letter from every A_j, I_2 is the longest following interval with the same property, etc. Note that every pair $I_i I_{i+1}$ has a subsequence a, b (where b is the first term of I_{i+1}) such that $a, b \in A_j$ for some j and $a \neq b$. Now arguing similarly as in (1), we see that

$$t \leq 2\binom{m}{2}(l-1)+2.$$

□

The shifted Fibonacci numbers $(F_n)_{n\geq 1} = (1, 2, 3, 5, 8, 13, 21, \dots)$ are defined by $F_1 = 1, F_2 = 2$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 3$. The explicit formula is

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right).$$

By induction, $F_n \leq 2^{n-1}$ for every $n \geq 1$.

The next theorem identifies the jump from the polynomial to the exponential growth and shows that $n \mapsto F_n$ is the first superpolynomial growth rate. Although it is fully subsumed in the more general Theorem 3.8, we give a sketch of the proof. We think that it may be interesting and instructive for the reader to compare how the concepts used here develop later in the more complicated proof of Theorem 3.8.

Theorem 3.4 *Let Π be any hereditary class of permutations. Then exactly one of the following possibilities holds.*

(1) There is a constant $c > 0$ such that $|\Pi_n| \leq n^c$ for all $n \geq 1$.

(2) $|\Pi_n| \geq F_n$ for all $n \geq 1$.

Proof. (Extended sketch.) We split any permutation π into $\pi = S_1 S_2 \dots S_m$ where S_1 is the longest initial monotone segment, S_2 is the longest following monotone segment, and so on. We mark the elements in S_i by i and read the marks from bottom to top (that is, from left to right in π^{-1}). This way we obtain a word $u(\pi)$ over the alphabet $[m]$, where $m = m(\pi)$ is the number of the monotone segments S_i. For example,

$$\text{if } \pi = (3,5,4,2,1,7,8,6,9) \text{ then } m(\pi) = 4 \text{ and } u(\pi) = 2,2,1,2,1,4,3,3,4$$

because $S_1 = 3,5$, $S_2 = 4,2,1$, $S_3 = 7,8$, and $S_4 = 6,9$. Note that π is determined uniquely by $u(\pi)$ and an m-tuple of signs $(\pm,\pm,\ldots,\pm)$ in which $+$ indicates an increasing segment and $-$ a decreasing one. For every pair S_i, S_{i+1} we fix an interval $T_i = T_{\pi,i} = [\min\{a,b,c\}, \max\{a,b,c\}]$ where a,b,c is a non-monotone subsequence of S_iS_{i+1} (such a subsequence certainly exists). In our example, for $i = 3$ we may set $a,b,c = 7,8,6$ and $T_3 = [6,8]$.

For a hereditary permutation class Π and π ranging over Π we distinguish four cases. Case 1a: $m(\pi)$ is bounded and so is $\ell(u(\pi))$. Case 1b: $m(\pi)$ is bounded and $\ell(u(\pi))$ is unbounded. Case 2a: $m(\pi)$ is unbounded and the maximum number of mutually intersecting intervals in the system $S(\pi) = \{T_{\pi,1}, T_{\pi,3}, T_{\pi,5}, \ldots\}$ is unbounded as well. Case 2b: $m(\pi)$ is unbounded and so is the maximum number of mutually disjoint intervals in the system $S(\pi)$.

In case 1a we use (2) of Lemma 3.3 and deduce the polynomial upper bound of claim (1). In case 1b, the class Π unboundedly alternates and, by Lemma 3.2, $|\Pi_n| \geq 2^{n-1} \geq F_n$. In case 2a, the class Π again unboundedly alternates and $|\Pi_n| \geq 2^{n-1} \geq F_n$. In case 2b, it follows by Theorem 3.1 and the definition of $T_{\pi,i}$, that either for every $n \geq 1$ we have $(2,1,4,3,6,5,\ldots,2n,2n-1) \in \Pi$ or for every $n \geq 1$ we have $(2n-1,2n,2n-3,2n-2,\ldots,1,2) \in \Pi$. Using the hereditarity of Π, we conclude that in this case, $|\Pi_n| \geq F_n$. $\square$

To state Theorem 3.8, we need a few more definitions and lemmas. For k an integer and F a power series, $[x^k]F$ denotes the coefficient at x^k in F. We define the family of generalized Fibonacci numbers $F_{n,k} \in \mathbf{N}$, where $k \geq 1$ and n are integers, by

$$F_{n,k} = [x^n] \frac{1}{1 - x^k - x^{k-1} - \cdots - x}.$$

In particular, $F_{n,1} = 1$ for every $n \geq 1$ and $F_{n,2} = F_n$. More generally, $F_{n,k} = 0$ for $n < 0$, $F_{0,k} = 1$, and

$$F_{n,k} = F_{n-1,k} + F_{n-2,k} + \cdots + F_{n-k,k} \quad \text{for } n > 0.$$

Lemma 3.5 *Let $k \geq 1$ be fixed.*

(1) For $n \to \infty$, we have the asymptotics

$$F_{n,k} = c_k \alpha_k^n + O(\beta_k^n), \; c_k = \frac{\alpha_k^k(\alpha_k - 1)^2}{\alpha_k^{k+1} - (k+1)\alpha_k + k},$$

where α_k is the largest positive real root of $x^k - x^{k-1} - x^{k-2} - \cdots - 1$ and β_k is a constant such that $0 < \beta_k < \alpha_k$.

(2) The roots α_k satisfy inequalities $1 = \alpha_1 < \alpha_2 < \alpha_3 < \ldots < 2$, and $\alpha_k \to 2$ as $k \to \infty$.

(3) For all integers m and n,

$$F_{m,k} \cdot F_{n,k} \leq F_{m+n,k}.$$

(4) For every $n \geq 1$ we have

$$F_{n,k} \leq 2^{n-1} \quad \textit{and} \quad F_{n,n} = 2^{n-1}.$$

Proof. (1) Since

$$\sum_{n\geq 0} F_{n,k} x^n = \frac{1}{1 - x^k - x^{k-1} - \cdots - x},$$

the asymptotics of $F_{n,k}$ follows by the standard technique of decomposing rational functions into partial fractions (see, for example, Stanley [30, p. 202]). We need to prove only that α_k is a simple root of the reciprocal polynomial $p_k(x) = x^k - x^{k-1} - x^{k-2} - \cdots - 1$ and that on the complex circle $|z| = \alpha_k$, the polynomial p_k has no other root besides α_k. The constant β_k can then be set to ε+the second largest modulus of a root of p_k. The form of the coefficient c_k follows by a simple manipulation from the identity $c_k = \alpha_k^{k-1}/p_k'(\alpha_k)$ provided by the partial fractions decomposition.

Clearly, $1 \leq \alpha_k < 2$. Since $xp_k' - kp_k = x^{k-1} + 2x^{k-2} + \cdots + k$, we have $p_k'(\alpha_k) > k(k+1)/4$ and α_k is a simple root of p_k. Since $p_k = (x^{k+1} - 2x^k + 1)/(x-1)$, $p_k(x) = 0$ is equivalent to $x^k = 1/(2-x)$. It is clear that no z, $|z| = \alpha_k$, $z \neq \alpha_k$, satisfies this equation.

(2) This is immediate from the identity $\alpha_k^k = 1/(2 - \alpha_k)$ used in (1).

(3) and (4): These are easy to verify inductively by the recurrence for $F_{n,k}$. We only prove (3). We proceed by induction on $m + n$. For $m < 0$ or $n < 0$ the inequality is true. It also holds for $m = n = 0$. Let $m \geq 0$ and $n \geq 1$. Then

$$F_{m,k}F_{n,k} = F_{m,k} \sum_{i=n-k}^{n-1} F_{i,k} \leq \sum_{i=m+n-k}^{m+n-1} F_{i,k} = F_{m+n,k}.$$

□

We list approximate values of the first few roots α_k:

k	2	3	4	5	6	10
α_k	1.61803	1.83928	1.92756	1.96594	1.98358	1.99901

Let A be a finite alphabet equipped with a weight function $w : A \to \mathbf{N}$. The weight $w(u)$ of a word $u = u_1, u_2, \ldots, u_m \in A^*$ is the sum $w(u_1) + w(u_2) + \cdots + w(u_m)$. We set

$$p(w, n) = \#\{u \in A^* : \ w(u) = n\}.$$

Lemma 3.6 *Let $k \geq 1$ be fixed.*

(1) If $A = \{a_1, a_2, \ldots, a_k\}$ and $w(a_i) = i$ for $i = 1, \ldots, k$, then $p(w, n) = F_{n,k}$ for every $n \geq 1$.

(2) If $A = \{a_1, a_2, \ldots, a_{k+1}\}$ and $w(a_i) = i$ for $i = 1, \ldots, k$ and $w(a_{k+1}) = k$, then $p(w, n) \geq 2^{n-1}$ for every $n \geq 1$.

Proof. In the general situation we have the identity

$$\sum_{n=0}^{\infty} p(w,n)x^n = \frac{1}{1-\sum_{a\in A} x^{w(a)}}.$$

Now (1) is clear since then $\sum_{a\in A} x^{w(a)} = x^k + x^{k-1} + \cdots + x$.

In (2), we have

$$\sum_{n=0}^{\infty} p(w,n)x^n = \frac{1}{1-(2x^k + x^{k-1} + \cdots + x)}$$

and the inequality $p(w,n) \geq 2^{n-1}$ follows by induction from the recurrence

$$p(w,n) = p(w,n-1) + \cdots + p(w,n-k+1) + 2p(w,n-k) \ (n>0)$$

starting from $p(w,n) = 0$ for $n < 0$ and $p(w,0) = 1$. □

In (2), one might be interested in a more precise bound. Since $1-(2x^k+x^{k-1}+\cdots+x) = (1-2x)(x^{k-1}+x^{k-2}+\cdots+1)$, the decomposition into partial fractions gives

$$p(w,n) = [x^n]\left(\frac{\alpha}{1-2x} + \frac{\beta_1}{1-x/\zeta_1} + \cdots + \frac{\beta_{k-1}}{1-x/\zeta_{k-1}}\right)$$

where $\alpha, \beta_1, \ldots \beta_{k-1} \in \mathbf{C}$ are suitable constants and ζ_i are the k-th roots of unity distinct from 1. Thus $\alpha = 1/\sum_{i=0}^{k-1}(\frac{1}{2})^i$ and, for $n \to \infty$, we obtain the asymptotics

$$p(w,n) = \left(\frac{1}{2} + \frac{1}{2^{k+1}-2}\right)\cdot 2^n + O(1).$$

An *upward reduction* of a permutation π, $|\pi| = n$, is a partition $[n] = [1,r]\cup[r+1,n]$, where $1 \leq r < n$, such that $\pi([1,r]) < \pi([r+1,n])$. If π has no upward reduction, we say that π is *upward irreducible*. The set Irr^+ consists of all upward irreducible permutations and $\mathrm{Irr}^+_n = \{\pi \in \mathrm{Irr}^+ : \ |\pi| = n\}$. Every permutation π of $[n]$ has a unique decomposition $\pi|I_1, \ldots, \pi|I_m$, called the *upward decomposition* of π, in which $I_1 < I_2 < \ldots < I_m$ are intervals partitioning $[n]$ such that $\pi(I_1) < \pi(I_2) < \ldots < \pi(I_m)$ and every restriction $\pi|I_i$ is upward irreducible. (This decomposition can be obtained by iterating the upward reductions). We call the permutations $\pi|I_i$ the *upward blocks* of π. Notions symmetric to these are obtained in the obvious way, replacing the appropriate signs $<$ by the opposite signs $>$. Thus we get the definitions of *downward reductions*, *downward irreducibility*, *downward decompositions*, *downward blocks*, and the sets Irr^- and Irr^-_n.

We prove that one can delete an entry from any upward irreducible permutation in such a way that the result is upward irreducible. Needless to say, the same holds for downward irreducible permutations.

Lemma 3.7 *For every $\pi \in \mathrm{Irr}^+_n$, $n > 1$, there is some $i \in [n]$ such that $\pi|([n]\backslash\{i\})$ is in Irr^+_{n-1}.*

Proof. For a permutation π of $[n]$ and $i \in [n]$ we say that i is a *record* of π if $\pi(j) < \pi(i)$ for every $j < i$. Let $1 = r_1 < r_2 < \ldots < r_m \leq n$ be the records of π. It is easy to see that π is upward irreducible if and only if for every $i = 1, 2, \ldots, m-1$ there is a j, $r_{i+1} < j \leq n$, with $\pi(j) < \pi(r_i)$. Suppose that π, $|\pi| = n \geq 2$, is upward irreducible and consider the set $A = \{i \in [n] : \ r_m < i \leq n \ \& \ \pi(i) > \pi(r_{m-1})\}$; if $m = 1$, we set $A = [2, n]$. If $A \neq \emptyset$, the deletion of any $i \in A$ leaves an upward irreducible permutation. If $A = \emptyset$, we delete $i = r_m$. $\Box$

We remark that it is easy to find examples of permutations of arbitrary length such that the statement of Lemma 3.7 is satisfied with only two indices i.

If π is a permutation, $|\pi| = n$, and $I_1 < I_2 < \ldots < I_m$ is a partition of $[n]$ into m nonempty intervals, we associate with π (as in the sketched proof of Theorem 3.4) the word $u(\pi) = u_1, u_2, \ldots, u_n$ over the alphabet $[m]$ by setting $u_i = j$ if $\pi^{-1}(i) \in I_j$. Note that π is uniquely determined by $u(\pi)$ and the m restrictions $\pi|I_i$.

Also, we associate with π the word $v^+(\pi)$ over the alphabet Irr^+ describing the upward decomposition of π. By $h^+(\pi) \in \mathbf{N}$ we denote the maximum size of an upward block of π appearing in the upward decomposition of π. Thus if $h^+(\pi) = k$ then $v^+(\pi) \in (\mathrm{Irr}_1^+ \cup \ldots \cup \mathrm{Irr}_k^+)^*$. In the analogous way we define $v^-(\pi)$ and $h^-(\pi)$.

Theorem 3.8 *Let Π be any hereditary class of permutations. Then either Π is finite, or exactly one of the following possibilities holds.*

(1) There is a unique $k \geq 1$ and a constant $c > 0$ such that $F_{n,k} \leq |\Pi_n| \leq F_{n,k} \cdot n^c$ for all $n \geq 1$.

(2) $|\Pi_n| \geq 2^{n-1}$ for all $n \geq 1$.

Proof. The *k-decomposition*, where $k \geq 2$ is an integer, of a permutation π, $|\pi| = n$, is the unique partition of $[n]$ into the intervals $U_1 < U_2 < \ldots < U_m$ such that U_1 is the longest initial interval of $[n]$ with $h^+(\pi|U_1) < k$ or $h^-(\pi|U_1) < k$, U_2 is the longest following interval with the same property, etc. We call the intervals U_i the *k-segments* of π. The number m of k-segments of π is denoted by $s_k(\pi)$.

Let Π be an infinite hereditary permutation class. Let $s_k(\Pi) = \max\{s_k(\pi) : \ \pi \in \Pi\}$. We set $s_1(\Pi) = \infty$. For every fixed $k \geq 1$ we prove the following claims.

Claim A. If $s_k(\Pi) = \infty$ then $|\Pi_n| \geq F_{n,k}$ for every $n \geq 1$.

Claim B. If $s_k(\Pi) < \infty$ then either $|\Pi_n| \geq 2^{n-1}$ for every $n \geq 1$ or $|\Pi_n| \leq F_{n,k-1} \cdot c_1 n^{c_2}$ for every $n \geq 1$ and some constants $c_1, c_2 > 0$.

This will prove the theorem. To see this, note that either $s_k(\Pi) = \infty$ for every $k \geq 1$ or there is a $k \geq 1$ such that $s_k(\Pi) = \infty$ but $s_{k+1}(\Pi) < \infty$. In the former case, claim A implies that $|\Pi_n| \geq F_{n,n} = 2^{n-1}$ for every $n \geq 1$ (by (4) of Lemma 3.5). In the latter case, we apply claim A with k and claim B with $k+1$ and conclude that either again $|\Pi_n| \geq 2^{n-1}$ for every $n \geq 1$ or that $F_{n,k} \leq |\Pi_n| \leq F_{n,k} \cdot n^c$ for every $n \geq 1$ (c_1 was absorbed in the enlarged c_2).

Proof of Claim A. For a $\pi \in \Pi$ with the k-segments $U_1 < U_2 < \ldots < U_{s_k(\pi)}$, we set $T_{\pi,i} = [\min(\pi(U_iU_{i+1})), \max(\pi(U_iU_{i+1}))]$. Note that, by the definition of k-segments and Lemma 3.7, every restriction $\pi|U_iU_{i+1}$ contains a member of Irr_k^+ and a member of Irr_k^-. We consider the system of intervals

$$S(\pi) = \{T_{\pi,1}, T_{\pi,3}, T_{\pi,5}, \ldots, T_{\pi,r}\}, \quad \text{where } r = 2\lceil (s_k(\pi) - 1)/2 \rceil - 1.$$

By the Ramsey theorem, either for every $m \geq 1$ there is a $\pi \in \Pi$ such that $S(\pi)$ contains m mutually intersecting intervals or for every $m \geq 1$ the same holds with mutually disjoint intervals. In the former case, it is easy to see that π must have at least $m/2$ alternations, since all members of a system of mutually intersecting intervals must have a point in common. By Lemma 3.2 and (4) of Lemma 3.5, $|\Pi_n| \geq 2^{n-1} \geq F_{n,k}$ for every $n \geq 1$.

In the latter case, for every $m \geq 1$ there is a $\pi \in \Pi$ for which $[|\pi|]$ can be partitioned into m intervals $I_1 < I_2 < \ldots < I_m$ such that every restriction $\pi|I_i$ contains a member of Irr_k^+ and a member of Irr_k^-, and for every $i \neq j$ we have $\pi(I_i) > \pi(I_j)$ or $\pi(I_i) < \pi(I_j)$. By Theorem 3.1, we may assume that $\pi(I_1) < \pi(I_2) < \ldots < \pi(I_m)$ or $\pi(I_1) > \pi(I_2) > \ldots > \pi(I_m)$. Let $\pi(I_1) < \pi(I_2) < \ldots < \pi(I_m)$; the other case is similar. Since m may be arbitrarily large and $|\mathrm{Irr}_k^+| \leq k!$, we may use the pigeon-hole principle and assume that there is one fixed $\sigma \in \mathrm{Irr}_k^+$ that is contained, for every $m \geq 1$, in every $\pi|I_i$, $1 \leq i \leq m$. By Lemma 3.7, there is a set of permutations $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ such that $\sigma_i \in \mathrm{Irr}_i^+$, $\sigma_i \subset \sigma_{i+1}$, and $\sigma_k = \sigma$. Since Π is hereditary, for every word u over the alphabet Σ there is a $\tau \in \Pi$ (contained in π) such that $v^+(\tau) = u$. Clearly, different words u determine different permutations τ. Using (1) of Lemma 3.6 (where the weight function is $w(\sigma_i) = |\sigma_i| = i$), we conclude that $|\Pi_n| \geq F_{n,k}$ for every $n \geq 1$. This finishes the proof of Claim A.

Proof of Claim B. We have $k \geq 2$ and there is a constant K such that $s_k(\pi) \leq K$ for every $\pi \in \Pi$. If $\pi \in \Pi_n$ and $U_1 < U_2 < \ldots < U_{s_k(\pi)}$ is the partition of $[n]$ into the k-segments of π, we consider the word $u(\pi)$ over $[K]$ as defined above the theorem.

For $1 \leq i \leq s_k(\pi)$ and $1 \leq j \leq k-1$ we define $v_{i,j}^+(\pi)$ as the subword of $v^+(\pi|U_i)$ consisting of the occurrences of the letters from the subalphabet Irr_j^+. The word $v_{i,j}^-(\pi)$ is defined in the obvious symmetric manner. Recall that $\ell(u)$ is the length of the longest alternating subsequence of u. Let $\ell(u(\Pi)) = \max\{\ell(u(\pi)) : \pi \in \Pi\}$ and, for $1 \leq i \leq K$ and $1 \leq j \leq k-1$, $\ell(v_{i,j}^+(\Pi)) = \max\{\ell(v_{i,j}^+(\pi)) : \pi \in \Pi\}$. The quantity $\ell(v_{i,j}^-(\Pi))$ is defined analogously. (For $i > s_k(\pi)$ we set $\ell(v_{i,j}^+(\pi)) = \ell(v_{i,j}^-(\pi)) = 0$.) We distinguish two complementary cases.

Case B1. *One of the* $2K(k-1)+1$ *quantities* $\ell(u(\Pi))$, $\ell(v_{i,j}^+(\Pi))$, *and* $\ell(v_{i,j}^-(\Pi))$ *equals* ∞.

We prove that then always

$$|\Pi_n| \geq 2^{n-1} \quad \text{for all} \quad n \geq 1.$$

For unbounded $\ell(u(\pi))$ we can find a $\pi \in \Pi$ with as many alternations in π^{-1} as we wish and thus $|\Pi_n| \geq 2^{n-1}$ for every $n \geq 1$ by Lemma 3.2. For unbounded $\ell(v_{i,j}^+(\pi))$ (the

argument for $v^-_{i,j}(\pi)$ is the same) there is a $j \in [k-1]$ (in fact, necessarily $j \in [3, k-1]$) and two distinct permutations $\tau, \sigma \in \mathrm{Irr}^+_j$ such that for every alternating word v over $\{\sigma, \tau\}$ there is a $\pi \in \Pi$ with $v^+(\pi) = v$. Using Lemma 3.7 again, we can take a set of permutations $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_j\}$ such that $\sigma_i \in \mathrm{Irr}^+_i$, $\sigma_i \subset \sigma_{i+1}$, and $\sigma_j = \sigma$. By the hereditarity of Π, for every word v over the alphabet $\Sigma \cup \{\tau\}$ there is a $\pi \in \Pi$ with $v^+(\pi) = v$. By (2) of Lemma 3.6, $|\Pi_n| \geq 2^{n-1}$ for every $n \geq 1$. This finishes the proof of case B1.

Case B2. *There is a constant $L > 0$ such that $\ell(u(\Pi)) \leq L$ and $\ell(v^+_{i,j}(\Pi)) \leq L$, $\ell(v^-_{i,j}(\Pi)) \leq L$ for every $1 \leq i \leq K, 1 \leq j \leq k-1$.*

We prove the upper bound

$$|\Pi_n| \leq F_{n,k-1} \cdot c_1 n^{c_2} \quad \text{for all} \quad n \geq 1.$$

Every $\pi \in \Pi_n$ is uniquely determined by the word $u(\pi) \in [K]^*$ together with the $s_k(\pi) \leq K$ restrictions $\pi|U_i$. For $s_k(\pi) < i \leq K$ we set $U_i = \emptyset$. Let $R(m)$ be the number of possible restrictions $\pi|U_i$ with $|U_i| = m$. If we prove that $R(m) \leq F_{m,k-1} \cdot c_3 m^{c_4}$ for all $m \geq 1$ and constants $c_3, c_4 > 0$ (depending only on K and L), we are done since (3) of Lemma 3.5 and (2) of Lemma 3.3 imply that

$$\begin{aligned}
|\Pi_n| &\leq \sum_{u \in W_{K,L,n}} R(|U_1|)R(|U_2|)\ldots R(|U_K|) \quad (\text{note that } |U_1| + \cdots + |U_K| = n) \\
&\leq \sum_{u \in W_{K,L,n}} F_{n,k-1} \cdot c_3^K n^{c_4 K} \\
&\leq F_{n,k-1} \cdot c_3^K n^{c_4 K} \cdot (K+1)^{c_5} n^{c_5} \quad \left(\text{where } \; c_5 = \binom{K}{2}(L-1)+1\right) \\
&\leq F_{n,k-1} \cdot c_1 n^{c_2}.
\end{aligned}$$

It remains to show that $R(m) \leq F_{m,k-1} \cdot c_3 m^{c_4}$. Let σ be a generic restriction $\pi|U_i$ with $|U_i| = m$ and π ranging over Π_n. We have that $h^+(\sigma) < k$ or $h^-(\sigma) < k$; we may assume the former. For $1 \leq j \leq k-1$ we write $v^+_j(\sigma)$ for $v^+_{i,j}(\pi)$. By the hypothesis of case B2, $\ell(v^+_j(\sigma)) \leq L$ for every $1 \leq j \leq k-1$. Since $v^+(\sigma) \in (\mathrm{Irr}^+_1 \cup \ldots \cup \mathrm{Irr}^+_{k-1})^*$, we apply (3) of Lemma 3.3 and conclude that there is a partition into intervals

$$v^+(\sigma) = J_1 J_2 \ldots J_M \quad \text{with} \quad M \leq 2\binom{1! + \cdots + (k-1)!}{2}(L-1) + 2 = N$$

such that every J_i uses at most one letter from any subalphabet Irr^+_j. Let $Q(m)$ be the number of $\tau \in \Pi_m$ such that $h^+(\tau) < k$ and $v^+(\tau)$ uses at most one letter from every Irr^+_j. Then, by (1) of Lemma 3.6,

$$Q(m) \leq 1! \cdot 2! \cdot \ldots \cdot (k-1)! \cdot F_{m,k-1} = c_6 \cdot F_{m,k-1}$$

because $v^+(\tau) \in \Sigma^*$ for a transversal Σ of the $k-1$ sets $\mathrm{Irr}^+_1, \ldots, \mathrm{Irr}^+_{k-1}$, and there are at most c_6 such transversals.

So, by the bound on $Q(m)$, (3) of Lemma 3.5, and the bound $M \leq N$, the number of σs satisfies

$$\begin{aligned} R(m) &\leq \sum_{m_1+\cdots+m_N=m} Q(m_1)Q(m_2)\ldots Q(m_N) \quad \text{(summing over } m_i \geq 0) \\ &\leq \sum_{m_1+\cdots+m_N=m} F_{m,k-1} \cdot c_6^N \leq F_{m,k-1} \cdot c_6^N (m+1)^N \\ &\leq F_{m,k-1} \cdot c_3 m^{c_4}. \end{aligned}$$

This finishes the proof of case B2, of claim B, and of the whole theorem. □

The growth rate $n \mapsto F_{n,k}$ is attained by the hereditary class of permutations π whose upward blocks are decreasing sequences of length at most k.

Let $a_n = |\mathrm{Irr}_n^+| = |\mathrm{Irr}_n^-|$. It follows from the upward decompositions that the numbers a_n of upward irreducible permutations satisfy the recurrence $a_n = n! - \sum_{k=1}^{n-1} a_k(n-k)!$ where $n \geq 2$ and $a_1 = 1$. From this we calculate that

$$(|\mathrm{Irr}_n^+|)_{n\geq 1} = (1, 1, 3, 13, 71, 461, 3447, 29093, 273343, 2829325, \ldots).$$

This is sequence A003319 of Sloane [27]. It appears three times in Comtet's textbook [14, pp. 84, 262, and 295]. Since

$$1 \geq \frac{a_n}{n!} = 1 - \sum_{k=1}^{n-1} \frac{a_k}{k!} \cdot \frac{1}{\binom{n}{k}} \geq 1 - \left(\frac{2}{n} + \frac{2(n-3)}{n(n-1)}\right) = 1 - \frac{4n-8}{n(n-1)}$$

for every $n \geq 3$, we conclude that $a_n = |\mathrm{Irr}_n^+| = |\mathrm{Irr}_n^-| = n!(1 - O(1/n))$. See [14, p. 295] for a more precise asymptotic expansion. Thus almost every permutation is upward irreducible (downward irreducible). For $|\mathrm{Irr}_n^+ \cap \mathrm{Irr}_n^-|$ we have the same asymptotics, because $|\mathrm{Irr}_n^+ \cap \mathrm{Irr}_n^-| = 2a_n - n!$. It follows that almost every permutation is both upward irreducible and downward irreducible.

4 Constant and polynomial growths

We look in more detail on the slow growths and begin with the constant growth. Let π be a permutation of $[n]$. For $r \in \mathbf{N}$, we say that π has the *r-intrusion property* if there are subsets $X, Y \subset [n]$ and an element $x \in [n]$ such that $X < x < Y$, $|X|, |Y| \geq r$, and $\pi|(X \cup Y)$ is monotone but $\pi|(X \cup Y \cup \{x\})$ is not. We say that π has the *r-union property* if there are subset $X, Y \subset [n]$ such that $X < Y$, $|X|, |Y| \geq r$, and both restrictions $\pi|X$ and $\pi|Y$ are monotone but $\pi|(X \cup Y)$ is not.

Lemma 4.1 *Let Π be any hereditary class of permutations.*

(1) If for every $r \geq 1$ there is a $\pi \in \Pi$ such that π or π^{-1} has the r-intrusion property, then $|\Pi_n| \geq n$ for all $n \geq 1$.

(2) If for every $r \geq 1$ there is a $\pi \in \Pi$ with the r-union property, then $|\Pi_n| \geq n$ for all $n \geq 1$.

(3) Suppose $\pi \neq \tau$ are two permutations of $[n]$ and $I \subset [n]$ is such that all three sets $I, \pi(I)$, and $\tau(I)$ are intervals in $[n]$ and both restrictions $\pi|I$ and $\tau|I$ are monotone. Then for every subset $J \subset I$ such that $|I| - |J| \geq 2$ we have $\pi|([n]\backslash J) \neq \tau|([n]\backslash J)$.

Proof. (1) We may assume that for every $r \geq 1$ there is a $\pi \in \Pi_{2r+1}$ such that $\pi|([2r+1]\backslash\{r+1\})$ is increasing but $\pi(r) > \pi(r+1)$; the other possible cases are very similar. Thus for every n and m, $1 \leq m \leq n-1$, there is a $\pi_m \in \Pi_n$ such that $\pi|[m]$ is increasing but $\pi(m) > \pi(m+1)$. The permutations π_m are mutually distinct and together with $(1, 2, \ldots, n) \in \Pi_n$ they show that $|\Pi_n| \geq n$.

(2) If π, $|\pi| = 2n$, is such that $\pi|[n]$ and $\pi|[n+1, 2n]$ are monotone but π is not, then $\pi(n-1), \pi(n), \pi(n+1)$ or $\pi(n), \pi(n+1), \pi(n+2)$ is non-monotone. ¿From this it easily follows, as in (1), that there are n distinct permutations σ, $|\sigma| = n$, such that $\sigma \subset \pi$. Thus $|\Pi_n| \geq n$ for all $n \geq 1$.

(3) The restrictions of π and τ on $[n]\backslash J$ must be different because at least 2 terms remained from the monotone sequences $\pi|I$ and $\tau|I$ and thus π and τ can be completely reconstructed from the restrictions. □

Theorem 4.2 *Let Π be any hereditary class of permutations. Then exactly one of the following possibilities holds.*

(1) $|\Pi_n|$ is constant for $n \geq n_0$.

(2) $|\Pi_n| \geq n$ for all $n \geq 1$.

Proof. We may assume that Π is a hereditary permutation class such that, for some $r \geq 1$, for every $\pi \in \Pi$ neither π nor π^{-1} has the r-intrusion property and π does not have the r-union property. If r does not exist, we are done because (1) and (2) of Lemma 4.1 then imply that (2) holds. Now let $\pi \in \Pi_n$ be arbitrary and let $n \geq 9r^2$. Consider the longest monotone subsequence of π, determined by $X \subset [n]$. Thus $\pi|X$ is monotone, $|X|$ is maximum, and, by Theorem 3.1, $|X| \geq 3r$. We partition X into the sets $X_1 < Y < X_2$ where $|X_1| = |X_2| = r$. So $|Y| \geq r$. Since neither π nor π^{-1} has the r-intrusion property and $|X|$ is maximum, both Y and $\pi(Y)$ must form an interval in $[n]$. Let $A = [\min Y - 1]\backslash X_1$ and $B = [\max Y + 1, n]\backslash X_2$. Using the assumption that π does not have the r-union property and invoking Theorem 3.1, we see that $|A|, |B| \leq r^2$. We conclude that for every $\pi \in \Pi_n$, $n \geq 9r^2$, there is a subset $Y \subset [n]$ such that $\pi|Y$ is monotone, both Y and $\pi(Y)$ form an interval in $[n]$, and $n - |Y| \leq 2(r^2 + r) = R$. If R is enlarged to $R = 9r^2$, the conclusion holds for every $n \geq 1$.

For all $n \geq 1$, $|\Pi_n| \leq 2(R+1)^2 R!$ because we have two possibilities for $\pi|Y$, at most $R+1$ ways to place Y, the same number of ways to place $\pi(Y)$, and at most $R!$ possibilities for $\pi|([n]\backslash Y)$. Let

$$k = \limsup_{n \to \infty} |\Pi_n|$$

and $n_0 \in \mathbf{N}$ be such that $n_0 \geq 2R+2$ and $|\Pi_n| \leq k$ for $n \geq n_0$. We show that in fact, $|\Pi_n| = k$ for $n \geq n_0$. Let $n \geq n_0$ be arbitrary, $m \geq n$ be such that $|\Pi_m| = k$, and let $\Pi_m = \{\pi_1, \pi_2, \ldots, \pi_k\}$. These k permutations satisfy the hypothesis of (3) of Lemma 4.1 with $I = [R+1, m-R]$. For $J = [n-R+1, m-R]$ we have $J \subset I$, $|I| - |J| \geq 2$, and $\sigma_i = \pi_i|([m]\backslash J) \in \Pi_n$ for $1 \leq i \leq k$. By (3) of Lemma 4.1, all σ_i are distinct. Hence $|\Pi_n| \geq k$ and $|\Pi_n| = k$. □

All possible constant growths are attained. The growth $n \mapsto 0$ is attained by $\Pi = \emptyset$ and $n \mapsto k$, $n \geq k$, is attained by $\{\pi \subset (1, 2, \ldots, n, n+k, n+k-1, \ldots, n+1) : \ n \in \mathbf{N}\}$. Similarly, $n \mapsto n$ is attained by $\{\pi \subset (1, 2, \ldots, n, 2n, 2n-1, \ldots, n+1) : n \in \mathbf{N}\}$.

We proceed to the polynomial growth and partially characterize polynomially growing counting functions of hereditary permutation classes. For this, we need to look first at the partial order $(\mathbf{N}^m, \leq)$, where $m \in \mathbf{N}$ and $a = (a_1, \ldots, a_m) \leq (b_1, \ldots, b_m) = b$ means that $a_i \leq b_i$ for every $i = 1, \ldots, m$. We say that a subset $S \subset \mathbf{N}^m$ is *hereditary* if $a \leq b \in S$ always implies $a \in S$. For $a \in \mathbf{N}^m$ we define $\|a\| = a_1 + a_2 + \cdots + a_m$. For a (hereditary) subset $S \subset \mathbf{N}^m$ and $n \in \mathbf{N}$ we set

$$S_n = \{a \in S : \ \|a\| = n\}.$$

The elements of S_n can be represented by partitions of $[n]$ into m nonempty intervals, and therefore the next lemma is a particular case of Theorem 3.1 in [17]. However, the direct proof is not too difficult; and for the sake of completeness, we give it here.

Lemma 4.3 *For every hereditary set $S \subset \mathbf{N}^m$ there is a number $M \in \mathbf{N}$ and $(M+1)^2$ integers $a_{i,j}$, $0 \leq i, j \leq M$, so that for every $n \geq n_0$ we have*

$$|S_n| = \sum_{i,j=0}^{M} a_{i,j} \binom{n-i}{j}.$$

Proof. We say that $S \subset \mathbf{N}^m$ is *canonical* if there is an m-tuple $b \in (\mathbf{N} \cup \{\infty\})^m$ so that

$$S = \{a \in \mathbf{N}^m : \ a_i < b_i \text{ for every } i = 1, \ldots, m\},$$

where $a_i < \infty$ means that the i-th coordinate of a is unrestricted. Canonical sets are hereditary and for a canonical S determined by b we have

$$|S_n| = [x^n] \left(\frac{x}{1-x}\right)^{\#(b_i=\infty)} \prod_{b_i<\infty} \sum_{j=1}^{b_i-1} x^j \quad \text{for } n \geq 1,$$

where the empty sum equals 0. If $b_i = \infty$ for no i, then $|S_n| = 0$ for $n \geq n_0 = \|b\| - m + 1$, and the formula is true with all $a_{i,j}$ zero. Otherwise, the formula follows by the binomial theorem. Thus the lemma holds in the case when S is a canonical set, even with nonnegative $a_{i,j}$. It is clear that every intersection of canonical sets is again a canonical set. Therefore, by the inclusion-exclusion principle, the lemma holds more generally for every finite union of canonical sets (now we may get negative $a_{i,j}$).

Let $S \subset \mathbf{N}^m$ be any hereditary set. It suffices to show that S is a finite union of canonical sets. S is determined by the set B of minimal elements in $(\mathbf{N}^m \backslash S, \leq)$. The set B is an antichain. It is known, and not too difficult to prove, that every antichain in $(\mathbf{N}^m, \leq)$ is finite (this is the combinatorial core of Hilbert's basis theorem), see (for example) Nash-Williams [23, Lemma 1] for a more general result. So $B = \{b^1, b^2, \ldots, b^r\}$ and

$$S = \{a \in \mathbf{N}^m : \ a \not\geq b^i \text{ for every } i = 1, \ldots, r\}.$$

Thus

$$a \in S \iff \bigwedge_{i=1}^{r} \bigvee_{j=1}^{m} (a_j < b^i_j).$$

The right hand side of the equivalence is equivalent to

$$\bigvee_{j_1,\ldots,j_r} \bigwedge_{i=1}^{r} (a_{j_i} < b^i_{j_i})$$

where in the disjunction the j_i's range over all r-tuples from $[m]$. Since every conjuction $\bigwedge_{i=1}^{r}(a_{j_i} < b^i_{j_i})$ defines a canonical set, S is indeed a union of m^r canonical sets and the lemma follows. □

A *canonical form* of a word u is the form $u = u_1^{a_1}, u_2^{a_2}, \ldots, u_m^{a_m}$ where $a_i \in \mathbf{N}$, $u_i \neq u_{i+1}$ for $i = 1, 2, \ldots, m-1$, and $u_i^{a_i}$ abbreviates a_i repetitions of the letter u_i. Let $u(\pi) = u_1, u_2, \ldots, u_n$ be the word over $[m]$ determined by the 2-decomposition $U_1 < U_2 < \ldots < U_m$ of a permutation π of length n; $u_i = j \Leftrightarrow \pi^{-1}(i) \in U_j$ and every $\pi|U_i$ is monotone (see the proofs of Theorems 3.4 and 3.8). Recall that π is uniquely determined by $u(\pi)$ and the m-tuple $s(\pi) \in \{+,-\}^m$ whose i-th component records whether $\pi|U_i$ is increasing or decreasing. Every $i < m$ appears in $u(\pi)$ at least twice and $i = m$ may appear only once. Let $u(\pi) = w_1^{a_1}, w_2^{a_2}, \ldots, w_r^{a_r}$ be the canonical form of $u(\pi)$. The *reduced* form of $u(\pi)$ is the word

$$u(\pi)^* = w = w_1^{b_1}, w_2^{b_2}, \ldots, w_r^{b_r}$$

where $b_i = 1$ if $w_i = m$ or if $w_j = w_i$ for some $j \neq i$, else $b_i = 2$. We define $E(w) = \{i \in [r] : \ b_i = 2\}$ and $e(\pi) \in \mathbf{N}^r$ by $e_i = a_i$ if $i \notin E(w)$ and $e_i = a_i - 1$ if $i \in E(w)$.

Let Π be any hereditary permutation class. We split Π into (nonhereditary) subclasses by the equivalence $\sim$: $\pi \sim \sigma$ iff $u(\pi)^* = u(\sigma)^* = w = w_1^{b_1}, w_2^{b_2}, \ldots, w_r^{b_r}$ and $s(\pi) = s(\sigma)$. In one equivalence subclass X, the permutations are fully determined by the r-tuples

$$e(X) = \{e(\pi) : \ \pi \in X\} \subset \mathbf{N}^r.$$

Note that $e(X)$ is hereditary (in the sense of the previous lemma) and that, for $\pi \in X$, $|\pi| = \|e(\pi)\| + |E(w)|$. By Lemma 4.3, there are $(M+1)^2$ integers $a_{i,j}$ such that

$$|X_n| = \#\{\pi \in X : \ |\pi| = n\} = \sum_{i,j=0}^{M} a_{i,j} \binom{n-i}{j}$$

for all $n \geq n_0$.

Theorem 4.4 *If Π is a hereditary class of permutations such that $|\Pi_n| \leq n^c$ for all $n \geq 1$ and a constant $c > 0$, then there is a number $M \in \mathbf{N}$ and $(M+1)^2$ integers $a_{i,j}$, $0 \leq i, j \leq M$, such that for all $n \geq n_0$ we have*

$$|\Pi_n| = \sum_{i,j=0}^{M} a_{i,j} \binom{n-i}{j}.$$

Proof. The proof of Theorem 3.8 (case B2, $k = 2$) shows that if $|\Pi_n| \leq n^c$ for all $n \geq 1$, then for all $\pi \in \Pi$ we have $m = s_2(\pi) \leq K$ and $\ell(u(\pi)) \leq L$ for some constants $K, L > 0$. Thus by (1) of Lemma 3.3, the length of the reduced form of $u(\pi)$ is bounded by some $d > 0$, and we have at most $(K+1)^d$ possible reduced forms $u(\pi)^*, \pi \in \Pi$. Hence the equivalence $\sim$ on Π has at most $2^K(K+1)^d$ subclasses. We select n_0 large enough so that for every of the subclasses X the above argument applies and for $n \geq n_0$ the number $|X_n|$ has the form $\sum_{i,j=0}^{M} a_{i,j}\binom{n-i}{j}$, with integral $a_{i,j}$ (M and $a_{i,j}$ depend on X). Then for $n \geq n_0$ also the finite sum

$$|\Pi_n| = \sum_{X} |X_n|$$

has the form $\sum_{i,j=0}^{M} a_{i,j}\binom{n-i}{j}$. □

It is an interesting question to fully characterize those polynomials that can be realized (for $n \geq n_0$) as $n \mapsto |\Pi_n|$. For example, it follows from Theorem 4.2 that no polynomial $\binom{n-i}{1} = n - i$, $i \in \mathbf{N}$, can be realized as a counting function. Note that a part of Theorem 4.2, the fact that every bounded counting function must be eventually constant, is an immediate corollary of the last theorem.

5 Concluding remarks

The fact that the containment order of permutations admits an infinite antichain has been known for a long time. The earliest references are Laver [21, p. 9], Pratt [24], and Tarjan [31]. Kruskal [20, p. 304] mentions Laver's (counter)example four years earlier and Laver himself "[uses] a construction of Jenkyns and Nash-Williams" [16]. Thus the idea seems to go back to the late 1960s. The recent reference is Spielman and Bóna [28]. See Atkinson, Murphy and Ruškuc [6] for further results on permutation antichains.

Hereditary classes of permutations and their counting functions have been investigated before by Atkinson [5] who, together with West [33], gives the counting function $n \mapsto |\Pi_n|$ for every hereditary Π of the form $\Pi = \mathrm{Forb}(\{\alpha, \beta\})$ where $|\alpha| = 3, |\beta| = 4$, and $\alpha \not\subset \beta$. Our approach is much inspired by the works of Scheinerman and Zito [26] and Balogh, Bollobás and Weinreich [7, 8, 9] on the hereditary classes and monotone classes of graphs. (For graphs, hereditarity means that the class is closed with respect to induced subgraphs, while monotonicity means that it is closed with respect to all subgraphs.) As far as we know, graphs are the only combinatorial structures for which the counting functions of hereditary classes have been systematically investigated from a 'global' viewpoint. One global result (although cast in the 'local' $\mathrm{Forb}(X)$ language) on hereditary classes of set

partitions is in Klazar [17, Theorem 3.1]. The counting functions of the hereditary classes of set partitions are further investigated in [19].

The question posed after Theorem 2.1, whether there are $2^{\aleph_0}$ hereditary permutation classes with counting functions mutually incomparable by the eventual dominance, has a positive answer for graph hereditary classes, see [8, Theorem 10]. Note also that if we restrict our attention to polynomially growing permutation classes, then by Theorem 4.4, the eventual dominance is a linear order.

References

[1] R. M. Adin and Y. Roichman, Shape avoiding permutations, *J. Comb. Theory, Ser. A*, **97** (2002), 162–176.

[2] M. H. Albert, M. D. Atkinson, C. C. Handley, D. A. Holton and W. Stromquist, On packing densities of permutations, *Electr. J. Comb.*, **9** (2002), R5, 20 pages.

[3] N. Alon and E. Friedgut, On the number of permutations avoiding a given pattern, *J. Comb. Theory, Ser. A*, **89** (2000), 133–140.

[4] R. Arratia, On the Stanley–Wilf conjecture for the number of permutations avoiding a given pattern, *Electr. J. Comb.*, **6** (1999), N1, 4 pages.

[5] M. D. Atkinson, Restricted permutations, *Discrete Math.*, **195** (1999), 27–38.

[6] M. D. Atkinson, M. M. Murphy and N. Ruškuc, Partially well-ordered sets of permutations, to appear in *Journal of Order*.

[7] J. Balogh, B. Bollobás and D. Weinreich, The speed of hereditary properties of graphs, *J. Comb. Theory, Ser. B*, **79** (2000), 131–156.

[8] J. Balogh, B. Bollobás and D. Weinreich, The penultimate rate of growth for graph properties, *Eur. J. Comb.*, **22** (2001), 277–289.

[9] J. Balogh, B. Bollobás and D. Weinreich, Measures on monotone properties of graphs, *Discrete Appl. Math.*, **116** (2002), 17–36.

[10] M. Bóna, *Exact and asymptotic enumeration of permutations with subsequence conditions*, Ph.D. thesis, M.I.T, 1997.

[11] M. Bóna, Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps, *J. Comb. Theory, Ser. A*, **80** (1997), 257–272.

[12] M. Bóna, Permutations avoiding certain patterns: the case of length 4 and some generalizations, *Discrete Math.*, **175** (1997), 55–67.

[13] M. Bóna, The solution of a conjecture of Stanley and Wilf for all layered patterns, *J. Comb. Theory, Ser. A*, **85** (1999), 96–104.

[14] L. Comtet, *Advanced Combinatorics*, D. Reidel, Dordrecht 1974.

[15] H. Davenport and A. Schinzel, A combinatorial problem connected with differential equations, *Amer. J. Math.*, **87** (1965), 684–694.

[16] T. A. Jenkyns and C. St. J. A. Nash-Williams, Counterexamples in the theory of well-quasi-ordered sets, *Proof Techniques in Graph Theory, Proc. 2nd Ann Arbor Graph Theory Conference, 1968*, Academic Press, New York 1969; pp. 87–91.

[17] M. Klazar, Counting pattern-free set partitions I: A generalization of Stirling numbers of the second kind, *Eur. J. Comb.*, **21** (2000), 367–378.

[18] M. Klazar, The Füredi–Hajnal conjecture implies the Stanley–Wilf conjecture, *Proc. 12th international conference FPSAC'00, Moscow, 2000*, Springer, Berlin 2000; pp. 250–255.

[19] M. Klazar, Counting pattern-free set partitions III: Growth rates of the hereditary classes, in preparation.

[20] J. B. Kruskal, The theory of well-quasi-ordering: A frequently discovered concept, *J. Comb. Theory, Ser. A*, **13** (1972), 297–305.

[21] R. Laver, Well-quasi-orderings and sets of finite sequences, *Math. Proc. Camb. Philos. Soc.*, **79** (1976), 1–10.

[22] L. Lovász, *Combinatorial Problems and Exercises*, Akadémiai Kiadó, Budapest 1979.

[23] C. St. J. A. Nash-Williams, On well-quasi-ordering finite trees, *Proc. Camb. Philos. Soc.*, **59** (1963), 833–835.

[24] V. R. Pratt, Computing permutations with double-ended queues, parallel stacks, and parallel queues, *Proc. 5th Annual ACM Symposium on Theory of Computing*, Assoc. Comput. Mach., New York 1973; pp. 268–277.

[25] A. Regev, Asymptotic values for degrees associated with strips of Young diagrams, *Adv. Math.*, **41** (1981), 115–136.

[26] E. R. Scheinerman and J. Zito, On the size of hereditary classes of graphs, *J. Comb. Theory, Ser. B*, **61** (1994), 16–39.

[27] N. J. A. Sloane, editor (2002), The On-Line Encyclopedia of Integer Sequences, published electronically at
`http://www.research.att.com/~njas/sequences/`.

[28] D. Spielman and M. Bóna, An infinite antichain of permutations, *Electr. J. Comb.*, **7** (2000), N2, 4 pages.

[29] Z. Stankova-Frenkel and J. West, A new class of Wilf-equivalent permutations, to appear in *Journal of Algebraic Combinatorics*.

[30] R. P. Stanley, *Enumerative Combinatorics, Volume 1*, Wadsworth & Brooks/Cole, Monterey, Ca 1986.

[31] R. E. Tarjan, Sorting using networks of queues and stacks, *J. Assoc. Comput. Mach.*, **19** (1972), 341–346.

[32] P. Valtr, private communication, January 2000.

[33] J. West, Generating trees and forbidden sequences, *Dicrete Math.*, **157** (1996), 363–374.

Strukturuntersuchungen für Shop-Scheduling-Probleme:

Anzahlprobleme, potentielle Optimalität und neue Enumerationsalgorithmen

DISSERTATION

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.),

genehmigt durch
die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

von Diplommathematiker Martin HARBORTH

geb. am:	21. Oktober 1968
in	Braunschweig

Gutachter:	Prof. Dr. Heidemarie Bräsel
	Prof. Dr. Johannes Terno
	Prof. Dr. Peter Brucker

Eingereicht am:	21.05.1999
Verteidigung am:	31.08.1999

Abstract

Structural analysis of shop scheduling problems: number problems, potential optimality, and new enumeration algorithms.

This thesis deals with the number and the structure of the solutions for shop scheduling problems. The basic notation and preliminaries which are relevant for the problems in consideration can be found in Chapter 2.

We are starting the structural investigation from the classical open shop problem with n jobs and m machines. A feasible solution of such a problem is represented by a sequence graph (directed acyclic graph) or a sequence (certain Latin rectangle). In Chapter 3 we discuss the modeling concepts and we prove some statements about sequence graphs and sequences.

Chapter 4 contains the general determination of the number of sequences. Here, exact formulae for the number of sequences for open shop problems with up to three machines and an arbitrary number of jobs or vice versa have been found. Furthermore, new upper and lower bounds for the number of $n \times m$–sequences are developed. These results are mainly based on the estimation of the chromatic polynomial of the Hamming graph $K_n \times K_m$.

The main topic of Chapter 5 is a new enumeration method which gives us the ability to generate all $n \times m$–sequences efficiently at least for small values of n and m. This procedure results from the construction of equivalence classes in which we collect sequences with the same basic structure.

In Chapter 6 we present and compare two concepts which serve for the determination of optimality criteria for sequences. With the aid of the concept of irreducibility we are able to reduce the set of all sequences to a set of potentially-optimal sequences, in which there is always an optimal sequence regardless of the given processing times. By means of the concept of stability we can characterize optimal sequences for given processing times, which remain optimal if there are certain deviations from these processing times. Such sequences are called stable sequences.

In Chapter 7 we discuss various methods for the enumeration of potentially-optimal sequences. These methods allow us to generate a comparatively small set of potentially-optimal sequences only instead of the whole set of $n \times m$–sequences for a given problem. Thus we can restrict to such a small set if we are searching for an optimal sequence for a given open shop problem with n jobs and m machines.

From the ratio of the total number of sequences and the corresponding number of potentially-optimal sequences we can deduce interesting statements about the differences in the hardness of the classical job shop problem with different machine orders of the jobs.

Danksagung

Für ihre fachkundige und freundliche Betreuung und Unterstützung beim Erstellen dieser Arbeit bin ich Prof. Dr. Heidemarie Bräsel und Dr. Thomas Tautenhahn sehr dankbar. Die Anfertigung dieser Dissertation wurde mir anhand eines Stipendiums im Rahmen der Graduiertenförderung des Landes Sachsen-Anhalt sowie durch das vom Land Sachsen-Anhalt finanzierte Projekt „Lateinische Rechtecke in der Schedulingtheorie“ ermöglicht. Mein ganz besonderer Dank gilt Dr. Eva Nuria Müller und Dipl.-Math. Per Willenius, denn ihre Korrekturvorschläge waren mir bei der Durchsicht der Arbeit eine große Hilfe. Ebenso möchte ich mich bei meinen Eltern und Freunden für die Unterstützung in vielen Bereichen bedanken.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Die *Schedulingtheorie* befaßt sich mit der Optimierung der zeitlichen Zuordnung von knappen Ressourcen zu bestimmten Aktivitäten und kann als ein Teilgebiet des *Operations Research* aufgefaßt werden. Wegen der großen praktischen Relevanz in der betrieblichen Produktionsplanung erfährt die Schedulingtheorie seit den fünfziger Jahren eine enorme Entwicklung, die unter anderem an der Vielzahl von Publikationen in diesem Bereich abgelesen werden kann. Durch die obige Beschreibung der Optimierungsprobleme in der Schedulingtheorie ergibt sich eine große Anzahl verschiedener Problemtypen. Zusätzlich sind die aus den praktischen Gegebenheiten abgeleiteten Modellierungen dieser Probleme durch die verschiedenen Restriktionen und Anforderungen sehr vielfältig. Viele Schedulingprobleme treten auch in Lebensbereichen auf, die nicht direkt aus dem Anwendungsgebiet der Produktionsplanung stammen. Beispielsweise seien hier Probleme bei der Stundenplanerstellung in Schulen oder bei der Optimierung des zeitlichen Zusammenspiels von Computerprozessoren genannt.

Typischerweise existiert für Schedulingprobleme eine sehr große Anzahl zulässiger Zuordnungen bzw. Lösungen. Beim Auffinden einer optimalen Lösung kann man sich zwar in der Regel auf eine bestimmte endliche Menge von zulässigen Lösungen beschränken, aber die Anzahl dieser Lösungen ist meistens immer noch so groß, daß sich deren Untersuchung selbst mit Hilfe moderner Computertechnik als zu aufwendig herausstellt. In diesem Zusammenhang ist eine Unterscheidung zwischen einfachen und schwierigen Problemen bezüglich der Laufzeit entsprechender Lösungsalgorithmen wünschenswert.

Die Anwendung der *Komplexitätstheorie* für Schedulingprobleme ist zu einem wichtigen Instrument geworden, denn sie erlaubt eine formale Interpretation der empirischen Unterscheidung zwischen einfachen und schwierigen kombinatorischen Optimierungsproblemen. Diese Unterscheidung beruht auf einer Identifizierung der *einfachen Probleme* mit Problemen, deren Lösung eine Zeit in Anspruch nimmt, die nur durch eine polynomiale Funktion der Problemgröße beschränkt ist, während

es für die *schweren* bzw. $\mathcal{NP}$*-vollständigen Probleme* unwahrscheinlich ist, daß ein polynomialer Lösungsalgorithmus existiert.

Problemstellung und Modellierung

Bei der gängigen Terminologie für Schedulingprobleme ist eine Menge von *Aufträgen* und eine Menge von *Maschinen* gegeben, wobei die Aktivitäten den Aufträgen entsprechen und die Ressourcen durch die Maschinen repräsentiert werden. Ein Schedulingproblem mit mehr als einem Auftrag und mehr als einer Maschine, bei dem zu gegebenen Aufträgen und Maschinen jeder Auftrag auf jeder Maschine für eine bestimmte Zeit zu bearbeiten ist, wird als *Shop-Scheduling-Problem* bezeichnet. Es gelten die für die meisten Schedulingprobleme üblichen Bedingungen, daß zu jedem Zeitpunkt jede Maschine höchstens einen Auftrag gleichzeitig bearbeitet, und jeder Auftrag auf höchstens einer Maschine gleichzeitig bearbeitet werden kann.

Ein spezielles Shop-Scheduling-Problem, das sogenannte *Open-Shop-Problem*, ist zentraler Ausgangspunkt der hier vorgestellten Untersuchungen. Bei diesem Problem ist die Reihenfolge, in der ein Auftrag von den Maschinen bearbeitet wird und in der eine Maschine die Aufträge bearbeitet, für alle Aufträge und Maschinen beliebig wählbar. Die Optimierungsaufgabe liegt in der Bestimmung dieser Reihenfolgen, so daß eine zulässige Lösung entsteht und dabei eine gegebene Zielfunktion minimiert wird, die für gewöhnlich als eine Funktion in den Fertigstellungszeiten der Aufträge definiert ist.

Diese Art von Schedulingproblemen tritt in vielen Lebensbereichen auf. Stellvertretend wird ein Beispiel aus der Praxis gegeben: In Kraftfahrzeugwerkstätten kann die Ablaufplanung für die anfallenden Inspektionen als ein bestimmtes Open-Shop-Problem aufgefaßt werden. Die Aufträge entsprechen hierbei den Fahrzeugen und die Abteilungen übernehmen die Rolle der Maschinen. In den Abteilungen werden die bezüglich ihrer Reihenfolge voneinander unabhängigen Wartungsarbeiten (z. B. Ölwechsel, Prüfung der Bremsen, Prüfung der Elektrik, usw.) durchgeführt, wobei die Bearbeitungszeiten für die einzelnen Wartungsarbeiten vorher bekannt sind. Es wird angenommen, daß zu jedem Zeitpunkt immer nur höchstens eine Wartungsarbeit an einem Fahrzeug vorgenommen, und höchstens ein Fahrzeug in einer Abteilung gewartet werden kann. Das Ziel ist eine möglichst geringe Gesamtbearbeitungszeit für alle anfallenden Inspektionen. Auf diese Weise sind alle Fahrzeuge wieder so früh wie möglich einsatzbereit.

Sehr anschaulich können Shop-Scheduling-Probleme anhand von Graphen modelliert werden. Ein *Graph* G ist ein geordnetes Paar $G = (V, E)$, bestehend aus einer Menge $V \neq \varnothing$ von *Knoten* und einer Menge E von *Kanten*, wobei jede Kante eine zweielementige Teilmenge $\{u, v\}$ mit $u, v \in V$ ist. Bei einem *Digraphen* bzw. *gerichteten Graphen* $G' = (V', E')$ besteht die Menge E' aus geordneten Knoten-

paaren (u, v) mit $u, v \in V'$. In diesem Fall wird häufig auch von *gerichteten* bzw. *orientierten Kanten* gesprochen. Zur Illustration einer zulässigen Lösung kann eine bestimmte Klasse von Digraphen verwandt werden. Diese *Ablaufgraphen* stehen mit den in der Schedulingtheorie häufig benutzten *disjunktiven Graphen* in engem Zusammenhang, auf den in Kapitel 3 näher eingegangen wird.

Zur Modellierung der Lösungen von Shop-Scheduling-Problemen bietet sich neben den Ablaufgraphen auch das *Blockmatrizenmodell* (siehe BRÄSEL [7]) an, bei dem jede zulässige Lösung eines Problems durch ein eindeutig bestimmtes lateinisches Rechteck repräsentiert ist. Ein *lateinisches Rechteck*[1] ist eine Matrix mit Einträgen aus einer endlichen Menge S, wobei jedes Element aus S höchstens einmal in jeder Zeile und höchstens einmal in jeder Spalte vorkommt.

Die lateinischen Rechtecke, die eine zulässige Lösung beschreiben, haben spezielle Eigenschaften und werden als *Pläne* bezeichnet. Jedem Eintrag eines Planes entspricht eine *Operation*, d. h. einem Bearbeitungsschritt eines Auftrags auf einer Maschine, und jeder Eintrag gibt gleichzeitig die Position seiner zugehörigen Operation innerhalb der Reihenfolge des gesamten Produktionsablaufs wieder.

Strukturuntersuchungen

Die Anzahl der Pläne eines Shop-Scheduling-Problems wird mit steigender Auftrags- und Maschinenanzahl schnell sehr groß und unhandlich. Man kann sich dieses Wachstum zum Beispiel an der Anzahl der Möglichkeiten veranschaulichen, für alle Maschinen die Reihenfolgen der auf ihnen zu bearbeitenden Aufträge festzulegen. Diese Anzahl beträgt $(n!)^m$ bei einem Shop-Scheduling-Problem mit n Aufträgen und m Maschinen, und im Falle $n = 6,\ m = 3$ sind das bereits 373 248 000 Möglichkeiten! Deshalb wird bei vielen *Approximationsalgorithmen* zur Lösung eines Shop-Scheduling-Problems auf eine vollständige Plan-Enumeration verzichtet. Auf diese Weise nähert man den optimalen Zielfunktionswert zwar in der Regel nur bis auf einen bestimmten Faktor an, aber der entsprechende Algorithmus besitzt dafür eine wesentlich kürzere Laufzeit.

Trotzdem ist die Enumeration aller Pläne eines Shop-Scheduling-Problems häufig von Nutzen, denn es können damit allgemeine Aussagen über die Schwierigkeit des betrachteten Problems getroffen werden. Offensichtlich hängt die Auswahl der Pläne, die hinsichtlich der Optimalität günstiger als andere sind, in starkem Maße von der Struktur des zugrunde liegenden Ablaufgraphen ab. Da bereits während der Enumeration das Erkennen von ungünstigen Strukturen möglich ist, kann man viele Pläne schon vor ihrer vollständigen Erzeugung von der weiteren Betrachtung

[1] Im 18. Jh. hat EULER [34] wahrscheinlich als erster den Begriff *lateinisches Quadrat* geprägt, denn er benutzte damals lateinische Buchstaben als Einträge für entsprechende quadratische Matrizen.

ausschließen. Bei der Plan-Enumeration in der vorliegenden Arbeit spielen die vorgegebenen Zeiten für die Bearbeitung eines Auftrags auf einer Maschine zunächst keine Rolle. Die Untersuchungen für Shop-Scheduling-Probleme, die sich auf die rein kombinatorische Frage nach der Art und Anzahl der zugehörigen Pläne bzw. Ablaufgraphen unabhängig von den gegebenen Bearbeitungszeiten beziehen, heißen *Strukturuntersuchungen*.

Die ersten Ergebnisse auf dem Gebiet der Strukturuntersuchungen stammen von AKERS und FRIEDMAN [1] sowie von CONWAY, MAXWELL und MILLER [26]. Die Autoren ermitteln in diesen Arbeiten für bestimmte Shop-Scheduling-Probleme günstige Grundstrukturen zulässiger Lösungen. Auf diese Weise kann man sich bei der Suche nach optimalen Lösungen auf einen kleineren Suchraum beschränken.

In [14, 15] haben BRÄSEL und M. KLEINAU die Resultate aus [1] und [26] für Open-Shop-Probleme verallgemeinert. Darauf aufbauend sind im Rahmen der vorliegenden Arbeit neue Enumerationstechniken entwickelt worden. Mit Hilfe dieser Techniken ist die Plan-Enumeration und die Charakterisierung günstiger Planstrukturen für größere Formate als bisher möglich. Auf diese Weise kann man zum Beispiel alle Pläne mit bis zu 6 Aufträgen und 3 Maschinen vollständig enumerieren und dabei die Pläne mit günstigen Strukturen identifizieren. Eine Reihe von neuen theoretischen Ergebnissen im Bereich der Anzahlproblematik bestätigen und ergänzen die durch die Enumeration entstandenen numerischen Werte. Insgesamt können die hier präsentierten Resultate auch als Weiterentwicklung der Überlegungen und Algorithmen zur Plan-Enumeration aus der Dissertation von M. KLEINAU [55] aufgefaßt werden. Die in [55] benutzten Begriffe und Konzepte werden dabei erweitert und an die gängige graphentheoretische Terminologie angepaßt.

Kapitelübersicht

In Kapitel 2 werden die Grundlagen aus dem Bereich der Schedulingtheorie bereitgestellt, die für die hier untersuchten Probleme relevant sind. Kapitel 3 enthält die Beschreibung der Konzepte und Ergebnisse im Zusammenhang mit der Modellierung von Schedulingproblemen durch Ablaufgraphen und Pläne.

In Kapitel 4 werden bekannte und neue Ergebnisse im Bereich der Anzahlproblematik für Pläne eines gegebenen Formats vorgestellt. Neben einer ausführlichen Übersicht über den aktuellen Stand auf dem Gebiet der Pläne, die den klassischen lateinischen Rechtecken entsprechen, werden in diesem Kapitel neue exakte Werte sowie obere und untere Schranken für die Anzahl allgemeiner Pläne entwickelt. Der zentrale Gegenstand in Kapitel 5 ist ein neues Enumerationsverfahren zur Erzeugung und Anzahlbestimmung der Pläne eines gegebenen Formats. Vorbereitend dazu werden Methoden beschrieben, durch die man Pläne mit gleichartigen Eigenschaften zusammenfassen kann. Weiterhin enthält dieses Kapitel Anzahlbestimmungen, bei

denen jeweils ausschließlich die Reihenfolge, in der ein Auftrag von den Maschinen bearbeitet wird, eine Rolle spielt.

Kapitel 6 ist der Charakterisierung von Plänen mit günstigen Grundstrukturen gewidmet. Es handelt sich um *potentiell-optimale* Pläne, unter denen unabhängig von den gegebenen Bearbeitungszeiten stets ein Plan existiert, der eine optimale Lösung des entsprechenden Shop-Scheduling-Problems repräsentiert. Weiterhin werden in diesem Kapitel Zusammenhänge zwischen potentieller Optimalität und sogenannter Stabilität von Plänen hergestellt. Verschiedene Verfahren zur Enumeration der potentiell-optimalen Pläne werden in Kapitel 7 behandelt. Abschließende Bemerkungen und Einstufungen der erzielten Ergebnisse sollen diese Arbeit in Kapitel 8 abrunden.

Kapitel 2

Grundlagen

Dieses Kapitel behandelt einige Grundbegriffe aus der Scheduling- und Komplexitätstheorie, die notwendig für das Verständnis der Problemstellungen sind.

In der vorliegenden Arbeit werden ausschließlich *deterministische Scheduling-probleme* behandelt, bei denen im Gegensatz zu den *stochastischen Problemen* alle problemdefinierenden Parameter vor dem Optimierungsprozeß bereits bekannt sind. Diese Probleme können nochmals in *Single-Stage-* und *Multi-Stage-Probleme* unterteilt werden, je nachdem ob zur Fertigstellung eines Auftrags eine oder mehr als eine Maschine benötigt wird. Es wird sich hier mit der Klasse der Multi-Stage-Probleme beschäftigt, zu der auch die *Shop-Scheduling-Probleme* gehören, die im anschließenden Abschnitt definiert werden und in der Regel schwerer als vergleichbare Single-Stage-Probleme sind.

2.1 Problem-Klassifikation

Parallel zur Entwicklung der Schedulingtheorie gestaltet sich die Verfeinerung einer detaillierten Problemklassifikation, dessen erste Grundlage im Buch von Conway, Maxwell und Miller [26] geschaffen wurde. Das Klassifikationsschema von Graham *et al.* [41] ist eine Weiterentwicklung des Schemas [26] und umfaßt eine sehr große Anzahl der Schedulingprobleme. Im folgenden werden nur die in der vorliegenden Arbeit benötigten Begriffe und Symbole in Anlehnung an das Klassifikationsschema [41] eingeführt.

Bei einem *Shop-Scheduling-Problem* wird für n, $m \in \mathbb{N}$ mit $n, m \geq 2$ eine Menge von n *Aufträgen (jobs)* $\{J_1, \ldots, J_n\}$ betrachtet, die auf einer Menge von m *Maschinen* $\{M_1, \ldots, M_m\}$ zu bearbeiten sind. Dabei bearbeitet jede Maschine höchstens einen Auftrag gleichzeitig, und jeder Auftrag kann höchstens auf einer Maschine gleichzeitig bearbeitet werden. Jeder Auftrag J_i, $i = 1, \ldots, n$, besteht aus einer Menge von m *Operationen* $\{o_{i1}, \ldots, o_{im}\}$, wobei jede Operation o_{ij}, $j = 1, \ldots, m$,

des Auftrags J_i auf der Maschine M_j während der *Bearbeitungszeit (processing time)* $p_{ij} > 0$ ausgeführt werden muß.

Zwischen je zwei Operationen o_{ik}, o_{il} eines Auftrags J_i oder je zwei Operationen o_{kj}, o_{lj} einer Maschine M_j können *Vorrangbedingungen (precedence constraints)* gegeben sein. Beispielsweise bedeutet eine Vorrangbedingung der Form $o_{i1} \rightarrow o_{i2}$, daß die Operation o_{i1} bereits bearbeitet sein muß, bevor mit der Bearbeitung der Operation o_{i2} des Auftrags J_i begonnen werden darf.

Bei einer Zuordnung der Maschinen zu den Aufträgen gibt die *Fertigstellungszeit (completion time)* $c_{ij} > 0$ der Operation o_{ij} den Zeitpunkt an, bei dem die Bearbeitung der Operation o_{ij} beendet ist. Die *Fertigstellungszeit* C_i des Auftrags J_i bezeichnet den Zeitpunkt, nachdem die Bearbeitung der letzten Operation dieses Auftrags beendet ist, also $C_i = \max_j(c_{ij})$.

Ziel eines Shop-Scheduling-Problem ist das Finden einer zulässigen Lösung, d. h. einer zulässigen zeitlichen Zuordnung von Maschinen und Aufträgen, so daß die Fertigstellungszeiten C_i der Aufträge J_i einem bestimmten Optimalitätskriterium genügen. Mit Hilfe der drei Felder $\boldsymbol{\alpha}|\boldsymbol{\beta}|\boldsymbol{\gamma}$ werden im Klassifikationsschema in [41] die verschiedenen Maschinen- (α), Auftrags- (β) und Optimalitätsmerkmale (γ) von Shop-Scheduling-Problemen wiedergegeben.

Maschinenumgebung ($\boldsymbol{\alpha}$)

Das erste Klassifikationsfeld $\boldsymbol{\alpha = \alpha_1 \alpha_2}$ spezifiziert die Maschinenmerkmale, wobei beim ersten Teilfeld α_1 hier vier Symbole von Interesse sind: $\boldsymbol{\alpha_1 \in \{J, F, O, G\}}$. Diese Symbole bezeichnen Sonderfälle des Shop-Scheduling-Problems, für die ganz bestimmte oder überhaupt keine Vorrangbedingungen bestehen:

$\boldsymbol{\alpha_1 = J}$: Beim *Job-Shop-Problem* sind für jeden Auftrag J_i bezüglich seiner Operationen o_{ij} $(j = 1, \ldots, m)$ Vorrangbedingungen der Form $o_{i,j_1} \rightarrow o_{i,j_2} \rightarrow \cdots \rightarrow o_{i,j_m}$ festgelegt.

$\boldsymbol{\alpha_1 = F}$: Das *Flow-Shop-Problem* ist ein Job-Shop-Problem, bei dem für jeden Auftrag die Vorrangbedingungen dieselben sind, also $o_{i1} \rightarrow o_{i2} \rightarrow \cdots \rightarrow o_{im}$ für alle $i = 1, \ldots, n$.

$\boldsymbol{\alpha_1 = O}$: Beim *Open-Shop-Problem* bestehen zwischen den Operationen o_{ij} keinerlei Vorrangbedingungen.

$\boldsymbol{\alpha_1 = G}$: Beim *General-Shop-Problem* existieren Vorrangbedingungen zwischen beliebigen Operationen einer Maschine bzw. eines Auftrags.

$\boldsymbol{\alpha_2 \in \{m, \circ\}}$: Das Symbol α_2 bezeichnet die *Maschinenanzahl*. Für $\alpha_2 = m$ mit $m \in \mathbb{N}$ wird eine konstante Anzahl m der Maschinen vorausgesetzt, während

man für das leere Symbol $\alpha_2 = \circ$ eine variable Maschinenanzahl annimmt, die dann als Teil der Eingabe eines Algorithmus zur Lösung des betreffenden Shop-Scheduling-Problems aufzufassen ist.

Auftragseigenschaften (β)

Das zweite Klassifikationsfeld $\boldsymbol{\beta = \beta_1, \beta_2, \ldots}$ stellt eine Kombination unterschiedlicher Typen von Nebenbedingungen für die Aufträge J_i dar. Die in der vorliegenden Arbeit betrachteten Nebenbedingungen sind:

$\beta_1 \in \{p_{ij} = 1, \circ\}$: Der Eintrag $p_{ij} = 1$ bedeutet, daß jede Operation o_{ij} die Bearbeitungszeit 1 hat, d. h. es bestehen sogenannte *Einheitsbearbeitungszeiten*.

$\beta_2 \in \{\underline{p} \leq p_{ij} \leq \overline{p}, \circ\}$: Im Fall $\beta_2 = \underline{p} \leq p_{ij} \leq \overline{p}$ gibt es *konstante untere und obere Schranken* für die Bearbeitungszeiten p_{ij} der Operationen o_{ij}.

$\beta_3 \in \{n = k, \circ\}$: Der Eintrag β_3 kann weitere Symbole mit naheliegender Interpretation wie z. B. $n = 2$ für ein Problem mit zwei Aufträgen haben.

Optimalitätskriterium (γ)

Das dritte Klassifikationsfeld $\boldsymbol{\gamma \in \{C_{\max}, \sum C_i, \ldots\}}$ gibt als Optimalitätskriterium die zu minimierende Zielfunktion γ an. Die Zielfunktion ist immer eine Funktion in den Fertigstellungszeiten C_i der Aufträge J_i, also $\gamma = \gamma(C_1, C_2, \ldots, C_n)$. Eine Zielfunktion γ heißt *regulär*, wenn γ nicht-fallend in den C_i ist. Das heißt, wenn bei zwei verschiedenen Lösungen A und B eines Shop-Scheduling-Problems für die Fertigstellungszeiten C_i^A und C_i^B die Beziehungen $C_i^A \leq C_i^B$ für alle $i = 1, \ldots, n$ gelten, folgt für jede reguläre Zielfunktion die Ungleichung

$$\gamma(C_1^A, \ldots, C_n^A) \leq \gamma(C_1^B, \ldots, C_n^B).$$

Es gibt eine Reihe von möglichen Optimalitätskriterien, wobei hier nur die folgenden regulären Zielfunktionen betrachtet werden:

$\gamma = C_{\max}$: Das Symbol $C_{\max}$ bezeichnet die zu minimierende *Gesamtbearbeitungszeit (makespan)*. Dies ist die größte Fertigstellungszeit C_i über allen Aufträgen J_i, also $C_{\max} = \max(C_1, \ldots, C_n)$.

$\gamma = \sum C_i$: Dieses Symbol bezeichnet als Optimalitätskriterium die *Summe der Fertigstellungszeiten (total flow time)* über allen Aufträgen J_i.

2.2 Sequenzen und Schedules

Bei den in dieser Arbeit betrachteten Shop-Scheduling-Problemen wird jeder Auftrag J_i stets genau einmal auf jeder Maschine M_j bearbeitet. Also kann eine Operation o_{ij} stets mit der Bearbeitung des Auftrags J_i auf der Maschine M_j für alle $i = 1, \ldots, n$ und $j = 1, \ldots, m$ identifiziert werden. Eine einzelne *Reihenfolge* $o_{i,j_1} \prec o_{i,j_2}$ bedeutet, daß mit der Bearbeitung der Operation o_{i,j_2} erst begonnen wird, nachdem die Bearbeitung der Operation o_{i,j_1} abgeschlossen ist. In einer Lösung für ein Shop-Scheduling-Problem wird für einen Auftrag J_i die Reihenfolge der Maschinen, auf denen J_i bearbeitet wird, als *technologische Reihenfolge* des Auftrags J_i bezeichnet. Analog dazu heißt für eine Maschine M_j die Reihenfolge der auf ihr bearbeiteten Aufträge die *organisatorische Reihenfolge* der Maschine M_j. Offensichtlich kann eine technologische bzw. organisatorische Reihenfolge auch als eine Operationen-Reihenfolge $o_{i,j_1} \prec o_{i,j_2} \prec \cdots \prec o_{i,j_m}$, bzw. $o_{i_1,j} \prec o_{i_2,j} \prec \cdots \prec o_{i_n,j}$ aufgefaßt werden.

Definition 2.2.1 Die *Technologie* ist die Menge der technologischen Reihenfolgen für alle Aufträge J_i, $i = 1, \ldots, n$. Die *Organisation* ist die Menge der organisatorischen Reihenfolgen für alle Maschinen M_j, $j = 1, \ldots, m$.

Beim Open-Shop-Problem sind sowohl die Technologie als auch die Organisation frei wählbar, während beim Job-Shop- und Flow-Shop-Problem die Technologie durch die gegebenen Vorrangbedingungen der Form „$\rightarrow$“ bereits festgelegt ist, und bei der Suche nach einer optimalen Lösung nur noch verschiedene Organisationen betrachtet werden.

Definition 2.2.2 Eine *Sequenz* eines Shop-Scheduling-Problems ist eine zulässige Kombination von Technologie und Organisation. Zulässig bedeutet hierbei, daß der durch die Kombination charakterisierte Produktionsablauf endlich ist.

Es handelt sich bei einer auf diese Weise definierten Sequenz eigentlich um eine „Multi-Sequenz“, denn es werden gleichzeitig stets alle technologischen und organisatorischen Reihenfolgen durch eine gegebene Sequenz beschrieben.

Ein *Schedule* ist die Realisierung einer Sequenz, bei der jeder Operation o_{ij} des gegebenen Shop-Scheduling-Problems ein fester Startzeitpunkt zugeordnet ist. Ein Schedule heißt *zulässig*, wenn neben dem durch die Sequenz garantierten endlichen Produktionsablauf alle weiteren problemspezifischen Bedingungen erfüllt sind. Zu einer gegebenen Sequenz kann ein Schedule anhand der vorgegebenen Bearbeitungszeiten p_{ij} der Operationen o_{ij} wie folgt bestimmt werden: Unter Berücksichtigung der durch Technologie und Organisation gegebenen Bearbeitungsreihenfolgen wird jede Operation o_{ij} so früh wie möglich ausgeführt. Die Laufzeit eines entsprechenden Algorithmus zur Berechnung der Start- und Fertigstellungszeiten der Operationen

wird später bei der Beschreibung der Matrix der Fertigstellungszeiten (Matrix C) im Blockmatrizenmodell auf Seite 23 behandelt. Ein Schedule, der auf die beschriebene Weise einer Sequenz eindeutig zugeordnet ist, heißt *semiaktiver Schedule*. Es liegt also ein semiaktiver Schedule vor, wenn keine Bearbeitung einer Operation früher beendet werden kann, ohne dabei eine technologische oder organisatorische Reihenfolge zu verändern oder eine der weiteren problemspezifischen Bedingungen zu verletzen.

In dieser Arbeit werden ausschließlich reguläre Zielfunktionen betrachtet. Offensichtlich genügt es, zur Minimierung solcher Zielfunktionen den untersuchten Lösungsbereich auf semiaktive Schedules zu beschränken (siehe FRENCH [35]).

Existenz optimaler Schedules

Die technologische Reihenfolge eines Auftrags J_i kann als Permutation der Indizes $j = 1, \ldots, m$ der Maschinen M_j aufgefaßt werden. Daher gibt es $m!$ mögliche technologische Reihenfolgen für einen Auftrag J_i. Dementsprechend kann eine gesamte Technologie für alle n Aufträge auf $(m!)^n$ verschiedene Weisen gebildet werden. Analog dazu gibt es bei einem Shop-Scheduling-Problem mit n Aufträgen und m Maschinen $(n!)^m$ mögliche Organisationen.

Für feste Werte von n und m ist die Anzahl aller möglichen Kombinationen der Technologien und Organisationen offensichtlich endlich. Da die Menge der Sequenzen für ein Problem mit n Aufträgen und m Maschinen gleichzeitig die Menge der zulässigen Kombinationen von Technologien und Organisationen darstellt, ist auch die Anzahl der Sequenzen endlich. Die oben beschriebene Zuordnung eines semiaktiven Schedules zu einer Sequenz ist eindeutig, daher folgt die Endlichkeit auch für die Anzahl der zu betrachtenden semiaktiven Schedules. Diese Endlichkeit sichert für ein Shop-Scheduling-Problem die Existenz eines Schedules, der bezüglich der gegebenen Zielfunktion optimal ist.

2.3 Komplexitätsergebnisse

Die Werkzeuge der Komplexitätstheorie sind wichtige Hilfsmittel zur Einschätzung der Schwierigkeit von kombinatorischen Optimierungsproblemen. Diese Theorie findet seinen Ursprung in Arbeiten von COOK [27] und KARP [51] Anfang der siebziger Jahre.

Eine ausführliche Übersicht zum Themenbereich der Komplexität im Zusammenhang mit rechnergestützten Lösungen von Entscheidungs- und Optimierungsproblemen ist dem Buch von GAREY und JOHNSON [36] zu entnehmen. In der vorliegenden Arbeit werden Begriffe wie z. B. *polynomialer Algorithmus*, *$\mathcal{NP}$-Vollständigkeit*, *polynomiale Reduktion* und *polynomiale Äquivalenz* als bekannt vorausgesetzt und

meist ohne weitere Erklärung benutzt. An dieser Stelle sei darauf hingewiesen, daß mit $\mathcal{P}$ bzw. $\mathcal{NP}$ die Klasse der Entscheidungsprobleme bezeichnet wird, zu deren Lösung deterministische bzw. nichtdeterministische Algorithmen mit polynomialer Laufzeit bekannt sind. Weiterhin heißt ein Optimierungsproblem *$\mathcal{NP}$-schwer ($\mathcal{NP}$-hard)*, wenn das zugehörigen Entscheidungsproblem $\mathcal{NP}$-vollständig ist, und damit im Falle der bisher nicht bestätigten Hypothese $\mathcal{P} \neq \mathcal{NP}$ für ein solches Problem kein deterministischer Algorithmus mit polynomialer Laufzeit existiert.

Die Terminologie für die Komplexität von Enumerationsproblemen ist nicht so weit verbreitet wie die von Entscheidungs- und Optimierungsproblemen. Daher wird auf diese Terminologie näher in Kapitel 5 bei der Betrachtung der Komplexität der Plan-Enumeration eingegangen.

Ziel bei der Lösung eines Shop-Scheduling-Problems ist das Finden einer optimalen und zulässigen Kombination aus Technologie und Organisation. Eine Einstufung eines solchen Problems in die Klasse $\mathcal{P}$ oder in die Klasse der $\mathcal{NP}$-schweren Probleme stellt ein Komplexitätsergebnis dar. Bei Brucker und Knust [20] ist eine ausführliche Übersicht bekannter Komplexitätsergebnisse für verschiedene Schedulingprobleme zu finden. Diese Zusammenstellung wird regelmäßig aktualisiert und ist im World-Wide-Web unter

`http://www.mathematik.uni-osnabrueck.de/research/OR/class/`

abrufbar.

Viele Schedulingprobleme lassen sich mittels elementarer polynomialer Reduktionen (siehe z. B. [18]) bezüglich ihrer Komplexität paarweise vergleichen. Bei den Definitionen im Anschluß werden die Begriffe „schwerere“ und „einfachere Probleme“ stets gemäß dieser Reduktionen aufgefaßt. Ein Problem aus der Klasse $\mathcal{P}$ heißt *maximal polynomial lösbar*, wenn alle schwereren Probleme $\mathcal{NP}$-schwer oder offen sind. Ein Problem heißt *minimal $\mathcal{NP}$-schwer*, wenn es $\mathcal{NP}$-schwer ist und alle einfacheren Probleme aus $\mathcal{P}$ oder offen sind. Ein Problem heißt *minimal offen*, wenn sein Komplexitätsstatus unbekannt ist, aber alle einfacheren Probleme aus $\mathcal{P}$ sind. Schließlich wird ein Problem *maximal offen* genannt, wenn sein Komplexitätsstatus unbekannt ist, aber alle schwereren Probleme bereits $\mathcal{NP}$-schwer sind. Offensichtlich genügt die Aufzählung aller bekannten Probleme aus diesen vier Bereichen zur vollständigen Beschreibung des Standes der Forschung bei der komplexitätstheoretischen Einstufung von Schedulingproblemen. Daher enthält die Übersicht [20] ausschließlich Komplexitätsergebnisse für Schedulingprobleme aus diesen Problemklassen.

Im Rahmen eines Projekts zur Entwicklung des Programmpakets LiSA (siehe Bräsel *et al.* [9]) wurde auf der Basis der Daten aus [20] eine Datenbank im BibTeX-Format erstellt. Die Datensätze dieser Datenbank umfassen die Literaturquellen, in denen die Komplexität (entweder maximal polynomial lösbar oder minimal $\mathcal{NP}$-schwer) der Schedulingprobleme nachgewiesen ist. Mit Hilfe des Pro-

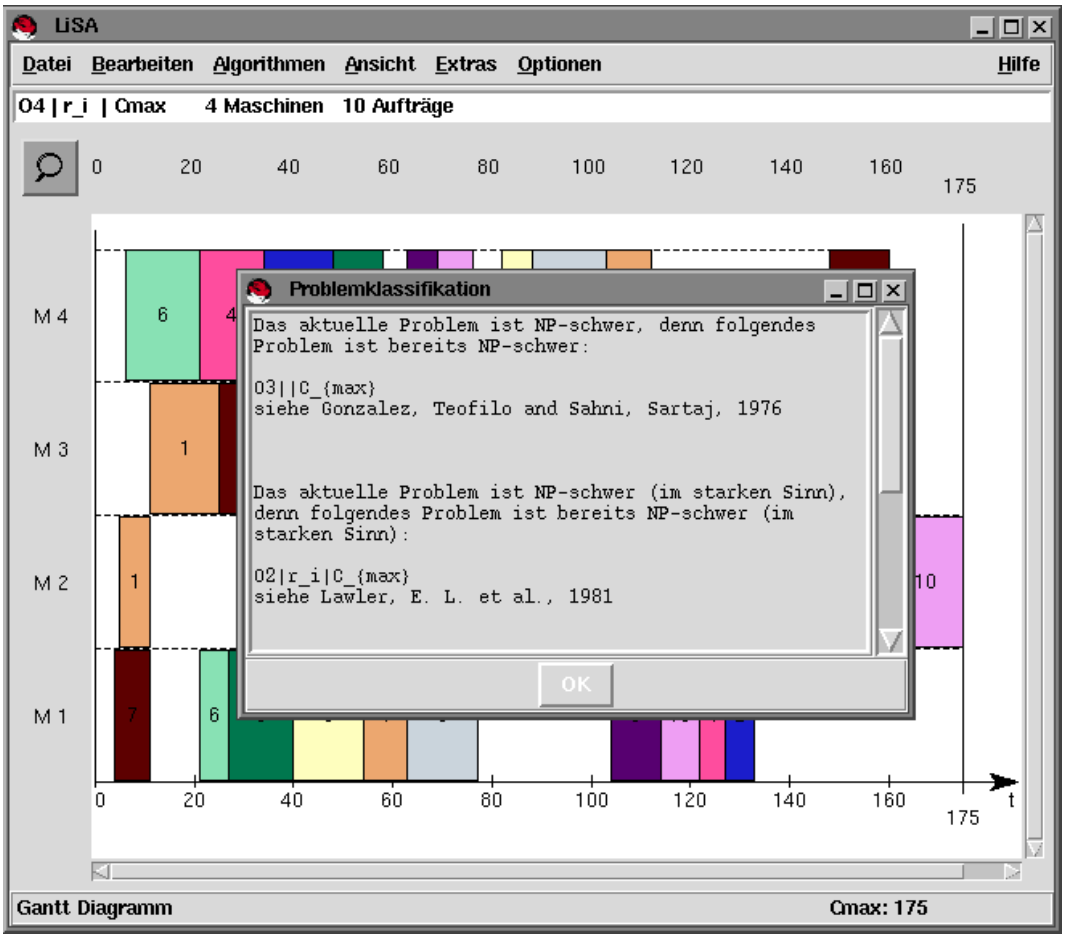


Abbildung 2.1: Die Klassifikation von Schedulingproblemen in LiSA.

grammpakets LiSA ist unter anderem eine bequeme und interaktive Benutzung dieser BibTeX-Datenbank möglich.

Beispielsweise ist in Abbildung 2.1 die Programmausgabe bei der Problemklassifikation für das Open-Shop-Problem $O4|r_i|C_{\max}$ dargestellt. Ein eigenes LiSA-Fenster enthält die Literaturquellen, die die Zugehörigkeit dieses Open-Shop-Problems zur Klasse der $\mathcal{NP}$-schweren Probleme zeigen. Die eindeutigen Identifikationsschlüssel der entsprechenden BibTeX-Datensätze (vgl. Abbildung 2.2) setzen sich in der Regel aus der entsprechenden „Mathematical Reviews Number“ oder „Zentralblatt-Nummer“ zusammen. Im `ANNOTE`-Feld wird jeweils codiert, welche Schedulingprobleme in der gegebenen Literaturquelle vorkommen, und zu welchen Komplexitätsklassen diese Probleme zugeordnet werden können.

```
@article {MR55:2108,
   AUTHOR = {Gonzalez, Teofilo and Sahni, Sartaj},
    TITLE = {Open shop scheduling to minimize finish time},
  JOURNAL = {J. Assoc. Comput. Mach.},
   VOLUME = 23,
     YEAR = 1976,
   NUMBER = 4,
    PAGES = {665--679},
   ANNOTE = {$O2||C_{\max}$ is in $P$;\\ $O3||C_{\max}$ is
                  $\NP$-hard.}
}

@article {MR82m:90091,
   AUTHOR = {Lawler, E. L. and Lenstra, J. K. and Rinnooy Kan, A. H. G.},
    TITLE = {Minimizing maximum lateness in a two-machine open shop},
  JOURNAL = {Math. Oper. Res.},
   VOLUME = 6,
     YEAR = 1981,
   NUMBER = 1,
    PAGES = {153--158},
     ISSN = {0364-765X},
   ANNOTE = {$O2|p_{ij}=1,prec,r_i|L_{\max}$ is in $P$;\\
                  $O2|r_i|C_{\max}$ is $*\NP$-hard;\\ $O2||L_{\max}$
                  is $*\NP$-hard;\\ $O2|pmtn|\sum{U_i}$ is
                  $\NP$-hard.}
}
```

Abbildung 2.2: Zwei Datensätze der BibTeX-Datenbank von LiSA.

Kapitel 3

Konzepte der Modellierung

Um eine strukturelle Vorstellung der untersuchten Schedulingprobleme gewinnen zu können, sind geeignete Modellierungen von großem Nutzen. In diesem Kapitel werden einige Konzepte und Ergebnisse behandelt, mit denen die Strukturuntersuchungen für Shop-Scheduling-Probleme in sinnvoller Weise realisiert werden können.

Viele Lösungsverfahren basieren auf der Veranschaulichung der Probleme durch geeignete Graphen. In der vorliegenden Arbeit wird auf die Terminologie der Graphentheorie im Buch von HARARY [44] zurückgegriffen, solange nicht abweichende oder zusätzliche Bezeichnungen definiert sind. Es werden stets Graphen $G = (V, E)$ ohne Schlingen und Mehrfachkanten betrachtet.[1]

Während des Optimierungsprozesses bei einem Shop-Scheduling-Problem sind stets für bestimmte Paare von Operationen Vorrangbedingungen bzw. Reihenfolgen festgelegt. Zur Illustration der verschiedenen Lösungsstrategien werden sogenannte disjunktive Graphen benutzt. Ein *disjunktiver Graph* $G^* = (V, C \cup D)$ zu gegebenen Vorrangbedingungen eines Shop-Scheduling-Problems ist anhand der Mengen V, C und D definiert:

- $V = \{\, o_{ij} \mid 1 \leq i \leq n,\, 1 \leq j \leq m \,\}$

  Die Knotenmenge besteht aus allen Operationen o_{ij}. Jeder Knoten o_{ij} besitzt als Gewicht die zugehörige Bearbeitungszeit p_{ij}.

- Die Menge C der *konjunktiven (gerichteten) Kanten* mit

  $$C = \Big\{\ (o_{ij}, o_{kl}) \ \Big|\ o_{ij}, o_{kl} \in V, \big((i = k) \vee (j = l)\big) \wedge \big(o_{ij} \rightarrow o_{kl}\big)\ \Big\}.$$

  Die konjunktiven Kanten repräsentieren die gegebenen Vorrangbedingungen $(o_{ij} \rightarrow o_{kl})$ zwischen je zwei Operationen (o_{ij} und o_{kl}), die zu einem Auftrag J_i bzw. zu einer Maschine M_j gehören.

[1] Eine *Schlinge* eines Graphen $G = (V, E)$ ist eine Kante $\{v, w\} \in E$ mit $v, w \in V$ und $v = w$. Wenn in G mehr als eine Kante aus E zwei Knoten $v, w \in V$ verbindet, werden diese Kanten $\{v, w\}$ als *Mehrfachkanten* von G bezeichnet.

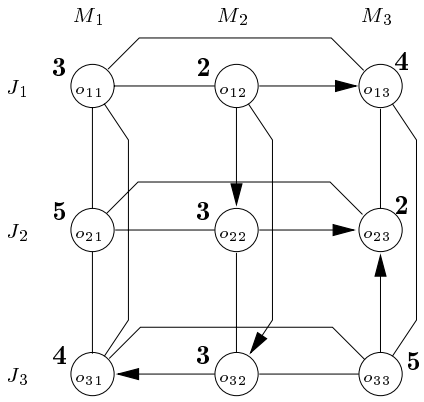


Abbildung 3.1: Ein disjunktiver Graph mit 9 Operationen.

- Die Menge D der *disjunktiven (ungerichteten) Kanten* mit

$$D = \Big\{ \{o_{ij}, o_{kl}\} \;\Big|\; o_{ij}, o_{kl} \in V, \\ \big((i = k) \vee (j = l)\big) \;\wedge\; \big((o_{ij}, o_{kl}) \notin C \;\wedge\; (o_{kl}, o_{ij}) \notin C\big) \Big\}.$$

  Die disjunktiven Kanten bestehen zwischen Operationen des gleichen Auftrags J_i oder der gleichen Maschine M_j, zwischen denen keine konjunktive Kante existiert.

Diese auf Roy und Sussmann [84] zurückgehenden Graphen besitzen gerichtete sowie ungerichtete Kanten und eignen sich gut zur schrittweisen Konstruktion zulässiger Lösungen für Shop-Scheduling-Probleme. Zum Beispiel wird in Abbildung 3.1 ein General-Shop-Problem mit 3 Aufträgen und 3 Maschinen und den Vorrangbedingungen $o_{12} \to o_{13},\ o_{12} \to o_{22},\ o_{12} \to o_{32},\ o_{22} \to o_{23},\ o_{32} \to o_{31},\ o_{33} \to o_{23}$ mit Hilfe eines disjunktiven Graphen modelliert.

Zur sukzessiven Bestimmung einer vollständigen und zulässigen Zuordnung zwischen den Maschinen M_j und den Aufträgen J_i müssen den Kanten $d \in D$ in $G^* = (V, C \cup D)$ Orientierungen zugewiesen werden, so daß keine gerichteten Kreise entstehen. Disjunktive Kanten werden in konjunktive umgewandelt, d. h. für je zwei Operationen o_{ij}, o_{il} oder o_{ij}, o_{kj}, die entweder zu einem Auftrag J_i oder zu einer Maschine M_j gehören, wird auf diese Weise eine Reihenfolge („$\prec$“) festgelegt, da solche Operationen nicht parallel bearbeitet werden können.

Am Ende dieses Prozesses existieren ausschließlich gerichtete Kanten, d. h. es gilt $D = \varnothing$. Der dadurch entstandene azyklische Digraph[2] beschreibt eine Sequenz,

[2]Ein *azyklischer Digraph* ist ein Digraph, der keinen gerichteten Kreis enthält. In Harary [44] wird ein solcher Digraph als *kreisloser Digraph* bezeichnet.

d. h. eine zulässige Kombination aus Technologie und Organisation. Die Klasse der azyklischen Digraphen, die man auf diese Weise mit Sequenzen assoziiert, werden im folgenden Abschnitt gesondert eingeführt, da sie in dieser Arbeit im Rahmen der Strukturuntersuchungen eine zentrale Rolle spielen.

3.1 Ablaufgraphen

Wenn Technologie und Organisation eines Shop-Scheduling-Problems vollständig vorgegeben sind, ist zwischen je zwei Operationen o_{ij} und o_{kl}, die zu einem Auftrag bzw. zu einer Maschine gehören, entweder $o_{ij} \prec o_{kl}$ oder $o_{kl} \prec o_{ij}$ als Reihenfolge festgelegt. Diese Reihenfolgen sind entweder durch direkte Vorgänger-Nachfolger-Beziehungen oder transitiv bestimmt. So ergeben sich z. B. durch die Reihenfolge $o_{i,j_1} \prec o_{i,j_2} \prec \ldots \prec o_{i,j_m}$ für den Auftrag J_i gleichzeitig transitiv die Reihenfolgen $o_{i,j_1} \prec o_{i,j_m},\ o_{i,j_2} \prec o_{i,j_m},\ \ldots,\ o_{i,j_{m-2}} \prec o_{i,j_m}$.

Definition 3.1.1 Es sei A eine Sequenz. Der azyklische Digraph $G(A) = (V, E)$ mit $V = \{\, o_{ij} \mid 1 \leq i \leq n,\ 1 \leq j \leq m \,\}$, $E = E_{TR} \cup E_{OR}$ und

$$\begin{aligned} E_{TR} &= \{\, (o_{i,j_1}, o_{i,j_2}) \mid o_{i,j_1}, o_{i,j_2} \in V,\ o_{i,j_1} \prec o_{i,j_2} \,\}, \\ E_{OR} &= \{\, (o_{i_1,j}, o_{i_2,j}) \mid o_{i_1,j}, o_{i_2,j} \in V,\ o_{i_1,j} \prec o_{i_2,j} \,\} \end{aligned}$$

heißt *Ablaufgraph (sequence graph)* der Sequenz A. Dabei repräsentieren E_{TR} und E_{OR} die Technologie und Organisation von A.

Dieser Digraph $G(A)$ entspricht einem disjunktiven Graphen, dessen disjunktive Kanten bereits sämtlich auf die im vorangegangenen Abschnitt beschriebene Weise in konjunktive Kanten umgewandelt wurden. In Abbildung 3.2 wird exemplarisch ein Ablaufgraph gezeigt, der aus einer (vollständigen) azyklischen Orientierung des disjunktiven Graphen aus Abbildung 3.1 hervorgeht.

Im Bereich der rein graphentheoretischen Terminologie ist diese Klasse von Digraphen in folgendem Zusammenhang bekannt: Das *kartesisches Produkt* $G = H_1 \times H_2$ zweier Graphen $H_1 = (V_1, E_1)$ und $H_2 = (V_2, E_2)$ ist der Graph $G = (V, E)$ mit $V = V_1 \times V_2$, bei dem zwei Knoten $(v_1, v_2), (w_1, w_2) \in V$ mit $v_i, w_i \in V_i$ genau dann benachbart sind, wenn $v_1 = w_1$ und $(v_2, w_2) \in E_2$ oder $v_2 = w_2$ und $(v_1, w_1) \in E_1$ gilt. Ein *Hamming-Graph* $K_n \times K_m$ ist das kartesische Produkt aus zwei vollständigen Graphen K_n und K_m. Es ist leicht zu sehen, daß der Ablaufgraph $G(A)$ einer Sequenz A für ein Shop-Scheduling-Problem mit n Aufträgen und m Maschinen einer azyklischen Orientierung des indizierten[3] Hamming-Graphen

[3]Ein Graph $G = (V, E)$ heißt *indizierter Graph*, wenn seine Knoten mit festen Indizes identifiziert und durch diese unterschieden werden.

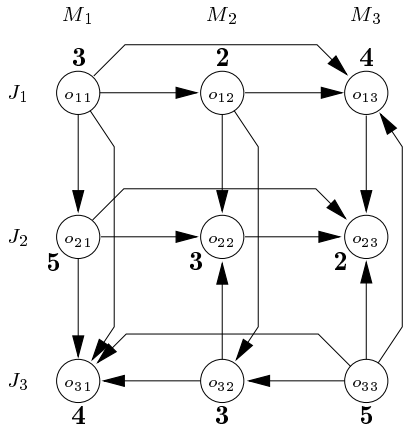


Abbildung 3.2: Ein 3×3–Ablaufgraph.

$K_n \times K_m$ entspricht, dessen Knoten mit den Operationen o_{ij} mit $i = 1, \ldots, n$ und $j = 1, \ldots, m$ identifiziert werden. Daher wird bei diesen azyklischen indizierten Digraphen im weiteren von $n \times m$–*Ablaufgraphen* gesprochen.

Eine Sequenz A läßt sich eindeutig durch den zugehörigen $n \times m$–Ablaufgraphen $G(A)$ beschreiben. Umgekehrt stellt jede azyklische Orientierung des Hamming-Graphen $K_n \times K_m$, dessen Knoten mit den Operationen o_{ij} indiziert sind, eindeutig eine zulässige Kombination von Technologie und Organisation dar, also ist die Zuordnung von Sequenzen zu Ablaufgraphen eineindeutig.

Im Zusammenhang mit Ablaufgraphen ist die sogenannte Prozedur des topologischen Sortierens[4] von großer Bedeutung. Eine *topologische Sortierung* der Knotenmenge V eines Digraphen $G = (V, E)$ mit $|V| = p$ ist eine Abbildung $\varrho : V \to \{1, \ldots, k\}$ mit $k \leq p$, so daß für alle $v, w \in V$ mit $(v, w) \in E$ die Beziehung $\varrho(v) < \varrho(w)$ gilt. Beim topologischen Sortieren der Knoten eines Digraphen G wird versucht, eine solche Abbildung ϱ von V zu finden. Offensichtlich existiert eine topologische Sortierung ϱ von V genau dann, wenn $G = (V, E)$ azyklisch ist.

Für einen azyklischen Digraphen $G = (V, E)$ sei $k_{\min}$ der kleinste Wert, für den eine topologische Sortierung $\varrho : V \to \{1, \ldots, k_{\min}\}$ existiert. Im folgenden wird stets diejenige topologische Sortierung ϱ betrachtet, die jedem Knoten $v \in V$ den kleinsten möglichen Wert aus $\{1, \ldots, k_{\min}\}$ zuordnet. Dann ist ϱ eindeutig bestimmt, und der Wert $\varrho(v)$ heißt *Rang* des Knotens $v \in V$. Der Rang $\varrho(v)$ eines Knotens $v \in V$ entspricht der Knotenanzahl eines längsten Weges in G, der im

[4]Das *topologische Sortieren* wurde erstmals im Zusammenhang mit PERT-Netzwerken behandelt (PERT steht für „Project Evaluation Review Technique“ – siehe Lasser [59] und Kahn [50]).

Knoten v endet, also ist zum Beispiel $\varrho(w) = 1$ für alle Quellen $w \in V$.

Der folgende Satz hilft bei der Beantwortung der Frage, ob es sich bei einem gegebenen Digraphen um den Ablaufgraphen einer Sequenz handelt.

Satz 3.1.2 [12] *Es sei $G = (V, E)$ ein Digraph. Das Problem der Entscheidung, ob die Knoten von G so indiziert werden können, daß G der Ablaufgraph einer Sequenz ist, kann in der Zeit $O(|E|)$ entschieden werden.*

Beweis: Zum Digraphen $G = (V, E)$ wird zunächst der zugrunde liegende Graph[5] $[G]$ betrachtet. Für einen ungerichteten Graphen mit q Kanten kann in der Zeit $O(q)$ festgestellt werden, ob es sich dabei um einen Hamming-Graphen $K_n \times K_m$ handelt (siehe Imrich und Klavžar [47]). Falls $[G]$ tatsächlich ein Hamming-Graph des Typs $K_n \times K_m$ für natürliche Zahlen n und m ist, liefert der in [47] beschriebene Algorithmus eine entsprechende Indizierung der Knoten von $[G]$. Anschließend kann topologisches Sortieren auf V angewandt werden, um zu testen, ob die Orientierung von G azyklisch ist. Die Laufzeit des Algorithmus zur Erzeugung einer topologischen Sortierung beträgt $O(p+q)$ für einen Digraphen mit p Knoten und q Kanten (siehe z. B. Simon [87]). Da für einen Hamming-Graphen $K_n \times K_m$ mit $n, m \geq 2$ die Anzahl seiner Knoten niemals größer als die Anzahl seiner Kanten ist, folgt insgesamt die Aussage des Satzes. □

3.2 Pläne

Für eine Vielzahl der hier untersuchten Algorithmen ist es sinnvoll, die Ablaufgraphen in komprimierter Form anhand von speziellen Matrizen darzustellen. Zu diesem Zweck wird in diesem Abschnitt eine eineindeutige Zuordnung von bestimmten lateinischen Rechtecken zu Ablaufgraphen beschrieben, auf der auch das von Bräsel [7] eingeführte *Blockmatrizenmodell* basiert.

Es sei $n \leq m \leq r$. Ein *lateinisches Rechteck* $\mathcal{L}_{n,m,r}$ ist eine $n \times m$–Matrix mit Einträgen aus der Belegungsmenge $S = \{1, \ldots, r\}$, wobei jeder Eintrag in jeder Zeile und Spalte höchstens einmal auftritt. Ein lateinisches Rechteck mit $n = m = r$ heißt *lateinisches Quadrat*. Viele Probleme im Zusammenhang mit lateinischen Quadraten und Rechtecken werden bei Dénes und Keedwell [29, 30] behandelt. Eine aktuell erschienene Übersicht von Laywine und Mullen [60] zeigt die vielfältigen Anwendungen lateinischer Rechtecke in verschiedenen Bereichen der Diskreten Mathematik. In diesen Monographien ist jedoch nicht die Anwendung lateinischer Rechtecke in der Schedulingtheorie enthalten, die im folgenden beschrieben wird.

[5] Der einem Digraphen G *zugrunde liegende Graph* $[G]$ ist der ungerichtete Graph, der durch Ersetzen der gerichteten durch ungerichtete Kanten in G entsteht.

Definition 3.2.1 Ein $n \times m$*–Plan* ist ein lateinisches Rechteck $\mathcal{L}_{n,m,r} = (l_{ij})$, in dem zu jedem Eintrag $l_{ij} > 1$ der Wert $l_{ij} - 1$ als Eintrag in der Zeile i oder Spalte j auftritt.

Die Menge aller $n \times m$–Pläne bildet eine Klasse von lateinischen Rechtecken des Typs $\mathcal{L}_{n,m,r}$. Der folgende Satz zeigt, daß die Elemente dieser Klasse von lateinischen Rechtecken den im vorangegangenen Abschnitt eingeführten Digraphen zugeordnet werden können.

Satz 3.2.2 *Jedem $n \times m$–Ablaufgraphen, dessen Knoten mit den Operationen o_{ij} eines Shop-Scheduling-Problems identifiziert werden, kann eineindeutig ein $n \times m$–Plan zugeordnet werden.*

Beweis: Es sei $G = (V, E)$ ein $n \times m$–Ablaufgraph, dessen Knoten mit den Operationen o_{ij}, $i = 1, \ldots, n$ und $j = 1, \ldots, m$ eines Shop-Scheduling-Problems mit n Aufträgen und m Maschinen identifiziert werden. Weiter sei $A = (a_{ij})$ die $n \times m$–Matrix, die aus den Rängen ϱ der Knoten von G besteht, also $a_{ij} = \varrho(o_{ij})$ für alle i, j. Für je zwei Operationen $o_{i,j_1}, o_{i,j_2} \in V$ eines Auftrags J_i gilt $(o_{i,j_1}, o_{i,j_2}) \in E$ oder $(o_{i,j_2}, o_{i,j_1}) \in E$, damit ist $a_{i,j_1} \neq a_{i,j_2}$ für alle $j_1 \neq j_2$. Analoges gilt für je zwei Operationen, die zu einer Maschine M_j gehören, also ist A ein lateinisches Rechteck. Weil G nur gerichtete Kanten zwischen Operationen enthält, die zum gleichen Auftrag oder zur gleichen Maschine gehören, erfüllt A die in Definition 3.2.1 beschriebene zusätzliche Bedingung für die Einträge eines Plans.

Sei umgekehrt ein $n \times m$–Plan $A = (a_{ij})$ gegeben. Zu A wird der Digraph $G = (V, E)$ mit $V = \{o_{ij} | 1 \leq i \leq n, 1 \leq j \leq m\}$ und $E = E_{TR} \cup E_{OR}$ definiert, wobei

$$\begin{aligned} E_{TR} &= \{ (o_{i,j_1}, o_{i,j_2}) \mid o_{i,j_1}, o_{i,j_2} \in V,\ a_{i,j_1} < a_{i,j_2} \}, \\ E_{OR} &= \{ (o_{i_1,j}, o_{i_2,j}) \mid o_{i_1,j}, o_{i_2,j} \in V,\ a_{i_1,j} < a_{i_2,j} \} \end{aligned}$$

ist. Die Beziehungen der Form $a_{ij_1} < a_{ij_2}$ induzieren Reihenfolgen $o_{ij_1} \prec o_{ij_2}$, in denen die Operationen o_{ij_1} und o_{ij_2} bearbeitet werden. Die Gesamtheit dieser Beziehungen repräsentiert eine Technologie. Analog dazu ergibt sich die Organisation. Die Existenz eines gerichteten Weges in G von einem Knoten o_{ij} zu einem Knoten o_{kl} setzt die Bedingung $a_{ij} < a_{kl}$ voraus. Wegen $a_{ij} \neq a_{kl}$ für $i = j \ \wedge \ k = l$ ist G azyklisch. Insgesamt folgt also, daß G ein $n \times m$–Ablaufgraph ist. □

Dieser Satz zeigt, daß jeder Plan mit genau einem Ablaufgraphen korrespondiert. In Abbildung 3.4 (Seite 22) ist beispielsweise ein 3×3–Plan und der zugehörige 3×3–Ablaufgraph zu sehen. Aufgrund dieser Korrespondenz können Pläne auch wie folgt definiert werden (vgl. Definition 3.2.1).

Definition 3.2.3 Ein $n \times m$*–Plan* ist eine $n \times m$–Matrix $A = (a_{ij})$, die aus den Rängen der Knoten o_{ij} $(i = 1, \ldots, n;\ j = 1, \ldots, m)$ eines $n \times m$–Ablaufgraphen besteht, also $a_{ij} = \varrho(o_{ij})$ für alle i, j.

Im folgenden wird bei einem Eintrag a_{ij} eines Plans A häufig auch vom *Rang* $\varrho(o_{ij})$ der Operation o_{ij} gesprochen. Ein Ablaufgraph entspricht einer zulässigen Kombination von Technologie und Organisation und damit der in Abschnitt 2.2 definierten Sequenz eines Shop-Scheduling-Problems. Wegen Satz 3.2.2 können solche Sequenzen ebenfalls anhand von Plänen repräsentiert werden. Die Ränge der Knoten des zugehörigen Ablaufgraphen werden durch topologisches Sortieren ermittelt. In diesem Zusammenhang erhält man folgende komplexitätstheoretische Aussage.

Satz 3.2.4 *Für $n \leq m$ kann der Plan zu einem gegebenen $n \times m$–Ablaufgraphen in in der Zeit $O(m^3)$ berechnet werden.*

Beweis: Der Rang eines Knotens v in einem azyklischen Digraphen $G = (V, E)$ ist die Anzahl der Knoten eines in v endenden längsten Weges von G. Die Bestimmung der Ränge der Knoten in G basiert auf einer topologischen Sortierung ϱ von V. Das topologische Sortieren der Knoten eines azyklischen Digraphen mit p Knoten und q Kanten benötigt $O(p + q)$ Zeit (vgl. Satz 3.1.2). Für einen $n \times m$–Ablaufgraphen $G = (V, E)$ gilt $|V| = nm$ und $|E| = nm(m-1)/2 + mn(n-1)/2$, daher können für $n \leq m$ die Ränge seiner Knoten und damit sein zugeordneter Plan mit dem Zeitaufwand $O(m^3)$ bestimmt werden. □

Definition 3.2.5 Ein *reduzierter Ablaufgraph* ist ein Ablaufgraph ohne transitive Kanten.

Folgerung 3.2.6 *Zu einem gegebenen reduzierten $n \times m$–Ablaufgraphen kann der zugehörige Plan in der Zeit $O(nm)$ berechnet werden.*

Beweis: Diese Aussage ergibt sofort sich aus Satz 3.2.4, denn ein reduzierter $n \times m$–Ablaufgraph besitzt maximal $n(m-1) + m(n-1)$ Kanten. □

Oft wird ein zu einer Sequenz zugehöriger Schedule, der zusätzlich zu den gegebenen Reihenfolgen der Sequenz die Informationen über die Fertigstellungszeiten C_i der Aufträge J_i enthält, anhand des sogenannten *Gantt-Diagramms* dargestellt (vgl. Abbildung 3.3). Erstmals werden derartige Diagramme in Clark [24] und Porter [75] erwähnt. Gantt-Diagramme können je nach der Bedeutung ihrer vertikalen Achse entweder *auftrags-* oder *maschinenorientiert* sein. Das in Abbildung 3.3 dargestellte Gantt-Diagramm ist maschinenorientiert. Die horizontale Achse ist die Zeitachse; an ihr können die Start- und Fertigstellungszeiten jeder Operation o_{ij} abgelesen werden.

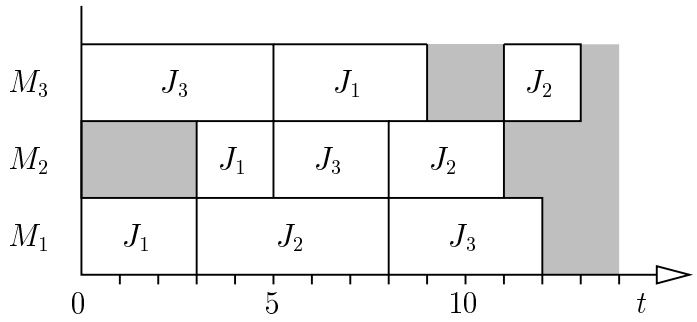


Abbildung 3.3: Das Gantt-Diagramm eines semiaktiven Schedules.

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 4 & 3 & 1 \end{pmatrix}$$

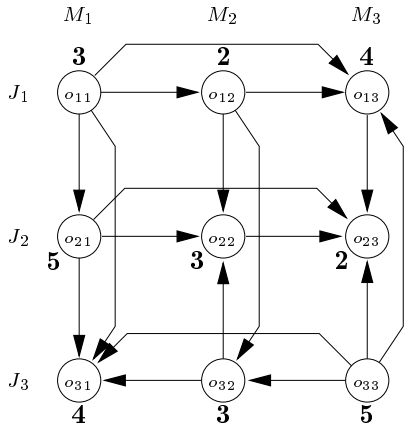


Abbildung 3.4: Ein Plan A und sein zugeordneter Ablaufgraph $G(A)$.

Eine alternative Methode für die Modellierung von Lösungen bzw. Schedules von Shop-Scheduling-Probleme stellt das auf Bräsel [7] zurückgehende *Blockmatrizenmodells* dar. Es sei A ein beliebiger $n \times m$–Plan und $G(A)$ der zugeordnete $n \times m$–Ablaufgraph (siehe Abbildung 3.4). Neben dem Plan A gehören im Blockmatrizenmodell noch vier weitere Typen von $n \times m$–Matrizen zur Beschreibung eines gegebenen Shop-Scheduling-Problems und einer zugehörigen zulässigen Lösung:

$TR = (tr_{ij})$: Die *Technologie-Matrix TR* besteht aus Zeilen, die Permutationen der Zahlen $1, \ldots, m$ sind. Diese Permutationen geben die technologischen Reihenfolgen wieder: Ein Eintrag tr_{ij} bedeutet die Position der Maschine M_j in der technologischen Reihenfolge des Auftrags J_i.

$OR = (or_{ij})$: Die *Organisations-Matrix OR* besteht aus Spalten, die jeweils Permutationen der Zahlen $1, \ldots, n$ sind und damit die organisatorischen Reihen-

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 4 & 3 & 1 \end{pmatrix} \quad TR = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \quad OR = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 3 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

$$P = \begin{pmatrix} 3 & 2 & 4 \\ 5 & 3 & 2 \\ 4 & 3 & 5 \end{pmatrix} \quad C = \begin{pmatrix} 3 & 5 & 9 \\ 8 & 11 & 13 \\ 12 & 8 & 5 \end{pmatrix}$$

Abbildung 3.5: Beispiel-Matrizen im Blockmatrizenmodell.

folgen angeben: Ein Eintrag or_{ij} beschreibt die Position des Auftrags J_i in der organisatorischen Reihenfolge auf der Maschine M_j.

$P = (p_{ij})$: Für alle i, j gibt der Eintrag p_{ij} in der *Bearbeitungszeit-Matrix* P für die Operationen o_{ij} die zugehörige Bearbeitungszeit p_{ij} an.

$C = (c_{ij})$: Der Eintrag c_{ij} in der Matrix C entspricht für alle i, j der *Fertigstellungszeit c_{ij} der Operation o_{ij}*. Die Einträge dieser Matrix C lassen sich aus dem zugrunde liegenden Plan A zusammen mit der Bearbeitungsmatrix P in der Zeit $O(nm)$ bestimmen, wenn der Plan A günstig abgespeichert ist (siehe Bräsel [8]). Bei der Matrixdarstellung $C = (c_{ij})$ eines Schedules kann die Gesamtbearbeitungszeit $C_{\max}$ des zugehörigen Plans A mit Hilfe der Gleichung

$$C_{\max} = \max_{\substack{1 \le i \le n \\ 1 \le j \le m}} (c_{ij}) \tag{3.1}$$

angegeben werden. Neben dem Gantt-Diagramm handelt es sich bei der Matrix C um eine weitere Möglichkeit, einen semiaktiven Schedule darzustellen.

Man stellt fest, daß für Shop-Scheduling-Probleme im Blockmatrizenmodell die Sequenzen durch Pläne und die Schedules durch Matrizen C der Fertigstellungszeiten repräsentiert werden.

Zum Plan A und seinem 3×3–Ablaufgraphen $G(A)$ aus Abbildung 3.4 sind in Abbildung 3.5 die zugehörigen Matrizen TR und OR zu finden. Weiterhin ist die Matrix P mit Bearbeitungszeiten p_{ij} aufgeführt, die mit den Gewichten der Knoten o_{ij} in Abbildung 3.4 übereinstimmen. Für den sich aus A und P ergebenen semiaktiven Schedule (Matrix C in Abbildung 3.5) gilt in diesem Fall $C_{\max} = 13$. Dieser Schedule korrespondiert mit dem in Form eines Gantt-Diagramms gezeichneten Schedule aus Abbildung 3.3. Das Gantt-Diagramm stellt damit den eindeutig

bestimmten semiaktiven Schedule zum Ablaufgraphen $G(A)$ aus Abbildung 3.4 dar, dessen Knoten mit den vorgegebenen Bearbeitungszeiten p_{ij} gewichtet sind.

Kapitel 4

Plan-Anzahlen

In diesem Kapitel werden Ergebnisse über die Anzahl der Elemente in verschiedenen Plan-Klassen erläutert. Jedem Plan entspricht ein spezielles lateinisches Rechteck. Viele der hier vorgestellten Resultate stammen daher aus Arbeiten über die Anzahl lateinischer Rechtecke. Darüber hinaus werden bisher unbekannte Anzahlen für allgemeine Pläne bestimmt, die sich nicht direkt auf die Anzahlen entsprechender lateinischer Rechtecke zurückführen lassen. Zunächst werden im ersten Abschnitt diejenigen Pläne behandelt, die mit den sogenannten *klassischen* lateinischen Rechtecken korrespondieren.

4.1 Rangminimale Pläne

Ein $n \times m$–Plan ($n \leq m$) mit maximalem Eintrag m heißt *rangminimaler $n \times m$–Plan*. Jeder rangminimale $n \times m$–Plan repräsentiert offensichtlich eine optimale Sequenz für das zugehörige Open-Shop-Problem $Om|p_{ij} = 1|\,C_{\max}$ mit n Aufträgen. Dies ist ein Shop-Scheduling-Problem mit Einheitsbearbeitungszeiten. Einen Überblick über die Komplexität von Open-Shop-Problemen mit Einheitsbearbeitungszeiten ist bei Brucker *et al.* [19] und Tautenhahn [95] zu finden.

Es sei $L(n, m, r)$ mit $n \leq m \leq r$ die Anzahl der lateinischen Rechtecke $\mathcal{L}_{n,m,r}$ mit Einträgen aus $\{1, 2, \ldots, r\}$. Im Fall $r = m$ entspricht $L(n, m, m)$ der Anzahl $P(n, m)$ der rangminimalen $n \times m$–Pläne, denn die Bedingung für Pläne aus Definition 3.2.1 ist bei lateinischen Rechtecken $\mathcal{L}_{n,m,m}$ trivialer Weise erfüllt. Die Anzahl $P(n, m)$ ist in der Literatur auch als Anzahl *klassischer lateinischer Rechtecke* ($\mathcal{L}_{n,m,m} := \mathcal{L}_{n,m}$) bekannt.

Die Anzahl $P(n) := P(n, n)$ der quadratischen rangminimalen Pläne entspricht der Anzahl lateinischer Quadrate der Ordnung n. Die Bestimmung dieser Anzahl ist das ursprüngliche und bekannteste Enumerationsproblem in diesem Bereich. Die Berechnung von $P(n)$ erweist sich bereits für $n \geq 6$ als eine nicht-triviale Aufgabe.

n	$\dfrac{P(n)}{n!(n-1)!}$
1	1
2	1
3	1
4	4
5	56
6	9 408
7	16 942 080
8	535 281 401 856
9	377 597 570 964 258 816
10	7 580 721 483 160 132 811 489 280

Tabelle 4.1: Anzahlen rangminimaler quadratischer Pläne.

Im Zusammenhang mit den bis heute bekannten Werten für $n \leq 10$ (vgl. Tabelle 4.1) sind viele Arbeiten verschiedener Autoren entstanden. An dieser Stelle sei erwähnt, daß die Berechnung des bisher größten bekannten Wertes $P(10)$ von McKay und Rogoyski [68] im Jahre 1995 veröffentlicht wurde. Verschiedene Quellen für die Berechnungen der Anzahlen $P(n)$ mit $6 \leq n \leq 9$ können z.B. in [29, 68] nachgeschlagen werden.

Geschlossene Formeln

Auch die Behandlung der Werte $P(n,m)$ für $n \neq m$ tritt in der Literatur vielfach auf. Es sind jedoch keine exakte Formeln für $m \geq n \geq 5$ bekannt. Offensichtlich gilt $P(1,m) = m!$. Mit Hilfe der Rencontre-Zahlen[1] D_m läßt sich $P(2,m)$ darstellen (siehe z.B. [76]):

Satz 4.1.1 *Für alle $m \geq 2$ gilt*

$$P(2,m) = m!\ D_m \qquad mit \qquad D_m = m! \sum_{k=0}^{m} \frac{(-1)^k}{k!}.$$

[1]Die *Rencontre-Zahl* D_m ist die Anzahl der Derangements der Ordnung m. Ein *Derangement* der Ordnung m ist eine fixpunktfreie Permutation der Ordnung m, also eine Permutation, die in keiner Position mit der Identität übereinstimmt.

m	$\frac{P(2,m)}{m!}$	$\frac{P(3,m)}{m!}$	$\frac{P(4,m)}{m!}$
2	1		
3	2	2	
4	9	24	24
5	44	552	1 344
6	265	21 280	393 120
7	1 854	1 073 760	155 185 920
8	14 833	70 299 264	88 390 995 840
9	133 496	5 792 853 248	69 761 852 246 016
10	1 334 961	587 159 944 704	74 175 958 614 030 336

Tabelle 4.2: Anzahlen rangminimaler $n \times m$–Pläne für $n = 2, 3$ und 4.

Die Anzahl $P(3,m)$ ist unter verschiedenen Gesichtspunkten untersucht worden, da ihre Bestimmung die ersten Schwierigkeiten darstellt, siehe Bogart und Longyear [6], Jacob [48], Kerawala [52, 53], Riordan [80, 81, 82], und Yamamoto [104]. In [81] gibt Riordan eine elegante Formel für $P(3,m)$ an, die auf die Rencontre-Zahlen D_m und die Ménage-Zahlen[2] U_m zurückgeht.

Satz 4.1.2 [81] *Für alle $m \geq 3$ gilt*

$$\begin{aligned} P(3,m) &= m! \sum_{k=0}^{\lfloor m/2 \rfloor} \binom{m}{k} D_k D_{m-k} U_{m-2k} \\ &\quad mit \quad U_m = \sum_{k=0}^{m} (-1)^k \frac{2m}{2m-k} \binom{2m-k}{k} (m-k)! \\ &\quad und \quad U_0 = 1. \end{aligned}$$

Für $n = 4$ hat Light in [61] ein Verfahren zur Enumeration rangminimaler $4 \times m$–Pläne entwickelt, das auf einer Reihe von Rekursionen im Zusammenhang mit bestimmten Diagrammen beruht. Mit deren Hilfe konnte er die Werte $P(4,m)$ für $m \leq 8$ bestimmen. Im Fall $m = 8$ hat Light allerdings irrtümlich einen falschen Wert angegeben, wie ein Vergleich von $P(4,8)$ mit den entsprechenden Resultaten in [68, 71] zeigt. Die Autoren Mullen und Purdy haben in [71] einige Fehler

[2] Die *Ménage-Zahl* U_m ist die Anzahl der Permutationen der Ordnung m, die jeweils in keiner Position mit der Identität und einem Zyklus der Länge m übereinstimmen.

in der Literatur zur Enumeration lateinischer Rechtecke bzw. rangminimaler Pläne aufgedeckt. Der falsche Wert $P(4,8)$ aus [61] ist ihnen allerdings verborgen geblieben, obwohl sie diesen Wert selbst korrekt auflisten und die Arbeit von LIGHT [61] zitieren.

Eine Übersicht über die Anzahl rangminimaler zwei-, drei- und vierzeiliger Pläne mit jeweils bis zu zehn Spalten gibt Tabelle 4.2. Die Werte für $P(4,m)$ wurden aus [68] übernommen, da die Werte $P(4,m)$ aus [61] nur bis $m = 7$ korrekt angegeben sind.

Die in den oben zitierten Arbeiten angewandten Methoden zur Berechnung der Werte $P(n,m)$ mit festem n sind nicht einheitlich und ergeben für $n \geq 5$ keine befriedigenden Resultate. Eine mögliche Begründung für die Unbrauchbarkeit dieser Methoden bei größeren Formaten liefert der nächste Unterabschnitt über die Erweiterung rangminimaler Pläne.

Es besteht ein Zusammenhang zwischen der Anzahl der rangminimalen Pläne und sogenannten Permanenten; dies zeigt Satz 4.1.4, der eine allgemeine Formel für $P(n,m)$ darstellt. Zur Vorbereitung benötigen wir:

Definition 4.1.3 Es sei $n \leq m$ und $S_m(n)$ die Menge aller n-Permutationen der Elemente $\{1,\ldots,m\}$. Die *Permanente* einer $n \times m$–Matrix $B = (b_{ij})$ ist durch

$$\operatorname{per}(B) = \sum_{\pi \in S_m(n)} b_{1,\pi(1)} b_{2,\pi(2)} \cdots b_{n,\pi(n)} \tag{4.1}$$

definiert.

Bei der Permanente einer Matrix B haben alle Terme im Gegensatz zur Determinante von B positives Vorzeichen. Obwohl die Determinante einer $n \times n$–Matrix effizient (also polynomial) berechnet werden kann, ist zur Berechnung ihrer Permanente unter der Annahme $\mathcal{P} \neq \mathcal{NP}$ die Existenz eines polynomialen Algorithmus nicht zu erwarten (siehe VALIANT [99]). Eine Formel für $P(n,m)$, die auf Permanenten von $(0,1)$–Matrizen beruht, ist bei SHAO und WEI [85] zu finden. Mit Hilfe von Permutationsmatrizen[3] und dem Prinzip der Inklusion und Exklusion wurde der folgenden Satz beweisen.

Satz 4.1.4 [85] *Für alle $n, m \in \mathbb{N}$ mit $n \leq m$ gilt*

$$P(n,m) = m! \sum_{B \in \mathcal{B}_{n,m}} (-1)^{\sigma(B)} \binom{\operatorname{per}(B)}{m}, \tag{4.2}$$

wobei $\mathcal{B}_{n,m}$ die Menge aller $n \times m$–$(0,1)$–Matrizen ist und $\sigma(B)$ die Anzahl der Null-Elemente von B angibt.

[3]Eine *Permutations-Matrix* ist eine $(0,1)$–Matrix, in der jede Zeile und jede Spalte genau ein Eins-Element enthält.

Trotz der Einfachheit von (4.2) ist dieser Ausdruck für eine effiziente Methode zur expliziten Berechnung der Werte $P(n,m)$ für größere n und m nicht geeignet, da die Anzahl der Terme in (4.2) gemäß (4.1) mit n bzw. m exponentiell wächst. Im anschließenden Abschnitt wird deutlich, daß die Permanenten bestimmter $(0,1)$–Matrizen auch bei der Erweiterung rangminimaler Pläne von großer Bedeutung sind.

Erweiterung rangminimaler Pläne

Die Ansätze zur Bestimmung der Anzahlen $P(n,m)$ für $n \leq 4$ beruhen hauptsächlich auf der Abzählung der Möglichkeiten, eine weitere Zeile zu rangminimalen Plänen so hinzuzufügen, daß wiederum rangminimale Pläne des gewünschten größeren Formats entstehen. Daß die Existenz von Erweiterungen rangminimaler Pläne immer gesichert ist, zeigt der anschließende Satz.

Satz 4.1.5 [29] *Es sei $n < m$. Jeder rangminimale $n \times m$–Plan ist zu einem quadratischen rangminimalen $m \times m$–Plan erweiterbar.*

Es ist nun von Interesse, ob das Prinzip der Erweiterung rangminimaler Pläne ebenfalls zur effizienten Berechnung der Anzahl entsprechender Pläne größeren Formats angewandt werden kann. Es wird zunächst anhand eines Beispiels der Zusammenhang zwischen der zeilenweisen Erweiterung rangminimaler $n \times m$–Pläne und perfekten Matchings[4] in $(m-n)$–regulären Teilgraphen des $K_{m,m}$ sowie Permanenten assoziierter $m \times m$–$(0,1)$–Matrizen veranschaulicht.

Beispiel 4.1.6 Es sei ein rangminimaler 3×5–Plan A gegeben mit

$$A = \begin{pmatrix} 1 & 4 & 3 & 5 & 2 \\ 2 & 5 & 1 & 3 & 4 \\ 4 & 3 & 2 & 1 & 5 \end{pmatrix}. \tag{4.3}$$

Zu A wird der 2-reguläre bipartite Graph $G_A = (V,E)$ mit $V = \{v_1, \ldots, v_5\} \cup \{w_1, \ldots, w_5\}$ und

$$E = \{\{v_i, w_j\} \,|\, \text{Eintrag } i \text{ existiert } \textit{nicht} \text{ in Spalte } j \text{ von } A\},$$

definiert, siehe Abbildung 4.1. Offensichtlich ist die Anzahl der Möglichkeiten, A zu einem rangminimalen 4×5–Plan zu erweitern, gleich der Anzahl perfekter Matchings in G_A, denn jedes perfekte Matching von G_A entspricht einer möglichen 4. Zeile. Die Anzahl der Erweiterungsmöglichkeiten kann ebenso mittels der Permanente einer

[4] Ein *perfektes Matching* eines Graphen $G = (V,E)$ mit $|V| = 2p$ ist eine Menge von p Kanten aus E, in der keine zwei Kanten einen gemeinsamen Knoten besitzen.

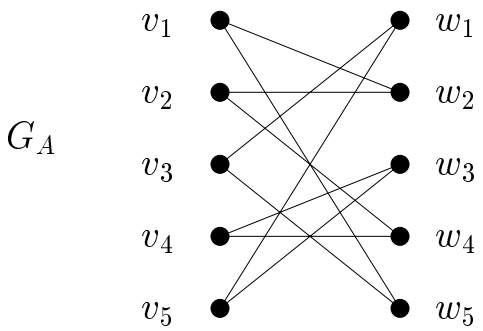


Abbildung 4.1: Der bipartite Graph G_A zu Beispiel 4.1.6.

bestimmten quadratischen $(0,1)$–Matrix ausgedrückt werden. Im quadratischen Fall ergibt sich aus (4.1) die Beziehung

$$\mathrm{per}(B) = \sum_{\pi \in S_m} b_{1,\pi(1)} b_{2,\pi(2)} \cdots b_{m,\pi(m)} \tag{4.4}$$

als Definition der Permanente einer $m \times m$–Matrix B, wobei S_m die Menge aller Permutationen der Zahlen $\{1, \ldots, m\}$ ist. Zu einem Plan A sei die $(0,1)$–Matrix $B = (b_{ij})$ durch

$$b_{ij} = \begin{cases} 1, \text{ falls Eintrag } i \textit{ nicht} \text{ in Spalte } j \text{ von } A \text{ existiert;} \\ 0, \text{ sonst;} \end{cases} \tag{4.5}$$

gegeben. Für den gemäß (4.3) gegebenen Plan A bekommt man also die 5×5–Matrix

$$B = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Offensichtlich betragen die Zeilen- und Spaltensummen von B jeweils $5-3=2$, und der Ausruck $\mathrm{per}(B)$ gibt gerade die Anzahl der Möglichkeiten an, den rangminimalen Plan A zu einem rangminimalen 4×5–Plan zu erweitern. In unserem Beispiel gilt $\mathrm{per}(B) = 2$ und die Matrizen

$$A' = \begin{pmatrix} 1 & 4 & 3 & 5 & 2 \\ 2 & 5 & 1 & 3 & 4 \\ 4 & 3 & 2 & 1 & 5 \\ 3 & 2 & 5 & 4 & 1 \end{pmatrix} \quad \text{und} \quad A'' = \begin{pmatrix} 1 & 4 & 3 & 5 & 2 \\ 2 & 5 & 1 & 3 & 4 \\ 4 & 3 & 2 & 1 & 5 \\ 5 & 1 & 4 & 2 & 3 \end{pmatrix}$$

sind die beiden Erweiterungsmöglichkeiten von A zu rangminimalen 4×5–Plänen.

In [56] hat U. KLEINAU gezeigt, daß die Enumeration der Möglichkeiten, einen gegebenen rangminimalen $n \times m$–Plan mit $n < m$ durch Hinzufügen einer Zeile zu einem rangminimalen $(n+1) \times m$–Plan zu erweitern, $\mathcal{NP}$-schwer ist:

Satz 4.1.7 [56] *Das Problem der Enumeration rangminimaler Pläne durch ein zeilenweise Erweiterung rangminimaler Pläne kleineren Formats ist $\#\mathcal{P}$-vollständig.*

Die Terminologie für die Komplexitätsklassen von Enumerationsproblemen (insbesondere die Klasse der $\#\mathcal{P}$-vollständigen Probleme) wird in Abschnitt 5.5 ausführlich erläutert (siehe dazu auch [99, 100]). Die Aussage dieses Satzes ist in [56] durch eine polynomiale Reduktion des Problems der Enumeration perfekter Matchings in regulären bipartiten Graphen auf das Enumerationsproblem für rangminimale Pläne bewiesen. Für das erste Problem hat U. KLEINAU in [56] die $\#\mathcal{P}$-Vollständigkeit gezeigt.

Mit diesem Resultat wissen wir, daß die Existenz eines polynomialen Algorithmus zur Abzählung der Erweiterungsmöglichkeiten rangminimaler Pläne unwahrscheinlich ist. Das Ergebnis verdeutlicht die Ursache des Scheiterns aller Bemühungen, anhand zeilenweiser Erweiterung von rangminimalen Plänen das Problem der Bestimmung der Anzahl rangminimaler $n \times m$–Pläne allgemein lösen zu können.

Komplettierung partieller lateinischer Rechtecke

Eine zu Satz 4.1.7 verwandte Aussage stammt aus einer Arbeit von COLBOURN [25] von 1984 und wird im folgenden vorgestellt.

Definition 4.1.8 Es sei $n \leq m$. Ein *partielles lateinisches Rechteck* ist ein lateinisches Rechteck $\mathcal{L}_{n,m}$, bei dem einige der nm Einträge aus $\{1, \ldots, m\}$ fehlen.

Satz 4.1.9 [25] *Das Entscheidungsproblem „Ist ein gegebenes partielles lateinisches Rechteck komplettierbar?“ ist $\mathcal{NP}$-vollständig.*

Bei diesem Ansatz wird von partiellen Matrizen ausgegangen, bei denen beliebige Zellen unbesetzt sein können, während in [56] ausschließlich ganze Zeilen hinzugefügt werden. Der Beweis von Satz 4.1.9 wird durch polynomiale Reduktion eines graphentheoretischen Problems geführt, das dem Problem der perfekten Matchings in [56] ähnelt: Besitzt ein gegebener tripartiter Graph eine Partition der Kantenmenge in Dreiecke?

Dieser Satz zeigt, daß im Fall $\mathcal{P} \neq \mathcal{NP}$ keine gute Charakterisierung für komplettierbare partielle lateinische Rechtecke zu erwarten ist. Das mit dem $\mathcal{NP}$-vollständigen Entscheidungsproblem in [25] assoziierte Enumerationsproblem, also das Problem der Enumeration der verschiedenen Möglichkeiten, ein partielles

lateinisches Rechteck zu komplettieren, ist offensichtlich auch $\mathcal{NP}$-schwer. Nichtsdestotrotz ist im Laufe der Zeit eine Vielzahl von notwendigen und hinreichenden Bedingungen für die Komplettierbarkeit partieller lateinischer Rechtecke entwickelt worden, siehe dazu TAUTENHAHN [95] und VAN LINT [101].

In einer 1998 erschienenen Arbeit entwickeln MCKAY und WANLESS [69] die rangminimalen $n \times m$–Pläne, die die meisten Erweiterungen zu entsprechenden $(n+1) \times m$–Plänen besitzen. Bei diesen Untersuchungen wird die Äquivalenz zum Problem der Bestimmung maximaler Permanenten von $m \times m$–$(0,1)$–Matrizen sowie zum Problem der Bestimmung $(m-n)$-regulärer Teilgraphen des $K_{m,m}$ mit maximaler Anzahl perfekter Matchings ausgenutzt. Die Bedingungen für rangminimale Pläne mit maximaler Anzahl der Erweiterungsmöglichkeiten werden meist in Form von Eigenschaften im zugrundeliegenden regulären bipartiten Graphen ausgedrückt. Das Problem der Identifizierung der rangminimalen $n \times m$–Pläne, die eine maximale Anzahl der Erweiterungsmöglichkeiten besitzen, ist in [69] zum einen für $n = 2$, m beliebig und zum anderen für $n \geq 2, k \geq 5$ und $m = kn$ komplett gelöst worden.

Asymptotische Resultate

Da sich die exakte Bestimmung der rangminimalen Pläne für größere Formate als außerordentlich schwierig gestaltet, sind asymptotische Resultate von Interesse. Ein asymptotischer Ausdruck für $P(n,m)$ ist erstmals 1946 in einer Arbeit von ERDÖS und KAPLANSKY [33] erschienen.

Satz 4.1.10 [33] *Für $n = o((\log m)^{3/2})$ gilt*

$$P(n,m) \sim (m!)^n \; e^{-n(n-1)/2}. \tag{4.6}$$

In [105] hat YAMAMOTO gezeigt, daß (4.6) sogar für $n = o(m^{1/3})$ gilt. Eine weitere Verbesserung dieses Resultats ist STEIN [93] im Jahr 1978 gelungen.

Satz 4.1.11 [93] *Für $n = o(m^{1/2})$ gilt*

$$P(n,m) \sim (m!)^n \; e^{-\binom{n}{2} - \frac{n^3}{6m}}. \tag{4.7}$$

In einer kürzlich erschienenen Arbeit verwendet SKAU [88] Ideen von VAN LINT [101], um zu zeigen, daß die Schranke $n = o(m^{1/2})$ für die Gültigkeit von (4.7) bestmöglich ist. Genauer gesagt wächst $P(n,m)$ für $n > m^{1/2+\varepsilon}$ mit $\varepsilon > 0$ langsamer als die rechte Seite von (4.7). Wie viele Resultate im Bereich der asymptotischen Bestimmung von $P(n,m)$ entsteht auch das Ergebnis in [88] durch die Abschätzung der Permanente $\mathrm{per}(B)$ der $m \times m$–$(0,1)$–Matrix B, die die Anzahl der Möglichkeiten angibt, eine $(n+1)$–te Zeile zu einem rangminimalen $n \times m$–Plan mit $n < m$ hinzuzufügen.

Umfangreiche probabilistische Untersuchungen von GODSIL und MCKAY [38] haben in diesem Zusammenhang eine wesentliche Verbesserung der vorangegangenen Resultate gebracht. Das im anschließenden Satz zusammengefaßte Ergebnis stellt zur Zeit die beste bekannte asymptotische Abschätzung für $P(n,m)$ dar.

Satz 4.1.12 [38] *Für $n = o(m^{6/7})$ gilt*

$$P(n,m) \sim (m!)^n \left(\frac{m(m-1)\cdots(m-n+1)}{m^n}\right)^m \left(1-\frac{n}{m}\right)^{-m/2} e^{-n/2}.$$

Eine interessante Verallgemeinerung des oben erwähnten Ergebnisses von YAMAMOTO [105] ist die Bestimmung der asymptotischen Anzahl der sogenannten B-lateinischen Rechtecke durch GREEN [42], die mit Hilfe der Techniken erzielt wurden, die bereits STEIN [93] benutzt hat.

Es sei B eine feste Menge mit $B \subset \mathbb{N}$. Eine $n \times m$–Matrix mit Einträgen aus $S = \{1, \ldots, m\}$ heißt *B-lateinisches Rechteck*, wenn jeder Eintrag in jeder Zeile genau einmal, und jeder Eintrag aus $B_m = B \cap S$ in jeder Spalte höchstens einmal auftritt. Das heißt, in einem B-lateinischen Rechteck ist im Gegensatz zum gewöhnlichen lateinischen Rechteck nur für einen Teil der Einträge aus S eine Wiederholung innerhalb der Spalten verboten. Für $B = S$ handelt es sich dagegen um ein gewöhnliches lateinisches Rechteck.

Der Ausdruck $L_B(n,m)$ bezeichne die Anzahl der B-lateinischen Rechtecke des Formats $n \times m$.

Satz 4.1.13 [42] *Wenn $1/|B_m| = O(n^{-\alpha})$ für ein festes α mit $\frac{1}{2} < \alpha \leq 1$ ist, dann gilt für $n = o(m^{(2\alpha-1)/3})$ die Abschätzung*

$$L_B(n,m)^{m/|B_m|} \sim (m!)^n e^{n(n-1)/2}.$$

Im Fall gewöhnlicher lateinischer Rechtecke ist $|B_m| = m$. Dieser Satz reduziert sich dann zu Satz 4.1.10 von ERDÖS und KAPLANSKY [33] mit der Schranke $n = o(m^{1/3})$ von YAMAMOTO [105].

4.2 Allgemeine lateinische Rechtecke

Im Fall der Anzahl allgemeiner lateinischer Rechtecke lassen sich nur die Werte $L(1,m,r)$ und $L(2,m,r)$ ohne große Schwierigkeiten exakt bestimmen (siehe z. B. PRANESACHAR [76]). Offensichtlich gilt $L(1,m,r) = r!/(r-m)!$.

Satz 4.2.1 *Für alle $m, r \in \mathbb{N}$ gilt*

$$L(2,m,r) = \frac{r!}{(r-m)!} \sum_{k=0}^{m} (-1)^k \binom{m}{k} \frac{(r-k)!}{(r-m)!}.$$

In [3] haben ATHREYA, PRANESACHAR und SINGHI die Technik der Möbius-Inversion[5] benutzt, um eine einheitliche Methode zur Enumeration von lateinischen Rechtecken zu entwickeln. Mit Hilfe dieser Methode sind die Anzahlen $L(n,m,r)$ für $n = 3,4$ berechnet worden. Die Überlegungen in [3] beruhen auf einer Korrespondenz zwischen allgemeinen lateinischen Rechtecken und sogenannten zulässigen Färbungen bestimmter Graphen.

Eine *zulässige λ-Färbung* eines Graphen $G = (V,E)$ ist eine Funktion $f : V \to \{1,2,\ldots,\lambda\}$ mit $f(v) \neq f(w)$ für alle $v,w \in V$ mit $\{v,w\} \in E$. Das *chromatische Polynom* $\chi(G,\lambda)$ eines ungerichteten Graphen $G = (V,E)$ ist das eindeutig bestimmte Polynom in λ, das für alle $\lambda \in \mathbb{N}$ die Anzahl der zulässigen λ–Färbungen von G angibt.[6] Ein *vollständiger bipartiter Graph* $K_{n,m}$ ist ein Graph, dessen Knotenmenge so in zwei Mengen V_1 und V_2 mit $|V_1| = n, |V_2| = m$ partitioniert werden kann, daß jeder Knoten aus V_1 mit jedem aus V_2 benachbart ist und sonst keine Nachbarschaften bestehen. Der *Kantengraph (line graph)* eines Graphen $G = (V,E)$ ist der Graph $l(G) = (V_l, E_l)$ mit $V_l = E$, bei dem je zwei Knoten aus V_l genau dann benachbart sind, wenn die entsprechenden Kanten in G einen gemeinsamen Endknoten haben.

Hilfssatz 4.2.2 *Für alle $n,m,r \in \mathbb{N}$ gilt $L(n,m,r) = \chi(l(K_{n,m}),r)$.*

Beweis: Offensichtlich entspricht der Kantengraph $l(K_{n,m})$ dem Hamming-Graphen $K_n \times K_m$. In diesem Graphen korrespondiert jeder Knoten mit einem Eintrag des lateinischen Rechtecks, so daß jeweils alle Knoten einer Zeile bzw. einer Spalte paarweise benachbart sind und bei einer zulässigen Färbung verschiedene Farben besitzen. Der Wert $\chi(K_n \times K_m, r)$ gibt die Anzahl der möglichen r-Färbungen des Hamming-Graphen Graphen $K_n \times K_m$ an. Diese Anzahl entspricht damit der Anzahl $L(n,m,r)$, da die Einträge $1,\ldots,r$ als r verschiedene Farben aufgefaßt werden können. □

Eine Spezialisierung der Resultate in [3] ist eine Formel, die das chromatische Polynom von $l(K_{n,m})$, also die Anzahl $L(n,m,r)$, als Linearkombination der chromatischen Polynome bestimmter, durch Partitionen konstruierter Graphen ausdrückt. Auf diese Weise können die Zahlen $L(3,m,r)$ und $L(4,m,r)$ berechnet werden.

Satz 4.2.3 [3] *Für alle $m,r \in \mathbb{N}$ gilt*

$$L(3,m,r) = \frac{r!m!}{((r-m)!)^3} \sum_{\alpha+\beta+\gamma=m} (-1)^\beta 2^\gamma \frac{((r-m+\alpha)!)^2}{\alpha!\gamma!} \binom{3r-3m+3\alpha+\beta+2}{\beta}.$$

[5] Die *Möbius-Inversion* ist eine effiziente Methode zur Berechnung der Summanden, die bei der Anwendung des Prinzips der Inklusion und Exklusion vorkommen (siehe ROTA [83]).

[6] Die Funktion $\chi(G,\lambda)$ ist von BIRKHOFF [4] erstmals 1912 eingeführt worden. Es ist leicht zu zeigen, daß es sich bei $\chi(G,\lambda)$ tatsächlich um ein Polynom in λ handelt (siehe z. B. [77]).

Da die Formel in [3] für $L(4,m,r)$ sehr umfangreich ist, wird auf deren Darstellung hier verzichtet. Für $r = m$ ergeben sich aus Satz 4.2.1 und Satz 4.2.3 die Werte $P(2,m)$ und $P(3,m)$ gemäß Satz 4.1.1 und 4.1.2. Tabelle 4.3 auf Seite 37 und Tabelle 4.4 auf Seite 42 enthalten die Anzahlen $L(2,m,r)$ und $L(3,m,r)$ für $m \leq 10$, jeweils als Summe über alle möglichen Werte r mit $m \leq r \leq nm$.

In [72] hat NECHVATAL, ebenfalls mit Hilfe der Technik der Möbius-Inversion, asymptotische Ergebnisse für $L(n,m,r)$ erzielt. Diese Ergebnisse können als Verallgemeinerung der Resultate von ERDÖS und KAPLANSKY [33] für $P(n,m)$ im vorangegangenen Unterabschnitt aufgefaßt werden.

4.3 Allgemeine Pläne

Es sei $P(n,m,r)$ die Anzahl der $n \times m$–Pläne mit maximalem Eintrag bzw. Rang r, wobei $m \leq r \leq nm$ ist. Aufgrund der Definition der Pläne ist es offensichtlich, daß für alle r die Beziehung $P(n,m,r) \leq L(n,m,r)$ gilt, d. h. die Anzahl lateinischer Rechtecke ist im allgemeinen nur eine grobe obere Schranke für die entsprechende Anzahl der Pläne. Es sind für festes n weniger Anzahlen $P(n,m,r)$ als $L(n,m,r)$ bekannt.

In [14] haben BRÄSEL und M. KLEINAU im Jahr 1992 eine Enumerationsmethode für die Anzahl der $n \times m$–Pläne für kleine Werte von n und m vorgestellt. In Kapitel 5 wird eine Weiterentwicklung des Algorithmus aus [14] behandelt, die zu einer effektiveren Plan-Enumeration führt.

Im Rahmen der folgenden Unterabschnitte wird eine bekannte exakte Formel für die Anzahl aller $2 \times m$–Pläne angegeben, und es werden neue Ergebnisse für die Anzahlen aller $3 \times m$– und $4 \times m$–Pläne entwickelt.

Pläne des Formats $2 \times m$

Für die Gesamtanzahl aller $n \times m$–Pläne wird $P_{n,m} := \sum_{r=m}^{nm} P(n,m,r)$ geschrieben. Ein geschlossener Ausdruck für $P_{2,m}$ erscheint erstmals bei BRÄSEL und M. KLEINAU [13]. Diese Werte geben Aufschluß über die Anzahl aller zulässigen Kombinationen von Technologien und Organisationen des Problems $Om|n = 2|C_{\max}$ bzw. die entsprechende Anzahl für das Problem $O2||C_{\max}$ mit Auftragsanzahl n. Die Beziehung $P_{2,m} = P_{n,2}$ gilt für $n = m$ offensichtlich aus Symmetriegründen.

Satz 4.3.1 [13] *Für alle $m \in \mathbb{N}$ gilt*

$$P_{2,m} = \sum_{r=m}^{2m} P(2,m,r) = m! \sum_{k=0}^{m} \frac{m!}{k!} \binom{m}{k}. \tag{4.8}$$

Beweis: In [1] haben AKERS und FRIEDMAN gezeigt, daß die Anzahl der zulässigen Organisationen zu einer gegebenen Technologie des Problems $Jm|n = 2|C_{\max}$ gleich $m + 1 + \sum_{k=2}^{m} |\pi_k|$ ist, wobei π_k die Menge der geordneten k-Tupel $(j_1, \ldots, j_k)$ von Maschinen $M_{j_1}, \ldots, M_{j_k}$ ist, die sich in derselben technologischen Reihenfolge beider Aufträge befinden, also

$$o_{1,j_1} \prec o_{1,j_2} \prec \cdots \prec o_{1,j_k} \quad \text{und} \quad o_{2,j_1} \prec o_{2,j_2} \prec \cdots \prec o_{2,j_k}.$$

Durch Summation über alle möglichen Technologien entsteht die verallgemeinerte Aussage (4.8) für das Open-Shop-Problem: Die Operationen-Reihenfolge $o_{1,1} \prec o_{1,2} \prec \cdots \prec o_{1,m}$ repräsentiere die technologische Reihenfolge von J_1. Unter den $m!$ technologischen Reihenfolgen von J_2 tritt ein festes, in natürlicher Reihenfolge geordnetes k-Tupel von Maschinen genau $m!/k!$–mal auf. Für die Wahl eines solchen k-Tupels gibt es $\binom{m}{k}$ Möglichkeiten. Der konstante Summand $m + 1$ aus der Formel von AKERS und FRIEDMAN kommt für jede der $m!$ technologischen Reihenfolgen von J_2 hinzu. Durch Multiplikation mit $m!$ als Anzahl der technologischen Reihenfolgen von J_1 ergibt sich

$$P_{2,m} = m!\left[m!(m+1) + \sum_{k=2}^{m} \frac{m!}{k!}\binom{m}{k}\right] = m!\ \sum_{k=0}^{m} \frac{m!}{k!}\binom{m}{k}.$$

□

In Tabelle 4.3 ist die Anzahl zweizeiliger Pläne gemäß (4.8) im Vergleich mit der Anzahl lateinischer Rechtecke $\mathcal{L}_{2,m,r}$ mit $m \le r \le 2m$ für $m = 2, \ldots, 10$ (jeweils reduziert um den Faktor $m!$) dargestellt.

Azyklische Orientierungen und chromatische Polynome

Ein $n \times m$–Plan entspricht einem $n \times m$–Ablaufgraphen und somit einer azyklischen Orientierung des Hamming-Graphen $K_n \times K_m$. Im folgenden wird gezeigt, daß nicht nur die Bestimmung der Anzahl allgemeiner lateinischer Rechtecke (Abschnitt 4.2), sondern auch die Bestimmung der Anzahl aller $n \times m$–Pläne eng mit der Bestimmung des chromatischen Polynoms des Hamming-Graphen $K_n \times K_m$ verbunden ist.

Es sei $\alpha(G)$ die Anzahl der azyklischen Orientierungen eines Graphen G. In [92] hat STANLEY erstmals $\alpha(G)$ mit dem chromatischen Polynom $\chi(G, \lambda)$ von G in Zusammenhang gebracht. Der entsprechende Satz in [92] macht sogar eine allgemeinere Aussage. Hier wird jedoch ausschließlich die folgende, für die Bestimmung der Anzahlen $P_{n,m}$ interessante Spezialisierung der Aussage in [92] bewiesen.

Satz 4.3.2 [92] *Für jeden Graphen $G = (V, E)$ gilt $\alpha(G) = (-1)^{|V|}\chi(G, -1)$.*

m	$P_{2,m} = \sum_{r=m}^{2m} \frac{P(2,m,r)}{m!}$	$\sum_{r=m}^{2m} \frac{L(2,m,r)}{m!}$
2	7	52
3	34	1 786
4	209	89 334
5	1 546	5 860 548
6	13 327	476 670 186
7	130 922	46 306 142 594
8	1 441 729	5 232 708 447 382
9	17 572 114	674 452 363 859 548
10	234 662 231	97 662 704 169 789 056

Tabelle 4.3: Anzahlen der $2 \times m$–Pläne und der lateinischen Rechtecke $\mathcal{L}_{2,m,r}$.

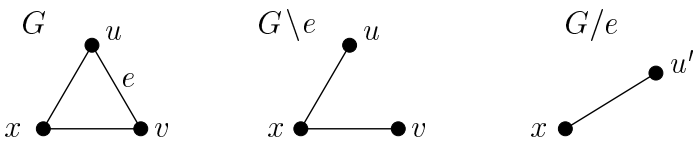


Abbildung 4.2: Zur Definition der Graphen $G\backslash e$ und G/e.

Beweis: Es ist bekannt, daß das chromatische Polynom $\chi(G,\lambda)$ eines Graphen $G = (V,E)$ eindeutig durch die drei folgenden Bedingungen bestimmt ist (siehe z. B. [77], Theorem 2.2, 2.5 und 2.6):

(i) $\chi(G_0,\lambda) = \lambda$, wobei G_0 der Graph ist, der aus einem Knoten besteht,
(ii) $\chi(G \cup H,\lambda) = \chi(G,\lambda)\chi(H,\lambda)$, wobei $G \cup H$ die Vereinigung zweier disjunkter Graphen G und H ist,
(iii) $\chi(G,\lambda) = \chi(G\backslash e,\lambda) - \chi(G/e,\lambda)$ für alle $e \in E$, wobei $G\backslash e$ bzw. G/e der Graph ist, der aus einem Graphen $G = (V,E)$ durch Löschen bzw. Kontraktion der Kante e entsteht (vgl. Abbildung 4.2).

Es reicht also zu zeigen, daß für $\alpha(G) = (-1)^{|V|}\chi(G,-1)$ die entsprechenden Bedingungen

(i') $\alpha(G_0) = 1$,
(ii') $\alpha(G \cup H) = \alpha(G)\alpha(H)$,
(iii') $\alpha(G) = \alpha(G\backslash e) + \alpha(G/e)$

gelten. Die Anzahl azyklischer Orientierungen des trivialen Graphen G_0 ist eins, und die Anzahl der azyklischen Orientierungen eines aus zwei Komponenten bestehenden Graphen ist das Produkt der entsprechenden Zahlen für die Komponenten, da die azyklischen Orientierungen unabhängig voneinander sind. Also gelten offensichtlich die Bedingungen (i') und (ii'). Im folgenden wird nun auch die Gültigkeit von (iii') gezeigt.

Es sei $\mathcal{O}$ eine azyklische Orientierung von $G\backslash e$, wobei $e = \{u, v\}$ die gelöschte Kante ist. Weiterhin sei $\mathcal{O}_1$ die Orientierung von G, die durch Hinzufügen von $u \to v$ zu $\mathcal{O}$ entsteht, und $\mathcal{O}_2$ die entsprechende Orientierung durch Hinzufügen von $v \to u$. Es ist schnell einzusehen (vgl. Abbildung 4.3, linke Hälfte), daß für jede azyklische Orientierung $\mathcal{O}$ von $G\backslash e$ entweder $\mathcal{O}_1$ oder $\mathcal{O}_2$ azyklisch ist, außer in $\alpha(G/e)$ Fällen, in denen sowohl $\mathcal{O}_1$ als auch $\mathcal{O}_2$ azyklisch sind (vgl. Abbildung 4.3, rechte Hälfte). Also gilt $\alpha(G) = \alpha(G\backslash e) + \alpha(G/e)$. □

Einen alternativen Beweis der Aussage dieses Satzes hat Vo in [102] gegeben. Der Beweis beruht auf sogenannten geordneten kantenfreien Partitionen der Knoten eines Graphen, und wird im folgenden skizziert.

Es sei $G = (V, E)$ ein Graph. Eine *Partition* von V ist eine Menge disjunkter Teilmengen von V, deren Vereinigung V ergibt. Eine *kantenfreie Partition* von V ist eine Partition in unabhängige[7] Knotenmengen. Eine *geordnete kantenfreie Partition* von G ist eine kantenfreie Partition, bei der eine Reihenfolge der unabhängigen Knotenmengen festgelegt ist. Ist π_k die Anzahl der kantenfreien Partitionen von G in k unabhängige Knotenmengen ist, so gilt offensichtlich

$$\chi(G, \lambda) = \sum_{k=1}^{|V|} \pi_k \ \lambda(\lambda - 1) \cdots (\lambda - k + 1), \tag{4.9}$$

und mit $\lambda = -1$ erhält man

$$(-1)^{|V|}\chi(G, -1) = \sum_{k=1}^{|V|} (-1)^{|V|-k} \pi_k k! \,. \tag{4.10}$$

Es sei Π_G die Menge aller geordneten kantenfreien Partitionen von V. Es gilt $|\Pi_G| = \sum_{k=1}^{|V|} \pi_k k!$. Wenn man in dieser Summe jedem Element P von Π_G das Vorzeichen $(-1)^{|V|-k}$ zuordnet, wird deutlich, daß beim Summieren nur die Fixpunkte einer vorzeichenumkehrenden Involution[8] $i : \Pi_G \to \Pi_G$ übrig bleiben, da

[7] In einem Graphen $G = (V, E)$ heißt eine Knotenmenge $V' \subseteq V$ *unabhängig*, wenn keine zwei Knoten aus V' benachbart sind.

[8] Eine *Involution* ist eine Abbildung i mit $i(i(a)) = a$ für alle Elemente a, auf denen i definiert ist.

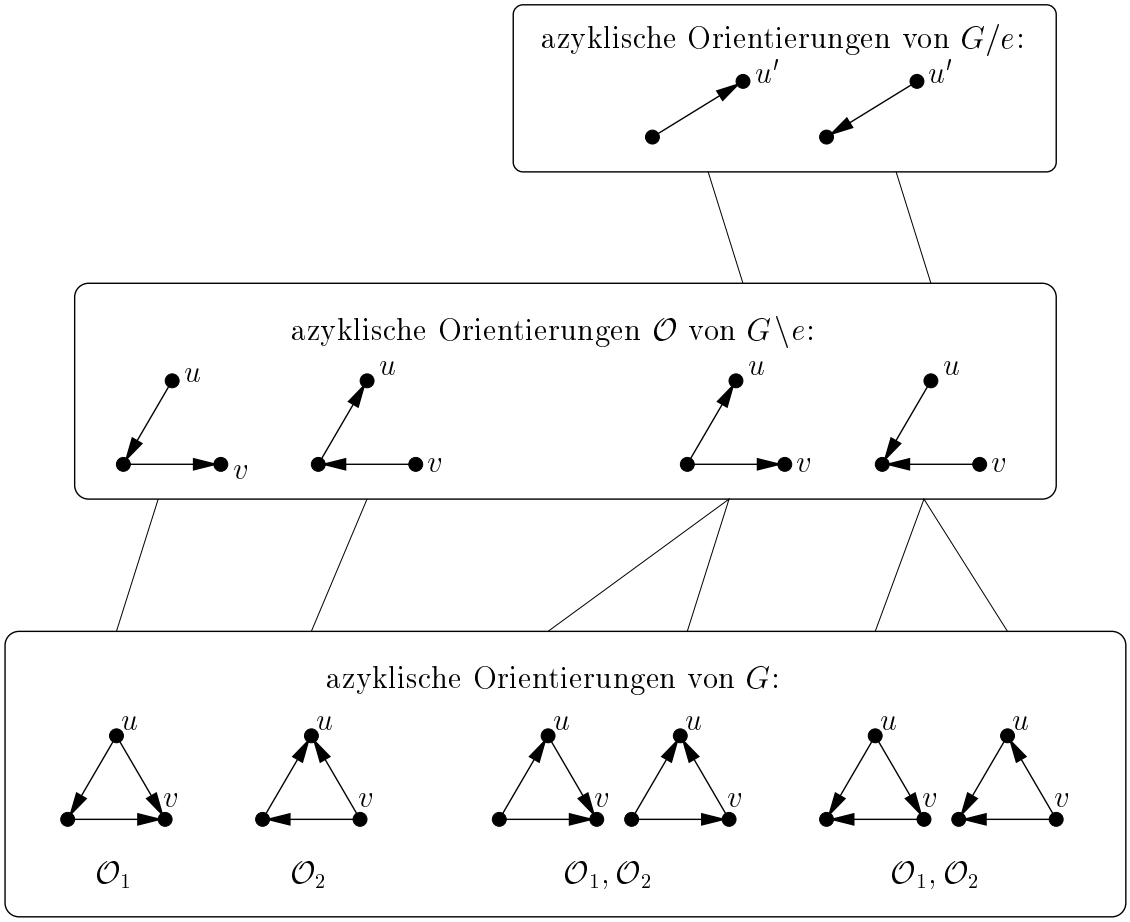


Abbildung 4.3: Zum Beweis von Satz 4.3.2.

sich alle anderen Elemente von Π_G durch Anwendung von i zu Null summieren. In [102] definiert Vo nun eine solche Involution i auf Π_G, deren Fixpunktmenge gerade die sogenannten diskreten Basis-Partitionen[9] von V sind, die wiederum den azyklischen Orientierungen von G eineindeutig zugeordnet werden können. Daher werden auf der rechten Seite von (4.10) die azyklischen Orientierungen von G gezählt, und der alternative Beweis von Satz 4.3.2 ist damit vollständig.

Da jeder $n \times m$–Plan mit einer azyklischen Orientierung des Hamming-Graphen $K_n \times K_m$ eineindeutig korrespondiert, kann man anhand von Satz 4.3.2 die Anzahl der Pläne mit Hilfe des chromatischen Polynoms der Hamming-Graphen darstellen.

Satz 4.3.3 *Für alle $n, m \in \mathbb{N}$ gilt*

$$P_{n,m} = \alpha(K_n \times K_m) = (-1)^{nm}\chi(K_n \times K_m, -1).$$

□

Wenn es eine effiziente Methode zur Berechnung des chromatischen Polynoms von Hamming-Graphen gibt, kann mit ihrer Hilfe die Anzahl aller Pläne eines gegebenen Formats $n \times m$ berechnet werden. Der Komplexitätsstatus dieses Enumerationsproblems ist bis heute unbekannt (siehe Abschnitt 5.5).

Pläne des Formats $3 \times m$ und $4 \times m$

Mit Hilfe des gerade beschriebenen Zusammenhangs und den Ergebnissen aus Abschnitt 4.2 über die Anzahl allgemeiner lateinischer Rechtecke kann neben der Bestimmung der Anzahl $P_{2,m}$ auch eine Formel für die Anzahl der dreizeiligen Pläne erstellt werden, denn das chromatische Polynom der Hamming-Graphen $K_3 \times K_m$ ist bekannt.

Satz 4.3.4 *Für festes $m \in \mathbb{N}$ gilt*

$$P_{3,m} = (-1)^m \tfrac{\lambda!m!}{((\lambda-m)!)^3} \sum_{\alpha+\beta+\gamma=m} (-1)^\beta \; 2^\gamma \; \tfrac{((\lambda-m+\alpha)!)^2}{\alpha!\gamma!} \; \binom{3\lambda-3m+3\alpha+\beta+2}{\beta}$$

mit $\lambda = -1$.

Beweis: Aufgrund von Satz 4.3.3 ist $P_{3,m}$ gleich dem Betrag des chromatischen Polynoms $\chi(K_3 \times K_m, \lambda)$ an der Stelle $\lambda = -1$. Hilfssatz 4.2.2 zeigt, daß das

[9] Die Elemente einer Knotenmenge V seien linear geordnet. Eine *diskrete Basis-Partition* ist eine geordnete kantenfreie Partition, bei der alle Teilmengen mit mehr als einem Knoten in 1-elementige Teilmengen aufgeteilt sind, wobei diese bezüglich der linearen Ordnung von V absteigend sortiert sind.

chromatische Polynom $\chi(K_3 \times K_m, \lambda)$ dem Ausdruck $L(3, m, \lambda)$ entspricht. Die Aussage des Satzes folgt dann wegen der Formel für $L(3, m, \lambda)$ aus Satz 4.2.3. □

Bemerkung 4.3.5 Wenn beim Ausdruck $P_{3,m}$ der Term $1/((\lambda-m)!)^2$ in die Summe hineingezogen wird, sieht man, daß es sich auf der rechten Seite um ein Polynom in λ vom Grad $3m$ handelt: Vor der Summe verbleibt mit $\lambda!/(\lambda - m)!$ der Anteil λ^m. In der Summe ergibt

$$\left(\frac{(\lambda - m + \alpha)!}{(\lambda - m)!}\right)^2$$

den Anteil $\lambda^{2\alpha}$ und der Binomialkoeffizient den Anteil λ^β. Wegen $m = \alpha + \beta + \gamma$ gilt $2\alpha + \beta \leq 2m$, also kommt durch die Summe der Anteil λ^{2m} zu λ^m noch hinzu. Beispielsweise lauten die beiden Polynome auf der rechten Seite für

$$\begin{array}{lll} m = 1: & -\lambda^3 + 3\lambda^2 - 2\lambda & \text{und für} \\ m = 2: & \lambda^6 - 9\lambda^5 + 34\lambda^4 - 67\lambda^3 + 67\lambda^2 - 26\lambda. & \end{array}$$

Mit $\lambda = -1$ ergibt sich $P_{3,1} = 6$ und $P_{3,2} = 204$.

Es liegt nun nahe, die gleiche Vorgehensweise auch für die Anzahl $P_{4,m}$ anzuwenden, da analog zu Satz 4.2.3 die Arbeit von Athreya, Pranesachar und Singhi [3] auch eine Formel für $L(4, m, \lambda)$ enthält. Allerdings scheint diese Formel nicht korrekt zu sein, denn schon im einfachsten Fall (für $m = 1$ und $\lambda = 4$) ergibt sich ein Wert, der nicht der Anzahl der zulässigen 4-Färbungen des Hamming-Graphen $K_4 \times K_1$ bzw. der Anzahl der Lateinischen Rechtecke $\mathcal{L}_{4,1,4}$ entspricht. Diese Anzahl $L(4, 1, 4)$ beträgt 4!. Mit der Formel aus [3] ergibt sich jedoch $L(4, 1, 4) = 137952$.

Der Beweis der Formel in [3] ist nur angedeutet. Eine Anfrage an C. R. Pranesachar (einer der Autoren von [3]) blieb bisher ohne klärenden Erfolg.

Analog zu Tabelle 4.3 auf Seite 37 für Pläne und lateinische Rechtecke vom Format $2 \times m$ enthält Tabelle 4.4 die Werte entsprechender Matrizen des Formats $3 \times m$. Tabelle 4.3 und Tabelle 4.4 veranschaulichen deutlich, daß in der betrachteten Menge der lateinischen Rechtecke nur relativ wenig Elemente die zusätzliche Bedingung eines Plans erfüllen. Die Anzahl der lateinischen Rechtecke ist also nur eine sehr unscharfe obere Schranke für die Anzahl der zugehörigen Pläne. Im folgenden Abschnitt werden nun schärfere obere und untere Schranken für die Anzahl $P_{n,m}$ hergeleitet.

m	$P_{3,m} = \sum_{r=m}^{3m} \frac{P(3,m,r)}{m!}$	$\sum_{r=m}^{3m} \frac{L(3,m,r)}{m!}$
3	3 194	9 432 636
4	155 544	32 338 932 048
5	10 736 592	185 278 786 748 496
6	989 958 592	1 602 418 389 749 579 136
7	116 976 844 224	19 524 505 523 383 344 567 936
8	17 177 847 282 048	318 946 995 329 678 929 562 127 360
9	3 061 325 835 300 608	6 730 548 553 292 744 342 990 592 919 680
10	649 679 086 266 011 904	178 253 947 720 328 843 939 901 662 766 677 760

Tabelle 4.4: Anzahlen der $3 \times m$–Pläne und der lateinischen Rechtecke $\mathcal{L}_{3,m,r}$.

4.4 Obere und untere Schranken

Die Formeln für die $3 \times m$– und $4 \times m$–Pläne sind bereits vergleichsweise umfangreich und kompliziert. Aufgrund der erwähnten Resultate im Zusammenhang mit der Komplexität der Erweiterung rangminimaler Pläne sind erst recht keine einfachen Ausdrücke für Formate $n \times m$ mit $n \geq 5$ zu erwarten. Es wird daher in diesem Abschnitt nach oberen und unteren Schranken für die Gesamtanzahl $P_{n,m}$ aller $n \times m$–Pläne gesucht.

In [55] hat M. Kleinau Abschätzungen für die Anzahl zulässiger Lösungen von Job-Shop-Problemen des Typs $Jm||C_{\max}$ mit n Aufträgen gegeben. Im folgenden werden obere und untere Schranken für die Anzahl $P_{n,m}$ aller $n \times m$–Pläne bzw. aller zulässigen Lösungen des Open-Shop-Problems $Om||C_{\max}$ mit n Aufträgen entwickelt.

Zum Auffinden oberer und unterer Schranken für die Anzahl der $n \times m$–Pläne genügt es nach Satz 4.3.3, das chromatische Polynom des Hamming-Graphen $K_n \times K_m$ nach oben und unten entlang der negativen reellen Achse abzuschätzen. In Arbeiten von Dohmen [31, 32] über Schranken für chromatische Polynome $\chi(G,k)$ werden ausschließlich Abschätzungen für positive ganzzahlige λ bzw. reelle Werte $\lambda \geq 1$ entwickelt. Die Interpretation von $\chi(G,\lambda)$ als Anzahl zulässiger λ-Färbungen des Graphen G ist nur für positive ganzzahlige λ sinnvoll. Für negative Werte von λ sind die Ergebnisse aus [31, 32] unbrauchbar und können daher hier im Zusammenhang mit azyklischen Orientierungen nicht verwandt werden. Es wird nun nach Abschätzungen des Polynoms $\chi(G,\lambda)$ gesucht, die auch für negative λ Gültigkeit besitzen.

Obere Schranke durch Gerüst-Anzahl

In [49] haben KAHALE und SCHULMAN eine obere Schranke für die Anzahl $\alpha(G)$ der azyklischen Orientierungen eines Graphen G auf der Grundlage der Gerüste[10] eines zu G verwandten Graphen G' hergeleitet. Diese Schranke stellt eine Verbesserung der vorher bekannten oberen Schranken dar, die ausschließlich auf den Knotengraden der Knoten eines Graphen basieren. Eine Verallgemeinerung dieser Ergebnisse ergibt zusätzlich Abschätzungen für das chromatische Polynom $\chi(G, \lambda)$ für negative reelle Argumente λ.

Für einen ungerichteten Graphen G sei G' der erweiterte Graph, der aus G durch Hinzufügen eines Knoten u entsteht, wobei u zu allen Knoten von G benachbart ist. Weiterhin sei $\tau(G)$ die Anzahl der Gerüste eines Graphen G. Die entwickelten oberen Schranken für $\alpha(G)$ beruhen auf einem in [49] gezeigten Zusammenhang zwischen den Anzahlen $\alpha(G)$ und $\tau(G')$.

Hilfssatz 4.4.1 [49] *Für einen beliebigen Graphen G gilt $\alpha(G) \leq \tau(G')$.*

Die *Admittanzmatrix* $Q(G)$ eines Graphen $G = (V, E)$ mit $V = \{1, \ldots, p\}$ ist die $p \times p$–Matrix $Q(G) = (q_{ij})$ mit

$$q_{ij} = \begin{cases} -1, & \text{falls die Knoten } i \text{ und } j \text{ benachbart sind,} \\ 0, & \text{falls } i \neq j \text{ und } i \text{ ist nicht zu } j \text{ benachbart,} \\ d(i), & \text{falls } i = j, \end{cases}$$

wobei $d(i)$ den Grad des Knoten i angibt.

Es sei $Q(G)_i$ die Matrix, die sich durch Streichung der i-ten Zeile und i-ten Spalte der Admittanzmatrix $Q(G)$ ergibt. Der folgende, ursprünglich auf KIRCHHOFF [54] zurückgehende Matrix-Gerüst-Satz zeigt, daß die Anzahl der Gerüste eines beliebigen Graphen mit Hilfe der Admittanzmatrix bestimmbar ist.

Satz 4.4.2 [44] *Es sei $G = (V, E)$ ein Graph mit $V = \{1, \ldots, p\}$. Dann gilt $\tau(G) = \det(Q(G)_i)$ für beliebiges i mit $1 \leq i \leq p$.*

Die *Kirchhoff-Matrix* von G ist $K(G) = Q(G) + E$, dabei bezeichnet E die Einheitsmatrix. Anhand von Hilfssatz 4.4.1 und durch Anwenden des Matrix-Gerüst-Satzes auf G' (Streichung der zum Knoten u gehörenden Zeile und Spalte) folgt unmittelbar die anschließende Aussage.

Satz 4.4.3 [49] *Es sei G ein beliebiger Graph und $K(G)$ seine Kirchhoff-Matrix. Dann gilt $\alpha(G) \leq \det(K(G))$.*

[10] Ein zyklenfreier Graph mit p Knoten und $p - 1$ Kanten heißt *Baum*. Es sei $G = (V, E)$ ein Graph. Ein Baum $T = (V_T, E_T)$ mit $V_T = V$ und $E_T \subseteq E$ heißt *aufspannender Baum* bzw. *Gerüst* von G.

Dieser Satz liefert zusammen mit einer geeigneten Abschätzung für die Kirchhoff-Matrix von Hamming-Graphen $K_n \times K_m$ eine obere Schranke für $P_{n,m}$.

Satz 4.4.4 *Für alle $n, m \in \mathbb{N}$ gilt*

$$P_{n,m} \leq \left[\frac{(n+m-1)}{e^{\left(\frac{1}{2d} - \frac{1}{2d^2} + \frac{1}{12d^3} - \frac{1}{4d^4} + O(\frac{1}{d^5})\right)}} \right]^{nm}$$

mit $d = n + m - 2$.

Beweis: Wegen Satz 4.3.3 und Satz 4.4.3 ist $P_{n,m} \leq \det(K(K_n \times K_m))$. Weiterhin gilt für jeden d-regulären Graphen $G = (V, E)$ mit $|V| = p$ die Beziehung $\det(K(G)) \leq (d+1)^p \exp(-p(\frac{1}{2d} - \frac{1}{2d^2} + \frac{1}{12d^3} - \frac{1}{4d^4} + O(\frac{1}{d^5})))$, siehe [49]. Die Aussage des Satzes folgt, da der Hamming-Graph $K_n \times K_m$ ein $(n+m-2)$–regulärer Graph mit nm Knoten ist. □

Zwei untere Schranken

In [37] wird von GODDARD *et al.* erstmals eine untere Schranke für die Anzahl azyklischer Orientierungen eines Graphen G in Abhängigkeit der Gradfolge von G bewiesen.

Satz 4.4.5 [37] *Für jeden Graphen $G = (V, E)$ mit $V = \{1, \ldots, p\}$ gilt*

$$\alpha(G) \geq \prod_{i=1}^{p} \left((d_i + 1)!\right)^{\frac{1}{d_i+1}},$$

wobei d_i den Grad des Knoten i bezeichnet.

Im Fall $G = K_n \times K_m$ kann man für die Anzahl der $n \times m$–Pläne eine untere Schranke herleiten.

Folgerung 4.4.6 *Für alle $n, m \in \mathbb{N}$ gilt*

$$P_{n,m} \geq \left((n+m-1)!\right)^{\frac{nm}{n+m-1}}. \tag{4.11}$$

Diese untere Schranke zeigt bereits das enorme Ansteigen der Anzahlen $P_{n,m}$ mit wachsenden Werten n und m. Zur Veranschaulichung ist die Funktion $f(n, m) = \left((n+m-1)!\right)^{nm/(n+m-1)}$ für das Intervall $[0, 1000]$ des Wertebereichs in Abbildung 4.4 dargestellt. In [14] haben BRÄSEL und M. KLEINAU eine andere untere Schranke für $P_{n,m}$ entwickelt, die auf der Analyse eines ersten Enumerationsalgorithmus für Pläne beruht.

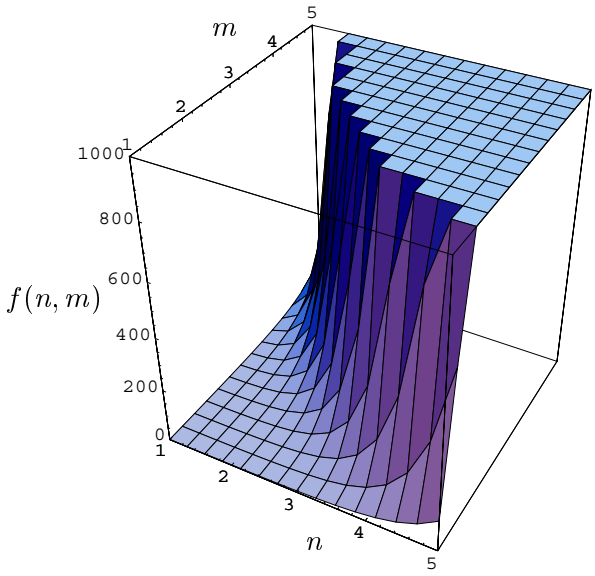


Abbildung 4.4: Untere Schranke (4.11) für die Anzahl aller $n \times m$–Pläne.

Satz 4.4.7 [14] *Für alle $n, m \in \mathbb{N}$ gilt*

$$P_{n,m} \geq \prod_{i=0}^{n-1} \frac{(m+i)!}{i!}. \tag{4.12}$$

Der folgende Satz zeigt, daß die untere Schranke (4.11) schlechter als (4.12) ist.

Satz 4.4.8 *Für alle $n, m \in \mathbb{N}$ gilt*

$$\prod_{i=0}^{n-1} \frac{(m+i)!}{i!} \geq \big((n+m-1)!\big)^{\frac{nm}{n+m-1}}. \tag{4.13}$$

Beweis: Die rechte Seite von (4.13) ist offensichtlich eine symmetrische Funktion von n und m. Die linke Seite ist ebenfalls symmetrisch, denn mit $M = \min(n, m)$ gilt

$$\prod_{i=0}^{n-1} \frac{(m+i)!}{i!} = \prod_{i=0}^{m-1} \frac{(n+i)!}{i!} = \prod_{i=1}^{M} \frac{(m+n-i)!}{(M-i)!}. \tag{4.14}$$

Sei also ohne Beschränkung der Allgemeinheit $n \leq m$, d. h. zu zeigen ist

$$\prod_{i=1}^{n} \frac{(m+n-i)!}{(n-i)!} \geq \big((n+m-1)!\big)^{\frac{nm}{n+m-1}}.$$

Durch Logarithmieren erhält man

$$(m+n-1)\sum_{i=1}^{n}\log(m+n-i)! \geq$$
$$\geq mn\log(m+n-1)! + (m+n-1)\sum_{i=1}^{n}\log(n-i)!\,.$$

Die Anwendung der Logarithmen-Gesetze sowie weitere elementare Umformungen führen schließlich auf

$$n(n-1)\sum_{i=1}^{m}\log(n-1+i) + (m+n-1)\sum_{i=1}^{n-1} i\log i \geq$$
$$\geq mn\sum_{i=1}^{n-1}\log i + (m+n-1)\sum_{i=1}^{n-1} i\log(m+i). \tag{4.15}$$

Es wird nun gezeigt, daß für $n \leq m$ die linke Seite der Ungleichung (4.15) mit m schneller wächst als die rechte Seite von (4.15), d. h. zu zeigen ist die Gültigkeit der Ungleichung

$$\begin{aligned} &n(n-1)\sum_{i=1}^{m+1}\log(n-1+i) + (m+n)\sum_{i=1}^{n-1} i\log i \\ &-n(n-1)\sum_{i=1}^{m}\log(n-1+i) - (m+n-1)\sum_{i=1}^{n-1} i\log i \geq \\ &\geq (m+1)n\sum_{i=1}^{n-1}\log i + (m+n)\sum_{i=1}^{n-1} i\log(m+1+i) \\ &- mn\sum_{i=1}^{n-1}\log i - (m+n-1)\sum_{i=1}^{n-1} i\log(m+i). \end{aligned} \tag{4.16}$$

Vereinfachungen von (4.16) führen auf

$$\sum_{i=1}^{n-1}(m+n-i)\log(m+i) \geq m(n-1)\log(m+n) + \sum_{i=1}^{n-1}(n-i)\log i$$

bzw.

$$\sum_{i=1}^{n-1}\left\{(n-i)\log\left(1+\frac{m}{i}\right) - m\log\left(1+\frac{n-i}{m+i}\right)\right\} \geq 0.$$

Diese Ungleichung ist gültig, wenn für alle $i = 1, \ldots, n-1$ die Ungleichung

$$(n-i)\log\left(1+\frac{m}{i}\right) \geq m\log\left(1+\frac{n-i}{m+i}\right)$$

gilt. Wegen $x > \log(1+x)$ mit $x > 0$ reicht es zu zeigen, daß für alle $i = 1, \ldots, n-1$ die Ungleichung

$$(n-i)\log\left(1+\frac{m}{i}\right) \geq \frac{m(n-i)}{m+i}$$

bzw.

$$\left(1+\frac{i}{m}\right)\log\left(1+\frac{m}{i}\right) \geq 1 \tag{4.17}$$

gilt. Dies ist der Fall, da für festes i, $1 \leq i \leq n-1 \leq m-1$ die erste Ableitung der linken Seite von (4.17) nach m positiv ist und (4.17) für das kleinste m, also $m = i+1$, erfüllt ist, wie sich leicht nachweisen läßt.

Damit ist die Gültigkeit von (4.16) gezeigt. Wegen der bereits erwähnten Symmetrie der beiden Seiten von (4.13) muß nun Beziehung (4.13) bzw. (4.15) nur noch für $n = m$ nachgewiesen werden. Aus (4.15) mit $n = m$ ergibt sich

$$\begin{aligned} n(n-1)\sum_{i=1}^{n}\log(n-1+i) + (2n-1)\sum_{i=1}^{n-1} i\log i \geq \\ \geq n^2\sum_{i=1}^{n-1}\log i + (2n-1)\sum_{i=1}^{n-1} i\log(n+i). \end{aligned} \tag{4.18}$$

Die linke Seite von (4.18) wächst mit n schneller als die rechte Seite, wenn

$$\begin{aligned} &(n+1)n\sum_{i=1}^{n+1}\log(n+i) + (2n+1)\sum_{i=1}^{n} i\log i \\ &-n(n-1)\sum_{i=1}^{n}\log(n-1+i) - (2n-1)\sum_{i=1}^{n-1} i\log i \geq \\ &\qquad \geq (n+1)^2\sum_{i=1}^{n}\log i + (2n+1)\sum_{i=1}^{n} i\log(n-1+i) \\ &\qquad\quad - n^2\sum_{i=1}^{n-1}\log i - (2n-1)\sum_{i=1}^{n-1} i\log(n+i) \end{aligned} \tag{4.19}$$

gilt. Durch elementare Umformungen erhält man

$$\begin{aligned} \sum_{i=1}^{n}(2n-2i+1)\log\left(1+\frac{n}{i}\right) + \sum_{i=1}^{n} 2n\log(n+i) \geq \\ \geq n^2\log(2n+1) + n^2\log(2n). \end{aligned} \tag{4.20}$$

Die beiden Summen auf der linken Seite lassen sich durch

$$\begin{aligned}\sum_{i=1}^{n}(2n-2i+1)\log\left(1+\frac{n}{i}\right) &= \\ &= (2n-1)\log(n+1)+\sum_{i=2}^{n}(2n-2i+1)\log\left(1+\frac{n}{i}\right) \\ &\geq (2n-1)\log(n+1)+[1+3+5+\cdots+(2(n-1)-1)]\log 2 \\ &= (2n-1)\log(n+1)+(n-1)^2\log 2 \\ &= (2n-1)\log\frac{n+1}{2}+n^2\log 2\end{aligned} \tag{4.21}$$

und

$$\begin{aligned}2n\sum_{i=1}^{n}\log(n+i) &\geq 2n\left(n\log n+\frac{n}{2}(\log(2n)-\log n)\right) \\ &= 2n^2\log n+n^2\log 2\end{aligned} \tag{4.22}$$

abschätzen, wobei sich die rechte Seite von (4.22) durch Berechnung der Trapezfläche unterhalb der Funktion $\log x$ im Intervall $[n,2n]$ ergibt. Für die rechte Seite von (4.20) erhält man

$$\begin{aligned}n^2\log(2n+1)+n^2\log(2n) &\leq n^2\log(2(n+1))+n^2\log(2n) \\ &= n^2\log(n+1)+n^2\log n+2n^2\log 2.\end{aligned} \tag{4.23}$$

Schließlich führt die Anwendung von (4.21)–(4.23) auf (4.20) zu

$$(2n-1)\log\frac{n+1}{2}+2n^2\log 2+2n^2\log n\geq n^2\log(n+1)+n^2\log n+2n^2\log 2$$

bzw.

$$(2n-1)\log\frac{n+1}{2}\geq n^2\log\left(1+\frac{1}{n}\right).$$

Wegen $x>\log(1+x)$ mit $x>0$ reicht es zu zeigen, daß $(2n-1)\log\frac{n+1}{2}\geq n$ bzw. $(2-\frac{1}{n})\log\frac{n+1}{2}\geq 1$ gilt, was für $n\geq 3$ offensichtlich der Fall ist. Da Ungleichung (4.19) auch für $n=1,2$ gilt, ist insgesamt die Monotonie von (4.18) für alle $n\in\mathbb{N}$ bewiesen. Schließlich prüft man schnell die Gültigkeit von (4.18) für $n=1,2$ und 3 nach. Damit gilt (4.18) für alle $n\in\mathbb{N}$ und der Satz ist bewiesen. □

Kapitel 5

Plan-Enumeration

In diesem Kapitel geht es um die Enumeration der zulässigen Lösungen eines Open-Shop-Problems mit n Aufträgen und m Maschinen, d. h. es wird für gegebene Werte n und m die Menge aller $n \times m$–Pläne erzeugt.

In [14] haben Bräsel und M. Kleinau 1992 erstmals eine Enumerationsmethode entwickelt, mit der die Werte $P(n, m, r)$ in den Fällen

- $n = 2, \quad 2 \leq m \leq 8 \quad$ und
- $n = 3, \quad 3 \leq m \leq 4$

für $m \leq r \leq nm$ mit Hilfe eines Computers bestimmt werden können. Um die Pläne auch für größere Formate $n \times m$ aufzählen zu können, werden in diesem Kapitel die Verfahren aus [14] in vielfacher Hinsicht modifiziert.

In Abschnitt 5.1 werden verschiedene Äquivalenzrelationen behandelt, die die Menge aller $n \times m$–Pläne jeweils in disjunkte Äquivalenzklassen partitionieren. Weiterhin wird für die Menge der Äquivalenzklassen ein geeignetes Vertretersystem beschrieben. Die daraus gewonnenen Erkenntnisse bilden die Basis für einen neuen effizienten Algorithmus zur Enumeration aller Pläne eines Open-Shop-Problems mit n Aufträgen und m Maschinen.

Die Abschnitte 5.2 und 5.3 dokumentieren diesen Enumerationsalgorithmus, wobei die Enumeration der Technologien eines Open-Shop-Problems vorangestellt ist. Die Beschreibung des gesamten Algorithmus mit den zugehörigen Teilprozeduren ist in Abschnitt 5.3 enthalten. Zum Abschluß wird in Abschnitt 5.4 eine Zusammenfassung und Auswertung der neu erzielten Werte für die Anzahl der $n \times m$–Pläne gegeben.

5.1 Pläne gleicher Struktur

Die in den folgenden Unterabschnitten vorgestellten Äquivalenzrelationen (Isomorphie, Äquivalenz und Struktur-Äquivalenz) sind für eine effiziente Aufzählung und Charakterisierung der Pläne mit gleichen Eigenschaften bzw. Strukturen grundlegend. Ein Plan A eines Shop-Scheduling-Problems mit n Aufträgen und m Maschinen wird mit der $n \times m$–Matrix $A = (a_{ij})$ identifiziert, die aus den Rängen $a_{ij} = \varrho(o_{ij})$ der Knoten bzw. Operationen o_{ij} des zugehörigen $n \times m$–Ablaufgraphen besteht.

Isomorphie von Plänen

Die $n \times m$–Pläne bilden eine spezielle Klasse innerhalb der Menge der lateinischen Rechtecke $\mathcal{L}_{n,m,r}$. Die $n \times m$–Pläne mit $n = m$ heißen *quadratische Pläne*. Die rangminimalen quadratischen Pläne (mit $n = m = r$) sind meist unter dem Namen *lateinische Quadrate* bekannt.

Lateinische Quadrate können als Multiplikationstafeln von Quasigruppen[1] aufgefaßt werden. In [29] bezeichnen DÉNES und KEEDWELL zwei lateinische Quadrate als *isomorph*, wenn sie durch *dieselbe* Permutation von Zeilen, Spalten und Elementen der Belegungsmenge ineinander überführt werden können, d. h. wenn die entsprechenden Quasigruppen isomorph sind (bezüglich des Isomorphie von Quasigruppen)[2].

Die Übertragung des für lateinische Quadrate verwandten Begriffs „isomorph“ auf $n \times n$–Pläne ist nicht sinnvoll, da die Permutation von Elementen der Belegungsmenge für allgemeine quadratische Pläne keine abgeschlossene Operation ist, d. h. wenn Elemente eines Planes vertauscht werden, führt dies zu Matrizen, die nicht notwendig der zusätzlichen Eigenschaft für Pläne (vgl. Definition 3.2.1) genügen müssen.

In [17] bezeichnet BROWN zwei lateinische Rechtecke als *isomorph*, wenn sie durch Permutationen von Zeilen, Spalten und Elementen ineinander überführt werden können (vgl. auch die *Isotopie* von lateinischen Quadraten und zugehörigen Quasigruppen in DÉNES und KEEDWELL [29]). Das heißt, im Fall isomorpher lateinischer Rechtecke müssen im Gegensatz zur obigen Definition der isomorphen lateinischen Quadrate die verwendeten Permutationen nicht identisch sein. Aus dem gleichen Grund wie bei der „Isomorphie von lateinischen Quadraten“ ist die Übertragung des Begriffs „Isomorphie von lateinischen Rechtecken“ auf $n \times m$–Pläne

[1] Eine Menge Q heißt *Quasigruppe*, wenn auf ihr eine Verknüpfung $(\cdot)$ definiert ist, und für jedes Paar $a, b \in Q$ die Gleichungen $a \cdot x = b$ und $y \cdot a = b$ eindeutig nach x bzw. y auflösbar sind.

[2] Zwei Quasigruppen G und G' heißen *isomorph*, wenn es eine bijektive Abbildung $\varphi : G \to G'$ gibt mit $\varphi(ab) = \varphi(a)\varphi(b)$ für alle $a, b \in G$.

ebenfalls unangebracht. Es wird daher die Isomorphie von Plänen folgendermaßen definiert:

Definition 5.1.1 Zwei Pläne A und B heißen *isomorph*, $A \cong B$, wenn A durch eine Permutation ϱ von Zeilen und eine Permutation σ von Spalten in B überführt werden kann.

Ein Isomorphismus (ϱ, σ), der einen Plan A in sich selbst überführt, heißt *Automorphismus* von A. Die Isomorphie von Plänen ist eine Äquivalenzrelation, die die Menge der Pläne in disjunkte *Isomorphieklassen* aufteilt. Der Begriff „Isomorphie" wurde so gewählt, daß die Pläne einer Isomorphieklasse bei passender Indizierung der Aufträge J_i und Maschinen M_j identisch sind.

Äquivalenz von Plänen

Im quadratischen Fall ist bei der Zusammenfassung von Plänen mit gleichen Eigenschaften jeweils nicht nur eine andere Indizierung der Aufträge J_i und Maschinen M_j denkbar, sondern die Rolle der Aufträge kann auch komplett mit der der Maschinen vertauscht werden.

Zu einer gegebenen $n \times n$–Matrix A wird die Matrix, die sich durch *Transposition*, also durch Spiegelung von A an ihrer Hauptdiagonalen, ergibt, mit A^T bezeichnet. Offensichtlich ist A^T genau dann ein $n \times n$–Plan, wenn A einer ist. Analog zur Terminologie bei allgemeinen Matrizen heißt A^T der *transponierte Plan* von A.

Definition 5.1.2 Zwei Pläne A und B heißen *äquivalent*, $A \equiv B$, wenn $A \cong B$ oder $A \cong B^T$ gilt.

Wird für Pläne stets ein festes Format $n \times m$ mit $n \neq m$ betrachtet, so gilt für zwei Pläne A und B genau dann $A \cong B$, wenn $A \equiv B$ ist, d. h. die Begriffe „Isomorphie" und „Äquivalenz" fallen in diesem Fall zusammen. Wie leicht zu sehen ist, muß für zwei quadratische Pläne A und B mit $A \equiv B$ allerdings nicht notwendig $A \cong B$ gelten, d. h. für $n = m$ ist die Anzahl nicht-isomorpher Pläne mindestens so groß wie die Anzahl nicht-äquivalenter Pläne.

Im nächsten Unterabschnitt wird deutlich, daß die Äquivalenz von $n \times m$–Plänen mit der Isomorphie der zugehörigen $n \times m$–Ablaufgraphen gleichbedeutend ist. Zunächst wird jedoch ein Algorithmus entwickelt, der in polynomialer Zeit entscheidet, ob zwei gegebene Pläne A und B äquivalent sind oder nicht.

Definition 5.1.3 Ein Plan $A = (a_{ij})$ ist in *Normalform*, wenn $a_{11} = 1$ gilt, und die Einträge der ersten Zeile und der ersten Spalte jeweils aufsteigend sind.

Algorithmus 5.1.4 *Plan-Äquivalenz*

Eingabe: Zwei beliebige $n \times m$–Pläne $A = (a_{ij})$ und $B = (b_{ij})$.
Ausgabe: Eine Äquivalenz (in Form von Zeilen- und Spaltenpermutationen sowie Transposition), falls eine existiert.

1. Bringe A durch geeignete Zeilen- und Spaltenpermutation ϱ_A und σ_A in eine Normalform $A' = (a'_{ij})$ mit $a'_{11} = 1$.

2. Für alle Einträge $b_{kl} = 1$ in B:

   (a) Bringe B durch geeignete Zeilen- und Spaltenpermutation ϱ_B und σ_B in die Normalform $B' = (b'_{ij})$ mit $b'_{11} = b_{kl}$.

   (b) Wenn $A' = B'$ ist:

   A und B sind äquivalent. Die Äquivalenz wird durch die Zeilenpermutation $\varrho_A \varrho_B^{-1}$ und Spaltenpermutation $\sigma_A \sigma_B^{-1}$ beschrieben.

   (c) Wenn $n = m$ ist:

   i. Setze $B'' := B'^T$.

   ii. Wenn $A' = B''$ ist:

   A und B sind äquivalent. Die Äquivalenz wird durch die Zeilenpermutation $\varrho_A \varrho_B^{-1}$ und Spaltenpermutation $\sigma_A \sigma_B^{-1}$ sowie durch Transposition beschrieben.

Die Korrektheit des Algorithmus 5.1.4 folgt aus den vorangegangenen Betrachtungen, insbesondere aus Definition 5.1.2 und Definition 5.1.3.

Satz 5.1.5 *Für $n \leq m$ kann die Äquivalenz von $n \times m$–Plänen in der Zeit $O(nm^2)$ entschieden werden.*

Beweis: Es wird Algorithmus 5.1.4 betrachtet: Die Sortierung der Zeilen und Spalten, die zur Konstruktion der Normalformen in Schritt 1 und 2a erforderlich ist, benötigt $O(n \log n + m \log m)$ Zeit. Mit $n \leq m$ ergibt sich $O(m \log m)$ als obere Schranke. Die Vergleiche der Elemente zweier Matrizen in Schritt 2b und 2c können für $n \leq m$ jeweils in der Zeit $O(m^2)$ ausgeführt werden. Da B ein lateinisches Rechteck ist, gibt es höchstens n verschiedene Normalformen von B. Also muß der gesamte Schritt 2 maximal n-mal wiederholt werden, d. h. die Zeitkomplexität für den gesamten Algorithmus beträgt $O(nm^2)$. □

Isomorphie-Problem für Ablaufgraphen

Zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ heißen *isomorph*, wenn es eine bijektive Abbildung $\varphi : V_1 \to V_2$ gibt, so daß für alle Paare von Knoten $v, w \in V_1$ genau dann $\{v, w\} \in E_1$ gilt, wenn $\{\varphi(v), \varphi(w)\} \in E_2$ ist. Das heißt, zwei Graphen sind isomorph, wenn man die Knoten des einen Graphen auf die des anderen so abbilden kann, daß die Nachbarschaften zwischen den Knoten erhalten bleiben. Eine Bijektion φ mit dieser Eigenschaft heißt *Isomorphismus*. Diese Definition kann für Digraphen angepaßt werden, indem man für die Kanten die ungeordneten Paare von Knoten jeweils durch geordnete ersetzt.

Nach Satz 3.2.2 besteht eine eineindeutige Beziehung zwischen $n \times m$–Plänen und zugehörigen $n \times m$–Ablaufgraphen. Der folgende Hilfssatz gibt eine Charakterisierung äquivalenter Pläne anhand von Isomorphismen zwischen den zugehörigen azyklischen Digraphen. Diese Charakterisierung kann ebenfalls als Definition der Äquivalenz von Plänen benutzt werden.

Hilfssatz 5.1.6 [12] *Zwei Pläne A und B sind genau dann äquivalent, wenn die zugehörigen Ablaufgraphen $G(A)$ und $G(B)$ isomorph sind.*

Beweis: Offensichtlich besteht ein $n \times m$–Ablaufgraph $G(A)$ aus $n + m$ azyklischen Turnieren[3]. Dabei handelt es sich um n Turniere mit m Knoten, die mit den Zeilen des zugehörigen Plans A korrespondieren und m Turniere mit n Knoten entsprechend für die Spalten von A. Zwei Turniere von $G(A)$ sind genau dann knotendisjunkt, wenn sie entweder zu zwei verschiedenen Zeilen oder zu zwei verschiedenen Spalten von A gehören. Deshalb können die Knoten von Zeilen und die Knoten von Spalten permutiert werden, ohne die Grundstruktur der Nachbarschaften zwischen den Knoten des entsprechenden Ablaufgraphen G zu verändern. Für jeden Plan A gibt es offensichtlich auch eine Bijektion zwischen den Knotenmengen der Ablaufgraphen $G(A)$ und $G(A^T)$, die die Nachbarschaften invariant läßt. Die Permutationen und die eventuelle Matrix-Transposition, die den Plan A in den Plan B überführen, legen also auf diese Weise den Isomorphismus zwischen den zugehörigen Ablaufgraphen fest. □

Es wird im weiteren der in diesem Hilfssatz dargestellte Zusammenhang zur Gewinnung einer Komplexitätsaussage für eines der bekanntesten Probleme aus der Graphentheorie benutzt: Das Problem der Entscheidung, ob zwei gegebene Graphen isomorph sind, heißt *Graphen-Isomorphie-Problem*. Die Komplexität dieses Problems ist bis heute ungeklärt, d. h. es ist weder ein polynomialer Lösungsalgorithmus bekannt, noch konnte gezeigt werden, daß es sich um ein $\mathcal{NP}$-vollständiges Problem handelt.

[3]Ein *Turnier* $T = (V, E)$ mit $|V| = n$ ist eine Orientierung des vollständigen Graphen K_n.

Das Graphen-Isomorphie-Problem spielt in der Komplexitätstheorie eine bedeutende Rolle, denn es ist eines der Probleme in der Klasse $\mathcal{NP}$, die möglicherweise weder in der Klasse $\mathcal{P}$ liegen noch $\mathcal{NP}$-vollständig sind. Falls $\mathcal{P} \neq \mathcal{NP}$ gilt, existieren tatsächlich solche Probleme, die zwischen diesen beiden Komplexitätsklassen liegen (siehe LADNER [57]). Die große Bedeutung des Graphen-Isomorphie-Problems in diesem Bereich kommt auch durch die besondere Erwähnung in frühen Arbeiten der Komplexitätstheorie [27, 51, 36] zum Ausdruck.

Die Komplexität des *Digraphen-Isomorphie-Problems*, also des entsprechenden Entscheidungsproblems für gerichtete Graphen, ist polynomial äquivalent zum Graphen-Isomorphie-Problem (siehe MILLER [70]). Während für beliebige Graphen bzw. Digraphen bis heute kein effizienter Algorithmus zur Entscheidung über die Existenz eines Isomorphismus im allgemeinen Fall bekannt ist, hat man bereits einige Graphenklassen bestimmt, in denen dieses Problem in polynomialer Zeit gelöst werden kann: Es sind polynomiale Algorithmen z. B. im Falle von planaren Graphen (HOPCROFT und WONG [46]), Graphen mit beschränkter maximaler Knotenanzahl (LUKS [64]), Graphen mit beschränktem durchschnittlichen Geschlecht (CHEN [22]) und Intervallgraphen (LUEKER und BOOTH [63]) bekannt.

Zur Lösung des Digraphen-Isomorphie-Problems sind polynomiale Algorithmen für minimale serien-parallele Digraphen[4] (VALDES, TARJAN und LAWLER [98]) und für zyklische Turniere[5] (PONOMARENKO [74]) entwickelt worden.

Im anschließenden Satz zeigen wir, daß die Menge aller $n \times m$–Ablaufgraphen eine weitere Klasse von Digraphen ist, in der die Entscheidung über die Existenz eines Isomorphismus in polynomialer Zeit möglich ist.

Satz 5.1.7 [12] *Es ist in polynomialer Zeit entscheidbar, ob ein Isomorphismus zwischen zwei azyklischen Orientierungen des Hamming-Graphen $K_n \times K_m$ existiert.*

Beweis: Die Menge der azyklischen Orientierungen des Hamming-Graphen $K_n \times K_m$ entspricht der Menge der $n \times m$–Ablaufgraphen. Einem gegebenen $n \times m$–Ablaufgraphen kann nach Satz 3.2.2 eindeutig ein $n \times m$–Plan zugeordnet werden. Satz 3.2.4 zeigt, daß es in polynomialer Zeit möglich ist, zu einem gegebenen Ablaufgraphen den zugehörigen Plan zu berechnen. Zusammen mit dem polynomialen Algorithmus 5.1.4 zur Entscheidung, ob zwei Pläne äquivalent sind, und Hilfssatz 5.1.6 folgt, daß in der Klasse der azyklischen Orientierungen des Hamming-Graphen vom Typ $K_n \times K_m$ das Graphen-Isomorphie-Problem polynomial lösbar ist. □

[4]Ein *minimaler serien-paralleler Digraph* (MSP-Digraph) ist ein Digraph, der sich durch eine Folge von seriellen und parallelen Kompositionen rekursiv aus kleineren MSP-Digraphen konstruieren läßt, wobei der triviale Digraph mit nur einem Knoten auch ein MSP-Digraph ist (Rekursionsanfang).

[5]Ein Turnier $T = (V, E)$ heißt *zyklisch*, wenn seine Automorphismengruppe, d. h. die Gruppe der Isomorphismen $\varphi : V \to V$, die zyklische Permutation $(1, 2, \ldots, n)$ enthält.

Struktur-Äquivalenz von Plänen

Es ist sinnvoll, bei der Zusammenfassung von $n \times m$–Plänen mit gleichen Eigenschaften nicht nur die jeweilige Neuindizierung der Aufträge J_i und Maschinen M_j untereinander (Isomorphie) sowie zusätzlich die komplette Vertauschung von Aufträgen und Maschinen im Fall $n = m$ (Äquivalenz) zuzulassen. Die Grundstruktur eines Planes ändert sich ebenfalls nicht, wenn alle Reihenfolgen von Technologie und Organisation vollständig umgekehrt werden. Daher wird in diesem Abschnitt zusätzlich zur Isomorphie „$\cong$“ und zur Äquivalenz „$\equiv$“ eine weitere Äquivalenzrelation für Pläne betrachtet, die auf der Umkehrung aller Reihenfolgen bzw. gerichteten Kanten im entsprechenden Ablaufgraphen basiert. Bei dieser Äquivalenzrelation handelt es sich um die sogenannte Struktur-Äquivalenz.

Es sei A ein Plan. Der Plan, der sich durch Umkehrung der Technologie und Organisation von A ergibt, heißt *Umkehrplan* $\overline{A}$ von A

Definition 5.1.8 Zwei Pläne A und B heißen *struktur-äquivalent*, $A \equiv_S B$, wenn $A \equiv B$ oder $A \equiv \overline{B}$ gilt.

Diese durch „$\equiv_S$“ definierte Äquivalenzrelation ist nicht so streng gefaßt wie die Äquivalenz („$\equiv$“) und die Isomorphie („$\cong$“): Offensichtlich folgt für zwei Pläne A und B aus $A \cong B$ die Beziehung $A \equiv B$, und $A \equiv B$ impliziert die Beziehung $A \equiv_S B$.

Vertretersysteme

In [12] haben Bräsel, Harborth und Willenius die Anzahl der Isomorphie- bzw. Äquivalenzklassen in der Menge der Pläne folgendermaßen bestimmt: Es werden zunächst alle $n \times m$–Pläne mit Hilfe des vollständigen Enumerationsalgorithmus (Bräsel und M. Kleinau [14]) für kleine Werte n und m erzeugt. Danach müssen diese $n \times m$–Pläne paarweise anhand von Algorithmus 5.1.4 verglichen werden, um letztlich nur die nicht-isomorphen bzw. nicht-äquivalenten Pläne zu zählen.

Diese Methode der Enumeration nicht-isomorpher bzw. nicht-äquivalenter Pläne ist aufgrund der großen Anzahl der durchzuführenden paarweisen Vergleiche zur schnellen Berechnung der gewünschten Plan-Anzahl bereits für $m \geq n \geq 4$ unzureichend. Es wird nun die in [11] entwickelte neue Grundlage vorgestellt, mit der alle Pläne eines gegebenen Formats effektiver als in [12] erzeugt bzw. gezählt werden können.

Zur Enumeration der Äquivalenzklassen bestimmter kombinatorischer Objekte wendet man häufig algebraische Methoden an, mit denen in geeigneter Weise nur die Vertreter eines disjunkten Vertretersystems für die Äquivalenzklassen erzeugt oder gezählt werden.

Es seien $A = (a_{ij})$ und $B = (b_{ij})$ zwei $n \times m$–Matrizen mit $a_{ij}, b_{ij} \in \mathbb{N}$ für alle $i = 1, \ldots, n$ und $j = 1, \ldots, m$. Die Matrix A heißt *lexikographisch kleiner* als die Matrix B ($A <_{lex} B$), wenn es ein Indexpaar (k, l) gibt, für das $a_{kl} < b_{kl}$ gilt und für alle Indexpaare (i, j) mit $1 \leq i < k,\ 1 \leq j \leq m$ und $i = k,\ 1 \leq j < l$ die Beziehung $a_{ij} = b_{ij}$ erfüllt ist. Auf diese Weise ist auf der Menge S der $n \times m$–Matrizen mit Einträgen aus $\mathbb{N}$ die *lexikographische Ordnung* der Form $A_1 <_{lex} A_2 <_{lex} \cdots <_{lex} A_r$ für die Matrizen $A_1, A_2, \ldots, A_r$ gegeben. Offensichtlich handelt es sich bei der Menge S mit dieser Ordnung um eine linear geordnete Menge[6].

Definition 5.1.9 Es sei S die Menge aller $n \times m$–Matrizen mit Einträgen aus $\mathbb{N}$. Eine Menge $Q \subseteq S$ werde durch eine Äquivalenzrelation in die disjunkten Teilmengen $Q_1, \ldots, Q_r$ partitioniert, also $Q = Q_1 \cup \cdots \cup Q_r$ und $Q_i \cap Q_j = \varnothing$ für $i \neq j$. Es sei $\min_{lex}(Q_i)$ das lexikographische Minimum der Elemente der Menge $Q_i \subseteq S$. Dann ist die Funktion f mit

$$\begin{aligned} f : \{1, \ldots, r\} &\rightarrow S, \\ i &\mapsto f(i) = \min_{lex}(Q_i) \end{aligned} \tag{5.1}$$

die Auswahlfunktion eines Vertretersystems für die gegebene Äquivalenzrelation auf Q.

Isomorphie, Äquivalenz und Struktur-Äquivalenz stellen Äquivalenzrelationen dar, durch die die Menge aller $n \times m$–Pläne jeweils in disjunkte Äquivalenzklassen partitioniert wird. In den folgenden Abschnitten beruhen die benutzten Vertretersysteme für die Äquivalenzklassen in der Menge der Technologien und Pläne direkt oder indirekt auf der Auswahlfunktion (5.1). Es wird also häufig die lexikographische kleinste Matrix als Repräsentant ihrer Klasse benutzt.

In Abbildung 5.1 sind unter allen 2×2–Plänen alle Vertreter einer Isomorphie- bzw. Äquivalenzklasse hervorgehoben. Außerdem sind in dieser Abbildung die vier verschiedenen Isomorphieklassen zeilenweise angeordnet.

5.2 Technologie-Anzahlen

Im ersten Teil neuen Enumerationsalgorithmus für $n \times m$–Pläne werden zunächst nur Technologien betrachtet. Eine Technologie wird stets mit der Matrix *TR* identifiziert (siehe Abbildung 3.5, Seite 23). Diese $n \times m$–Matrix *TR* besteht für ein Shop-Scheduling-Problem mit n Aufträgen und m Maschinen aus n Zeilen, die jeweils Permutationen der Zahlen $1, \ldots, m$ darstellen.

[6] Eine *linear geordnete Menge* ist eine Menge in der je zwei Elemente anhand einer zweistelligen Relation vergleichbar sind.

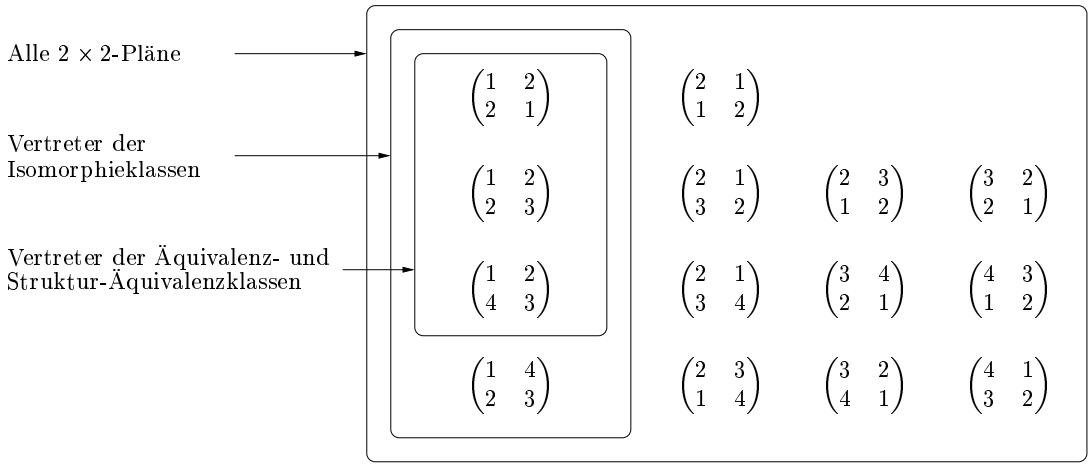


Abbildung 5.1: Ein Vertretersystem der 2×2–Pläne.

$n \backslash m$	2	3	4	5	6
2	4	36	576	14 400	518 400
3	8	216	13 824	1 728 000	373 248 000
4	16	1 296	331 776	207 360 000	268 738 560 000
5	32	7 776	7 962 624	24 883 200 000	193 491 763 200 000
6	64	46 656	191 102 976	2 985 984 000 000	139 314 069 504 000 000
7	128	279 936	4 586 471 424	358 318 080 000 000	100 306 130 042 880 000 000

Tabelle 5.1: Gesamtanzahlen der $n \times m$–Technologien.

Im folgenden wird das in Abschnitt 5.1 angewandte Prinzip der Zusammenfassung von Plänen gleicher Struktur auf die Technologien TR übertragen, um sich unter der Gesamtanzahl der Technologien eines Formats (vgl. Tabelle 5.1) auf die strukturell verschiedenen konzentrieren zu können.

Definition 5.2.1 Zwei Technologien TR^1 und TR^2 heißen *isomorph* ($TR^1 \cong TR^2$), wenn TR^1 durch eine Permutation ϱ von Zeilen und eine Permutation σ von Spalten in TR^2 überführt werden kann.

Zur Abschätzung der Anzahl nicht-isomorpher Technologien ist die Betrachtung sogenannter reduzierter Technologien TR sinnvoll. In der folgenden Definition und im weiteren Teil dieses Kapitels wird die Gesamtheit aller Einträge einer Zeile TR_i der Technologie TR stets als Zeilenvektor $TR_i = (tr_{i1}, tr_{i2}, \ldots, tr_{im})$ aufgefaßt.

Definition 5.2.2 Eine $n \times m$–Technologie TR heißt *reduziert*, wenn

$$TR_1 = (1, 2, \ldots, m)$$

ist, und $TR_i \leq_{lex} TR_{i+1}$ für alle $1 \leq i \leq n-1$ gilt.

Hilfssatz 5.2.3 [11] *Für die Anzahl $I_{TR}(n,m)$ der Isomorphieklassen der Technologien TR des Formats $n \times m$ gilt*

$$\frac{1}{n}\binom{m!+n-2}{n-1} \leq I_{TR}(n,m) \leq \binom{m!+n-2}{n-1}. \tag{5.2}$$

Beweis: Jede Technologie TR kann durch Zeilen- und Spaltenpermutationen in eine reduzierte Form überführt werden. Also ist die Anzahl der reduzierten TR's eine obere Schranke für die Anzahl nicht-isomorpher TR's. Da die erste Zeile einer reduzierten TR festgelegt ist, entspricht die Anzahl der reduzierten TR's der Anzahl der $(n-1)$–Kombinationen der Menge der Permutationen der Länge m mit Wiederholung, und diese Anzahl ist $\binom{m!+n-2}{n-1}$.

Es sei k die Anzahl verschiedener Zeilenvektoren in Technologien TR einer Isomorphieklasse. Dann enthält diese Isomorphieklasse höchstens k verschiedene reduzierte TR's, denn jede der k verschiedenen Zeilen kann durch Anwendung geeigneter Zeilen- und Spaltenpermutation als erste Zeile benutzt werden. Da eine Technologie TR höchstens n verschiedene Zeilen besitzt, gilt die in (5.2) angegebene untere Schranke. □

Ein *Automorphismus* (ϱ, σ) einer Technologie TR ist eine Zeilen- und Spaltenpermutation, die TR in sich selbst überführt. Ein Automorphismus (ϱ, σ) heißt *nichttrivial*, wenn die Spaltenpermutation σ nicht die Identität ist.

Die in Hilfssatz 5.2.3 gegebenen Schranken für die Anzahl nicht-isomorpher Technologien sind nicht scharf. Satz 5.2.5 zeigt, daß sich die Anzahl $I_{TR}(n,m)$ für unendlich viele Paare (n,m) exakt angeben läßt. Zum Beweis dieses Ergebnisses benötigt man folgendes vorbereitendes Resultat:

Hilfssatz 5.2.4 [11] *Es gibt genau dann eine $n \times m$–Technologie TR mit einem nichttrivialen Automorphismus, wenn n durch eine Zahl $p \in \mathbb{N}$ mit $1 < p \leq m$ teilbar ist.*

Beweis: Es sei TR eine $n \times m$–Technologie mit einem nichttrivialen Automorphismus (ϱ, σ). Dann enthält die Spaltenpermutation σ mindestens einen Zyklus der Länge p mit $1 < p \leq m$. Es gilt $TR_{\varrho(i)} = \sigma(TR_i)$ für alle i, da (ϱ, σ) ein Automorphismus ist. Sei r die Länge des Zyklus von ϱ, der i enthält. Dann gilt $\varrho^r(i) = i$ und damit $\sigma^r(TR_i) = TR_i$. Also ist σ^r die Identität und r ein Vielfaches von p. Die gleiche Schlußfolgerung kann für alle Zeilen i, $1 \leq i \leq n$ angewandt werden. Jeder Zyklus in ϱ hat daher als Länge ein Vielfaches von p. Da n die Summe aller Zykluslängen von ϱ ist, ist n ebenfalls Vielfaches von p.

Sei umgekehrt $n = sp$ mit $1 \leq p \leq m$. Dann ist die Technologie $TR = (tr_{ij})$ mit

$$tr_{ij} = \begin{cases} \big((\lceil i/s \rceil + j - 2) \mod p \big) + 1 & \text{für } j \leq p; \\ j & \text{für } p < j \leq m; \end{cases} \tag{5.3}$$

für alle $1 \leq i \leq n$, eine $n \times m$–Technologie, die einen nichttrivialen Automorphismus besitzt. □

Die im Beweis angegebene Matrix ist die einfachste Form einer Technologie mit einem nichttrivialen Automorphismus. Zum Beispiel ergibt sich aus (5.3) mit $n = 6$, $m = 7$ und $p = 3$ die Technologie

$$(tr_{ij}) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 1 & 4 & 5 & 6 & 7 \\ 2 & 3 & 1 & 4 & 5 & 6 & 7 \\ 2 & 3 & 1 & 4 & 5 & 6 & 7 \end{pmatrix}. \tag{5.4}$$

Die Aufzählung der Technologien bereitet nur dann Schwierigkeiten, wenn nichttriviale Automorphismen existieren. Ansonsten kann die Anzahl $I_{TR}(n,m)$ der paarweise nicht-isomorphen Technologien anhand des folgenden Satzes leicht berechnet werden.

Satz 5.2.5 [11] *Es sei n durch keine Zahl p mit $1 < p \leq m$ teilbar. Dann gilt*

$$I_{TR}(n,m) = \sum_{k=1}^{n} \binom{m! - 1}{k - 1} \binom{n-1}{k-1} \frac{1}{k}. \tag{5.5}$$

Beweis: Es wird zunächst die Anzahl der reduzierten Technologien TR mit genau k paarweise verschiedenen Zeilenvektoren $TR_i,\ i = 1, \ldots, k$ betrachtet. Der erste Zeilenvektor einer reduzierten Technologie ist stets $TR_1 = (1, 2, 3, \ldots, m)$, daher verbleiben $\binom{m!-1}{k-1}$ Möglichkeiten, diese Menge von insgesamt k verschiedenen Zeilenvektoren auszuwählen. Weiterhin tritt jeder dieser Zeilenvektoren $TR_i,\ i = 1, \ldots, k$ mit einer Häufigkeit $h_i \geq 1$ auf, wobei offensichtlich $n = h_1 + h_2 + \cdots + h_k$ gilt. Die Anzahl der Möglichkeiten, die Zahl n auf diese Weise in k positive Summanden zu zerlegen, ist $\binom{n-1}{k-1}$. Also existieren insgesamt $\binom{m!-1}{k-1}\binom{n-1}{k-1}$ reduzierte Technologien TR mit genau k verschiedenen Zeilenvektoren. Wie bereits im Beweis von Hilfssatz 5.2.3 gezeigt wurde, enthält jede Isomorphieklasse von Technologien mit k verschiedenen Zeilen maximal k verschiedene reduzierte Technologien. Nach Voraussetzung ist n durch keine Zahl p mit $1 < p \leq m$ teilbar. Wegen Hilfssatz 5.2.4 hat jede solche Isomorphieklasse in diesem Fall genau k verschiedene reduzierte Technologien. Die Aussage des Satzes folgt also, wenn man über k summiert und dabei jeweils den Faktor $1/k$ hinzufügt. □

Anwendung des Cauchy-Frobenius-Hilfssatzes

In diesem Abschnitt wird eine weit verbreitete algebraische Methode angewendet, um eine exakte Formel für die Anzahl $I_{TR}(n, m)$ der Isomorphieklassen der $n \times m$–Technologien herleiten zu können. Das zentrale Hilfsmittel in diesem Zusammenhang ist der sogenannte Cauchy-Frobenius-Hilfssatz, der zunächst zitiert wird. Vorbereitend dazu benötigt man die anschließend erklärten Begriffe.

Es sei G eine Gruppe von Elementen, die auf einer endlichen Menge X operiere, also $G \times X \to X, (g, x) \mapsto g(x)$. Man kann die Gruppe G bezüglich X in sogenannte Bahnen zerlegen: Für ein Element $x \in X$ heißt die Menge $G(x) = \{g(x) \mid g \in G\}$ die *Bahn* von x. Der *Stabilisator* von x ist die Menge $G_x = \{g \in G \mid g(x) = x\}$. Offensichtlich ist der Stabilisator von x eine Untergruppe von G. Weiterhin kann gezeigt werden, daß stets $|G(x)|\ |G_x| = |G|$ gilt.

Hilfssatz 5.2.6 [21] *Sei G eine Gruppe, die auf einer endlichen Menge X operiere. Mit $B(G)$ werde die Menge der Bahnen von G auf X bezeichnet. Es gilt*

$$|B(G)| = \frac{1}{|G|} \sum_{g \in G} |\{x \in X : g(x) = x\}|. \tag{5.6}$$

Dieser Cauchy-Frobenius-Hilfssatz ist irrtümlich auch unter dem Namen „Burnside's Lemma“ bekannt. Dieser Irrtum ist durch folgenden Zusammenhang begründet: Während Burnside in [21] noch die entsprechende Literaturquelle zitiert, ist dies in der 2. Auflage von [21] nicht mehr der Fall. Daher sind viele Autoren später davon

ausgegangen, daß dieser Hilfssatz auf BURNSIDE zurückgeht. Der Hilfssatz ist jedoch CAUCHY und FROBENIUS bereits deutlich früher bekannt gewesen. Historische Anmerkungen hierzu findet man in [5, 28, 73, 103].

Die *symmetrische Gruppe* S_m ist die Gruppe der Ordnung $m!$, die aus allen Permutationen der Zahlen $1, 2, \ldots, m$ besteht. Es sei π ein Element von S_m. Der kleinste Wert k, für den π^k die Identität ist, heißt *Ordnung* der Permutation π. Der Cauchy-Frobenius-Hilfssatz gibt uns im Zusammenhang mit der Enumeration der Pläne von Shop-Scheduling-Problemen die Möglichkeit, die Anzahl der nicht-isomorphen Technologien exakt anzugeben:

Satz 5.2.7 [11] *Es sei $n_{k,m}$ die Anzahl der Permutationen der Ordnung k in der symmetrischen Gruppe S_m. Dann gilt*

$$I_{TR}(n,m) = \frac{1}{m!} \sum_{k|n} n_{k,m} \binom{\frac{m!}{k} + \frac{n}{k} - 1}{\frac{n}{k}}. \tag{5.7}$$

Beweis: Der Cauchy-Frobenius-Hilfssatz wird direkt angewandt. Es sei X die Menge der reduzierten Technologien TR und G die symmetrische Gruppe S_m. Der Wert $I_{TR}(n,m)$ ist dann gerade die Anzahl der Bahnen von G auf X. Jedes Element $\sigma \in S_m$ wird als Spaltenpermutation der Technologie TR aufgefaßt, und es werde zu σ stets automatisch die Zeilenpermutation angewandt, die wieder eine reduzierte Technologie erzeugt. Für alle Spaltenpermutationen σ wird jeweils die Anzahl der Technologien TR gesucht, für die σ ein Automorphismus ist.

Offensichtlich kann σ nur dann ein Automorphismus sein, wenn die Ordnung von σ ein Teiler von n ist. Es sei k die Ordnung von σ in S_m, dann sind alle Zeilenvektoren $TR_i, \sigma(TR_i), \sigma^2(TR_i), \ldots, \sigma^{k-1}(TR_i)$ verschieden. Daher enthält jede Technologie, die unter σ gleich bleibt, entweder alle oder keinen dieser Zeilenvektoren. Das heißt, durch einen ausgewählten Zeilenvektor sind stets automatisch weitere $k-1$ festgelegt, und man erhält als Anzahl verschiedener Technologien TR, für die σ Automorphismus ist, die Anzahl der n/k–Kombinationen aus $m!/k$ solchen Mengen von Zeilenvektoren mit Wiederholung. Durch Summation über alle Spaltenpermutationen σ, deren Ordnung Teiler von n ist, ergibt sich (5.7). □

Wenn die Anzahl $n_{k,m}$ der Permutationen der Ordnung k in der symmetrischen Gruppe S_m für alle k mit $k|n$ bekannt ist, liefert dieser Satz sofort die Anzahl $I_{TR}(n,m)$ nicht-isomorpher Technologien des Formats $n \times m$. Eine Übersicht über die Werte $I_{TR}(n,m)$ für $1 \leq n \leq 9$ und $1 \leq m \leq 6$ gibt Tabelle 5.2. Beim Vergleich mit Tabelle 5.1 wird die erhebliche Reduzierung der Technologie-Anzahlen deutlich, die durch die Beschränkung auf nicht-isomorphe Technologien entsteht.

$n\backslash m$	2	3	4	5	6
2	2	5	17	73	398
3	2	10	111	2 467	86 787
4	3	24	762	76 044	15 688 744
5	3	42	4 095	1 876 255	2 270 743 529
6	4	83	19 941	39 096 565	274 382 326 290
7	4	132	84 825	703 593 825	28 457 281 936 435
8	5	222	329 214	11 169 676 185	2 586 055 570 098 800
9	5	335	1 168 740	158 855 852 180	209 183 155 674 562 575

Tabelle 5.2: Anzahlen nicht-isomorpher $n \times m$–Technologien.

Struktur-Isomorphie von Technologien

In diesem Abschnitt wird in Anlehnung an die Pläne auch für die Technologien eine Erweiterung der einfachen Isomorphie („$\cong$“) gegeben. Da eine Technologie TR durch Transposition nicht notwendig wieder in eine Technologie überführt wird, kann die Äquivalenz von Plänen nicht für Technologien TR angewandt werden. Man kann aber die Technologien als „strukturell gleichwertig“ identifizieren, in denen die technologischen Reihenfolgen aller Aufträge vollständig umgekehrt sind. Die Technologie, die sich durch Umkehrung aller technologischen Reihenfolgen in TR ergibt, heißt *Umkehrtechnologie* $\overline{TR}$ von TR.

Definition 5.2.8 Zwei Technologien TR^1 und TR^2 heißen *struktur-isomorph*, geschrieben $TR^1 \cong_S TR^2$, wenn $TR^1 \cong TR^2$ oder $TR^1 \cong \overline{TR^2}$ gilt.

Für die Anzahl $S_{TR}(n,m)$ der nicht-struktur-isomorphen Technologien TR läßt sich analog zu Satz 5.2.7 der Cauchy-Frobenius-Hilfssatz anwenden, wobei dann die Gruppe $S_m \times \mathbb{Z}_2$ anstelle von S_m zugrunde gelegt werden muß, da jeweils die Operation der Umkehrung aller technologischen Reihenfolgen hinzukommt.[7] Das Abzählen der Technologien TR, die unter einer Operation $(\sigma, u) \in S_m \times \mathbb{Z}_2$ fix bleiben, ist jedoch weitaus schwieriger als im Fall ohne mögliche Umkehrtechnologien, daher ergibt sich eine kompliziertere Formel als (5.7), auf deren Darstellung hier verzichtet wird. In Tabelle 5.3 wird analog zu Tabelle 5.2 eine Übersicht über die Werte $S_{TR}(n,m)$ gegeben. Beim Vergleich mit Tabelle 5.1 (Seite 57) kann wiederum die deutliche Anzahlreduzierung festgestellt werden: Während z. B. die Gesamtanzahl der 7×6–Technologien bei $1,003 \times 10^{20}$ liegt, ist $I^*_{TR}(7,6) \approx 1,423 \times 10^{13}$.

[7] Die Gruppe $\mathbb{Z}_2$ ist die *zyklische Gruppe* der Ordnung 2; und die Gruppe $S_m \times \mathbb{Z}_2$ ist das *direkte Produkt* aus den Gruppen S_m und $\mathbb{Z}_2$.

$n\backslash m$	2	3	4	5	6
2	2	4	13	45	230
3	2	7	67	1 269	43 767
4	3	16	434	38 356	7 854 456
5	3	26	2 175	939 395	1 135 495 745
6	4	50	10 385	19 556 801	137 193 369 114
7	4	76	43 353	351 827 297	14 228 666 657 843
8	5	126	167 102	5 585 002 649	1 293 028 139 377 488
9	5	185	589 648	79 428 476 802	104 591 581 641 412 531

Tabelle 5.3: Anzahlen nicht-struktur-isomorpher $n \times m$–Technologien.

Mit Hilfe der Datenbank *On-Line Encyclopedia of Integer Sequences*, die von SLOANE gepflegt wird und sich im World-Wide-Web unter der Adresse

`http://www.research.att.com/~njas/sequences/`

befindet, läßt sich ein interessanter Zusammenhang zwischen der Anzahl sogenannter Armbänder und den Werten aus Tabelle 5.2 und 5.3 entdecken. Auf diese Korrespondenz wird im folgenden näher eingegangen.

Definition 5.2.9 Ein *k-Armband (k-bracelet)* ist eine Äquivalenzklasse von zyklischen Null-Eins-Folgen der Länge k unter den Operationen der Drehung und Spiegelung.[8]

Die Anzahl der k-Armbänder entspricht der Anzahl der Möglichkeiten, schwarze und weiße Perlen auf ein Armband mit insgesamt k Perlen aufzuziehen. Dabei gehören diejenigen Darstellungen der Armbänder einer Äquivalenzklasse an, die durch Drehung und Achsenspiegelung ineinander überführt werden können. So sind zum Beispiel die zyklischen Folgen 001011 und 001101 gleichwertig (siehe Abbildung 5.2).

Satz 5.2.10 *Die Anzahl der nicht-struktur-isomorphen $n \times 3$–Technologien TR ist gleich der Anzahl der $(n+6)$–Armbänder mit n weißen und 6 schwarzen Perlen.*

Beweis: Es sei $f : S_3 \to \{0, 1, \ldots, 5\}$ eine Funktion mit

$$\begin{array}{l} (1,2,3) \mapsto 0,\ (1,3,2) \mapsto 1,\ (2,3,1) \mapsto 2, \\ (3,2,1) \mapsto 3,\ (3,1,2) \mapsto 4,\ (2,1,3) \mapsto 5, \end{array} \tag{5.8}$$

[8]Ein Armband (bracelet) ist in der Literatur häufig auch unter dem Namen *Perlenkette (necklace)* bekannt (siehe z. B. `http://sue.csc.uvic.ca/~cos/inf/neck/NecklaceInfo.html`).

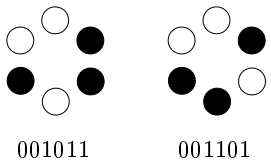


Abbildung 5.2: Ein 6-Armband in zwei verschiedenen Darstellungen.

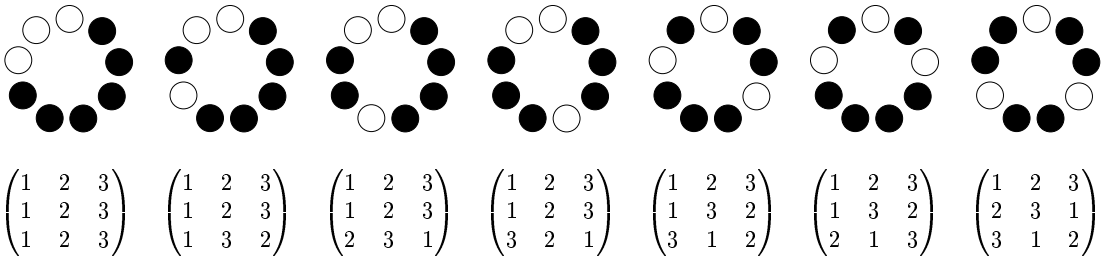


Abbildung 5.3: Eine Bijektion zwischen 9-Armbändern und den Repräsentanten der Struktur-Isomorphie-Klassen von 3×3–Technologien.

wobei S_3 die symmetrische Gruppe der Ordnung 3 ist. Jeder Zeilenvektor TR_i einer $n \times 3$–Technologie TR wird als Permutation der Zahlen $\{1, 2, 3\}$ aufgefaßt. Eine Technologie TR heißt *f-geordnet*, wenn $f(TR_i) \leq f(TR_{i+1})$ für alle $i = 1, \ldots, n-1$ gilt. Offensichtlich kann jede Technologie durch eine geeignete Zeilenpermutation in eine f-geordnete Technologie TR überführt werden. Es läßt sich nun eine Bijektion zwischen den f-geordneten Technologien TR des Formats $n \times 3$ und den Darstellungen der $(n+6)$–Armbänder formulieren.

Eine f-geordnete Technologie TR wird auf eine Darstellung eines Armbandes folgendermaßen abgebildet: Für $i = 1, 2, \ldots, n$ ist die Anzahl der schwarzen Perlen, die sich zwischen der obersten Perle und der i-ten weißen Perle im mathematisch positiven Sinne befinden, gleich $f(TR_i)$, wobei die Funktion f durch (5.8) gegeben ist (siehe z. B. Abbildung 5.3 im Fall $n = 3$). Es ist dabei zu beachten, daß jeweils n der Darstellungen der $(n+6)$–Armbänder nicht unterschieden werden. Und zwar handelt es sich jeweils um diejenigen Darstellungen, für die bezüglich der gerade eingeführten Abbildung kein Urbild definiert ist, da sich zwischen der obersten Perle und einer i-ten Perle > 5 schwarze Perlen befinden (siehe z. B. Abbildung 5.4 im Fall $n = 3$).

Zu jeder der unterschiedenen Darstellungen der $(n+6)$–Armbänder gibt es genau ein Urbild, daher handelt es sich insgesamt bei dieser Abbildung um eine eineindeutige Zuordnung.

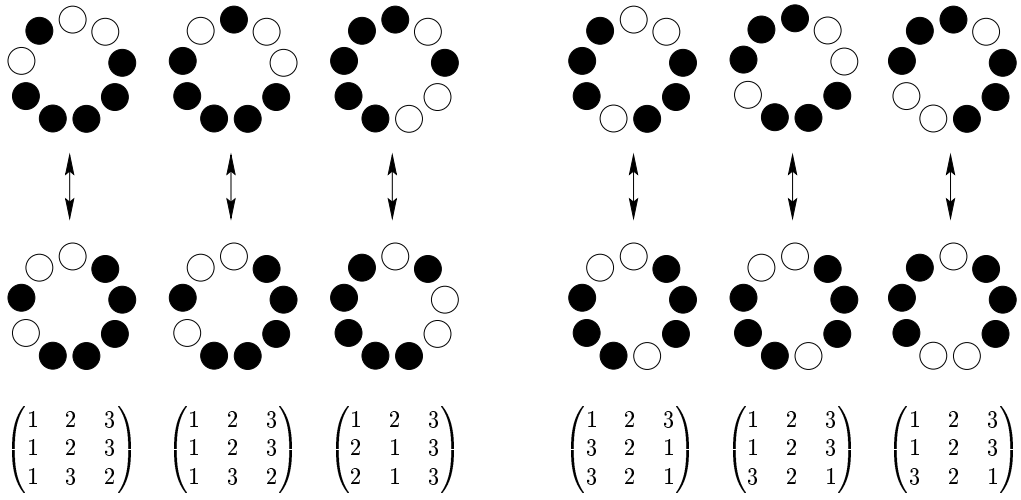


Abbildung 5.4: Zwei Beispiele für die Identifizierung von jeweils 3 Darstellungen der 9-Armbändern (es werden jeweils die obere und untere Darstellung nicht unterschieden).

Da von den $n+6$ Positionen, in die ein $(n+6)$–Armband gedreht werden kann, n nicht unterschieden werden, ist die Gruppe der Drehspiegelungen, die auf dieser Menge von Darstellungen der $(n+6)$–Armbänder operiert, die Diedergruppe[9] D_{12}. Die Gruppe der Operationen (Spaltenpermutationen und Umkehrung), die auf den f-geordneten $n \times 3$–Technologien definiert sind, ist $S_3 \times \mathbb{Z}_2$. Es kann gezeigt werden, daß die Gruppen D_{12} und $S_3 \times \mathbb{Z}_2$ isomorph sind, daher ist die Anzahl der Struktur-Isomorphieklassen von $n \times 3$–Technologien tatsächlich gleich der Anzahl der verschiedenen $(n+6)$–Armbänder mit n weißen und sechs schwarzen Perlen. □

Die gesuchte Anzahl der $(n+6)$–Armbänder mit n weißen und sechs schwarzen Perlen läßt sich auch direkt anhand des Cauchy-Frobenius-Hilfssatzes ermitteln, indem gezählt wird, wieviele Darstellungen der $(n+6)$–Armbänder unter den Operationen der Diedergruppe $D_{2(n+6)}$ fix bleiben (in diesem Fall werden alle $n+6$ „Drehpositionen“ unterschieden).

Der anschließende Satz zeigt, daß für kein $m > 3$ eine Satz 5.2.10 entsprechende Beziehung existiert.

Satz 5.2.11 *Die Anzahl der nicht-struktur-isomorphen $n \times m$–Technologien TR (n variabel) entspricht für kein festes $m > 3$ der Anzahl der $(n+m!)$–Armbänder mit*

[9] Die *Diedergruppe* D_{2n} ist eine Gruppe der Ordnung $2n$, die als Menge von Drehspiegelungen des regelmäßigen n-Ecks aufgefaßt werden kann.

n weißen und $m!$ schwarzen Perlen.

Beweis: Wie im Beweis zu Satz 5.2.10 wird mittels einer geeigneten Funktion $f : S_m \to \{1, 2, \ldots, m!\}$ eine Bijektion zwischen den f-geordneten $n \times m$–Technologien TR und den $(n + m!)$–Armbändern mit n weißen und $m!$ schwarzen Perlen hergestellt, wobei bei den Darstellungen der $(n + m!)$–Armbänder wieder jeweils n Positionen nicht unterschieden werden. Die Gruppe der Drehspiegelungen, die auf diesen Armbändern operiert, ist die Diedergruppe $D_{2(m!)}$. Die Gruppe, die auf den f-geordneten $n \times m$–Technologien TR operiert, ist $S_m \times \mathbb{Z}_2$. Jede Diedergruppe $D_{2(m!)}$ hat ein Element der Ordnung $m!$. Aber kein Element von $S_m \times \mathbb{Z}_2$ hat für $m > 3$ ein Element der Ordnung $m!$, daher sind die Gruppen $D_{2(m!)}$ und $S_m \times \mathbb{Z}_2$ für kein $m > 3$ isomorph, und die im Satz genannten Anzahlen (m fest, n variabel) stimmen nicht überein. □

5.3 Ein neuer Enumerationsalgorithmus

In diesem Abschnitt wird ein neuer Algorithmus zur Enumeration aller $n \times m$–Pläne für fest vorgegebene Werte n und m vorgestellt. Die gesamte Plan-Enumeration gliedert sich zunächst in zwei Teilprozeduren: Lexikographische Technologie-Erzeugung und Technologie-Minimalitätstest. Diese Teilprozeduren werden im anschließenden Unterabschnitt behandelt, bevor die gesamte Plan-Enumeration Mittelpunkt des darauf folgenden Unterabschnitts ist. Einige Ergebnisse und Bestandteile der benutzten Verfahren sind bereits in [11] enthalten. In der vorliegenden Arbeit sind die zur Plan-Enumeration benötigten Algorithmen im Gegensatz zu [11] ausführlich anhand von Pseudocode-Programmen beschrieben.

Erzeugung von Permutationen

Ein Isomorphismus (ϱ, σ) oder Struktur-Isomorphismus (ϱ, σ, u), der eine Technologie TR^1 auf eine Technologie TR^2 abbildet, definiert gleichzeitig eine Bijektion zwischen den beiden Mengen von Plänen, deren Technologien entweder TR^1 oder TR^2 entsprechen. Diese Bijektion bildet einen Plan A mit TR^1 jeweils auf einen Plan B mit TR^2 ab. Daher ist es zur Enumeration nicht-isomorpher Pläne zunächst ausreichend, ausschließlich nicht-isomorphe bzw. nicht-struktur-isomorphe Technologien zu generieren.

Mit Hilfe der anschließend beschriebenen Algorithmen wird ein Vertretersystem für nicht-isomorphe Technologien auf der Grundlage von Auswahlfunktion (5.1) erzeugt. Zuerst werden Technologien TR in lexikographischer Reihenfolge ausgegeben (Algorithmus 5.3.1). Dann erfolgt für alle so konstruierten Technologien TR jeweils

ein Minimalitätstest (Algorithmus 5.3.3), der entscheidet, ob es sich bei der gegebenen Technologie um das lexikographische Minimum ihrer (Struktur-)Isomorphieklasse handelt. Offensichtlich ist das lexikographische Minimum der Technologien TR stets reduziert, daher reicht es bei der *Lexikographischen Technologie-Erzeugung* aus, ausschließlich reduzierte Technologien zu generieren.

Algorithmus 5.3.1 *Lexikographische Technologie-Erzeugung*

Eingabe: Parameter n und m.
Ausgabe: Alle reduzierten $n \times m$–Technologien in lexikographisch aufsteigender Reihenfolge.

1. Setze $TR := TR^0$, wobei TR^0 die lexikographisch kleinste $n \times m$–Technologie ist, also
$$TR^0 = \begin{pmatrix} 1 & 2 & 3 & \cdots & m \\ 1 & 2 & 3 & \cdots & m \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2 & 3 & \cdots & m \end{pmatrix}.$$

2. Bestimme den größten Zeilenindex i von TR, so daß der Zeilenvektor TR_i gemäß Schritt 3 noch lexikographisch erhöht werden kann. Falls kein solches i existiert, Stop!

3. Erhöhe den Zeilenvektor $TR_i = (t_{i1}, t_{i2}, \ldots, t_{im})$ wie folgt:

   (a) Bestimme den größten Spaltenindex j mit $t_{i,j-1} < t_{ij}$, d. h. es gilt $t_{ij} > t_{i,j+1} > \cdots > t_{i,m-1} > t_{im}$ oder $j = m$.

   (b) Bestimme den größten Spaltenindex $k \geq j$ mit $t_{i,j-1} < t_{ik}$, d. h. t_{ik} ist die kleinste Zahl unter den $t_{ij}, \ldots, t_{im}$, die größer als t_{ij-1} ist.

   (c) Setze $TR_{i'} := (t_{i1}, \ldots, t_{i,j-2}, t_{ik}, t_{im}, \ldots, t_{i,k+1}, t_{i,j-1}, t_{i,k-1}, \ldots, t_{ij})$, d. h. die Reihenfolge von $t_{ij}, t_{i,j+1}, \ldots, t_{im}$ wird umgekehrt und $t_{i,j-1}$ wird mit t_{ik} vertauscht.

4. Setze $TR' := (TR_1, TR_2, \ldots, TR_{i-1}, TR_{i'}, TR_{i'}, \ldots, TR_{i'})^T$ und gib TR' aus.

5. Setze $TR := TR'$ und gehe zu Schritt 2.

Satz 5.3.2 *Die Prozedur der Erzeugung der lexikographisch nachfolgenden Technologie zu einer gegebenen reduzierten $n \times m$–Technologie kann in der Zeit $O(nm)$ ausgeführt werden.*

Beweis: Die Korrektheit von Algorithmus 5.3.1 ist offensichtlich. Die Schritte der Erhöhung einer Zeile (Schritte 3a-3c) benötigen jeweils $O(m)$ Zeit. Da maximal $n-1$ Zeilen auf die beschriebene Weise lexikographisch erhöht werden müssen, wird für den Schritt von einer reduzierten Technologie TR zur lexikographisch folgenden im schlechtesten Fall $O(nm)$ Zeit benötigt. □

Für jede Technologie TR', die bei der *Lexikographischen Technologie-Erzeugung* ausgegeben wird, kann zur Entscheidung, ob TR' ein Vertreter ihrer Isomorphieklasse ist, der folgende Algorithmus angewandt werden.

Algorithmus 5.3.3 *Technologie-Minimalitätstest*

Eingabe: Eine beliebige $n \times m$–Technologie TR^*.
Ausgabe: Entscheidung, ob die gegebene Technologie TR^* das lexikographische Minimum ihrer Isomorphieklasse ist.

1. Setze $TR := TR^*$
2. Für alle $i := 1, \ldots, n$:
   (a) Setze $TR' := (TR_i, TR_2, \ldots, TR_1, \ldots, TR_n)^T$, d. h. vertausche den ersten mit dem i-ten Zeilenvektor von TR.
   (b) Wende die Spaltenpermutation auf TR' an, die den ersten Zeilenvektor von TR' in $(1, 2, 3, \ldots, m)$ überführt und speichere das Ergebnis in TR''.
   (c) Wende die Zeilenpermutation auf TR'' an, die TR'' in die reduzierte Form TR''' überführt, wobei die erste Zeile von TR'' fix bleibt.
   (d) Wenn $TR''' <_{lex} TR^*$ gilt:
   TR^* ist nicht das lexikographische Minimum, Stop!
3. TR^* ist das lexikographische Minimum, gib TR^* aus, Stop!

Algorithmus 5.3.3 kann neben der Entscheidung über das lexikographische Minimum bezüglich Isomorphie auch zur entsprechenden Entscheidung bezüglich Struktur-Isomorphie benutzt werden. Zu diesem Zweck muß jeweils vor dem Abschluß (Schritt 3) die Beziehung $TR := \overline{TR^*}$ gesetzt und wieder zum Anfang von Schritt 2 gesprungen werden. Da die Umkehrtechnologie $\overline{TR}$ einer $n \times m$–Technologie in $O(nm)$ Zeit berechnet werden kann, beeinträchtigt diese Komplexität nicht den gesamten Zeitaufwand für den Test auf Minimalität bezüglich $<_{lex}$:

Satz 5.3.4 *Mit Hilfe von Algorithmus 5.3.3 kann in der Zeit $O(n^2 m \log n)$ festgestellt werden, ob es sich bei einer gegebenen $n \times m$–Technologie TR um das lexikographische Minimum ihrer Isomorphieklasse handelt.*

Beweis: Die Korrektheit von Algorithmus 5.3.3 ist klar. Da bei der Implementation des Algorithmus zur Realisierung der Spaltenpermutation in Schritt 2b nur Zeiger umgesetzt werden müssen, genügt dafür $O(m)$ Zeit. Die Sortierung der Zeilen in Schritt 2c, durch die wieder eine reduzierte Technologie erzeugt wird, benötigt $O(nm \log n)$ Zeit. Der lexikographische Vergleich (Schritt 2d) kann in der Zeit $O(nm)$ ausgeführt werden. Da man schließlich alle Operationen in Schritt 2 maximal n-mal durchführen muß, folgt als gesamte Zeitkomplexität die Schranke $O(n^2 m \log n)$. □

Mit Hilfe der *Lexikographischen Technologie-Erzeugung* und dem *Technologie-Minimalitätstest* ist es möglich, die theoretisch erzielten Ergebnisse über die Anzahl der Isomorphie- und Struktur-Isomorphieklassen aus Abschnitt 5.2 zu verifizieren. Tatsächlich stammen die Werte aus Tabelle 5.2 und Tabelle 5.3 ursprünglich aus Berechnungen mit Hilfe der gerade beschriebenen Algorithmen. Dabei können die gewünschten Anzahlen z. B. für $n \leq 5$ und $m = 6$ auf einer schnellen Workstation innerhalb von wenigen Sekunden berechnet und ausgegeben werden.

Modifiziertes Einfügeverfahren

Bei dem von Bräsel und M. Kleinau [13, 14] entwickelten Einfügeverfahren zur Enumeration aller $n \times m$–Pläne werden für feste Werte n und m die Operationen o_{ij} sukzessive in partielle Ablaufgraphen bzw. sogenannte Teilpläne eingefügt.

Definition 5.3.5 Ein *Teilplan* $A = (a_{ij})$ ist ein Plan, in dem einige Zellen leer sind, und bei dem zu jedem vorhandenen Eintrag $a_{ij} > 1$ der Wert $a_{ij} - 1$ als Eintrag in der Zeile i oder Spalte j existiert.

In unvollständig besetzten Matrizen kennzeichnen werden die leeren Zellen ohne Einträge stets mit „$\cdot$“ gekennzeichnet.

Beispiel 5.3.6 Es seien die Matrizen

$$A = \begin{pmatrix} 3 & \cdot & 2 \\ \cdot & 2 & 1 \\ 4 & \cdot & 3 \end{pmatrix} \qquad \text{und} \qquad B = \begin{pmatrix} \cdot & 5 & 3 \\ 3 & \cdot & 2 \\ 2 & 1 & \cdot \end{pmatrix}$$

gegeben. Die Matrix A ist ein Teilplan, B ist es nicht, da für den Eintrag $b_{1,2} = 5$ kein Eintrag $b_{1j} = 4$ oder $b_{i2} = 4$ in der ersten Zeile oder zweiten Spalte existiert.

Das anschließend vorgestellte modifizierte Einfügeverfahren baut auf der im vorangegangenen Abschnitt beschriebenen Konstruktion nicht-isomorpher Technologien TR auf. Das heißt, die technologischen Reihenfolgen sind bereits vorgegeben und

dazu werden jeweils vollständige Pläne erzeugt, indem die Operationen o_{ij} so in die partiellen organisatorischen Reihenfolgen eingefügt werden, daß keine Zyklen entstehen. Der Ausdruck $TR(A)$ bezeichne im folgenden stets die Technologie eines Plans A, und in der Variablen $Aut(A)$ wird im Laufe des Algorithmus die Anzahl der Automorphismen des Plans A gespeichert.

Algorithmus 5.3.7 *Plan-Enumeration*

Eingabe: Parameter n und m.
Ausgabe: Vertretersystem für die Isomorphieklassen der $n \times m$–Pläne, Gesamtanzahl $P_{n,m}$ aller $n \times m$–Pläne.

1. Setze $P_{n,m} := 0$.
2. Erzeuge ein Vertretersystem S^* für die Isomorphieklassen der $n \times m$–Technologien TR auf der Grundlage der lexikographischen Minima mittels Kombination aus Algorithmus 5.3.1 und 5.3.3.
3. Für alle Vertreter TR aus S^*, die in Schritt 2 erzeugt wurden:
   (a) Initialisiere eine $n \times m$–Matrix A, die ausschließlich leere Zellen enthält.
   (b) Für alle Operationen $o_{ij},\ i := 1, \ldots, n,\ j := 1, \ldots, m$:
      i. Aktualisiere die Matrix A anhand folgender Prozedur: Füge die Operation o_{ij} gemäß der Technologie TR in die Organisation der Maschine M_j als direkten Nachfolger einer in dieser Organisation bereits existierenden Operation ein, oder falls es noch keine derartige Operation gibt, füge o_{ij} als Quelle der Organisation von M_j ein.
      ii. Prüfe die Zulässigkeit von A, d. h. teste, ob A ein $n \times m$–Teilplan ist.
4. Für alle (vollständigen) Pläne A, die in Schritt 3 rekursiv erzeugt werden:
   (a) Setze $Aut(A) := 0$.
   (b) Für alle nichttrivialen Automorphismen (ϱ, σ) von $TR(A)$:
      i. Vergleiche A lexikographisch mit dem $n \times m$–Plan A', der gemäß (ϱ, σ) zu A isomorph ist (vgl. Algorithmus 5.1.4).
      ii. Wenn (ϱ, σ) ein Automorphismus von A ist:
         Setze $Aut(A) := Aut(A) + 1$.
   (c) Wenn für alle Pläne A' aus Schritt 4b die Beziehung $A \leq_{lex} A'$ gilt:
      i. Gib den Plan A aus.
      ii. Setze $P_{n,m} := P_{n,m} + (n!m!)/Aut(A)$.

5. Gib $P_{n,m}$ aus.

Es folgen einige kurze Erläuterungen zu einzelnen Teilen dieser *Plan-Enumeration*. Weitere Details zur Implementation sind in [11] beschrieben.

Die Einfügungen der Operationen in Schritt 3b wird in einer festgelegten Reihenfolge durchgeführt, die in bestimmten Fällen eventuell noch optimiert werden kann. Der gesamte Schritt 3 erfolgt durch Backtracking, um in systematischer Weise alle möglichen $n \times m$–Pläne erzeugen zu können, die zur gegebenen Technologie TR gehören. Bei jeder Aktualisierung der Matrix A wird versucht, die Ränge der Operationen zu ermitteln (topologisches Sortieren im assoziierten Digraphen). Wenn keine topologische Sortierung existiert, handelt es sich nicht um einen $n \times m$–Teilplan, d. h. die aktuelle Matrix A ist nicht zulässig, und die Weiterführung der Einfügungen von Operationen in A wird abgebrochen.

Das bei der *Plan-Enumeration* erzeugte Vertretersystem für die Isomorphieklassen der $n \times m$–Pläne basiert nicht auf der Auswahlfunktion (5.1), die jeweils das lexikographische Minimum als Repräsentant einer Isomorphieklasse auswählt. Da man die Informationen ausnutzen will, die bereits aus der lexikographischen Technologie-Erzeugung und dem Minimalitätstest in Schritt 2 stammen, wird für das zu erzeugende Vertretersystem der $n \times m$–Pläne die folgende, bezüglich (5.1) leicht modifizierte Auswahlfunktion verwendet:

Definition 5.3.8 Für zwei $n \times m$–Pläne A und B gelte $A \ll_{lex} B$, genau dann wenn $TR(A) <_{lex} TR(B)$ ist oder $TR(A) =_{lex} TR(B)$ und $A <_{lex} B$ gilt. Es sei $\mathcal{P}_{n,m}$ die Menge aller $n \times m$–Pläne, und $\mathcal{P}_1, \ldots, \mathcal{P}_r$ seien ihre Isomorphieklassen, also gilt $\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_r = \mathcal{P}_{n,m}$ und $\mathcal{P}_i \cap \mathcal{P}_j = \varnothing$ für $i \neq j$. Weiterhin sei $\min_{\ll}(\mathcal{P}_i)$ das Minimum der Pläne aus $\mathcal{P}_i$ bezüglich $\ll$. Dann ist die Funktion f mit

$$\begin{aligned} f : \{1, \ldots, r\} &\rightarrow \mathcal{P}_{n,m}, \\ i &\mapsto f(i) = \min_{\ll}(\mathcal{P}_i) \end{aligned} \tag{5.9}$$

die Auswahlfunktion des Vertretersystems der $n \times m$–Pläne, die bei der *Plan-Enumeration* zugrunde gelegt wird.

In Schritt 3 von Algorithmus 5.3.7 werden ausschließlich solche Pläne erzeugt, die für zwei Aufträge mit gleicher technologischer Reihenfolge bereits lexikographisch sortiert sind. Das heißt, wenn in einem Plan zwei Aufträge J_i und J_k mit $i < k$ die gleiche technologische Reihenfolge besitzen, wird die Operation o_{i1} vor o_{k1} ausgeführt. Auf diese Weise müssen in Schritt 4 von Algorithmus aufgrund der zugrunde gelegten Auswahlfunktion (5.9) nur dann lexikographische Vergleiche für einen Plan A ausgeführt werden, wenn $TR(A)$ nichttriviale Automorphismen besitzt.

Die Variable $Aut(A)$ in Schritt 4 dient zur Berechnung der Anzahl der Automorphismen von A. Der Ausdruck $P_{n,m} := P_{n,m} + (n!m!)/Aut(A)$ in Schritt 4(c)ii läßt sich wiederum gruppentheoretisch deuten (vgl. Seite 60): Die Zeilen- und Spaltenpermutationen (ϱ, σ) werden als Elemente der Gruppe $S_n \times S_m$ aufgefaßt, die auf der Menge der $n \times m$–Pläne operiert. Die Bahn eines Plans A entspricht dann der Isomorphieklasse, der A angehört, und der Stabilisator des Plans A ist die Menge der Automorphismen von A. Es ist bekannt, daß die Anzahl der Elemente der Isomorphieklasse, der A angehört, gleich dem Verhältnis von der Anzahl der Gruppenelemente zur Anzahl der Automorphismen von A ist. Das heißt, die wiederholte Ausführung von Schritt 4(c)ii bei der *Plan-Enumeration* bedeutet, daß für alle Isomorphieklassen die Anzahl der zugehörigen Pläne jeweils zur Variablen $\mathcal{P}_{n,m}$ addiert wird. Auf diese Weise wird am Ende der Prozedur neben der Anzahl der Isomorphieklassen auch die Gesamtanzahl aller $n \times m$–Pläne berechnet, obwohl nicht alle Pläne tatsächlich erzeugt wurden.

Durch Anpassung der lexikographischen Tests in Schritt 4 ist es ebenso möglich, neben der Anzahl der Isomorphieklassen auch die Anzahl der Plan-Klassen bezüglich Äquivalenz und Struktur-Äquivalenz mittels Erzeugung entsprechender Vertretersysteme zu bestimmen. Eine Übersicht über die anhand dieses Algorithmus erzielten Werte gibt der folgende Abschnitt.

5.4 Numerische Auswertungen

In diesem Abschnitt werden die Resultate für die verschiedenen Anzahlen der $n \times m$–Pläne zusammengestellt, verglichen und kommentiert. Im folgenden sei stets $P_{n,m}$ die Gesamtanzahl aller $n \times m$–Pläne, $I_{n,m}$ die Anzahl der nicht-isomorphen, $A_{n,m}$ die Anzahl der nicht-äquivalenten und $S_{n,m}$ die Anzahl der nicht-struktur-äquivalenten $n \times m$–Pläne.

In Tabelle 5.4 fällt auf, daß für $n \neq m$ stets $I_{n,m} = A_{n,m}$ gilt, was auf die Gleichbedeutung der Begriffe Isomorphie und Äquivalenz im nicht-quadratischen Fall hinweist. Weiterhin gilt $A_{n,m}/S_{n,m} \leq 2$, da die Anzahl der nicht-äquivalenten Pläne höchstens um die Hälfte reduziert werden kann, wenn zu jedem Plan A sein Umkehrplan $\overline{A}$ als strukturell gleichwertig eingestuft wird. Da allerdings die Wahrscheinlichkeit, daß ein zufällig ausgewählter Plan A zu $\overline{A}$ äquivalent ist, mit wachsenden n und m abnimmt, kann man folgendes feststellen.

Beobachtung 5.4.1 *Für* $(n + m) \to \infty$ *gilt*

$$\frac{A_{n,m}}{S_{n,m}} = 2 - o(1). \tag{5.10}$$

n	m	$P_{n,m}$	$I_{n,m}$	$A_{n,m}$	$S_{n,m}$
2	2	14	4	3	3
2	3	204	17	17	12
3	3	19 164	533	280	147
2	4	5 016	106	106	68
3	4	3 733 056	25 924	25 924	13 100
4	4	6 941 592 576	12 051 574	6 028 059	3 017 369
2	5	185 520	773	773	422
3	5	1 288 391 040	1 789 432	1 789 432	895 388
4	5	26 549 943 275 520	9 218 730 304	9 218 730 304	4 609 489 912
2	6	9 595 440	6 671	6 671	3 495
3	6	712 770 186 240	164 993 112	164 993 112	82 507 654
2	7	659 846 880	65 461	65 461	33 193
3	7	589 563 294 888 960	19 496 140 704	19 496 140 704	9 748 141 078

Tabelle 5.4: Anzahlen der $n \times m$–Pläne im Vergleich.

$n \setminus m$	2	3	4	5	6	7
2	21.4286	5.9113	1.3557	0.2275	0.0364	0.0050
3		0.7671	0.3509	0.0695	0.0116	0.0017
4			0.0435	0.0174	??	??

Tabelle 5.5: Verhältnisse der Anzahlen nicht-struktur-äquivalenter $n \times m$–Pläne zu den jeweiligen Gesamtanzahlen (in %).

Diese Beobachtung wird durch die Werte in Tabelle 5.4 bestätigt.

Ziel der Einführung der Struktur-Äquivalenz ist es, die Plan-Untersuchungen, also die Untersuchungen von Lösungen für Shop-Scheduling-Probleme, auf solche Pläne zu beschränken, die sich in ihrer Grundstruktur bezüglich der Wege in den zugehörigen Ablaufgraphen unterscheiden. Tabelle 5.5 zeigt, daß die Anzahl nicht-struktur-äquivalenter $n \times m$–Pläne deutlich kleiner als die Gesamtanzahl der Pläne des gleichen Formats ist, d. h. es ergibt sich damit wie gewünscht eine erhebliche Reduzierung der zu betrachtenden Pläne.

Die zur Enumeration benötigten Algorithmen wurden in C++ implementiert und auf einem Pentium-PC-133-Mhz-Rechner getestet. Das gesamte Programm ist so angelegt, daß für ein gegebenes Format $n \times m$ in *einem* Durchlauf *alle* Zahlen, also die Gesamtanzahl $P_{n,m}$ sowie die Anzahlen $I_{n,m}, A_{n,m}$ und $S_{n,m}$ für die drei Plan-Äquivalenzklassen bestimmt werden. Die dazu benötigte CPU-Zeit zur

n	m	CPU-Zeit in Sek.	# versuchter Komplettierungen	# erzeugter $n \times m$–Pläne	$P_{n,m}$
2	2	0.01	6	4	14
2	3	0.01	33	15	204
3	3	0.04	892	384	19 164
2	4	0.01	226	84	5 016
3	4	0.66	38 704	15 638	3 733 056
4	4	838.15	14 184 294	6 872 356	6 941 592 576
2	5	0.03	1 546	491	185 520
3	5	35.73	2 430 856	923 073	1 288 391 040
4	5	1118.57	10 247 426 194	4 656 870 459	26 549 943 275 520
2	6	0.23	13 908	3 888	9 595 440
3	6	3617.67	235 692 637	83 310 542	712 770 186 240
2	7	2.08	137 532	34 709	659 846 880
3	7	3537.27	29 593 003 309	9 756 803 163	589 563 294 888 960

Tabelle 5.6: Statistische Werte für die Berechnung der Plan-Anzahlen.

vollständigen Enumeration ist in Tabelle 5.6 enthalten. Weiterhin zeigt diese Tabelle die Anzahl der versuchten Komplettierungen von Teilplänen gemäß Schritt 3 in Algorithmus 5.3.7, sowie die Anzahl der darunter erfolgreichen Komplettierungen, welche der Anzahl der im Algorithmus tatsächlich erzeugten Pläne entspricht. Zu Vergleichszwecken ist nochmals zusätzlich die Gesamtanzahl $P_{n,m}$ aufgeführt.

Während die benötigte CPU-Zeit für kleinere Formate noch unerheblich ist, wächst mit steigenden Werten n und m enorm. Mit dem vorliegenden Programm ist die Berechnung der Anzahlen für das Format 5×5 in vernünftiger Zeit nicht mehr möglich. Die Anzahlbestimmung für die quadratischen Formate ($n = m$) ist allerdings auch besonders aufwendig, da bei den Tests auf Äquivalenz- und Struktur-Äquivalenz jeweils zusätzlich die transponierten Pläne betrachtet werden müssen.

5.5 Komplexität der Plan-Enumeration

Die von Cook [27] und Karp [51] eingeführte Komplexitätstheorie, die häufig zur Einschätzung der Schwierigkeit von kombinatorischen Optimierungsproblemen dient, wurde in Arbeiten von Valiant [99, 100] für Enumerationsprobleme erweitert. Viele der bekannten Entscheidungs- bzw. Optimierungsprobleme sind mit solche Enumerationsproblemen auf natürliche Weise verbunden. Zum Beispiel gehört

zum Problem der Bestimmung eines Hamiltonschen Kreises[10] in einem gegebenen Graphen G das Enumerationsproblem „Wieviele solcher Hamiltonschen Kreise gibt es für G?“

Neben der bereits in Abschnitt 2.3 erwähnten Klasse $\mathcal{P}$ von Entscheidungsproblemen gibt es im Bereich der Anzahlproblematik die Komplexitätsklasse $\#\mathcal{P}$, die aus allen nichtdeterministisch-polynomial-lösbaren Enumerationsproblemen besteht. Die Klasse $\#\mathcal{P}$ wurde von VALIANT definiert, um die zusätzliche Schwierigkeit bei der Enumeration widerzuspiegeln zu können. Analog zur Terminologie der $\mathcal{NP}$-vollständigen Probleme sind die *$\#\mathcal{P}$-vollständigen* Enumerationsprobleme (gesprochen: Anzahl-$\mathcal{P}$-vollständigen Probleme) die schwierigsten Probleme in der Komplexitätsklasse $\#\mathcal{P}$. Genauer gesagt, wird ein Enumerationsproblem Π als *$\#\mathcal{P}$-vollständig* bezeichnet, wenn $\Pi \in \#\mathcal{P}$ ist und alle Probleme $\Pi' \in \#\mathcal{P}$ auf Π polynomial reduzierbar sind (siehe GAREY und JOHNSON [36]).

Die Enumerationsprobleme, die zu $\mathcal{NP}$-vollständigen Entscheidungsproblemen gehören, sind offensichtlich $\mathcal{NP}$-schwer. Allgemein gilt, daß die Enumerationsprobleme mindestens so schwierig sind wie die zugehörigen Entscheidungsprobleme. Es gibt Enumerationsprobleme, die $\#\mathcal{P}$-vollständig sind, obwohl das zugrundeliegende Entscheidungsproblem polynomial lösbar ist. Beispielsweise ist das Problem „Gibt es in einem gegebenen bipartiten Graphen $G = (V, E)$ ein perfektes Matching?“ in der Zeit $O(|V|^{5/2})$ lösbar (siehe HOPCROFT und KARP [45]), während für das zugehörige Enumerationsproblem „Wieviele perfekte Matchings gibt es in einem bipartiten Graphen G?“ von VALIANT [99] die $\#\mathcal{P}$-Vollständigkeit nachgewiesen wurde.

In Abschnitt 4.3 zeigen Satz 4.3.2 und 4.3.3 bereits, daß das Problem der Enumeration aller $n \times m$–Pläne bzw. der Enumeration aller azyklischen Orientierungen des Hamming-Graphen $K_n \times K_m$ mit dem der Bestimmung des chromatischen Polynoms von $K_n \times K_m$ komplexitätstheoretisch korrespondiert. Satz 4.3.2 von STANLEY [92] stammt aus dem Jahr 1973. In [62] hat LINIAL 1986 gezeigt, daß das Problem der Enumeration der azyklischen Orientierungen eines Graphen G zu den schwierigsten Problemen in der Klasse $\#\mathcal{P}$ gehört:

Satz 5.5.1 [62] *Das Problem der Enumeration der azyklischen Orientierungen eines Graphen G ist $\#\mathcal{P}$-vollständig.*

Beweis: Die *Verbindung (join)* zweier Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ mit $V_1 \cap V_2 = \varnothing$ ist definiert als

$$G_1 + G_2 = (V_1 \cup V_2 \ , \ E_1 \cup E_2 \cup \{\{v_1, v_2\} : v_1 \in V_1, v_2 \in V_2\}). \tag{5.11}$$

[10] Ein *Hamiltonscher Kreis* eines Graphen G ist ein Kreis in G, der alle Knoten von G durchläuft.

Es sei $G = (V, E)$ ein Graph mit $|V| = p$. Für das chromatische Polynom der Verbindung $G + K_n$ gilt offensichtlich

$$\chi(G + K_n, \lambda) = \lambda(\lambda - 1) \cdots (\lambda - n + 1)\chi(G, \lambda - n). \tag{5.12}$$

Nach Satz 4.3.2 ist das Problem der Enumeration der azyklischen Orientierungen eines beliebigen Graphen G äquivalent zur Bestimmung des chromatischen Polynoms von G an der Stelle $\lambda = -1$. Das Problem der Bestimmung von $\chi(G, -1)$ läßt sich polynomial auf das Problem der Berechnung von $\chi(G, \lambda)$ transformieren:

Mit $\chi(G, -1)$ haben wir auch $\chi(G + K_n, -1)$ für $n = 1, \ldots, p$. Also kann mittels (5.12) der Ausdruck $\chi(G, -j)$ für $2 \leq j \leq p + 1$ berechnet werden. Damit ist auch das chromatische Polynom von G bestimmt, denn $\chi(G, \lambda)$ ist ein Polynom in λ vom Grad p.

Das Entscheidungsproblem „Hat ein Graph G eine zulässige λ-Färbung?“ ist für $\lambda \geq 3$ $\mathcal{NP}$-vollständig, siehe [36]. Daher ist das assoziierte Enumerationsproblem $\#\mathcal{P}$-vollständig und somit auch das Problem der Bestimmung des chromatischen Polynoms von G. □

Dieser Satz zeigt die Schwierigkeit der Bestimmung der Anzahl azyklischer Orientierungen eines beliebigen Graphen G. Im Zusammenhang mit der Enumeration von Plänen ist man allerdings ausschließlich an der Anzahl der azyklischen Orientierungen von Hamming-Graphen $K_n \times K_m$ interessiert.

Innerhalb einiger Graphenklassen kann das chromatische Polynom auf einfache Weise bestimmt werden. Bei diesen Graphen G existiert wegen Satz 4.3.2 daher auch ein effektiver und exakter Ausdruck für $\alpha(G)$. Es stellt sich nun also die Frage, ob die Bestimmung des chromatischen Polynoms von Hamming-Graphen $K_n \times K_m$ in polynomialer Zeit möglich ist. In diesem Zusammenhang ist ein kürzlich erschienenes Resultat von Rezaie [79] interessant:

Satz 5.5.2 [79] *Es sei P_n ein Weg mit n Knoten. Für alle $n, m \in \mathbb{N}$ gilt*

$$\chi(P_n \times K_m, \lambda) = \chi(K_m, \lambda)^n \left(\sum_{i=0}^{m} \frac{(-1)^i \binom{m}{i}}{\chi(K_i, \lambda)} \right)^{n-1}, \tag{5.13}$$

wobei $\chi(K_m, \lambda) = \lambda(\lambda - 1) \cdots (\lambda - m + 1)$ ist.

Zur rekursiven Berechnung von chromatischen Polynomen wird sehr häufig die folgende, bereits im Beweis zu Satz 4.3.2 verwandte Eigenschaft benutzt.

Hilfssatz 5.5.3 [77] *Es sei e eine Kante eines Graphen G, dann gilt*

$$\chi(G, \lambda) = \chi(G \backslash e, \lambda) - \chi(G/e, \lambda), \tag{5.14}$$

wobei $G\backslash e$ bzw. G/e der Graph ist, der aus G durch Löschen bzw. Kontraktion der Kante e entsteht.

Die Prozedur in [79] zur Bestimmung von $\chi(P_n \times K_m, \lambda)$ macht in jedem Schritt von der Existenz eines Knotens vom Grad 1 Gebrauch, um mit Hilfe der Anwendung von Eigenschaft (5.14) eine rekursive Beziehung herleiten zu können. Auch für allgemeine Bäume T_n mit n Knoten führt dieselbe Vorgehensweise zum Erfolg. Also erhält man für $\chi(T_n \times K_m, \lambda)$ ebenfalls die Formel (5.13).

Es stellt sich heraus, daß sich die rekursive Prozedur in [79] nicht auf Hamming-Graphen $K_n \times K_m$ übertragen läßt, denn in diesem Fall fehlen die in den Wegen P_n bzw. Bäumen T_n enthaltenen Knoten vom Grad 1. Das Problem der Bestimmung von $\chi(K_n \times K_m, \lambda)$ ist bis heute ungelöst. Weiterhin sind bisher weder geschlossene Formeln für $\chi(C_n \times K_m, \lambda)$ noch für $\chi(P_n \times P_m, \lambda)$ bekannt (siehe [23]).

Kapitel 6

Optimalitätskriterien für Pläne

Zu jedem Plan eines Shop-Scheduling-Problems kann der eindeutig zugeordnete semiaktive Schedule $C = (c_{ij})$ mit dem Zeitaufwand $O(nm)$ berechnet werden (siehe Seite 23). Daher ist die Bestimmung des semiaktiven Schedules zu einem gegebenen Plan ein polynomial lösbares Problem. Eine Vielzahl der Shop-Scheduling-Probleme ist jedoch $\mathcal{NP}$-schwer. Die Schwierigkeit dieser Probleme liegt also bereits in der Konstruktion eines optimalen Plans.

In diesem Kapitel werden verschiedene Kriterien behandelt, mit denen „ungünstige“ Pläne bei der Suche nach einem optimalen Plan ausgeschlossen werden können. Die hier vorgestellten Kriterien sind unabhängig bzw. nur bedingt abhängig von den gegebenen Bearbeitungszeiten p_{ij} für die Operationen o_{ij} des gegebenen Shop-Scheduling-Problems.

6.1 Potentiell-optimale Pläne

Bei jedem Job-Shop-Problem ist eine Technologie bereits fest vorgegeben. Zur Lösung des Problems muß eine günstige Organisation gefunden werden. In [2] hat ASHOUR für ein Job-Shop-Problem unter allen Organisationen erstmals zwischen zulässigen, potentiell-optimalen und optimalen Organisationen unterschieden. Bei der in [2] verwandten Terminologie für Job-Shop-Probleme wird im Gegensatz zur vorliegenden Arbeit (vgl. Definition 2.2.1 und 2.2.2) eine Organisation als „sequence“ bezeichnet.

Definition 6.1.1 Für ein gegebenes Shop-Scheduling-Problem sei $\mathcal{S}^*$ eine echte Teilmenge der Menge aller zulässigen Lösungen mit der Eigenschaft, daß $\mathcal{S}^*$ für jede beliebige Bearbeitungszeit-Matrix $P = (p_{ij})$ mindestens eine optimale Lösung enthält. Dann heißt $\mathcal{S}^*$ eine *potentiell-optimale* Menge, und die Elemente von $\mathcal{S}^*$ heißen *potentiell-optimale* Lösungen.

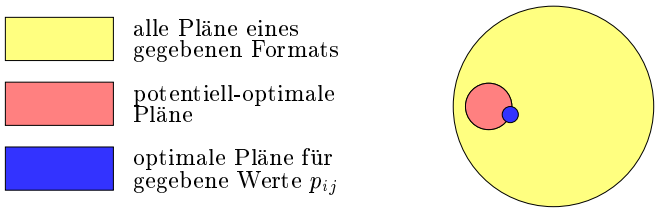


Abbildung 6.1: Die potentiell-optimalen Elemente im Mengensystem der Pläne.

Während die Eigenschaft eines Plans, potentiell-optimal zu sein, unabhängig von der Wahl der Bearbeitungszeiten p_{ij} ist, hängt eine optimale Lösung von den Werten p_{ij} sowie von der gegebenen Zielfunktion ab.

Im folgenden ist zunächst das Open-Shop-Problem Ausgangspunkt für die Strukturuntersuchung der zugehörigen Lösungen. Zur Lösung eines Open-Shop-Problems ist neben der Organisation auch eine Technologie zu bestimmen. Solche Sequenzen werden anhand von Plänen, also durch bestimmte lateinische Rechtecke, modelliert. Eine allgemeine Klassifikation der Pläne hinsichtlich Optimalität ist Abbildung 6.1 zu entnehmen. Ziel von Kapitel 6 und 7 ist es, geeignete Charakterisierungen für eine potentiell-optimale Menge von Plänen zu entwickeln, um „günstige“ Pläne effizient enumerieren zu können.

Für das Problem $J|n = 2|C_{\max}$ haben Akers und Friedman in [1] eine eindeutige Charakterisierung einer potentiell-optimalen Menge von Sequenzen anhand von sogenannten freien Maschinen angegeben. In einer Sequenz für $J|n = 2|C_{\max}$ wird eine Maschine als *freie Maschine* bezeichnet, wenn auf ihr die beiden Aufträge direkt aufeinanderfolgend bearbeitet werden und zur gleichen Zeit auf den anderen Maschinen keine Bearbeitungen stattfinden. In [1] wird gezeigt, daß alle Sequenzen ohne freie Maschinen zusammen für $J|n = 2|C_{\max}$ eine Menge potentiell-optimaler Sequenzen bilden.

Ähnliche Ergebnisse für Flow-Shop-Probleme des Typs $F||C_{\max}$ haben Conway, Maxwell und Miller in [26] erzielt. Es zeigt sich, daß diejenigen Sequenzen eine Menge potentiell-optimaler Sequenzen bilden, bei denen die Aufträge auf den Maschinen M_1 und M_2 in der gleichen organisatorische Reihenfolge stehen, und ebenfalls die Maschinen M_{m-1} und M_m eine gleiche organisatorische Reihenfolge besitzen. In [26] werden daraus auch Resultate für andere reguläre Zielfunktionen abgeleitet.

Eine Verallgemeinerung dieser Ergebnisse für Open-Shop-Probleme mit beliebiger Auftrags- und Maschinenanzahl gestaltet sich schwierig. Eine mögliche allgemeine Charakterisierung einer potentiell-optimalen Menge von Sequenzen ist durch das anschließend vorgestellte und erstmals von M. Kleinau in [55] eingeführte Konzept der *Irreduzibilität* von Plänen gegeben (siehe auch [15]). Im folgenden wird

dieses Konzept erweitert, ergänzt und präzisiert. Weiterhin werden mehrere Begriffe an die gängige Terminologie der Graphentheorie angepaßt, insbesondere im Fall sogenannter Vergleichbarkeitsgraphen.

6.2 Konzept der Irreduzibilität

Auf der Menge aller $n \times m$–Pläne ist eine spezielle Halbordnung definiert, deren minimale Elemente eine Menge potentiell-optimaler Pläne bilden. Zur Beschreibung dieser Halbordnung sind folgende Vorbetrachtungen notwendig.

Es sei A ein $n \times m$–Plan und $G(A) = (V, E)$ der zugehörige Ablaufgraph. Im folgenden handelt es sich bei den betrachteten Wegen in $G(A)$ stets um gerichtete Wege. Die hier dargestellten Aussagen über Pläne gelten für alle Shop-Scheduling-Probleme. Für einen Weg $W = (v_0, v_1, \ldots, v_k)$ in $G(A)$ sei $V_W = \{v_0, v_1, \ldots, v_k\}$ die Menge seiner Knoten. Weiterhin sei $\mathcal{W}_A(v_k)$ die Menge aller Wege in $G(A)$, die im Knoten $v_k \in V$ enden. Aufgrund der Definition des Ablaufgraphen $G(A) = (V, E)$ ist leicht zu sehen, daß sich die Fertigstellungszeit $c_{ij}(A)$ einer Operation o_{ij} im Plan A anhand von

$$c_{ij}(A) = \max_{W \in \mathcal{W}_A(o_{ij})} \left\{ \sum_{o_{kl} \in V_W} p_{kl} \right\} \tag{6.1}$$

berechnen läßt, wobei die p_{kl} die Bearbeitungszeiten der Operationen o_{kl} angeben.

Definition 6.2.1 Ein Weg $W \in \mathcal{W}_A(v_k)$ in $G(A)$ heißt *dominant*, wenn kein anderer Weg $W' \in \mathcal{W}_A(v_k)$ mit $V_W \subset V_{W'}$ existiert.

Es sei $\mathcal{W}_A^*(v_k) \subseteq \mathcal{W}_A(v_k)$ die Menge aller dominanten Wege in $G(A)$, die im Knoten v_k enden. Da die Bearbeitungszeiten p_{ij} für alle $i = 1, \ldots, n$ und $j = 1, \ldots, m$ nicht negativ sind, genügt es, sich bei der Bestimmung der Fertigstellungszeiten $c_{ij}(A)$ auf dominante Wege zu beschränken, d. h. es folgt unmittelbar

$$c_{ij}(A) = \max_{W \in \mathcal{W}_A^*(o_{ij})} \left\{ \sum_{o_{kl} \in V_W} p_{kl} \right\}. \tag{6.2}$$

Wenn Gleichung (3.1) auf einen bestimmten Plan A bezogen wird, so wird für die zugehörige Gesamtbearbeitungszeit $C_{\max}(A)$ die Gleichung

$$C_{\max}(A) = \max_{\substack{1 \le i \le n \\ 1 \le j \le m}} \{c_{ij}(A)\}. \tag{6.3}$$

geschrieben.

Definition 6.2.2 Ein Plan B heißt *reduzierbar* auf einen Plan A, geschrieben $A \preceq B$, wenn für jeden dominanten Weg W_a in $G(A)$ ein dominanter Weg W_b in $G(B)$ mit $V_{W_a} \subseteq V_{W_b}$ existiert. Zwei Pläne A und B heißen *ähnlich*, symbolisiert durch $A \sim B$, wenn $A \preceq B$ und $B \preceq A$ gilt. Ein Plan B wird *streng reduzierbar* auf einen Plan A genannt, wenn $A \preceq B$ und $A \not\sim B$ gilt. In diesem Fall wird $A \prec B$ geschrieben.

Beispiel 6.2.3 Es werden die beiden 4×4–Pläne

$$A = \begin{pmatrix} 1 & 6 & 7 & 8 \\ 2 & 5 & 6 & 7 \\ 3 & 4 & 5 & 1 \\ 4 & 7 & 1 & 2 \end{pmatrix} \quad \text{und} \quad A' = \begin{pmatrix} 2 & 6 & 7 & 8 \\ 1 & 5 & 6 & 7 \\ 3 & 4 & 5 & 1 \\ 4 & 7 & 1 & 2 \end{pmatrix}$$

betrachtet. Die Pläne $A = (a_{ij})$ und $A' = (a'_{ij})$ unterscheiden sich nur in den Einträgen a_{11}, a_{21} bzw. a'_{11}, a'_{21}. Daher sind in den entsprechenden Ablaufgraphen $G(A)$ und $G(A')$ nur diejenigen dominanten Wege verschieden, die in a_{11} bzw. a'_{21} starten. Diese dominanten Wege besitzen jedoch jeweils die gleiche Menge von Knoten, also gilt $A \sim A'$.

Definition 6.2.4 Ein Plan B heißt *irreduzibel*, wenn es keinen Plan A mit $A \prec B$ gibt.

Die Relation '$\prec$' induziert eine Halbordnung auf der Menge aller Pläne eines gegebenen Formats $n \times m$. Die minimalen Elemente dieser Halbordnung sind gerade die irreduziblen Pläne. Man kann nachprüfen, daß die Pläne A und A' aus Beispiel 6.2.3 irreduzibel sind. Es existieren also ähnliche irreduzible Pläne.

Für einen Plan A ist der Plan, der sich durch Umkehrung aller seiner technologischen und organisatorischen Reihenfolgen ergibt, der Umkehrplan $\overline{A}$ von A. Offensichtlich ist stets $A \sim \overline{A}$, und es gelten für zwei Pläne A und B mit $A \preceq B$ die Beziehungen

$$\overline{A} \preceq B, \quad A \preceq \overline{B} \quad \text{und} \quad \overline{A} \preceq \overline{B}. \tag{6.4}$$

An dieser Stelle sei darauf hingewiesen, daß die Definition

$$A \prec B \;:=\; A \preceq B \;\wedge\; A \neq B \;\wedge\; A \neq \overline{B}$$

in [55] im Zusammenhang mit der Irreduzibilität nicht zweckmäßig ist, denn aus $A \sim B$ folgt nicht notwendig $A = B \;\vee\; A = \overline{B}$. Es existieren also ähnliche Pläne A, B mit $A \neq B$ und $A \neq \overline{B}$. Beispielsweise sind alle Pläne paarweise ähnlich, deren zugeordnete Ablaufgraphen einen Weg enthalten, der alle Operationen o_{ij}

($i = 1, \ldots, n$ und $j = 1, \ldots, m$) umfaßt. Weiterhin sind z. B. die beiden Pläne aus Beispiel 6.2.3 ähnlich. Aus diesem Grund wird stets

$$A \prec B := A \preceq B \ \wedge \ A \not\sim B, \tag{6.5}$$

gemäß Definition 6.2.2 gesetzt.

Offensichtlich handelt es sich bei der Ähnlichkeit („$\sim$“) von Plänen um eine Äquivalenzrelation. Zwischen der Ähnlichkeit und den Äquivalenzrelationen aus Abschnitt 5.1 (Isomorphie, Äquivalenz und Struktur-Äquivalenz) besteht ein grundlegender Unterschied: Die Ähnlichkeit zweier Pläne A und B beruht ausschließlich auf dem direkten Vergleich der Knotenmengen der gerichteten Wege in den zugehörigen Ablaufgraphen $G(A)$ und $G(B)$. Bei den anderen Äquivalenzrelationen gehören zwei Pläne einer Äquivalenzklasse an, wenn alle Wegstrukturen in den zugehörigen Ablaufgraphen bei entsprechender Vertauschung von Aufträgen und Maschinen bzw. Umkehrung der Orientierungen identisch bleiben.

Dieser Unterschied läßt sich anhand von Beispiel 6.2.3 darstellen: Die Pläne A und A' sind zwar ähnlich ($A \sim A'$), aber nicht-struktur-äquivalent ($A \not\equiv_S A'$). Das heißt, diese Pläne lassen sich durch Vertauschung von Zeilen und Spalten, durch Matrix-Transposition oder durch Umkehrung aller Reihenfolgen nicht ineinander überführen, obwohl sie in den entsprechenden Ablaufgraphen bezüglich der Mengen von Knoten der gerichteten Wege gleichartig sind.

Die Eigenschaft eines Plans, irreduzibel zu sein, ist offensichtlich invariant bezüglich Isomorphie, Äquivalenz und Struktur-Äquivalenz. Das bedeutet zum Beispiel: Wenn A und B zwei struktur-äquivalente Pläne sind ($A \equiv_S B$) und A irreduzibel ist, dann ist auch B irreduzibel. Dieser Zusammenhang wird später bei der Enumeration der irreduziblen Pläne ausgenutzt, bei der nur die nicht-struktur-äquivalenten Vertreter irreduzibler Pläne erzeugt werden.

Der anschließende Satz stellt die eigentlich Motivation für die Einführung des Konzepts der Irreduzibilität dar.

Satz 6.2.5 [15] *Es seien A und B zwei Pläne eines Open-Shop-Problems des Typs $O||C_{\max}$ mit n Aufträgen und m Maschinen. Weiterhin sei $A \preceq B$. Dann gilt*

$$C_{\max}(A) \leq C_{\max}(B). \tag{6.6}$$

Beweis: Die Aussage des Satzes folgt direkt aus der Definition von „$\preceq$“ zusammen mit den Beziehungen (6.2) und (6.3). □

Aus $A \prec B$ läßt sich nicht notwendig $C_{\max}(A) < C_{\max}(B)$ folgern. Nur wenn es im Fall gegebener Bearbeitungszeiten $p_{ij} > 0$ ($i = 1, \ldots, n$ und $j = 1, \ldots, m$) für

$A \prec B$ einen eindeutigen kritischen Weg[1] W_b in $G(B)$ gibt, für den kein dominanter Weg W_a in $G(A)$ mit $V_{W_a} = V_{W_b}$ existiert, gilt $C_{\max}(A) < C_{\max}(B)$.

Die anschließende Folgerung zeigt, daß es hilfreich ist, den Suchraum auf die Menge der irreduziblen Pläne einzuschränken, wenn man nach einem optimalen Plan für ein Open-Shop-Problem sucht.

Folgerung 6.2.6 *Es sei $\mathcal{P}^*_{n,m}$ die Menge aller irreduziblen $n \times m$–Pläne. Für jedes Open-Shop-Problem des Typs $O||C_{\max}$ mit n Aufträgen und m Maschinen enthält $\mathcal{P}^*_{n,m}$ einen optimalen Plan.*

Beweis: Es sei B ein optimaler Plan eines gegebenen Open-Shop-Problems. Falls kein Plan A mit $A \prec B$ existiert, ist B irreduzibel. Sei daher B streng reduzierbar. Dann existiert eine Menge $\{A_1, A_2, \ldots, A_k\}$ von Plänen mit $k \geq 1$ und $A_1 \prec A_2 \prec \cdots \prec A_k \prec B$, so daß A_1 irreduzibel ist. Wegen $A_1 \prec A_2 \prec \cdots \prec A_k \prec B$ ist auch $A_1 \preceq B$ und nach Satz 6.2.5 gilt dann $C_{\max}(A_1) \leq C_{\max}(B)$. Damit ist der irreduzible Plan A_1 ebenfalls optimal. □

Die Menge $\mathcal{P}^*_{n,m}$ der irreduziblen $n \times m$–Pläne ist eine Teilmenge von $\mathcal{P}_{n,m}$, der Menge aller $n \times m$–Pläne für $O||C_{\max}$ mit n Aufträgen und m Maschinen. Da sich in $\mathcal{P}^*_{n,m}$ jede beliebige Bearbeitungszeit-Matrix eine optimale Lösung befindet, ist $\mathcal{P}^*_{n,m}$ eine potentiell-optimale Menge von Plänen im Sinne von Definition 6.1.1.

Es stellt sich heraus, daß die Menge $\mathcal{P}^*_{n,m}$ keine minimale Menge von potentiell-optimalen Plänen ist: Die beiden irreduziblen Pläne A und A' aus Beispiel 6.2.3 sind ähnlich. Das heißt, immer wenn A für eine gegebene Menge von Bearbeitungszeiten p_{ij} optimal ist, so ist es auch A', und man kann sich bei der Suche nach einer minimalen Menge potentiell-optimaler Pläne stets auf einen der beiden Pläne beschränken.

Es existiert eine echte Teilmenge $\mathcal{M} \subset \mathcal{P}^*_{n,m}$, in der sich unabhängig von den gegebenen Bearbeitungszeiten stets ein optimaler Plan befindet. Eine minimale Menge $\mathcal{M}$ von potentiell-optimalen Plänen heißt *unvermeidbar*. Eine solche unvermeidbare Menge $\mathcal{M}$ von Plänen ist im Gegensatz zur Menge $\mathcal{P}^*_{n,m}$ jedoch im allgemeinen nicht eindeutig bestimmt (siehe [96, 97]).

6.3 Stabilität optimaler Pläne

Thema der ersten beiden Abschnitte dieses Kapitels war die Beschreibung potentiell-optimaler bzw. irreduzibler Pläne, deren potentielle Optimalität unabhängig von den gegebenen Bearbeitungszeiten p_{ij} ist.

[1] Ein *kritischer Weg* in einem Ablaufgraphen ist ein gerichteter Weg W, dessen Knotenmenge V_W ausschließlich aus Operationen o_{ij} besteht, die nicht später begonnen werden können, ohne die Gesamtbearbeitungszeit $C_{\max}$ zu verlängern (sogenannte kritische Operationen).

In diesem Abschnitt spielen bei der sogenannten Stabilität von Plänen die Bearbeitungszeiten p_{ij} eine größere Rolle. Im Falle einer gegebenen Bearbeitungszeit-Matrix $P = (p_{ij})$ wird für einen zugehörigen optimalen Plan A untersucht, für welche Abweichungen von p_{ij} der Plan A optimal bleibt.

Stabilitätsradius eines optimalen Plans

Arbeiten von SOTSKOV, STRUSEVICH und TANAEV [89, 94] dienen als Ausgangspunkt und Grundlage der Definitionen im Zusammenhang mit dem Stabilitätsradius optimaler Pläne. Zur Vereinfachung wird im folgenden für die Menge der Bearbeitungszeiten p_{ij} anstelle der Bearbeitungszeit-Matrix $P = (p_{ij})$ der Vektor $\boldsymbol{p} = (p_{11}, p_{12}, \ldots, p_{nm})$ geschrieben, bei dem die Elemente von P zeilenweise hintereinander stehen.

Eine optimale Lösung eines Shop-Scheduling-Problems ist im allgemeinen nicht eindeutig. Es sei $\mathcal{S}^*(\boldsymbol{p})$ die Menge aller Pläne, die bezüglich des Bearbeitungszeitvektors $\boldsymbol{p}$ optimal sind. Weiterhin sei $\mathbb{R}_+^{nm}$ der nm-dimensionale Raum nicht-negativer reeller Vektoren mit der *Maximum-* bzw. *Tschebyschev-Metrik*, d. h. zwischen zwei Vektoren $\boldsymbol{p}, \boldsymbol{p'} \in \mathbb{R}_+^{nm}$ mit

$$\boldsymbol{p} = (p_{11}, p_{12}, \ldots, p_{nm}) \text{ und } \boldsymbol{p'} = (p'_{11}, p'_{12}, \ldots, p'_{nm})$$

wird der *Abstand* durch

$$d(\boldsymbol{p}, \boldsymbol{p'}) := \max_{\substack{1 \le i \le n \\ 1 \le j \le m}} |p_{ij} - p'_{ij}|$$

definiert. Eine *abgeschlossene Kugel* mit dem *Mittelpunkt* $\boldsymbol{p}$ und dem *Radius* ε ist die Menge

$$K_\varepsilon(\boldsymbol{p}) := \{\boldsymbol{p'} \in \mathbb{R}_+^{nm} \mid d(\boldsymbol{p}, \boldsymbol{p'}) \le \varepsilon\}.$$

Eine abgeschlossene Kugel $K_\varepsilon(\boldsymbol{p})$ heißt *Stabilitätskugel* eines optimalen Plans $A \in \mathcal{S}^*(\boldsymbol{p})$, wenn der Plan A für jeden Vektor dieser Kugel optimal bleibt, d. h. wenn $A \in \mathcal{S}^*(\boldsymbol{p'})$ für alle $\boldsymbol{p'} \in K_\varepsilon(\boldsymbol{p})$ gilt. Der *Stabilitätsradius* $\varepsilon(A, \boldsymbol{p})$ eines optimalen Plans $A \in \mathcal{S}^*(\boldsymbol{p})$ ist der maximale Wert, den der Radius einer Stabilitätskugel von A um $\boldsymbol{p}$ annehmen kann, also

$$\varepsilon(A, \boldsymbol{p}) := \max\{r \in \mathbb{R}^+ \mid K_r(\boldsymbol{p}) \text{ ist Stabilitätskugel von } A \in \mathcal{S}^*(\boldsymbol{p})\}.$$

Ein optimaler Plan $A \in \mathcal{S}^*(\boldsymbol{p})$ heißt *stabil*, wenn $\varepsilon(A, \boldsymbol{p}) > 0$ gilt. Für die Lösung von Shop-Scheduling-Problemen sind optimale Pläne $A \in \mathcal{S}^*(\boldsymbol{p})$ mit einem möglichst großen Stabilitätsradius interessant, denn derartige Pläne bleiben auch bei relativ starken Abweichungen der Bearbeitungszeiten p_{ij} optimal im Gegensatz zu anderen Plänen mit vergleichsweise geringerem Stabilitätsradius. Es stellt sich nun die Frage, ob es stets stabile optimale Pläne gibt, und ob in bestimmten Fällen Pläne existieren, die für alle Bearbeitungszeitvektoren $\boldsymbol{p}$ optimal bleiben.

Zusammenhang zwischen Stabilität und Irreduzibilität

In diesem Unterabschnitt wird ein Zusammenhang zwischen dem Stabilitätsradius optimaler Pläne und ihrer Irreduzibilität hergestellt. Auf diese Weise ist es unter anderem möglich, wichtige Ergebnisse aus [89, 90] bezüglich der Stabilität optimaler Pläne mit Hilfe des Konzepts der Irreduzibilität zu formulieren.

Für ein Open-Shop-Problem mit n Aufträgen und m Maschinen ist $\mathcal{P}_{n,m}$ die Menge aller $n \times m$–Pläne. Bei den anderen Shop-Scheduling-Problemen (Job-Shop-, Flow-Shop- und General-Shop-Problem) sind die entsprechenden Mengen ($\mathcal{P}^J_{n,m}$, $\mathcal{P}^F_{n,m}$, $\mathcal{P}^G_{n,m}$) offensichtlich kleiner als $\mathcal{P}_{n,m}$, da nicht jeder Plan aus $\mathcal{P}_{n,m}$ die gegebenen Vorrangbedingungen zwischen den Operationen erfüllt. Bei einem General-Shop-Problem $G||C_{\max}$ werden die gegebenen Vorrangbedingungen zwischen den Operationen anhand der disjunktiven Kanten in der Menge C des disjunktiven Graphen $G^* = (V, C \cup D)$ repräsentiert (vgl. Abbildung 3.1). In diesem Fall ist ein Plan A genau dann in der zugehörigen Menge $\mathcal{P}^G_{n,m}$ enthalten, wenn für seinen Ablaufgraphen $G(A) = (V, E)$ die Beziehung $C \subset E$ erfüllt ist.

Das Konzept der Irreduzibilität kann auch für die Shop-Scheduling-Probleme mit Vorrangbedingungen übernommen werden. So ist z. B. bei einem gegebenen General-Shop-Problem mit zugrundeliegender Menge $\mathcal{P}^G_{n,m}$ von Plänen ein Plan B genau dann *irreduzibel*, wenn es keinen Plan $A \in \mathcal{P}^G_{n,m}$ mit $A \prec B$ gibt. Im restlichen Teil dieses Abschnitts wird bei der Benutzung der Begriffe „irreduzibel“, „reduzierbar“, usw. stets von dieser Beschränkung auf diejenigen Pläne ausgegangen, die für das jeweils gegebene General-Shop-Problem relevant sind.

Die folgenden Aussagen für General-Shop-Probleme sind offensichtlich genauso auf Open-Shop-, Job-Shop- und Flow-Shop-Probleme übertragbar, da diese Probleme Spezialfälle des General-Shop-Problems sind. Für das Problem $G||C_{\max}$ beschreibe der Ausdruck $\mathcal{S}^*(\boldsymbol{p})$ anschließend stets die Menge der optimalen Pläne zum gegebenen Bearbeitungszeitvektor $\boldsymbol{p}$.

Satz 6.3.1 *Es sei $A \in \mathcal{S}^*(\boldsymbol{p})$ ein Plan für das Problem $G||C_{\max}$ mit $\varepsilon(A, \boldsymbol{p}) > 0$. Ist A' ein Plan mit $A' \preceq A$, so gilt $A' \in \mathcal{S}^*(\boldsymbol{p})$ und $\varepsilon(A', \boldsymbol{p}) \geq \varepsilon(A, \boldsymbol{p})$.*

Beweis: Es sei A ein optimaler Plan für $G||C_{\max}$. Aus $A' \preceq A$ folgt wegen Satz 6.2.5 die Ungleichung $C_{\max}(A') \leq C_{\max}(A)$, also ist A' auch optimal. Der Stabilitätsradius eines Plans A zum Bearbeitungszeitvektor $\boldsymbol{p} = (p_{11}, \ldots, p_{nm})$ kann gemäß [89] anhand von

$$\varepsilon(A, \boldsymbol{p}) = \inf \left\{ d(\boldsymbol{p}, \boldsymbol{p'}) \;\middle|\; \boldsymbol{p'} \in \mathbb{R}^{nm}_+, \max_{W_a \in \mathcal{W}^*_A} \sum_{o_{ij} \in V_{W_a}} p'_{ij} > \min_{\substack{B \in \mathcal{P}^G_{n,m} \\ B \neq A}} \max_{W_b \in \mathcal{W}^*_B} \sum_{o_{ij} \in V_{W_b}} p'_{ij} \right\} \tag{6.7}$$

berechnet werden, wobei $\mathcal{W}_A^*$ die Menge der dominanten Wege im Ablaufgraphen $G(A)$ des Plans A ist. Wegen $A' \preceq A$ gibt es zu jedem $W_{A'} \in \mathcal{W}_{A'}^*$ ein $W_A \in \mathcal{W}_A^*$ mit $V_{W_{A'}} \subseteq V_{W_A}$. Die Gültigkeit der Ungleichung im entsprechenden Ausdruck (6.7) für $\varepsilon(A', \boldsymbol{p})$ hängt daher von höchstens sovielen Komponenten p'_{ij} ab wie im Fall $\varepsilon(A, \boldsymbol{p})$. Also gibt es für die Wahl der Bearbeitungszeitvektoren $\boldsymbol{p'} \in \mathbb{R}_+^{nm}$ beim Abstand $d(\boldsymbol{p}, \boldsymbol{p'})$ in $\varepsilon(A', \boldsymbol{p})$ mindestens soviele Freiheitsgrade wie in $\varepsilon(A, \boldsymbol{p})$. Damit ist $\varepsilon(A', \boldsymbol{p}) \geq \varepsilon(A, \boldsymbol{p})$. □

Der anschließende Satz gibt ein eineindeutiges Kriterium für die Existenz von stabilen Plänen.

Satz 6.3.2 *Es sei $A \in \mathcal{S}^*(\boldsymbol{p})$ ein Plan für das Problem $G||C_{\max}$. Es ist genau dann $\varepsilon(A, \boldsymbol{p}) > 0$, wenn für alle $B \in \mathcal{S}^*(\boldsymbol{p})$ die Beziehung $A \preceq B$ gilt.*

Beweis: Die in [89] angegebene indirekte Beweisführung kann offensichtlich ohne Schwierigkeiten an die Terminologie des Konzepts der Irreduzibilität angepaßt werden, da der Berechnung des Stabilitätsradius von A stets die Anzahl der Knoten der dominanten Wege im Ablaufgraphen $G(A)$ zugrundeliegt. □

Folgerung 6.3.3 *Jeder eindeutig optimale Plan ist stabil.*

Analog zu Satz 6.3.2 kann nun auch ein Kriterium für Pläne mit unbegrenzter Stabilität anhand des Konzepts der Irreduzibilität formuliert werden.

Satz 6.3.4 *Es sei $A \in \mathcal{S}^*(\boldsymbol{p})$ ein Plan für das Problem $G||C_{\max}$. Es ist genau dann $\varepsilon(A, \boldsymbol{p}) = \infty$, wenn für alle $B \in \mathcal{P}_{n,m}^G$ die Beziehung $A \preceq B$ gilt.*

Beweis: Analog zu Satz 6.3.2. □

Für ein General-Shop-Problem existiert also genau dann ein Plan mit unendlichem Stabilitätsradius, wenn es einen irreduziblen Plan gibt, auf den alle anderen Pläne dieses Problems reduzierbar sind. Offensichtlich kann die Bedingung

$$A \preceq B \quad \text{für alle} \quad B \in \mathcal{P}_{n,m}^G$$

in diesem Satz für General-Shop-Probleme nur dann erfüllt sein, wenn n und m klein sind oder schon einer Vielzahl von technologischen und organisatorischen Reihenfolgen vorgegeben sind, denn „$\preceq$“ ist im allgemeinen keine lineare Ordnung, d. h. im allgemeinen sind nicht alle Pläne bezüglich „$\preceq$“ vergleichbar.

Da eine optimale Lösung eines Shop-Scheduling-Problems im allgemeinen nicht eindeutig ist, erscheint die Darstellung der qualitativen Unterschiede zwischen verschiedenen optimalen Plänen für gegebene Bearbeitungszeiten p_{ij} mit Hilfe der Konzepte der Irreduzibilität und der Stabilität sinnvoll. Abbildung 6.2 zeigt unter allen

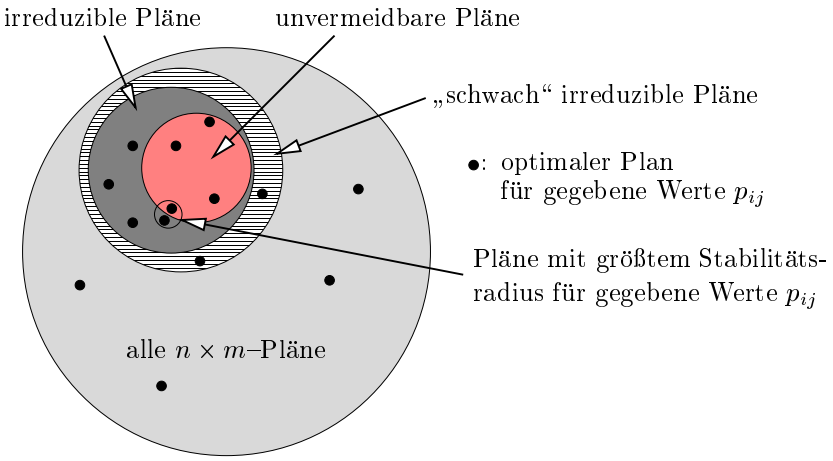


Abbildung 6.2: Qualitative Unterschiede zwischen verschiedenen optimalen Plänen.

Plänen eines gegebenen Formats $n \times m$ die verschiedenen Arten optimaler Pläne. Es ist klar, daß unter den optimalen Plänen diejenigen Pläne im Falle von möglichen Abweichungen der vorgegebenen Werte p_{ij} vorzuziehen sind, die maximalen Stabilitätsradius besitzen. Mit „schwach“ irreduziblen Plänen sind in Abbildung 6.2 diejenigen Pläne gemeint, die sich während eines Enumerationsalgorithmus durch die Anwendung polynomial nachprüfbarer hinreichender Bedingungen für die Reduzibilität als Kandidaten für irreduzible Pläne herausstellen (vgl. Abschnitt 7.3).

Die explizite Berechnung des Stabilitätsradius eines optimalen Plans gemäß [89] ist kompliziert und zeitintensiv. In [16] haben Bräsel, Sotskov und Werner erstmals auch Stabilitätsradien von Plänen für diejenigen Shop-Scheduling-Probleme betrachtet, die die Summe der Fertigstellungszeiten ($\sum C_i$) als Optimalitätskriterium besitzen. Beim Vergleich mit dem $C_{\max}$-Kriterium (siehe auch Sotskov, Tanaev und Werner [90]) stellt sich heraus, daß ein optimaler Plan im Falle der Gesamtbearbeitungszeit $C_{\max}$ gewöhnlich einen größeren Stabilitätsradius aufweist als bei $\sum C_i$. Weiterhin haben die Stabilitätsradien von optimalen Plänen bei dem $C_{\max}$-Kriterium eine größere Varianz. Beim $\sum C_i$-Kriterium besitzen alle optimalen Pläne sogar häufig den gleichen Stabilitätsradius.

Kapitel 7

Enumeration irreduzibler Pläne

Zur Bestimmung der Anzahl $P^*_{n,m}$ der irreduziblen $n \times m$ Pläne werden zwei verschiedene Enumerationsmethoden vorgestellt. Die Enumeration irreduzibler Pläne anhand von Vergleichbarkeitsgraphen ist Gegenstand der Abschnitte 7.1 und 7.2. Bei dieser Methode werden Vertreter der Ähnlichkeitsklassen (Äquivalenzklassen bezüglich der Relation „$\sim$“) erzeugt. Mit Hilfe der zweiten Methode, die in Abschnitt 7.3 und 7.4 beschrieben ist, können alle bzw. alle nicht-struktur-äquivalenten irreduziblen Pläne enumeriert werden. Eine effiziente Implementation dieser Methode, die auf der Enumeration in Kapitel 5 basiert, liefert eine Reihe von numerischen Resultaten, auf die abschließend in Abschnitt 7.5 eingegangen wird.

7.1 Reduzibilität zwischen zwei Plänen

Es sei $G = (V, E)$ ein azyklischer Digraph. Ein Weg $W = (v_0, v_1, \ldots, v_k)$ in G enthält die Knoten $v_0, v_1, \ldots, v_k$ und die Kanten $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$. Zu $G = (V, E)$ wird der Digraph $G^{tc} = (V, E^{tc})$ definiert, in dem für alle $v, w \in V$ genau dann $(v, w) \in E^{tc}$ ist, wenn $v \neq w$ ist und in G ein Weg von v nach w existiert. Der Digraph $G^{tc} = (V, E^{tc})$ heißt *transitive Hülle (transitive closure)* von $G = (V, E)$. Eine Kante (v, w) eines azyklischen Digraphen heißt *redundant*, wenn es einen Weg von v nach w gibt, der die Kante (v, w) nicht enthält. Als *transitive Reduktion (transitive reduction)* von $G = (V, E)$ wird der Digraph $G_{tr} = (V, E_{tr})$ bezeichnet, der keine redundante Kanten enthält, und dessen transitive Hülle gleich der transitiven Hülle von G ist. Abbildung 7.2 zeigt die transitive Hülle $G^{tc}(A)$ und die transitive Reduktion $G_{tr}(A)$ des Ablaufgraphen $G(A)$ aus Abbildung 7.1.

Mit $[G]$ wird der ungerichtete Graph bezeichnet, der einem Digraphen G zugrunde liegt. Weiterhin sei an dieser Stelle daran erinnert, daß es sich bei $n \times m$–Ablaufgraphen um indizierte Digraphen handelt, d. h. jeder Knoten eines Ablaufgraphen wird jeweils mit einer bestimmten Operation o_{ij} des betrachteten Shop-Schedu-

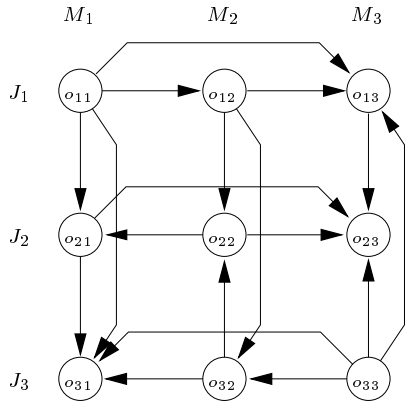


Abbildung 7.1: Ein Ablaufgraph $G(A)$.

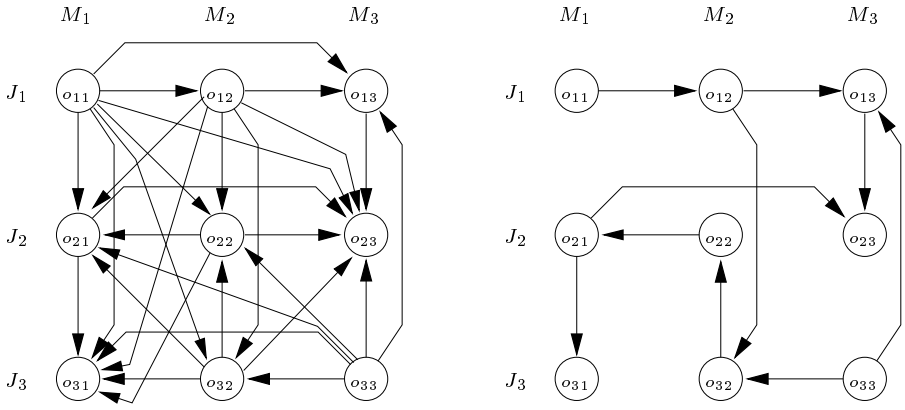


Abbildung 7.2: Die transitive Hülle $G^{tc}(A)$ und die transitive Reduktion $G_{tr}(A)$ von $G(A)$.

ling-Problems identifiziert, da sonst keine eindeutige Beziehung zwischen Plänen und Ablaufgraphen im Sinne von Satz 3.2.2 bestünde. Für zwei indizierte (Di-)Graphen $G_1 = (V, E_1)$ und $G_2 = (V, E_2)$ mit derselben Knotenmenge V wird $G_1 \subset G_2$, $G_1 \subseteq G_2$, bzw. $G_1 = G_2$ geschrieben, falls $E_1 \subset E_2$, $E_1 \subseteq E_2$ bzw. $E_1 = E_2$ gilt. Der folgende Satz gibt ein eineindeutiges Kriterium für die strenge Reduzibilität eines Plans B auf einen Plan A.

Satz 7.1.1 [10] *Es seien A und B zwei $n \times m$–Pläne und $G(A)$ bzw. $G(B)$ die zugehörigen $n \times m$–Ablaufgraphen. Es gilt genau dann $A \prec B$, wenn $[G^{tc}(A)] \subset [G^{tc}(B)]$ ist.*

Beweis: Angenommen, es ist $A \prec B$. Dann gibt es zu jedem Weg W_a in $G(A)$ einen Weg W_b in $G(B)$ mit $V_{W_a} \subseteq V_{W_b}$. Außerdem existiert in $G(B)$ mindestens ein dominanter Weg W_b^*, für den es in $G(A)$ keinen dominanten Weg gibt, der alle Knoten von W_b^* enthält. Gemäß der Definition der transitiven Hülle entspricht jeder dominante Weg eines Digraphen G einer maximalen Clique in $[G^{tc}]$. Daher gilt also $[G^{tc}(A)] \subset [G^{tc}(B)]$.

Ist umgekehrt $[G^{tc}(A)] \subset [G^{tc}(B)]$, so sind alle Cliquen aus $[G^{tc}(A)]$ in Cliquen aus $[G^{tc}(B)]$ enthalten, und es gibt in $[G^{tc}(B)]$ eine maximale Clique C_b, deren Knotenmenge echt größer ist als die einer entsprechenden maximalen Clique C_a in $[G^{tc}(A)]$. Jede orientierte Clique besitzt einen Hamiltonschen Weg (siehe RÉDEI [78]). Es kann schnell eingesehen werden, daß dieser Hamiltonsche Weg eindeutig ist, wenn die Orientierung der Clique transitiv ist. Die den Graphen $[G^{tc}(A)]$ und $[G^{tc}(B)]$ zugrunde liegenden Orientierungen sind transitiv. Seien also W_a^* bzw. W_b^* die eindeutigen dominanten Wege der Ablaufgraphen $G(A)$ bzw. $G(B)$, die den maximalen Cliquen C_a bzw. C_b in den Graphen $[G^{tc}(A)]$ bzw. $[G^{tc}(B)]$ entsprechen. Es ist $V_{W_a^*} \subset V_{W_b^*}$ und für alle anderen Wege W_a in $G(A)$ existiert ein Weg W_b in $G(B)$ mit $V_{W_a} \subseteq V_{W_b}$. Also gilt $A \prec B$. □

Mit Hilfe dieses Satzes läßt sich relativ leicht ein Algorithmus konstruieren, der für zwei gegebene $n \times m$–Pläne A und B anhand der zugehörigen Ablaufgraphen $G(A)$ und $G(B)$ testet, ob $A \prec B$, $A \preceq B$ oder $A \sim B$ gilt. Für diesen Test ist die Berechnung der transitiven Hülle eines Ablaufgraphen sowie das Überprüfen der Relation $[G^{tc}(A)] \subset [G^{tc}(B)]$ notwendig. Ein hierzu in [10] benutztes Verfahren hat die Zeitkomplexität $O(n^2m^2)$.

Das Problem der Berechnung der transitiven Hülle bzw. der transitiven Reduktion ist sehr häufig untersucht worden (siehe z. B. [43]). Es werden nun in diesem Zusammenhang zwei Arbeiten mit den besten, bis heute bekannten Laufzeiten zitiert.

Der in [40] vorgestellte Algorithmus von GORALČÍKOVÁ und KOUBEK berechnet die transitive Hülle $G^{tc} = (V, E^{tc})$ eines azyklischen Digraphen $G = (V, E)$ in der

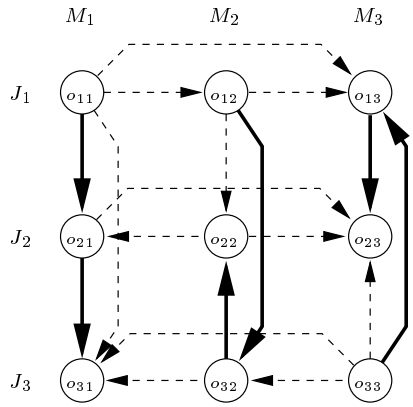


Abbildung 7.3: Eine Ketten-Zerlegung des Ablaufgraphen $G(A)$ aus Abbildung 7.1.

Zeit $O(|V| \cdot |E_{tr}| + |E^{tc}|)$, wobei E_{tr} bzw. E^{tc} die Menge der Kanten der zugehörigen transitiven Reduktion G_{tr} bzw. Hülle G^{tc} darstellt. Da für einen Ablaufgraphen $G = (V, E)$ stets die Beziehung $O(|E^{tc}|) = O(|V| \cdot |E_{tr}|)$ gilt, reduziert sich in diesem Fall die Laufzeit für die Berechnung der transitiven Hülle zu $O(|V| \cdot |E_{tr}|)$. Die transitive Reduktion $G_{tr}(A)$ eines $n \times m$–Ablaufgraphen $G(A)$ enthält keine redundante Kanten. Also sind für jeden Auftrag J_i bzw. für jede Maschine M_j höchstens die direkten Vorgänger-Nachfolger-Beziehungen der Technologie bzw. Organisation in $G_{tr}(A)$ enthalten. Das heißt, in der transitiven Reduktion $G_{tr}(A)$ existieren maximal $n(m-1) + m(n-1)$ gerichtete Kanten (vgl. Abbildung 7.2). Also folgt für die Laufzeit des Algorithmus zur Bestimmung der transitiven Hülle eines $n \times m$–Ablaufgraphen insgesamt die Schranke $O(n^2m^2)$. Mit Hilfe sogenannter Ketten-Zerlegungen konnte SIMON in [86] die Laufzeit für die Berechnung der transitiven Hülle eines Graphen verringern:

Definition 7.1.2 Es sei $G = (V, E)$ ein Digraph. Eine *Ketten-Zerlegung (chain decomposition)* von G ist eine Partition $P = \{P_1, \ldots, P_k\}$ von V in disjunkte nichtleere Mengen P_i mit $V = P_1 \cup \ldots \cup P_k$, bei der jede Menge P_i, $i = 1, \ldots, k$ einen gerichteten Weg (auch *Kette* genannt) in G aufspannt, wenn transitive Kanten ignoriert werden. Die Zahl k heißt *Weite (width)* der Ketten-Zerlegung P.

Als Beispiel ist in Abbildung 7.3 eine Ketten-Zerlegung der Weite 3 eines 3×3–Ablaufgraphen dargestellt.

In [86] wird gezeigt, daß sich eine Ketten-Zerlegung eines Digraphen $G = (V, E)$ in der Zeit $O(|V| + |E|)$ konstruieren läßt. Darauf aufbauend hat SIMON den Algorithmus in [40] verbessert, so daß die transitive Hülle von $G = (V, E)$ in $O(k \cdot |E_{tr}|)$ berechnet werden kann, wobei k die Weite einer Ketten-Zerlegung von G ist. Die

Laufzeit des Algorithmus in [40] wird dadurch verkürzt, daß anstelle der Bestimmung aller Knoten $w \in V$ die von einem gegebenen Knoten $v \in V$ über einen Weg aus erreichbar sind, nur die jeweiligen ersten Knoten der Mengen P_i der Ketten-Zerlegung bestimmt werden, die von v aus erreichbar sind. Für diesen Schritt ergibt sich also anstelle von $O(|V|)$ die Laufzeit $O(k)$, da die bereits bestehende Information der Ketten-Zerlegung auf diese Weise ausgenutzt wird.

Offensichtlich kann die Menge der Knoten eines $n \times m$–Ablaufgraphen stets in n bzw. m Ketten zerlegt werden (siehe Abbildung 7.3). Für die Bestimmung der transitiven Hülle eines $n \times m$–Ablaufgraphen anhand des Algorithmus von Simon [86] ergibt sich wegen $|E_{tr}| = O(nm)$ für $n \leq m$ also insgesamt ein Zeitaufwand der Größenordnung $O(n^2 m)$. Allerdings benötigt der Test, ob die Relationen $\subset$, $\subseteq$ oder $=$ zwischen den Graphen $[G^{tc}(A)]$ und $[G^{tc}(B)]$ bestehen, bereits $O(n^2 m^2)$ Zeit. Also verringert sich trotz des verbesserten Algorithmus von Simon [86] nicht die Laufzeit für den gesamten Algorithmus, der gemäß Satz 7.1.1 zwei Pläne A und B auf Reduzibilität testet.

7.2 Enumeration bezüglich Ähnlichkeitsklassen

Ein *Vergleichbarkeitsgraph (comparability graph)* G^* ist ein ungerichteter Graph, der sich transitiv orientieren läßt. Mit Hilfe bestimmter Vergleichbarkeitsgraphen wird in diesem Abschnitt eine Enumerationsmethode vorgestellt, die auf einer neuen Charakterisierung irreduzibler Pläne beruht.

Es sei $H_{n\times m} = (V_H, E_H)$ der in Abschnitt 3.1 eingeführte Hamming-Graph $K_n \times K_m$. Die Knotenmenge V_H korrespondiert in naheliegender Weise mit der Menge der Operationen o_{ij} des betrachteten Shop-Scheduling-Problems mit n Aufträgen und m Maschinen. Weiterhin sei $G(A)$ der $n \times m$–Ablaufgraph eines beliebigen $n \times m$–Plans A. Jede Kante e des Graphen $[G^{tc}(A)]$ mit $e \notin E_H$ heißt *Diagonalkante*. Aufgrund von Satz 7.1.1 kann die folgende Charakterisierung irreduzibler Pläne gegeben werden.

Folgerung 7.2.1 *Ein $n \times m$–Plan A ist genau dann irreduzibel, wenn kein Vergleichbarkeitsgraph G^* mit $H_{n\times m} \subseteq G^* \subset [G^{tc}(A)]$ existiert.*

In [10] haben Bräsel *et al.* mit Hilfe dieses Zusammenhangs einen Enumerationsalgorithmus für irreduzible Pläne konstruiert. Der Algorithmus beruht auf der Entwicklung ungerichteter Graphen G durch sukzessives Hinzufügen von Diagonalkanten zum Hamming-Graphen $H_{n\times m}$. Sobald unter diesen Graphen ein Vergleichbarkeitsgraph G^* gefunden wird, handelt es sich um einen inklusions-minimalen Vergleichbarkeitsgraphen bezüglich Kantenanzahl unter der Voraussetzung $H_{n\times m} \subseteq G^*$.

Die zum Erkennen des Vergleichbarkeitsgraphen notwendige transitive Orientierung von G^* liefert dann einen zugehörigen irreduziblen Plan A' mit $G^* = [G^{tc}(A')]$.

Für diesen Algorithmus zur Enumeration irreduzibler Pläne spielt die transitive Orientierung von Graphen eine zentrale Rolle. In [65, 66, 67, 91] haben McConnell und Spinrad das Problem der transitiven Orientierung von Graphen ausführlich untersucht und mit Hilfe der sogenannten modularen Dekomposition (Zerlegung eines Graphen in bestimmte Komponenten) schnelle Algorithmen zu dessen Lösung gefunden. Das aktuellste Resultat [67] beschreibt einen Algorithmus für die transitive Orientierung eines Graphen $G = (V, E)$ mit Zeitkomplexität $O(|V| + |E|)$. Da dieser Algorithmus jedoch nicht erkennt, ob es sich beim betrachteten Graphen G überhaupt um einen Vergleichbarkeitsgraphen handelt, d. h. ob sich G überhaupt transitiv orientieren läßt, muß zum Erkennen eines Vergleichbarkeitsgraphen zusätzlich der Algorithmus von Simon [86] angewandt werden. Es wird dabei getestet, ob sich die gefundene Orientierung von G von seiner transitiven Hülle unterscheidet. Ist dies nicht der Fall, liegt ein Vergleichbarkeitsgraph vor. Im Rahmen des Enumerationsalgorithmus ist für das Erkennen von Vergleichbarkeitsgraphen G^* mit $H_{n\times m} \subseteq G^* \subset [G^{tc}(A)]$ die Zeitkomplexität des Verfahrens aus [86] nicht dominierend. Der zugrunde liegende Graph $[G^{tc}(A)]$ der transitiven Hülle eines Ablaufgraphen $G(A)$ enthält maximal $O(n^2m^2)$ Kanten. Diese Schranke ist für die Laufzeit eines Verfahrens zum Erkennen entsprechender Vergleichbarkeitsgraphen entscheidend.

Neben den hier erwähnten Verfahren zur transitiven Orientierung und zum Erkennen von Vergleichbarkeitsgraphen bildet die Auswahl der Reihenfolge der hinzuzufügenden Diagonalkanten einen wichtigen Bestandteil des in [10] beschriebenen Enumerationsalgorithmus. Bei dieser Enumeration wird ein Vertretersystem M für die Ähnlichkeitsklassen in der Menge $\mathcal{P}^*_{n,m}$ aller irreduziblen $n \times m$–Pläne erzeugt.

7.3 Hinreichende Bedingungen

Aufbauend auf Ergebnissen von M. Kleinau [55] werden in diesem Abschnitt hinreichende Bedingungen für die Reduzibilität von Plänen hergeleitet. Offensichtlich korrespondieren hinreichende Bedingungen für die Reduzibilität eines Plans B mit entsprechenden notwendigen Bedingungen für die Irreduzibilität von B. Anhand dieser Bedingungen können im Verlaufe des Plan-Enumerationsalgorithmus aus Abschnitt 5.3 diejenigen Pläne in effizienter Weise eliminiert werden, deren zugeordnete Ablaufgraphen ungünstige Wegstrukturen besitzen. Durch diese Selektion ist die Menge der enumerierten Pläne im Vergleich zur Menge aller Pläne schon erheblich eingeschränkt.

Die folgenden hinreichende Bedingungen für die strenge Reduzibilität eines ge-

gebenen Plans B beruhen jeweils auf langen gerichteten Wegen im zugeordneten Ablaufgraphen $G(B)$. Die Beweise von Satz 7.3.1 – 7.3.4 erscheinen in [10, 11]. Exemplarisch wird hier nur Satz 7.3.1 bewiesen.

Satz 7.3.1 [11] *Es sei B ein Plan, der eine Operation o_{ij} mit folgenden Eigenschaften enthält: Die Operation o_{ij} besitzt mindestens einen Nachfolger, aber kein Nachfolger von o_{ij} in der Zeile i bzw. Spalte j hat einen direkten Vorgänger außerhalb der Zeile i bzw. Spalte j. Dann gibt es einen Plan A mit $A \prec B$.*

Beweis: Es wird ein Plan A konstruiert, indem die Operation o_{ij} gelöscht und als Senke in die technologische Reihenfolge des Auftrags J_i und die organisatorische Reihenfolge der Maschine M_j wiedereingefügt wird. Wenn auf diese Weise im zugehörigen Ablaufgraphen eine neue Menge von Operationen entstehen würde, die auf einem gemeinsamen Weg liegen, dann muß diese Menge o_{ij} enthalten, da der Rest des Plans unverändert bleibt. Angenommen, es gibt eine Operation o_{kl}, die auf einem Weg mit o_{ij} in $G(A)$ liegt, aber nicht in $G(B)$. Da o_{ij} Senke in $G(A)$ ist, muß dieser Weg von o_{kl} nach o_{ij} gerichtet sein, und an einer Stelle in Zeile i oder Spalte j eintreten. Da es keinen Weg von o_{kl} nach o_{ij} in $G(B)$ gibt, muß die Operation, bei der der neue Weg in $G(A)$ in Zeile i oder Spalte j eintritt, ein Nachfolger von o_{ij} in $G(B)$ sein. Dies ist ein Widerspruch zur im Satz gemachten Annahme. Daher gilt $A \preceq B$.

Es wird nun gezeigt, daß in $G(A)$ weniger Mengen von Operationen existieren, die jeweils auf einem gemeinsamen Weg liegen. Ohne Beschränkung der Allgemeinheit gelte für die Einträge von B die Beziehung

$$b_{\max} = \max\{b_{pj} | p = 1, \ldots, n\} \geq \{b_{iq} | q = 1, \ldots, m\}.$$

Es sei k so gewählt, daß $b_{kj} = b_{\max}$ ist. Also ist o_{kj} in $G(B)$ ein Nachfolger von o_{ij} und hat nach Annahme keinen Vorgänger außerhalb von Spalte j. Weiterhin ist jede Operation o_{kl} mit $l \neq j$ ein Nachfolger von o_{kj} und liegt daher in $G(B)$ auf einem gemeinsamen Weg mit o_{ij}. Wegen $b_{kl} > b_{\max}$ gibt es in $G(A)$ keinen Weg, der in o_{kl} startet und in Zeile i oder Spalte j eintritt. Andererseits gibt es in $G(A)$ keine Wege, die in o_{ij} starten. Wegen Satz 7.1.1 gilt also $A \prec B$. □

Satz 7.3.2 [10] *Es sei B ein $n \times m$–Plan, bei dem jeder Auftrag J_i, $i = 1, \ldots, n$ als erstes auf derselben Maschine M_j bearbeitet wird. Dann gibt es einen Plan A mit $A \prec B$.*

Satz 7.3.3 [11] *Es sei $n, m \geq 3$ und B ein $n \times m$–Plan mit maximalen Rang $r \geq nm - 2$. Dann gibt es einen Plan A mit $A \prec B$.*

Satz 7.3.4 [11] *Es sei $B = (b_{ij})$ ein Plan, der zwei Aufträge J_i und J_k sowie eine Maschine M_j enthält, so daß o_{ij} die letzte Operation von J_i, o_{kj} die erste Operation von J_k, und o_{kj} direkter Nachfolger von o_{ij} in der organisatorischen Reihenfolge von M_j ist. Wenn ein $l \neq j$ existiert mit $b_{kl} \leq b_{ij} + 3$ oder $b_{kj} \leq b_{il} + 3$, dann gibt es einen Plan A mit $A \prec B$.*

Dieser Satz ist eine alternative Formulierung eines ursprünglich in [55] erzielten Resultats. Er liefert eine hinreichende Bedingung dafür, daß der Plan B durch Umdrehen einer gerichteten Kante im zugehörigen Ablaufgraphen $G(B)$ zu einem Plan A streng reduzierbar ist. Ein irreduzibler Plan darf daher notwendiger Weise keine der im Satz 7.3.4 beschriebenen Eigenschaften haben.

7.4 Enumeration durch Ausschlußverfahren

Die Dissertation von M. Kleinau [55] enthält einen Algorithmus, der entscheidet, ob ein gegebener Plan B irreduzibel ist. Dieser Irreduzibilitätstest benötigt exponentiellen Zeitaufwand und ist daher zur direkten Anwendung auf jeden enumerierten Plan ungeeignet, weil die Anzahl aller $n \times m$–Pläne bereits für vergleichsweise kleine Werte von n und m sehr groß ist (siehe Tabelle 5.4).

Der hier vorgestellte neue Enumerationsalgorithmus basiert auf Algorithmus 5.3.7 zur Enumeration aller $n \times m$–Pläne für gegebene n und m. An geeigneten Stellen in Algorithmus 5.3.7 werden die Bedingungen aus dem vorangegangenen Abschnitt angewandt, um die streng reduzierbaren Pläne verwerfen zu können. Es ist leicht zu sehen, daß die notwendigen Bedingungen für die Irreduzibilität gemäß Satz 7.3.4–7.3.3 jeweils in polynomialer Zeit getestet werden können. Zum Teil ist es sogar möglich, diese Bedingungen bereits auf Teilpläne anzuwenden, so daß auf die vollständige Erzeugung bestimmter Pläne ganz verzichtet werden kann, falls ein Teilplan bereits eine der notwendigen Bedingungen für die Irreduzibilität verletzt.

Bei Anwendung der Tests aufgrund Satz 7.3.4–7.3.3 stellt sich heraus, daß diejenigen Verfahren sehr effektiv sind, die auf dem Wiedereinsetzen einer Operation als Quelle oder Senke beruhen. Das heißt, bei Anwendung dieser Verfahren werden die streng reduzierbaren Pläne am häufigsten als solche erkannt. Daher wird im folgenden die Zeitkomplexität dieses Verfahrens angegeben.

Satz 7.4.1 [11] *Es sei $n \leq m$. Das Problem, ob ein $n \times m$–Plan B durch Löschen und Wiedereinfügen einer Operation als Quelle oder Senke auf einen Plan A streng reduziert werden kann, ist in $O(n^2m^2)$ Zeit entscheidbar.*

Beweis: Die transitive Hülle $G^{tr}(B)$ des Ablaufgraphen $G(B)$ eines Plans B läßt sich wie schon gezeigt für $n \leq m$ in $O(n^2m)$ Zeit bestimmen. Der Beweis dieses Satzes in [11] zeigt, daß für eine feste Operation o_{ij} maximal $O(nm)$ gerichtete

Wege in $G^{tr}(B)$ getestet werden müssen, um über die strenge Reduzibilität von B entscheiden zu können. Da diese Prozedur für alle nm Operationen in $G(B)$ durchgeführt werden muß, ergibt sich insgesamt die Zeitkomplexität $O(n^2m^2)$. □

Falls ein Plan während des Enumerationsalgorithmus alle beschriebenen notwendigen Bedingungen für die Irreduzibilität erfüllt, wird ein abschließender Test auf Irreduzibilität angewandt, der allerdings exponentiellen Zeitaufwand benötigt. Dieser Test basiert auf sogenannten Implikationsklassen, in die die Menge der gerichteten Kanten des Ablaufgraphen $G(B)$ zum betrachteten Plan B partitioniert werden kann.

Die Einführung der Implikationsklassen ist folgendermaßen motiviert: Angenommen, für einen Plan B gibt es im Graphen $[G^{tc}(B)]$ keine ungerichtete Kante $\{o_{ij}, o_{kl}\}$ zwischen den Operationen o_{ij} und o_{kl}. Dann existiert diese Kante nach Satz 7.1.1 auch nicht in $[G^{tc}(A)]$ für jeden Plan A mit $A \prec B$. Also enthält $G(A)$ entweder die beiden gerichteten Kanten (o_{ij}, o_{kj}) und (o_{kl}, o_{kj}) oder (o_{kj}, o_{ij}) und (o_{kj}, o_{kl}). In dieser Weise bedingen sich die Orientierungen der beiden Kanten $\{o_{ij}, o_{kj}\}$ und $\{o_{kj}, o_{kl}\}$ gegenseitig.

Zur Definition der Implikationsklassen wird die Terminologie von GOLUMBIC [39] übernommen: Es wird von einem gegebenen ungerichteten Graphen $[G^{tc}(A)]$ ausgegangen. Eine gerichtete Kante (o_{ij}, o_{kj}) in $G(A)$ *erzwingt direkt* die Kante (o_{kl}, o_{kj}), geschrieben, $(o_{ij}, o_{kj})\Gamma(o_{kl}, o_{kj})$, wenn $\{o_{ij}, o_{kl}\}$ nicht in $[G^{tc}(A)]$ existiert. Diese binäre Relation wird analog für $(o_{kj}, o_{ij})\Gamma(o_{kj}, o_{kl})$ definiert. Die transitive Hülle Γ^{tc} von Γ ist offensichtlich eine Äquivalenzrelation. Die Äquivalenzklassen bezüglich Γ^{tc} heißen *Implikationsklassen*. Zwei Kanten e und f sind genau dann in derselben Implikationsklasse ($e\,\Gamma^{tc} f$), wenn es Kanten $e_1, \ldots, e_k$ mit

$$e\Gamma e_1\Gamma e_2\Gamma \cdots \Gamma e_k\Gamma f$$

gibt.

Wenn ein Plan B mit $G(B) = (V, E)$ durch Umkehrung der Orientierungen der Kanten einer bestimmten Menge $E' \subseteq E$ zu einem Plan A reduziert werden soll, so müssen offensichtlich jeweils die Orientierungen aller Kanten einer Implikationsklasse umgekehrt werden, da sonst zusätzliche Wege in $G(B)$ entstehen würden. Es ergibt sich die folgende Aussage.

Satz 7.4.2 [11] *Es sei B ein Plan mit Ablaufgraph $G(B) = (V, E)$. Wenn alle Kanten aus E zur gleichen Implikationsklasse gehören, dann ist B irreduzibel.*

Beweis: Da alle Kanten von $G(B)$ zu einer Implikationsklasse gehören, wird der Plan B nur dann reduziert, wenn die Orientierungen aller Kanten in $G(B)$ umgekehrt werden. Wegen $B \sim \overline{B}$ ist B irreduzibel. □

Beim abschließenden Test auf Irreduzibilität eines Plans B werden die Kanten aus $G(B)$ in ihre Implikationsklassen partitioniert, wobei k die Anzahl dieser Implikationsklassen sei. Danach wird für jede Teilmenge $\mathcal{T}$ der Menge aller Implikationsklassen der Plan A bzw. Ablaufgraph $G(A)$ konstruiert, der durch Umkehrung aller Orientierungen der Kanten entsteht, die zu den Implikationsklassen aus $\mathcal{T}$ gehören. Da alle 2^k Teilmengen der Menge der Implikationsklassen geprüft werden müssen, ist dieser Test auf Irreduzibilität nicht in polynomialer Zeit realisierbar.

Ablauf des Enumerationsalgorithmus

Es wird von der *Plan-Enumeration* (Algorithmus 5.3.7) ausgegangen. In Schritt 2 werden bereits bei der Technologie-Erzeugung jeweils Tests gemäß Satz 7.3.1 und 7.3.2 angewandt. Dabei werden nur diejenigen Pläne vollständig erzeugt, die keine der Bedingungen aus Satz 7.3.1 oder 7.3.2 erfüllen. Auf jeden vollständig erzeugten Plan wird zunächst der Test bezüglich seines maximalen Rangs (gemäß Satz 7.3.3) angewandt (im Algorithmus 5.3.7 zwischen Schritt 3 und 4). Nach dem Ermitteln der Automorphismen der Pläne bezüglich Isomorphie, Äquivalenz oder Struktur-Äquivalenz in Schritt 4 werden die übrigbleibenden Vertreter anhand von Satz 7.3.2 und 7.3.4 auf mögliche strenge Reduzibilität hin überprüft. Bevor der abschließende exponentielle Irreduzibilitätstest aufgrund der Betrachtung der Implikationsklassen für die verbleibenden Pläne gestartet wird, testet man für jede Operation o_{ij} eines Plans, ob der Plan durch Löschen und Wiedereinsetzen von o_{ij} als Quelle bzw. Senke streng reduziert werden kann (vgl. Satz 7.4.1). Am Ende des Algorithmus werden auf diese Weise ausschließlich die irreduziblen Vertreter der betrachteten Äquivalenzklassen als Pläne ausgegeben.

Der anschließende Unterabschnitt zeigt, daß ein polynomialer Irreduzibilitätstest bisher nur im Fall einer bestimmten Klasse von unvollständigen Mengen von Operationen bekannt ist.

Polynomialer Test auf Irreduzibilität bei bestimmten unvollständigen Mengen von Operationen

Es sei $\mathcal{J} = \{J_1, \ldots, J_n\}$ die Menge der Aufträge und $\mathcal{M} = \{M_1, \ldots, M_m\}$ die Menge der Maschinen eines Shop-Scheduling-Problems. Weiterhin sei $G_{n,m} = (\mathcal{J} \cup \mathcal{M}, \mathcal{O})$ der assoziierte bipartite Graph, bei dem die Kantenmenge $\mathcal{O} \subseteq \mathcal{J} \times \mathcal{M}$ die Menge der Operationen o_{ij} ($i = 1, \ldots, n$ und $j = 1, \ldots, m$) ist. Es werden nun Shop-Scheduling-Probleme mit $\mathcal{O} \subsetneq \mathcal{J} \times \mathcal{M}$ betrachtet. Die Lösungen von derartigen Problemen mit *unvollständigen Mengen von Operationen* werden anhand von Teilplänen $A = (a_{ij})$ repräsentiert, wobei für alle $i = 1, \ldots, n$ und $j = 1, \ldots, m$ der leere Eintrag $a_{ij} = \cdot$ gesetzt wird, wenn $o_{ij} \notin \mathcal{O}$ ist. Ansonsten geben die Einträge $a_{ij} \neq \cdot$

n	m	$P_{n,m}$	$P^*_{n,m}$	$S_{n,m}$	$S^*_{n,m}$
2	2	14	2	3	1
2	3	204	12	12	1
3	3	19 164	516	147	7
2	4	5 016	72	68	2
3	4	3 733 056	32 688	13 100	123
4	4	6 941 592 576	27 106 560	3 017 369	12 073
2	5	185 520	480	422	2
3	5	1 288 391 040	2 932 560	895 388	2 073
4	5	26 549 943 275 520	??	4 609 489 912	5 936 306
2	6	9 595 440	3 600	3 495	3
3	6	712 770 186 240	352 098 720	82 507 654	40 933
2	7	659 846 880	30 240	33 193	3
3	7	589 563 294 888 960	??	9 748 141 078	??

Tabelle 7.1: Anzahlen irreduzibler $n \times m$–Pläne und Gesamtanzahlen der $n \times m$–Pläne.

analog zu den Plänen die Reihenfolge der Bearbeitung der Operationen wieder (vgl. Definition 5.3.5).

Für Teilpläne eines Shop-Scheduling-Problems, dessen zugrundeliegende Menge von Operationen $\mathcal{O}$ als Kantenmenge im bipartiten Graphen $G_{n,m}$ einen Baum aufspannt, hat TAUTENHAHN in [96] einen polynomialen Irreduzibilitätstest entwickelt.

7.5 Numerische Auswertungen

In Tabelle 7.1 werden die Anzahlen der irreduziblen bzw. nicht-struktur-äquivalenten irreduziblen $n \times m$–Pläne ($P^*_{n,m}$ bzw. $S^*_{n,m}$) den entsprechenden Gesamtanzahlen für kleine Werte n und m gegenübergestellt. Es zeigt sich, daß jeweils nur ein kleiner Prozentsatz einer Menge von $n \times m$–Plänen irreduzibel ist (vgl. Tabelle 7.2 für $P^*_{n,m}/P_{n,m}$ in %). Die Algorithmen zur Lösung von Shop-Scheduling-Problemen, die ausschließlich in der Menge der irreduziblen Pläne nach einem Optimum suchen, sind folglich wesentlich effektiver als diejenigen, deren Suchraum durch die Menge aller zulässigen Lösungen gebildet wird.

Die Effektivität der verschiedenen Tests aufgrund der hinreichenden Bedingungen für die Reduzibilität von Plänen gemäß Satz 7.3.1 – 7.3.4 und Satz 7.4.1 kann z. B. an der Anzahl der 3×4–Pläne abgelesen werden, die diese Bedingungen erfüllen (siehe Tabelle 7.3). Alle bis auf den letzten Test (Irreduzibilitätstest gemäß der

$n \setminus m$	2	3	4	5	6	7
2	14.30	5.88	1.44	0.26	0.038	0.0046
3		2.69	0.88	0.23	0.049	??
4			0.39	??	??	??

Tabelle 7.2: Verhältnisse der Anzahlen irreduzibler $n \times m$–Pläne zu den jeweiligen Gesamtanzahlen (in %).

Test gemäß ...	# 3×4–Pläne
(ohne Test)	13100
Satz 7.3.1 und 7.3.2, ausschließlich auf die Technologien angewandt	6081
Satz 7.3.3	5640
Satz 7.3.2, ausschließlich auf die Organisationen angewandt	5050
Satz 7.3.4	4291
Satz 7.4.1	256
Umkehrung der Kanten aus Implikationsklassen	123

Tabelle 7.3: Anzahlen nicht-struktur-äquivalenter 3×4–Pläne, die nach Anwendung der verschiedenen Tests auf Irreduzibilität übrigbleiben.

Umkehrung der Kanten aus den Implikationsklassen) sind in polynomialer Zeit ausführbar. Außerdem zeigt sich, daß dieser exponentielle Irreduzibilitätstest auf nur ca. 2% aller nicht-struktur-äquivalenten 3×4–Pläne angewandt werden muß, und vergleichbare Werte gelten auch für $n, m \geq 3$.

Die Werte in Tabelle 7.4 legen die Vermutung nahe, daß die hinreichende Bedingung für die strenge Reduzibilität bezüglich des maximalen Rangs aus Satz 7.3.3 für $n, m \geq 3$ noch verschärft werden kann.

Während die maximale Anzahl von Implikationsklassen für $n \leq 3$ und $m \leq 4$ noch relativ klein ist (≤ 4), wächst dieser Wert mit steigenden n bzw. m stark an. Dies liefert eine Begründung dafür, daß bereits die Anzahl der irreduziblen 4×5–Pläne nicht mehr in angemessener Zeit auf die beschriebene Methode berechenbar ist, denn der abschließende Irreduzibilitätstest muß auf alle Teilmengen der Menge aller Implikationsklassen eines Plans angewandt werden.

Die in der vorliegenden Arbeit vorgestellten Algorithmen zur Enumeration der irreduziblen Pläne sind zu einem gesamten C++–Programm zusammengefügt worden, so daß für gegebene n und m alle Werte für die irreduziblen $n \times m$–Pläne in

$n \times m$	max. Rang r	# irred. Pläne
2×2	2	1
2×3	3	1
3×3	3	1
	4	3
	5	3
2×4	4	2
3×4	4	4
	5	40
	6	75
	7	4
4×4	4	4
	5	88
	6	1847
	7	5845
	8	3932
	9	355
	10	2

$n \times m$	max. Rang r	# irred. Pläne
2×5	5	2
3×5	5	38
	6	541
	7	1153
	8	334
	9	7
2×6	6	3
3×6	6	658
	7	8428
	8	19744
	9	10844
	10	1248
	11	11
2×7	7	3

Tabelle 7.4: Anzahlen der nicht-struktur-äquivalenten irreduziblen $n \times m$–Pläne mit maximalem Rang r.

$n \times m$	# Implikationsklassen	
	durchschnittlich	maximal
2×2	1.00	1
2×3	1.00	1
3×3	1.14	2
2×4	1.00	1
3×4	1.30	4
4×4	1.53	9
2×5	1.00	1
3×5	1.87	8
2×6	1.00	1
3×6	2.97	18
2×7	1.00	1

Tabelle 7.5: Durchschnittliche und maximale Anzahlen der Implikationsklassen unter den nicht-struktur-äquivalenten irreduziblen $n \times m$–Plänen.

den Tabellen dieses Abschnitts in einem Durchlauf errechnet werden können. Der gesamte Programmdurchlauf benötigt auf einem PC Pentium I (133 Mhz) im Fall $n = m = 4$ ca. 184 Minuten.

Abschließend wird nun ein Beispiel angeführt, das drei spezielle irreduzible 4×4–Pläne zeigt.

Beispiel 7.5.1 Es seien die Pläne A_1, A_2 und A_3 mit

$$A_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 5 & 10 \\ 5 & 8 & 4 & 9 \\ 6 & 7 & 8 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 3 & 8 & 9 \\ 5 & 6 & 4 & 10 \\ 6 & 1 & 7 & 2 \\ 7 & 2 & 3 & 4 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 2 & 6 & 7 & 8 \\ 4 & 5 & 6 & 7 \\ 1 & 2 & 9 & 3 \\ 3 & 4 & 8 & 1 \end{pmatrix}$$

gegeben. Die Pläne A_1 und A_2 sind die einzigen beiden nicht-struktur-äquivalenten irreduziblen 4×4–Pläne mit maximalem Rang 10. Der Plan A_3 besitzt 9 Implikationsklassen, welches der maximalen Anzahl von Implikationsklassen unter den irreduziblen 4×4–Plänen entspricht.

Die Pläne A_1 und A_2 sind also die einzigen beiden nicht-struktur-äquivalenten 4×4–Pläne, deren zugehörige Ablaufgraphen $G(A_1)$ und $G(A_2)$ einen gerichteten Weg der Länge 10 enthalten. Obwohl ein Weg der Länge 10 relativ lang ist für optimale Pläne des Formats 4×4, können A_1 und A_2 bei der Suche nach einem optimalen Plan nicht von vorneherein ausgeschlossen werden, da keine Pläne A' mit $A' \prec A_1$ bzw. $A' \prec A_2$ existieren.

Der Plan A_3 ist ein Beispiel eines 4×4-Plans, bei dem während der Enumeration der abschließende Test auf Irreduzibilität am meisten Zeit beansprucht, weil A_3 die maximale Anzahl von Implikationsklassen besitzt. In diesem Fall müssen also beim Irreduzibilitätstest alle Kombinationen der 9 Implikationsklassen bezüglich der Orientierung ihrer Kanten überprüft werden.

Kapitel 8

Schlußbemerkungen

Die Grundlage vieler Ergebnisse dieser Arbeit ist die Identifikation von zulässigen Lösungen eines Open-Shop-Problems mit Plänen und Ablaufgraphen.

Es werden zunächst neue theoretische Resultate über die Anzahl der Pläne eines Open-Shop-Problems erzielt. So ist es nun möglich, die Anzahl der $3 \times m$–Pläne anhand einer Summenformel direkt anzugeben. Dieses Ergebnis baut auf einer in [3] gezeigten allgemeinen Formel für die Anzahl lateinischer Rechtecke des Formats $3 \times m$ mit maximalem Eintrag $r \leq 3m$ auf. Da auch für die entsprechenden lateinischen Rechtecke des Formats $4 \times m$ in [3] eine Formel erscheint, liegt die gleiche Vorgehensweise für die $4 \times m$–Pläne nahe. Die Korrektheit der Formel aus [3] für die 4-zeiligen lateinischen Rechtecke ist jedoch fragwürdig, da sich schon beim Einsetzen kleiner m und r Abweichungen von den zu erwartenden Anzahlen ergeben. Die fragliche Formel ist sehr kompliziert und der Beweis in [3] ist nur kurz mit dem Hinweis auf den nächstkleineren Fall skizziert. Daher verspricht ein weiterer intensiver Kontakt mit den Autoren dieser Arbeit Aussicht auf Erfolg bezüglich einer allgemeinen Formel für die $4 \times m$–Pläne.

Für $n \times m$–Pläne mit $n \geq 4$ werden neue obere und untere Schranken bewiesen. Während die obere Schranke eine echte Verbesserung der bisher bekannten Ergebnisse liefert, erweist sich die neue untere Schranke nach langer Rechnung als etwas schlechter im Vergleich zu einer bereits bekannten unteren Schranke in [14].

Sowohl die Bestimmung einer allgemeinen Formel für die Anzahl der $3 \times m$–Pläne als auch die Methoden zur Gewinnung der neuen Schranken basieren auf der Enumeration der $n \times m$–Ablaufgraphen bzw. der azyklischen Orientierungen des entsprechenden Hamming-Graphen $K_n \times K_m$. Die auf diese Weise erzielten Schranken können durch geeignete Ausnutzung der Eigenschaften der Ablaufgraphen eventuell noch verbessert werden.

Die Enumeration der azyklischen Orientierungen eines Graphen G steht in direktem Zusammenhang mit dem chromatischen Polynom von G. Zur Bestimmung der Plan-Anzahlen ist also das chromatische Polynom des Hamming-Graphen $K_n \times K_m$

grundlegend. Obwohl für das chromatische Polynom der Graphen des Typs $P_n \times K_m$ eine geschlossene Formel bekannt ist, bleibt das Problem der allgemeinen Bestimmung von $\chi(K_n \times K_m)$ bis heute ungelöst. Wenn gezeigt werden kann, daß dieses Problem $\#\mathcal{P}$-vollständig ist, so ist es auch das Problem der Bestimmung der Anzahl aller $n \times m$–Pläne.

Es ist nicht bekannt, ob über die Isomorphie zweier beliebigen Graphen in polynomialer Zeit entschieden werden kann, oder ob dieses Problem $\mathcal{NP}$-vollständig ist. In dieser Arbeit wird gezeigt, daß die Ablaufgraphen eine Klasse von Digraphen bilden, in der das Isomorphie-Problem polynomial lösbar ist. Dieses Ergebnis wird ausgenutzt, um Pläne mit gleichartiger Struktur zu identifizieren. Unter anderem auf dieser Grundlage wird ein Algorithmus zur Enumeration der $n \times m$–Pläne entwickelt.

Das in [15, 55] eingeführte Konzept der Irreduzibilität wird anhand neuer Terminologie und neuer Charakterisierungen irreduzibler Pläne erweitert. Darauf aufbauend sowie mit Hilfe der beschriebenen Enumeration aller $n \times m$–Pläne, wird ein Algorithmus zur Enumeration der irreduziblen Pläne entwickelt. Es stellt sich heraus, daß die Menge der irreduziblen Pläne keine minimale Menge potentiell-optimaler Pläne ist. Die Menge der irreduziblen $n \times m$–Pläne ist für gegebene n, m jedoch eindeutig bestimmt, während dies im Fall einer minimalen Menge von potentiell-optimalen Plänen nicht der Fall sein muß. Im Bereich der Bestimmung einer minimalen Menge von potentiell-optimalen Plänen bestehen noch vielfältige Forschungsaufgaben. Weiterhin ist bis heute unbekannt, ob das Entscheidungsproblem „Ist ein gegebener Plan irreduzibel?“ polynomial lösbar oder $\mathcal{NP}$-vollständig ist.

Die Enumerationsalgorithmen können für Probleme mit Vorrangbedingungen angepaßt werden, so daß mit Hilfe der Verhältnisse aus der Anzahl irreduzibler Pläne zur Anzahl der jeweils zulässigen Pläne Aussagen über die Schwierigkeit der Job-Shop-Probleme mit bestimmten Technologien getroffen werden können. Auf diese Weise lassen sich die praktischen Erfahrungen bezüglich der unterschiedlichen Dauer für das Berechnen einer optimalen Lösung verschiedener Job-Shop-Probleme theoretisch begründen (siehe [10]).

Es werden in dieser Arbeit Zusammenhänge zwischen der Irreduzibilität und der Stabilität von Plänen hergestellt. Da die Stabilität von Plänen für Algorithmen zur Lösung von Shop-Scheduling-Problemen mit nicht genau vorgegebenen Bearbeitungszeiten p_{ij} eine wesentliche Rolle spielen, könnte hier auch der Einsatz der entwickelten notwendigen Bedingungen für die Irreduzibilität eines Plans angewandt werden.

Wenn bei einem Shop-Scheduling-Problem für alle Bearbeitungszeiten nur jeweils untere und obere Schranken ($a_{ij} \leq p_{ij} \leq b_{ij}$) bekannt sind, ist also denkbar, die durch einen Lösungsalgorithmus entwickelten Pläne so weit wie möglich zu redu-

zieren, da diese Pläne im allgemeinen bezüglich Optimalität stabiler sind. Weitere Untersuchungen in diesem Bereich versprechen Ergebnisse für Shop-Scheduling-Probleme, bei denen Abweichungen von den Bearbeitungszeiten zugelassen sind. Solche Shop-Scheduling-Probleme sind bei der praktischen Anwendung offensichtlich von großem Interesse (vgl. LAI *et. al* [58]).

Anhang A

Symbolverzeichnis

$A \prec B$ ..... Plan B ist streng reduzierbar auf Plan B

$A \preceq B$ ..... Plan B ist reduzierbar auf Plan B

$A \sim B$ ..... Pläne A und B sind ähnlich

$A \cong B$ ..... Pläne A und B sind isomorph

$A \equiv B$ ..... Pläne A und B sind äquivalent

$A \equiv_S B$ ..... Pläne A und B sind struktur-äquivalent

$A_{n,m}$ ..... Anzahl der Äquivalenzklassen von $n \times m$–Plänen

$\alpha(G)$ ..... Anzahl der azyklischen Orientierungen des Graphen G

$\chi(G,k)$ ..... chromatisches Polynom des Graphen G

D_{2n} ..... Diedergruppe der Ordnung $2n$

$\det(B)$ ..... Determinante einer Matrix B

$G = (V,E)$ .. Graph bzw. Digraph mit der Knotenmenge V und der Menge E von Kanten bzw. gerichteten Kanten

$[G]$ ..... zugrunde liegende Graph eines Digraphen G

$G_1 \cup G_2$ .... Vereinigung zweier disjunkter Graphen G_1 und G_2

$G_1 + G_2$ .... Verbindung zweier disjunkter Graphen G_1 und G_2

$I_{n,m}$ ..... Anzahl der Isomorphieklassen von $n \times m$–Plänen

$I_{TR}(n,m)$ ... Anzahl der Isomorphieklassen von $n \times m$–Technologien

$K(G)$ ..... Kirchhoff-Matrix eines Graphen G

$l(G)$ ..... Kantengraph eines Graphen G

$\mathcal{L}_{n,m,r}$ ..... lateinisches Rechteck des Formats $n \times m$ mit der Belegungsmenge $\{1, \ldots, r\}$ (für $n \leq m \leq r$)

$L(n, m, r)$ ... Anzahl der lateinischen Rechtecke $\mathcal{L}_{n,m,r}$, deren Einträge aus der Menge $\{1, \ldots, r\}$ stammen

$\mathbb{N}$ ..... Menge der natürlichen Zahlen $\{1, 2, 3, \ldots\}$

$\#\mathcal{P}$ ..... Klasse aller Enumerationsprobleme, für die ein nichtdeterministischer polynomialer Lösungsalgorithmus existiert

$\mathcal{NP}$ ..... Klasse aller Entscheidungsprobleme, für die ein nichtdeterministischer polynomialer Lösungsalgorithmus existiert

$O(g(n))$ ..... Klasse aller Funktionen $f(n)$, für die eine Konstante $C > 0$ existiert, so daß $|f(n)| \leq C|g(n)|$ für alle $n \geq n_0$ ist.

$o(g(n))$ ..... Klasse aller Funktionen $f(n)$, bei denen für jedes $\varepsilon > 0$ ein $n_0(\varepsilon)$ existiert mit $|f(n)| \leq \varepsilon|g(n)|$ für alle $n \geq n_0(\varepsilon)$.

$o_{ij} \prec o_{kl}$ .... Operation o_{ij} wird vor o_{kl} bearbeitet

$o_{ij} \rightarrow o_{kl}$ .... Vorrangbedingung zwischen Operationen o_{ij} und o_{kl}

$\mathcal{P}$ ..... Klasse aller Entscheidungsprobleme, für die ein deterministischer polynomialer Lösungsalgorithmus existiert

$P(n)$ ..... Anzahl der rangminimalen $n \times n$–Pläne (= Anzahl lateinischer Quadrate der Ordnung n)

$P(n, m)$ .... Anzahl rangminimaler $n \times m$–Pläne

$P(n, m, r)$ .. Anzahl der $n \times m$–Pläne mit maximalem Rang r, $m \leq r \leq nm$

P_n ..... Weg mit n Knoten

$P_{n,m}$ ..... Anzahl der $n \times m$–Pläne

$P^*_{n,m}$ ..... Anzahl irreduzibler $n \times m$ Pläne

$\mathcal{P}_{n,m}$ ..... Menge der $n \times m$–Pläne bzw. der zulässigen Lösungen eines Open-Shop-Problems mit n Aufträgen und m Maschinen

$\mathcal{P}^*_{n,m}$ ..... Menge der irreduziblen $n \times m$–Pläne

$\mathcal{P}^F_{n,m}$ ..... Menge der $n \times m$–Pläne bezüglich eines Flow-Shop-Problems mit n Aufträgen und m Maschinen

$\mathcal{P}^G_{n,m}$ ..... Menge der $n \times m$–Pläne bezüglich eines General-Shop-Problems mit n Aufträgen und m Maschinen

$\mathcal{P}^J_{n,m}$ ..... Menge der $n \times m$–Pläne bezüglich eines Job-Shop-Problems mit n Aufträgen und m Maschinen

$\mathrm{per}(B)$ ..... Permanente einer Matrix B

$\varrho(v)$ ..... Rang eines Knotens v in einem Digraphen

S_n ..... Permutationsgruppe, bestehend aus den Permutationen der Zahlen $\{1, 2, \ldots, n\}$

$S_{n,m}$ ..... Anzahl der Struktur-Äquivalenzklassen von $n \times m$–Plänen

$S_{TR}(n, m)$ ... Anzahl der Struktur-Isomorphieklassen von $n \times m$–Technologien

T_n ..... Baum mit n Knoten

$\tau(G)$ ..... Anzahl der aufspannenden Bäume eines Graphen G

V_W ..... Menge der Knoten eines Weges W in einem (Di-)Graphen

$\mathbb{Z}_n$ ..... zyklische Gruppe der Ordnung n

Literaturverzeichnis

[1] S. B. Akers und J. Friedman, A non-numerical approach to production scheduling problems, *Oper. Res.* **3** (1955), 429–442.

[2] S. Ashour, *Sequencing Theory*, vol. 69 of Lecture Notes in Economical and Mathematical Systems, Springer, 1972.

[3] K. B. Athreya, C. R. Pranesachar, und N. M. Singhi, On the number of Latin rectangles and chromatic polynomial of $L(K_{r,s})$, *European J. Combin.* **1** (1980), 9–17.

[4] G. D. Birkhoff, A determinant formula for the number of ways of coloring a map, *Ann. Math.* **14** (1912), 42–46.

[5] K. P. Bogart, An obvious proof of Burnside's lemma, *Amer. Math. Monthly* **98**, 10 (1991), 927–928.

[6] K. P. Bogart und J. Q. Longyear, Counting 3 by n Latin rectangles, *Proc. Amer. Math. Soc.* **54** (1976), 463–467.

[7] H. Bräsel, *Lateinische Rechtecke und Maschinenbelegung*, Habilitationsschrift, Technische Universität "Otto von Guericke" Magdeburg, 1990.

[8] H. Bräsel, *Schedulingtheorie: Mathematische Modelle und Methoden*, Universität Kaiserslautern, Fachbereich Mathematik, 1996. Vorlesungsskript, Wintersemester 1995/1996.

[9] H. Bräsel, L. Dornheim, M. Harborth, T. Tautenhahn, I. Wasmund, P. Willenius, und A. Winkler, LiSA – A Library of Scheduling Algorithms. Dynamic Survey, http://fma2.math.uni-magdeburg.de/~lisa/.

[10] H. Bräsel, M. Harborth, T. Tautenhahn, und P. Willenius, On the hardness of the classical job shop problem. To appear in *Ann. Oper. Res.*

[11] H. Bräsel, M. Harborth, T. Tautenhahn, und P. Willenius, On the set of solutions of an open shop problem. To appear in *Ann. Oper. Res.*

[12] H. Bräsel, M. Harborth, und P. Willenius, Isomorphism for digraphs and sequences of shop scheduling problems. To appear in *J. Combin. Math. Combin. Comput.*

[13] H. Bräsel und M. Kleinau, On number problems for the open shop problem, in *System Modelling and Optimization, Proc. 15th IFIP Conf.*, P. Kall, ed., vol. 180 of Lecture Notes in Control and Inform. Sci., Berlin, 1992, Springer, 145–154.

[14] H. Bräsel und M. Kleinau, On the number of feasible schedules of the open-shop-problem – an application of special Latin rectangles, *Optimization* **23** (1992), 251–260.

[15] H. Bräsel und M. Kleinau, New steps in the amazing world of sequences and schedules, *Math. Methods Oper. Res.* **43** (1996), 195–214.

[16] H. Bräsel, Y. N. Sotskov, und F. Werner, Stability of a schedule minimizing mean flow time, *Math. Comput. Modelling* **24**, 10 (1996), 39–53.

[17] J. W. Brown, Enumeration of Latin squares with application to order 8, *J. Combin. Theory Ser. B.* **5** (1968), 177–184.

[18] P. Brucker, *Scheduling Algorithms*, Springer, Berlin, 1995.

[19] P. Brucker, B. Jurisch, und M. Jurisch, Open shop problems with unit time operations, *Z. Oper. Res.* **37** (1993), 59–73.

[20] P. Brucker und S. Knust, Complexity results of scheduling problems. Dynamic Survey, http://www.mathematik.uni-osnabrueck.de/research/OR/class/.

[21] W. Burnside, *Theory of Groups of Finite Order*, University Press, Cambridge, 1897.

[22] J. Chen, A linear-time algorithm for isomorphism of graphs of bounded average genus, *SIAM J. Discrete Math.* **7**, 4 (1994), 614–631.

[23] G. L. Chia, Some problems on chromatic polynomials, *Discrete Math.* **172**, 1-3 (1997), 39–44. Chromatic polynomials and related topics (Shanghai, 1994).

[24] W. Clark, *The Gantt Chart*, Pitman and Sons, London, 3rd ed., 1952.

[25] C. J. Colbourn, The complexity of completing partial Latin squares, *Discrete Appl. Math.* **8**, 1 (1984), 25–30.

[26] R. W. Conway, W. L. Maxwell, und L. W. Miller, *Theory of Scheduling*, Addison-Wesley, Reading, MA, 1967.

[27] S. A. Cook, The complexity of theorem-proving procedures, in *Proc. 3rd Annual ACM Symp. Theory of Computing*, 1971, 151–158.

[28] N. G. de Bruijn, A note on the Cauchy-Frobenius lemma, *Indag. Math.* **41** (1979), 225–228.

[29] J. Dénes und A. D. Keedwell, *Latin Squares and their Applications*, Academic Press, New York and London, 1974.

[30] J. Dénes und A. D. Keedwell, *Latin Squares: New Developments in the Theory and Applications*, vol. 46 of Ann. Discrete Math., North-Holland, Amsterdam, 1991.

[31] K. Dohmen, Lower bounds and upper bounds for chromatic polynomials, *J. Graph Theory* **17**, 1 (1993), 75–80.

[32] K. Dohmen, Bounds to the chromatic polynomial of a graph, *Results Math.* **33**, 1-2 (1998), 87–88.

[33] P. Erdös und I. Kaplansky, The asymptotic number of Latin rectangles, *Amer. J. Math.* **68** (1946), 230–236.

[34] L. Euler, Recherches sur une nouvelle espèce de quarrés magiques, *Ges. Werke* **7** (1782), 291–392.

[35] S. French, *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Horwood, Chichester, 1982.

[36] M. R. Garey und D. S. Johnson, *Computers and Intractability - a Guide to the Theory of NP-Completeness*, W. H. Freeman & Co, New York, 1979.

[37] W. Goddard, C. Kenyon, V. King, und L. J. Schulman, Optimal randomized algorithms for local sorting and set-maxima, *SIAM J. Comput.* **22**, 2 (1993), 272–283.

[38] C. D. Godsil und B. D. McKay, Asymptotic enumeration of Latin rectangles, *J. Combin. Theory Ser. B* **48** (1990), 19–44.

[39] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Comput. Sci. Appl. Math., Academic Press, New York, 1980.

[40] A. Goralčíková und V. Koubek, A reduct-and-closure algorithm for graphs, in *Mathematical Foundations of Computer Science. (Proc. Eighth Sympos., Olomouc, 1979)*, J. Becvar, ed., vol. 74 of Lect. Notes Comput. Sci., Berlin - New York, 1979, Springer, 301–307.

[41] R. L. Graham, E. L. Lawler, J. K. Lenstra, und A. H. G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, *Ann. Discrete Math.* **5** (1979), 287–326.

[42] T. A. Green, Asymptotic enumeration of generalized Latin rectangles, *J. Combin. Theory Ser. A* **51**, 2 (1989), 149–160.

[43] M. Habib, M. Morvan, und J.-X. Rampon, On the calculation of transitive reduction-closure of orders, *Discrete Math.* **111** (1993), 289–303.

[44] F. Harary, *Graphentheorie*, Oldenbourg, München, 1974. Germ. Transl. of "Graph Theory", Addison-Wesley, Reading, 1969.

[45] J. E. Hopcroft und R. M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* **2** (1973), 225–231.

[46] J. E. Hopcroft und J. K. Wong, Linear time algorithm for isomorphism of planar graphs, in *Proc. 6th ann. ACM Symp. Theory Comput.*, 1974, 172–184.

[47] W. Imrich und S. Klavžar, On the complexity of recognition Hamming graphs and related classes of graphs, *European J. Combin.* **17**, 2/3 (1996), 209–221.

[48] S. M. Jacob, The enumeration of the Latin rectangle of depth three, *Amer. J. Math.* **31** (1930), 329–354.

[49] N. Kahale und L. J. Schulman, Bounds on the chromatic polynomial and on the number of acyclic orientations of a graph, *Combinatorica* **16**, 3 (1996), 383–397.

[50] A. B. Kahn, Topological sorting of large networks, *Comm. ACM* **5** (1962), 558–562.

[51] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, R. E. Miller und J. W. Thatcher, eds., New York, 1972, Plenum, 85–103.

[52] S. M. Kerawala, The enumeration of the Latin rectangle of depth three by means of difference equation, *Bull. Calcutta Math. Soc.* **33** (1941), 119–127.

[53] S. M. Kerawala, The asymptotic number of three-deep Latin rectangles, *Bull. Calcutta Math. Soc.* **39** (1947), 71–72.

[54] G. Kirchhoff, Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird, *Ann. Phys. Chem.* **72** (1847), 497–508.

[55] M. Kleinau, *Zur Struktur von Shop-Scheduling-Problemen: Anzahlprobleme, Reduzierbarkeit und Komplexität*, Dissertation, Technische Universität "Otto von Guericke" Magdeburg, 1993.

[56] U. Kleinau, *Zur Struktur und Lösung verallgemeinerter Shop-Scheduling-Probleme*, Dissertation, Technische Universität "Otto von Guericke" Magdeburg, 1993.

[57] R. E. Ladner, On the structure of polynomial time reducibility, *J. Assoc. Comput. Mach.* **22** (1975), 155–171.

[58] T.-C. Lai, Y. N. Sotskov, N. Y. Sotskova, und F. Werner, Optimal makespan scheduling with given bounds of processing times, *Math. Comput. Modelling* **26**, 3 (1997), 67–86.

[59] D. J. Lasser, Topological sorting of a list of randomly-numbered elements of a network, *Comm. ACM* **4** (1961), 12.

[60] C. F. Laywine und G. L. Mullen, *Discrete mathematics using Latin squares*, John Wiley & Sons Inc., New York, 1998. A Wiley-Interscience Publication.

[61] J. Light, F. W., A procedure for the enumeration of $4 \times n$ Latin rectangles, *Fibonacci Quart.* **11**, 3 (1973), 241–246.

[62] N. Linial, Hard enumeration problems in geometry and combinatorics, *SIAM J. Algebraic Discrete Methods* **7**, 2 (1986), 331–335.

[63] G. S. Lueker und K. S. Booth, A linear time algorithm for deciding interval graph isomorphism, *J. Assoc. Comput. Mach.* **26**, 2 (1979), 183–195.

[64] E. M. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comput. Syst. Sci.* **25**, 1 (1982), 42–65.

[65] R. M. McConnell und J. P. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, in *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994)*, New York, 1994, ACM, 536–545.

[66] R. M. McConnell und J. P. Spinrad, Modular decomposition and transitive orientation, Preprint-Reihe Mathematik 475, Technische Universität Berlin, Fachbereich 3, 1995.

[67] R. M. McConnell und J. P. Spinrad, Linear-time transitive orientation, in *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (New Orleans, LA, 1997)*, New York, 1997, ACM, 19–25.

[68] B. D. McKay und E. Rogoyski, Latin squares of order 10, *Electron. J. Combin.* **2** (1995), Note 3, approx. 4 pp. (electronic).

[69] B. D. McKay und I. M. Wanless, Maximising the permanent of $(0, 1)$-matrices and the number of extensions of Latin rectangles, *Electron. J. Combin.* **5**, 1 (1998), Research Paper 11, 20 pp. (electronic).

[70] G. L. Miller, Graph isomorphism, general remarks, *J. Comput. System Sci.* **18** (1979), 128–142.

[71] G. L. Mullen und D. Purdy, Some data concerning the number of Latin rectangles, *J. Combin. Math. Combin. Comput.* **13** (1993), 161–165.

[72] J. R. Nechvatal, Asymptotic enumeration of generalized Latin rectangles, *Utilitas Math.* **20** (1981), 273–292.

[73] P. M. Neumann, A lemma that is not Burnside's, *Math. Sci.* **4** (1979), 133–141.

[74] I. N. Ponomarenko, Polynomial time algorithms for recognizing and isomorphism testing of cyclic tournaments, *Acta Appl. Math.* **29**, 1/2 (1992), 139–160.

[75] D. B. Porter, The Gantt chart as applied to production scheduling and control, *Naval Res. Logist. Quart.* **15** (1968), 311–317.

[76] C. R. Pranesachar, Enumeration of Latin rectangles via SDR's, in *Combinatorics and Graph Theory*, S. B. Rao, ed., vol. 885 of Lecture Notes in Math., Springer, Berlin, 1981, 380–390.

[77] R. C. Read und W. T. Tutte, Chromatic polynomials, in *Selected topics in graph theory, 3*, Academic Press, San Diego, CA, 1988, 15–42.

[78] L. Rédei, Ein kombinatorischer Satz, *Acta Litt. Sci. Szeged* **7** (1934), 39–43.

[79] M. Rezaie, Chromatic polynomial of Cartesian product of graphs, in *Proceedings of the 28th Annual Iranian Mathematics Conference, Part 1 (Tabriz, 1997)*, Tabriz Univ., Tabriz, 1997, 447–450.

[80] J. Riordan, Three-line Latin rectangles, *Amer. Math. Monthly* **51** (1944), 450–452.

[81] J. Riordan, Three-line Latin rectangles II, *Amer. Math. Monthly* **53** (1946), 18–20.

[82] J. Riordan, A recurrence relation for three-line Latin rectangles, *Amer. Math. Monthly* **59** (1952), 159–162.

[83] G.-C. Rota, On the foundations of combinatorial theory. I: Theory of Möbius functions, *Z. Wahrsch. Verw. Gebiete* **2** (1964), 340–368.

[84] B. Roy und B. Sussmann, Les problèmes d'ordonnancement avec contraintes disjonctives, Note DS No.9 bis, Montrouge, 1964.

[85] J. Y. Shao und W. D. Wei, A formula for the number of Latin squares, *Discrete Math.* **110**, 1-3 (1992), 293–296.

[86] K. Simon, An improved algorithm for transitive closure on acyclic digraphs, *Theoret. Comput. Sci.* **58** (1988), 325–346.

[87] K. Simon, *Effiziente Algorithmen für perfekte Graphen*, Teubner, Stuttgart, 1992.

[88] I. Skau, A note on the asymptotic number of Latin rectangles, *European J. Combin.* **19**, 5 (1998), 617–620.

[89] Y. N. Sotskov, Stability of an optimal schedule, *European J. Oper. Res.* **55** (1991), 91–102.

[90] Y. N. Sotskov, V. S. Tanaev, und F. Werner, Stability radius of an optimal schedule: a survey and recent developments, in *Industrial applications of combinatorial optimization*, Kluwer Acad. Publ., Dordrecht, 1998, 72–108.

[91] J. Spinrad, On comparability and permutation graphs, *SIAM J. Comput.* **14**, 3 (1985), 658–670.

[92] R. P. Stanley, Acyclic orientations of graphs, *Discrete Math.* **5** (1973), 171–178.

[93] C. M. Stein, Asymptotic evaluation of the number of Latin rectangles, *J. Combin. Theory Ser. A* **25** (1978), 38–49.

[94] V. S. Tanaev, Y. N. Sotskov, und V. A. Strusevich, *Scheduling theory. Multi-stage systems*, Kluwer Academic Publishers Group, Dordrecht, 1994. Translated and revised from the 1989 Russian original by the authors.

[95] T. Tautenhahn, *Open-Shop-Probleme mit Einheitsbearbeitungszeiten*, Dissertation, Otto-von-Guericke-Universität Magdeburg, 1993.

[96] T. Tautenhahn, Irreducible sequences - an approach to interval edge colouring trees, Preprint No. OR83, Faculty of Mathematical Studies, University of Southampton, 1996.

[97] T. Tautenhahn und P. Willenius, Irreducibility and unavoidability of sequences, Preprint, Otto-von-Guericke-Universität Magdeburg, 1999. To appear.

[98] J. Valdes, R. E. Tarjan, und E. L. Lawler, The recognition of series parallel digraphs, *SIAM J. Comput.* **11**, 2 (1982), 298–313.

[99] L. G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* **8**, 2 (1979), 189–201.

[100] L. G. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* **8**, 3 (1979), 410–421.

[101] J. H. van Lint und R. M. Wilson, *A Course in Combinatorics*, University Press, Cambridge, 1992, ch. 17, 157–171.

[102] K. P. Vo, Graph colorings and acyclic orientations, *Linear and Multilinear Algebra* **22**, 2 (1987), 161–170.

[103] E. M. Wright, Burnside's lemma: A historical note, *J. Combin. Theory Ser. B.* **30** (1981), 89–90.

[104] K. Yamamoto, An asymptotic series for the number of three-line Latin rectangles, *J. Math. Soc. Japan* **1** (1949), 226–241.

[105] K. YAMAMOTO, On the asymptotic number of Latin rectangles, *Japan. J. Math.* **21** (1951), 113–119.

Index

Lebenslauf

Martin Harborth, geboren am 21. Oktober 1968 in Braunschweig (Deutschland).

Schulbesuch

1975-1981	Grundschule Schuntersiedlung und Orientierungsstufe Nibelungenschule, Braunschweig
1981-1988	Gymnasium Martino-Katharineum, Braunschweig
1988	Abiturprüfung bestanden am 18. Mai

Studium

WS 1988/89	Beginn des Studiums an der Technischen Universität Braunschweig, Studiengang Mathematik und Chemie für das höhere Lehramt
WS 1989/90	Wechsel zum Studiengang Mathematik-Diplom, Nebenfach Informatik
1994	Diplomprüfung mit der Gesamtnote „mit Auszeichnung“ abgelegt; TU Braunschweig, 24. Oktober

Studentische Nebentätigkeiten

1990-1992	Betreuung von Übungen als Hilfsassistent; TU Braunschweig
1992-1995	Entwicklung und Wartung von Datenbanken und Fachinformationssystemen in der Bibliothek der mathematischen Institute; TU Braunschweig
seit 1995	Verwaltung und Betreuung der WWW-Internet-Präsentation der Fakultät für Mathematik, Pflege der Datenbank der Fakultätsbibliothek; Otto-von-Guericke-Universität Magdeburg, seit dem 1. Juli

Promotion

1995-1997	Stipendium zur Graduiertenförderung des Landes Sachsen-Anhalt; Otto-von-Guericke-Universität Magdeburg, 1. April 1995 bis 30. September 1997
seit 1997	Wissenschaftlicher Mitarbeiter im Rahmen des vom Land Sachsen-Anhalt finanzierten Projekts „Lateinische Rechtecke in der Schedulingtheorie“; Otto-von-Guericke-Universität Magdeburg, seit dem 1. Oktober

On Infinite Families of Sequences with One and Two Valued Autocorrelation and Two Valued Crosscorrelation Function

Marc Gysin[1] and Jennifer Seberry[2] *

[1] School of Information Technology
James Cook University
Townsville, QLD 4811
Australia
e-mail: marc@cs.jcu.edu.au
[2] Centre for Computer Security Research
School of Information Technology and Computer Science
University of Wollongong
Wollongong, NSW 2500
Australia
e-mail: jennie@uow.edu.au

Abstract. We show how to construct infinite families of sequences that have one and two valued autocorrelation and two valued crosscorrelation function. These sequences are obtained via the discrete Fourier transform of integer sequences. The sequences obtained can be complex valued or having entries $\in \{0, 1, \ldots, p\}$, p prime, depending on the construction used.

1 Introduction

We shall make use of the following notations: (i) $\mathcal{Z}$, $\mathcal{R}$ and $\mathcal{C}$ will denote the integers, real numbers and complex numbers, respectively; (ii) if $a \in \mathcal{C}$ then a^* is its complex conjugate and $Re(a)$ and $Im(a)$ denote its real and imaginatory part, respectively; (iii) when talking about a sequence of length ℓ, subscripts are to be taken reduced modulo ℓ.

Let $\mathcal{S}$ be a set and let $X = \{x_0, \ldots, x_{\ell-1}\}$ be a sequence where $x_i \in \mathcal{S}$, for $i = 0, \ldots, \ell - 1$. We call X a *binary sequence* or *ternary sequence* if $\mathcal{S} = \{-1, 1\}$ or $\mathcal{S} = \{-1, 0, 1\}$, respectively.

The *periodic autocorrelation function* $P_X(s)$, of a sequence X with shift s is defined as:

$$P_X(s) = \sum_{i=0}^{\ell-1} x_i x_{i+s}.^{1}$$

We are interested in one or two binary or ternary sequence(s) X or X, Y such that

$$P_X(s) = c \text{ or } P_X(s) + P_Y(s) = c, \quad s = 1, \ldots, \ell - 1 \tag{1}$$

If we let $w = P_X(0)$ or $w = P_X(0) + P_Y(0)$ and $w \neq c$ then we say that the periodic autocorrelation function of X or X and Y is *two valued*. If $w = c$ then we say the periodic autocorrelation function is *one valued*.

The *periodic crosscorrelation function* $C_{X,Y}(s)$ of two sequences X, Y with shift s is defined as:

$$C_{X,Y}(s) = \sum_{i=0}^{\ell-1} x_i y_{i+s}$$

* Research supported by Large ARC Grants A9803826, A49703117 and a small ARC Grant. This paper has been written while the first author was at the University of Wollongong.

[1] In the terms of the above sum, the second factor is *not* the complex conjugate as seen in many definitions of the periodic autocorrelation function. The reason why we do not take the complex conjugate is to keep definitions consistent throughout this paper which otherwise would not be possible.

Note that for $s \neq 0$ generally $C_{X,Y}(s) \neq C_{Y,X}(s)$. If

$$C_{X,Y}(s) = c, \quad s = 1, \ldots, \ell - 1 \tag{2}$$

and $C_{X,Y}(0) = w$, then we say that the periodic crosscorrelation function of X and Y is *one valued* or *two valued*, if $w = c$ or $w \neq c$, respectively.

Binary or ternary sequences satisfying (1) or (2) play an important role in communication and combinatorial design theory, [GavLem94], [GerSeb79], [Paterson98], [SebYam92]. Unfortunately, such sequences are hard to find for larger lengths ℓ.

We generalise and let $\mathcal{S} = \mathcal{C}$, or $\mathcal{S} = \{0, \ldots, p-1\}$, where p is a prime and show how to construct infinite families of sequences having properties (1) and (2) for any length ℓ. If $\mathcal{S} = \mathcal{C}$, all the calculations are to be done in the field of complex numbers, whereas for $\mathcal{S} = \{0, \ldots, p-1\}$ all the calculations are in the field $GF(p)$.

2 The Constructions

Let $\mathcal{S}_1 = \mathcal{Z}$, where $\mathcal{Z}$ are the integers and let $\mathcal{S}_2 = \mathcal{C}$. We start with an integer sequence $A = \{a_0, \ldots, a_{\ell-1}\}$, $a_k \in \mathcal{S}_1$ and we let $X = \{x_0, \ldots, x_{\ell-1}\}$, $Y = \{y_0, \ldots, y_{\ell-1}\}$ where

$$x_k = \sum_{j=0}^{\ell-1} a_j e^{2\pi ijk/\ell}, \quad y_k = \sum_{j=0}^{\ell-1} a_j e^{-2\pi ijk/\ell} \tag{3}$$

and $i^2 = -1$. Observe that x_k is the k–th element of the discrete Fourier transform of the sequence A and $x_k = x^*_{-k} = y^*_k$. Also $x_k, y_k \in \mathcal{S}_2$.

We first prove:

Lemma 1. *Altering the sign of one element of A does not affect the periodic crosscorrelation function of X and Y. More precisely, let A and $\tilde{A}$ be two integer sequences such that $\tilde{a}_p = -a_p$ for some $p \in \{0, \ldots, \ell-1\}$ and $\tilde{a}_k = a_k$ for all other elements. Let X, Y and $\tilde{X}$, $\tilde{Y}$ be the sequences obtained from A and $\tilde{A}$, respectively according to (3). Then*

$$C_{\tilde{X},\tilde{Y}}(s) = C_{X,Y}(s) \text{ and } C_{\tilde{Y},\tilde{X}}(s) = C_{Y,X}(s) \tag{4}$$

for all $s = 0, \ldots, \ell - 1$.

Proof. For symmetry reasons it is sufficient to prove $C_{X,Y}(s) = C_{\tilde{X},\tilde{Y}}(s)$. Consider

$$\Delta_s = C_{\tilde{X},\tilde{Y}}(s) - C_{X,Y}(s).$$

We have

$$\Delta_s = \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} \tilde{a}_j \tilde{a}_u e^{2\pi i(jk-uk-us)/\ell} - \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} a_j a_u e^{2\pi i(jk-uk-us)/\ell}$$

which is

$$-2a_p\Big(\sum_{k=0}^{\ell-1}\sum_{u=0,u\neq p}^{\ell-1} a_u e^{2\pi i(pk-uk-us)/\ell} + \sum_{k=0}^{\ell-1}\sum_{j=0,j\neq p}^{\ell-1} a_j e^{2\pi i(jk-pk-ps)/\ell}\Big)$$

because of the construction of $\tilde{A}$ and A. We exchange the two sum–operators and obtain

$$-2a_p\Big(\sum_{u=0,u\neq p}^{\ell-1}\sum_{k=0}^{\ell-1} a_u e^{2\pi ik(p-u)/\ell} e^{-2\pi ius/\ell} + \sum_{j=0,j\neq p}^{\ell-1}\sum_{k=0}^{\ell-1} a_j e^{2\pi ik(j-p)/\ell} e^{-2\pi ips/\ell}\Big)$$

which can be written as

$$-2a_p\Big(\sum_{u=0,u\neq p}^{\ell-1} a_u e^{-2\pi ius/\ell}\sum_{k=0}^{\ell-1} e^{2\pi ik(p-u)/\ell} + \sum_{j=0,j\neq p}^{\ell-1} a_j e^{-2\pi ips/\ell}\sum_{k=0}^{\ell-1} e^{2\pi ik(j-p)/\ell}\Big)$$

Because $u \neq p \neq j$ the two innermost sums both evaluate to zero. Therefore, all the summations are over zero and $\Delta_s = 0$. Because no specifications have been made about s, $\Delta_s = 0$, for all $s \in \{0, \ldots, \ell-1\}$.

The above lemma allows us to prove the following:

Theorem 1. *Let $A = \{a_0, \ldots, a_{\ell-1}\}$ be any integer sequence such that $a_0 = |a|$ and $|a_1| = |a_2| = \ldots = |a_{\ell-1}| = b$. Then*

$$C_{X,Y}(0) = \ell(a^2 - b^2) + \ell^2 b^2$$
$$C_{X,Y}(s) = \ell(a^2 - b^2), \quad s = 1, \ldots, \ell-1$$

and the same is true for $C_{Y,X}(s)$.

In other words, the periodic crosscorrelation function of X and Y is two valued with values $\ell(a^2 - b^2) + \ell^2 b^2$ and $\ell(a^2 - b^2)$, respectively.

Proof. Because of Lemma 1 we are allowed to assume that $a_0 = a$, $a_1 = a_2 = \ldots = a_{\ell-1} = b$. Consider now $C_{X,Y}(s)$ for $s \neq 0$. We have

$$\begin{aligned} C_{X,Y}(s) &= \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} a_j a_u e^{2\pi i(jk-uk-us)/\ell} \\ &= \sum_{k=0}^{\ell-1}\Big(\sum_{u=1}^{\ell-1} abe^{-2\pi iu(k+s)/\ell} + \sum_{j=1}^{\ell-1} abe^{2\pi ijk/\ell} + a^2\Big) + \sum_{k=0}^{\ell-1}\sum_{j=1}^{\ell-1}\sum_{u=1}^{\ell-1} b^2 e^{2\pi i(jk-uk-us)/\ell} \end{aligned}$$

We first evaluate the two leftmost sums.

$$\begin{aligned} &\sum_{k=0}^{\ell-1}\Big(\sum_{u=1}^{\ell-1} abe^{-2\pi iu(k+s)/\ell} + \sum_{j=1}^{\ell-1} abe^{2\pi ijk/\ell} + a^2\Big) \\ &= \ell a^2 - 2\ell ab + \sum_{k=0}^{\ell-1}\Big(\sum_{u=0}^{\ell-1} abe^{-2\pi iu(k+s)/\ell} + \sum_{j=0}^{\ell-1} abe^{2\pi ijk/\ell}\Big) \end{aligned}$$

The two innermost sums of this expressions evaluate to zero, except for the case $k = -s$ and $k = 0$, respectively. In this case both innermost sums evaluate to ℓab. Therefore,

$$\begin{aligned} &\sum_{k=0}^{\ell-1}\Big(\sum_{u=1}^{\ell-1} abe^{-2\pi iu(k+s)/\ell} + \sum_{j=1}^{\ell-1} abe^{2\pi ijk/\ell} + a^2\Big) \\ &= \ell a^2 - 2\ell ab + (\ell-1)(0+0) + \ell ab + \ell ab = \ell a^2. \end{aligned}$$

The rightmost sum is:

$$\begin{aligned} &\sum_{k=0}^{\ell-1}\sum_{j=1}^{\ell-1}\sum_{u=1}^{\ell-1} b^2 e^{2\pi i(jk-uk-us)/\ell} \\ &= \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} b^2 e^{2\pi i(jk-uk-us)/\ell} - \sum_{k=0}^{\ell-1}\Big(\sum_{u=1}^{\ell-1} b^2 e^{-2\pi iu(k+s)/\ell} + \sum_{j=1}^{\ell-1} b^2 e^{2\pi ijk/\ell} + b^2\Big) \end{aligned}$$

Now similarly to the above

$$\sum_{k=0}^{\ell-1}(\sum_{u=1}^{\ell-1} b^2 e^{-2\pi iu(k+s)/\ell} + \sum_{j=1}^{\ell-1} b^2 e^{2\pi ijk/\ell} + b^2) = \ell b^2$$

and it can be shown (using similar techniques) that for $s \neq 0$

$$\sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} b^2 e^{2\pi i(jk-uk-us)/\ell} = 0$$

Therefore, the rightmost sum is $-\ell b^2$ and $C_{X,Y}(s) = \ell(a^2 - b^2)$, $s \neq 0$. It remains to consider $C_{X,Y}(0)$. Since no assumptions have been made about s except for

$$\sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} b^2 e^{2\pi i(jk-uk-us)/\ell} = 0$$

we know

$$C_{X,Y}(0) = \ell(a^2 - b^2) + \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} b^2 e^{2\pi i(jk-uk)/\ell}$$

But

$$\sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} b^2 e^{2\pi i(jk-uk)/\ell} = b^2 \sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1}\sum_{k=0}^{\ell-1} e^{2\pi ik(j-u)/\ell} = b^2\ell^2$$

because the innermost sum vanishes except for $j = u$ when it assumes the value ℓ. Therefore, $C_{X,Y}(0) = \ell(a^2 - b^2) + \ell^2 b^2$.

Corollary 1. *Let the sequence X be as in Theorem 1 and write $x_k = r_k + iw_k$. Let R, W be the sequences $\{r_0, \ldots, r_{\ell-1}\}$ and $\{w_0, \ldots, w_{\ell-1}\}$, respectively. Then*

$$P_R(s) + P_W(s) = \begin{cases} \ell(a^2 - b^2) + \ell^2 b^2 & s = 0 \\ \ell(a^2 - b^2) & s = 1, \ldots, \ell - 1 \end{cases}$$

and

$$C_{R,W}(s) = C_{W,R}(s),$$

for $s = 0, \ldots, \ell - 1$.

Proof. Consider $C_{X,Y}(s) = \sum_{k=0}^{\ell-1}(r_k + iw_k)(r_{k+s} - iw_{k+s}) = P_R(s) + P_W(s) + i(C_{W,R}(s) - C_{R,W}(s))$.

We now show how to construct sequences with one valued periodic autocorrelation function.

Lemma 2. *Let A be an integer sequence satisfying for $k \neq 0$, $a_k \neq 0 \Longrightarrow a_{-k} = 0$. Then "deleting" one element not at the beginning of A does not affect the periodic autocorrelation function of X and Y. More precisely, let A be as above and let $\tilde{A}$ be an integer sequences such that $\tilde{a}_p = 0$ for some $p \neq 0$ where $a_p \neq 0$ and $\tilde{a}_k = a_k$ for all other elements. Let X, Y and $\tilde{X}$, $\tilde{Y}$ be the sequences obtained from A and $\tilde{A}$, respectively according to (3). Then*

$$P_{\tilde{X}}(s) = P_X(s) \text{ and } P_{\tilde{Y}}(s) = P_Y(s) \tag{5}$$

for all $s = 0, \ldots, \ell - 1$.

Proof. Consider

$$\Delta_s = P_X(s) - P_{\tilde{X}}(s).$$

We have

$$\begin{aligned}
\Delta_s &= \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} a_j a_u e^{2\pi i(jk+uk+us)/\ell} - \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} \tilde{a}_j \tilde{a}_u e^{2\pi i(jk+uk+us)/\ell} \\
&= \sum_{k=0}^{\ell-1}\Big(\sum_{u=0,u\neq p}^{\ell-1} a_p a_u e^{2\pi i(pk+uk+us)/\ell} + \sum_{j=0,j\neq p}^{\ell-1} a_j a_p e^{2\pi i(jk+pk+ps)/\ell} + a_p^2 e^{2\pi ip(2k+s)/\ell}\Big) \\
&= \sum_{u=0,u\neq p}^{\ell-1} a_p a_u \sum_{k=0}^{\ell-1} e^{2\pi i(pk+uk+us)/\ell} + \sum_{j=0,j\neq p}^{\ell-1} a_j a_p \sum_{k=0}^{\ell-1} e^{2\pi i(jk+pk+ps)/\ell} + a_p^2 \sum_{k=0}^{\ell-1} e^{2\pi ip(2k+s)/\ell}
\end{aligned}$$

Consider now the three sums:

$$\begin{aligned}
\sum_{k=0}^{\ell-1} e^{2\pi i(pk+uk+us)/\ell} &= e^{2\pi ius/\ell} \sum_{k=0}^{\ell-1} e^{2\pi i(k(p+u))/\ell} \\
\sum_{k=0}^{\ell-1} e^{2\pi i(jk+pk+ps)/\ell} &= e^{2\pi ips/\ell} \sum_{k=0}^{\ell-1} e^{2\pi i(k(j+p))/\ell} \\
\sum_{k=0}^{\ell-1} e^{2\pi i(p(2k+s))/\ell} &= e^{2\pi ips/\ell} \sum_{k=0}^{\ell-1} e^{2\pi i(k2p)/\ell}
\end{aligned}$$

The last of the three sums is zero because $p \neq 0$. The first and second sum vanish if and only if $u \neq -p$ and $j \neq -p$, respectively. But if $u = -p$ then either $a_p = 0$ or $a_u = 0$ by the assumption about A. Hence,

$$a_p a_u \sum_{k=0}^{\ell-1} e^{2\pi i(pk+uk+us)/\ell} = 0$$

Similarly for the second sum we always have

$$a_j a_p \sum_{k=0}^{\ell-1} e^{2\pi i(jk+pk+ps)/\ell} = 0$$

Therefore $\Delta_s = 0$, for $s = 0, \ldots, \ell - 1$.

The above lemma allows us to prove the following:

Theorem 2. *Let $A = \{a_0, \ldots, a_{\ell-1}\}$ be any integer sequence such that for $k \neq 0$, $a_k \neq 0 \Longrightarrow a_{-k} = 0$. Then*

$$\begin{aligned}
P_X(s) &= \ell a_0^2 \\
P_Y(s) &= \ell a_0^2
\end{aligned}$$

for $s = 0, \ldots, \ell - 1$.

In other words, the periodic autocorrelation function of X or Y is one valued with value ℓa_0^2.

Proof. Because of Lemma 2 we are allowed to assume $a_1 = a_2 = \ldots = a_{\ell-1} = 0$. Now

$$P_X(s) = \sum_{k=0}^{\ell-1}\sum_{j=0}^{\ell-1}\sum_{u=0}^{\ell-1} a_j a_u e^{2\pi i(jk+uk+us)/\ell} = \sum_{k=0}^{\ell-1} a_0^2 = \ell a_0^2.$$

Corollary 2. *Let the sequence X be as in Theorem 2 and write $x_k = r_k + iw_k$. Let R, W be the sequences $\{r_0, \ldots, r_{\ell-1}\}$ and $\{w_0, \ldots, w_{\ell-1}\}$, respectively. Then*

$$P_R(s) - P_W(s) = \ell a^2$$

and

$$C_{R,W}(s) + C_{W,R}(s) = 0,$$

for $s = 0, \ldots, \ell - 1$.

Proof. Consider $P_X(s) = \sum_{k=0}^{\ell-1}(r_k + iw_k)(r_{k+s} + iw_{k+s}) = P_R(s) - P_W(s) + i(C_{W,R}(s) + C_{R,W}(s))$.

Special Constructions

The first construction can be enhanced by putting additional conditions on the integer sequence A. We describe this in the following lemma.

Lemma 3. *Let A be an integer sequence satisfying the conditions of Theorem 1. Then*

(i) if in addition $a_k = a_{-k}$ we have

$$P_X(0) = P_Y(0) = \ell(a^2 - b^2) + \ell^2 b^2$$
$$P_X(s) = P_Y(s) = \ell(a^2 - b^2), \quad s = 1, \ldots, \ell - 1$$

(ii) if in addition $a_k = -a_{-k}$ and ℓ is odd then

$$P_X(0) = P_Y(0) = \ell(a^2 + b^2) - \ell^2 b^2$$
$$P_X(s) = P_Y(s) = \ell(a^2 + b^2), \quad s = 1, \ldots, \ell - 1$$

Observe that we now have sequences X or Y with two valued periodic autocorrelation function.

Proof. The proof of (i) is very simple. Because of $a_k = a_{-k}$, X and Y are both real valued. Also $x_k = y_k = x_{-k} = y_{-k}$. That is, $X = Y$, and so, $C_{X,Y}(s) = C_{Y,X}(s) = P_X(s) = P_Y(s)$, for $s = 0, \ldots, \ell - 1$. For (ii) let $w_k = Im(x_k)$. By construction $x_k = a + iw_k$. From Theorem 1 we know that

$$C_{X,Y}(s) = \sum_{k=0}^{\ell-1} x_k y_{k+s} = \ell a^2 + \sum_{k=0}^{\ell-1} w_k w_{k+s} = \begin{cases} \ell(a^2 - b^2) + \ell^2 b^2 & s = 0 \\ \ell(a^2 - b^2) & s = 1, \ldots, \ell - 1 \end{cases}$$

Hence

$$\sum_{k=0}^{\ell-1} w_k w_{k+s} = \begin{cases} -\ell b^2 + \ell^2 b^2 & s = 0 \\ -\ell b^2 & s = 1, \ldots, \ell - 1 \end{cases}$$

Now

$$P_X(s) = \sum_{k=0}^{\ell-1} x_k x_{k+s} = \ell a^2 - \sum_{k=0}^{\ell-1} w_k w_{k+s} = \begin{cases} \ell(a^2 + b^2) - \ell^2 b^2 & s = 0 \\ \ell(a^2 + b^2) & s = 1, \ldots, \ell - 1 \end{cases}$$

3 Altering $\mathcal{S}_1$ and $\mathcal{S}_2$

All the proofs "go through" if we let $\mathcal{S}_1 = \mathcal{R}$ or $\mathcal{S}_1 = \mathcal{C}$, that is, if A is a real or complex valued sequence. If $\mathcal{S}_1 = \mathcal{Z}$, we can also choose $\mathcal{S}_2 = \{0, \ldots, p^\alpha - 1\}$, p prime. Let us briefly focus on this last case. Assume that we want to construct sequences with the above properties of length ℓ. We then have to choose p and α such that $\ell \mid p^\alpha - 1$. Let g be a primitive root of $GF(p^\alpha)$ and let $\tilde{g} = g^{\frac{p^\alpha - 1}{\ell}}$. The sequences X and Y are then obtained by

$$x_k = \sum_{j=0}^{\ell-1} a_j \tilde{g}^{jk} \text{ and } y_k = \sum_{j=0}^{\ell-1} a_j \tilde{g}^{-jk}$$

where all the calculations are to be done in $GF(p^\alpha)$. Because of $\sum_{j=0}^{\ell-1} \tilde{g}^j = 0$, all the proofs from Section 2 remain valid.

4 Examples

Example 1:
Let $\mathcal{S}_1 = \mathcal{Z}$ and $\mathcal{S}_2 = \mathcal{C}$. Let $\ell = 6$ and $A_1 = \{2, 3, -3, 3, 3, -3\}$ and $A_2 = \{2, -3, 3, -3, -3, -3\}$ then

$$X_1 = \{5, -1, 5 + 10.39i, -1, 5 - 10.39i, -1\}$$
$$Y_1 = \{5, -1, 5 - 10.39i, -1, 5 + 10.39i, -1\}$$

and for X_2, Y_2:

$$X_2 = \{-7, 2 + 5.2i, 2 - 5.2i, 11, 2 + 5.2i, 2 - 5.2i\}$$
$$Y_2 = \{-7, 2 - 5.2i, 2 + 5.2i, 11, 2 - 5.2i, 2 + 5.2i\}$$

and

$$C_{X_1,Y_1}(s) = C_{Y_1,X_1}(s) = C_{X_2,Y_2}(s) = C_{Y_2,X_2}(s) = \begin{cases} 294 & s = 0 \\ -30 & s = 1, \ldots, 5 \end{cases}$$

Example 2:
Let $\mathcal{S}_1 = \mathcal{Z}$ and $\mathcal{S}_2 = \mathcal{C}$. Let $\ell = 9$ and $A_1 = \{1, 5, 0, -3, 4, 0, 0, 0, 0\}$ and $A_2 = \{1, 0, 0, 0, 0, -5, 2, 7, 1\}$ then

$$\begin{aligned} X_1 = \{&7, 2.57 + 1.98i, 6.43 + 4.95i, -6.5 + 7.79i, -1.5 - 4.83i, \\ &-1.5 + 4.83i, -6.5 - 7.79i, 6.43 - 4.95i, 2.57 - 1.98i\} \\ Y_1 = \{&7, 2.57 - 1.98i, 6.43 - 4.95i, -6.5 - 7.79i, -1.5 + 4.83i, \\ &-1.5 - 4.83i, -6.5 + 7.79i, 6.43 + 4.95i, 2.57 + 1.98i\} \end{aligned}$$

and for X_2, Y_2:

$$\begin{aligned} X_2 = \{&6, 6.68 - 7.56i, -10.23 - 4.86i, 1.5 + 9.53i, 3.55 - 2.5i, \\ &3.55 + 2.5i, 1.5 - 9.53i, -10.23 + 4.86i, 6.68 + 7.56i\} \\ Y_2 = \{&6, 6.68 + 7.56i, -10.23 + 4.86i, 1.5 - 9.53i, 3.55 + 2.5i, \\ &3.55 - 2.5i, 1.5 + 9.53i, -10.23 - 4.86i, 6.68 - 7.56i\} \end{aligned}$$

and

$$P_{X_1}(s) = P_{Y_1}(s) = P_{X_2}(s) = P_{Y_2}(s) = 9,$$

for $s = 0, \ldots, 8$.

Example 3:
(As Example 1 but now with $\mathcal{S}_2 = \{0, \ldots, 12\}$, $p = 13$. We let $g = 2$, $\tilde{g} = g^2 = 4$.)

$$X_1 = \{5, 12, 8, 12, 2, 12\}$$
$$Y_1 = \{5, 12, 2, 12, 8, 12\}$$

and for X_2, Y_2:

$$X_2 = \{6, 10, 7, 11, 10, 7\}$$
$$Y_2 = \{6, 7, 10, 11, 7, 10\}$$

and

$$C_{X_1,Y_1}(s) = C_{Y_1,X_1}(s) = C_{X_2,Y_2}(s) = C_{Y_2,X_2}(s) = \begin{cases} 8 & s = 0 \\ 9 & s = 1, \ldots, 5 \end{cases}$$

Example 4:
(As Example 1 but now with $\mathcal{S}_2 = \{0, \ldots, 36\}$, $p = 37$. We let $g = 2$, $\tilde{g} = g^6 = 27$.)

$$X_1 = \{5, 36, 27, 36, 20, 36\}$$
$$Y_1 = \{5, 36, 20, 36, 27, 36\}$$

and for X_2, Y_2:

$$X_2 = \{30, 13, 28, 11, 13, 28\}$$
$$Y_2 = \{30, 28, 13, 11, 28, 13\}$$

and

$$C_{X_1,Y_1}(s) = C_{Y_1,X_1}(s) = C_{X_2,Y_2}(s) = C_{Y_2,X_2}(s) = \begin{cases} 35 & s = 0 \\ 7 & s = 1, \ldots, 5 \end{cases}$$

Example 5:
(As Example 2 but now with $\mathcal{S}_2 = \{0, \ldots, 18\}$, $p = 19$. We let $g = 2$, $\tilde{g} = g^2 = 4$.)

$$X_1 = \{7, 17, 11, 4, 11, 3, 2, 7, 4\}$$
$$Y_1 = \{7, 4, 7, 2, 3, 11, 4, 11, 17\}$$

and for X_2, Y_2:

$$X_2 = \{6, 4, 6, 8, 7, 10, 14, 1, 10\}$$
$$Y_2 = \{6, 10, 1, 14, 10, 7, 8, 6, 4\}$$

and

$$P_{X_1}(s) = P_{Y_1}(s) = P_{X_2}(s) = P_{Y_2}(s) = 9,$$

for $s = 0, \ldots, 8$.

Example 6:
Let $\mathcal{S}_1 = \mathcal{Z}$ and $\mathcal{S}_2 = 0, \ldots, 18$, $p = 19$, $g = 2$, $\tilde{g} = g^2 = 4$. Let $\ell = 9$ and $A = \{2, 1, 1, -1, -1, -1, -1, 1, 1\}$ then

$$X = Y = \{1, 7, 3, 17, 15, 15, 17, 3, 7\}$$

and

$$P_X(s) = P_Y(s) = \begin{cases} 5 & s = 0 \\ 0 & s = 0, \ldots, 8 \end{cases}$$

5 A Computer–Search

We have shown constructions that yield sequences with special periodic autocorrelation function for every length ℓ. We can implement a search–program that searches through all sequences which have special periodic autocorrelation function and then checks which ones have certain additional properties (for example all its elements are in $\{-1, 0, 1\}$). "Traditional searches" for such sequences go precisely the other way round: typically a search–program searches through all sequences which have certain properties (for example all its elements are in $\{-1, 0, 1\}$) *and then* checks for special periodic autocorrelation function.

Computational Results

We let $\mathcal{S}_1 = \{-1, 0, 1\}$ and search through sequences R and W according to Lemma 3, Corollaries 1 and 2 and *hope* that $r_k, w_k \in \mathcal{Z}$. We got many results. Table 1 and 2 show a few examples. A sample search is given in the appendix.

Results obtained are somewhat disappointing since the sequences obtained expose a rather simple pattern. The sequences obtained may be constructed directly rather than via the theory and search in this paper.

Length ℓ	Sequences R and W	$P_R(s) + P_W(s)$
8	$R = \{2, 2, 2, 2, -6, 2, 2, 2\}$	$64, \ s = 0$
	$W = \{0, 0, 0, 0, 0, 0, 0, 0\}$	$0, \ s \neq 0$
12	$R = \{6, 2, -4, 2, 0, 2, 2, 2, 0, 2, 4, 2\}$	$144, \ s = 0$
	$W = \{0, 2, 0, 4, 0, 2, 0, -2, 0, -4, 0, 2\}$	$0, \ s \neq 0$

Table 1. Sample sequences R and W obtained via Corollary 1

Length ℓ	Sequences R and W	$P_R(s) - P_W(s)$
8	$R = \{6, 0, -2, 0, 2, 0, -2, 0\}$	0
	$W = \{0, 2, 4, -2, 0, 2, -4, -2\}$	
12	$R = \{1, 0, 2, 0, -2, 0, -1, 0, -2, 0, 2, 0\}$	0
	$W = \{0, 0, 0, 3, 0, 0, 0, 0, 0, -3, 0, 0\}$	

Table 2. Sample sequences R and W obtained via Corollary 2

References

[GavLem94] A. Gavish and A. Lempel, On ternary complementary sequences, *IEEE Transactions on Information Theory*, 40, 2, 522–526, 1994.

[GerSeb79] A.V. Geramita and J.Seberry, *Orthogonal Designs: Quadratic Forms and Hadamard Matrices*, Marcel Dekker, New York – Basel, 1979.

[GysSeb97] M. Gysin and J. Seberry, An experimental search and new combinatorial designs via a generalisation of cyclotomy, *Journal of Combinatorial Mathematics and Combinatorial Computing*, 27, 143–160, 1998.

[IreRos82] K. Ireland, M. Rosen, *A Classical Introduction to Modern Number Theory*, Springer–Verlag, New York, 1982.

[Paterson98] , K.G. Paterson, Binary sequence sets with Favorable Correlations from Difference Sets and MDS Codes, *IEEE Transactions on Information Theory*, Vol. 44, 1, 172–180, 1998.

[Schroeder84] M.R. Schroeder, *Number Theory in Science and Communication*, Springer–Verlag, New York, 1984.

[SebYam92] J. Seberry and M. Yamada, Hadamard matrices, sequences and block designs, *Contemporary Design Theory – a Collection of Surveys*, eds. J.Dinitz and D.R. Stinson, John Wiley and Sons, New York, 431–560, 1992.

[Sloane73] N.J.A. Sloane, *A Handbook of Integer Sequences*, Academic Press, New York, 1973.

A Complete Results from Exhaustive Computer–Searches

Construction Lemma 3 (i) with $\ell = 18$, $a = b = 1$, $r_k \in \mathcal{Z}$

(Some information about the seeding sequence A is also printed immediately after the sequence R. We set $a_0 = a = 1$ and then a_1 to a_9 = "sequence printed out" and a_{10} to a_{17} follow from a_1 to a_8.)

```
R: 018* 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 *
+++++++++
R: 016*002*-02*002*-02*002*-02*002*-02*002*-02*002*-02*002*-02*002*-02*002*-02*002*
++++++++-
R: 014*002*002*-04*002*002*-04*002*002*-04*002*002*-04*002*002*-04*002*002*-04*002*002*
+++++-+++
R: 012*004* 0 *-02* 0 *004*-06*004* 0 *-02* 0 *004*-06*004* 0 *-02* 0 *004*
+++++-++-
R: 014*-02*002*004*002*-02*-04*-02*002*004*002*-02*-04*-02*002*004*002*-02*
++-++++++
R: 012* 0 * 0 *006* 0 * 0 *-06* 0 * 0 *006* 0 * 0 *-06* 0 * 0 *006* 0 * 0 *
++-++++++-
R: 010* 0 *004* 0 *004* 0 *-08* 0 *004* 0 *004* 0 *-08* 0 *004* 0 *004* 0 *
```

```
++-++-+++
R: 008*002*002*002*002*002*-10*002*002*002*002*002*-10*002*002*002*002*002*
++-++-++-
R: 006* 0 * 0 *006* 0 * 0 *006* 0 * 0 *-12* 0 * 0 *006* 0 * 0 *006* 0 * 0 *
+-+-++++-+
R: 004*002*-02*008*-02*002*004*002*-02*-10*-02*002*004*002*-02*008*-02*002*
+-+-++++--
R: 002*002*002*002*002*002*002*002*002*-16*002*002*002*002*002*002*002*002*
+-+-+-+-+
R:  0 *004* 0 *004* 0 *004* 0 *004* 0 *-14* 0 *004* 0 *004* 0 *004* 0 *004*
+-+-+-+--
R: 002*-02*002*010*002*-02*002*-02*002*-08*002*-02*002*-02*002*010*002*-02*
+---++++-+
R:  0 * 0 * 0 *012* 0 * 0 * 0 * 0 * 0 *-06* 0 * 0 * 0 * 0 * 0 *012* 0 * 0 *
+---++++--
R: -02* 0 *004*006*004* 0 *-02* 0 *004*-12*004* 0 *-02* 0 *004*006*004* 0 *
+---+-+-+
R: -04*002*002*008*002*002*-04*002*002*-10*002*002*-04*002*002*008*002*002*
+---+-+--
R: 006* 0 * 0 *-06* 0 * 0 *006* 0 * 0 *012* 0 * 0 *006* 0 * 0 *-06* 0 * 0 *
-+++-+-++
R: 004*002*-02*-04*-02*002*004*002*-02*014*-02*002*004*002*-02*-04*-02*002*
-+++-+-+-
R: 002*002*002*-10*002*002*002*002*002*008*002*002*002*002*002*-10*002*002*
-+++---++
R:  0 *004* 0 *-08* 0 *004* 0 *004* 0 *010* 0 *004* 0 *004* 0 *-08* 0 *004*
-+++---+-
R: 002*-02*002*-02*002*-02*002*-02*002*016*002*-02*002*-02*002*-02*002*-02*
-+-+-+-++
R:  0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 *018* 0 * 0 * 0 * 0 * 0 * 0 * 0 * 0 *
-+-+-+-+-
R: -02* 0 *004*-06*004* 0 *-02* 0 *004*012*004* 0 *-02* 0 *004*-06*004* 0 *
-+-+---++
R: -04*002*002*-04*002*002*-04*002*002*014*002*002*-04*002*002*-04*002*002*
-+-+---+-
R: -06* 0 * 0 * 0 * 0 * 0 *012* 0 * 0 * 0 * 0 * 0 *012* 0 * 0 * 0 * 0 * 0 *
--+--+--+
R: -08*002*-02*002*-02*002*010*002*-02*002*-02*002*010*002*-02*002*-02*002*
--+--+---
R: -10*002*002*-04*002*002*008*002*002*-04*002*002*008*002*002*-04*002*002*
--+-----+
R: -12*004* 0 *-02* 0 *004*006*004* 0 *-02* 0 *004*006*004* 0 *-02* 0 *004*
--+------
R: -10*-02*002*004*002*-02*008*-02*002*004*002*-02*008*-02*002*004*002*-02*
-----+--+
R: -12* 0 * 0 *006* 0 * 0 *006* 0 * 0 *006* 0 * 0 *006* 0 * 0 *006* 0 * 0 *
-----+---
R: -14* 0 *004* 0 *004* 0 *004* 0 *004* 0 *004* 0 *004* 0 *004* 0 *004* 0 *
--------+
R: -16*002*002*002*002*002*002*002*002*002*002*002*002*002*002*002*002*002*
---------
```

Nr of sequences found: 00032

GENERALISED PATTERN AVOIDANCE

ANDERS CLAESSON

ABSTRACT. Recently, Babson and Steingrímsson have introduced generalised permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. We will consider pattern avoidance for such patterns, and give a complete solution for the number of permutations avoiding any single pattern of length three with exactly one adjacent pair of letters. For eight of these twelve patterns the answer is given by the Bell numbers. For the remaining four the answer is given by the Catalan numbers. We also give some results for the number of permutations avoiding two different patterns. These results relate the permutations in question to Motzkin paths, involutions and non-overlapping partitions. Furthermore, we define a new class of set partitions, called monotone partitions, and show that these partitions are in one-to-one correspondence with non-overlapping partitions.

1. INTRODUCTION

In the last decade a wealth of articles has been written on the subject of pattern avoidance, also known as the study of "restricted permutations" and "permutations with forbidden subsequences". Classically, a pattern is a permutation $\sigma \in \mathcal{S}_k$, and a permutation $\pi \in \mathcal{S}_n$ avoids σ if there is no subsequence in π whose letters are in the same relative order as the letters of σ. For example, $\pi \in \mathcal{S}_n$ avoids 132 if there is no $1 \leq i < j < k \leq n$ such that $\pi(i) < \pi(k) < \pi(j)$. In [4] Knuth established that for all $\sigma \in \mathcal{S}_3$, the number of permutations in $\mathcal{S}_n$ avoiding σ equals the nth Catalan number, $C_n = \frac{1}{1+n}\binom{2n}{n}$. One may also consider permutations that are required to avoid several patterns. In [5] Simion and Schmidt gave a complete solution for permutations avoiding any set of patterns of length three. Even patterns of length greater than three have been considered. For instance, West showed in [8] that permutations avoiding both 3142 and 2413 are enumerated by the Schröder numbers, $S_n = \sum_{i=0}^{n} \binom{2n-i}{i} C_{n-i}$.

In [1] Babson and Steingrímsson introduced generalised permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. The motivation for Babson and Steingrímsson in introducing these patterns was the study of Mahonian statistics, and they showed that essentially all Mahonian permutation statistics in the literature can be written as linear combinations of such patterns. An example of a generalised pattern is $(a\text{–}cb)$. An $(a\text{–}cb)$-subword of a permutation $\pi = a_1 a_2 \cdots a_n$ is a subword $a_i a_j a_{j+1}$, $(i < j)$, such that $a_i < a_{j+1} < a_j$. More generally, a pattern p is a word over the alphabet $a < b < c < d \cdots$ where two adjacent letters may or may not be separated by a dash. The absence of a dash between two adjacent letters in a p indicates that the corresponding letters in a p-subword of a permutation must be adjacent. Also, the ordering of the letters in the p-subword must match the ordering of the letters in the pattern. This definition, as well as any other definition in the introduction, will be stated rigorously in Section 2. All classical patterns are generalised patterns where each pair of adjacent letters is separated by a dash. For example, the generalised pattern equivalent to 132 is $(a\text{–}c\text{–}b)$.

Date: October 8, 2001.

We extend the notion of pattern avoidance by defining that a permutation avoids a (generalised) pattern p if it does not contain any p-subwords. We show that this is a fruitful extension, by establishing connections to other well known combinatorial structures, not previously shown to be related to pattern avoidance. The main results are given below.

P	$\lvert\mathcal{S}_n(P)\rvert$	*Description*
a–bc	B_n	Partitions of $[n]$
a–cb	B_n	Partitions of $[n]$
b–ac	C_n	Dyck paths of length $2n$
a–bc, ab–c	B_n^*	Non-overlapping partitions of $[n]$
a–bc, a–cb	I_n	Involutions in $\mathcal{S}_n$
a–bc, ac–b	M_n	Motzkin paths of length n

Here $\mathcal{S}_n(P) = \{\pi \in \mathcal{S}_n : \pi \text{ avoids } p \text{ for all } p \in P\}$, and $[n] = \{1, 2, \dots, n\}$. When proving that $|\mathcal{S}_n(a\text{–}bc,\, ab\text{–}c)| = B_n^*$ (the nth Bessel number), we first prove that there is a one-to-one correspondence between $\{a\text{–}bc, ab\text{–}c\}$-avoiding permutations and *monotone partitions*. A partition is monotone if its non-singleton blocks can be written in increasing order of their least element and increasing order of their greatest element, simultaneously. This new class of partitions is then shown to be in one-to-one correspondence with non-overlapping partitions.

2. Preliminaries

By an *alphabet* X we mean a non-empty set. An element of X is called a *letter*. A *word* over X is a finite sequence of letters from X. We consider also the *empty word*, that is, the word with no letters; it is denoted by ϵ. Let $x = x_1 x_2 \cdots x_n$ be a word over X. We call $|x| := n$ the *length* of x. A *subword* of x is a word $v = x_{i_1} x_{i_2} \cdots x_{i_k}$, where $1 \le i_1 < i_2 < \cdots < i_k \le n$. A *segment* of x is a word $v = x_i x_{i+1} \cdots x_{i+k}$. If X and Y are two linearly ordered alphabets, then two words $x = x_1 x_2 \cdots x_n$ and $y = y_1 y_2 \cdots y_n$ over X and Y, respectively, are said to be *order equivalent* if $x_i < x_j$ precisely when $y_i < y_j$.

Let $X = A \cup \{\text{–}\}$ where A is a linearly ordered alphabet. For each word x let $\bar{x}$ be the word obtained from x by deleting all dashes in x. A word p over X is called a *pattern* if it contains no two consecutive dashes and $\bar{p}$ has no repeated letters. By slight abuse of terminology we refer to the *length of a pattern* p as the length of $\bar{p}$. If the ith letter in p is a dash precisely when the ith letter in q is a dash, and p and q are order equivalent, then p and q are *equivalent*. In what follows all patterns will be over the alphabet $\{a, b, c, d, \dots\} \cup \{\text{–}\}$ where $a < b < c < d < \cdots$.

Let $[n] := \{1, 2, \dots, n\}$ (so $[0] = \emptyset$). A *permutation* of $[n]$ is bijection from $[n]$ to $[n]$. Let $\mathcal{S}_n$ be the set of permutations of $[n]$. We shall usually think of a permutation π as the word $\pi(1)\pi(2)\cdots\pi(n)$ over the alphabet $[n]$. In particular, $\mathcal{S}_0 = \{\epsilon\}$, since there is only one bijection from $\emptyset$ to $\emptyset$, the empty map. We say that a subword σ of π is a *p-subword* if by replacing (possibly empty) segments of π with dashes we can obtain a pattern q equivalent to p such that $\bar{q} = \sigma$. However, all patterns that we will consider will have a dash at the beginning and one at the end. For convenience, we therefore leave them out. For example, $(a\text{–}bc)$ is a pattern, and the permutation 491273865 contains three $(a\text{–}bc)$-subwords, namely 127, 138, and 238. A permutation is said to be *p-avoiding* if it does not contain any p-subwords. Define $\mathcal{S}_n(p)$ to be the set of p-avoiding permutations in $\mathcal{S}_n$ and, more generally, $\mathcal{S}_n(A) = \bigcap_{p\in A} \mathcal{S}_n(p)$.

We may think of a pattern p as a permutation statistic, that is, define $p\,\pi$ as the number of p-subwords in π, thus regarding p as a function from $\mathcal{S}_n$ to $\mathbb{N}$. For example, $(a\text{–}bc)\,491273865 = 3$. In particular, π is p-avoiding if and only if $p\,\pi = 0$. We say that

two permutation statistics stat and stat$'$ are *equidistributed* over $A \subseteq \mathcal{S}_n$, if

$$\sum_{\pi \in A} x^{\mathrm{stat}\,\pi} = \sum_{\pi \in A} x^{\mathrm{stat}'\,\pi}.$$

In particular, this definition applies to patterns.

Let $\pi = a_1 a_2 \cdots a_n \in \mathcal{S}_n$. An i such that $a_i > a_{i+1}$ is called a *descent* in π. We denote by $\operatorname{des}\pi$ the number of descents in π. Observe that des can be defined as the pattern (ba), that is, $\operatorname{des}\pi = (ba)\pi$. A *left-to-right minimum* of π is an element a_i such that $a_i < a_j$ for every $j < i$. The number of left-to-right minima is a permutation statistic. Analogously we also define *left-to-right maximum*, *right-to-left minimum*, and *right-to-left maximum*.

In this paper we will relate permutations avoiding a given set of patterns to other better known combinatorial structures. Here follows a brief description of these structures. Two excellent references on combinatorial structures are [7] and [6].

Set partitions. A *partition* of a set S is a family, $\pi = \{A_1, A_2, \ldots, A_k\}$, of pairwise disjoint non-empty subsets of S such that $S = \cup_i A_i$. We call A_i a *block* of π. The total number of partitions of $[n]$ is called a *Bell number* and is denoted B_n. For reference, the first few Bell numbers are

$$1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, 678570, 4213597.$$

Let $S(n,k)$ be the number of partitions of $[n]$ into k blocks; these numbers are called the *Stirling numbers of the second kind*.

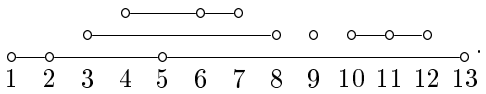
Non-overlapping partitions. Two blocks A and B of a partition π *overlap* if

$$\min A < \min B < \max A < \max B.$$

A partition is *non-overlapping* if no pairs of blocks overlap. Thus

$$\pi = \{\{1, 2, 5, 13\}, \{3, 8\}, \{4, 6, 7\}, \{9\}, \{10, 11, 12\}\}$$

is non-overlapping. A pictorial representation of π is

$\pi =$ 

Let B_n^* be the number of non-overlapping partitions of $[n]$; this number is called the nth *Bessel number* [3, p. 423]. The first few Bessel numbers are

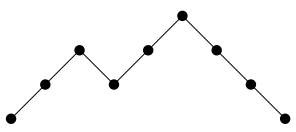
$$1, 1, 2, 5, 14, 43, 143, 509, 1922, 7651, 31965, 139685, 636712.$$

We denote by $S^*(n,k)$ the number of non-overlapping partitions of $[n]$ into k blocks.

Involutions. An *involution* is a permutation which is its own inverse. We denote by I_n the number of involutions in $\mathcal{S}_n$. The sequence $\{I_n\}_0^\infty$ starts with

$$1, 1, 2, 4, 10, 26, 76, 232, 764, 2620, 9496, 35696, 140152.$$

Dyck paths. A *Dyck path* of length $2n$ is a lattice path from $(0,0)$ to $(2n,0)$ with steps $(1,1)$ and $(1,-1)$ that never goes below the x-axis. Letting u and d represent the steps $(1,1)$ and $(1,-1)$ respectively, we code such a path with a word over $\{u,d\}$. For example, the path

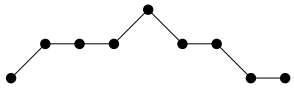


is coded by $uuduuddd$. A *return step* in a Dyck path δ is a d such that $\delta = \alpha u \beta d \gamma$, for some Dyck paths α, β, and γ. A useful observation is that every non-empty Dyck path δ can be uniquely decomposed as $\delta = u\alpha d\beta$, where α and β are Dyck paths. This is the so-called *first return decomposition* of δ.

The nth *Catalan number* $C_n = \frac{1}{n+1}\binom{2n}{n}$ counts the number of Dyck paths of length $2n$. The sequence of Catalan numbers starts with

$$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012.$$

Motzkin paths. A *Motzkin path* of length n is a lattice path from $(0,0)$ to $(n,0)$ with steps $(1,0)$, $(1,1)$, and $(1,-1)$ that never goes below the x-axis. Letting ℓ, u, and d represent the steps $(1,0)$, $(1,1)$, and $(1,-1)$ respectively, we code such a path with a word over $\{\ell, u, d\}$. For example, the path



is coded by $u\ell\ell ud\ell d\ell$. If δ is a non-empty Motzkin path, then δ can be decomposed as $\delta = \ell\gamma$ or $\delta = u\alpha d\beta$, where α, β and γ are Motzkin paths.

The nth *Motzkin number* M_n is the number of Motzkin paths of length n. The first few of the Motzkin numbers are

$$1, 1, 2, 4, 9, 21, 51, 127, 323, 835, 2188, 5798, 15511.$$

3. Three classes of patterns

Let $\pi = a_1 a_2 \cdots a_n \in \mathcal{S}_n$. Define the *reverse* of π as $\pi^r := a_n \cdots a_2 a_1$, and define the *complement* of π by $\pi^c(i) = n + 1 - \pi(i)$, where $i \in [n]$.

Proposition 1. *With respect to being equidistributed, the twelve pattern statistics of length three with one dash fall into the following three classes.*

(i) a–bc, c–ba, ab–c, cb–a.
(ii) a–cb, c–ab, ba–c, bc–a.
(iii) b–ac, b–ca, ac–b, ca–b.

Proof. The bijections $\pi \mapsto \pi^r$, $\pi \mapsto \pi^c$, and $\pi \mapsto (\pi^r)^c$ give the equidistribution part of the result. Calculations show that these three distributions differ pairwise on $\mathcal{S}_4$. □

4. Permutations avoiding a pattern of class one or two

Proposition 2. *Partitions of $[n]$ are in one-to-one correspondence with $(a$–$bc)$-avoiding permutations in $\mathcal{S}_n$. Hence $|\mathcal{S}_n(a$–$bc)| = B_n$.*

First proof. Recall that the Bell numbers satisfy $B_0 = 1$, and

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k.$$

We show that $|\mathcal{S}_n(a$–$bc)|$ satisfy the same recursion. Clearly, $\mathcal{S}_0(a$–$bc) = \{\epsilon\}$. For $n > 0$, let $M = \{2, 3, \ldots, n+1\}$, and let S be a k element subset of M. For each $(a$–$bc)$-avoiding permutation σ of S we construct a unique $(a$–$bc)$-avoiding permutation π of $[n+1]$. Let τ be the word obtained by writing the elements of $M \setminus S$ in decreasing order. Define $\pi := \sigma 1 \tau$.

Conversely, if $\pi = \sigma 1 \tau$ is a given $(a$–$bc)$-avoiding permutation of $[n+1]$, where $|\sigma| = k$, then the letters of τ are in decreasing order, and σ is an $(a$–$bc)$-avoiding permutation of the k element set $\{2, 3, \ldots, n+1\} \setminus \{i : i \text{ is a letter in } \tau\}$. □

Second proof. Given a partition π of $[n]$, we introduce a standard representation of π by requiring that:

(a) Each block is written with its least element first, and the rest of the elements of that block are written in decreasing order.
(b) The blocks are written in decreasing order of their least element, and with dashes separating the blocks.

Define $\widehat{\pi}$ to be the permutation we obtain from π by writing it in standard form and erasing the dashes. We now argue that $\widehat{\pi} := a_1 a_2 \cdots a_n$ avoids $(a\text{–}bc)$. If $a_i < a_{i+1}$, then a_i and a_{i+1} are the first and the second element of some block. By the construction of $\widehat{\pi}$, a_i is a left-to-right minimum, hence there is no $j \in [i-1]$ such that $a_j < a_i$.

Conversely, π can be recovered uniquely from $\widehat{\pi}$ by inserting a dash in $\widehat{\pi}$ preceding each left-to-right minimum, apart from the first letter in $\widehat{\pi}$. Indeed, it easy to see that the partition, π, in this way obtained is written in standard form. Thus $\pi \mapsto \widehat{\pi}$ gives the desired bijection. □

Example. As an illustration of the map defined in the above proof, let

$$\pi = \{\{1,3,5\},\{2,6,9\},\{4,7\},\{8\}\}.$$

Its standard form is 8–47–296–153. Thus $\widehat{\pi} = 847296153$.

Proposition 3. *Let $L(\pi)$ be the number of left-to-right minima of π. Then*

$$\sum_{\pi \in \mathcal{S}_n(a-bc)} x^{L(\pi)} = \sum_{k \geq 0} S(n,k) x^k.$$

Proof. This result follows readily from the second proof of Proposition 2. We here give a different proof, which is based on the fact that the Stirling numbers of the second kind satisfy

$$S(n,k) = S(n-1,k-1) + kS(n-1,k).$$

Let $T(n,k)$ be the number of permutations in $\mathcal{S}_n(a\text{–}bc)$ with k left-to-right minima. We show that the $T(n,k)$ satisfy the same recursion as the $S(n,k)$.

Let π be an $(a\text{–}bc)$-avoiding permutation of $[n-1]$. To insert n in π, preserving $(a\text{–}bc)$-avoidance, we can put n in front of π or we can insert n immediately after each left-to-right minimum. Putting n in front of π creates a new left-to-right minimum, while inserting n immediately after a left-to-right minimum does not. □

Proposition 4. *Partitions of $[n]$ are in one-to-one correspondence with $(a\text{–}cb)$-avoiding permutations in $\mathcal{S}_n$. Hence $|\mathcal{S}_n(a\text{–}cb)| = B_n$.*

Proof. Let π be a partition of $[n]$. We introduce a standard representation of π by requiring that:

(a) The elements of a block are written in increasing order.
(b) The blocks are written in decreasing order of their least element, and with dashes separating the blocks.

(Note that this standard representation is different from the one given in the second proof of Proposition 2.) Define $\widehat{\pi}$ to be the permutation we obtain from π by writing it in standard form and erasing the dashes. It easy to see that $\widehat{\pi}$ avoids $(a\text{–}cb)$. Conversely, π can be recovered uniquely from $\widehat{\pi}$ by inserting a dash in between each descent in $\widehat{\pi}$. □

Example. As an illustration of the map defined in the above proof, let

$$\pi = \{\{1,3,5\},\{2,6,9\},\{4,7\},\{8\}\}.$$

Its standard form is 8–47–269–135. Thus $\widehat{\pi} = 847269135$.

Proposition 5.

$$\sum_{\pi\in\mathcal{S}_n(a-cb)} x^{1+\operatorname{des}\pi} = \sum_{k\geq 0} S(n,k)x^k.$$

Proof. From the proof of Proposition 4 we see that π has $k+1$ blocks precisely when $\widehat{\pi}$ has k descents. □

Proposition 6. *Involutions in $\mathcal{S}_n$ are in one-to-one correspondence with permutations in $\mathcal{S}_n$ that avoid $(a\text{–}bc)$ and $(a\text{–}cb)$. Hence*

$$|\mathcal{S}_n(a\text{–}bc, a\text{–}cb)| = I_n.$$

Proof. We give a combinatorial proof using a bijection that is essentially identical to the one given in the second proof of Proposition 2.

Let $\pi \in \mathcal{S}_n$ be an involution. Recall that π is an involution if and only if each cycle of π is of length one or two. We now introduce a standard form for writing π in cycle notation by requiring that:

(a) Each cycle is written with its least element first.
(b) The cycles are written in decreasing order of their least element.

Define $\widehat{\pi}$ to be the permutation obtained from π by writing it in standard form and erasing the parentheses separating the cycles.

Observe that $\widehat{\pi}$ avoids $(a\text{–}bc)$: Assume that $a_i < a_{i+1}$, that is $(a_i\, a_{i+1})$ is a cycle in π, then a_i is a left-to-right minimum in π. This is guaranteed by the construction of $\widehat{\pi}$. Thus there is no $j < i$ such that $a_j < a_i$.

The permutation $\widehat{\pi}$ also avoids $(a\text{–}cb)$: Assume that $a_i > a_{i+1}$, then a_{i+1} must be the smallest element of some cycle. Whence a_{i+1} is a left-to-right minimum in $\widehat{\pi}$.

Conversely, if $\widehat{\pi} := a_1 \dots a_n$ is an $\{a\text{–}bc, a\text{–}cb\}$-avoiding permutation then the involution π is given by: $(a_i\, a_{i+1})$ is a cycle in π if and only if $a_i < a_{i+1}$. □

Example. The involution $\pi = 826543719$ written in standard form is

$$(9)(7)(4\,5)(3\,6)(2)(1\,8),$$

and hence $\widehat{\pi} = 974536218$.

Proposition 7. *The number of permutations in $\mathcal{S}_n(a\text{–}bc, a\text{–}cb)$ with $n-k-1$ descents equals the number of involutions in $\mathcal{S}_n$ with $n-2k$ fixed points.*

Proof. Under the bijection $\pi \mapsto \widehat{\pi}$ in the proof of Proposition 6, a cycle of length two in π corresponds to an occurrence of (ab) in $\widehat{\pi}$. Hence, if π has $n-2k$ fixed points, then $\widehat{\pi}$ has $n-k-1$ descents. □

Corollary 8.

$$\sum_{\pi\in\mathcal{S}_n(a-bc,a-cb)} x^{1+\operatorname{des}\pi} = \sum_{k=0}^{n} \binom{n}{k}\binom{n-k}{k}\frac{k!}{2^k}x^{n-k}.$$

Proof. Let I_n^k denote the number of involutions in $\mathcal{S}_n$ with k fixed points. Then Proposition 7 is equivalently stated as

$$\sum_{\pi\in\mathcal{S}_n(a-bc,a-cb)} x^{1+\operatorname{des}\pi} = \sum_{k\geq 0} I_n^{n-2k}x^{n-k}. \tag{1}$$

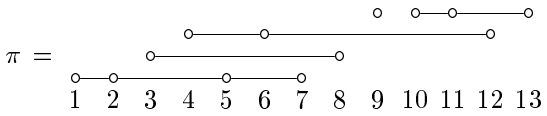
The result now follows from the well-known and easily to derived formula

$$I_n^k = \binom{n}{k}\binom{n-k}{r}\frac{r!}{2^r}, \quad \text{where } r = \frac{n-k}{2},$$

for $n-k$ even, with $I_n^k = 0$ for $n-k$ odd. □

Definition 9. Let π be an arbitrary partition whose non-singleton blocks $\{A_1, \ldots, A_k\}$ are ordered so that for all $i \in [k-1]$, $\min A_i > \min A_{i+1}$. If $\max A_i > \max A_{i+1}$ for all $i \in [k-1]$, then we call π a *monotone partition*. The set of monotone partitions of $[n]$ is denoted by $\mathcal{M}_n$.

Example. The partition



is monotone.

Proposition 10. *Monotone partitions of $[n]$ are in one-to-one correspondence with permutations in $\mathcal{S}_n$ that avoid $(a\text{–}bc)$ and $(ab\text{–}c)$. Hence*

$$|\mathcal{S}_n(a\text{–}bc, ab\text{–}c)| = |\mathcal{M}_n|.$$

Proof. Given π in $\mathcal{M}_n$, let $A_1\text{–}A_2\text{–}\cdots\text{–}A_k$ be the result of writing π in the standard form given in the second proof of Proposition 2, and let $\widehat{\pi} = A_1A_2\cdots A_k$. By the construction of $\widehat{\pi}$ the first letter in each A_i is a left-to-right minimum. Furthermore, since π is monotone the second letter in each non-singleton A_i is a right-to-left maximum. Therefore, if xy is an (ab)-subword of $\widehat{\pi}$, then x is left-to-right minimum and y is a right-to-left maximum. Thus $\widehat{\pi}$ avoids both $(a\text{–}bc)$ and $(ab\text{–}c)$.

Conversely, given $\widehat{\pi}$ in $\mathcal{S}_n(a\text{–}bc, ab\text{–}c)$, let $A_1\text{–}A_2\text{–}\cdots\text{–}A_k$ be the result of inserting a dash in $\widehat{\pi}$ preceding each left-to-right minimum, apart from the first letter in $\widehat{\pi}$. Since $\widehat{\pi}$ is $(ab\text{–}c)$-avoiding, the second letter in each non-singleton A_i is a right-to-left maximum. The second letter in A_i is the maximal element of A_i when A_i is viewed as a set. Thus $\pi = \{A_1, A_2, \ldots, A_k\}$ is monotone. □

We now show that there is a one-to-one correspondence between monotone partitions and non-overlapping partitions. The proof we give is strongly influenced by the paper [3], in which Flajolet and Schot showed that the ordinary generating function of the Bessel numbers admits a nice continued fraction expansion

$$\sum_{n\geq 0} B_n^* x^n = \cfrac{1}{1 - 1\cdot x - \cfrac{x^2}{1 - 2\cdot x - \cfrac{x^2}{1 - 3\cdot x - \cfrac{x^2}{\ddots}}}},$$

and using that as a starting point they derived the asymptotic formula

$$B_n^* \sim \sum_{k\geq 0} \frac{k^{n+2}}{(k!)^2}.$$

Proposition 11. *Monotone partitions of $[n]$ are in one-to-one correspondence with non-overlapping partitions of $[n]$. Hence $|\mathcal{M}_n| = B_n^*$.*

Proof. Let π be a non-overlapping partition of $[n]$. From π we will create a new partition by successively inserting $1, 2, \ldots, n$, in this order, into this new partition. During this process a block is labelled as either *open* or *closed*. More formally, in each step $k = 1, 2, \ldots, n$ in this process we will have a partition σ of $[k]$ together with a function from σ to the set of labels $\{\text{open}, \text{closed}\}$. Before we start we also need a labelling of the blocks of π. Actually we need n such labellings, one for each $k \in [n]$: At step k a block B of π is labelled open if $\max B > k$ and closed otherwise. For ease

of language, we say that a block is open if it is labelled open, and closed if it is labelled closed.

(a) If k is the minimal element of a non-singleton block of π, then create a new block $\{k\}$ and label it open.
(b) If k is the maximal element of a non-singleton block of π, then insert k into the open block with the smallest minimal element, and label it closed.
(c) If k belongs to a non-singleton block B of π and is not the minimal or the maximal element of B, and B has the ith largest minimal element of the open blocks of π, then insert k into the open block with the ith largest minimal element.
(d) If $\{k\}$ is a block of π then create a new block $\{k\}$ and label it closed.

Define $\Phi(\pi)$ as the partition obtained from π by applying the above process. Observe that $\Phi(\pi)$ is monotone. Indeed, the two crucial observations are (i) in (b) we label the open block with the smallest minimum closed, and (ii) a block labelled closed has received all its elements.

Conversely, we give a map Ψ that to each monotone partition π of $[n]$ gives a unique non-overlapping partition $\Psi(\pi)$ of $[n]$. Define Ψ the same way as Φ is defined, except for case (c), where we instead of inserting k into the block labelled open with the *smallest* minimal element, insert k into the block labelled open with the *largest* minimal element. It is easy to see that Φ and Ψ are each others inverses and hence they are bijections. □

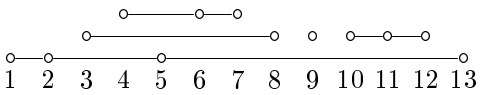
Corollary 12. *The non-overlapping partitions of* $[n]$ *are in one-to-one correspondence with permutations in* $\mathcal{S}_n$ *that avoid* $(a{-}bc)$ *and* $(ab{-}c)$*. Hence*

$$|\mathcal{S}_n(a{-}bc, ab{-}c)| = B_n^*.$$

Proof. Follows immediately from Proposition 10 together with Proposition 11. □

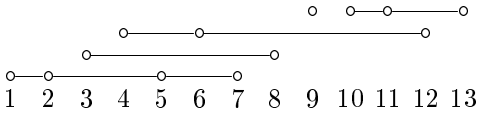
Example. By the proof of Proposition 11, the non-overlapping partition

π =



corresponds to the monotone partition

$\Phi(\pi)$ =



that according to the proof of Proposition 10 corresponds to the $\{a{-}bc, ab{-}c\}$-avoiding permutation

$$\widehat{\Phi(\pi)} = 10\ 13\ 11\ 9\ 4\ 12\ 6\ 3\ 8\ 1\ 7\ 5\ 2.$$

Proposition 13. *Let* $L(\pi)$ *be the number of left-to-right minima of* π*. Then*

$$\sum_{\pi \in \mathcal{S}_n(a-bc, ab-c)} x^{L(\pi)} = \sum_{k \geq 0} S^*(n,k) x^k.$$

Proof. Under the bijection $\pi \mapsto \widehat{\pi}$ in the proof of Proposition 10, the number of blocks in π determines the number of left-to-right minima of $\widehat{\pi}$, and vice versa. The number of blocks is not changed by the bijection Ψ in the proof of Proposition 11. □

5. Permutations avoiding a pattern of class three

In [4] Knuth observed that there is a one-to-one correspondence between $(b\text{–}a\text{–}c)$-avoiding permutations and Dyck paths. For completeness and future reference we give this result as a lemma, and prove it using a bijection which rests on the first return decomposition of Dyck paths. First we need a definition. For each word $x = x_1x_2\cdots x_n$ without repeated letters, we define the *projection* of x onto $\mathcal{S}_n$, which we denote $\mathrm{proj}(x)$, by

$$\mathrm{proj}(x) = a_1a_2\cdots a_n\,, \quad \text{where } \; a_i = |\{j \in [n] : x_j \le x_i\}|.$$

Equivalently, $\mathrm{proj}(x)$ is the permutation in $\mathcal{S}_n$ which is order equivalent to x. For example, $\mathrm{proj}(265) = 132$.

Lemma 1. $|\mathcal{S}_n(b\text{–}a\text{–}c)| = C_n$.

Proof. Let $\pi = a_1a_2\cdots a_n$ be a permutation of $[n]$ such that $a_k = 1$. Then π is $(b\text{–}a\text{–}c)$-avoiding if and only if $\pi = \sigma 1\tau$, where $\sigma := a_1\cdots a_{k-1}$ is a $(b\text{–}a\text{–}c)$-avoiding permutation of $\{n, n-1, \ldots, n-k+1\}$, and $\tau := a_{k+1}\cdots a_n$ is a $(b\text{–}a\text{–}c)$-avoiding permutation of $\{2, 3, \ldots, k\}$.

We define recursively a mapping Φ from $\mathcal{S}_n(b\text{–}a\text{–}c)$ onto the set of Dyck paths of length $2n$. If π is the empty word, then so is the Dyck path determined by π, that is, $\Phi(\epsilon) = \epsilon$. If $\pi \neq \epsilon$, then we can use the factorisation $\pi = \sigma 1\tau$ from above, and define $\Phi(\pi) = u\,(\Phi \circ \mathrm{proj})(\sigma)\,d\,(\Phi \circ \mathrm{proj})(\tau)$. It is easy to see that Φ may be inverted, and hence is a bijection. □

Lemma 2. *A permutation avoids $(b\text{–}ac)$ if and only if it avoids $(b\text{–}a\text{–}c)$.*

Proof. The sufficiency part of the proposition is trivial. The necessity part is not difficult either. Assume that π contains a $(b\text{–}a\text{–}c)$-subword. Then there is a segment $Bm_1\cdots m_r$ of π, where, for some $j < r$, $m_j < B$ and $m_r > B$. Now choose the largest i such that $m_i < B$, then $m_{i+1} > B$. □

Proposition 14. *Dyck paths of length $2n$ are in one-to-one correspondence with $(b\text{–}ac)$-avoiding permutations in $\mathcal{S}_n$. Hence*

$$|\mathcal{S}_n(b\text{–}ac)| = \frac{1}{n+1}\binom{2n}{n}.$$

Proof. Follows immediately from Lemmas 1 and 2. □

Proposition 15. *Let $L(\pi)$ be the number of left-to-right minima of π. Then*

$$\sum_{\pi \in \mathcal{S}_n(b\text{–}ac)} x^{L(\pi)} = \sum_{k \ge 0} \frac{k}{2n-k}\binom{2n-k}{n}x^k.$$

Proof. Let $R(\delta)$ denote the number of return steps in the Dyck path δ. It is well known (see [2]) that the distribution of R over all Dyck paths of length $2n$ is the distribution we claim that L has over $\mathcal{S}_n(b\text{–}ac)$. .

Let γ be a Dyck path of length $2n$, and let $\gamma = u\alpha d\beta$ be its first return decomposition. Then $R(\gamma) = 1 + R(\beta)$. Let $\pi \in \mathcal{S}_n(b\text{–}ac)$, and let $\pi = \sigma 1\tau$ be the decomposition given in the proof of Lemma 1. Then $L(\pi) = 1 + L(\sigma)$. The result now follows by induction. □

In addition, it is easy to deduce that left-to-right minima, left-to-right maxima, right-to-left minima, and right-to-left maxima all share the same distribution over $\mathcal{S}_n(b\text{–}ac)$.

Proposition 16. *Motzkin paths of length n are in one-to-one correspondence with permutations in $\mathcal{S}_n$ that avoid $(a\text{–}bc)$ and $(ac\text{–}b)$. Hence*

$$|\mathcal{S}_n(a\text{–}bc, ac\text{–}b)| = M_n.$$

Proof. We mimic the proof of Lemma 1. Let $\pi \in \mathcal{S}_n(a\text{–}bc, ac\text{–}b)$. Since π avoids $(ac\text{–}b)$ it also avoids $(a\text{–}c\text{–}b)$ by Lemma 2 via $\pi \mapsto (\pi^c)^r$. Thus we may write $\pi = \sigma n \tau$, where $\pi(k) = n$, σ is an $\{a\text{–}bc, ac\text{–}b\}$-avoiding permutation of $\{n-1, n-2, \ldots, n-k+1\}$, and τ is an $\{a\text{–}bc, ac\text{–}b\}$-avoiding permutation of $[n-k]$. If $\sigma \neq \epsilon$ then $\sigma = \sigma' r$ where $r = n-k+1$, or else an $(a\text{–}bc)$-subword would be formed with n as the 'c' in $(a\text{–}bc)$. Define a map Φ from $\mathcal{S}_n(a\text{–}bc, ac\text{–}b)$ to the set of Motzkin paths by $\Phi(\epsilon) = \epsilon$ and

$$\Phi(\pi) = \begin{cases} \ell\,(\Phi \circ \mathrm{proj})(\sigma) & \text{if } \pi = n\sigma, \\ u\,(\Phi \circ \mathrm{proj})(\sigma)\,d\,\Phi(\tau) & \text{if } \pi = \sigma r n \tau \text{ and } r = n-k+1. \end{cases}$$

It is routine to find the inverse of Φ. □

Example. Let us find the Motzkin path associated with the $\{a\text{–}bc, ac\text{–}b\}$-avoiding permutation 76453281.

$$\begin{aligned} \Phi(76453281) &= u\Phi(54231)d\Phi(1) \\ &= u\ell\Phi(4231)d\ell \\ &= u\ell\ell\Phi(231)d\ell \\ &= u\ell\ell ud\Phi(1)d\ell \\ &= u\ell\ell ud\ell d\ell \end{aligned}$$

Acknowledgement

I am greatly indebted to my advisor Einar Steingrímsson, who put his trust in me and gave me the opportunity to study mathematics on a postgraduate level. This work has benefited from his knowledge, enthusiasm and generosity.

References

[1] E. Babson and E. Steingrímsson. Generalized permutation patterns and a classification of the Mahonian statistics. *Séminaire Lotharingien de Combinatoire*, B44b:18pp, 2000.
[2] E. Deutsch. Dyck path enumeration. *Discrete Math.*, 204(1-3):167–202, 1999.
[3] P. Flajolet and R. Schott. Non–overlapping partitions, continued fractions, Bessel functions and a divergent series. *European Journal of Combinatorics*, 11:421–432, 1990.
[4] D. E. Knuth. *The art of computer programming*, volume 1. Addison-Wesley, 1973.
[5] R. Simion and F. W. Schmidt. Restricted permutations. *European Journal of Combinatorics*, 6:383–406, 1985.
[6] N. J. A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences*. Academic Press, 1995. http://www.research.att.com/~njas/sequences/.
[7] R. P. Stanley. *Enumerative Combinatorics*, volume 1. Cambridge University Press, 1997.
[8] J. West. Generating trees and the Catalan and Schröder numbers. *Discrete Mathematics*, 146:247–262, 1995.

Matematik, Chalmers tekniska högskola och Göteborgs universitet, S-412 96 Göteborg, Sweden

E-mail address: claesson@math.chalmers.se

Gončarov Polynomials and Parking Functions

JOSEPH P. S. KUNG
Department of Mathematics, University of North Texas, Denton, TX 76203, U.S.A.
E-mail: kung@unt.edu

and

CATHERINE YAN [1]
Department of Mathematics, Texas A & M University, College Station, TX 77843, U.S.A.
E-mail: cyan@math.tamu.edu

Proposed running head: Gončarov Polynomials

[1]Supported by NSF grant DMS-0070574. Part of this research was carried out at the Institute for Advanced Study and was supported by NSF grant DMS-9729992.

Abstract

Let $\mathbf{u}$ be a sequence of non-decreasing positive integers. A $\mathbf{u}$-parking function of length n is a sequence $(x_1, x_2, \ldots, x_n)$ whose order statistics (the sequence $(x_{(1)}, x_{(2)}, \ldots, x_{(n)})$ obtained by rearranging the original sequence in non-decreasing order) satisfy $x_{(i)} \leq u_i$. The Gončarov polynomials $g_n(x; a_0, a_1, \ldots, a_{n-1})$ are polynomials defined by the biorthogonality relation:

$$\varepsilon(a_i) D^i g_n(x; a_0, a_1, \ldots, a_{n-1}) = n! \delta_{in},$$

where $\varepsilon(a)$ is evaluation at a. Gončarov polynomials form a "natural basis" of polynomials for working with $\mathbf{u}$-parking functions. For example, the number of $\mathbf{u}$-parking functions of length n is $(-1)^n g_n(0; u_1, u_2, \ldots, u_n)$. Gončarov polynomials also satisfy a linear recursion obtained by expanding x^n as a linear combination of Gončarov polynomials. The combinatorial structure underlying this recursion is a decomposition of an arbitrary sequence of positive integers into two subsequences: a "maximum" $\mathbf{u}$-parking function and a subsequence consisting of terms of higher values. From this combinatorial decomposition, we derive linear recursions for sum enumerators, expected sums of $\mathbf{u}$-parking functions, and higher moments of sums of $\mathbf{u}$-parking functions. These recursions yield explicit formulas for these quantities in terms of Gončarov polynomials.

Key Words: Gončarov polynomials, parking functions, linear recurrence, sum enumerators, factorial moments

Corresponding author:
Catherine H. Yan
Department of Mathematics, Texas A&M University, College Station, TX 77843-3368
E-mail: cyan@math.tamu.edu

1 Introduction

We shall think of finite sequences $(x_1, x_2, \ldots, x_n)$ interchangably as sequences and functions with domain $\{1, 2, \ldots, n\}$. If $(x_1, x_2, \ldots, x_n)$ is a sequence of real numbers of length n, then the sequence $(x_{(1)}, x_{(2)}, \ldots, x_{(n)})$ of *order statistics* is obtained by rearranging the original sequence $(x_1, x_2, \ldots, x_n)$ in non-decreasing order. Let $\mathbf{u}$ be a non-decreasing sequence $(u_1, u_2, u_3, \ldots)$ of positive integers. A $\mathbf{u}$-*parking function* of length n is a sequence $(x_1, x_2, \ldots, x_n)$ of length n whose sequence of order statistics satisfies $x_{(i)} \leq u_i$.

We shall call $(1, 2, 3, \ldots)$-parking functions *ordinary* parking functions. Intuitively, an ordinary parking function can take on as many smaller values as one wishes, but it cannot take on too many larger values. Ordinary parking functions originated in the theory of hashing and searching in computer science (see [11, 9]). They have been extensively studied. In particular, it is known that the number of ordinary parking functions of length n is

$$(n+1)^{n-1},$$

a formula which is closely related to Cayley's formula for the number of labelled trees. This relation with trees had motivated much work in this area, particularly in finding bijections between ordinary parking functions and labelled trees. Less obvious, perhaps, is the observation that the formula is (up to a sign) an evaluation of an Abel polynomial. It is this observation which led us to Gončarov polynomials.

Gončarov polynomials (see [1, 2, 7]) arose in the following special case of Hermite interpolation in numerical analysis.

Gončarov Interpolation. Given two sequences of real or complex numbers $a_0, a_1, \ldots, a_n$ and $b_0, b_1, \ldots, b_n$, find a polynomial $p(x)$ of degree n such that for each $i, 0 \leq i \leq n$, the ith derivative $p^{(i)}(x)$ evaluated at a_i equals b_i.

The natural basis of polynomials for this interpolation problem is the sequence of Gončarov polynomials defined in Section 3. A special case of this is *Abel interpolation*, where the point a_i is the integer i. The Gončarov polynomials for this case are the Abel polynomials.

The appearance of Abel polynomials in both the enumeration of parking functions and Abel interpolation was one of the motivations behind this paper. We shall show that the enumerative theory of ordinary parking functions can be generalized to $\mathbf{u}$-parking functions using Gončarov polynomials. We hope that it will become evident that the Gončarov polynomials are the natural basis of polynomials for working with parking functions, even in the ordinary case. In particular, we shall give explicit linear recursions which would allow one to compute any specific moment of the sum of a random $\mathbf{u}$-parking function of length n.

The approach in this paper is to apply results about Gončarov polynomials to parking functions. We start with a discussion of a general theory of biorthogonal polynomials in Section 2 and specialize this theory to Gončarov polynomials in Section 3. In Section 4, we present a combinatorial description of the coefficients of Gončarov polynomials in terms of rankings on ordered partitions. The key tool in this paper, a decomposition of an arbitrary sequence of positive integers into two subsequences, a "maximum" $\mathbf{u}$-parking functions of length m and a subsequence all of whose terms are strictly larger than u_m, is given in Section 5. An immediate application yields formulas for the number of parking functions (Section 5). This decomposition also yields results about sum enumerators (Section 6), expected sums (Section 7), and higher moments of sums of parking functions (Sections 11 and 12). In Section 10, we discuss the conjecture that the expected sum is an increasing function of the "gaps" $u_{i+1} - u_i$ in the sequence $\mathbf{u}$. We also derive formulas for the expected sum in the "classical" case when the sequence $\mathbf{u}$ is an arithmetic progression. Two methods are used. The first, involving Abel identities, is presented in Section 8. The second, using a matrix inverse relation, is presented in Section 9. With substantially more work, the matrix method can also be used to obtain formulas for higher moments of sums of classical parking functions. We shall present this in [14]. We end this paper with a brief discussion of variants of parking functions (Section 13) and some historical remarks (Section 14).

We shall use the following notation. If a and b are integers with $a \leq b$, then the *discrete interval* $[a, b]$ is the set $\{a, a+1, a+2, \ldots, b\}$.

2 Sequences of biorthogonal polynomials

We shall need several results about Gončarov polynomials in this paper. Many of these results are special cases of a general algebraic, that is to say, non-analytic, theory of sequences of polynomials biorthogonal to a sequence of linear functionals. Although this theory must be well-known (for some examples, see [1] or [2]), we have not been able to find an explicit description in the literature.

Consider the vector space $\mathcal{P}$ of all polynomials in the variable x over a field F of characteristic zero. Let $D : \mathcal{P} \to \mathcal{P}$ be the differentiation operator. For a scalar a in the field F, let

$$\varepsilon(a) : \mathcal{P} \to F,\ p(x) \mapsto p(a)$$

be the linear functional which evaluates $p(x)$ at a.

Let $\varphi_s(D), s = 0, 1, 2, \ldots$ be a sequence of linear operators on $\mathcal{P}$ of the form

$$\varphi_s(D) = D^s \sum_{r=0}^{\infty} b_{sr} D^r,$$

where the coefficients b_{s0} are assumed to be non-zero. Note that, although $\varphi_s(D)$ are infinite formal sums, they become finite sums when applied to a specific polynomial. Then there exists a unique sequence $p_n(x), n = 0, 1, 2, \ldots$ of polynomials such that $p_n(x)$ has degree n and

$$\varepsilon(0)\varphi_s(D)p_n(x) = n!\delta_{sn}, \tag{2.1}$$

where δ_{sn} is the Kronecker delta. To see this, let

$$p_n(x) = \sum_{k=0}^{n} c_{nk} x^k.$$

Then, for a given index n, the orthogonality relations are equivalent to the following upper triangular system of linear equations in the unknowns $c_{n,0}, c_{n,1}, c_{n,2}, \ldots, c_{n,n}$:

$$\begin{aligned}
b_{00}c_{n0} + b_{01}c_{n1} + 2!b_{02}c_{n2} + 3!b_{03}c_{n3} + \ldots + \quad n!b_{0n}c_{nn} &= 0\\
b_{10}c_{n1} + 2!b_{11}c_{n2} + 3!b_{12}c_{n3} + \ldots + n!b_{1,n-1}c_{nn} &= 0\\
2!b_{20}c_{n2} + 3!b_{21}c_{n3} + \ldots + n!b_{2,n-2}c_{nn} &= 0\\
&\ldots\\
n!b_{n0}c_{nn} &= n!.
\end{aligned}$$

This system of linear equations can be solved uniquely for every index n. Hence, the polynomials $p_n(x)$ exist and they are uniquely determined by the orthogonality relations (2.1). Note also that $p_n(x)$ depends only on the operators $\varphi_0(D), \varphi_1(D), \ldots, \varphi_{n-1}(D)$. When solving this system, we need only divide by the diagonal entries b_{s0}. Hence, if we put on the extra assumption that the entries b_{s0} all equal 1, then $p_n(x)$ is monic and the coefficients of $p_n(x)$ are polynomials in the entries b_{sr}.

The polynomial sequence $p_n(x)$ is said to be *biorthogonal* to the sequence $\varphi_s(D)$ of operators, or, as some would prefer, the sequence $\varepsilon(0)\varphi_s(D)$ of linear functionals. Using Cramer's rule to solve the linear system and Laplace's expansion to group the results, we obtain the following *determinantal formula*:

$$p_n(x) = \frac{n!}{b_{00}b_{10}\cdots b_{n0}} \begin{vmatrix}
b_{00} & b_{01} & b_{02} & \ldots & b_{0,n-1} & b_{0n}\\
0 & b_{10} & b_{11} & \ldots & b_{1,n-2} & b_{1,n-1}\\
0 & 0 & b_{20} & \ldots & b_{2,n-3} & b_{2,n-2}\\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\
0 & 0 & 0 & \ldots & b_{n-1,0} & b_{n-1,1}\\
1 & x & x^2/2! & \ldots & x^{n-1}/(n-1)! & x^n/n!
\end{vmatrix}. \tag{2.2}$$

Another important consequence of the fact that the initial segment $\varphi_s(D), s = 0, 1, 2, \ldots, n$ gives a non-singular upper triangular system of linear equations is that if $p(x)$ is a degree-n polynomial, then the conditions

$$\varepsilon(0)\varphi_i(D)p(x) = 0 \quad \text{for} \quad 0 \leq i \leq n$$

imply that $p(x)$ is identically zero. In particular, if $p(x)$ has degree n, then

$$p(x) = \sum_{i=0}^{n} \frac{\varepsilon(0)\varphi_i(D)p(x)}{i!} p_i(x). \tag{2.3}$$

This gives an *expansion formula*. Furthermore, the unique solution to the interpolation problem, *given numbers $d_0, d_1, \ldots, d_n$, find a degree-n polynomial $p(x)$ such that for $i = 0, 1, \ldots, n$,*

$$\varepsilon(0)\varphi_i(D)p(x) = d_i,$$

is given by the formula

$$p(x) = \sum_{i=0}^{n} \frac{d_i p_i(x)}{i!}. \tag{2.4}$$

Since

$$\varepsilon(0)\varphi_i(D)x^n = n! b_{i,n-i},$$

a special case of equation (2.3) or equation (2.4) is

$$x^n = \sum_{i=0}^{n} \frac{n! b_{i,n-i} p_i(x)}{i!}. \tag{2.5}$$

Equation (2.5) gives a *linear recursion* for $p_n(x)$. These linear recursions are perhaps the most efficient way to calculate the sequence $p_n(x)$ explicitly on a computer. Multiplying these equations by $t^n/n!$, summing over all non-negative integers n, and rearranging the right-hand side into products, we obtain the following formal power series equation (which is an instance of what one might called an *Appell relation*):

$$e^{xt} = \sum_{n=0}^{\infty} \frac{p_n(x)\varphi_n(t)}{n!}. \tag{2.6}$$

Another way to prove the Appell relation (2.6) is to observe that when one applies $\varphi_s(D)$ to both sides, one obtains the same result. Observe also that when restricted to the subspace $\mathcal{P}_m$ of all polynomials of degree less than or equal to m in $\mathcal{P}$, the operators D^s are expressible as linear combinations of the operators $\varphi_t(D), t = 0, 1, 2, \ldots, m$. Hence, one also obtains the same result when D^s is applied to both sides of the Appell relation, that is, the coefficient of x^s are the same on both sides.

We end with a matrix version of the linear recursion. We can rewrite the first $n+1$ instances of equation (2.5) as the matrix equation

$$\overrightarrow{x^i} = \mathcal{B}\,\overrightarrow{p_i(x)},$$

where

$$\overrightarrow{x^i} = [1, x, x^2, \ldots, x^n]^T,$$

$$\overrightarrow{p_i(x)} = [p_0(x), p_1(x), p_2(x), \ldots, p_n(x)]^T,$$

and $\mathcal{B}$ is the $(n+1) \times (n+1)$ lower triangular matrix

$$\left[\binom{i}{j} (i-j)! b_{j,i-j} \right]_{0 \leq i,j \leq n}.$$

We use the convention that the binomial coefficient $\binom{i}{j}$ is zero if $j > i$. For example, when $n = 3$, we have

$$\begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ b_{01} & 1 & 0 & 0 \\ 2b_{02} & 2b_{11} & 1 & 0 \\ 6b_{03} & 6b_{12} & 3b_{21} & 1 \end{bmatrix} \begin{bmatrix} 1 \\ p_1(x) \\ p_2(x) \\ p_3(x) \end{bmatrix}$$

However, we also have

$$\overrightarrow{p_i(x)} = \mathcal{C}\, \overrightarrow{x^i},$$

where $\mathcal{C}$ is the $(n+1) \times (n+1)$ lower triangular *coefficient matrix*

$$[c_{ij}]_{0 \le i,j \le n}$$

whose entries c_{ij} are coefficients of the polynomials $p_i(x)$. We use the convention that c_{ij} is zero when $j > i$. Hence, we conclude that the two lower triangular matrices $\mathcal{B}$ and $\mathcal{C}$ are inverses of each other. In particular,

$$\overrightarrow{p_i(x)} = \mathcal{B}^{-1}\, \overrightarrow{x^i}. \tag{2.7}$$

This gives a determinantal formula for $p_n(x)$ which is row and column reducible to equation (2.2).

Summarizing, we have shown that the biorthogonality relations, the linear recursions, the Appell relation, and the matrix form of the linear recursions all define the same sequence $p_n(x)$ of polynomials.

Sequences of polynomials of binomial type are special cases of sequences of biorthogonal polynomials. We shall use a description of polynomials of binomial type given in the classic paper of Mullin and Rota [16]. Recall that a sequence $p_n(x)$ of polynomials is of binomial type if and only if

$$\sum_{n=0}^{\infty} p_n(x) \frac{t^n}{n!} = e^{xf(t)}, \tag{2.8}$$

for some formal power series $f(t)$ such that $f(0) = 0$ and $Df(0) \neq 0$. These conditions are equivalent to the condition that $f(t)$ have a compositional inverse in the ring of formal power series. Let $g(t)$ be the compositional inverse of $f(t)$. Then, substituting $g(t)$ for t in equation (2.8), we obtain the Appell relation

$$e^{xt} = \sum_{n=0}^{\infty} p_n(x) \frac{[g(t)]^n}{n!}.$$

From this, we conclude that sequences of polynomials of binomial type are precisely sequences of polynomials biorthogonal to operator sequences of the form

$$\varphi_s(D) = [g(D)]^s,$$

where $g(t)$ is a formal power series with $g(0) = 0$ and $Dg(0) \neq 0$.

3 Algebraic properties of Gončarov polynomials

Let $(a_0, a_1, a_3, \dots)$ be a sequence of numbers or variables called *nodes*. The sequence of *Gončarov polynomials*

$$g_n(x; a_0, a_1, \dots, a_{n-1}),\ n = 0, 1, 2, \dots$$

is the sequence of polynomials biorthogonal to the operators

$$E^{a_s} D^s,$$

where for any number or variable a, the operator E^a is the shift by a, that is,

$$E^a p(x) = p(x + a).$$

Because $\varepsilon(0)E^a = \varepsilon(a)$, the sequence of Gončarov polynomials $g_n(x; a_0, a_1, \ldots, a_{n-1})$ are defined by the orthogonality relations

$$\varepsilon(a_s)D^s g_n(x; a_0, a_1, \ldots, a_{n-1}) = n!\delta_{sn}.$$

Since

$$E^a = \sum_{r=0}^{\infty} \frac{a^r D^r}{r!} = e^{a_i D},$$

the sequence of Gončarov polynomials is biorthogonal to the sequence

$$D^s \sum_{r=0}^{\infty} \frac{a_s^r D^r}{r!}.$$

As indicated by the notation, $g_n(x; a_0, a_1, \ldots, a_{n-1})$ depends only on the nodes $a_0, a_1, \ldots, a_{n-1}$. Indeed, from equation (2.2), we have the *determinantal formula*,

$$g_n(x; a_0, a_1, \ldots, a_{n-1}) = n! \begin{vmatrix} 1 & a_0 & \frac{a_0^2}{2!} & \frac{a_0^3}{3!} & \cdots & \frac{a_0^{n-1}}{(n-1)!} & \frac{a_0^n}{n!} \\ 0 & 1 & a_1 & \frac{a_1^2}{2!} & \cdots & \frac{a_1^{n-2}}{(n-2)!} & \frac{a_1^{n-1}}{(n-1)!} \\ 0 & 0 & 1 & a_2 & \cdots & \frac{a_2^{n-3}}{(n-3)!} & \frac{a_2^{n-2}}{(n-2)!} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & 1 & a_{n-1} \\ 1 & x & \frac{x^2}{2!} & \frac{x^3}{3!} & \cdots & \frac{x^{n-1}}{(n-1)!} & \frac{x^n}{n!} \end{vmatrix}.$$

From equations (2.5) and (2.6), we have the *linear recursion*

$$x^n = \sum_{i=0}^{n} \binom{n}{i} a_i^{n-i} g_i(x; a_0, a_1, \ldots, a_{i-1})$$

and the *Appell relation*

$$e^{xt} = \sum_{n=0}^{\infty} g_n(x; a_0, a_1, \ldots, a_{n-1}) \frac{t^n e^{a_n t}}{n!}.$$

Finally, from equation (2.3), we have the *expansion formula*. If $p(x)$ is a polynomial of degree n, then

$$p(x) = \sum_{i=0}^{n} \frac{\varepsilon(a_i)D^i p(x)}{i!} g_i(x; a_0, a_1, \ldots, a_{i-1}).$$

We turn now to properties specific to the sequence of Gončarov polynomials. The Gončarov polynomials can be equivalently defined by the *differential relations*

$$D g_n(x; a_0, a_1, \ldots, a_{n-1}) = n g_{n-1}(x; a_1, a_2, \ldots, a_{n-1}),$$

with initial conditions

$$g_n(a_0; a_0, a_1, \ldots, a_{n-1}) = \delta_{0n}.$$

(To see this, check that the orthogonality relations are satisfied.) Integrating the differential relations, we obtain the *integral relation*

$$g_n(x; a_0, a_1, \ldots, a_{n-1}) = n \int_{a_0}^{x} g_{n-1}(t; a_1, a_2, \ldots, a_{n-1}) dt.$$

Iterating this, we obtain the *integral formula*

$$g_n(x; a_0, a_1, \ldots, a_{n-1}) = n! \int_{a_0}^{x} dt_1 \int_{a_1}^{t_1} dt_2 \cdots \int_{a_{n-1}}^{t_{n-1}} dt_n.$$

The integral relation makes it clear (by induction) that $g_n(x; a_0, a_1, \ldots, a_{n-1})$ is a homogeneous polynomial with integer coefficients in the variables $x, a_0, a_1, \ldots, a_{n-1}$ of total degree n. It also gives a quick way to calculate Gončarov polynomials of low degree by hand. For example,

$$\begin{aligned} g_0(x) &= 1, \\ g_1(x; a_0) &= x - a_0, \\ g_2(x; a_0, a_1) &= x^2 - 2a_1x + 2a_0a_1 - a_0^2, \\ g_3(x; a_0, a_1, a_2) &= x^3 - 3a_2x^2 + (6a_1a_2 - 3a_1^2)x - a_0^3 + 3a_0^2a_2 - 6a_0a_1a_2 + 3a_0a_1^2. \end{aligned}$$

Using a change of variable, the integral relation and induction, or, observing that the differential operator is "shift-invariant" or commutes with shifts, one obtains the following useful *shift formula:*

$$g_n(x + \xi; a_0 + \xi, a_1 + \xi, \ldots, a_{n-1} + \xi) = g_n(x; a_0, a_1, \ldots, a_{n-1}).$$

The integral formula also suggests a formula which shows the effect of shifting or perturbing a single node. Using the identity

$$\int_{a_m}^{t} F(t)dt = \int_{a_m}^{a_m+b_m} F(t)dt + \int_{a_m+b_m}^{t} F(t)dt$$

at the mth integral in the integral formula, we obtain the *perturbation formula:*

$$\begin{aligned} & g_n(x; a_0, \ldots, a_{m-1}, a_m + b_m, a_{m+1}, \ldots, a_{n-1}) = g_n(x; a_0, \ldots a_{m-1}, a_m, a_{m+1}, \ldots, a_{n-1}) \\ - \quad & \binom{n}{m} g_{n-m}(a_m + b_m; a_m, a_{m+1}, \ldots, a_{n-1}) g_m(x; a_0, a_1, \ldots, a_{m-1}). \end{aligned}$$

Applying the perturbation formula repeatedly, we can perturb any subset of nodes. For example, the following formula allows us to perturb an initial segment of length $n - m + 1$:

$$\begin{aligned} & g_n(x; a_0 + b_0, a_1 + b_1, \ldots, a_{n-m} + b_{n-m}, a_{n-m+1}, \ldots, a_{n-1}) \\ = \quad & g_n(x; a_0, a_1, \ldots, a_{n-m}, a_{n-m+1}, \ldots, a_{n-1}) \\ & - \sum_{i=0}^{n-m} \binom{n}{i} g_{n-i}(a_i + b_i; a_i, a_{i+1}, \ldots, a_{n-1}) g_i(x; a_0 + b_0, a_1 + b_1, \ldots, a_{i-1} + b_{i-1}). \end{aligned}$$

In general, perturbation formulas can also be obtained by expanding the unperturbed polynomial $g_n(x; a_0, a_1, \ldots, a_{n-1})$ as a series in suitably perturbed Gončarov polynomials.

In general, there are no nice closed-form expressions for Gončarov polynomials. But such expressions exist for two special cases studied in analysis. The first is the case when all the nodes a_i equals a. In this case,

$$g_n(x; a, a, \ldots, a) = (x - a)^n$$

and Gončarov interpolation is just expansion as a power series at $x = a$. For this case, the linear recursion specializes to the binomial identity

$$x^n = \sum_{i=0}^{n} \binom{n}{i} a^{n-i}(x - a)^i,$$

The second case (which includes the first as a special case) is when $a_0, a_1, a_2, \ldots$ form an arithmetic progression. This is the case of *Abel polynomials* and we have

$$g_n(x; y, y + b, y + 2b, \ldots, y + (n - 1)b) = (x - y)(x - y - nb)^{n-1}. \tag{3.1}$$

In particular,

$$g_n(x; 0, 1, 2, \ldots, n - 1) = x(x - n)^{n-1}.$$

The linear recursion is

$$x^n = \sum_{i=0}^{n} \binom{n}{i} (y+ib)^{n-i}(x-y)(x-y-ib)^{i-1}.$$

Substituting $x+y$ for x in the second identity, we obtain *Abel's binomial theorem,*

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} (y+ib)^{n-i}x(x-ib)^{i-1}.$$

With the substitution $x+y+nb$ for x, $y+nb$ for y, and $-b$ for b, we obtain *Hurwitz's versions* of Abel's binomial theorem:

$$(x+y+nb)^n = \sum_{i=0}^{n} \binom{n}{i} (y+(n-i)b)^{n-i}x(x+ib)^{i-1},$$

or, changing indices from i to $n-i$,

$$(x+y+nb)^n = \sum_{i=0}^{n} \binom{n}{i} (y+ib)^{i}x(x+(n-i)b)^{n-i-1}. \tag{3.2}$$

Differentiating both sides of equation (3.2) with respect to y, we obtain

$$\begin{aligned} n(x+y+nb)^{n-1} &= \sum_{i=1}^{n} \binom{n}{i} i(y+ib)^{i-1}x(x+(n-i)b)^{n-i-1} \\ &= \sum_{i=1}^{n} \binom{n}{i-1} (n-i+1)(y+ib)^{i-1}x(x+(n-i)b)^{n-i-1}. \end{aligned}$$

Taking the case $n-1$ of this identity and setting $x=b$ and $y=a$, we obtain

$$(n-1)(a+nb)^{n-2} = \sum_{i=1}^{n-1} \binom{n-1}{i-1} b^{n-i-1}(n-i)^{n-i-1}(a+ib)^{i-1}. \tag{3.3}$$

We shall need identity (3.3) in Section 8.

Abel's binomial theorem is a member of a family of Abel identities studied by Hurwitz, Riordan and others (see [21], pp. 18 to 22). The following identity is a slightly modified version of the identity called $A_n(x,y;1,-1)$ from this family:

$$\sum_{i=0}^{n} \binom{n}{i} (x+ib)^{i+1}y(y+(n-i)b)^{n-i-1} = \sum_{i=0}^{n} \frac{n!}{i!}(x+y+nb)^i b^{n-i}(x+(n-i)b).$$

We shall use two special cases of this Abel identity in Section 8.

A proof of this identity can be found in [21], but it is part of a larger proof and difficult to extract. For this reason, we provide a simple self-contained proof. Observe that the left hand side is an expansion in terms of Abel polynomials $y(y+mb)^{m-1}$ in y with nodes at $-mb$. Hence, the identity follows from the following computation, where D_y is differentiation with respect to y :

$$\begin{aligned} & \varepsilon(-(n-i)b)D_y^{n-i}\left(\sum_{j=0}^{n} \frac{n!}{j!}(x+y+nb)^j b^{n-j}(x+(n-j)b)\right) \\ = \; & n!\sum_{k=0}^{i} \frac{b^{i-k}(x+ib)^k(x+(i-k)b)}{k!}. \end{aligned}$$

The zeroth term in the sum is $b^i(x+ib)$. When $k>0$, we can rewrite the kth term as

$$\frac{b^{i-k}(x+ib)^{k+1}}{k!} - \frac{b^{i-(k-1)}(x+ib)^k}{(k-1)!}.$$

Hence, the sum telescopes and the right hand side equals

$$\frac{n!(x+ib)^{i+1}}{i!}.$$

The identity now follows from the expansion formula.

The first special case is obtained by setting $x = a + b$ and $y = b$ in the case $n - 2$ of the Abel identity. Doing so, we obtain

$$\sum_{i=0}^{n-2} \binom{n-2}{j} (a + (j+1)b)^{j+1} b^{n-j-3} (n-1-i)^{n-j-3}$$

$$= \sum_{i=0}^{n-2} \frac{(n-2)!}{j!} (a+nb)^j b^{n-j-3} (a + (n-j-1)b). \tag{3.4}$$

Setting $x = a$ and $y = 0$ on the left hand side, we obtain the second special case:

$$(a+nb)^{n+1} = \sum_{i=0}^{n} \frac{n!}{i!} (a+nb)^i b^{n-i} (a + (n-i)b),$$

or, changing indices from i to $n - i$,

$$(a+nb)^{n+1} = \sum_{i=0}^{n} \frac{n!}{(n-i)!} (a+nb)^{n-i} b^i (a+ib). \tag{3.5}$$

4 Coefficients of Gončarov polynomials

The main result in this section is a combinatorial interpretation of the coefficients of Gončarov polynomials. We first show that it suffices to consider only the constant terms.

Expanding $g_n(x + y; a_0, \ldots, a_{n-1})$ as a Taylor expansion in x and using the differential relations, we obtain

$$g_n(x+y; a_0, a_1, \ldots, a_{n-1}) = \sum_{i=0}^{n} \binom{n}{i} g_{n-i}(y; a_i, a_{i+1}, \ldots, a_{n-1}) x^i. \tag{4.1}$$

This is a shifted or parametrized analogue of a Sheffer relation, but *not* an actual Sheffer relation unless all the nodes a_i are equal. Thus, the Gončarov polynomials may be viewed as a "shifted" Sheffer sequence for the operator D (see [17]). The beginnings of a theory of "shifted" or "decentralized" umbral calculus has been developed in [22].

Setting $y = 0$ in equation (4.1), we obtain

$$g_n(x; a_0, a_1, \ldots, a_{n-1}) = \sum_{i=0}^{n} \binom{n}{i} g_{n-i}(0; a_i, a_{i+1}, \ldots, a_{n-1}) x^i. \tag{4.2}$$

Thus, coefficients of Gončarov polynomials are constant terms of (shifted) Gončarov polynomials. In particular, we have the following special case of equation (2.7).

(4.1) Lemma. Let $\mathcal{A}$ be the lower triangular matrix

$$\left[\binom{i}{j} a_j^{i-j} \right]_{0 \le i,j \le n}.$$

Then, its inverse $\mathcal{A}^{-1}$ is the lower triangular coefficient matrix

$$\left[\binom{i}{j} g_{i-j}(0; a_j, a_{j+1}, \ldots, a_{i-1})\right]_{0 \le i,j \le n}.$$

In particular,

$$\mathcal{A}^{-1}\overrightarrow{x^i} = \overrightarrow{g_i(x; a_0, a_1, \ldots, a_{n-1})}.$$

We shall now give a combinatorial interpretation of the constant terms of Gončarov polynomials. This interpretation is obtained by considering the number f_n of monomials in the constant term $g_n(0; a_0, a_1, \ldots, a_{n-1})$, counted with multiplicity. The sequence f_n starts $1, 1, 3, 13, 75, \ldots$. Using, say, the integral relation, it is easy to show that the numbers f_n satisfy the recurrence

$$f_n = \sum_{i=1}^{n} \binom{n}{i} f_{n-i}$$

and have exponential generating function

$$\sum_{n=0}^{\infty} \frac{f_n t^n}{n!} = \frac{1}{2 - e^t}.$$

From this, we see (from [23], say) that f_n is the number of *preferential arrangements*, or ordered partitions of the set with n elements. These observations suggest that there is an interpretation of the constant term $g_n(0; a_0, a_1, \ldots, a_{n-1})$ in terms of objects related to ordered partitions.

From an ordered partition $B_1, B_2, \ldots, B_m$ of a set $\{x_1, x_2, \ldots, x_n\}$ with n elements, one can associate a *ranking* $\rho : \{x_1, x_2, \ldots, x_n\} \to \{0, 1, 2, \ldots, n-1\}$ as follows: if an element x_i is in the jth block B_j, then defined

$$\rho(x_i) = \sum_{l<j} |B_l|.$$

In particular, $\rho(x_i) = 0$ whenever x_i is in the first block B_1. We define the order $|\rho|$ to be the size of the image of ρ, which is also the number of blocks in the ordered partition associated with ρ. For example, from the ordered partition $\{2, 4\}$, $\{5\}$, $\{1, 3\}$ of $\{1, 2, 3, 4, 5\}$, one obtains the ranking defined by $\rho(2) = \rho(4) = 0$, $\rho(5) = 2$, and $\rho(1) = \rho(3) = 3$. Rankings are characterized by the property: for every element x_i, there are exactly $\rho(x_i)$ elements x_j such that $\rho(x_j) < \rho(x_i)$.

(4.2) Theorem.

$$g_n(0; a_0, a_1, \ldots, a_{n-1}) = \sum_{\rho} (-1)^{|\rho|} \prod_{j=0}^{n-1} a_{\rho(j)},$$

where the sum ranges over all rankings ρ of $\{1, 2, \ldots, n\}$.

Proof. The theorem holds when $n = 0$. When $n > 0$, the constant terms of Gončarov polynomials satisfy the recursion

$$g_n(0; a_0, a_1, \ldots, a_{n-1}) = -\sum_{i=0}^{n-1} \binom{n}{i} a_i^{n-i} g_i(0; a_0, a_1, \ldots, a_{i-1})$$

obtained by setting $x = 0$ in the linear recursion. We shall show that the sum on the right hand side of the equation in Theorem 4.2 satisfies the same recursion. Let $\mathcal{R}[n]$ be the set of all rankings on $\{1, 2, \ldots, n\}$. Divide $\mathcal{R}[n]$ into groups $\mathcal{R}[n, i]$ according to the maximum value i taken by the ranking, so that

$$\mathcal{R}[n, i] = \{\rho : \max\{\rho(1), \rho(2), \ldots, \rho(n)\} = i\}.$$

If ρ is in $\mathcal{R}[n,i]$, then the inverse image $\rho^{-1}(i)$ must contain exactly $n-i$ numbers. Thus, there is a bijection between rankings ρ in $\mathcal{R}_i$ and pairs consisting of an i-element subset of $\{1,2,\ldots,n\}$ (the complement of $\rho^{-1}(i)$) and a ranking ρ' (having order $|\rho|-1$) on that i-element subset obtained by restricting ρ. Hence,

$$\sum_{\rho\in\mathcal{R}[n]}(-1)^{|\rho|}\prod_{j=1}^{n}a_{\rho(j)} = -\sum_{i=0}^{n-1}a_i^{n-i}\binom{n}{i}\left(\sum_{\rho\in\mathcal{R}[n,i]}(-1)^{|\rho'|}\prod_{j=0}^{i-1}a_{\rho(j)}\right).$$

Since both sides of the equation in Theorem 4.2 satisfy the same recursion and initial condition, they are equal by induction.

By Theorem 4.2 and the shift formula, we obtain the following formula for Gončarov polynomials.

$$\begin{aligned} g_n(x;a_0,a_1,\ldots,a_{n-1}) &= g_n(0;a_0-x,\ldots,a_{n-1}-x)\\ &= \sum_{\rho}(-1)^{|\rho|}\prod_{i=1}^{n}(a_{\rho(i)}-x). \end{aligned}$$

Abel polynomials are intimately related to the enumeration of trees. In particular, if one set $a_i=i$, then the constant term $(-1)^n g_n(0;a_0,a_1,\ldots,a_n)$ is the number of labelled trees on $n+1$ vertices. Is there an interpretation for $(-1)^n g_n(0;a_0,a_1,\ldots,a_n)$ in terms of labelled trees?

5 A decomposition for sequences of positive integers

In this section, we describe the combinatorial decomposition underlying the theory of parking functions. For us, this decomposition was motivated by the linear recursion for Gončarov polynomials. After discovering this decomposition, we found out from Julian Gilbey that the special case of this decomposition for ordinary parking functions was already used by Konheim and Weiss in the *first* paper [11] on the subject.

(5.1) Theorem. Let $(u_1,u_2,\ldots,u_n)$ be a sequence of non-decreasing positive integers and let x be a positive integer. Then, every sequence $(x_1,x_2,\ldots,x_n)$ of length n with terms x_i integers from the discrete interval $[1,x]$ can be decomposed into a pair of subsequences

$$(x_{i_1},x_{i_2},\ldots,x_{i_m}),\ (x_{j_1},x_{j_2},\ldots,x_{j_{n-m}})$$

such that the first subsequence $(x_{i_1},x_{i_2},\ldots,x_{i_m})$ is a **u**-parking function of length m, and all the terms in the second subsequence, the complementary subsequence of length $n-m$ obtained by removing the terms in the first subsequence from $(x_1,x_2,\ldots,x_n)$ are in the discrete interval $[u_{m+1}+1,x]$. This decomposition provides a bijection between all sequences of length n with terms in $[1,x]$ and all pairs of complementary subsequences, the first a **u**-parking function of length m and the second a sequence of length $n-m$ taking values in $[u_{m+1}+1,x]$.

Proof. Consider the sequence $(x_{(1)},x_{(2)},\ldots,x_{(n)})$ of order statistics. Let m be the maximum index such that

$$x_{(i)}\le u_i \quad \text{for} \quad i=1,2,\ldots,m. \tag{5.1}$$

Then, the subsequence $(x_{i_1},x_{i_2},\ldots,x_{i_m})$ from which the sequence $(x_{(1)},x_{(2)},\ldots,x_{(m)})$ was obtained by rearrangement is a **u**-parking function of length m. Furthermore, m is the maximum index satisfying condition (5.1) if and only if

$$x_{(n)}\ge x_{(n-1)}\ge\ldots\ge x_{(m+1)}>u_{m+1},$$

or, equivalently, the *complementary* subsequence $(x_{j_1},x_{j_2},\ldots,x_{j_{n-m}})$, obtained by deleting the subsequence $(x_{i_1},x_{i_2},\ldots,x_{i_m})$ from the original sequence, takes values in the interval $[u_{m+1}+1,x]$. Since the maximum

index m and hence, the set $\{i_1, i_2, \ldots, i_m\}$ are uniquely determined by the sequence $(x_1, x_2, \ldots, x_n)$, and any pair of subsequences satisfying the conditions in the theorem can be reassembled into a sequence in $[1, x]^n$, this decomposition yields a bijection.

It will be useful to state the decomposition more explicitly.

(5.2) Corollary. There is a bijection between the set $[1, x]^n$ of all length-n integer sequences with terms in the discrete interval $[1, x]$ and the disjoint union of cartesian products

$$\bigcup_{\{i_1, i_2, \ldots, i_m\}} \mathrm{Park}(i_1, i_2, \ldots, i_m) \times [u_{m+1} + 1, x]^{n-m},$$

where $\mathrm{Park}(i_1, i_2, \ldots, i_m)$ is the set of length-m $\mathbf{u}$-parking functions indexed by the set $\{i_1, i_2, \ldots, i_m\}$ and $[u_{m+1} + 1, x]^{n-m}$ is the set of length-$(n - m)$ integer sequences with terms in $[u_{m+1} + 1, x]$ indexed by the complement of $\{i_1, i_2, \ldots, i_m\}$.

Let $P_n(\mathbf{u})$ be the number of $\mathbf{u}$-parking functions of length m. Since $P_n(\mathbf{u})$ depends only on the first n terms of $\mathbf{u}$, we will often write $P_n(u_1, u_2, \ldots, u_n)$ instead of $P_n(\mathbf{u})$ to make explicit the parameters on which $P_n(\mathbf{u})$ is dependent. The decomposition in Theorem 5.1 yields the following identity.

(5.3) Corollary. Let x be an integer greater than or equal to u_n. Then

$$x^n = \sum_{m=0}^{n} \binom{n}{m} (x - u_{m+1})^{n-m} P_m(u_1, u_2, \ldots, u_m).$$

Comparing the recursion in Corollary 5.3 with the linear recursion for Gončarov polynomials given in Section 3, we obtain

$$P_n(u_1, u_2, \ldots, u_n) = g_n(x; x - u_1, x - u_2, \ldots, x - u_n).$$

By the shift formula, the Gončarov polynomial equals

$$g_n(0; -u_1, -u_2, \ldots, -u_n).$$

Since the Gončarov polynomial $g_n(x; a_0, a_1, \ldots, a_{n-1})$ is a homogeneous polynomial of total degree n in $x, a_0, a_1, \ldots, a_{n-1}$, we have

$$g_n(0; -u_1, -u_2, \ldots, -u_n) = (-1)^n g_n(0; u_1, u_2, \ldots, u_n).$$

All three forms of the formula for $P_n(\mathbf{u})$ are useful.

(5.4) Theorem.

$$\begin{aligned} P_n(u_1, u_2, \ldots, u_n) &= g_n(x; x - u_1, x - u_2, \ldots, x - u_n) \\ &= g_n(0; -u_1, -u_2, \ldots, -u_n) \\ &= (-1)^n g_n(0; u_1, u_2, \ldots, u_n). \end{aligned}$$

When $u_i = a + (i - 1)b$, we obtain the following special case.

(5.5) Corollary.

$$P_n(a, a + b, a + 2b, \ldots, a + (n - 1)b) = a(a + nb)^{n-1}.$$

In particular, we have rederived the classic formula for ordinary parking functions:

$$P_n(1, 2, 3, \ldots, n) = (n + 1)^{n-1}.$$

From the fact that Gončarov polynomials are homogeneous, we obtain another consequence of Theorem 5.4.

(5.6) Corollary.

$$P_n(bu_1, bu_2, \ldots, bu_n) = b^n P(u_1, u_2, \ldots, u_n).$$

Any reasonable formula for Gončarov polynomials yields a reasonable formula for parking functions. We give an example which is motivated by results in [17] and [31]. Consider the sequence $a_0, a_1, \ldots, a_{n-m}, c + (n-m+1)d, c + (n-m+2)d, \ldots, c + (n-1)d$ of n nodes. This sequence can be obtained by perturbing the arithmetic progression $c, c+d, \ldots, c+(n-1)d$ by $b_i = a_i - (c + id)$ for $i = 0, 1, \ldots, n-m$. Using the perturbation formula, we have

$$\begin{aligned} & g_n(x; a_0, a_1, \ldots, a_{n-m}, c + (n-m+1)d, c + (n-m+2)d, \ldots, c + (n-1)d) \\ = \; & (x-c)(x-c-nd)^{n-1} \\ & - \sum_{i=0}^{n-m} \binom{n}{i} (a_i - c - id)(a_i - c - nd)^{n-i-1} g_i(x; a_0, a_1, \ldots, a_{i-1}). \end{aligned}$$

Using this and Theorem 5.4, we obtain the following result.

(5.7) Corollary. If $c + (n-m+1)d \geq a_{n-m}$, then

$$\begin{aligned} & P_n(u_1, u_2, \ldots, u_{n-m+1}, c + (n-m+1)d, c + (n-m+2)d, \ldots, c + (n-1)d) \\ = \; & c(c+nd)^{n-1} - \sum_{i=0}^{n-m} \binom{n}{i} (c + id - u_{i+1})(c + id - u_{i+1})^{n-i-1} P_i(u_1, u_2, \ldots, u_i). \end{aligned}$$

Note that c need not be positive and some of the terms in the sum may be negative in Corollary 5.7.

By the determinantal formula for Gončarov polynomials in Section 3, we have the discrete analog of a result for real-valued parking functions usually attributed to Steck [28].

(5.8) Corollary. The number $P_n(u_1, u_2, \ldots, u_n)$ of $\mathbf{u}$-parking functions of length n equals $(-1)^n n! \det \mathcal{D}$, where $\mathcal{D}$ is the matrix with ijth entry equal to

$$\frac{u_i^{j-i+1}}{(j-i+1)!}$$

if $j - i + 1 \geq 0$ and 0 otherwise.

Note that Lemma 4.1 and Jacobi's formula for the inverse of a matrix yields another determinantal formula for $P_n(\mathbf{u})$. However, this formula can easily be derived from the formula in Corollary 5.8 by row and column operations.

6 Sum enumerators of parking functions

In this section, we extend several results for enumerators of trees and ordinary parking functions to $\mathbf{u}$-parking functions. Let $\mathbf{u}$ be a sequence of non-decreasing positive integers. The *sum enumerator* $S_n(q; \mathbf{u})$

for the set of **u**-parking functions is the polynomial in q defined by

$$S_n(q;\mathbf{u}) = \sum_{(a_1,a_2,\ldots,a_n)} q^{a_1+a_2+\ldots+a_n-n}$$

where the sum ranges over all **u**-parking functions $(a_1, a_2, \ldots, a_n)$. The sum enumerator may be regarded as a "q-analogue" of $P_n(\mathbf{u})$. The sum enumerator for a subset $\mathcal{S}$ of $[1,x]^n$ is defined analogously by summing over all sequences in $\mathcal{S}$. Sum enumerators are *multiplicative* in the following sense. Suppose that $\mathcal{S}_1$ and $\mathcal{S}_2$ are two sets of subsequences on disjoint index sets. Then the sum enumerator of the cartesian product $\mathcal{S}_1 \times \mathcal{S}_2$ consisting of all sequences formed by combining a subsequence from $\mathcal{S}_1$ and a subsequence from $\mathcal{S}_1$ is the product of the sum enumerators of $\mathcal{S}_1$ and $\mathcal{S}_2$.

For a **u**-parking function, the maximum value of the ith order statistic $x_{(i)}$ is at most u_i and hence, $u_i - x_{(i)} \geq 0$. The *reversed sum enumerator* $R_n(q;\mathbf{u})$ is defined by

$$R_n(q;\mathbf{u}) = \sum_{(a_1,a_2,\ldots,a_n)} q^{u_1+u_2+\ldots+u_n-(a_1+a_2+\ldots+a_n)},$$

where the sum ranges over all **u**-parking functions $(a_1, a_2, \ldots, a_n)$. Equivalently,

$$R_n(q;\mathbf{u}) = q^{u_1+u_2+\ldots+u_n-n} S_n(1/q;\mathbf{u}). \tag{6.1}$$

The reversed sum enumerator is a polynomial in the variable q of degree $u_1 + u_2 + \ldots + u_n - n$.

(6.1) Lemma.

$$(1+q+q^2+\ldots+q^{x-1})^n = \sum_{m=0}^{n} \binom{n}{m} (q^{u_{m+1}} + q^{u_{m+1}+1} + \ldots + q^{x-1})^{n-m} S_m(q;\mathbf{u}).$$

Proof. Since sum enumerators are multiplicative, the sum enumerator of $[1,x]^n$ is

$$(1+q+q^2+\ldots+q^{x-1})^n.$$

For the same reason, the sum enumerator of functions which are decomposed into a **u**-parking function of length m and a sequence in $[u_{m+1}+1, x]^{n-m}$ is

$$(q^{u_{m+1}} + q^{u_{m+1}+1} + \ldots + q^{x-1})^{n-m} S_m(q,\mathbf{u}).$$

The recursion now follows.

Comparing this recursion with the linear recursion in Corollary 5.3, we obtain the following theorem.

(6.2) Theorem.

$$S_n(q;\mathbf{u}) = P_n(1+q+\ldots+q^{u_1-1}, 1+q+\ldots+q^{u_2-1}, \ldots, 1+q+\ldots+q^{u_n-1}).$$

Theorem 6.2 can also be obtain directly using a decomposition for the set of **u**-parking functions due to Pitman and Stanley [19]. Given a **u**-parking function $(\beta_1, \beta_2, \ldots, \beta_n)$, we can associate an ordinary parking function $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ by setting $\alpha_i = r$ if β_i is in the discrete interval $[u_{r-1}+1, u_r]$. Conversely, given an ordinary parking function $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, there are

$$(u_{\alpha_1} - u_{\alpha_1-1})(u_{\alpha_2} - u_{\alpha_2-1}) \cdots (u_{\alpha_n} - u_{\alpha_n-1})$$

u-parking functions associated with it. These are obtained by choosing a number from each discrete interval $[u_{\alpha_j-1}+1, u_{\alpha_j}]$. Here, we use the convention that $u_0 = 0$. Hence,

$$P_n(u_1, u_2, \ldots, u_n) = \sum_{(\alpha_1,\alpha_2,\ldots,\alpha_n)} (u_{\alpha_1} - u_{\alpha_1-1})(u_{\alpha_2} - u_{\alpha_2-1}) \cdots (u_{\alpha_n} - u_{\alpha_n-1}),$$

where the sum ranges over all ordinary parking functions of length n. Replacing the number of elements $u_{\alpha_j} - u_{\alpha_j - 1}$ in the discrete interval $[u_{\alpha_{j-1}} + 1, u_{\alpha_j}]$ by its sum enumerator and using the fact that sum enumerators are multiplicative, we obtain Theorem 6.2.

Using Theorem 5.3, Theorem 6.1, and the shift formula, we can express sum enumerators in terms of Gončarov polynomials:

$$S_n(q; \mathbf{u}) = g_n\left(\frac{1}{1-q}; \frac{q^{u_1}}{1-q}, \frac{q^{u_2}}{1-q}, \ldots, \frac{q^{u_n}}{1-q}\right).$$

By homogeneity of Gončarov polynomials,

$$(1-q)^n S_n(q; \mathbf{u}) = g_n(1; q^{u_1}, q^{u_2}, \ldots, q^{u_n}).$$

Hence, sum enumerators satisfy the simpler linear recursion

$$1 = \sum_{m=0}^{n} \binom{n}{m} q^{u_{m+1}(n-m)} (1-q)^m S_m(q; \mathbf{u}). \tag{6.2}$$

They also satisfy the following Appell relation

$$\exp(t) = \sum_{n=0}^{\infty} (1-q)^n S_n(q; \mathbf{u}) \exp(q^{u_{n+1}} t) \frac{t^n}{n!}.$$

In the case of ordinary parking functions, $u_i = i$ and we have

$$(1-q)^n S_n(q; 1, 2, \ldots, n) = g_n(1; q, q^2, \ldots, q^n).$$

For example,

$$\begin{aligned} (1-q)^2 S_2(q; 1, 2) &= 1 - 3q^2 + 2q^3 \\ (1-q)^3 S_3(q; 1, 2, 3) &= 1 - 4q^3 - 3q^4 + 12q^5 - 6q^6. \end{aligned}$$

One does not expect simple generating functions for sum enumerators in general. However, when u_i is an arithmetic progression, we can group terms together to obtain a recursion which yields a simple exponential generating function. We shall show how this can be done for reversed sum enumerators.

Substituting $1/q$ for q in equation (6.2) and using equation (6.1), we obtain

$$q^{u_1+u_2+\ldots+u_n} = \sum_{m=0}^{n} \binom{n}{m} (q-1)^m R_m(q; \mathbf{u}) q^{-(n-m)u_{m+1} + u_{m+1} + u_{m+2} + \ldots + u_n}.$$

If the exponent

$$-(n-m)u_{m+1} + u_{m+1} + u_{m+2} + \ldots + u_n$$

is a function $\tau(n-m)$ depending only on $n-m$, then we have

$$q^{u_1+u_2+\ldots+u_n} = \sum_{m=0}^{n} \binom{n}{m} (q-1)^m R_m(q; \mathbf{u}) q^{\tau(n-m)}.$$

Multiplying this by $t^n/n!$, summing over all non-negative integers n, and manipulating the resulting formal power series, we obtain

$$\sum_{n=0}^{\infty} (q-1)^n R_n(q; \mathbf{u}) \frac{t^n}{n!} = \frac{\displaystyle\sum_{n=0}^{\infty} q^{u_1+u_2+\ldots+u_n} \frac{t^n}{n!}}{\displaystyle\sum_{n=0}^{\infty} q^{\tau(1)+\tau(2)+\ldots+\tau(n)} \frac{t^n}{n!}}.$$

The condition that the exponent is a function $\tau(n-m)$ of $n-m$ is in fact very strong. Consider the case $n-m=2$. Then the condition implies that for all m, $-2u_{m+1}+u_{m+1}+u_{m+2}$ equals a number $\tau(2)$ independently of m, that is, $u_{m+2}-u_{m+1}$ is a constant b for all m. This in turn implies that $\mathbf{u}$ is an arithmetic progression with common difference b. Conversely, if $u_i = a+(i-1)b$, then

$$\sum_{j=1}^{n} u_j = an + b\binom{n}{2}$$

and

$$\sum_{j=1}^{n} \tau(j) = b\binom{n}{2}.$$

We have thus proved the following theorem, which is best possible.

(6.3) Theorem. Let $\mathbf{u}$ be the arithmetic progression $(a, a+b, a+2b, \dots)$. Then

$$\sum_{n=0}^{\infty}(q-1)^n R_n(q;\mathbf{u})\frac{t^n}{n!} = \frac{\displaystyle\sum_{n=0}^{\infty} q^{an+b\binom{n}{2}}\frac{t^n}{n!}}{\displaystyle\sum_{n=0}^{\infty} q^{b\binom{n}{2}}\frac{t^n}{n!}}.$$

The reversed sum enumerator $R_n(q;\mathbf{u})$ also enumerates the number of inversions for certain sequences of rooted b-forests. For more details about this and the relation between rooted b-forests and generalized parking functions, see [32]. In particular, the reversed sum enumerator $R_n(q;1,2,\dots,n)$ for ordinary parking functions equals the inversion enumerator $I_n(q)$ for labelled trees (see [15, 12, 26, 27]). Hence, we obtain, as a special case of Theorem 6.3, the following result of Stanley ([26, 27]):

$$\sum_{n=0}^{\infty}(q-1)^n I_n(q)\frac{t^n}{n!} = \frac{\displaystyle\sum_{n=0}^{\infty} q^{\binom{n+1}{2}}\frac{t^n}{n!}}{\displaystyle\sum_{n=0}^{\infty} q^{\binom{n}{2}}\frac{t^n}{n!}}.$$

The theory of sum enumerators suggests that Gončarov polynomials with nodes forming a geometric progression $1, q, q^2, \dots$ are worthy of study. For example,

$$\begin{aligned}
g_2(x;1,q) &= x^2-2qx+2q-1,\\
g_3(x;1,q,q^2) &= x^3-3q^2x^2+(6q^3-3q^2)x+-6q^3+6q^2-1,\\
g_4(x;1,q,q^2,q^3) &= x^4-4q^3x^3+(12q^5-6q^4)x^2+(-24q^6+24q^5-4q^3)x\\
&\quad +24q^6-36q^5+6q^4+8q^3-1.
\end{aligned}$$

These Gončarov polynomials can be regarded as q-analogues of Abel polynomials.

7 Expected sums of parking functions

In the remainder of this paper, we shall use methods from elementary probability theory. A subset $\mathcal{S}$ of the set $[1,x]^n$ of length-n sequences with terms in the discrete interval $[1,x]$ can be made into a discrete probability space with a uniform probability measure by assigning a probability of $1/|\mathcal{S}|$ to each sequence

in $\mathcal{S}$. When $\mathcal{S}$ is $[1,x]^n$, then each sequence has probability $1/x^n$. In this case, the probability measure can also be obtained by choosing each term x_i independently and randomly with uniform distribution from the discrete interval $[1,x]$.

Given a subset $\mathcal{S}$ of length-n sequences, we define the random variable S_n to be the sum $x_1+x_2+\ldots+x_n$ of a random sequence in $\mathcal{S}$. The *expected sum* of a random sequence from $\mathcal{S}$ is the expectation $E[S_n]$. Let $(x)_k$ be the *k-falling factorial*, that is,

$$(x)_k = x(x-1)\cdots(x-k+1).$$

The *kth (falling) factorial moment of the sum of a random sequence* in $\mathcal{S}$ is the expectation $E[(S_n)_k]$. More explicitly, $E[(S_n)_k]$ equals

$$\frac{1}{|\mathcal{S}|}\sum_{(x_1,x_2,\ldots,x_n)\in\mathcal{S}}(x_1+x_2+\ldots+x_n)_k.$$

In particular, let $E_k(n;\mathbf{u})$ be the *kth falling factorial moment of the sum of a random* $\mathbf{u}$*-parking function*, that is,

$$E_k(n;\mathbf{u}) = \frac{1}{P_n(\mathbf{u})}\sum_{(x_1,x_2,\ldots,x_n)}(x_1+x_2+\ldots+x_n)_k,$$

where the sum ranges over all $\mathbf{u}$-parking functions of length n.

The decomposition in Theorem 5.1 also gives recursions for expected values of moments of sums of parking functions. From these recursions, one can, with some difficulty, get explicit formulas for the moments. In this section, we shall show how this can be done for the first moment or the expected sum. We begin with the linear recursion.

(7.1) Theorem. The expected sums of $\mathbf{u}$-parking functions satisfy the following linear recursion:

$$\frac{n(x+1)}{2} = \sum_{m=0}^{n}\binom{n}{m}\frac{(x-u_{m+1})^{n-m}P_m(\mathbf{u})}{x^n}\left(E_1(m;\mathbf{u})+\frac{(n-m)(x+u_{m+1}+1)}{2}\right).$$

Proof. We derive the expected sum of a sequence $(x_1,x_2,\ldots,x_n)$ in $[1,x]^n$ in two different ways. Since the expected value of any term x_i is $(1+x)/2$, the expected sum of a random sequence in $[1,x]^n$ is $n(1+x)/2$, the left hand side of the recursion.

By Theorem 5.1, each sequence in $[1,x]^n$ decomposes into a $\mathbf{u}$-parking function of length m and a sequence in $[u_{m+1}+1,x]^{n-m}$. For a fixed m-element subset $\{i_1,i_2,\ldots,i_m\}$ of $\{1,2,\ldots,n\}$, consider the subset of sequences decomposing into a length-m $\mathbf{u}$-parking function indexed by $\{i_1,i_2,\ldots,i_m\}$ and a length-$(n-m)$ sequence in $[u_{m+1}+1,x]^{n-m}$ indexed by the complement. The probability that a random sequence is in this set is

$$\frac{(x-u_{m+1})^{n-m}P_m(u_1,u_2,\ldots,u_m)}{x^n}$$

and the expected sum of such a sequence is

$$E_1(m;u_1,u_2,\ldots,u_m)+\frac{(n-m)(x+u_{m+1}+1)}{2}.$$

The right hand side of the recursion can now be obtained by conditioning on the event that the maximal subsequence forming a $\mathbf{u}$-parking function is indexed by $\{i_1,i_2,\ldots,i_m\}$ and summing over subsets of the index set $\{1,2,\ldots,n\}$. This completes the proof of Theorem 7.1.

The recursion in Theorem 7.1 gives an Appell relation for the expected sums. Let $\mathbf{a}$ be the sequence defined by

$$a_i = x-u_{i+1},$$

with $0\le i<\infty$. Then the expected sum $E_1(n;u_1,u_2,\ldots,u_n)$, as a function of $u_1,u_2,\ldots,u_n$, becomes a function of x and $a_0,a_1,\ldots,a_{n-1}$. Let

$$e_n^{(1)}(x;a_0,a_1,\ldots,a_{n-1}) = E_1(n;x-a_0,\ldots,x-a_{n-1}).$$

In terms of $e_n^{(1)}(x;\mathbf{a})$, the recursion in Theorem 7.1 becomes

$$\begin{aligned}\frac{nx^n(x+1)}{2} &= \sum_{m=0}^{n}\binom{n}{m}a_m^{n-m}g_m(x;\mathbf{a})e_m^{(1)}(x;\mathbf{a}) \\ &+ \sum_{m=0}^{n}\binom{n}{m}a_m^{n-m}g_m(x;\mathbf{a})(n-m)\left(\frac{2x-a_m+1}{2}\right).\end{aligned}$$

From the recursion, we conclude that $g_n(x;\mathbf{a})e_n^{(1)}(x;\mathbf{a})$ is the sum of two homogeneous polynomials in x and $a_0, a_1, \ldots, a_{n-1}$, one having total degree $n+1$ and the other having total degree n. The sum $g_n(x;\mathbf{a})e_n^{(1)}(x;\mathbf{a})$ is easier to work with than the expected sum $e_n^{(1)}(x;\mathbf{a})$. We shall derive an Appell relation and an explicit formula for $g_n(x;\mathbf{a})e_n^{(1)}(x;\mathbf{a})$ in terms of Gončarov polynomials.

We begin with the Appell relation. Multiplying both sides by $t^n/n!$ and summing over n, we get

$$\frac{(1+x)xt}{2}e^{xt}$$

on the left hand side. For the first sum on the right hand side, we get

$$\sum_{n=0}^{\infty}\left[\sum_{m=0}^{n}\binom{n}{m}a_m^{n-m}g_m(x;\mathbf{a})e_m^{(1)}(x;\mathbf{a})\right]\frac{t^n}{n!} = \sum_{m=0}^{\infty}g_m(x;\mathbf{a})e_m^{(1)}(x;\mathbf{a})\frac{e^{a_m t}t^m}{m!}.$$

For the second sum, we get

$$\begin{aligned}&\sum_{n=0}^{\infty}\left[\sum_{m=0}^{n}\binom{n}{m}a_m^{n-m}g_m(x;\mathbf{a})(n-m)\left(\frac{2x-a_m+1}{2}\right)\right]\frac{t^n}{n!} \\ =\ & t\sum_{n=1}^{\infty}\left[\sum_{m=0}^{n-1}\binom{n-1}{m}a_m^{(n-1)-m}g_m(x;\mathbf{a})\left(a_m\left(x+\frac{1}{2}\right)-\frac{1}{2}a_m^2\right)\right]\frac{t^{n-1}}{(n-1)!} \\ =\ & t\sum_{m=0}^{\infty}g_m(x;\mathbf{a})\left[\left(x+\frac{1}{2}\right)a_m-\frac{1}{2}a_m^2\right]\frac{e^{a_m t}t^m}{m!}.\end{aligned}$$

Therefore we obtain the following Appell relation.

(7.2) Theorem.

$$\begin{aligned}&\sum_{n=0}^{\infty}g_n(x;\mathbf{a})e_n^{(1)}(x;\mathbf{a})\frac{e^{a_n t}t^n}{n!} \\ =\ & \frac{(1+x)xt}{2}e^{xt}-\sum_{n=0}^{\infty}g_n(x;\mathbf{a})\left[\left(x+\frac{1}{2}\right)a_n t-\frac{1}{2}a_n^2 t\right]\frac{e^{a_n t}t^n}{n!}.\end{aligned}$$

Our next objective is to derive an expression for $g_n(x;\mathbf{a})e_n^{(1)}(x;\mathbf{a})$ as a linear combination of Gončarov polynomials. This gives an formula to compute $e_n^{(1)}(x;\mathbf{a})$ assuming that the Gončarov polynomials are already computed. We remark that from a computer algebra point of view, the linear recursion in Theorem 7.1 is a very efficient way to calculate a specific expected sum, but "explicit" formulas are also useful.

We shall use the following vector notation introduced in Section 2. If $f_i(x), i = 0, 1, 2, \ldots, n$, is a sequence of polynomials, then

$$\overrightarrow{f_i(x)} = (f_0(x), f_1(x), \ldots, f_n(x))^T.$$

In particular, the linear recursions for Gončarov polynomials can be rewritten as

$$\mathcal{A}\ \overrightarrow{g_i(x;\mathbf{a})} = \overrightarrow{x^i},$$

where $\mathcal{A}$ is the matrix defined in Lemma 4.1. Similarly, we can rewrite the linear recursion in Theorem 7.1 as

$$\frac{x(1+x)}{2}\overrightarrow{ix^{i-1}} = \mathcal{A}\ \overrightarrow{g_i(x;\mathbf{a})e_i^{(1)}(x;\mathbf{a})} + \mathcal{B}\ \overrightarrow{\left(\frac{2x-a_i+1}{2}\right)g_i(x;\mathbf{a})}.$$

where $\mathcal{B}$ is the $(n+1)\times(n+1)$ matrix

$$\left[i\binom{i-1}{j}a_j^{i-j}\right]_{0\le i,j,\le n}.$$

Note that, as always, we use the convention that the binomial coefficient $\binom{i}{j}$ is zero if $j > i$. Applying the inverse of $\mathcal{A}$ to both sides, we obtain

$$\frac{x(1+x)}{2}\mathcal{A}^{-1}\ \overrightarrow{ix^{i-1}} = \overrightarrow{g_i(x;\mathbf{a})e_i^{(1)}(x;\mathbf{a})} + \mathcal{A}^{-1}\mathcal{B}\ \overrightarrow{\left(\frac{2x-a_i+1}{2}\right)g_i(x;\mathbf{a})}. \tag{7.1}$$

By Lemma 4.1, the inverse of $\mathcal{A}$ is the coefficient matrix of the Gončarov polynomials. Hence, observing that ix^{i-1} is the derivative of x^i,

$$\mathcal{A}^{-1}\overrightarrow{ix^{i-1}} = \overrightarrow{Dg_i(x;\mathbf{a})}.$$

Using the differential relation for Gončarov polynomials, we conclude that the left hand side of equation (7.2) equals

$$\frac{x(1+x)}{2}\overrightarrow{ig_{i-1}(x;a_1,a_2,\ldots,a_{i-1})},$$

where we use the convention (consistent with the differential relation) that Gončarov polynomials with negative indices are identically zero.

To simplify the right hand side, consider the matrix $\mathcal{A}^{-1}\mathcal{B}$. Since both $\mathcal{A}$ and $\mathcal{B}$ are lower triangular and the diagonal entries of $\mathcal{B}$ are zero, $\mathcal{A}^{-1}\mathcal{B}$ is lower triangular with zero diagonal. In particular, the ij-entry of $\mathcal{A}^{-1}\mathcal{B}$ is zero if $i \le j$. Suppose that $i > j$. Then by Lemma 4.1, the ij-th entry of $\mathcal{A}^{-1}\mathcal{B}$ equals

$$\begin{aligned}
&\sum_{k=0}^{n}\binom{i}{k}g_{i-k}(0;a_k,\ldots,a_{i-1})k\binom{k-1}{j}a_j^{k-j}\\
=\quad &(i-j)\binom{i}{j}a_j\sum_{t=0}^{n-j-1}\binom{i-j-1}{t}g_{i-j-1-t}(0;a_{j+1+t},\ldots,a_{i-1})a_j^t.
\end{aligned}$$

By equation (4.2),

$$g_{i-j}(x;a_j,\ldots,a_{i-1}) = \sum_{t=0}^{i-j}\binom{i-j}{t}g_t(0;a_{i-t},\ldots,a_{i-1})x^{i-j-t}.$$

Taking the derivative on both sides, we obtain

$$\begin{aligned}
Dg_{i-j}(x;a_j,\ldots,a_{i-1}) &= (i-j)\sum_{t=0}^{i-j-1}\binom{i-j-1}{t}g_t(0;a_{i-t},\ldots,a_{i-1})x^{i-j-t-1}\\
&= (i-j)\sum_{t=0}^{i-j-1}\binom{i-j-1}{t}g_{i-j-1-t}(0;a_{j+1+t},\ldots,a_{i-1})x^t.
\end{aligned}$$

We conclude that the ijth entry of $\mathcal{A}^{-1}\mathcal{B}$ equals

$$\binom{i}{j} a_j Dg_{i-j}(a_j; a_j, a_{j+1}, \ldots, a_{i-1}).$$

By the differential relation,

$$Dg_{i-j}(x; a_j, a_{j+1}, \ldots, a_{i-1}) = (i-j) g_{i-j-1}(x; a_{j+1}, a_{j+2}, \ldots, a_{i-1})$$

Hence, an alternate way to write the ijth entry of $\mathcal{A}^{-1}\mathcal{B}$ is

$$i\binom{i-1}{j} a_j g_{i-j-1}(a_j; a_{j+1}, a_{j+2}, \ldots, a_{i-1}).$$

Putting all the above into equation (7.1), we obtain the following theorem.

(7.3) Theorem. The sum $g_n(x; \mathbf{a}) e_n^{(1)}(x; \mathbf{a})$ equals

$$\begin{aligned} & \frac{nx(1+x)}{2} g_{n-1}(x; a_1, a_2, \ldots, a_{n-1}) \\ - \quad & \frac{n}{2} \sum_{i=0}^{n-1} \binom{n-1}{i} a_i (2x - a_i + 1) g_{n-i-1}(a_i; a_{i+1}, a_{i+2}, \ldots, a_{n-1}) g_i(x; a_0, a_1, \ldots, a_{i-1}). \end{aligned}$$

Setting $x = 0$ and $a_i = -u_{i+1}$ and using Theorem 5.4 and the shift formula, we obtain a formula for the expected sum in terms of the sequence $\mathbf{u}$.

(7.4) Theorem. The expected sum $E_1(n; \mathbf{u})$ equals

$$\frac{n}{2} \sum_{j=1}^{n} \binom{n-1}{j-1} u_j (u_j + 1) \frac{P_{n-j}(u_{j+1} - u_j, u_{j+2} - u_j, \ldots, u_n - u_j) P_{j-1}(u_1, u_2, \ldots, u_{j-1})}{P_n(u_1, u_2, \ldots, u_n)}.$$

This formula, a sum of positive terms, should have an revealing combinatorial interpretation.

8 The classical case with Abel identities

In this section, we shall give several equivalent formulas for the expected sum $E_1(n; a, a+b, \ldots, a+(n-1)b)$. We shall often abbreviate our notation and write $E_1(n; a, b)$ instead of $E_1(n; a, a+b, \ldots, a+(n-1)b)$. Using Theorem 7.4 and Corollary 5.5, we obtain the following formula for the expected sum.

(8.1) Theorem.

$$E_1(n; a, b) = \frac{n}{2} \sum_{i=0}^{n-1} \binom{n-1}{i} (a + ib + 1) b^{n-i-1} (n-i)^{n-i-2} \frac{(a+ib)^i}{(a+nb)^{n-1}}.$$

In the remainder of this section, we shall use Theorem 7.3 to obtain other formulas for expected sums. There are many – almost too many – such formulas, due mainly to the existence of Abel identities discussed in Section 3. We shall use the following substitutions

$$x = a,\ a_0 = 0,\ a_1 = -b,\ a_2 = -2b,\ \ldots,\ a_n = -(n-1)b.$$

We begin by calculating explicitly several values of Gončarov polynomials. By equation (3.1),

$$\begin{aligned} g_n(a;0,-b,-2b,\ldots,-(n-1)b) &= P_n(a,a+b,a+2b,\ldots,a+(n-1)b) \\ &= a(a+nb)^{n-1}, \end{aligned}$$

$$g_{n-1}(a;-b,-2b,\ldots,-(n-1)b) = (a+b)(a+nb)^{n-2},$$

$$\begin{aligned} g_{n-i-1}(-ib;-(i+1)b,\ldots,-(n-1)b) &= b[(n-i)b]^{n-i-2} \\ &= b^{n-i-1}(n-i)^{n-i-2}. \end{aligned}$$

Substituting these values into the formula in Theorem 7.3, we obtain

$$\begin{aligned} & a(a+bn)^{n-1}e_n^{(1)}(a;0,-b,-2b,\ldots,-(n-1)b) \\ = \quad & \frac{na(a+1)}{2}(a+b)(a+bn)^{n-2} \\ & +n\sum_{i=0}^{n-1}\binom{n-1}{i}ib^{n-i}(n-i)^{n-i-2}\left(\frac{2a+ib+1}{2}\right)a(a+ib)^{i-1}. \end{aligned}$$

The sum in this expression can be simplified slightly (by manipulating binomial coefficients) to

$$n\sum_{i=1}^{n-1}\binom{n-1}{i-1}b^{n-i}(n-i)^{n-i-1}\left(\frac{2a+ib+1}{2}\right)a(a+ib)^{i-1}.$$

We break up this sum into two parts by writing

$$\frac{2a+ib+1}{2} = \frac{a+1}{2}+\frac{a+ib}{2}.$$

The first part is the following sum

$$\frac{nab(a+1)}{2}\sum_{i=1}^{n-1}\binom{n-1}{i-1}b^{n-i-1}(n-i)^{n-i-1}(a+ib)^{i-1}.$$

By equation (3.3), the sum equals $(n-1)(a+nb)^{n-2}$. Regrouping terms, we conclude that the first part equals

$$\binom{n}{2}ab(a+1)(a+nb)^{n-2}.$$

When this quantity is added to $na(a+1)(a+b)(a+nb)^{n-2}/2$, we get the refreshingly simple result $na(a+1)(a+nb)^{n-1}/2$. The second part,

$$\frac{na}{2}\sum_{i=1}^{n-1}\binom{n-1}{i-1}b^{n-i}(n-i)^{n-i-1}(a+ib)^{i}, \tag{8.1}$$

does not simplify into a single term. Hence, the following theorem gives a reasonable formula for the expected sum.

(8.2) Theorem. The expected sum $E_1(n;a,b)$ equals

$$\frac{n(a+1)}{2}+\frac{n}{2}\sum_{i=1}^{n-1}\binom{n-1}{i-1}b^{n-i}(n-i)^{n-i-1}\frac{(a+ib)^i}{(a+nb)^{n-1}}.$$

For ordinary parking function, this formula specializes to

$$E_1(n;1,1) = n + \frac{n}{2}\sum_{i=1}^{n-1}\binom{n-1}{i-1}(n-i)^{n-i-1}\frac{(i+1)^i}{(n+1)^{n-1}}. \tag{8.2}$$

This formula does not have the same form (and is not obviously the same) as the formula obtained by Gessel and Sagan [5] or Knuth [10] for ordinary parking functions, which states

$$E_1(n;1,1) = \binom{n+1}{2} - \frac{1}{2}\sum_{i=2}^{n}\binom{n}{i}i!(n+1)^{1-i}. \tag{8.3}$$

This formula suggests a third formula for $E_1(n;a,b)$ which specializes to equation (8.3) when both a and b are set equal to 1.

(8.3) Theorem.

$$E_1(n;a,b) = \frac{n(a+1)}{2} + b\binom{n}{2} - \frac{1}{2}\sum_{i=2}^{n}\binom{n}{i}\frac{i!b^i}{(a+nb)^{i-1}}.$$

There are two ways to prove Theorem 8.3. The first way to show that the expressions in Theorems 8.2 and 8.3 are equal. This can be done using a computer algebra program (see [18]) or by traditional methods. We will leave the computer algebra method to our silicon-based friends. The traditional method requires using two Abel identities. It is not particularly illuminating *per se*, but an intermediate form turns out to be useful later. Thus, it is worthwhile to show this method in some detail. The main step is to transform the sum (8.1) into a suitable form.

Using equation (3.4), the sum (8.1) equals

$$\frac{ab^2n(n-1)}{2}\sum_{j=0}^{n-2}\frac{(n-2)!}{j!}(a+nb)^j b^{n-j-3}(a+(n-j-1)b).$$

Changing indices from j to $n-j$ and regrouping terms, this becomes

$$\frac{a}{2}\sum_{j=2}^{n}\frac{n!}{(n-j)!}(a+nb)^{n-j}b^{j-1}(a+(j-1)b).$$

Hence, we have another formula for the expected sum. This formula will be used in Section 10.

(8.4) Theorem. The expected sum $E_1(n;a,a+b,\ldots,a+(n-1)b)$ equals

$$\frac{n(a+1)}{2} + \frac{1}{2}\sum_{j=2}^{n}\frac{n!}{(n-j)!}\frac{b^{j-1}(a+(j-1)b)}{(a+nb)^{j-1}}.$$

This is not yet Theorem 8.3 and we need identity (3.5). Extracting the first two terms in the sum, moving them to the left, simplifying the left hand side, finding that there is a factor of b on the left hand side, and dividing by b, we obtain :

$$n(n-1)b(a+nb)^{n-1} = \sum_{j=2}^{n}\frac{n!}{(n-j)!}(a+nb)^{n-j}b^{j-1}(a+jb).$$

Applying the last identity, we reach the required form for the sum (8.1). Dividing by $a(a+nb)^{n-1}$, we arrive finally at the equation in Theorem 8.3.

9 The classical case with an inverse relation

The second way to prove Theorem 8.3 is to proceed directly from the linear recursion. This method yields an interesting and simpler special case of the linear recursion. Consider the linear recursion in Theorem 7.1 with $u_{i+1} = a + ib$. Multiplying both sides by x^n, we obtain

$$\begin{aligned} & \frac{nx^n(1+x)}{2} \\ = & \sum_{i=0}^{n} \binom{n}{i} (x-a-ib)^{n-i} a(a+ib)^{i-1} \left(E_1(i;a,b) + \frac{(n-i)(x+a+ib+1)}{2} \right). \end{aligned}$$

This identity holds for all integers x greater than or equal to $a + (n-1)b$. Hence, it is a polynomial identity in x and holds for all real numbers x. Setting $x = 0$ and rearranging terms, we have

$$\begin{aligned} & \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} a(a+ib)^{n-1} E_1(i;a,b) \\ = & -\frac{1}{2} \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} (n-i) \left[a(a+ib)^{n-1} + a(a+ib)^n \right]. \end{aligned} \tag{9.1}$$

When $n = 0$, equation (9.1) says that $E_1(0;a,b) = 0$, as expected. When $n \geq 1$, the right hand side of equation (9.1) can be simplified slightly to

$$\frac{n}{2} \sum_{i=0}^{n-1} (-1)^{n-1-i} \binom{n-1}{i} \left[a(a+ib)^{n-1} + a(a+ib)^n \right].$$

In this form, it can be written as a single term by using the following lemma.

(9.1) Lemma.

$$\sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} a(a+ib)^m = ab^n n! \sum_{r=0}^{m-n} \binom{m}{n+r} a^{m-n-r} b^r S(n+r,n).$$

where $S(m,n)$ is a Stirling number of the second kind and equals the number of partitions of an m-element set into n non-empty blocks.

Proof. Expand $(a+bi)^m$ with the binomial theorem, use the identity of Stirling ([29]; see, for example, [25], page 34):

$$\sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} i^m = n! S(m,n),$$

and observe that $S(m,n) = 0$ if $m < n$.

The sum on the right hand side in Lemma 9.1 is empty if $m \leq n-1$. Hence, if $m \leq n-1$,

$$\sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} a(a+ib)^m = 0.$$

Two other useful cases of Lemma 9.1 are

$$\sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} a(a+ib)^n = n! ab^n$$

and

$$\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}a(a+ib)^{n+1} = n!ab^n\left[b\binom{n+1}{2}+a(n+1)\right].$$

Using Lemma 9.1 (for the case $n-1$), the right hand side of equation (9.1) can be written as a single term and we obtain the following simpler linear recursion for the expected sum when $n \geq 1$:

$$\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}a(a+ib)^{n-1}E_1(i;a,b) = \frac{ab^{n-1}n!}{2}\left[b\binom{n}{2}+an+1\right]. \tag{9.2}$$

With a reasonable linear recursion in hand, we have two ways of proving Theorem 8.3. The first and somewhat unsatisfactory way is to check that the formula for $E_1(n;a,b)$ given in Theorem 8.3 yields $E_1(0) = 0$ and satisfies the linear recursion (9.2). The checking can be done easily by hand (using Lemma 9.1) or by a computer algebra program. Either way, we obtain Theorem 8.3.

The second is to "discover" the solution in a systematic way. This method will be necessary for finding formulas for the higher moments for which we have no reasonable guess. See [14]. We begin by transforming the recursion into a matrix equation.

Let $\mathcal{P}$ be the $(N+1)\times(N+1)$ lower triangular matrix

$$\left[(-1)^{n-i}\binom{n}{i}a(a+ib)^{n-1}\right]_{0\leq n,i\leq N}.$$

For example, when $N=3$, $\mathcal{P}$ is the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -a & a & 0 & 0 \\ a^2 & -2a(a+b) & a(a+2b) & 0 \\ -a^3 & 3a(a+b)^2 & -3a(a+2b)^2 & a(a+3b)^2 \end{bmatrix}$$

Using the vector notation introduced in Section 2, we can rewrite equation (9.2) as

$$\mathcal{P}\overrightarrow{E_1(i;a,b)} = \overrightarrow{\frac{i!ab^{i-1}}{2}\left[b\binom{i}{2}+ai+1\right]}. \tag{9.3}$$

Our next step is to find the inverse of $\mathcal{P}$. Let $\mathcal{Q}$ be the $(N+1)\times(N+1)$ lower triangular matrix

$$\left[\binom{i}{j}\frac{j!b^j}{(a+ib)^{j-1}}\right]_{0\leq i,j\leq N}.$$

For example, when $N=3$, $\mathcal{Q}$ is the matrix

$$\begin{bmatrix} a & 0 & 0 & 0 \\ a+b & b & 0 & 0 \\ a+2b & 2b & \dfrac{2b^2}{a+2b} & 0 \\ a+3b & 3b & \dfrac{6b^2}{a+3b} & \dfrac{6b^3}{(a+3b)^2} \end{bmatrix}$$

(9.2) Lemma.

$$\mathcal{P}\mathcal{Q} = \mathcal{N}\mathcal{L}$$

where $\mathcal{N}$ is the diagonal matrix whose iith entry is $ab^i i!$ and $\mathcal{L}$ is the lower triangular matrix with all ijth entries equal to 1 whenever $i \geq j$ and 0 otherwise.

Proof. The njth entry of the product $\mathcal{P}\mathcal{Q}$ equals

$$\sum_{i=j}^{n}(-1)^{n-i}\binom{n}{i}\binom{i}{j}j!b^j a(a+ib)^{n-j}.$$

Changing indices from i to $i-j$ and regrouping terms, this can be simplified to

$$\frac{n!b^j}{(n-j)!}\sum_{i=0}^{n-j}(-1)^{n-j-i}\binom{n-j}{i}a((a+jb)+ib)^{n-j}.$$

By the case $n-j$ of Lemma 9.1, the sum equals $(n-j)!ab^{n-j}$ and the lemma follows.

Lemma 9.2 can be rephrased as follows.

(9.3) Lemma.

$$\mathcal{P}^{-1} = \mathcal{Q}\mathcal{L}^{-1}\mathcal{N}^{-1}.$$

The inverse matrices $\mathcal{N}^{-1}$ and $\mathcal{L}^{-1}$ have simple interpretations when acting on a column vector $\overrightarrow{a_i}$. Multiplying on the left by $\mathcal{N}^{-1}$ divides the ith coordinate a_i by $ab^i i!$. The matrix $\mathcal{L}$ is the summation matrix and sends the vector $\overrightarrow{a_i}$ to the vector whose ith coordinate is $a_0 + a_1 + \ldots + a_i$. Hence, the inverse $\mathcal{L}^{-1}$ is the backward difference matrix, with all diagonal entries 1, all subdiagonal entries -1, and all other entries zero. Hence, multiplying the vector $\overrightarrow{a_i}$ on the left by $\mathcal{L}^{-1}$ results in the vector $\overrightarrow{a_i - a_{i-1}}$, obtained by taking the backward difference of the coordinates a_i, with the convention that $a_{-1} = 0$.

Hence, when we apply Lemma 9.3 to the matrix equation (9.3), we obtain

$$\overrightarrow{E_1(i;a,b)} = \frac{1}{2b}Q(0, a+1, a+b, a+2b, \ldots, a+(i-1)b, \ldots, a+(N-1)b)^T.$$

Writing out the nth coordinate explicitly, we obtain the formula in Theorem 8.4.

To obtain Theorem 8.3, we need to "precondition" and consider the *adjusted* sum S_n^*, defined by

$$S_n^* = \frac{n(a+1)}{2} + b\binom{n}{2} - S_n.$$

For ordinary parking functions, the adjusted sum is the reversed sum (defined in Section 6), but this is not true in general.

Substituting S_n^* into equation (9.1) and simplifying using Lemma 9.1, we obtain

$$\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}a(a+ib)^{n-1}E[S_i^*] = -\frac{n!ab^n(n-1)}{2}.$$

This converts to the matrix equation

$$\mathcal{P}\overrightarrow{E[S_i^*]} = -\frac{1}{2}(0, 0, 2ab^2, 12ab^3, \ldots, i!ab^i(i-1), \ldots, N!ab^N(N-1))^T.$$

Inverting this, we obtain

$$\overrightarrow{E[S_i^*]} = -\frac{1}{2}Q(0, 0, 1, 1, \ldots, 1)^T,$$

from which the formula in Theorem 8.3 follows immediately.

10 Order properties of expected sums

In this section, we shall consider the following conjecture.

(10.1) Conjecture. The expected sum $E_1(n; u_1, u_2, \ldots, u_n)$ is an increasing function of n and the gaps $u_{i+1} - u_i$.

This conjecture may seem obvious. However, there are two factors to consider when n and the gaps are increased. On the positive side, the parking functions are allowed to take on higher values. On the negative side, there are more parking functions, and since parking functions cannot take on too many higher values, the sample might consist mostly of parking functions with smaller sums. Our intuition is that the positive factor always predominates.

We begin with a simple general result supporting this conjecture.

(10.2) Proposition. If γ is a rational number greater than 1, then

$$E_1(n; u_1, u_2, \ldots, u_n) < E_1(n; \gamma u_1, \gamma u_2, \ldots, \gamma u_n).$$

Proof. Writing $u_j(u_j + 1)$ as $u_j^2 + u_j$ in the formula in Theorem 7.5, we can write $E_1(n; \mathbf{u})P_n(\mathbf{u})$ as the sum $F(\mathbf{u}) + G(\mathbf{u})$ of two homogeneous functions in the variables $u_1, u_2, \ldots, u_n$, where $F(\mathbf{u})$ has total degree $n + 1$ and $G(\mathbf{u})$ has total degree n. Using Corollary 5.6, we have

$$\begin{aligned} E_1(n; \gamma u_1, \gamma u_2, \ldots, \gamma u_n) &= \frac{\gamma^{n+1} F(\mathbf{u}) + \gamma^n G(\mathbf{u})}{\gamma^n P_n(\mathbf{u})} \\ &= \frac{(\gamma - 1)F(\mathbf{u})}{P_n(\mathbf{u})} + \frac{F(\mathbf{u}) + G(\mathbf{u})}{P_n(\mathbf{u})} \\ &= \frac{(\gamma - 1)F(\mathbf{u})}{P_n(\mathbf{u})} + E_1(n; u_1, u_2, \ldots, u_n). \end{aligned}$$

Since $\gamma > 1$, the proposition follows.

For the classical case, when $u_i = a + (i - 1)b$, Conjecture 10.1 states that the expected sums $E_1(n; a, b)$ are increasing functions of n, a, and b. We shall verify this special case.

(10.3) Lemma. If $0 < a < c$,

$$E_1(n; a, b) < E_1(n; c, b).$$

Proof. Use the formula in Theorem 8.3 and observe that $-(c + nb)^{-1} > -(a + nb)^{-1}$.

Hence, $E_1(n; a, b)$ is an increasing function of a (for fixed n and b).

(10.4) Lemma.

$$E_1(n; a, b) < E_1(n + 1; a, b).$$

Proof. Rewrite the formula in Theorem 8.3 in the form

$$E_1(n; a, b) = \frac{n(a + 1)}{2} + b\binom{n}{2} - \frac{b}{2} \sum_{i=2}^{n} \frac{(n)_i}{(\gamma + n)^{i-1}},$$

where $(n)_i$ is a falling factorial and $\gamma = a/b$. Then, the forward difference

$$E_1(n + 1; a, b) - E_1(n; a, b)$$

equals

$$\frac{a + 1}{2} + bn - \frac{b}{2} \sum_{i=2}^{n} \left[\frac{(n + 1)_i}{(n + 1 + \gamma)^{i-1}} - \frac{(n)_i}{(n + \gamma)^{i-1}} \right] - \frac{b}{2} \frac{(n + 1)!}{(n + 1 + \gamma)^n}.$$

By an elementary induction argument, one can show that for γ a positive real number and $i = 2, 3, \ldots, n$,

$$\frac{(n+1)_i}{(n+1+\gamma)^{i-1}} - \frac{(n)_i}{(n+\gamma)^{i-1}} < 1. \tag{10.1}$$

The induction argument runs as follows. If $i = 2$,

$$\frac{(n+1)n}{n+1+\gamma} - \frac{n(n-1)}{n+\gamma} = \frac{n^2+2\gamma n+n}{n^2+2\gamma n+n+\gamma^2+\gamma} < 1.$$

Now assume that the inequality is true for $i \le k$. When $i = k+1$,

$$\begin{aligned}
& \frac{(n+1)_{k+1}}{(n+1+\gamma)^k} - \frac{(n)_{k+1}}{(n+\gamma)^k} \\
= & \left[\frac{(n+1)_k}{(n+1+\gamma)^{k-1}} - \frac{(n)_k}{(n+\gamma)^{k-1}}\right]\frac{n-k+1}{n+\gamma+1} + \frac{(n)_k}{(n+\gamma)^{k-1}}\left[\frac{n-k+1}{n+\gamma+1} - \frac{n-k}{n+\gamma}\right] \\
< & \frac{n-k+1}{n+\gamma+1} + \frac{(n)_k}{(n+\gamma)^{k-1}}\left[\frac{(\gamma+k)}{(n+\gamma)(n+\gamma+1)}\right] \\
= & 1 - \frac{\gamma+k}{n+\gamma+1} + \frac{(n)_k}{(n+\gamma)^k}\cdot\frac{\gamma+k}{n+\gamma+1} \\
< & 1.
\end{aligned}$$

From inequality (10.1), we conclude that

$$\begin{aligned}
E_1(n+1;a,b) - E_1(n;a,b) &> bn + \frac{a+1}{2} - \frac{b(n-1)}{2} - \frac{b}{2} \\
&= \frac{bn+a+1}{2} \\
&> 0.
\end{aligned}$$

(10.5) Lemma. If c is an integer strictly greater than b, then

$$E_1(n;a,b) < E_1(n;a,c).$$

Proof. Rewrite the formula for the expected sum $E_1(n;a,a+b,\ldots,a+(n-1)b)$ given in Theorem 8.4 in the form

$$\frac{n(a+1)}{2} + \frac{1}{2}\sum_{j=2}^{n}\frac{n!}{(n-j)!}\left[\frac{ab^{j-1}}{(a+nb)^{j-1}} + \frac{(j-1)b^j}{(a+nb)^{j-1}}\right].$$

From this, the lemma follows from the easy inequality: if $a > 0$ and $c > b$, then

$$\frac{b}{a+nb} < \frac{c}{a+nc}.$$

The three lemmas imply the following theorem.

(10.6) Theorem. The expected sum $E_1(n;a,a+b,\ldots,a+(n-1)b)$ is a strictly increasing function of n, a and b.

11 Factorial moments of parking functions

The decomposition in Section 5 can also be used to obtain linear recursions for higher factorial moments of sums of random parking functions. Let $\mathbf{u}$ be a sequence of non-decreasing positive integers. Let $\mathbf{a}$ be the

sequence defined by $a_j = x - u_{j+1}$ and let

$$e_i^{(k)}(x; a_0, \ldots, a_{n-1}) = E[(S_i)_k],$$

the k-factorial moment of the sum of a random $\mathbf{u}$-parking function as a function of $x, a_0, a_1, \ldots, a_{i-1}$. The *factorial moment generating function* $\mathcal{S}_i(t; \mathbf{a})$ for $\mathbf{u}$-parking functions of length i is defined by the following formula:

$$\mathcal{S}_i(t; \mathbf{a}) = \sum_{k=0}^{\infty} e_i^{(k)}(x; \mathbf{a}) \frac{t^k}{k!}.$$

Given a discrete interval $[\alpha, \beta]$, let $U_i(\alpha, \beta)$ be the sum of a random (integer) sequence chosen with uniform distribution from the space $[\alpha, \beta]^i$ of all length-i sequences with terms in $[\alpha, \beta]$. Then $U_i(\alpha, \beta)$ can also be thought of as a length-i random sequence obtained by choosing each term independently with uniform distribution from $[\alpha, \beta]$. The factorial moments of $U_i(\alpha, \beta)$ are known and they can be expressed in a compact form by exponential generating functions (see, for example, [8]). Let

$$\mathcal{U}_i(t; \alpha, \beta) = \sum_{k=0}^{\infty} E[(U_i(\alpha, \beta))_k] \frac{t^k}{k!}.$$

Then

$$\mathcal{U}_i(t; \alpha, \beta) = \left(\frac{(1+t)^{\beta+1} - (1+t)^{\alpha}}{(\beta - \alpha + 1)t} \right)^i.$$

(11.1) Theorem. Let k be a positive integer. Then the factorial moments of the sum of a random $\mathbf{u}$-parking function of length n satisfies the following linear recursion:

$$E[(U_n(1, x))_k] = \sum_{m=0}^{n} \binom{n}{m} \frac{a_m^{n-m} g_m(x; \mathbf{a})}{x^n} \left(\sum_{j=0}^{k} \binom{k}{j} e_m^{(j)}(x; \mathbf{a}) E[(U_{n-m}(u_{m+1} + 1, x))_{k-j}] \right).$$

Proof. The proof is almost the same as the proof of Theorem 7.1. Consider the event that the maximum subsequence forming a $\mathbf{u}$-parking function is indexed by $\{i_1, i_2, \ldots, i_m\}$. Because the length-$m$ $\mathbf{u}$-parking function and the length-$(n-m)$ sequence from $[u_{m+1} + 1, x]^{n-m}$ are chosen independently and an analogue of the binomial theorem holds for falling factorials, the expected value of $(U_n(1, x))_k$ conditioned on this event is

$$\sum_{j=0}^{k} \binom{k}{j} e_m^{(j)}(x; \mathbf{a}) E[(U_{n-m}(u_{m+1} + 1, x))_{k-j}].$$

Summing over the conditional expectations, we obtain the linear recursion.

As with the expected sum, it follows from the linear recursion that $g_i(x; \mathbf{a}) e_i^{(k)}(x; \mathbf{a})$ is a sum of $k+1$ homogeneous polynomial in the variables $x, a_0, a_1, \ldots, a_{i-1}$ having total degree i, $i+1$, $\ldots$, and $i+k$.

When Theorem 11.1 is restated in terms of exponential generating functions, we obtain the following linear recursion for the factorial moment generating functions $\mathcal{S}_i(t; \mathbf{a})$:

$$x^n \mathcal{U}_n(t; 1, x) = \sum_{i=0}^{n} \binom{n}{i} a_i^{n-i} g_i(x; \mathbf{a}) \mathcal{S}_i(t; \mathbf{a}) \mathcal{U}_{n-i}(t; u_{i+1} + 1, x), \tag{11.1}$$

We can use the matrix method in Section 7 to rewrite equation (11.1) in the following more compact form:

$$\mathcal{M} \overrightarrow{g_i(x; \mathbf{a}) \mathcal{S}_i(t; \mathbf{a})} = \overrightarrow{x^i \mathcal{U}_i(t; 1, x)},$$

where $\mathcal{M}$ is the lower triangular matrix with ijth entry equal to

$$\binom{i}{j} a_j^{i-j} \mathcal{U}_{i-j}(t; u_{j+1} + 1, x)$$

if $i \geq j$ and zero if $i < j$. From this linear equation, one can obtain by Cramer's rule a rather complicated determinantal formula for $\mathcal{S}_i(t;\mathbf{a})$. This determinantal formula seems to have no simple form.

After this, it is a pleasant surprise that there is a simple Appell relation for $\mathcal{S}_i(t;\mathbf{a})$. First observe that

$$\begin{aligned}\sum_{n=0}^{\infty} x^n \mathcal{U}_n(t;1,x)\frac{q^n}{n!} &= \sum_{n=0}^{\infty}\left[\frac{(1+t)^{x+1}-(1+t)}{t}\right]^n \frac{q^n}{n!} \\ &= \exp(\frac{q}{t}((1+t)^{x+1}-(1+t)))\end{aligned}$$

and

$$\sum_{n=i}^{\infty} a_i^{n-i}\mathcal{U}_{n-i}(t;u_{i+1}+1,x)\frac{q^{n-i}}{(n-i)!} = \exp(\frac{q}{t}((1+t)^{x+1}-(1+t)^{1+x-a_i})).$$

Hence, multiplying equation (11.1) by $q^n/n!$, summing over all non-negative integers n, and dividing both sides by $\exp(q(i+t)^{x+1}/t)$, we obtain

$$\exp(-\frac{q}{t}(1+t)) = \sum_{i=0}^{\infty} g_i \mathcal{S}_i(t)\exp(-\frac{q}{t}(1+t)^{1+x-a_i})\frac{q^i}{i!}.$$

Changing variables from q to qt, we obtain the following Appell relation.

(11.3) Theorem.

$$\exp(-q(1+t)) = \sum_{i=0}^{\infty} g_i(x;\mathbf{a})\mathcal{S}_i(t;\mathbf{a})\exp(-q(1+t)^{1+x-a_i})\frac{t^i q^i}{i!}.$$

The left hand side of the Appell relation does not depend on x (which is not surprising, as the linear recursion from which it is derived holds for all sufficiently large integer x). Hence, simpler Appell relations can be obtained by setting x to be 0 or any convenient constant or variable.

12 Second factorial moments of sums of parking functions

The second power moment is of particular importance in estimating the spread of a distribution. In this section, we shall derive an explicit formula for the second factorial moment of the sum of a random **u**-parking function with the matrix method used in Section 7.

We start with the linear recursion for the second factorial moment.

(12.1) Lemma. The second factorial moment $e_n^{(2)}(x;\mathbf{a})$ satisfies the linear recursion

$$\begin{aligned}& x^n E[(U_n(1,x))_2] \\ = & \sum_{m=0}^{n}\binom{n}{m}a_m^{n-m}g_m(x;\mathbf{a}) \\ & \cdot(e_m^{(2)}(x;\mathbf{a}) + 2e_m^{(1)}(x;\mathbf{a})E[U_{n-m}(x-a_m+1,x)]) + E[(U_{n-m}(x-a_m+1,x))_2]\end{aligned}$$

with

$$E[U_{n-m}(x-a_m+1,x)] = \frac{(n-m)(2x-a_m+1)}{2},$$

$$E[(U_{n-m}(x-a_m+1,x))_2] = \frac{(2x-a_m+1)^2(n-m)^2}{4} + (n-m)\left[\frac{a_m^2}{12} - \frac{2x-a_m}{2} - \frac{7}{12}\right],$$

and

$$E[(U_n(1,x))_2] = \frac{n(n-1)(x+1)^2}{4} + \frac{n(x^2-1)}{3}.$$

Next, we rewrite the linear recursion as the following system of linear equations:

$$\begin{aligned}
& \frac{1}{4}(x+1)^2\overrightarrow{i(i-1)x^i} + \frac{1}{3}(x^2-1)\overrightarrow{ix^i} \\
= \; & \overrightarrow{\mathcal{A}g_i(x;\mathbf{a})e_i^{(2)}(x;\mathbf{a})} \\
& \overrightarrow{+\mathcal{B}g_i(x;\mathbf{a})e_i^{(1)}(x;\mathbf{a})(2x-a_i+1) + g_i(x;\mathbf{a})(x^2 - xa_i + \frac{1}{3}(a_i^2-1))} \\
& + \; \overrightarrow{\mathcal{C}\left(\frac{2x-a_i+1}{2}\right)^2 g_i(x;\mathbf{a})},
\end{aligned}$$

where $\mathcal{A}$, $\mathcal{B}$ are the lower triangular matrices described in Section 7 and $\mathcal{C}$ is the lower triangular matrix with ijth entry equal to

$$i(i-1)\binom{i-2}{j}a_j^{i-j}$$

if $i > j+1$ and zero otherwise.

As in Section 7, we apply $\mathcal{A}^{-1}$ to both sides of the linear equation. Using Lemma 4.1, the left hand side simplifies to

$$\frac{x^2(x+1)^2}{4}\overrightarrow{D^2 g_i(x;\mathbf{a})} + \frac{x(x^2-1)}{3}\overrightarrow{Dg_i(x;\mathbf{a})}.$$

To simplify the right hand side, we need the entries of $\mathcal{A}^{-1}\mathcal{B}$ and $\mathcal{A}^{-1}\mathcal{C}$. The entries of $\mathcal{A}^{-1}\mathcal{B}$ were calculated in Section 7. The entries of $\mathcal{A}^{-1}\mathcal{C}$ can be calculated using a similar method. Indeed, the ijth entry of $\mathcal{A}^{-1}\mathcal{C}$ is

$$\binom{i}{j}a_j^2 D^2 g_{i-j}(a_j; a_j, a_{j+1}, \ldots, a_{i-1})$$

if $i \geq j$ and zero otherwise. Using these facts and the differential relation for Gončarov polynomials, we obtain the equation:

$$\begin{aligned}
& g_n(x;a_0,a_1,\ldots,a_{n-1})e_n^{(2)}(x;a_0,a_1,\ldots,a_{n-1}) \\
= \; & \frac{x^2(x+1)^2}{4}n(n-1)g_{n-2}(x;a_2,a_3,\ldots,a_{n-1}) + \frac{x(x^2-1)}{3}ng_{n-1}(x;a_1,a_2,\ldots,a_{n-1}) \\
& -\frac{n(n-1)}{4}\sum_{i=0}^{n-2}\binom{n-2}{i}a_i^2(2x-a_i+1)^2 \\
& \cdot g_{n-i-2}(a_i;a_{i+2},a_{i+3},\ldots,a_{n-1})g_i(x;a_0,a_1,\ldots,a_{i-1}) \\
& -n\sum_{i=0}^{n-1}\binom{n-1}{i}a_i g_{n-i-1}(a_i;a_{i+1},a_{i+2},\ldots,a_{n-1}) \\
& \quad \cdot\Big[(2x-a_i+1)g_i(x;a_0,a_1,\ldots,a_{i-1})e_i^{(1)}(x;a_0,a_1,\ldots,a_{i-1}) \\
& \quad + \left(x^2 - xa_i + \frac{a_i^2-1}{3}\right)g_i(x;a_0,a_1,\ldots,a_{i-1})\Big].
\end{aligned}$$

Setting $x=0$ and $a_i = -u_{i+1}$, we obtain the following theorem.

(12.2) Theorem

$$\begin{aligned}
& P(u_1,u_2,\ldots,u_n)E_2(n;u_1,u_2,\ldots,u_n)\\
= \;& n\sum_{i=0}^{n-1}\binom{n-1}{i}u_{i+1}P_{n-i-1}(u_{i+2}-u_{i+1},u_{i+3}-u_{i+1},\ldots,u_n-u_{i+1})\\
& \quad\cdot\left[(u_{i+1}+1)P_i(u_1,\ldots,u_i)E_1(i;u_1,\ldots,u_i)+\left(\frac{u_{i+1}^2-1}{3}\right)P_i(u_1,\ldots,u_i)\right]\\
& -\frac{n(n-1)}{4}\sum_{i=0}^{n-2}\binom{n-2}{i}u_{i+1}^2(u_{i+1}+1)^2\\
& \quad\cdot P_{n-i-2}(u_{i+3}-u_{i+1},u_{i+4}-u_{i+1},\ldots,u_n-u_{i+1})P_i(u_1,\ldots,u_i).
\end{aligned}$$

On comparing the formulas in Theorems 7.4 and 12.2, it is evident that one can obtain, in a mechanical way, formulas for any higher moments and that these formulas has a recognizable pattern.

For many applications, asymptotic formulas are much more useful than explicit formulas. The only known results are asymptotic formulas for the expected sum and second moment of random ordinary parking functions. These formulas can be extracted from [20, 24, 30] (see also [4]). Briefly,

$$\begin{aligned}
\mu_n = E[S_n] &\sim \binom{n+1}{2}-\sqrt{\frac{\pi}{8}}n^{3/2},\\
E[S_n^2] &\sim \binom{n+1}{2}^2-n(n+1)\sqrt{\frac{\pi}{8}}n^{3/2}+\frac{5n^3}{12}+\sqrt{\frac{\pi}{8}}n^{3/2}.
\end{aligned}$$

Hence, if σ_n is the variance of S_n, then

$$\sigma_n^2 \sim (\frac{5}{12}-\frac{\pi}{8})n^3 \approx 0.0239676n^3.$$

Using these formulas and Chebyshev's inequality,

$$\Pr(|S_n-\mu_n|\geq\sqrt{\frac{\pi}{8}}n^{3/2})\leq\frac{8\sigma^2}{\pi}\approx 0.061033,$$

so that about 94% of ordinary parking functions have sums which are at least

$$\binom{n+1}{2}-\sqrt{\frac{\pi}{2}}n^{3/2}.$$

Moreover, as $K\to\infty$,

$$\Pr(|S_n-\mu_n|>Kn^{3/2})\leq\frac{\sigma_n^2}{K^2}\to 0,$$

in other words, when n is large, most ordinary parking functions have sums which are close to $\binom{n+1}{2}$, the largest possible value. Can one prove a similar result for **u**-parking functions? A less speculative unsolved problem is to extend asymptotical results for ordinary parking functions to the classical case when **u** is an arithmetic progression.

13 Variants of parking functions

13.1. Reversed parking functions.

Let $\mathbf{u}$ be a sequence of non-decreasing positive integers. A *reversed* $\mathbf{u}$*-parking function* of length n on the discrete interval $[1, x]$ is a sequence $(x_1, x_2, \ldots, x_n)$ with terms in $[1, x]$ whose sequence of order statistics satisfies $x_{(i)} \geq u_i$. The *suites majeures* of Kreweras [12] are special cases of reversed parking functions.

As the astute reader might have noticed, Gončarov polynomials are better matched with reversed parking functions. For example, if $x \geq u_n$, the number of reversed $\mathbf{u}$-parking functions on $[1, x]$ is simply $g_n(x; u_1, u_2, \ldots, u_n)$. Almost all the results about Gončarov polynomials stated in this paper can be given combinatorial proofs using reversed parking functions. We note also that the slight incompatibility of Gončarov polynomials and parking functions is overcome in this paper by the substitution $u_{i+1} = x - a_i$. This allows us to shift and reflect the domain, so that we are essentially working with reversed parking functions.

13.2. Injective parking functions.

An ordinary parking function is injective or one-to-one if and only if it is a permutation. Thus, injective $\mathbf{u}$-parking functions may be considered generalizations of permutations.

Let $Q_n(u_1, u_2, \ldots, u_n)$ be the number of injective $\mathbf{u}$-parking functions of length n. Since it is almost immediate that the decomposition for integer sequences or discrete functions described in Section 5 works when restricted to injective functions, we have the following theorem.

(13.1) Theorem. Let x be an integer greater than or equal to u_n. Then

$$(x)_n = \sum_{m=0}^{n} \binom{n}{m} (x - u_{m+1})_{n-m} Q_m(u_1, u_2, \ldots, u_m).$$

Injective parking functions has a theory parallel to the one given in this paper. It is based on "difference" Gončarov polynomials (see [13] for the definition). For example, $Q_n(u_1, u_2, \ldots, u_n)$ equals $n! \det \mathcal{F}$, where $\mathcal{F}$ is the matrix with ijth entry equal to

$$\frac{(-u_i)_{j-i+1}}{(j-i+1)!}$$

if $j - i + 1 \geq 0$ and 0 otherwise.

13.3. Real-valued parking functions.

Let $\mathbf{u}$ be a non-decreasing sequence of non-negative real numbers. A *real-valued parking function of length* n is a sequence $(x_1, x_2, \ldots, x_n)$ of non-negative real numbers whose sequence of order statistics satisfies $x_{(i)} \leq u_i$. Using exactly the same proof, one can prove that sequences of length n with terms in the continuous interval $[0, x]$ satisfies the following decomposition, analogous to the one given in Corollary 5.2.

(13.2) Theorem. There is a bijection between the set $[0, x]^n$ of all length-n sequences with terms in the continuous interval $[0, x]$ and the disjoint union of cartesian products

$$\bigcup_{\{i_1, i_2, \ldots, i_m\}} \mathrm{Park}(i_1, i_2, \ldots, i_m) \times (u_{m+1}, x]^{n-m},$$

where $\mathrm{Park}(i_1, i_2, \ldots, i_m)$ is the set of real-valued length-m $\mathbf{u}$-parking functions indexed by $\{i_1, i_2, \ldots, i_m\}$ and $(u_{m+1}, x]^{n-m}$ is the set of length-$(m-n)$ sequences with terms in the continuous half-open interval $(u_{m+1}, x]$ indexed by the complement of $\{i_1, i_2, \ldots, i_m\}$.

Let $\bar{P}_n(\mathbf{u})$ be the probability that a random sequence $(X_1, X_2, \ldots, X_n)$ with the terms X_i chosen independently with uniform distribution from $[0, x]$ is a $\mathbf{u}$-parking function. Then, by Theorem 13.2, $\bar{P}_n(\mathbf{u})$ satisfies the following linear recursion:

$$1 = \sum_{m=0}^{n} \binom{n}{m} \frac{(x - u_{m+1})^{n-m}}{x^{n-m}} \bar{P}_m(u_1, u_2, \ldots, u_m).$$

Comparing this recursion with the recursion in Corollary 5.3, we obtain the following theorem.

(13.3) Theorem.

$$\bar{P}_n(u_1, u_2, \ldots, u_n) = \frac{P_n(u_1, u_2, \ldots, u_n)}{x^n}.$$

This theorem has appeared earlier in the paper [19]. Pitman and Stanley proved this theorem using their decomposition for **u**-parking functions (which works for real numbers u_i also) described in Section 6.

Theorems 5.4 and 13.3 together imply that

$$\bar{P}_n(u_1, u_2, \ldots, u_n) = \frac{(-1)^n g_n(0; u_1, u_2, \ldots, u_n)}{x^n}.$$

The analogue for "reversed" real-valued **u**-parking functions is usually stated in terms of an integral.

(13.4) Theorem. Let $0 \le u_n \le u_{n-1} \le \ldots \le u_1 \le x$. Then the probability that a length-n sequence $(X_1, X_2, \ldots, X_n)$ with terms X_i chosen independently with uniform distribution from $[0, x]$ satisfies the conditions $X_{(i)} \ge u_i, i = 1, 2, \ldots, n$, is

$$\frac{n!}{x^n} \int_{u_1}^{x} \int_{u_2}^{t_1} \cdots \int_{u_n}^{t_{n-1}} dt_n dt_{n-1} \ldots dt_1.$$

Proof. Condition on the size of the $(n-1)$st order statistics and use some well-known facts (see, for example, [3], Section 1.7) about independence and densities of the order statistics for a sequence of independent uniformly distributed random variables on $[0, x]$. See, for example, [28], [17], or [13].

This theorem seems to be first stated and proved by Steck [28] and rediscovered by many others.

Using the decomposition, we can also obtain the following recursion for the expected sums $\bar{E}_1(n; \mathbf{u})$ of random length-n real-valued **u**-parking functions:

$$\frac{nx}{2} = \sum_{m=0}^{n} \binom{n}{m} \frac{(x - u_{m+1})^{n-m} P_m(\mathbf{u})}{x^n} \left(\bar{E}_1(m; \mathbf{u}) + \frac{(n-m)(x + u_{m+1})}{2} \right).$$

This recursion can be obtained from the recursion for integer-valued parking functions by deleting all terms not of total degree $n+1$. Hence, $P_n(\mathbf{u})\bar{E}_1(n; \mathbf{u})$ is a homogeneous polynomial in the variables $u_1, u_2, \ldots, u_n$ of total degree $n+1$ and equals

$$\frac{n}{2} \sum_{j=1}^{n} \binom{n-1}{j-1} u_j^2 \frac{P_{n-j}(u_{j+1} - u_j, u_{j+2} - u_j, \ldots, u_n - u_j) P_{j-1}(u_1, u_2, \ldots, u_{j-1})}{P_n(u_1, u_2, \ldots, u_n)}.$$

In the ordinary case, we have the formula

$$\bar{E}_1(n; a, a+b, \ldots, a+(n-1)b) = \frac{na}{2} + \frac{1}{2} \sum_{j=2}^{n} \frac{n!}{(n-j)!} \frac{b^{j-1}(a + (j-1)b)}{(a+nb)^{j-1}}.$$

14 Historical remarks

The idea behind parking functions has occurred in many different subjects and its history is replete with rediscoveries. No one paper contains a complete overview, but if one combines four papers, one can obtain a reasonably complete picture. The first paper is Niederhausen [17]. It offers a good survey of the use of real-valued parking functions in statistics up to around 1980. A comprehensive account of how parking

functions occur in statistics and the study of certain polytopes and arrangements of hyperplanes can be found in Pitman and Stanley [19]. An excellent bibliography of work on bijections between ordinary parking functions and labelled trees (to around 2000) can be found in Gilbey and Kalikow [6]. Finally, a clear discussion of hashing and its relations to ordinary parking functions can be found in the paper of Flajolet, Poblete and Viola [4].

References

[1] R. P. Boas and R. C. Buck, "Polynomial Expansion of Analytic Functions," Springer-Verlag, Heidelberg, 1958.

[2] P. J. Davis, "Interpolation and Approximation," Blaisdell, Waltham MA, 1963; reprinted, Dover, New York, 1975.

[3] W. Feller, "An Introduction to Probability Theory and its Applications," Vol. II, 2nd Edition, Wiley, New York, 1971.

[4] P. Flajolet, P. Poblete and A. Viola, On the analysis of linear probing hashing, *Algorithmica* **22** (1998), 490–515.

[5] I. M. Gessel and B. E. Sagan, The Tutte polynomial of a graph, depth-first search, and simplicial complex partitions, *Electron. J. Combin.* **3** (1996), No. 2, R9.

[6] J. D. Gilbey and L. H. Kalikow, Parking functions, valet functions and priority queues, *Discrete Math.* **197/198** (1999), 351–373.

[7] V. L. Gončarov, "Theory of interpolation and approximation of functions," Gosudarstv. Izdat. Tehn.-Teor. Lit., Moscow, 1954.

[8] N. L. Johnson, S. Kotz, and A. W. Kemp, "Univariate Discrete Distributions," Wiley, New York, 1993.

[9] D. E. Knuth, "Sorting and Searching, The Art of Computer Programming, Vol. 3," Addison-Wesley, Reading, MA, 1973.

[10] D. E. Knuth, Linear probing and graphs, average-case analysis for algorithms, *Algorithmica* **22** (1998), 561–568.

[11] A. G. Konheim and B. Weiss, An occupancy discipline and applications, *SIAM J. Appl. Math.* **14** (1966), 1266–1274.

[12] G. Kreweras, Une famille de polynômes ayant plusieurs propriétés énumeratives, *Period. Math. Hungar.* **11** (1980), 309–320.

[13] J. P. S. Kung, A probabilistic interpretation of the Gončarov and related polynomials, *J. Math. Anal. Appl.* **79** (1981), 349–351.

[14] J. P. S. Kung and C. H. Yan, Moments of sums of parking functions, preprint 2001.

[15] C. L. Mallows and J. Riordan, The inversion enumerator for labeled trees, *Bull. Amer. Math. Soc.* **74** (1968), 92–94.

[16] R. Mullin and G.-C. Rota, On the foundations of combinatorial theory. III. Theory of binomial enumeration, in B. Harris, ed., "Graph Theory and its Applications," Academic Press, New York, 1970, pp. 167–213; reprinted in J. P. S. Kung, ed., "Gian-Carlo Rota on Combinatorics," Birkhäuser, Boston and Basel, 1995, pp. 118–147.

[17] H. Niederhausen, Sheffer polynomials for computing exact Kolmogorov-Smirnov and Rényi type distributions, *Ann. Statist.* **9** (1981) 923–944.

[18] M. Petkovšek, H. S. Wilf, and D. Zeilberger, "$A = B$," A. K. Peters, Wellesley, MA, 1996.

[19] J. Pitman and R. Stanley, A polytope related to empirical distributions, plane trees, parking functions, and the associahedron, Preprint, 1999.

[20] A. Rényi, On connected graphs I, *MTA Mat. Kut. Int. Közl.*, **4** (1959), 385–388.

[21] J. Riordan, "Combinatorial Identities," Wiley, New York, 1968.

[22] S. Roman, Polynomials, power series and interpolation, *J. Math. Anal. Appl.* **80** (1981), 333–371.

[23] N. J. A. Sloane and S. Plouffe, "The Encyclopedia of Integer Sequences," Academic Press, San Diego, CA, 1995.

[24] J. Spencer, Enumerating graphs and Brownian motion, *Comm. Pure Appl. Math.* **50** (1997), 291–294.

[25] R. P. Stanley, "Enumerative combinatorics," Vol. 1, 2nd edition, Cambridge Univ. Press, Cambridge, 2001.

[26] R. P. Stanley, Hyperplane arrangements, interval orders, and trees, *Proc. Nat. Acad. Sci.* **93** (1996), 2620–2625.

[27] R. Stanley, Hyperplane arrangements, parking functions, and tree inversions, in B. Sagan and R. Stanley, eds., "Mathematical essays in honor of Gian-Carlo Rota," Birkhäuser, Boston and Basel, 1998, pp. 359–375.

[28] G. P. Steck, The Smirnov two-sample tests as rank test, *Ann. Math. Statist.* **40** (1968), 1449–1466.

[29] J. Stirling, Methodus differentialis Newtoniana illustrata, *Philos. Trans.* **30** (1719), 1050–1070.

[30] E. M. Wright, The number of connected sparsely edged graphs, *J. Graph Theory*, **1** (1977), 317–330.

[31] C. H. Yan, On the enumeration of generalized parking functions, Proceedings of the 31st Southeastern Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, FL, 2000), *Congressus Numerantium,* **147** (2000), 201–209.

[32] C. H. Yan, Generalized parking functions, tree inversions and multicolored graphs, *Adv. Appl. Math.*, to appear, 2001.

RESTRICTED 132-DUMONT PERMUTATIONS

Toufik Mansour [1]

Department of Mathematics, Chalmers University of Technology, S-41296 Göteborg, Sweden

toufik@math.chalmers.se

Abstract

In [D] Dumont showed that certain classes of permutations on n letters are counted by the Genocchi numbers. In particular, Dumont showed that the $(n+1)$st Genocchi number is the number of permutations on $2n$ letters with the following properties: (1) each even integer must be followed by a smaller integer (this rule disallows the sequence from ending with an even integer), (2) each odd integer is either followed by a larger integer or is final in the sequence. We call such permutations by *Dumont permutations of the first kind*. In this paper we study the number of Dumont permutations of the first kind on n letters avoiding the pattern 132 and avoiding (or containing exactly once) an arbitrary pattern on k letters. In several interesting cases the generating function depends only on k.

Keywords: Dumont permutations, restricted permutations, generating functions.

1. Introduction

Classical patterns. Let $\alpha \in \mathfrak{S}_n$ and $\tau \in \mathfrak{S}_k$ be two permutations. We say that α *contains* τ if there exists a subsequence $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ such that $(\alpha_{i_1}, \ldots, \alpha_{i_k})$ is order-isomorphic to τ; in such a context τ is usually called a *pattern*. We say that α *avoids* τ, or is τ-*avoiding*, if such a subsequence does not exist. The set of all τ-avoiding permutations in $\mathfrak{S}_n$ is denoted $\mathfrak{S}_n(\tau)$. For an arbitrary finite collection of patterns T, we say that α avoids T if α avoids any $\tau \in T$; the corresponding subset of $\mathfrak{S}_n$ is denoted $\mathfrak{S}_n(T)$.

While the case of permutations avoiding a single pattern has attracted much attention, the case of multiple pattern avoidance remains less investigated. In particular, it is natural, as the next step, to consider permutations avoiding pairs of patterns τ_1, τ_2. This problem was solved completely for $\tau_1, \tau_2 \in \mathfrak{S}_3$ (see [SS]), for $\tau_1 \in \mathfrak{S}_3$ and $\tau_2 \in \mathfrak{S}_4$ (see [W]), and for $\tau_1, \tau_2 \in \mathfrak{S}_4$ (see [Bo1, Km] and references therein). Several recent papers [CW, MV1, Kr, MV2, MV3, MV4] deal with the case $\tau_1 \in \mathfrak{S}_3$, $\tau_2 \in \mathfrak{S}_k$ for various pairs τ_1, τ_2. Another natural question is to study permutations avoiding τ_1 and containing τ_2 exactly t times. Such a problem for certain $\tau_1, \tau_2 \in \mathfrak{S}_3$ and $t = 1$ was investigated in [R], and for certain $\tau_1 \in \mathfrak{S}_3$, $\tau_2 \in \mathfrak{S}_k$ in [RWZ, MV1, Kr]. The tools involved in these papers include generating trees, continued fractions, Chebyshev polynomials, and Dyck words. Also, the tools involved in these papers include many sequences, for example sequence of Catalan numbers, Fibonacci numbers, and Pell numbers.

[1] Research financed by EC's IHRP Programme, within the Research Training Network "Algebraic Combinatorics in Europe", grant HPRN-CT-2001-00272

We denote the nth *Catalan number* by $C_n = \frac{1}{n+1}\binom{2n}{n}$. The generating function for the Catalan numbers is denoted by $C(x)$, that is, $C(x) = \sum_{n\geq 0} C_n x^n = \frac{1-\sqrt{1-4x}}{2x}$.

Generalized patterns. In [BS] introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. We write a classical pattern with dashes between any two adjacent letters of the pattern, say 1342, as 1-3-4-2, and if we write, say 24-3-1, then we mean that if this pattern occurs in permutation $\pi \in \mathfrak{S}_n$, then the letters in the permutation π that correspond to 2 and 4 are adjacent (see [C]). For example, the permutation $\pi = 35421$ has only two occurrences of the pattern 23-1, namely the subsequences 352 and 351, whereas π has four occurrences of the pattern 2-3-1, namely the subsequences 352, 351, 342, and 341.

Claesson [C] presented a complete solution for the number of permutations avoiding any single generalized pattern of length three with exactly one adjacent pair of letters. Claesson and Mansour [CM] presented a complete solution for the number of permutations avoiding any double generalized patterns of length three with exactly one adjacent pair of letters. Kitaev [Ki] investigate simultaneous avoidance of two or more 3-letters generalized patterns without internal dashes. Later, Mansour [M1, M2] (for more details see [M3]) presented a general approach to study the number of permutations avoiding 1-3-2 and avoiding (or containing exactly once) an arbitrary generalized pattern.

Dumont permutations. Dumont [D] showed that certain classes of permutations in $\mathfrak{S}_n$ are counted by the Genocchi numbers (see [SP, Sequence A001469(M3041)]). Dumont showed that the $(n+1)$st Genocchi number is the number of permutations in $\mathfrak{S}_{2n}$ with the following properties: (1) each even integer must be followed by a smaller integer (this rule disallows the sequence from ending with an even integer), (2) each odd integer is either followed by a larger integer or is final in the sequence. We call such permutations by *Dumont permutations of the first kind*. For example, 2143, 3421, and 4213 are the all Dumont permutations of the first kind of length 4.

Dumont [D] defined another type of permutations in $\mathfrak{S}_n$ and showed that the $(n+1)$st Genocchi number is the number of permutations in $\mathfrak{S}_{2n}$ with the following properties: (1) $\pi_i < i$ for any even position i, (2) $\pi_i \geq i$ for any odd position i. We call such permutations by *Dumont permutations of the second kind*. For example, 2143, 3142, 4132 are the all Dumont permutations of the second kind of length 4.

Remark 1.1. *Let $\pi \in \mathfrak{S}_n$ be any Dumont permutation of the second kind; since $\pi_2 < 2$ we get $\pi_2 = 1$. Hence, it is easy to see that there are no Dumont permutations of the second kind in $\mathfrak{S}_n(132)$ for all $n \geq 4$. So, in this paper we discus only the case of Dumont permutations of the first kind.*

We define for all $r \geq 2$,

$$Q_r(x) = 1 + \frac{x^2 Q_{r-1}(x)}{1 - x^2 Q_{r-2}(x)}. \tag{1.1}$$

We denote the solution of Recurrence 1.1 with $Q_0(x) = 0$ and $Q_1(x) = 1$ by $F_r(x)$, and we denote the solution of Recurrence 1.1 with $Q_0(x) = Q_1(x) = 1$ by $G_r(x)$. For example, $F_2(x) = 1 + x^2$, $F_3(x) = \frac{1+x^4}{1-x^2}$, $G_2(x) = \frac{1}{1-x^2}$, and $G_3(x) = \frac{1-x^2+x^4}{(1-x^2)^2}$. Evidently, $F_r(x)$ and $G_r(x)$ are a rational functions in x^2, and for all $r \geq 1$,

$$F_r(x) = 1 + \sum_{j=1}^{r-1} \frac{x^{2j}}{\prod_{m=r-1-j}^{r-2}(1 - x^2 F_m(x))} \text{ and } G_r(x) = 1 + \sum_{j=1}^{r-1} \frac{x^{2j}}{\prod_{m=r-1-j}^{r-2}(1 - x^2 G_m(x))}. \tag{1.2}$$

Example 1.2. *Using Recurrence 1.1 it is easy to see that*

$$F_4(\sqrt{x}) = \sum_{n\geq 0}(f_{n+2}+f_n-2)x^n \text{ and } G_4(\sqrt{x}) = 1+x+\sum_{n\geq 2}(3\cdot 2^{n-2}-1)x^n,$$

where f_n is the nth Fibonacci number.

Organization of the paper. In this paper we use generating function techniques to study those Dumont permutations of the first kind which avoid 132 and avoid (or contain exactly once) an arbitrary pattern on k letters. In several interesting cases the generating function depends only on k.

The paper is organized as follows. The case of Dumont permutations of the first kind avoiding both 132 and τ is treated in Section 2. We present a simple structure for any Dumont permutation of the first kind avoiding 132. This structure can be obtained explicitly for several interesting cases, including classical patterns and generalized patterns. This allows us to find explicitly some statistics on Dumont permutations of the first kind which avoid 132. The case of avoiding 132 and containing another pattern τ exactly once is treated in Section 3. Again, we find explicitly the generating function for several interesting cases of τ, including classical patterns and generalized patterns.

Most of the explicit solutions obtained in Sections 2-4 involve the generating functions $F_k(x)$ and $G_k(x)$.

2. Dumont permutations of the first kind which avoid 132 and another pattern

Let $\mathfrak{D}^{(1)}_\tau(n)$ denote the number of Dumont permutations of the first kind in $\mathfrak{S}_n(132,\tau)$, and let $\mathfrak{D}^{(1)}_\tau(x) = \sum_{n\geq 0}\mathfrak{D}^{(1)}_\tau(n)x^n$ be the corresponding generating function. In this section we describe a method for enumerating Dumont permutations of the first kind which avoid 132 and another pattern and we use our method to enumerate $\mathfrak{D}^{(1)}_\tau(n)$ for various τ. We begin with an observation concerning the structure of the Dumnot permutations of the first kind avoiding 132 which holds immediately from definitions.

Proposition 2.1. *For any $\pi\in\mathfrak{D}^{(1)}_n(132)$ such that $\pi_j = n$, there holds one of the following assertions:*

1. *if n is odd number then $\pi = (\pi', n)$, where $\pi'\in\mathfrak{D}^{(1)}_{n-1}(132)$;*
2. *if n is even number then $\pi = (\pi', n, \pi'')$ such that π' is a Dumont permutation of the first kind on the numbers $n-j+1, n-j+2, \dots, n-1$, π'' is nonempty Dumont permutation of the first kind on the numbers $1, 2, \cdots, n-j$, and $j = 1, 2, 4, \dots, n-2$.*

2.1. $\tau = \varnothing$**.** As a corollary of Proposition 2.1 we find an explicit formula for the number of 132-avoiding Dumont permutations of the first kind in $\mathfrak{S}_n$.

Theorem 2.2. *The generating function for the number of 132-avoiding Dumont permutations of the first kind in $\mathfrak{S}_n$ is given by $(1+x)C(x^2)$. In other words, the number of 132-avoiding Dumont permutations of the first kind in $\mathfrak{S}_n$ is given by $C_{[n/2]}$, which is the $[n/2]$th Catalan number.*

Proof. By Proposition 2.1, we have two possibilities for block decomposition of an arbitrary $\pi\in\mathfrak{D}^{(1)}_n(132)$. Let us write an equation for $\mathfrak{D}^{(1)}_\varnothing(x)$. The contribution of the first decomposition above equals

$$\sum_{n\geq 0}\mathfrak{D}^{(1)}_\varnothing(2n+1)x^{2n+1} = x\sum_{n\geq 0}\mathfrak{D}^{(1)}_\varnothing(2n)x^{2n},$$

equivalently,

$$\mathfrak{D}^{(1)}_{\varnothing}(x) - \mathfrak{D}^{(1)}_{\varnothing}(-x) = x(\mathfrak{D}^{(1)}_{\varnothing}(x) + \mathfrak{D}^{(1)}_{\varnothing}(-x)). \tag{2.1}$$

The contribution of the second decomposition above equals

$$\sum_{n\geq 1} \mathfrak{D}^{(1)}_{\varnothing}(2n)x^{2n} = \sum_{n\geq 1} \mathfrak{D}^{(1)}_{\varnothing}(2n-1)x^{2n} + \sum_{n\geq 1}\sum_{j=0}^{n} \mathfrak{D}^{(1)}_{\varnothing}(2j+1)\mathfrak{D}^{(1)}_{\varnothing}(2n+2-2j)x^{2n},$$

equivalently,

$$\begin{aligned}\mathfrak{D}^{(1)}_{\varnothing}(x) + \mathfrak{D}^{(1)}_{\varnothing}(-x) - 2 = \\ = x(\mathfrak{D}^{(1)}_{\varnothing}(x) - \mathfrak{D}^{(1)}_{\varnothing}(-x)) + \tfrac{x}{2}(\mathfrak{D}^{(1)}_{\varnothing}(x) - \mathfrak{D}^{(1)}_{\varnothing}(-x))(\mathfrak{D}^{(1)}_{\varnothing}(x) + \mathfrak{D}^{(1)}_{\varnothing}(-x) - 2).\end{aligned} \tag{2.2}$$

By putting $\mathfrak{D}^{(1)}_{\varnothing}(x) = (1+x)A(x)$ in Equations 2.1 and 2.2 it is easy to see that $A(x) = C(x^2)$. □

2.2. **A classical pattern $\tau = 12\dots k$.** Let us start by the following example.

Example 2.3. *By definitions we have $\mathfrak{D}^{(1)}_{1}(x) = 1$ and $\mathfrak{D}^{(1)}_{12}(x) = 1 + x + x^2$.*

The case of varying k is more interesting. As an extension of Example 2.3, let us consider the case $\tau = 12\dots k$.

Theorem 2.4. *Let $A_k(x) = \frac{1}{2}(\mathfrak{D}^{(1)}_{12\dots k}(x) + \mathfrak{D}^{(1)}_{12\dots k}(-x))$ and $B_k(x) = \frac{1}{2}(\mathfrak{D}^{(1)}_{12\dots k}(x) - \mathfrak{D}^{(1)}_{12\dots k}(-x))$ for all $k \geq 0$. Then*

$$A_k(x) = F_k(x), \qquad B_k(x) = xF_{k-1}(x), \quad \text{and} \quad \mathfrak{D}^{(1)}_{12\dots k}(x) = F_k(x) + xF_{k-1}(x).$$

Proof. Using the same arguments as in the proof of Theorem 2.2 we get

$$\mathfrak{D}^{(1)}_{12\dots k}(x) - \mathfrak{D}^{(1)}_{12\dots k}(-x) = x(\mathfrak{D}^{(1)}_{12\dots (k-1)}(x) + \mathfrak{D}^{(1)}_{12\dots (k-1)}(-x)),$$

and

$$\begin{aligned}\mathfrak{D}^{(1)}_{12\dots k}(x) + \mathfrak{D}^{(1)}_{12\dots k}(-x) - 2 = x(\mathfrak{D}^{(1)}_{12\dots k}(x) - \mathfrak{D}^{(1)}_{12\dots k}(-x)) + \\ + \tfrac{x}{2}(\mathfrak{D}^{(1)}_{12\dots (k-1)}(x) - \mathfrak{D}^{(1)}_{12\dots (k-1)}(-x))(\mathfrak{D}^{(1)}_{12\dots k}(x) + \mathfrak{D}^{(1)}_{12\dots k}(-x) - 2).\end{aligned}$$

The rest is easy to check by the definitions of A_k and B_k. □

Example 2.5. *Theorem 2.4, for $k = 3$, yields $\mathfrak{D}^{(1)}_{123}(x) = \frac{1+x+x^4-x^5}{1-x^2}$. In other words, the number of 132-avoiding Dumont permutation of the first kind in $\mathfrak{S}_n(123)$ is given by $1+(-1)^n$ for all $n \geq 4$, and 1 for $n = 0, 1, 2, 3$. An another example, Theorem 2.4, for $k = 4$, yields $\mathfrak{D}^{(1)}_{1234}(x) = \frac{1+2x+x^2+2x^6+x^7+x^8}{(1+x)(1-x^2-x^4)}$. In other words, the number of 132-avoiding Dumont permutation of the first kind in $\mathfrak{S}_n(1234)$ is $f_{n/2+2} + f_{n/2} - 2$ if n is even number, otherwise 2 for all $n \geq 2$, where f_n is the nth Fibonacci number.*

As an extension of Theorem 2.4, let us define

$$\mathfrak{A}(x_1, x_2, x_3, \dots) = \sum_{\pi \in \mathfrak{D}^{(1)}} \prod_{j\geq 1} x_1^{12\dots j(\pi)},$$

where $\mathfrak{D}^{(1)}$ is the set of all Dumont permutations of the first kind of all sizes including the empty permutation, and $\tau(\pi)$ is the number of occurrences of τ in π. Let

$$\begin{aligned}A^{(1)}(x_1, x_2, x_3, \dots) = \tfrac{1}{2}(\mathfrak{A}(x_1, x_2, x_3, \dots) + \mathfrak{A}(-x_1, x_2, x_3, \dots)), \\ B^{(1)}(x_1, x_2, x_3, \dots) = \tfrac{1}{2}(\mathfrak{A}(x_1, x_2, x_3, \dots) - \mathfrak{A}(-x_1, x_2, x_3, \dots)).\end{aligned}$$

Using the same arguments as in the proof of Theorem 2.4, we obtain the following.

Theorem 2.6. *We have*

$$A^{(1)}(x_1,x_2,x_3,\dots)=1+\frac{x_1^2A^{(1)}(x_1x_2,x_2x_3,x_3x_4,\dots)}{1-x_1^2x_2A^{(1)}(x_1x_2^2x_3,x_2x_3^2x_4,x_3x_4^2x_5,\dots)},$$

and

$$B^{(1)}(x_1,x_2,x_3,\dots)=x_1A^{(1)}(x_1x_2,x_2x_3,x_3x_4,\dots).$$

Aa an application to Theorem 2.6, for $x_1=x$ and $x_j=1$, $j\geq 2$, we get that

$$B^{(1)}(x,1,1,\dots)=xA^{(1)}(x,1,1,\dots),$$

and

$$A^{(1)}(x,1,1,\dots)=\cfrac{1}{1-\cfrac{x^2}{1-\cfrac{x^2}{\ddots}}}=C(x^2).$$

Hence, we have $\mathfrak{D}^{(1)}_{\varnothing}(x)=(1+x)C(x^2)$ (see Theorem 2.2).

An another application for Theorem 2.6 is the number of right to left maxima. Let $\pi\in\mathfrak{S}_n$, π_i is a *right to left maxima* if $\pi_i>\pi_j$ for all $i<j$. We denote the number of right to left maxima of π by $rlm(\pi)$. In [BCS, Proposition 5] proved

$$lrm(\pi)=\sum_{j\geq 1}12\dots j(\pi)(-1)^{j-1}.$$

Therefore,

$$\sum_{\pi\in\mathfrak{D}^{(1)}}x^{|\pi|}y^{rlm(\pi)}=\mathfrak{A}(xy,y^{-1},y,y^{-1},\dots)$$

together with Theorem 2.6 and $A^{(1)}(x,1,1,\dots)=C(x^2)$ we get

$$\sum_{\pi\in\mathfrak{D}^{(1)}}x^{|\pi|}y^{rlm(\pi)}=1+xC(x^2)y+\sum_{n\geq 2}x^{2n-2}C^{n-1}(x^2)y^n.$$

Corollary 2.7. *The generating function for the number of Dumont permutations of the first kind avoiding* 132 *and having exactly k right to left maxima is given by $x^{2k-2}C^{k-1}(x^2)$ for all $k\geq 2$, and $x^kC^k(x^2)$ for $k=0,1$.*

2.3. **A classical pattern $\tau=2134\dots k$.** Similarly as in Theorem 2.4, we obtain the case $\tau=2134\dots k$.

Theorem 2.8. *For all $k\geq 2$,*

$$\mathfrak{D}^{(1)}_{213\dots k}(x)=G_{k-1}(x)+xG_{k-2}.$$

Example 2.9. *Theorem 2.8 for $k=3,4$ yields $\mathfrak{D}^{(1)}_{213}(x)=\frac{1+x-x^3}{1-x}$ and $\mathfrak{D}^{(1)}_{2134}(x)=\frac{1+x-x^2-x^3+x^4}{(1-x^2)^2}$.*

2.4. **A generalized pattern 12-3-$\cdots$-k.** In this subsection we use the notation of generalized patterns (see Section 1). For example, we write the classical pattern 132 as 1-3-2.

By definitions, we get $\mathfrak{D}_{12}^{(1)}(x) = 1+x+x^2$. So, by the same arguments as in the proof of Theorem 2.4, together with

$$\mathfrak{D}_{12}^{(1)}(x) = \mathfrak{D}_{1\text{-}2}^{(1)}(x) = 1 + x + x^2,$$

we obtain the following.

Theorem 2.10. *For all $k \geq 1$,*

$$\mathfrak{D}_{12\text{-}3\text{-}\ldots\text{-}k}^{(1)}(x) = \mathfrak{D}_{1\text{-}2\text{-}3\text{-}\ldots\text{-}k}^{(1)}(x) = F_k(x) + xF_{k-1}(x).$$

A comparison of Theorem 2.4 with Theorem 2.10 suggests that there should exist a bijection between the sets $\mathfrak{S}_n(1\text{-}3\text{-}2, 12\text{-}3\text{-}\cdots\text{-}k)$ and $\mathfrak{S}_n(1\text{-}3\text{-}2, 1\text{-}2\text{-}3\text{-}\cdots\text{-}k)$. However, we failed to produce such a bijection, and finding it remains a challenging open question.

Now, let us define

$$\mathfrak{B}(x_1, x_2, x_3, \ldots) = \sum_{\pi \in \mathfrak{D}^{(1)}} x_1^{1(\pi)} \prod_{j \geq 2} x_1^{12\text{-}3\text{-}\ldots\text{-}j(\pi)},$$

where $\mathfrak{D}^{(1)}$ is the set of all Dumont permutations of the first kind of all sizes including the empty permutation, and $\tau(\pi)$ is the number of occurrences of τ in π. Let

$$A^{(2)}(x_1, x_2, x_3, \ldots) = \tfrac{1}{2}(\mathfrak{B}(x_1, x_2, x_3, \ldots) + \mathfrak{B}(-x_1, x_2, x_3, \ldots)),$$
$$B^{(2)}(x_1, x_2, x_3, \ldots) = \tfrac{1}{2}(\mathfrak{B}(x_1, x_2, x_3, \ldots) - \mathfrak{B}(-x_1, x_2, x_3, \ldots)).$$

Using the same arguments as those in the proof of Theorem 2.4, we get

Theorem 2.11.

$$A^{(2)}(x_1, x_2, x_3, \ldots) = 1 + \frac{x_1^2(1 - x_2 + x_2 A^{(2)}(x_1, x_2x_3, x_3x_4, \ldots))}{1 - x_1^2x_2(1 - x_2x_3 + x_2x_3A^{(2)}(x_1, x_2x_3^2x_4, x_3x_4^2x_5, \ldots))},$$

and

$$B^{(2)}(x_1, x_2, x_3, \ldots) = x_1 - x_1x_2 + x_1x_2A^{(2)}(x_1, x_2x_3, x_3x_4 \ldots).$$

Let $\pi \in \mathfrak{S}_n$; we say π_j is a *rise* for π if $\pi_j < \pi_{j+1}$ for all $j = 1, 2, \ldots, n-1$. We denote the number of rises of π by $rises(\pi)$. By definitions, we have

$$\sum_{\pi \in \mathfrak{D}^{(1)}} x^{|\pi|} y^{rises(\pi)} = x - xy + (1+xy)A^{(2)}(x, y, 1, 1, \ldots),$$

so an application for Theorem 2.11 we get

Corollary 2.12. *The generating function $\sum_{\pi \in \mathfrak{D}^{(1)}} x^{|\pi|} y^{rises(\pi)}$ is given by*

$$\frac{1 + xy - 2x^2y + 2x^2y^2 - (1+xy)\sqrt{1 - 4x^2y}}{2x^2y^2}.$$

In other words, the generating function for Dumont permutations of the first kind avoiding 1-3-2 with exactly k rises is given by $C_k x^{2k+1} + C_{k+1}x^{2k+2}$ for all $k \geq 1$, and $1 + x + x^2$ for $k = 0$, where C_m is the mth Catalan number.

2.5. **A generalized pattern $\tau = 21\text{-}3\text{-}\cdots\text{-}k$.** In this subsection, we use the notation of generalized patterns (see Section 1). For example, we write the classical pattern 132 as 1-3-2.

By definitions, we get $\mathfrak{D}^{(1)}_{21}(x) = 1 + x$. So, by the same arguments as in the proof of Theorem 2.4 together with

$$\mathfrak{D}^{(1)}_{21}(x) = \mathfrak{D}^{(1)}_{2\text{-}1}(x) = 1 + x,$$

we obtain the following.

Theorem 2.13. *For all $k \geq 2$,*

$$\mathfrak{D}^{(1)}_{21\text{-}3\text{-}\ldots\text{-}k}(x) = \mathfrak{D}^{(1)}_{2\text{-}1\text{-}3\text{-}\ldots\text{-}k}(x) = G_{k-1}(x) + xG_{k-2}(x).$$

A comparison of Theorem 2.8 with Theorem 2.13 suggests that there should exist a bijection between the sets $\mathfrak{S}_n(1\text{-}3\text{-}2, 21\text{-}3\text{-}\cdots\text{-}k)$ and $\mathfrak{S}_n(1\text{-}3\text{-}2, 2\text{-}1\text{-}3\text{-}\cdots\text{-}k)$. However, we failed to produce such a bijection, and finding it remains a challenging open question.

Now, let us define

$$\mathfrak{C}(x_1, x_2, x_3, \dots) = \sum_{\pi \in \mathfrak{D}^{(1)}} x_1^{1(\pi)} \prod_{j \geq 2} x_1^{21\text{-}3\text{-}\ldots\text{-}j(\pi)},$$

where $\mathfrak{D}^{(1)}$ is the set of all Dumont permutations of the first kind of all sizes including the empty permutation, and $\tau(\pi)$ is the number of occurrences of τ in π. Let

$$\begin{aligned} A^{(3)}(x_1, x_2, x_3, \dots) &= \tfrac{1}{2}(\mathfrak{C}(x_1, x_2, x_3, \dots) + \mathfrak{C}(-x_1, x_2, x_3, \dots)), \\ B^{(3)}(x_1, x_2, x_3, \dots) &= \tfrac{1}{2}(\mathfrak{C}(x_1, x_2, x_3, \dots) - \mathfrak{C}(-x_1, x_2, x_3, \dots)). \end{aligned}$$

Using the same arguments as in the proof of Theorem 2.4, we get the following.

Theorem 2.14. *We have*

$$A^{(3)}(x_1, x_2, x_3, \dots) = 1 + \frac{x_1^2 x_2 A^{(3)}(x_1, x_2x_3, x_3x_4, \dots)}{1 - x_1^2 x_2 A^{(3)}(x_1, x_2x_3^2x_4, x_3x_4^2x_5, \dots)},$$

and

$$B^{(3)}(x_1, x_2, x_3, \dots) = x_1 A^{(3)}(x_1, x_2x_3, x_3x_4 \dots).$$

Let $\pi \in \mathfrak{S}_n$; we say that π_j is a *descent* for π if $\pi_j > \pi_{j+1}$ for all $j = 1, 2, \dots, n-1$. We denote the number of descents of π by $descents(\pi)$. By definitions, we have

$$\sum_{\pi \in \mathfrak{D}^{(1)}} x^{|\pi|} y^{decents(\pi)} = (1 + x) A^{(3)}(x, y, 1, 1, \dots),$$

therefore an application for Theorem 2.14 we get

Corollary 2.15. *The generating function $\sum_{\pi \in \mathfrak{D}^{(1)}} x^{|\pi|} y^{descents(\pi)}$ is given by $(1+x)C(x^2y)$. In other words, the generating function for Dumont permutations of the first kind avoiding 1-3-2 with exactly k descents is given by $C_k x^{2k+1} + C_k x^{2k+2}$ for all $k \geq 0$, where C_m is the mth Catalan number.*

2.6. **A classical pattern $\tau = 23\dots k1$.** Again, Proposition 2.1 gives a complete answer for $\tau = 23\dots k1$.

Theorem 2.16. *For all $k \geq 3$,*

$$\mathfrak{D}^{(1)}_{23\dots k1}(x) = 1 + x + \frac{x^2(1+x)}{1 - x^2 - x^2 F_{k-3}(x)}.$$

Proof. Using the same arguments as in the proof of Theorem 2.2 we get

$$\mathfrak{D}^{(1)}_{23\dots k1}(x) - \mathfrak{D}^{(1)}_{23\dots k1}(-x) = x(\mathfrak{D}^{(1)}_{23\dots k1}(x) + \mathfrak{D}^{(1)}_{23\dots k1}(-x)),$$

and

$$\begin{aligned}\mathfrak{D}^{(1)}_{23\dots k1}(x) + \mathfrak{D}^{(1)}_{23\dots k1}(-x) - 2 = x(\mathfrak{D}^{(1)}_{23\dots k1}(x) - \mathfrak{D}^{(1)}_{23\dots k1}(-x)) + \\ + \tfrac{x}{2}(\mathfrak{D}^{(1)}_{12\cdots(k-2)}(x) - \mathfrak{D}^{(1)}_{12\dots(k-2)}(-x))(\mathfrak{D}^{(1)}_{23\dots k1}(x) + \mathfrak{D}^{(1)}_{23\dots k1}(-x) - 2).\end{aligned}$$

The rest is easy to check by the definitions of $F_k(x)$ together with Theorem 2.4. □

Example 2.17. *Theorem 2.16, for $k = 5$, yields $\mathfrak{D}^{(1)}_{23451}(x) = \frac{(1+x)(1-x^2-x^4)}{1-2x^2-x^4}$. In other words, the number of Dumont permutation of the first kind in $\mathfrak{S}_n(132, 23451)$ is given by $P_{[n/2]}$, which is the $[n/2]$th Pell number for all $n \geq 2$.*

3. Dumont permutations of the first kind which avoid 132 and contain another pattern exactly once

Let $\mathfrak{D}^{(1)}_{\tau;r}(n)$ denote the number of Dumont permutations of the first kind in $\mathfrak{S}_n(132)$ containing τ exactly r times, and let $\mathfrak{D}^{(1)}_{\tau;r}(x) = \sum_{n\geq 0} \mathfrak{D}^{(1)}_{\tau;r}(n)x^n$ be the corresponding generating function.

3.1. A classical pattern $\tau = 12\dots k$.

Theorem 3.1. *Let*

$$A_k(x) = \frac{x^2}{1 - x^2 F_{k-2}(x)} A_{k-1}(x) + \frac{x^4 F_{k-1}(x)}{(1 - x^2 F_{k-2}(x))^2} A_{k-2}(x)$$

for all $k \geq 2$, where $A_1(x) = 0$ and $A_2(x) = x^4$. Then for all $k \geq 2$

$$\mathfrak{D}^{(1)}_{12\dots k;1}(x) = A_k(x) + xA_{k-1}(x).$$

Proof. By Proposition 2.1, we have two possibilities for block decomposition of an arbitrary π in $\mathfrak{D}^{(1)}_n(132)$. Let us write an equation for $\mathfrak{D}^{(1)}_{12\dots k;1}(x)$. The contribution of the first decomposition above is

$$\sum_{n\geq 0} \mathfrak{D}^{(1)}_{12\dots k;1}(2n+1)x^{2n+1} = x \sum_{n\geq 0} \mathfrak{D}^{(1)}_{12\dots(k-1);1}(2n)x^{2n},$$

equivalently

$$\mathfrak{D}^{(1)}_{12\dots k;1}(x) - \mathfrak{D}^{(1)}_{12\dots k;1}(-x) = x(\mathfrak{D}^{(1)}_{12\dots(k-1);1}(x) + \mathfrak{D}^{(1)}_{12\dots(k-1);1}(-x)). \tag{3.1}$$

The contribution of the second decomposition above is

$$\sum_{n\geq 1}\mathfrak{D}^{(1)}_{12\dots k;1}(2n)x^{2n}=\sum_{n\geq 1}\mathfrak{D}^{(1)}_{12\dots k;1}(2n-1)x^{2n}+$$
$$+\sum_{n\geq 1}\sum_{j=0}^{n}\mathfrak{D}^{(1)}_{12\dots(k-1);1}(2j+1)\mathfrak{D}^{(1)}_{12\dots k;0}(2n+2-2j)x^{2n}+$$
$$+\sum_{n\geq 1}\sum_{j=0}^{n}\mathfrak{D}^{(1)}_{12\dots(k-1);0}(2j+1)\mathfrak{D}^{(1)}_{12\dots k;1}(2n+2-2j)x^{2n},$$

equivalently

$$\begin{aligned}\mathfrak{D}^{(1)}_{12\dots k;1}(x)+\mathfrak{D}^{(1)}_{12\dots k;1}(-x)&=x(\mathfrak{D}^{(1)}_{12\dots k;1}(x)-\mathfrak{D}^{(1)}_{12\dots k;1}(-x))+\\ &+\tfrac{x}{2}(\mathfrak{D}^{(1)}_{12\dots(k-1);1}(x)-\mathfrak{D}^{(1)}_{12\dots(k-1);1}(-x))(\mathfrak{D}^{(1)}_{12\dots k;0}(x)+\mathfrak{D}^{(1)}_{12\dots k;0}(-x)-2)+\\ &+\tfrac{x}{2}(\mathfrak{D}^{(1)}_{12\dots(k-1);0}(x)-\mathfrak{D}^{(1)}_{12\dots(k-1);0}(-x))(\mathfrak{D}^{(1)}_{12\dots k;1}(x)+\mathfrak{D}^{(1)}_{12\dots k;1}(-x)).\end{aligned}\tag{3.2}$$

Using Theorem 2.4, Equation 3.1, Equation 3.2, and Definition 1.1, we get the desired result. □

Example 3.2. *Theorem 3.1 for $k=3$ we get*

$$\mathfrak{D}^{(1)}_{123;1}(x)=\frac{x^5(1+x-x^2)}{1-x^2},$$

and for $k=4$ we get

$$\mathfrak{D}^{(1)}_{1234;1}(x)=\frac{x^7(1+x-3x^2+2x^3+3x^4+3x^5-x^6+x^7)}{(1-x^2)(1-x^2-x^4)^2}.$$

As an extension of Theorem 3.1, let us consider the case $r\geq 1$. Theorem 2.6, for given k and r, yields an explicit formula for $\mathfrak{D}^{(1)}_{12\dots k;r}(x)$. For example, for $k=3$ and $r=0,1,2,3,4$, we have the following.

Theorem 3.3. *We have*

(i) $\mathfrak{D}^{(1)}_{123;0}(x)=\dfrac{1+x+x^4-x^5}{1-x^2}$;

(ii) $\mathfrak{D}^{(1)}_{123;1}(x)=\dfrac{x^5(1+x-x^2)}{1-x^2}$;

(iii) $\mathfrak{D}^{(1)}_{123;2}(x)=\dfrac{x^5(1+x^2)(1+2x-2x^2-x^3+x^4)}{(1-x^2)^2}$;

(iv) $\mathfrak{D}^{(1)}_{123;3}(x)=\dfrac{x^7(1+x-x^2+x^3-x^4-x^5+x^6)}{(1-x^2)^2}$;

(v) $\mathfrak{D}^{(1)}_{123;4}(x)=\dfrac{x^9(1+x^2)(-1-3x+3x^2+3x^3-3x^4-x^5+x^6)}{(1-x^2)^2}$.

3.2. **A classical pattern $\tau=2134\dots k$.** Similarly to Theorem 3.1, we have

Theorem 3.4. *Let*

$$A_k(x)=\frac{x^2}{1-x^2G_{k-2}(x)}A_{k-1}(x)+\frac{x^4G_{k-1}(x)}{(1-x^2G_{k-2}(x))^2}A_{k-2}(x)$$

for all $k\geq 4$, where $A_1(x)=A_2(x)=x^2$ and $A_3(x)=\frac{x^4}{1-x^2}$. Then, for all $k\geq 2$,

$$\mathfrak{D}^{(1)}_{213\dots k;1}(x)=A_k(x)+xA_{k-1}(x).$$

3.3. **A generalized patterns $\tau = 12\text{-}3\text{-}\cdots\text{-}k$ and $\tau = 21\text{-}3\text{-}\cdots\text{-}k$.** Similarly to Theorem 3.1, we get

Theorem 3.5. *Let*

$$A_k(x) = \frac{x^2}{1-x^2F_{k-2}(x)}A_{k-1}(x) + \frac{x^4F_{k-1}(x)}{(1-x^2F_{k-2}(x))^2}A_{k-2}(x)$$

for all $k \geq 4$, where $A_1(x) = x^2$ and $A_2(x) = 2x^4$. Then, for all $k \geq 2$,

$$\mathfrak{D}^{(1)}_{12\text{-}3\text{-}\cdots\text{-}k;1}(x) = A_k(x) + xA_{k-1}(x).$$

As an extension of Theorem 3.5, let us consider the case $r \geq 1$. Theorem 2.11, for given k and r, yields an explicit formula for $\mathfrak{D}^{(1)}_{12\text{-}3\text{-}\cdots\text{-}k;r}(x)$. For example, for $k = 3$ and $r = 0, 1, 2, 3, 4$, we have the following.

Theorem 3.6. *We have*

(i) $\mathfrak{D}^{(1)}_{12\text{-}3;0}(x) = \dfrac{1+x+x^4-x^5}{1-x^2}$;

(ii) $\mathfrak{D}^{(1)}_{12\text{-}3;1}(x) = \dfrac{x^5(2+3x-4x^2-x^3+2x^4)}{(1-x^2)^2}$;

(iii)$\mathfrak{D}^{(1)}_{12\text{-}3;2}(x) = \dfrac{x^7(2+2x-6x^2-x^3+6x^4+x^5-2x^6)}{(1-x^2)^3}$;

(iv) $\mathfrak{D}^{(1)}_{12\text{-}3;3}(x) = \dfrac{x^7(3+5x-10x^2-9x^3+10x^4+3x^5+4x^7-5x^8-x^9+2x^10)}{(1-x^2)^4}$;

(v) $\mathfrak{D}^{(1)}_{12\text{-}3;4}(x) = \dfrac{x^9(5+5x-23x^2-7x^3+40x^4-x^5-30x^6+5x^7+5x^8-x^9+5x^{10}+x^{11}-2x^{12})}{(1-x^2)^5}$.

Similarly to Theorem 3.1, we have

Theorem 3.7. *Let*

$$A_k(x) = \frac{x^2}{1-x^2G_{k-2}(x)}A_{k-1}(x) + \frac{x^4G_{k-1}(x)}{(1-x^2G_{k-2}(x))^2}A_{k-2}(x)$$

for all $k \geq 4$, where $A_1(x) = A_2(x) = x^2$, $A_3(x) = \frac{x^4}{1-x^2}$, and $A_4(x) = \frac{x^6(2-x^2)}{(1-x^2)^3}$. Then, for all $k \geq 2$,

$$\mathfrak{D}^{(1)}_{21\text{-}3\text{-}\cdots\text{-}k;1}(x) = A_k(x) + xA_{k-1}(x).$$

As an extension of Theorem 3.7, let us consider the case $r \geq 1$. Theorem 2.14, for given k and r, yields an explicit formula for $\mathfrak{D}^{(1)}_{21\text{-}3\text{-}\cdots\text{-}k;r}(x)$. For example, for $k = 3$ and $r = 0, 1, 2, 3, 4$, we have the following.

Theorem 3.8. *We have*

(i) $\mathfrak{D}^{(1)}_{21\text{-}3;0}(x) = \dfrac{1+x+x^4-x^5}{1-x^2}$;

(ii) $\mathfrak{D}^{(1)}_{21\text{-}3;1}(x) = \dfrac{x^3(1+x-x^2)}{1-x^2}$;

(iii) $\mathfrak{D}^{(1)}_{21\text{-}3;2}(x) = \dfrac{x^5(1+2x-2x^2-x^3+x^4)}{(1-x^2)^2}$;

(iv) $\mathfrak{D}^{(1)}_{21\text{-}3;3}(x) = \dfrac{x^5(1+x-x^2+x^3-x^4-x^5+x^6)}{(1-x^2)^2}$;

(v) $\mathfrak{D}^{(1)}_{21\text{-}3;4}(x) = \dfrac{x^7(1+2x-2x^2-2x^5+2x^6+x^7-x^8)}{(1-x^2)^3}$.

4. Further results

Here we present three different directions to generalize the results of the previous sections. The first of these directions is to consider one occurrence of the classical pattern 132. For example, the following result is true.

Theorem 4.1. *There does not exist a Dumont permutation of the first kind containing* 132 *(classical pattern) exactly once.*

Proof. Let $\pi = (\pi', n, \pi'')$ be a Dumont permutation of the first kind of length n, which contain the pattern 132 exactly once. It is easy to see that there does not exist a Dumont permutation of the first kind where $n = 0, 1, 2, 3$. Suppose $n \geq 4$, and let us assume by induction on n that there does not exist a Dumont permutation of the first kind of length $m \leq n-1$ containing 132 exactly once. To prove this property for n, let us consider the following two cases together with using Proposition 2.1: n is either an even number, or n is an odd number.

1. Let n be an odd number. Since π is a Dumont permutation of the first kind, we get $\pi'' = \varnothing$, so π contains 132 exactly once if and only if π' contains 132 exactly once.
2. Let n be an even number. Since π is a Dumont permutation of the first kind we have $\pi'' \neq \varnothing$. Now, let us consider two cases: either n does not appear in the occurrence of 132, or n does it.
   (a) Let the occurrence of 132 does not contain the element n. So, every element of π' greater than every element of π''. Therefore, either π' is a Dumont permutation of the first kind of length $m \leq n-2$ contains 132 exactly once, or π'' is a Dumont permutation of the first kind of length $m \leq n-1$ contains 132 exactly once.
   (b) Let the occurrence of 132 contains the element n. So, $\pi = (\pi', a, n, \pi'', a+1, \pi''')$ (see [MV4]) such that $\pi_p = n$ and $\pi_q = a+1$, where every element of π' is greater than every element of π'' and every element of π'' is greater than every element of π'''. Since n is even number and maximal in π we have that a is an odd number, so $a+1$ is an even number. Therefore, by using Proposition 2.1 we get that p, q are even numbers, (π', a) is of odd length, and π'' is of even length. On the other hand, $q = p+1+|\pi''|$, so q is an odd number, a contradiction.

Hence, by induction on n we get the desired result. □

The second direction is to consider more than one additional restriction. For example, the following result is true.

Theorem 4.2. *Let $k \geq 2$. The generating function for the number of Dumont permutations of the first kind in $\mathfrak{S}_n$(1-3-2, 1-2-3$\cdots$-k, 2-1-3$\cdots$-k) is given by*

$$G_{k-1}(x) + xG_{k-2}(x).$$

A comparison of Theorem 4.2 with Theorem 2.8 suggests that there should exist a bijection between the sets $\mathfrak{S}_n$(1-3-2, 2-1-3-$\cdots$-k) and $\mathfrak{S}_n$(1-3-2, 1-2-3$\cdots$-k, 2-1-3-$\cdots$-k). However, we failed to produce such a bijection, and finding it remains an open question.

The third direction is to consider another 3-letter pattern instead of 1-3-2.

Theorem 4.3. *The number of Dumont permutation of the second kind in* $\mathfrak{S}_n(3\text{-}2\text{-}1)$ *is the same as the number of Dumont permutation of the first kind in* $\mathfrak{S}_n(2\text{-}3\text{-}1)$ *(or in* $\mathfrak{S}_n(3\text{-}1\text{-}2)$*) which is equal to* $C_{[n/2]}$.

Acknowledgments: The author are grateful to S. Kitaev for his careful reading of the manuscript.

References

[Bo1] M. Bóna, The permutation classes equinumerous to the smooth class, *Electron. J. Combin.* **5** (1998) #R31.

[BS] E. Babson and E. Steingrímsson, Generalized permutation patterns and a classification of the Mahonian statistics, *Séminaire Lotharingien de Combinatoire*, B44b:18pp, (2000).

[BCS] P. Brändén, A. Claesson, and E. Steingrímsson, Continued fractions and increasing subsequences in permutations, *Discr. Math.*, to appear.

[C] A. Claesson, Generalised pattern avoidance, *European Journal of Combinatorics*, **22** (2001) 961–973.

[CM] A. Claesson and T. Mansour, Permutations avoiding a pair of generalized patterns of length three with exactly one dash, preprint CO/0107044.

[CW] T. Chow and J. West, Forbidden subsequences and Chebyshev polynomials, *Discr. Math.* **204** (1999) 119–128.

[D] D. Dumont, Interpretations combinatoires des nombres de Genocchi, *Duke Math. J.* **41** (1974), 305–318.

[Ki] S. Kitaev, Multi-avoidance of generalised patterns, to appear in Discrete Mathematics.

[Km] D. Kremer, Permutations with forbidden subsequences and a generalized Schröder number, *Discr. Math.* **218** (2000) 121–130.

[Kr] C. Krattenthaler, Permutations with restricted patterns and Dyck paths, *Adv. in Applied Math.* **27** (2001), 510–530.

[M1] T. Mansour, Continued fractions and generalized patterns, *European Journal of Combinatorics*, **23:3** (2002), 329–344.

[M2] T. Mansour, Restricted 1-3-2 permutations and generalized patterns, *Annals of Combinatorics* **6** (2002), 1–12.

[M3] T. Mansour, Continued fractions, statistics, and generalized patterns, *Ars Combinatorica*, to appear (2002), preprint CO/0110040.

[MV1] T. Mansour and A. Vainshtein, Restricted permutations, continued fractions, and Chebyshev polynomials *Electron. J. Combin.* **7** (2000) #R17.

[MV2] T. Mansour and A. Vainshtein, Restricted 132-avoiding permutations, *Adv. Appl. Math.* **126** (2001), 258–269.

[MV3] T. Mansour and A. Vainshtein, Layered restrictions and Chebychev polynomials (2000), *Annals of Combinatorics*, *Annals of Combinatorics* **5** (2001), 451–458.

[MV4] T. Mansour and A. Vainshtein, Restricted permutations and Chebyshev polynomials, *Séminaire Lotharingien de Combinatoire* **47** (2002), Article B47c.

[R] A. Robertson, Permutations containing and avoiding 123 and 132 patterns, *Discrete Mathematics and Theoretical Computer Science*, **3** (1999) 151–154.

[RWZ] A. Robertson, H. Wilf, and D. Zeilberger, Permutation patterns and continuous fractions, *Electron. J. Combin.* **6** (1999) #R38.

[SS] R. Simion, F.W. Schmidt, Restricted Permutations, *Europ. J. of Combinatorics* **6** (1985), 383–406.

[SP] N.J.A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, New York (1995).

[W] J. West, Generating trees and forbidden subsequences, *Discr. Math.* **157** (1996), 363–372.

Moments, Narayana Numbers, and the Cut and Paste for Lattice Paths

R. A. Sulanke
Boise State University
Boise, ID, USA
January 28, 2002.

Abstract. Let $\mathcal{U}(n)$ denote the set of lattice paths that run from $(0,0)$ to $(n,0)$ with permitted steps $(1,1)$, $(1,-1)$, and perhaps a horizontal step. Let $\mathcal{E}(n+2)$ denote the set of paths in $\mathcal{U}(n+2)$ that run strictly above the horizontal axis except initially and finally. First we review *the cut and paste bijection* which relates points under paths of $\mathcal{E}(n+2)$ to points on paths of $\mathcal{U}(n)$. We apply it to obtain enumerations, some involving the Narayana distribution. We extend the bijection to a formula relating arbitrary factorial moments for the paths of $\mathcal{E}(n+2)$ to moments for the paths of $\mathcal{U}(n)$. This formula produces some additional results for moments and for the total area of the paths of $\mathcal{E}(n+2)$.

Key phases: lattice path moments, Catalan numbers, Narayana distribution, Schröder numbers, square-triangular numbers.

1 Introduction

Consider lattice paths in the integer plane represented as concatenations of the directed steps types: $U := (1,1)$, $D := (1,-1)$, and, perhaps, $H := (h,0)$ where h is a positive integer. When the steps are weighted, the weight of a path P, denoted by $|P|$, is the product of the weights of its steps. The weight of a path set $\mathcal{X}$, denoted $|\mathcal{X}|$, is the sum of the weights of its paths. For a given step set $\mathcal{S}$, let $\mathcal{U}(n)$ denote the set of all *unrestricted* paths running from $(0,0)$ to $(n,0)$. Let $\mathcal{C}(n)$ denote the set of paths in $\mathcal{U}(n)$ *constrained* never to pass beneath the horizontal axis. Let $\mathcal{E}(n)$ denote the set of paths in $\mathcal{C}(n)$ that are *elevated* strictly above the horizontal axis except at their initial and final points. E.g., for the unit-weighted steps of $\mathcal{S} = \{U, D\}$ and for $n \geq 0$, we have that $|\mathcal{U}(2n)|$ is the central binomial coefficient $\binom{2n}{n}$, $\mathcal{C}(2n)$ are the Dyck paths of length $2n$, and $|\mathcal{C}(2n)| = |\mathcal{E}(2n+2)|$ is a Catalan number $\frac{1}{n+1}\binom{2n}{n}$.

For any step set $\mathcal{S}$ and any path P running from $(0,0)$ to $(n,0)$, let

$$(0,p_0),(1,p_1),(2,p_2),\ldots,(x,p_x),\ldots,(n,p_n) \tag{1}$$

denote the lattice points traced by the path. (When P uses an $(h,0)$ step for $h \geq 2$, the trace points need not be at the ends of steps.) For any real valued function f defined on the

integers and for any path set $\mathcal{X}(n)$, $\mathcal{X}(n) \subseteq \mathcal{U}(n)$, the *weighted moment with respect to the formula* $f(y)$ is designated as

$$\mu(\mathcal{X}(n), f(y)) = \sum_{P \in \mathcal{X}(n)} |P| \sum_{x=0}^{n} f(p_x).$$

For $f(y) = 1/(n+1)$ and $\mathcal{X}(n) \subseteq \mathcal{U}(n)$, $\mu(\mathcal{X}(n), 1/(n+1)) = |X(n)|$. By the trapezoid rule, if $f(y) = y$, $\mu(\mathcal{E}(n), y)$ is the sum of the weighted areas of the regions bounded by the paths of $\mathcal{E}(n)$ and the horizontal axis. When we permit only the unit-weighted steps of $\mathcal{S} = \{U, D\}$, we will see later that $\mu(\mathcal{E}(2n+2), y) = 4^n$ for $n \geq 0$.

This paper is a direct continuation of the paper [6] which introduces *the cut and paste* bijective method relating lattice points under elevated paths of $\mathcal{E}(n+2)$ to points on the unrestricted paths of $\mathcal{U}(n)$. In [6] the method yielded results about zeroth, first, and second moments for $\mathcal{E}(n+2)$, in particular

$$\mu(\mathcal{E}(n+2), y^2) = \mu(\mathcal{U}(n), 1) = (n+1)|\mathcal{U}(n)|. \tag{2}$$

Section 2 will review the cut and paste method. Section 3 will give some illustrations of the method obtained by restricting its domain and codomain. *In particular, the cut and paste delivers the Narayana numbers.* Section 4 will extend the method to a result that relates factorial moments for $\mathcal{E}(n+2)$ to factorial moments for $\mathcal{U}(n)$. The paper concludes with additional means for handling moments and with consequences of Section 4 including results related to the Schröder, central Delannoy, and square-triangular numbers.

Some notation and background. In this paper n will denote an arbitrary nonnegative integer. Usually, U, D, and H $(= (h, 0))$ will denote unit-weighted steps, while U_t and H_s will denote steps weighted by the indeterminates t and s. For notational brevity, we will allow h to be either a positive integer or '∞'. Effectively, $\{U, D, (\infty, 0)\} = \{U, D\}$, and the power z^∞ will make no contribution to any power series.

For any $\mathcal{S} = \{U_t, D, H_s\}$, consider the following generating functions: $c(z) := \sum_{n\geq 0} |\mathcal{C}(n)|z^n$, $e(z) := tz^2c(z) = \sum_{n\geq 0} |\mathcal{E}(n+2)|z^{n+2}$, and $u(z) := \sum_{n\geq 0} |\mathcal{U}(n)|z^n$.

From the known decompositions of paths sets we have,

$$\begin{aligned} c(z) &= 1 + sz^h c(z) + tz^2 c(z)^2 & (3)\\ u(z) &= 1 + sz^h u(z) + 2tz^2 c(z)u(z) & (4) \end{aligned}$$

To see (4) note that every path in $\mathcal{U}(n)$ either has zero length, begins with H, or begins with U or D followed by a constrained path or its reflection and later returns to the horizontal axis for the first time. Identity (3) follows in a similar manner. Solving these yields

$$\begin{aligned} e(z) &= tz^2c(z) = (1 - sz^h - \sqrt{(1-sz^h)^2 - 4tz^2})/2,\\ u(z) &= 1/\sqrt{(1-sz^h)^2 - 4tz^2}. & (5) \end{aligned}$$

We remark that Example 6.3.8 of [9] extends easily to an alternative derivation of (5).

Recall the rising factorial, $z^{\overline{k}}$, defined so $z^{\overline{k}} = z(z+1)\cdots(z+k-1)$ for positive integer k, $z^{\overline{0}} = 1$, and $z^{\overline{k}} = 0$ for negative integer k. Then, for $k \geq 0$, $\binom{r+k}{k} = (r+1)^{\overline{k}}/k!$. For any statement A, we define its truth value by $\chi(A)$ so that $\chi(A) = 1$ if A is true, and $= 0$ otherwise.

2 The cut and paste method

Here we define *the cut and paste method*, which was presented with more detail, including its invertibility, in [6]. Let $\mathcal{S} = \{U, D, H\}$. First we need the notion of a *dot*. Given a path $P \in \mathcal{E}(n+2)$, given a lattice point (x, y) lying strictly under P but weakly above the horizontal axis, and given an integer k, a *dot* is a triple, $[P, (x, y), k]$. The index k permits the existence of more than one distinguishable dot at some points. With the notation of (1), the domain for our proposed bijection is

$$\text{DOTS}(n+2) := \{[P, (x, y), k] : P \in \mathcal{E}(n+2), 0 < x < n+2, 0 \leq y < p_x, -p_x + y < k < p_x - y\}.$$

This domain can be partitioned into triangular arrays of dots, with one array corresponding to each lattice point on the trace of each elevated path in $\mathcal{E}(n+2)$. Thus there will be $(n+1)|\mathcal{E}(n+2)|$ triangular arrays. E.g., if $P = UUDUUDDD$ and if $x = 5$, then $p_5 = 3$ and the corresponding array appears as

$$\begin{array}{ccccccc} & & & [P,(5,2),0] & & & \\ & & [P,(5,1),-1] & [P,(5,1),0] & [P,(5,1),1] & & \\ [P,(5,0),-2] & & [P,(5,0),-1] & [P,(5,0),0] & [P,(5,0),1] & & [P,(5,0),2] \end{array}$$

The codomain for the proposed bijection is a set of pointed paths, each path being pointed, i.e. marked, by a distinguished lattice point on its trace. Hence the codomain is

$$\text{POINTS}(n) := \{(P, (x, p_x)) : P \in \mathcal{U}(n), 0 \leq x \leq n\}.$$

We now define

$$\phi : \text{DOTS}(n+2) \to \text{POINTS}(n) \tag{6}$$

First assume $k \geq 0$. Each $[P, (x, y), k] \in \text{DOTS}(n+2)$ determines four points on P (See Fig. 1):

– Let θ be the point on P directly above (x, y); i.e., $\theta := (x, p_x)$.

– Let $\epsilon = (\epsilon_1, \epsilon_2)$ be the nearest point on P to the left of (x, y) such that $\epsilon_2 = p_x - k - 1$. (This indicates the role k plays in defining the bijection.)

– Let $\lambda = (\lambda_1, \lambda_2)$ be the nearest point on P to the left of (x, y) such that $\lambda_2 = y$.

– Let $\rho = (\rho_1, \rho_2)$ be the nearest point on P to right of (x, y) such that $\rho_2 = y$.

Let L_1 be that subpath of P running from $(0, 0)$ to λ; let L_2 be that subpath of P running from λ to ϵ; let R_1 be that subpath of P running from ϵ to ρ; let R_2 be that subpath of P running from ρ to $(n+2, 0)$. Some of these subpaths may be empty. Here $P = L_1 L_2 R_1 R_2$.

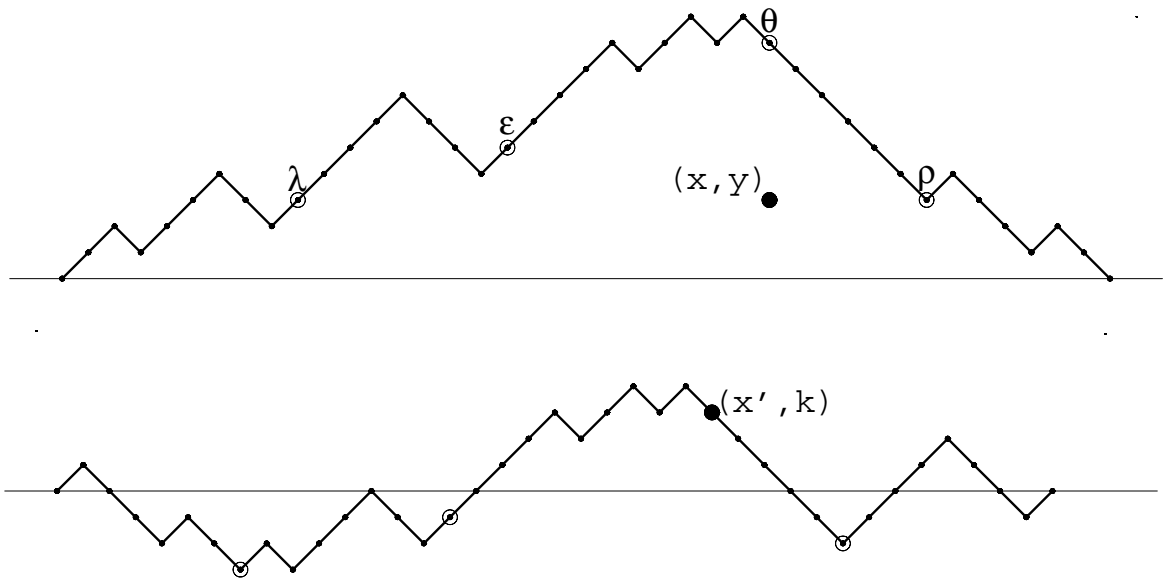


Figure 1: For $k = 3$, the dot $[P, (27, 3), 3]$ in DOTS(40) and its image $[P', (25, 3)]$ in POINTS(38).

Let $\overline{L_1R_1}$ be the path obtained from the concatenation L_1R_1 by deleting its first and last steps. When $y = 0$, L_1 has zero length, and we will use $\overline{\overline{R_1}}$ to denote $\overline{L_1R_1}$. Define

$$\phi([L_1L_2R_1R_2, (x, y), k]) = (R_2\overline{L_1R_1}L_2, (x', k)) \tag{7}$$

where the point θ is moved along with $\overline{R_1}$ so that $x' = x + n + \lambda_1 - \epsilon_1 - \rho_1 - 1$.

If $k < 0$, let REFL(P') denote the reflection of the path P' about the x-axis and define

$$\phi([P, (x, y), k]) = (\text{REFL}(P'), (x', k))$$

where $\phi([P, (x, y), |k|]) = (P', (x', |k|))$.

3 Examples of the cut and paste

Here we illustrate the cut and paste method by restricting its domain and codomain to prove some known results, mainly concerning the Narayana distribution and the large Schröder numbers. Other such examples appear in [6].

3.1 Cardinality results analogous to the cycle lemma

For $\mathcal{S} = \{U_t, D, H_s\}$ let $\mathcal{E}(n, m)$ and $\mathcal{U}(n, m)$ denote those subsets of $\mathcal{E}(n)$ and $\mathcal{U}(n)$, respectively, where each path has m U-steps.

Proposition 1 *For $m \geq 0$,*

$$(m + 1)|\mathcal{E}(n + 2, m + 1)| = t|\mathcal{U}(n, m)|.$$

When $\mathcal{S} = \{U, D\}$, this formula reduces to $(n+1)|\mathcal{E}(2n+2)| = |\mathcal{U}(2n)|$, and thus the cut and paste explains the factor $(n+1)$. The paper [6] compares this explanation with that given by the classical cycle lemma of [4].

To obtain the proposition, place $m+1$ dots on the x-axis under each path P in $\mathcal{E}(n+2, m+1)$ so that exactly one dot is located directly below the final point of each U step. More specifically, apply the restricted bijection

$$\phi : \{[P, (x,0), 0] : P \in \mathcal{E}(n+2, m+1) \text{ and } x \text{ below a final point of a } U \text{ step}\}$$

$$\rightarrow \{[P', (0,0)] : P' \in \mathcal{E}(n, m)\}.$$

We see that the weight of the domain is $(m+1)|\mathcal{E}(n+2, m+1)|$ while the weight of the codomain is $t|\mathcal{U}(n, m)|$, where the factor t corresponds to the deletion of a U in the cut and paste. $\square$

We define the elevated large Schröder paths to be the paths in $\overline{\mathcal{E}}(n+2)$ having $\overline{\mathcal{S}} = \{U, D, (2,0)\}$. (Here and in the following, when we embellish '$\mathcal{S}$', we embellish '$\mathcal{E}$' and '$\mathcal{U}$' correspondingly.) We thus take the large Schröder numbers to be defined in terms of the cardinality of these path sets. Specifically, $(|\overline{\mathcal{E}}(2n+2)|)_{n\geq 0} = (1, 2, 6, 22, 90, 394, ...)$. (Sequence A006318 of [8].)

The proposition shows that large Schröder numbers can be formulated as

$$|\overline{\mathcal{E}}(2n+2)| = \sum_{m\geq 0} |\overline{\mathcal{E}}(2n+2, m+1)| =$$

$$\sum_{m\geq 0} \frac{1}{(m+1)} |\overline{\mathcal{U}}(2n, m)| = \sum_{m\geq 0} \frac{(m+n)!}{(m+1)!\ m!\ (n-m)!}. \tag{8}$$

3.2 The Narayana distribution in terms of oddly positioned up steps

Consider the step set $\widetilde{\mathcal{S}} = \{U_t, U, D\}$ where U_t is a step of weight t that must be oddly positioned on any path of $\widetilde{\mathcal{E}}(2n+2)$, U is a unit-weighted step that must be evenly positioned on any path of $\widetilde{\mathcal{E}}(2n+2)$, and D is unit weighted.

Proposition 2 *For $n \geq 1$,*

$$|\widetilde{\mathcal{E}}(2n+2)| = \sum_{i=1}^{n} \frac{1}{i} \binom{n-1}{i-1} \binom{n}{i-1} t^i = \sum_{i=1}^{n} \frac{1}{n} \binom{n}{i} \binom{n}{i-1} t^i, \tag{9}$$

where $\frac{1}{n}\binom{n}{i}\binom{n}{i-1}$ is a Narayana number. (Sequence A001263 in [8].)

For $t = 1$, $(|\widetilde{\mathcal{E}}(2n+2)|)_{n\geq 1} = (1, 2, 5, 14, 42, ...)$, the Catalan numbers less the first term. For $t = 2$, $(|\widetilde{\mathcal{E}}(2n+2)|)_{n\geq 1} = (2, 6, 22, 90, 394, ...)$, the large Schröder numbers less the first term.

For this proof, use the unit weighted steps, U and H. For any path Q, let $W_o(Q)$ ($W_e(Q)$, respectively) denote the number of oddly (evenly, respectively) positioned U steps on Q, translated if necessary to begin at $(0,0)$. Let $\mathcal{E}(2n+2,i)$ denote the subset of $\mathcal{E}(2n+2)$ whose paths have i oddly positioned steps. Let

$$\begin{aligned} A &:= \{[P,(x,0),0] : P \in \mathcal{E}(2n+2,i) \text{ and } x \text{ is a final abscissa of an oddly positioned } U\},\\ B &:= \{(P,(0,0)) : P \in \mathcal{U}(2n) \text{ and the lowest point of } P \text{ has an even ordinate}\},\\ C &:= \{P'' : P'' \text{ runs from } (0,0) \text{ to } (2n-1,1) \text{ and } W_e(P'') = i-1\}. \end{aligned}$$

Restricting the cut and paste establishes the bijection $\phi : A \to B$. However, since this map does not preserve the number of oddly positioned U steps, we need a bijection $\nu : B \to C$ correcting this situation for which the composition $\nu \circ \phi$ is weight preserving.

For $y = 0$, P factors as L_2R_1 and its image under ϕ factors as $\overline{\overline{R_1}}\ L_2$. We now define $\nu(\overline{\overline{R_1}}\ L_2)$. Immediately, $W_e(\overline{\overline{R_1}}) = W_o(R_1) - 1$. If L_2 has zero length then $W_o(L_2) = W_e(L_2) = 0$. If L_2 has positive length, write $\overline{\overline{R_1}}\ L_2$ as a sequence of steps: $\overline{\overline{R_1}}\ L_2 = r_1r_2\cdots r_{2n-x+1}\ell_1\ell_2\cdots\ell_{x-1}$. The step ℓ_1 begins with a negative even ordinate, namely, $1-p_x$. Let ℓ_j denote the last step on $\overline{\overline{R_1}}\ L_2$ that begins with ordinate equal -1. Put

$$\overline{\overline{R_1}}\ L_2' := r_1r_2\cdots r_{2n-x+1}\ell_j\ell_{j+1}\cdots\ell_{x-1}\ell_1\ell_2\cdots\ell_{j-1}.$$

Since on $\overline{\overline{R_1}}\ L_2'$ the relocated step ℓ_1 is the last step to begin with ordinate $2-p_x$, L_2 is recoverable from L_2'. Moreover, $W_e(L_2') = W_o(L_2)$.

If L_2 has zero length, let P' denote $\overline{\overline{R_1}}$; otherwise, let P' denote $\overline{\overline{R_1}}\ L_2'$ and notice that we have $W_e(P') = W_o(P) - 1$. If P' terminates with a D step, let P'' denote the path obtained from P' when this final D is deleted. For the sake of recovering P' notice that the lowest point on P'' has even ordinate. On the other hand, if P' terminates with a U step, let P'' denote the path obtained when this (evenly positioned) last step is removed and the leftmost lowest D step is changed into an (evenly positioned) U step. Here, for the sake of recovering P', notice that the lowest point on P'' is unique and has an odd ordinate. In either case, P'' terminates at $(2n-1,1)$ and $W_e(P'') = W_e(P') = W_o(P) - 1 = i - 1$. Thus we define $\nu(\overline{\overline{R_1}}\ L_2) = P''$.

To determine $|C|$, observe that each path must have $i-1$ of its $n-1$ even step positions filled with a U step and must have $n-(i-1)$ of its n odd step positions filled with a U step. Hence $|C| = \binom{n-1}{i-1}\binom{n}{n-i+1} = \frac{i}{n}\binom{n}{i}\binom{n}{i-1}$, while $|A| = i|\mathcal{E}(2n+2,i)|$. $\square$

3.3 The Narayana distribution in terms of peaks

For this application we count elevated paths with respect to the number of bicolored peaks. Again we derive the Narayana distribution and the large Schröder numbers. On any path, a 'right-hand turn' or a 'peak' is the intermediate vertex of a consecutive UD pair. For $\widehat{\mathcal{S}} = \{U, D\}$, let $\widehat{\mathcal{E}}(n,b,r)$ denote the set of elevated paths using the steps U and D and having b blue peaks and r red peaks.

Proposition 3 *For* $1 \le b+r \le n$,

$$|\widehat{\mathcal{E}}(2n+2,b,r)| = \frac{1}{b+r}\binom{n-1}{b+r-1}\binom{n}{b+r-1}\binom{b+r}{b}.$$

For each path P in $\widehat{\mathcal{E}}(2n+2,0,i)$, place i dots on the x-axis below the peaks (all being red) of the path. With $k=0$ each dot is mapped by ϕ to a point $\phi([P,(x,0),0]) = [\overline{\overline{R_1}L_2},(0,0)]$ where the image path begins with a D step and has $i-1$ right-hand turns. If we tilt each image path counterclockwise by 45 degrees, one can check that in the tilted path there would be $\binom{n-1}{i-1}$ ways to choose the abscissae and $\binom{n}{i-1}$ ways to choose the ordinates for the intermediate vertices of the right-hand turns, where these turns uniquely determine the path. Thus, $i|\widehat{\mathcal{E}}(2n+2,0,i)| = \binom{n-1}{i-1}\binom{n}{i-1}$. Now, allowing b of the peaks to be independently colored blue, while the remainder are red, yields the factor $\binom{b+r}{b}$. □

When we disallow blue peaks, the proposition shows that $\mathcal{E}(2n+2,0,r)$ is a Narayana number. When we do not limit the coloring or the number of peaks, we see that the number of paths in the $\widehat{\mathcal{E}}(2n+2)$ with independently bicolored peaks is the large Schröder number:

$$\begin{aligned}\sum_b\sum_r |\widehat{\mathcal{E}}(2n+2,b,r)| &= \sum_b\sum_i |\widehat{\mathcal{E}}(2n+2,b,i-b)| = \\ \sum_i\sum_b |\widehat{\mathcal{E}}(2n+2,b,i-b)| &= \sum_{i=1}^{n}\frac{1}{i}\binom{n-1}{i-1}\binom{n}{i-1}2^i \qquad (10)\end{aligned}$$

Now return to the large Schröder paths, considered in (8), which used $\overline{\mathcal{S}} = \{U,D,(2,0)\}$. In that notation, $\overline{\mathcal{E}}(2n,n+1-j)$ will be the set of elevated paths having j of the $(2,0)$ steps. There is a simple matching between $\cup_r\widehat{\mathcal{E}}(2n+2,b,r)$ and $\overline{\mathcal{E}}(2n+2,n+1-b)$ that is obtained by transforming each UD pair with a blue intermediate vertex into a $(2,0)$ step and by removing the color red. Hence the number of paths in $\overline{\mathcal{E}}(2n+2)$ is also counted by large Schröder numbers of (10).

3.4 The 'area' under peaks

We will consider the total area under the paths of $\mathcal{E}(n+2)$ more extensively in Section 6. Here, for $\mathcal{S} = \{U_t, D, H_s\}$, we will sum the heights of peaks over all of the constrained paths in $\mathcal{C}(n)$. Equivalently, by the manner in which dots are arrayed under each trace point of the paths of $\mathcal{E}(n+2)$, we will find the weighted cardinality of the dots with $k=1$ under the peaks of the paths of $\mathcal{E}(n+2)$. By the cut and paste one can check that the restricted bijection is

$$\begin{aligned}\phi : \{[P,(x,y),1] \in \mathrm{DOTS}(n+2) : (x,p_x) \text{ is a peak }\} \\ \to \{(P,(x,1)) \in \mathrm{POINTS}(n) : p_{x-1} = p_{x+1} = 0\}.\end{aligned}$$

Here, each dot in the restricted domain is mapped to an unrestricted path with a marked UD with intermediate vertex having ordinate 1. Each marked path results from the concatenation of three paths, namely, an unrestricted path from $(0,0)$ to the marked UD followed by an unrestricted path to $(n,0)$. Hence, with tz^2 corresponding to the marked UD, we have

Proposition 4

$$\sum_{n\geq 0}\sum_{P\in\mathcal{C}(n)}\sum_{0<x<n}\chi((x,p_x)\ \textit{is a peak})p_x z^n = tz^2u(z)^2.$$

Consequently, for $\mathcal{S}=\{U,D\}$, the power series for the sum of heights of the peaks on constrained paths is $tz^2u(z)^2 = z^2(1-4z^2)^{-1}$, whose coefficients are powers of 4.

4 Relating moments for $\mathcal{E}(n+2)$ to those for $\mathcal{U}(n)$

To obtain our principal consequence of the the cut and paste bijection, we assign a value to each dot and its image. Consider any real valued function, ρ, defined on $\mathbb{Z}\times\mathbb{Z}$. With r viewed as an index, the cut and paste yields trivially

$$\sum_{[P,(x,y),k]\in \mathrm{DOTS}(n+2)}\rho(k,r) = \sum_{(P,(x,k))\in \mathrm{POINTS}(n)}\rho(k,r) = \sum_{(P,(x,y))\in \mathrm{POINTS}(n)}\rho(y,r). \quad (11)$$

We call a formula $\rho(y,r)$ a 'b-moment' if, for arbitrary $(y,r)\in\mathbb{Z}\times\mathbb{Z}$,

$$\rho(y,r+1) = \chi(y\geq 0)\sum_{0\leq j\leq y}\rho(j,r). \quad (12)$$

We use the name, 'b-moment', since a b-moment is easily seen to be a linear combination of binomial coefficients. Thus, if ρ is a b-moment, the left side of (11) becomes

$$\begin{aligned}\sum_P\sum_x\sum_{y=0}^{p_x-1}\sum_{k=0}^{p_x-y-1}\rho(k,r) &= \sum_P\sum_x\sum_{y=0}^{p_x-1}\rho(p_x-y-1,r+1)\\ &= \sum_P\sum_x\rho(p_x-1,r+2).\end{aligned}$$

This identity and (11), which is a consequence of the cut and paste, yields

Proposition 5 *For $\mathcal{S}=\{U_t,D,H_s\}$, for integer r, and for any b-moment $\rho(y,r)$,*

$$\mu(\mathcal{E}(n+2),\rho(y-1,r+2)) = t\mu(\mathcal{U}(n),\rho(y,r)). \quad (13)$$

Specifically,

$$\mu(\mathcal{E}(n+2),\tbinom{y-1+r+2}{y-1}) = t\mu(\mathcal{U}(n),\tbinom{y+r}{y}) \quad (14)$$

and for $r\geq 0$,

$$\frac{\mu(\mathcal{E}(n+2),y^{\overline{r+2}})}{(r+2)!} = \frac{t\mu(\mathcal{U}(n),y^{\overline{r}})}{r!}. \quad (15)$$

As the classic example, we use (14) to prove (2):

$$\mu(\mathcal{E}(n+2), y^2) = \mu(\mathcal{E}(n+2), 2\tbinom{y+1}{y-1} - \tbinom{y}{y-1})$$

$$= \mu(\mathcal{U}(n), 2\tbinom{y}{y} - \tbinom{y-1}{y}) = \mu(\mathcal{U}(n), 2\chi(y \geq 0) - \chi(y = 0)) = \mu(\mathcal{U}(n), 1),$$

where the last identity holds since over the paths of $\mathcal{U}(n)$ there are the same number of points with positive ordinate as with negative ordinate.

5 Recurrences of moment generating functions

We will recast a recurrence for factorial moments for elevated paths like one given by Chapman [3]. (See also [7] and [12]). We will then use Proposition 5 to convert that recurrence into one for moments for unrestricted paths. For a given b-moment $\rho(y, r)$ and for $\mathcal{S} = \{U_t, D, H_s\}$, let

$$e_r(z) := \sum_{n \geq 0} \mu(\mathcal{E}(n+2), \rho(y-1, r))z^{n+2} \quad \text{and} \quad u_r(z) := \sum_{n \geq 0} \mu(\mathcal{U}(n), \rho(y, r))z^n.$$

Proposition 6 *For integer r,*

$$(1 - tz^2c(z)^2)e_r(z) = e_{r-1}(z) \tag{16}$$

$$(1 - tz^2c(z)^2)u_r(z) = u_{r-1}(z) \tag{17}$$

$$e_r(z) = \frac{1}{2}(1 + (1 - sz^h)u(z))e_{r-1}(z) \tag{18}$$

We prove (16) in the form

$$e_r(z) = e_{r-1}(z) + tz^2c(z)^2e_r(z). \tag{19}$$

Since ρ is a b-moment we have $\rho(i, r) = \rho(i, r-1) + \rho(i-1, r)$ and hence

$$\sum_{P \in \mathcal{E}(n+2)} \sum_{x=1}^{n+1} \rho(p_x - 1, r) = \sum_{P \in \mathcal{E}(n+2)} \sum_{x=1}^{n+1} \rho(p_x - 1, r-1) + \sum_{P \in \mathcal{E}(n+2)} \sum_{x=1}^{n+1} \rho(p_x - 2, r). \tag{20}$$

Consider the rightmost double sum. Each $P \in \mathcal{E}(n+2)$, satisfies $P = UQD$ where $Q \in \mathcal{C}(n)$ and Q can be factored uniquely so that each factor, Q', is a translation of some elevated path. Suppose Q' can be translated to belong to $\mathcal{E}(n')$, for $2 \leq n' \leq n$. Each time Q' appears as a factor in some of the concatenations forming the paths of $\mathcal{C}(n)$, it is preceded by a (perhaps void) constrained path and followed by a (perhaps void) constrained path; thus Q' makes $\sum_{i+i'=n-n'} |\mathcal{C}(i)||\mathcal{C}(i')|$ appearances in the paths of $\mathcal{C}(n)$. Since each factor Q' begins and ends with points of ordinate 1, i.e., since each Q' is 'doubly elevated', the

moment contribution to $\sum_{P\in\mathcal{E}(n+2)}\sum_{x=1}^{n+1}\rho(p_x-2,r)$ of Q', translated to begin at $(0,0)$, is $\sum_{x=1}^{n'-1}\rho(q'_x-1,r)$ times its frequency of appearances. Hence,

$$\sum_{P\in\mathcal{E}(n+2)}\sum_{x=1}^{n+1}\rho(p_x-2,r)=\sum_{n'=2}^{n}\sum_{Q\in\mathcal{E}(n')}\sum_{i+i'=n-n'}|\mathcal{C}(i)||\mathcal{C}(i')|\sum_{x=1}^{n'-1}\rho(q'_x-1,r) \tag{21}$$

and the corresponding generating function is $tz^2c(z)^2e_r(z)$. The identities (20) and (21) yield (19). An application of Proposition 5 to (16) yields (17). A straightforward computation yields (18). □

6 Further examples

6.1 Areas and intercepts

Take $\mathcal{S}=\{U,D,H\}$ and let $\rho(y,r)=\binom{r+y}{y}$. Since $\rho(y-1,1)=y$, the left side of (13) becomes the first moment $\sum_{P\in\mathcal{E}(n+2)}\sum_x p_x$, which by the trapezoid rule is the total area bounded between the horizontal axis and the paths of $\mathcal{E}(n+2)$. Since $\rho(y,-1)=\chi(y=0)$, the right side of (13) becomes $\sum_{P\in\mathcal{U}(n)}\sum_x\chi(p_x=0)$ which is the total number of intercepts of the horizontal axis by the paths of $\mathcal{U}(n)$. More generally, if we consider $\rho(y,r)=\binom{r+y-y_0}{y-y_0}$ for nonnegative y_0, then we can use (13) to show

Proposition 7 *For $\mathcal{S}=\{U,D,H\}$ and for $y_0\geq 0$, the total area of the regions under the paths of $\mathcal{E}(n+2)$ and above the horizontal line $y=y_0$ is equal to the number of intercepts of that line by the paths of $\mathcal{U}(n)$.*

Next we give further results concerning intercepts, area, and the generating function $u(z)^2$, whose formula is obtained from (5).

Proposition 8 *For $\mathcal{S}=\{U,D,H\}$, the generating function for the number of intercepts of the horizontal axis by the step end points on the traces of the paths of $\mathcal{U}(n)$ satisfies*

$$\sum_{n\geq 0}\sum_{P\in\mathcal{U}(n)}\sum_{0\leq x\leq n}\chi(\ (x,0)\ \textit{is a step end point on}\ P\ )z^n=u(z)^2.$$

This proposition follows by observing that each intercept contributing to the inner summations results from the concatenation of two paths, an unrestricted path from $(0,0)$ to the intercept followed by an unrestricted path from the intercept to $(n,0)$. □

Proposition 9 *For $\mathcal{S}=\{U,D,H\}$, the generating function for the number of intercepts of the horizontal axis by lattice points on the traces of paths of $\mathcal{U}(n)$ satisfies*

$$\sum_{n\geq 0}\mu(\mathcal{U}(n),\chi(y=0))z^n=(1+(h-1)z^h)u(z)^2. \tag{22}$$

Equivalently, the generating function for the total area under the paths of $\mathcal{E}(n+2)$ satisfies

$$\sum_{n\geq 0}\mu(\mathcal{E}(n+2),y)z^{n+2}=z^2(1+(h-1)z^h)u(z)^2.$$

Intercepts which are end points of steps make a contribution to the generating function as in Proposition 8. When a step lies on the horizontal axis, the $h-1$ intercepts which are interior to a step can be collapsed along with the step to become the intercept of a step end point on a path of belonging to $\mathcal{U}(n-h)$ and thus make an adjusted contribution to the right side of (22). The second identity follows by the initial remarks of this section. $\square$

6.2 Two comparable models Here we examine some comparable, yet different, models whose area results are derivable from the the cut and paste, through Proposition 9. In the two cases below we will consider instances of a recurrence: If $a(n)$ is defined so that $\sum_n a(n)z^n = u(z)^2$, then

$$a(n) = 4ta(n-2) + 2sa(n-h) - s^2a(n-2h) \tag{23}$$

which follows the comparison of coefficients and (5).

Case for $s = t+1$ and $h = 1$: Here the step set is $\mathcal{S}^* = \{U_t, D, (0,1)_{t+1}\}$. For $t = 1$, $e^*(z) = 1z^2 + 2z^3 + 5z^4 + 14z^5 + 42z^6...$, a generating function for the Catalan numbers less the first term. For $t = 2$, $e^*(z) = 2z^2 + 6z^3 + 22^4 + 90z^5 + 394z^6...$, a generating function for the large Schröder numbers less its first term; it is not a curiosity that the cardinalities of $\mathcal{E}^*(n+2)$ and $\widetilde{\mathcal{E}}(2n+2)$ (see (9)) agree as one can establish a straightforward isomorphism $\alpha : \mathcal{U}^*(n) \to \widetilde{\mathcal{U}}(2n)$ using the step replacement rules: $\alpha(U_t) = U_tU$, $\alpha(D) = DD$, and $\alpha(H_{t+1}) = \{U_tD, DU\}$.

For $t = 1$ and $t = 2$, $u^*(z)$ is the generating function for the central binomial coefficients and the central Delannoy numbers, respectively. (See Sequence A001850 of [8] and Section 6.5 of [9].)

The weighted area under the elevated paths of $\mathcal{E}^*(n+2)$, satisfies

$$\sum_{n\geq 0} \mu(\mathcal{E}^*(n+2), y)z^{n+2} = tz^2\mathcal{U}^*(z)^2 = \frac{tz^2}{((t+1)z-1)^2 - 4tz^2}$$

For $t = 1$, $z^2u^*(z) = \frac{z^2}{1-4z}$, a generating function whose coefficients are powers of 4. We remark that there is a simple bijection from $\mathcal{E}^*(n+2)$ to a set of parallogram polyominoes of perimeter $2n+4$ which is area preserving as discussed in [11]. Further attention to this result appears in [5].

For $t = 2$, $2z^2u^*(z)^2 = \frac{2z^2}{1-6z+z^2} = 2z^2 + 12z^3 + 70z^4 + 408z^5...$, where the coefficients are double the square-triangular numbers, or, equivalently, every other Pell number. (See sequences A000129, A001109, and A001542 in [8].) By (23), for $t = 2$, $u^*(z)^2 = \sum_n a^*(n)z^n$ satisfies

$$a^*(n) = 6a^*(n-1) - a^*(n-2) \tag{24}$$

subject to $a^*(2) = 2$ and $a^*(3) = 12$. This recurrence in terms of the total the areas of zebras (i.e., column-bicolored parallogram polyominoes) was a principal topic of [11].

Case for $t = 1$: Here $\overline{\mathcal{S}} = \{U, D, (0,h)_s\}$. For $h = \infty$, $\overline{e}(z)$ is a generating function for the Catalan numbers. For $s = 1$ and $h = 1$, $\overline{e}(z)$ is the generating function for the Motzkin

numbers. For $s=1$ and $h=2$, $\overline{\mathcal{C}}(n)$ are the usual large Schröder paths with $\overline{e}(z)$ being the generating function for the large Schröder numbers, as noted after formula (10).

For $h=\infty$, $\overline{u}(z)$ gives the central binomial coefficients. For $s=h=1$, $\overline{u}(z)$ gives the central trinomial numbers. For $s=1$ and $h=2$, $\overline{u}(z)$ gives the central Delannoy numbers.

By Proposition 9 the area under the elevated paths of $\overline{\mathcal{E}}(n+2)$, satisfies

$$\sum_{n\geq 0}\mu(\overline{\mathcal{E}}(n+2),y)z^{n+2}=\frac{1+(h-1)z^h}{(sz^h-1)^2-4z^2}.$$

For $h=\infty$, the coefficients of this power series are powers of 4. (See [13].) For $s=1$ and $h=2$, this power series becomes $\frac{z^2(1+z^2)}{1-6z^2+z^4}=1z^2+7z^4+41z^6+239z^8+1393z^{10}...$, where the coefficients correspond to pairwise sums of consecutive square-triangular numbers, or equivalently to every other pairwise sum of consecutive Pell numbers, as noted in [2]. (See sequence A002315 in [8].) For $\overline{\mathcal{E}}(2n+2)$, we remark that the square-triangular numbers give both the sums of the ordinates of the trace points restricted to be end points of steps and the sums of the ordinates of the trace points which are the mid points of steps. By (23), for $s=1$ and $h=2$, $\overline{u}(z)^2=\sum_n \overline{a}(n)z^n$ satisfies

$$\overline{a}(n)=6\overline{a}(n-2)-\overline{a}(n-4) \tag{25}$$

subject to $\overline{a}(2)=1$ and $\overline{a}(4)=7$, as noted in [2]. Compare this recurrence to (24). For recent considerations of (25) and other references, see [1]. For a bijective approach to the recurrences for the cardinality, area, and the second moments for large Schröder paths see [10].

6.3 Moments for unrestricted paths about the horizontal axis

Consider the rising factorial moments of the distances from the horizontal axis for trace points of the paths of $\mathcal{U}(n)$. While our computations are not consequences of the cut and paste, we include them since they are analogous to the proofs of Propositions 6 and 8.

Proposition 10 *For $m>0$,*

$$\sum_{n\geq 0}\sum_{P\in\mathcal{U}(n)}\sum_{0\leq x\leq n}|p_x|^{\overline{m}}z^n=2u(z)^2\sum_{n\geq 0}\mu(\mathcal{E}(n+2),y^{\overline{m}})z^n.$$

To prove this, note that since $|p_x|^{\overline{m}}>0$ there is no contribution to the left side from the intercepts. Also note that each path in $\mathcal{U}(n)$ can be factored into subpaths which begin and end on the horizontal axis, each factor being either an elevated path P' whose translation belongs to $\mathcal{E}(n'+2)$ or its reflection. Since the factor P', or its reflection (requiring the multiple 2), is preceded and followed by a unrestricted path and since $\sum_{1\leq x\leq n'+1}|p'_x|^{\overline{m}}$ (when P' is translated to begin at the origin) is a summand of $\mu(\mathcal{E}(n'+2),y^{\overline{m}})$, the proposition now follows in a manner of the proof of (16). $\square$.

6.4 Two Catalan configurations

Here we give two configurations enumerated by the Catalan numbers but not appearing in the catalog of Exercise 6.19 of [9].

Proposition 11 *For $\mathcal{S} = \{U, D\}$ and for $n \geq 0$, the total number of intercepts of the horizontal axis by the Dyck paths running from $(0,0)$ to $(2n,0)$ is $\frac{1}{n+2}\binom{2n+2}{n+1}$. Moreover, if the values 1, -2, and 1, are assigned respectively to the points of ordinate 0, 1, and 2 on the unrestricted paths running from $(0,0)$ to $(2n,0)$, then the sum of these values over all these paths is the same Catalan number.*

More generally, for $\mathcal{S} = \{U_t, D, H_s\}$, we claim that

$$\sum_{n\geq 0} \mu(\mathcal{U}(n), (-1)^y \tbinom{2}{y}) z^n = \sum_{n\geq 0} \mu(\mathcal{U}(n), \tbinom{y-3}{y}) z^n =$$

$$\sum_{n\geq 0} t^{-1}\mu(\mathcal{E}(n+2), \tbinom{y-2}{y-1}) z^n = \sum_{n\geq 0} \mu(\mathcal{C}(n), \chi(y=0)) z^n = c(z)^2.$$

To see the last identity, notice that $\mu(\mathcal{C}(n), \chi(y=0))$ is the weight of the set of intercept-marked paths form from constrained paths of $C(n)$. Since each intercept is realized as the concatenation of a constrained path from $(0,0)$ to the intercept with a constrained path from the intercept to $(n,0)$, $\mu(\mathcal{C}(n), \chi(y=0)) = \sum_i |C(i)||C(n-i|$. To finish the proof of the proposition, notice that $z^2c(z)^2 = c(z) - 1$ for $\mathcal{S} = \{U, D\}$. $\square$

Acknowledgement. The author thanks Elisa Pergola for her contribution to the proof of Proposition 2.

References

[1] E. Barcucci, E. and S. Rinaldi, Some linear recurrences and their combinatorial interpretation by means of regular languages. *Theoret. Comput. Sci.* 255 (2001), no. 1-2, 679-686.

[2] J. Bonin, L. Shapiro, and R. Simion, Some q-analogues of the Schröder numbers arising from combinatorial statistics on lattice paths, *J. Statistical Planning and Inference* 34 (1993) 35-55.

[3] R. Chapman, Moments of Dyck paths, *Disc. Math.*, 204 (1999) 113-117.

[4] N. Dvoretzky and Th. Motzkin, A problem of arrangements, *Duke Math J.* 14 (1947) 305-313.

[5] A. Del Lungo, M. Nivat, R. Pinzani, S. Rinaldi, The area of parallelogram polyominoes and a tiling game, Fun with Algorithms, 2, *Proceedings in Informatics* 10 (2001) 85-101.

[6] E. Pergola, R. Pinzani, S. Rinaldi, R.A. Sulanke, Lattice path moments by cut and paste, preprint, 2001.

[7] L. Shapiro, W-J Woan, and S. Getu, Runs, slides, and moments, *SIAM, J. of Disc. Math.* 4, (1983) 459-466.

[8] N. J. A. Sloane, *On-line Encyclopedia of Integer Sequences*, `http://www.research.att.com/~njas/sequences/index.html`

[9] R. P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge University Press, 1999

[10] R. A. Sulanke, Bijective Recurrences concerning Schröder Paths, *Electronic Journal of Combinatorics*, Vol. 5(1), R47, 1998

[11] R. A. Sulanke, Three Recurrences for Parallelogram Polyominoes, *J. of Difference Eq. and its Appl.*, 5 (1999) 155-176.

[12] R.A. Sulanke, Recurrences for moments of generalized Motzkin paths, *Journal of Integer Sequences*, Vol. 3 (2000), Article 00.1.1

[13] W-J Woan, L. Shapiro, and D. G. Rogers, The Catalan numbers, the Lebesgue integral and 4^{n-2}, *Am. Math. Monthly*, 104, (1997) 926-931.

Gaïa: a package for the random generation of combinatorial structures

Paul Zimmermann[1]

Gaïa is a computer algebra package that helps counting and drawing random combinatorial structures of various sorts. It is an implementation of the calculus developed by Ph. Flajolet, B. Van Cutsem and the author in [5]. Given a combinatorial specification and an integer n, it draws a random object uniformly amongst all size n structures. It applies to all decomposable structures, either labelled or unlabelled, including trees of various kinds, surjections, set partitions, permutations, functional graphs of many sorts.

Some applications of random generation are: (i) analyzing the average case complexity of algorithms by making simulations to guess or to check analytic results, (ii) checking the correctness of programs by feeding them with random inputs, (iii) getting ideas about some parameter of a class of objects, for example the height of trees or the number of connected components of graphs, (iv) simply drawing a random object.

Uniform random generation is difficult because there is generally no closed formula for the number A_n of data structures of size n, and secondly most methods require an explicit bijection with integers modulo A_n, but such a bijection is known only in a few cases (for example permutations and integer partitions, see the `combinat` package).

The main idea underlying the Gaïa system is first to transform the specification of a combinatorial class into a *standard* specification restricted to atoms and union, product, *pointing* constructors; then the standard specification is translated into counting and drawing procedures using some well-defined *templates*. This ensures a *really uniform* random generation in $O(n \log n)$ arithmetic operations in the worst case, after a $O(n^2)$ preprocessing to compute the counting sequences up to size n.

This article explains how to define a class of decomposable combinatorial structures with Gaïa, how to count the number of structures of a given size, how to generate a random structure and how to use it. Details about the algorithms used will be found in [5] and [6].

A simple example

Once you have properly installed Gaïa as a Maple package (see the section **Installing the package** below), it is very easy to generate a random object, for example a random binary tree:

```
% maple
> with(gaia):
> binary_tree := { B = Union(Z, Prod(B,B)) }:
> draw(binary_tree,unlabelled,B,7);

            Prod(Prod(Z, Prod(Z, Prod(Prod(Prod(Z, Z), Z), Z))), Z)

> draw(binary_tree,labelled,B,5);
```

[1]Inria Lorraine, Nancy, France, Paul.Zimmermann@loria.fr. This work was partly supported by the Esprit Basic Research Action No. 7141 (Alcom II).

```
Prod(Prod(Prod(Z[2], Prod(Z[5], Z[1])), Z[4]), Z[3])
```

The command `with(gaia)` loads the package, then one defines the grammar for binary trees, one draws an unlabelled tree of size 7 and a labelled one of size 5. The first two arguments of the `draw` command define a combinatorial specification, that is a grammar and a labelling type (see the section **Defining a combinatorial specification** below). The third argument indicates the type of object to be generated (the specification may define several types) and the last one the desired size.

The function `count` is similar to `draw`, except it gives the number of objects of a given size:

```
> count(binary_tree,labelled,B,33);
```

```
4822199239911149788434590729198926777763128934400000000
```

Defining a combinatorial specification

A class of decomposable combinatorial structures either contains only one object, or is built from simpler classes by means of *constructors*. The elementary classes are Epsilon, which denotes an object of size zero, and Z, which denotes an object of size one. The available constructors are:

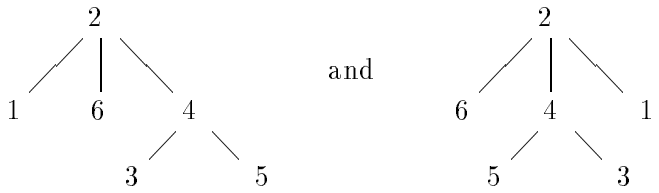
Atom	object of size 1 (Z is a predefined atom)
Union(A, B, ...)	disjoint union of the classes A, B, ...
Prod(A, B, ...)	product of the classes A, B, ...
Set(A)	all sets whose elements are in A
Sequence(A)	all sequences with elements of A
Cycle(A)	all directed cycles with elements of A.

For the constructors Set, Sequence and Cycle, it is possible to add some restrictions on the cardinality: for example, Set($A, \mathrm{card} \geq 1$) means all non empty sets whose elements are in A, Sequence($A, \mathrm{card} \leq 3$) means all sequences of at most three elements of A, and Cycle($A, \mathrm{card} = 5$) means all cycles of five elements from A.

A specification is a grammar and a labelling type, which is either 'labelled' or 'unlabelled'. In the labelled universe, each atom has a unique label, which is an integer from 1 to n, where n is the size of the whole object. In other words, the labels define a total order on all n atoms. In the unlabelled universe, there is no label. The grammar itself is a set of productions of the form $A = \langle \mathrm{rhs} \rangle$, where A is the name of the class being defined, and $\langle \mathrm{rhs} \rangle$ is an expression involving elementary classes, constructors and other classes. Below are some grammars and the corresponding combinatorial objects they define in the labelled universe.

$\{A = \mathrm{Prod}(Z, \mathrm{Set}(A))\}$	non plane trees
$\{B = \mathrm{Union}(Z, \mathrm{Prod}(B, B))\}$	plane binary trees
$\{C = \mathrm{Prod}(Z, \mathrm{Sequence}(C))\}$	plane general trees
$\{D = \mathrm{Set}(\mathrm{Cycle}(Z))\}$	permutations
$\{E = \mathrm{Set}(\mathrm{Cycle}(A)), A = \mathrm{Prod}(Z, \mathrm{Set}(A))\}$	functional graphs
$\{F = \mathrm{Set}(\mathrm{Set}(Z, \mathrm{card} \geq 1))\}$	set partitions
$\{G = \mathrm{Union}(Z, \mathrm{Prod}(Z, \mathrm{Set}(G, \mathrm{card} = 3)))\}$	non plane ternary trees
$\{H = \mathrm{Union}(Z, \mathrm{Set}(H, \mathrm{card} \geq 2))\}$	hierarchies
$\{L = \mathrm{Set}(\mathrm{Set}(\mathrm{Set}(Z, \mathrm{card} \geq 1), \mathrm{card} \geq 1))\}$	3-balanced hierarchies
$\{M = \mathrm{Sequence}(\mathrm{Set}(Z, \mathrm{card} \geq 1))\}$	surjections

A non plane tree (type A) is a root node (Z) to which are attached some subtrees that may take any position around the root, thus forming a set; the set may be empty, and this gives a terminal node, that is a leaf. For example,



represent the same labelled non plane tree. In plane binary trees (type B), the number of subtrees is restricted to be two or zero, and they are ordered. Thus we get the grammar $B = \mathrm{Union}(Z, \mathrm{Prod}(Z, B, B))$, or simply $B = \mathrm{Union}(Z, \mathrm{Prod}(B, B))$ if we do not count internal nodes. A plane general tree (type C) is similar to a non plane tree except the subtrees are ordered (now the two pictures above represent two different plane trees), thus we just replace the Set by a Sequence construction in the grammar of A.

For permutations (type D), we could represent a permutation on $\{1 \ldots n\}$ by the sequence of its images $\sigma_1 \ldots \sigma_n$, for example the sequence $6, 2, 5, 1, 3, 4$ would represent the permutation $\sigma_1 = 6, \sigma_2 = 2, \sigma_3 = 5, \sigma_4 = 1, \sigma_5 = 3, \sigma_6 = 4$. This would give the grammar $D = \mathrm{Sequence}(Z)$. But usually it is more convenient to work on the cycle decomposition, for example $(164)(2)(35)$ for the above permutation, which is defined by $D = \mathrm{Set}(\mathrm{Cycle}(Z))$. This last grammar is in some sense "more precise", the construction $\mathrm{Set}(\mathrm{Cycle}(\cdot))$ being equivalent to $\mathrm{Sequence}(\cdot)$ for labelled objects.

Functional graphs (type E) are graphs of functions on $\{1 \ldots n\}$. Such a function f has two kinds of points: cyclic points i such that some iterate of f on i goes back to i, such as $4, 8, 10, 11, 14$ on Figure 1, and other points, which are non-cyclic. Starting from any point, and iterating the function, we attain necessarily a cyclic point in a finite number of iterations (this is the trick used in Pollard's algorithm to find a factor of an integer). The set of points that go to the same cyclic point is a non plane tree (type A). A partition of a set is exactly a

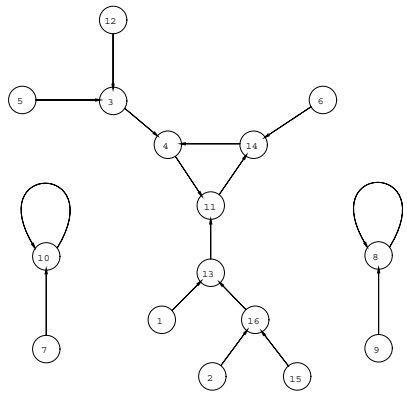


Figure 1: The graph of $x \longrightarrow x^2 + 12 \bmod 17$.

set of non-empty sets, the latter being defined by $\mathrm{Set}(Z, \mathrm{card} \geq 1)$, thus we get the grammar of F. Non plane ternary trees (type G) are defined like non plane trees, except the number of subtrees is either 0 or 3: in the above grammar, we simplified $\mathrm{Prod}(Z, \mathrm{Set}(G, \mathrm{card} = 0))$ into Z.

A hierarchy (type H) is similar to a non plane tree too, but unary nodes are forbidden, thus the number of subtrees is either zero or greater or equal to two. Three-balanced hierarchies (type L) are balanced non plane trees (all leaves are at the same level) of height exactly 3. Finally, a surjection (type M) from $\{1 \ldots n\}$ to a totally ordered set is equivalent to a sequence of non empty sets (the integers with image the smallest element are in the first set, those with image the second smallest one are in the second set, and so on).

Other combinatorial objects are defined by the following grammars in the unlabelled universe.

$\{A = \mathrm{Set}(\mathrm{Sequence}(Z, \mathrm{card} \geq 1))\}$	integer partitions
$\{B = \mathrm{Sequence}(\mathrm{Union}(Y, Z)), Y = \mathrm{Atom}\}$	binary sequences
$\{C = \mathrm{Cycle}(\mathrm{Set}(Z, \mathrm{card} \geq 1))\}$	necklaces
$\{D = \mathrm{Prod}(Z, \mathrm{Set}(D))\}$	rooted unlabelled trees
$\{E = \mathrm{Set}(\mathrm{Cycle}(D)), D = \mathrm{Prod}(Z, \mathrm{Set}(D))\}$	random mappings patterns
$\{F = \mathrm{Union}(Z, \mathrm{Set}(F, \mathrm{card} = 2))\}$	non plane binary trees
$\{G = \mathrm{Union}(Z, \mathrm{Set}(G, \mathrm{card} = 3))\}$	non plane ternary trees
$\{H = \mathrm{Union}(Z, \mathrm{Set}(H, \mathrm{card} \geq 2))\}$	unlabelled hierarchies
$\{M = \mathrm{Sequence}(\mathrm{Set}(Z, \mathrm{card} \geq 1))\}$	integer compositions

It should be noticed that the same grammar may define different kinds of objects. As an example, $\mathrm{Sequence}(\mathrm{Set}(Z, \mathrm{card} \geq 1))\}$ defines surjections in the labelled universe, but integer compositions in the unlabelled universe.

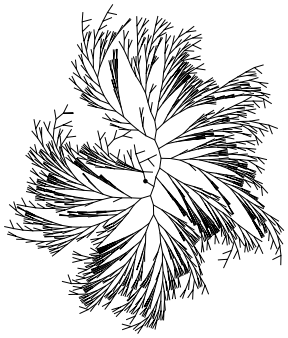
Here again, the specifications are explained as follows. An integer partition, for example $17 = 12 + 3 + 1 + 1$, is equivalent to a set of boxes of integer length, with repetitions allowed: $\{\square\square\square\square\square\square\square\square\square\square\square\square, \square\square\square, \square, \square\}$. Such a box is simply a non empty sequence of atoms: $\mathrm{Sequence}(Z, \mathrm{card} \geq 1)$. A necklace (type C) is a cycle of non empty sets of beads. By the way, let us remark that a set of beads $\mathrm{Set}(Z, \mathrm{card} \geq 1)$ is equivalent to a sequence of beads $\mathrm{Sequence}(Z, \mathrm{card} \geq 1)$ in the unlabelled universe.

Rooted non plane trees D have exactly the same grammar as in the labelled case. Similarly, random mappings patterns (type E) are the "skeletons" of functional graphs. Trees and hierarchies (types F, G and H) are defined like in the labelled case.

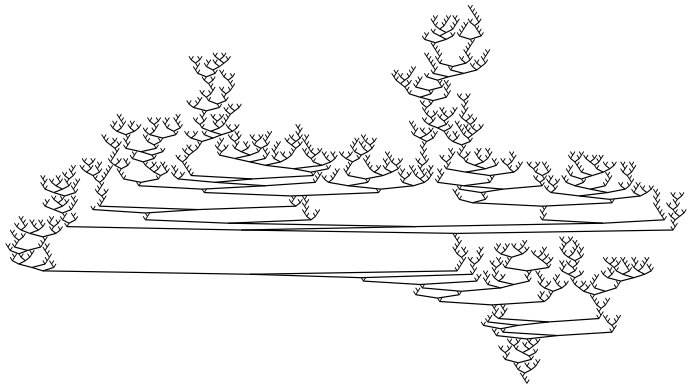
Figure 2 shows two objects of size 1000 generated using Gaïa: the first one is the binary search tree corresponding to a random permutation of size 1000 (type D in the labelled case), the second one is a plane binary tree. The left drawing was produced using a special-purpose Maple routine, and the right one was obtained using the algorithm described in [11] (Gaïa only produces a Maple expression, it does not include any graphical instruction). These examples show some values of interest that could be examined on combinatorial objects: the height of different kinds of trees, the number of sets in a random set partition, or the number of terms in a random integer partition, the distribution of degrees in general trees, the number of cycles in a permutation, ...

Using and printing objects generated by Gaïa

All objects produced by Gaïa are valid Maple expressions. They are either names (possibly labelled) representing atoms, or inert functions for all constructors. Thus you can access the components of an object with the usual Maple functions `op`, `nops`. For example, the following function computes the size of an object:



A binary search tree of size 1000.
`{D=Set(Cycle(Z))},labelled`



A binary plane tree of size 1000.
`{B=Union(Z,Prod(B,B))},unlabelled`

Figure 2: Two random objects generated with Gaïa.

```
size := proc(e)
    if type(e,epsilon) then 0
    elif type(e,name) then 1
    else convert(map(procname,e),`+`)
    fi
end:
```

We can check it rapidly:

```
> size(draw(binary_tree,unlabelled,B,20));
```

```
                                    20
```

If you want your objects to be printed another way than the default, you can easily do it by redefining the functions `gaia/print/xxx` where `xxx` is a constructor. Take for example Cayley trees, which are printed by default as follows:

```
> Cayley := {A = Prod(Z,Set(A))},labelled:
> draw(Cayley,A,4);
```

```
  Prod(Z[2], Set(Prod(Z[1], EmptySet), Prod(Z[4], Set(Prod(Z[3], EmptySet)))))
```

If you want to use Maple curly-bracket notation instead, just redefine `gaia/print/Set` for general sets and `gaia/print/EmptySet` for empty sets:

```
> `gaia/print/Set` := () -> {args}:
> `gaia/print/EmptySet`  := () -> {}:
> draw(Cayley,A,4);
```

```
      Prod(Z[1], {Prod(Z[3], {{}, Prod(Z[2], {})}), {{}, Prod(Z[4], {})}})
```

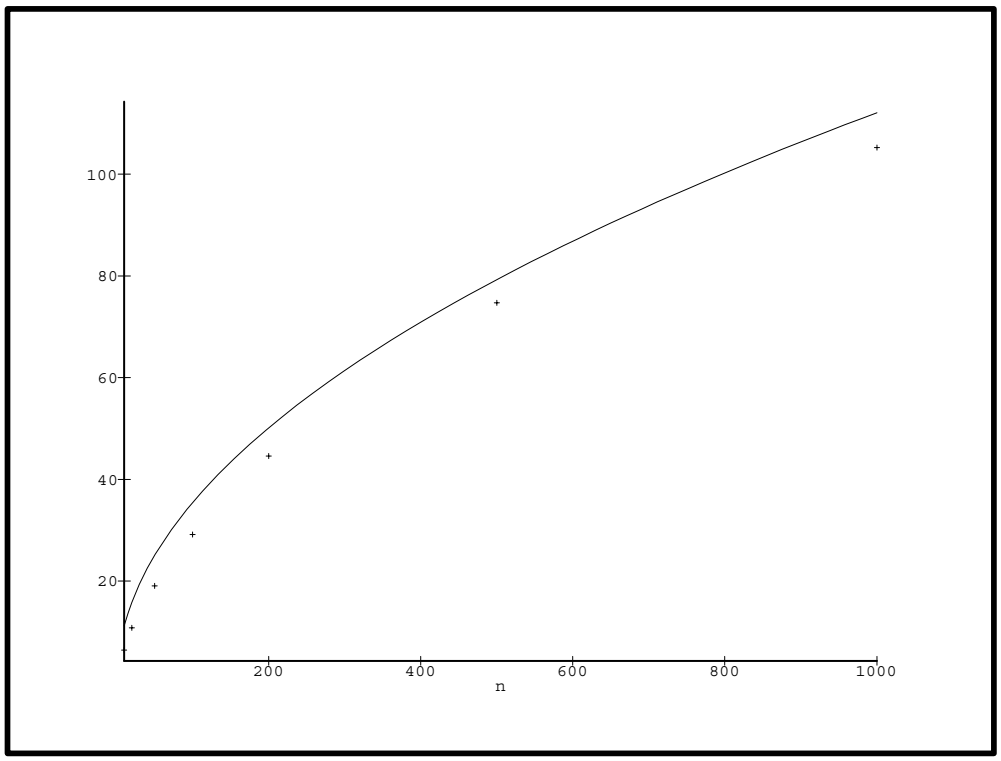
Notice that the `gaia/print/xxx` functions do not only modify the way objects are *printed* like the `print/xxx` functions of Maple, but really modify the internal structure of the objects (and consequently user-defined functions like `size` above may have to be redefined accordingly). This behaviour enables one to work further with random objects.

For example, suppose we want to analyze the height of unlabelled binary trees. We first write a `height` function:

```
height := proc(b)
    if type(b,name) then 0 else 1+max(height(op(1,b)),height(op(2,b))) fi
end:
```

and we are ready to experiment and compare to the actual result of $2\sqrt{\pi n} + O(n^{1/4+\epsilon})$ from [4, Theorem B page 200]. We plot for different sizes the average height over 100 random binary trees.

```
> s:=NULL:
> for n in [10,20,50,100,200,500,1000] do
>   l:=seq(height(draw(binary_tree,unlabelled,B,n)),i=1..100);
>   s:=s,[n,stats[average](l)]
> od:
> exper:=plot([s],n=10..1000,style=POINT):
> theor:=plot(2*sqrt(Pi*n),n=10..1000):
> plots[display]({exper,theor});
```



Similarly, one could analyze the path length of binary trees, the number of cycles in a random permutation, the number of connected components of a random functional graph, the number of elements in a set partition or an integer partition, the average node degree in a random hierarchy, and so on.

Some advanced examples

A lot of combinatorial structures encountered in the literature are decomposable, that is expressible by a specification in Gaïa. For example we saw in the section **Defining a combinatorial specification** that a functional graph on $\{1 \ldots n\}$ is a set of cycles, each cycle being made of non plane trees; a functional digraph on $\{1 \ldots n\}$ is similar, except the cycles must have at least two elements. We can easily check the figures given in [9, p. 70]:

```
> sys:={F=Set(Cycle(D)),D=Prod(Z,Set(D)),FD=Set(Cycle(D,card>=2))},unlabelled:
> seq(count(sys,FD,n),n=1..11);

                0, 1, 2, 6, 13, 40, 100, 291, 797, 2273, 6389

> seq(count(sys,F,n),n=1..11);

              1, 3, 7, 19, 47, 130, 343, 951, 2615, 7318, 20491
```

Another beautiful example was suggested by Volker Strehl. We consider bicolored functional graphs on $\{1 \dots n\}$, where each point has a color, either blue or red, and has at most one ancestor of each color. The corresponding specification is the following, with `Ab` (resp. `Ar`) denoting trees with a blue (resp. red) root, and `E` denoting bicolored functional graphs.

```
> sys := {Ab = Union(b,Prod(b,A),Prod(b,Ab,Ar)),
        Ar = Union(r,Prod(r,A),Prod(r,Ab,Ar)),
        A = Union(Ab,Ar),
        A2 = Union(Prod(r,Ab),Prod(b,Ar)),
        C = Cycle(Union(A2,b,r)),
        E = Set(C),
        b = Atom,
        r = Atom}, labelled:

> seq(count(sys,E,n),n=0..9);

      1, 2, 12, 120, 1680, 30240, 665280, 17297280, 518918400, 17643225600
```

The numbers found are exactly $(2n)!/n!$ up to $n = 9$. It is left as an exercise to the reader to check if this is true for every n. This is a typical example of research in combinatorics: defining with Gaïa a particular kind of objects, computing the first numbers, looking for an explicit formula or for similar sequences in Sloane's book [12], and perhaps deriving a bijection with other combinatorial objects.

The list of combinatorial constructors given above is not complete. In fact, the system itself uses two other constructors, `Theta` and `Int`. The construction `Theta(A)` produces objects of type `A` with one atom having a special mark, and `Int(A)` simply erases the mark in the objects of type `A`. Thus the constructor `Int` is only valid for marked objects. These two constructors are used in the *standard form* of combinatorial specifications (see [5] for more details). As an example, the standard form of the labelled specification `A=Set(B)` is:

```
> standardform({A = Set(B)},labelled);

 {T1 = Prod[Set](T0, A), T2 = Int(T1), A = Union(EmptySet, T2), T0 = Theta(B)}
```

which means that (an object of type) A is either the empty set or T_2, T_2 being an object of type T_1 without the mark, T_1 being the product of T_0 and A, and T_0 being a marked object of type B.

In the unlabelled case, the standard specification uses a third constructor, the generalized diagonal `Delta` defined in [6]:

```
> standardform({A = Set(B)},unlabelled);

   {T1 = Delta[Set](T0), T2 = Prod[Set](T1, A), T0 = Theta(B), T3 = Int(T2),

       A = Union(EmptySet, T3)}
```

These three constructions `Theta`, `Int` and `Delta` allow you to define a wider class of structures. The following specifies for example unrooted non plane trees (the reader is not necessarily supposed to understand the specification, which is based on the notion of *similar node* defined in [9]).

```
> sys:={T=Prod(Z,Set(T)),t=Int(Union(T,Prod(T,Delta[Set(2)](Theta(T))),
                                  Delta[2](Theta(T))))},unlabelled:
> sum('count(sys,t,n)*x^n',n=1..10);
```

$$x + x^2 + x^3 + 2\,x^4 + 3\,x^5 + 6\,x^6 + 11\,x^7 + 23\,x^8 + 47\,x^9 + 106\,x^{10}$$

A lot of examples in the book of Harary and Palmer can be checked in the same manner, like in those of Comtet [3], Goulden and Jackson [7] and Bollobás [2].

Installing the package

For those who have an access to Internet, the Gaïa package is available by anonymous ftp from the machine `ftp.inria.fr`:

```
% ftp ftp.inria.fr
Name (ftp.inria.fr:zimmerma): anonymous
Password: <your e-mail address>
ftp> cd INRIA/Projects/algo/gaia
ftp> bin
ftp> get gaia1.1.tar.Z
ftp> quit
% uncompress gaia1.1.tar.Z
% tar xvf gaia1.1.tar
```

This will create the following files: `gaia.mpl`, `gfun.mpl`, `gaia.test` and `README.tex`. Then you must create a Maple ".m" file from the files `gaia.mpl` and `gfun.mpl`. To do this, type

```
% maple -s -q < gaia.mpl
% maple -s -q < gfun.mpl
```

You have now two files `gaia.m` and `gfun.m`. To be able to load the Gaïa package easily from Maple, add in your `.mapleinit` file (in your home directory) the line

```
libname := `/users/eureca/zimmerma/Gaia`,libname:
```

(`/users/eureca/zimmerma/Gaia` is the directory where the file `gaia.m` lies). Once you have created the file `gaia.m` and updated your `.mapleinit` file, just check that all works properly:

```
% maple -q < gaia.test
Total time=   215.133
```

Further developments. Due to the exponential growth of the counting sequences coefficients, the more expensive operations are those that deal with those huge numbers (the number of unlabelled binary trees of size 1000 has 597 digits). For this kind of computation, Maple is not as efficient as some specialized libraries like GMP [8], BigNum [10] or Pari [1]. An interface with these multiprecision libraries is in preparation. It works as follows: in Maple, you type

```
> compile(binary_tree,unlabelled,gmp,`foo.c`);
```

and this creates a C program `foo.c` that generates random unlabelled binary trees, using the multiprecision library GMP. The generation of random objects is about ten times faster with the C interface. The trees on page 5 were generated in about 10 seconds each using this C interface. Please contact the author for more information on this.

Once a random object was generated, Gaïa is not able to generate the *next* one, like the function `nextpart` of the `combinat` package. This ability would be very useful, because it would enable one to list all objects of a given size. Unfortunately, as already said in the introduction, this would require an explicit bijection between objects of size n and integers modulo A_n. This seems to be awkward with the methods of [5, 6].

Acknowledgement. The author thanks Bruno Salvy for his comments on a previous version of this article, and the referees for their careful reading and interesting remarks.

References

[1] Batut, C., Bernardi, D., Cohen, H., and Olivier, M. *User's Guide to PARI-GP*, Dec. 1991. Available by anonymous ftp from `megrez.ceremab.u-bordeaux.fr` or `math.ucla.edu`.

[2] Bollobás, B. *Random Graphs*. Academic Press, 1985.

[3] Comtet, L. *Advanced Combinatorics*. Reidel, Dordrecht, 1974.

[4] Flajolet, P., and Odlyzko, A. The average height of binary trees and other simple trees. *J. Comput. Syst. Sci. 25* (1982), 171–213.

[5] Flajolet, P., Zimmermann, P., and Cutsem, B. V. A calculus for the random generation of combinatorial structures. *Theoretical Comput. Sci.*. 29 pages. To appear. Also available as Inria Research Report number 1830.

[6] Flajolet, P., Zimmermann, P., and Cutsem, B. V. A calculus of random generation: Unlabelled structures. In preparation.

[7] Goulden, I. P., and Jackson, D. M. *Combinatorial Enumeration*. John Wiley, New York, 1983.

[8] Granlund, T. *GNU MP: The GNU Multiple Precision Arithmetic Library*, 1.2 ed., Dec. 1991. Available by anonymous ftp from `sics.se`.

[9] Harary, F., and Palmer, E. M. *Graphical Enumeration*. Academic Press, 1973.

[10] Hervé, J.-C., Serpette, B., and Vuillemin, J. BigNum: A Portable and Efficient Package for Arbitrary-Precision Arithmetic. Tech. Rep. 2, Digital Paris Research Laboratory, May 1989.

[11] Reingold, E. M., and Tilford, J. S. Tidier drawings of trees. *IEEE Trans. Softw. Eng. SE-7*, 2 (Mar. 1981), 223–228.

[12] Sloane, N. J. A. *A Handbook of Integer Sequences*. Academic Press, 1973.

Les Cahiers du GERAD ISSN: 0711–2440

What Forms Do Interesting Conjectures Have in Graph Theory ?

P. Hansen, M. Aouchiche, G. Caporossi
H. Mélot, D. Stevanović

G-2002-46

August 2002
Revised: August 2003,
Draft. Do not cite without the authors' permission.

What Forms Do Interesting Conjectures Have in Graph Theory ?

Pierre Hansen
GERAD and HEC Montréal

Mustapha Aouchiche
École Polytechnique de Montréal

Gilles Caporossi
GERAD and HEC Montréal

Hadrien Mélot
University of Mons-Hainault
Belgium

Dragan Stevanović
University of Nis
Yugoslavia

August, 2002
Revised: August, 2003

Les Cahiers du GERAD
G–2002–46

Abstract

Conjectures in graph theory have multiple forms and involve graph invariants, graph classes, subgraphs, minors and other concepts in premisses and/or conclusions. Various abstract criteria have been proposed in order to find interesting ones with computer-aided or automated systems for conjecture-making. Beginning with the observation that famous theorems (and others) have first been conjectures, if only in the minds of those who obtained them, we review forms that they take. We also give examples of conjectures of such forms obtained with the help of, or by, computers when it is the case. It appears that many forms are unexplored and so computer-assisted and automated conjecture-making in graph theory, despite many successes, is pretty much at its beginning.

Keywords: graph, conjecture, computer-aided system, automated system, invariant, subgraph, minor.

Résumé

Les conjectures en théorie des graphes ont des formes multiples et impliquent des invariants graphiques, des classes de graphes, des sous-graphes, des mineurs et d'autres concepts dans les prémisses et/ou conclusions. Divers critères abstraits ont été proposés afin de trouver des conjectures intéressantes avec l'assistance de l'ordinateur ou à l'aide de systèmes automatisés. A partir de l'observation que les théorèmes célèbres (et les autres) ont d'abord été des conjectures, ne fut-ce que dans l'esprit de ceux qui les ont obtenus, on passe en revue les formes qu'elles peuvent prendre. On donne également des exemples pour les formes pour lesquelles des systèmes assistés ou automatisés ont donné des résultats. Il apparait que de nombreuses formes sont inexplorées et en conséquence la recherche de conjectures assistée par ordinateur ou automatisée, malgré de nombreux succès, en est encore à ses débuts.

Mots clés: graphe, conjecture, système assisté, système automatisé, invariant, sous-graphe, mineur.

1 Introduction

"*What makes a mathematical result interesting?*" This difficult question of mathematical philosophy is seldom discussed, despite its obvious interest. Recently, needs of computer-assisted or automated systems for finding interesting new concepts, theorems or conjectures have given it some actuality, notably in graph theory. Views of several famous scientists on this topic are interspersed with discussions of graph theoretical conjectures in the large *Written on the wall* file of Fajtlowicz [50]. Colton *et al.* [32] and Larson [67], also address this question in detail.

We next mention and briefly discuss a few proposed criteria:

(a) *simplicity*: simple formulae are the most used ones, and thus the most likely to have many consequences. They also have the most potential falsifiers, as explained by Popper in his famous book "*The Logic of scientific discovery*"[76]. However, it may be hard to find many simple, new and true formulae. Moreover, some of them may be trivial, e.g., that the clique number of a graph is not larger than its chromatic number.

In a similar vein, one might suggest the two following criteria:

(b) *centrality*: conjectures should preferably involve the most central concepts of graph theory as e.g. connectedness, stability, colorability, and so forth. To illustrate, some new concepts proved to be interesting and lead to numerous results, as e.g. *pancyclicity* or having elementary cycles of all possible lengths, introduced by Bondy [10], which is close to the basic concept of cycle. This is far from being always the case for the numerous new concepts which nowadays proliferate and, to some extent, threaten the unity of graph theory.

(c) *problem solving*: instead of considering centrality in terms of concepts, one may examine it in terms of problems posed by scientists in a given field. This leads to another criterion, again stated by Popper in *"The Logic of Scientific Discovery"* [76]: "Only if it is the answer to a problem – a difficult, a fertile problem, a problem of some depth – does a truth, or a conjecture about the truth, become relevant to science. This is so in pure mathematics, and it is so in the natural sciences."

A quite different criterion is the following:

(d) *surprisingness*: Conway's answer to the question "What makes a good conjecture?" was "It should be outrageous" [50]. This means a trained mathematician finds something contrary to what suggests his well-educated intuition, and so gets a new insight. Of course, it remains to be examined whether some explanation may be found, together with new results, or the conjecture will remain an isolated curiosity.

(e) *distance between concepts* is one version of surprisingness: a conjecture will be the more interesting the farther the concepts involved are one from another. This implies an operational notion of distance, either in the conjecture-making program or possibly in a lattice of graph-theoretical concepts.

Another view comes from information theory:

(f) *information-content* relative to databases of conjectures and graphs. A conjecture is interesting if it tells more, for at least one graph than the conjunction of all other conjectures. This is the criterion of the "DALMATIAN" version of Graffiti [50], discussed in [60]. It also means the conjecture should not be redundant.

A more demanding related criterion is:

(g) *sharpness:* the conjecture should be best possible in the weak sense, i.e., sharp for some values of the parameters, or in the strong sense, i.e., sharp for all values of the parameters compatible with the existence of a graph [60].

In addition to such abstract criteria one might take a pragmatic view and say that a conjecture is interesting if it has attracted the attention of mathematicians, whoever they may be. This is fairly tautological. Note, moreover, that popularity of a result depends not only on its intrinsic merits but also on its visibility (Journal where it was published, computer systems which mention it or give access to it, as well as relations and aptitude for marketing of its author(s)).

In this paper, we follow a different approach, beginning from the observation that *well-known theorems in graph-theoretical books and papers were first conjectures, if only in the minds of those which proved them.* Instead of seeking an abstract and general criterion we more modestly try to find what forms have a number of well-known results in graph theory. On this base we reflect on what is done by available conjecture making systems, and what remains to be done.

Let us recall the definition of conjecture in Bouvier and George's [13] *Dictionary of Mathematics*:

Conjecture: *An a priori hypothesis on the exactness or falseness of a statement of which one ignores the proof.*

As a *statement* is a very general concept in mathematics, one can expect to find conjectures of many forms. We are, as mentioned above, interested here in the various forms of graph-theoretic conjectures. We therefore make a tentative, and necessarily incomplete, catalog of such forms using books by Berge [6], Biggs [7], Bondy and Murty [12], Busacker and Saaty [20], Cvetković, Doob and Sachs [34], Haynes, Hedetniemi and Slater [61] and a few others prominent among which is Chung and Graham's book *Erdös on Graphs* [30].

We also mention, with an example if possible, if a form has been explored by one or another system for computer-assisted or automated conjecture-making in graph theory. In accordance with the terminology of [60] we say a conjecture has been obtained *with* a system if this was done in computer-assisted mode and *by* a system if this was done in (fully) automated mode. Note that several systems can be used in either of those modes. Moreover, we mention some cases where systems, designed for other purposes, could be used for conjecture-making. As will be seen, many unexplored cases remain, most of which could apparently be explored by some enhanced version of one or another existing system.

2 Algebraic relations

2.1 General form

A first class of graph-theoretic conjectures consist in algebraic relations between graph invariants, i.e., quantities which are independent of vertices and edge labelings. Such relations may be valid for any graph G or for some particular class of graphs.

To date, this class of conjectures is the most studied, but far from the only one, in computer-assisted and automated conjecture-making, see [60] for a discussion.

Let R denote a relation and C a class of graphs; any graph G can be associated with a boolean variable, true (or equal to 1) if G belongs to this class and false (or equal to 0) otherwise [14] [16].

The general form of conjectures considered in this section can then be written

$$R \,|\, C \qquad (\text{or } \; C \Rightarrow R)$$

which reads:

"For any graph of class C, relation R holds".

If a relation holds for all graphs, C can be omitted.

We now review theorems and conjectures of this form, considering first R, then C, and going from the simplest to the more elaborate ones.

2.2 Linear relations and extensions

Let $G = (V, E)$ be a simple undirected graph without loops, with *order* $n = |V|$ and *size* $m = |E|$. Let $\alpha(G)$ denote the *independence number* of G, i.e., the largest number of pairwise non adjacent vertices, $\nu(G)$ the *matching number* of G, i.e., the largest number of pairwise non-incident edges, $\tau(G)$ the *vertex covering number* of G, i.e., the smallest number of vertices in a set such that each edge contains at least one of those vertices, and $\epsilon(G)$, the *edge covering number* of G, i.e., the smallest number of edges in a set such that each vertex belongs to at least one of those edges. Denote by R_1 the class of linear equalities between invariants of G.

Theorem 1 (Norman, Rabin [71], Gallai [55]) *For any graph G with matching number $\nu(G)$, edge covering number $\epsilon(G)$, vertex covering number $\tau(G)$, independence number $\alpha(G)$ and order n,*

$$\nu(G) + \epsilon(G) = n$$

and if G has no isolated vertex

$$\alpha(G) + \tau(G) = n.$$

Such equalities, valid for all graphs (or for a very large class) are rare. They are more common for particular classes of graphs. Recall that a *tree* T is a connected graph without

cycles (paths with the last vertex equal to the first one). Let $\omega(G)$ denote the *clique number* of G, i.e., the largest number of pairwise adjacent vertices and $\chi(G)$ the *chromatic number* of G, i.e., the smallest number of colors to be assigned to the vertices of G such that no pair of adjacent vertices get the same color.

Theorem 2 (Folklore) *For any tree T,*

$$m = n - 1,$$

$$\omega(T) = 2$$

and

$$\chi(T) = 2.$$

Observe that coefficients of invariants in these relations are equal to 1. This need not always be the case.

Let n_1 denote the number of *pending vertices* of G, i.e., the number of vertices each belonging to a single edge. Recall the *distance* l_{ij} between a pair of vertices v_i and v_j of a graph G is the number of edges in a shortest path joining them. The *eccentricity* ecc_i of a vertex v_i is the largest distance between that vertex and another one. A *center* of G is a vertex v_i with smallest eccentricity; this eccentricity is called the *radius* of G. The *diameter* $D(G)$ of a graph G is the maximum eccentricity of its vertices, (or the largest distance between two vertices of G). The *index* (or *spectral radius*) of G is the largest eigenvalue of its *adjacency matrix* $A = (a_{ij})$, where $a_{ij} = 1$ if v_i and v_j are adjacent and 0 otherwise.

Conjecture 1 (Caporossi, Hansen [25] [24]) *For any tree T of size m and order n with n_b black and n_w white vertices, $n = n_b + n_w$, with minimum index, independence number $\alpha(T)$, n_1 pending vertices, radius r and diameter $D(T)$,*

$$2\alpha(T) - m - n_1 + 2r(T) - D(T) = 0.$$

This conjecture, obtained by AGX, is open. It is unlikely that an equality conjecture with as many invariants could be found by hand. Note that coefficients of invariants are small integers. AGX can also obtain conjectures with real numbers (approximated to a reasonable extent, as computations are made by machine).

Let d_j, for $j = 1, 2, .., n$, denote the *degree* of vertex v_j,i.e., the number of edges incident with v_j. Recall that the *Randic index* [79] of a graph $G = (V, E)$ is defined by

$$Ra(G) = \sum_{(i,j)/\{v_i,v_j\}\in E} \frac{1}{\sqrt{d_i d_j}}$$

and the irregularity $irr(G)$ [1] of G by

$$irr(G) = \sum_{(i,j)/\{v_i,v_j\}\in E} |d_i - d_j|.$$

Conjecture 2 *For any tree T of size m with maximum degree $\Delta \leq 3$ and maximum irregularity $irr(T)$, Randic index $Ra(T)$, and n_1 pending vertices,*

$$Ra(T) = -0.027421\, irr(T) + 0.538005\, m - 0.1104848\, n_1 + 0.614014.$$

This conjecture is proved in the Appendix. Extremal trees have vertices of degree 3 and 1 alternatingly, as far as possible. Note that the system GRAPH [33] [35] could also have been used to find such extremal trees interactively, and, after characterizing them, possibly lead to the above result.

Linear inequalities form a class R_2 of relations and are more common in graph theory than linear equalities. Let $\chi'(G)$ denote the *edge-chromatic number* (or chromatic index) of G, i.e., the smallest number of colors needed to color the edges of G such that no two incident edges have the same color.

Theorem 3 (Vizing [85]) *For any graph G with maximum degree Δ and chromatic index $\chi'(G)$*

$$\Delta \leq \chi'(G) \leq \Delta + 1.$$

Many linear inequality conjectures have been obtained by several systems, and proved, refuted or remain open. We mention a few. Let $\bar{l}(G)$ denote the average distance between pairs of vertices of G.

Conjecture 3 (Graffiti 2, Fajtlowicz [50]) *For any connected graph G with average distance $\bar{l}(G)$ and independence number $\alpha(G)$,*

$$\bar{l}(G) \leq \alpha(G).$$

Conjecture 4 (Graffiti 3, Fajtlowicz [50]) *For any connected graph G with average distance $\bar{l}(G)$ and Randic index $Ra(G)$,*

$$\bar{l}(G) \leq Ra(G).$$

Both conjectures were obtained with Graffiti; the former was proved by Chung [29] and the latter is open.

A *chemical graph* G has maximum degree 4 (due to the valency of carbon).

Conjecture 5 (Caporossi, Hansen [25] [24]) *For any chemical graph G with Randic index $Ra(G)$, size m and n_1 pending vertices,*

$$Ra(G) \geq \frac{m + n_1}{4}.$$

This conjecture, obtained by AGX, was proved using arguments based on linear programming.

A shorter proof is the following. Let $G = (V, E)$ and $E = E_1 \cup E_2$ where E_1 denotes the edges of G adjacent to a leaf and E_2 those which have both endvertices of degree at least 2. $|E_2| = |E| - |E_1| = m - n_1$. Moreover, for any edge $\{v_i, v_j\} \in E_1, 1/\sqrt{d_i d_j} \geq 1/2$ as d_i and $d_j \leq 4$ and one of d_i and d_j is equal to 1, and for any edge $\{v_i, v_j\} \in E_2, 1/\sqrt{d_i d_j} \geq 1/4$. Hence, $Ra(G) \geq (m - n_1)/4 + n_1/2 = (m + n_1)/4$. □

A third class of relations, R_3, is obtained by using floor and ceiling operators.

Let $\gamma(G)$ denote the *domination number* of G (or *exterior stability number*), i.e., the smallest number of vertices in a set such that any vertex not in the set is adjacent to one in the set; let $g(G)$, the *girth* of G denote the length of the smallest cycle of G.

Theorem 4 (Brigham, Dutton [15]) *For any graph G with minimum degree $\delta \geq 2$ and girth $g(G) \geq 5$,*

$$\gamma(G) \leq \left\lceil \frac{n - \lfloor g(G)/3 \rfloor}{2} \right\rceil.$$

Not much has been done regarding the use of the operators $\lfloor a \rfloor$ (floor of a, or largest integer not larger than a) and $\lceil a \rceil$ (ceiling of a or smallest integer not smaller than a) in computer-assisted or automated conjecture-making in graph theory. Exceptions are a few conjectures obtained with Graffiti [38] and the following conjecture. Recall that the *distance polynomial* of a graph G is defined as

$$P(G) = n + mx + \sum_{k=1}^{\lfloor \frac{D(G)}{2} \rfloor} p_k x^k,$$

where p_k denotes the number of pairs of vertices v_j, v_l at distance k. Then this polynomial will be *palindromic* if

$$p_k = p_{D(G)-k} \qquad k = 0, 1, 2, .., \lfloor \frac{D(G)}{2} \rfloor.$$

and the *distance to the palindrome condition* is defined as

$$dist(G) = \sum_{k=0}^{\lfloor \frac{D(G)}{2} \rfloor} |p_{D(G)-k} - p_k|.$$

Clearly if $dist(G) = 0$ the polynomial is palindromic. AGX [22] could find trees T with a palindromic distance polynomial $P(T)$ and an even diameter $D(T)$ (finding graphs G with a palindromic distance polynomial is easy) but not with an odd diameter $D(T)$. However, its use led to

Conjecture 6 (Caporossi *et al.*[22]) *For any tree T with odd diameter $D(T)$,*

$$dist(T) \geq \lceil \frac{n}{2} \rceil.$$

This conjecture is open (and apparently hard). It was obtained interactively with AGX; however the non-automated part was easy as AGX produced trees T with odd diameter and distances $dist(T)$ equal to 5,6,6,7,7,8,8 and so forth for $n = 10$ to $n = 50$ without exception, from where the conjecture follows immediately.

2.3 Non-linear relations

A fourth class of relations, $R4$, involves powers of invariants or products of them. Usually powers are squares, cubes, inverses, square or cubic roots. Products usually involve only a pair of invariants. Recall that the *complementary graph* $\bar{G}$ of a graph G has an edge joining vertices v_i and v_j if and only if G has not.

Theorem 5 (Nordhaus, Gaddum [70]) *For any graph G of order n with chromatic number $\chi(G)$,*

$$2\sqrt{n} \leq \chi(G) + \chi(\bar{G}) \leq n + 1$$

and

$$n \leq \chi(G).\chi(\bar{G}) \leq \frac{(n+1)^2}{2} = \frac{n^2}{2} + n + \frac{1}{2}.$$

Systems Graffiti and AGX led to several conjectures with powers or products of invariants. Define [50] the temperature t_j of vertex v_j of G as

$$t_j = \frac{d_j}{n - d_j} \quad j = 1, 2, .., n.$$

Conjecture 7 (Graffiti 834, Fajtlowicz [50]) *For any connected graph G with average distance $\bar{l}(G)$ and temperature of vertices of the complementary graph $t_j(\bar{G})$, $j = 1, \ldots n$,*

$$\bar{l}(G) \leq 1 + \max_j \ t_j(\bar{G}).$$

This conjecture could be reformulated as

$$(1 + \delta(G))\, \bar{l}(G) \leq n,$$

and was refuted by AGX [26]; the counter-example consists of two triangles joined by a path with seven edges. A weaker, but simple and elegant, conjecture is the following:

Conjecture 8 (Graffiti 127, Fajtlowicz [50]) *For any connected graph G*

$$\delta(G).\bar{l}(G) \leq n.$$

After this conjecture remained open for more than 10 years, a stronger result, implying it as a corollary, was obtained by Beezer *et al.* [5].

The *energy* E of a graph G can be defined [57] [56] as

$$E = \sum_{i=1}^{n} |\lambda_i|$$

where the λ_i , $i = 1, 2, \ldots, n$ are the eigenvalues of the adjacency matrix $A(G)$ of G.

Conjecture 9 (Caporossi *et al.* [21]) *For any graph* G,

$$E \geq 2\sqrt{m}$$

and

$$E \geq \frac{4m}{n}.$$

Both relations, obtained with AGX, could easily be proved.

A fifth, rare, class of relations, R_5, involve exponentials or logarithms.

Theorem 6 (Berge [6]) *For any connected graph* G *with a maximum degree* $\Delta \geq 2$ *and radius* $r(G)$,

$$r(G) \geq \frac{log(n\,\Delta - n + 1)}{log(\Delta)}$$

A few other conjectures involving logarithms were recently obtained with Graffiti [38].

Let $\rho(G)$ denote the *path covering number* of G, i.e., the smallest number of vertex disjoint paths needed to cover all vertices of G.

Conjecture 10 and 11 (De La Vina *et al.* [38]) *For any graph* G *with independence number* $\alpha(G)$, *radius* $r(G)$ *and path covering number* $\rho(G)$,

$$\alpha(G) \geq r(G) + ln(\rho(G))$$

and

$$\alpha(G) \geq ln(r(G)) + \rho(G).$$

These conjectures are open.

2.4 Qualitative relations

Relations of another form, i.e., *qualitative* ones, define class R_6. They are rarely used in graph theory but quite frequent in other fields such as economics [81], particularly in *comparative statics*. Qualitative relations describe trends of invariants. e.g:

"invariant i_1 *increases when invariant* i_2 *increases"*

or

"invariant i_1 *decreases when invariant* i_2 *increases",*

which may be expressed by

$$\frac{\Delta i_1}{\Delta i_2} > 0 \qquad \text{and} \qquad \frac{\Delta i_1}{\Delta i_2} < 0$$

respectively, where Δi_2 is an increase in invariant i_2 and Δi_1 the corresponding change in the invariant i_1.

A tree with n vertices is bipartite and its vertices can be colored, say, in black and white; let n_b and n_w denote the numbers of black and of white vertices respectively (with $n_b + n_w = n$). In [37] color-constrained trees, i.e., trees with fixed n and $n_b \geq n_w$, and with minimum index are studied. This led to the following result:

Conjecture 12 (Cvetković *et al.* [37]) *For all trees T with n vertices, n_b black ones and n_w white ones, $n_b \geq n_w$, the minimum value of the index λ_1 (T) increases monotonously with $n_b - n_w$.*

This qualitative conjecture was obtained with AGX and is proved in the cited reference.

2.5 Conditions

We next discuss the classes C of graphs G which are the most used in conjectures of the type $R|C$. Several of them have already been illustrated by examples given above.

A first class, C_1, is composed of *conditions necessary for the invariants $i_1, i_2, ...$ used in the relation R to be defined*. Quite often the graph will have to be *connected*, i.e., any two vertices must be joined by a path.

Examples are conjectures 3,4,7 and 8 above where connectedness is needed for average distance not to be infinite. In other conjectures, such as those on trees, e.g. conjecture 6 above, connectedness is implicit, as a tree is a connected graph without cycles.

Another class C_2 consists of *conditions eliminating trivial cases.* An example is that there should be no isolated points, i.e., the minimum degree $\delta(G) \geq 1$. This is illustrated by the second formula of Gallai's theorem (Theorem 1 above).

Forbidden subgraphs can also be used to obtain well-known classes of graphs, which we denote collectively by C_3.

A first case is *triangle-free* graphs.

Theorem 7 (Fraughnaugh, Locke [54]) *For any connected triangle-free 3-regular graph G with independence number $\alpha(G)$ and order n,*

$$\frac{\alpha(G)}{n} \geq \frac{11}{30} - \frac{2}{15n} \qquad \left(or\ \alpha(G) \geq \frac{11}{30} n - \frac{2}{15} \right)$$

Conjecture 13 (Graffiti 116, Fajtlowicz [50]) *For any triangle-free graph G with index $\lambda_1(G)$ and Randić index $Ra(G)$,*

$$\lambda_1(G) \leq Ra(G).$$

This has been proved by Favaron, Mahéo and Saclé [51].

A generalization is to consider graphs without odd cycles C_{2k+1} for all positive integers k, i.e., *bipartite graphs*.

Theorem 8 (König [64]) *For any bipartite graph G with matching number $\nu(G)$ and vertex covering number $\tau(G)$,*

$$\nu(G) = \tau(G).$$

A more drastic condition is to exclude all cycles, which of course gives trees, if connectivity is assumed, and *forests* otherwise.

Conjecture 1 above does not hold for all trees; the following one does

Conjecture 14 (Caporossi, Hansen [25] [24]) *For any tree T,*

$$\alpha(T) \leq \frac{1}{2}(m + n_1 + D(T) - 2r(T))$$

and

$$\alpha(T) \geq \frac{1}{2}(m + n_1 + D(T) - 2r(T) - [\frac{n-2}{2}]).$$

Symbols are defined above. Both relations were found with AGX; the former is proved in [25] and the latter in [24].

A generalization consists in defining a new class C_4, in terms of excluded subgraphs of G obtained by applying some operations. A first such operation is an homomorphism, i.e., removal of degree 2 vertices: if $d_j = 2$ and the neighbors of v_j are v_i, v_k, remove v_j and replace its two incident edges by an edge joining v_i and v_k. Then G is *planar* if it contains no induced subgraph homomorphic to K_5 or $K_{3,3}$ (see below).

Theorem 9 (the four-color theorem, Appel, Haken [2] [3] [4])
If G is planar and has chromatic number $\chi(G)$ then

$$\chi(G) \leq 4.$$

This result was conjectured already in 1852 and was proved in 1976, with important computer aid; see also the more recent and shorter, but still computer-aided proof of Robertson *et al.* [80].

3 Conditions for belonging to a class of graphs

A second class of graph theoretic conjectures consists in necessary and/or sufficient conditions, expressed as algebraic relations, for a graph G to belong to a particular class C. Sufficient conditions appear most often. Their general form is

$$C \Leftarrow R$$

which reads:

"For any graph G, relation R implies G belongs to class C".

Necessary conditions have the form discussed in section 2, i.e., $C \Rightarrow R$. In rare cases, necessary and sufficient conditions are available: $C \Leftrightarrow R$. One can have also conditions valid only for some classes of graphs, e.g. $(C_1 \Leftarrow R) \mid C_2$. Recall that a graph is *Hamiltonian* if and only if there exists a cycle of G going once and only once through each vertex.

Theorem 10 *A graph G of order $n \geq 3$ with degree sequence $d_1 \leq d_2 \leq \ldots d_n$ is Hamiltonian if one of the following conditions holds:*
(i) (Dirac [40]) $d_k \geq \frac{n}{2}$ for all $k = 1, 2, \ldots n$;
(ii) (Ore [72] $d_u + d_v \geq n$ for all pairs of non adjacent vertices u,v;
(iii) (Pósa [77]) $d_k > k$ for all k with $1 \leq k \leq \frac{n}{2}$;
(iv) (Bondy [9]) $d_j + d_k \geq n$ for all j, k with $d_j \leq j, d_k \leq k - 1$.

Instead of a single relation R, one could have a conjunction or a disjunction of relations (as shown in the previous theorem, when the four conditions are taken jointly) or some more complicated logical combination of relations.

Relations of this form do not appear to have been much studied with computer-assisted or automated conjecture-making systems. One possible approach would be to consider conjectures which have not yet been refuted or proved, for some class C of graphs and test, on a database of examples or with an optimization routine, if one or several of them appear to be sufficient for G to belong to C.

Another approach would be to study conjectures valid for critical graphs related to the property defining C (*i.e.,* graphs G belonging to class C but who cease to be so if a vertex or an edge is removed), then to see if these conjectures hold for all graphs of C, or can be modified for this to be the case.

4 Inclusions between classes of graphs

A third class of graph-theoretic conjectures describes inclusion between classes $C_1, C_2, \ldots$ of graphs. The simplest form is then

$$C_1 \subseteq C_2$$

or, in rare cases,

$$C_1 \equiv C_2$$

which read

"All graphs of class C_1 belong to class C_2"

e.g.

"All trees are bipartite graphs"

and

"A graph belongs to class C_1 if and only if it belongs to class C_2"

e.g.

"A tree is a connected graph without cycles"

(this is sometimes taken as a definition but one can also use the following one: "A tree is a connected graph with $n-1$ edges").

Definitions of classes can be more general, *e.g.,* correspond to boolean expressions on simple classes of graphs or subgraphs in G, or possibly some graph derived from G by transformation such as removing vertices of degree 2.

Theorem 11 (Kuratowski [65]) *A graph G is planar if and only if it does not contain an induced subgraph homeomorphic to K_5 or $K_{3,3}$.*

The system *Graph Theorist* developed by Epstein [42] [43] [44] [45] represents classes of graphs by constructive definitions, *i.e.,* properties are associated with the classes of graphs satisfying them and algorithms are specified to construct (at least in principle) all graphs of these classes. Then inclusion among classes is studied leading to conjectures and their proof.

Such conjectures seldom appear to be new, the aim of Graph Theorist being more to understand mathematical reasoning than derive new results.

Relations of the above form do not appear to have been studied with other conjecture-making systems in graph theory.

5 Implications between relations

A further class of conjectures relates to implications and equivalences between relations $R_1, R_2, \ldots$, *i.e.,* they are of the form

$$R_1 \Rightarrow R_2$$

or

$$R_1 \Leftrightarrow R_2$$

Again these forms may be generalized to consider conjunctions, disjunctions or more complex logical expressions of several relations.

These forms are basic in mathematics and graph theory. They correspond to several problems:

5.1 Corollaries

The conjecture is then that corollary R_2 is a consequence of theorem R_1.

Conjecture 15 *The lower bound (*Berge *[6]) on the independence number $\alpha(G)$ of any graph G of order n and size m*

$$\alpha(G) \geq \frac{n^2}{2m+n}$$

is implied by the lower bound (Favaron *et al.* [52])

$$\alpha(G) \geq \left\lceil \frac{2n - \frac{2m}{\lceil \frac{2m}{n} \rceil}}{\lceil \frac{2m}{n} \rceil + 1} \right\rceil .$$

This is indeed the case, the latter bound being best possible for all n and m compatible with the existence of a simple graph.

Conjecture 16 [52] *The second relation in Conjecture 15 is equivalent to the following one (proposed earlier in* [59]*):*

$$\alpha(G) \geq \left\lceil n - \frac{2m}{1 + \lfloor \frac{2m}{n} \rfloor} \right\rceil + \left\lceil \frac{n - \lceil n - 2m/(1 + \lfloor \frac{2m}{n} \rfloor)(1 + \lfloor \frac{2m}{n} \rfloor) \rceil}{2 + \lfloor \frac{2m}{n} \rfloor} \right\rceil .$$

This conjecture is correct (but stated without proof in [52]).

Corroborating, refuting or strengthening conjectures such as the two last ones can be done in several ways:
(i) enumerating small graphs with systems such as Nauty or geng [69];
(ii) building interactively a counter-example, with a system such as GRAPH [33] [36];
(iii) minimizing the difference between the right hand-sides of both conjectures with AGX while parametrizing on n and m [21] [24].

5.2 Redundancy

If a relation R_2 is implied by a relation R_1 in a database, it may be viewed as redundant (and possibly deleted). Given R_1 and R_2, AGX is well-adapted to test a conjecture for redundancy: it will minimize (or maximize) the latter under the constraint that the former holds. This can be extended to testing a conjecture such as $R_1, R_2, \ldots R_k$ imply R_{k+1}, as well as to equivalence. However, this leads to refuting or corroborating one such conjecture not to finding it.

More generally,

> "When a new inequality relating graph invariants is discovered INGRID can be employed to determine if the same or better bounds can be obtained from previously known results" ([17] p.170).

To that effect, INGRID [17] can find among all relations of a large database if there is a small subset of them which imply a given relation. Thus given a set of relations $\Re = \{R_1, R_2, ...R_p\}$ and a relation R, INGRID discovers a statement of the form

$$R_{i_1} \cap R_{i_2} \cap \ldots R_{i_k} \;\Rightarrow\; R$$

where $k \ll p$. An example follows:

Conjecture 17 (Brigham *et al.* [17]) *The known relation between spectral radius λ_1, chromatic number χ and size m of a graph G*

$$\lambda_1 \leq \sqrt{2m\frac{(\chi-1)}{\chi}}$$

and

$$\chi \leq \lfloor 1 + \frac{1}{2}\sqrt{1+8m} \rfloor$$

imply the relation (Stanley [83])

$$\lambda_1 \leq -1 + \sqrt{1+8m}.$$

INGRID works as follows: it has built into it 458 relations between 37 graph invariants. The user can enter values or ranges of values for any of the invariants and INGRID then returns, using the relations, values or ranges of values for the remaining invariants. There is also a tracking function which allows the user to see the sequence of relations which led to the result, if desired.

INGRID may be used in interactive or in automated mode, i.e., in the latter case, after posing a question one just records the results in terms of values or intervals of values for invariants and of relations used.

It thus appears that the tools it uses for "helping to test the effectiveness of new theorems", as is discussed in this subsection, as well as for "helping derive theorems", which is discussed in the next subsection, are automated.

Brigham *et al.* comment as follows on the above example ([17] p.170):

> "With this insight we were able to show analytically that substitution of the second inequality into the first always produces a better bound than Stanley's except for one class of extremal graphs where they are equal. This in no way diminishes the value of Stanley's result, which gives an elegant direct relationship between λ_1 and e, but the exercise showed we need not include it in INGRID's knowledge base."

So, in this case, INGRID make a conjecture, which was later proved by hand. Observe that INGRID [17], as Graffiti's DALMATIAN heuristic ([49, p. 370]), does not include a relation in its database of relation if it is not informative. In the former case, this means it is implied by the union of all previous ones and in the latter case that this is true for the restricted set of graphs in the database of examples.

5.3 Paths towards new relations

The conjecture making function of INGRID just described can be extended to help finding new relations. Indeed, "INGRID does not of itself find new theorems relating graph

invariants, but it can be a valuable tool in aiding a researcher to do just that" ([17] p.170). Assuming an unknown but interesting relation exists between two invariants i_1 and i_2, one may vary one of them, observe the influence on the bounds of the other and use the tracking function to see which relations (implying quite different invariants than i_1 and i_2) are invoked by the system in computing these bounds. This leads to a conjecture of the form

"Relations $R_1, R_2, \ldots R_k$ in the database lead to a relation between invariants i_1 and i_2."

Then algebraic manipulations can be used to derive this relation, as illustrated by the next example:

Conjecture 18 (Brigham *et al.* [17]) *The relations*

$$\Delta \leq \lambda_1^2,$$

$$\nu \geq \frac{n}{\Delta - 1},$$

$$\epsilon \leq n - \nu$$

and

$$\theta_0 \leq \alpha$$

where the symbols are described above, except for the clique cover number $\theta_0 = \chi(\bar{G})$, imply relation(s) between λ and θ_0.
This indeed led to the relations

$$\theta_0 \leq n[\lambda_1^2/(1+\lambda_1^2)]$$

and

$$\theta_0 \leq \frac{1}{2} + [n(n-1) - \lambda_1(\lambda_1 - 1) + \frac{1}{4}]^2,$$

which could be proved and are new.

6 Structural conjectures

Many theorems in graph theory specify partially or completely the structure of some classes of graphs. In particular extremal graphs, *i.e.,* graphs for which an invariant takes its minimum or maximum value have been much studied, as shown in Bollobas' book [8] on that topic. Critical graphs have also received much attention.

Theorem 12 (Turan [84]): *If G is a graph of order n with independence number $\alpha(G)$, and minimum number of edges, then G is isomorphic to the graph $G_{n,k}$ composed of k disjoint cliques, r of which have q vertices and the others $k-r$ of which have $q-1$ vertices, where r and q are such that $n = q(k-1) + r$.*

This result has been generalized in many ways.

The energy of a graph has been defined above, and two lower bounds in terms of m and n given.

Conjecture 19: *For any graph G with energy $E(G)$, and size m the bound*

$$E(G) \geq 2\sqrt{m}$$

is attained if and only if G is complete bipartite.

This conjecture obtained with AGX, is proved in [21].

The Randic index of a graph has also been defined above.

Conjecture 20: *For any chemical tree T (with a maximum degree 4) of given size m, the Randic index is minimum if and only if it belongs to one of the three families represented in Figure 1 or is obtained from such a tree by iterated removal of three pending edges incident with a same vertex and their addition at another pending vertex.*

This conjecture, obtained with AGX, is proved in [23].

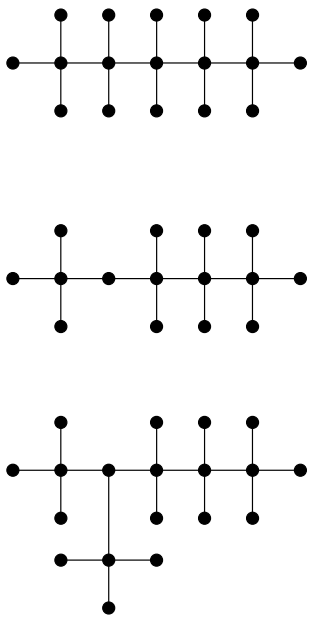


Figure 1: Three classes of chemical trees with minimum Randic index.

Dendrimers [41] are trees with a given maximum degree Δ which are as regular as possible (*i.e.,* regular except for pending vertices) and symmetric around one central vertex (see Figure 2a). It has long been surmised that:

Conjecture 21: [62] [63] *Dendrimers have minimum Wiener index (or total distance between pairs of vertices) among all trees with maximum degree Δ and the same order n.*

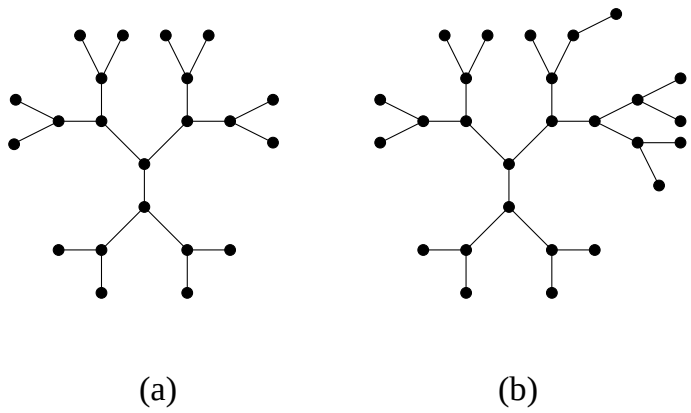


Figure 2: Dendrimers without and with additional edges.

AGX has corroborated this conjecture, and led to observe that if the number of edges does not correspond to that one of a dendrimer, additional edges should be as close as possible (see Figure 2b). This conjecture was recently proved, independently by Fischermann *et al.* [53] and by Zheng [86].

7 Counting and Enumerating

Many graph theoretic theorems give the number of graphs satisfying some specific property, often as a function of size, and sometimes provide also an implicit list of all such graphs. Another related type of problem is to find the minimum order of graphs which satisfy a given property. Computers have been extensively used in enumerative tasks from graph theory. They have led to many computer-assisted conjectures and proofs.

7.1 Counting graphs

A graph is labeled if its vertices are numbered $1, 2, \ldots n$. Two isomorphic graphs are viewed as different when their vertices are not labeled in the same way.

Theorem 13 (Cayley [27]): *There are n^{n-2} labeled trees on $n \geq 2$ vertices.*

An approach to finding conjectures of this type would be to enumerate all graphs satisfying a given property for $n = 1, 2, \ldots$ with a powerful system such as *geng* [69], then *(i)* to check if the resulting sequence of numbers is known with the Online Encyclopedia of Integer Sequences [82] ;
(ii) if not, use tools from algebra to study the sequence (and submit it to the Encyclopedia).

7.2 Enumerating graphs

Benzenoids are molecules which can be represented as planar polyhexes, *i.e.,* simply connected regions of the hexagonal lattice. They can also be viewed as graphs. Many algo-

rithms have been proposed for enumerating polyhexes with a given number h of hexagons (see [18] for a recent survey). The first few values are given in Table 1. However, no closed form formula for these series could be found.

h	N(h)	h	N(h)	h	N(h)
1	1	9	6505	17	1751594643
2	1	10	30086	18	8553649747
3	3	11	141229	19	41892642772
4	7	12	669584	20	205714411986
5	22	13	3198256	21	1012565172403
6	81	14	15367577	22	4994807695197
7	331	15	74207910	23	24687124900540
8	1435	16	359863778	24	122238208783203

Table 1: Number of planar polyhexes ($N(h)$) according to h

Conjecture 22: *There is no closed-form formula giving the number of polyhexes with h hexagons.*

While this conjecture could be refuted, it is hard to see how to prove it.

8 Ramseyian Theorems and Conjectures

Conjectures considered up to now are expressed in terms of invariants of a graph G and structure of such a graph. Another class of results is less direct: one considers a property which must hold for all partitions of a given type defined on G, most frequently all colorings of its edges using a given number of colors. Then the effect of the imposition of this property on an invariant $i(G)$, most often its order, is studied. To illustrate let us consider all bicoloring of the edges of G. The classical Ramsey number $r(k)$ is the smallest order of a graph G such that all such bicolorings induce a K_k in G or in $\bar{G}$.

Very few Ramsey numbers are known [30], so generalized Ramsey numbers in which one considers a subgraph G_1 in G or G_2 in $\bar{G}$ have been extensively studied. Computer enumeration played an important role: in a recent version of his "Dynamic Survey" on "Small Ramsey Numbers", Radzizowski [78] cites 71 papers which report on automated or computer-assisted determination of generalized Ramsey numbers or bounds on them. In this last case, conjectures are sometimes made on what is the most likely value.

More general questions have been asked, often by Erdös and his collaborators.

Conjecture 23 (Burr, Erdös [19]) *For every graph G on n vertices in which every subgraph has average degree at most c,*

$$r(G) \leq c'n$$

where the constraint c' depends only on n.

A conjecture of the same form for subgraphs with maximum degree Δ,

$$r(G) \leq c(\Delta)n$$

was made by the same authors and proved to hold by Chvatal *et al.* [31].

An example in which edge 3-colorings are considered is the following:

Conjecture 24 (Bondy and Erdös [11]) *Let C_p be a cycle with p vertices; then*

$$r(C_p, C_p, C_p) \leq 4p - 3.$$

Luczak [68] has shown that $r(C_p, C_p, C_p) \leq 4p + o(p)$.

Other problems concern the number of classes in a family of partition defined on a graph G.

Conjecture 24 (Erdös, Gallai, 1959 [46]) *Every connected graph on n vertices can be edge-partitioned into almost $\lfloor (n+1)/2 \rfloor$ paths.*

Instead of partitions of edges of G, one may also consider all subgraphs of G of a given type, such as, e.g. cliques. This leads to new questions, e.g.:

Problem 1 (Erdös *et al.* 1992 [47]) *Estimate the cardinality, denoted by $T(G)$, of a smallest set of vertices in G that shares some vertex with every maximal clique of G.*

While computers do not appear to have been used in the study of this problem, it seems that a specialized algorithm could prove useful.

9 Conclusions

In order to get a clear view of what are interesting conjectures in graph theory, we followed up on the observation that famous theorems in this field (as in others) were first conjectures, if only in the minds of those which proved them. This suggests a rich variety of forms. We attempted to classify them, taking into account the work done in computer-assisted or automated conjecture-making. Thus we could provide examples of a number of cases in which one or another system was successful.

Moreover, it appears that

(i) there are many classes of conjectures which have not yet been explored with or by conjecture-making systems (the more so as the present classification is exploratory and certainly not exhaustive).

(ii) different systems appear to each have their strong points and none seems presently able to obtain interesting conjectures in all the cases where the others do.

Therefore, there is much work to do, both in modifying existing systems for doing in different ways tasks done by others and expanding them to tackle new conjecture-making tasks. Clearly, while computer-assisted and automated conjecture-making is successful, the field is still at its beginning.

References

[1] M. O. Alberston. The irregularity of a graph. *Ars Combinatoria*, 46:215–225, 1997.

[2] K. Appel and W. Haken. Every planar map is four colorable. part i. discharging. *Illinois Journal of Math.*, 21:429–490, 1977.

[3] K. Appel and W. Haken. Every planar map is four colorable. part ii. reducibility. *Illinois Journal of Math.*, 21:491–567, 1977.

[4] K. Appel and W. Haken. Every planar map is four colorable. *A.M.S. Contemp. Math.*, 98:1–743, 1989.

[5] R. A. Beezer, J. Riegsecker, and B. A. Smith. Using minimum degree to bound average distance. *Discrete Mathematics*, 226:365–377, 2001.

[6] C. Berge. *Graphes et Hypergraphes.* Dunod, Paris, 1970.

[7] N. Biggs. *Algebraic Graph Theory.* Combridge University Press, 1974.

[8] B. Bollobas. *Extremal graph theory*, volume 11. London Mathematical Society Monographs, 1978.

[9] J. A. Bondy. Properties of graphs with constraints on degrees. *Studia Sc. Math. Hung.*, 4:473–475, 1969.

[10] J. A. Bondy. Pancyclic graphs. i. *J. Combin. Theory Ser. B*, 11:80–84, 1971.

[11] J. A. Bondy and P. Erdös. Some new problems and results in graph theory and other branches of combinatorial mathematics. *Lect. Notes in Mathematics*, 885:9–17, 1981.

[12] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications.* McMillan, London, 1972.

[13] A. Bouvier and M. George. *Dictionnaire des Mathematiques, (french).* Paris, Presses Universitaires de France, 1979.

[14] R.C. Brigham and R.D. Dutton. A Compilation of Relations between Graphs Invariants. *Networks*, 15:73–107, 1985.

[15] R.C. Brigham and R.D. Dutton. Bounds on the domination number of a graph. *Quart. J. Math. Oxford Ser. 2*, 41:269–275, 1990.

[16] R.C. Brigham and R.D. Dutton. A Compilation of Relations between Graphs Invariants. Supplement 1. *Networks*, 21:421–455, 1991.

[17] R.C. Brigham, R.D. Dutton, and F. Gomez. INGRID. A graphs invariant manipulator. *J. Symb. Comp.*, 7:163–177, 1989.

[18] G. Brinkmann, G. Caporossi, and P. Hansen. A survey and new results on computer enumeration of polyhex and fusene hydrocarbons. *Les Cahiers du GERAD*, G-2002-17, 2002.

[19] S.A. Burr and P. Erdös. On the magnitude of generalized Ramsey numbers for graphs. In: *Infinite and Finite Sets*, Vol. 1, *Colloq. Math. Soc. Janos Bolyai*, 10:219–240, Amsterdam: North-Holland, 1975.

[20] R. G. Busacker and T. L. Saaty. *Finite Graphs and Networks*. McGraw-Hill Book Company, 1965.

[21] G. Caporossi, D. Cvetković, I. Gutman, and P. Hansen. Variable Neighborhood Search for Extremal Graphs. 2. Finding Graphs with Extremal Energy. *J. Chem. Inf. Comput. Sci.*, 39:984–996, 1999.

[22] G. Caporossi, A.A. Dobrynin, I. Gutman, and P. Hansen. Trees with Palindromic Hosoya Polynomials. *Graph Theory Notes of New-York*, 37:10–16, 1999.

[23] G. Caporossi, I. Gutman, and P. Hansen. Variable Neighborhood Search for Extremal Graphs. 4. Chemical Trees with Extremal Connectivity Index. *Computers and Chemistry*, 23:469–477, 1999.

[24] G. Caporossi and P. Hansen. Variable neighborhood search for extremal graphs 5: Three ways to automate finding conjectures. *Discrete Mathematics*, 2003 (in press).

[25] G. Caporossi and P. Hansen. Finding Relations in Polynomial Time. In *Proceedings of the XVI International Joint Conference on Artificial Intelligence*, pages 780–785, 1999.

[26] G. Caporossi and P. Hansen. Variable Neighborhood Search for Extremal Graphs. 1. The Autographix System. *Discrete Mathematics*, 212:29–44, 2000.

[27] A. Cayley. A theorem on trees. *Quart. J. Math.*, 23:376–378, 1889.

[28] S.C. Chou. *Mechanical Geometry Theorem Proving*. Mathematics and its Applications, 41, Dordrecht: Reidel, 1988.

[29] F.R.K. Chung. The average distance and the independence number. *Journal of Graph Theory*, 12:229–235, 1988.

[30] F. Chung, and R. Graham. *Erdös on Graphs. His Legacy of Unsolved Problems*. A.K. Peters, Natic, Massachusetts, 1999.

[31] V. Chvatal, V. Rödl, E. Szemeredi and W.T. Trotter. The Ramsey number of a graph with bounded maximum degree. *J. Combinatorial Theory* B 39:239–243, 1983.

[32] S. Colton. Refactorable numbers - a machine invention. *Journal of Integer Sequences*, 2, 1999.

[33] D. Cvetković. *"Graph" an Expert System for the Classification and Extension of the Knowledge in the Field of Graph Theory, User's Manual*. Elektrothn. Fak. Beograd, 1983.

[34] D. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs - Theory and Applications*. Academic Press New York, 1980.

[35] D. Cvetković and I. Gutman. The computer system graph: a useful tool in chemical graph theory. *Comput. Chem.*, 7:640–644, 1985.

[36] D. Cvetković and S. Simić. Graph theorical results obtained with support of the expert system "graph" -an extended survey-. submitted.

[37] D. Cvetković, S. Simić, G. Caporossi, and P. Hansen. Variable Neighborhood Search for Extremal Graphs. 3. On the Largest Eigenvalue of Color–Constrained Trees. *Linear and Multilinear Algebra*, 49:143-160, 2001.

[38] E. DeLaVina, S. Fajtlowicz, and B. Waller. On conjectures of Griggs and Graffiti. preprint (2002).

[39] E. DeLaVina. Some History of the Development of Graffiti, preprint. 2003.

[40] G. A. Dirac. Some theorems on abstract graphs. *Proc. London Math. Soc.*, 2:69–81, 1952.

[41] A. A. Dobrynin, R. Entringer, and I. Gutman. Wiener index of trees: Theory and applications. *Acta Applicandae Mathematicae*, 66:211–249, 2001.

[42] S. L. Epstein. Ph.D. Thesis, Rutgers University, 1983.

[43] S. L. Epstein. On the discovery of mathematical theorems. *In Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 194–197, 1987.

[44] S. L. Epstein. Learning and discovery: One system's search for mathematical knowledge. *Comput. Intell.*, 4:42–53, 1988.

[45] S. L. Epstein and N. S. Sridharan. Knowledge presentation for mathematical discovery: Three experiments in graph theory. *J. Applied Intelligence*, 1:7–33, 1991.

[46] P. Erdös, and T. Gallai. On maximal paths and circuit of graphs. *Acta Math. Acad. Sci. Hungarica*, 10:337–356, 1959.

[47] P. Erdös, T. Callai, and Z. Tuza. Covering the cliques of a graph with vertices. In: *Topological, Algebrical and Combinatorial Structures Frolik's Memorial Volume. Discrete Mathematics*, 108:279–289, 1992.

[48] S. Fajtlowicz. On Conjectures and Methods of Graffiti. *Proceedings of the Fourth Clemson Mini-Conference on Discrete Mathematics*, Clemson, 1989.

[49] S. Fajtlowicz. On conjectures of Graffiti – V. In *Seventh International Quadrennial Conference on Graph Theory*, 1, 367–376, 1995.

[50] S. Fajtlowicz. Written on the wall. version 03-1997 (updated regularily), 1997.

[51] O. Favaron, M. Mahéo, and J-F. Saclé. On the residue of a graph. *Journal of Graph Theory*, 15:39–64, 1991.

[52] O. Favaron, M. Mahéo, and J-F. Saclé. Some eigenvalue properties in graphs (conjectures of graffiti. ii). *Discr. Math.*, 111:197–220, 1993.

[53] M. Fischermann, A. Hoffman, D. Rautenbach, L. Szekely, and L. Vollmann. Wiener index versus maximum degree in trees. *Discrete Applied Mathematics*, 122:127–137, 2002.

[54] K. Fraughnaugh and S. C. Locke. 11/30 (finding large independent sets in connected triangle-free 3-regular graphs). *J. Combin. Theory Ser. B*, 65 no. 1:51–72, 1995.

[55] T. Gallai. Maximum – minimum Safze uber Graphen (German). *Acta Mth. Acad. Sci. Hungarica*, 9:395–434, 1959.

[56] I. Gutman. Total π-electron energy of benzenoid hydrocarbons. *Topics in Current Chemistry*, 162:29–63, 1992.

[57] I. Gutman and S.J. Cyvin. *Introduction to the Theory of Benzenoid Hydrocarbons.* Springer-Verlag, 1989.

[58] I. Gutman, P. Hansen, and H. Mélot. Variable neighborhood search for extremal graphs. 10. Comparing measures of irregularity for chemical trees. in preparation, 2002.

[59] P. Hansen. Degrés et nombre de stabilité d'un graphe. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 17:213–220, 1975.

[60] P. Hansen. How far is, should, is and could be conjecture-making in graph theory and automated process. submitted, 2002, revised 2003.

[61] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs.* Dekker, New York, 1998.

[62] S. L. Lee, I. Gutman, Y. N. Yeh and J. C. Chen. Wiener numbers of dendrimers. *Match-Comm. Math. Chem.*, 30:103–115, 1994.

[63] S. L. Lee, I. Gutman, Y. N. Yeh and Y. L. Luo. Some recent results in the theory of Wiener number. *Indian J. Chem.*, 32 A:651–661, 1993.

[64] D. König. Graphs and matrices. (hungarian). *Mat. Fiz. Lapok*, 38:116–119, 1931.

[65] C. Kuratowski. Sur le problème des courbes gauches en topologie. (french). *Fund. Math.*, 5:271–283, 1930.

[66] I. Lakatos. *Proofs and Refutations.* Cambridge University Press, Cambridge, 1976.

[67] C. Larson. Intelligent machinery and mathematical discovery. *Graph Theory Notes of New York*, XLII:8–17, 2002.

[68] T. Luczak. $R(C_n, C_n, C_n) \leq (4+0(1))n$. *Journal of Combinatorial Theory B*, 75:179–187, 1999.

[69] B.D. McKay. nauty user's guide (version 1.5). Technical Report. TR-CS-90-02, Department of Computer Science, Australian National University, 1990.

[70] E. A. Nordhaus and J. W. Gaddum. On complementary graphs. *Amer. Math. Monthly*, 63:175–177, 1956.

[71] R. Z. Norman and M. O. Rabin. An algorithm for a minimum cover of graph. *Proc. Amer. Math. Soc.*, 10:315–319, 1959.

[72] O. Ore. Arc covering of graphs. *Ann. Math. Pura Appl.*, 55:315–321, 1961.

[73] G. Polya. *Mathematical Discovery. On Understanding, Learning and Teaching Problem Solving.* Combined edition, Wiley, New-York, 1962.

[74] G. Polya. *Mathematics and Plausible Reasoning, Volume 1. (Induction and Analogy in Mathematics).* Princeton University Press, Princeton, 1954.

[75] G. Polya. *Mathematics and Plausible Reasoning, Volume 2. (Patterns of Plausible Inference).* Princeton University Press, Princeton, 1954.

[76] K. Popper. *The Logic of Scientific Discovery*. Hutchinson, London, 1959.

[77] O. Pósa. A theorem covering hamilton lines. *Magyar Tud. Akad. Mat. Kutato Int Zözl.*, 7:225–226, 1962.

[78] S.P. Radzizowski. Small Ramsey numbers. Dynamic survey 1. *Electronic Journal of Combinatorics*, 1994. Updated 1998.

[79] M. Randic. On characterization of molecular branching. *Journal of the American Chemical Society*, 97:6609–6615, 1975.

[80] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. The four-colour theorem. *Journal of Combinatorial Theory, Ser. B*, 70:2–44, 1997.

[81] P.A. Samuelson and W.D. Nordhaus. *Economics*. Irwin McGraw-Hill, 1998 (16th edition).

[82] N. Sloane. The on-line encyclopedia of integer sequences. http://www.research.att.com/minjas/sequences/

[83] R.P. Stanley. A bound on the spectral radius of graphs with e edges. *Linear Algebra and Applications*, 87:267–289, 1987.

[84] P. Turán. An extremal problem in graph theory. *Mat. Fiz. Lapok*, 48:436–452, 1941.

[85] V.G. Vizing. On an estimate of the chromatic class of a p-graph. (russian). *Metody Diskret. Analiz.*, 3:25–30, 1964.

[86] M. Zheng. Minimum total distance d-trees. Presentation at the DIMACS Workshop on *Computer-Generated Conjectures from Graph-Theoretic and Chemical Databases*, November 12–16, 2001.

Appendix. Proof of Conjecture 2

The trees with maximum degree $\Delta \leq 3$ found by *AGX* with (conjectured) maximum irregularity are represented on Figure 3.

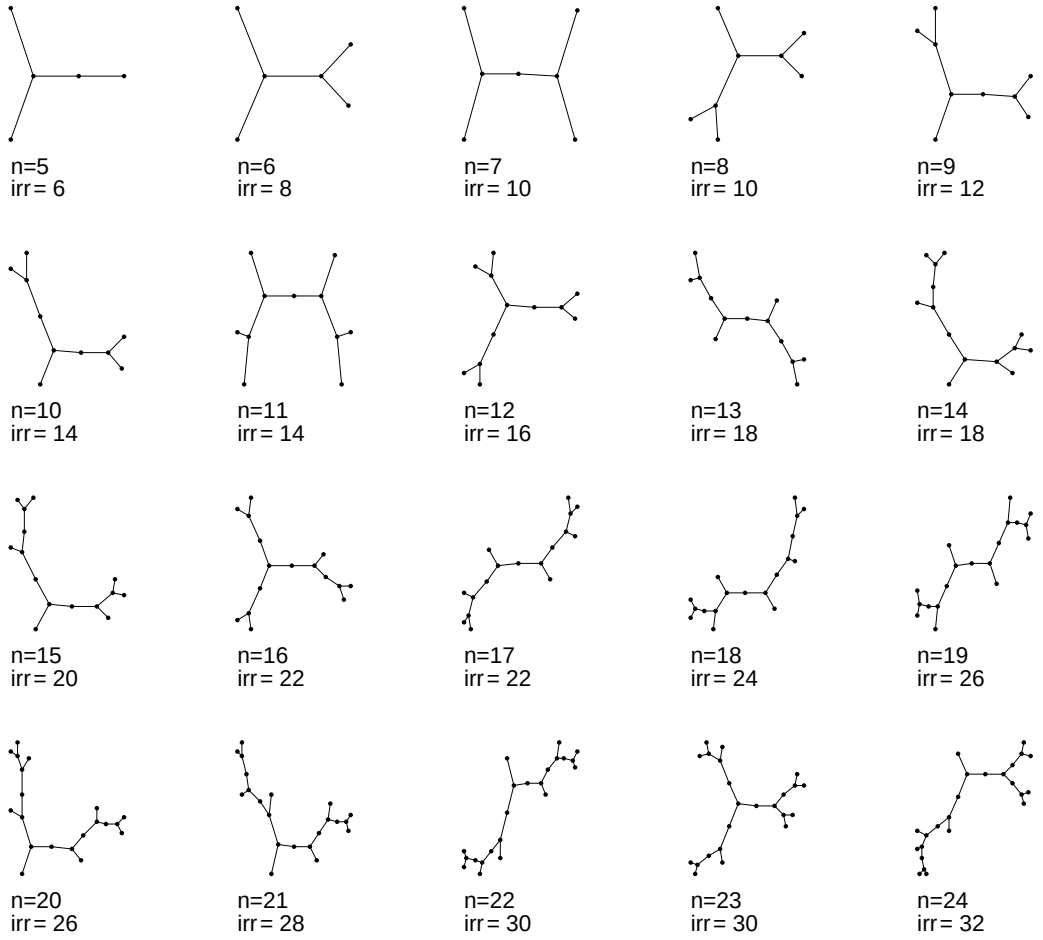


Figure 3: Extremal trees with $\Delta \leq 3$ and maximum irregularity found by *AGX*

These extremal trees are used in the following proofs, illustrating also the help provided by *AGX* in getting proofs.

Theorem 14 *For any tree T with $\Delta \leq 3$,*

$$\begin{aligned} irr(T) &\leq \tfrac{4n+2}{3} && \text{if } n(mod\ 3) = 1, \\ &\leq \tfrac{4n-n(mod\ 3)}{2} && \text{otherwise.} \end{aligned}$$

Proof. Let T be a tree with maximum degree $\Delta \leq 3$ and denote by x_{ij} the number of edges of T with endvertices of degree i and j.

By definition of the irregularity,

$$irr(T) = x_{12} + 2x_{13} + x_{23}. \tag{9.1}$$

We first solve the following system of five linear equations which holds for all trees with $\Delta \leq 3$:

$$\begin{aligned} x_{12} + x_{13} &= n_1 && (9.2)\\ x_{12} + 2x_{22} + x_{23} &= 2n_2 && (9.3)\\ x_{13} + x_{23} + 2x_{33} &= 3n_3 && (9.4)\\ n_1 + 2n_2 + 3n_3 &= 2n - 2 && (9.5)\\ n_1 + n_2 + n_3 &= n. && (9.6) \end{aligned}$$

with unkowns x_{13}, x_{23}, n_1, n_2 and n_3. That gives :

$$\begin{aligned} x_{13} &= \frac{1}{3}(n - 4x_{12} - x_{22} + x_{33} + 5) && (9.7)\\ x_{23} &= \frac{1}{3}(2n + x_{12} - 2x_{22} - 4x_{33} - 8) && (9.8)\\ n_1 &= \frac{1}{3}(n - x_{12} - x_{22} + x_{33} + 5) && (9.9)\\ n_2 &= \frac{1}{3}(n + 2x_{12} + 2x_{22} - 2x_{33} - 4) && (9.10)\\ n_3 &= \frac{1}{3}(n - x_{12} - x_{22} + x_{33} - 1). && (9.11) \end{aligned}$$

Replacing x_{13} by (9.7) and x_{23} by (9.8) in (9.1) gives

$$irr(G) = \frac{1}{3}(4n - 4x_{12} - 4x_{22} - 2x_{33} + 2) \tag{9.12}$$

which is maximal for a fixed number of vertices when the values x_{12} and x_{33} are equal to zero.

If $n(\text{mod } 3) = 1$, we can choose $x_{12} = 0$, $x_{22} = 0$ and $x_{33} = 0$ because the solutions given in Eqs. (9.7) – (9.11) are in integers. In this case, $x_{13} = (n+5)/3$, $x_{23} = (2n-8)/3$ and $irr(T) = (4n+2)/3$.

If $n(\text{mod } 3) = 0$, x_{12}, x_{22} and x_{33} cannot be all equal to zero because the solutions are no more in integers. Looking at (9.12), the best choice is to take $x_{12} = x_{22} = 0$ and $x_{33} = 1$ which is a feasible case. In this case, $irr(T) = 4n/3$.

If $n(\text{mod } 3) = 2$, there are three feasible solutions with the same irregularity value. One can choose $x_{12} = x_{22} = 0$ and $x_{33} = 2$, or $x_{12} = 1$ and $x_{22} = x_{33} = 0$, or $x_{22} = 1$ and

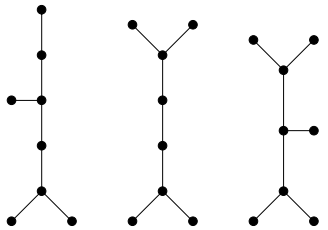


Figure 4: Three trees with maximum irregularity, $\Delta \leq 3$ and $n = 8$

$x_{12} = x_{33} = 0$. These solutions lead to $irr(T) = (4n-2)/3$. Figure 4 shows three different trees with maximum irregularity and $n = 8$. □

The graphs found by *AGX* (see Figure 3) are extremal for the irregularity by Theorem 14. The proof of this theorem gives a good characterization of these graphs in terms of x_{ij}. We now prove Conjecture 2, which was obtained automatically by *AGX* from these extremal trees.

Theorem 15 *For any tree T of size m with $\Delta \leq 3$ and maximum irregularity $irr(T)$, Randic index $Ra(T)$, and n_1 pending vertices,*

$$Ra(T) = -0.027421\ irr(T) + 0.538005\ m - 0.110484\ n_1 + 0.614014.$$

Proof. Before proceeding to the proof itself, we find which real values *AGX* has approximated. To do this, we choose 4 extremal trees given by the system (see Figure 5), compute their values for Ra, irr, m and n_1 and subsitute these values in

$$Ra = a\ irr + b\ m + c\ n_1 + d \tag{9.13}$$

where a, b, c, d are the real values sought for. For instance, the tree T_1 on Figure 5 has $Ra(T_1) = 1/\sqrt{2} + 2/\sqrt{3} + 1/\sqrt{6}$, $irr(T_1) = 6$, $m(T_1) = 4$ and $n_1(T_1) = 3$. That gives the following system of equations with unknows a, b, c and d :

$$\begin{aligned}
6a + 4b + 3c + d &= \frac{1}{\sqrt{2}} + \frac{2}{\sqrt{3}} + \frac{1}{\sqrt{6}}, && (9.14)\\
8a + 5b + 4c + d &= \frac{4}{\sqrt{3}} + \frac{1}{3}, && (9.15)\\
10a + 6b + 4c + d &= \frac{4}{\sqrt{3}} + \frac{2}{\sqrt{6}}, && (9.16)\\
10a + 7b + 5c + d &= \frac{5}{\sqrt{3}} + \frac{2}{3}. && (9.17)
\end{aligned}$$

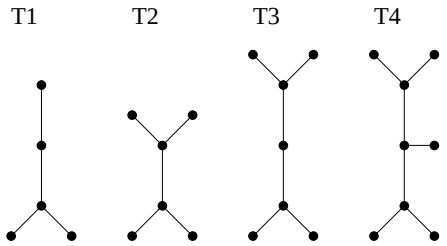


Figure 5: Four extremal trees with $\Delta \leq 3$ and maximum irregularity found by *AGX*

The unique solution of this system is

$$a = -\frac{\sqrt{2}}{4} + \frac{\sqrt{3}}{6} + \frac{\sqrt{6}}{12} - \frac{1}{6}, \tag{9.18}$$

$$b = \frac{\sqrt{2}}{2} - \frac{\sqrt{3}}{3} + \frac{\sqrt{6}}{6}, \tag{9.19}$$

$$c = -\frac{\sqrt{2}}{2} + \frac{2\sqrt{3}}{3} - \frac{\sqrt{6}}{2} + \frac{2}{3}, \tag{9.20}$$

$$d = \frac{3\sqrt{2}}{2} - \sqrt{3} + \frac{\sqrt{6}}{2} - 1. \tag{9.21}$$

A numerical approximation of these irrational values corresponds to the values given by *AGX* in the conjecture.

Let T be a tree with maximum degree $\Delta \leq 3$. We have that

$$m = x_{12} + x_{13} + x_{22} + x_{23} + x_{33}, \tag{9.22}$$

and

$$n = x_{12} + x_{13} + x_{22} + x_{23} + x_{33} + 1. \tag{9.23}$$

Moreover, by definition of the irregularity

$$irr(T) = x_{12} + 2x_{13} + x_{23}, \tag{9.24}$$

and by definition of the Randic index

$$Ra(T) = \frac{x_{12}}{\sqrt{2}} + \frac{x_{13}}{\sqrt{3}} + \frac{x_{22}}{2} + \frac{x_{23}}{\sqrt{6}} + \frac{x_{33}}{3}. \tag{9.25}$$

By Theorem 14, if $n(\text{mod } 3) = 1$,

$$x_{13} = (n+5)/3, \tag{9.26}$$

and

$$x_{12} = x_{22} = x_{33} = 0. \tag{9.27}$$

Substituting (9.27) in (9.23) and (9.23) in (9.26) gives

$$x_{23} = 2x_{13} - 6. \tag{9.28}$$

By (9.27) and (9.28), Eqs. (9.22), (9.24) and (9.25) become

$$m = 3x_{13} - 6, \tag{9.29}$$

$$irr(T) = 4x_{13} - 6, \tag{9.30}$$

and

$$Ra(T) = x_{13}\frac{\sqrt{3}+\sqrt{6}}{3} - \sqrt{6}, \tag{9.31}$$

respectively. Moreover, Eq. (9.2) gives

$$n_1 = x_{13} \tag{9.32}$$

Replace irr by (9.30), m by (9.29), n_1 by (9.32) and a, b, c, d by Eqs. (9.18) – (9.21) in the right-hand-side of (9.13) and simplify. This leads to

$$x_{13}\frac{\sqrt{3}+\sqrt{6}}{3} - \sqrt{6},$$

which is equal to the Randic index of T given by (9.32).

The other cases are similar.

If $n(\text{mod } 3) = 0$, we start with $x_{13} = (n+6)/3$, $x_{33} = 1$ and $x_{12} = x_{22} = 0$ and modify the remainder of the proof in consequence.

If $n(\text{mod } 3) = 2$ we start with the three different solutions given in Theorem 14 and apply the same ideas in each case.

□

Les Cahiers du GERAD ISSN: 0711–2440

How Far Is, Should and Could Be Conjecture-Making in Graph Theory an Automated Process ?

Pierre Hansen

G-2002-44

August 2002
Revised: August 2003
Draft. Do not cite without the author's permission.

How Far Is, Should and Could Be Conjecture-Making in Graph Theory an Automated Process ?

Pierre Hansen

GERAD
and
École des Hautes Études Commerciales
Montréal

August, 2002
Revised: August, 2003

Les Cahiers du GERAD
G–2002–44

Abstract

Computer-assisted and automated conjecture-making in graph theory is reviewed, focusing on the three operational systems GRAPH, Graffiti and AutoGraphiX (AGX). A series of possible enhancements, mostly through hybridisation of these systems, are proposed as well as several research paths for development of the area.

Keywords: graph, conjecture, computer-assisted, automated.

Résumé

On passe en revue la génération de conjectures assitée par ordinateur et automatisée en théorie des graphes, en considérant plus particulièrement les trois systèmes opérationnels GRAPH, Graffiti et AutoGraphiX (AGX). Une série d'améliorations possibles, le plus souvent par hybridation de ces systèmes, sont proposées ainsi que plusieurs voies de recherche pour le développement du domaine dans son ensemble.

Mots clés: graphe, conjecture, assisté par ordinateur, automatisé.

1 Introduction

1.1 Conjectures

Roget's New thesaurus [147] defines *conjecture* as

> "a judgment, estimate or opinion arrived at by guessing: guess, guesswork, speculation, supposition, surmise".

So, uncertainty is stressed. In mathematics, the word "conjecture" has a more precise meaning. In their *Dictionary of Mathematics*, Bouvier and George [22] define it as follows:

> "Conjecture: An *a priori* hypothesis on the exactness or falseness of a statement of which one ignores the proof".

Knowledge should back this hypothesis, and make the conjecture valuable, as stressed by Mac Lane [128]:

> "Conjecture has long been accepted in mathematics, but the customs are clear. If a mathematician has really studied the subject and made advances therein, then he is entitled to formulate an insight as a conjecture, which usually has the form of a specific proposed theorem. Riemann, Poincaré, Hilbert, Mordell, Bieberbach, and many others have made such deep conjectures".

Further examples of important conjectures, this time in graph theory, are the four-color conjecture [149], proved in 1976 by Appel and Haken [7] [8] [9] (see also the more recent proof of Robertson *et al.* [146]) and the strong perfect graph conjecture of Berge [16] [17] very recently proved by Chudnovsky *et al.* [47] [48]. The former may have initially been a happy guess of a student, Francis Guthrie, but was popularized by a major mathematician, Augustus de Morgan. Its (correct) proof took 125 years and was computer-assisted. No proof without partial automation is known. The latter was proposed in 1961 by one of the most prominent graph theorist of the time. It took 41 years and the work of scores of mathematicians to be finally proved (without computer assistance). So, conjecturing supposes knowledge and insight. Guessing is easy but conjecturing is hard; we will see that this holds for computers as for humans.

1.2 Automation

Again according to Mac Lane [128], the sequence for the understanding of mathematics may be:

> "Intuition, Trial, Error, Speculation, Conjecture, Proof".

Proof is the ultimate goal and has attracted the most attention, including in attempts to automate mathematics. Yet, it is far from the whole story. Hardy [114] reminds us that

> "All physicists and a good many quite respectable mathematicians are contemptuous about proof".

Discovering interesting or beautiful conjectures, even if someone else proves (or refutes) them, is of importance.

Clearly, theorems are first conjectures, possibly known as such only to those who prove them. Often, only the final result, i.e., the theorem, is published. The discovery process is not explained, and further discoveries may be made more difficult than necessary. A few mathematicians and philosophers of science have focused on this process. Prominent among them are Euler, Polya [138] [139] and Lakatos [123]. Recent studies of the application of Popper's ideas in mathematics [140] and their development are also of interest [94] [95].

Automated theorem proving is a well-developed field, with numerous researchers, tens of books and a rapidly increasing record of successes [162]. A good example is the recent 16-line automated proof by Mc Cune [133] of the Robbins conjecture:

"All Robbinsonian algebras are Boolean algebras",

which had been open for 63 years.

In graph theory, only simple propositions can at present be proved in an entirely automated way (see Section 2 for a brief discussion). Computer-assisted proofs, mostly based on enumeration routines, are becoming common. To illustrate, the survey of Radzizowski [145] on small Ramsey numbers mentions computer-assisted results from 71 papers.

In contrast, computer-assisted and automated conjecture-making in mathematics, the mathematical branch of discovery science, has attracted few researchers up to now, despite some notable successes. Outside of graph theory one may mention the important work of several mathematicians on integer relation detection [115] [11] [21]. This led, among other applications to Apéry-like formulae for $\zeta(4n+3)$, new Euler sums and formulae for various constants including one for π with the astonishing consequence that one can compute, in base 2, the digits of π beginning at any place (e.g. from the trillionth' one) without knowing the previous ones. While the important work of Wu [163] [164], and Chou *et al.* [42] [44] [45] in plane geometry is mostly aimed at automating proofs, it includes conjecture-making routines. This led to the discovery of new families of Pascal conics [44]; recently a procedure for finding *all* relations implied by a given configuration of lines and curves in the plane has been obtained [45]. Hájek and Havránek [100] [101] and Hájek and Holeňa [102] have studied mathematical formulations for a general theory of the mechanization of hypothesis formation. They introduce formal logics for that purpose and may have been one of the sources of inspiration for the system GRAPH discussed below.

Graph-theoretic work will be discussed throughout this paper. This will be done by a study and discussion of some of the best developed systems, no general mathematical framework for making conjectures in graph theory being in current use. But a preliminary question should be addressed:

How far should conjecture-making in graph theory be automated?

Langley [121] comments as follows on discovery science systems:

> "Although the term *computational discovery* suggests an automated process, close inspection of the literature reveals that the human developer or user plays an important role in any successful project. Early computational research on scientific discovery downplayed this fact and emphasized the automation aspect

> in general keeping with the goals of artificial intelligence at the time. However, the new climate in AI systems that advise humans rather than replace them, and recent analyses of machine learning applications... suggest an important role for the developer".

Two things should be distinguished here: on the one hand, that knowledge due to the developer, and possibly many others (e.g. numerous algorithms for computing graph theoretic invariants) is embedded in the system appears to be necessary to obtain conjectures; on the other hand, that the user may interact or not with the system, leading to computer-assisted or to automated discoveries.

In graph theory, three additional reasons may be adduced for preferring computer-assisted systems to automated ones:

(i) the difficulty of automation may limit the scope of problems addressed;

(ii) the ultimate goal being proof, interaction with the system is more likely to lead to insights about how to prove the conjectures found than just reading their statement;

(iii) such interaction may also be very fruitful from the pedagogical point of view. This question will be discussed further in Section 3.

However, automating a system for making conjectures in graph theory is a challenge, and may lead to original ways of addressing this problem. Moreover, comparison of this work with the treatment of similar problems within close fields such as automated theorem proving or data mining, may foster cross-fertilization.

This author's view is that both computer-assisted and automated conjecture-making are of interest; in this paper, the main focus is on the latter.

1.3 Definitions

We will adopt the following terminology: an *automated system* will be synonymous with a *fully* automated one, and this means that

(i) *input should be limited to the problem statement* which implies further information on the problem or closely related ones cannot be introduced *at that time*, but may of course already belong to one or another of the systems databases;

(ii) *there should be no human intervention between problem statement and output of the results*;

(iii) *output of the results should be the final step*, which implies there should be no human selection of those conjectures about the problem under study which are publicized; of course the users are then free to choose those they will try to prove.

Otherwise, the system will be called *computer-assisted*.

This is in keeping with usual practice. To illustrate points (i) and (ii), "Deep Blue" [30] [118] is an automated system which includes considerable knowledge about chess-playing (and Kasparov's way to play chess) due to the developers. In competition it can be tuned before a game but not when this game is in process, and it only receives notice of the opponent's moves [118].

To illustrate point (iii) observe that some researchers (e.g. [120]) claim that computers can compose poetry and try to make their point by selecting among a large output, obtained by their system from some poets vocabulary, usually very short "poems" which appear to make sense. If one generates a sufficiently large number of "poems" and selects drastically among them, this is bound to work (to some extent), but the conclusion is far from clear.

When an automated system makes conjectures we will say they are obtained *by* the system; when a computer-assisted system does so, we will say the conjectures are obtained *with* the system.

Refuting or corroborating conjectures known beforehand with a computerized system will be referred to as *testing* them; conjectures which are corroborated may be improved (e.g. stronger bounds may be considered) and this will be called *strengthening* them; finding new conjectures will be called *conjecture-making*, and can be unassisted (or done by hand), computer-assisted or automated.

To the best of our knowledge, present systems for conjecture-making in graph-theory are either computer-assisted or can be used both in computer-assisted mode and, in rare cases, in automated mode. The question of whether one computer-assisted system is more automated than another cannot be answered in a clear-cut way as different systems perform different tasks. Therefore, we will describe these tasks, state which of them are automated and how, which are not and how they are done, and let the reader judge.

About half a dozen systems for conjecture-making in graph theory and other close purposes have been developed. We distinguish between *experimental* and *operational* systems. An experimental system explores an idea, without necessarily leading to new results (or to just a few, due to its developers); its aim is often to understand the way mathematicians reason or to help them in various tasks. Such systems, while they may be inactive for the time being, have potential, particularly in conjunction with others, as discussed briefly in various places of this paper. They include:

(a) the *INGRID* system of Brigham and Dutton [25] [26] [27] [28], which manipulates formulae on graph invariants from a database to compute bounds on some invariants when others are limited to some range. INGRID can be used to

  (i) help solve practical problems,

  (ii) derive new theorems (by selecting relations leading to them),

  (iii) test the effectiveness of new theorems (by showing they are or not consequences of one or several previously known ones),

  (iv) test conjectures (viewed as "temporary theorem" to see if this implies some contradiction),

  (v) resolve open problems (by showing they imply some contradiction), and

  (vi) help to study graph theory.

  As explained in [113], some of these functions may be viewed as obtaining particular types of conjectures.

(b) the *graph theorist* system of Epstein [73] [74] [75] [76]; this knowledge intensive, specific domain learning system uses algorithmic descriptions of classes of graphs such as connected, acyclic, bipartite and so forth. It mainly uses theory-driven discovery of concepts, conjectures and theorems, based upon search heuristics, but also infers explanations from factual input about graphs.

There are three main operational systems:

(a) the GRAPH system, developed by Cvetković and co-workers [59] [60] [54] [61] [55] [56] [62] [63], which pioneered the man-machine type of research in graph theory. Built between 1980 and 1984 this system was extensively used to find conjectures and prove theorems in graph theory (usually the latter only being published), with an emphasis on algebraic graph theory. Cvetković and Simić [64] review 92 papers by 23 authors on GRAPH, its uses and results obtained with it from 1982 onwards. GRAPH comprises

  (i) a bibliographic component, BIBLI,

  (ii) an algorithmic component, ALGOR, and

  (iii) an automated theorem proving one, THEOR.

(b) the Graffiti system, due to Fajtlowicz [81] [80] [82][83] [84] [85] [79] [87] and developed since the mid-eighties, with from 1990 onwards collaboration of De La Vina, notably in the development of its DALMATIAN version. This system generates a large number of *a priori* conjectures, under the form of algebraic relations between graph invariants, then selects among them, by eliminating false or uninteresting conjectures through testing them on a database of graphs, applying heuristics and building counter-examples. Conjectures which pass these correctness and interestingness tests are proposed, after further selection, to the mathematical community in the large email file "Written on the Wall" which is updated from time to time. More than 70 mathematicians, among them some famous ones, sent proofs, or refutations of those conjectures, listed in that file. Many papers on proofs, and more often disproofs, sometimes with corrected results which led to further developments or strengthened conjectures, have been published. De La Vina [67] lists 75 such papers, technical reports and theses from 1986 onwards.

(c) the AutoGraphiX (AGX) system, due to Caporossi and Hansen [37] [31] [65] [34] [35] [106] [33] [6] [32] [36] [107] which generates many extremal or near-extremal graphs for some invariant or formula involving several invariants, then derives various results from them. This system may be used to

  (i) find a graph satisfying given constraints;

  (ii) find optimal or near-optimal values for a graph invariant on a family of graphs with given constraints;

  (iii) refute, corroborate or strengthen a conjecture;

  (iv) make a conjecture in computer-assisted or automated mode;

(v) suggest ideas of proof.

A series of papers on the system, its uses, results and comparative performance have been published. Aouchiche [5] lists 40 papers on AGX and its results, or related to its results, published since 1999, submitted, or to appear.

Collectively, this number of papers (over 200) is among the largest in the field of discovery science.

Some programs from graph theory not designed specifically for making conjectures may be useful to do so, either on their own or in conjunction with others. This is the case in enumeration where e.g. programs such as *Nauty* and *geng* of McKay [131] [132] helped to conjecture and then determine many Ramsey numbers.

Conjectures can also be obtained by serendipity. As explained in more detail in [104], a program for coloring planar graphs written in Mathematica by Wagon, always used 3 colors when applied to rhombic Penrose tilings; Sibley and Wagon [152] then proved 3 colors suffice, a problem that had been open for 20 years. Another example relies on a program from mathematical programming: a mixed-integer formulation of the problem of determining the Clar number of a benzenoid [110], due to Hansen and Zheng [111], never used branching. The conjecture that linear programming sufficed to solve this problem was later proved by Abeledo and Atkinson [1] [2].

1.4 Plan of the paper

This paper has two complementary aims:

(i) *assess the state-of-the art* in computer-assisted and automated conjecture-making in graph theory. This will be done in the next three sections, devoted respectively to GRAPH, Graffiti and AGX, with special emphasis on their conjecture-making functions;

(ii) *make a series of proposals for advancement* of this field. They will be interspersed in the next three sections and will take two forms. First, *Proposed Enhancements* (PE) will suggest ways to improve specific steps or functions of the system under study; they will often be suggested by ways to solve similar problems in other systems and the suggestions will then amount to hybridizing them. Second, *Research paths* (RP) will draw attention upon open problems or general questions related to conjecture-making in graph theory, as well as links to establish with other domains of research. They are often long-term goals, sometimes quite speculative. Separation between study of systems and proposals will be indicated by numbering them PEk or RPk, with a □ sign as the end of the corresponding statement.

The three operational systems GRAPH, Graffiti and AGX will be studied in sections 2, 3 and 4 respectively. Conclusions will be drawn in Section 5.

2 Graph

As mentioned in the introduction, GRAPH has three components, BIBLI, ALGOR and THEOR. ALGOR is the most directly related to conjecture-making but both BIBLI and THEOR bear upon problems of importance for conjecture-making systems too. So we examine all three of them in turn.

2.1 BIBLI

The GRAPH system uses a formalized subset of the everyday English language, called Graph Theoretic Computer Language. It is described in [59]. It is an interactive language used from a terminal keyboard; in recent versions a mouse can be used also for some operations.

The BIBLI component is devoted to bibliographic data processing: it allows storage and retrieval of information on papers, books, proceedings, reports, abstracts, manuscripts and documents. Its functions, rarely available at the time of inception, are now in wide use in systems accessible on the web such as *Google, Web of Science* or *Citeseer*, but it remains useful for tailor-made bibliographies such as that one of the book of Cvetković *et al.* "*Recent Results in the Theory of Graph Spectra*" [57].

While very large amounts of data are now available online and special sites devoted to graph theory, such as the *Graph Theory White Pages* are open to the general public, the documentation problem in graph theory is far from solved (All those who have painstakingly derived a series of conjectures, transformed them by proof into theorems only to find in a last check most or all results to be known but expressed in a different language are well aware of this problem). Indeed, the graph theory literature is vast, dispersed over many fields, growing in a savage way and, as a consequence, terminology is far from unified. Moreover, due to dependence between concepts, the same results can take different forms e.g. in the graph G or its complement $\bar{G}$, or after eliminating one or another invariant by a linear equation such as those of Gallai's theorem [92]. Finally, some results can be expressed in different ways because concepts have a nonlinear dependence. To illustrate, even if one knows that the Wiener index of a tree T [72] is another name for the sum of distances between pairs of vertices of T, one might miss equivalent results expressed in terms of average distance between pairs of distinct vertices of T.

Brigham and Dutton [26] [27] have gathered 458 relations between graph invariants, used in their system INGRID. They can help in checking whether a result is new, but if this has to be done with a chance of success, a much more comprehensive system should be built, in a collaborative effort, similar to that which gave rise to Sloane's On-line Encyclopedia of Integer Sequences [157]. The following research paths sketch how this might be done:

RP1. *Find linear equality relations between graph invariants.* Consider a large number of graph invariants and programs to compute them (available in the cited systems, in Graphbase [119] or LEDA [134] and on the Web). Compute values of these invariants for a large set of graphs. Then use the numerical relation-finding routine of AGX (see Section 4)

to obtain a basis of affine relations on these invariants. If some new relations are found, prove them. □

RP2. *Define a standard set of invariants* in terms of which all others will be expressed and (one or several) *standard forms for relations in graph theory.* Write a translator program which will express (as far as possible) any formula in standard form and conversely express a standard-form formula in one or all equivalent forms. Programs for algebraic manipulations such as Mathematica [161] or Matlab [130] might be used for that purpose. □

RP3. *Organize a site* for interactive addition to and consultation of a database of graph theory relations. These relations might be valid for all graphs, or for important families of subgraphs, e.g. bipartite, triangle-free, of girth at least 5, and so forth. □

Another important open problem, related to storing graph theory relations is to find if a given relation is redundant, i.e., implied by one or more relations already in the database. This can be done by finding a graph within a database for which it is not the case, as in the DALMATIAN version of Graffiti [84] (see Section 3) or in an algebraic way as in INGRID [28] or by showing that the relation is not best possible (assuming a best possible relation is known).

Given invariants $i_1, i_2, \ldots, i_p$ of a graph G one can define, as in [109], a *canonical form* for relations involving these invariants as

$$i_k \leq f(i_1, i_2, \ldots, i_{k-1}, i_{k+1}, \ldots, i_p) \tag{2.1}$$

or

$$i_k \geq g(i_1, i_2, \ldots, i_{k-1}, i_{k+1}, \ldots, i_p); \tag{2.2}$$

such relations are *sharp* (or best possible) if for all values of $i_1, i_2, \ldots, i_{k-1}, i_{k+1}, \ldots, i_p$ compatible with the existence of a graph there is a graph such that the relation is satisfied as an equality. A set of canonical relations is *complete* if the $2p$ relations (2.1) and (2.2) on $x_1, x_2, \ldots, x_p$ are sharp. One such set for the three parameters $\alpha(G)$ (independence number), n (order) and m (size) is given in [109]. The relation [105] [88]

$$\alpha(G) \geq \left\lceil \frac{2n - \frac{2m}{\lceil \frac{2m}{n} \rceil}}{\lceil \frac{2m}{n} \rceil + 1} \right\rceil \tag{2.3}$$

is sharp, while the following one, derived from Turan's theorem [159],

$$\alpha(G) \geq \frac{n^2}{2m + n} \tag{2.4}$$

is not. It is thus redundant but might be kept also if one is more interested in simplicity than in sharpness. Observe also that if a sharp relation is known one might consider that it is not useful to compare it to another one, yet the latter could also be sharp and simpler as is the case for (2.3) which is equivalent to but simpler than the relation given in [105].

2.2 ALGOR

This part of the system GRAPH is directly connected to conjecture-making ([59], p20):

> "The part of the system "GRAPH" described is primarily meant as a means for quick[ly] checking, disproving or making conjectures in graph theory. Facilities provided by the system enable to get the answer on a great number of questions on graphs of a reasonable size in a few seconds (of course, what does a reasonable size mean depends on the problem considered)."

Also:

> "Another situation in which the system can help is the following. Many results in graph theory begin with an observation which proves the desired statement for all but a finite number of graphs. These exceptional graphs are, as a rule, of a small size. The next part of the proof consists then in checking whether the statements hold for these graphs and that can be performed with the help of the system."

ALGOR solves a series of problems on particular graphs. They can be divided as follows: ([59], p11):

(a) manipulative tasks (setting and displaying values of the mentioned objects (i.e., graphs, values of the type integer, real and complex, and families of sets of integer values),

(b) creating common graphs (e.g. complete graphs, circuits, etc) or random graphs,

(c) creating graphs by performing graph-theoretic operations (e.g. complement of a graph, product of two graphs, etc),

(d) relabelling (points or lines of) graphs (by given permutations, at random, etc),

(e) determining integer invariants in graphs (e.g. number of some subgraphs, order of some point, etc.),

(f) determining real invariants of a graph (e.g. eigenvalues, eigenvectors, etc),

(g) checking properties of graphs, (e.g. whether a graph is planar or hamiltonian, whether two graphs are isomorphic, etc),

(h) listing families of graph characteristics (e.g. point degrees, components; etc).

Each group of operations is characterized by a verb in the commands used. They have a simple and transparent form, e.g.

CREATE < g-name > [AS] < type of graph > [OF] [ORDER] < integer >,

for instance:

CREATE $G1$ CIRCUIT OF ORDER 12

or

FORM [g-name] [AS] [THE] < integer > [TH] < operations > [GRAPH]

[OF] < g-name >,
for instance:

FORM H AS THE 4TH SUBDIVISION GRAPH OF G

The operations are: DISTANCE, (PATH), POWER, SUBDIVISION, (TRAIL), (WALK). Names in parentheses correspond to operations not yet implemented when [53] was written.

PE1. Complete GRAPH by enriching its functions as planned. This task is in progress in the system NEWGRAPH, currently developed. □

Determining invariants is broken down in four categories:

(a) Invariants of the graph
(b) Invariants of point of the graph
(c) Invariants of a given size
(d) Invariants of two points of the graph

Commands for invariants of a graph have two forms:

(i) determining the number of objects in the graph, which can be (AUTOMORPHISMS), BLOCKS, BRIDGES, CENTRAL POINTS, (CIRCUITS), CLIQUES, (COCLIQUES), COMPONENTS, CUTPOINTS, (INDEPENDENT LINES), LINES, LOOPS, MAXDEGREE, MINDEGREE, (ORBITS), PENDANT LINES, (PENTAGONS), POINTS, QUADRANGLES, TRIANGLES;

(ii) computing the value of an invariant such as (CHROMATIC CLASS), (CHROMATIC INDEX), CHROMATIC NUMBER, CIRCONFERENCE, (CLIQUE NUMBER), (COARSENESS), (COMPLEXITY), (CROSSING NUMBER), CYCLOMATIC NUMBER, (DETERMINANT), DIAMETER, (EXTERIOR STABILITY), (GENUS), GIRTH, (INTERIOR STABILITY), (LINE CONNECTIVITY), (PERMANENT), (POINT CONNECTIVITY), RADIUS, RANK, (THICKNESS).

For instance:

DETERMINE THE NUMBER OF TRIANGLES OF G,
DETERMINE DH THE DIAMETER OF H.

A point invariant such as DEGREE, ECCENTRICITY, etc would be found by making a command such as

DETERMINE DEGREE OF 7 OF G

where 7 is the label of a point. Commands for invariants involving two points or real invariants of a graph are similar. Possible objects are (CIRCUITS CONTAINING), COMMON NEIGHBOURS, (DISJOINT PATHS), DISTANCE, LINE LABEL, LINES INCIDENT, (PATHS), (TRAILS), (WALKS) in the former case and (ADMITTANCE SPECTRUM), (ANGLES), BOND ORDERS, CHARGES, DISTANCE, INDEX, (DISTANCE SPECTRUM), EIGENVALUES, EIGENVECTORS, ENERGY, (MAIN ANGLES), (R-SPECTRUM), SEIDEL SPECTRUM in the latter.

The GRAPH system can check many properties of graphs such as ACYCLIC, BIPARTITE, BLOCK, (BLOCK CUTPOINT GRAPH), (BLOCK GRAPH), CIRCUIT, (CLIQUE GRAPH),

COMPLETE, CONNECTED, (CUTPOINT GRAPH), EULERIAN, FOREST, HAMILTONIAN, HYPOHAMILTONIAN, (INTERVAL GRAPH), LINE GRAPH, LOOPLESS, (MOORE GRAPH), (OUTERPLANER), (PERFECT), PLANAR, (PRIME), (SELFCOMPLEMENTARY), (SELFDUAL), SEMI REGULAR, (SEMITOTAL LINE GRAPH), (SEMITOTAL POINT GRAPH), STRONGLY REGULAR, (SUBDIVISION GRAPH), (TOTAL GRAPH), TOTALY DISCONNECTED, (TRAVERSIBLE), TREE, TRIANGLE FREE, TRIVIAL, UNICYCLIC, WHEEL, WITHOUT MULTIPLE LINES.

Commands are for instance

CHECK WHETHER $G1$ IS PLANAR,
CHECK WHETHER $G2$ IS A TREE.

or, for properties of a point of a graph:

CHECK WHETHER THE POINT 5 IS ISOLATED IN G,

or of two graphs

CHECK WHETHER $G1$ AND $G2$ ARE ISOMORPHIC.

Clearly the system GRAPH can answer a large number of questions regarding particular graphs. It can also check for graphs with some property among several lists of graphs, e.g. connected graphs up to 6 points, regular graphs up to 7 points, trees up to 10 points, cubic graphs op to 12 points, etc.

Results of GRAPH consist, as mentioned above, of computer-assisted conjectures, refutations and proofs. Most of the published results are theorems, and while mention of system GRAPH is made, details on how it led interactively to conjectures, refutations or proofs are unfortunately not given except in [59] (automated theorem-proving is discussed in more detail [62] [56]).

We list a couple of results obtained with GRAPH, see [64] for a more comprehensive set. Let G be a graph, v a distinguished vertex, and $N_1(v)$, $N_2(v)$ a partition of the neighbours of v. If G' is obtained from $G - v$ by adding vertices v_1, v_2 and edges $\{v_1, w\}$ with $w \in N_1(v)$ and $\{v_1, w\}$ with $w \in N_2(v)$, G' is obtained by *splitting* vertex v.

The following result was conjectured with the system GRAPH and proved in [153]: *If G is a connected graph and G' is obtained from G by splitting a vertex then $\lambda_1(G') < \lambda_1(G)$* (where $\lambda_1(G)$ is the *index* of G or largest eigenvalue of its adjacency matrix).

Denote by $\rho(k)$ the largest eigenvalue of the graph obtained from the cycle C_n with $n \geq 6$ by adding an edge between two vertices at distance $k = 2, 3, \ldots, \lfloor n/2 \rfloor$. On the basis of experiments conducted with GRAPH it was conjectured that $\rho(k)$ is monotonous and decreases. This was proved in [148] [154].

2.3 THEOR

The THEOR component of GRAPH is designed for computer-assisted or automated theorem-proving in graph theory, and is described in Cvetković and Pevac [62]. We only discuss it briefly as this paper's topic is not automated theorem proving. Graph theory

is formalized using a special first-order predicate calculus, called "arithmetic graph theory" (AGT). It contains point variables, line variables, integer variables, graph names, constraints, function names, operations over graphs and predicates.

The effectiveness of the prover depends largely on a set of lemmas which represent beginner's knowledge of graph theory. The user may select more advanced lemmas.

A resolution-based prover is a subsystem of a natural deduction interactive theorem prover. The interactive prover provides a proof for a given goal sentence P by splitting it into subgoals, which are further split, thus generating a proof tree memorized by the system. This tree is a rooted one, and the user can move the current root, i.e., select the subgoal next considered. He can also inform the system about the truth of a subgoal. The resolution-based prover can be applied to any subgoal and the proof is completed when all subgoals are proved. Subgoals may be processed by case analysis, forward chaining, *reductio ad absurdum*, simplification or extension of the formula, expressing it in an equivalent form, etc.

A completely automated proof of the simple sentence

> "If the graph is connected, then the graph is trivial or there is no point x such that x is isolated"

is obtained and has 10 lines. The sentence

> "If the graph is not connected, then the complement is connected"

is proved interactively, in 38 lines. Further examples are given in [56].

These examples show the difficulty inherent in full formalization of graph theory. Its language, close to English, is deceptively simple. The situation is much easier in logic [162], or in plane geometry where a method of reduction of problems to systems of linear and quadratic equations applies, see e.g. Chou [43]. But as the speed of equally priced computers has augmented since the time GRAPH was developed by a factor of 10^4 to 10^5 and automated theorem proving made much progress, another attempt might be worthwhile.

PE2. Test the automated theorem proving approach of THEOR with a modern computer and a prover such as OTTER [136]. □

Should this attempt be successful, it should meet a wish of Fajtlowicz [84]:

> "the problem of trivial conjectures could be solved if we had automated theorem provers capable of proving the easiest conjectures of Graffiti . . . "

3 Graffiti

3.1 Structure

The Graffiti program is discussed in the series of papers "*On conjectures of Graffiti*" [81] [80] [82] [83] [84] a paper "*On conjectures and methods of Graffiti*" [87] as well as in the more recent paper "*Towards fully automated fragments of graph theory*" [85], and a couple of papers of Larson [124] [125]. De La Vina [68] presents the system Graffiti.pc and, very

recently, some recollections about early development and use of Graffiti [69]. Conjectures obtained with Graffiti and their status i.e., proved, refuted or open, are listed in [79].

There are many versions of Graffiti, not all of which appear to have been fully documented [69]. The two main ones appear to be the initial version (with a few developments) described in [80] [81] [82] [83] [87] and the DALMATIAN version described in [84] [85] [124] [125] and [68]. In this subsection we list the steps of both of them. These steps will be discussed in detail in the following subsections.

Unfortunately, no complete and precise description of all steps of the process of obtaining conjectures with Graffiti has been provided. Instead, partial and informal descriptions of the automated steps are scattered over a good half-dozen publications; information about the other steps is given similarly, but in much less detail. This makes rational discussion of the Graffiti system and its applications extremely difficult as it must be preceded by a long reconstruction process, i.e., finding what really happened, or happens, from scant and sometimes contradictory information (as e.g. when computing invariants is attributed to Graffiti in one place and to Algernon in others). The paper of De La Vina [68], written after the first version of the present paper was completed, and remarks of an anonymous referee have been very helpful in this reconstruction process.

Graffiti uses two databases; a database of graphs and a database of conjectures. The former contains graphs proposed by the authors or other researchers, which have refuted some conjectures, together with precomputed values for all invariants considered in the system. The latter contains conjectures generated by the system and not refuted or viewed as non-interesting, or possibly in the DALMATIAN version, viewed as non-informative.

Steps of the process of finding conjectures with the initial version of Graffiti appear to be the following:

Step 1. Problem statement: Find relations between a set of invariants $i_1(G), i_2(G) \ldots$ chosen by the user.

Step 2. Conjecture generation: The program generates a set of inequalities of the forms $i_1(G) \leq i_2(G)$, $i_1(G) \leq i_2(G) + i_3(G)$, or similar ones using the selected invariants and possibly small integers (mostly 1).

Step 3. Correctness Test: The program evaluates the inequalities obtained. If one graph refutes them, they are deleted.

Step 4. Heuristic Tests (see below): The program deletes the conjectures which do not pass the test.

Step 5. Counter-example: Find by hand (a) counter-example(s) to at least one of the new conjectures. If one is found, delete the corresponding conjecture.

Step 6. Update of Graph Database: If at least one counter-example has been found compute values of all invariants for the corresponding graph(s). Adds these graphs to the database of graphs and return to Step 3.

Step 7. Elimination of true conjectures: Prove by hand easy new and true conjectures and eliminate them from the database of conjectures (if they are not judged to be interesting).

Step 8. Selection of conjectures: Select, by hand, among the remaining conjectures those considered to be worthy of publication. Make them known, e.g. by including them in the "Written on the wall" file.

Fajtlowicz ([80] p.189) comments as follows on this process, and its interactive character:

"Graffiti makes conjectures by first verifying that it does not know a counter-example to a formula and then by deciding whether the formula makes an interesting conjecture. The first function of the program is highly interactive because a user is expected to find counterexamples to false conjectures and then describe them to the program."

Steps of the process of finding conjectures with the DALMATIAN version of Graffiti appear to be the following:

Step 1. Problem statement: Find lower (or upper) bounds for a user-selected invariant.

Step 2. Conjecture Generation: The program generates an inequality and evaluates the values of both sides of all graphs in the database.

Step 3. DALMATIAN test for informativeness (see below): The program deletes the conjecture if it does not pass the test.

Step 4. Correctness test: The program deletes the conjecture if the inequality does not hold for at least one graph in the database.

Step 5. Other heuristic tests (see below): The program deletes the conjecture if it does not pass one of these tests.

Step 6. Database updating: The program shelves conjectures viewed as less informative due to the addition of the new conjecture.

Step 7. Test for ending conjecture generation: If for each graph in the database of graphs, there is a conjecture for the selected invariant and direction of inequality in the database of conjectures which is sharp (i.e., satisfied as an equality), proceed to the next step. Otherwise, return to Step 2.

Step 8. Counter-example: Find, by hand, a counter-example to one at least of the inequalities generated.

Step 9. Updating database of graphs: If a counter-example has been found, compute with an auxiliary program (Called Algernon) the values of all invariants for this graph, introduce it, together with those values in the database of graphs and return to Step 2.

Step 10. Elimination of true conjectures: Prove by hand easy new and true conjectures and eliminate them from the database of conjectures.

Step 11. Selection of conjectures: Select by hand among the remaining conjectures, those considered to be worthy of publication. Make them known, e.g., by including them in the "Written on the Wall" file.

Note that the correctness test now follows the first interestingness test; the reason appears to be that the DALMATIAN test is quicker than the other one on average.

Observe that as the new conjectures have the same left-hand side invariant and direction of inequality they may be viewed as a system. Note that the procedure described does not necessarily converge (a simple example is given below). It may thus have to be stopped manually, after some time.

At this point, a divergence of opinion between the authors of the Graffiti system and the present author should be clearly stated. Fajtlowicz focuses on what is automated and wishes to limit Graffiti to Steps 1 to 7 above. When they are finished, which constitutes a *round*, the user takes over, does whatever he wishes (eventually with the help of Algernon) and may proceed or not to a further round. So the non-automated part of the conjecture generation process is viewed to be in some sense, outside of Graffiti, while the final conjectures are still attributed to Graffiti alone, as shown by referring to them as "conjectures of Graffiti" or "conjectures obtained by Graffiti".

This author could only accept this view if what is not automated did not substantially affect the final result, i.e., the list of conjectures to be publicized. That steps 8 to 11 play an important role will be documented in the following subsections. Note also that isolating automated parts from the other ones, and giving them a name, then considering the remaining parts to be outside of the process, can lead to a claim that the resulting process is (fully) automated, for any interactive process. The present author cannot agree with such an argument and therefore views Graffiti as a computer-assisted system and not a (fully) automated one. The reader is left to judge.

3.2 Problem statement and generation of *a priori* conjectures.

In the initial version, the problem statement consists in specifying the invariants to be studied (e.g., a set of 20 from the rich library of Graffiti) as well as, possibly, operators such as sum, maximum, minimum, complement etc acting on them, and the desired form of the relations derived. The program then generates systematically such relations.

Forms of conjectures are simple ones, such as $i_1 \leq i_2$ or $i_1 \leq i_2 + i_3$ or sometimes $i_1 + i_2 \leq i_3 + i_4$. Later, ratios were introduced and finally a real algebra on the invariants.

In the DALMATIAN version, the problem statement step has the following form: Find lower (or upper, instead) bounds for a (user selected) invariant. The system then generates a term, as right-hand side of the inequality. This term is obtained by selecting invariants and performing unary or binary operations on them. Examples of such operations are the reciprocal, the natural logarithm, ceiling, addition and multiplication [68].

Details on how this is done, i.e., how many invariants and operations are chosen, within which set, according to which rules and whether or not there is any further user intervention before the session or at the moment the user states his query, are not given. As a consequence, results of Graffiti cannot be reproduced by other researchers.

As the conjecture-generation step conditions the results obtained, it should be analyzed carefully.

First, one may note that the system does not *at this stage*, use any knowledge of graph theory at all, so one should speak of *guesses* rather than *conjectures* (that the subsequent process, which uses graph theoretic algorithms as well as heuristics transforms or not these guesses into conjectures by its selection process will be the crucial point).

In view of this lack of knowledge, one may expect that initially

(i) many conjectures will be false;

(ii) many conjectures will be true but trivial;

(iii) if a very large number of conjectures are generated some of them may be interesting.

Reading all papers written on Graffiti and its conjectures suggests that all three propositions, including the redeeming third one, are true. Fajtlowicz comments as follows on trivial conjectures ([81], p.113) obtained with the initial version of Graffiti. "The number of conjectures, particularly those which are completely trivial, is the main problem and more than half of the program consists of various heuristics whose purpose is detection of trivial and otherwise non-interesting but true conjectures". As documented below in the subsection on selection of conjectures, a substantial number of the selected ones remains false with the initial version and also, to a lesser extent, with the DALMATIAN one.

Second, generation of some important formulae may be, in practice, out of reach of Graffiti, even if the necessary invariants and operations are available, because their algebraic expression is too complex. To illustrate, consider again the bound (2.3) on the stability number $\alpha(G)$. It implies only 2 invariants, m and n, but 12 product, division, sum, subtraction or upper bound operation. The probability that the right invariants and operations, as well as their order can be found *a priori* must be extremely small.

Consequently, Graffiti is not a good tool for obtaining strongest conjectures, i.e., graph theoretical bounds which are best possible in the strong sense, that is, as formula (2.3), tight for all m and n. That other systems, together with a few algebraic manipulations, can do so is illustrated in [107] for the case of an upper bound on the irregularity of a graph.

A related problem arises if the formulae have numerical coefficients; Graffiti introduces a few, usually small, integers. However, if the coefficients are real ones, the number of possible formulae is infinite even in the linear case. How could Graffiti guess *a priori* the right values in such a case?

Third, observe that no computer is needed to generate systematically relations between graph invariants: the (tedious) task of writing down $i_1 \leq i_2, i_1 \geq i_2, i_1 \leq i_3$ and so on can be done by hand without any difficulty; enumerating relations with more complicated forms as done in the DALMATIAN version is only slightly more complicated. Programming this task is also easy.

Fourth, while some *a priori* conjectures are simple and appealing, more complicated ones might not be attractive. To illustrate, the formulae

$$\bar{l}(G) \leq \alpha(G) \qquad \text{(Graffiti 2)}$$

where $\bar{l}(G)$ denotes the average distance between distinct vertices of G and

$$r(G) \leq \alpha(G) \qquad \text{(Graffiti 0)}$$

where $r(G)$ denotes the radius of G, or minimum over all vertices of the largest distance to another vertex, have attracted mathematicians and led to several papers; contrarywise, most mathematicians might consider that the conjecture

"The minimum of derivative of eigenvalues of the gravity matrix is $\leq n$/average distance" (Graffiti 150)

is too complicated and specialized.

Fifth, *a priori* conjectures of Graffiti may not have the simplest form they may take. To illustrate the *temperature* t_j of a vertex j is defined by Fajtlowicz as

$$t_j = \frac{d_j}{n - d_j}$$

where d_j is the *degree* of j. The conjecture

$$\bar{l}(G) \leq 1 + \max_j t_j(\bar{G}) \qquad \text{(Graffiti 834)}$$

where $\bar{G}$ is the complementary graph of G, can be reformulated into

$$(1 + \delta(G))\bar{l}(G) \leq n$$

which is simpler, more intuitive, and was refuted [37].

PE3. Add to Graffiti a translation routine which would automatically simplify conjectures. □

3.3 Dalmatian and other heuristics

We now describe and discuss the various heuristics designed to select *interesting* conjectures among those listed *a priori*. The DALMATIAN one [84] is the most recent and apparently also the most powerful. It is based on the notion of *information content* (or informativeness). Basically, a conjecture on an invariant is considered as interesting if and only if it provides some new information for at least one graph in the system's database, i.e. it provides for that graph a strictly better bound than all previous relations. Otherwise, the conjecture is deleted. If it is added to the database of conjectures it may happen that some other conjectures are no more informative and are *shelved*, i.e., kept separately of the database of conjectures (or tagged); if later on some conjecture(s) giving a better or equal bound on i_1 is (are) refuted they can be *unshelved*, or considered as interesting conjecture once again.

Several comments are in order. First, the definition of interestingness on which the dalmatian heuristic is based is *local*, as it depends on the database of conjectures and the database of graphs of Graffiti, and *unstable*, as these databases evolve over time. This implies this definition is not *universal*, i.e., contrary to other mathematical definitions, it cannot be used by all researchers in all places with consistent answers as to whether a conjecture is or not interesting.

Second, the definition may be too *lax*, if the database of conjectures is small or the database of graphs is large (but this would be only temporary as new conjectures are introduced and initial ones shelved), or too *severe* if the database of conjectures is large. Indeed, the situation in which the values of a large set of invariants and many relations on the invariant i_1 under study are known is atypical in graph theory research. Much more often, graph theorists study one invariant as a function of two or three others, ignoring temporarily the other ones.

Four other heuristics were used in early versions of Graffiti. The IRIN heuristic "deletes conjectures which follow from others by transitivity" [81]. The CNCL heuristic deletes conjectures "...in which one invariant on the left is always smaller than an invariant on the right" [81]. The ECHO heuristic [80] applies to conjectures defined for restricted classes of graphs: "its main idea is that a conjecture about a class of objects A is considered noninteresting if it can be generalized to a larger class B ..., the background of A". This heuristic appears still to be used in recent versions of Graffiti. The BEAGLE heuristic is based upon the idea that conjectures involving concepts of a different type are more likely to be interesting [82].

The idea of difference in concept types is related to a representation of concepts as a rooted tree: a graph G is associated with the root and various numerical invariants to its vertices. A concept is a *descendant* of another one if it is computed in terms of that one. The distance between vertices in the tree can be viewed as a distance between the corresponding concepts.

The BEAGLE heuristic removes conjectures involving concepts that are too close; it appears that the DALMATIAN also removes most but not all of them. Larson ([124] p.12) comments on this as follows:

"The BEAGLE heuristic of Graffiti was central to early versions of the program [82]. The function was largely superseded with the introduction of the DALMATIAN heuristic."

Note that the BEAGLE heuristic, as the DALMATIAN one, is defined in terms of the Graffiti system. One may wonder if distance between concepts in graph theory could be defined in a more general mathematical way. This seems to be the case, as lattices of graph theoretic concepts are considered by the Graph Theorist system of Epstein [74] [75] [76] as well as by the Hardy-Ramanujan system of Colton [50] (which is more often applied, however to algebra or number theory than to graph problems). A concept of distance follows. It seems worthy of further study to see to what extent this framework, or more general ones, apply:

RP4. Apply the theory as *formal concept analysis* [93] to graph theory definitions and see if a concept of distance between concepts can be derived. In particular, study to what extent concepts in graph theory can be represented by a lattice (or several). Deduce new concepts from this(these) lattice(s). □

Note that Graffiti is not designed for finding new concepts (except in the trivial sense that any inequality can be viewed as defining a new concept); it is claimed however in one place ([84]) that

> "*the current version can define its own properties.* One of the properties discovered by Graffiti is the class of all graphs in which the smallest eigenvalue has multiplicity 1. Graffiti defined this concept because it knew many examples of such graphs".

However, as no routine for concept discovery is described in the papers on Graffiti, this appears to be more an observation of the user than a discovery of the system.

PE4. Add to Graffiti a data mining routine to find frequent patterns in its database of graphs, as well as a routine and a database to check if they correspond or not to known concepts. □

Considering results of the heuristics, one may note that

(i) some of the conjectures of Graffiti which passed the tests are simple and attracted much attention of graph theorists;

(ii) the simplest ones are of the form $i_1 \leq i_2$, and the best known is probably conjecture Graffiti 2: *For any graph* G

$$\bar{l}(G) \leq \alpha(G),$$

(where $\bar{l}$ denotes average distance and α the independence number) proved by Chung [49].

It is surprising, as it connects very different concepts, on the one hand average distance, based on paths and on the other hand independence, based on non-adjacency. Perhaps this is the reason why it was not suggested by anyone before.

Other conjectures of the same form involve concepts which had been little studied, or not studied at all, by mathematicians at the time they were introduced into Graffiti; this is the case for the Randić index [144] defined for any graph $G = (V, E)$ by

$$Ra(G) = \sum_{i,j/v_i,v_j \in E} \frac{1}{\sqrt{d_i d_j}}$$

where d_j is the degree of vertex v_j. This concept appears in the following conjecture: *For any connected graph* G

$$\bar{l}(G) \leq Ra(G), \qquad \text{(Graffiti 3)}$$

which is still open (conjectures involving the Randić index tend to be hard to prove as the value of this invariant may increase or decrease upon addition of an edge to the graph considered).

Yet other conjectures use concepts invented by Fajtlowicz. The Havel-Hakimi operation on the set of degrees of vertices of a graph, ranked in order of non-increasing values, consist in deleting the first degree d_1 and reducing by 1 the next d_1 degrees. Havel [116] and Hakimi [103] independently proved that a degree sequence is *graphical*, i.e., corresponds to a graph, if and only if the degree sequence obtained by the above operation does. Iterating this operation finally leads to a series of zeros; their number is the *residue* $Re(G)$ Fajtlowicz considered it as an invariant and obtained with Graffiti the conjecture: *For any graph* G,

$$Re(G) \leq \alpha(G), \qquad \text{(Graffiti 69)}$$

which was proved by Favaron, Mahéo and Saclé [88]. Several further papers [66] [90] [96] followed.

The question of whether or not the concepts involved in a conjecture bear upon its interestingness has not been much studied. Fajtlowicz notes that finding new concepts is

not difficult at all, contrary to the case of conjectures. Indeed concepts are not true or false, but simple or not, convenient or not and, more importantly, able or not to unify previous results. Finding new ones by computer is as easy as making guesses, but finding interesting ones may be another matter. Fajtlowicz argues that any sufficiently simple concept is interesting. While this may be true for most concepts which Fajtlowicz invented, as he found several nice ones (see Written on the Wall, *passim*), and attracted attention of mathematicians to them, it is hard to agree with his argument in general. Indeed, graph theory suffers from a plethora of concepts, the number of which suggests several questions.

First, to illustrate, one might argue that average distance is a simpler, or more central, concept than residue, and that the independence number is simpler and more central than both. Indeed, independence depends only on the basic concept of adjacency, average distance on the central concept of paths and their length while Residue depends on a particular algorithm. Of course, both average distance and residue give lower bounds on the independence number, and could be used in a branch-and-bound algorithm to determine its value. This may not be their main attraction, particularly for average distance which gives a usually loose bound.

Then considering general questions, we may propose:

RP5. Define the *simplicity* of a concept by the *minimal* number of operations to be applied to a graph G to compute it (operations not being considered here as elementary operations as in complexity theory but in more abstract terms as "checking adjacency for all pairs of vertices" or "computing all shortest distances between pairs of vertices"). This research would continue that of Graffiti on distance between concepts. □

RP6. Do the same as RP5 but using the concept of *Information (or Kolmogorov) Complexity*[127], i.e. the minimum length of a program to compute the invariant considered.

RP7. Evaluate empirically the importance of concepts in graph theory by a statistical analysis of their use in the literature. □

The next research proposal is inspired by the analysis of research networks as done in scientometrics [141] [126].

RP8. Construct a network of graph-theoretical concepts by associating them to the the vertices of a complete graph, and weighting edges by the number of times concepts corresponding to their end vertices are used in the same paper of some chosen corpus. Then analyze this network with standard tools of scientometrics to find central concepts, cliques of concepts used jointly, distance between concepts and other information. □

3.4 Refutation

Conjectures which passed the heuristic tests (or some of them) are tested on the database of graphs for correctness. If they do not hold for one of these graphs they are deleted.

Several remarks on the selection of graphs, heuristic or exact algorithms and graph representation are in order, as these questions bear upon the efficiency of the refutation process.

First, checking conjectures on the few hundred graphs of the database is not a severe test. Indeed, the classes of graphs under consideration are usually infinite.

Other systems are more powerful and/or more original in this respect: GRAPH uses interactive modifications, which constitute an informal descent method and can also get out of local optima; AGX applies the efficient and versatile Variable Neighborhood Search metaheuristic (see below); *Geng* and other enumeration programs list systematically much larger sets of graphs; INGRID combines relations between graph invariants, assuming the conjecture to be true, i.e., a *temporary theorem*, in order to derive a contradiction.

Some hybrids of Graffiti and enumeration programs have been sporadically explored: Fajtlowicz mentions using the CaGe program of Brinkmann [29] to generate fullerenes and De La Vina [68] applies Makeg of Mc Kay to obtain all trees satisfying given constraints and uses them in Graffiti.pc. She proposes as criterion of interestingness the *touch number* or number of graphs for which the conjecture is sharp (a criterion already used informally in [31] where it seems to have been mentioned in print, without the name, for the first time). In a recent paper, De La Vina [69] claims it was used in Graffiti since the early 90's, but for some reason it was not mentioned in the previous papers on that system, and notes that with a large database, conjectures with an important touch number tend to be true. Such a development appears to be promising.

Second, graphs in the Graffiti database are often those which refuted some conjecture and were proposed by various researchers. A set of 195 of them is described in the "*Graphs of Graffiti*" file [156]. The implicit assumption behind their selection appears to be that graphs which have refuted some conjecture may be more useful than randomly generated ones to refute others. This appears to be worthy of further study.

RP9. Study statistically which graphs are the most efficient for refuting conjectures of a given corpus, representative of the various types of algebraic ones. Examine also which conjectures are hardest to refute. □

Third, Graffiti (or Algernon) uses heuristics to compute the value of invariants such as the independence number, which are NP-complete to determine. This introduces an unnecessary error for small graphs; moreover up-to-date heuristics and metaheuristics could be used for evaluating such invariants, instead of simple heuristics such as MAXINE ([83]) for the independence number which are adequate for small graphs but not competitive for larger ones (see e.g. [14] [112] for state-of-the-art heuristics for the clique or independence number).

PE5. Replace heuristics for NP-hard invariants in Graffiti by exact algorithms, coupled with the best available heuristics for the same problems, to be used on large instances. □

In addition to automated refutation the process of finding conjectures with Graffiti uses further counter-examples obtained by hand by the user. In the DALMATIAN version, after introducing the corresponding graph(s) into the system a conjecture is generated again.

Here, knowledge and work of the user is incorporated and may strongly influence the quality of the conjectures obtained. Indeed, it is well known that when discovering and proving a theorem one often goes through a sequence of conjectures and refutations getting

progressively closer to the correct statement. So this procedure is certainly reasonable and appears to be efficient; however, it is not automated. Probably, as discussed above, in the present state of graph theoretical theorem proving, it could not be. However, what is examined here is the impact of the counter-example obtained by hand on the new, further conjectures obtained. This is essential to evaluate how far the process of finding conjectures with Graffiti is automated.

To illustrate, consider Graffiti conjecture 117. Initially, this conjecture was stated as follows: *For any connected graph*

$$\bar{l}(G) \leq \sum_{j=1}^{n} \frac{1}{d_j}$$

It was disproved by Erdös, Pach and Spencer [77]. Fajtlowicz then proposed the weaker version:

For any connected graph G with girth $g(G) \geq 5$, average distance is not more than inverse degree (where inverse degree is shorthand for the sum of inverses of degrees of all vertices) and surmised that given the known counter-examples, Graffiti would come up with that version. Granting the hypothetical, it remains that a non-trivial result by famous graph theorists was needed to transform an initial conjecture which turned out to be false into an interesting and still open one.

Another example is Graffiti's conjectures 67 and 119; they involve the new invariant $f(G)$ defined as the *maximum frequency of occurrence of a degree in G* (or mode of the degree sequence). For conjecture 67, i.e.,

For any graph G without K_3 (i.e., *with* $g(G) \geq 4$)

$$\chi(G) \leq f(G)$$

counter-examples were found by Staton and later by Erdös and Staton [78]; knowing some counter-examples, conjecture 119 was obtained:

For any graph G without K_3 or K_4, (i.e., *with* $g(G) \geq 5$) $\chi(G) \leq f(G)$.

So, once again, a counter-example obtained by hand was needed to transform a false conjecture into an interesting open one. Recently, Caro [39] proved that this last conjecture is true for all sufficiently large graphs.

The fact that this step is not automated does not appear to be discussed in the parts of papers on Graffiti which concern automation. One may wonder how often one had recourse to counter-examples obtained by hand before reaching the conjectures publicized in "Written on the Wall". Very recently some information on that point has been provided in [80], it is stated that

"... in the 1980's once the conjectures were output, then as described by Fajtlowicz in [81] he would categorize the program's conjectures as *false*, *proven* and *open*. Counter-examples to conjectures were reported to the program, the program was re-executed and again the conjectures would be categorized. As further described in [80] after a few rounds of this process, as is the academic custom, Fajtlowicz announced the open conjectures".

3.5 Proofs

Many conjectures of Graffiti are true but trivial. Some of them are deleted as they are not informative according to the criterion of the DALMATIAN heuristic. This selection process could be made much more efficient by considering true relations (theorems) as well as conjectures.

PE6. Add to Graffiti a database of theorems containing both classical ones and others, proved with possible help of that or other systems. Then apply the DALMATIAN heuristic with a joint database of conjectures and theorems. □

True conjectures which pass the DALMATIAN heuristic test are studied by the user, and discarded if they appear to be trivial (which is not synonymous with, but implies the conclusion that they are trivial to prove). No operational system for theorem-proving in graph theory being available, this is done by hand.

Note that if a database of theorems is available it can also be used, as in INGRID [28], to find if a conjecture is implied by one or several theorems from that database and which. Then, if the resulting system is not too complicated, a proof might be obtained automatically by a system for algebraic manipulations such as Mathematica [161] or Matlab [130].

Presently, that one conjecture obtained with Graffiti (or a theorem if it has been proved) follows from another is only discovered with a web database or by a chance remark from one or another graph theorist.

To illustrate, the conjecture Graffiti 1 is: *For any graph G,*

$$\chi \leq 1 + rank(A(G))$$

where $A(G)$ is the adjacency matrix of G. Jaeger told Fajtlowicz ([79], p5) that Van Nuffelen [160] had proposed earlier the stronger conjecture

For any graph G,

$$\chi \leq rank(A(G)).$$

Both conjectures were refuted by Alon and Seymour [4].

This example shows the interest of a database of graph theory formulae, as discussed in Section 2.

3.6 Selection of conjectures

Until the version of Graffiti comprising the DALMATIAN heuristic, conjectures which passed the tests of the heuristics and could neither be refuted nor proved were further selected by the user. This new heuristic raised big hopes ([84], p 370):

> "There are strong indications that the new version of Graffiti can be used so that it will make very few trivial conjectures ... If these early indications, based on test runs, are right, it would mean that the program can be fully automated and can make conjectures without any help of humans. By contrast, as I was always clearly stating this, conjectures of previous versions of Graffiti had to be approved by myself, before they were included in "Written on the Wall"."

However, it seems that proofs of easy true conjectures are still done by hand, perhaps some non-automated selection of conjectures still takes place and counter-examples obtained by hand are added to the database within the conjecture-making process. Automation of Graffiti is further discussed when considering the "Little Red Riding Hood" version of Graffiti in Subsection 3.8 below.

A few studies allow evaluation of the proportion of conjectures of "Written on the Wall" which are false. The two first of them correspond to the initial version. Favaron, Mahéo and Saclé [88] studied extensively eigenvalue properties of graphs conjectured with Graffiti. They proved 3 of them in their original form, 9 others as corrolaries of stronger results and disproved 49 of them. Brewster, Dineen and Faber [24] program a series of invariants and tested about 200 conjectures of Graffiti using a database of all graphs with up to 10 vertices. They refuted 49 of these conjectures (some with such simple graphs as a single edge and proved one).

As the DALMATIAN heuristic is more selective than previous ones, one may wonder if conjectures obtained with the DALMATIAN version of Graffiti are more often true than before. They are numbered from 700 upwards in "Written on the Wall". Pujol [142] studied 12 conjectures, in that range pertaining to cubic graphs. For that purpose he used the AutoGraphiX system (see Section 4 below) in interactive mode together with a program for cubic graph enumeration, due to Brinkmann [29]. 5 out of the 12 conjectures could be refuted. For the other ones, it was shown that a minimal counter-example would have at least 18 vertices. While this is a small sample, it nevertheless indicates that the proportion of false conjectures obtained with the DALMATIAN version of Graffiti, and after elimination of false or trivially true conjectures by both automated and non-automated methods may still be large.

3.7 Minuteman and Discriminant Analysis

The Minuteman version of Graffiti [86] is designed to solve problems of discriminant analysis, i.e., separating entities from given sets by values of a function, which corresponds geometrically to a surface, often a hyperplane. A motivating application was to discover stability sorting patterns of fullerenes. An additional routine works as follows ([85] p.21):

> "To study conjectures, objects are sorted by the difference between both sides of the inequality and sometimes when this is done for fullerene conjectures they show a conspicuous pattern by displaying the known stable examples on the top of the list and those with the largest sum of eigenvalues (i.e., presumed candidates for the least stable) at the bottom".

We do not discuss the chemical relevance of the patterns and conjectures so found here. Regarding the routine, note that checking if there is a pattern in the one-dimensional data obtained for a conjecture is done visually. It could of course easily be automated and simple statistical tests applied.

Now, if computer-assisted or automated systems for conjecture-making in graph theory are still rare, the situation is completely different in discriminant analysis. Indeed, this is a well established field, beginning in statistics at least 65 years ago [91] and presently central to data mining. Automated methods to find separating planes or surfaces in low or high dimensional spaces are operational for various criteria. Let us just mention that if perfect separation by a plane is possible this can be done by linear programming [129] and that otherwise one can use *decision trees* [143] [23], *support vector machines* [46], *logical analysis of data* (LAD, [19]) or other methods.

So while Minuteman is far from the state of the art in discriminant analysis, it suggests the interest of using more powerful discrimination methods in graph theory. In a similar vein, Colton [52] recently stressed that mathematics could be viewed as a new field for data mining. Some techniques using Boolean variables appear particularly well-suited to the case of graph problems, e.g. LAD and decision trees.

RP10. Apply decision trees and LAD to discriminant problems in graph theory. Such problems may be mathematical ones (e.g. belonging or not to a particular class of graphs) or applications based on measurements relative to the problem under study (as in the fullerene example discussed above). Compare results with those of other conjecture-making systems. □

RP11. Study criteria for approximate separation in graph theory using various discriminant analysis methods, both for mathematical problems and for applications. Examine when and how an approximate separation (e.g. a linear one) can lead to an exact one (e.g. by restricting the class of graphs considered or barring exceptional cases). □

3.8 Pedagogical versions of Graffiti

Computer systems have long been used with success in teaching graph theory. This was already the case of GRAPH [59], Chinn [41] reports on her use of INGRID for that purpose and the "CABRI-graphes" system [38] developed in Grenoble led to the widely distributed "CABRI-géomètre" package.

Recently, versions of Graffiti devoted to teaching graph theory with an active pedagogy were developed. They met with equal success when used in special project classes. Pepper [137] gives an enthusiastic record of his discovery and use of Graffiti, and Chervenka [40] describes more briefly how she used De La Vina's Graffiti.pc [68].

The main difference with previous versions of Graffiti is in use: initially the database of graphs is empty. When a first graph is entered, conjectures are formulated and the corresponding invariants studied. These conjectures are often easy to prove or refute. For that reason, more work is asked from the students than merely to provide a counter-example: they are requested

(i) if the conjecture is refuted, to find a smallest counter-example in terms of number of vertices and, as a secondary criterion, of number of edges;

(ii) if the conjecture is true, to determine whether it is NP-hard or not to determine if a graph G satisfies the relation (assumed to be an inequality) as an equality.

While such tasks are initially easy to accomplish, their difficulty will augment with the number of graphs in the database. Graffiti does not contain routines to do them automatically or in computer-aided mode. At an early stage, this is reasonable if one wants the students to practice their refuting and proving skills. Later, they might want to have some help, which could be provided by a system such as GRAPH or AGX.

PE7. Add to the pedagogical versions of Graffiti a program for visualizing graphs on screen, modifying them online and computing automatically a series of invariants or formulae involving invariants. □

While the main advantage of such an enhancement would be to make interaction with the system easier and more effective, another one would be to show students that tedious computations may be delegated to the machine, so that they may concentrate on reasoning.

PE8. Add to the pedagogical versions of Graffiti a routine similar to AGX's function for evaluating invariants subject to constraints: then use it if the relation is false to find smallest counter-examples by parametrizing on numbers of vertices and edges and attempting to find a graph which does not satisfy the given relation. □

Such an enhancement should be made available only after students have tried to find minimum counter-examples on their own, and submitted them to the system.

PE9. With the same function, and assuming that the relation is true, find graphs which satisfy it as an equality, to help estimate how difficult it is to recognize them. □

The same comment as for PE8 holds here too.

In the "Little Red Riding Hood" version, the task of proving true conjectures is ignored. It is then claimed ([85] p. 18) that it is "an offshot aimed at fully automating the program apart from the invention of concepts".

Observe however that no additional functions have been automated since the last general version. Some tasks have been abandoned (proving true conjectures) and some others, done by hand, made harder (finding counter-examples which are smallest possible). As previously, the fact that generation of counter-examples is not automated is overlooked in the comment cited above. In fact, steps of automated generation of conjectures alternate with steps of finding smallest counter-examples, in what appears to be a typically interactive man-machine process.

3.9 Complexity and the $P = NP$ problem

Some considerations on the condition for stopping of "Red Burton" and its relation to complexity issues, i.e., the $P = NP$ problem, are given in ([85] p.24). As many researchers (e.g. Smale [158]) consider this last problem as the most important one of computer science we examine this text in detail.

A first paragraph tells us that:

> "Once in a while it may happen that all conjectures of a given round are true. The natural interpretation of this situation-called *bingo*- is that for every object (under consideration, not just those in the database of the program) there is

> a conjecture made in this round such that the left and the right sides of the inequality have the same value for this object. Unless this indeed is the case, supplying the program with a counter-example to this situation will still break the stalemate and one can proceed to the next round."

One may wonder if this interpretation is "natural"; the set of objects (graphs) under consideration may be very large, and in some cases, discussed below, infinite. Extrapolating the fact that there is a tight relation for every object in the database to this much larger set is a very risk step. But, clearly, as indicated, if this property does not hold and one can find an object for which there is no tight relation, one can proceed to the next round. Note that this task is different from those of Red Burton as described earlier in [85], in which one asks for objects which *refute* a conjecture, not for objects for which no conjecture is *tight*.

The next paragraph of the text begins as follows

> "Most of the interesting runs of the program will yield at least one false conjecture in each round. This will always happen if the leading invariant L is NP-hard and all the remaining invariants from N are polynomially computable. These versions of the program will run forever modifying some of its conjectures after each round. Some of the conjectures are cyclically and some are continuously repeated in rounds providing more and more experimental evidence for their correctness."

The second sentence does not appear to be true. No proof is given and a counter-example is easy to find: let the leading invariant L be the independence number α, the class of graphs under consideration being all non-trivial graphs, i.e., all graphs with at least one edge and the only graph in the database in the first round a star, say S_4. Then the system will give the relation $\alpha(G) \leq n - 1$ and the first round will end without a false conjecture. If another graph, say C_4, is introduced, there will be an infinite, incomplete second round, still without a false conjecture. Moreover, if as stated at the beginning of the third sentence

> "These versions of the program will run forever ...",

it does not seem they can lead to a "bingo" which implies that they stop. This contradicts the first statement of the remainder of this second paragraph, next reproduced, which gets to the main question:

> "One can still end up with a correct *bingo* but this would imply $P = NP$ in which case the more appropriate term for the situation would be "big bang". Penrose does not question that in a sense a machine's insights may be superior to human. It is not unthinkable that $P = NP$ can be proved, because machines may conjure up hundred of novel radius, average distance, residue, and δ-like bounds, constituting a valid bingo."

For the last sentence to make sense, one should write "big bang" instead of "bingo". Then, one may wonder if it is true, and if it has information content. Recall from elementary

logic that B holds because of A is equivalent to the implication $A => B$ and means that A is false or B is true. Let B denote the proposition "It is not unthinkable that $P = NP$". As long as it has not been proved that $P \neq NP$ this is a tautology, i.e., certainly true. But then the implication holds regardless of the antecedent A, i.e., one can adopt for A any statement whatsoever, true or false, instead of "machines may conjure...". So for the last sentence to have information content, one must show that a "big bang" has some *plausibility* not that it is merely *possible*. For this to be done along the proposed lines one must show it is plausible that one can:

(a) find relations given sets of graphs (various systems do this);

(b) find in each round graphs which are not tight for any of the relations involving the chosen invariant and direction in the database of conjectures. This task increases in difficulty with the size of that database. Moreover, such graphs are likely to be increasingly and finally enormously large (clearly no machine could find such graphs if they must have billions of vertices).

(c) prove that *all* relations considered in the last round are true;

(d) prove that there exists *no* graph under consideration, the set of which is necessarily *infinite* for the problem under study to bear upon $P = NP$, for which none of the relations considered in the last round are tight.

Clearly, this proof scheme is incredibly difficult to carry out. Except for step (a) the necessary steps are not even listed, nor of course discussed. As no argument is provided for a "big bang" to be plausible, the last sentence of the cited text has no information content. In other words, that "machines may conjure up hundred of novel ... bounds" provides no argument of any weight for or against $P \neq NP$.

4 AutoGraphiX (AGX)

4.1 Uses and structure

As mentioned in the introduction AGX has several aims. We focus here on computer-assisted and automated conjecture-making. Indeed, AGX can be used in both modes, and the steps involved as well as their sequence must be carefully distinguished.

When working in computer-assisted mode, AGX's follows the following ones.

Step 1. Problem formulation.

Step 2. Obtention of a set extremal or near-extremal graphs for the chosen objective subject to the stated constraints.

Step 3. Visual display of the graphs found and parametric value curves.

Step 4. Interactive improvement of graphs which do not appear to be optimal.

Step 5. Interactive derivation of structural and algebraic conjectures.

When AGX is used in automated mode, steps 3 to 5 are replaced by the following ones

Step 6. Recognition of extremal graphs belonging to known families.

Step 7. Determination of linear equations between invariants associated with all or some subset of the external graphs obtained by the numerical method.

Step 8. Determination of linear inequality relations between invariants by the geometric method.

Step 9. Determination of linear or nonlinear relations between invariants by the algebraic method.

Step 10. Results: output external graphs found, families to which they belong, parametric curves of values for the objective, and conjectures found.

Note that not all methods for finding conjectures automatically need be used in the same experiment: one of steps 7, 8 or 9 suffices; step 6 is also optional except if step 9 is used.

4.2 Problem formulation

When the aim of using AGX is conjecture-making, one leading invariant is usually selected and others (most often n and m) used as parameters. Moreover, the class of graphs considered is specified by constraints, which will be added to the objective function with large coefficients (as in Lagrangian relaxation). Such coefficients must be chosen to be sufficiently large to exclude any graph not in the class considered; if this is not possible, a large value indicating a contradiction will be obtained.

Moreover, in some cases it is necessary to add a secondary criterion or progressive series of weights in order to transform the graphs in directions which will tend to satisfy the constraints.

An example occurred at Graph Theory Day 42, where after a presentation on *Computers in Graph Theory* [104] a demonstration of AGX was made. Cowen [53] asked for graphs with a maximum number of K_4 for a given number of K_3. This last number was chosen as a parameter and the number of K_4 maximized, which led on the spot to rediscovery of a series of extremal graphs for those parameters. Running AGX for a longer time gave a series of further extremal graphs of larger size.

At another (early) demonstration of the system, Seymour [151] asked for cubic graphs of diameter 3 with a maximum number of vertices. A first try where the diameter was minimized under the constraint that all degrees be equal to 3 yielded examples with 14 and 16 vertices but not more (the constraint on the degree was imposed by penalizing the numbers of vertices of degree smaller or greater than 3 increasingly with their distance to that value). Adding as secondary criterion minimization of the average distance, and so smoothing the objective function, led to cubic graphs of diameter 3 with 18 and 20 vertices in 35 seconds and 1 minute respectively; the latter graph is optimal.

Presently AGX disposes of about 60 invariants to be used in the objective function and constraints. They are *order*, *size*, *independence number*, *chromatic number*, *chromatic index*, *minimum degree*, *maximum degree*, *average distance*, *degrees of the vertices*, *eigenvalues of the adjacency matrix* and others.

However, this is not a very large set, as compared with those of GRAPH, Graffiti, LEDA or other systems.

PE10. Add to AGX routines to compute the main graph theoretic invariants not yet included (e.g. *matching* number, *domination* number, etc) □

Graph invariants are invented every day, so if AGX is to accommodate all needs of the users, it must let them add their own routines for their favorite invariants.

PE11. Construct a version of AGX in which the user can add routines to compute new invariants. □

This enhancement is being implemented in the new version, AGX2, of AGX, which is currently being built.

At present, standard algebraic expressions can be taken in the objective function and constraints as well as some simple graph transformations such as complementation. Other operations should be made possible.

PE12. Add to AGX routines for the main graph operations, such as sum or product of graph, etc, as done in GRAPH.

4.3 Finding extremal graphs

The principle of *AGX* is to use heuristic optimization to find a family of extremal or near-extremal graphs for some objective, subject to constraints, then to exploit the corresponding information.

Heuristic optimization in *AGX* follows the Variable Neighborhood Search (VNS) metaheuristic [108], or framework for building heuristics. VNS exploits the still rather new idea of systematic change of neighborhood within the search. This is done in two ways: first in a descent routine, called Variable Neighborhood Descent (VND), which leads to a local optimum, and, second, in a systematic effort to get away from this local optimum by applying increasingly strong perturbations and descents.

Rules of VNS are as follows:

0. Select the set of neighborhood structures $N_k, k = 1, \ldots, k_{\max}$ that will be used in the search for a better local optimum, and a stopping condition. Find an initial solution (or graph) x.

Repeat until the stopping condition is met:

1. Set $k = 1$;
2. Until $k = k_{\max}$, repeat the following steps
   (a) (*shaking*) generate a point x' at random from the k^{th} neighborhood of x (i.e., $x' \in N_k(x)$):
   (b) (*descent*) Apply the Variable Neighborhood Descent routine with x' as initial solution: denote by x'' the local optimum obtained;
   (c) (*improvement or continuation*) If the solution x'' so obtained is better than the best known one x, move there ($x \leftarrow x''$) and continue the search within $N_1(x)(k = 1)$; otherwise set $k \leftarrow k + 1$.

The stopping condition may be a maximum number of iterations, a maximum CPU time or a maximum number of iterations or CPU time since the last improvement.

Rules of VND are as follows:

0. Select the set of neighborhood structures $N'_k, k = 1, 2, \ldots, k'_{max}$ that will be used in the descent. Consider an initial solution x.

Main step: Set $k = 1$ and $i = FALSE$ (*improvement indicator*).

Until $k = k'_{max}$, *repeat the following steps:*

(a) Find the best neighbor x' of x in $N'_k(x)$;

(b) If the solution x' so obtained is better than x, set $x \leftarrow x'$ and $i = TRUE$;

(c) Set $k \leftarrow k + 1$;

(d) if $k = k'_{max}$ and $i = TRUE$ set $k = 1$.

In words, VND applies a series of transformations to the current graph, keeping each time that transformation giving the best improvement. It there is no improvement within the current neighborhood, VND proceeds to the next one. If there is no further improvement when considering all neighborhoods in turn, VND stops; otherwise it begins again at the first neighborhood.

Moves corresponding to the different VND neighborhoods in AGX are the following: *rotation* of an edge, *deletion* of an edge, *addition* of an edge, *move* of an edge, i.e., deletion plus addition, *detour*, i.e., removal of an edge and addition of two edges between endpoints of the deleted one and a vertex not adjacent to either of their endpoints, *short cut*, i.e., the operation that is the reverse of detour, 2-*opt*, i.e., removal of two non adjacent edges, and addition of two different edges connecting the endpoints of the removed ones: *add pendant vertex*: i.e., add a new edge from an old vertex to a new one; *delete vertex* of bounded degree and all adjacent edges.

The neighborhoods rotation, addition, deletion and move are the most frequently used, and the least time consuming ones.

If all moves within a neighborhood are examined before choosing the last one, only graphs of moderate size may be considered, particularly if the objective function to be computed after each potential move is hard to evaluate. A speed-up can be obtained by using a "first improvement" instead of a "best improvement" rule.

The choice of moves is presently left to the user (with a standard option of using them all). However, this choice could be automated:

PE.13 Add a routine which evaluates the effect of all moves during an initial period, then selects for continuation of the search those which proved to be the most efficient. □

One could also try to find new moves systematically:

PE.14 Construct moves by all possible transformations on a small graphs (e.g. with 4 vertices). Eliminate redundant ones which are the same as others up to symmetry. □

Contrary to VND, which uses systematically a series of different local moves, VNS makes random use of more global moves, often deriving from a simple principle. The most

frequently used one is to repeat a move k times, e.g., one first moves an edge chosen at random (a move in $N_1(x)$) then 2 (a move in $N_2(x)$) and so on.

VNS appears to be quite powerful, i.e., very often, but not always, it gives extremal graphs. Cases where it does not give the best graph, which can often be recognized by comparison with graphs obtained for close values of the parameters, can be exploited to define new neighborhoods.

RP12. Systematically explore cases in which the neighborhoods of VND and VNS are not enough to find consistently extremal graphs. Define new neighborhoods accordingly and study the complexity of implementing them.

4.4 Display of results

Results of *AGX*, when used in interactive mode are of two types:

a) Extremal or near-extremal graphs;

b) Parametric curves of values of the objective function.

Extremal graphs can be visualized on screen or printed. Drawings can be modified interactively by moving vertices; classes of vertices or of edges can also be highlighted, in various colors (e.g. edges which are critical for some invariants, edges of a spanning tree or a shortest path tree, vertices of various degrees, ...)

Up to now only simple tools of graph drawing have been implemented in *AGX*, i.e., a specialized routine for representation of trees with edges parallel to the axes, a "spring" type heuristic to avoid cluttering parts of the drawing with closely spaced vertices, and a few more. As the field of graph drawing is very active (see e.g. Di Battista et *et al.* [70] [71]) further results obtained there could be exploited. Note however that the frequently adopted criterion of minimizing edge crossings does not seem adequate for AGX's needs. Easy recognition of subgraphs of one or another type (*cliques*, *cycles*, ...) seems more important.

RP13. Study precise needs of AGX for graph drawing and how they can be met by methods of that field. In particular consider ways to make structure (particular subgraphs, graphs formed from them) visible in individual graphs as well as in sequences of graphs.□

While there may be no closed-form formulae for some invariants on general graphs, there may be some for particular classes of graphs. Recognizing them can lead to conjectures, as discussed further below.

Curves of values can be represented in three dimensions, corresponding usually to some invariant i_1, n, and m. These curves can be rotated, superposed, isolated, etc... Moreover, graphs corresponding to particular points on these curves e.g. minima or maxima can be displayed in a window. Finally, if one has some idea about a conjecture it can be introduced into AGX, checked, displayed with the curves, and both the differences in ordinates and the points of contact highlighted.

These facilities should be extended to higher dimensions.

PE15. Add to *AGX* a routine for projection of points in R^p with $p > 3$ but moderate, corresponding to extremal graphs, on subspaces with 2 or 3 dimensions. □

4.5 Recognizing structure interactively

When visualizing the extremal graphs obtained with AGX, it is not uncommon to find that

(i) they belong to some well-known family, e.g. paths, circuits, trees, stars, bipartite, complete,...,

or

(ii) they have some recognizable but more complicated structure.

There may be some exceptions among them, and one should then find out whether this is due to the VNS heuristic not finding the (or an) extremal graph for the corresponding values of the parameters or to the particularities of the objective function under study.

Usually, significant differences in structure or an outlier position with respect to the curve of values make such exceptions conspicuous. One can then deduce from close examples what might be the true extremal graph for those parameter values and build it by moving edges with the mouse; then the system will compute its value and, if it is better than the previous near-extremal graph, substitute them.

This step is not mandatory, and not used when applying AGX in automated mode (outliers may be removed in other ways, see below). However, it could be automated, or at least more automated than it presently is.

PE16. Augment the number of routines for recognition of classes of graphs in AGX.□

PE17. Add a routine which will test if extremal graphs frequently belong to some parameterized family; for those parameter values for which it is not the case, compute their value and substitute them if there is an improvement. □

Note that such developments are close to those needed in the third (algebraic) way to find conjectures automatically (see below). The automated parts correspond to step 6 of AGX, which will not be discussed further.

4.6 Obtaining conjectures interactively

Conjectures most often made have the two following forms (see also [113])

(i) *Algebraic relations* between graph invariants, valid for some class of graphs (e.g. all graphs, connected, bipartite, split, stars, trees, complete...)

(ii) Description of the *structure* of extremal graphs (i.e., of a known or new class of graphs) or of a subset of them.

Conjectures of the former type can be obtained from the parametric curves of values of the objective. Consider for instance the *energy* E of a graph defined [31] [97] [98] as

$$E = \sum_{i=1}^{n} |\lambda_i|.$$

Minimizing this function with parameters n and m, then superposing the curves of $E(m)$ for fixed n shows very clearly all values to be above a parabola. Its equation is then readily found and leads to the lower bound [31]

$$E \geq 2\sqrt{m}.$$

Moreover, the equation of this curve can be entered in AGX, which represents it in the plane of values and highlights points where it is attained, i.e., graphs reaching the bound, as well as differences for other points. One can thus see if the bound is sharp and if it remains so over the range of parameter values or not. In the case discussed, when m becomes large the curve lies increasingly below observed values. Then, looking at curves for one value of n at a time suggests a linear lower bound for each, from where the inequality

$$E \geq \frac{4m}{n}$$

follows. It is sharp for fewer values than the first bound, though still sharp several times.

Conjectures of the second type are obtained by examining the graphs obtained and, possibly, exploiting conjectures on these graphs obtained automatically (see below). Sometimes, results are straightforward. For instance minimizing with AGX the energy of unicyclic graphs (a problem of interest to chemists) led to extremal graphs which were cycles for $n \leq 7$ or n =9, 10, 11, 13 and 15 and 6-cycles with an appended path for all other values of n considered. The natural conjecture that these and only these graphs were the true extremal ones [31] has recently been partially proved [99] [117].

4.7 Numerical method of conjecture-making

We now turn to the automated mode of using AGX and consider the three ways in which this has been done (up to now). A first method uses the mathematics of principal component analysis to find resemblances between objects, in the form of affine relations they all satisfy, instead of differences as usually done.

The method works as follows [35] [36]:

(a) Find extremal or near-extremal graphs for some objective with AGX;

(b) Filter this set to remove outliers (optional but often useful);

(c) Compute values for a set of invariants on all remaining graphs;

(d) Center the vectors of values for each invariant (thus transforming the problems of finding affine relations into that of finding linear ones);

(e) Compute the variance-covariance matrix V between centered vectors;

(f) Diagonalize V, with, however, some empty lines if there are relations. In the resulting matrix V', $Dim(I_m(V))$ lines contain non-zero terms and correspond to independent variables. The remaining $n - Dim(I_m(V))$ lines contain only zeros and correspond to dependent variables which may be expressed as linear combinations of the independent ones. These relations form a basis of the null-space of V. Using the initial data one can then compute the right-hand sides of the corresponding affine relations.

To illustrate, consider the irregularity $irr(G)$ of a graph G as defined by Albertson [3]: let the *imbalance* imb_{ij} of edge (v_i, v_j) of G be defined by

$$imb_{ij} = |d_i - d_j|$$

and the irregularity $irr(G)$ of (G) by

$$irr(G) = \sum_{i,j|(v_i,v_j)\in E} imb_{ij}.$$

Applying AGX [107] led automatically to the following conjectures valid for graphs G with maximum irregularity:

$$\begin{aligned} r(G) &= 1 \\ \chi(G) &= \omega(G) \\ n &= \Delta + 1 \\ \alpha(G) &= -\omega(G) + \Delta + 2, \end{aligned}$$

from where it follows that

$$\alpha(G) + \omega(G) = n + 1$$

which implies that the extremal graphs are split graphs, i.e., graphs consisting of a clique, a disjoint independent set and edges joining vertices of the clique to those of the independent set. (For further use of this information and of the external graphs found, see [107]).

Several comments are in order. First, note that the algorithm described takes polynomial time: if the number of graphs considered is fixed and t invariants are computed, it requires $O(t^3)$ time.

Second, observe that it gives relations for *subsets* of the set of invariants considered, not necessarily for the whole set. So the combinatorial problem of finding the right subset is avoided. This implies that given a sufficiently large set of graphs of some class, and sufficient computing time, one could find a basis of affine relations among a large set of invariants (maybe several hundred of them). Should such relations exist and be up to now unnoticed, it would prove that interesting relations may be found without focussing on a particular problem, or domain (such as e.g. problems of distances in graphs).

Third, the algorithm subsumes some other ones, which have met with success. For instance the BACON algorithm developed by Simon and co-workers [122], gives *rational reconstructions* (or possible reasonings) for great discoveries of the past in physics and chemistry. It uses four rules, given a set of observations involving several variables:

(a) If a variable is constant, a law has been found;

(b) If a variable is a linear function of another one, a law has been found;

(c) If a variable increases while another one decreases, add a new variable equal to their product, and iterate;

(d) If a variable increases while another one increases, add a new variable equal to their ratio and iterate.

BACON rediscovered Kepler's third law, in three iterations only, as well as several other famous ones. The numerical method of AGX reproduced these results in much less computer time [36]. These laws are expressed as monomials, i.e., products of variables with integer powers. Taking logarithms gives affine functions.

However, there are many more complicated cases: Langley *et al.* [122] have observed that laws in chemistry may take a more general form, the logic of which had, apparently, not yet been studied [155]: in addition to the variables, there are substance-specific constants, such as e.g. *specific heat*. BACON could be extended to this case.

RP14. Study how to extend the numerical method of AGX in order to apply it to problems with both variables and substance-specific constants. □

Fourth, one would clearly like to extend the discovery of conjectures to more general cases than affine relations. Note first that inequalities are obtained in a straightforward way: it suffices to check on which side lie graphs which are not extremal for the objective under study. Then one might add, as new variables, products of variables, or simple powers such as squares, cubes, inverses, square roots and the like.
All this increases the number of variables, and thus augments the number of graphs needed to obtain relations, as well as computing time, but does not change the method itself.

If e.g. only products of two variables are considered it is still possible to consider a few tens of variables. One would like to do better than this brute-force approach, and in view of results obtained by *support-vector machines*, this seems to be possible.

RP15. Study selection of product and power terms in finding nonlinear conjectures between invariants in graph theory. Devise corresponding heuristics. □

Fifth, even more general sets of relations involving *signomial functions* (polynomials in several variables with arbitrary powers and signs) have been studied by using neural networks. These have reconstructed with fairly good precision a set of such equations.

RP16. Compare conjecture-making by the numerical method of AGX and by neural networks; define hybrids where neural networks are used to find the form of the relations and AGX to find the precise values of coefficients. □

Sixth, to determine affine relations, which are equalities, numerical precision is required, and hence control of errors. Standard tools of numerical analysis are used to do so, but the guarantee of finding all affine relations is not complete. To attain such a goal one would need computations in error-free arithmetic (i.e., making computations with rational numbers using a sufficient number of digits to avoid all approximation errors), which has been used in solution of equations associated with Euler sums [12], but are very time consuming.

4.8 Geometric method of conjecture-making

Consider a set of extremal graphs for some objective; they correspond to points in the $\mathbb{R}^p$ space of invariants (or in a sub-space of selected invariants), each of which is associated with

one of the p axes. Then constructing the convex hull of these points with a gift-wrapping algorithm (as e.g. implemented in the package of Avis and Fukuda [10]) immediately yields a set of conjectures in the form of linear inequalities: for each invariant, faces passing below all points, or above all points correspond to lower and upper bounds.

To illustrate, consider chemical graphs, in which $\Delta \leq 4$ due to the valency of carbon. The geometric method of AGX could find the two following relations in a very small computing time:

$$Ra(G) \geq \frac{n}{3} + \frac{m}{12}$$

and

$$Ra(G) \geq \frac{1}{4}(m + n_1).$$

The main difficulty with this approach is to avoid undue extrapolation. In the case of a function of a single invariant say $i_1(n)$, if it is concave, just the next graph could disprove the conjecture.

Therefore the conjectures obtained are systematically tested by looking for the few extremal graph(s) following those used to find them. Also the *touch number* criterion discussed above is of interest here: if the inequality found is sharp at only a couple of points, it appears to be of little interest. Conversely, if it is sharp for many or even most values of the parameters, as is the case for the two relations just cited, it is clearly interesting.

One would then like to obtain often sharp relations even when the relationships between invariants of extremal graphs are nonlinear. Again this could be done by introducing new variables, i.e., going to a higher dimensions. There are some limitations here, as gift-wrapping algorithms may become very time consuming with only 10 variables or so.

RP17. Examine how to transform functions in order to get nonlinear relations through the geometric method. Compare results with those of the numerical approach for the same problems.

4.9 Algebraic method for conjecture-making

The principle of the third method is to recognize extremal graphs for some objective function, then to use relations between invariants valid for those classes of graphs in order to obtain new relations, which are conjectured to hold in general.

To illustrate, consider the objective function $Ra(G) - \bar{l}(G)$, (which corresponds to conjecture Graffiti 3, i.e., $\bar{l}(G) \leq Ra(G)$). Minimizing this relation systematically gave stars, for which the Randić index is equal to $\sqrt{n-1}$ (and is minimum for fixed n as shown by Bollobas and Erdös [18]) and the average distance is $2 - \frac{2}{n}$. This leads to the conjecture *For any connected graph G*

$$Ra(G) - \bar{l}(G) \geq \sqrt{n-1} + \frac{2}{n} - 2,$$

which strengthens Graffiti 3. If true, the new bound is sharp for all $n \geq 1$.

The difficulty of this method is the large amount of information needed: on the one hand, specific algorithms are required to recognize to which class belong extremal graphs, and on the other hand a database of relations between graph invariants is needed for each class considered. Presently this method is working in experimental mode.

PE18. Extend the set of graph recognition routines of AGX. □

PE19. Extend the database of relations between graph invariants. □

PE20. Couple the algebraic method with Mathematica or Matlab to simplify the relations obtained. □

Clearly it will not always be the case that extremal graphs all belong to a single well-defined class, for which relations are known. A first difficulty is then that one will have to use lower or upper bounds (e.g. if all one can find is that extremal graphs are trees), although that would not change the approach too much.

PE21. Extend the algebraic approach to manipulate bounds rather than equalities between invariants. □

Another extension would be to recognize the various classes of graphs which are extremal for some values of the parameters (a problem already evoked above) and modify again the way bounds are computed and relations obtained.

Finally, once again one should compare methods.

RP18. Compare systematically results of the three methods proposed for automated conjecture-making on the same set of problems, including some which led to well-known graph theorems. Deduce from this comparison intimations about what makes a relation difficult to find for one or all of them. □

5 Conclusions

Computer-assisted and automated conjecture-making in graph-theory appears to be very successful and has led collectively to more than 200 papers research reports and theses. This makes it probably the most active subfield of discovery science.

Three systems are operational and largely used: GRAPH, Graffiti and AGX. Their principles are different: interactive computing, generation of a priori conjectures and selection amongst them, heuristic optimization to get extremal graphs and deduction of conjectures from them. All three have large parts which are automated, but only the last can presently be used in (fully) automated mode, that is with a problem statement unaccompanied by further information, no human intervention between problem statement and reading the final results as well as no selection among results so obtained. Note that this is not the only way to use this system, nor necessarily the most efficient one, as interactive modification of the extremal graphs obtained may give insight on how to prove the conjectures it delivers.

All three systems (and others) are susceptible of fuller automation in the near future. A series of suggestions on 21 possible enhancements are given in this paper, as well as a list of 18 more general questions, or research paths, of possible interest to the whole field.

As a final point, observe that the conjectures considered in this paper are mainly algebraic inequalities (or, in some rare case, equalities) among graph invariants. As discussed more fully in [113] there are many other forms which interesting conjectures in graph theory can take. So there is plenty of room for further achievement in this young and promising field.

Acknowledgments:

This paper was written in part during a visit to SMG, University of Brussels; support of the *Research in Brussels* program is gratefully acknowledged as well as NSERC grant # 105574-98. Thanks to Mustapha Aouchiche, Gilles Caporossi, Hadrien Mélot and Dragan Stevanović for discussions as well as Dragos Cvetković and Siemion Fajtlowicz for correspondence which helped to clarify issues discussed.

References

[1] ABELEDO, H., and ATKINSON, G.W. The Clar and Fries problems for benzenoid hydrocarbons are linear programs. In: *Discrete Mathematical Chemistry*, P. Hansen, P. Fowler, and M. Zheng, Eds., vol. 51 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, Providence RI, 2000, pp. 1–8.

[2] ABELEDO, H., and ATKINSON, G.W. Polyhedral combinatorics of benzenoid problems. Proceedings of IPCO VI, Houston (1998). *Lecture Notes in Computer Science*, New-York, Springer 1412.

[3] ALBERTSON, M.O. The irregularity of a graph. *Ars Combinatoria*, 46 (1997) 215–225.

[4] ALON, N. and SEYMOUR, P. A counter-example to the rank-coloring conjecture. *Journal of Graph Theory*, 13 (1989) 523–525.

[5] AOUCHICHE, M. www.gerad.ca/AGX. A Bibliography on AutoGraphiX, its Results and Related Topics (forthcoming).

[6] AOUCHICHE, M., CAPOROSSI, G., and HANSEN, P. Variable neighborhood search for extremal graphs 8. Variations on Graffiti 105. *Congr. Numer.*, 148 (2001) 129–144.

[7] APPEL, K., and HAKEN, W. Every planar map is four colorable. Part I. Discharging. *Illinois J. Math.*, 21 (1977) 429–490.

[8] APPEL, K., and HAKEN, W. Every planar map is four colorable. Part II. Reducibility. *Illinois J. Math.*, 21 (1977) 491–567.

[9] APPEL, K., and HAKEN, W. Every planar map is four colorable. *Contemp. Math.*, 98 (1989) 1–743.

[10] AVIS, P., and FUKUDA, K. *lrs home page*; *cdd and ccd plus page.*

[11] BAILEY, D. Integer Relation Detection. *Computing in Science and Engineering* 2 (2000) 24–28.

[12] BAILEY, D.H., BORWEIN, P.B. and PLOUFFE, S.A. New formulas for picking up pieces of Pi. *Science News*, 148 (1995) 279.

[13] BAILEY, D.H., BORWEIN, P.B. and PLOUFFE, S.A. On the rapid computation of various polylogarithmic constants. *Mathematics of Computation*, 66 (1997) 903–913.

[14] BATTITI, R., and PROTASI, M. Reactive local search for the maximum clique problem. *Algorithmica* 29 (2001) 610–637.

[15] BEEZER, R.A., RIEGSECKER, J. and SMITH, B.A. Using minimum degree to bound average distance. *Discrete Mathematics*, 226 (2001) 365–377.

[16] BERGE, C. Färbung von Graphen deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung), *Wissenschaftliche Zeitschrift, Martin-Luther-Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe*, (1961) 114–115.

[17] BERGE, C. Perfect graphs I. *Six papers on graph theory.* Indian Statistical Institute, Calcutta (1963).

[18] BOLLOBAS, B. and ERDÖS, P. Graphs of extremal Weights. *Ars combinatoria*, 50 (1998) 255–233.

[19] BOROS, E., HAMMER, P.L., IBARAKI, T., MAYORAZ, E., and MUCHNIK, I. An implementation of logical analysis of data. *IEEE TRANS. On Knowledge and Data Engineering*, 12 (2000) 292-306.

[20] BORWEIN, J., BRADLEY, R. Empirically determined Apéry-like formulae for zeta (4n+3). *Experimental Mathematics* 6 (1997) 181–194.

[21] BORWEIN, J.M., LISONĚK, P. Applications of integer relation algorithms. *Discrete Mathematics* 217 (2000) 65–82.

[22] BOUVIER, A., and GEORGE, M. *Dictionnaire des Mathématiques.* Presses Universitaires de France (1979). (in french)

[23] BREIMAN, L., FRIEDMAN J., STONE, C.J. and OLSHEN, R.A. *Classification and Regression Trees.* Chapman and Hall (1984).

[24] BREWSTER, T.L., DINNEEN, M.J. and FABER, V. A computational attack on the conjectures of Graffiti: New counterexamples and proofs. *Discrete Mathematics* 147 (1995) 35–55.

[25] BRIGHAM, R.C., and DUTTON, R.D. INGRID: A software tool for extremal graph theory research. *Congressum Numerantium*, 39 (1983) 337–352.

[26] BRIGHAM, R.C., and DUTTON, R.D. A compilation of relations between graph invariants. *Networks*, 15 (1985) 73–107.

[27] BRIGHAM, R.C., and DUTTON, R.D. A compilation of relations between graph invariants. Supplement 1. *Networks*, 21 (1991) 421–455.

[28] BRIGHAM, R.C., DUTTON, R.D., and GOMEZ, F. INGRID: A graph invariant manipulator. *J. Symb. Comp.*, 7 (1989) 163–177.

[29] CAGE. The chemical and abstract graph environment. Homepage: http://www.mathematik.uni-bielefeld.de/∼CaGe/ .

[30] CAMPBELL, M., HOANE, A.J. and HSU, F.M. Deep Blue. *Artificial Intelligence* 134 (2002) 57–83.

[31] CAPOROSSI, G., CVETKOVIC, D., GUTMAN, I., and HANSEN, P. Variable neighborhood search for extremal graphs 2. Finding graphs with extremal energy. *J. Chem. Inf. Comp. Sci.*, 39 (1999) 984–996.

[32] CAPOROSSI, G., DOBRYNIN, A.A., HANSEN, P., and GUTMAN, I. Trees with palindromic Hosoya polynomials. *Graph Theory Notes N.Y.*, 37 (1999) 10–16.

[33] CAPOROSSI, G., FOWLER, P.W., HANSEN, P., and SONCINI, A. Variable neighborhood for extremal graphs 7. Polyenes with maximum HOMO-LUMO gap. *Chemical Physics Letters*.

[34] CAPOROSSI, G., GUTMAN, I., and HANSEN, P. Variable neighborhood search for extremal graphs 4. Chemical trees with extremal connectivity index. *Computers and Chemistry*, 23 (1999) 469–477.

[35] CAPOROSSI, G., and HANSEN, P. Variable neighborhood for extremal graphs 5. Three ways to automate finding conjectures. *Discrete Mathematics*. (To appear).

[36] CAPOROSSI, G., and HANSEN, P. Finding Relations in Polynomial Time. In *XVIth International Joint Conference on Artificial Intelligence (IJCAI)* (Stockholm, 1999), vol. 2.

[37] CAPOROSSI, G., and HANSEN, P. Variable neighborhood search for extremal graphs 1. The system AutoGraphiX. *Discr. Math.*, 212 (2000) 29–44.

[38] CARBONNEAUX, Y., LABORDE, J.-N. and MADANI, M. Cabri-graphes: A tool for research and teaching in graph theory. In *Lecture Notes in Computer Science*. Vol. 1027, Berlin:Springer, 1995, pp. 123–127.

[39] CARO, Y. Colorability, frequency and Graffiti-119. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 27 (1998) 129–134.

[40] CHERVENKA, B. Graffiti.pc Red Burton Style – A Student's perspective. *preprint*, (2002).

[41] CHINN, P.Z. Discovery-method teaching in graph theory. *Annals of Discrete Mathematics* 55 (1993) 375–384.

[42] CHOU, S.C. Proving and Discovering Theorem in Elementary Geometrics using Wu's Method, Ph.D. Thesis, Department of Mathematics, University of Texas, Austin (1985).

[43] CHOU, S.C. *Mechanical Geometry Theorem Proving*. Mathematics and its Applications, 41, Dordrecht: Reidel, 1988.

[44] CHOU, S.C., GAO, X.S. The computer searches for Pascal conics. *Computers and Mathematics with Applications* 29 (1995) 63–71.

[45] CHOU, S.C., GAO, X.S., ZHANG, J.Z. A deductive database approach to automated geometry theorem proving and discovering. *Journal of Automated Reasoning* 25 (2000) 129–246.

[46] CHRISTIANI, N., and SHAW-TAYLOR, J. *Support Vector Machines*. Cambridge: Cambridge University Press (2001).

[47] CHUDNOVSKY, M., ROBERTSON, N., SEYMOUR, P., and THOMAS, R. Progress on perfect graphs, *Mathematical Programming* B 97 (2003) 405–422.

[48] CHUDNOVSKY, M., ROBERTSON, N., SEYMOUR, P., and THOMAS, R. The strong perfect graph tehorem, manuscript. http://www.gatech.adv/∼thomas/sqge.html .

[49] CHUNG, F. The average distance is not more than the independence number. *J. Graph Theory*, 12 (1988) 229–235.

[50] COLTON, S. Refactorable numbers – A machine invention. *Journal of Integer Sequences*, 2 (1999).

[51] COLTON, S. On the notion of interestingness in automated mathematical discovery. *International Journal of Human Computer Studies special issue on Machine Discovery*, 53 (2000).

[52] COLTON, S. Mathematics: A new domain for data mining. *IJCAI 01 Proceedings*, 2001.

[53] COWEN, R. Personal Communication at Graph Theory Day 42. DIMACS, Rutgers, November 2001.

[54] CVETKOVIĆ, D. Discussing graph theory with a computer, II: Theorems suggested by the computer. *Publ. Inst. Math. (Beograd)*, 33(47) (1983) 29–33.

[55] CVETKOVIĆ, D. Discussing graph theory with a computer, IV: Knowledge organisation and examples of theorem proving. In *Proc. Fourth Yugoslav Seminar on Graph Theory* (Novi Sad, 1983), pp. 43–68.

[56] CVETKOVIĆ, D. Discussing graph theory with a computer, VI: Theorems proved with the aid of the computer. *Cl. Sci. Math. Natur., Sci. Math.*, T. XCVII (1988), No. 16, 51–70.

[57] CVETKOVIĆ, D., DOOB, M., GUTMAN, I., and TORGASEV, A. Recent results in the theory of graph spectra. *Annals of Discrete Mathematics*, 36 (1988) 1–306.

[58] CVETKOVIĆ, D. JOVANOVIC, A., RADO SAVLIEVIĆ, Z. and SIMIĆ, S. Coplanar graphs. Univ. Beograd, Publ. Elektrotekn. Fak. Mat., 2 (1991) 67–81.

[59] CVETKOVIĆ, D., and KRAUS, L. "Graph" an expert system for the classification and extension of the knowledge in the field of graph theory, User's manual. Elektrotehn. Fak., Beograd, 1983.

[60] CVETKOVIĆ, D., KRAUS, L., and SIMIĆ, S. Discussing graph theory with a computer, I: Implementation of graph theoretic algorithms. *Univ. Beograd Publ. Elektrotehn. Fak, Ser. Mat. Fiz. No. 716 – No. 734* (1981) 100–104.

[61] CVETKOVIĆ, D., and PEVAC, I. Discussing graph theory with a computer, III: Man-machine theorem proving. *Publ. Inst. Math. (Beograd)*, 34(48) (1983) 37–47.

[62] CVETKOVIĆ, D., and PEVAC, I. Man-machine theorem proving in graph theory. *Artificial Intell.*, 35 (1988) 1–23.

[63] CVETKOVIĆ, D., and SIMIĆ, S. Graph theoretical results obtained by the support of the expert system "Graph". *Cl. Sci. Math. Natur., Sci. Math.*, T. CVII (1994), No. 19, 19–41.

[64] CVETKOVIĆ, D., and SIMIĆ, S. Graph theoretical results obtained with support of the expert system "GRAPH"– An extended survey. *(submitted)*

[65] CVETKOVIĆ, D., SIMIĆ, S., CAPOROSSI, G., and HANSEN, P. Variable neighborhood search for extremal graphs 3. On the largest eigenvalue of color-constrained trees. *Lin. and Multilin. Algebra*, 2 (2001) 143–160.

[66] DANKELMANN, P. Average distance and the independence number. *Discrete Applied Mathematics*, 51 (1994) 73–83.

[67] DE LA VINA, E. Bibliography on conjectures of Graffitti. http://cms.dt.uh.edu/faculty/delavinae/research/wowref.htm , 2000.

[68] DE LA VINA, E. Graffiti.pc. *Graph Theory Notes of New York*, XLII (2002) 26–30.

[69] DE LA VINA, E. Some history of the development of Graffiti. Submitted for publication, 2003.

[70] DI BATTISTA, G., EADES, P., TAMASSIA, R., and TOLLIS, I.G. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications 4*, 5 (1994) 235–282.

[71] DI BATTISTA, G., EADES, P., TAMASSIA, R., and TOLLIS, I.G. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[72] DOBRYNIN, A.A., ENTRINGER, R. and GUTMAN, I. Wiener index of trees: Theory and applications. *Acta Applicandae Mathematicae*, 66 (2001) 211–240.

[73] EPSTEIN, S.L. Ph.D. Thesis, Rutgers University, 1983.

[74] EPSTEIN, S.L. On the discovery of mathematical theorems. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (Milan, Italy, 1987), pp. 194–197.

[75] EPSTEIN, S.L. Learning and discovery: one system's search for mathematical knowledge. *Comput. Intell.*, 4 (1988) 42–53.

[76] EPSTEIN, S.L., and SRIDHARAN, N.S. Knowledge representation for mathematical discovery: Three experiments in graph theory. *J. Applied Intelligence*, 1 (1991) 7–33.

[77] ERDÖS, P., PACH, J., and SPENCER, J. On the mean distance between points of a graph. *Congressus Numerantium*, 64 (1988) 121–124.

[78] ERDÖS, P., FAJTLOWICZ, S., and STATON, W. Degree sequences in the triangle-free graphs, *Discrete Mathematics*, 92 (1991) 85–88.

[79] FAJTLOWICZ, S. Written on the Wall. A regularly updated file accessible from http://www.math.uh.edu/∼clarson/ .

[80] FAJTLOWICZ, S. On conjectures of Graffiti – II. *Congr. Numer.*, 60 (1987) 187–197.

[81] FAJTLOWICZ, S. On conjectures of Graffiti. *Discrete Math.*, 72 (1988) 113–118.

[82] FAJTLOWICZ, S. On conjectures of Graffiti – III. *Congr. Numer.*, 66 (1988) 23–32.

[83] FAJTLOWICZ, S. On conjectures of Graffiti – IV. *Congr. Numer.*, 70 (1990) 231–240.

[84] FAJTLOWICZ, S. On conjectures of Graffiti – V. In *Seventh International Quadrennial Conference on Graph Theory*. (1995), Vol. 1, pp. 367–376.

[85] FAJTLOWICZ, S. Toward fully automated fragments of graph theory. *Graph Theory Notes of New York*, XLII (2002) 18–25.

[86] FAJTLOWICZ, S. *Fullerene Expanders, a List of Conjectures of Minuteman*. Available from the author.

[87] FAJTLOWICZ, S. On conjectures and methods of Graffiti. In *Proceedings of the 4^{th} Clemson Miniconference on Discrete Mathematics*, Clemson (1989).

[88] FAVARON, O., MAHÉO, M., and SACLÉ, J.-F. On the residue of a graph. *J. Graph Theory*, 15 (1991) 39–64.

[89] FAVARON, O., MAHÉO, M., and SACLÉ, J.-F. Some eigenvalue properties in graphs (Conjectures of Graffiti-II). *Discrete Mathematics* 111 (1993) 197–220.

[90] FIRBY, P., and HAVILAND, J., Independence and average distance in graphs. *Discrete Applied Mathematics*, 75 (1997) 27–37.

[91] FISHER, R.A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (1936) 179–188.

[92] GALLAI, T. Maximum-minimum Satze uber Graphen (german). *Acta Math. Acad. Sci. Hungar.*, 9 (1958) 395–434.

[93] GANTER, B., and WILLE, R. *Formal Concept Analysis – Mathematical Foundations*. Berlin: Springer (1999).

[94] GLAS, E. The 'Popperian Programme' and Mathematics. Part 1: The Faillibilist Logic of Mathematical Discovery. *Studies in History and Philosophy of Science*, 32(1) (2001) 119–137.

[95] GLAS, E. The 'Popperian Programme' and Mathematics. Part 2: From Quasi-Empiriciam to Mathematical Research Programmes. *Studies in History and Philosophy of Science*, 32(1) (2001) 355–376.

[96] GRIGGS, J.R., and KLEITMAN, D.J. Independence and the Havel-Hakimi residue. *Discrete Mathematics*, 127 (1994) 209–212.

[97] GUTMAN, I. Total π-electron energy of benzenoid hydrocarbon. *Topics in Current Chemistry*, 162 (1992) 29–63.

[98] GUTMAN, I., and CYVIN, S. *Introduction to the Theory of Benzenoid Hydrocarbons*. Springer-Verlag, 1989.

[99] GUTMAN, I. and HOU, Y.P. Bipartite unicyclic graphs with greatest energy. *Match-Commun. Math. comp. Chem.* (43) (2001) 17–28.

[100] HÁJEK, P. and HAVRÁNEK, T. On generation of inductive hypotheses. *International Journal of Man-Machine Studies* 9 (1977) 415–438.

[101] HÁJEK, P. and HAVRÁNEK, T. *Mechanizing Hypothesis Formation. Mathematical Foundations for a General Theory*, Berlin: Springer, 1978.

[102] HÁJEK, P. and HOLEŇA, M. Formal logics of discovery and hypothesis formation by machine. *Theoretical Computer Science* 292 (2003) 345–357.

[103] HAKIMI, S.L. On realizability of a set of integers as degrees of the vertices of a linear graph. 1. *Journal of SIAM*, 10 (1962) 496–506.

[104] HANSEN, P. Computers in graph theory. *Graph Theory Notes of New York XLIII* (2002) 20–34.

[105] HANSEN, P. Degrés et nombre de stabilité d'un graphe. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 17 (1975) 213–220.

[106] HANSEN, P., and MÉLOT, H. Variable neighborhood for extremal graphs 6. Analysing bounds for the connectivity index. *Journal of Chemical Information and Chemical Sciences*, (2002).

[107] HANSEN, P., and MÉLOT, H. Variable neighborhood search for extremal graphs. 9. Bounding the irregularity of a graph, in S. Fajtlowicz *et al.* (eds.), *Graphs and Discovery* , American Mathematical Society, forthcoming.

[108] HANSEN, P., and MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. *European J. of Oper. Res.*, 130 (2001) 449–467.

[109] HANSEN, P., and ZHENG, M.L. Sharp bounds on the order, size, and stability number of graphs. *Networks*, 23 (1993) 99–102.

[110] HANSEN, P., and ZHENG, M.L. Upper bounds for the Clar number of a benzenoid hydrocarbon. *Faraday Transactions*, 88 (1992) 75–83.

[111] HANSEN, P., and ZHENG, M.L. The Clar number of a benzenoid hydrocarbon and linear programming. *Journal of Math. Chem.*, 15 (1994) 93–107.

[112] HANSEN, P., MLADENOVIC, N., and UROSEVIC, D. Variable neighborhood search for the maximum clique. *Les Cahier du GERAD*, G–2001–08, submitted.

[113] HANSEN, P., AOUCHICHE, M., CAPOROSSI, C., MÉLOT, H., and STEVANOVIĆ, D. What forms have interesting conjectures in graph theory? *Les Cahiers du GERAD*, G–2002–46, 2002, submitted.

[114] HARDY, G. *A Mathematician's Apology.* Cambridge: Cambridge University Press, 1992.

[115] HASTAD, J., JUST, B., LAGARIAS, T.C., SCHNORR, C.P. Polynomial time algorithms for finding integer relations among real numbers. *SIAM Journal on Computing* 18 (1989) 859–881.

[116] HAVEL, V., A remark on the existence of finite graphs. *Casopis Pest. Mat.* 80 (1955) 477–480.

[117] HOU, Y.P. Unicyclic graphs with minimum energy. *J. Mat. Chem.*, 29 (2001) 163–168.

[118] HSU, F.-H. *Behind Deep Blue*, Princeton: Princeton University Press, 2002.

[119] KNUTH, D. *The Stanford Graphbase: A Platform for Combinatorial Computing.* Addison-Wesley, Reading, Massachusetts, 1993.

[120] KURZWEIL, R. *The Age of Spiritual Machines.* London: Penguin, 2002.

[121] LANGLEY, P. The Computer-Aided Discovery of Scientific Knowledge. *Discovery Science: Proceedings of the First International Conference on Discovery Science. Lecture Notes in Artificial Intelligence*, 25–39, (1998).

[122] LANGLEY, P., SIMON, H.A, BRADSHAW, G.L., and ZYTKOW, J.M. *Scientific Discovery, Computational Explorations of the Creative Process.* Cambridge, Mass: MIT Press.

[123] LAKATOS, I. *Proofs and Refutations.* Cambridge, Mass: Cambridge University Press, 1976.

[124] LARSON, C. Intelligent machinery and mathematical discovery. *Graph Theory Notes of New York*, XLII (2002) 8–17.

[125] LARSON, C. On progress in the automation of mathematical conjecture-making. *preprint*, (2002).

[126] LEYDESDORFF, L. *The Challenge of Scientometrics: The Development of Measurement and Self-Organization of Scientific Communications.* Universal Publisher (2001).

[127] LI, M., AND VITANY, P. *An Introduction to Kolmogorov Complexity and its Applications.* New York: Springer, 1997.

[128] MAC LANE, S. Comment on "Theoretical Mathematics": Towards a cultural synthesis of mathematics and theoretical physics. *Bulletin of the American Mathematical Society*, 30 (1994) 13–15.

[129] MANGASARIAN, O.L. Arbitrary-norm separating plane. *Operations Research Letters*, 24 (1999) 15–23.

[130] MATHWORKS, Inc. Matlab: The Language of Technical Computing. The MathWorks, Inc.

[131] MC KAY, B.D. Nauty user's guide (version 1.5). Tech. Rep. TR-CS-90-02, Department of Computer Science, Australian National University, 1990.

[132] MCKAY, B.D. Isomorph-free exhaustive generation. *J. Algorithms*, 26 (1998) 306–324.

[133] MC CUNE, W. Solution of the Robbins problem. *J. Automated Reasoning*, 19 (1977) 263–276.

[134] MEHLHORN, K., and NÄHGER, S. LEDA: A platform for combinatorial and geometric computing. *Communications of the ACM*, 38(1) (1995) 96–102.

[135] MLADENOVIĆ, N., and HANSEN, P. Variable neighborhood search. *Computers and Operations Research*, 29 (1997) 1097–1100.

[136] OTTER. An Automated Deduction System. Web Site.

[137] PEPPER, R. On New Didactics of Mathematics-Learning Graph Theory via Graffiti. *Preprint*, (2002).

[138] POLYA, G. *Mathematics and Plausible Reasoning, Volume 1. (Induction and Analogy in Mathematics).* Princeton: Princeton University Press, 1954.

[139] POLYA, G. *Mathematics and Plausible Reasoning, Volume 2. (Patterns of Plausible Inference).* Princeton: Princeton University Press, 1954.

[140] POPPER, K. *The Logic of Scientific Discovery*. Hutchinson, London, 1959.

[141] PRICE, D. DE SOLLA, *Little Science, Big Science*. New York; Columbia University Press (1963).

[142] PUJOL, F. Étude d'un système automatisé en théorie des graphes (french). Travail de fin d'études IIE, sous la direction de Gilles Caporossi et Pierre Hansen. Rapport final. GERAD. 1999.

[143] QUINLAN, J.R. *C4.5. Programs for Machine Learning*. Morgan Kaufmann, 1993.

[144] RANDIĆ, M. On characterization of molecular branching. *Journal of the American Chemical Society*, 97 (1975) 6609–6615.

[145] RADZISZOWSKI, S.P. Small Ramsey numbers. Dynamic survey 1. *Electronic Journal of Combinatorics* (1994). Updated 1998.

[146] ROBERTSON, N., SANDERS, D., SEYMOUR, P., and THOMAS, R. The four-color theorem. *J. Combinatorial Theory, Ser. B*, 70 (1997) 2–44.

[147] ROGET'S II. The New Thesaurus@Bartleby.com

[148] ROWLINSON, P. A deletion–contraction algorithm for the characteristic polynomial of a multigraph. *Proceedings of the Royal Society of Edinburgh A*, 105 (1987) 153–160.

[149] SAATY, T., and KAINEN, P. *The Four-Color Problem: Assaults and Conquest*. New-York: Dover (1986).

[150] SAMUELSON, P.A. *Economics*. New York: Mc Graw Hill, 1968.

[151] SEYMOUR, P. Personal Communication at the Graph Coloring and Applications Workshop. CRM, Montreal, May 1998.

[152] SIBLEY, T., and WAGON, S. Rhombic Penrose tillings can be 3-colored. *American Mathematical Monthly*, (2000) 251–253.

[153] SIMIĆ, S. Some results on the largest eigenvalue of a graph. *Ars Combinatoria*, 24A (1987) 211–219.

[154] SIMIĆ, S., and KOCIĆ, V. On the largest eigenvalue of some homeomorphic graphs. *Publ. Inst. Math.* (Beograd) 40 (1986) 3–9.

[155] SHEN, W., and SIMON, H.A. Fitness Requirements for scientific theories containing recursive theoretical terms. *British Journal for the Philosophy of Science*, 44 (1993) 641-652.

[156] SKIENA, S. The Graphs of Graffiti:, *directory*, of a collection of 195 graphs from the database of Graffiti. The graphs have been converted to Combinatorica format. The database consists mostly of counterexamples, most of which were found by Noga Alon, Robert Beezer, Tony Brewster, Michael Dineen, Shui-Tain Chen, Paul Erdös, Siemion Fajtlowicz, Odile Favaron, Maryvonne Maheo, J. Riegsecker, Jean-Franois Sacle, Michael Saks, Paul Seymour, James Shearer, B.A. Smith, William Staton and Peter Winkler.

[157] SLOANE, N. The On-Line Encyclopedia of Integer Sequences, interactive Web Site.

[158] SMALE, S. Mathematical problems for the next century. *The Mathematical Intelligence* 20 (1998) 7–15.

[159] TURAN, P. An extremal problem in graph theory (in Hungarian) *Mat. Fiz. Lapok*, 48 (1941) 436–452.

[160] VAN NUFFELEN, C. A bound for the chromatic number of a graph. *American Mathematical Monthly*, 83 (1976) 265–266.

[161] WOLFRAM, Research Inc. *Mathematica Language and Software*. Wolfram Research, Inc.

[162] WOS, L. *The Automation of Reasoning: An Experimenter's Notebook with other Tutorial*. New-York, Academic Press (1996).

[163] WU, W.-T. On the decision problem and the mechanization of theorems proving in elementary geometry. *Scientia Simica* 21 (1978) 157–179.

[164] WU, W.-T. Basic principles of mechanical theorem proving in geometrics. *Journal of Systems Science and Mathematical Sciences* 4 (1984) 207–235, republished in *Journal of Automated Reasoning* 2 (1986) 221–252.

Experimental Mathematics: Recent Developments and Future Outlook

David H. Bailey[1] and Jonathan M. Borwein[2]

[1] Lawrence Berkeley Laboratory, Berkeley, CA 94720, USA, dhbailey@lbl.gov.***

[2] Gordon M. Shrum Professor of Science, Centre for Experimental and Constructive Mathematics, Simon Fraser University, Burnaby, BC, Canada, jborwein@cecm.sfu.ca.†

1 Introduction

While extensive usage of high-performance computing has been a staple of other scientific and engineering disciplines for some time, research mathematics is one discipline that has heretofore not yet benefited to the same degree. Now, however, with sophisticated mathematical computing tools and environments widely available on desktop computers, a growing number of remarkable new mathematical results are being discovered partly or entirely with the aid of these tools. With currently planned improvements in these tools, together with substantial increases expected in raw computing power, due both to Moore's Law and the expected implementation of these environments on parallel supercomputers, we can expect even more remarkable developments in the years ahead.

This article briefly discusses the nature of mathematical experiment. It then presents a few instances primarily of our own recent computer-aided mathematical discoveries, and sketches the outlook for the future. Additional examples in diverse fields and broader citations to the literature may be found in [16] and its references.

2 Preliminaries

The crucial role of high performance computing is now acknowledged throughout the physical, biological and engineering sciences. Numerical experimentation, using increasingly large-scale, three-dimensional simulation programs, is now a staple of fields such as aeronautical and electrical engineering, and research scientists heavily utilize computing technology to collect and analyze data, and to explore the implications of various physical theories.

*** Bailey's work supported by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC03-76SF00098.

† Borwein's work supported by the Natural Sciences and Engineering Research Council of Canada and the Networks of Centres of Excellence programme.

However, "pure" mathematics (and closely allied areas such as theoretical physics) only recently has begun to capitalize on this new technology. This is ironic, because the basic theoretical underpinnings of modern computer technology were set out decades ago by mathematicians such as Alan Turing and John Von Neumann. But only in the past decade, with the emergence of powerful mathematical computing tools and environments, together with the growing availability of very fast desktop computers and highly parallel supercomputers, as well as the pervasive presence of the Internet, has this technology reached the level where the research mathematician can enjoy the same degree of intelligent assistance that has graced other technical fields for some time.

This new approach is often termed *experimental mathematics*, namely the utilization of advanced computing technology to explore mathematical structures, test conjectures and suggest generalizations. And there is now a thriving journal of *Experimental Mathematics*. In one sense, there is nothing new in this approach — mathematicians have used it for centuries. Gauss once confessed, "I have the result, but I do not yet know how to get it." [2]. Hadamard declared, "The object of mathematical rigor is to sanction and legitimize the conquests of intuition, and there was never any other object for it." [34]. In recent times Milnor has stated this philosophy very clearly:

> If I can give an abstract proof of something, I'm reasonably happy. But if I can get a concrete, computational proof and actually produce numbers I'm much happier. I'm rather an addict of doing things on computer, because that gives you an explicit criterion of what's going on. I have a visual way of thinking, and I'm happy if I can see a picture of what I'm working with. [35]

What is really meant by an *experiment* in the context of mathematics? In *Advice to a Young Scientist*, Peter Medawar [31] identifies four forms of experiment:

> 1. The *Kantian* experiment is one such as generating "the classical non-Euclidean geometries (hyperbolic, elliptic) by replacing Euclid's axiom of parallels (or something equivalent to it) with alternative forms."
> 2. The *Baconian* experiment is a contrived as opposed to a natural happening, it "is the consequence of 'trying things out' or even of merely messing about."
> 3. The *Aristotelian* experiment is a demonstration: "apply electrodes to a frog's sciatic nerve, and lo, the leg kicks; always precede the presentation of the dog's dinner with the ringing of a bell, and lo, the bell alone will soon make the dog dribble."
> 4. The *Galilean* experiment is "a critical experiment – one that discriminates between possibilities and, in doing so, either gives us confidence in the view we are taking or makes us think it in need of correction."

The first three are certainly common in mathematics. However, as discussed in detail in [15], the Galilean experiment is the only one of the four forms which can make experimental mathematics a truly serious enterprise.

3 Tools of the Trade

The most obvious development in mathematical computing technology has been the growing availability of powerful symbolic computing tools. Back in the 1970s, when the first symbolic computing tools became available, their limitations were quite evident — in many cases, these programs were unable to handle operations that could be done by hand. In the intervening years these programs, notably the commercial products such as Maple and Mathematica, have greatly improved. While numerous deficiencies remain, they nonetheless routinely and correctly dispatch many operations that are well beyond the level that a human could perform with reasonable effort.

Another recent development that has been key to a number of new discoveries is the emergence of practical integer relation detection algorithms. Let $x = (x_1, x_2, \cdots, x_n)$ be a vector of real or complex numbers. x is said to possess an integer relation if there exist integers a_i, not all zero, such that $a_1x_1 + a_2x_2 + \cdots + a_nx_n = 0$. By an *integer relation algorithm*, we mean a practical computational scheme that can recover the vector of integers a_i, if it exists, or can produce bounds within which no integer relation exists. The problem of finding integer relations was studied by numerous mathematicians, including Euclid and Euler. The first general integer relation algorithm was discovered in 1977 by Ferguson and Forcade [24]. There is a close connection between integer relation detection and finding small vectors in an integer lattice, and thus one common solution to the integer relation problem is to apply the Lenstra-Lenstra-Lovasz (LLL) lattice reduction algorithm [30]. At the present time, the most effective scheme for integer relation detection is Ferguson's "PSLQ" algorithm [23,6].

Integer relation detection, as well as a number of other techniques used in modern experimental mathematics, relies heavily on very high precision arithmetic. The most advanced tools for performing high precision arithmetic utilize fast Fourier transforms (FFTs) for multiplication operations. Armed with one of these programs, a researcher can often effortlessly evaluate mathematical constants and functions to precision levels in the many thousands of decimal digits. The software products Maple and Mathematica include relatively complete and well-integrated multiple precision arithmetic facilities, although until very recently they did not utilize FFTs, or other accelerated multiplication techniques. One may also use any of several freeware multiprecision software packages [3,22] and for many purposes tools such as Matlab, MuPAD or more specialized packages like Pari-GP are excellent.

High precision arithmetic, when intelligently used with integer relation detection programs, allows researchers to discover heretofore unknown mathematical identities. It should be emphasized that these numerically discovered "identities" are only approximately established. Nevertheless, in the cases we are aware of, the results have been numerically verified to hundreds and in some cases thousands of decimal digits beyond levels that could reasonably be dismissed as numerical artifacts. Thus while these "identities" are not firmly established in a formal sense, they are supported by very compelling numerical evidence. After all, which is more compelling, a formal proof that in its full exposition requires hundreds of difficult pages of reasoning, fully understood by only two or three colleagues, or the numerical verification of a conjecture to 100,000 decimal digit accuracy, subsequently validated by numerous subsidiary computations? In the same way, these tools are often even more useful as a way of *excluding* the possibility of hoped for relationships, as in equation (1) below.

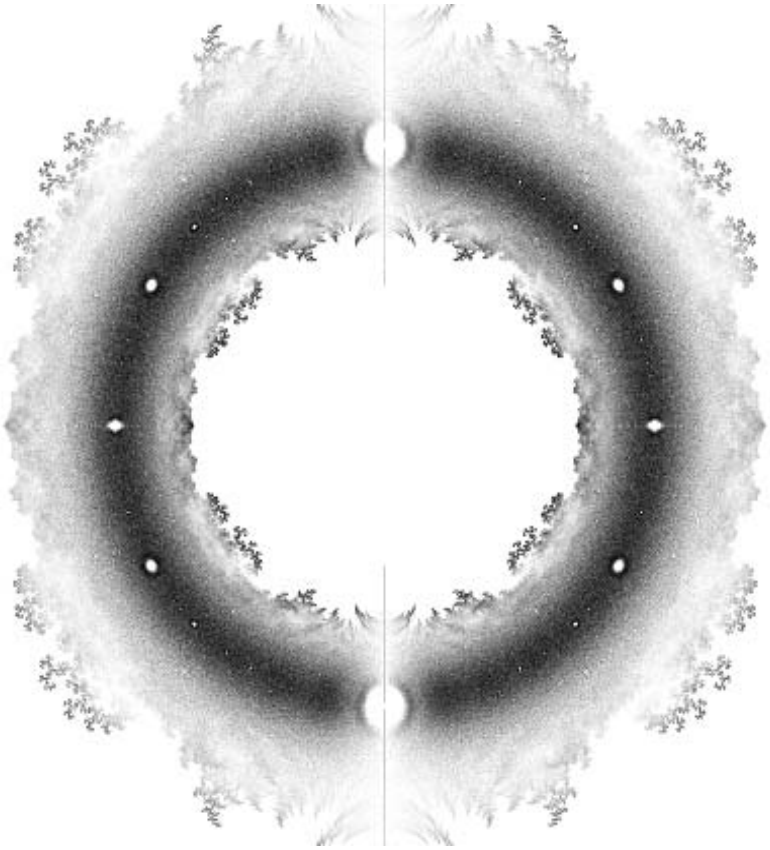


FIGURE 1(A-D): -1/1 POLYNOMIALS (TO BE SET IN COLOR)

We would be remiss not to mention the growing power of visualization especially when married to high performance computation. The pictures

in FIGURE 1 represents the zeroes of all polynomials with ± 1 coefficients of degree at most 18. One of the most striking features of the picture, its fractal nature excepted, is the appearance of different sized "holes" at what transpire to be roots of unity. This observation which would be very hard to make other than pictorially led to a detailed and rigorous analysis of the phenomenon and more [17,27]. They were lead to this analysis by the interface which was built for Andrew Odlyzko's seminal online paper [32].

One additional tool that has been utilized in a growing number of studies is Sloane and Plouffe's *Encyclopedia of Integer Sequences* [36]. As the title indicates, it identifies many integer sequences based on the first few terms. A very powerful on-line version is also available and is a fine example of the changing research paradigm. Another wonderful resource is Stephen Finch's "Favorite Mathematical Constants," which contains a wealth of frequently updated information, links and references on 125 constants, [25], such as the *hard hexagon constant* $\kappa \approx 1.395485972$ for which Zimmermann obtained a minimal polynomial of degree 24 in 1996.[1]

In the following, we illustrate this – both new and old – approach to mathematical research using a handful of examples with which we are personally familiar. We will then sketch some future directions in this emerging methodology. We have focussed on the research of our own circle of direct collaborators. We do so for resaons of fmailiarity and because we believe it is representative of broad changes in the way mathematics is being done rather than to claim primacy for our own skills or expertise.

4 A New Formula for Pi

Through the centuries mathematicians have assumed that there is no shortcut to determining just the n-th digit of π. Thus it came as no small surprise when such a scheme was recently discovered [5]. In particular, this simple algorithm allows one to calculate the n-th hexadecimal (or binary) digit of π without computing any of the first $n-1$ digits, without the need for multiple-precision arithmetic software, and requiring only a very small amount of memory. The one millionth hex digit of π can be computed in this manner on a current-generation personal computer in only about 30 seconds run time.

This scheme is based on the following remarkable formula, whose formal proof involves nothing more sophisticated than freshman calculus:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left[\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right]$$

This formula was found using months of PSLQ computations, after corresponding but simpler n-th digit formulas were identified for several

[1] See `http://www.mathsoft.com/asolve/constant/square/square.html`.

other constants, including log(2). This is likely the first instance in history that a significant new formula for π was discovered by a computer.

Similar base-2 formulas are given in [5,21] for a number of other mathematical constants. In [20] some base-3 formulas were obtained, including the identity

$$\pi^2 = \frac{2}{27}\sum_{k=0}^{\infty}\frac{1}{729^k}\quad\left[\frac{243}{(12k+1)^2} - \frac{405}{(12k+2)^2} - \frac{81}{(12k+4)^2}\right.$$
$$- \frac{27}{(12k+5)^2} - \frac{72}{(12k+6)^2} - \frac{9}{(12k+7)^2}$$
$$\left. - \frac{9}{(12k+8)^2} - \frac{5}{(12k+10)^2} + \frac{1}{(12k+11)^2}\right]$$

In [8], it is shown that the question of whether π, log(2) and certain other constants are normal can be reduced to a plausible conjecture regarding dynamical iterations of the form $x_0 = 0$,

$$x_n = (bx_{n-1} + r_n) \bmod 1$$

where b is an integer and $r_n = p(n)/q(n)$ is the ratio of two nonzero polynomials with $\deg(p) < \deg(q)$. The conjecture is that these iterates either have a finite set of attractors or else are equidistributed in the unit interval. In particular, it is shown that the question of whether π is normal base 16 (and hence base 2) can be reduced to the assertion that the dynamical iteration $x_0 = 0$,

$$x_n = \left(16x_{n-1} + \frac{120n^2 - 89n + 16}{512n^4 - 1024n^3 + 712n^2 - 206n + 21}\right) \bmod 1$$

is equidistributed in $[0, 1)$. There are also connections between the question of normality for certain constants and the theory of linear congruential pseudorandom number generators. All of these results derive from the discovery of the individual digit-calculating formulas mentioned above. For details, see [8].

5 Identities for the Riemann Zeta Function

Another application of computer technology in mathematics is to determine whether or not a given constant α, whose value can be computed to high precision, is algebraic of some degree n or less. This can be done by first computing the vector $x = (1, \alpha, \alpha^2, \cdots, \alpha^n)$ to high precision and then applying an integer relation algorithm. If a relation is found for x, then this relation vector is precisely the set of integer coefficients of a polynomial satisfied by α. Even if no relation is found, integer relation detection programs can produce bounds within which no relation can exist. In fact, exclusions of this type are solidly established by integer relation calculations, whereas "identities" discovered in this fashion are only approximately established, as noted above.

Consider, for example, the following identities, with that for $\zeta(3)$ due to Apéry [10,14]:

$$\zeta(2) = 3\sum_{k=1}^{\infty} \frac{1}{k^2\binom{2k}{k}}$$
$$\zeta(3) = \frac{5}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^3\binom{2k}{k}}$$
$$\zeta(4) = \frac{36}{17}\sum_{k=1}^{\infty} \frac{1}{k^4\binom{2k}{k}}$$

where $\zeta(n) = \sum_k k^{-n}$ is the Riemann zeta function at n. These results have led many to hope that

$$Z_5 = \zeta(5)/\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^5\binom{2k}{k}} \tag{1}$$

might also be a simple rational or algebraic number. However, computations using PSLQ established, for instance, that if Z_5 satisfies a polynomial of degree 25 or less, then the Euclidean norm of the coefficients must exceed 2×10^{37}. Given these results, there is no "easy" identity, and researchers are licensed to investigate the possibility of multi-term identities for $\zeta(5)$. One recently discovered [14], using a PSLQ computation, was the polylogarithmic identity

$$\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^5\binom{2k}{k}} = 2\zeta(5) + 80\sum_{k=1}^{\infty}\left[\frac{1}{(2k)^5} - \frac{L}{(2k)^4}\right]\rho^{2k} - \frac{4}{3}L^5 + \frac{8}{3}L^3\zeta(2) + 4L^2\zeta(3)$$

where $L = \log(\rho)$ and $\rho = (\sqrt{5}-1)/2$. This illuastrates neatly that one can only find a closed form if one knows where to look.

Other earlier evaluations involving the central binomial coefficient suggested general formulas [12], which were pursued by a combination of PSLQ and heavy-duty symbolic manipulation. This led, most unexpectedly, to the identity

$$\sum_{k=1}^{\infty} \zeta(4k+3)z^{4k} = \sum_{k=1}^{\infty} \frac{1}{k^3(1-z^4/k^4)} = \frac{5}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^3\binom{2k}{k}(1-z^4/k^4)} \prod_{m=1}^{k-1} \frac{1+4z^4/m^4}{1-z^4/m^4}.$$

Experimental analysis of the first ten terms showed that the rightmost above series necessarily had the form

$$\frac{5}{2}\sum_{k=1}^{\infty} \frac{(-1)^{k-1}P_k(z)}{k^3\binom{2k}{k}(1-z^4/k^4)}$$

where

$$P_k(z) = \prod_{j=1}^{k-1} \frac{1 + 4z^4/j^4}{1 - z^4/j^4}.$$

Also discovered in this process was the intriguing *equivalent* combinatorial identity

$$\binom{2n}{n} = \sum_{k=1}^{\infty} \frac{2n^2 \prod_{i=1}^{n-1}(4k^4 + i^4)}{k^2 \prod_{i=1, i\neq k}^{n}(k^4 - i^4)}.$$

This evaluation was discovered as the result of an serendipitous error in an input to Maple[2]— the computational equivalent of discovering penicillin after a mistake in a Petri dish.

With the recent proof of this last conjectured identity, by Almkvist and Granville [1], the above identities have now been rigorously established. But other numerically discovered "identities" of this type appear well beyond the reach of current formal proof methods. For example, in 1999 British physicist David Broadhurst used a PSLQ program to recover an explicit expression for $\zeta(20)$ involving 118 terms. The problem required 5,000 digit arithmetic and over six hours computer run time. The complete solution is given in [6].

6 Identification of Multiple Sum Constants

Numerous identities were experimentally discovered in some recent research on multiple sum constants. After computing high-precision numerical values of these constants, a PSLQ program was used to determine if a given constant satisfied an identity of a conjectured form. These efforts produced empirical evaluations and suggested general results [4]. Later, elegant proofs were found for many of these specific and general results [13], using a combination of human intuition and computer-aided symbolic manipulation. Three examples of experimentally discovered re-

[2] Typing 'infty' for 'infinity' revealed that the program had an algorithm when a formal variable was entered.

sults that were subsequently proven are:

$$\sum_{k=1}^{\infty}\left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^2 (k+1)^{-4} = \frac{37}{22680}\pi^6 - \zeta^2(3)$$

$$\sum_{k=1}^{\infty}\left(1+\frac{1}{2}+\cdots+\frac{1}{k}\right)^3 (k+1)^{-6} = \zeta^3(3) + \frac{197}{24}\zeta(9) + \frac{1}{2}\pi^2\zeta(7) - \frac{11}{120}\pi^4\zeta(5) - \frac{37}{7560}\pi^6\zeta(3)$$

$$\sum_{k=1}^{\infty}\left(1-\frac{1}{2}+\cdots+(-1)^{k+1}\frac{1}{k}\right)^2 (k+1)^{-3} = 4\,\mathrm{Li}_5(\tfrac{1}{2}) - \frac{1}{30}\ln^5(2) - \frac{17}{32}\zeta(5) - \frac{11}{720}\pi^4\ln(2) + \frac{7}{4}\zeta(3)\ln^2(2) + \frac{1}{18}\pi^2\ln^3(2) - \frac{1}{8}\pi^2\zeta(3)$$

where again $\zeta(n) = \sum_{j=1}^{\infty} j^{-n}$ is a value of the Riemann zeta function, and $\mathrm{Li}_n(x) = \sum_{j=1}^{\infty} x^j j^{-n}$ denotes the classical polylogarithm function.

More generally, one may define *multi-dimensional Euler sums* (or *multiple zeta values*) by

$$\zeta\begin{pmatrix} s_1,\ s_2\ \cdots\ s_r \\ \sigma_1,\ \sigma_2\ \cdots\ \sigma_r \end{pmatrix} := \sum_{k_1>k_2>\cdots>k_r>0} \frac{\sigma_1^{k_1}}{k_1^{s_1}}\,\frac{\sigma_2^{k_2}}{k_2^{s_2}}\,\cdots\,\frac{\sigma_r^{k_r}}{k_r^{s_r}}$$

where $\sigma_j = \pm 1$ are signs and $s_j > 0$ are integers. When all the signs are positive, one has a multiple zeta value. The integer r is the sum's depth and $s_1 + s_2 + \cdots + s_r$ is the weight. These sums have connections with diverse fields such as knot theory, quantum field theory and combinatorics. Constants of this form with alternating signs appear in problems such as computation of the magnetic moment of the electron.

Multi-dimensional Euler sums satisfy many striking identities. The discovery of the more recondite identities was facilitated by the development of Hölder convolution algorithms that permit very high precision numerical values to be rapidly computed. See [13] and a computational interface at `www.cecm.sfu.ca/projects/ezface+/`. One beautiful general identity discovered by Zagier [37] in the course of similar research is

$$\zeta(3,1,3,1,\ldots,3,1) = \frac{1}{2n+1}\zeta(2,2,\ldots,2) \;=\; \frac{2\pi^{4n}}{(4n+2)!}$$

where there are n instances of '$(3,1)$' and '2' in the arguments to $\zeta(\cdot)$. This has now been proven in [13] and the proof, while entirely conventional, was obtained by guided experimentation. A related conjecture for which overwhelming evidence but no hint of a proof exists is the

"identity"

$$8^n \zeta \begin{pmatrix} 2, & 1, & 2, & 1, \cdots, & 2, & 1 \\ -1, & 1, & -1, & 1, \cdots, & -1, & 1 \end{pmatrix} = \zeta(2,1,2,1,\ldots,2,1).$$

Along this line, Broadhurst conjectured, based on low-degree numerical results, that the dimension of the space of Euler sums with weight w is the Fibonacci number $F_{w+1} = F_w + F_{w-1}$, with $F_1 = F_2 = 1$. In testing this conjecture, complete reductions of all Euler sums to a basis of size F_{w+1} were obtained with PSLQ at weights $w \leq 9$. At weights $w = 10$ and $w = 11$ the conjecture was stringently tested by application of PSLQ in more than 600 cases. At weight $w = 11$ such tests involve solving integer relations of size $n = F_{12} + 1 = 145$. In a typical case, each of the 145 constants was computed to more than 5,000 digit accuracy, and a working precision level of 5,000 digits was employed in an advanced "multi-pair" PSLQ program. In these problems the ratios of adjacent coefficients in the recovered integer vector usually have special values, such as $11! = 39916800$. These facts, combined with confidence ratios typically on the order of 10^{-300} in the detected relations, render remote the chance that these identities are spurious numerical artifacts, and lend substantial support to this conjecture [6].

7 Mathematical Computing Meets Parallel Computing

The potential future power of highly parallel computing technology has been underscored in some recent results. Not surprisingly, many of these computations involve the constant π, underscoring the enduring interest in this most famous of mathematical constants. In 1997 Fabrice Bellard of INRIA used a more efficient formula, similar to the one mentioned in section three, programmed on a network of workstations, to compute 150 binary digits of π starting at the *trillionth* position. Not to be outdone, 17-year-old Colin Percival of Simon Fraser University in Canada organized a computation of 80 binary digits of π beginning at the five trillionth position, using a network of 25 laboratory computers. He an many others are presently computing binary digits at the quadrillionth position on the web [33]. As we write, the most recent computational result was Yasumasa Kanada's calculation (September 1999) of the first 206 billion decimal digits of π. This spectacular computation was made on a Hitachi parallel supercomputer with 128 processors, in little over a day, and employed the Salamin-Brent algorithm [10], with a quartically convergent algorithm from [10] as an independent check.

Several large-scale parallel integer relation detection computations have also been performed in the past year or two. One arose from the discovery by Broadhurst that

$$\alpha^{630} - 1 = \frac{(\alpha^{315}-1)(\alpha^{210}-1)(\alpha^{126}-1)^2(\alpha^{90}-1)(\alpha^3-1)^3(\alpha^2-1)^5(\alpha-1)^3}{(\alpha^{35}-1)(\alpha^{15}-1)^2(\alpha^{14}-1)^2(\alpha^5-1)^6\alpha^{68}}$$

where $\alpha = 1.176280818\ldots$ is the largest real root of Lehmer's polynomial [29]

$$0 = 1 + \alpha - \alpha^3 - \alpha^4 - \alpha^5 - \alpha^6 - \alpha^7 + \alpha^9 + \alpha^{10}.$$

The above cyclotomic relation was first discovered by a PSLQ computation, and only subsequently proven. Broadhurst then conjectured that there might be integers a, b_j, c_k such that

$$a\,\zeta(17) = \sum_{j=0}^{8} b_j\,\pi^{2j}(\log\alpha)^{17-2j} + \sum_{k\in D(\mathcal{S})} c_k\,\mathrm{Li}_{17}(\alpha^{-k})$$

where the 115 indices k are drawn from the set, $D(\mathcal{S})$, of positive integers that divide at least one element of

$$\begin{aligned}\mathcal{S} = \{&29, 47, 50, 52, 56, 57, 64, 74, 75, 76, 78, 84, 86, 92, 96, 98, 108, 110, 118,\\ &124, 130, 132, 138, 144, 154, 160, 165, 175, 182, 186, 195, 204, 212, 240,\\ &246, 270, 286, 360, 630\}.\end{aligned}$$

Indeed, such a relation was found, using a parallel multi-pair PSLQ program running on a SGI/Cray T3E computer system at Lawrence Berkeley Laboratory. The run employed 50,000 decimal digit arithmetic and required approximately 44 hours on 32 processors. The resulting integer coefficients are as large as 10^{292}, but the "identity" nonetheless was confirmed to 13,000 digits beyond the level of numerical artifact [7].

8 Connections to Quantum Field Theory

In another surprising recent development, David Broadhurst has found, using these methods, that there is an intimate connection between Euler sums and constants resulting from evaluation of Feynman diagrams in quantum field theory [18,19]. In particular, the renormalization procedure (which removes infinities from the perturbation expansion) involves multiple zeta values. As before, a fruitful theory has emerged, including a large number of both specific and general results [13].

Some recent quantum field theory results are even more remarkable. Broadhurst has now shown [20], using PSLQ computations, that in each of ten cases with unit or zero mass, the finite part the scalar 3-loop tetrahedral vacuum Feynman diagram reduces to 4-letter "words" that represent iterated integrals in an alphabet of seven "letters" comprising the one-forms $\Omega := dx/x$ and $\omega_k := dx/(\lambda^{-k} - x)$, where $\lambda := (1 + \sqrt{-3})/2$ is the primitive sixth root of unity, and k runs from 0 to 5. A 4-letter word is a 4-dimensional iterated integral, such as

$$U := \zeta(\Omega^2\omega_3\omega_0) = \int_0^1 \frac{dx_1}{x_1}\int_0^{x_1}\frac{dx_2}{x_2}\int_0^{x_2}\frac{dx_3}{(-1-x_3)}\int_0^{x_3}\frac{dx_4}{(1-x_4)} = \sum_{j>k>0}\frac{(-1)^{j+k}}{j^3k}.$$

There are 7^4 such four-letter words. Only two of these are primitive terms occurring in the 3-loop Feynman diagrams: U, above, and

$$V := \text{Real}[\zeta(\Omega^2\omega_3\omega_1)] = \sum_{j>k>0} \frac{(-1)^j \cos(2\pi k/3)}{j^3 k}.$$

The remaining terms in the diagrams reduce to products of constants found in Feynman diagrams with fewer loops. These ten cases as shown in Figure 1. In these diagrams, dots indicate particles with nonzero rest mass. The formulas that have been found, using PSLQ, for the corresponding constants are given in Table 2. In the table the constant $C = \sum_{k>0} \sin(\pi k/3)/k^2$.

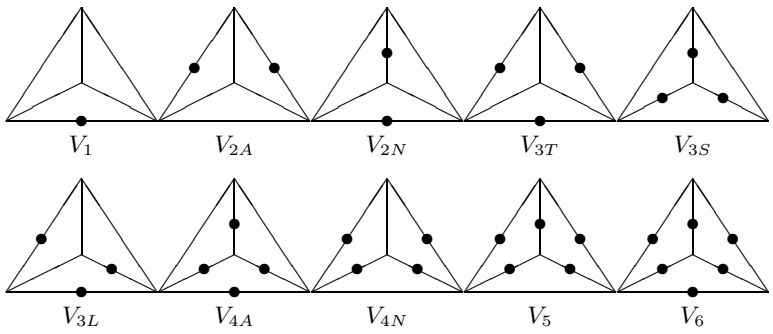


Fig. 1. The ten tetrahedral cases

V_1	$= 6\zeta(3) + 3\zeta(4)$
V_{2A}	$= 6\zeta(3) - 5\zeta(4)$
V_{2N}	$= 6\zeta(3) - \frac{13}{2}\zeta(4) - 8U$
V_{3T}	$= 6\zeta(3) - 9\zeta(4)$
V_{3S}	$= 6\zeta(3) - \frac{11}{2}\zeta(4) - 4C^2$
V_{3L}	$= 6\zeta(3) - \frac{15}{4}\zeta(4) - 6C^2$
V_{4A}	$= 6\zeta(3) - \frac{77}{12}\zeta(4) - 6C^2$
V_{4N}	$= 6\zeta(3) - 14\zeta(4) - 16U$
V_5	$= 6\zeta(3) - \frac{469}{27}\zeta(4) + \frac{8}{3}C^2 - 16V$
V_6	$= 6\zeta(3) - 13\zeta(4) - 8U - 4C^2$

Table 1. Formulas found by PSLQ for the ten cases of Figure 1

9 A Note of Caution

In spite of the remarkable successes of this methodology, some caution is in order. First of all, the fact that an identity is established to high precision is *not* a guarantee that it is indeed true. One example is

$$\sum_{n=1}^{\infty} \frac{[n \tanh \pi]}{10^n} \approx \frac{1}{81}$$

which holds to 267 digits, yet is not an exact identity, failing in the 268'th place. Several other such bogus "identities" are exhibited and explained in [11].

More generally speaking, caution must be exercised when extrapolating results true for small n to all n. For example,

$$\int_0^{\infty} \frac{\sin(x)}{x}\, dx = \frac{\pi}{2}$$
$$\int_0^{\infty} \frac{\sin(x)}{x} \frac{\sin(x/3)}{x/3}\, dx = \frac{\pi}{2}$$
$$\cdots$$
$$\int_0^{\infty} \frac{\sin(x)}{x} \frac{\sin(x/3)}{x/3} \cdots \frac{\sin(x/13)}{x/13}\, dx = \frac{\pi}{2}$$

yet

$$\int_0^{\infty} \frac{\sin(x)}{x} \frac{\sin(x/3)}{x/3} \cdots \frac{\sin(x/15)}{x/15}\, dx = \frac{467807924713440738696537864469}{935615849440640907310521750000}\pi.$$

When this fact was recently observed by a researcher using a mathematical software package, he concluded that there must be a "bug" in the software. Not so. What is happening here is that

$$\int_0^{\infty} \frac{\sin(x)}{x} \frac{\sin(x/h_1)}{x/h_1} \cdots \frac{\sin(x/h_n)}{x/h_n}\, dx = \frac{\pi}{2}$$

only so long as $1/h_1 + 1/h_2 + \cdots + 1/h_n < 1$. In the above example, $1/3 + 1/5 + \cdots + 1/13 < 1$, but with the addition of $1/15$, the sum exceeds 1 and the identity no longer holds [9]. Changing the h_n lets this pattern persist indefinitely but still fail in the large.

10 Future Outlook

Computer mathematics software is now becoming a staple of university departments and government research laboratories. Many university departments now offer courses where the usage of one of these software

packages is an integral part of the course. But further expansion of these facilities into high schools has been inhibited by a number of factors, including the fairly high cost of such software, the lack of appropriate computer equipment, difficulties in standardizing such coursework at a regional or national level, a paucity of good texts incorporating such tools into a realistic curriculum, lack of trained teachers and many other demands on their time.

But computer hardware continues its downward spiral in cost and its upward spiral in power. It thus appears that within a very few years, moderately powerful symbolic computation facilities can be incorporated into relatively inexpensive hand calculators, at which point it will be much easier to successfully integrate these tools into high school curricula. Thus it seems that we are poised to see a new generation of students coming into university mathematics and science programs who are completely comfortable using such tools. This development is bound to have a profound impact on the future teaching, learning and doing of mathematics.

A likely and fortunate spin-off of this development is that the commercial software vendors who produce these products will likely enjoy a broader financial base, from which they can afford to further enhance their products geared at serious researchers. Future enhancements are likely to include more efficient algorithms, more extensive capabilities mixing numerics and symbolics, more advanced visualization facilities, and software optimized for emerging symmetric multiprocessor and highly parallel, distributed memory computer systems. When combined with expected increases in raw computing power due to Moore's Law — improvements which almost certainly will continue unabated for at least ten years and probably much longer — we conclude that enormously more powerful computer mathematics systems will be available in the future.

We only now are beginning to experience and comprehend the potential impact of computer mathematics tools on mathematical research. In ten more years, a new generation of computer-literate mathematicians, armed with significantly improved software on prodigiously powerful computer systems, are bound to make discoveries in mathematics that we can only dream of at the present time. Will computer mathematics eventually replace, in near entirety, the solely human form of research, typified by Andrew Wiles' recent proof of Fermat's Last Theorem? Will computer mathematics systems eventually achieve such intelligence that they discover deep new mathematical results, largely or entirely without human assistance? Will new computer-based mathematical discovery techniques enable mathematicians to explore the realm, proved to exist by Gödel, Chaitin and others, that is fundamentally beyond the limits of formal reasoning?

11 Conclusion

We have shown a small but we hope convincing selection of what the present allows and what the future holds in store. We have hardly mentioned the growing ubiquity of web based computation, or of pervasive access to massive data bases, both public domain and commercial. Neither have we raised the human/computer interface or intellectual property issues and the myriad other not-purely-technical issues these raise.

Whatever the outcome of these developments, we are still persuaded that mathematics is and will remain a uniquely human undertaking. One could even argue that these developments confirm the fundamentally human nature of mathematics. Indeed, Reuben Hersh's arguments [26] for a humanist philosophy of mathematics, as paraphrased below, become more convincing in our setting:

1. *Mathematics is human.* It is part of and fits into human culture. It does not match Frege's concept of an abstract, timeless, tenseless, objective reality.
2. *Mathematical knowledge is fallible.* As in science, mathematics can advance by making mistakes and then correcting or even re-correcting them. The "fallibilism" of mathematics is brilliantly argued in Lakatos' *Proofs and Refutations* [28].
3. *There are different versions of proof or rigor.* Standards of rigor can vary depending on time, place, and other things. The use of computers in formal proofs, exemplified by the computer-assisted proof of the four color theorem in 1977, is just one example of an emerging nontraditional standard of rigor.
4. *Empirical evidence, numerical experimentation and probabilistic proof all can help us decide what to believe in mathematics.* Aristotelian logic isn't necessarily always the best way of deciding.
5. *Mathematical objects are a special variety of a social-cultural-historical object.* Contrary to the assertions of certain post-modern detractors, mathematics cannot be dismissed as merely a new form of literature or religion. Nevertheless, many mathematical objects can be seen as shared ideas, like Moby Dick in literature, or the Immaculate Conception in religion.

Certainly the recognition that "quasi-intuitive" analogies can be used to gain insight in mathematics can assist in the learning of mathematics. And honest mathematicians will acknowledge their role in discovery as well.

We look forward to what the future will bring.

References

1. G. Almkvist and A. Granville, "Borwein and Bradley's Apéry-like formulae for $\zeta(4n+3)$", *Experimental Mathematics* **8** (1999), 197–204.

2. Issac Asimov and J. A. Shulman, ed., *Isaac Asimov's Book of Science and Nature Quotations*, Weidenfield and Nicolson, New York, 1988, pg. 115.
3. David H. Bailey, "A Fortran-90 Based Multiprecision System", *ACM Transactions on Mathematical Software*, **21** (1995), pg. 379-387. Available from `http://www.nersc.gov/~dhbailey`.
4. David H. Bailey, Jonathan M. Borwein and Roland Girgensohn, "Experimental Evaluation of Euler Sums", *Experimental Mathematics*, **4** (1994), 17–30.
5. David H. Bailey, Peter B. Borwein and Simon Plouffe, "On The Rapid Computation of Various Polylogarithmic Constants", *Mathematics of Computation*, **66**,(1997, 903–913.
6. David H. Bailey and David Broadhurst, "Parallel Integer Relation Detection: Techniques and Applications". Available from `http://www.nersc.gov/~dhbailey`.
7. David H. Bailey and David Broadhurst, "A Seventeenth-Order Polylogarithm Ladder". Available from `http://www.nersc.gov/~dhbailey`.
8. David H. Bailey and Richard E. Crandall, "On the Random Character of Fundamental Constant Expansions", manuscript (2000). Available from `http://www.nersc.gov/~dhbailey`.
9. David Borwein and Jonathan M. Borwein, "Some Remarkable Properties of Sinc and Related Integrals", CECM Preprint 99:142, available from `http://www.cecm.sfu.ca/preprints`.
10. Jonathan M. Borwein and Peter B. Borwein, *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*, John Wiley and Sons, New York, 1987.
11. J. M. Borwein and P. B. Borwein, "Strange Series and High Precision Fraud", *American Mathematical Monthly*, **99** (1992), 622–640.
12. J.M. Borwein and D.M. Bradley, "Empirically determined Apéry–like formulae for zeta(4n+3)," *Experimental Mathematics*, **6** (1997), 181–194.
13. Jonathan M. Borwein, David M. Bradley, David J. Broadhurst and Peter Lisonek, "Special Values of Multidimensional Polylogarithms", *Trans. Amer. Math. Soc.*, in press. CECM Preprint 98:106, available from `http://www.cecm.sfu.ca/preprints`.
14. Jonathan M. Borwein, David J. Broadhurst and Joel Kamnitzer, "Central binomial sums and multiple Clausen values," preprint, November 1999. CECM Preprint 99:137, , available from `http://www.cecm.sfu.ca/preprints`.
15. J.M. Borwein, P.B. Borwein, R. Girgensohn and S. Parnes, "Making Sense of Experimental Mathematics," *Mathematical Intelligencer*, **18**, Number 4 (Fall 1996), 12–18.
16. Jonathan M. Borwein and Robert Corless, "Emerging tools for experimental mathematics," *MAA Monthly*, **106**(1999), 889–909. CECM Preprint 98:110, , available from `http://www.cecm.sfu.ca/preprints`.
17. Peter. B. Borwein and Christopher Pinner, "Polynomials with $\{0, +1, -1\}$ Coefficients and Root Close to a Given Point," *Canadian J. Mathematics* **49** (1998), 887–915.

18. David J. Broadhurst, John A. Gracey and Dirk Kreimer, "Beyond the Triangle and Uniqueness Relations: Non-zeta Counterterms at Large N from Positive Knots", *Zeitschrift für Physik*, **C75** (1997), 559–574.
19. David J. Broadhurst and Dirk Kreimer, "Association of Multiple Zeta Values with Positive Knots via Feynman Diagrams up to 9 Loops", *Physics Letters*, **B383** (1997), 403–412.
20. David J. Broadhurst, "Massive 3-loop Feynman Diagrams Reducible to SC* Primitives of Algebras of the Sixth Root of Unity", preprint, March 1998, to appear in *European Physical Journal C*. Available from `http://xxx.lanl.gov/abs/hep-th/9803091` .
21. David J. Broadhurst, "Polylogarithmic Ladders, Hypergeometric Series and the Ten Millionth Digits of $\zeta(3)$ and $\zeta(5)$", preprint, March 1998. Available from `http://xxx.lanl.gov/abs/math/9803067`.
22. Sid Chatterjee and Herman Harjono, "MPFUN++: A Multiple Precision Floating Point Computation Package in C++", University of North Carolina, Sept. 1998. Available from `http://www.cs.unc.edu/Research/HARPOON/mpfun++`.
23. Helaman R. P. Ferguson, David H. Bailey and Stephen Arno, "Analysis of PSLQ, An Integer Relation Finding Algorithm", *Mathematics of Computation*, **68** (1999), 351–369.
24. Helaman R. P. Ferguson and Rodney W. Forcade, "Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher Than Two", *Bulletin of the American Mathematical Society*, **1** (1979), 912–914.
25. Stphen Finch, "Favorite Mathematical Constants", `http://www.mathsoft.com/asolve/constant/constant.html`.
26. Reuben Hersh, "Fresh Breezes in the Philosophy of Mathematics", the *American Mathematical Monthly*, August-September 1995, 589–594.
27. Loki Jörgenson, "Zeros of Polynomials with Constrained Roots", `http//www.cecm.sfu.ca/personal/loki/Projects/Roots/Book`.
28. Imre Lakatos, *Proofs and Refutations: The Logic of Mathematical Discovery*, Cambridge University Press, 1977.
29. Derrick H. Lehmer, "Factorization of Certain Cyclotomic Functions", *Annals of Mathematics*, **34** (1933), 461–479.
30. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz, "Factoring Polynomials with Rational Coefficients", *Mathematische Annalen*, **261** (1982), 515-534.
31. P. B. Medawar, *Advice to a young Scientist*, Harper Colophon, New York, 1981.
32. Andrew Odlyzko, "Zeros of polynomials with 0,1 coefficients", `http://www.cecm.sfu.ca/organics/authors/odlyzko/` and `/organics/papers/odlyzko/support/polyform.html`.
33. Colin Percival, "Pihex: A Distributed Effort To Calculate Pi", `http://www.cecm.sfu.ca/projects/pihex/`.
34. George Polya, *Mathematical Discovery: On Understanding, Learning, and Teaching Problem Solving*, Combined Edition, New York, Wiley and Sons, 1981, pg. 129.

35. Ed Regis, *Who Got Einstein's Office?*, Addison-Wesley, 1986, pg. 78.
36. N.J.A. Sloane and Simon Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, 1995. The on-line version can be accessed at `http://www.research.att.com/~njas/sequences/Seis.html`.
37. Don Zagier, *Values of zeta functions and their applications*, First European Congress of Mathematics, Volume II, Birkhäuser, Boston, 1994, 497–512.

Evaluating the "Small Scope Hypothesis" for Code

Darko Marinov Alexandr Andoni Dumitru Daniliuc Sarfraz Khurshid
MIT Laboratory for Computer Science
200 Technology Square
Cambridge, MA 02139
{marinov,andoni,dumi,khurshid}@lcs.mit.edu

ABSTRACT

The "small scope hypothesis" argues that a high proportion of bugs in a system can be found by exhaustively checking the system within some small scope. In software testing, this exhaustive checking corresponds to testing the program for all inputs in a given scope. In object-oriented programs, an input is constructed from objects of different classes; a test input is within a scope s if at most s objects of any given class appear in it.

This paper evaluates the hypothesis for several implementations of data structures, including some from the Java Collections Framework. We measure how statement coverage, branch coverage, and rate of mutant killing vary with scope. For systematic input generation and correctness checking, we use the Korat tool. This paper presents Korat extensions that enable faster input generation and correctness checking. This paper also presents the Ferastrau tool that we have developed for mutation testing of Java programs. Experimental results show that exhaustive testing within small scopes can achieve complete coverage and kill most of the mutants, even for intricate methods that manipulate complex data structures. The results also show that Korat can efficiently generate inputs and check correctness for these scopes.

1. INTRODUCTION

The "small scope hypothesis" [16] argues that a high proportion of bugs in a system can be found by exhaustively checking the system within some small scope. This hypothesis is a well-known underlying principle of model checking [11]. For example, several case studies [18, 19] used the Alloy modeling language [15] to build *abstract models* of systems and check them with the Alloy Analyzer [17], an automatic tool for exhaustive checking of Alloy models. These studies revealed bugs in the actual systems, providing empirical evidence in support of the hypothesis. However, the studies did not directly check actual implementation code.

The challenge in evaluating/exploiting the hypothesis for code is doing exhaustive checking of code. Our approach uses systematic testing for all inputs within a given scope. In object-oriented programs, an input is constructed from objects of different classes; a test input is within a scope s if at most s objects of any class appear in it. For test input generation and correctness checking, we use Korat (Section 3), a tool that we have developed for testing Java programs [8].

The heart of Korat is a technique for systematically generating all (non-isomorphic) inputs that satisfy a Java predicate, i.e., inputs for which the predicate returns `true`. We have used this technique for specification-based, black-box testing [7]: given a specification for a method, Korat automatically generates all test inputs (within a given small scope) that satisfy the method precondition; Korat then executes the method on each test input and uses the method postcondition as a test oracle to check the correctness of each output. For specifications, Korat uses the Java Modeling Language (JML) [22], and for checking correctness, Korat builds on the JML tool-set [10]. This paper also presents how our technique can be used for white-box testing (Section 3.4), which can reduce total testing time.

Using tools for systematic testing, we have found bugs in several applications [25], including a networking architecture [2], a constraint solver for first-order logic [17], and a fault-tree analyzer [31]. Korat has been also reimplemented in the AsmL Test Generator tool (AsmLT) [1] and successfully used for testing an XPath compiler [30]. Scalability of systematic testing tools does not depend as much on the complexity/size of the tested code as it depends on the complexity of data that the code operates on. This paper focuses on implementations of several Java data structures, including some from the Java Collections Framework [32]. We evaluate the "small scope hypothesis" for these programs using code coverage and mutation testing.

Code coverage is a common criterion for assessing the quality of a test suite [7]. Measuring code coverage involves executing the program on each input and recording statements and branches that get executed. Statement (branch) coverage is then the ratio of the number of executed statements (branches) to the number of total statements (branches) in the program; *complete coverage* is the ratio of 100%. Since Korat uses executable specifications, we also measure *specification coverage* [9].

Mutation testing is another criterion for assessing the quality of a test suite [14, 27]. Mutation testing determines how many bugs a test suite can find. It proceeds in two steps. In the first step, several *mutant*s are generated from the original program, by performing one or more syntactic modifications as specified by *mutation operators*, e.g., replacing a variable with another variable (of a compatible type), say `n.left` with `n.right`. These operators corresponds to typical bugs that programmers make. For several languages, including Java, possible operators are characterized in [3, 20, 21, 28].

In the second step, the original program and each mutant are executed on each input and the corresponding outputs are compared. If a mutant generates an output different than the original program, the test input is said to *kill* the mutant. For a given set of inputs, the rate of mutant killing is the ratio of the number of killed mutants to the total number of mutants. Mutation testing tools were implemented for some languages, such as Mothra [21] for Fortran and Proteum [13] for C. We have implemented Ferastrau (Section 4) for Java; to the best of our knowledge, this is the first tool for mutation testing of Java programs.

The experimental results show that systematic testing within small scopes can achieve complete coverage and kill almost all of the mutants, even for intricate methods that manipulate complex data structures. We also compare systematic testing with randomly selected test inputs; the results show that systematic testing for all inputs within some scope can be more effective than random testing with bigger inputs. These results provide evidence that the "small scope hypothesis" holds for data structures.

Moreover, evaluating the hypothesis is not only about characterizing benchmarks; it also determines whether a tool for systematic testing can be practically used, i.e., how big a scope it can test in a given time. Once we establish that the "small scope hypothesis" holds for some type of benchmarks (or we cannot establish that), we dispense complex testing metrics, and use the scope itself as a metric. The experimental results show that for all benchmarks and the scopes that give high quality test suites, Korat can generate all inputs and check correctness in less than five minutes, often within a few seconds. We show how Korat can generate inputs even faster using a library of *dedicated generators* (Section 3.2) that also make specifications easier to write.

Previous work [8] has presented the basic ideas of Korat. The new contributions of this paper are:

- Evaluation of the "small scope hypothesis" for several data structure implementations;
- Introduction of dedicated generators, a Korat extension that allows faster input generation and easier specification writing;
- Application of Korat technique to white-box testing;
- Evaluation of the Korat tool;
- Design and implementation of Ferastrau, a tool for mutation testing of Java programs.

2. EXAMPLE

This section illustrates how programmers can use Korat to test their programs. As a running example, we use a method for removing an element from a set implemented as a binary search tree. Figure 1 shows JML-annotated Java code that declares a binary tree and its `remove` method. Each object of the class `SearchTree` represents a binary search tree. The `size` field contains the number of nodes in the tree. Objects of the inner class `Node` represent nodes of the trees. The elements of the set are stored in the `info` fields. The elements implement the interface `Comparable`, which provides the method `compareTo` for comparisons. Appendix A shows the full code for the `remove` method.

```
class SearchTree {
    Node root; // root node
    int size;  // number of nodes in the tree
    static class Node {
        Node left;  // left child
        Node right; // right child
        Comparable info; // data
    }

    /*@ normal_behavior // non-exceptional specification
      @     // precondition
      @     requires repOk();
      @     // postcondition
      @     ensures repOk() && !contains(info) &&
      @             \result == \old(contains(info));
      @*/
    boolean remove(Comparable info) { ... }

    boolean repOk() {
        // checks that empty tree has size zero
        if (root == null) return size == 0;
        // checks that the input is a tree
        if (!isAcyclic()) return false;
        // checks that size is consistent
        if (numNodes(root) != size) return false;
        // checks that data is ordered
        if (!isOrdered(root)) return false;
        return true;
    }
}
```

Figure 1: Example code and specification.

The JML annotations specify partial correctness for the example `remove` method. The `normal_behavior` annotation specifies that if the precondition (annotation `requires`) is satisfied at the beginning of the method, then the method must satisfy the postcondition (annotation `ensures`) at the end, and it must return without raising an exception. The method `repOk` is a Java predicate that checks the *representation invariant* [24] of the corresponding data structure. For illustrative purposes, we put `repOk` in the precondition and postcondition; in practice, it is usually given as a class invariant (annotation `invariant`) that is implicitly conjoined with the precondition and postcondition [22]. Good programming practice [24] suggests that implementations of abstract data types provide these predicates, as they are useful for checking correctness of the implementations.

In this example, `repOk` checks if the input is a valid binary search tree with the correct `size`. First, `repOk` checks if the tree is empty. If not, `repOk` checks that there are no undirected cycles along `left` and `right`, that the number of nodes reachable from `root` is `size`, and that all elements in the left (right) subtree of a node are smaller (larger) than the element in that node. Appendix A shows the full code for `repOk` (and the methods it invokes). The same `repOk` is also used for `add` and other methods in `SearchTree`. Manually developing a high-quality test suite for all methods in a data structure is typically much harder than writing `repOk` invariant that Korat uses to automatically generate test inputs.

The method `contains` checks that the tree contains the given element. The JML keyword `\result` denotes the return value of the method. In this example, `remove` returns `true` iff it removes an element from the tree. The JML keyword `\old` denotes that its expression should be evaluated in the pre-state, i.e., the state immediately before the method's invocation.

To test the `remove` method in a black-box setting, Korat first generates valid inputs for the method. Each input is a

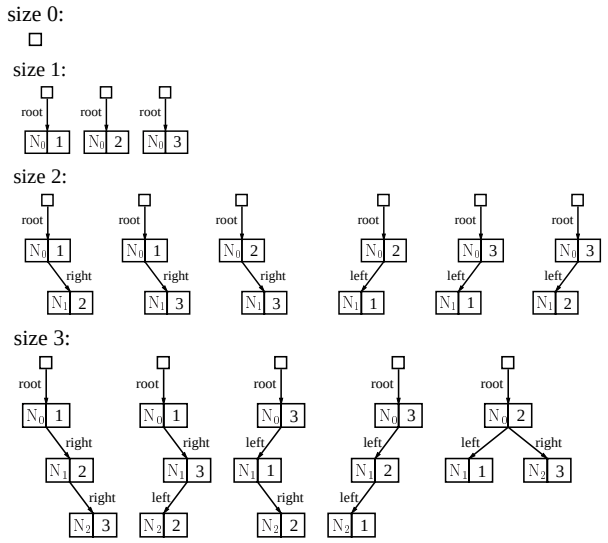


Figure 2: Trees generated for scope three.

pair of a tree and an element. The precondition defines valid inputs: the tree satisfies `repOk`, and the element is unconstrained. To limit the number of inputs, Korat uses a *finitization* (Section *3.1.1*) that specifies bounds on both the number of objects to be used to construct data structures and the values stored in the fields of these objects. For trees, finitization specifies the maximum number of nodes and the possible elements; a tree is in scope s if it has at most s nodes and s elements. Two trees are *isomorphic* if they have the same branching structure and isomorphic elements, irrespective of the identity of the actual nodes or elements in the trees.

Given a finitization and bounds, Korat generates all non-isomorphic input pairs that satisfy the precondition. For example, in scope three, Korat generates 45 input pairs that are the Cartesian product of the 15 trees shown in Figure 2 and the three elements. For the `SearchTree` benchmark, we use Korat to generate inputs and check correctness of `remove` and `add` methods. As another example, in the scope seven, Korat generates 41300 input pairs for both these methods in less than ten seconds. With dedicated generators (Section 3.2), it takes less than three seconds to generate these inputs.

Korat uses the JML tool-set [10] to translate method postconditions (and JML assertions) into Java runtime assertions. After generating the inputs, Korat invokes the method, with assertions, on each input and reports a counterexample if the method fails to satisfy the postcondition. This process checks the correctness of the method for the given scope. For example, for scope seven, Korat takes less than two seconds to check both `remove` and `add` for all 41300 inputs.

We evaluate the "small scope hypothesis" by measuring how coverage and the rate of mutant killing vary with the scope. We use our Ferastrau framework for mutation testing. The "output" for `remove` consists of both its `boolean` return value and the value of the receiver tree in the post-state, i.e., the state immediately after the method's invocation. Figure 3 shows the variation for the `SearchTree` benchmark; a certain small scope is sufficient to achieve complete coverage and kill most of the mutants. Korat generates inputs and checks

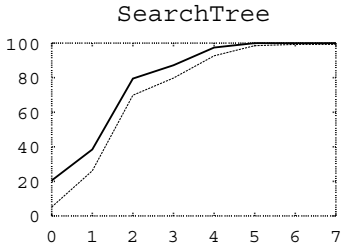


Figure 3: Variation of statement coverage (thick line) and rate of mutant killing (thin line) with scope.

correctness for these scopes in less than 15 seconds.

3. KORAT

This section describes Korat [8], a tool that automates both test-input generation and correctness checking for Java programs. The heart of Korat is a technique for generating inputs that satisfy a Java predicate (Section 3.1). We show how to apply this technique to black-box (Section 3.3) and white-box (Section 3.4) testing by constructing appropriate predicates from method preconditions and postconditions.

3.1 Valid input generation

Given a Java predicate and a bound on its input, Korat automatically generates all non-isomorphic inputs that are *valid*, i.e., inputs for which the predicate returns `true`. Korat uses a *finitization* (Section *3.1.1*) to bound the *state space* (Section *3.1.2*) of predicate inputs. Korat uses backtracking (Section *3.1.3*) to systematically explore this state space. Korat generates *candidate inputs* and invokes the predicate on them to check their validity. Naive checking of all possible candidate inputs would prohibit searching very large state spaces. Korat uses two optimizations: 1) pruning based on accessed fields and 2) generating only non-isomorphic candidates. These optimizations speed up the search without compromising its soundness and completness.

Korat prunes the search based on the following observation: if the predicate returns without reading some fields of a candidate input, the validity of the candidate must be independent of the values of those fields. Korat monitors accesses that the predicate makes for each execution to determine which fields it reads. To monitor the accesses, Korat instruments the predicate and all the methods that the predicate transitively invokes.

Each candidate that Korat generates has one root object; a tuple of objects is essentially one object of a tuple class (Section 3.3). In Java, structure isomorphism is defined based on object identity; two candidates are isomorphic if the parts of their object graphs reachable from the root are isomorphic:

Definition: Let $O_1, \ldots, O_n$ be some sets of objects from n classes. Let $O = O_1 \cup \ldots \cup O_n$, and suppose that candidates consist only of objects from O, i.e., pointer fields of objects in O can either be `null` or point to other objects in O. Let P be the set consisting of `null` and all values of primitive types, such as `int`. Let $r \in O$ be a root object, and let $R_C(r)$ be the set of all objects reachable from r in C. Two candidates, C and C', are *isomorphic* iff there exists a

```
Finitization finSearchTree(int numNode,
    int minSize, int maxSize, int minInfo, int maxInfo) {
  Finitization f = new Finitization(SearchTree.class);
  ObjSet nodes = f.createObjects("Node", numNode);
  nodes.add(null);
  f.set("root", nodes);
  f.set("size", new IntSet(minSize, maxSize));
  f.set("Node.left", nodes);
  f.set("Node.right", nodes);
  f.set("Node.info", new IntegerSet(minInfo, maxInfo));
  return f;
}
Finitization finSearchTree(int scope) {
  return finSearchTree(scope, 0, scope, 1, scope);
}
```

Figure 4: Two finitizations for the `repOk` method.

permutation π on $O \cup P$ that is identity on P and that maps objects from O_i to objects from O_i for all $1 \leq i \leq n$, such that:

$$\forall o \in R_C(r).\ \forall f \in fields(o).\ \forall v \in O \cup P.$$
$$o.f == v \text{ in } C \Leftrightarrow \pi(o).f == \pi(v) \text{ in } C',$$

where the operator `==` is Java's comparison by object identity. Isomorphism between candidates partitions the state space into *isomorphism partitions*. Since candidates and valid inputs are rooted and edge-labeled, it is easy to check isomorphism. However, Korat does not do that explicitly; instead, it avoids generating isomorphic valid inputs by not even considering isomorphic candidates.

In summary, Korat generates all non-isomorphic valid inputs within specified bounds; the search has these properties:

- **Soundness:** Korat does not generate any input for which the predicate returns `false`.
- **Completness:** Korat generates at least one input from each isomorphism partition for which the predicate returns `true`.
- **Optimality:** Korat generates at most one input from each isomorphism partition for which the predicate returns `true`.

We next desribe the most relevant parts of Korat, which allows us to present recent extensions; more details on Korat can be found in [8]. For illustration, we consider that the predicate is the `repOk` method from `SearchTree`, and we show how Korat generates valid trees. (Section 3.3 presents how Korat generates valid test inputs for the `remove` method.)

3.1.1 Finitization

To generate a finite state space for predicate's inputs, the search algorithm needs a finitization, i.e., a set of bounds that limits the size of the inputs. The inputs can consist of objects from several classes, and the finitization specifies the number of objects for each of those classes. A set of objects from one class forms a *class domain*. The finitization also specifies a set of values for each field; this set forms a *field domain*, which is a union of some class domains.

In the spirit of Extreme Programming [5] that uses the implementation language familiar to programmers for testing and specification, Korat provides a `Finitization` class that allows finitizations to be written in Java. Korat automatically generates a finitization *skeleton* from the type declarations in the Java code. The AsmLT [1] additionally provides a GUI for generating skeletons. Testers can further specialize or generalize this skeleton.

Figure 4 shows two finitizations for the example `repOk` method. For `finSearchTree(s)`, Korat generates all valid

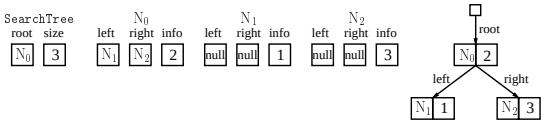


Figure 5: Candidate that is a valid `SearchTree`.

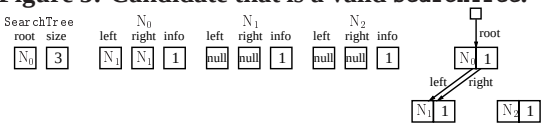


Figure 6: Candidate that is not a valid `SearchTree`.

inputs within scope `s`. The `createObjects` method specifies that the input contains at most `numNode` objects from the class `Node`. The `set` method specifies a field domain for each field.

3.1.2 State space

Korat uses the finitization presented in Figure 4 to construct the state space of inputs to the `repOk` method. Consider the case for `finSearchTree(3)`. Korat first allocates one `SearchTree` object and three `Node` objects. These `Node` objects form the `Node` class domain. Korat then assigns a field domain and a unique *identifier* to each field. The identifier is the index into the *candidate vector*. In this example, the vector has length 11: the single `SearchTree` object has two fields (`root` and `size`) and the three `Node` objects have three fields each (`left`, `right`, and `info`).

A *candidate* input is represented by a valuation of the candidate vector. The state space of inputs consists of all possible valuations of the candidate vector, i.e., it is the Cartesian product of the field domains for all fields. In this example, the domain for `root`, `left`, and `right` has four elements (`null` and three `Node` objects), the domain for `size` has four elements, and the domain for `info` has three elements. Therefore, the state space has $4 \cdot 4 \cdot (4 \cdot 4 \cdot 3)^3 = 1769472 > 2^{20}$ potential candidates. For `scope`$= n$, the state space has $(n+1)^{2(n+1)} \cdot n^n$ potential candidates. Figure 5 shows an example candidate tree that is a valid binary search tree with three nodes. Not all valuations represent valid binary search trees. Figure 6 shows an example candidate tree that is not a tree; `repOk` returns `false` for this candidate.

3.1.3 Search

To systematically explore the state space, Korat orders all the elements in every class domain and every field domain. The ordering in each field domain is consistent with the orderings in the class domains, and all the values that belong to the same class domain occur consecutively in the ordering of each field domain.

Each candidate input is a vector of *field domain indices* into the corresponding field domains. For our running example with scope 3, assume: the `Node` class domain is ordered [N_0,N_1,N_2]; the field domain for `root`, `left`, and `right` is ordered [`null`,N_0,N_1,N_2] (`null` by itself forms a class domain); the domain for `size` is ordered `[0,1,2,3]`; and the domain for `info` is ordered [`Int(1)`,`Int(2)`,`Int(3)`]. According to this ordering, the candidate in Figure 5 (Figure 6) corresponds to the valuation `[1,3,2,3,1,0,0,0,0,0,2]` (`[1,3,2,2,0,0,0,0,0,0,0]`) for candidate vector.

The search starts with the candidate vector set to all zeros. For each candidate, Korat sets fields in the objects according to the values in the vector. Korat then executes the predicate to check the validity of the current candidate. During the execution, Korat monitors the fields that the predicate accesses. Specifically, Korat builds a *field-ordering*: a list of the field identifiers ordered by the first time the predicate accesses the corresponding field. As an illustration, consider the invocation of `repOk` on the candidate shown in Figure 6. In this case, `repOk` accesses only the fields [`root`,N_0`.left`,N_0`.right`] (in that order) before returning `false`. Hence, the field-ordering that Korat builds is `[0,2,3]`.

After the predicate returns, Korat generates the next candidate vector backtracking on the accessed fields. Korat first increments the field domain index for the last field in the field-ordering. If the index exceeds the domain size, Korat resets the index to zero, increments the domain index of the previous field in the field-ordering, and so on. Continuing with our example, the next candidate takes the next value for N_0`.right`, which is N_2 by the above order; the other fields do not change. This prunes from the search $4^5 \cdot 3^3 = 27648$ candidate vectors of the form `[1,_,2,2,_,_,_,_,_,_]` that have the (partial) valuation: `root`=N_0, N_0`.left`=N_1, N_0`.right`=N_1. The pruning does not rule out any valid data structure because `repOk` did not read the other fields, and it would have returned `false` irrespective of the values of those fields. If the predicate returns `true`, Korat outputs all (non-isomorphic) candidates that have the same values for the accessed fields as the current candidate. The search then backtracks to the next candidate.

Recall that Korat orders the values in the class and field domains. Additionally, each execution of the predicate on a candidate imposes an order on the fields in the field-ordering. Together, these orders induce a lexicographic order on the candidates. The Korat search algorithm generates inputs in the lexicographical order. Moreover, Korat avoids generating multiple candidates that are isomorphic to one another: for each isomorphism partition, Korat generates only the lexicographically smallest candidate in that partition. Conceptually, Korat avoids generating isomorphic candidates by incrementing field domain indices by more than one. This optimization is presented in detail in [8].

3.2 Dedicated generators

Korat provides a library of *dedicated generators* that make it easier to write spefications and also enable faster generation of valid inputs. Certain checks are common in class invariants (`repOk` methods), e.g., that a linked data structure is acyclic along some fields or that an array has all elements different or ordered. The library provides methods for these checks. The specifications, as well as any other code, can use these library methods; in regular execution, these methods behave like other Java methods. However, when Korat generates valid inputs, it uses the special knowledge about these methods to further optimize its search.

In `SearchTree`, `repOk` invokes the method `isAcyclic` that checks that the nodes reachable from the `root` field form a tree along the `left` and `right` fields. Appendix A shows one way to write `isAcyclic`; it has about 20 lines of code. Instead, we could just use the library method `korat.isTree(root, new String[]{"left", "right"})`. This method is parametrized over the root node and the names of the fields. Given a root node, `isTree` checks that the reachable nodes form a tree; essentially, it means that no node repeats in the traversal of the nodes reachable from the root. The search for the library method is implemented to take into account this fact.

When Korat generates an input that satisifes `isTree` along some fields, it does not try all (non-isomorphic) possibilities for those fields. Instead, each field is either `null` or points to a node that is not already in the tree. In our example `finSearchTree(s)`, this reduces the number of possiblities for one field from `s`$+1$ to 2. In the library, the implementation of `isTree` uses the basic dedicated generator `korat.isIn(field, set)` that, while searching, assigns to the `field` only the values from the `set`, and while checking, checks that the value of `field` is in the `set`.

The library includes the basic dedicated generators for checking: that a value is in a set, that two values are equal, that a value is less/greater than another value, and that a value is of a certain class (`instanceof`). The library also includes generators for combining other generators for checking: negation, conjunction, and disjunction. Finally, the library includes several higher-level generators, implemented using basic generators, which check structural constraints such as acyclicity or that elements of an array are sorted.

It is easy to add new generators; in theory, we could even add for each data structure that we consider a special-purpose generator that generates all valid inputs without any backtracking. For example, such a generator for red-black trees was developed and used for testing in [4]. However, we do not do that; the library that we use in the experiments has only generators that are applicable for several data structures. In practice, we do not expect Korat users to extend the library, but instead to use Korat as general-purpose search.

3.3 Black-box Testing

In black-box testing, Korat tests a method without considering the method's code. Korat systematically generates inputs that satisfy the method precondition, executes the method on each of the inputs and checks the output using a *test oracle*. To generate test inputs for a method m, Korat first constructs a Java class corresponding to the m's inputs and a predicate corresponding to the m's precondition. Korat then generates valid inputs for that predicate; each of these inputs corresponds to a valid test input for m. For the `remove` method from Section 2, the corresponding class and the predicate `removePre` are shown in Figure 7. The predicate simply invokes `repOk` on the (implicit) `this` parameter of `remove`; the parameter `info` is unconstrained.

3.3.1 Checking correctness

After generating all valid test inputs for a method, Korat invokes the method on each input and checks each output with a test oracle. A simple test oracle could check partial correctness of a method by invoking `repOk` in the post-state to check if the method preserves its class invariant. If the result is `false`, the method under test is incorrect, and the

```
class SearchTree_remove {  // inputs to "remove"
    SearchTree This;       // (implicit) "this" parameter
    Comparable info;       // "info" parameter

    // for black-box testing of "remove"
    boolean removePre() {  // precondition for "remove"
        return This.repOk();
    }

    // for white-box testing of "remove"
    boolean removeFail() { // failure for "remove"
        if (!removePre()) return false;
        try { // invoke "remove" with JML assertions
            This.remove(info);
        } catch (JMLAssertionException e) {
            return true;   // postcondition not satisfied
        }
        return false;
    }
}
```

Figure 7: Class for inputs to the `remove` method.

	testing framework		
testing activity	JUnit	jmlunit	Korat
generating test inputs		-	√
generating test oracle		√	√
running tests	√	√	√

Table 1: Comparison of several testing frameworks for Java. Automated testing activities are indicated with '√'; jmlunit generates inputs using directly the Cartesian product, which cannot handle very large input spaces.

input provides a concrete counterexample.

The Korat tool currently uses the JML tool-set to automatically generate test oracles from method postconditions (and method assertions in general), as in the jmlunit framework [10]. The JML tool-set translates JML postconditions (and assertions) into runtime Java assertions. If an execution of a method violates such an assertion, an exception is raised. Test oracle catches these exceptions and reports correctness violations. These exceptions are different from the exceptions that the method specification allows, and Korat leverages JML to check both normal and exceptional behavior of methods. More details on the JML tool-set and translation can be found in [22].

Korat can also use jmlunit to combine JML test oracles with JUnit [6], a popular framework for unit testing of Java modules. JUnit automates test execution and error reporting, but requires programmers to provide test inputs and test oracles. In jmlunit, the Cartesian product is directly used to generate test inputs, which cannot handle very large input spaces. Additionally, jmlunit does not generate complex data structures, but requires users to create and provide them. Korat further automates and optimizes generation of test inputs, thus automating the entire testing process. Table 1 summarizes the comparison of these testing frameworks.

3.4 White-box Testing

In white-box testing, Korat tests a method considering the method's code. To test a method m, Korat first constructs a predicate corresponding to the negation of m's correctness. If a valid input is found for this predicate, m is incorrect, and the input provides a counterexample. For the `remove` method, the corresponding predicate `removeFail` is shown in Figure 7. This predicate first invokes `removePre`; if it is not satisfied, the input is not a valid test input for `remove` and cannot be a counterexample. If the input is valid, `remove` is executed, together with the JML-translated assertions. If this execution raises a JML exception, `remove` failed to satisfy its specification.

The difference between predicates for white-box and black-box testing is in the invocation of the method under test; in our example, `removeFail` invokes `remove`, but `removePre` does not. This means that for generating valid inputs to `removeFail`, Korat instruments `remove`, among other methods, and monitors the accesses that `remove` makes to the candidate. This by itself makes one execution of `remove` slower. But it "opens" the body of `remove` for the optimizations that Korat performs to prune the search. In general, this can significantly reduce the time to test the method.

4. MUTATION TESTING

This section presents design and implementation of Ferastrau, a tool for mutation testing of Java programs. *Mutation testing* is a criterion for assessing the quality of a set of test inputs [14, 27]. Mutation testing proceeds in two steps. In the first step, a set of *mutant*s is generated from the original program by applying *mutation operators* to perform one or more syntactic modifications. Section 4.1 presents mutant generation in Ferastrau. In the second step, the original program and each mutant are executed on each input and the corresponding outputs are compared. If a mutant generates an output different than the original program, the test input is said to *kill* the mutant. Section 4.2 presents how Ferastrau executes mutants and compares the outputs.

4.1 Mutant generation

We have implemented mutant generation by changing the Sun's `javac` compiler. Ferastrau performs a source-to-source translation: it parses each class of the original program into an abstract syntax tree, applies some mutation operators to the trees, and outputs the source of the mutants. Ferastrau applies the following mutation operators:

- Mutate a Java operator to another operator (of the same type), e.g., '`+`' to '`-`', '`==`' to '`!=`', '`<`' to '`<=`' etc.
- Mutate a variable to another variable (of a compatible type), e.g., a local variable `i` to `j` or an instance variable `n.left` to `n.right`.
- Mutate an invocation of a method to another method (of a compatible signature). (Ferastrau does not replace some special methods, such as `notify`; programmers typically do not make such mistakes.)

The above operators modify only the code of methods, and not classes, i.e., do not add/remove a method or a field. These operators correspond to subtle mistakes that manifest only for non-trivial inputs, as the results in Section 5.3 show. It is easy to add new operators to Ferastrau to test different kind of mistakes.

Ferastrau generates mutant classes that have the same name as the corresponding original classes. For reasons explained below, Ferastrau provides two approaches: 1) generate the same classes with both the original program and the mutants or 2) generate different classes. Suppose that the original programs contains `temp.right` that is to be

mutated to `left.right`. The first approach uses *metamutants* [33]: the mutations are guarded by boolean variables that are appropriately set during mutant execution; it generates one class with `(MUT ? left : temp).right`. The second approach simply generates `left/*temp*/.right` in another class.

4.2 Mutant execution

After generating the mutants, Ferastrau uses a set of test inputs to perform mutation testing. Our experiments use inputs generated by Korat. Ferastrau executes the original program and the mutants for each input and compares their respective outputs. Ferastrau assumes that the original program terminates for all test inputs; mutation testing tools for other languages [13, 21] make the same assumption. Since Ferastrau operates on Java and has to handle potentially large number of inputs, additional questions arise:

- How to compare outputs and name mutated classes?
- Whether to execute the original program and the mutants in a single run or in separate runs?
- How to handle non-termination and exceptional termination of the original program and the mutants?

We next describe how Ferastrau addresses these questions and then list the criteria that Ferastrau uses to kill a mutant.

Recall that the "output" of a method refers to both the return value and the objects in the post-state. Comparison is easy when these are primitive values, but the objects can represent complex structures. Ferastrau by default uses `equals` methods to compare outputs, following Java convention of using `equals` for equality comparisons of objects. This allows comparisons based on *abstract* values; for example, two binary search trees that implement sets may be structurally different at the *concrete* level of the implementation, but if they represent the same set, they are equal according to the `equals` method. The use of `equals` requires that Ferastrau generates mutant classes that have the same name as the corresponding original classes.

Ferastrau executes the original program and the mutants in a single run; otherwise, it would need to serialize all the outputs, which could produce very large files for inputs exhaustively generated by Korat. When Ferastrau generates the original program and the mutants in different classes, it needs to execute several classes with the same name in a single Java Virtual Machine (JVM). Ferastrau then uses a different `ClassLoader` [32] to load in the classfiles of the original program and each mutant. To compare objects, Ferastrau uses serialization through a buffer in memory. This approach works better for large code with small data. When Ferastrau uses metamutants, the guarding boolean variables slow down the execution. This approach works better for small code with large data.

Ferastrau assumes that the original program terminates for all test inputs, either normally or exceptionally. These exceptions are allowed by the specification, and they are not errors. Ferastrau handles non-termination of mutants by running them in a separate thread and setting a time limit for execution. The mutants can terminate either normally or exceptionally. Ferastrau catches all exceptions (in terms of Java, all `Throwable` objects) that the executions raise. This allows Ferastrau to compare the outputs, even when they are exceptional, as well as to catch all errors in the mutants. This handles the situations when the mutant runs out of stack or heap memory and JVM raises `StackOverflowError` or `OutOfMemoryError`.

Ferastrau uses the following criteria to kill a mutant:

- The mutant's output does not satisfy some class invariant (`repOk`), which is a precondition for `equals`.
- The mutant's output differs from the output of the original program; any of the outputs can be normal or exceptional.
- The mutant's execution exceeds the time limit.
- The mutant's execution runs out of memory.

5. EXPERIMENTAL RESULTS

This section presents the experiments that evaluate the "small scope hypothesis" and the Korat tool. We first discuss Korat's performance for test input generation and checking method correctness. We then discuss how the coverage and the rate of mutant killing vary with the scope. We finally compare exhaustive testing with randomly selected test inputs. We performed all timed experiments on a Linux machine with a 1.8GHz Pentium 4 processor using Sun's Java 2 SDK1.3.1 JVM.

5.1 Benchmarks and methods

Table 2 lists the benchmarks and methods that we use to measure Korat's performance. We use Korat to generate inputs and check the correctness of outputs for the *target* methods. These methods implement the standard operations on their corresponding data structures [12]. Executing these methods also tests some *helper* methods because they are invoked either when executing the target methods or when checking their correctness (e.g., from postconditions).

`SearchTree` is presented in Section 2. `DisjSet` is an array-based implementation of the fast union-find data structure [12]; this implementation uses both path compression and rank estimation heuristics to improve efficiency. `HeapArray` is an array-based implementation of the heap (priority queues) data structure. `BinomialHeap` and `FibonacciHeap` are dynamic data structures that also implement heaps, but differ in complexity for certain operations [12].

`LinkedList` is the implementation of linked lists in the Java Collections Framework, a part of the standard Java libraries [32]. This implementation uses doubly-linked, circular lists. This benchmark is also representative for linked data structures such as stacks and queues. The elements in `LinkedList` are arbitrary objects; `SortedList` is structurally identical to `LinkedList`, but the elements are sorted. This benchmark is similar to the examples used in some shape analyses [23, 26]. `TreeMap` implements the `Map` interface using red-black trees [12]. `HashSet` implements the `Set` interface, backed by a hash table [12].

`AVTree` implements the *intentional name* trees that describe properties of services in the Intentional Naming System (INS) [2], an architecture for service location in dynamic networks. The original implementation of INS had errors that we revealed with exhaustive testing [25] and corrected. We use the corrected version as the original program in these experiments, but (some of) the mutants have errors.

benchmark	"target" methods	some "helper" methods	# ncnb lines	# branches	# mutants
SearchTree	add, remove	contains	85	20	272
DisjSet	union, find	compressPath	29	8	243
HeapArray	insert, extractMax	heapifyUp, heapifyDown	51	9	274
BinomialHeap	insert, extractMin union, delete	contains, decrease merge, findMin	182	33	292
FibonacciHeap	insert, extractMin union, delete	contains, decrease cascadingCut, cut, consolidate	171	31	297
LinkedList	add, remove, reverse	contains, ListIterator.next	102	16	244
SortedList	insert, remove sort, merge	contains	176	29	231
TreeMap	put, remove	get, fixAfterInsertion containsKey, fixAfterDeletion rotateLeft, rotateRight	230	47	293
HashSet	add, remove	contains, HashMap.containsKey HashMap.put, HashMap.remove HashMap.rehash	113	20	244
AVTree	lookup	extract	199	26	205

Table 2: Benchmarks and target methods. Each benchmark is named after the main class; Korat generates data structures that also contain objects from other classes. Korat generates inputs and checks outputs for the target methods, thereby also testing helper methods. We tabulate the number of non-comment non-blank lines of source code in all those methods, the number of branches, and the number of mutants generated by Ferastrau.

5.2 Test generation and correctness checking

Table 3 shows Korat's performance for test generation and correctness checking for some scopes. Appendix B presents the results for many other scopes. For each benchmark, all size parameters and maximum elements are set to the scope value. For each benchmark, the tabulated scope is sufficient to achieve the maximum coverage and kill almost all the mutants. We tabulate the time Korat takes to generate all valid test inputs (without and with dedicated generators) and to check the correctness of methods. All times are elapsed real times in seconds from the start of Korat to its completion, without the JVM initialization that takes around 0.5 seconds.

Number of inputs that is generated is the sum of numbers of inputs for *all* target methods. Similarly, the generation and checking times are sums of times for all target methods. We use Korat to separately generate inputs for each method. However, when two methods have the same precondition (e.g., `remove` and `add` for `SearchTree`), we could reuse the inputs and thus reduce the generation time. The postconditions for all methods specify typical partial correctness properties; they require resulting data structures to be valid and to (not) contain the input elements, depending on the method.

For scopes in Table 3, the size of the search space is between 2^{25} and 2^{150}. The actual size of search spaces for several data structures can be found in [8]; for some scopes in those experiments, as well as for some scopes in Appendix B, Korat explores search spaces with size over 2^{250}. In all cases, Korat completes in less than two minutes, often in just a few seconds. The use of dedicated generators reduces the generation times for up to 75% (for `SearchTree`). Since dedicated generators have a higher overhead, their use sometimes increases the generation time, specially for small scopes. But in all cases, dedicated generators make it easier to write specifications.

These results show that Korat can efficiently generate all inputs even for very large search spaces, primarily because the search pruning allows Korat to explore only a tiny fraction of these spaces. The key to effective pruning is backtracking based on fields accessed during `repOk`'s executions. Without backtracking, and even with isomorphism optimization, Korat would consider infeasibly many candidates. Isomorphism optimization further reduces the number of considered candidates, but it mainly reduces the number of valid inputs. As shown in [8], Korat generates exactly the number of non-isomorphic data structures given in the Sloane's On-Line Encyclopedia of Integer Sequences [29].

5.3 Coverage and mutant testing

Table 3 also shows specification/code coverage and the rate of mutant killing. Since Korat uses executable specifications, we measure *specification coverage* [9] as code coverage for the predicate that corresponds to the method's precondition (e.g., `removePre`). We measure this coverage while Korat generates valid inputs for the predicate, i.e., valid test cases for the method. For most benchmarks, the tabulated scopes achieve complete coverage, both for statements and branches. It is not always 100%, because finitizations do not even put for fields some values that do not satisfy the predicate (e.g., `finSearchTree` does not put `null` for `info`). Specification coverage typically reaches maximum before code coverage (Appendix B).

Figure 8 shows graphs that relate scope with the statement coverage of code and the rate of mutant killing. The code coverage is measured for all target and helper methods, since they are all executed. For most benchmarks, Korat generates inputs that achieve complete coverage, both for statements and branches. For other benchmarks, the coverage is not complete because no input for target methods could trigger some exceptional behavior of helper methods.

For example, the (target) `reverse` method for lists creates a `ListIterator` and invokes some (helper) methods on it. In general, these helper methods could raise exceptions, such as `ConcurrentModificationException` or `NoSuchElementException`, but the target methods never invoke the helper methods in such a way. In terms of JML specifications, the target methods invoke the helper methods in pre-states that satisfy the precondition for `normal_behavior`, and not for `exceptional_behavior`.

		generation					checking			
		gen.	ded.	spec. coverage		#	time	code coverage		mutants
benchmark	scope	[sec]	[sec]	st. [%]	br. [%]	inputs	[sec]	st. [%]	br. [%]	killed [%]
SearchTree	7	9.03	2.19	94.74	96.67	41300	1.25	100.00	100.00	99.26
DisjSet	5	10.91	9.87	100.00	100.00	1246380	19.93	100.00	100.00	95.06
HeapArray	7	7.09	6.21	90.00	92.86	1175620	17.58	100.00	100.00	96.71
BinomialHeap	7	35.60	28.06	97.67	98.00	2577984	75.96	100.00	100.00	96.91
FibonacciHeap	5	14.14	12.94	97.78	98.28	941058	23.37	100.00	100.00	88.88
LinkedList	7	0.74	0.71	100.00	100.00	58175	1.54	90.57	84.38	99.59
SortedList	7	22.68	21.13	100.00	100.00	1047608	37.91	92.50	89.66	97.40
TreeMap	7	3.28	1.75	100.00	100.00	12754	0.73	100.00	91.49	89.76
HashSet	7	3.38	2.88	89.47	92.31	54844	1.55	100.00	100.00	92.21
AVTree	5	87.13	43.41	96.67	96.88	417878	134.51	94.12	92.31	93.65

Table 3: Korat's performance for test generation (with regular and dedicated generators), specification coverage (statement and branch), correctness checking, code coverage (statement and branch), and rate of mutant killing. All times are elapsed real times in seconds from the start of Korat to its completion. For all benchmarks and their sufficient scopes, Korat takes less than five minutes to generate all inputs and check correctness.

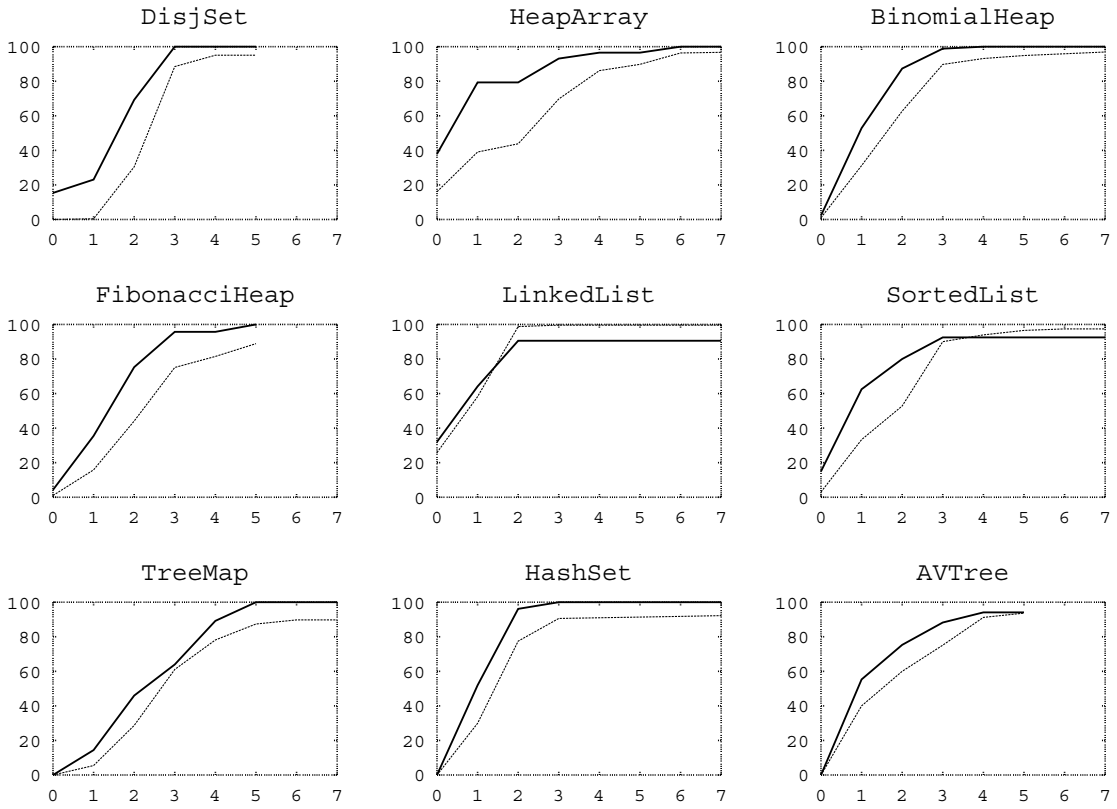


Figure 8: Variation of statement code coverage (thick line) and rate of mutant killing (thin line) with scope. For *all* benchmarks, Korat generates inputs that achieve the maximum coverage that is possible without directly generating inputs for helper methods.

For mutant testing, we use Ferastrau to generate between 200 and 300 mutants for each benchmark. We instruct Ferastrau to mutate the target methods and the helper methods the invoke, but not the helper methods that only specifications invoke. For most benchmarks, Korat generates inputs that kill over 90% of the mutants. We tried to manually inspect if the mutants that are not killed are, although syntactically different from the original program, *semantically* equivalent to it and thus no input could kill them. Due to the complexity of the benchmark methods, we were not able to definitely establish the equivalence for all surviving mutants, but those that we managed to inspect were indeed equivalent.

Notice that for some of the benchmarks the rate of mutant killing increases with scope even after achieving complete coverage. This can be expected because complete statement and branch coverage (or for that matter, any coverage criteria) does not guarantee absence of bugs [7]. Because of this, we take as sufficient the scope for which almost all mutants are killed, and not the scope that just achieves complete coverage. For all benchmarks and their respective sufficient scopes, Korat can generate all inputs and check correctness in less than five minutes, often within a few seconds. Korat can thus be effectively used for systematic testing of these benchmarks and similar data structures.

benchmark	scope	random mutants killed [%]	exhaustive scope-1	exhaustive scope
SearchTree	7	99.26	=	=
DisjSet	5	95.06	=	=
HeapArray	7	95.99	<	<
BinomialHeap	7	95.10	<	<
FibonacciHeap	5	86.87	>	<
LinkedList	7	99.59	=	=
SortedList	7	96.40	<	<
TreeMap	7	89.08	<	<
HashSet	7	91.39	<	<
AVTree	5	93.17	>	<

Table 4: Comparison of exhaustive testing with randomly selected test inputs. '=' means that both sets are equally good, '<' that random testing is worse, and '>' that random testing is better.

5.4 Random selection

We next evaluate the importance of exhaustive testing within a scope. Consider one benchmark, and let $T(s)$ be the set of all (non-isomorphic) test inputs within scope s for that benchmark. From $T(s)$, we randomly select a subset $R(s)$ whose cardinality is the same as the cardinality of $T(s-1)$. We then compare the quality of $R(s)$ against $T(s-1)$ and $T(s)$. For comparison, we use the rate of mutant killing. This criterion most directly measures the quality of test suite in detecting faults; the results are similar for code coverage. It is important to notice that randomly selected inputs are also generated with Korat; for complex data structures, it is not possible to simply generate random inputs.

Table 4 shows the comparison for all benchmarks. In most cases, randomly selected test inputs give a lower rate of mutant killing; only for `FibonacciHeap` and `AVTree`, the rate is higher for randomly selected inputs than for all inputs from the smaller scope. This means that the exhaustive testing for all inputs within some scope can be more effective than random testing with bigger inputs.

6. CONCLUSIONS

The "small scope hypothesis" argues that a high proportion of bugs can be found by testing the program for all test inputs within some small scope. In object-oriented programs, a test input is constructed from objects of different classes; a test input is within a scope of s if at most s objects of any given class appear in it. This paper evaluated the hypothesis for several implementations of data structures. We measured how statement coverage, branch coverage, and rate of mutant killing vary with scope. We used Korat and its extensions to perform exhaustive testing. This paper also presented the Ferastrau tool that we developed for mutation testing of Java programs.

The experimental results show that exhaustive testing within small scopes can achieve complete coverage and kill almost all of the mutants for data structure benchmarks, and additionally that exhaustive testing within some scope can be sometimes more effective than random testing with bigger inputs. The results also show that Korat can be used effectively to generate inputs and check correctness for these scopes. These results, together with previous studies that used systematic testing to expose bugs in real application [25], suggest that techniques that rely on exhaustive generation within scope $[1, 8, 34]$ are worth pursuing.

7. REFERENCES

[1] The AsmL test generator tool. `http://research.microsoft.com/fse/asml/doc/AsmLTester.html`.

[2] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. The design and implementation of an intentional naming system. In *Proc. 17th ACM Symposium on Operating Systems Principles (SOSP)*, Kiawah Island, Dec. 1999.

[3] H. Agrawal, R. A. DeMillo, R. Hathaway, W. Hsu, W. Hsu, E. W. Krauser, R. J. Martin, A. P. Mathur, and E. H. Spafford. Design of mutant operators for the c programming language. Technical Report SERC-TR-41-P, Purdue University, West Lafayette, IN, 1989.

[4] T. Ball, D. Hoffman, F. Ruskey, R. Webber, and L. J. White. State generation and automated class testing. *Software Testing, Verification & Reliability*, 10(3):149–170, 2000.

[5] K. Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000.

[6] K. Beck and E. Gamma. Test infected: Programmers love writing tests. *Java Report*, 3(7), July 1998.

[7] B. Beizer. *Software Testing Techniques*. International Thomson Computer Press, 1990.

[8] C. Boyapati, S. Khurshid, and D. Marinov. Korat: Automated testing based on Java predicates. In *Proc. International Symposium on Software Testing and Analysis (ISSTA)*, July 2002.

[9] J. Chang and D. J. Richardson. Structural specification-based testing: Automated support and experimental evaluation. In *Proc. 7th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, pages 285–302, Sept. 1999.

[10] Y. Cheon and G. T. Leavens. A simple and practical approach to unit testing: The JML and junit way. In *Proc. European Conference on Object-Oriented Programming (ECOOP)*, June 2002.

[11] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, Cambridge, MA, 1999.

[12] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 1990.

[13] M. E. Delamaro and J. C. Maldonado. Proteum—A tool for the assessment of test adequacy for C programs. In *Conference on Performability in Computing Systems (PCS 96)*, New Brunswick, NJ, July 1996.

[14] R. A. DeMillo, R. J. Lipton, and F. G. Sayward. Hints on test data selection: Help for the practicing programmer. *Computer*, 4(11):34–41, Apr. 1978.

[15] D. Jackson. Micromodels of software: Modelling and analysis with Alloy, 2001. `http://sdg.lcs.mit.edu/alloy/book.pdf`.

[16] D. Jackson and C. A. Damon. Elements of style: Analyzing a software design feature with a counterexample detector. *IEEE Transactions on Software Engineering*, 22(7), July 1996.

[17] D. Jackson, I. Schechter, and I. Shlyakhter. ALCOA: The Alloy constraint analyzer. In *Proc. 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, June 2000.

[18] D. Jackson and K. Sullivan. COM revisited: Tool-assisted modeling of an architectural framework. In *Proc. 8th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, San Diego, CA, 2000.

[19] S. Khurshid and D. Jackson. Exploring the design of an intentional naming scheme with an automatic constraint analyzer. In *Proc. 15th IEEE International Conference on Automated Software Engineering (ASE)*, Grenoble, France, Sep 2000.

[20] S.-W. Kim, J. Clark, and J. McDermid. Class mutation: Mutation testing for object oriented programs. In *FMES 2000*, Oct. 2000.

[21] K. N. King and A. J. Offutt. A Fortran language system for mutation-based software testing. *Software–Practice and Experience*, 21(7):685–718, 1991.

[22] G. T. Leavens, A. L. Baker, and C. Ruby. Preliminary design of JML: A behavioral interface specification language for Java. Technical Report TR 98-06i, Department of Computer Science, Iowa State University, June 1998. (last revision: Aug 2001).

[23] T. Lev-Ami and M. Sagiv. TVLA: A system for implementing static analyses. In *Proc. Static Analysis Symposium*, Santa Barbara, CA, June 2000.

[24] B. Liskov. *Program Development in Java: Abstraction, Specification,*

and Object-Oriented Design. Addison-Wesley, 2000.

[25] D. Marinov and S. Khurshid. TestEra: A novel framework for automated testing of Java programs. In *Proc. 16th IEEE International Conference on Automated Software Engineering (ASE)*, San Diego, CA, Nov. 2001.

[26] A. Moeller and M. I. Schwartzbach. The pointer assertion logic engine. In *Proc. SIGPLAN Conference on Programming Languages Design and Implementation*, Snowbird, UT, June 2001.

[27] J. Offutt and R. Untch. Mutation 2000: Uniting the orthogonal. In *Mutation 2000: Mutation Testing in the Twentieth and the Twenty First Centuries*, San Jose, CA, Oct. 2000.

[28] J. Offutt, J. Voas, and J. Payne. Mutation operators for Ada. Technical Report ISSE-TR-96-09, George Mason University, Fairfax, VA, Oct. 1996.

[29] N. J. A. Sloane, S. Plouffe, J. M. Borwein, and R. M. Corless. The encyclopedia of integer sequences. *SIAM Review*, 38(2), 1996. `http://www.research.att.com/~njas/sequences/Seis.html`.

[30] K. Stobie. Adavanced modeling, model based test generation, and Abstract state machine Language AsmL. `http://www.sasqag.org/pastmeetings/asml.ppt`, 2003.

[31] K. J. Sullivan, D. Coppit, and J. B. Dugan. The Galileo fault tree analysis tool. In *Proc. of the 29th International Symposium on Fault Tolerant Computing*, pages 232–235, June 1999.

[32] Sun Microsystems. *Java 2 Platform, Standard Edition, v1.3.1 API Specification*. `http://java.sun.com/j2se/1.3/docs/api/`.

[33] R. Untch, A. J. Offutt, and M. J. Harrold. Mutation testing using mutant schemata. In *Proc. International Symposium on Software Testing and Analysis (ISSTA)*, pages 139–148, 1993.

[34] M. Vaziri and D. Jackson. Checking properties of heap-manipulating procedures with a constraint solver. In *Proc. 9th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, Warsaw, Poland, Apr. 2003. (to appear).

APPENDIX

A. FULL CODE FOR THE EXAMPLE

```
import java.util.*;
class SearchTree {
    Node root; // root node
    int size;  // number of nodes in the tree
    static class Node {
        Node left;  // left child
        Node right; // right child
        Comparable info; // data
    }

    /*@ normal_behavior // non-exceptional specification
      @     // precondition
      @     requires repOk();
      @     // postcondition
      @     ensures repOk() && !contains(info) &&
      @             \result == \old(contains(info));
      @*/
    boolean remove(Comparable info) {
        Node parent = null;
        Node current = root;
        while (current != null) {
            int cmp = info.compareTo(current.info);
            if (cmp < 0) {
                parent = current;
                current = current.left;
            } else if (cmp > 0) {
                parent = current;
                current = current.right;
            } else {
                break;
            }
        }
        if (current == null) return false;
        Node change = removeNode(current);
        if (parent == null) {
            root = change;
        } else if (parent.left == current) {
            parent.left = change;
        } else {
            parent.right = change;
        }
        return true;
    }

    Node removeNode(Node current) {
        size--;
        Node left = current.left, right = current.right;
        if (left == null) return right;
        if (right == null) return left;
        if (left.right == null) {
            current.info = left.info;
            current.left = left.left;
            return current;
        }
        Node temp = left;
        while (temp.right.right != null) {
            temp = temp.right;
        }
        current.info = temp.right.info;
        temp.right = temp.right.left;
        return current;
    }

    boolean repOk() {
        // checks that empty tree has size zero
        if (root == null) return size == 0;
        // checks that the input is a tree
        if (!isAcyclic()) return false;
        // checks that size is consistent
        if (numNodes(root) != size) return false;
        // checks that data is ordered
        if (!isOrdered(root)) return false;
        return true;
    }

    private boolean isAcyclic() {
        Set visited = new HashSet();
        visited.add(root);
        LinkedList workList = new LinkedList();
        workList.add(root);
        while (!workList.isEmpty()) {
            Node current = (Node)workList.removeFirst();
            if (current.left != null) {
                // checks that the tree has no cycle
                if (!visited.add(current.left))
                    return false;
                workList.add(current.left);
            }
            if (current.right != null) {
                // checks that the tree has no cycle
                if (!visited.add(current.right))
                    return false;
                workList.add(current.right);
            }
        }
        return true;
    }

    private int numNodes(Node n) {
        if (n == null) return 0;
        return 1 + numNodes(n.left) + numNodes(n.right);
    }

    private boolean isOrdered(Node n) {
        return isOrdered(n, null, null);
    }

    private boolean isOrdered(Node n, Compara-
ble min, Comparable max) {
        if (n.info == null) return false;
        if ((min != null && n.info.compareTo(min) <= 0) ||
            (max != null && n.info.compareTo(max) >= 0))
            return false;
        if (n.left != null)
            if (!isOrdered(n.left, min, n.info))
                return false;
        if (n.right != null)
            if (!isOrdered(n.right, n.info, max))
                return false;
        return true;
    }
}
```

B. EXPERIMENTAL RESULTS

		generation					checking			
		gen.	ded.	spec. coverage		#	time	code coverage		mutants
benchmark	scope	[sec]	[sec]	st. [%]	br. [%]	inputs	[sec]	st. [%]	br. [%]	killed [%]
SearchTree	1	0.06	0.01	57.89	60.00	4	0.06	38.46	40.00	26.10
	2	0.05	0.01	94.74	96.67	20	0.06	79.49	87.50	69.85
	3	0.07	0.10	94.74	96.67	90	0.07	87.18	92.50	79.77
	4	0.17	0.10	94.74	96.67	408	0.14	97.44	97.50	92.64
	5	0.38	0.25	94.74	96.67	1880	0.24	100.00	100.00	98.52
	6	1.39	0.52	94.74	96.67	8772	0.46	100.00	100.00	99.26
	7	9.03	2.19	94.74	96.67	41300	1.25	100.00	100.00	99.26
DisjSet	1	0.01	0.01	61.54	55.00	4	0.04	23.08	25.00	0.41
	2	0.01	0.01	100.00	95.00	30	0.09	69.23	68.75	30.45
	3	0.04	0.04	100.00	100.00	456	0.09	100.00	100.00	88.47
	4	0.29	0.31	100.00	100.00	18280	0.43	100.00	100.00	95.06
	5	10.91	9.87	100.00	100.00	1246380	19.93	100.00	100.00	95.06
HeapArray	1	0.01	0.01	80.00	85.71	16	0.04	79.31	66.67	39.05
	2	0.01	0.01	90.00	92.86	75	0.05	79.31	66.67	43.79
	3	0.02	0.02	90.00	92.86	396	0.09	93.10	83.33	69.70
	4	0.08	0.09	90.00	92.86	2240	0.17	96.55	88.89	86.13
	5	0.22	0.21	90.00	92.86	15352	0.38	96.55	94.44	89.78
	6	0.90	0.71	90.00	92.86	118251	1.88	100.00	100.00	96.35
	7	7.09	6.21	90.00	92.86	1175620	17.58	100.00	100.00	96.71
BinomialHeap	1	0.02	0.01	62.79	62.00	12	0.07	52.87	57.58	31.16
	2	0.03	0.02	93.02	94.00	54	0.08	87.36	84.85	62.67
	3	0.12	0.09	93.02	94.00	336	0.14	98.85	96.97	89.72
	4	0.40	0.30	97.67	98.00	1800	0.24	100.00	98.48	93.15
	5	0.81	0.65	97.67	98.00	16848	0.69	100.00	100.00	94.86
	6	3.30	2.35	97.67	98.00	159642	4.61	100.00	100.00	95.89
	7	35.60	28.06	97.67	98.00	2577984	75.96	100.00	100.00	96.91
FibonacciHeap	1	0.01	0.07	55.55	51.72	12	0.07	35.48	43.55	15.82
	2	0.03	0.03	91.11	93.10	108	0.09	75.27	80.64	44.10
	3	0.28	0.24	97.78	98.28	1632	0.24	95.70	98.39	75.08
	4	1.22	0.90	97.78	98.28	34650	1.08	95.70	98.39	81.48
	5	14.14	12.94	97.78	98.28	941058	23.37	100.00	100.00	88.88
LinkedList	1	0.01	0.01	100.00	100.00	15	0.08	64.15	68.75	58.19
	2	0.01	0.01	100.00	100.00	50	0.09	90.57	84.38	98.77
	3	0.03	0.03	100.00	100.00	169	0.12	90.57	84.38	99.59
	4	0.07	0.07	100.00	100.00	627	0.16	90.57	84.38	99.59
	5	0.18	0.18	100.00	100.00	2584	0.26	90.57	84.38	99.59
	6	0.33	0.31	100.00	100.00	11741	0.48	90.57	84.38	99.59
	7	0.74	0.71	100.00	100.00	58175	1.54	90.57	84.38	99.59
SortedList	1	0.03	0.04	71.43	62.50	7	0.11	62.50	50.00	33.33
	2	0.04	0.07	100.00	100.00	36	0.11	80.00	74.14	52.81
	3	0.07	0.07	100.00	100.00	188	0.15	92.50	89.66	90.04
	4	0.22	0.20	100.00	100.00	1066	0.28	92.50	89.66	93.93
	5	0.53	0.48	100.00	100.00	7427	0.50	92.50	89.66	96.53
	6	1.94	1.77	100.00	100.00	73263	2.57	92.50	89.66	97.40
	7	22.68	21.13	100.00	100.00	1047608	37.91	92.50	89.66	97.40
TreeMap	1	0.02	0.02	57.14	63.33	6	0.06	14.41	14.89	5.46
	2	0.03	0.03	100.00	100.00	28	0.06	45.95	50.00	28.66
	3	0.07	0.04	100.00	100.00	96	0.09	63.96	73.40	61.09
	4	0.18	0.15	100.00	100.00	328	0.15	89.19	85.11	78.15
	5	0.38	0.31	100.00	100.00	1150	0.24	100.00	91.49	87.37
	6	0.94	0.61	100.00	100.00	3924	0.38	100.00	91.49	89.76
	7	3.28	1.75	100.00	100.00	12754	0.73	100.00	91.49	89.76
HashSet	1	0.01	0.01	57.89	69.23	4	0.04	51.92	50.00	29.91
	2	0.01	0.01	89.47	92.31	34	0.05	96.15	95.00	77.45
	3	0.06	0.05	89.47	92.31	212	0.09	100.00	100.00	90.57
	4	0.23	0.22	89.47	92.31	1170	0.19	100.00	100.00	90.98
	5	0.36	0.34	89.47	92.31	3638	0.27	100.00	100.00	91.39
	6	0.91	0.71	89.47	92.31	12932	0.62	100.00	100.00	91.80
	7	3.38	2.88	89.47	92.31	54844	1.55	100.00	100.00	92.21
AVTree	1	0.01	0.01	53.33	56.25	2	0.07	55.29	51.92	40.00
	2	0.05	0.03	90.00	87.50	86	0.14	75.29	78.85	60.00
	3	0.21	0.17	96.67	96.88	1702	0.78	88.23	84.61	75.12
	4	3.16	1.86	96.67	96.88	27734	8.36	94.12	92.31	91.21
	5	87.13	43.41	96.67	96.88	417878	134.51	94.12	92.31	93.65

Table 5: Korat's performance for test generation (with regular and dedicated generators), specification coverage (statement and branch), correctness checking, code coverage (statement and branch), and rate of mutant killing. All times are elapsed real times in seconds from the start of Korat to its completion. For all benchmarks and their sufficient scopes, Korat takes less than five minutes to generate all inputs and check correctness.

La fonction τ de Ramanujan.

François Brunault.

Exposé au séminaire des doctorants de théorie
des nombres de Chevaleret, le 18 mars 2003.

Nous supposerons connues les bases de la théorie des formes modulaires (définition d'une forme modulaire de poids $k \geq 4$ pair et de niveau 1). Le lecteur pourra se référer à [Z] qui est une très bonne introduction.

1 Définitions.

Pour tout entier $k \geq 4$ pair, on définit la *série d'Eisenstein de poids k* par :

$$G_k(z) = \frac{(k-1)!}{2(2\pi i)^k} \sum_{(m,n)\in\mathbf{Z}^2}{}' \frac{1}{(mz+n)^k}. \tag{1}$$

Le symbole $\sum'$ indique que l'on somme sur les $(m,n) \neq (0,0)$. Cette série converge absolument car l'exposant de $(mz+n)$ est > 2. Elle définit une fonction holomorphe sur le demi-plan de Poincaré $\mathcal{H} = \{z \in \mathbf{C},\ \Im(z) > 0\}$. Il n'est pas très difficile de vérifier que G_k est modulaire de poids k, c'est-à-dire

$$G_k\left(\frac{az+b}{cz+d}\right) = (cz+d)^k G_k(z) \qquad \left(\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbf{Z})\right). \tag{2}$$

La série d'Eisenstein G_k est une *forme modulaire de poids k*. Elle admet le développement de Fourier suivant, que l'on peut obtenir grâce à la formule de sommation de Poisson :

$$G_k(z) = -\frac{B_k}{2k} + \sum_{n=1}^{\infty} \sigma_{k-1}(n) e^{2i\pi nz}, \tag{3}$$

où B_k désigne le k-ième nombre de Bernoulli [S], et $\sigma_{k-1}(n)$ désigne la somme des puissances $(k-1)$-ièmes des diviseurs positifs de n.

Nous noterons $q = e^{2i\pi z}$, de telle sorte que G_k peut être vue comme une série entière en q, de rayon de convergence égal à 1. Notons également que le membre de droite de (3) a encore un sens pour $k = 2$, ce qui permet de définir G_2. En revanche, G_2 ne vérifie plus la condition de modularité (2). Pour les premières valeurs de k, on calcule facilement les développements suivants :

$$\begin{array}{rcccccccccccccc} G_2 & = & -\frac{1}{24} & + & q & + & 3q^2 & + & 4q^3 & + & 7q^4 & + & 6q^5 & + & 12q^6 & + & \dots \\ G_4 & = & \frac{1}{240} & + & q & + & 9q^2 & + & 28q^3 & + & 73q^4 & + & 126q^5 & + & 252q^6 & + & \dots \\ G_6 & = & -\frac{1}{504} & + & q & + & 33q^2 & + & 244q^3 & + & 1057q^4 & + & 3126q^5 & + & 8052q^6 & + & \dots \end{array}$$

Ces développements renferment de nombreuses propriétés arithmétiques. Par exemple, le coefficient de q^6 est toujours égal au produit du coefficient de q^2 par le coefficient de q^3 (c'est une conséquence de la multiplicativité de la fonction σ_{k-1}). D'autres propriétés existent, on peut par exemple chercher la relation entre le coefficient de q^2 et celui de q^4.

Remarque 1. *Le membre de droite de (3) a encore un sens et est non trivial pour k impair. On ne peut en revanche pas l'écrire sous la forme (1) car, pour des raisons de parité, cette dernière série s'annule identiquement pour $k \geq 3$ impair. Il serait donc intéressant d'interpréter autrement le membre de droite de (3) lorsque k est impair.*

Nous allons maintenant définir la fonction τ de Ramanujan.

Définition 2. *Soit Δ la série entière en q suivante*

$$\Delta = 8000G_4^3 - 147G_6^2. \tag{4}$$

Pour tout entier $n \geq 1$, on note $\tau(n)$ le coefficient de q^n dans Δ. On a donc par définition

$$\Delta = \sum_{n=1}^{\infty} \tau(n) q^n. \tag{5}$$

La fonction τ sur $\mathbf{N}^*$ ainsi obtenue est appelée fonction τ de Ramanujan.

Le calcul des premiers termes donne

$$\Delta = q - 24q^2 + 252q^3 - 1472q^4 + 4830q^5 - 6048q^6 + \dots. \tag{6}$$

Proposition 3. *La série entière Δ, vue comme fonction holomorphe sur $\mathcal{H}$, est une forme modulaire de poids 12.*

Démonstration. On sait que G_4 est de poids 4, et que G_6 est de poids 6. En conséquence G_4^3 et G_6^2 sont modulaires de poids respectifs $4 \times 3 = 12$ et $6 \times 2 = 12$. Il en résulte que Δ est également une forme modulaire de poids 12. □

Notons que par choix des coefficients devant G_4^3 et G_6^2, le terme constant du développement de Fourier de Δ vaut 0. On dit que Δ est une *forme parabolique* de poids 12. Cela signifie que

$$\lim_{\Im(z) \to +\infty} \Delta(z) = 0.$$

On peut même voir que $\Delta(z)$ décroît exponentiellement vite en $\Im(z)$, lorsque $\Im(z) \to +\infty$.

Avant d'entamer l'étude de la fonction τ, signalons que $\tau(n)$ a été calculé par Ramanujan pour $1 \leq n \leq 30$, puis par Lehmer pour $1 \leq n \leq 300$. Le calcul efficace de la fonction τ est l'objet de recherches actuelles [C].

2 Une congruence de Ramanujan.

Nous commençons par la proposition suivante.

Proposition 4. *La fonction τ est à valeurs entières : pour tout $n \geq 1$, on a $\tau(n) \in \mathbf{Z}$.*

Démonstration. Il est clair a priori que la fonction τ est à valeurs rationnelles. La difficulté vient du fait que les termes constants de G_4 et G_6 ne sont pas entiers.

Posons

$$G_4 = \frac{1}{240} + H_4 \qquad \text{et} \qquad G_6 = -\frac{1}{504} + H_6.$$

On a alors

$$\begin{aligned}\Delta &= 8000G_4^3 - 147G_6^2 \\ &= 8000\left(\frac{1}{240} + H_4\right)^3 - 147\left(-\frac{1}{504} + H_6\right)^2 \\ &= 8000H_4^3 - 147H_6^2 + 100H_4^2 + \frac{5H_4 + 7H_6}{12}.\end{aligned}$$

Il suffit donc de montrer que $\frac{5H_4+7H_6}{12}$ est à coefficients entiers. Or, par définition de G_4 et G_6, le n-ième coefficient de cette série entière vaut $\frac{5\sigma_3(n)+7\sigma_5(n)}{12}$.

Il s'agit donc de montrer que 12 divise $5\sigma_3(n) + 7\sigma_5(n)$, pour tout $n \geq 1$. Or

$$\begin{aligned}5\sigma_3(n) + 7\sigma_5(n) &= \sum_{d|n} 5d^3 + 7d^5 \\ &\equiv \sum_{d|n} 7d^5 - 7d^3 \qquad \pmod{12} \\ &\equiv 7\sum_{d|n} d^3(d+1)(d-1) \pmod{12} \\ &\equiv 0 \qquad \pmod{12}\end{aligned}$$

car $d^3(d+1)(d-1)$ est divisible par 12 pour tout $d \in \mathbf{Z}$ (en effet il l'est par 4, et par 3). □

Proposition 5. *On a la congruence (dite de Ramanujan)*

$$\tau(n) \equiv \sigma_{11}(n) \pmod{691} \qquad (n \geq 1). \tag{7}$$

Démonstration. Nous admettrons que l'espace M_{12} des formes modulaires de poids 12 est un espace vectoriel complexe de dimension 2 (voir [Z] pour une démonstration). En conséquence le sous-espace S_{12} des formes paraboliques est de dimension 1, et il est engendré par Δ. On a

$$G_{12} = \frac{691}{65520} + \underbrace{\cdots}_{\in \mathbf{Z}[[q]]} \in M_{12},$$
$$G_6^2 = \frac{1}{504^2} + \underbrace{\cdots}_{\in \frac{1}{504}\mathbf{Z}[[q]]} \in M_{12}.$$

Nous en déduisons

$$\underbrace{65520 G_{12}}_{\in \mathbf{Z}[[q]]} - 691 \times \underbrace{504^2 G_6^2}_{\in \mathbf{Z}[[q]]} \in S_{12} \cap \mathbf{Z}[[q]].$$

Il existe donc α complexe tel que $65520 G_{12} - 691 \times 504^2 G_6^2 = \alpha\Delta \in S_{12} \cap \mathbf{Z}[[q]]$. Puisque $\tau(1) = 1$, on a nécessairement $\alpha \in \mathbf{Z}$. En identifiant les n-ièmes coefficients des séries entières on obtient

$$65520\sigma_{11}(n) \equiv \alpha\tau(n) \pmod{691} \qquad (n \geq 1).$$

En faisant $n = 1$ on obtient $\alpha \equiv 65520 \equiv 566 \pmod{691}$, en particulier α est inversible modulo 691 (qui est premier). En simplifiant l'équation ci-dessus par α, on obtient $\sigma_{11}(n) \equiv \tau(n) \pmod{691}$, ce qui est la congruence recherchée. □

Il existe beaucoup d'autres congruences vérifiées par les nombres $\tau(n)$. Voici quelques exemples

$$\tau(n) \equiv n\sigma_3(n) \pmod{7} \qquad (n \geq 1) \tag{8}$$
$$\tau(n) \equiv n^2\sigma_7(n) \pmod{27} \qquad (n \geq 1) \tag{9}$$

Pour plus de détails sur les congruences vérifiées par la fonction τ, ainsi que le lien avec les représentations l-adiques, on pourra se reporter à l'exposé de Serre [S2], qui est par ailleurs un très bon exposé (c'est un pléonasme) sur la fonction τ de Ramanujan.

3 Une interprétation elliptique de Δ.

Théorème 6. *(Jacobi) On a l'identité de séries formelles suivante*

$$\Delta = q\prod_{n=1}^{\infty}(1-q^n)^2 4. \tag{10}$$

La démonstration de ce résultat peut être trouvée dans [Z] ou dans [S]. Un corollaire de ce théorème est que Δ ne s'annule pas sur $\mathcal{H}$.

La forme modulaire Δ est intimement liée aux courbes elliptiques. En effet, pour $z \in \mathcal{H}$, notons E_z la surface de Riemann compacte définie par

$$E_z = \frac{\mathbf{C}}{\mathbf{Z} + z\mathbf{Z}}.$$

On sait que E_z est isomorphe à la courbe elliptique sur $\mathbf{C}$ définie par l'équation

$$E_z : y^2 = 4x^3 - g_2(z)x - g_3(z)$$

où l'on a posé $g_2(z) = 20 \cdot (2\pi)^4 G_4(z)$ et $g_3(z) = -\frac{7}{3}(2\pi)^6 G_6(z)$ (attention au changement d'indice, nous avons adopté ici les notations standard).

Proposition 7. *La valeur de la forme modulaire Δ en z est égale, à un facteur près, au discriminant de la courbe elliptique E_z :*

$$\Delta(E_z) := g_2(z)^3 - 27g_3(z)^2 = (2\pi)^{12}\Delta(z) \qquad (z \in \mathcal{H}).$$

Le discriminant d'une courbe elliptique sur un corps K n'est défini qu'à un élément de $(K^*)^{12}$ près. Ici $K = \mathbf{C}$, donc $(K^*)^{12} = \mathbf{C}^*$. Cela explique le terme 'à un facteur près' dans la proposition précédente.

4 Propriétés arithmétiques de la fonction Δ.

Le développement de Fourier (6) de Δ, que nous récrivons ici :

$$\Delta = q - 24q^2 + 252q^3 - 1472q^4 + 4830q^5 - 6048q^6 + \dots$$

a des propriétés arithmétiques très intéressantes. En guise d'exercice (et sans lire la suite !), on peut chercher la relation entre les coefficients de q^2, q^3 et q^6, ou encore celle entre les coefficients de q^2 et q^4.

Ramanujan a le premier observé, et conjecturé en 1916, que les coefficients $\tau(n)$ sont multiplicatifs, i.e. satisfont

$$\tau(mn) = \tau(m)\tau(n) \qquad (m \text{ et } n \geq 1 \text{ premiers entre eux}). \tag{11}$$

On le vérifie ici pour $m = 2$ et $n = 3$. Cette conjecture a été démontrée un an plus tard par Mordell. Pour donner une idée de la démonstration de Mordell, nous sommes amenés à introduire les formes modulaires de Hecke.

Définition 8. *Soit $f = \sum_{n=0}^{\infty} a_n q^n \in M_k$ une forme modulaire de poids $k \geq 4$ pair. On dit que f est une* forme de Hecke *lorsque $f \neq 0$ et*

$$a_{mn} = a_m a_n \qquad (m \text{ et } n \geq 1 \text{ premiers entre eux}). \tag{12}$$

Sous cette hypothèse on a toujours $a_1 = 1$. On dit que les formes de Hecke sont *normalisées*. Les séries d'Eisenstein G_k ($k \geq 4$ pair) sont des exemples de formes de Hecke. Un théorème célèbrede Hecke affirme que les formes de Hecke de M_k (resp. S_k) forment une base de M_k (resp. S_k). La conjecture de Ramanujan découle immédiatement de ce théorème : l'espace S_{12} auquel appartient Δ est de dimension 1, et l'on a $\tau(1) = 1$, par conséquent Δ est une forme de

Hecke. En réalité, il n'est pas nécessaire d'utiliser le théorème de Hecke dans toute sa force pour démontrer la conjecture de Ramanujan. On peut se débrouiller en introduisant les opérateurs de Hecke (ce qu'a fait Mordell). On montre alors également la relation de récurrence suivante

$$\tau(p^{n+2}) = \tau(p)\tau(p^{n+1}) - p^{11}\tau(p^n) \qquad (p \text{ premier}, n \geq 0). \tag{13}$$

Cette relation permet de ramener le calcul des $\tau(n)$ $(n \geq 1)$ à celui des $\tau(p)$, p premier.

5 Ordre de grandeur de la fonction τ.

Intéressons-nous maintenant à l'ordre de grandeur de $\tau(n)$. Commençons par l'ordre de grandeur des coefficients de Fourier des séries d'Eisenstein.

Proposition 9. *Soit k un entier pair ≥ 2. On a l'estimation suivante pour le n-ième coefficient de Fourier de G_k, lorsque n tend vers l'infini :*

$$a_n(G_k) = \sigma_{k-1}(n) = \begin{cases} O(n^{k-1}) & \text{si } k \geq 4, \\ O(n^{1+\epsilon}) & \text{si } k = 2 \qquad (\epsilon > 0). \end{cases} \tag{14}$$

Il n'est pas difficile de voir que ces estimations sont les meilleures possibles, du point de vue de l'exposant de n. À l'aide de la définition (4) de Δ et de cette proposition, on peut montrer à la main que

$$\tau(n) = O(n^{11}).$$

Il existe en fait un résultat plus général.

Théorème 10. *Soient k un entier pair ≥ 4 et $f = \sum_{n=0}^{\infty} a_n q^n \in M_k$ une forme modulaire de poids k. Alors on a l'estimation, lorsque n tend vers l'infini :*

$$a_n = O(n^{k-1}) \tag{15}$$

et

$$a_n = O(n^{\frac{k}{2}}) \qquad \text{si } f \in S_k. \tag{16}$$

En particulier, $\tau(n) = O(n^6)$.

On pourra trouver une démonstration dans [Z].

On peut encore améliorer l'exposant lorsque $f \in S_k$, mais cela demande beaucoup plus de travail !

Théorème 11. *(Deligne). Soit $f = \sum_{n=1}^{\infty} a_n q^n \in S_k$ une forme de Hecke de poids k pair ≥ 4. Alors*

$$|a_p| \leq 2p^{\frac{k-1}{2}} \qquad (p \text{ premier}) \tag{17}$$

ou de façon équivalente

$$|a_n| \leq \sigma_0(n) n^{\frac{k-1}{2}} \qquad (n \geq 1). \tag{18}$$

Ici, $\sigma_0(n)$ est le nombre de diviseurs > 0 de n. En particulier, lorsque n tend vers l'infini :

$$\tau(n) = O(n^{\frac{11}{2}+\epsilon}) \qquad (\epsilon > 0). \tag{19}$$

Ce résultat a été conjecturé par Ramanujan dans le cas de Δ, et par Petersson dans le cas général. En 1969, Deligne a montré que ce résultat était une conséquence des conjectures de Weil portant sur les variétés algébriques sur les corps finis. Il a ensuite démontré les conjectures de Weil, en 1974.

Signalons une autre conséquence du théorème de Deligne : soit $f = \sum_{n=1}^{\infty} a_n q^n \in S_k$ une forme parabolique quelconque. Définissons la fonction L de f par

$$L(f,s) := \sum_{n=1}^{\infty} \frac{a_n}{n^s} \qquad (s \in \mathbf{C}). \tag{20}$$

Alors $L(f,s)$ converge pour $\Re(s) > \frac{k+1}{2}$. On peut démontrer de manière élémentaire que la fonction $L(f,s)$ se prolonge en une fonction entière sur $\mathbf{C}$ (ceci n'utilise pas le théorème de Deligne).

Notons que le problème de l'estimation des coefficients de Fourier des formes modulaires (ou plus généralement des formes automorphes) est un des problèmes majeurs de la théorie des nombres.

6 Une conjecture pour finir.

Terminons ce petit tour d'horizon de la fonction τ par la conjecture de Lehmer.

Conjecture 12. *(Lehmer) Pour tout entier $n \geq 1$, on a $\tau(n) \neq 0$.*

Par la propriété de multiplicativité (12), on se ramène au cas où n est une puissance d'un nombre premier. En utilisant la relation de récurrence (13), il me semble (mais je ne l'ai pas rédigé) que l'on peut se ramener au cas où n est un nombre premier p.

Conjecture 13. *Pour tout nombre premier p, on a $\tau(p) \neq 0$.*

À l'heure actuelle, la conjecture de Lehmer est connue pour $n \leq 22689242781695999$ [JK]. Un problème lié à la conjecture de Lehmer est le suivant :

Problème ouvert 1. *Pour quels nombres premiers p a-t-on $\tau(p) \equiv 0 \pmod{p}$?*

À l'aide d'un ordinateur, on trouve que les premières valeurs de p satisfaisant $\tau(p) \equiv 0 \pmod{p}$ sont $p = 2$, 3, 5, 7 et 2411.

La condition $\tau(p) \equiv 0 \pmod{p}$ se traduit conjecturalement en terme de la représentation l-adique associée à τ (voir [S2]). Plus généralement, il est intéressant d'étudier les propriétés de τ d'un point de vue géométrique, c'est-à-dire en étudiant les propriétés du *motif* associé.

Pour de plus amples renseignements sur la fonction τ de Ramanujan, on pourra se reporter à la page web [Sl], qui contient de nombreuses références. Attention cependant, car j'ai trouvé un lien vers une page qui démontre tout bonnement la conjecture de Lehmer !

Références

[C] Charles, C. D., Computing the Ramanujan Tau Function. http ://www.cs.wisc.edu/~cdx/CompTau.pdf

[JK] Jordan, B., Kelly, B., The vanishing of the Ramanujan Tau function, Preprint, 1999.

[S] Serre, J.-P., Cours d'arithmétique. Presses Universitaires de France (1970).

[S2] Serre, J.-P., Une interprétation des congruences relatives à la fonction τ de Ramanujan, Séminaire Delange-Pisot-Poitou 9 (1967/68), Théorie des Nombres, exposé 14 (1969). Traduction anglaise http ://public.csusm.edu/public/FranzL/publ/serre.pdf

[Sl] Sloane, N. J. A. Suite A000594. In *The On-Line Encyclopedia of Integer Sequences*. http ://www.research.att.com/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi ?Anum=000594

[Z] Zagier, D. Introduction to Modular Forms. In *From Number Theory To Physics*, Waldschmidt, Moussa, Luck, Itzykson. Springer (1992), pp. 238-291.

ANALYTIC COMBINATORICS

—

SYMBOLIC COMBINATORICS

PHILIPPE FLAJOLET & ROBERT SEDGEWICK

Algorithms Project
INRIA Rocquencourt
78153 Le Chesnay
France

Department of Computer Science
Princeton University
Princeton, NJ 08540
USA

First Edition, May 25, 2002

♡ ♡ ♡ ♡ ♡ ♡

ABSTRACT

This booklet develops in nearly 200 pages the basics of combinatorial enumeration through an approach that revolves around generating functions. The major objects of interest here are words, trees, graphs, and permutations, which surface recurrently in all areas of discrete mathematics. The text presents the core of the theory with chapters on unlabelled enumeration and ordinary generating functions, labelled enumeration and exponential generating functions, and finally multivariate enumeration and generating functions. It is largely oriented towards applications of combinatorial enumeration to random discrete structures and discrete mathematics models, as they appear in various branches of science, like statistical physics, computational biology, probability theory, and, last not least, computer science and the analysis of algorithms.

Acknowledgements. This work was supported in part by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT). The authors are grateful to Xavier Gourdon who incited us to add a separate chapter on multivariate generating functions and to Brigitte Vallée for many critical suggestions regarding the presentation and global organization of this text. This booklet would be substantially different (and much less informative) without Neil Sloane's *Encyclopedia of Integer Sequences*, Steve Finch's *Mathematical Constants*, both available on the internet. Bruno Salvy and Paul Zimmermann have developed algorithms and libraries for combinatorial structures and generating functions that are based on the MAPLE system for symbolic computations and have proven to be immensely useful.

"*Symbolic Combinatorics*" is a set of lecture notes that are a component of a wider book project titled *Analytic Combinatorics*, which will provide a unified treatment of analytic methods in combinatorics. This text is partly based on an earlier document titled "The Average Case Analysis of Algorithms: Counting and Generating Functions", INRIA Res. Rep. #1888 (1993), 116 pages, which it now subsumes.

FOREWORD

Analytic Combinatorics aims at predicting precisely the asymptotic properties of structured combinatorial configurations, through an approach that bases itself extensively on analytic methods. Generating functions are the central objects of the theory.

Analytic combinatorics starts from an exact enumerative description of combinatorial structures by means of generating functions, which make their first appearance as purely formal algebraic objects. Next, generating functions are interpreted as analytic objects, that is, as mappings of the complex plane into itself. In this context, singularities play a key rôle in extracting the functions' coefficients in asymptotic form and extremely precise estimates result for counting sequences. This chain is applicable to a large number of problems of discrete mathematics relative to words, trees, permutations, graphs, and so on. A suitable adaptation of the theory finally opens the way to the analysis of parameters of large random structures.

Analytic combinatorics can accordingly be organized based on three components:

— *Symbolic Combinatorics* develops systematic "symbolic" relations between some of the major constructions of discrete mathematics and operations on generating functions which exactly encode counting sequences.

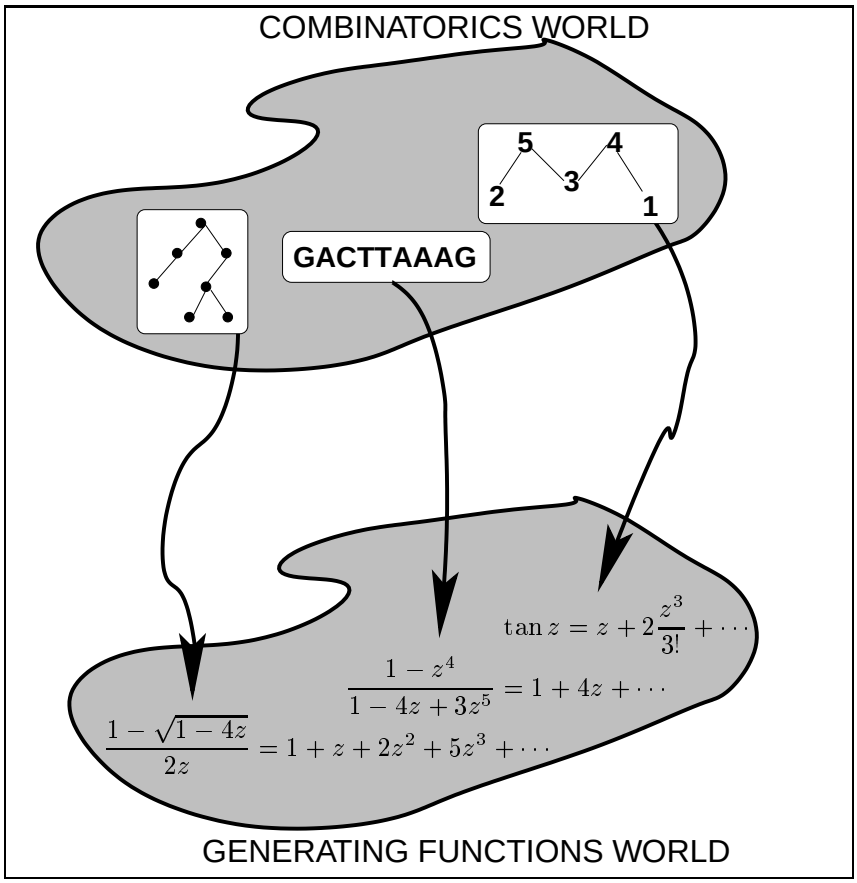
— *Singular combinatorics* elaborates a collection of methods by which one can extract asymptotic counting informations from generating functions, once these are viewed as analytic (holomorphic) functions over the complex domain. Singularities then appear to be a key determinant of asymptotic behaviour.

— *Random Combinatorics* concerns itself with probabilistic properties of large random structures—which properties hold with "high" probability, which laws govern randomness in large objects? In the context of analytic combinatorics, this corresponds to a deformation (adding auxiliary variables) and a perturbation (examining the effect of small variations of such auxiliary variables) of the standard enumerative theory.

The approach to quantitative problems of discrete mathematics provided by analytic combinatorics can be viewed as an *operational calculus* for combinatorics. The booklets, of which this is the first installment, expose this view by means of a very large number of examples concerning classical combinatorial structures (like words, trees, permutations, and graphs). What is aimed at eventually is an effective way of quantifying "metric" properties of large random structures. Accordingly, the theory is susceptible to many applications, within combinatorics itself, but, perhaps more importantly, within other areas of science where discrete probabilistic models recurrently surface, like statistical physics, computational biology, or electrical engineering. Last but not least, the analysis of algorithms and data structures in computer science has served and still serves as an important motivation in the development of the theory.

This booklet specifically exposes *Symbolic Combinatorics*, which is a unified algebraic theory dedicated to the setting up of functional relations between counting generating functions. As it turns out, a collection of general (and simple) theorems provide a

systematic translation mechanism between combinatorial constructions and operations on generating functions. (This translation process is a purely formal one, hence the name of "symbolic combinatorics" that we have adopted to characterize it.) Precisely, as regards basic counting, two parallel frameworks coexist—one for unlabelled structures and ordinary generating functions, the other for labelled structures and exponential generating functions. Furthermore, within the theory, parameters of combinatorial configurations can be easily taken into account by adding supplementary variables. Three chapters then compose this booklet: Chapter I deals with unlabelled objects; Chapter II develops in a parallel way labelled objects; Chapter III treats multivariate aspects of the theory suitable for the analysis of parameters of combinatorial structures.



Contents

Leonhard Euler (born, 15 April 1707 in Basel, Switzerland; died, 18 Sept 1783 in St Petersburg, Russia) was the first to relate classical analysis and combinatorics in a publication of 1753. Euler showed how to enumerate the triangulations of an n-gon: first, he modelled the combinatorial counting problem by a recurrence, then introduced the corresponding generating function; finally he solved the resulting equation and expanded the generating function using classical analysis, thereby providing a closed-form solution to the original counting problem and discovering the "Catalan numbers".

(Pictures are from *The MacTutor History of Mathematics* archive hosted by the University of St Andrews.)

CHAPTER I

Combinatorial Structures and Ordinary Generating Functions

Laplace discovered the remarkable correspondence between set theoretic operations and operations on formal power series and put it to great use to solve a variety of combinatorial problems.
— GIAN–CARLO ROTA [**122**]

This chapter and the next are devoted to enumeration, where the question is to determine the number of combinatorial configurations described by finite rules, and do so for all possible sizes. For instance, how many different permutations are there of size 17? of size n, for general n? what if some constraints are imposed, e.g., no four elements of increasing order in a row? The counting sequences are exactly encoded by *generating functions*, and, as we shall see, *generating functions are the central mathematical object* of combinatorial analysis. We examine here a framework that, contrary to more traditional treatments based on recurrences, explains the surprising efficiency of generating functions in the solution of combinatorial enumeration problems.

This chapter serves to introduce the *symbolic* approach to combinatorial enumerations. The principle is that many general set–theoretic constructions admit a direct translation as operations over generating functions. This is made concrete by means of a "dictionary" based on a core of important constructions, which includes the operations of union, cartesian product, sequence, set, multiset, and cycle. (Supplementary operations like pointing and substitution can be also be similarly treated.)

In this way, a language describing elementary combinatorial classes is set up. The problem of enumerating a class of combinatorial structures then simply reduces to finding a proper *specification*, a sort of formal "grammar", for the class in terms of the basic constructions. The translation into generating functions then becomes a purely mechanical "symbolic" process.

We show here how to describe in such a context integer partitions and compositions, as well as several elementary string and tree enumeration problems. A parallel approach, developed in Chapter II, applies to labelled objects and exponential generating functions, and in contrast the plain structures considered in this chapter are called *unlabelled*. The methodology is susceptible to multivariate extensions with which many characteristic parameters of combinatorial objects can also be analysed in a unified manner: this is to be examined in Chapter III. It also has the great merit of connecting nicely with complex asymptotic methods that exploit analyticity properties and singularities, to the effect that very precise asymptotic estimates are usually available whenever the symbolic method applies—a systematic treatment forms the basis of the next booklet in the series, *Analytic Combinatorics, Singular Combinatorics* (Chapters IV–VI).

I. 1. Symbolic enumeration methods

First and foremost, combinatorics deals with *discrete objects*, that is, objects that can be finitely described by construction rules. Examples are words, trees, graphs, geometric configurations, permutations, allocations, functions from a finite set into itself, and so on. A major question is to *enumerate* such objects according to some characteristic parameter(s).

DEFINITION I.1. *A* combinatorial class, *or simply a* class, *is a finite or denumerable set on which a* size *function is defined, satisfying the following conditions: the size of an element is a nonnegative integer; the number of elements of any given size is finite.*

If $\mathcal{A}$ is a class, the size of an element $\alpha \in \mathcal{A}$ is denoted by $|\alpha|$, or $|\alpha|_{\mathcal{A}}$ in the few cases where the underlying class needs to be made explicit. Given a class $\mathcal{A}$, we consistently let $\mathcal{A}_n$ be the set of objects in $\mathcal{A}$ that have size n and use the same group of letters for the counts $A_n = \mathrm{card}(\mathcal{A}_n)$ (alternatively, also $a_n = \mathrm{card}(\mathcal{A}_n)$). An axiomatic presentation is then as follows: a combinatorial class is a pair $(\mathcal{A}, |\cdot|)$ where $\mathcal{A}$ is at most denumerable and the mapping $|\cdot| \in (\mathcal{A} \mapsto \mathbb{N})$ is such that the inverse image of any integer is finite.

DEFINITION I.2. *The* counting sequence *of a combinatorial class $\mathcal{A}$ is the sequence of integers $\{A_n\}_{n\geq 0}$ where $A_n = \mathrm{card}(\mathcal{A}_n)$ is the number of objects in class $\mathcal{A}$ that have size n.*

Consider for instance the set $\mathcal{W}$ of binary words, which are words over a binary alphabet,

$$\mathcal{W} := \{\ldots 00, 01, 10, 11, 000, 001, 010, \ldots, 1001101, \ldots\},$$

if the binary alphabet is $\mathcal{A} = \{0,1\}$. The set $\mathcal{P}$ of permutations is

$$\mathcal{P} = \{\ldots 12, 21, 123, 132, 213, 231, 312, 321, 1234, \ldots, 532614, \ldots\},$$

since a permutation of $I_n := [1\,..\,n]$ is a bijective mapping that is representable by an an array $\left(\begin{smallmatrix} 1 & 2 & \cdots & n \\ \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{smallmatrix}\right)$ or equivalently by the sequence $\sigma_1\sigma_2\cdots\sigma_n$ of distinct elements from $\mathcal{I}_n$; The set $\mathcal{T}$ of triangulations is comprised of triangulations of convex polygonal domains which are decompositions into non-overlapping triangles. (For the purpose of the present discussion, the reader may content herself with what is suggested by Figure 1; the formal specification of triangulations appears on p. 18.) The sets $\mathcal{W}$, $\mathcal{P}$, and $\mathcal{T}$ constitute combinatorial classes, with the convention that the size of a word is its length, the size of a permutation is the number of its elements, the size of a triangulation is the number of triangles it comprises. The corresponding counting sequences are then given by

$$W_n = 2^n, \qquad P_n = n!, \qquad T_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!\,n!}, \tag{1}$$

where the initial values are

(2)

n	0	1	2	3	4	5	6	7	8	9	10
W_n	1	2	4	8	16	32	64	128	256	512	1024
P_n	1	1	2	6	24	120	720	5040	40320	362880	3628800
T_n	1	1	2	5	14	42	132	429	1430	4862	16796

Indeed elementary counting principles, namely, for finite sets $\mathcal{B}$ and $\mathcal{C}$

$$\begin{cases} \mathrm{card}(\mathcal{B}\cup\mathcal{C}) &= \mathrm{card}(B) + \mathrm{card}(C) \qquad (\text{provided } \mathcal{B}\cap\mathcal{C} = \emptyset) \\ \mathrm{card}(\mathcal{B}\times\mathcal{C}) &= \mathrm{card}(B)\cdot\mathrm{card}(Y), \end{cases} \tag{3}$$

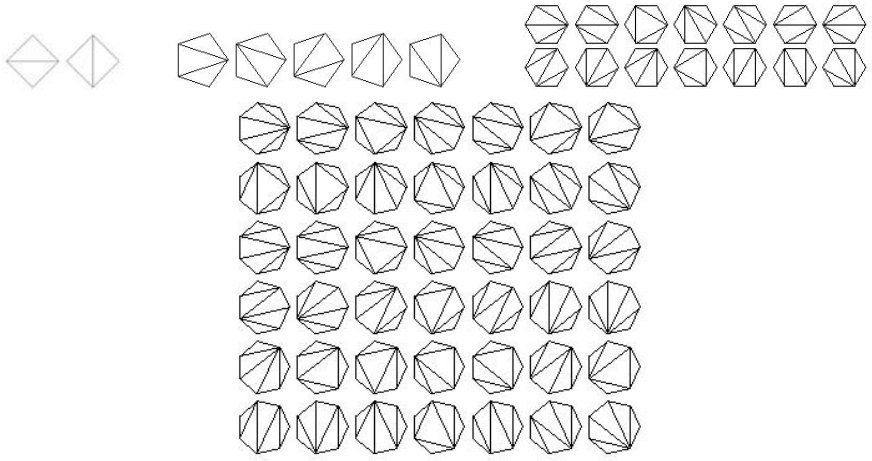


FIGURE 1. The class $\mathcal{T}$ of all triangulations of regular polygons (with size defined as the number of triangles) is a combinatorial class. The counting sequence starts as $T_0 = 1,\ T_1 = 1,\ T_2 = 2,\ T_3 = 5,\ T_4 = 14,\ T_5 = 42,\ T_6 = 132,\ \ldots$. Euler determined the OGF $T(z) = \sum_n T_n z^n$ as

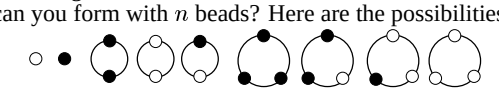
$$T(z) = \frac{1 - \sqrt{1-4z}}{2z},$$

from which there results that $T_n = \frac{1}{n+1}\binom{2n}{n}$. These numbers are known as the *Catalan numbers* (p. 17).

lead directly to expressions for words (W_n) and permutations (P_n). The sequence $W_n = 2^n$ has a well-known legend associated with the invention of the chess game: the inventor was promised by his king one grain of rice for the first square of the chessboard, two for the second, four for the third, and so on; the king naturally could not deliver... As to the number of permutations, it has been known for more than 1500 years and Knuth [**86**, p. 23] refers to the Hebrew *Book of Creation* (c. A.D.. 400), and to the *Anuyogadv ārasutra* (India, c. A.D. 500) for the explicit formula $n! = 1 \cdot 2 \cdots n$. Following Euler (1707–1783), the counting of triangulations (T_n) is best approached by generating functions: the modified binomial coefficients so obtained are known as Catalan numbers (see the discussion p. 17) and are central in combinatorial analysis (Section I. 5.3).

▷ **1.** *Permutations and factorials.* For a permutation in $\mathcal{P}_n$ written as a sequence of distinct numbers, there are n places where one can accommodate n, $n-1$ remaining places for $n-1$, and so on. Therefore, by (3), the number of permutations is $n \cdot (n-1) \cdots = n!$. ◁

▷ **2.** *Necklaces.* You are given tons of beads of two colours, ∘ and •. How many different types of necklace designs can you form with n beads? Here are the possibilities for $n = 1, 2, 3$:



This can be reformulated as the problem of finding the counting sequence of the class of necklaces defined formally as all the possible circular arrangements of two letters. The counting sequence starts as $2, 3, 4, 6, 8, 14, 20, 36, 60, 108, 188, 352$. The solution appears later in this chapter, p. 40. ◁

Two combinatorial classes $\mathcal{A}$ and $\mathcal{B}$ are said to be *isomorphic*, which is written $\mathcal{A} \cong \mathcal{B}$, iff their counting sequences are identical. This is equivalent to saying that there exists a bijection from $\mathcal{A}$ to $\mathcal{B}$ that preserves size, and one also says that $\mathcal{A}$ and $\mathcal{B}$ are *bijectively*

equivalent. Since we are only interested in counting problems, it proves often convenient to identify isomorphic classes and plainly consider them as identical. We then confine the notation $\mathcal{A} \cong \mathcal{B}$ (instead of $\mathcal{A} = \mathcal{B}$) to the few cases where combinatorial isomorphism rather than plain identity needs to be emphasized.

DEFINITION I.3. *The* ordinary generating function *(OGF) of a sequence* $\{A_n\}$ *is the formal power series*

$$A(z) = \sum_{n=0}^{\infty} A_n z^n. \tag{4}$$

The ordinary generating function *(OGF) of a combinatorial class* $\mathcal{A}$ *is the generating function of the numbers* $A_n = \text{card}(\mathcal{A}_n)$. *Equivalently, the OGF of class* $\mathcal{A}$ *is*

$$A(z) = \sum_{n \geq 0} A_n z^n = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|}. \tag{5}$$

It is also said that the variable z marks *size in the generating function.*

We adhere to a systematic *naming convention*: classes, their counting sequences, and their generating functions are systematically denoted by the same groups of letters: for instance, $\mathcal{A}$ for a class, $\{A_n\}$ (or $\{a_n\}$) for the counting sequence, and $A(z)$ (or $a(z)$) for its OGF. Also, we let generally $[z^n]f(z)$ denote the operation of extracting the coefficient of z^n in the formal power series $f(z) = \sum f_n z^n$, so that

$$[z^n] \left(\sum_{n \geq 0} f_n z^n \right) = f_n. \tag{6}$$

(The coefficient extractor notation reads as "coefficient of z^n in $f(z)$".)

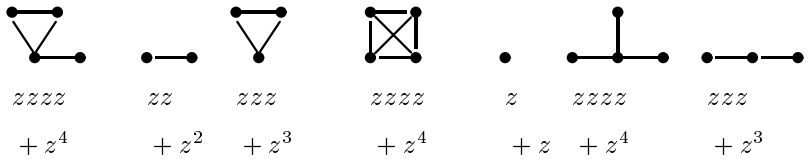
The OGF's corresponding to Eq. (1) are then

$$W(z) = \frac{1}{1-2z}, \qquad P(z) = \sum_{n=0}^{\infty} n!\, z^n, \qquad T(z) = \frac{1-2z-\sqrt{1-4z}}{2z}.$$

The OGF's $W(z)$ and $T(z)$ can be interpreted as standard analytic objects, upon assigning to the formal variable z values in the complex domain $\mathcal{C}$. In effect, the series $W(z)$ and $T(z)$ converge in a neighbourhood of 0 and represent complex functions analytic at the origin, while the OGF $P(z)$ is a purely formal power series (its radius of convergence is 0) that can nonetheless be subjected to the usual algebraic operations of power series; see APPENDIX: *Formal power series*, p. 169. (Permutation enumeration is most conveniently approached by exponential generating functions developed in Chapter II.)

The second "combinatorial" form in (5) results straightforwardly from observing that the term z^n occurs as many times as there are objects in $\mathcal{A}$ having size n. This form shows that generating functions are nothing but a reduced representation of the combinatorial class[1], where "internal" structures are destroyed and elements contributing to size ("atoms") are replaced by the variable z. Here is an illustration: start with a (finite) family of graphs $\mathcal{G}$, with size taken as the number of vertices [line 1]. Each vertex in each graph is replaced by the variable z and the graph structure is "forgotten" [line 2]; then the monomials corresponding to each graph are formed [line 3] and the generating function is obtained [line 4] by gathering all the monomials:

[1]This observation of which great use was made by Schützenberger as early as the 1950's and 1960's "explains" why many similarities are to be found between combinatorial structures and generating functions.

$\mathcal{G}$ = 

$$G(z) = z + z^2 + 2z^3 + 3z^4$$

For instance, there are three graphs of size 4, in agreement with the fact that $[z^4]G(z) = 3$. If size had been instead defined by number of edges, another generating function would have resulted, namely, with y marking size: $1 + y + y^2 + 2y^3 + y^4 + y^6$. If both number of vertices and number of edges are of interest, then a bivariate generating function, $G(z, y) = z + z^2y + z^3y^2 + z^3y^3 + z^4y^3 + z^4y^4 + z^4y^6$; such multivariate generating functions are developed systematically in Chapter III.

A path often taken in the older or more traditional literature consists in decomposing the structures to be enumerated into smaller structures either of the same type or of simpler types, and then in extracting from such a decomposition *recurrence relations* satisfied by the $\{A_n\}$. In this context, the recurrence relations are either solved directly—whenever they are simple enough—or by means of *ad hoc* generating functions, then introduced as a mere technical artefact.

In the framework to be described, classes of combinatorial structures are built *directly* in terms of simpler classes by means of a collection of elementary combinatorial *constructions*. (This closely resembles the description of formal languages by means of grammars, as well as the construction of structured data types in programming languages.) The approach developed here has been termed "symbolic", as it relies on a formal specification language for combinatorial structures. Specifically, it is based on so–called *admissible constructions* that admit direct translations into generating functions. In this chapter, the generating functions considered are ordinary generating functions.

DEFINITION I.4. *Assume that Φ is a construction that associates to a finite collection of classes $\mathcal{B}, \mathcal{C}, \cdots$ a new class*

$$\mathcal{A} := \Phi[\mathcal{B}, \mathcal{C}, \ldots],$$

in a finitary way: each $\mathcal{A}_n$ depends on finitely many of the $\{\mathcal{B}_j\}, \{\mathcal{C}_j\}, \ldots$. Then Φ is admissible iff the counting sequence $\{A_n\}$ of $\mathcal{A}$ only depends on the counting sequences $\{B_j\}, \{C_j\}, \ldots$ of $\mathcal{B}$ and $\mathcal{C}$:

$$\{A_n\} = \Xi[\{B_j\}, \{C_j\}].$$

In that case, there exists a well defined operator Ψ relating the associated ordinary generating functions

$$A(z) = \Psi[B(z), C(z), \ldots].$$

As an introductory example, take the construction of cartesian product that forms ordered pairs (equivalently, "records" in classical programming languages):

$$(a) \qquad \mathcal{A} = \mathcal{B} \times \mathcal{C} \quad \text{iff} \quad \mathcal{A} = \{\, \alpha = (\beta, \gamma) \mid \beta \in \mathcal{B},\ \gamma \in \mathcal{C} \,\},$$

the size of a pair $\alpha = (\beta, \gamma)$ being defined by $|\alpha|_{\mathcal{A}} = |\beta|_{\mathcal{B}} + |\gamma|_{\mathcal{C}}$. Then, considering all possibilities, the counting sequences corresponding to $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are related by the convolution relation

$$(b) \qquad A_n = \sum_{k=0}^{n} B_k C_{n-k}.$$

We recognize here the formula for a product of two power series. Therefore, with $A(z) = \sum_{n\geq 0} A_n z^n$ etc, one has

$$A(z) = B(z) \cdot C(z). \tag{c}$$

Thus in our terminology, the cartesian product is admissible: *A cartesian product translates as a product of OGF's.*

Similarly, let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be combinatorial classes satisfying

$$\mathcal{A} = \mathcal{B} \cup \mathcal{C}, \qquad \text{with} \quad \mathcal{B} \cap \mathcal{C} = \emptyset, \tag{d}$$

with size defined in a consistent manner. One has

$$A_n = B_n + C_n, \tag{e}$$

which, at generating function level, means

$$A(z) = B(z) + C(z). \tag{f}$$

Thus, *a disjoint union translates as a sum of generating functions*.

The correspondences Eq. (a)–(c) and (d)–(f) summarized by the table

$$\begin{cases} \mathcal{A} = \mathcal{B} \cup \mathcal{C} & \Longrightarrow \quad A(z) = B(z) + C(z) \qquad \text{(provided } \mathcal{B} \cap \mathcal{C} = \emptyset\text{)} \\ \mathcal{A} = \mathcal{B} \times \mathcal{C} & \Longrightarrow \quad A(z) = B(z) \cdot C(z) \end{cases}$$

are clearly very general ones. (Compare with Eq. (3).) Their merit is that they can be stated as general-purpose translation rules that only need to be established once and for all. As soon as the problem of counting elements of a disjoint union or a cartesian product is recognized, it becomes possible to dispense altogether with the intermediate stages of writing explicitly coefficient relations like (f) or recurrences like (b). This is the spirit of the symbolic method for combinatorial enumerations. Its interest lies in the fact that several powerful set-theoretic constructions are amenable to such a treatment.

I. 2. Admissible constructions and specifications

The main goal of this section is to introduce formally the basic constructions that constitute the core of a specification language for combinatorial structures. This core is based on disjoint unions (or sums) and on Cartesian products that we have just discussed. We shall introduce the constructions of sequence, cycle, multiset, and powerset. A class is (fully) constructible if it can be defined from primal elements by means of these constructions. The generating function of any such class satisfies functional equations that can be transcribed systematically from a specification; see Figure 2.

First, we assume given a class $\mathcal{E}$ called the *neutral class* that consists of a single object of size 0; any such an object of size 0 is called a *neutral object*. and is usually denoted by symbols like ϵ or $\mathbf{1}$. The reason for this terminology becomes clear if one considers the combinatorial isomorphism

$$\mathcal{A} \cong \mathcal{E} \times \mathcal{A} \cong \mathcal{A} \times \mathcal{E}.$$

We also assume as given an *atomic class* $\mathcal{Z}$ comprising a single element of size 1; any such element is called an atom; the atom may used to describe a generic node in a tree or graph, in which case it may be represented by a circle ($\bullet$ or $\circ$), but also a generic letter in a word, in which case it may be instantiated as $a, b, c, \ldots$. Distinct copies of the neutral or atomic class may also be subscripted by indices in various ways. Thus, for instance we may use the classes $\mathcal{Z}_a = \{a\}$, $\mathcal{Z}_b = \{b\}$ (with a, b of size 1) to build up binary words over the alphabet $\{a, b\}$, or $\mathcal{Z}_\bullet = \{\bullet\}$, $\mathcal{Z}_\circ = \{\circ\}$ (with $\bullet, \circ$ taken to be of size 1) to build

1. The main constructions of union, product, sequence, set, multiset, and cycle and their translation into generating functions (Theorem I.1).

Construction		*OGF*
Union	$\mathcal{A} = \mathcal{B} + \mathcal{C}$	$A(z) = B(z) + C(z)$
Product	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$A(z) = B(z) \cdot C(z)$
Sequence	$\mathcal{A} = \mathfrak{S}\{\mathcal{B}\}$	$A(z) = \dfrac{1}{1 - B(z)}$
Set	$\mathcal{A} = \mathfrak{P}\{\mathcal{B}\}$	$A(z) = \exp\left(B(z) - \dfrac{1}{2}B(z^2) + \cdots\right)$
Multiset	$\mathcal{A} = \mathfrak{M}\{\mathcal{B}\}$	$A(z) = \exp\left(B(z) + \dfrac{1}{2}B(z^2) + \cdots\right)$
Cycle	$\mathcal{A} = \mathfrak{C}\{\mathcal{B}\}$	$A(z) = \log\dfrac{1}{1 - B(z)} + \dfrac{1}{2}\log\dfrac{1}{1 - B(z^2)} + \cdots$

2. The translation for sets, multisets, and cycles constrained by the number of components (Theorem I.2).

$$
\begin{aligned}
\mathfrak{S}_k\{\mathcal{B}\} &: \quad B(z)^k \\
\mathfrak{P}_2\{\mathcal{B}\} &: \quad \tfrac{B(z)^2}{2} - \tfrac{B(z^2)}{2} \\
\mathfrak{M}_2\{\mathcal{B}\} &: \quad \tfrac{B(z)^2}{2} + \tfrac{B(z^2)}{2} \\
\mathfrak{C}_2\{\mathcal{B}\} &: \quad \tfrac{B(z)^2}{2} + \tfrac{B(z^2)}{2} \\
\mathfrak{P}_3\{\mathcal{B}\} &: \quad \tfrac{B(z)^3}{6} - \tfrac{B(z)\,B(z^2)}{2} + \tfrac{B(z^3)}{3} \\
\mathfrak{M}_3\{\mathcal{B}\} &: \quad \tfrac{B(z)^3}{6} + \tfrac{B(z)\,B(z^2)}{2} + \tfrac{B(z^3)}{3} \\
\mathfrak{C}_3\{\mathcal{B}\} &: \quad \tfrac{B(z)^3}{3} + \tfrac{2B(z^3)}{3} \\
\mathfrak{P}_4\{\mathcal{B}\} &: \quad \tfrac{B(z)^4}{24} - \tfrac{B(z)^2B(z^2)}{4} + \tfrac{B(z)B(z^3)}{3} + \tfrac{B(z^2)^2}{8} - \tfrac{B(z^4)}{4} \\
\mathfrak{M}_4\{\mathcal{B}\} &: \quad \tfrac{B(z)^4}{24} + \tfrac{B(z)^2B(z^2)}{4} + \tfrac{B(z)B(z^3)}{3} + \tfrac{B(z^2)^2}{8} + \tfrac{B(z^4)}{4} \\
\mathfrak{C}_4\{\mathcal{B}\} &: \quad \tfrac{B(z)^4}{4} + \tfrac{B(z^2)^2}{2} + \tfrac{B(z^4)}{2}.
\end{aligned}
$$

3. The additional constructions of pointing and substitution (Section I. 6).

Construction		*OGF*
Pointing	$\mathcal{A} = \Theta\mathcal{B}$	$A(z) = z\frac{d}{dz}B(z)$
Substitution	$\mathcal{A} = \mathcal{B} \circ \mathcal{C}$	$A(z) = B(C(z))$

FIGURE 2. A "dictionary" of constructions applicable to *unlabelled* structures, together with their translation into ordinary generating functions (OGFs). (The labelled counterpart of this table appears in Figure 2 of Chapter II, p. 67.)

trees. Similarly, we may introduce $\mathcal{E}_\Box, \mathcal{E}_1, \mathcal{E}_2$ to denote a class comprising the neutral objects $\Box, \epsilon_1, \epsilon_2$ respectively. Clearly, the generating functions of a neutral class $\mathcal{E}$ and an atomic class $\mathcal{Z}$ are

$$E(z) = 1, \qquad Z(z) = z,$$

corresponding to the unit 1, and the variable z, of generating functions.

I. 2.1. Basic constructions. Here are described a few powerful constructions that build upon disjoint unions and cartesian products, and form sequences, sets, and cycles.

First consider the *disjoint union* also called the *combinatorial sum* of classes, the intent being to capture the union of disjoint sets, but without the burden of carrying extraneous disjointness conditions. We formalize the (combinatorial) sum of two classes $\mathcal{B}$ and $\mathcal{C}$ as the union (in the standard set–theoretic sense) of two *disjoint* copies, say $\mathcal{B}^\Box$ and $\mathcal{C}^\Diamond$, of $\mathcal{B}$ and $\mathcal{C}$. A picturesque way to view the construction is as follows: first choose two distinct colours and repaint the elements of $\mathcal{B}$ with the $\Box$-colour and the elements of $\mathcal{C}$ with the $\Diamond$-colour. This is made precise by introducing two distinct "markers" $\Box$ and $\Diamond$, each a neutral object (*i.e.*, of size zero); the disjoint union $\mathcal{B} + \mathcal{C}$ of $\mathcal{B}, \mathcal{C}$ is then defined as the standard set-theoretic union,

$$\mathcal{B} + \mathcal{C} := (\{\Box\} \times \mathcal{B}) \cup (\{\Diamond\} \times \mathcal{C}) .$$

The size of an object in a disjoint union $\mathcal{A} = \mathcal{B} + \mathcal{C}$ is by definition inherited from its size in its class of of origin. One reason behind the definition[2] adopted here of disjoint union is that the combinatorial sum of two classes is always well-defined. Furthermore, we have ($\cong$ represents combinatorial isomorphism)

$$\mathcal{B} + \mathcal{C} \cong \mathcal{B} \cup \mathcal{C} \quad \text{whenever} \quad \mathcal{B} \cap \mathcal{C} = \emptyset.$$

Disjoint union in the above sense is thus equivalent to a standard union whenever it is applied to disjoint sets. Then, because of disjointness, one has the implication

$$\mathcal{A} = \mathcal{B} + \mathcal{C} \quad \Longrightarrow \quad A_n = B_n + C_n \quad \Longrightarrow \quad A(z) = B(z) + C(z),$$

so that disjoint union is admissible. Note that, in contrast, standard set-theoretic union is not admissible since

$$\text{card}(\mathcal{B}_n \cup \mathcal{C}_n) = \text{card}(\mathcal{B}_n) + \text{card}(\mathcal{C}_n) - \text{card}(\mathcal{B}_n \cap \mathcal{C}_n),$$

and information on the "internal structure" of $\mathcal{B}$ and $\mathcal{C}$ (i.e., the nature of this intersection) is needed in order to be able to enumerate the elements of their union.

With the convention of identifying isomorphic classes, sum and product acquire pleasant algebraic properties: sums and cartesian products become commutative and associative operations, e.g.,

$$(\mathcal{A} + \mathcal{B}) + \mathcal{C} = \mathcal{A} + (\mathcal{B} + \mathcal{C}), \qquad \mathcal{A} \times (\mathcal{B} \times \mathcal{C}) = (\mathcal{A} \times \mathcal{B}) \times \mathcal{C},$$

while distributivity holds, $(\mathcal{A} + \mathcal{B}) \times \mathcal{C} = (\mathcal{A} \times \mathcal{C}) + (\mathcal{B} \times \mathcal{C})$. (The proofs are simple verifications from the definitions and the notion of combinatorial isomorphism.)

Next, we turn to the sequence construction. If $\mathcal{C}$ is a class then the *sequence* class $\mathfrak{S}\{\mathcal{C}\}$ is defined as the infinite sum

$$\mathfrak{S}\{\mathcal{C}\} = \{\epsilon\} + \mathcal{C} + (\mathcal{C} \times \mathcal{C}) + (\mathcal{C} \times \mathcal{C} \times \mathcal{C}) + \cdots$$

[2] It would have been inconvenient to have a construction that translates into generating functions under some external condition—disjointness—of a logical nature that would need to be established separately in each particular case.

with ϵ being a neutral structure (of size 0). (The neutral structure in this context plays a rôle similar to that of the "empty" word in formal language theory, while the sequence construction is somewhat analogous to the Kleene star operation ('*'); see APPENDIX: *Regular languages*, p. 171.) It is then readily checked that the construction $\mathcal{A} = \mathfrak{S}\{\mathcal{C}\}$ defines a proper class satisfying the finiteness condition for sizes if and only if $\mathcal{C}$ contains no object of size 0. From the definition of size for sums and products, there results that the size of a sequence is to be taken as the sum of the sizes of its components:

$$\gamma = (\alpha_1, \ldots, \alpha_\ell) \qquad \Longrightarrow \qquad |\gamma| = |\alpha_1| + \cdots + |\alpha_\ell|.$$

▷ **3.** *Natural numbers.* Let $\mathcal{Z} := \{\bullet\}$ with $\bullet$ an atom (of size 1). Then $\mathcal{I} = \mathfrak{S}\{Z\} \setminus \{\epsilon\}$ is a way of describing natural integers in unary notation: $\mathcal{I} = \{\bullet,\ \bullet\bullet,\ \bullet\bullet\bullet, \ldots\}$. The corresponding OGF is $I(z) = z/(1-z) = z + z^2 + z^3 + \cdots$. ◁

▷ **4.** *Interval coverings.*

Let $\mathcal{Z} := \{\bullet\}$ be as before. Then $\mathcal{A} = \mathcal{Z} + (\mathcal{Z} \times \mathcal{Z})$ is a set of two elements, $\bullet$ and $(\bullet, \bullet)$, which we choose to draw as $\{\bullet,\ \bullet\text{–}\bullet\}$. Then $\mathcal{C} = \mathfrak{S}\{\mathcal{A}\}$ contains elements like

$$\bullet,\ \bullet\bullet,\ \bullet\text{–}\bullet,\ \bullet\bullet\text{–}\bullet,\ \bullet\text{–}\bullet\bullet,\ \bullet\text{–}\bullet\bullet\text{–}\bullet,\ \bullet\ \bullet\ \bullet\ \bullet\ .$$

With the notion of size adopted, the objects of size n in $\mathcal{C} = \mathfrak{S}\{\mathcal{Z} + (\mathcal{Z} \times \mathcal{Z})\}$ are (isomorphic to) the *coverings* of the interval $[0, n]$ by matches of length either 1 or 2. The generating function

$$C(z) = 1 + z + 2\,z^2 + 3\,z^3 + 5\,z^4 + 8\,z^5 + 13\,z^6 + 21\,z^7 + 34\,z^8 + 55\,z^9 + 89\,z^{10} + \cdots,$$

is, as we shall see shortly (p. 24), the OGF of Fibonacci numbers. ◁

Cycles are merely sequences defined up to a circular shift of their components, the notation being $\mathfrak{C}\{\mathcal{B}\}$. Thus, $\mathfrak{C}\{\mathcal{B}\} := \mathfrak{S}\{\mathcal{B}\}/\mathbf{S}$ with $\mathbf{S}$ the equivalence relation between sequences defined by $(\alpha_1, \ldots, \alpha_r)\,\mathbf{S}\,(\beta_1, \ldots, \beta_r)$ iff there exists some circular shift σ of $[1\,..\,n]$ such that for all j, $\beta_j = \alpha_{\sigma(j)}$; in other words, for some d, one has $\beta_j = \alpha_{1+(j+d) \bmod n}$. Here is for instance a depiction of the cycles formed from the 8 and 16 sequences of lengths 3 and 4 over two types of objects (a, b): the number of cycles is 4 (for $n = 3$) and 6 (for $n = 4$). Sequences are grouped into equivalence classes according to the relation $\mathbf{S}$.

$$\begin{array}{c} aaa \\ aab\ aba\ baa \\ abb\ bba\ bab \\ bbb \end{array} \qquad \begin{array}{c} aaaa \\ aaab\ aaba\ abaa\ baaa \\ aabb\ bbba\ bbaa\ baab \\ abab\ baba \\ abbb\ bbba\ bbab\ babb \\ bbbb \end{array}$$

This construction corresponds to the formation of directed cycles. We make only a limited use of it for unlabelled objects; however, its counterpart plays a rather important rôle in the context of labelled structures and exponential generating functions.

Multisets are like finite sets (that is the order between element does not count) but arbitrary repetitions of elements are allowed. The notation is $\mathcal{A} = \mathfrak{M}\{\mathcal{B}\}$ when $\mathcal{A}$ is obtained by forming all finite multisets of elements from $\mathcal{B}$. The precise way of defining $\mathfrak{M}\{\mathcal{B}\}$ is as a quotient: $\mathfrak{M}\{\mathcal{B}\} := \mathfrak{S}\{\mathcal{B}\}/\mathbf{R}$ with $\mathbf{R}$ the equivalence relation between sequences defined by $(\alpha_1, \ldots, \alpha_r)\,\mathbf{R}\,(\beta_1, \ldots, \beta_r)$ iff, there exists some arbitrary permutation σ of $[1\,..\,n]$ such that for all j, $\beta_j = \alpha_{\sigma(j)}$. The *powerset* class (or set class) $\mathcal{A} = \mathfrak{P}\{\mathcal{B}\}$ is defined as the class consisting of all *finite* subsets of class $\mathcal{B}$, or equivalently, as the class $\mathfrak{P}\{\mathcal{B}\} \subset \mathfrak{M}\{\mathcal{B}\}$ formed of multisets that involve no repetitions. We again need to make explicit the way the size function is defined when such constructions are performed: like for products and sequences, the size of a composite object—set, multiset, or cycle—is defined as the sum of the sizes of its components.

In what follows, we also want to impose restrictions on the number of components allowed in sequences, sets, multisets, and cycles. Let $\mathfrak{K}$ be any of $\mathfrak{S}, \mathfrak{C}, \mathfrak{M}, \mathfrak{P}$ and let Ω be

a predicate over the integers, then $\mathfrak{K}_{\Omega}\{\mathcal{A}\}$ will represent the class of objects constructed by $\mathfrak{K}$ but with a number of components constrained to satisfy Ω. Then, the notations

$$\mathfrak{S}_{=k} \text{ (or simply } \mathfrak{S}_k),\ \mathfrak{S}_{>k},\ \mathfrak{S}_{1\,..\,k}$$

refer to sequences whose number of components are exactly k, larger than k, or in the interval $1\,..\,k$ respectively. For example, one has

$$\mathfrak{S}_k\{\mathcal{B}\} := \overbrace{\mathcal{B}\times\cdots\mathcal{B}}^{k \text{ times}} \equiv \mathcal{B}^k, \qquad \mathfrak{M}_k\{\mathcal{B}\} := \mathfrak{S}_k\{\mathcal{B}\}/\mathbf{R}, \quad \mathfrak{S}_{\geq k}\{\mathcal{B}\} \cong \mathcal{B}^k \times \mathfrak{S}\{\mathcal{B}\}.$$

Similarly $\mathfrak{S}_{\mathrm{odd}}, \mathfrak{S}_{\mathrm{even}}$ will denote sequences with an odd or even number of components, and so on.

I. 2.2. The admissibility theorem for ordinary generating functions. This section shows that any specification of a constructible class translates directly into generating function equations. The cycle construction involves the Euler totient function $\varphi(k)$ defined as the number of integers in $[1,k]$ that are relatively prime to k (APPENDIX: *Arithmetical functions*, p. 165).

THEOREM I.1 (Admissible unlabelled constructions). *The constructions of union, cartesian product, sequence, multiset, powerset, and cycle are all admissible. The associated operators are*

$$\begin{aligned}
&\textit{Union:} && \mathcal{A}=\mathcal{B}+\mathcal{C} && \Longrightarrow && A(z)=B(z)+C(z)\\
&\textit{Product:} && \mathcal{A}=\mathcal{B}\times\mathcal{C} && \Longrightarrow && A(z)=B(z)\cdot C(z)\\
&\textit{Sequence:} && \mathcal{A}=\mathfrak{S}\{\mathcal{B}\} && \Longrightarrow && A(z)=\frac{1}{1-B(z)}\\
&\textit{Cycle:} && \mathcal{A}=\mathfrak{C}\{\mathcal{B}\} && \Longrightarrow && A(z)=\sum_{k=1}^{\infty}\frac{\varphi(k)}{k}\log\frac{1}{1-B(z^k)}.\\
&\textit{Multiset:} && \mathcal{A}=\mathfrak{M}\{\mathcal{B}\} && \Longrightarrow && A(z)=\left\{\begin{array}{l}\displaystyle\prod_{n\geq 1}(1-z^n)^{-B_n}\\ \displaystyle\exp\left(\sum_{k=1}^{\infty}\frac{1}{k}B(z^k)\right)\end{array}\right.\\
&\textit{Powerset:} && \mathcal{A}=\mathfrak{P}\{\mathcal{B}\} && \Longrightarrow && A(z)=\left\{\begin{array}{l}\displaystyle\prod_{n\geq 1}(1+z^n)^{B_n}\\ \displaystyle\exp\left(\sum_{k=1}^{\infty}\frac{(-1)^{k-1}}{k}B(z^k)\right)\end{array}\right.
\end{aligned}$$

For the sequence, cycle, and set constructions, it is assumed that $B_0=0$.

The class $\mathcal{E}=\{\epsilon\}$ consisting of the neutral structure only, and the class $\mathcal{Z}$ consisting of a single "atomic" object (node, letter) of size 1 have OGFs

$$E(z)=1 \qquad \text{and} \qquad Z(z)=z.$$

PROOF. *Union:* Let $\mathcal{A}=\mathcal{B}+\mathcal{C}$. Since the union is *disjoint*, and the size of an $\mathcal{A}$–element coincides with its size in $\mathcal{B}$ or $\mathcal{C}$, one has $A_n=B_n+C_n$ and

$$A(z)=B(z)+C(z),$$

as has discussed earlier. Alternatively, the translation rule follows directly from the combinatorial form of generating functions as

$$\sum_{\alpha\in\mathcal{A}} z^{|\alpha|} = \sum_{\alpha\in\mathcal{B}} z^{|\alpha|} + \sum_{\alpha\in\mathcal{C}} z^{|\alpha|}.$$

Cartesian Product: The admissibility result for $\mathcal{A} = \mathcal{B}\times\mathcal{C}$ has been discussed already. It follows from $A_n = \sum_{k=0}^{n} B_k C_{n-k}$ that

$$A(z) = B(z) \times C(z).$$

Note also the alternative direct derivation based on the combinatorial form of GF's,

$$\sum_{\alpha\in\mathcal{A}} z^{|\alpha|} = \sum_{(\beta,\gamma)\in(\mathcal{B}\times\mathcal{C})} z^{|\beta|+|\gamma|} = \left(\sum_{\beta\in\mathcal{B}} z^{|\beta|}\right) \times \left(\sum_{\gamma\in\mathcal{C}} z^{|\gamma|}\right),$$

as follows from distributing products over sums. The result readily extends to an arbitrary number of factors.

Sequence: Admissibility for $\mathcal{A} = \mathfrak{S}\{\mathcal{B}\}$ (with $\mathcal{B}_0 = \emptyset$) follows from the union and product relations. One has

$$\mathcal{A} = \{\epsilon\} + \mathcal{B} + (\mathcal{B}\times\mathcal{B}) + (\mathcal{B}\times\mathcal{B}\times\mathcal{B}) + \cdots,$$

so that

$$A(z) = 1 + B(z) + B(z)^2 + B(z)^3 + \cdots = \frac{1}{1-B(z)},$$

where the geometric sum converges in the sense of formal power series since $[z^{\,0}]B(z) = 0$, by assumption.

Set (or powerset) construction: Let $\mathcal{A} = \mathfrak{P}\{\mathcal{B}\}$ and first take $\mathcal{B}$ to be finite. Then, the class $\mathcal{A}$ of all the finite subsets of $\mathcal{B}$ is isomorphic to a product,

$$\mathfrak{P}\{\mathcal{B}\} \cong \prod_{\beta\in\mathcal{B}} (\{\epsilon\} + \{\beta\})$$

with ϵ a neutral structure of size 0. Indeed, distributing the products in all possible ways forms all the possible combinations, i.e., sets, of elements of $\mathcal{B}$ with no repetition allowed. The technique is similar to what is required to establish identities like

$$(1+a)(1+b)(1+c) = 1 + [a+b+c] + [ab+bc+ac] + abc,$$

where all combinations of variables appear. Then, directly from the combinatorial form (4) of OGF's and the sum and product rules, we find

$$A(z) = \prod_{\beta\in\mathcal{B}} (1 + z^{|\beta|}) = \prod_{n} (1+z^n)^{B_n}.$$

The "*exp-log transformation*", $A(z) = \exp(\log A(z))$, then yields

$$\begin{aligned} A(z) &= \exp\Big(\sum_{n=1}^{\infty} B_n \log(1+z^n)\Big) \\ &= \exp\Big(\sum_{n=1}^{\infty} B_n \cdot \sum_{k=1}^{\infty} \frac{z^{nk}}{k}\Big) \\ &= \exp\Big(\frac{B(z)}{1} - \frac{B(z^2)}{2} + \frac{B(z^3)}{3} - \cdots\Big), \end{aligned} \tag{7}$$

where the second line results from expanding the logarithm,

$$\log(1+u) = \frac{u}{1} - \frac{u^2}{2} + \frac{u^3}{3} - \cdots$$

and the third line results from exchanging summations.

The proof extends to the case of $\mathcal{B}$ being infinite by noting that each $\mathcal{A}_n$ depends only on those $\mathcal{B}_j$ for which $j \leq n$, to which the relations given above for the finite case apply. Precisely, let $\mathcal{B}^{(\leq m)} = \sum_{k=1}^{m} \mathcal{A}_j$ and $\mathcal{A}^{(\leq m)} = \mathfrak{P}\{\mathcal{B}^{(\leq m)}\}$. Then, with $O(z^{m+1})$ denoting any series that has no term of degree $\leq m$, one has

$$A(z) = A^{(\leq m)}(z) + O(z^{m+1}) \qquad \text{and} \qquad B(z) = B^{(\leq m)}(z) + O(z^{m+1}).$$

On the other hand, $A^{(\leq m)}(z)$ and $B^{(\leq m)}(z)$ are connected by the fundamental exponential relation (7) , since $\mathcal{A}^{(\leq m)}$ is finite. Letting m tend to infinity, there follows in the limit

$$A(z) = \exp\Big(\frac{B(z)}{1} - \frac{B(z^2)}{2} + \frac{B(z^3)}{3} - \cdots\Big).$$

(See APPENDIX: *Formal power series*, p. 169 for definitions of formal convergence.) The necessary condition for validity is that $[z^0]B(z) = 0$, a restriction that also applies to multisets and cycles.

Multiset: First for finite $\mathcal{B}$ (with $\mathcal{B}_0 = \emptyset$), the multiset class $\mathcal{A} = \mathfrak{M}\{\mathcal{B}\}$ is definable by

$$\mathfrak{M}\{\mathcal{B}\} \cong \prod_{\beta\in\mathcal{B}} \mathfrak{S}\{\beta\}.$$

In words, any multiset can be sorted, in which case it can be viewed as formed of a sequence of repeated elements β_1, followed by a sequence of repeated elements β_2, where $\beta_1, \beta_2, \ldots$ is a canonical listing of the elements of $\mathcal{B}$. The relation translates into generating functions by the product and sequence rules,

$$\begin{aligned} A(z) &= \prod_{\beta\in\mathcal{B}} (1-z^{|\beta|})^{-1} = \prod_{n=1}^{\infty} (1-z)^{-B_n} \\ &= \exp\Big(\sum_{n=1}^{\infty} B_n \log(1-z^n)^{-1}\Big) \\ &= \exp\Big(\frac{B(z)}{1} + \frac{B(z^2)}{2} + \frac{B(z^3)}{3} + \cdots\Big), \end{aligned}$$

where the exponential form results from the "exp-log transformation". The case of an infinite class $\mathcal{B}$ follows similarly by a continuity argument.

Cycle: The translation of the cycle relation $\mathcal{A} = \mathfrak{C}\{\mathcal{B}\}$ is

$$A(z) = \sum_{k=1}^{\infty} \frac{\varphi(k)}{k} \log \frac{1}{1-B(z^k)},$$

where $\varphi(k)$ is the Euler totient function: $\varphi(k)$ equals the number of integers in $[1,k]$ that are relatively prime to k, with $\varphi(1) = 1$. The first terms, with $L_k = \log(1-B(z^k))^{-1}$ are

$$A(z) = \frac{1}{1}L_1 + \frac{1}{2}L_2 + \frac{2}{3}L_3 + \frac{2}{4}L_4 + \frac{4}{5}L_5 + \frac{2}{6}L_6 + \frac{6}{7}L_7 + \cdots.$$

This translation was first established by Read within the framework of Pólya's theory of counting [**115**]. An elementary combinatorial derivation based on [**58**] is given in APPENDIX: *Cycle construction*, p. 168. □

The results for sets, multisets, and cycles are particular cases of the well known *Pólya theory* that deals more generally with the enumeration of objects under group symmetry actions [**113, 115**]. This theory is exposed in many textbooks, see for instance [**28, 76**]. The approach adopted here consists in considering simultaneously all possible values of the number of components by means of bivariate generating functions. Powerful generalizations within the theory of species are presented in the book [**13**].

Restricted constructions. An immediate formula for OGF's is that of the *diagonal* Δ of a cartesian product $\mathcal{B} \times \mathcal{B}$ defined as

$$\mathcal{A} \equiv \Delta(\mathcal{B} \times \mathcal{B}) := \{(\beta, \beta) \mid \beta \in \mathcal{B}\}.$$

Then, clearly $A_{2n} = B_n$ so that

$$A(z) = B(z^2).$$

The diagonal construction permits us to access the class of all unordered pairs of (distinct) elements of $\mathcal{B}$, which is $\mathcal{A} = \mathfrak{P}_2\{\mathcal{B}\}$. A direct argument then runs as follows: the unordered pair $\{\alpha, \beta\}$ is associated to the two ordered pairs (α, β) and (β, α) except when $\alpha = \beta$, where an element of the diagonal is obtained. In other words, one has the combinatorial isomorphism,

$$\mathfrak{P}_2\{\mathcal{B}\} + \mathfrak{P}_2\{\mathcal{B}\} + \Delta(B \times B) \cong B \times B,$$

meaning that

$$2A(z) + B(z^2) = B(z)^2.$$

The resulting translation into OGFs is thus

$$\mathcal{A} = \mathfrak{P}_2\{\mathcal{B}\} \qquad \Longrightarrow \qquad A(z) = \frac{1}{2}B(z)^2 - \frac{1}{2}B(z^2).$$

Similarly, for multisets, we find

$$\mathcal{A} = \mathfrak{M}_2\{\mathcal{B}\} \qquad \Longrightarrow \qquad A(z) = \frac{1}{2}B(z)^2 + \frac{1}{2}B(z^2),$$

while for cycles one has $\mathfrak{C}_2 \cong \mathfrak{M}_2$, and

$$\mathcal{A} = \mathfrak{C}_2\{\mathcal{B}\} \qquad \Longrightarrow \qquad A(z) = \frac{1}{2}B(z)^2 + \frac{1}{2}B(z^2).$$

This type of direct reasoning could be extended to treat triples, and so on, but the computations (if not the reasoning) tend to grow out of control. An approach based on multivariate generating functions generates *simultaneously* all cardinality restricted constructions.

THEOREM I.2 (Component-restricted constructions). *The OGF of sequences with k components $\mathcal{A} = \mathfrak{S}_k\{\mathcal{B}\}$ satisfies*

$$A(z) = B(z)^k.$$

The OGF of sets, $\mathcal{A} = \mathfrak{P}_k\{\mathcal{B}\}$, is a polynomial in the quantities $B(z), \dots, B(z^k)$,

$$A(z) = [u^k] \exp\left(\frac{u}{1}B(z) - \frac{u^2}{2}B(z^2) + \frac{u^3}{3}B(z^3) - \cdots\right).$$

The OGF of multisets, $\mathcal{A} = \mathfrak{M}_k\{\mathcal{B}\}$, is

$$A(z) = [u^k] \exp\left(\frac{u}{1}B(z) + \frac{u^2}{2}B(z^2) + \frac{u^3}{3}B(z^3) + \cdots\right).$$

The OGF of cycles, $\mathcal{A} = \mathfrak{C}_k\{\mathcal{B}\}$, is

$$A(z) = [u^k] \sum_{\ell=1}^{\infty} \frac{\varphi(\ell)}{\ell} \log \frac{1}{1 - u^\ell B(z^\ell)}.$$

The explicit forms for small values of k are summarized in Figure 2.

PROOF. The result for sequences is obvious since $\mathfrak{S}_k\{\mathcal{B}\}$ means $\mathcal{B} \times \cdots \times \mathcal{B}$ (k times). For the other constructions, the proof makes use of the techniques of Theorem I.1, but it is best based on bivariate generating functions that are otherwise developed fully in Chapter III to which we refer for details. The idea consists in describing all composite objects and introducing a supplementary marking variable to keep track of the number of components.

Take $\mathfrak{K}$ to be a construction amongst $\mathfrak{S}, \mathfrak{C}, \mathfrak{M}, \mathfrak{P}$, set $\mathcal{A} = \mathfrak{K}\{\mathcal{B}\}$, and let $\chi(\alpha)$ for $\alpha \in \mathcal{A}$ be the parameter "number of $\mathcal{B}$–components". Define the multivariate quantities

$$\begin{aligned} A_{n,k} &:= \text{card}\left\{\alpha \in \mathcal{A} \;\middle|\; |\alpha| = n,\ \chi(\alpha) = k\right\} \\ A(z,u) &:= \sum_{n,k} A_{n,k} u^k z^n = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|} u^{\chi(\alpha)}. \end{aligned}$$

For instance, a direct calculation shows that, for sequences, there holds

$$\begin{aligned} A(z,u) &= \sum_{k \geq 0} u^k B(z)^k \\ &= \frac{1}{1 - uB(z)}. \end{aligned}$$

For multisets and powersets, a simple adaptation of the already seen argument gives $A(z,u)$ as

$$A(z,u) = \prod_n (1 - uz^n)^{-B_n}, \qquad A(z,u) = \prod_n (1 + uz^n)^{B_n},$$

respectively. The result follows from there by the "exp-log transformation" upon extracting $[u^k]A(z,u)$. □

▷ **5.** *Vallée's identity.* Let $\mathcal{M} = \mathfrak{M}\{\mathcal{C}\}$, $\mathcal{P} = \mathfrak{P}\{\mathcal{C}\}$. Separating elements of $\mathcal{C}$ according to the parity of the number of times they appear in a multiset gives rise to the identity

$$M(z) = P(z)M(z^2).$$

(Hint: a multiset contains elements of either odd or even multiplicity.) Accordingly, one can deduce the translation of powersets from the formula for multisets. Iterating the relation above yields $M(z) = P(z)P(z^2)P(z^4)P(z^8)\cdots$, that is closely related to the binary representation of numbers and to Euler's identity on page 28. ◁

▷ **6.** *Sets with distinct component sizes.* Let $\mathcal{A}$ be the class of the finite sets of elements from $\mathcal{B}$, with the additional constraint that no two elements in a set have the same size. One has

$$A(z) = \prod_{n=1}^{\infty} (1 + B_n z^n).$$

Similar identities serve for instance in the analysis of polynomial factorization algorithms [**49**]. ◁

▷ **7.** *Sequences without repeated components.* These have generating function formally given by

$$\int_0^\infty \exp\left(\sum_{k \geq 1} (-1)^{j-1} \frac{u^j}{j} A(z^j) \right) e^{-u}\, du.$$

(This form is based on the Eulerian integral: $k! = \int_0^\infty e^u u^k\, dn$.) ◁

I. 2.3. Constructibility and combinatorial specifications. In the framework just introduced, the class of all binary words is described by

$$\mathcal{W} = \mathfrak{S}\{\mathcal{A}\} \qquad \text{where} \qquad \mathcal{A} = \{a, b\},$$

the ground alphabet, comprises two elements (letters) of size 1. The size of a binary word then coincides with its length (the number of letters it contains). In other words, we start from basic atomic elements and build up words by forming freely all the objects determined by the sequence construction. Such a combinatorial description of a class that only involves a composition of basic constructions applied to initial classes $\mathcal{E}, \mathcal{Z}$ is said to be an *iterative* (or *nonrecursive*) *specification*. Other examples already encountered include binary necklaces (Ex. 2, p. 3) and the natural integers (Ex. 3, p. 9) respectively defined by

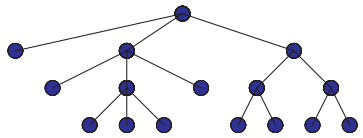
$$\mathcal{N} = \mathfrak{C}\{\mathcal{Z} + \mathcal{Z}\} \qquad \text{and} \qquad \mathcal{I} = \mathfrak{S}_{\geq 1}\{\mathcal{Z}\}.$$

From there, one can construct ever more complicated objects. For instance,

$$\mathcal{P} = \mathfrak{M}\{\mathcal{I}\} \equiv \mathfrak{M}\{\mathfrak{S}_{\geq 1}\{\mathcal{Z}\}\}$$

means the class of multisets of natural integers, which is isomorphic to the class of integer partitions (see Section I. 3 below for a detailed discussion). As such examples demonstrate, a specification that is iterative can be represented as a single term built on $\mathcal{E}, \mathcal{Z}$ and the constructions $+, \times, \mathfrak{S}, \mathfrak{C}, \mathfrak{M}, \mathfrak{P}$. An iterative specification can be equivalently listed by naming some of the subterms (for instance partitions in terms of natural integers themselves defined as sets of atoms).

We next turn our attention to trees (cf. also APPENDIX: *Tree concepts*, p. 174 for basic definitions). In graph theory, a tree is classically defined as an undirected graph that is connected and acyclic. Additionally, a tree is *rooted* if a particular vertex is distinguished—the "root". Computer scientists commonly make use of trees called *plane* that are rooted but also embedded in the plane. In other words, the ordering of subtrees attached to any node matters. Here, we will give the name of "general plane trees" to such rooted plane trees and call $\mathcal{G}$ their class, where size is the number of vertices; see [**130**]. (The term "general" refers to the fact that all nodes degrees are allowed.) For instance a general tree of size 16, drawn with the root on top, is:

$\tau =$ 

As a consequence of the definition, if one interchanges, say, the second and third root subtrees, then this will result in a different tree—the original tree and its variant are not homeomorphically equivalent. (General trees are thus comparable to graphical renderings of genealogies, where children are ordered by age.). Although we have introduced plane trees as 2-dimensional diagrams, it is obvious that any tree also admits a linear representation: a tree τ with root ζ and root subtrees $\tau_1, \ldots, \tau_r$ (in that order) can be seen as the object $\zeta\,\boxed{\tau_1, \ldots, \tau_r}$, where the box encloses similar representations of subtrees. Typographically, a box $\boxed{\cdot}$ may be reduced to a matching pair of parentheses, '$(\cdot)$', and one gets in this way a linear description that illustrates the correspondence between trees viewed as plane diagrams and functional terms of mathematical logic and computer science.

Trees are best described recursively. A tree is a root to which is attached a (possibly empty) sequence of trees. In other words, the class $\mathcal{G}$ of general trees is definable by the

recursive equation

$$\mathcal{G} = \mathcal{Z} \times \mathfrak{S}\{\mathcal{G}\}, \tag{8}$$

where $\mathcal{Z}$ comprises a single atom written ζ and denoting a generic node.

Although recursive definitions are familiar to computer scientists, the specification (8) may look dangerously circular to some. One way of making good sense of it is via an adaptation of the numerical technique of iteration. Start with $\mathcal{G}^{[0]} = \emptyset$, the empty set, and define successively the classes

$$\mathcal{G}^{[j+1]} = \mathcal{Z} \times \mathfrak{S}\{\mathcal{G}^{[j]}\}.$$

For instance, $\mathcal{G}^{[1]} = \mathcal{Z} \times \mathfrak{S}\{\emptyset\} = \{(\zeta, \epsilon)\} \cong \{\zeta\}$ describes (the linear representation of) the tree of size 1, and

$$\begin{aligned}
\mathcal{G}^{[2]} &= \left\{\zeta,\ \zeta\boxed{\zeta},\ \zeta\boxed{\zeta,\zeta},\ \zeta\boxed{\zeta,\zeta,\zeta},\ \dots,\right\}\\
\mathcal{G}^{[3]} &= \left\{\zeta,\ \zeta\boxed{\zeta},\ \zeta\boxed{\zeta,\zeta},\ \zeta\boxed{\zeta,\zeta,\zeta},\ \dots\right.\\
&\qquad \left.\zeta\boxed{\zeta,\boxed{\zeta}},\ \zeta\boxed{\zeta,\boxed{\zeta,\zeta}},\ \zeta\boxed{\zeta,\boxed{\zeta,\zeta}},\ \zeta\boxed{\zeta,\boxed{\zeta,\zeta},\boxed{\zeta,\zeta}},\dots\right\}.
\end{aligned}$$

First, each $\mathcal{G}^{[j]}$ is well-defined since it corresponds to a purely iterative specification. Next, we have the inclusion $\mathcal{G}^{[j]} \subset \mathcal{G}^{[j+1]}$, ($\mathcal{G}^{[j]}$ admits of a simple interpretation as the class of all trees of height $< j$). We can therefore regard the complete class $\mathcal{G}$ as defined by the "limit" of the $\mathcal{G}^{[j]}$: $\mathcal{G} := \bigcup_j \mathcal{G}^{[j]}$. (There, '$\cup$' represents the usual set-theoretic union.)

$\triangleright$ **8.** *Limes superior of classes.* Let $\{\mathcal{A}^{[j]}\}$ be any increasing sequence of combinatorial classes, in the sense that $\mathcal{A}^{[j]} \subset \mathcal{A}^{[j+1]}$. If $\mathcal{A}^{[\infty]} = \bigcup_j \mathcal{A}^{[j]}$ is a combinatorial class, then the corresponding OGF's satisfy $A^{[\infty]}(z) = \lim_{j\to\infty} A^{[j]}(z)$ in the formal topology (APPENDIX: *Formal power series*, p. 169). $\triangleleft$

In all generality, a *specification* for an r–tuple $\vec{\mathcal{A}} = (\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(r)})$ of classes is a collection of r equations,

$$\left\{\begin{aligned}
\mathcal{A}^{(1)} &= \Xi_1(\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(r)})\\
\mathcal{A}^{(2)} &= \Xi_2(\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(r)})\\
&\dots\\
\mathcal{A}^{(r)} &= \Xi_r(\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(r)})
\end{aligned}\right. \tag{9}$$

where each Ξ_i denotes a term built from the $\mathcal{A}$'s using the constructions of disjoint union, cartesian product, sequence, set, multiset, and cycle, as well as the "initial structures" $\mathcal{E}$ and $\mathcal{Z}$. We also say that the system is a specification of $\mathcal{A}^{(1)}$. A specification for a class of combinatorial structures is thus a sort of formal grammar defining that class. The system (9) corresponds to an iterative specification if it is stictly upper-triangular, that is, $\mathcal{A}^{(r)}$ is defined solely in terms of initial classes $\mathcal{Z}, \mathcal{E}$; the definition of $\mathcal{A}^{(r-1)}$ only involves $\mathcal{A}^{(r)}$, etc, so that $\mathcal{A}^{(1)}$ can be equivalently described by a single term. Otherwise, the system is said to be *recursive*. In the latter case, the semantics of recursion is identical to the one introduced in the case of trees: start with the "empty" vector of classes, $\vec{\mathcal{A}}^{[0]} := (\emptyset, \dots, \emptyset)$, iterate $\vec{\mathcal{A}}^{[j+1]} = \vec{\Xi}\left[\vec{\mathcal{A}}^{[j]}\right]$, and finally take the limit.

DEFINITION I.5. *A class of combinatorial structures is said to be* constructible *iff it admits a (possibly recursive) specification in terms of sum, product, sequence, set, multiset, and cycle constructions.*

At this stage, we have therefore defined a specification language for combinatorial structures which is some fragment of set theory with recursion added. Each constructible class has by virtue of Theorem I.1 an ordinary generating function for which defining equations can be produced systematically. In fact, it is even possible to use computer algebra systems in order to compute it *automatically*! See [**56**] for the description of such a system.

COROLLARY I.1. *The generating function of a constructible class is a component of a system of generating function equations whose terms are built from*

$$1,\ z,\ +\ ,\ \times\ ,\ \Phi_{\mathfrak{S}},\ \Phi_{\mathfrak{C}},\ \Phi_{\mathfrak{M}},\ \Phi_{\mathfrak{P}},$$

where

$$\begin{cases} \Phi_{\mathfrak{S}}[f] &=& \dfrac{1}{1-f}, & \Phi_{\mathfrak{C}}[f] &=& \displaystyle\sum_{k=1}^{\infty} \frac{\varphi(k)}{k} \log \frac{1}{1-f(z^k)}, \\ \Phi_{\mathfrak{M}}[f] &=& \exp\left(\displaystyle\sum_{k=1}^{\infty} \frac{f(z^k)}{k} \right), & \Phi_{\mathfrak{P}}[f] &=& \exp\left(\displaystyle\sum_{k=1}^{\infty} (-1)^{k-1} \frac{f(z^k)}{k} \right). \end{cases}$$

Thus, iterative classes have explicit generating functions involving compositions of the basic operators only, while recursive structures have OGF's that are only accessible indirectly via systems of functional equations. As we see at various places in this chapter, the following classes are constructible: binary words, binary trees, general trees, integer partitions, integer compositions, nonplane trees, polynomials over finite fields, necklaces, and wheels.

For instance, the OGF of binary words corresponding to $\mathcal{W} = \mathfrak{S}(\mathcal{Z} + \mathcal{Z})$ is

$$W(z) = \frac{1}{1-2z},$$

whence the expected result that $W_n = 2^n$.

For the class $\mathcal{G}$ of general trees, constructibility leads to an equation defining $G(z)$ implicitly,

$$G(z) = \frac{z}{1-G(z)}.$$

From this point on, basic algebra does the rest. First the original equation is equivalent (in the ring of formal power series) to $G - G^2 - z = 0$. Next, the quadratic equation is solvable by radicals, and one finds

$$\begin{aligned} G(z) &= \tfrac{1}{2}\left(1 - \sqrt{1-4z}\right) \\ &= z + z^2 + 2\,z^3 + 5\,z^4 + 14\,z^5 + 42\,z^6 + 132\,z^7 + 429\,z^8 + \cdots \\ &= \sum_{n \geq 1} \frac{1}{n} \binom{2n-2}{n-1} z^n. \end{aligned}$$

(The conjugate root $\overline{G}(z)$ is to be discarded since it involves a term z^{-1} as well as negative coefficients; the expansion results from Newton's binomial theorem applied to $(1+x)^{1/2}$ at $x = -4z$.)

The numbers

$$\text{(10)} \qquad \mathrm{C}_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!\,n!} \qquad \text{with OGF} \quad \mathrm{C}(z) = \frac{1-\sqrt{1-4z}}{2z}$$

are known as the Catalan numbers (*EIS* **A000108**)[3] in the honour of Eugène Catalan (1814-1894), a French and Belgian mathematician who developed many of their properties. In

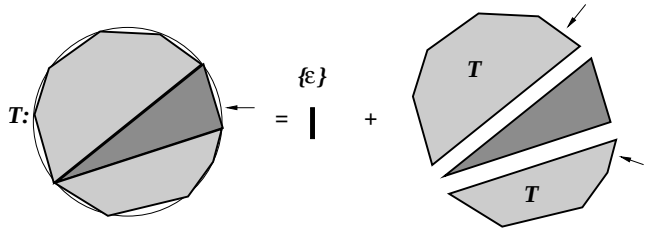
[3] Throughout this book, a reference like *EIS* **A000108** points to Sloane's *Encyclopedia of Integer Sequences* that is available in electronic form [**132**] or as a book by Sloane and Ploufe [**133**].

summary, *general trees are enumerated by Catalan numbers:*

$$G_n = \mathrm{C}_{n-1} \equiv \frac{1}{n}\binom{2n-2}{n-1}, \qquad \textit{where } \mathrm{C}_n \textit{ is a Catalan number.}$$

For this reason the term *Catalan tree* is often employed as synonymous to "general (rooted unlabelled plane) tree".

We can now conclude with the enumeration of triangulations, one of our three leading examples at the beginning of this chapter. Fix n points regularly spaced on a circle and conventionally numbered from 0 to $n-1$ (for instance the nth roots of unity). A triangulation is defined as a maximal decomposition of the regular n-gon into $n-2$ triangles; the size of the triangulation is taken as the number of triangles, that is, $n-2$. Given a triangulation, we define its "root" as a triangle chosen in some conventional and unambiguous manner (e.g., at the start, the triangle that contains the two smallest labels). Then, a triangulation decomposes into its root triangle and two subtriangulations (that may well be "empty") appearing on the left and right sides of the root triangle; the decomposition is illustrated by the following diagram (where the arrow points to a possible choice of roots):



The class $\mathcal{T}$ of all triangulations can be specified recursively as

$$\mathcal{T} = \{\epsilon\} + (\mathcal{T} \times \nabla \times \mathcal{T}),$$

provided that we consider a 2-gon (a diameter) as giving rise to an empty triangulation. Consequently, the OGF satisfies the equation $T = 1 + zT^2$ and

$$T(z) = \frac{1}{2z}\left(1 - \sqrt{1-4z}\right).$$

As a result, *triangulations are enumerated by Catalan numbers*:

$$T_n = \mathrm{C}_n \equiv \frac{1}{n+1}\binom{2n}{n}, \qquad \textit{where } \mathrm{C}_n \textit{ is a Catalan number.}$$

This particular result goes back to Euler and Segner (1753), a century before Catalan; see Figure 1 for first values and p. 48 for related bijections.

▷ **9.** *A variant specification of triangulations.* Consider the class $\mathcal{U}$ of "nonempty" triangulation of the n-gon, that is, we exclude the 2-gon and the coresponding "empty" triangulation of size 0. Then, $\mathcal{U} = \mathcal{T} \setminus \{\epsilon\}$ admits the specification

$$\mathcal{U} = \nabla + (\nabla \times \mathcal{U}) + (\mathcal{U} \times \nabla) + (\mathcal{U} \times \nabla \times \mathcal{U})$$

which also leads to the Catalan numbers via $U = z(1+U)^2$. ◁

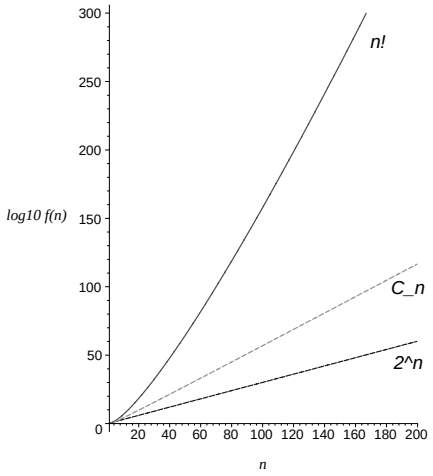


FIGURE 3. The growth regimes of three sequences $f(n) = 2^n,\ \mathrm{C}_n,\ n!$, with a plot of $\log_{10} f(n)$ versus n.

I. 2.4. Asymptotic interpretation of counting sequences. Even in simplest cases, counting sequences delivered by the symbolic method may not be too easy to interpret directly. On the other hand, from a quick glance at the table of initial values of W_n, P_n, T_n given in Eq. (2), it is apparent that W_n grows more slowly than T_n, which itself grows more slowly than P_n. The classification of growth rates of counting sequences belongs properly to asymptotic analysis, of which a thorough treatment is presented in Chapters III–V. Here, we content ourselves with a few remarks based on elementary real analysis. (The basic notations are described in APPENDIX: *Asymptotic Notation*, p. 166.)

The sequence $W_n = 2^n$ grows exponentially and, in such an extreme simple case, the exact form coincides with the asymptotic form. The sequence $P_n = n!$ must grow at a faster asymptotic regime. But how fast? The answer is provided by what is known as "Stirling's formula", that is, an approximation to the factorial numbers due to the Scottish mathematician James Stirling (1692–1770):

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n}\left(1 + O(\frac{1}{n})\right) \qquad (n \to +\infty). \tag{11}$$

This formula shows that the factorial numbers grow superexponentially fast, and in particular, grow much faster than W_n. The ratios of the exact values to Stirling's approximations

n:	1	2	5	10	100	1,000
$\frac{n!}{n^n e^{-n}\sqrt{2\pi n}}$:	1.084437	1.042207	1.016783	1.008365	1.000833	1.000083

shows an *excellent quality* of the asymptotic estimate: the error is only 8% for $n = 1$, less than 1% for $n = 10$, and less than 1 per thousand for any n greater than 100.

Stirling's formula in turn gives access to the asymptotic form of the Catalan numbers, by means of a simple calculation:

$$\mathrm{C}_n = \frac{1}{n+1}\frac{(2n)!}{(n!)^2} \sim \frac{1}{n}\frac{(2n)^{2n}e^{-2n}\sqrt{4\pi n}}{n^{2n}e^{-2n}2\pi n},$$

n	C_n	$C_n^\star$	$C_n^\star/C_n$
1	1	2.25	2.25675 83341 91025 14779 23178
10	16796	18707.89	1.11383 05127 52445 89437 89064
100	$0.89651 \cdot 10^{57}$	$0.90661 \cdot 10^{57}$	1.01126 32841 24540 52257 13957
1000	$0.20461 \cdot 10^{598}$	$0.20484 \cdot 10^{598}$	1.00112 51328 1542 41647 01282
10000	$0.22453 \cdot 10^{6015}$	$0.22456 \cdot 10^{6015}$	1.00011 25013 28127 92913 51406
100000	$0.17805 \cdot 10^{60199}$	$0.17805 \cdot 10^{60199}$	1.00001 12500 13281 25292 96322
1000000	$0.55303 \cdot 10^{602051}$	$0.55303 \cdot 10^{602051}$	1.00000 11250 00132 81250 29296

FIGURE 4. The Catalan numbers C_n, their Stirling approximation $\mathrm{C}_n^\star = 4^n/\sqrt{\pi n^3}$, and the ratio $\mathrm{C}_n^\star \,/\, \mathrm{C}_n$.

which simplifies to

$$\mathrm{C}_n \sim \frac{4^n}{\sqrt{\pi n^3}}. \tag{12}$$

Thus, the growth of Catalan numbers is roughly comparable to an exponential, 4^n, modulated by a "polynomial" factor, here $1/\sqrt{\pi n^3}$. A surprising consequence of this asymptotic estimate to the area of boolean function complexity appears in Example 12 below.

Altogether, the asymptotic number of general trees and triangulations is well summarized by a simple formula. Approximations become more and more accurate as n becomes large. Figure 1 exemplifies the quality of the approximation with subtler phenomena apparent on the figures and well explained by asymptotic theory. Such asymptotic formulæ then make comparison between the growth rates of sequences easy.

▷ **10.** *The complexity of coding.* A company specialized in computer aided design has sold to you a scheme that (they claim) can encode any triangulation of size $n \geq 100$ using at most $1.5n$ bits of storage. After reading these pages, what do you do? [Hint: sue them!] See also Ex. 21 for related coding arguments. ◁

▷ **11.** *Experimental asymptotics.* From the data of Figure 4, guess the value of $\mathrm{C}^\star_{10^7} \,/\, \mathrm{C}_{10^7}$ and of $\mathrm{C}^\star_{5\cdot 10^6} \,/\, \mathrm{C}_{5\cdot 10^6}$ to 25D. (See, e.g., [**89**] for related asymptotic expansions and [**22**] for similar properties.) ◁

The interplay between combinatorial structure and asymptotic structure is indeed the principal theme of this book. We shall see that a vast majority of the generating functions provided by the symbolic method, however complicated, lead to similarly simple asymptotic estimates.

I. 3. Integer compositions and partitions

This section and the next one provide first illustrations of the symbolic method and of counting via specifications. In this framework, generating functions are obtained with hardly any computation. At the same time, many counting refinements follow from a basic combinatorial construction. The most direct applications described here relate to the additive decomposition of integers into summands with the classical combinatorial-arithmetic structures of partitions and compositions. The specifications are iterative and they simply combine two levels of constructions of type $\mathfrak{S}, \mathfrak{C}, \mathfrak{M}, \mathfrak{P}$.

I. 3.1. Compositions and partitions. First the definitions:

DEFINITION I.6. *A composition of an integer n is a sequence $(x_1, x_2, \ldots, x_k)$ of integers (for some k) such that*

$$n = x_1 + x_2 + \cdots + x_k, \qquad x_j \geq 1.$$

A partition of an integer n is a sequence $(x_1, x_2, \ldots, x_k)$ of integers (for some k) such that

$$n = x_1 + x_2 + \cdots + x_k \qquad \textit{and} \qquad x_1 \geq x_2 \geq \cdots \geq x_k.$$

In both cases, the x_i's are called the summands or the parts and the quantity n is called the size of the composition or the partition.

Graphically, compositions may be seen as as "ragged-landscapes" (represent the summands vertically) or equivalently as alignments of balls with dividing lines, the "balls-and-bars" model; in contrast, partitions appear as "staircases" also known as Ferrers diagrams [**28**, p. 100]; see Figure 5. We let $\mathcal{C}$ and $\mathcal{P}$ denote the class of of all compositions and all partitions. Since a set can always be presented in sorted order, the difference between compositions and partitions lies in the fact that the order of summands *does* or *does not* matter. This is reflected by the use of a sequence construction (for $\mathcal{C}$) against a multiset construction (for $\mathcal{P}$). In this perspective, it proves convenient to regard 0 as obtained by the empty sequence of summands ($k = 0$), and we shall do so from now on.

First, let $\mathcal{I} = \{1, 2, \ldots\}$ denote the combinatorial class of all integers at least 1 (the summands), and let the size of each integer be its value. Then, the OGF of $\mathcal{I}$ is

$$I(z) = \sum_{n \geq 1} z^n = \frac{z}{1-z}, \tag{13}$$

since $I_n = 1$ for $n \geq 1$, corresponding to the fact that there is exactly one object in $\mathcal{I}$ for each size $n \geq 1$. If integers are represented in unary, say by small balls, one has,

$$\mathcal{I} = \{1,\ 2,\ 3,\ \ldots\} = \{\bullet,\ \bullet\bullet,\ \bullet\bullet\bullet,\ \ldots\} \cong \mathfrak{S}_{\geq 1}\{\bullet\}, \tag{14}$$

which is another way to view the equality $I(z) = z/(1-z)$.

From their definition, the classes $\mathcal{C}$ and $\mathcal{P}$ can then be specified as

$$\mathcal{C} = \mathfrak{S}\{\mathcal{I}\}, \qquad \mathcal{P} = \mathfrak{M}\{\mathcal{I}\}. \tag{15}$$

In sequences, the order of components is taken into accout, which precisely models compositions. In multisets, order is not taken into account (while repetitions are allowed), so that we do have an adequate specification of partitions. In both cases, size is correctly inherited additively from summands.

FIGURE 5. Graphical representations of compositions and partitions: (left) the composition $1 + 3 + 1 + 4 + 2 + 3 = 14$ with its "ragged-landscape" and "balls-and-bars" models; (right) the partition $8+8+6+5+4+4+4+2+1+1 = 43$ with its staircase (Ferrers diagram) model.

0	1	1
10	512	42
20	524288	627
30	536870912	5604
40	549755813888	37338
50	562949953421312	204226
60	576460752303423488	966467
70	590295810358705651712	4087968
80	604462909807314587353088	15796476
90	618970019642690137449562112	56634173
100	633825300114114700748351602688	190569292
110	649037107316853453566312041152512	607163746
120	664613997892457936451903530140172288	1844349560
130	680564733841876926926749214863536422912	5371315400
140	696898287454081973172991196020261297061888	15065878135
150	713623846352979940529142984724747568191373312	40853235313
160	730750818665451459101842416358141509827966271488	107438159466
170	748288838313422294120286634350736906063837462003712	274768617130
180	766247770432944429179173513575154591809369561091801088	684957390936
190	784637716923335095479473677900958302012794430558004314112	1667727404093
200	803469022129495137770981046170581301261101496891396417650688	3972999029388
210	822752278660603021077484591278675252491367932816789931674304512	9275102575355
220	842498333348457493583344221469363458551160763204392890034487820288	21248279009367
230	862718293348820473429344482784628181556388621521298319395315527974912	47826239745920
240	883423532389192164791648750371459257913741948437809479060803100646309888	105882246722733
250	904625697166532776746648320380374280103671755200316906558262375061821325312	230793554364681

FIGURE 6. For $n = 0, 10, 20, \ldots, 250$ (left), the number of compositions C_n (middle) and the number of partitions (right). The figure illustrates the difference in growth between $C_n = 2^{n-1}$ and $P_n = e^{O(\sqrt{n})}$.

First, the specification $\mathcal{C} = \mathfrak{S}\{\mathcal{I}\}$ admits, by Theorem I.1, a direct translation into OGF:

$$C(z) = \frac{z}{1 - I(z)}. \tag{16}$$

The collection of equations (13), (16) thus fully determines $C(z)$:

$$\begin{aligned} C(z) &= \frac{1}{1 - \frac{z}{1-z}} = \frac{1-z}{1-2z} \\ &= 1 + z + 2z^2 + 4z^3 + 8z^4 + 16z^5 + 32z^6 + \cdots . \end{aligned}$$

From there, the counting problem for compositions is solved by a straightforward expansion of the OGF: one has

$$C(z) = \left(\sum_{n \geq 0} 2^n z^n \right) - \left(\sum_{n \geq 0} 2^n z^{n+1} \right),$$

implying

$$C_n = 2^{n-1}, \quad n \geq 1; \qquad C_0 = 1.$$

(Naturally, the C_n bear no relation to the Catalan numbers, C_n.) This agrees with basic combinatorics since a composition of n can be viewed as the placement of $n-1$ separation bars between n aligned balls (the "balls and bars" model of Figure 5), of which there are clearly 2^{n-1} possibilities.

Next, the form of the partition generating function derives from Theorem I.1; the general translation mechanism provides the relation

$$P(z) = \exp\left(I(z) + \frac{1}{2} I(z^2) + \frac{1}{3} I(z^3) + \cdots \right) \quad \text{with} \quad I(z) = \frac{z}{1-z}. \tag{17}$$

In a special case like this, it is just as easy, however, to appeal directly to the product representation and get the more familiar form

$$\begin{aligned} P(z) &= \prod_{m=1}^{\infty} \frac{1}{1 - z^m} \\ &= 1 + z + 2\,z^2 + 3\,z^3 + 5\,z^4 + 7\,z^5 + 11\,z^6 + 15\,z^7 + 22\,z^8 + 30\,z^9 + \cdots . \end{aligned} \tag{18}$$

	Spec.	OGF	coeff.	asympt.
Composition	$\mathfrak{S}\{\mathfrak{S}_{\geq 1}\{Z\}\}$	$\frac{1-z}{1-2z}$	2^{n-1}	$\frac{1}{2}2^n$
—, sum. $\leq r$	$\mathfrak{S}\{\mathfrak{S}_{1\,..\,r}\{Z\}\}$	$\frac{1-z}{1-2z+z^{r+2}}$	Eq. (19)	$c_r \rho_r^{-n}$
—, k sum.	$\mathfrak{S}_k\{\mathfrak{S}_{\geq 1}\{Z\}\}$	$\frac{z^k}{(1-z)^k}$	$\binom{n-1}{k-1}$	$\frac{n^{k-1}}{(k-1)!}$
Partitions	$\mathfrak{M}\{\mathfrak{S}_{\geq 1}\{Z\}\}$	$\prod_{m=1}^{\infty}(1-z^m)^{-1}$	—	$\frac{1}{4n\sqrt{3}}e^{\pi\sqrt{\frac{2n}{3}}}$
—, sum. $\leq r$	$\mathfrak{M}\{\mathfrak{S}_{1\,..\,r}\{Z\}\}$	$\prod_{m=1}^{r}(1-z^m)^{-1}$	—	$\frac{n^{r-1}}{r!(r-1)!}$
—, $\leq k$ sum.	$\cong \mathfrak{M}\{\mathfrak{S}_{1\,..\,k}\{Z\}\}$	$\prod_{m=1}^{k}(1-z^m)^{-1}$	—	$\frac{n^{k-1}}{k!(k-1)!}$
Cyclic comp.	$\mathfrak{C}\{\mathfrak{S}_{\geq 1}\{Z\}\}$	Eq. (23)	Eq. (24)	$\frac{2^n}{n}$
Part., distinct sum.	$\mathfrak{P}\{\mathfrak{S}_{\geq 1}\{Z\}\}$	$\prod_{m=1}^{\infty}(1+z^m)$	—	$\frac{3^{3/4}}{12n^{3/4}}e^{\pi\sqrt{\frac{n}{3}}}$

FIGURE 7. Partitions and compositions: specifications, generating functions, counting sequences, and asymptotic approximation.

Contrary to compositions that are counted by the explicit formula 2^{n-1}, so simple form exists for p_n. Asymptotic analysis of the OGF (17) based on the saddle point shows that $P_n = e^{O(\sqrt{n})}$. In fact a very famous theorem of Hardy and Ramanujan later improved by Rademacher, see [**4**], provides a full expansion of which the asymptotically dominant term is

$$P_n \sim \frac{1}{4n\sqrt{3}} \exp\left(\pi\sqrt{\frac{2n}{3}}\right).$$

There are consequently much fewer partitions than compositions (Figure 6).

▷ **12.** *A recurrence for the partition numbers.* Logarithmic differentiation gives

$$z\frac{P'(z)}{P(z)} = \sum_{n=1}^{\infty}\frac{nz^n}{1-z^n} \qquad \text{implying} \quad nP_n = \sum_{j=1}^{n-1}\sigma(j)P_{n-j},$$

where $\sigma(n)$ is the sum of the divisors of n (e.g., $\sigma(6) = 1+2+3+6 = 12$). Consequently, $P_1, \ldots, P_N$ can be computed in $O(N^2)$ integer-arithmetic operations. (The technique is generally applicable to powersets and multisets; see also Ex. 33. Ex. 18 further lowers the bound in the case of partitions to $O(N\sqrt{N})$.) ◁

When considering variations of the scheme (14), a number of counting results follow rather straightfowardly. We discuss below the case of compositions and partitions with restricted summands, as well as with a fixed number of parts. First, we state:

PROPOSITION I.1. *Let $\mathcal{T} \subseteq \mathcal{I}$ be a subset of the positive integers. The OGF of the classes $\mathcal{C}^{\mathcal{T}} := \mathfrak{S}\{\mathfrak{S}_{\mathcal{T}}\{\mathcal{Z}\}\}$ and $\mathcal{P}^{\mathcal{T}} := \mathfrak{M}\{\mathfrak{S}_{\mathcal{T}}\{\mathcal{Z}\}\}$ of compositions and partitions having summands restricted to $\mathcal{T}$ is given by*

$$C^{\mathcal{T}}(z) = \frac{1}{1-\sum_{n\in T} z^n} = \frac{1}{1-T(z)}, \qquad P^{\mathcal{T}}(z) = \prod_{n\in\mathcal{T}}\frac{1}{1-z^n}.$$

PROOF. The statement results directly from Theorem I.1. □

EXAMPLE 1. *Compositions with restricted summands.* In order to enumerate the class $\mathcal{C}^{\{1,2\}}$ of compositions of n whose parts are only allowed to be taken from the set $\{1,2\}$, simply write

$$\mathcal{C}^{\{1,2\}} = \mathfrak{S}\{\mathcal{I}^{\{1,2\}}\} \qquad \text{with } \mathcal{I}^{\{1,2\}} = \{1,2\}.$$

Thus, in terms of generating functions, the relation

$$C^{\{1,2\}}(z) = \frac{1}{1 - I^{\{1,2\}}(z)}$$

holds (see Eq. (16)), with

$$I^{\{1,2\}}(z) = z + z^2.$$

Then,

$$C^{\{1,2\}}(z) = \frac{1}{1 - z - z^2} = 1 + z + 2z^2 + 3z^3 + 5z^4 + +8z^5 + 13z^6 + \cdots$$

and the number of compositions of n in this class is expressed by a Fibonacci number,

$$C_n^{\{1,2\}} = \mathrm{F}_{n+1} \text{ where } \mathrm{F}_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right].$$

In particular, the rate of growth is of the exponential type φ^n, where

$$\varphi := \frac{1+\sqrt{5}}{2}$$

is the golden ratio.

Similarly, compositions such that all their summands lie in the set $\{1, 2, \ldots, r\}$ have generating function

$$C^{\{1,\ldots,r\}}(z) = \frac{1}{1 - z - z^2 - \cdots z^r} = \frac{1}{1 - z\frac{1-z^r}{1-z}} = \frac{1-z}{1 - 2z + z^{r+1}},$$

and the corresponding counts are given by generalized Fibonacci numbers. A double combinatorial sum expresses these counts

$$C_n^{\{1,\ldots,r\}} = [z^n] \sum_j \left(\frac{z(1-z^r)}{(1-z)}\right)^j = \sum_{j,k} (-1)^k \binom{j}{k} \binom{n - rk - 1}{j-1}. \tag{19}$$

Asymptotically, for any fixed r, one checks that there is a unique root ρ_r of the denominator $1 - 2z + z^{r+1}$ in $(\frac{1}{2}, 1)$, that this root dominates all the other roots, and that it is simple. Consequently, one has

$$C_n^{\{1,\ldots,r\}} \sim c_r \rho_r^{-n} \quad \text{for fixed } r \text{ as } n \to \infty. \tag{20}$$

The quantity ρ_r plays a rôle similar to that of the golden ratio when $r = 2$. Details of the asymptotic analysis are discussed in Chapter 4. □

▷ **13.** *Compositions into primes.* The additive decomposition of integers into primes is still surrounded with mystery. For instance, it is not known whether every even number is the sum of two primes (Goldbach's conjecture). However, the number of compositions of n into prime summands (*any* number of summands is permitted) is $B_n = [z^n]B(z)$ where

$$\begin{aligned} B(z) &= \left(1 - \sum_{p \text{ prime}} z^p\right)^{-1} = \left(1 - z^2 - z^3 - z^5 - z^7 - z^{11} - \cdots\right)^{-1} \\ &= 1 + z^2 + z^3 + z^4 + 3\,z^5 + 2\,z^6 + 6\,z^7 + 6\,z^8 + 10\,z^9 + 16\,z^{10} + \cdots \end{aligned}$$

(*EIS* **A023360**) and complex asymptotic method make it *easy* from there to determine the asymptotic form $B_n \sim 0.30365 \cdot 1.47622^n$; see Chapter 4. ◁

EXAMPLE 2. *Partitions with resricted summands and denumerants.* Whenever summands are restricted to a finite set, the special partitions that result are called denumerants. A popular denumerant problem consists in finding the number of ways of giving change of 99 cents using coins that are pennies (1 ¢), nickels (5 ¢), dimes (10 ¢) and quarters (25 ¢). (The order in which the coins are taken does not matter and repetitions are allowed.) For the case of a finite $\mathcal{T}$, we predict from Proposition 2 that $P^{\mathcal{T}}(z)$ is always a *rational* function with poles that are at roots of unity; also the $P_n^{\mathcal{T}}$ satisfy a linear recurrence related to the structure of $\mathcal{T}$. The solution to the original coin change problem is found to be

$$[z^{99}]\,\frac{1}{(1-z)(1-z^5)(1-z^{10})(1-z^{25})} = 242.$$

In the same vein, one proves [**28**, p. 108] that

$$P_n^{\{1,2\}} = \lceil \frac{2n+3}{4} \rfloor \qquad P_n^{\{1,2,3\}} = \lceil \frac{(n+3)^2}{12} \rfloor.$$

There $\lceil x \rfloor \equiv \lfloor x + \frac{1}{2} \rfloor$ denotes the integer closest to the real number x. Such results are typically obtained by the two step process: (i) decompose the rational generating function into simple fractions; (ii) compute the coefficients of each simple fraction and combine them to get the final result [**28**, p. 108].

The general argument also gives the generating function of partitions whose summands lie in the set $\{1, 2, \ldots, r\}$ as

$$P^{\{1,\ldots,r\}}(z) = \prod_{m=1}^{r} \frac{1}{1-z^m}. \tag{21}$$

In other words, we are enumerating partitions according to the value of the largest summand. One then has by looking at the poles

$$P_n^{\{1,\ldots,r\}} \sim c_k n^{k-1} \text{ with } c_k = \frac{1}{k!(k-1)!}.$$

A similar argument provides the asymptotic form of $P_n^{\mathcal{T}}$ when $\mathcal{T}$ is an arbitrary finite set:

$$P_n^{\mathcal{T}} \sim \frac{1}{\tau}\frac{n^{r-1}}{(r-1)!} \qquad \text{with } \tau := \prod_{n\in\mathcal{T}} n, \quad r := \text{card}(\mathcal{T}).$$

This result is due to Schur and is proved in Chapter IV. □

We next examine the statistic of the number of summands. Let $\mathcal{C}^{(k)}$ denote the class of compositions made of k summands, k a fixed integer ≥ 1. One has

$$\mathcal{C}^{(k)} = \mathcal{I} \times \mathcal{I} \times \cdots \times \mathcal{I},$$

where the number of terms in the cartesian product is k, and $\mathcal{I}$ still represents the summands, i.e., the class of positive integers. From there, the corresponding generating function is found to be

$$C^{(k)} = \big(I(z)\big)^k \qquad \text{with} \qquad I(z) = \frac{z}{1-z}.$$

The number of compositions of n having k parts is thus

$$C_n^{(k)} = [z^n]\frac{z^k}{(1-z)^k} = \binom{n-1}{k-1},$$

a result which constitutes a combinatorial refinement of $C_n = 2^{n-1}$. Note that the formula $C_n^{(k)} = \binom{n-1}{k-1}$ also results directly from the "balls and bars" model of compositions (Figure 5).

Partitions, are naturally represented as collections of points (the staircase model of Figure 5) in the $\mathbb{N} \times \mathbb{N}$ lattice. A geometric symmetry around the main diagonal (also known in the specialized literature as conjugation) exchanges number of summands and value of largest summand, so that the OGF $P^{(\leq k)}(z)$ of partitions with at most k summands coincides with the OGF of partitions with summands all at most k already enumerated in (21)

$$P^{(\leq k)}(z) \equiv P^{\{1,\ldots,k\}} = \prod_{m=1}^{k} \frac{1}{1-z^m}; \tag{22}$$

consequently the OGF of partitions with *exactly* k summands, $P^{(k)}(z) = P^{(\leq k)}(z) - P^{(\leq k-1)}(z)$, evaluates to

$$P^{(k)}(z) = \frac{z^k}{(1-z)(1-z^2)\cdots(1-z^k)}.$$

▷ **14.** *Compositions with summands bounded in number and size.* The number of compositions of size n with k summands each at most r is

$$[z^n] \left(z \frac{1-z^r}{1-z} \right)^k,$$

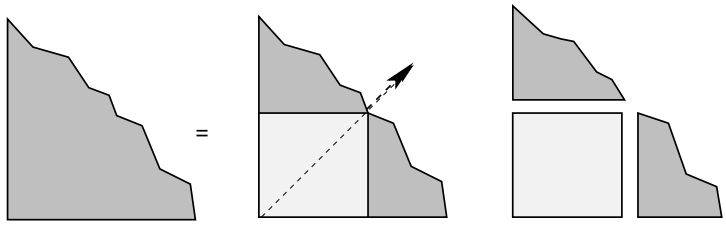
and is expressible as a simple binomial convolution. ◁

▷ **15.** *Partitions with summands bounded in number and size.* The number of partitions of size n with at most k summands each at most ℓ is

$$[z^n] \frac{(1-z)(1-z^2)\cdots(1-z^{k+\ell})}{((1-z)(1-z^2)\cdots(1-z^k))\cdot((1-z)(1-z^2)\cdots(1-z^\ell))}.$$

(The verification by recurrence is easy.) The GF reduces to the binomial coefficient $\binom{k+\ell}{k}$ as $z \to 1$; it is known as a Gaussian binomial coefficient, denoted $\binom{k+\ell}{k}_z$, or a "q-analogue" of the binomial coefficient [**4, 28**]. ◁

The last problem of this section exemplifies the close interplay betwen combinatorial decompositions and special function identities, which constitutes a recurrent theme of classical combinatorial analysis. The diagram of any partition contains a uniquely determined square (the "Durfee square") that is maximal, as exemplified by the following diagram:



This decomposition gives the identity

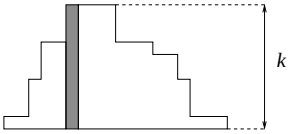
$$\prod_{n=1}^{\infty} \frac{1}{1-z^n} = \sum_{k \geq 0} \frac{z^{k^2}}{((1-z)\cdots(1-z^k))^2},$$

expressing, via (21) and (22), the combinatorial isomorphism (k is the size of the Durfee square)

$$\mathcal{P} \cong \bigcup_{k \geq 0} \left(\mathcal{Z}^{k^2} \times \mathcal{P}^{(\leq k)} \times \mathcal{P}^{\{1,\ldots,k\}} \right),$$

itself nothing but a formal rewriting of the geometric decomposition. As time goes, we shall make greater and greater use of such "direct" translations of object descriptions into generating function equations.

▷ **16.** *Stack polyominos.* These are diagrams of compositions such that for some j, one has $1 \leq x_1 \leq x_2 \leq \cdots \leq x_j \geq x_{j+1} \geq \cdots \geq x_k \geq 1$. The diagram representation of stack polyominos,



$$\longleftrightarrow \quad \mathcal{P}^{\{1,\ldots,k-1\}} \times \mathcal{Z}^k \times P^{\{1,\ldots,k\}}$$

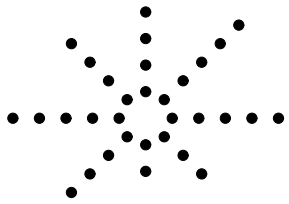
translates immediately into the OGF

$$S(z) = \sum_{k\geq 1} \frac{z^k}{1-z^k} \frac{1}{\left((1-z)(1-z^2)\cdots(1-z^{k-1})\right)^2},$$

once use is made of the partition GFs $P^{\{1,\ldots,k}(z)$ of (21). The book of van Rensburg [**144**] describes many such constructions and their relation to certain models of statistical physics. ◁

I. 3.2. Integer related constructions. Finally, we say a few words about the two constructions of cycle and powerset that haven't been yet applied to $\mathcal{I}$. First, the class $\mathcal{D} = \mathfrak{C}\{I\}$ comprises *cyclic compositions*, that is, compositions defined up to circular shift; so, for instance $2+3+1+2+5, 3+1+2+5+2$, etc, are identified. Alternatively, we may view elements of $\mathcal{D}$ as "wheels" composed of circular arrangements of segments (taken up to circular symmetry).

A "wheel" (cyclic composition):



By the cycle construction, the OGF is

(23)
$$\begin{aligned} D(z) &= \sum_{k=1}^{\infty} \frac{\varphi(k)}{k} \log\left(1 - \frac{z^k}{1-z^k}\right)^{-1} = \sum_{k=1}^{\infty} \frac{\varphi(k)}{k} \left(\log(1-z^k) - \log(1-2z^k)\right) \\ &= z + 2\,z^2 + 3\,z^3 + 5\,z^4 + 7\,z^5 + 13\,z^6 + 19\,z^7 + 35\,z^8 + \cdots . \end{aligned}$$

The coefficients are thus (*EIS* **A008965**)

$$D_n = \frac{1}{n} \sum_{k \mid n} \varphi(k)(2^{n/k} - 1) \equiv -1 + \frac{1}{n} \sum_{k \mid n} \varphi(k) 2^{n/k} \sim \frac{2^n}{n}. \tag{24}$$

(Notice that D_n is of the same asymptotic order as $\frac{1}{n}C_n$, which is suggested by circular symmetry of wheels, but $D_n \sim C_n/(2n)$.)

More interestingly perhaps, the class $\mathcal{Q} = \mathfrak{P}\{I\}$ is the subclass of $\mathcal{P} = \mathfrak{M}\{I\}$ corresponding to *partitions into distinct summands*: these are determined like in Definition I.6 but with the strict inequalities $x_k > \cdots > x_1$, so that the OGF is

$$Q(z) = \prod_{n\geq 1} (1 + z^n).$$

The coefficients are not amenable to closed from. However the saddle point method (Chapter 6) yields the approximation:

$$Q_n \sim \frac{3^{3/4}}{12n^{3/4}} \exp\left(\pi\sqrt{\frac{n}{3}}\right), \tag{25}$$

which has a shape similar to that of P_n.

▷ **17.** *Odd versus distinct summands.* The partitions of n into odd summands ($\mathcal{O}_n$) and into distinct summands ($\mathcal{Q}_n$) are equinumerous. Indeed, one has

$$Q(z) = \prod_{m=1}^{\infty}(1+z^m), \qquad O(z) = \prod_{j=0}^{\infty}(1-z^{2j+1})^{-1}.$$

Equality results from substituting $(1+a) = (1-a^2)/(1-a)$ with $a = z^m$,

$$Q(z) = \frac{\mathbf{1-z^2}}{1-z}\,\frac{\mathbf{1-z^4}}{\mathbf{1-z^2}}\,\frac{\mathbf{1-z^6}}{1-z^3}\,\frac{\mathbf{1-z^8}}{\mathbf{1-z^4}}\,\frac{\mathbf{1-z^{10}}}{1-z^5}\cdots = \frac{1}{1-z}\,\frac{1}{1-z^3}\,\frac{1}{1-z^5}\cdots,$$

and simplification of the numerators with half of the denominators (in boldface). ◁

Let $\mathcal{I}^{\mathrm{pow}} = \{1,2,4,8,\ldots\}$ be the set of powers of 2. The corresponding $\mathcal{P}$ and $\mathcal{Q}$ partitions have OGFs

$$\begin{aligned} P^{\mathrm{pow}}(z) &= \prod_{j=0}^{\infty}\frac{1}{1-z^{2^j}} \\ &= 1+z+2\,z^2+2\,z^3+4\,z^4+4\,z^5+6\,z^6+6\,z^7+10\,z^8+10\,z^9+\cdots \\ Q^{\mathrm{pow}}(z) &= \prod_{j=0}^{\infty}(1+z^{2^j}) \\ &= 1+z+z^2+z^3+z^4+z^5+\cdots. \end{aligned}$$

The first sequence $1,1,2,2,\ldots$ is the "binary partition sequence" (*EIS* **A018819**); the difficult asymptotic analysis was performed by de Bruijn [**34**] who obtained an estimate that involves subtle fluctuations and is of the global form $e^{O(\log^2 n)}$. The function $Q^{\mathrm{pow}}(z)$ reduces to $(1-z)^{-1}$ since every number has a unique additive decomposition into powers of 2. Accordingly, the identity

$$\frac{1}{1-z} = \prod_{j=0}^{\infty}(1+z^{2^j})$$

first observed by Euler is sometimes nicknamed the "computer scientist's identity" as it expresses the fact that every number admits a unique binary representation.

There exists a rich set of identities satisfied by partition generating functions—this fact owes to deep connections with elliptic functions, modular forms, and q-analogues of special functions on the one hand, basic combinatorics and number theory on the other hand. See [**4, 28**] for an introduction to this fascinating subject.

▷ **18.** *Euler's pentagonal number theorem.* This famous identity expresses $1/P(z)$ as

$$\prod_{n\geq 1}(1-z^n) = \sum_{k\in\mathbb{Z}}(-1)^k z^{k(3k+1)/2}.$$

It is proved formally and combinatorially in [**28**, p. 105]. As a consequence, the numbers $\{P_j\}_{j=0}^{N}$ can be determined in $O(N\sqrt{N})$ arithmetic operations. ◁

▷ **19.** *Lattice points.* The number of lattice points with integer coordinates that belong to the closed ball of radius n in d–dimensional space is

$$[z^{n^2}]\frac{1}{1-z}(\Theta(z))^d \qquad \text{where} \qquad \Theta(z)=1+2\sum_{n=1}^{\infty} z^{n^2}.$$

(Such OGF's are useful in cryptography [**93**] and estimates may be obtained from the saddle point method.) ◁

I. 4. Words and regular languages

First a finite *alphabet* $\mathcal{A}$ whose elements are called *letters* is fixed. Each letter is taken to have size 1, *i.e.*, it is an atom. A *word* is then any finite sequence of letters, usually written without separators. So, for us, with the choice of the latin alphabet ($\mathcal{A} = \{\mathsf{a},\ldots,\mathsf{z}\}$), sequences written as ygololihq, philology, zgrmblglps are words. The set of all words (often written as $\mathcal{A}^\star$ in formal linguistics) will be consistently denoted by $\mathcal{W}$ here. Following a well-established tradition in theoretical computer science and formal linguistics, any subset of $\mathcal{W}$ is called a *language* (or formal language, when the distinction with natural languages has to be made).

From the definition of the set of words $\mathcal{W}$, one has

$$\mathcal{W} \cong \mathfrak{S}\{\mathcal{A}\} \quad \text{implying} \quad W(z)=\frac{1}{1-mz}, \tag{26}$$

where m is the cardinality of the alphabet, *i.e.*, the number of letters. The generating function gives us (in an admittedly devious way) the counting result

$$W_n = m^n.$$

As is usual with symbolic methods, many enumerative consequences usually result from a given construction, and it is precisely the purpose of this section to examine some of them.

We shall introduce two frameworks that each have great expressive power to describe languages. The first one is iterative (*i.e.*, nonrecursive) and it bases itself on "regular specifications" that only involve sums, products, and sequences; the other one that is recursive (but of a very simple form) is best conceived of in terms of finite automata and is equivalent to linear systems of equations. It turns out that both frameworks determine the same family of languages, the *regular languages*, though the equivalence is nontrivial, and each particular problem usually admits a preferred representation. The resulting GFs are invariably rational functions.

I. 4.1. Regular specifications. Consider first words (or strings) over the binary alphabet $\mathcal{A}=\{a,b\}$. There is an alternative way to construct binary strings. It is based on the observation that (with a minor adjustment at the beginning) a string decomposes into a succession of "blocks" each formed with a single b followed by an arbitrary (possibly empty) sequence of a's. For instance $aaabaababaabbabbaaa$ decomposes as

$$aaa \,||\, baa \mid ba \mid baa \mid b \mid ba \mid b \mid baaa.$$

Omitting redundant[4] symbols, we have the alternative decomposition:

$$\mathcal{W} \cong \mathfrak{S}\{a\}\,\mathfrak{S}\{b\,\mathfrak{S}\{a\}\}. \tag{27}$$

[4]As is usual when dealing with words, we omit writing explicitly redundant braces '$\{,\}$' and cartesian products '$\times$'. Thus, for instance, $\mathfrak{S}\{a+b\}$ and $\{ab\}$ are shorthand notations for $\mathfrak{S}\{\{a\}+\{b\}\}$ and $\{\{a\}\times\{b\}\}$.

A check is provided by computing the OGF corresponding to this new specification,

$$W(z) = \frac{1}{1-z}\frac{1}{1-z\frac{1}{1-z}}, \tag{28}$$

which reduces to $(1-2z)^{-1}$ as it should.

The interest of the decomposition just seen is to take into account various other interesting properties, for example longest runs. Denote by $a^{<k} := \mathfrak{S}_{<k}\{a\}$ the collection of all words formed with the letter a only and whose length is between 0 and $k-1$; the corresponding OGF is $1+z+z^2+\cdots+z^{k-1} = (1-z^k)/(1-z)$. The collection $\mathcal{W}^{\langle k\rangle}$ of words which do not have k consecutive a's is described by an amended form of (27), namely

$$\mathcal{W}^{\langle k\rangle} = a^{<k}\,\mathfrak{S}\{ba^{<k}\}. \tag{29}$$

The corresponding OGF obtains immediately from (29)

$$W^{\langle k\rangle}(z) = \frac{1-z^k}{1-z}\cdot\frac{1}{1-z\frac{1-z^k}{1-z}} = \frac{1-z^k}{1-2z+z^{k+1}}.$$

This is therefore the generating functions of words whose longest run of consecutive a's is of length $< k$. From this computation and some asymptotic analysis, it can be deduced that the longest run of a's in a random binary string of length n is about $\log_2 n$. Such asymptotic aspects will be further explored in later chapters.

▷ **20.** *Runs in arbitrary alphabets.* For an alphabet of cardinality m, the quantity

$$\frac{1-z^k}{1-mz+(m-1)z^{k+1}}$$

is the OGF of words without k consecutive occurrences of a designated letter. ◁

The case of longest runs exemplifies the expressive power of nested constructions involving sequences. We set:

DEFINITION I.7. *An iterative specification that only involves atoms (e.g., letters of a finite alphabet $\mathcal{A}$) together with combinatorial sums, cartesian products, and sequence constructions is said to be a* regular specification.

*A language $\mathcal{L}$ is said to be S-*regular *(specification-regular) if there exists a regular specification $\mathcal{R}$ such that $\mathcal{L}$ and $\mathcal{R}$ are combinatorially isomorphic, $\mathcal{L} \cong \mathcal{R}$.*

It is a non-trivial fact that the notion of S-regularity introduced here coincides with the usual notion of regularity in formal language theory. See APPENDIX: *Regular languages*, p. 171 for explanations. From the definition and the basic theorem regarding admissibility (Theorem I.1), one has immediately:

PROPOSITION I.2. *Any S-regular language has an OGF that is a* rational function. *This OGF is obtained from a regular specification of the language by translating each letter into the variable z, disjoint unions into sums, cartesian products into products, and sequences into quasi-inverses, $(1-\cdot)^{-1}$.*

This result is technically shallow but its importance derives from the fact that regular languages have great expressive power devolving from their rich closure properties as well as their relation to finite automata discussed in the next subsection.

EXAMPLE 3. *Combinations and spacings.* The specification $\mathcal{L} = \mathfrak{S}\{a\}\,(b\mathfrak{S}\{a\})^k$ describes unambiguously the set of words that contain exactly k occurrences of the letter b.

The OGF is $L(z) = z^k/(1-z)^{k+1}$, and the number of words in the language satisfies

$$L_n = [z^n]\frac{z^k}{(1-z)^{k+1}} = \binom{n}{k}.$$

Each word of length n is characterized by the positions of its letters b, which means the choices of k positions amongst n possible ones. Formal language theory thus gives us back the well-known count of combinations by binomial coefficients.

Let $\binom{n}{k}_{<d}$ be the number of combinations of k elements amongst $[1, n]$ with constrained spacings: no element can be at distance d or more from another element. The refinement

$$\mathcal{L}^{[d]} = \mathfrak{S}\{a\}\,(b\mathfrak{S}_{<d}\{a\})^{k-1}\,(b\mathfrak{S}\{a\})$$

provides the generating function

$$\sum_{n\geq 0}\binom{n}{k}_{<d} z^n = \frac{z^k(1-z^d)^{k-1}}{(1-z)^{k+1}},$$

which is equivalent to a binomial convolution expression for $\binom{n}{k}_{<d}$. (This problem is clearly analogous to compositions with bounded summands.) □

EXAMPLE 4. *Double run statistics.* By forming maximal groups of equal letters in words, one finds easily that, for a binary alphabet,

$$\mathcal{W} = \mathfrak{S}\{b\}\,\mathfrak{S}\{a\mathfrak{S}\{a\}\,b\mathfrak{S}\{b\}\}\,\mathfrak{S}\{a\}.$$

Let $\mathcal{W}^{\langle\alpha,\beta\rangle}$ be the class of all words that have at most α consecutive a's and at most β consecutive b's. The specification of $\mathcal{W}$ produces a specification of $\mathcal{W}^{\langle\alpha,\beta\rangle}$, upon replacing $\mathfrak{S}\{a\}, \mathfrak{S}\{b\}$ by $\mathfrak{S}_{<\alpha}\{a\}, \mathfrak{S}_{<\beta}\{b\}$ internally, and by $\mathfrak{S}_{\leq\alpha}\{a\}, \mathfrak{S}_{\leq\beta}\{b\}$ externally. In particular, the OGF of binary words that never have more than r consecutive equal letters is found to be (set $\alpha = \beta = r$)

$$W^{\langle r,r\rangle} = \frac{(1-z^{r+1})^2}{1-2z+2z^{r+2}-z^{2r+2}} \tag{30}$$

Révész in [**121**] tells the following amusing story attributed to T. Varga: " A class of high school children is divided into two sections. In one of the sections, each child is given a coin which he throws two hundred times, recording the resulting head and tail sequence on a piece of paper. In the other section, the children do not receive coins, but are told instead that they should try to write down a 'random' head and tail sequence of length two hundred. Collecting these slips of paper, [a statistician] then tries to subdivide them into their original groups. Most of the time, he succeeds quite well."

The statistician's secret is to determine the probability distribution of the maximum length of runs of consecutive letters in a random binary word of length n (here $n = 200$). The probability of this parameter to equal k is

$$\frac{1}{2^n}\left(W_n^{\langle k,k\rangle} - W_n^{\langle k-1,k-1\rangle}\right)$$

and is fully determined by (30). The probabilities are then easily computed using any symbolic algebra package: For $n = 200$, the values found are

k	3	4	5	6	7	8	9	10	11	12
$\mathbb{P}$:	$6.54\,10^{-8}$	$7.07\,10^{-4}$	0.0339	0.1660	0.2574	0.2235	0.1459	0.0829	0.0440	0.0226

Thus, in a randomly produced sequence of length 200, there are usually runs of length 7 or more: the probability of the event turns out to be close to 80% (and there is still a probability of about 8% to have a run of length 11 or more). On the other hand most children (and adults) are usually afraid of writing down runs longer than 4 or 5 as this is felt as strongly "non-random". Hence, the statistician simply selects the slips that contain runs of length 6 or more. Et voilà! □

▷ **21.** *Coding without long runs.* Because of hysteresis in magnetic heads, certain storage devices cannot store binary sequences that have more than 4 consecutive 0's or more than 4 consecutive 1's. A coding scheme that transforms an arbitrary binary string into a string obeying this constraint will be called "acceptable".

From the GF, one finds $[z^{11}]W^{\langle 4,4\rangle}(z) = 1546 > 2^{10} = 1024$. Consequently, a code can be built that translates 10 bit blocks into acceptable 11 bit blocks, and only needs a built-in table of size 1024. Such a code has a loss factor of 10%.

Any acceptable code must use asymptotically at least $1.056n$ bits to encode strings of n bits. (Hint: let α be the root near $\frac{1}{2}$ of $1 - 2\alpha + 2\alpha^6 - \alpha^{10} = 0$, which is a pole of $W^{\langle 4,4\rangle}$. One has $\log_2(1/\alpha) = 1.05621$.) Thus, a loss of at least 5% *must* be incurred because of the coding constraint. See Ex. 10 for related coding theory arguments. This limit rate of 1.056 can be approached arbitrarily well, albeit with codes of growing complexity. ◁

EXAMPLE 5. *Patterns in a random text.* A sequence of letters that occurs in the right order, but not necessarily contiguously in a text is said to be a "hidden pattern". For instance the pattern "*combinatorics*" is to be found hidden in Shakespeare's Hamlet (Act I, Scene 1)

comb at in which our v a lian t Hamlet [...] f or fe i t [...] Whi c h he s tood ...

A census shows that there are in fact $1.63\,10^{39}$ occurrences hidden somewhere amongst the 120,057 letters that constitute the text. Is this the sign of a secret encouragement passed to us by the author of Hamlet?

Take a fixed finite alphabet $\mathcal{A}$ comprising m letters ($m = 26$ for English). Let $\mathfrak{p} = p_1 p_2 \cdots p_k$ be a word of length k. Consider the regular specification

$$\mathcal{O} = \mathfrak{S}\{\mathcal{A}\}\, p_1\, \mathfrak{S}\{\mathcal{A}\}\, p_2\, \mathfrak{S}\{\mathcal{A}\} \cdots \mathfrak{S}\{\mathcal{A}\}\, p_{k-1}\, \mathfrak{S}\{\mathcal{A}\}\, p_k\, \mathfrak{S}\{\mathcal{A}\}.$$

An element of $\mathcal{O}$ is a $(2k+1)$-tuple whose first component is an arbitrary word, whose second component is the letter p_1, and so on, with letters of the pattern and free blocks alternating . In other terms, any $\omega \in \mathcal{O}$ represents precisely one possible occurrence of the hidden pattern $\mathfrak{p}$ in a text built over the alphabet $\mathcal{A}$. The associated OGF is simply

$$O(z) = \frac{z^k}{(1 - mz)^{k+1}}.$$

The ratio between the number of occurrences and the number of words of length n then equals

$$\Omega_n = \frac{[z^n]O(z)}{m^n} = m^{-k}\binom{n}{k}, \tag{31}$$

and this quantity represents the expected number of occurrences of the hidden pattern in a random word of length n, assuming all such words to be equally likely. For the parameters corresponding to the text of Hamlet ($n = 120,057$) and the pattern "*combinatorics*" ($k = 13$), the quantity Ω_n evaluates to $6.96\,10^{38}$. The number of hidden occurrences observed is thus 23 times higher than what the uniform model predicts! However, similar methods make it possible to take into account nonuniform letter probabilities (see Chapter III): based on the frequencies of letters in the English text itself, the expected number of occurrences is found to be $1.71\,10^{39}$—this is now only within 5% of what is observed.

Thus, Shakespeare did not (probably) conceal in his text any message relative to combinatorics.

In the same vein, one can describe all the occurrences of a fixed word $\mathfrak{p} = p_1 p_2 \cdots p_k$ as a *contiguous* block (a "factor") in texts:

$$\widehat{\mathcal{O}} = \mathfrak{S}\{\mathcal{A}\}\,(p_1 p_2 \cdots p_k)\,\mathfrak{S}\{\mathcal{A}\},$$

so that the OGF is

$$\widehat{O}(z) = \frac{z^k}{(1-mz)^2}.$$

Consequently, the expected number of such contiguous occurrences satisfies

$$\widehat{\Omega}_n = m^{-k}(n-k+1) \sim \frac{n}{m^k}. \tag{32}$$

□

For patterns, the estimation of the mean in (31) and (32) can be easily obtained by direct probabilistic reasoning. The example is only meant to demonstrate a symbolic approach to pattern statistics that proves extremely versatile: it can accommodate various notions of patterns (e.g., we may impose maximal spacings between letters) and provide valuable informations on probability distributions as well; see [**50**]. Such methods are of interest in the statistical analysis of texts and in assessing the significance of patterns detected in molecular biology; see [**149**, Ch. 12] for an introduction. From the combinatorial standpoint, these examples illustrate the counting of structures that are richer than words (namely, pattern occurrences) by means of regular specifications.

▷ **22.** *Patterns with gaps.* If less than d symbols of the text must separate the letters of the pattern in order to form a valid occurrence, then the OGF of occurrences is

$$z^k \frac{(1-m^d z^d)^{k-1}}{(1-mz)^{k+1}}.$$

See [**50**] for variations of this theme. ◁

I. 4.2. Finite automata. Let again a finite alphabet $\mathcal{A}$ be fixed. We first define a simple device that is able to "process" words over the alphabet and has wide descriptive power as regards structural properties of words.

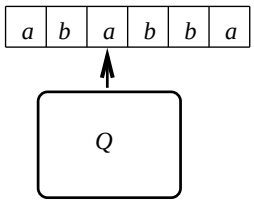
DEFINITION I.8. *A* finite automaton *is a directed multigraph whose edges are labelled by letters of the alphabet. It is customary to call the vertices by the name of* states *and denote by Q the set of states. An initial state $q_0 \in Q$ and a set of final states $Q_f \subseteq Q$ are also designated. A word $w = w_1 \dots w_n$ is* accepted *by the automaton if there exists a path in the multigraph connecting the initial state q_0 to one of the final states of Q_f and whose sequence of edge labels is precisely $w_1, \dots, w_n$.*

An automaton is said to be deterministic *if for each pair (q, α) with $q \in Q$ and $\alpha \in A$ there exists at most one edge (one also says a transition) starting from q that is labelled by the letter α. A language is said to be A–*regular *(automaton regular) if it coincides with the set of words accepted by a deterministic finite automaton.*

The following equivalence theorem is briefly discussed in the Appendix (see APPENDIX: *Regular languages*, p. 171):

> THEOREM (Kleene–Rabin–Scott). *For a language, the following four conditions are equivalent: (i) to be S-regular (i.e., representable by a regular specification); (ii) to be A-regular (i.e., recognizable by a deterministic finite automaton); (iii) to be the set of words accepted by a nondeterministic finite automaton; (iv) to be described by a standard regular* expression.

In the case of a deterministic automaton, it is easy to determine whether a word w is accepted: it suffices to start from the initial state q_0, scan the letters of the word from left to right, and follow at each stage the only transition permitted; the word is accepted if the state reached in this way after scanning the last letter of w is a final state. A deterministic automaton is thus a simple processing device that has a finite instruction set governing its evolution when characters are read. Here is a rendering:



As an illustration, consider the class $\mathcal{L}$ of all words w that contain the pattern abb as a factor (the letters of the pattern should appear contiguously). Such words are recognized by a finite automaton with 4 states, q_0, q_1, q_2, q_3. The construction is classical: state q_j is interpreted as meaning "*the first j characters of the pattern have just been scanned*", and the corresponding automaton appears in Figure I. 4.2. The initial state is q_0, and there is a unique final state q_3.

We next examine the way generating functions can be obtained from a deterministic automaton. The process was first discovered in the late 1950's by Chomsky and Schützenberger [**26**]. It proves convenient at this stage to introduce Iverson's bracket notation: for a predicate P, the variable $[\![P]\!]$ has value 1 if P is true and 0 otherwise.

PROPOSITION I.3. *Let G be a deterministic finite automaton with state set $Q = \{q_0, \ldots, q_s\}$, initial state q_0, and set of final states $\overline{Q} = \{q_{i_1}, \ldots, q_{i_f}\}$. The generating function of the language $\mathcal{L}$ of all words accepted by the automaton is a rational function that is determined under matrix form as*

$$L(z) = \mathrm{u}(I - zT)^{-1}\mathrm{v}.$$

There the transition matrix T is defined by

$$T_{i,j} = \text{card}\,\{\alpha \in \mathcal{A} \text{ such that an edge } (q_i, q_j) \text{ is labelled by } \alpha\};$$

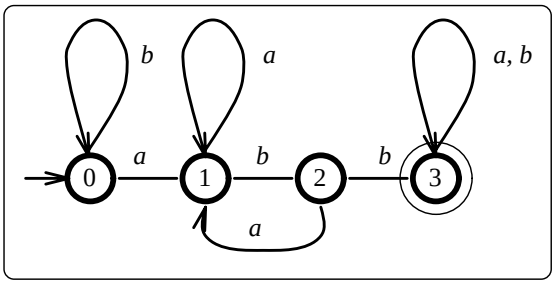


FIGURE 8. Words that contain the pattern abb are recognized by a 4–state automaton with initial state q_0 and final state q_3.

the line vector u *is the vector* $(1,0,0,\dots,0)$ *and the column vector* $\mathrm{v} = (v_0,\dots,v_s)^t$ *is such that* $v_j = [\![q_j \in \overline{Q}]\!]$.

In particular, by Cramer's rule, the OGF of a regular language is the quotient of two sparse determinants whose structure directly reflects the automaton transitions.

PROOF. For $j \in \{0,\dots,s\}$, introduce the class (language) $\mathcal{L}_j$ of all words w such that the automaton, when started in state q_j, terminates in one of the final states after having read w. The following relation holds for any j:

$$\mathcal{L}_j \cong \Delta_j + \left(\sum_{\alpha\in\mathcal{A}} \{\alpha\}\mathcal{L}_{(q_j\circ\alpha)} \right); \tag{33}$$

there Δ_j is the class $\{\epsilon\}$ formed of the word of length 0 if q_j is final and the empty set ($\emptyset$) otherwise; the notation $(q_j \circ \alpha)$ designates the state reached in one step from state q_j upon reading letter α. The justification is simple: a language $\mathcal{L}_j$ contains the word of length 0 only if the corresponding state q_j is final; a word of length ≥ 1 that is accepted starting from state q_j has a first letter α followed by a word that must lead to an accepting state when starting from state $q_j \circ \alpha$.

The translation of (33) is then immediate:

$$L_j(z) = [\![q_j \in \overline{Q}]\!] + z \sum_{\alpha\in\mathcal{A}} L_{(q_j\circ\alpha)}(z). \tag{34}$$

The collection of all the equations as j varies forms a linear system: with $\mathrm{L}(z)$ the column vector $(L_0(z),\dots,L_s(z))$, one has

$$\mathrm{L}(z) = \mathrm{v} + zT\,\mathrm{L}(z),$$

where v and T are as described in the statement. The result follows by matrix inversion upon observing that $L(z) \equiv L_0(z)$. □

For instance, consider the automaton recognizing the pattern abb as given in Figure 8. The languages $\mathcal{L}_j$ (where L_j is the set of accepted words when starting fom state q_j) are connected by the system of equations

$$\begin{array}{lcllll} \mathcal{L}_0 &=& a\mathcal{L}_1 &+ b\mathcal{L}_0 & & \\ \mathcal{L}_1 &=& a\mathcal{L}_1 &+ b\mathcal{L}_2 & & \\ \mathcal{L}_2 &=& a\mathcal{L}_1 &+ b\mathcal{L}_3 & & \\ \mathcal{L}_3 &=& a\mathcal{L}_3 &+ b\mathcal{L}_3 &+ \epsilon, & \end{array}$$

which directly reflects the graph structure of the automaton. This gives rise to a set of equations for the associated OGFs

$$\begin{array}{lcllll} L_0 &=& zL_1 &+ zL_0 & & \\ L_1 &=& zL_1 &+ zL_2 & & \\ L_2 &=& zL_1 &+ zL_3 & & \\ L_3 &=& zL_3 &+ zL_3 &+ 1. & \end{array}$$

Solving the system, we find the OGF of all words containing the pattern abb: it is $L_0(z)$ since the initial state of the automaton is q_0, and

$$L_0(z) = \frac{z^3}{(1-z)(1-2z)(1-z-z^2)}. \tag{35}$$

The partial fraction decomposition

$$L_0(z) = \frac{1}{1-2z} - \frac{2+z}{1-z-z^2} + \frac{1}{1-z},$$

then yields

$$L_{0,n} = 2^n - \mathrm{F}_{n+3} + 1,$$

with F_n a Fibonacci number. In particular the number of words of length n that do *not* contain abb is $\mathrm{F}_{n+3} - 1$, a quantity that grows at an exponential rate of φ^n, with $\varphi = (1+\sqrt{5})/2$ the golden ratio. Thus, all but an exponentially vanishing proportion of the strings of length n contain the given pattern abb, a fact that was otherwise to be expected on probabilistic grounds. (For instance, from the previous subsection, a random word contains a large number, about $\sim n/8$, of occurrences of the pattern abb.)

This example is simple enough that one can also come up with an equivalent regular expression describing $\mathcal{L}_0$: an accepting path in the automaton of Figure 8 loops around state 0 with a sequence of b, then reads an a, loops around state 1 with a sequence of a's and moves to state 2 upon reading a b; then there should be letters making the automaton passs through states 1-2-1-2-$\cdots$-1-2 and finally a b followed by an arbitrary sequence of a's and b's at state 3. This corresponds to the specification

$$\mathcal{L}_0 = \mathfrak{S}\{b\}\, a\mathfrak{S}\{a\}b\, \mathfrak{S}\{a\mathfrak{S}\{a\}b\}\, b\mathfrak{S}\{a+b\},$$

which gives back a form equivalent to (35), namely,

$$L_0(z) = \frac{z^3}{(1-z)^2(1-\frac{z^2}{1-z})(1-2z)}.$$

The general construction that reduces systematically finite automata to regular specifications is due to the logician Kleene and is discussed in APPENDIX: *Regular languages*, p. 171.

EXAMPLE 6. *Words containing or excluding a pattern.* Fix an arbitrary pattern $\mathfrak{p} = p_1 p_2 \cdots p_k$ and let $\mathcal{L}$ be the language of words containing *at least* one occurrence of $\mathfrak{p}$ as a contiguous block. The construction given for the particular pattern $\mathfrak{p} = abb$ generalizes in an easy manner: there exists a deterministic finite automaton with $k+1$ states that recognizes $\mathcal{L}$, the states corresponding to the prefixes of the pattern $\mathfrak{p}$. Thus, the OGF $L(z)$ is *a priori* a rational function of degree at most $k+1$. (The corresponding automaton is in fact known as a Knuth–Morris–Pratt automaton [**88**].)

The automaton construction provides the OGF $L(z)$ in determinantal form but the relation between this rational form and the structure of the pattern is not transparent. An explicit construction due to Guibas and Odlyzko [**74**] nicely circumvents this problem; it is based on an "equational" specification that yields an alternative linear system. The fundamental notion is that of an *autocorrelation vector*. For a given $\mathfrak{p}$, this vector of bits $c = (c_0, \ldots, c_{k-1})$ is most conveniently defined in terms of Iverson's bracket as

$$c_i = [\![p_1 p_2 \cdots p_{k-i} = p_{i+1} p_{i+2} \cdots p_k]\!].$$

In other words, the bit c_i is determined by shifting $\mathfrak{p}$ right by i positions and putting a 1 if the remaining letters match the original. For instance, with $\mathfrak{p} = aabbaa$, one has

```
a a b b a a
___________
a a b b a a          1
  a a b b a a        0
    a a b b a a      0
      a a b b a a    0
        a a b b a a  1
          a a b b a a 1
```

The autocorrelation is then $c = (1, 0, 0, 0, 1, 1)$. The *autocorrelation polynomial* is defined as

$$c(z) := \sum_{j=0}^{k-1} c_j z^n.$$

For the example pattern, this gives $c(z) = 1 + z^4 + z^5$.

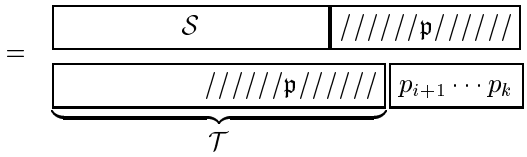
Let $\mathcal{S}$ be the language of words with *no* occurrence of $\mathfrak{p}$ and $\mathcal{T}$ the language of words that end with $\mathfrak{p}$ but have no other occurrence of $\mathfrak{p}$. First, by appending a letter to a word of $\mathcal{S}$, one finds a nonempty word either in $\mathcal{S}$ or $\mathcal{T}$, so that

$$\mathcal{S} + \mathcal{T} = \{\epsilon\} + \mathcal{S} \times \mathcal{A}. \tag{36}$$

Next, appending a copy of the word $\mathfrak{p}$ to a word in $\mathcal{S}$ may only give words that contain $\mathfrak{p}$ at or "near" the end. Precisely, the decomposition based on the leftmost occurrence of $\mathfrak{p}$ in $\mathcal{S}\mathfrak{p}$ is

$$\mathcal{S} \times \{\mathfrak{p}\} = \mathcal{T} \times \sum_{c_i \neq 0} \{p_{i+1} p_{i+2} \cdots p_k\}, \tag{37}$$

corresponding to the configurations



The translation of the system (36), (37) into OGF's then gives:

The OGF of words not *containing the pattern* $\mathfrak{p}$ *is*

$$S(z) = \frac{c(z)}{z^k + (1 - mz)c(z)}, \tag{38}$$

where m *is the alphabet cardinality,* $k = |\mathfrak{p}|$ *the pattern length, and* $c(z)$ *the autocorrelation polynomial,* $c(z) = \sum_i c_i z^i$.

Similarly, the GF's of words containing at least once the pattern (anywhere) and containing it only once at the end are

$$L(z) = \frac{z^k}{(1 - mz)(z^k + (1 - mz)c(z))}, \quad T(z) = \frac{z^k}{z^k + (1 - mz)c(z)},$$

respectively. □

▷ **23.** *Waiting times in strings.* Let $\mathcal{L} \subset \mathfrak{S}\{a, b\}$ be a language and $S = \{a, b\}^\infty$ be the set of infinite strings with the product probability induced by $\Pr(a) = \mathbb{P}(b) = \frac{1}{2}$. The probability that a random string $\omega \in S$ starts with a word of L is $\widehat{L}(1/2)$, where $\widehat{L}(z)$ is the OGF of the "prefix language" of L, that is, the set of words $w \in L$ that have no strict prefix belonging to $\mathcal{L}$. The GF $\widehat{L}(z)$ serves to express the expected time at which a word in $\mathcal{L}$ is first encountered: this is $1/2\widehat{L}'(1/2)$. For a regular language, this quantity must be a rational number. ◁

▷ **24.** *A probabilistic paradox on strings.* In a random infinite sequence, a pattern $\mathfrak{p}$ of length k first occurs on average at time $2^k c(1/2)$, where $c(z)$ is the correlation polynomial. For instance, the pattern $\mathfrak{p} = abb$ tends to occur "sooner" (at average position 8) than $\mathfrak{p}' = aaa$ (at average position 14). See [**74**] for a thorough discussion. Here are for instance the epochs at which $\mathfrak{p}$ and $\mathfrak{p}'$ are first found in a sample of 20 runs

$\mathfrak{p}$: 3, 4, 5, 5, 6, 6, 7, 8, 8, 8, 8, 9, 9, 10, 11, 14, 15, 15, 16, 21

$\mathfrak{p}'$: 3, 4, 8, 8, 9, 10, 11, 11, 11, 12, 17, 22, 23, 27, 27, 27, 44, 47, 52, 52.

On the other hand, patterns of the same length have the same expected number of occurrences, which is puzzling. (The catch is that, due to overlaps of $\mathfrak{p}'$ with itself, occurrences of $\mathfrak{p}'$ tend to occur in clusters, but, then, clusters tend to be separated by wider gaps than for $\mathfrak{p}$; eventually, no contradiction occurs.) $\triangleleft$

$\triangleright$ **25.** *Borges's Theorem.* Take any fixed set Π of finite patterns. A random text of length n contains all the patterns of the set Π (as contiguous blocks) with probability tending to 1 exponentially fast as $n \to \infty$. (Reason: the rational functions $S(z/2)$ with $S(z)$ as in (38) have no pole in $|z| \leq 1$; see also Chapter 4.)

Note: similar properties hold for many random combinatorial structures They are sometimes called "Borges's Theorem" as a tribute to the famous Argentinian writer Jorge Luis Borges (1899–1986) who, in his essay *"The Library of Babel"*, describes a library so huge as to contain: "Everything: the minutely detailed history of the future, the archangels' autobiographies, the faithful catalogues of the Library, thousands and thousands of false catalogues, the demonstration of the fallacy of those catalogues, the demonstration of the fallacy of the true catalogue, the Gnostic gospel of Basilides, the commentary on that gospel, the commentary on the commentary on that gospel, the true story of your death, the translation of every book in all languages, the interpolations of every book in all books." $\triangleleft$

In general, automata are useful in establishing *a priori* the rational character of generating functions. They are also surrounded by interesting analytic properties (e.g., Perron-Frobenius theory that characterizes the dominant poles) and by asymptotic probability distributions of associated parameters that are normally Gaussian. They are most conveniently used for proving existence theorems, then supplemented when possible by regular specifications that may lead to more explicit expressions.

$\triangleright$ **26.** *Variable length codes.* A finite set $\mathcal{F} \subset \mathcal{W}$, where $\mathcal{W} = \mathfrak{S}\{\mathcal{A}\}$ is called a *code* if any word of $\mathcal{W}$ decomposes in at most one manner into factors that belong to $\mathcal{F}$ (with repetitions allowed). For instance $\mathcal{F} = \{a, ab, bb\}$ is a code and $aaabbb = a|a|ab|bb$ has a unique decomposition; $\mathcal{F}' = \{a, aa, b\}$ is not a code since $aaa = a|aa = aa|a = a|a|a$. The OGF of the set $\mathcal{S}_{\mathcal{F}}$ of all words that admit a decomposition into factors all in $\mathcal{F}$ is a computable rational function, irrespective of whether $\mathcal{F}$ is a code. (Hint: use a construction by automaton.) A finite set $\mathcal{F}$ is a code iff $S_{\mathcal{F}}(z) = (1 - F(z))^{-1}$. Consequently, the property of being a code can be decided in polynomial time using linear algebra. The book of Berstel and Perrin [**16**] develops systematically the theory of such "variable-length" codes; see also the construction of the "Aho–Corasick" automaton in [**1**]. $\triangleleft$

$\triangleright$ **27.** *Knight's tours.* For the number of knight's tours on an $n \times w$ chessboard (with fixed w and varying n), the OGF is a rational function. In statistical physics, such automata related methods are commonly used and known as *transfer matrix methods*. $\triangleleft$

I. 4.3. Word related constructions. Words can encode any combinatorial structure. We detail here one example that demonstrates the usefulness of such encodings: it is relative to set partitions and Stirling numbers. The point to be made is that some amount of "combinatorial preprocessing" is sometimes necessary in order to bring combinatorial structures into the framework of symbolic methods.

EXAMPLE 7. *Set partitions and Stirling partition numbers.* A *set partition* is a partition of a finite domain into a certain number of nonempty sets, also called blocks. For instance, if the domain is $\mathcal{D} = \{\alpha, \beta, \gamma, \delta\}$, there are 15 ways to partition it (Figure 9). Let $\mathcal{S}_n^{(k)}$ denote the collection of all partitions of the set $[1 \,..\, n]$ into k non–empty blocks and $S_n^{(k)} = \text{card}(\mathcal{S}_n^{(k)})$ the corresponding cardinality. The basic object under consideration here is a *set partition* (not to be confused with integer partitions considered earlier).

It is possible to find an encoding of partitions in $\mathcal{S}_n^{(k)}$ of an n–set into k blocks by words over a k letter alphabet, $\mathcal{B} = \{b_1, b_2, \ldots, b_k\}$ as follows:

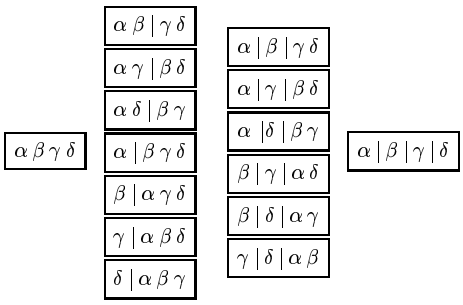


FIGURE 9. The 15 ways to partition a four-element domain into blocks correspond to $S_4^{(1)} = 1, \quad S_4^{(2)} = 7, \quad S_4^{(3)} = 6, \quad S_4^{(4)} = 1$.

Consider a set partition ϖ that is formed of k blocks. Identify each block by its smallest element called the block *leader*; then sort the block leaders into increasing order. Define the index of a block as the rank of its leader amongst all the k leaders, with ranks conventionally starting at 1.
Scan the elements 1 to n in order and produce sequentially n letters from the alphabet $\mathcal{B}$: for an element belonging to the block of index r, produce the letter b_r.

For instance to $n = 6$, $k = 3$, the set partition $\varpi = \{\{6,4\},\{5,1,2\},\{3,7,8\}\}$, is reorganized by putting leaders in first position of the blocks and sorting them,

$$\varpi = \{\overbrace{\{\underline{1},2,5\}}^{b_1},\overbrace{\{\underline{3},7,8\}}^{b_2},\overbrace{\{\underline{4},6\}}^{b_3}\},$$

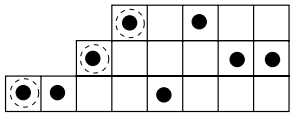
so that the encoding is

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ b_1 & b_1 & b_2 & b_3 & b_1 & b_3 & b_2 & b_2 \end{pmatrix}.$$

In this way, a partition is encoded as a word of length n over $\mathcal{B}$ with the additional properties that: (i) all k letters occur; (ii) the first occurrence of b_1 precedes the first occurrence of b_2 which itself precedes the first occurrence of b_3, etc. Thus $\mathcal{S}_n^{(k)}$ is mapped $\mathcal{S}_n^{(k)}$ into words of length n in the language

$$b_1\,\mathfrak{S}\{b_1\}\cdot b_2\,\mathfrak{S}\{b_1+b_2\}\cdot b_3\,\mathfrak{S}\{b_1+b_2+b_3\}\;\cdots\;b_k\,\mathfrak{S}\{b_1+b_2+\cdots+b_k\}. \tag{39}$$

(The encoding is clearly revertible.) Graphically, this can be rendered by an "irregular staircase" representation, like



where the staircase has length n and height k, each column contains exactly one element, and the columns exposed North-West are systematically filled.

The language specification immediately gives the OGF

$$S^{(k)}(z) = \frac{z^k}{(1-z)(1-2z)(1-3z)\cdots(1-kz)}.$$

The partial fraction expansion of $S^{(k)}(z)$ is readily computed,

$$S^{(k)}(z) = \frac{1}{k!}\sum_{j=0}^{k}\binom{k}{j}\frac{(-1)^{k-j}}{1-jz}, \quad \text{so that} \quad S_n^{(k)} = \frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}\binom{k}{j}j^n.$$

In particular, one has

$$S_n^{(1)} = 1;\ S_n^{(2)} = \frac{1}{2!}(2^n - 2);\ S_n^{(3)} = \frac{1}{3!}(3^n - 3\cdot 2^n + 3).$$

These numbers are known as the Stirling numbers of the second kind, or better, as the Stirling partition numbers, and the $S_n^{(k)}$ are nowadays usually denoted by $\left\{ {n \atop k} \right\}$; see APPENDIX: *Stirling numbers*, p. 173. □

The counting of set partitions could eventually be done successfully thanks an encoding into words, and the corresponding language forms a constructible class of combinatorial structures (actually a regular language). In the next chapter, we shall examine another approach to the counting of set partitions that is based on labelled structures and exponential generating functions.

We conclude this section with a brief mention of "circular words". Let $\mathcal{A}$ be a binary alphabet, viewed as comprised of beads of two distinct colours. The class $\mathcal{N} = \mathfrak{C}\{\mathcal{A}\}$ represents the set of words to taken up to circular shift of their letters. Equivalently, with $\mathcal{A} = \{\bullet, \circ\}$, the class $\mathcal{N}$ describes "necklaces" (p. 3). The OGF of necklaces is given the cyle construction operator:

$$\begin{aligned} N(z) &= \sum_{k=1}^{\infty}\frac{\varphi(k)}{k}\log\frac{1}{1-2z^k} \\ &= 2\,z + 3\,z^2 + 4\,z^3 + 6\,z^4 + 8\,z^5 + 14\,z^6 + 20\,z^7 + 36\,z^8 + 60\,z^9 + \cdots . \end{aligned}$$

Consequently, one has

$$N_n = \frac{1}{n}\sum_{k\ |\ n}\varphi(k)2^{n/k}. \tag{40}$$

This is sequence *EIS* **A000031** and one has $N_n = D_n + 1$ where D_n is the wheel count, p. 27. [The connection is easily explained combinatorially: start from a wheel and repaint in white all the nodes that are not on the basic circle; then fold them onto the circle.] The same argument proves that the number of necklaces over an m-ary alphabet is obtained by replacing 2 by m in (40).

I. 5. Trees and tree-like structures

This section is concerned with basic tree enumerations. Trees are, as we saw, the prototypical recursive structure. There, recursive specifications normally lead to nonlinear equations (and systems of such equations) over generating functions. The Lagrange inversion theorem is useful in solving the simplest category of problems. The functional equations furnished by the symbolic method are then conveniently exploited by the asymptotic theory of Chapter 5. a certain type of analytic behaviour appears to be universal in trees, namely a $\sqrt{\ }$–singularity; as a consequence, most trees families occurring in the combinatorial world have counting sequences obeying the asymptotic form $C\,A^{\,n}n^{-3/2}$.

I. 5.1. Plane trees. Plane trees are also sometimes called ordered trees. There, the subtrees dangling from a node are ordered between themselves. Alternatively, these trees may be viewed as abstract graph structures accompanied by an embedding into the plane; see APPENDIX: *Tree concepts*, p. 174 for key concepts associated with trees. They are precisely described in terms of unions, cartesian products, and sequence constructions. Here, we restrict attention to rooted trees.

First, consider the class $\mathcal{G}$ of "general" plane trees where all node degrees are allowed; it satisfies the recursive specification (already discussed on p. 17,

$$\mathcal{G} = \mathcal{Z} \times \mathfrak{S}\{\mathcal{G}\}, \tag{41}$$

and, accordingly, $G(z)$ is determined by

$$G(z) = \frac{z}{1 - G(z)}, \qquad \text{hence} \qquad G(z) = \frac{1 - \sqrt{1-4z}}{2}.$$

As a result, the number of general trees of size n is the Catalan number C_{n-1}:

$$G_n = \frac{1}{n}\binom{2n-2}{n-1} = \frac{1}{2n-1}\binom{2n-1}{n} = \frac{(2n-2)!}{n!\,(n-1)!}.$$

Many classes of trees defined by all sorts of constraints on properties of nodes appear to be of interest in combinatorics and in related areas like logic and computer science. Let Ω be a subset of the integers that contains 0. Define the class $\mathcal{T}^{\Omega}$ of Ω-restricted trees as formed of trees such that the outdegrees of nodes are constrained to lie in Ω. Thus, for instance $\Omega = \{0, 2\}$ determines binary trees, where each node has either 0 or 2 descendants; $\Omega = \{0, 1, 2\}$ and $\Omega = \{0, 3\}$ determine respectively unary-binary trees and ternary trees; the case of general trees corresponds to $\Omega = \mathbb{Z}_{\geq 0}$. In what follows, an essential rôle is played by the (ordinary) characteristic function of Ω, namely

$$\phi(u) := \sum_{\omega \in \Omega} u^{\omega}.$$

It is in terms of this characteristic function that Ω-restricted trees can be enumerated as shown by the following statement:

PROPOSITION I.4. *The ordinary generating function $T^{\Omega}(z)$ of the class $\mathcal{T}^{\Omega}$ of Ω-restricted trees is determined implicitly by the equation*

$$T(z) = z\,\phi(T(z)),$$

where ϕ is the ordinary characteristic of Ω, namely $\phi(u) := \sum_{\omega \in \Omega} u^{\omega}$. The tree counts are given by

$$T_n^{\Omega} \equiv [z^n]T^{\Omega}(n) = \frac{1}{n}\,[u^{n-1}]\phi(u)^n. \tag{42}$$

PROOF. The GF equation is a direct consequence of the specification $\mathcal{T}^{\Omega} = \mathcal{Z}\mathfrak{S}_{\Omega}\{\mathcal{T}\}$ and of the obvious translation of Ω-restricted sequences:

$$\mathcal{A} = \mathfrak{S}_{\Omega}\{B\} \quad \Longrightarrow \quad A(z) = \phi(B(z)).$$

This shows that $T = T^{\Omega}$ is related to z by functional inversion:

$$z = \frac{T}{\phi(T)}.$$

The Lagrange Inversion Theorem precisely states that the expansion of an inverse function (here T) are determined simply by coefficients of powers of the "direct" function (that involves ϕ): see APPENDIX: *Lagrange Inversion*, p. 170. This is precisely what is expressed by (42. □

The statement extends trivially to the case where Ω is a multiset of integers, that is, a set of integers with repetitions allowed. For instance, $\Omega = \{0, 1, 1, 3\}$ corresponds to unary-ternary trees with two types of unary nodes, say, having one of two colours; in this case, the characteristic is $\phi(u) = u^0 + 2u^1 + u^3$. The theorem gives back the enumeration of general trees, where $\phi(u) = (1-u)^{-1}$, by way of the binomial theorem applied to $(1-u)^{-n}$. In general, it implies that, whenever Ω comprises r elements, $\Omega = \{\omega_1, \ldots, \omega_r\}$, the tree counts are expressed as an $(r-1)$-fold summation of binomial coefficients (use the multinomial expansion). An important special case detailed below is when Ω has only two elements.

▷ **28.** *Forests.* Consider ordered k-forests of trees defined by $\mathcal{F} = \mathfrak{S}_k\{\mathcal{T}\}$. The Bürmann form of Lagrange inversion implies

$$[z^n]F(z) \equiv [z^n]T(z)^k = \frac{k}{n}[u^{n-k}]\ \phi(u)^n.$$

In particular, one has for forests of general trees ($\phi(u) = (1-u)^{-1}$):

$$[z^n]\left(\frac{1-\sqrt{1-4z}}{2}\right)^k = \frac{k}{n}\binom{2n-k-1}{n-1};$$

the coefficients are also known as "ballot numbers". ◁

EXAMPLE 8. *"Regular" (t-ary) trees.* A tree is said to be t-regular or t-ary if Ω consists only of the elements $\{0, t\}$. In other words, all internal nodes have degree t exactly, hence the name. Let $\mathcal{A} := \mathcal{T}^{\{0,t\}}$. In an element of $\mathcal{A}$, a node is either terminal or it has exactly t children. In this case, the characteristic is $\phi(u) = 1 + u^t$ and the binomial theorem combined with the Lagrange inversion formula gives

$$\begin{aligned} A_n &= \frac{1}{n}\,[u^{n-1}]\,(1+u^t)^n \\ &= \frac{1}{n}\binom{n}{\frac{n-1}{t}} \quad \text{provided } n \equiv 1 \bmod t. \end{aligned}$$

As the formula shows, only trees of total size of the form $n = t\nu + 1$ exist (a well-known fact otherwise easily checked by induction), and

$$A_{t\nu+1} = \frac{1}{t\nu+1}\binom{t\nu+1}{\nu} = \frac{1}{(t-1)\nu+1}\binom{t\nu}{\nu}. \tag{43}$$

A particular rôle is played by binary trees. Then a form equivalent to (43) reads:

> *The number of plane binary trees having a total of $2\nu+1$ nodes (i.e., ν binary nodes and $\nu+1$ external nodes) is the Catalan number* $\mathrm{C}_\nu = \frac{1}{\nu+1}\binom{2\nu}{\nu}$.

In this book, we shall use $\mathcal{B}$ to denote the class of binary trees. Size will be freely measured, depending on context and convenience, by recording internal, external, or all nodes.

There is a variant of the determination of (43) that avoids congruence restrictions. Let $\mathcal{A}$ be the class of t-ary trees and define the class $\widehat{\mathcal{A}}$ of "pruned" trees as trees of $\mathcal{A}$ deprived of all their external nodes. The trees in $\widehat{\mathcal{A}}$ now have nodes that are of degree at most t. In order to make $\widehat{\mathcal{A}}$ bijectively equivalent to $\mathcal{A}$, it suffices to regard trees of $\widehat{\mathcal{A}}$ as having $\binom{t}{j}$ possible types of nodes of degree j for any $j \in [0, t]$: each node type in $\widehat{\mathcal{A}}$

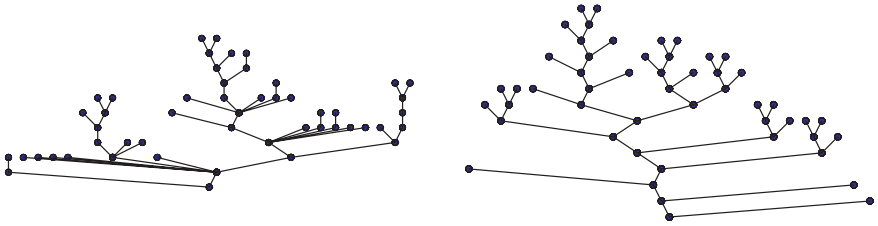


FIGURE 10. A general tree of $\mathcal{G}_{51}$ (left) and a binary tree of $\mathcal{T}_{51}^{\{0,2\}}$ (right) drawn uniformly at random amongst the C_{50} and C_{25} possible trees respectively, with $\mathrm{C}_n = \frac{1}{n+1}\binom{2n}{n}$ the nth Catalan number.

plainly encodes which of the original $t-j$ subtrees have been pruned. The equations above immediately generalize to the case of an Ω with multiplicities. One finds $\widehat{\phi}(u) = (1+u)^t$ and $\widehat{A}(z) = z\widehat{\phi}(\widehat{A}(z))$, so that, by Lagrange inversion,

$$\widehat{A}_\nu = \frac{1}{\nu}\binom{t\nu}{\nu-1},$$

yet another equivalent form of (43), since, by basic combinatorics, $\widehat{A}_\nu = A_{t\nu+1}$. □

▷ **29.** *Motzkin numbers.* Let $M(z)$ be the generating function for unary-binary trees ($\Omega = \{0,1,2\}$):

$$M(z) = z(1 + M(z) + M(z)^2) \qquad \Longrightarrow \qquad M(z) = \frac{1 - z - \sqrt{1-2z-3z^2}}{2z}.$$

One has $M(z) = z + z^2 + 2\,z^3 + 4\,z^4 + 9\,z^5 + 21\,z^6 + 51\,z^7 + \cdots$. The coefficients $M_n = [z^n]M(z)$ are given in Lagrange form as

$$M_n = \frac{1}{n}\sum_k \binom{n}{k}\binom{n-k}{k-1},$$

and called Motzkin numbers (*EIS* **A001006**). ◁

▷ **30.** *Yet another variant of t-ary trees.* Let $\widetilde{\mathcal{A}}$ be the class of t-ary trees, but with size now defined as the number of external nodes (leaves). Then, one has

$$\widetilde{\mathcal{A}} = \mathcal{Z} + \mathfrak{S}_k\{\widetilde{\mathcal{A}}\}.$$

The binomial formula for $\widetilde{A}_n$ follows from Lagrange inversion applied to $\widetilde{A} = z/(1-\widetilde{A}^{t-1})$. ◁

EXAMPLE 9. *Hipparchus of Rhodes and Schröder.* In 1870, the German mathematician Ernst Schröder (1841–1902) published a paper entitled *Vier combinatorische Probleme*. The paper had to do with the number of terms that can be built out of n variables using nonassociative operations. In particular, the second of his four problems asks for the number of ways a string of n identical letters, say x, can be "bracketted". The rule is best stated recursively: x itself is a bracketting and if $\sigma_1, \sigma_2, \ldots, \sigma_k$ with $k \geq 2$ are bracketted expressions, then the k-ary product $(\sigma_1)(\sigma_2)\cdots(\sigma_k)$ is a bracketting.

Let $\mathcal{S}$ denote the class of all brackettings, where size is the number of variables. Then, the recursive definition is readily translated into the formal specification

$$\mathcal{S} = \mathcal{Z} + \mathfrak{S}_{\geq 2}\{\mathcal{S}\}, \quad \mathcal{Z} = \{x\}. \tag{44}$$

To each bracketting of size n is associated a tree whose external nodes contain the variable x (and determine size), with internal nodes corresponding to brackettings and having

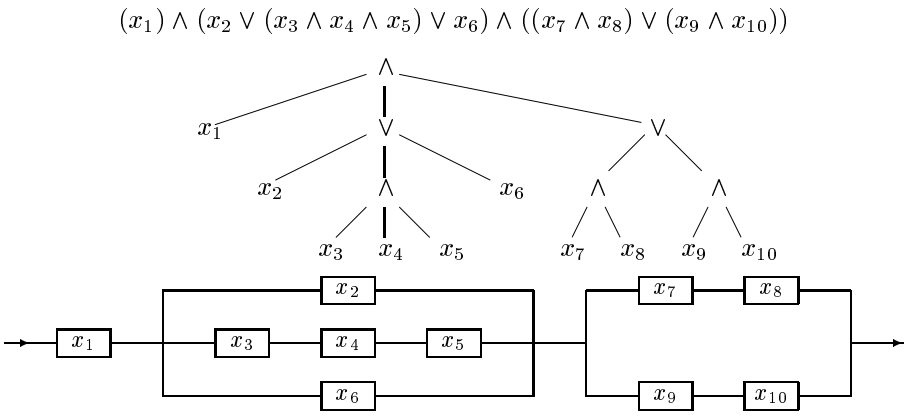


FIGURE 11. An and–or positive proposition of the conjunctive type (top), its associated tree (middle), and an equivalent planar series-paralllel network of the serial type (bottom).

degree at least 2 (while not contributing to size). The functional equation satisfied by the OGF is then

$$S(z) = z + \frac{S(z)^2}{1 - S(z)}. \tag{45}$$

This is not *a priori* of the type corresponding to Proposition I.4 because *not* all nodes contribute to size in this particular application. However, the quadratic equation induced by (45) can be solved, giving

$$\begin{aligned} S(z) &= \frac{1}{4}\left(1 + z - \sqrt{1 - 6z + z^2}\right) \\ &= z + z^2 + 3z^3 + 11z^4 + 45z^5 + 197z^6 + 903z^7 + 4279z^8 + 20793z^9 \\ &\qquad + 103049z^{10} + 518859z^{11} + \cdots, \end{aligned}$$

where the coefficients are *EIS* **A001003**. (These numbers also count series-parallel networks of a specified type (e.g., serial in Figure 11, bottom), where placement in the plane matters.)

In an instructive paper, Stanley [**136**] discusses a page of Plutarch's *Moralia* where there appears the following statement:

> "Chrysippus says that the number of compound propositions that can be made from only ten simple propositions exceeds a million. (Hipparchus, to be sure, refuted this by showing that on the affirmative side there are 103,049 compound statements, and on the negative side 310,952.)"

It is notable that the tenth number of Hipparchus of Rhodes[5] (*c.* 190–120B.C.) is precisely $S_{10} = 103,049$. This is, for instance, the number of logical formulæ that can be formed from ten boolean variables $x_1, \ldots, x_{10}$ (used once each and in this order) using

[5]This was first observed by David Hough in 1994; see [**136**]. In [**75**], Habsieger *et al.* further note that $\frac{1}{2}(S_{10} + S_{11}) = 310,954$, and suggest a related interpretation (based on negated variables) for the other count given by Hipparchus.

Tree variety	1	2	3	4	5	6	7	8	n	$+\infty$
Plane gen. $\mathcal{G} = \mathcal{Z} \times \mathfrak{S}\{\mathcal{G}\}$	1	1	2	5	14	42	132	429	$\frac{1}{n}\binom{2n-2}{n-1}$	$\sim 4^{n-1}/\sqrt{\pi n^3}$
Plane bin. $\mathcal{T} = \mathcal{Z} + \mathfrak{S}_2\{\mathcal{T}\}$	1	1	2	5	14	42	132	429	$\frac{1}{n}\binom{2n-2}{n-1}$	$\sim 4^{n-1}/\sqrt{\pi n^3}$
Unord. gen. $\mathcal{H} = \mathcal{Z} \times \mathfrak{M}\{\mathcal{H}\}$	1	1	2	4	9	20	48	115	–	$\sim \lambda \cdot \beta^n / n^{3/2}$
Unord. bin. $\mathcal{U} = \mathcal{Z} + \mathfrak{M}_2\{\mathcal{U}\}$	1	1	1	2	3	6	11	23	–	$\lambda_2 \cdot \beta_2^n / n^{3/2}$

FIGURE 12. The number of rooted trees of type plane/unordered and general/binary for $n = 1\,..\,8$ and the corresponding asymptotic forms where $\lambda \doteq 0.43992$, $\beta \doteq 2.95576$; $\lambda_2 \doteq 0.79160$, $\beta_2 \doteq 2.48325$. For binary trees, size is by convention the number of external nodes.

and–or connectives in alternation (no "negation"), upon starting from the top in some conventional fashion (e.g, with an and-clause); see Figure 11[6]. Hipparchus was naturally not cognizant of generating functions, but with the technology of the time (and a rather remarkable mind!), he would still be able to discover a recurrence equivalent to (45),

$$S_n = [\![n \geq 2]\!] \left(\sum_{n_1+\cdots+n_k=n} S_{n_1} S_{n_2} \cdots S_n \right) + [\![n = 1]\!], \tag{46}$$

where the sum has only 42 essentially different terms for $n = 10$ (see [**136**] for a discussion), and finally determine S_{10}. □

▷ **31.** *The Lagrangean form of Schröder's GF.* The generating function $S(z)$ admits the form

$$S(z) = z\phi(S(z)) \quad \text{where} \quad \phi(y) = \frac{1-y}{1-2y}$$

is the OGF of compositions. Consequently, one has

$$\begin{aligned} S_n &= \frac{1}{n}[u^{n-1}] \left(\frac{1-u}{1-2u} \right)^n \\ &= \frac{(-1)^{n-1}}{n} \sum_k (-2)^k \binom{n}{k+1} \binom{n+k-1}{k} \\ &= \frac{1}{n} \sum_{k=0}^{n-2} \binom{2n-k-2}{n-1} \binom{n-2}{k}. \end{aligned}$$

Is there a direct combinatorial relation to compositions? ◁

▷ **32.** *Faster determination of Schröder numbers.* By forming a differential equation satisfied by $S(z)$ and extracting coefficients, one obtains a recurrence

$$(n+2)S_{n+2} - 3(2n+1)S_{n+1} + (n-1)S_n = 0,$$

that entails a fast determination (in linear time) of the S_n. In contrast, Hipparchus's recurrence implies an algorithm of complexity $e^{O(\sqrt{n})}$ in the number of arithmetic operations involved. ◁

[6] Any functional term admits a unique tree representation. Here, as soon as the root type has been fixed (e.g., an $\wedge$ connective), the others are determined by level parity. The constraint of node degrees ≥ 2 in the tree means that no superfluous connectives are used. Finally, any monotone boolean expression can be represented by a series-parallel network: the x_j are viewed as switches with the *true* and *false* values being associated with closed and open circuits, respectively.

I. 5.2. Nonplane tree. An *unordered tree*, also called *nonplane* tree, is a tree in the general graph–theoretic sense, so that there is no order distinction between subtrees emanating from a common node. The unordered trees considered here are furthermore rooted, meaning that one of the nodes is distinguished as the root. Accordingly, in the language of constructible structures, a rooted *unordered* tree is a root node linked to a *multiset* of trees. Thus, the class $\mathcal{H}$ of all unordered trees, admits the recursive specification

$$\mathcal{H} = \mathcal{Z} \times \mathfrak{M}\{\mathcal{H}\},$$

which translates into the *functional equation*

$$\begin{aligned} H(z) &= z(1-z)^{-H_1}(1-z^2)^{-H_2}(1-z^3)^{-H_3}\cdots \\ &= z\exp\left(H(z)+\frac{1}{2}H(z^2)+\frac{1}{3}H(z^3)+\cdots\right). \end{aligned}$$

The first form is due to Cayley in 1857 [**17**, p. 43]; it does not admit a closed form solution, though the equation permits one to determine all the H_n recurrently (*EIS* **A000081**)

$$H(z) = z + z^2 + 2z^3 + 4z^4 + 9z^5 + 20z^6 + 48z^7 + 115z^8 + 286z^9 + 719z^{10} + \cdots .$$

In addition, the local analysis of the singularities of $H(z)$ (Chapter 4) yields a *bona fide* asymptotic expansion for H_n, a fact first discovered by Pólya [**115**] who proved that

$$H_n \sim \lambda \cdot \frac{\beta^n}{n^{3/2}}, \tag{47}$$

for some positive constants $\lambda \doteq 0.43992$ and $\beta \doteq 2.95576$.

▷ **33.** *Fast determination of the Cayley–Pólya numbers.* Logarithmic differentiation of the equation satisfied by $H(z)$ provides for the H_n a recurrence that permits one to compute H_n in time polynomial in n. (Note: a similar technique applies to the partition numbers P_n; see p. 23.) ◁

The enumeration of the class of trees defined by an arbitrary set Ω of nodes degree immediately results from the translation of sets of fixed cardinality.

PROPOSITION I.5. *Let $\Omega \subset \mathbb{N}$ be a finite set of integers containing 0. Define the "exponential characteristic"*

$$\overline{\phi}(u) = \sum_{\omega\in\Omega} \frac{u^\omega}{\omega!}.$$

The OGF $U(z)$ of nonplane trees with degrees constrained to lie in Ω satisfies the functional equation

$$U(z) = z\overline{\phi}(U(z)) + z\Phi(U(z^2), U(z^3), \ldots),$$

for some polynomial Φ.

PROOF. The class of trees satisfies the combinatorial equation,

$$\mathcal{U} = \mathcal{Z} \times \mathfrak{M}_\Omega\{\mathcal{U}\} \qquad \left(\mathfrak{M}_\Omega\{\mathcal{U}\} \equiv \sum_{\omega\in\Omega} \mathfrak{M}_\omega\{\mathcal{U}\}\right),$$

where the multiset construction reflects non-planarity, since subtrees stemming from a node can be freely rearranged between themselves and may appear repeated. Theorem I.2 implies that the translation of $\mathfrak{M}_k\{\mathcal{A}\}$ is $A(z)^k/k!$ plus a polynomial form in $\{A(z^k)\}_{k\geq 2}$; the result follows. □

Once more, there are no explicit formulæ but only functional equations implicitly determining the generating functions. However, as we shall see in Chapter 4, the equations may be used to analyse the dominant singularity of $U(z)$. It is found that a "universal" law governs the singularities of simple tree generating functions that are of the type $\sqrt{1-z/\rho}$, corresponding to a general asymptotic scheme (see Figure 12),

$$U_n^{\Omega} \sim \lambda_{\Omega} \frac{(\beta_{\Omega})^n}{\sqrt{n^3}}. \tag{48}$$

Many of these questions have their origin in combinatorial chemistry, starting with Cayley in the 19th century [**17**, Ch. 4]. Pólya reexamined these questions, and in his important paper published in 1937 [**113**] he developed at the same time a general theory of combinatorial enumerations under group actions and of asymptotics methods giving rise to estimates like (48). See the book by Harary and Palmer [**76**] for more on this topic or Read's edition of Pólya's paper [**115**].

▷ **34.** *Binary nonplane trees.* Unordered binary trees with size measured by the number of external nodes are described by the equation $\mathcal{U} = \mathcal{Z} + \mathfrak{M}_2\{\mathcal{U}\}$. The functional equation determining $U(z)$ is

$$U(z) = z + \frac{1}{2}U(z)^2 + \frac{1}{2}U(z^2); \quad U(z) = z + z^2 + z^3 + 2z^4 + 3z^5 + \cdots. \tag{49}$$

The asymptotic analysis of the coefficients (*EIS* **A001190**) was carried out by Otter [**111**] who established an estimate of type (48). (The values of the constants are summarized in Figure 12.) The quantity U_n is also the number of structurally distinct products of n elements under a commutative nonassociative binary operation. ◁

▷ **35.** *Hierarchies.* Define the class $\mathcal{K}$ of hierarchies to be trees without nodes of outdegree 1 and size determined by the number of external nodes. The corresponding OGF satisfies (Cayley 1857, see [**17**, p.43])

$$K(z) = \frac{1}{2}z + \frac{1}{2}\left[\exp(K(z) + \frac{1}{2}K(z^2) + \cdots) - 1\right],$$

from which the first values are found (*EIS* **A000669**)

$$K(z) = z + z^2 + 2z^3 + 5z^4 + 12z^5 + 33z^6 + 90z^7 + 261z^8 + 766z^9 + 2312z^{10} + \cdots.$$

These numbers also enumerate topologically equivalent series-parallel networks (with no plane embedding imposed) as well as hierarchies in statistical classification theory [**142**]. They are the nonplanar analogues of the Hipparchus–Schröder's numbers on p. 43. ◁

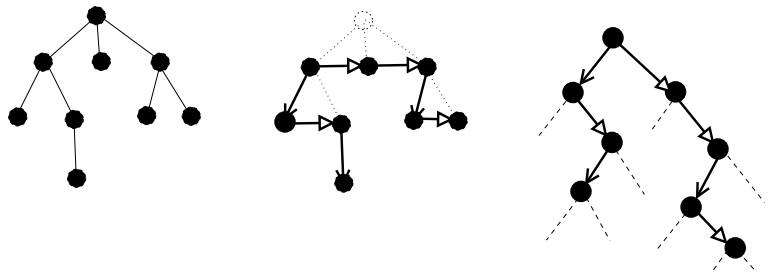
I. 5.3. Tree related constructions. Trees underlie recursive structures of all sorts. A first illustration is provided by the fact that the Catalan numbers, $\mathrm{C}_n = \frac{1}{n+1}\binom{2n}{n}$ count general trees ($\mathcal{G}$) of size $n+1$, binary trees ($\mathcal{B}$) of size n (if size is defined as the number of internal nodes), as well as triangulations ($\mathcal{T}$) comprised of n triangles. The combinatorialist John Riordan even coined the name "Catalan domain" for the area within combinatorics that deals with objects enumerated by Catalan numbers, and Stanley's book contains an exercise [**137**, Ex. 6.19] whose statement alone spans ten full pages, with a lists of 66 types of objects(!) belonging to the Catalan domain. We shall illustrate the importance of Catalan numbers by describing a few fundamental correspondences the "explain" the occurrence of Catalan numbers in relation to the already encountered classes $\mathcal{G}, \mathcal{B}, \mathcal{T}$.

The combinatorial isomorphism relating $\mathcal{G}$ and $\mathcal{B}$ (albeit with a shift in size) coincides with a classical technique of computer science [**85**, Sec. 2.3.2]. To wit, a general tree can be represented in such a way that every node has two types of links, one pointing to the leftmost child, the other to the next sibling in left-to-right order. Under this representation, if the root of the general tree is left aside, then every node is linked to two other (possibly

empty) subtrees. In other words, general trees with n nodes are equinumerous with pruned binary trees with $n-1$ nodes:

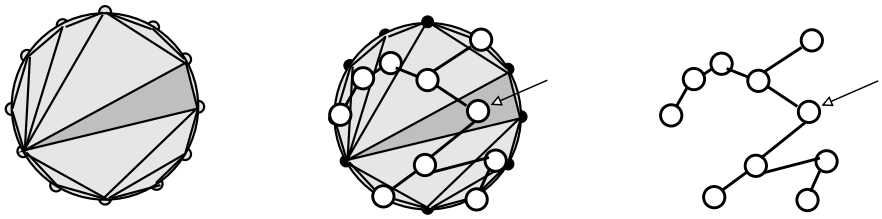
$$\mathcal{G}_n \cong \mathcal{B}_{n-1}.$$

Graphically, this is illustrated as follows:



The rightmost tree is a binary tree drawn in a conventional manner, following a 45° tilt. This justifies the name of "rotation correspondence" often given to this transformation.

The relation betwen binary trees $\mathcal{B}$ and triangulations $\mathcal{T}$ is equally simple: draw a triangulation; define the root triangle as the one that contains the edge connecting two designated vertices (for instance, the vertices numbered 0 and 1); associate to the root triangle the root of a binary tree; next, associate recursively to the subtriangulation on the left of the root triangle a left subtree; do similarly for the right subtriangulation giving rise to a right subtree.



Under this correspondence, tree nodes correspond to triangle faces, while edges connect adjacent triangles. What this correspondence proves is the combinatorial isomorphism

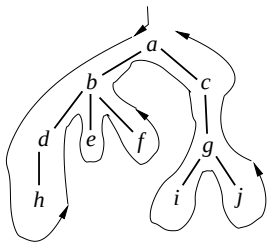
$$\mathcal{T}_n \cong \mathcal{B}_n.$$

We turn next to different types of objects that are in correspondence with trees. These can be interpreted as words encoding tree traversals, and interpretd geometrically as paths in the discrete plane $\mathbb{Z} \times \mathbb{Z}$.

EXAMPLE 10. *Tree codes and Łukasiewicz words.* Any tree can be traversed starting from the root, proceeding depth-first (and left-to-right), and backtracking upwards once a

subtree has been completely traversed. For instance, in the tree

(50) $\tau =$



the first visits to nodes take place in the following order

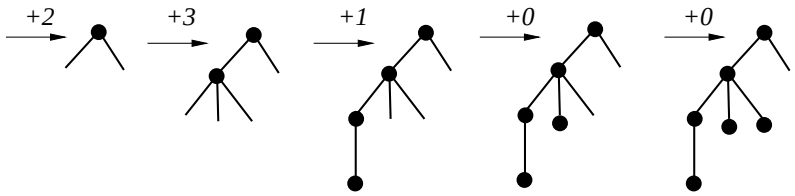
$$a, \quad b, \quad d, \quad h, \quad e, \quad f, \quad c, \quad g, \quad i, \quad j\,.$$

(Note: the tags $a, b, \ldots$ added for convenience in order to distinguish nodes have no special meaning; only the abstract tree shape matters here.) This order is known as *preorder* or *prefix order* since a node is preferentially visited before its children.

Given a tree, the listing of the outdegrees of nodes in prefix order will be called the preorder degree sequence. For the tree of (50), this is

$$\sigma = (2, 3, 1, 0, 0, 0, 1, 2, 0, 0).$$

It is a fact that the degree sequence determines the tree unambiguously. Indeed, given the degree sequence, the tree is reconstructed step by step, adding nodes one after the other at the leftmost available place. For σ, the first steps are then



Next, if one represents degree j by a "symbol" f_j, then the degree sequence becomes a *word* over the infinite alphabet $\mathcal{F} = \{f_0, f_1, \ldots\}$, for instance,

$$\sigma \rightsquigarrow f_2 f_3 f_1 f_0 f_0 f_0 f_1 f_2 f_0 f_0.$$

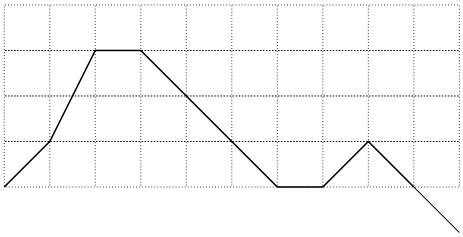
This can be interpreted in logical language a denotation for a functional term built out symbols from $\mathcal{F}$, where f_j represents a "function" of degree j. The correspondence even becomes obvious if superfluous parentheses are added at appropriate place to delimitate scope:

$$\sigma \rightsquigarrow f_2(f_3(f_1(f_0), f_0, f_0), f_1(f_2(f_0, f_0))).$$

Such codes are known as Łukasiewicz codes[7], in recognition of the work of the Polish logician with that name. Jan Łukasiewicz (1878–1956) introduced them in order to completely specify the *syntax* of terms in various logical calculi; they prove nowadays basic in the development of parsers and compilers in computer science.

[7] A less dignified name is "Polish prefix notation". The "reverse Polish notation" is a variant based on postorder that has been used in calculators since the 1970's.

Finally, a tree code can be rendered as a walk over the discrete lattice $\mathbb{Z} \times \mathbb{Z}$. Associate to any f_j (i.e., any node of outdegree j) the displacement $(1, j-1) \in \mathbb{Z} \times \mathbb{Z}$, and plot the sequence of moves starting from the origin. On the example one finds:

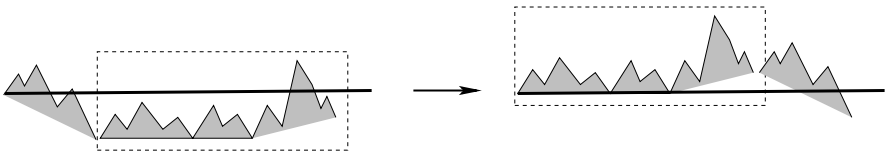


f_2	f_3	f_1	f_0	f_0	f_0	f_1	f_2	f_0	f_0
1	2	0	-1	-1	-1	0	1	-1	-1

There, the last line represents the vertical displacements. The resulting paths are known as Łukasiewicz paths. Such a walk is then characterized by two conditions: the vertical displacements are in the set $\{-1, 0, 1, 2, \ldots\}$; all the its points, except for the very last step, are always in the upper half-plane.

By this correspondence, the number of Łukasiewicz paths with n steps is the shifted Catalan number, $\frac{1}{n}\binom{2n-2}{n-1}$. □

▷ **36.** *Conjugacy principle and cycle lemma.* Let $\mathcal{L}$ be the class of all Łukasiewicz paths. Define a "relaxed" path as one that starts at level 0, ends at level -1 but is otherwise allowed arbitrary negative steps; let $\mathcal{M}$ be the corresponding class. Then, each relaxed path can be cut-and-pasted uniquely after its leftmost minimum as described here:



This associates to every relaxed path of length ν a unique standard path. A bit of combinatorial reasoning shows that correspondence is 1-to-ν (each element of $\mathcal{L}$ has *exactly* ν preimages.) One thus has $M_\nu = \nu L_\nu$. This correspondence preserves the number of steps of each type $(f_0, f_1, \ldots)$, so that the number of Łukasiewicz with ν_j steps of type f_j is

$$\frac{1}{\nu}[x^{-1}u_0^{\nu_0}u_1^{\nu_1}\cdots]\left(x^{-1}u_0 + u_1 + xu_2 + x^2u_3 + \cdots\right)^\nu = \frac{1}{\nu}\binom{\nu}{\nu_0, \nu_1, \ldots},$$

under the necessary condition $(-1)\nu_0 + 0\nu_1 + 1\nu_2 + 2\nu_3 + \cdots = -1$.

This combinatorial way of obtaining refined Catalan statistics is known as the "conjugacy principle" [**119**] or the "cycle lemma" [**40**]. Raney has derived from it a purely combinatorial proof of the Lagrange inversion formula [**119**] while Dvoretzky & Motzkin [**40**] have employed this technique to solve a number of counting problems related to circular arrangements. ◁

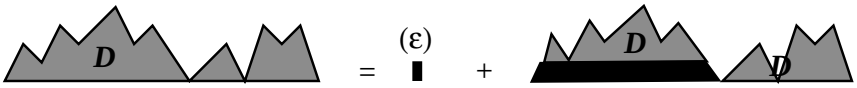
EXAMPLE 11. *Binary tree codes and Dyck paths.* Walks associated with binary trees have a very special form since the vertical displacements can only be $+1$ or -1. The resulting paths of Łukasiewicz type are then equivalently characterized as sequences of numbers $x = (x_0, x_1, \ldots, x_{2n}, x_{2n+1})$ satisfying the conditions

$$x_0 = 0; \qquad x_j \geq 0 \quad \text{for } 1 \leq j \leq 2n; \qquad |x_{j+1} - x_j| = 1; \qquad x_{2n+1} = -1. \tag{51}$$

These coincide with "gambler ruin sequences", a familiar object from probability theory: a player plays head and tails. He starts with no capital ($x_0 = 0$) at time 0; his total gain is x_j at time j; he is allowed no credit ($x_j \geq 0$) and loses at the very end of the game $x_{2n+1} = -1$; his gains are ± 1 depending on the outcome of the coin tosses ($|x_{j+1} - x_j| = 1$).

It is customary to drop the final step and consider "excursions' that take place in the upper half-plane. The resulting objects defined as sequences $(x_0 = 0, x_1, \ldots, x_{2n} = 0)$ satisfying the first three conditions of (51) are known as *Dyck paths*[8]. By construction, Dyck paths of length $2n$ correspond bijectively to binary trees with n internal nodes and are consequently enumerated by Catalan numbers. Let $\mathcal{D}$ be the combinatorial class of Dyck paths, with size defined as length. This property can also be checked directly: the quadratic decomposition

(52)



$$\mathcal{D} \quad = \quad \{\epsilon\} \; + \; (\nearrow \mathcal{D} \searrow) \times \mathcal{D}$$

induces for the OGF of Dyck paths the quadratic equation

$$D(z) = 1 + (zD(z)z)\, D(z),$$

from which the Catalan GF results, and $D_{2n} = \frac{1}{n+1}\binom{2n}{n}$, as expected. The decomposition (52) is known as the "first passage" decomposition as it is based on the first time the cumulated gains in the coin-tossing game pass through the value zero.

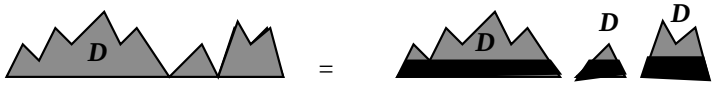
Dyck paths also arise in connection will well-parenthetized expressions. These are recognized by keeping a counter that records at each stage the excess of the number of opening brackets '(' over closing brackets ')'. Finally, one of the origins of Dyck path is the famous "ballot problem", which goes back to the nineteenth century [**99**]: there are two candidates A and B that stand for election, $2n$ voters, and the election eventually results in a tie; what is the probability that A is always ahead of or tied with B when the ballots are counted? The answer is

$$\frac{D_{2n}}{\binom{2n}{n}} = \frac{1}{n+1},$$

since there are $\binom{2n}{n}$ possibilities in total, of which the number of favorable cases is D_{2n}, a Catalan number. The central rôle of Dyck paths and Catalan numbers in problems coming from such diverse areas of science is quite remarkable. □

▷ **37.** *Dyck paths and general trees.* The class of Dyck paths admits an alternative sequence decomposition

(53)



$$\mathcal{D} = \mathfrak{S}\{\mathcal{Z} \times \mathcal{D} \times \mathcal{Z}\},$$

[8]Dyck paths are closely associated with free groups on one generator and are named after the German mathematician Walther (von) Dyck (1856–1934) who introduced free groups around 1880.

which again leads to the Catalan GF. The decomposition (53) is known as the "arch decomposition". It can also be directly related to traversal sequences of general trees, but with the directions of *edge* traversals being recorded (instead of traversals based on node degrees). $\triangleleft$

$\triangleright$ **38.** *Random generation of Dyck paths.* Dyck paths of length $2n$ can be generated uniformly at random in time linear in n. (Hint: By the conjugacy principle of Ex. 36, it suffices to generate uniformly a sequence of n a's and $n+1$ b's, then reorganize it according to the conjugacy principle. $\triangleleft$

$\triangleright$ **39.** *Motzkin paths and unary-binary trees.* Motzkin paths are defined by changing the third condition of (51) defining Dyck paths into $|x_{j+1} - x_j| \leq 1$. They appear as codes for unary-binary trees and are enumerated by the Motzkin numbers of Ex. 29. $\triangleleft$

EXAMPLE 12. *The complexity of boolean functions.* Complexity theory provides many surprising applications of enumerative combinatorics and asymptotic estimates. In general, one starts with a finite set of mathematical objects Ω and a combinatorial class $\mathcal{D}$ of "descriptions". By assumption, to every object of $\delta \in \mathcal{D}$ is associated an element $\mu(\delta) \in \Omega$, its "meaning"; conversely any object of Ω admits at least one description in $\mathcal{D}$, that is, the function μ is surjective. It is then of interest to quantify properties of the shortest description function defined for $\omega \in \Omega$ as

$$\sigma(\omega) := \min \left\{ |\delta|_{\mathcal{D}} \;\middle|\; \mu(\delta) = \omega \right\},$$

and called the "complexity" of element of Ω (with respect to $\mathcal{D}$).

We take here Ω to be the class of all boolean functions on m variables. Their number is $\|\Omega\| = 2^{2^m}$. As descriptions, we adopt the class of logical expressions involving the logical connectives $\vee, \wedge$ and pure or negated variables. Equivalently, $\mathcal{D}$ is the class of binary trees, where internal nodes are tagged by a logical disjunction ('$\vee$') or a conjunction ('$\wedge$'); each external node is tagged by either a boolean variable of $\{x_1, \ldots, x_m\}$ or a negated variable of $\{\neg x_1, \ldots, \neg x_m\}$. Define the size of a tree description as the number of internal nodes, that is, the number of logical operators. Then, one has

$$D_n = \left(\frac{1}{n+1} \binom{2n}{n} \right) \cdot 2^n \cdot (2m)^{n+1}, \tag{54}$$

as seen by counting tree shapes and possibilities for internal as well as external node tags.

The crux of the matter is that if the inequality

$$\sum_{j=0}^{\nu} D_j < \|\Omega\|, \tag{55}$$

holds, then there are not enough descriptions of size $\leq \nu$ to exhaust Ω. In other terms, there must exist at least one object in Ω whose complexity exceeds ν. If the left side of (55) is much smaller than the right side, then, it must even be the case that "most" Ω-objects have a complexity that exceeds ν.

In the case of boolean functions and tree descriptions, the asymptotic form (12) is available. There results from (54) that, for n, ν getting large, one has

$$D_n = O(16^n m^n n^{-3/2}), \qquad \sum_{j=0}^{\nu} D_j = O(16^\nu m^\nu \nu^{-3/2}).$$

Choose ν such that the second expression is $o(\|\Omega\|)$. This is ensured for instance by taking for ν the value

$$\nu(m) := \frac{2^m}{\log_2 m},$$

as verified by a simple asymptotic calculation. With this choice, one has the following suggestive statement:

> *A fraction tending to 1 (as $m \to \infty$) of boolean functions in m variables have tree complexity at least $2^m / \log_2 m$.*

Regarding upper bounds on boolean function complexity, a function always has a tree complexity that is at most $2^{m+1} - 3$. To see it, note that for $m = 1$, the 4 functions are

$$0 \equiv (x_1 \wedge \neg x_1), \quad 1 \equiv (x_1 \vee \neg x_1), \quad x_1, \quad \neg x_1.$$

Next, a function of m variables is representable by a technique known as the binary decision tree (BDT),

$$f(x_1, \ldots, x_{m-1}, x_m) = (\neg x_m \wedge f(x_1, \ldots, x_{m-1}, 0)) \vee (x_m \wedge f(x_1, \ldots, x_{m-1}, 1)),$$

which provides the basis of the induction as it reduces the representation of an m-ary function to the representation of two $(m-1)$-ary functions, consuming on the way three logical connectives.

Altogether, basic counting arguments have shown that "most" boolean functions have a tree-complexity that is "close" to the maximum possible, namely, $O(2^m)$. A similar result has been established by Shannon for the measure called circuit complexity: circuits are more powerful than trees, but Shannon's result states that *almost all boolean functions of m variables have circuit complexity $O(2^m/m)$*. See [**143**], especially the chapter by Li and Vitányi, for a discussion of such counting techniques within the framework of complexity theory. □

We finally conclude with a vast generalization of the previous examples.

DEFINITION I.9. *A class $\mathcal{T}$ of trees is said to be a* context-free variety of trees *if it coincides with the first component of a system of equations ($\mathcal{T} = \mathcal{S}_1$) of a recursive system*

$$\left\{ \begin{array}{lcl} \mathcal{S}_1 & = & \Phi_1(\mathcal{Z}, \mathcal{S}_1, \ldots, \mathcal{S}_r) \\ \vdots & \vdots & \vdots \\ \mathcal{S}_r & = & \Phi_r(\mathcal{Z}, \mathcal{S}_1, \ldots, \mathcal{S}_r), \end{array} \right. \tag{56}$$

where each Φ_j is a constructor that involves only the operations of combinatorial sum ($+$) and cartesian product ($\times$).

A combinatorial class $\mathcal{C}$ is said to be context-free *if it is combinatorially isomorphic to a context-free variety of trees: $\mathcal{C} \cong \mathcal{T}$.*

The classes of general trees ($\mathcal{G}$) and binary trees ($\mathcal{B}$) are context-free varieties of trees since they are specifiable as

$$\left\{ \begin{array}{lcl} \mathcal{G} & = & \mathcal{Z} \times \mathcal{F} \\ \mathcal{F} & = & \{\epsilon\} + (\mathcal{G} \times \mathcal{F}) \end{array} \right. , \qquad \mathcal{B} = \mathcal{Z} + (\mathcal{B} \times \mathcal{B}).$$

($\mathcal{F}$ designates ordered forests of general trees.) The Łukasiewicz language and the set of Dyck paths are context-free classes since they are bijectively equivalent to $\mathcal{G}$ and $\mathcal{T}$.

This terminology is an extension of the concept of context-free language in the theory of formal languages; there, one defines a context-free language as the language formed with words that are obtained as sequences of leaf tags (read in left-to-right order) of a context-free variety of trees. In formal linguistics, the one-to-one mapping between trees and words is not generally imposed; when it is satisfied, the context-free language is said to be *unambiguous*, since words and trees determine each other uniquely.

An immediate consequence of admissibility theorems is the following proposition first encountered by Chomsky and Schútzenberger [**26**] in the course of their research relating formal languages and formal power series:

PROPOSITION I.6. *A combinatorial class $\mathcal{C}$ that is* context-free *admits an OGF that is an* algebraic function. *In other words, there exists a bivariate polynomial $P(z, y) \in \mathbb{C}[z, y]$ such that*

$$P(z, C(z)) = 0.$$

PROOF. The context-free system (56) translates into a system

$$\left\{ \begin{array}{lcl} S_1(z) & = & \Psi_1(z, S_1(z), \dots, S_r(z)) \\ \vdots & \vdots & \vdots \\ S_r(z) & = & \Psi_r(z, S_1(z), \dots, S_r(z)), \end{array} \right.$$

where the Ψ_j are polynomials. This follows by the basic sum and product rules.

It is then well-known that algebraic elimination is possible in polynomials systems. here, it is possible to eliminate the auxiliary variables $S_2, \dots, S_r$, one by one, preserving the polynomial character of the system at each stage. The end result is then a single polynomial equation satisfied by $C(z) \equiv S_1(z)$.

Methods for performing polynomial elimination are well-known in algebra: one may appeal to a repeated use of resultants [**94**] or to Groebner basis algorithms. See Lang's classic treatise on algebra for resultants [**95**, V.§10] and the excellent introduction to Groebner bases provided by Cox, Little, and O'Shea in [**31**]. □

Proposition I. 5.3 justifies the importance of algebraic functions in enumerative theory and it will be put to use in later chapters of this book. It constitutes a counterpart of Proposition I.3 which asserts that rational generating functions arise from finite state devices.

I. 6. Additional constructions

This section is devoted to the presentation of two types of mechanisms that enrich the framework of constructions: the constructions of pointing and substitution, as well as the use of implicit combinatorial definitions,

I. 6.1. Pointing and substitution. Two more constructions, namely pointing and substitution, translate agreeably into generating functions. Combinatorial structures are viewed here as formed of "atoms" (words are composed of letters, graphs of nodes, etc) which determine their sizes. In this context, pointing means "pointing at a distinguished atom"; substitution, written $\mathcal{B} \circ \mathcal{C}$ or $\mathcal{B}[\mathcal{C}]$, means "substitute elements of $\mathcal{C}$ for atoms of $\mathcal{B}$".

DEFINITION I.10. *Let $\{\epsilon_1, \epsilon_2, \dots\}$ be a fixed collection of distinct neutral objects of size 0. The* pointing *of a class $\mathcal{B}$, noted $\mathcal{A} = \Theta\mathcal{B}$, is formally defined by*

$$\Theta\mathcal{B} := \sum_{n \geq 0} \mathcal{B}_n \times \{\epsilon_1, \dots, \epsilon_n\}.$$

The substitution *of $\mathcal{C}$ into $\mathcal{B}$ (also known as composition of $\mathcal{B}$ and $\mathcal{C}$), noted $\mathcal{B} \circ \mathcal{C}$ or $\mathcal{B}[\mathcal{C}]$, is formally defined as*

$$\mathcal{B} \circ \mathcal{C} \equiv \mathcal{B}[\mathcal{C}] := \sum_{k \geq 0} \mathcal{B}_k \times \mathfrak{S}_k\{\mathcal{C}\}.$$

If B_n is the number of $\mathcal{B}$ structures of size n, then nB_n can be interpreted as counting pointed structures where *one* of the n atoms composing a $\mathcal{B}$-structure has been distinguished (here by a special "pointer" of size 0 attached to it). Elements of $\mathcal{B} \circ \mathcal{C}$ may also be viewed as obtained by selecting in all possible ways an element $\beta \in \mathcal{B}$ and replacing each of its atoms by an arbitrary element of $\mathcal{C}$.

The interpretations above rely (silently) on the fact that atoms in an object can be eventually distinguished from each other. This can be obtained by "canonicalizing" [9] the representations of objects: first define inductively the lexicographic ordering for products and sequences; next represent powersets and multisets as increasing sequences with the induced lexicographic ordering (more complicated rules can also canonicalize cycles). In this way, any constructible object admits a unique "rigid" representation in which each particular atom is determined by its place. Such a canonicalization thus reconciles the abstract definition, Definition I.10, and the intuitive interpretation of pointing and substitution.

THEOREM I.3 (Pointing and substitution). *The constructions of pointing and substitution are admissible*[10]*:*

$$\begin{aligned} \mathcal{A} = \Theta\mathcal{B} \quad &\Longrightarrow \quad A(z) = z\partial_z B(z) \quad \partial_z := \frac{d}{dz} \\ \mathcal{A} = \mathcal{B} \circ \mathcal{C} \quad &\Longrightarrow \quad A(z) = B(C(z)) \end{aligned}$$

PROOF. By the definition of pointing, one has

$$A_n = n \cdot B_n \qquad \text{and} \qquad A(z) = z\frac{d}{dz}B(z).$$

From the definition of substitution, $\mathcal{A} = \mathcal{B}[\mathcal{C}]$ implies, by the sum and product rules,

$$A(z) = \sum_{k \geq 0} B_k \cdot (C(z))^k = B(C(z)),$$

and the proof is completed. □

▷ **40.** *Combinatorics of derivatives.* The combinatorial operation **D** of "eraser–pointing" points to an atom in an object and replaces it by a neutral object, otherwise preserving the overall structure of the object. The translation of **D** on OGFs is then simply $\partial \equiv \partial_z$. Classical identities of analysis then receive simple combinatorial interpretations, for instance,

$$\partial(A \times B) = (A \times \partial B) + (\partial A) \times B);$$

Leibniz's identity, $\partial^m(f \cdot g) = \sum_j \binom{m}{j} (\partial^j f) \cdot (\partial^{m-j} g)$, also follows from basic combinatorics. Similarly, for the "chain rule" $\partial(f \circ g) = ((\partial f) \circ g) \cdot \partial g$. ◁

As an example of pointing, consider the class $\mathcal{P}$ of all permutations written as words over integers starting from 1. One can go from a permutation of size $n-1$ to a permutation of size n by selecting a "gap" and inserting the value n. When this is done in all possible ways, it gives rise to the combinatorial relation

$$\mathcal{P} = \mathcal{E} + \Theta(\mathcal{Z} \times \mathcal{P}), \qquad \mathcal{E} = \{\epsilon\},$$

[9] Such canonicalization techniques also serve to develop fast algorithms for the exhaustive listing of objects of a given size as well as for the range of problems known as "ranking" and "unranking", with implications in fast random generation. See, e.g., [**103, 109, 152**] for the general theory as well as [**118, 157**] for particular cases like necklaces and trees.

[10] In this book, we borrow from differential algebra the convenient notation $\partial_z := \frac{d}{dz}$ to represent derivatives.

and to the corresponding ordinary differential equation for the OGF,

$$P(z) = 1 + z\frac{d}{dz}(zP(z)),$$

whose formal solution is $P(z) = \sum_{n\geq 0} n!z^n$.

As an example of substitution, consider the class $\mathcal{B}$ of (plane rooted) binary trees, where all nodes contribute to size. If at each node there is substituted a linear chain of nodes (linked by edges placed on top of the node), one forms an element of the class $\mathcal{M}$ of unary-binary trees; in symbols:

$$\mathcal{M} = \mathcal{B} \circ \mathfrak{S}\{\mathcal{Z}\} \quad \text{and} \quad M(z) = B\left(\frac{z}{1-z}\right).$$

Thus from the known OGF, $B(z) = (1 - \sqrt{1-4z^2})/(2z)$, one derives

$$M(z) = \frac{1 - \sqrt{1 - 4z^2(1-z)^{-2}}}{2z(1-z)^{-1}} = \frac{1 - z - \sqrt{1 - 2z - 3z^2}}{2z},$$

which matches the direct derivation on p. 43 (Motzkin numbers).

I. 6.2. Implicit structures. There are many cases where a combinatorial class $\mathcal{X}$ is determined by a relation $\mathcal{A} = \mathcal{B} + \mathcal{X}$, where $\mathcal{A}$ and $\mathcal{B}$ are known. In terms of generating functions, one has $A(z) = B(z) + X(z)$, so that

$$\mathcal{A} = \mathcal{B} + \mathcal{X} \quad \Longrightarrow \quad X(z) = A(z) - B(z).$$

For instance, the autocorrelation technique of Section I. 4.2 makes it possible to describe the class $\mathcal{S}$ of all words in $\mathcal{W}$ that do *not* contain a given pattern $\mathfrak{p}$, whereas the language of words containing the pattern is determined as the solution in $\mathcal{X}$ of the equation $\mathcal{W} = \mathcal{S}+\mathcal{X}$; see p. 36. Similarly, for products, basic algebra gives

$$\mathcal{A} = \mathcal{B} \times \mathcal{X} \quad \Longrightarrow \quad X(z) = \frac{A(z)}{B(z)}.$$

Here are the corresponding solutions for two of the composite constructions.

THEOREM I.4 (Implicit specifications). *The generating functions associated to the implicit equations in $\mathcal{X}$*

$$\mathcal{A} = \mathfrak{S}\{\mathcal{X}\}, \quad \mathcal{A} = \mathfrak{M}\{\mathcal{X}\}$$

are respectively

$$X(z) = 1 - \frac{1}{A(z)}, \quad X(z) = \sum_{k\geq 1} \frac{\mu(k)}{k} \log A(z^k),$$

where $\mu(k)$ is the Moebius function.

PROOF. For sequences, the relation $A(z) = (1 - X(z))^{-1}$ is readily inverted. For multisets, start from the fundamental relation of Theorem I.1 and take logarithms:

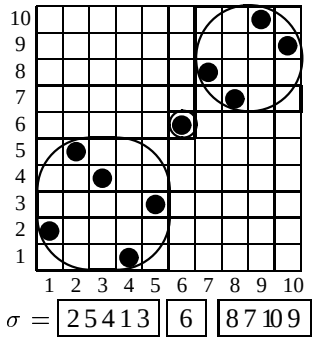
$$\log(A(z)) = \sum_{k=1}^{\infty} \frac{1}{k} X(z^k).$$

Let $L = \log A$ and $L_n = [z^n]L(z)$. One has

$$nL_n = \sum_{d \mid n} (dX_d),$$

to which it suffices to apply Moebius inversion; see APPENDIX: *Arithmetical functions*, p. 165. □

EXAMPLE 13. *Indecomposable permutations.* A permutation $\sigma = \sigma_1 \cdots \sigma_n$ (written here as a word of distinct letters) is said to be *decomposable* if, for some $k < n$, $\sigma_1 \cdots \sigma_k$ is a permutation of $\{\sigma_1, \ldots, \sigma_k\}$, *i.e.*, a strict prefix of the permutation is itself a permutation. Any permutation decomposes uniquely as a catenation of indecomposable permutations; for instance, here is the decomposition of $\sigma = 2\,5\,4\,1\,3\,6\,8\,7\,10\,9$:



Thus the class $\mathcal{P}$ of all permutations and the class $\mathcal{I}$ of indecomposable ones are related by

$$\mathcal{P} = \mathfrak{S}\{\mathcal{I}\}.$$

This determines $I(z)$ implicitly, and Theorem I.4 gives:

$$I(z) = 1 - \frac{1}{P(z)} \quad \text{where} \quad P(z) = \sum_{n \geq 1} n!\, z^n .$$

This example illustrates the implicit structure theorem, but also the possibility of *bona fide* algebraic calculations with power series even in cases where they are divergent (APPENDIX: *Formal power series*, p. 169). One finds

$$I(z) = z + z^2 + 3\,z^3 + 13\,z^4 + 71\,z^5 + 461\,z^6 + +3447\,z^7 + \cdots ,$$

where the coefficients are *EIS* **A003319** and

$$I_n = n! - \sum_{\substack{n_1+n_2=n \\ n_1,n_2 \geq 1}} (n_1!n_2!) + \sum_{\substack{n_1+n_2+n_3=n \\ n_1,n_2,n_3 \geq 1}} (n_1!n_2!n_3!) - \cdots .$$

From there, simple majorizations of the terms imply that $I_n \sim n!$, so that *almost all permutations are indecomposable*; see [**28**, p. 262]. □

▷ **41.** *2-dimensional wanderings.* A drunkard starts from the origin in the $\mathbb{Z} \times \mathbb{Z}$ plane and, at each second, he makes a step in either one of the four directions, NW, NE, SW, SE. The steps are thus $\nwarrow, \nearrow, \swarrow, \searrow$. Consider the class $\mathcal{L}$ of "primitive loops" defined as walks that start and end at the origin, but do not otherwise touch the origin. The GF of $\mathcal{L}$ is (*EIS* **A002894**)

$$L(z) = 1 - \frac{1}{\sum_{n=0}^{\infty} \binom{2n}{n}^2 z^{2n}} = 4\,z^2 + 20\,z^4 + 176\,z^6 + 1876\,z^8 + \cdots .$$

(Hint: a walk is determined by its projections on the horizontal and vertical axes; 1-dimensional walks that return to the origin in $2n$ steps are enumerated by $\binom{2n}{n}$.) In particular $[z^n]L(z/4)$ is the probability that the random walk first returns to the origin in n steps.

Such problems largely originate with Pólya and the implicit structure technique above was most likely known to him [**114**]. See [**24**] for similar multidimensional extensions. ◁

EXAMPLE 14. *Irreducible polynomials over finite fields.* Objects apparently "non–combinatorial" can sometimes be enumerated by symbolic methods. Here is an indirect construction relative to polynomials over finite fields. We fix a prime number p and consider the base field $\mathbb{F}_p$ of integers taken modulo p. The polynomial ring $\mathbb{F}_p[X]$ is the ring of polynomials in X with coefficients taken in $\mathbb{F}_p$. For all practical purposes, one may restrict attention to polynomials that are monic, that is, whose leading coefficient is 1.

First, let $\mathcal{P}$ be the class of all monic polynomials, with the size of a polynomial being its degree. Since a monic polynomial of degree n is described by a choice of n coefficients, one has

$$P \cong \mathfrak{S}\{\mathbb{F}_p\} \quad \text{and} \quad P(z) = \frac{1}{1-pz}, \quad P_n = p^n.$$

A polynomial is said to be *irreducible* if it does not decompose as a product of two polynomials of smaller degrees. By unique factorization, each monic polynomial decomposes uniquely into a product (with repetitions being possible) of monic irreducible polynomials. For instance, over $\mathbb{F}_3$, one has

$$X^{10} + X^8 + 1 = (X+1)^2(X+2)^2(X^6 + 2X^2 + 1).$$

Let I be the set of monic irreducible polynomials. The combinatorial isomorphism

$$\mathcal{P} \cong \mathfrak{M}\{\mathcal{I}\}$$

expresses precisely the unique factorization property. Thus, the irreducibles are determined implicitly from the class of all polynomials whose OGF is known. Theorem I.4 implies the identity

$$I(z) = \sum_{k \geq 1} \frac{\mu(k)}{k} \log \frac{1}{1-pz^k},$$

and, upon extracting coefficients,

$$I_n = \frac{1}{n} \sum_{k \mid n} \mu_k p^{n/k}.$$

In particular, I_n is asymptotic to p^n/n. This estimate constitutes the density theorem for irreducible polynomials:

> *The fraction of irreducible polynomials amongst all polynomials of degree n over the finite field $\mathbb{F}_p$ is asymptotic to $\frac{1}{n}$.*

This property is analogous to the Prime Number Theorem of number theory (which is technically much harder [**32**]), after which the proportion of prime numbers in the interval $[1, n]$ is asymptotic to $\frac{1}{\log n}$. (The derivation above is in essence due to Gauß. See Knopfmacher's book [**83**] for an abstract discussion of statistical properties of arithmetical semigroups.) □

▷ **42.** *Square-free polynomials.* Let Q be the class of monic square-free polynomials (*i.e.*, polynomials not divisible by the square of a polynomial). One has by "Vallée's identity" (p. 14) $Q(z) = P(z)/P(z^2)$, hence

$$Q_n = p^n - p^{n-1} \qquad (n \geq 1).$$

Berlekamp's book [**14**] discusses such facts together with relations to error correcting codes. ◁

▷ **43.** *Balanced trees.* The class $\mathcal{O}$ of balanced 2-3 trees is a familiar data structure [**86**], defined as (rooted planar) trees whose internal nodes have degree 2 or 3 and such that all leaves are at the same distance from the root. Only leaves contribute to size. Balanced trees satisfy an implicit equation based on combinatorial substitution:

$$\mathcal{O} = \mathcal{Z} + \mathcal{O}[(\mathcal{Z} \times \mathcal{Z}) + (\mathcal{Z} \times \mathcal{Z} \times \mathcal{Z})], \qquad O(z) = z + O(z^2 + z^3).$$

Odlyzko [**110**] has determined the growth of O_n (it is like φ^n/n, where $\varphi = (1+\sqrt{5})/2$ is the golden ratio, but involves subtle fluctuations). $\lhd$

I. 7. Notes

There are several lessons to be learnt from the uses that we have surveyed of symbolic combinatorics.

First, for a given class of problems, symbolic methods lead to a unified treatment that reveals a natural class of functions in which generating functions lie. Thus denumerants with a finite set of coin denominations always lead to rational generating functions with poles on the unit circle. Such an observation is useful since then a common strategy for coefficient extraction can be applied, in such a case, based on partial fraction expansion. In the same vein, the run statistics constitute a particular case of the general theorem of Chomsky and Schützenberger to the effect that the generating function of a regular language is necessarily a rational function. Theorems of this sort establish a bridge between combinatorial analysis and special functions. The example of counting set partitions shows that application of the symbolic method may require finding an adequate presentation of the combinatorial structures to be counted. In this way, bijective combinatorics enters the game in a nontrivial fashion.

Second, our introductory examples of compositions and partitions correspond to classes of combinatorial structures with *explicit* "iterative" definitions, a fact leading in turn to explicit generating function expressions. The tree examples then introduce *recursively defined* structures. In that case, the recursive definition translates into a *functional equation* that only determines the generating function implicitly. In simpler situations (like binary or general trees), the equation can be solved and explicit counting results still follow. In other cases (like non-planar trees) one can usually proceed with complex asymptotic analysis directly from the functional equation and obtain very precise *asymptotic estimates*; see Chapters IV and V.

Modern presentations of combinatorial analysis appear in the books of Comtet [**28**] (a beautiful book largely example driven), Stanley [**135, 137**] (a rich set with an algebraic orientation), and Wilf [**153**] (generating functions oriented). An elementary but insightful presentation of the basic techniques appears in Graham, Knuth, and Patashnik's classic [**71**], a popular book with a highly original design. An encyclopedic reference is the book of Jackson & Goulden [**68**] whose descriptive approach very much parallels ours.

The sources of the modern approaches to combinatorial analysis are hard to trace since they are usually based on earlier traditions and informally stated mechanisms that were well mastered by practicing combinatorial analysts. (See for instance MacMahon's book [**101**] *Combinatory Analysis* first published in 1917, the introduction of denumerant generating functions by Pólya as exposed in [**116**], or the "domino theory" in [**71**, Sec. 7.1].) One source in recent times is the Chomsky–Schützenberger theory of formal languages and enumerations [**26**]. Rota [**122**] and Stanley [**134, 137**] developed an approach which is largely based on partially ordered sets. Bender and Goldman developed a theory of "prefabs" [**11**] whose purposes are similar to the theory developed here. Joyal [**79**] proposed an especially elegant framework, the "theory of species", that addresses foundational issues in combinatorial theory and constitutes the starting point of the superb exposition by Bergeron, Labelle, and Leroux [**13**]. Parallel (but independent) developments by the "Russian School" are nicely synthetized in the books by Sachkov [**124, 125**].

One of the reasons for the revival of interest in combinatorial enumerations and properties of random structures is the analysis of algorithms, a subject founded in modern times by Knuth [**87**]. The symbolic ideas exposed here have been applied to the analysis of algorithms in surveys [**46, 147**] and are further exposed in our book [**130**]. Flajolet, Salvy, and Zimmermann [**56**] have shown how to use them in order to automate the analysis of some well characterized classes of combinatorial structures.

CHAPTER II

Labelled Structures and Exponential Generating Functions

Cette approche évacue pratiquement tous les calculs.
— DOMINIQUE FOATA &
MARCEL P. SCHÜTZENBERGER [**64**]

Many objects of classical combinatorics present themselves naturally as labelled structures where "atoms" of an object (typically nodes in a graph or a tree) bear distinctive integer labels. For instance the cycle decomposition of a permutation represents the permutation as an unordered collection of circular graphs whose nodes are labelled by integers.

Commonly encountered classes of labelled objects are permutations, set partitions, labelled graphs and labelled trees, graphs and mappings of a finite set into itself, as well as structures related to occupancy problems.

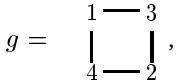
Operations on labelled structures are based on a special product, the labelled product that distributes labels between components. This operation is a natural analogue of the cartesian product for plain unlabelled objects The labelled product in turn leads to labelled analogues of the sequence, set, and cycle constructions.

The labelled constructions translate over exponential generating functions. The translation schemes are analytically simpler than in the unlabelled case considered in the previous chapter. Labelled constructions enable us to take into account structures that are in many ways combinatorially richer, in particular as regards order properties. They therefore constitute a facet with powerful descriptive powers of the symbolic method for combinatorial enumeration.

II. 1. Labelled classes and labelled product

Throughout this chapter, we consider *combinatorial classes* as broadly defined in Chapter I: we deal exclusively with finite objects; a combinatorial class is a set of objects, with a notion of size attached, so that the number of objects of each size is finite. However, the objects are now *labelled* in the sense that each "atom" carries with it an integer label and all the labels occurring in an object are distinct. Precisely, a *weakly labelled* object of size n bears n distinct labels that are integers in $\mathbb{Z}_{\geq 0}$. An object of size n is said to be (strongly or well) *labelled* if it is is weakly labelled and its collection of labels is the consecutive integer interval $[1\,..\,n]$. For a labelled class, the *size* function is systematically defined as the number of labels that the object contains.

As an example, consider the class $\mathcal{G}$ of labelled graphs. An element is by definition an undirected graph such that labels are supported by vertices. A particular labelled graph of size 4 is then

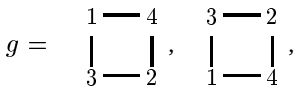
$g =$ 

,

which represents a graph whose vertices bear the labels $\{1, 2, 3, 4\}$ and whose set of edges is

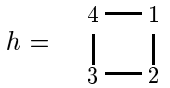
$$\{\,\{1,3\},\ \{2,3\},\ \{2,4\},\ \{1,4\}\,\}.$$

Only the abstract graph structure counts, so that this is the same abstract graph as in the alternative visual representations

$g =$ 

,

since in all three cases, the lists of edges coincide. However, this graph is different from

$h =$ 

,

since, for instance, 1 and 2 have become adjacent. Altogether, it can be seen that there are 3 different ways to build labelled graphs out of the common unlabelled quadrangle graph


.

See Figure 1 for details.

It is also convenient to introduce the neutral (empty, null) object ϵ that has size 0 and bears no label at all, and consider it as a special case of a labelled object; the *neutral class* $\mathcal{E}$ is then by definition $\mathcal{E} = \{\epsilon\}$. The (labelled) *atomic class* $\mathcal{Z} = \{\textcircled{1}\}$ is formed of a unique object of size 1 that bears the integer label $\textcircled{1}$.

The counting of labelled objects is normally achieved by means of exponential generating functions.

DEFINITION II.1. *The* exponential generating function *(EGF) of a sequence* $\{A_n\}$ *is the formal power series*

$$A(z) = \sum_{n=0}^{\infty} A_n \frac{z^n}{n!}. \tag{1}$$

The exponential generating function *(EGF) of a class* $\mathcal{A}_n$ *is the generating function of the numbers* $A_n = \text{card}(\mathcal{A}_n)$. *Equivalently, the EGF of class* $\mathcal{A}$ *is*

$$A(z) = \sum_{n\geq 0} A_n \frac{z^n}{n!} = \sum_{\alpha\in\mathcal{A}} \frac{z^{|\alpha|}}{|\alpha|!}.$$

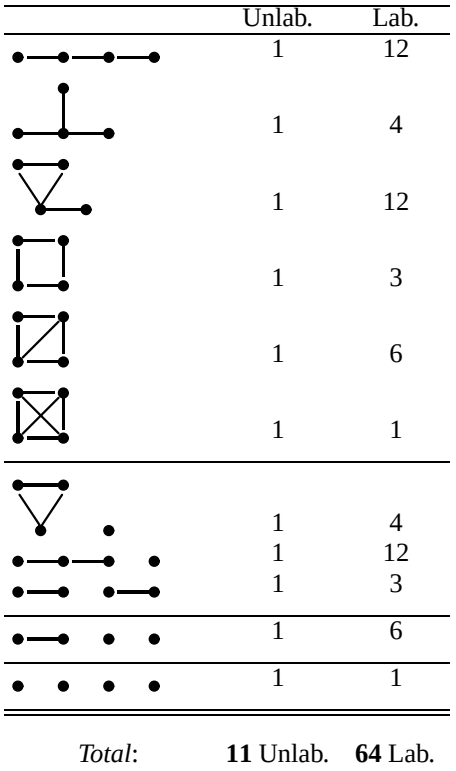
It is also said that the variable z marks *size in the generating function.*

With the standard notation for coefficients of series, the coefficient A_n in an exponential generating function is then recovered by

$$A_n = n! \cdot [z^n]\, A(z),$$

since $[z^n]A(z) = A_n/n!$ by the definition of EGFs and in accordance with the coefficient extractor notation, Eq. (6 of Chapter I.

Note that, like in the previous chapter, we adhere to a systematic naming convention for generating functions of combinatorial structures. A labelled class $\mathcal{A}$, its counting sequence $\{A_n\}$ (or a_n) and its exponential generating function $A(z)$ (or $a(z)$) will all be



Unlab.	Lab.
1	12
1	4
1	12
1	3
1	6
1	1
1	4
1	12
1	3
1	6
1	1

Total: **11** Unlab. **64** Lab.

There are $\widehat{G}_4 = \mathbf{11}$ unlabelled graphs of size 4, i.e., comprising 4 nodes when any number of edges is allowed (left column).
Each unlabelled graph corresponds to a variable number of labelled graphs (indicated in each case by the figure in the right column). For instance, the totally disconnected graph and the complete graph have only 1 labelling. In contrast the line graph has $\frac{1}{2}\,4! = 12$ possible labellings. For size 4, the number of labellings is seen here to vary between 1 and 12.
The total number of labelled graphs found is $G_4 = \mathbf{64} = 2^6$, in agreement with the general formula

$$G_n = 2^{n(n-1)/2}.$$

See p. 70 for details.

FIGURE 1. Unlabelled versus labelled graphs for size $n = 4$.

denoted by the same group of letters. Clearly, the EGF's of the neutral class and the atomic class are respectively

$$E(z) = 1, \qquad Z(z) = 1.$$

EXAMPLE 1. *Permutations.* The class $\{\mathcal{P}\}$ of all permutations is prototypical of labelled classes. Under the linear representation of permutations, it starts as

$$\mathcal{P} = \left\{ \epsilon\ ,\ \textcircled{1}\ ,\ \begin{matrix}\textcircled{1}\!-\!\textcircled{2}\\ \textcircled{2}\!-\!\textcircled{2}\end{matrix}\ ,\ \begin{matrix}\textcircled{1}\!-\!\textcircled{2}\!-\!\textcircled{3}\\ \textcircled{2}\!-\!\textcircled{3}\!-\!\textcircled{1}\\ \textcircled{3}\!-\!\textcircled{1}\!-\!\textcircled{2}\\ \textcircled{2}\!-\!\textcircled{1}\!-\!\textcircled{3}\\ \textcircled{1}\!-\!\textcircled{3}\!-\!\textcircled{2}\\ \textcircled{3}\!-\!\textcircled{1}\!-\!\textcircled{2}\end{matrix}\ ,\ \cdots \right\},$$

so that $P_0 = 1$, $P_1 = 1$, $P_2 = 2$, $P_3 = 6$, etc. There, by definition, all the possible orderings between the distinct atoms are taken into account so that the class $\mathcal{P}$ can be equivalently viewed as the class of all labelled linear digraphs (with an implicit direction, from left to right, say, in the representation). Accordingly, the class $\mathcal{P}$ of permutations has the counting sequence $P_n = n!$ (argument: there are n places at which to place the element 1, then $(n-1)$ possible places for 2, etc). Thus the EGF of $\mathcal{P}$ is

$$P(z) = \sum_{n\geq 0} n!\,\frac{z^n}{n!} = \sum_{n\geq 0} z^n = \frac{1}{1-z}.$$

Permutations, as they contain information relative to the order of their elements are essential in many applications related to order statistics. □

EXAMPLE 2. *Urns.* The class $\mathcal{U}$ of totally disconnected graphs starts as

$$\mathcal{U} = \left\{ \epsilon\ ,\ \textcircled{1}\ ,\ \boxed{\textcircled{1}\quad\textcircled{2}}\ ,\ \boxed{\begin{array}{cc}\textcircled{1} & \textcircled{2}\\ & \textcircled{3}\end{array}}\ ,\ \boxed{\begin{array}{cc}\textcircled{1} & \textcircled{2}\\ \textcircled{3} & \textcircled{4}\end{array}}\ ,\ \boxed{\begin{array}{ccc}\textcircled{1} & & \textcircled{2}\\ & \textcircled{5} & \\ \textcircled{3} & & \textcircled{4}\end{array}}\ ,\ \dots \right\}.$$

Order between the labelled atoms does *not* count, so that for each n, there is only *one* possible arrangement and $U_n = 1$. The class $\mathcal{U}$ can be regarded as the class of "urns", where an urn of size n contains n distinguishable balls in an unspecified (and irrelevant) order. The corresponding EGF is

$$U(z) = \sum_{n\geq 0} 1\, \frac{z^n}{n!} = \exp(z) = e^z.$$

(The fact that the EGF of the constant sequence $\{1\}$ is the exponential function explains the term "exponential generating function".) Alternatively, presenting elements of an urn in sorted order leads to a representation of urns as *sorted* linear graphs; for instance,

$$\textcircled{1}\!-\!\textcircled{2}\!-\!\textcircled{3}\!-\!\textcircled{4}\!-\!\textcircled{5}$$

is such an equivalent representation of the urn of size 5. Though urns may look trivial at first glance, they are of particular importance as building blocks of complex labelled structures (e.g., allocations of various sorts), as we shall see shortly. □

EXAMPLE 3. *Circular graphs.* Finally, the class of circular graphs, where cycles are oriented in some conventional manner (say, positively here) is

$$\mathcal{C} = \left\{ \textcircled{1}\ ,\ \begin{array}{c}\textcircled{1}\\ \textcircled{2}\end{array}\ ,\ \begin{array}{cc}\textcircled{1} & \\ \textcircled{2} & \textcircled{3}\end{array}\ ,\ \begin{array}{cc}\textcircled{1} & \\ \textcircled{3} & \textcircled{2}\end{array}\ ,\ \dots \right\}.$$

Cyclic graphs correspond bijectively to *cyclic permutations* . One has $C_n = (n-1)!$ (argument: a directed cycle is determined by the succession of elements that "follow" 1, hence by a permutation of $n-1$ elements). Thus, one has

$$C(z) = \sum_{n\geq 1} (n-1)!\frac{z^n}{n!} = \sum_{n\geq 1} \frac{z^n}{n} = \log\frac{1}{1-z},$$

where, as we shall see shortly, the logarithm is characteristic of circular arrangements of labelled objects. □

II. 2. Admissible labelled constructions

We now describe a toolkit of *constructions* that make it possible to build complex classes from simpler ones. Combinatorial sum or disjoint union is defined exactly as in Chapter I: it is the union of disjoint copies. Novelty here lies in the definition of a product that is adapted to labelled structures. The usual cartesian product is unsuitable since an ordered pair of two labelled objects is not well labelled—for instance the label 1 would invariably appear repeated twice. The labelled product translates naturally into exponential generating functions, and from there simple translation rules follow for labelled sequences, sets, and cycles.

As a preparation to the translation of labelled constructions, we first briefly review the effect of products over EGF's. If $a(z), b(z), c(z)$ are EGF's, with $a(z) = \sum_n a_n z^n/n!$ and so on, we have the *binomial convolution* formula

$$a(z) = b(z) \cdot c(z) \qquad \Longrightarrow \qquad a_n = \sum_{k=0}^{n} \binom{n}{k} b_k c_{n-k}, \tag{2}$$

since, by the usual product of formal power series,

$$\frac{a_n}{n!} = \sum_{k=0}^{n} \frac{b_k}{k!} \cdot \frac{c_{n-k}}{(n-k)!} \quad \text{and} \quad \binom{n}{k} = \frac{n!}{k!\,(n-k)!}.$$

In the same vein,

$$a(z) = a^{(1)}(z)\, a^{(2)}(z) \cdots a^{(r)}(z) \implies$$

$$a_n = \sum_{n_1+n_2+\cdots+n_r=n} \binom{n}{n_1, n_2, \ldots, n_r} a_n^{(1)} a_n^{(2)} \cdots a_n^{(r)}. \tag{3}$$

In Eq. (3) there occurs the multinomial coefficient

$$\binom{n}{n_1, n_2, \ldots, n_r} = \frac{n!}{n_1! n_2! \cdots n_r!}.$$

This multinomial coefficient also counts the number of ways of splitting n elements into r distinguished classes of cardinalities $n_1, \ldots, n_r$. This fact lies at the very heart of most enumerative applications of binomial convolutions and EGF's.

II. 2.1. Labelled constructions. A labelled object may be relabelled. *We only consider "consistent" relabellings defined by the fact that they preserve the order relations between labels.* Then two dual modes of relabellings prove important:

- *Reduction*: For a non–canonically labelled structure of size n, this operation reduces its labels to the standard interval $[1\,..\,n]$ while preserving the relative order of labels. For instance, the sequence $\langle 7, 3, 9, 2\rangle$ reduces to $\langle 3, 2, 4, 1\rangle$. We note $\rho(\alpha)$ the reduction of the structure α.
- *Expansion*: This operation is defined relative to a relabelling function $e \in [1\,..\,n] \mapsto \mathbb{Z}_{\geq 1}$ that is assumed to be strictly increasing. For instance, $\langle 3, 2, 4, 1\rangle$ may expand as $\langle 33, 22, 44, 11\rangle$, $\langle 7, 3, 9, 2\rangle$, and so on. We note $e(\alpha)$ the result of relabelling α by e.

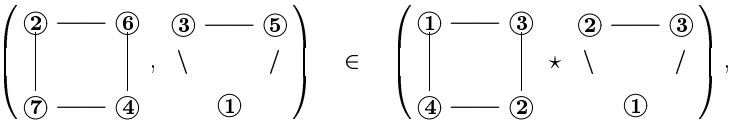
We next define a product called the *labelled product*, or simply *product* (originally this was named *partitional product* by Foata who proposed an early formalization in [**62**]). Given two labelled structures $\beta \in \mathcal{B}$ and $\gamma \in \mathcal{C}$, the product $\beta \star \gamma$ comprises the finite collection of objects that are ordered pairs (β', γ') of relabelled copies of (β, γ),

$$\beta \star \gamma := \{\, (\beta', \gamma') \;\big|\; (\beta', \gamma') \text{ is well–labelled},\ \rho(\beta') = \beta,\ \rho(\gamma') = \gamma \,\}, \tag{4}$$

the relabellings preserving the order structure present in β and γ. An equivalent form is via expansion of labels:

$$\beta \star \gamma = \{\, (e(\beta), f(\gamma) \;\big|\; \mathrm{Im}(e) \cap \mathrm{Im}(f) = \emptyset,\ \mathrm{Im}(e) \cup \mathrm{Im}(f) = [1\,..\,|\beta| + |\gamma|] \,\}, \tag{5}$$

where e, f are again assumed to be *increasing* with ranges $\mathrm{Im}(e), \mathrm{Im}(f)$. For instance, one has



as seen by reduction of the left pair or, dually, by expansion of the right pair.

If $\mathcal{B}$ and $\mathcal{C}$ are two classes of combinatorial structures, the labelled product $\mathcal{A} = \mathcal{B} \star \mathcal{C}$ is defined by the usual extension of operations to sets:

$$\mathcal{B} \star \mathcal{C} = \bigcup_{\beta \in \mathcal{B},\ \gamma \in \mathcal{C}} (\beta \star \gamma). \tag{6}$$

In summary:

DEFINITION II.2. *The* labelled product *of $\mathcal{B}$ and $\mathcal{C}$, denoted $\mathcal{B} \star \mathcal{C}$, is obtained by forming ordered pairs from $\mathcal{B} \times \mathcal{C}$ and performing all possible order consistent relabellings, ensuring that the resulting pairs are well–labelled, as described by (4) or (5), and (6).*

The corresponding counting sequences satisfy the relation,

$$A_n = \sum_{n_1 + n_2 = n} \binom{n}{n_1, n_2} B_{n_1} C_{n_2}.$$

There the binomial arises since the the number of relabellings involved in forming all the elements of $(\beta \star \gamma)$ is $\binom{n}{n_1, n_2}$, if $|\beta| - n_1$, $|\gamma| = n_2$ and $n_1 + n_2 = n$. The product $B_{n_1} C_{n_2}$ keeps track of all the possibilities for the $\mathcal{B}$ and $\mathcal{C}$ components. By (2), the binomial convolution corresponds to the product relation,

$$A(z) = B(z) \cdot C(z),$$

relating EGFs. Thus, the labelled product simply translates into the product operation on exponential generating functions.

The kth (labelled) *power* of $\mathcal{B}$ is defined as $(\mathcal{B} \star \mathcal{B} \cdots \mathcal{B})$, with k factors equal to $\mathcal{B}$. It is denoted $\mathfrak{S}_k\{\mathcal{B}\}$. This corresponds to forming k–sequences and performing all consistent relabellings. The (labelled) *sequence* class of $\mathcal{B}$ is denoted by $\mathfrak{S}\{\mathcal{B}\}$ and is defined by

$$\mathfrak{S}\{\mathcal{B}\} \stackrel{\text{def}}{=} \{\epsilon\} + \mathcal{B} + (\mathcal{B} * \mathcal{B}) + (\mathcal{B} * \mathcal{B} * \mathcal{B}) + \cdots = \bigcup_{k \geq 0} \mathfrak{S}_k\{\mathcal{B}\}.$$

The product relation for EGF's clearly extends to arbitrary products, as seen from the multinomial convolution formula (3), so that

$$\mathcal{A} = \mathfrak{S}_k\{\mathcal{B}\} \implies B(z) = B(z)^k,$$

and (assuming $B_0 \neq 0$)

$$\mathcal{A} = \mathfrak{S}\{\mathcal{B}\} \implies A(z) = \sum_{k=0}^{\infty} B(z)^k = \frac{1}{1 - B(z)}.$$

We denote by $\mathfrak{P}_k\{\mathcal{B}\}$ the class of k–sets formed from $\mathcal{B}$. The powerset class is defined formally, like in the unlabelled case, as the quotient $\mathfrak{P}\{\mathcal{B}\} := \mathfrak{S}_k\{\mathcal{B}\}/\mathbf{R}$ where the equivalence relation $\mathbf{R}$ indentifies two sequences when the components of one are a permutation of the components of the other (p. 9). In simple terms, a "set" is like a

1. The main constructions of union, and product, sequence, set, and cycle for labelled structures together with their translation into exponential generating functions.

Construction		EGF
Union	$\mathcal{A} = \mathcal{B} + \mathcal{C}$	$A(z) = B(z) + C(z)$
Product	$\mathcal{A} = \mathcal{B} \star \mathcal{C}$	$A(z) = B(z) \cdot C(z)$
Sequence	$\mathcal{A} = \mathfrak{S}\{\mathcal{B}\}$	$A(z) = \dfrac{1}{1 - B(z)}$
Set	$\mathcal{A} = \mathfrak{P}\{\mathcal{B}\}$	$A(z) = \exp(B(z))$
Cycle	$\mathcal{A} = \mathfrak{C}\{\mathcal{B}\}$	$A(z) = \log \dfrac{1}{1 - B(z)}$

2. The translation for sets, multisets, and cycles of fixed cardinality.

Construction		EGF
Sequence	$\mathcal{A} = \mathfrak{S}_k\{\mathcal{B}\}$	$A(z) = A(z)^k$
Set	$\mathcal{A} = \mathfrak{P}_k\{\mathcal{B}\}$	$A(z) = \dfrac{1}{k!}A(z)^k$
Cycle	$\mathcal{A} = \mathfrak{C}_k\{\mathcal{B}\}$	$A(z) = \dfrac{1}{k}A(z)^k$

3. The additional constructions of pointing and substitution.

Construction		EGF
Pointing	$\mathcal{A} = \Theta\mathcal{B}$	$A(z) = z\frac{d}{dz}B(z)$
Substitution	$\mathcal{A} = \mathcal{B} \circ \mathcal{C}$	$A(z) = B(C(z))$

4. The "boxed" product.

$$\mathcal{A} = (\mathcal{B}^{\square} \star \mathcal{C}) \implies A(z) = \int_0^z \left(\frac{d}{dt}B(t)\right) \cdot C(t)\, dt.$$

FIGURE 2. A "dictionary" of *labelled* constructions together with their translation into *exponential* generating functions (EGF's). The first constructions are counterparts of the unlabelled constructions of the previous chapter (the multiset construction is not meaningful here). The translation for composite constructions of bounded cardinality appears to be simple. Finally, the boxed product is specific to labelled structures. (Compare with the unlabelled counterpart, Figure 2 of Chapter I, p. 2.)

sequence, but the order between components is immaterial. The (labelled) *powerset* class of $\mathcal{B}$, denoted $\mathfrak{P}\{\mathcal{B}\}$, is defined by

$$\mathfrak{P}\{\mathcal{B}\} \stackrel{\text{def}}{=} \{\epsilon\} + \mathcal{B} + \mathfrak{P}_2\{\mathcal{B}\} + \cdots = \bigcup_{k\geq 0} \mathfrak{P}_k\{\mathcal{B}\}.$$

A labelled k–set is associated with exactly $k!$ different sequences. (There is here is subtle difference with the the unlabelled case where formulæ are more complex as an unlabelled sequence may contain repeated elements while components of a labelled sequence are all distinguished by their labels.) Thus in terms of EGF's, one has (assuming $\mathcal{B}_0 = \emptyset$)

$$\mathcal{A} = \mathfrak{P}_k\{\mathcal{B}\} \implies A(z) = \frac{B(z)^k}{k!},$$

$$\mathcal{A} = \mathfrak{P}\{\mathcal{B}\} \implies A(z) = \sum_{k=0}^{\infty} \frac{B(z)^k}{k!} = \exp(B(z)).$$

Note that the distinction between multisets and powersets is here immaterial, since by definition components of a set all have distinct labels: in the labelled universe, we have $\mathfrak{M} \equiv \mathfrak{P}$.

We also introduce the class of k–cycles, $\mathfrak{C}_k\{\mathcal{B}\}$ and the cycle class. The cycle class is defined formally, like in the unlabelled case, as the quotient $\mathfrak{C}\{\mathcal{B}\} := \mathfrak{S}_k\{\mathcal{B}\}/\mathbf{S}$ where the equivalence relation $\mathbf{S}$ indentifies two sequences when the components of one are a cyclic permutation of the components of the other (p. 9). In simple terms, a "cycle" is like a sequence, but components can be circularly shifted. In terms of EGF's, we have (assuming $\mathcal{B}_0 = \emptyset$)

$$\mathcal{A} = \mathfrak{C}_k\{\mathcal{B}\} \implies A(z) = \frac{B(z)^k}{k},$$

$$\mathcal{A} = \mathfrak{C}\{\mathcal{B}\} \implies A(z) = \sum_{k=0}^{\infty} \frac{B(z)^k}{k} = \log \frac{1}{1 - B(z)},$$

since each cycle admits exactly k representations as a sequence. In summary:

THEOREM II.1. *The constructions of labelled product, k–th power, and sequence class,*

$$\mathcal{A} = \mathcal{B} \star \mathcal{C}, \qquad \mathcal{A} = \mathfrak{S}_k\{\mathcal{B}\}, \qquad \mathcal{A} = \mathfrak{S}\{\mathcal{B}\}$$

are admissible:

$$A(z) = B(z) \cdot C(z), \qquad A(z) = B(z)^k, \qquad A(z) = \frac{1}{1 - B(z)}.$$

The constructions of k–set and powerset class

$$\mathcal{A} = \mathfrak{P}_k\{\mathcal{B}\}, \qquad \mathcal{A} = \mathfrak{P}\{\mathcal{B}\},$$

are admissible:

$$A(z) = \frac{1}{k!} B(z)^k, \qquad A(z) = \exp(B(z)).$$

The constructions of k–cycle and cycle class,

$$\mathcal{A} = \mathfrak{C}_k\{\mathcal{B}\}, \qquad \mathcal{A} = \mathfrak{C}\{\mathcal{B}\},$$

are admissible:

$$A(z) = \frac{1}{k} B(z)^k, \qquad A(z) = \log \frac{1}{1 - B(z)}.$$

Constructible classes. Like in the previous chapter, we say that a class of labelled objects is constructible if it admits a specification in terms of sums (disjoint unions), the labelled constructions of product, sequence, set, cycle, and the initial classes defined by the neutral structure of size 0 and the atomic node $\mathcal{N} = \{1\}$ of size 1. Amongst the elementary classes discussed in Section II. 1, one immediately recognizes that

$$\mathcal{P} = \mathfrak{S}\{\mathcal{Z}\}, \quad \mathcal{U} = \mathfrak{P}\{\mathcal{Z}\}, \quad \mathcal{C} = \mathfrak{C}\{\mathcal{Z}\},$$

specify permutations, urns, and circular graphs respectively. These are basic building blocks out of which more complex objects can be constructed. Set partitions ($\mathcal{S}$), surjections ($\mathcal{R}$), permutations ($\mathcal{P}$) under their cycle decomposition, and alignments ($\mathcal{O}$) are then particular constructible classes corresponding to

$$\mathcal{S} \simeq \mathfrak{P}\{\mathfrak{P}_{\geq 1}\{\mathcal{Z}\}\}, \quad \mathcal{R} \simeq \mathfrak{S}\{\mathfrak{P}_{\geq 1}\{\mathcal{Z}\}\}, \quad \mathcal{P} \simeq \mathfrak{P}\{\mathfrak{C}_{\geq 1}\{\mathcal{Z}\}\}, \quad \mathcal{O} \simeq \mathfrak{S}\{\mathfrak{C}_{\geq 1}\{\mathcal{Z}\}\}.$$

An immediate consequence of Theorem II.1 is the fact that the EGF of a constructible labelled class can be computed automatically.

THEOREM II.2. *The exponential generating function of a constructible class of labelled objects is a component of a system of generating function equations whose terms are built from* 1 *and* z *using the operators*

$$+\,,\ \times\,,\ Q(f) = \frac{1}{1-f}\,,\ E(f) = e^f\,,\ L(f) = \log\frac{1}{1-f}.$$

If we further allow cardinality restrictions in composite constructions, the operators f^k (for $\mathfrak{S}_k$), $f^k/k!$ (for $\mathfrak{P}_k$), and f^k/k (for $\mathfrak{C}_k$) are to be added to the list.

II. 2.2. Labelled versus unlabelled? Let $\mathcal{A}$ be a labelled class. If this class is constructible, it automatically has an unlabelled counterpart $\widehat{\mathcal{A}}$ that is obtained by interpreting all the intervening constructions as unlabelled ones, in the sense of Chapter I. Equivalently, one may view objects in $\widehat{\mathcal{A}}$ as obtained from objects of $\mathcal{A}$ by "forgetting the labels". This is formalized by identifying two labelled object if there is an *arbitrary* relabelling (not just order-consistent ones, as have been used so far) that transforms one into the other. For an object of size n, each equivalence class contains a priori between 1 and $n!$ elements. We state:

PROPOSITION II.1. *The counts of a labelled class* $\mathcal{A}$ *and its unlabelled counterpart* $\widehat{\mathcal{A}}$ *are related by*

$$\widehat{A}_n \leq A_n \leq n!\,\widehat{A}_n \quad \textit{or equivalently} \quad 1 \leq \frac{A_n}{\widehat{A}_n} \leq n!. \tag{7}$$

EXAMPLE 4. *Labelled and Unlabelled graphs.* This phenomenon has been already encountered in our discussion of graphs, where 4 labellings can be attached to the unlabelled quadrangle graph of size 4. If one considers instead the totally disconnected graph of size 4, then there exists exactly one labelled version (the "urn" of size 4) and one unlabelled version. Let generally G_n and $\widehat{G}_n$ be the number of graphs of size n in the labelled and unlabelled case respectively. One finds for $n = 1\,..\,18$

$\widehat{G}_n$ (unlabelled)	G_n (labelled)
1	1
2	2
4	8
11	64
34	1024
156	32768
1044	2097152
12346	268435456
274668	68719476736
12005168	35184372088832
1018997864	36028797018963968
165091172592	73786976294838206464
50502031367952	30223145490365729367654 4
29054155657235488	24758800785707605497982484 48
31426485969804308768	40564819207303340847894502572032
6400101570452755789492 8	13292279957849158729038070602803445 76
24593586415353293268371 9776	87112285931760246646623899502532662132736
17875777251456117005478 78190848	11417981541647679048466287755595961091061972992

The sequence $\{\widehat{G}_n\}$ constitutes *EIS* **A000088**, which can be obtained by an extension of methods of Chapter I; see [**76**, Ch. 4]. The sequence $\{G_n\}$ is determined directly by the fact that a graph of n vertices can have each of the $\binom{n}{2}$ possible edges either present or not, so that

$$G_n = 2^{\binom{n}{2}} = 2^{n(n-1)/2}.$$

The sequence of labelled counts obviously grows much faster than its unlabelled counterpart. We may then verify the inequality (7) in this particular case. The normalized ratios,

$$\rho_n := G_n/\widehat{G}_n, \quad \sigma_n := G_n/(n!\widehat{G}_n),$$

are observed to be

n	$\rho_n = G_n/\widehat{G}_n$	$\sigma_n = G_n/(n!\widehat{G}_n)$
1	1.000000000	1.0000000000
2	1.000000000	0.5000000000
3	2.000000000	0.3333333333
4	5.818181818	0.2424242424
5	30.11764706	0.2509803922
6	210.0512821	0.2917378918
8	21742.70663	0.5392536367
10	2930768.823	0.8076413203
12	446946830.2	0.9330800361
14	$0.8521603960 \cdot 10^{11}$	0.9774915111
16	$0.2076885783 \cdot 10^{14}$	0.9926428522
18	$0.6387404239 \cdot 10^{16}$	0.9976618880

From these data, it is natural to conjecture that σ_n tends (fast) to 1 as n tends to infinity. This is indeed a nontrivial fact originally established by Pólya (see Chapter 9 of [**76**] dedicated to asymptotics of graph enumerations):

$$\widehat{G}_n \sim \frac{1}{n!}\, 2^{\binom{n}{2}} \sim \frac{G_n}{n!}.$$

In other words, "almost all" graphs of size n should admit a number of labellings close to $n!$. (Combinatorially, this corresponds to the fact that in a random unlabelled graph, with high probability, all of the nodes can be distinguished based on the adjacency structure of the graph; in such a case, the graph has no nontrivial automorphism and the number of distinct labellings is $n!$ exactly.) □

The case of urns and totally disconnect graphs resorts to the other extreme situation where

$$\widehat{U}_n = U_n = 1.$$

The examples of graphs and urns illustrate the fact that, beyond the general bounds of Proposition II.1, there is no automatic way to translate between labelled and unlabelled enumerations, apart from computing separately the two GF's and comparing coefficients.

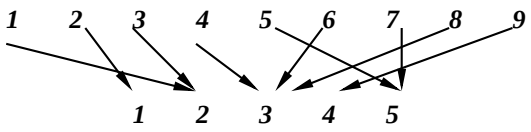
II. 3. Surjections, set partitions, and words

This section and the next are devoted to what could be termed nonrecursive structures of "level 2" defined by the fact that they combine two constructions. Here, we examine classes

$$\mathcal{R} = \mathfrak{S}\{\mathfrak{P}_{\geq 1}\{Z\}\} \qquad \text{and} \qquad \mathcal{S} = \mathfrak{P}\{\mathfrak{P}_{\geq 1}\{Z\}\},$$

corresponding to sequences-of-sets ($\mathcal{R}$) and sets-of-sets ($\mathcal{S}$) respectively. We shall see shortly (Section II. 3.1) that such abstract specifications model classical objects of discrete mathematics, namely surjections ($\mathcal{R}$) and set partitions ($\mathcal{S}$). (These constitute in a way labelled analogues of integer compositions and integer partitions in the unlabelled universe.) The symbolic methodology then extends naturally to words over a finite alphabet, where it opens access to an analysis of the frequencies of letters composing words. This in turn has useful consequences for the study of some classical random allocation problems, of which the birthday paradox and the coupon collector problem stand out (Section II. 3.2).

II. 3.1. Surjections and set partitions. In elementary mathematics, a surjection from a set A to a set B is a function from A to B that assumes each value *at least once* (an unto mapping). Fix some integer $r \geq 1$ and let $\mathcal{R}_n^{(r)}$ denote the class of all surjections from the set $[1\,..\,n]$ *onto* $[1\,..\,r]$ whose elements are also called r–surjections.. Here is a particular object of $\mathcal{R}_9^{(5)}$:



We set $\mathcal{R}^{(r)} = \bigcup_n \mathcal{R}_n^{(r)}$ and proceed to determine the corresponding EGF, $R^{(r)}(z)$. First, let us observe that an r–surjection $\phi \in \mathcal{R}_n^{(r)}$ is determined by the *ordered r–tuple* formed with the preimages, $\big(\phi^{-1}(1), \phi^{-1}(2), \ldots, \phi^{-1}(r)\big)$, themselves disjoint nonempty sets of integers that cover the interval $[1\,..\,n]$. In other words, one has the combinatorial specification

$$\mathcal{R}^{(r)} = \mathfrak{S}_r\{\mathcal{V}\}, \quad \mathcal{V} = \mathcal{U} \setminus \{\epsilon\} = \mathfrak{P}_{\geq 1}\{\mathcal{Z}\},$$

where $\mathcal{V}$ designates the class of urns ($\mathcal{U}$) that are nonempty. Consequently, the EGF satisfies

$$R^{(r)}(z) = (e^z - 1)^r, \tag{8}$$

in view of our earlier discussion of urns ($\mathcal{U}$) with EGF $U(z) = e^z$.

Equation (8) does solve the counting problem for surjections. For small r, one finds

$$R^{(2)}(z) = e^{2z} - 2e^z + 1, \qquad R^{(3)}(z) = e^{3z} - 3e^{2z} + 3e^z - 1,$$

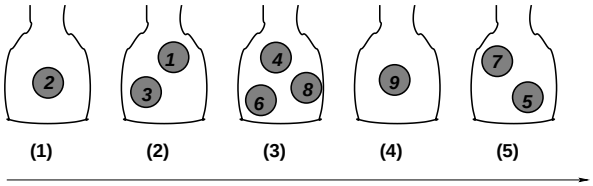
whence, by expanding,

$$R_n^{(2)} = 2^n - 2, \qquad R_n^{(3)} = 3^n - 3 \cdot 2^n + 3\,.$$

A surjection, here the mapping from $[1\,..\,9]$ onto $[1\,..\,5]$ given by the table

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 2 & 1 & 2 & 3 & 5 & 3 & 5 & 3 & 4 \end{pmatrix},$$

may be viewed as an ordered tuple of nonempty urns, or equivalently, linear sorted graphs



$$\begin{aligned} \sigma &= [\,\{2\},\quad \{3,1\},\quad \{6,4,8\},\quad \{9\},\quad \{5,7\}\,] \\ &= \Big[\,②,\quad ①—③,\quad ④—⑥—⑧,\quad ⑨,\quad ⑤—⑦\,\Big] \end{aligned}$$

corresponding to the collection of preimages of $1, 2, 3, 4, 5$.

FIGURE 3. The decomposition of surjections as sequences-of-sets.

The general formula follows similarly from expanding the rth power in (8) by the binomial theorem, and then extracting coefficients:

$$\begin{aligned} R_n^{(r)} &= n!\,[z^n] \sum_{j=0}^{r} \binom{r}{j} (-1)^j e^{r-j} z \\ &= \sum_{j=0}^{r} \binom{r}{j} (-1)^j (r-j)^n. \end{aligned} \tag{9}$$

▷ **1.** *A direct derivation of the surjection EGF.* One may verify the result provided by the symbolic method by returning to first principles. Since each preimage of a surjection is a nonempty set, the number of r-surjections is expressed by an r-fold convolution,

$$R_n^{(r)} = \sum_{(n_1, n_2, \cdots, n_r)} \binom{n}{n_1, n_2, \ldots, n_r}, \tag{10}$$

the sum being taken over $n_j \geq 1,\ n_1 + n_2 + \cdots + n_r = n$. (In this formula the indices n_j vary over all allowable cardinalities of preimages, and the multinomial coefficient counts the number of ways of distributing the elements of $[1\,..\,n]$ amongst the r preimages.) Introduce the numbers V_n by $V_0 = 0$ and $V_n = 1$ if $n \geq 1$. The formula (10) then assumes the simpler form

$$R_n^{(r)} \equiv \sum_{n_1, n_2, \ldots, n_r} \binom{n}{n_1, n_2, \ldots, n_r} V_{n_1} V_{n_2} \cdots V_{n_r}, \tag{11}$$

where the summation now extends to *all* tuples $(n_1, n_2, \ldots, n_r)$. The EGF of the V_n is $V(z) = \sum V_n z^n/n! = e^z - 1$. Thus the convolution relation (11) leads to (8). ◁

Let $\mathcal{S}_n^{(r)}$ denote the number of ways of partitioning the set $[1\,..\,n]$ into r disjoint and nonempty equivalence classes. We set $\mathcal{S}^{(r)} = \bigcup_n \mathcal{S}_n^{(r)}$; the corresponding objects are called *set partitions* (the latter not to be confused with integer partitions examined in Section I. 3). The enumeration problem for set partitions is closely related to that of surjections. Symbolically, a partition is determined as a labelled *set* of classes, each of which is a non-empty urn. Thus, one has

$$\mathcal{S}^{(r)} = \mathfrak{P}_r\{\mathcal{V}\}, \qquad \mathcal{V} = \mathfrak{P}_{\geq 1}\{\mathcal{Z}\}.$$

The basic formula connecting the two counting sequences results from there (or from direct reasoning):

$$S_n^{(r)} = \frac{1}{r!} R_n^{(r)} \qquad \text{and} \qquad S^{(r)}(z) = \frac{1}{r!}\,(e^z - 1)^r\,. \tag{12}$$

The rationale for (12) is that an r–partition is associated with a group of exactly $r!$ distinct r–surjections, two surjections belonging to the same group iff one obtains from the other by permuting the range values, $[1\,..\,r]$.

The numbers $S_n^{(r)} = n![z^n]S^{(r)}(z)$ are known as the Stirling numbers of the second kind, or better, the Stirling "partition" numbers. They were briefly encountered in the previous chapter and discussed in connection with encodings by words (Example 7 and Figure 9 of Chapter I). Knuth, following Karamata, advocated for the $S_n^{(r)}$ the notation $\left\{ {n \atop r} \right\}$. From (9), an explicit form also exists:

$$S_n^{(r)} \equiv \left\{ {n \atop r} \right\} = \frac{1}{r!} \sum_{j=0}^{r} \binom{r}{j} (-1)^j (r-j)^n. \tag{13}$$

The books by Graham, Knuth, and Patashnik [**71**] and Comtet [**28**] contain a thorough discussion of these numbers; see also APPENDIX: *Stirling numbers*, p. 173.

Define now the collection of all surjections and all set partitions by

$$\mathcal{R} = \bigcup_r \mathcal{R}^{(r)}, \qquad \mathcal{S} = \bigcup_r \mathcal{S}^{(r)}.$$

Thus $\mathcal{R}_n$ is the class of all surjections of $[1\,..\,n]$ onto *any* initial segment of the integers, and $\mathcal{S}_n$ is the class of all partitions of the set $[1\,..\,n]$ into *any* number of blocks (Figure 4). Symbolically, one has

$$\mathcal{R} = \mathfrak{S}\{\mathcal{V}\}, \qquad \mathcal{S} = \mathfrak{P}\{\mathcal{V}\}, \qquad \text{with} \quad \mathcal{V} = \mathfrak{P}_{\geq 1}\{\mathcal{Z}\}.$$

From there one finds

$$R(z) = \frac{1}{2 - e^z}, \qquad S(z) = e^{e^z - 1}, \tag{14}$$

since $V(z) = e^z - 1$ and $R(z) = (1 - V(z))^{-1}$, $S(z) = e^{V(z)}$. The numbers $R_n = n!\ [z^n]R(z)$ and $S_n = n![z^n]S(z)$ are called the *surjection numbers* (also, "preferential arrangements" numbers, *EIS* **A000670**) and the *Bell numbers* (*EIS* **A000110**) respectively. These numbers are well determined by expanding the EGFs:

$$\begin{aligned} R(z) &= 1 + z + 3\,\frac{z^2}{2!} + 13\,\frac{z^3}{3!} + 75\,\frac{z^4}{4!} + 541\,\frac{z^5}{5!} + 4683\,\frac{z^6}{6!} + 47293\,\frac{z^7}{7!} + \cdots \\ S(z) &= 1 + z + 2\,\frac{z^2}{2!} + 5\,\frac{z^3}{3!} + 15\,\frac{z^4}{4!} + 52\,\frac{z^5}{5!} + 203\,\frac{z^6}{6!} + 877\,\frac{z^7}{7!} + \cdots . \end{aligned}$$

Explicit expressions as finite double sums result from summing Stirling numbers,

$$R_n = \sum_{k \geq 0} k! \left\{ {n \atop k} \right\}, \qquad R_n = \sum_{k \geq 0} k! \left\{ {n \atop k} \right\},$$

where each Stirling number is itself a sum given by (13).

Alternatively, single (though infinite) sums result from the expansions

$$\left\{ \begin{aligned} R(z) &= \frac{1}{2}\,\frac{1}{1 - \frac{1}{2}e^z} \\ &= \sum_{k=0}^{\infty} \frac{1}{2^{k+1}} e^{kz} \end{aligned} \right. \qquad \text{and} \qquad \left\{ \begin{aligned} S(z) &= e^{e^z - 1} = \frac{1}{e} e^{e^z} \\ &= \frac{1}{e} \sum_{k=0}^{\infty} e^{kz}, \end{aligned} \right.$$

$n = 1, S_1 = 1$:

$\{1\}$

$n = 2, S_2 = 2$:

$\{2\}, \{1\}$, $\{1, 2\}$

$n = 3, S_3 = 5$:

$\{1\}, \{2, 3\}$, $\{1, 2\}, \{3\}$, $\{1, 2, 3\}$, $\{2\}, \{1\}, \{3\}$, $\{1, 3\}, \{2\}$

$n = 4, S_4 = 15$:

$\{1, 2, 4\}, \{3\}$, $\{1, 3, 4\}, \{2\}$, $\{1\}, \{2, 4\}, \{3\}$, $\{1, 3\}, \{2\}, \{4\}$, $\{1, 2, 3\}, \{4\}$, $\{4\}, \{1, 2\}, \{3\}$, $\{3, 4\}, \{1, 2\}$,
$\{1, 2, 3, 4\}$, $\{1\}, \{2, 3, 4\}$, $\{2, 3\}, \{1, 4\}$, $\{2\}, \{1, 4\}, \{3\}$, $\{3, 4\}, \{2\}, \{1\}$, $\{2\}, \{1\}, \{4\}, \{3\}$, $\{1, 3\}, \{2, 4\}$,
$\{1\}, \{4\}, \{2, 3\}$

$n = 5, S_5 = 52$:

$\{2\}, \{1\}, \{4\}, \{3, 5\}$, $\{1, 2, 3\}, \{5\}, \{4\}$, $\{1, 2, 4, 5\}, \{3\}$, $\{1\}, \{2, 4\}, \{3, 5\}$, $\{5\}, \{1\}, \{2, 4\}, \{3\}$,
$\{1, 5\}, \{2, 4\}, \{3\}$, $\{1\}, \{2, 3\}, \{4, 5\}$, $\{1, 3, 5\}, \{2, 4\}$, $\{1\}, \{4\}, \{2, 5\}, \{3\}$, $\{2, 4, 5\}, \{1, 3\}$, $\{5\}, \{1, 2, 3, 4\}$,
$\{1, 5\}, \{2, 3, 4\}$, $\{2\}, \{1\}, \{3\}, \{4, 5\}$, $\{5\}, \{2\}, \{1, 4\}, \{3\}$, $\{2, 3, 4, 5\}, \{1\}$, $\{1, 3, 5\}, \{2\}, \{4\}$, $\{2, 4, 5\}, \{1\}, \{3\}$,
$\{1, 2, 3, 5\}, \{4\}$, $\{2, 3, 5\}, \{1, 4\}$, $\{1, 3\}, \{5\}, \{2\}, \{4\}$, $\{4\}, \{1, 2\}, \{3, 5\}$, $\{1, 2, 3\}, \{4, 5\}$, $\{1, 2\}, \{3\}, \{4, 5\}$,
$\{1, 5\}, \{2\}, \{4\}, \{3\}$, $\{1, 3, 4\}, \{2, 5\}$, $\{3, 4\}, \{5\}, \{2\}, \{1\}$, $\{1, 3\}, \{5\}, \{2, 4\}$, $\{2\}, \{1, 3, 4, 5\}$, $\{1, 3\}, \{2\}, \{4, 5\}$,
$\{5\}, \{1\}, \{4\}, \{2, 3\}$, $\{5\}, \{4\}, \{1, 2\}, \{3\}$, $\{3, 4, 5\}, \{1, 2\}$, $\{5\}, \{2, 3\}, \{1, 4\}$, $\{1, 2, 3, 4, 5\}$, $\{1, 3, 4\}, \{5\}, \{2\}$,
$\{2\}, \{1, 4, 5\}, \{3\}$, $\{3, 4\}, \{1\}, \{2, 5\}$, $\{5\}, \{1\}, \{2, 3, 4\}$, $\{4\}, \{1, 2, 5\}, \{3\}$, $\{1, 2, 4\}, \{3, 5\}$, $\{3, 4\}, \{1, 2, 5\}$,
$\{1, 4\}, \{2, 5\}, \{3\}$, $\{3, 4\}, \{5\}, \{1, 2\}$, $\{5\}, \{1, 2, 4\}, \{3\}$, $\{1, 3\}, \{4\}, \{2, 5\}$, $\{2, 3\}, \{1, 4, 5\}$, $\{2\}, \{1\}, \{3, 4, 5\}$,
$\{2, 3, 5\}, \{1\}, \{4\}$, $\{5\}, \{2\}, \{1\}, \{4\}, \{3\}$, $\{2\}, \{1, 4\}, \{3, 5\}$, $\{1, 5\}, \{3, 4\}, \{2\}$, $\{1, 5\}, \{4\}, \{2, 3\}$

FIGURE 4. A listing of all set partitions for sizes $n = 1, 2, 3, 4, 5$.

from which coefficient extraction yields

$$R_n = \frac{1}{2}\sum_{k=0}^{\infty}\frac{k^n}{2^k} \qquad \text{and} \qquad S_n = \frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!}.$$

The formula for the Bell numbers was found by Dobinski in 1877.

The asymptotic analysis of the surjection numbers (R_n) will be performed in a later chapter by means of singularity analysis of the meromorphic function $R(z)$; that of Bell's partition numbers (S_n) is best done by means of the saddle point method. The asymptotic forms found are

$$R_n \sim \frac{n!}{2}\,\frac{1}{(\log 2)^{n+1}} \qquad \text{and} \qquad S_n \sim n!\,\frac{e^{e^{r(n)}-1}}{r(n)^{n+1}\sqrt{2\pi\exp(r(n))}}, \tag{15}$$

where $r(n)$ is the positive root of the equation $re^r = n$. One has $r(n) \approx \log n - \log\log n$, so that

$$\log S_n = n\left(\log n - \log\log n - 1 + o(1)\right).$$

Elementary derivations of these asymptotic forms are explored in the notes that follow.

▷ **2.** *Laplace's method for sums.* By examining ratios between successive terms in the sum expressing S_n, one determines the index k_0 near which the terms in Dobinski's formula are maximal. The general term of index $k = k_0 \pm h$, after scaling, is then found to be well approximated by the Gaussian function e^{-x^2}. A comparison with the Riemann sum of the Gaussian functions leads to the asymptotic form stated for S_n. This is an instance of the Laplace method for sums that is detailed in

De Bruijn's book [**35**]; see also [**130**]. The asymptotic estimation of R_n can be subjected to a similar treatment (Comtet). $\triangleleft$

$\triangleright$ **3.** *Cauchy's method for generating functions.* An approach different from the one in Ex. 2 bases itself on the fact that $R(z)$ has a singularity at a finite distance. Indeed, the function

$$R(z) - \frac{1}{2}\frac{1}{\log 2 - z}$$

is analytic for $|z| \leq 6$. (The singularity of $R(z)$ at $\log 2$ has been removed and the next poles are at $\log 2 \pm 2i\pi$.) Thus, one has

$$\frac{R_n}{n!} = \frac{1}{2}\left(\frac{1}{(\log 2)^{n+1}} + O(\frac{1}{6^n})\right)$$

by virtue of Cauchy's bounds for coefficients of analytic functions; see Chapter IV for details. $\triangleleft$

The line of reasoning adopted for the enumeration of surjections viewed as sequences-of-sets and partitions viewed as sets-of-sets yields a general result that is applicable to a wide variety of constrained objects.

PROPOSITION II.2. *Let $\mathcal{R}^{(A,B)}$ be the class of surjections where the cardinalities of the preimages lie in $A \subseteq \mathbb{Z}_{\geq 1}$ and the cardinality of the range belongs to B. The corresponding EGF is*

$$R^{(A,B)}(z) = \beta(\alpha(z)) \qquad \textit{where} \qquad \alpha(z) = \sum_{a\in A}\frac{z^a}{a!}, \quad \beta(z) = \sum_{b\in B} z^b.$$

Let $\mathcal{S}^{(A,B)}$ be the class of set partitions with part sizes in $A \subseteq \mathbb{Z}_{\geq 1}$ and with a number of blocks that belongs to B. The corresponding EGF is

$$S^{(A,B)}(z) = \beta(\alpha(z)) \qquad \textit{where} \qquad \alpha(z) = \sum_{a\in A}\frac{z^a}{a!}, \quad \beta(z) = \sum_{b\in B} \frac{z^b}{b!}.$$

PROOF. One has

$$\mathcal{R}^{(A,B)} = \mathfrak{S}_A\{\mathfrak{P}_B\{Z\}\} \qquad \text{and} \qquad \mathcal{S}^{(A,B)} = \mathfrak{P}_A\{\mathfrak{P}_B\{Z\}\},$$

where $\mathfrak{K}_X$ represents a construction with a number of components restricted to the integer set X. $\square$

EXAMPLE 5. *Set partitions with bounded block sizes.* Let $e_b(z)$ denote the truncated exponential function,

$$e_b(z) := 1 + \frac{z}{1!} + \frac{z^2}{2!} + \cdots + \frac{z^b}{b!}.$$

The EGFs

$$S^{\langle\leq b\rangle}(z) = \exp(e_b(z) - 1), \quad S^{\langle>b\rangle}(z) = \exp(e^z - e_b(z)),$$

correspond to partitions with all blocks of size $\leq b$ and all blocks of size $> b$, respectively. $\square$

$\triangleright$ **4.** The EGF of partitions without singleton parts is e^{e^z-1-z}. The EGF of "double surjections" (each preimage contains at least two elements) is $(2 - z - e^z)^{-1}$. $\triangleleft$

EXAMPLE 6. *Comtet's square.* An exercise in Comtet's book [**28**, Ex. 13, p. 225] serves beautifully to illustrate the power of symbolic methods. The question is to enumerate set partitions such that a parity constrained is satisfied by the number of blocks and/or the number of elements in each block. Then, the EGF's are tabulated as follows:

Set partitions	Any number of blocks	Odd number of blocks	Even number of blocks
Any block sizes	e^{e^z-1}	$\sinh(e^z-1)$	$\cosh(e^z-1)$
Odd block sizes	$e^{\sinh z}$	$\sinh(\sinh z)$	$\cosh(\sinh z)$
Even block sizes	$e^{\cosh z-1}$	$\sinh(\cosh z-1)$	$\cosh(\cosh z-1)$

The proof is a direct application of Proposition II.2, upon noting that

$$e^z, \quad \sinh z, \quad \cosh z$$

are the characteristic EGFs of $\mathbb{Z}_{\geq 0}$, $2\mathbb{Z}_{\geq 0}+1$, and $2\mathbb{Z}_{\geq 0}$ respectively. The sought EGFs are then obtained by forming the compositions

$$\left\{ \begin{array}{c} \exp \\ \sinh \\ \cosh \end{array} \right\} \circ \left\{ \begin{array}{c} \exp -1 \\ \sinh \\ \cosh -1 \end{array} \right\},$$

in accordance with general principles. □

II. 3.2. Applications to words and random allocations. The examples discussed now deal with enumerative problems that present themselves when analysing statistics on letters in words. They find applications in random allocations and the so-called "hashing algorithms" of computer science [**130**]. Fix an alphabet

$$\mathcal{X} = \{a_1, a_2, \ldots, a_r\}$$

of cardinality r, and let $\mathcal{W}$ be the class of all words over the alphabet $\mathcal{X}$, the size of a word being its length. A word of length n, $w \in \mathcal{W}_n$, can be viewed as an unconstrained function from $[1\,..\,n]$ to $[1\,..\,r]$, the function associating to each position the value of the corresponding letter in the word (canonically numbered from 1 to r). For instance, let $\mathcal{X} = \{a, b, c, d, r\}$ and take the letters of $\mathcal{X}$ canonically numbered as $a_1 = a, \ldots, a_5 = r$; for the word $w =$ '*abracadbra*', the table giving the position-to-letter mapping is

$$\left(\begin{array}{cccccccccccc} a & b & r & a & c & a & d & a & b & r & a \\ \hline 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 2 & 5 & 1 & 3 & 1 & 4 & 1 & 2 & 5 & 1 \end{array} \right),$$

which is itself determined by its sequence of preimages:

$$\overbrace{\{1,4,6,8,11\}}^{a=a_1}, \quad \overbrace{\{2,9\}}^{b=a_2}, \quad \overbrace{\{5\}}^{c=a_3}, \quad \overbrace{\{7\}}^{d=a_4}, \quad \overbrace{\{3,10\}}^{r=a_5}.$$

(Here, all preimages are nonempty, but this need not always the case.) The decomposition based on preimages then gives

$$\mathcal{W} \simeq \mathcal{U}^r \equiv \mathfrak{S}_r\{\mathcal{U}\}, \tag{16}$$

where $\mathcal{U}$ represents a possibly empty urn. As the EGF of $\mathcal{U}$ is $U(z) = e^z$, this construction implies that the EGF of all words is

$$W(z) = (e^z)^r = e^{rz}, \tag{17}$$

which yields back $W_n = r^n$, as was to be expected. For the situation where restrictions are imposed on the number of occurrences of letters, the decomposition (16) generalizes as follows.

PROPOSITION II.3. *Let $\mathcal{W}^{(A)}$ denote the family of words such that the number of occurrences of each letter lies in a set A. Then*

$$W^{(A)}(z) = (\alpha(z))^r \qquad \textit{where} \qquad \alpha(z) = \sum_{a \in A} \frac{z^a}{a!}. \tag{18}$$

Though this result is technically a shallow consequence of the symbolic method, it has several important applications in discrete probability; see [**130**, Ch. 8] for a discussion along the lines of the symbolic method.

EXAMPLE 7. *Restricted words.* The EGF of words containing *at most* b times each letter, and that of words containing *more* than b times each letter are

$$\mathcal{W}^{\langle \le b \rangle}(z) = (e_b(z))^r, \qquad \mathcal{W}^{\langle > b \rangle}(z) = (e^z - e_b(z))^r, \tag{19}$$

respectively. Taking $b = 1$ in the first formula gives the number of r–arrangements of n elements (i.e., ordered combinations of r elements amongst n) as

$$n! \, [z^n](1+z)^r = r! \binom{n}{r} = n(n-1)\cdots(n-r+1), \tag{20}$$

as anticipated; taking $b = 1$, but now in the second formula, gives back the number of r-surjections.

For general b, the generating functions of (19) contain valuable information on the least frequent and most frequent letter in random words. Some consequences are explored below. □

▷ **5.** *Number of different letters in words.* The probability that a random word of length n over an alphabet of cardinality r contains k different letters is

$$p_{n,k}^{(r)} := \frac{1}{r^n} \binom{r}{k} \left\{ n \atop k \right\} k!$$

(Choose k letters amongst r, then split the n positions into k distinguished nonempty classes.) The quantity $p_{n,k}^{(r)}$ is also the probability that a random mapping from $[1\,..\,n]$ to $[1\,..\,r]$ has an image of cardinality k.) ◁

▷ **6.** *Arrangements.* Define an *arrangement* of size n as an ordered combination of (some) elements of $[1\,..\,n]$, and let $\mathcal{A}$ be the class of all arrangements. Grouping together all the possible elements *not* present in the arrangement into an urn shows that a specification and its companion EGF are

$$\mathcal{A} \simeq \mathcal{U} \star \mathcal{P},\ \mathcal{U} = \mathfrak{P}\{\mathcal{Z}\},\ \mathcal{P} = \mathfrak{S}\{\mathcal{Z}\} \qquad \Longrightarrow \qquad A(z) = \frac{e^z}{1-z}.$$

The resulting counting sequence

$$A_n = \sum_{k=0}^{n} \frac{n!}{k!}$$

starts as $1, 2, 5, 16, 65, 326, 1957, 13700$ (*EIS* **A000522**); see also [**28**, p. 75]. ◁

▷ **7.** *Balls-switching-bins model.* There are m distinguishable balls and two bins (also called "urns") A and B. At any time $t = 1, 2, \ldots$, one of the balls changes bins. The EGF of the number of moves of duration $2n$ that start with urn A full (at $t = 0$) and end with urn A again full (at $t = 2n$) is

$$(2n)! \cdot [z^{2n}] \, (\cosh(z))^m.$$

[Hint: this the EGF enumerates mappings where each preimage has an even cardinality.] From there, one can generalize to the case where A contains k balls initially and ℓ balls finally. (This is Ehrenfest's simplified model of heat transfer that is analysed thoroughly in [**69**] by combinatorial methods.) ◁

EXAMPLE 8. *Random allocations (balls-in-bins model).* Throw at random n distinguishable balls into m distinguishable bins. A particular realization is described by a word of length n (balls are distinguishable, say, as numbers from 1 to n) over an alphabet of cardinality m (representing the bins chosen). Let Min and Max represent the size of the least filled and most filled bins, respectively. Then, the probabilistic model [1] has

$$\begin{aligned} \mathbb{P}\{\text{Max} \leq b\} &= n! \, [z^n] e_b \left(\tfrac{z}{m}\right)^m \\ \mathbb{P}\{\text{Max} > b\} &= n! \, [z^n] \left(e^{z/m} - e_b \left(\tfrac{z}{m}\right)\right)^m. \end{aligned} \tag{21}$$

The justification of this formula relies on the easy identity

$$\frac{1}{m^n} [z^n] f(z) \equiv [z^n] f(\frac{z}{m}), \tag{22}$$

and on the fact that a probability is determined as the ratio between the number of favorable cases (given by (19) and the total number of cases (m^n).

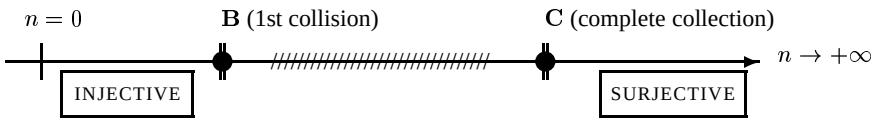
An especially interesting case is when m and n are asymptotically proportional, that is, $n/m = \alpha$ and α lies in a compact subinterval of $(0, +\infty)$. In that case, with probability tending to 1 as n tends to infinity, one has

$$\text{Min} = 0, \qquad \text{Max} \sim \frac{\log n}{\log \log n}.$$

In other words, there are almost surely empty urns (in fact many of them, see Ex. 8 in Chapter III) and the most filled urn grows logarithmically in size. Such probabilistic properties are best established by complex analytic methods (especially the saddle point method detailed in Chapter VI) based on exact generating representations like (19) and (21). They form the core of the reference book [**92**] by Kolchin, Sevastyanov, and Chistyakov. The resulting estimates are in turn invaluable in the analysis of hashing algorithms [**67, 86, 130**] to which the balls-in-bins model has been recognized to apply with great accuracy [**100**]. □

The next two examples illustrate applications of EGF's to two classical problems of probability theory, the "birthday paradox" and the "coupon collector problem". Assume there is a very long line of persons ready to enter a very large room one by one. Each person is let in and declares her birthday upon entering the room. How many people must enter in order to find two that have the same birthday? The "birthday paradox" is the counterintuitive fact that on average a birthday collision takes place as early as $n \doteq 24$. Dually, the "coupon collector problem" asks for the average number of persons that must enter in order to exhaust all the possible days in the year as birthdates. In this case, the answer is the rather large number $n' \doteq 2364$. (The term "coupon collection" alludes to the situation where images or coupons of various sorts are inserted in sales items and some premium is given to those who succeed in gathering a complete collection.) The birthday problem and the coupon collector problem are relative to a potentially infinite sequence of events; however, the fact that the first birthday collision or the first complete collection occurs at any fixed time n only involves finite events. The following diagram illustrates the events of interest:

[1] We let $\mathbb{P}(E)$ represent the probability of an event E and $\mathbb{E}(X)$ the expectation of the random variable X. Whenever necessary, subscripts may be used to indicate the probabilistic model of use.



In other words, we seek the time at which injectivity *ceases* to hold (the first birthday collision, B) and the time at which surjectivity *begins* to be satisfied (a complete collection, C). In what follows, we consider a year with r days (readers from earth may take $r = 365$) and let $\mathcal{X}$ represent an alphabet with r letters (the days in the year).

EXAMPLE 9. *Birthday paradox.* Let B be the time of the first collision, which is a random variable ranging between 2 and $r + 1$ (where the upperbound derives from the pigeonhole principle). A collision has not yet occurred at time n, if the sequence of birthdates $\beta_1, \ldots, \beta_n$ has no repetition. In other words, the function β from $[1, \, . \, n]$ to $\mathcal{X}$ must be injective; equivalently, $\beta_1, \ldots, \beta_n$ is an n-arrangement of r objects. Thus, we have the fundamental relation

$$\begin{aligned} \mathbb{P}\{B > n\} &= \frac{r(r-1)\cdots(r-n+1)}{r^n} \\ &= \frac{n!}{r^n}[z^n](1+z)^r \\ &= n!\,[z^n]\left(1 + \frac{z}{r}\right)^r, \end{aligned} \tag{23}$$

where the second line repeats (20) and the third results from the series transformation (22).

The expectation of the random variable B is

$$\mathbb{E}(B) = \sum_{n=0}^{\infty} \mathbb{P}\{B > n\}, \tag{24}$$

by virtue of a general formula valid for all discrete random variables. From (23), line 1, this gives us a sum expressing the expectation, namely,

$$\mathbb{E}(B) = 1 + \sum_{n=1}^{r} \frac{r(r-1)\cdots(r-n+1)}{r^n}.$$

For instance, with $r = 365$, one finds that the expectation is the rational number ,

$$\mathbb{E}(B) = \frac{12681\cdots06674}{51517\cdots40625} \doteq 24.61658,$$

where the denominator comprises as much as 864 digits.

An alternative form of the expectation derives from the generating function involved in (23), line 3. Let f be an entire function with nonnegative coefficients. Then the formula

$$f(z) = \sum_{n\geq 0} f_n z^n \quad \Longrightarrow \quad S := \sum_{n=0}^{\infty} f_n n! = \int_0^{\infty} e^{-t} f(t)\, dt, \tag{25}$$

is valid provided either the sum or the integral on the right converges. The reason is the usual Eulerian representation of factorials,

$$n! = \int_0^{\infty} e^{-t} t^n \, dt.$$

Applying this principle to (24) with the probabilities given by (23) (third line), one finds

$$\mathbb{E}(B) = \int_0^\infty e^{-t} \left(1 + \frac{t}{r}\right)^r \, dt. \tag{26}$$

This last form is easily amenable to asymptotic analysis and the Laplace method[2] delivers the estimation

$$\mathbb{E}(B) = \sqrt{\frac{\pi r}{2}} + \frac{2}{3} + O(r^{-1/2}), \tag{27}$$

as r tends to infinity. For instance, the asymptotic approximation provided by the first two terms of (27) is $24.61\mathit{19}$, which represents a relative error of only $2 \cdot 10^{-4}$.

The interest of such integral representations based on generating function is that they are *robust*: they adjust naturally to many kinds of combinatorial conditions. For instance, the expected time necessary for the first occurrence of the event "b persons have the same birthday" is found to have expectation given by the integral

$$I(r, b) := \int_0^\infty e^{-t} e_b \left(\frac{t}{r}\right)^r \, dt. \tag{28}$$

(The basic birthday paradox corresponds to $b = 2$.) The formula (28) was first derived by Klamkin and Newman in 1967; their paper [**80**] shows in addition that

$$I(r, b) \underset{r\to\infty}{\sim} \sqrt[b]{b!}\, \Gamma\left(1 + \frac{1}{b}\right) r^{1-1/b},$$

where the asymptotic form evaluates to 82.87 for $r = 365$ and $b = 3$, while the exact value of the expectation is 88.73891. Thus three-way collisions also tend to occur much sooner than one might think, with about 89 persons on average. Globally, such developments illustrate the versatility of the symbolic approach to many basic probabilistic problems. □

▷ **8.** *Birthday paradox with leap years.* Assume that the 29th of February exists precisely once every fourth year. What is the amplitude of the effect on the expectation of the first birthday collision? (Hint: one may wish to treat the general case of nonuniform date distributions;see Ex. 10 below.) ◁

EXAMPLE 10. *Coupon collector problem.* This problem is dual to the birthday paradox. We ask for the first time C when $\beta_1, \ldots, \beta_C$ contains all the elements of $\mathcal{X}$, that is, all the possible birthdates have been "collected". (The name "coupon collector" is due to the fact that in former times, chocolate bars would contain different coupons or images and collectors would be awarded some gift in exchange for a full collection.) In other words, the event $\{C \leq n\}$ means the equality between sets, $\{\beta_1, \ldots, \beta_n\} = \mathcal{X}$. Thus, the probabilities satisfy

$$\begin{aligned} \mathbb{P}\{C \leq n\} &= \frac{R_n^{(r)}}{r^n} = \frac{n! \left\{ {r \atop n} \right\}}{r^n} \\ &= \frac{n!}{r^n} [z^n] (e^z - 1)^r \\ &= n! [z^n] \left(e^{z/r} - 1\right)^r, \end{aligned} \tag{29}$$

by our earlier enumeration of surjections. The complementary probabilities are then

$$\mathbb{P}\{C > n\} = 1 - \mathbb{P}\{C \leq n\} = n! [z^n] \left(e^z - \left(e^{z/r} - 1\right)^r\right).$$

[2] Knuth [**85**, Sec. 1.2.11.3] uses this calculation as a pilot example for (real) asymptotic analysis; the quantity $\mathbb{E}(B)$ is related to Ramanujan's Q-function (see also Eq. (45) below) by $\mathbb{E}(B) = 1 + Q(r)$.

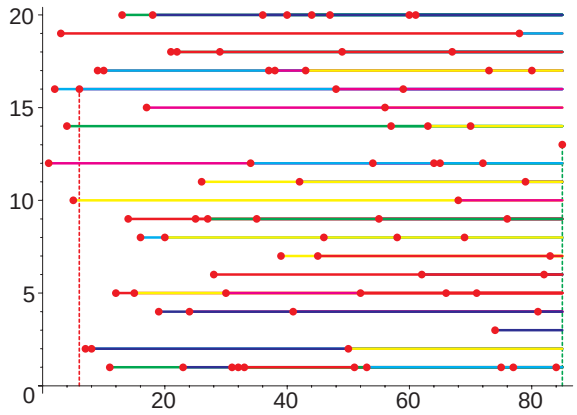


FIGURE 5. A sample realization of the "birthday paradox" and "coupon collection" with an alphabet of $r = 20$ letters. The first collision occurs at time $B = 6$ while the collection becomes complete at time $C = 87$.

An application of the Eulerian integral trick of (26) then provides a representation of the expectation of the time needed for a full collection as

$$\mathbb{E}(C) = \int_0^\infty \left(1 - (1 - e^{-t/r})^r\right)\, dt. \tag{30}$$

A simple calculation (expand by the binomial theorem and integrate termwise) shows that

$$\mathbb{E}(C) = r \sum_{j=1}^{r} \binom{r}{j} \frac{(-1)^{j-1}}{j},$$

which constitutes a first answer to the coupon collector problem in the form of an alternating sum. Alternatively, in (30), perform the change of variables $v = 1 - e^{-t/r}$, then expand and integrate termwise; this process provides the more tractable form

$$\mathbb{E}(C) = r\, \mathrm{H}_r, \tag{31}$$

where H_r is the harmonic number:

$$\mathrm{H}_r = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{r}.$$

(Formula (31) is by the way easy to interpret directly: one needs on average $1 = r/r$ trials to get the first day, then $r/(r-1)$ to get a different day, etc.)

Regarding (31), one has available the well-known formula (by comparing sums with integrals or by Euler-Maclaurin summation),

$$\mathrm{H}_r = \log r + \gamma + \frac{1}{2r} + O(r^{-2}), \quad \gamma \doteq 0.57721\,56649,$$

where γ is known as Euler's constant. Thus, the expected time for a full collection satisfies

$$\mathbb{E}(C) = r \log r + \gamma r + \frac{1}{2} + O(r^{-1}). \tag{32}$$

Here the "surprise" lies the nonlinear growth of the expected time for a full collection. For a year on earth, $r = 365$, the exact expected value is $\doteq 2364.64602$ while the approximation provided by the first three terms of (32) yields 2364.64625, representing a relative error of only one in ten millions.

Like before, the symbolic treatment adapts to a variety of situations, for instance, to multiple collections. The expected time till each item (birthday or coupon) is obtained b times (the standard case corresponds to $b = 1$) equals the quantity

$$J(r, b) = \int_0^\infty \left(1 - \left(1 - e_b(t/r)e^{-t/r}\right)^r\right) dt,$$

an expression that vastly generalizes (32). From there, one finds [**108**]

$$J(r, b) = n\left(\log n + (b-1)\log\log n + \gamma - \log(b-1)! + o(1)\right),$$

so that only a few more trials are needed in order to obtain additional collections. □

▷ **9.** *The little sister.* The coupon collector has a little sister to whom he gives his duplicates. Foata, Lass, and Han [**63**] show that the little sister misses on average H_r coupons when her big brother first obtains a complete collection. ◁

▷ **10.** *The original coupon collector problem.* A company issues coupons of r different types, type j being issued with probability p_j. Let C be the random variable representing the number of coupons that one needs to gather until a full collection with r different coupons is obtained. By the multivariate techniques of Chapter III, one has

$$\mathbb{P}\{C \le n\} = n![z^n]\prod_{j=1}^{r}(e^{p_j z} - 1).$$

The Eulerian integral gives for the expectation:

$$\mathbb{E}(C) = \int_0^\infty \left(1 - \prod_{j=1}^{r}(1 - e^{-p_i x})\right) dx.$$

See [**47**] for several variations on this theme and p. 141 for related context. ◁

What distinguishes a labelled structure from an unlabelled one? There is nothing intrinsic there, and everything is in the eye of the beholder! (Or rather in the type of construction adopted when modelling a specific problem.) Take the class of words $\mathcal{W}$ over an alphabet of cardinality r. The two generating functions

$$\widehat{W}(z) \equiv \sum_n W_n z^n = \frac{1}{1 - rz} \qquad \text{and} \qquad W(z) \equiv \sum_n W_n \frac{z^n}{n!} = e^{rz},$$

leading in both cases to $W_n = r^n$, correspond to two different ways of constructing words, the first one directly as an unlabelled sequence, the other one as a labelled power of letter positions. A similar situation arises for r–partitions, for which we found as OGF and EGF,

$$\widehat{S}^{(r)}(z) = \frac{z^r}{(1-z)(1-2z)\cdots(1-rz)} \qquad \text{and} \qquad S^{(r)}(z) = \frac{(e^z - 1)^r}{r!},$$

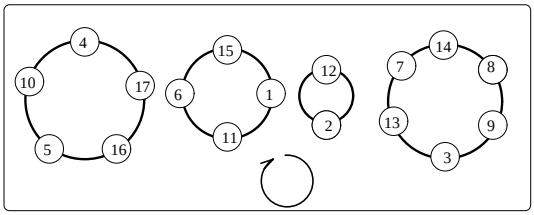
by viewing these either as unlabelled structures (an encoding via words of a regular language, see Section I.4.3) or directly as labelled structures.

II. 4. Alignments, permutations, and related structures

In this section, we start by considering the specifications,

$$\mathcal{O} = \mathfrak{S}\{\mathfrak{C}\{\mathcal{Z}\}\}, \qquad \text{and} \qquad \mathcal{P} = \mathfrak{P}\{\{\mathcal{Z}\}\}, \tag{33}$$

built by piling up two constructions, sequences-of-cycles and sets-of-cycles respectively. They define a new class of objects, called alignments ($\mathcal{O}$), while serving to specify permutations ($\mathcal{P}$) in a novel way as detailed below. (These specifications otherwise parallel surjections and set partitions.) Permutations are in this context examined under their cycle



A permutation may be viewed as a *set* of cycles that are labelled circular digraphs. The diagram shows the decomposition of the permutation

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \\ 11 & 12 & 13 & 17 & 10 & 15 & 14 & 9 & 3 & 4 & 6 & 2 & 7 & 8 & 1 & 5 & 16 \end{pmatrix}.$$

(Cycles read clockwise and i is connected to σ_i in the graph.)

FIGURE 6. The cycle decomposition of permutations.

decomposition, the corresponding enumerative results being the most important ones combinatorially (Section II. 4.1). In Section II. 4.2, we recapitulate the meaning of classes that can be defined iteratively by a combination of any two nested labelled constructions.

II. 4.1. Alignments and Permutations. Define first an alignment as a well-labelled sequence of cycles and let $\mathcal{O}$ be the class of all alignments. Let $\mathcal{P}$ be defined momentarily as the class of all sets of cycles. The corresponding specifications are then clearly the ones of (33).

By the symbolic method, alignments have EGF

$$\begin{aligned} O(z) &= \frac{1}{1-\log(1-z)^{-1}} \\ &= 1 + z + 3\frac{z^2}{21} + 14\frac{z^3}{3!} + 88\frac{z^4}{4!} + 694\frac{z^5}{5!} + \cdots, \end{aligned}$$

which does not simplify. The coefficients form *EIS* **A007840** ("ordered factorizations of permutations into cycles").

From elementary mathematics, it is known that a permutation admits a unique decomposition into cycles. Let $\sigma = \sigma_1 \dots \sigma_n$ be as permutation. Start with any element, say 1, and draw a directed edge from 1 to $\sigma(1)$, then continue connecting to $\sigma^{\,2}(1), \sigma^3(1)$, and so on; a cycle containing 1 is obtained after at most n steps. If one repeats the construction, taking at each stage an element not yet connected to earlier ones, the cycle decomposition of the permutation σ is obtained. This argument shows that the class of "sets-of-cycles" (corresponding to $\mathcal{P}$ in (33)) is isomorphic to the class of permutations as defined in Section II. 1:

$$\mathcal{P} = \mathfrak{P}\{\mathfrak{C}\{\mathcal{Z}\}\} \cong \mathfrak{S}\{\mathcal{Z}\}.$$

This combinatorial isomorphism is reflected by the obvious series identity

$$P(z) = \exp\left(\log \frac{1}{1-z}\right) = \frac{1}{1-z}.$$

In a sense, the property that exp and log are inverse of one another is an analytic reflex of the combinatorial fact that permutations uniquely decompose into cycles!

As regards combinatorial applications, what is especially fruitful is the variety of specializations of the construction of permutations from cycles. We state:

PROPOSITION II.4. *Let $\mathcal{P}^{(A,B)}$ be the class of permutations with cycle lengths in $A \subseteq \mathbb{Z}_{>0}$ and with a number of cycles that belongs to $B \subseteq \mathbb{Z}_{\geq 0}$. The corresponding EGF is*

$$P^{(A,B)}(z) = \beta(\alpha(z)) \qquad \textit{where} \qquad \alpha(z) = \sum_{a \in A} \frac{z^a}{a},\ \beta(z) = \sum_{b \in B} \frac{z^b}{b!}.$$

EXAMPLE 11. *Stirling cycle numbers.* The number of permutations of size n comprised of r cycles is determined by the explicit generating function

$$P_n^{(r)} = \frac{n!}{r!}\,[z^n]\,\left(\log \frac{1}{1-z}\right)^r. \tag{34}$$

These numbers are fundamental quantities of combinatorial analysis. They are known as the Stirling numbers of the first kind, or better, according to a proposal of Knuth, the *Stirling "cycle" numbers*. Together with the Stirling partition numbers, the properties of the Stirling cycle numbers are explored in the book by Graham, Knuth, and Patashnik [**71**] where they are denoted by $\left[{n \atop r}\right]$. See APPENDIX: *Stirling numbers*, p. 173. (Note that the number of alignments formed with r cycles is $r!\left[{n \atop r}\right]$.) As we shall see shortly (p. 99) Stirling numbers also surface in the enumeration of permutations by their number of records.

It is also of interest to determine what happens regarding cycles in a random permutation of size n. Clearly, when the uniform distribution is put on all elements of $\mathcal{P}_n$, each particular permutation has probability exactly $1/n!$. Since the probability of an event is the quotient of the number of favorables cases over the total number of cases, the quantity

$$p_{n,k} := \frac{1}{n!}\left[{n \atop k}\right]$$

is the probability that a random element of P_n has n cycles. This probabilities can be effectively determined for "reasonable" values of n from (34), preferably by means of a computer algebra system. Here are for instance selected values for $n = 100$:

$k:$	1	2	3	4	5	6	7	8	9	10
$p_{n,k}:$	0.01	0.05	0.12	0.19	0.21	0.17	0.11	0.06	0.03	0.01

so that, for this value of n, we expect in a vast majority of cases the number of cycles to be in the interval $[1, 10]$. (The residual probability is only about 0.005.) Under this probabilistic model, the mean is found to be about 5.18. Thus: *On average, a random permutation of size 100 has a little more than 5 cycles*.

Such procedures demonstrate a direct exploitation of symbolic methods. They do not however tell us how the number of cycles could depend on n as n varies. Such questions are to be examined systematically in Chapter III. Here, we shall content ourselves with a brief sketch. First, form the bivariate generating function,

$$P(z,u) := \sum_{r=0}^{\infty} P^{(r)}(z) u^r,$$

and observe that

$$\begin{aligned} P(z,u) &= \sum_{r=0}^{\infty} \frac{u^r}{r!}\left(\log \frac{1}{1-z}\right)^r = \exp\left(u \log \frac{1}{1-z}\right) \\ &= (1-z)^{-u}. \end{aligned}$$

Newton's binomial theorem then provides

$$[z^n](1-z)^{-u} = (-1)^n \binom{-u}{n}.$$

In other words, a simple formula

$$\sum_{k=0}^{n} \begin{bmatrix} n \\ k \end{bmatrix} u^k = u(u+1)(u+2)\cdots(u+n-1) \tag{35}$$

describes precisely the distribution of Stirling cycle numbers for any fixed value of n. From there, the expected number of cycles, $\mu_n := \sum_k k p_{n,k}$ is easily found (use logarithmic differentiation of (35)),

$$\mu_n = \mathrm{H}_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}.$$

In particular, one has $\mu_{100} \equiv \mathrm{H}_{100} \doteq 5.18738$. In general: *The mean number of cycles in a random permutation of size n grows logarithmically with n, $\mu_n \sim \log n$.* □

EXAMPLE 12. *Involutions and permutations without long cycles.* A permutation σ is an *involution* if $\sigma^2 = Id$ with Id the identity permutation. Quite clearly, an involution can have only cycles of sizes 1 and 2. The class $\mathcal{I}$ of all involutions thus satisfies $\mathcal{I} = \mathfrak{P}\{\mathfrak{C}_{1,2}\{\mathcal{Z}\}\}$. The translation is immediate:

$$I(z) \equiv \sum_n I_n \frac{z^n}{n!} = \exp\left(z + \frac{z^2}{2}\right). \tag{36}$$

This last equation then provides the formula

$$I_n = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{n!}{(n-2k)!2^k k!},$$

which solves the counting problem explicitly. A *pairing* is an involution without fixed point; in other words, only cycles of length 2 are allowed. The EGF and the number of all pairings are given by

$$J(z) = e^{z^2/2}, \qquad J_{2n} = 1 \cdot 3 \cdot 5 \cdots (2n-1)$$

as was to be anticipated from a direct reasoning.

Generally, the EGF of permutations, all of whose cycles (in particular the largest one) have length at most equal to r satisfies

$$B^{(r)}(z) = \exp\left(\sum_{j=1}^{r} \frac{z^j}{j}\right).$$

The numbers $b_n^{(r)} = [z^n]B^{(r)}(z)$ satisfy the recurrence

$$(n+1)b_{n+1}^{(r)} = (n+1)b_n^{(r)} - b_{n-r}^{(r)},$$

by which they can be computed fast. This gives access to the statistics of the longest cycle in a permutation. □

All perms	Derangements	Involutions	Pairings
$\frac{1}{1-z}$	$\frac{e^{-z}}{1-z}$	$e^{z+z^2/2}$	$e^{z^2/2}$
Shortest cycle $> r$		Longest cycle $\leq r$	
$\frac{e^{-\ell_r(z)}}{1-z}$		$e^{\ell_r(z)}$	

FIGURE 7. A summary of major EGFs related to permutations. There, $\ell_r(z) := \sum_{j=1}^{r} \frac{z^j}{j}$ is the "truncated logarithm".

▷ **11.** *Permutations such that $\sigma^e = Id$.* Such permutations are "roots of unity" in the symmetric group. Their EGF is

$$\exp\left(\sum_{d \mid e} \frac{z^d}{d}\right),$$

where the sum extends to all divisors d of e. ◁

EXAMPLE 13. *Derangements and permutations without short cycles.* Classically, a derangement is defined as a permutation without fixed points, i.e., $\sigma_i \neq i$ for all i. Given an integer r, an r–derangement is a permutation all of whose cycles (in particular the shortest one) have length larger than r. Let $\mathcal{D}^{(r)}$ be the class of all r–derangements. A specification is

$$\mathcal{D}^{(r)} = \mathfrak{P}\{\mathfrak{C}_{>r}\{\mathcal{Z}\}\}, \tag{37}$$

the corresponding EGF being then

$$D^{(r)}(z) = \exp\left(\sum_{j>r} \frac{z^j}{j}\right) = \frac{\exp(-\sum_{j=1}^{r} \frac{z^j}{j})}{1-z}. \tag{38}$$

For instance, when $r = 1$, a direct expansion yields

$$\frac{D_n^{(1)}}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!},$$

a truncation of the series expansion of $\exp(-1)$ that converges fast to e^{-1}. Phrased differently, the enumeration of derangements is a famous combinatorial problem with a pleasantly quaint nineteenth century formulation [**28**]: "A number n of people go to opera, leave their hats on hook in the cloakroom and grab them at random when leaving; the probability that nobody gets back his own hat is asymptotic to $1/e$, which is nearly 37%". (The usual proof uses an inclusion-exclusion argument Also, it is a sign of changing times that Motwani and Raghavan [**107**, p. 11] describe the problem as one of sailors that return to their ship in state of inebriation and choose random cabins to sleep in.) For the generalized derangement problem, there holds

$$\frac{D_n^{(r)}}{n!} \sim e^{-\mathrm{H}_r}, \tag{39}$$

(for any fixed r), as can be proved easily by complex asymptotic methods (Chapter IV). □

Like several other structures that we have been considering previously, permutation allow for transparent connections between structural constraints and the shapes of generating functions. The major counting results encountered in this section are summarized in Figure 7.

▷ **12.** *Parity constraints in permutations.* The EGF's of permutations having only even size cycles ($E(z)$) or odd size cycles ($O(z)$) are

$$E(z) = \exp\left(\frac{1}{2}\log\frac{1}{1-z^2}\right) = \frac{1}{\sqrt{1-z^2}}, \qquad O(z) = \exp\left(\frac{1}{2}\log\frac{1+z}{1-z}\right) = \sqrt{\frac{1+z}{1-z}}.$$

From the EGFs, one finds $E_{2n} = (1\cdot 3\cdot 5\cdots(2n-1))^2$, $O_{2n} = E_{2n}$, $O_{2n+1} = (2n+1)E_{2n}$.

The EGF's of permutations having an even number of cycles ($E^*(z)$) and an odd number of cycles ($O^*(z)$) are

$$E^*(z) = \cosh(\log\frac{1}{1-z}) = \frac{1}{2}\,\frac{2-z^2}{1-z}, \qquad O^*(z) = \sinh(\log\frac{1}{1-z}) = \frac{1}{2}\,\frac{z^2}{1-z}.$$

so that parity of the number of cycles is evenly distributed amongst permutations of size n as soon as $n \geq 2$. (The generating functions obtained in this way are analogous to the ones appearing in the discussion of "Comtet's square" in the previous section.) ◁

II. 4.2. Second level structures. Consider the three basic constructors of labelled sequence ($\mathfrak{S}$), set ($\mathfrak{P}$), and cycle ($\mathfrak{C}$). We can play the formal game of examining what the various combinations produce as combinatorial objects. Restricting attention to superpositions of two constructors (an "external" one applied to an "internal" one) gives 9 possibilities summarized by the following table:

ext.\int.	$\mathfrak{S}_{\geq 1}$	$\mathfrak{P}_{\geq 1}$	$\mathfrak{C}$
	"Labelled compositions" ($\mathcal{L}$)	Surjections ($\mathcal{R}$)	Alignments ($\mathcal{O}$)
$\mathfrak{S}$	$\mathfrak{S}\circ\mathfrak{S}$ $\frac{1-z}{1-2z}$	$\mathfrak{S}\circ\mathfrak{P}$ $\frac{1}{2-e^z}$	$\mathfrak{S}\circ\mathfrak{C}$ $\frac{1}{1-\log(1-z)^{-1}}$
	"Fragmented permutations" ($\mathcal{F}$)	Set partitions ($\mathcal{S}$)	Permutations ($\mathcal{P}$)
$\mathfrak{P}$	$\mathfrak{P}\circ\mathfrak{S}$ $e^{z/(1-z)}$	$\mathfrak{P}\circ\mathfrak{P}$ e^{e^z-1}	$\mathfrak{P}\circ\mathfrak{C}$ $\frac{1}{1-z}$
	"Supernecklaces^1"	*"Supernecklaces^2"*	*"Supernecklaces^3"*
$\mathfrak{C}$	$\mathfrak{C}\circ\mathfrak{S}$ $\log\frac{1-z}{1-2z}$	$\mathfrak{C}\circ\mathfrak{P}$ $\log(1-e^z)^{-1}$	$\mathfrak{C}\circ\mathfrak{C}$ $\log\frac{1}{1-\log(1-z)^{-1}}$

The classes of surjections, alignments, set partitions, and permutations appear naturally as $\mathfrak{S}\circ\mathfrak{P}$, $\mathfrak{S}\circ\mathfrak{C}$, $\mathfrak{P}\circ\mathfrak{P}$, and $\mathfrak{P}\circ\mathfrak{C}$. The other ones represent essentially nonclassical objects. The case of $\mathcal{L}$ corresponding to $\mathfrak{S}\circ\mathfrak{S}$ describes a class whose elements are (ordered) sequences of linear graphs. This can be interpreted as permutations with separators inserted, e.g, $53|264|5$, or alternatively as integer compositions with a labelling superimposed. Finally, the class $\mathcal{F} = \mathfrak{P}\{\mathfrak{S}_{\geq 1}\{\mathcal{Z}\}\}$ corresponds to unordered collections of permutations. In other words, "fragments" are obtained by breaking a permutation into pieces (pieces must be nonempty for definiteness). The interesting EGF is

$$F(z) = e^{z/(1-z)} = 1 + z + 3\frac{z^2}{2!} + 13\frac{z^3}{3} + 73\frac{z^4}{4!} + \cdots,$$

whose coefficients constitute *EIS* **A000262** ("sets of lists"). What we termed "superneck-laces" in the last column represents cyclic arrangements of composite objects existing in three brands.

All sorts of refinements, of which Figure 7 may give an idea, are clearly possible. We leave to the reader's imagination the task of determining which amongst the level 3 structures may be of combinatorial interest. . .

▷ **13.** *Counting specifications of level n.* The algebra of constructions satisfies the combinatorial isomorphism $\mathfrak{P}\{\mathfrak{C}\{\mathcal{X}\}\} \cong \mathfrak{S}\{X\}$ for all $\mathcal{X}$. How many different terms of length n can be built from three symbols $\mathfrak{C}, \mathfrak{P}, \mathfrak{S}$ satisfying a semi-group law ('$\circ$') together with the relation $\mathfrak{P} \circ \mathfrak{C} = \mathfrak{S}$? This determines the number of specifications of level n. (Hint: the OGF is rational as it corresponds to words with an excluded pattern.) ◁

II. 5. Labelled trees, mappings, and graphs

In this section, we consider labelled trees and certain labelled objects that are naturally associated with them, namely mappings and functional graphs on one side, graphs of small excess on the other side. Like in the unlabelled case considered in Section I. 6, the corresponding combinatorial classes are inherently recursive.

II. 5.1. Trees. The trees to be studied here are rooted and labelled, meaning as usual that a node is distinguished as the root and that nodes bear distinct integer labels. Labelled trees, like their unlabelled counterparts, exist in two varieties: (i) plane trees where an embedding in the plane is understood (or, equivalently, subtrees dangling from a node are ordered, say, from left to right); (ii) nonplane trees where no such embedding is imposed (such trees are then nothing but connected directed acyclic graphs with a distinguished root). Trees may be further restricted by the additional constraint that the node outdegrees should belong to a fixed set $\Omega \subseteq \mathbb{Z}_{\geq 0}$ where $\Omega \ni 0$.

We first dispose of the plane variety of labelled trees. Let $\mathcal{A}$ be the set of (rooted labelled) plane trees constrained by Ω. This family is specified by

$$\mathcal{A} = \mathcal{Z} \star \mathfrak{S}_\Omega\{\mathcal{A}\},$$

where $\mathcal{Z}$ represents the atomic class consisting of a single labelled node: $\mathcal{Z} = \{1\}$. The sequence construction appearing here reflects the planar embedding of trees, as subtrees

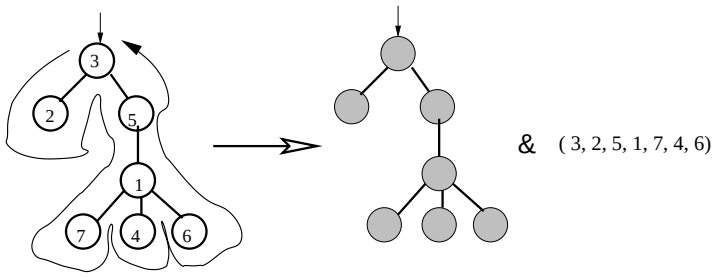


FIGURE 8. A labelled plane tree is determined by an unlabelled tree (the "shape") and a permutation of the labels $1, \ldots, n$.

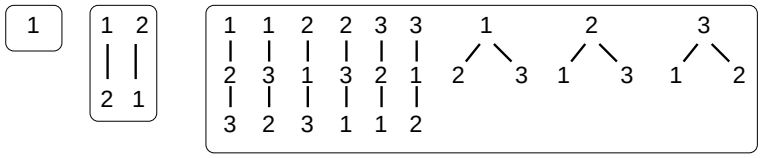


FIGURE 9. There are $T_1 = 1, T_2 = 2, T_3 = 9$, and in general $T_n = n^{n-1}$ Cayley trees of size n.

stemming from a common root are ordered between themselves. Accordingly, the EGF $A(z)$ satisfies

$$A(z) = z\phi(A(z)) \qquad \text{where} \quad \phi(u) = \sum_{\omega \in \Omega} u^{\omega}.$$

This is exactly the same equation as the one satisfied by the *ordinary* GF of Ω-restricted *unlabelled* plane trees (see Proposition I.4). Thus, $\frac{1}{n!}A_n$ is the number of unlabelled trees. In other words: *in the plane rooted case, the number of labelled trees equals $n!$ times the corresponding number of unrooted trees.* As illustrated by Figure 8, this is easily understood combinatorially: each labelled tree can be defined by its "shape" that is an unlabelled tree and by the sequence of node labels where nodes are traversed in some fixed order (preorder, say). Finally, one has, by Lagrange inversion,

$$A_n = n![z^n]A(z) = (n-1)![u^{n-1}]\phi(u)^n.$$

This simple analytic–combinatorial relation enables us to transpose all of the enumerative results of Section I.5.1 to plane labelled trees (upon multiplying the evaluations by $n!$, of course). In particular, the total number of "general" plane labelled trees (with no degree restriction imposed, i.e., $\Omega = \mathbb{Z}_{\geq 0}$) is

$$n! \times \frac{1}{n}\binom{2n-2}{n-1} = \frac{(2n-2)!}{(n-1)!} = 2^{n-1}\left(1 \cdot 3 \cdots (2n-3)\right).$$

The corresponding sequence starts as $1, 2, 12, 120, 1680$ and is *EIS* **A001813**.

We next turn to labelled nonplane trees (Figure 9) to which the rest of this section will be devoted. The class $\mathcal{T}$ of all such trees is definable by the symbolic equation

$$\mathcal{T} = \mathcal{Z} \star \mathfrak{P}\{\mathcal{T}\}, \tag{40}$$

where the set construction translates the fact that subtees stemming from the root are not ordered between themselves. From the specification (40), the EGF $T(z)$ is defined implicitly by the "functional equation"

$$T(z) = ze^{T(z)}. \tag{41}$$

The first few values are easily found:

$$T(z) = z + 2^1\frac{z^2}{2!} + 3^2\frac{z^3}{3!} + 4^3\frac{z^4}{4!} + 5^4\frac{z^5}{5!} + \cdots.$$

This leads to expect that

$$T_n = n^{n-1} \tag{42}$$

a fact proved (once more) by the Lagrange Inversion Theorem (see APPENDIX: *Lagrange Inversion*, p. 170):

$$\frac{T_n}{n!} = [z^n]T(z) = \frac{1}{n}[u^{n-1}](e^z)^n = \frac{n^{n-1}}{n!}.$$

The enumerative result $T_n = n^{n-1}$ is a famous one, attributed to the prolific British mathematician Arthur Cayley (1821–1895) who had keen interest in combinatorial mathematics and published altogether over 900 papers and notes. Consequently, formula (42) given by Cayley in 1889 is often referred to as "Cayley's formula" and unrestricted nonplane labelled tees are often called "Cayley trees". See [**17**, p. 51] for a historical discussion. The simplicity of Cayley's formula calls for a combinatorial explanation. The most famous one due to Prüfer (in 1918); see [**17**, p. 53] or [**105**, p. 5] for a description of the Prüfer encoding of trees by sequences. The function $T(z)$ is also known as the (Cayley) "tree function"; it is a close relative of the W–function [**29**] defined implicitly by $We^W = z$, which was introduced by the Swiss mathematician Johann Lambert (1728–1777) otherwise famous for first proving the irrationality of the number π.

A similar process gives the number of trees where all (out)degrees of nodes are restricted to lie in a set Ω. This corresponds to the specification

$$\mathcal{T}^{(\Omega)} = \mathcal{Z} \star \mathfrak{P}_\Omega\{\mathcal{T}^{(\Omega)}\},$$

which translates directly into an EGF equation,

$$T^{(\Omega)}(z) = z\overline{\phi}(T^{(\Omega)}(z)) \qquad \text{where} \quad \overline{\phi}(u) = \sum_{\omega\in\Omega} \frac{u^\omega}{\omega!},$$

and is amenable to Lagrange inversion. What this last formula involves is the "exponential characteristic" of the degree sequence (as opposed to the ordinary characteristic, in the planar case). In summary:

PROPOSITION II.5. *The number of trees, where all nodes have their outdegree in Ω, is*

$$T_n^{(\Omega)} = (n-1)![u^{n-1}](\overline{\phi}(u))^n \qquad \textit{where} \quad \overline{\phi}(u) = \sum_{\omega\in\Omega} \frac{u^\omega}{\omega!}.$$

▷ **14.** *Forests.* The number of unordered k–forests (i.e., k–sets of trees) is

$$F_n^{(k)} = n![z^n]\frac{(T(z))^k}{k!} = \frac{(n-1)!}{(k-1)!}[u^{n-k}](e^u)^n = \binom{n-1}{k-1} n^{n-k},$$

as follows from Bürmann's form of Lagrange inversion. ◁

▷ **15.** *Labelled hierarchies.* The class $\mathcal{L}$ of labelled hierarchies is formed of trees whose internal nodes are unlabelled and are constrained to have outdegree larger than 1, while leaves have labels attached to them. Like for other labelled structure, size is the number of labels (so that internal nodes do not contribute). Hierarchies satisfy the specification

$$\mathcal{L} = \mathcal{Z} + \mathfrak{P}_{\geq 2}\{\mathcal{L}\},$$

so that $L(z)$ satisfies $L = z + e^L - 1 - L$, and

$$L(z) = T(\frac{1}{2}e^{z/2-1/2}) + \frac{z}{2} - \frac{1}{2} = z + \frac{z^2}{2!} + 4\frac{z^3}{3!} + 26\frac{z^4}{4!} + 236\frac{z^5}{5!} + \cdots$$

(*EIS* **A000311**), with T being the Cayley tree function. The numbers count "phylogenetic trees" (used to describe the evolution of a genetically related group of organisms) and correspond to Schröder's "fourth problem"; see [**28**, p. 224] and Section I.5.2 for unlabelled analogues.

The class of binary (labelled) hierarchies defined by the additional fact that internal nodes can have degree 2 only corresponds to $\mathcal{M} = \mathcal{Z} + \mathfrak{P}_2\{\mathcal{M}\}$, so that

$$M(z) = 1 - \sqrt{1-2z} \qquad \text{and} \qquad M_n = 1\cdot 3\cdots(2n-3),$$

where the counting numbers are the odd factorials. ◁

II. 5.2. Mappings and functional graphs. Let $\mathcal{F}$ be the class of mappings (or "functions") from $[1\,..\,n]$ to itself. A mapping $f \in [1\,..\,n] \mapsto [1\,..\,n]$ can be represented by a directed graph over the set of vertices $[1\,..\,n]$ with an edge connecting x to $f(x)$, for all $x \in [1\,..\,n]$. The graphs so obtained are called *functional graphs* and they have the characteristic property that the outdegree of each vertex is exactly equal to 1.

Given a mapping (or function) f, upon starting from any point x_0, the succession of (directed) edges in the graph traverses the iterates of the mapping, $x_0, f(x_0), f(f(x_0)), \ldots$, and since the domain is finite, each such sequence must eventually loop on itself. When the operation is repeated, the elements group themselves into components. This leads to another characterization of functional graphs (Figure 10): *A functional graph is a set of connected functional graphs. A connected functional graph is a collection of rooted trees arranged in a cycle.*

Thus, with $\mathcal{T}$ being as before the class of all Cayley trees, and with $\mathcal{K}$ the class of all connected functional graphs, we have the specification:

$$\left\{ \begin{array}{rcl} \mathcal{F} & = & \mathfrak{P}\{\mathcal{K}\} \\ \mathcal{K} & = & \mathfrak{C}\{\mathcal{T}\} \\ \mathcal{T} & = & \mathcal{Z} \star \mathfrak{P}\{\mathcal{T}\}. \end{array} \right. \tag{43}$$

This translates at sight into a set of equations for EGF's

$$\left\{ \begin{array}{rcl} F(z) & = & e^{K(z)} \\ K(z) & = & \log \dfrac{1}{1-T(z)} \\ T(z) & = & z e^{T(z)}. \end{array} \right. \tag{44}$$

Eventually, the EGF $F(z)$ is found to satisfy $F = (1-T)^{-1}$. It can be checked from there, by Lagrange inversion once more, that we have

$$F_n = n^n,$$

as was to be expected (!) from the origin of the problem. More interestingly, Lagrange inversion also provides for the number of connected functional graphs (expand $\log(1 -$

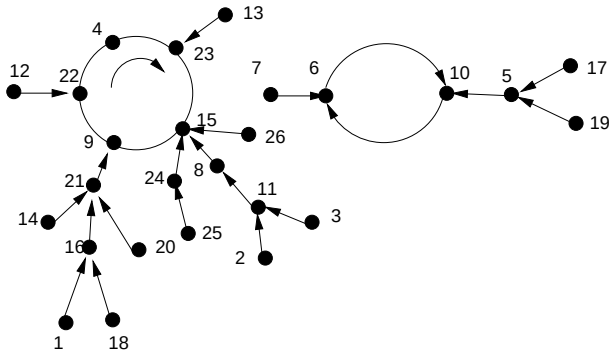


FIGURE 10. A functional graph of size $n = 26$ associated to the mapping φ such that $\varphi(1) = 16$, $\varphi(2) = \varphi(3) = 11$, $\varphi(4) = 23$, and so on.

$T)^{-1}$ and recover coefficients by Bürmann's form):

$$K_n = n^{n-1} Q(n) \quad \text{where} \quad Q(n) := 1 + \frac{n-1}{n} + \frac{(n-1)(n-2)}{n^2} + \dots . \tag{45}$$

The quantity $Q(n)$ that appears in (45) is a famous one that surfaces in many problems of discrete mathematics (including the birthday paradox, Equation (26)). Knuth has proposed to call it "Ramanujan's Q–function" as it already appears in the first letter of Ramanujan to Hardy in 1913. The asymptotic analysis can be done elementarily by developing a continuous approximation of the general term and approximating the resulting Riemman sum by an integral: this is an instance of the Laplace method for sums (see [**85**, Sec. 1.2.11.3], [**130**, Sec. 4.7] as well as Ex. 2). In fact, very precise estimates come out naturally from an analysis of the singularities of the EGF $K(z)$, as we shall see in Chapters IV and V. The net result is

$$K_n \sim n^n \sqrt{\frac{\pi}{2n}},$$

so that a fraction about $1/\sqrt{n}$ of all the graphs consist of a single component.

As is customary with the symbolic method, the constructions (43) also lead to a large number of related counting results. For instance, the mappings without fixed points, $((\forall x)\ f(x) \neq x)$ and those without $1, 2$–cycles, (additionally, $(\forall x)\ f(f(x)) \neq x$), have EGFs

$$\frac{e^{-T(z)}}{1-T(z)}, \qquad \frac{e^{-T(z)-T^2(z)/2}}{1-T(z)}.$$

The first equation is consistent with what a direct count yields, namely $(n-1)^n$, which is asymptotic to $e^{-1}n^n$, so that the fraction of mappings without fixed point is asymptotic to e^{-1}. The second one lends itself easily to complex-asymptotic methods that give

$$n![z^n]\frac{e^{-T-T^2/2}}{1-T} \sim e^{-3/2} n^n,$$

and the proportion is asymptotic to $e^{-3/2}$. These two particular estimates are of the same form as what has been found for permutations (the generalized derangements, Eq. (39)). Such facts that are not quite obvious by elementary probabilistic arguments are in fact neatly explained by the singular theory of combinatorial schemas developed in Chapter IV.

Next, idempotent mappings satisfying $f(f(x)) = f(x)$ correspond to $\mathcal{I} \cong \mathfrak{P}\{\mathcal{Z} \star \mathfrak{P}\{\mathcal{Z}\}\}$, so that

$$I(z) = e^{ze^z} \qquad \text{and} \qquad I_n = \sum_{k=0}^{n} \binom{n}{k} k^{n-k}.$$

(The specification translates the fact that idempotent mappings can have only cycles of length 1 on which are grafted sets of direct antecedents.) The latter sequence is *EIS* **A000248**, which starts as 1,1,3,10,41,196,1057. An asymptotic estimate can be derived either from the Laplace method or, better, from the saddle point method exposed in Chapter V.

Several analyses of this type are of relevance to cryptography and the study of random number generators. For instance, the fact that a random mapping over $[1\,..\,n]$ tends to reach a cycle in $O(\sqrt{n})$ steps led Pollard to design a Monte Carlo integer factorization algorithm, see [**86**, p. 371] and [**130**, Sec 8.8]. The algorithm once suitably optimised first led to the factorization of the Fermat number $F_8 = 2^{2^8} + 1$ obtained by Brent in 1980.

▷ **16.** *Binary mappings.* The class $\mathcal{BF}$ of binary mappings, where each point has either 0 or 2 preimages, is specified by

$$\mathcal{BF} = \mathfrak{P}\{\mathcal{K}\},\ \mathcal{K} = \mathfrak{C}\{\mathcal{P}\},\ \mathcal{P} = \mathcal{Z} \star \mathcal{B},\ \mathcal{B} = \mathcal{Z} \star \mathfrak{P}_{0,2}\{\mathcal{B}\}$$

All mappings $\frac{1}{1-T}$	Partial $\frac{e^T}{1-T}$	Injective partial $\frac{1}{1-z}e^{z/(1-z)}$	Surjection $\frac{1}{2-e^z}$	Bijection $\frac{1}{1-z}$
Connected ($\mathcal{K}$) $\log\frac{1}{1-T}$	No fixed point $\frac{e^{-T}}{1-T}$	Involution $e^{z+z^2/2}$	Idempotent e^{ze^z}	Binary $\frac{1}{\sqrt{1-2z^2}}$

FIGURE 11. A summary of various counting EGFs relative to mappings.

(planted trees $\mathcal{P}$ and binary trees $\mathcal{B}$ are needed), so that

$$BF(z) = \frac{1}{\sqrt{1-2z^2}}, \qquad BF_{2n} = \frac{((2n)!)^2}{2^n (n!)^2}.$$

The class $\mathcal{BF}$ is an approximate model of the behaviour of (modular) quadratic functions under iteration. See [**6, 53**] for a general enumerative theory of random mappings including degree-restricted ones. ◁

▷ **17.** *Partial mappings.* A partial mapping may be undefined at some points, where it can be considered as taking a special value, $\perp$. The iterated preimages of $\perp$ form a forest, while the remaining values organize themselves into a standard mapping. The class $\mathcal{PF}$ of partial mappings is thus specified by $\mathcal{PF} = \mathfrak{P}\{\mathcal{T}\} \star \mathcal{F}$, so that

$$PF(z) = \frac{e^{T(z)}}{1-T(z)} \qquad \text{and} \qquad PF_n = (n+1)^n.$$

This construction lends itself to all sorts of variations. For instance, the class PFI of *injective* partial maps is described as sets of chains of linear and circular graphs, $PFI = \mathfrak{P}\{\mathfrak{C}\{\mathcal{Z}\} + \mathfrak{S}_{\geq 1}\{\mathcal{Z}\}\}$, so that

$$PFI(z) = \frac{1}{1-z}e^{z/(1-z)}, \qquad PFI_n = \sum_{i=0}^{n} i! \left\{ {n \atop i} \right\}^2$$

(This is a symbolic rewriting of part of the paper [**20**].) ◁

The results of this section and the previous ones offer a wide number of counting results relative to maps satisfying various constraints. These are summarized in Figure 11.

II. 5.3. Labelled graphs. Random graphs form a major chapter of the theory of random discrete structures [**19, 78**]. We examine here enumerative results concerning graphs of low "complexity", that is, graphs which are very nearly trees. (Such graph for instance play an essential rôle in the analysis of early stages of the evolution of a random graph, when edges are successively added, as shown in [**51, 77**].)

The simplest of all connected graphs are certainly the ones that are acyclic. These are trees, but contrary to the case of Cayley trees, no root is specified. Let $\mathcal{U}$ be the class of all *unrooted* trees. Since a rooted tree (rooted trees are, as we know, counted by $T_n = n^{n-1}$) is an unrooted tree combined with a choice of a distinguished node (there are n possible such choices for trees of size n), one has

$$T_n = nU_n \qquad \text{implying} \qquad U_n = n^{n-2}.$$

At generating function level, this combinatorial equality translates into

$$U(z) = \int_0^z T(w)\,\frac{dw}{w},$$

which integrates to give (take T as the independent variable)

$$U(z) = T(z) - \frac{1}{2}T(z)^2.$$

Since $U(z)$ is the EGF of acyclic connected graphs, the quantity

$$A(z) = e^{U(z)} = e^{T(z)-T(z)^2/2},$$

is the EGF of all acyclic graphs. (Equivalently, these are unordered forests of unrooted trees.) Methods developed in Chapters IV and V imply immediately

$$A_n \sim e^{1/2}\, n^{n-2}.$$

Surprisingly, perhaps, there are barely more acyclic graphs than unrooted trees.

The *excess* of a graph is defined as the difference between the number of vertices and the number of nodes. For a connected graph, this is always -1 or more with the minimal value -1 being precisely attained by unrooted trees. The class $\mathcal{W}_k$ is the class of connected graphs of excess equal to k; in particular $\mathcal{U} = \mathcal{W}_{-1}$. The successive classes $\mathcal{W}_{-1}, \mathcal{W}_0, \mathcal{W}_1, \ldots$, may be viewed as describing connected graphs of increasing complexity.

The class $\mathcal{W}_0$ comprises all connected graphs with the number of edges equal to the number of vertices. Equivalently, a graph in $\mathcal{W}_0$ is a connected graph with exactly one cycle (a sort of "eye"), and for that reason, elements of $\mathcal{W}_0$ are sometimes referred to as "unicyclic components" or "unicycles". In a way, such a graph looks very much like an undirected version of a connected functional graph. Precisely, a graph of $\mathcal{W}_0$ consists of a cycle of length at least 3 (by definition, graphs have neither loops nor multiple edges) that is undirected (the orientation present in the usual cycle construction is killed by identifying cycles isomorphic up to reflection) and on which are grafted trees (these are implicitly rooted by the point at which they are attached to the cycle). With $\mathfrak{UC}$ representing the (new) undirected cycle construction, one thus has

$$\mathcal{W}_0 \cong \mathfrak{UC}_{\geq 3}\{\mathcal{T}\}.$$

We claim that this construction is reflected by the EGF equation

$$W_0(z) = \frac{1}{2}\log\frac{1}{1-T(z)} - \frac{1}{2}T(z) - \frac{1}{4}T(z)^2. \tag{46}$$

Indeed one has the isomorphism

$$\mathcal{W}_0 + \mathcal{W}_0 \cong \mathfrak{C}_{\geq 3}\{\mathcal{T}\},$$

since we may regard the two disjoint copies on the left as instantiating two possible orientations of the undirected cycle. The result of (46) then follows from the usual translation of the cycle construction. It is originally due to the Hungarian probabilist Rényi in 1959. Asymptotically, one finds (by methods of Chapter IV):

$$n![z^n]W_0 \sim \frac{1}{4}\sqrt{2\pi}n^{n-1/2} - \frac{5}{3}n^{n-1} + \frac{1}{48}\sqrt{2\pi}n^{n-3/2} + \cdots. \tag{47}$$

Finally, the number of graphs made only of trees and unicyclic components is

$$e^{W_{-1}(z)+W_0(z)} = \frac{e^{T/2-3T^2/4}}{\sqrt{1-T}},$$

and asymptotically,

$$n![z^n]e^{W_{-1}+W_0} = \Gamma(3/4)2^{-1/4}e^{-1/2}\pi^{-1/2}n^{n-1/4}\left(1+O(n^{-1/2})\right).$$

Such graphs stand just next to acyclic graphs in order of structural complexity.

▷ **18.** *2-Regular graphs.* This is based on Comtet's account [**28**, Sec. 7.3]. A 2–regular graph is an undirected graph in which each vertex has degree exactly 2. Connected 2–regular graphs are thus undirected cycles of length $n \geq 3$, so that the EGF of all 2–regular graphs is

$$R(z) = \frac{e^{-z/2-z^2/4}}{\sqrt{1-z}}.$$

Given n straight lines in general position, a cloud is defined to be a set of n intersection points no three being collinear. Clouds and 2–regular graphs are equinumerous. [Hint: Use duality.]

The general enumeration of r–regular graphs becomes somewhat more difficult when $r > 2$. Algebraic aspects are discussed in [**65, 68**] while Bender and Canfield [**9**] have determined the asymptotic formula (for rn even),

$$R_n^{(r)} \sim \sqrt{2}e^{(r^2-1)/4}\frac{r^{r/2}}{e^{r/2}r!}n^{rn/2},$$

for the number of r–regular graphs of size n. ◁

The previous discussion suggests considering more generally the enumeration of connected graphs according to excess. E. M. Wright made important contributions in this area [**154, 155, 156**] that are revisited in the famous "giant paper on the giant component" by Janson, Knuth, Łuczak, and Pittel [**77**]. Wright's result are summarized by the following proposition.

PROPOSITION II.6. *The EGF $W_k(z)$ of connected graphs with excess (of edges over vertices) equal to k is, for $k \geq 1$, of the form*

$$W_k(z) = \frac{P_k(T)}{(1-T)^{3k}}, \quad T \equiv T(z), \tag{48}$$

where P_k is a polynomial of degree $3k+2$. For any fixed k, as $n \to \infty$, one has

$$W_{k,n} = n![z^n]W_k(z) = \frac{P_k(1)\sqrt{2\pi}}{2^{3k/2}\Gamma\left(\frac{3}{2}k\right)}n^{n+(3k-1)/2}\left(1+O(n^{-1/2})\right). \tag{49}$$

The combinatorial part of the proof (not given here, see Wright's original papers or [**77**]) is an interesting exercise in graph surgery and symbolic methods. The analytic part of the statement follows straightforwardly from singularity analysis. The polynomials $P(T)$ and the constants $P_k(1)$ are determined by an explicit nonlinear recurrence; one finds for instance:

$$W_1 = \frac{1}{24}\frac{T^4(6-T)}{(1-T)^3}, \qquad W_2 = \frac{1}{2}\frac{T^4(2+28T-23T^2+9T^3-T^4)}{(1-T)^6}.$$

As explained in the giant paper [**77**], such results combined with complex analytic techniques provide with great detail information on the aspect of a random graph $\Gamma(n,m)$ with n nodes and m edges. In the sparse case where m is of the order of n, one finds the following properties to hold "*with high probability*" (w.h.p)[3], that is, with probability tending to 1 as $n \to \infty$.

- For $m = \mu n$, with $\mu < \frac{1}{2}$, the random graph $\Gamma(m,n)$ has w.h.p. only tree and unicycle components; the largest component is w.h.p. of size $O(\log n)$.
- For $m = \frac{1}{2}n + O(n^{1/3})$, w.h.p. there appear one or several semi-giant components that have size $O(n^{2/3})$.
- For $m = \mu n$, with $\mu > \frac{1}{2}$, there is w.h.p a unique giant component of size proportional to n.

[3]Synonymous expressions are "asymptotically almost surely" (a.a.s) and "in probability". The term "almost surely" is sometimes used, though it lends itself to confusion with continuous measures.

In each case, refined estimates follow from a detailed analysis of corresponding generating functions, which is a main theme of [**51**] and especially [**77**]. Raw forms of these results were first obtained by Erdős and Rényi who launched the subject in a famous series of papers dating from 1959–60; see the books [**19, 78**] for a probabilistic context and the paper [**10**] for the finest counting estimates available. In contrast, the enumeration of *all* connected graphs (irrespective of the number of edges, that is, without excess being taken into account) is a relatively easy problem treated in the next section. Many other classical aspects of the enumerative theory of graphs are covered in the book *Graphical Enumeration* by Harary and Palmer [**76**].

II. 6. Additional constructions

Like in the unlabelled case, pointing and substitution are available in the world of labelled structures (Section II. 6.1). Implicit definitions enlarge the scope of the symbolic method (Section II. 6.2) The inversion process needed to enumerate implicit structures are even simpler, since in the labelled universe sets and cycles have more concise translations as operators over EGF. Finally, and this departs significantly from Chapter I, the fact that integer labels are naturally ordered makes it possible to take into account certain order properties of combinatorial structures (Section II. 6.3).

II. 6.1. Pointing and substitution. The *pointing* of a class $\mathcal{B}$ is defined by

$$\mathcal{A} = \Theta\mathcal{B} \qquad \text{iff} \qquad \mathcal{A}_n = [1\,..\,n] \times \mathcal{B}_n.$$

In other words, in order to generate an element of $\mathcal{A}$, select one of the n labels and point at it. Clearly

$$A_n = n \cdot B_n \implies A(z) = z\frac{d}{dz}A(z).$$

The *composition* or *substitution* can be defined so that it corresponds *a priori* to composition of generating functions. It is formally defined as

$$\mathcal{B} \circ \mathcal{C} = \sum_{k=0}^{\infty} \mathcal{B}_k \times \mathfrak{P}_k\{\mathcal{C}\},$$

so that its EGF is

$$\sum_{k=0}^{\infty} B_k \frac{(C(z))^k}{k!} = B(C(z)).$$

A combinatorial way of realizing this definition and form $\mathcal{B} \circ \mathcal{C}$, is as follows: select some element of $\mathcal{B}$ of some size k, then a k–set of $\mathcal{C}^{\,k}$; the elements of the k–set are naturally ordered by value of their "leader" (the leader of an object being by convention the value of its smallest label); the element with leader of rank r is then substituted to the labelled node of value r in $\mathcal{B}$.

THEOREM II.3. *The combinatorial constructions of pointing and substitution are admissible.*

$$\mathcal{A} = \Theta\mathcal{B} \quad \Longrightarrow \quad A(z) = z\partial_z A(z), \qquad \partial_z \equiv \frac{d}{dz}$$
$$\mathcal{A} = \mathcal{B} \circ \mathcal{C} \quad \Longrightarrow \quad A(z) = B(C(z)).$$

For instance, the EGF of (relabelled) pairings of elements drawn from $\mathcal{A}$ is

$$e^{A(z)+A(z)^2/2},$$

since the EGF of involutions is $e^{z+z^2/2}$.

▷ **19.** *Standard constructions based on substitutions.* The sequence class of $\mathcal{A}$ may be defined by composition as $\mathcal{P} \circ \mathcal{A}$ where $\mathcal{P}$ is the set of all permutations. The powerset class of $\mathcal{A}$ may be defined as $\mathcal{U} \circ \mathcal{A}$ where $\mathcal{U}$ is the class of all urns. Thus,

$$\mathfrak{S}\{\mathcal{A}\} \cong \mathcal{P} \circ \mathcal{A}, \ \mathfrak{P}\{\mathcal{A}\} \cong \mathcal{U} \circ \mathcal{A}.$$

In this way, permutation, urns and circle graphs appear as archetypal classes in a development of combinatorial analysis based on composition.

Joyal's "theory of species" [**79**] and the book by Bergeron, Labelle, and Leroux [**13**] make a great use of such ideas and show that an extensive theory of combinatorial enumeration can be based on such ideas. ◁

▷ **20.** *Distinct component sizes.* The EGF's of permutations with cycles of distinct lengths and of set partitions with parts of distinct sizes are

$$\prod_{n=1}^{\infty}(1+\frac{z^n}{n}), \ \prod_{n=1}^{\infty}(1+\frac{z^n}{n!}).$$

The probability that a permutation of $\mathcal{P}_n$ has distinct cycle sizes tends to $e^{-\gamma}$, see [**72**, Sec. 4.1.6] for a Tauberian argument. ◁

II. 6.2. Implicit structures. Let $\mathcal{X}$ be a labelled class implicitly defined by either of the equations

$$\mathcal{A} = \mathcal{B} + \mathcal{X}, \qquad \mathcal{A} = \mathcal{B} \star \mathcal{X}.$$

Then, solving the corresponding EGF equations leads to

$$X(z) = A(z) - B(z), \qquad X(z) = \frac{A(z)}{B(z)},$$

respectively. For the composite labelled constructions $\mathfrak{S}, \mathfrak{P}, \mathfrak{C}$, the algebra is equally easy.

THEOREM II.4 (Implicit specifications). *The generating functions associated to the implicit equations in* $\mathcal{X}$

$$\mathcal{A} = \mathfrak{S}\{\mathcal{X}\}, \qquad \mathcal{A} = \mathfrak{P}\{\mathcal{X}\}, \qquad \mathcal{A} = \mathfrak{C}\{\mathcal{X}\},$$

are respectively

$$X(z) = 1 - \frac{1}{A(z)}, \qquad X(z) = \log A(z), \qquad X(z) = 1 - e^{-A(z)}.$$

EXAMPLE 14. *Connected graphs.* In the context of graphical enumerations, the labelled set construction takes the form of an enumerative formula relating a class of graphs $\mathcal{G}$ and the subclass of its connected graphs $\mathcal{K} \subset \mathcal{G}$:

$$\mathcal{G} = \mathfrak{P}\{\mathcal{K}\} \implies G(z) = e^{K(z)}.$$

This basic formula is known in graph theory [**76**] as the *exponential formula*.

Consider the class $\mathcal{G}$ of all (undirected) labelled graphs, the size of a graph being the number of its nodes. Since a graph is determined by the choice of its set of edges, there are $\binom{n}{2}$ potential edges each of which may be taken in or out, so that $G_n = 2^{\binom{n}{2}}$. Let $\mathcal{K} \subset \mathcal{G}$ be the subclass of all connected graphs. The exponential formula determines $K(z)$ implicitly,

$$\begin{aligned} K(z) &= \log\left(1 + \sum_{n\geq 1} 2^{\binom{n}{2}}\frac{z^n}{n!}\right) \\ &= z + \frac{z^2}{2!} + 4\frac{z^3}{3!} + 38\frac{z^4}{4!} + 728\frac{z^5}{5!}, \end{aligned}$$

where the sequence is *EIS* **A001187**. The series is divergent, that is, it has radius of convergence 0. It can nonetheless be manipulated as a formal series. Expanding by means of $\log(1+u) = u + u^2/2 + \cdots$, yields a complicated convolution expression for K_n:

$$K_n = 2^{\binom{n}{2}} - \frac{1}{2}\sum \binom{n}{n_1, n_2} 2^{\binom{n_1}{2}+\binom{n_2}{2}} + \frac{1}{3}\sum \binom{n}{n_1, n_2, n_3} 2^{\binom{n_1}{2}+\binom{n_2}{2}+\binom{n_3}{2}} - \cdots .$$

(The kth term is a sum over $n_1 + \cdots + n_k = n$, with $0 < n_j < n$.) Given the very fast increase of G_n with n, for instance

$$2^{\binom{n+1}{2}} = 2^n 2^{\binom{n}{2}},$$

a detailed analysis of the various terms of the expression of K_n shows predominance of the first sum, and, in that sum itself, predominance of the extreme terms corresponding to $n_1 = n-1$ or $n_2 = n-1$, so that

$$K_n = 2^{\binom{n}{2}} \left(1 - 2n2^{-n} + o(2^{-n})\right). \tag{50}$$

Thus, almost all labelled graphs of size n are connected. In addition, the error term decreases very fast: for instance, for $n = 18$, an exact computation based on the generating function formula reveals that a proportion only 0.0001373291074 of all the graphs are not connected—this is *extremely* close to the value 0.0001373291016 predicted by the second term in the asymptotic formula (50). Notice that here good use could be made of a purely divergent generating function for asymptotic enumeration purposes. □

▷ **21.** *Bipartite graphs.* A plane bipartite graph is a pair (G, ω) where G is labelled graph, $\omega = (\omega_W, \omega_E)$ is a bipartition of the nodes (into *West* and *East* categories), and the edges are such that they only connect nodes from ω_W to nodes of ω_E. A direct count shows that the EGF of plane bipartite graphs is

$$\Gamma(z) = \sum_n \gamma_n \frac{z^n}{n!} \text{ with } \gamma_n = \sum_k \binom{n}{k} 2^{k(n-k)}.$$

The EGF of plane bipartite graphs that are connected is $\log \Gamma(z)$.

A bipartite graph is a labelled graph whose nodes can be partitioned into two groups so that edges only connect nodes of different groups. The EGF of bipartite graphs is

$$\exp\left(\frac{1}{2}\log \Gamma(z)\right) = \sqrt{\Gamma(z)}.$$

[Hint. The EGF of a connected bipartite graph is $\frac{1}{2}\log\Gamma(z)$ as a factor of $\frac{1}{2}$ kills the East–West orientation present in a connected plane bipartite graph. See Wilf's book [**153**, p. 78] for details.] ◁

Note. The class of all graphs is not "fully" constructible in the sense that it does not admit a complete construction starting from single atoms and involving only sums, products, sets and cycles. (This assertion can be established rigorously by complex analysis since EGF's of constructible classes must have a nonzero radius of convergence.) In contrast, the special graphs encountered in this chapter, including graphs of fixed excess, are all constructible.

II. 6.3. Order constraints. A construction well suited to taking into account many order properties of combinatorial structures the modified labelled product,

$$\mathcal{A} = (\mathcal{B}^{\square} \star \mathcal{C}).$$

This denotes the subset of the product $\mathcal{B} \star \mathcal{C}$ formed with elements such that the smallest label is constrained to lie in the $\mathcal{B}$ component. (To make this definition consistent, it must be assumed that $B_0 = 0$.) We call this binary operation on structures the *boxed* product.

THEOREM II.5. *The boxed product is admissible.*

$$\mathcal{A} = (\mathcal{B}^{\square} \star \mathcal{C}) \quad \Longrightarrow \quad A(z) = \int_0^z (\partial_t B(t)) \cdot C(t)\, dt, \qquad \partial_t \equiv \frac{d}{dt}. \tag{51}$$

PROOF. The definition of boxed products implies the coefficient relation

$$A_n = \sum_{k=1}^{n} \binom{n-1}{k-1} B_k C_{n-k}.$$

The binomial coefficient that appears in the standard labelled product is now modified since only $n-1$ labels need to be distributed between the two components, $k-1$ going to the $\mathcal{B}$ component (that is constrained to contain the label 1 already) and $n-k$ to the $\mathcal{C}$ component. From the equivalent form

$$\frac{A_n}{n!} = \frac{1}{n} \sum_{k=0}^{n} \binom{n}{k} (kB_k)\, C_{n-k},$$

the result follows by taking EGF's. □

A useful special case is the min–rooting operation,

$$\mathcal{A} = \{1\}^{\square} \star \mathcal{C},$$

for which a variant definition goes as follows. Take in all possible ways elements $\gamma \in \mathcal{C}$, prepend an atom with a label smaller than the labels of γ, for instance 0, and relabel in the canonical way over $[1\,..\,(n+1)]$ by shifting label values. Clearly $A_{n+1} = C_n$ which yields

$$A(z) = \int_0^z C(t)\, dt,$$

a result also consistent with the general formula of boxed products.

For some applications, it is easier to impose constraints on the *maximal* label rather than the minimum. The max-boxed product written

$$\mathcal{A} = (\mathcal{B}^{\blacksquare} \star \mathcal{C}),$$

is then defined by the fact the maximum is constrained to lie in the $\mathcal{B}$-component of the labelled product. Naturally, the translation by an integral in (51) remains valid for this trivially modified boxed product.

▷ **22.** *Combinatorics of integration.* In the perspective of this book, integration by parts has an immediate interpretation. Indeed, the equality,

$$\int_0^z A'(t) \cdot B(t)\, dt = A(z) \cdot B(z) - \int_0^z A(t) \cdot B'(t)\, dt,$$

reads off as: *"The smallest label in an ordered pair, if it appears on the left, cannot appear on the right."* ◁

EXAMPLE 15. *Records in permutations.* Given a sequence of numerical data, $x = (x_1, \ldots, x_n)$ assumed all distinct, a *record* in that sequence is defined to be an element x_j such that $x_k < x_j$ for all $k < j$. (A record is an element "better" than its predecessors!) Figure 12 displays a numerical sequence of length $n = 100$ that has 7 records. Confronted to such data, a statistician will typically want to determine whether the data obey purely random fluctuations or there could be some indications of a "trend" or of a "bias" [**33**, Ch. 10]. (Think of the data as reflecting share prices or athletic records, say.) In particular, if the x_j are independently drawn from a continuous distribution, then the number

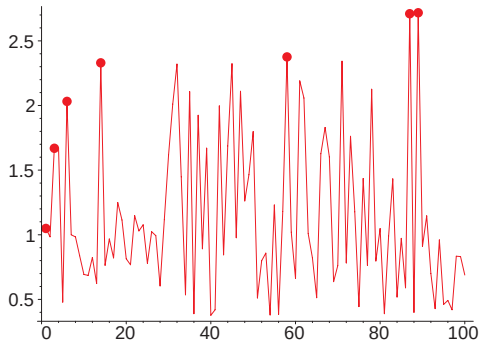


FIGURE 12. A numerical sequence of size 100 with records marked by circles: there are 7 records that occur at times $1, 3, 5, 11, 60, 86, 88$.

of records obeys the same laws as in a random permutation of $[1\,..\,n]$. This statistical preamble then invites the question: *How many permutations of n have k records?*

First, we start with a special brand of permutations, the ones that have their *maximum* at the beginning. Such permutations are defined as ('■' indicates the boxed product based on the maximum label)

$$\mathcal{Q} = (Z^{\blacksquare} \star \mathcal{P}),$$

where $\mathcal{P}$ is the class of all permutations. Observe that this gives the EGF

$$Q(z) = \int_0^z \left(\frac{d}{dt}t\right) \cdot \frac{1}{1-t}\, dt = \log \frac{1}{1-z},$$

implying the obvious result $Q_n = (n-1)!$ for all $n \geq 1$. These are exactly the permutations with *one* record. Next, consider the class

$$\mathcal{P}^{(k)} = \mathfrak{P}_k\{\mathcal{Q}\}.$$

The elements of $\mathcal{P}^{(k)}$ are unordered sets of cardinality k with elements of type $\mathcal{Q}$. Define the (max) leader of any component of $\mathcal{P}^{(k)}$ as the value of its maximal element. Then, if we place the components in sequence, ordered by increasing values of their leaders, then read off the whole sequence, we obtain a permutation with k records exactly. The correspondence[4] is easily revertible. Here is an illustration, with leaders underlined:

$$\begin{aligned}\{(\underline{\mathbf{7}}, 2, 6, 1),\ (\underline{\mathbf{4}}, 3),\ (\underline{\mathbf{9}}, 8, 5)\} &\cong [(\underline{\mathbf{4}}, 3),\ (\underline{\mathbf{7}}, 2, 6, 1),\ (\underline{\mathbf{9}}, 8, 5))] \\ &\cong \underline{\mathbf{4}}, 3, \underline{\mathbf{7}}, 2, 6, 1, \underline{\mathbf{9}}, 8, 5.\end{aligned}$$

Thus, the number of permutations with k records is determined by

$$P^{(k)}(z) = \frac{1}{k!}\left(\log \frac{1}{1-z}\right)^k, \qquad P_n^{(k)} = \begin{bmatrix} n \\ k \end{bmatrix},$$

where we recognize Stirling cycle numbers from Example 11. In other words:

> *The number of permutations of size n having k records is counted by the Stirling "cycle" number $\left[{n \atop k}\right]$.*

[4]This correspondence is easily extended to a transformation on permutations that maps the number of records to the number of cycles.In this case, it is known as Foata's fundamental correspondence [**98**, Sec. 10.2].

Returning to our statistical problem, the treatment of Example 84 (to be revisited in Chapter III) shows that the expected number of records in a random permutation of size n equals H_n, the harmonic number. One has $\mathrm{H}_{100} \doteq 5.18$, so that for 100 data items, a little more than 5 records are expected on average. The probability of observing 7 records or more is still about 23%, an altogether not especially rare event. In contrast, observing twice as many records, that is, 14, would be a fairly strong indication of a bias since, on random data, the event has probability very close to 10^{-4}. Altogether, the present discussion is consistent with the hypothesis for the data of Figure 12 to have been generated independently at random (and indeed they were). □

It is possible to base a fair part of the theory of labelled constructions on sums and products in conjunction with the boxed product. In effect, consider the three relations

$$\begin{aligned}
\mathcal{F} = \mathfrak{S}\{\mathcal{G}\} \quad &\Longrightarrow \quad f(z) = \frac{1}{1-g(z)}, \qquad f = 1 + gf \\
\mathcal{F} = \mathfrak{P}\{\mathcal{G}\} \quad &\Longrightarrow \quad f(z) = e^{g(z)}, \qquad f = \int g' f \\
\mathcal{F} = \mathfrak{C}\{\mathcal{G}\} \quad &\Longrightarrow \quad f(z) = \log \frac{1}{1-g(z)}, \quad f = \int g' \frac{1}{1-g}
\end{aligned}$$

The last column is easily checked to provide an alternative form of the standard operator corresponding to sequences, powersets, and cycles. Each case is then itself deduced directly from Theorem II.5 and the labelled product rule:

Sequences: they obey the recursive definition

$$\mathcal{F} = \mathfrak{S}\{\mathcal{G}\} \qquad \Longrightarrow \qquad \mathcal{F} \cong \{\epsilon\} + (\mathcal{G} \star \mathcal{F}).$$

Sets: we have

$$\mathcal{F} = \mathfrak{P}\{\mathcal{G}\} \qquad \Longrightarrow \qquad \mathcal{F} \cong \{\epsilon\} + (\mathcal{G}^{\blacksquare} \star \mathcal{F}),$$

which means that, in a set, one can always single out the component with the largest label, the rest of the components forming a set. In other words, when this construction is repeated, the elements of a set can be canonically arranged according to increasing values of their largest labels, the "leaders". (We recognize here a generalization of the construction used for records in permutations.)

Cycles: The element of a cycle that contains the largest label can be taken canonically as the cycle "starter", which is then followed by an arbitrary sequence of elements upon traversing the cycle in circular order. Thus

$$\mathcal{F} = \mathfrak{C}\{\mathcal{G}\} \qquad \Longrightarrow \qquad \mathcal{F} \cong (\mathcal{G}^{\blacksquare} \times \mathfrak{S}\{\mathcal{G}\}).$$

Greene [**73**] has developed a complete framework of labelled grammars based on standard and boxed labelled products. In its basic form, its expressive power is essentially equivalent to ours, because of the above relations. More complicated order constraints, dealing simultaneously with a collection of larger and smaller elements, can be furthermore taken into account within this framework.

▷ **23.** *Higher order constraints.* Let the symbols $\square$, $\boxdot$, $\blacksquare$ represent smallest, second smallest, and largest labels respectively. One has the correspondences (with $\partial_z = \frac{d}{dz}$)

$$\begin{aligned}
\mathcal{A} &= \left(\mathcal{B}^{\square} \star \mathcal{C}^{\blacksquare}\right) && \partial_z^2 A(z) = (\partial_z B(z)) \cdot (\partial_z C(z)) \\
\mathcal{A} &= \left(\mathcal{B}^{\square\,\blacksquare} \star \mathcal{C}\right) && \partial_z^2 A(z) = \left(\partial_z^2 B(z)\right) \cdot C(z) \\
\mathcal{A} &= \left(\mathcal{B}^{\square} \star \mathcal{C}^{\boxdot} \star \mathcal{D}^{\blacksquare}\right) && \partial_z^3 A(z) = (\partial_z B(z)) \cdot (\partial_z C(z)) \cdot (\partial_z D(z)),
\end{aligned}$$

and so on. These can be transformed into (iterated) integral representations. [See [**73**] for more.] ◁

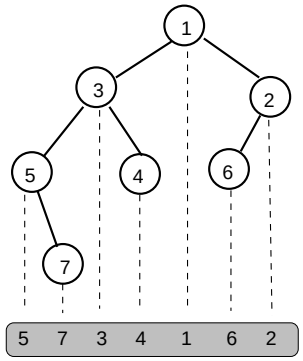


FIGURE 13. A permutation of size 7 and its increasing binary tree lifting.

The next two examples demonstrate the usefulness of min-rooting used in conjunction with recursion. In this way, trees satisfying some order conditions can be constructed and enumerated easily. This is turn gives access to new characteristics of permutations.

EXAMPLE 16. *Increasing binary trees and alternating permutations.* To each permutation, one can associate bijectively a binary tree of a special type[5] called an *increasing binary tree* and sometimes a heap–ordered tree or a tournament tree. This is a plane rooted binary tree in which internal nodes bear labels in the usual way, but with the additional constraint that node labels increase along any branch stemming from the root.

The correspondence (Figure 13) is as follows: Given a permutation of a set written as a word, $\sigma = \sigma_1\sigma_2\dots\sigma_n$, factor it in the form $\sigma = \sigma_L \cdot \min(\sigma) \cdot \sigma_R$, with $\min(\sigma)$ the smallest label value in the permutation, and σ_L, σ_R the factors left and right of $\min(\sigma)$. Then the binary tree $\beta(\sigma)$ is defined recursively in the format ⟨root, left,right⟩ by

$$\beta(\sigma) = \langle \min(\sigma), \beta(\sigma_L), \beta(\sigma_R)\rangle, \qquad \beta(\epsilon) = \epsilon.$$

The empty tree (consisting of a unique external node of size 0) goes with the empty permutation ϵ. Conversely, reading the labels of the tree in symmetric (infix) order gives back the original permutation. (The correspondence is described for instance in Stanley's book [**135**, p. 23–25] who says that "it has been primarily developed by the French", pointing at [**64**].)

Thus, the family $\mathcal{I}$ of binary increasing trees satisfies the recursive definition

$$\mathcal{I} = \{\epsilon\} + \left(\mathcal{Z}^{\square} \star \mathcal{I} \star \mathcal{I}\right),$$

which implies the nonlinear integral equation for the EGF

$$I(z) = 1 + \int_0^z I(t)^2\, dt.$$

This equation reduces to $I'(z) = I(z)^2$ and, under the initial condition $I(0) = 1$, it admits the solution $I(z) = (1-z)^{-1}$. Thus $I_n = n!$, which is consistent with the fact that there are as many increasing trees as there are permutations.

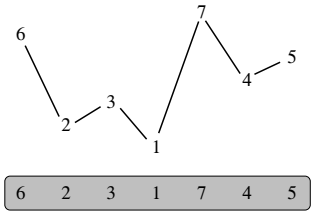
[5] Such trees are closely related to classical data structures of computer science, like heaps and binomial queues [**30, 129**].

The construction of increasing trees associated with permutation is instrumental in deriving EGF's relative to various local order patterns in permutations, like the number of ascents and descents, rises, falls, peaks and troughs, etc. We illustrate its use here by counting the number of *up-and-down* (or *zig-zag*) permutations, also known as *alternating* permutations. The result was first derived by Désiré André in 1881 by means of a direct recurrence argument.

A permutation $\sigma = \sigma_1\sigma_2\ldots\sigma_n$ is an alternating permutation if

$$\sigma_1 > \sigma_2 < \sigma_3 > \sigma_4 < \cdots, \tag{52}$$

so that pairs of consecutive elements form a succession of ups and downs; for instance,



Consider first the case of an alternating permutation of *odd* size. It can be checked that the corresponding increasing trees have no one–way branching nodes, so that they consist solely of binary nodes and leaves. Thus, the corresponding specification is

$$\mathcal{J} = \mathcal{Z} + \left(\mathcal{Z}^{\square} \star \mathcal{J} \star \mathcal{J}\right),$$

so that

$$J(z) = z + \int_0^z J(t)^2\, dt \qquad \text{and} \qquad \frac{d}{dz}J(z) = 1 + J(z)^2.$$

The equation admits separation of variables, which implies (with $J(0) = 0$)

$$J(z) = \tan(z) = z + 2\frac{z^3}{3!} + 16\frac{z^5}{5!} + 272\frac{z^7}{7!} + \cdots.$$

The coefficients J_{2n+1} are known as the *tangent numbers* or the *Euler numbers* of odd index (*EIS* **A000182**).

Alternating permutations of *even* size defined by the constraint (52) and denoted by $\overline{\mathcal{J}}$ can be determined from

$$\overline{\mathcal{J}} = \{\epsilon\} + \left(\mathcal{Z}^{\square} \star \mathcal{J} \star \overline{\mathcal{J}}\right),$$

since now all internal nodes of the tree representation are binary, except for the rightmost one that only branches on the left. Thus, $J'(z) = \tan(z)J(z)$, and the EGF is

$$\overline{J}(z) = \frac{1}{\cos(z)} = 1 + 1\frac{z^2}{2!} + 5\frac{z^4}{4!} + 61\frac{z^6}{6!} + 1385\frac{z^8}{8!} + \cdots,$$

where the coefficients $\overline{J}_{2n}$ are the *secant numbers* also known as Euler numbers of even index (*EIS* **A000364**). □

Use will be made later in this book (Chapter III, p. 17) of this important tree representation of permutations as it opens access to parameters like the number of descents, runs, and (once more!) records in permutations. Analyses of increasing trees also inform us of crucial performance issues regarding binary search trees, quicksort, and heap–like priority queue structures [**102, 130, 147, 148**].

▷ **24.** *Combinatorics of trigonometrics.* Interpret $\tan\frac{z}{1-z}$, $\tan\tan z$, $\tan(e^z - 1)$ as EGFs. ◁

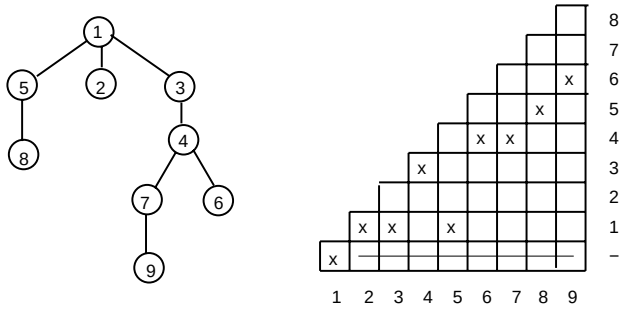


FIGURE 14. An increasing Cayley tree (left) and its associated regressive mapping (right).

EXAMPLE 17. *Increasing Cayley trees and regressive mappings.* An increasing Cayley tree is a Cayley tree (i.e., it is nonplane and rooted) whose labels along any branch stemming from the root form an increasing sequence. In particular, the minimum must occur at the root, and no plane embedding is implied. Let $\mathcal{K}$ be the class of such trees. The recursive specification is now

$$\mathcal{K} = \left(\mathcal{Z}^{\square} \star \mathfrak{P}\{\mathcal{K}\} \right).$$

The generating function thus satisfies the functional relations

$$K(z) = \int_0^z e^{K(t)}\, dt, \qquad K'(z) = e^{K(z)},$$

with $K'(0) = 0$. Integration of $K' e^{-K} = 1$ shows that

$$K(z) = \log \frac{1}{1-z} \qquad \text{and} \qquad K_n = (n-1)!.$$

Thus the number of increasing Cayley trees is $(n-1)!$, which is also the number of permutations of size $n-1$. These trees have been studied by Meir and Moon [**104**] under the name of "recursive trees", a terminology that we do not however retain here.

The simplicity of the formula $K_n = (n-1)!$ certainly calls for a combinatorial interpretation. In fact, an increasing Cayley tree is fully determined by its child parent relationship (Figure 14). Otherwise said, to each increasing Cayley tree τ, we associate a partial map $\phi = \phi_\tau$ such that $\phi(i) = j$ iff the label of the parent of i is j. Since the root of tree is an orphan, the value of $\phi(1)$ is undefined, $\phi(1) = \perp$; since the tree is increasing, one has $\phi(i) < i$ for all $i \geq 2$. A function satisfying these last two conditions is called a *regressive mapping*. The correspondence between trees and regressive mappings is then easily seen to be a bijective one.

Thus regressive mappings on the domain $[1\,..\,n]$ and increasing Cayley trees are equinumerous, so that we may as well use $\mathcal{K}$ to denote the class of regressive mappings. Now, a regressive mapping of size n is evidently determined by a single choice for $\phi(2)$ (since $\phi(2) = 1$), two possible choices for $\phi(3)$ (either of $1, 2$), and so on. Hence the fact that

$$K_n = 1 \cdot 2 \cdot 3 \cdots (n-1)$$

receives a natural interpretation. □

Regressive mappings can be also related directly to permutations. The construction that associates a regressive mapping to a permutation is called the "inversion table" construction; see [**86, 130**]. In short, given a permutation $\sigma = \sigma_1, \ldots, \sigma_n$, one can associate to it a function $\psi = \psi_\sigma$ from $[1\,..\,n]$ to $[0\,..\,n-1]$, by the rule

$$\psi(j) = \operatorname{card}\left\{k < j \;\middle|\; \sigma_k > \sigma_j\right\}.$$

(The function ψ is a trivial variant of a regressive mapping.) Summarizing, we have a double combinatorial connection,

$$\text{Increasing Cayley tree} \cong \text{Regressive mappings} \cong \text{Permutations},$$

that opens the way to yet more permutation enumerations.

▷ **25.** *Rotations and increasing trees.* An increasing Cayley tree can be canonically drawn by ordering descendants of each node from left to right according to their label values. The rotation correspondence (p. 48) then gives rise to a binary increasing tree. Hence, increasing Cayley trees and increasing binary trees are also directly related. ◁

II. 7. Notes

Labelled constructions are a frequently used paradigm of combinatorial analysis with applications to order statistics and graphical enumerations for instance. See the books by Comtet [**28**], Wilf [**153**], Stanley [**135**], or Goulden and Jackson [**68**] for many examples.

The labelled set construction and the exponential formula were recognized early by researchers working in the area of graphical enumerations [**76**]. Foata [**62**] proposed a detailed formalization in 1974 of labelled constructions, especially sequences and sets, under the names of partitional complex; a brief account is also given by Stanley in his survey [**134**]. This is parallel to the concept of "prefab" due to Bender and Goldman [**11**].

Greene developed a general framework of "labelled grammars" largely based on the boxed product with implications for the random generation of combinatorial structures. Joyal's theory of species [**79**], already mentioned in the previous chapter, is based on category theory; it presents the advantage of uniting in a common theory the unlabelled and the labelled worlds.

Flajolet, Salvy, and Zimmermann have developed a specification language closely related to the system exposed here. They show in [**56**] how to compile automatically specifications into generating functions; this is complemented by a calculus that produces fast random generation algorithms [**61**].

CHAPTER III

Combinatorial Parameters and Multivariate Generating Functions

Generating functions find averages, etc.
— HERBERT WILF [**153**]

Je n'ai jamais été assez loin pour bien sentir l'application de l'algèbre à la géométrie. Je n'aimais point cette manière d'opérer sans voir ce qu'on fait, et il me sembloit que résoudre un problème de géométrie par les équations, c'étoit jouer un air en tournant une manivelle.
— JEAN-JACQUES ROUSSEAU, *Les Confessions*, Livre VI

Many scientific endeavours, in probability theory and statistics, computer science and analysis of algorithms, statistical physics and computational biology demand precise quantitative informations on probabilistic properties of *parameters* of combinatorial objects. For the purpose of designing, analysing, and optimizing a sorting algorithm, it is for instance of interest to determine what the typical disorder of data obeying a given model of randomness is, and do so in the mean, or even in distribution, either exactly or asymptotically. The "exact" problem is then a refined counting problem with two parameters, size and additional characteristic; the "asymptotic" problem can be viewed as one of characterizing in the limit a family of probability laws indexed by the values of the possible sizes. As demonstrated in this chapter, the symbolic methods initially developed for counting combinatorial objects adapt gracefully to the analysis of various sorts of parameters of constructible classes, unlabelled and labelled alike.

Multivariate generating functions—ordinary or exponential—can keep track of the number of components in a composite construction, like a sequence, a (multi)set, or a cycle. Generally, multivariate generating functions give access to "inherited" parameters defined inductively over combinatorial objects. This includes the number of occurrences of designated "patterns" to be found in an object of a given size. From such generating functions, there result either explicit probability distributions or, at least, mean and variance evaluations. Essentially all the combinatorial classes discussed in the first two chapters are amenable to such a treatment. Typical applications are the number of summands in a composition, the number of blocks in a set partition, the number of cycles in a permutation, the root degree or path length of a tree, the number of fixed point in a permutation, the number of singleton blocks in a set partition, the number of leaves in trees of various sorts, and so on. Technically, the translation schemes that relate combinatorial constructions and multivariate generating functions present no major difficulty, since they appear to be natural (notational, even) refinements of the paradigm developed in Chapters I and II for the univariate case.

Beyond its technical aspects anchored in "symbolic combinatorics", this chapter also serves as a first encounter with the general area of "random combinatorics". The question is: *What does a random object of large size look like?* Multivariate generating functions

when combined with probabilistic inequalities often offer definitive answers. For instance, a large integer partition conforms with high probability to a deterministic profile, a large random permutation almost surely has at least one long cycle and a few short ones, and so on. Such a highly constrained behaviour of large objects may in turn serve to design, dedicated algorithms and optimize data structures; or it may serve to build statistical tests (when does one depart from randomness and detect a "signal" in large sets of observed data?). Randomness aspects form a recurrent theme of the book: they will be developed even further in Chapters IV–VII, after complex-asymptotic methods have been grafted on exact modelling by generating functions.

This chapter is organized as follows. Section III. 1 first introduces the basic notions of multivariate enumeration and multivariate generating function. There, we shall also discuss the relations with discrete probabilistic models, as the language of elementary probability theory does provide an intuitively appealing way to conceive of multivariate counting data. The symbolic method *per se* declined in its multivariate version is centrally developed in Sections III. 2 and III. 3: with suitable multi-index notations, the extension to the multivariate case is almost immediate. Recursive parameters that often arise from tree statistics form the subject of Section III. 4, while "universal" generating functions and combinatorial models are discussed in Section III. 5. Additional constructions like pointing, substitution, and order constraints lead to interesting developments, in particular, an original treatment of the inclusion-exclusion principle in Section III. 6. The chapter concludes with Section III. 7 that presents a brief abstract discussion of extremal parameters like height in trees or smallest and largest components in composite structures, which leads to families of univariate generating functions.

III. 1. Parameters, generating functions, and distributions

Our purpose here is to analyse various characteristics of combinatorial structures. Most of the time, we shall be interested in enumeration according to size *and* a single auxiliary parameter. However, the theory is best developed in full generality for the joint analysis of a *finite collection* of parameters.

DEFINITION III.1. *Consider a combinatorial class $\mathcal{A}$. A* parameter $\chi = (\chi_1, \dots, \chi_d)$ *on the class is a function from $\mathcal{A}$ to the set $\mathbb{N}^d$ of d-tuples of natural numbers. The* counting sequence *of $\mathcal{A}$ with respect to size and the parameter χ is then defined by*

$$A_{n,k_1,\dots,k_d} = \operatorname{card}\left\{\alpha \;\middle|\; |\alpha| = n,\ \chi_1(\alpha) = k_1, \dots, \chi_d(\alpha) = k_d\right\}.$$

We sometimes refer to such a parameter as a "multiparameter" (in particular when $d > 1$), as a "simple" or "scalar" parameter otherwise. One may take for $\mathcal{A}$ the class $\mathcal{P}$ of all permutations, and for $\chi \equiv \chi_1$ the parameter that associates to a permutation the number of its cycles. Natural questions are then: How many permutations of size n have k cycles? What is the expected number of cycles in a random permutation? Does this parameter have a distribution that can be made explicit? What are the features of this distribution in terms of "shape", "concentration", or limiting asymptotic behaviour. See Figure 1 for a first example related to binary words and Figure 2 for histograms relative to words and to cycles in permutations.

III. 1.1. Multivariate generating functions. Not too unexpectedly, the treatment of parameters in this book will be in terms of generating functions. The multi-index convention employed in various branches of mathematics greatly simplifies notations and is as follows: let $\mathbf{u} = (u_1, \dots, u_d)$ be a vector of d formal variables and $\mathbf{k} = (k_1, \dots, k_d)$

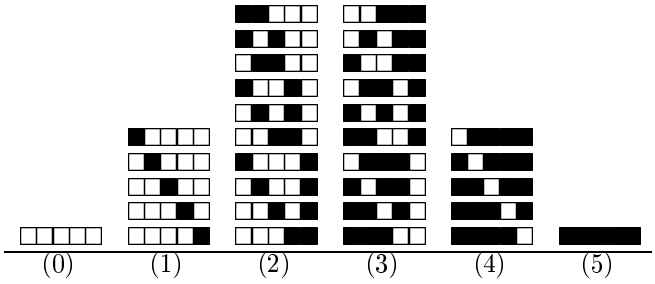


FIGURE 1. The set $\mathcal{W}_5$ of the 32 binary words over the alphabet $\{\square, \blacksquare\}$ enumerated according to the number of occurrences of the letter '$\blacksquare$' gives rise to the bivariate counting sequence $\{W_{5,j}\} = 1, 5, 10, 10, 5, 1$.

be a vector of integers of the same dimension; then, the multi-power $\mathbf{u}^{\mathbf{k}}$ is defined as the monomial

$$\mathbf{u}^{\mathbf{k}} := u_1^{k_1} u_2^{k_2} \cdots u_d^{k_d}. \tag{1}$$

With this notation, we have:

DEFINITION III.2. *Let $A_{n,\mathbf{k}}$ be a multi-index sequence of numbers, where $\mathbf{k} \in \mathbb{N}^d$. The* multivariate generating function (MGF) *of the sequence of either ordinary or exponential type is defined by*

$$\begin{aligned} A(z, \mathbf{u}) &= \sum_{n,k} A_{n,\mathbf{k}} \mathbf{u}^{\mathbf{k}} z^n \quad \textit{(ordinary MGF)} \\ A(z, \mathbf{u}) &= \sum_{n,k} A_{n,\mathbf{k}} \mathbf{u}^{\mathbf{k}} \frac{z^n}{n!} \quad \textit{(exponential MGF)}, \end{aligned} \tag{2}$$

where the multi-index convention is in force.

Given a class $\mathcal{A}$ and a parameter χ, the multivariate generating function (MGF) of the pair $\langle \mathcal{A}, \chi \rangle$ is the MGF of the corresponding counting sequence. In particular, one has the combinatorial forms

$$\begin{aligned} A(z, \mathbf{u}) &= \sum_{\alpha \in \mathcal{A}} \mathbf{u}^{\chi(\alpha)} z^{|\alpha|} \quad \textit{(ordinary MGF; unlabelled case)} \\ A(z, \mathbf{u}) &= \sum_{\alpha \in \mathcal{A}} \mathbf{u}^{\chi(\alpha)} \frac{z^{|\alpha|}}{|\alpha|!} \quad \textit{(exponential MGF; labelled case)}. \end{aligned} \tag{3}$$

One also says that $A(z, \mathbf{u})$ is the MGF of the combinatorial class with the formal variable u_j marking *the parameter χ_j and z* marking *size.*

From the very definition, $A(z, \mathbf{1})$ (with $\mathbf{1}$ a vector of all 1's) coincides with the counting generating function of $\mathcal{A}$, either ordinary or exponential as the case may be. One can then view an MGF as a "deformation" of a univariate GF by way of a parameter u, with the property for the multivariate GF to reduce to the univariate counting GF at $u = 1$.

In the case of a single parameter, these formulæ give rise to a *bivariate generating function*, also abbreviated as BGF. As already pointed out, this is the most frequently encountered situation in this book. In the many cases where the univariate versus multivariate distinction does not need to be stressed, we shall allow ourselves to use common (italic) letters to represent both scalars and vectors (so that $\mathbf{u} \mapsto u$ and $\mathbf{k} \mapsto k$): in such cases,

the multi-index convention is automatically understood as soon as $d > 1$. (In this way, generating functions can be written with the less intrusive notation $A(z, u)$.)

The counting of $\mathcal{A}$–structures according to size and values of the scalar parameter χ is entirely encoded into a bivariate generating function. In order to put the OGF and EGF cases under a common umbrella, set

$$\omega_n = 1 \text{ (OGF, unlabelled case)}, \qquad \omega_n = n! \text{ (EGF, labelled case)}.$$

One may then arrange the BGF either in powers of z or in powers of u:

$$\begin{aligned} A(z,u) &= \sum_n A_n(u)\frac{z^n}{\omega_n} \\ &= \sum_k A^{\langle k\rangle}(z)u^k. \end{aligned}$$

If one views the table of coefficients as a 2-dimensional table, the $A_n(u)$ describe the behaviour of χ over all objects of some fixed size n—these are sometimes called the "horizontal" GF's associated to the BGF; the $A^{\langle k\rangle}(z)$, also called "vertical" generating functions, count the objects in $\mathcal{A}$ associated to fixed values of the parameter χ. Here is a diagram that displays the GFs stemming from a single BGF $A(z, u)$ and justifies this "horizontal–vertical" terminology.

vertical GF's

$A^{\langle 0\rangle}(z)$ ↓	$A^{\langle 1\rangle}(z)$ ↓	$A^{\langle 2\rangle}(z)$ ↓		
$+A_{0,0}u^0z^0$	$+A_{0,1}u^1z^0$	$+A_{0,2}u^2z^0$	$\cdots\ \cdots$	$\longleftarrow A_0(u)$
$+A_{1,0}u^0z^1$	$+A_{1,1}u^1z^1$	$+A_{1,2}u^2z^1$	$\cdots\ \cdots$	$\longleftarrow A_1(u)$
$+A_{2,0}u^0z^2$	$+A_{2,1}u^1z^2$	$+A_{2,2}u^2z^2$	$\cdots\ \cdots$	$\longleftarrow A_2(u)$
$+A_{3,0}u^0z^3$	$+A_{3,1}u^1z^3$	$+A_{3,2}u^2z^3$	$\cdots\ \cdots$	$\longleftarrow A_3(u)$
$\vdots$	$\vdots$	$\vdots$		

} horizontal GFs;

see also [**130**]. (Technically, we are taking advantage of the isomorphism between formal power series: $\mathbb{C}[\![z, u]\!] \cong \mathbb{C}[\![u]\!][\![z]\!] \cong \mathbb{C}[\![z]\!][\![u]\!]$.) Accordingly, the coefficients $A_{n,k}$ are recovered by applying the coefficient operator repeatedly in any convenient order. For instance, for a simple parameter

$$A_{n,k} = \omega_n \cdot [u^k z^n]A(z,u) \equiv \omega_n \cdot [z^n]\Big([u^k]A(z,u)\Big) \equiv \omega_n \cdot [u^k]\Big([z^n]A(z,u)\Big),$$

As a first illustration, consider the binomial coefficient $\binom{n}{k}$, already discussed from a univariate point of view in Chapter I, as it counts the binary words of length n having k occurrences of a designated letter; see Figure 1. In order to compose the bivariate GF, start from the simplest case of Newton's binomial theorem and form directly the horizontal GFs:

$$W_n(u) := \sum_{k=0}^{n} \binom{n}{k} u^k = (1+u)^n,$$

Then a summation over all values of n gives the ordinary BGF

$$W(z,u) = \sum_{k,n\geq 0} \binom{n}{k} u^k z^n = \sum_{n\geq 0} (1+u)^n z^n = \frac{1}{1-z(1+u)}. \tag{4}$$

(There, the second equality results from a computation in $\mathbb{C}[\![u]\!][\![z]\!]$.) The vertical OGFs of the binomial coefficients are

$$W^{\langle k\rangle}(z) = \sum_{n\geq 0} \binom{n}{k} z^n = \frac{z^k}{(1-z)^{k+1}},$$

as results from a direct calculation based on Newton's binomial theorem with negative exponents, or via an expansion of the BGF with respect to u:

$$W(z,u) = \frac{1}{1-z}\frac{1}{1-u\frac{z}{1-z}} = \sum_{k\geq 0} u^k \frac{z^k}{(1-z)^{k+1}}.$$

Such calculations are typical of MGF manipulations. Observe that (4) reduces to the OGF $(1-2z)^{-1}$ of binary words, as it should, upon setting $u=1$.

▷ **1.** *Exponential GFs of binomial coefficients.* The exponential BGF of binomial coefficients is

$$\widetilde{W}(z,u) = \sum_{k,n} \binom{n}{k} u^k \frac{z^n}{n!} = \sum (1+u)^n \frac{z^n}{n!} = e^{z(1+u)}. \tag{5}$$

The vertical EGFs are $e^z z^k/k!$. The horizontal GFs are $(1+u)^n$, like in the ordinary case. ◁

As a second illustration, we saw in Chapter II (Example 11) that the number of permutations of size n having k cycles is the Stirling cycle number $\left[{n \atop k}\right]$. The EGF is, for fixed k, given by

$$P^{\langle k\rangle}(z) := \sum_n \left[{n \atop k}\right] \frac{z^n}{n!} = \frac{L(z)^k}{k!}, \qquad L(z) := \log\frac{1}{1-z}.$$

The starting point is thus a collection of vertical EGFs. From there, the exponential BGF is easily formed as follows:

$$\begin{aligned} P(z,u) &:= \sum_k P^{\langle k\rangle}(z) u^k \\ &= \sum_k \frac{u^k}{k!} L(z)^k \quad = \quad e^{uL(z)} \\ &= (1-z)^{-u}. \end{aligned} \tag{6}$$

The simplification is quite remarkable but altogether quite typical, as we shall see shortly, in the context of a labelled set construction.

An expansion of the BGF according to the variable z further gives by virtue of Newton's binomial theorem:

$$\begin{aligned} P(z,u) &= \sum_{n\geq 0} \binom{n+u-1}{n} z^n \\ P_n(u) &= u(u+1)\cdots(u+n-1) \equiv \sum_k \left[{n \atop k}\right] u^k. \end{aligned}$$

This last polynomial is a horizontal GF called the *Stirling cycle polynomial* of index n and it describes completely the distribution of the number of cycles in all permutations of size n. In passing, note that the relation

$$P_n(u) = P_{n-1}(u)(u+(n-1)),$$

is equivalent to a recurrence

$$\left[{n \atop k}\right] = (n-1)\left[{n-1 \atop k}\right] + \left[{n-1 \atop k-1}\right],$$

by which Stirling numbers are often defined and easily evaluated numerically; see also APPENDIX: *Stirling numbers*, p. 173. (The recurrence is otherwise susceptible to a direct combinatorial interpretation—add n either to an existing cycle or as a "new" singleton.)

▷ **2.** *Specializations of MGFs.* The exponential MGF of permutations with u_1, u_2 marking the number of 1-cycles and 2-cycles respectively turns out to be

$$P(z, u_1, u_2) = \frac{\exp\left((u_1 - 1)z + (u_2 - 1)\frac{z^2}{2}\right)}{1 - z}. \tag{7}$$

(This is to be proved later in this chapter, p. 137.) The formula is checked to be consistent with three already known specializations derived in Chapter II: (i) setting $u_1 = u_2 = 1$ gives back the counting off *all* permutations, $P(z, 1, 1) = (1-z)^{-1}$, as it should; (ii) setting $u_1 = 0$ and $u_2 = 1$ gives back the EGF of derangements, namely $e^{-z}/(1 - z)$; (iii) setting $u_1 = u_2 = 0$ gives back the EGF of permutations with cycles all of length greater than 2, $P(z, 0, 0) = e^{z+z^2/2}/(1 - z)$, a generalized derangement GF. In addition, the specialized BGF

$$P(z, u, 1) = \frac{e^{(u-1)z}}{1 - z},$$

enumerates permutations according to the number of singleton cycles. This last BGF itself interpolates between the EGF of derangements ($u = 0$) and the EGF of all permutations ($u = 1$). ◁

Concise expressions for BGFs like (4), (5), (6), or (7) are precious for deriving moments, variance, and even finer characteristics of distributions, as we see next.

III. 1.2. Distributions, moments, and generating functions. As indicated in the preamble to this chapter, the eventual goal of multivariate enumeration is the quantification of properties present with high regularity in large random structures. With this subsection and the next one, we momentarily digress from our primary objective in order to introduce the basic concepts of discrete probability needed to interpret multivariate counting sequences.

Consider a pair $\langle \mathcal{A}, \chi \rangle$, where $\mathcal{A}$ is a class and χ a parameter. The *uniform probability distribution* over $\mathcal{A}_n$ is defined as follows: the probability of any $\alpha \in \mathcal{A}_n$ is equal to $1/A_n$ and the probability of any set (or "event") $\mathcal{E} \subseteq \mathcal{A}_n$ is

$$\mathbb{P}\{\mathcal{E}\} = \frac{\text{card}(\mathcal{E})}{A_n}$$

("the number of favorable cases over the total number of cases"). For this uniform probabilistic model, we write

$$\mathbb{P}_n \qquad \text{and} \qquad \mathbb{P}_{\mathcal{A}_n},$$

whenever the size and the type of combinatorial structure considered need to be emphasized.

Next, take for simplicity the parameter χ to be scalar (i.e., $d = 1$). We regard χ as defining over each $\mathcal{A}_n$ a (discrete) *random variable* defined over the (discrete) probability space $\mathcal{A}_n$:

$$\mathbb{P}_{\mathcal{A}_n}\{\chi(\alpha) = k\} = \frac{A_{n,k}}{A_n} = \frac{A_{n,k}}{\sum_k A_{n,k}}.$$

This way of thinking enables us to make use of whichever probabilistic intuition might be available in any particular case, while allowing for a natural interpretation of data. Indeed, instead of noting that there are 381922055502195 permutations of size 20 that have 10 cycles, it is perhaps more informative to state the probability of the event, which is 0.00015, i.e., about 1.5 per ten thousand. Discrete distributions are conveniently represented by *histograms* or "bar charts", where the height of the bar above k indicates the value of $\mathbb{P}\{X = k\}$. Figure 2 displays in this way two classical combinatorial distributions. Given

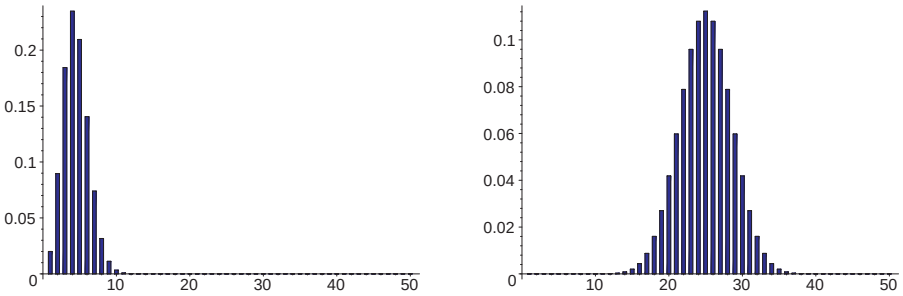


FIGURE 2. Histograms of two distributions. Left: the number of cycles in a random permutation of size 50 (Stirling cycle distribution). Right: the number of occurrences of a designated letter in a random binary word of length 50 (binomial distribution).

the uniform probabilistic model that we have been adopting, such histograms are eventually nothing but a condensed form of the "stacks" corresponding to exhaustive listings, like the one displayed in Figure 1.

An important information is provided by *moments*. Given a discrete random variable (RV) X, the *expectation* of $f(X)$ is defined as the linear functional

$$\mathbb{E}(f(X)) = \sum_k \mathbb{P}\{X = k\} \cdot f(k).$$

In particular, the (power) *moment* of order r is defined as

$$\mathbb{E}(X^r) = \sum_k \mathbb{P}\{X = k\} \cdot k^r.$$

Of special importance are the first two moments of the random variable X. The expectation (also mean or average) $\mathbb{E}(X)$ is

$$\mathbb{E}(X) = \sum_k \mathbb{P}\{X = k\} \cdot k.$$

The second moment $\mathbb{E}(X^2)$ gives rise to the *variance*,

$$\mathbb{V}(X) = \mathbb{E}\left((X - \mathbb{E}(X))^2\right) = \mathbb{E}(X^2) - \mathbb{E}(X)^2,$$

and, in turn, to the *standard deviation*

$$\sigma(X) = \sqrt{\mathbb{V}(X)}.$$

The mean deserves its name as first observed by Galileo Galilei (1564–1642): if a large number of draws are effected and values of X are observed, then the arithmetical mean of the observed values will normally be close to the expectation $\mathbb{E}(X)$. The standard deviation measures in a mean quadratic sense the dispersion of values around the expectation $\mathbb{E}(X)$.

Bivariate generating functions can be put to use in order to determine probability generating function and moments of parameters. Consider a BGF $A(z, u)$, where z marks size and u marks the parameter χ. Coefficient extraction then yields a polynomial

$$A_n(u) := \omega_n \cdot [z^n]A(z, u),$$

whose coefficients enumerate the configurations $\alpha \in \mathcal{A}_n$ according to the value of the χ parameter. Also, we have $A_n = A_n(1)$ the total number of objects in $\mathcal{A}_n$ having size n.

Consequently, the normalized polynomial

$$p_n(u) := \frac{A_n(u)}{A_n(1)} = \frac{[z^n]A(z,u)}{[z^n]A(z,1)}$$

is the *probability generating function* (PGF) of χ on $\mathcal{A}_n$ in the sense that

$$[u^k]p_n(u) = \mathbb{P}_{\mathcal{A}_n}\{\chi = k\}, \qquad \text{equivalently,} \qquad p_n(u) = \sum_k \mathbb{P}_{\mathcal{A}_n}\{\chi = k\}u^k.$$

Successive differentiations then give access to the moments of χ on A_n. In particular, one has

$$\begin{aligned} \mathbb{E}_{\mathcal{A}_n}(\chi) &= \left(\partial_u p_n(u)\right)_{u=1} & \partial_u &:= \frac{\partial}{\partial u} \\ \mathbb{E}_{\mathcal{A}_n}(\chi^2) &= \left(\partial_u^2 p_n(u) + \partial_u p_n(u)\right)_{u=1} & \partial_u^2 &:= \frac{\partial^2}{\partial u^2}, \text{ etc.} \end{aligned}$$

We thus get:

PROPOSITION III.1 (Moments from BGFs). *The moments of order 1 (mean) and of order 2 of a parameter χ are determined from the BGF $A(z,u)$ by differentiation and specialization at 1 as follows:*

$$\begin{aligned} \mathbb{E}_{\mathcal{A}_n}(\chi) &= \frac{[z^n]\partial_u A(z,1)}{[z^n]A(z,1)} \\ \mathbb{E}_{\mathcal{A}_n}(\chi^2) &= \frac{[z^n]\partial_u^2 A(z,1)}{[z^n]A(z,1)} + \frac{[z^n]\partial_u A(z,1)}{[z^n]A(z,1)}. \end{aligned}$$

In particular, the *standard deviation* is recovered from there by the usual formula,

$$\sigma(\chi)^2 = \mathbb{E}(\chi^2) - \mathbb{E}(\chi)^2.$$

As seen from basic definitions, the quantities

$$\Omega_n^{(k)} := \omega_n \cdot \left([z^n]\, \partial_u^k A(z,u)\big|_{u=1}\right)$$

give, up to normalization, the so-called *factorial moments*

$$\mathbb{E}\left(\chi(\chi-1)\cdots(\chi-k+1)\right) = \frac{1}{A_n}\Omega_n^{(k)}.$$

(Factorial moments and power moments are clearly connected by linear relations; as a matter of fact, the connection coefficients are Stirling numbers.) Most notably, $\Omega_n^{(1)}$ is the *cumulated value* of χ over all objects of $\mathcal{A}_n$:

$$\Omega_n^{(1)} \equiv \omega_n \cdot [z^n]\, \partial_u A(z,u)\big|_{u=1} = \sum_{\alpha\in\mathcal{A}_n} \chi(\alpha) \equiv A_n \cdot \mathbb{E}_{\mathcal{A}_n}(\chi).$$

EXAMPLE 1. *Moments of the Stirling cycle distribution.* Let us return to the example of cycles in permutations which is of interest in connection with certain sorting algorithms like bubble sort or insertion sort, maximum finding, and *in situ* rearrangement [**84**].

We are dealing with labelled objects, hence exponential generating functions. As seen earlier on p. 111, the BGF of permutations counted according to cycles is

$$P(z,u) = (1-z)^{-u}.$$

We have $P_n = n!$, while $\omega_n = n!$ since the BGF is exponential. (The number of permutations of size n being $n!$, the combinatorial normalization happens to coincide with the factor of $1/n!$ present in all exponential generating functions.)

By differentiation of the BGF with respect to u, then setting $u = 1$, we next get the expected number of cycles in a random permutation of size n as a Taylor coefficient

$$\mathbb{E}_n(\chi) = [z^n]\frac{1}{1-z}\log\frac{1}{1-z} = 1 + \frac{1}{2} + \cdots + \frac{1}{n}, \tag{8}$$

which is the harmonic number H_n. Thus, on average, a random permutation of size n has about $\log n + \gamma$ cycles, a well known fact of discrete probability theory.

For the variance, a further differentiation of the bivariate EGF gives

$$\sum_{n\geq 0}\mathbb{E}_n(\chi(\chi-1))z^n = \frac{1}{1-z}\left(\log\frac{1}{1-z}\right)^2. \tag{9}$$

From this expression (or from the Stirling polynomials), a calculation shows that

$$\sigma_n^2 = \left(\sum_{k=1}^{n}\frac{1}{k}\right) - \left(\sum_{k=1}^{n}\frac{1}{k^2}\right). \tag{10}$$

Thus, asymptotically,

$$\sigma_n \sim \sqrt{\log n}.$$

The standard deviation is of an order smaller than the mean, and therefore deviations from the mean have an asymptotically negligible probability of occurrence (see below the discussion of moment inequalities). Furthermore, the distribution was proved to be asymptotically Gaussian by V. Gončarov, around 1942, see [**66**] and Chapter VII. □

▷ **3.** *Stirling cycle numbers and harmonic numbers.* By the "exp-log trick" of Chapter I, the PGF of the Stirling cycle distribution satisfies

$$\frac{1}{n!}u(u+1)\cdots(u+n-1) = \exp\left(v\,\mathrm{H}_n - \frac{v^2}{2}\,\mathrm{H}_n^{(2)} + \frac{v^3}{3}\,\mathrm{H}_n^{(3)} + \cdots\right), \qquad u = 1+v$$

where $\mathrm{H}_n^{(r)}$ is the generalized harmonic number $\sum_{j=1}^{n} j^{-r}$. Consequently, any moment of the distribution is a polynomial in generalized harmonic numbers, cf (8) and (10). Also, the kth moment satisfies $\mathbb{E}_{\mathcal{P}_n}(\chi^k) \sim (\log n)^k$. (The same technique expresses the Stirling cycle number $\left[{n \atop k}\right]$ as a polynomial in generalized harmonic numbers $\mathrm{H}_{n-1}^{(r)}$.)

Alternatively, start from the expansion of $(1-z)^{-\alpha}$ and differentiate repeatedly with respect to α; for instance, one has

$$(1-z)^{-\alpha}\log\frac{1}{1-z} = \sum_{n\geq 0}\left(\frac{1}{\alpha} + \frac{1}{\alpha+1} + \cdots + \frac{1}{n-1+\alpha}\right)\binom{n+\alpha-1}{n}z^n.$$

while the next differentiation gives access to (10). ◁

The situation encountered with cycles in permutations is typical of iterative (non–recursive) structures. In many other cases, especially when dealing with recursive structures, the bivariate GF may satisfy complicated functional equations in two variables (see the example of path length in trees, Section III. 4 below) that do not make them available under an explicit form. Thus, exact expressions for the distributions are not always available, but asymptotic laws can be determined in a large number of cases (Chapter VII). In all cases, the BGF's are the central tool in obtaining mean and variance estimates, since their derivatives instantiated at $u = 1$ become univariate GFs that usually satisfy much simpler relations than the BGF's themselves.

III. 1.3. Moment inequalities. We conclude this section by a few remarks that make precise our earlier informal discussion of concentration.

Qualitatively speaking, families of distributions can be classified in two categories: the ones that are "concentrated" (i.e., the standard deviation is much smaller than the mean) and the ones that are "spread" (i.e., the standard deviation is at least as large as the mean). Figure 2 illustrates the phenomena at stake and suggests that both the Stirling cycle distributions and the binomial distributions are somehow concentrated. In contrast, the uniform distributions over $[0, n]$, which have totally flat histograms, are spread. Such informal observations are indeed supported by the Markov-Chebyshev inequalities:

PROPOSITION III.2 (Markov-Chebyshev inequalities). *Let X be a nonnegative random variable and Y an arbitrary real variable. One has*

$$\mathbb{P}\{X \geq t\mathbb{E}(X)\} \leq \frac{1}{t} \qquad \textit{(Markov inequality)}$$

$$\mathbb{P}\{|Y - \mathbb{E}(Y)| \geq t\sigma(X)\} \leq \frac{1}{t^2} \qquad \textit{(Chebyshev inequality).}$$

PROOF. Without loss of generality, one may assume that x has been scaled in such a way that $\mathbb{E}(X) = 1$. Define the function $f(x)$ whose value is 1 if $x \geq t$, and 0 otherwise. Then

$$\mathbb{P}\{X \geq t\} = \mathbb{E}(f(X)).$$

Since $f(x) \leq x/t$, the expectation on the right is less than $1/t$. Markov's inequality follows. Chebyshev's inequality then results from Markov's inequality applied to $X = |Y - \mathbb{E}(Y)|^2$. □

Proposition III.2 informs us that the probability of being much larger than the mean must decay (Markov) and that an upperbound on the decay is measured in units given by the standard deviation (Chebyshev). These bounds are *universal* in the sense that they hold for all random variables. In fact, in most cases of combinatorial interest, it is the case that far stronger decay rates—of an exponential nature—hold: see Chapter VII on multivariate asymptotics and limit distributions.

The next proposition formalizes a notion of concentration for distributions. It applies to a *family* of distributions indexed by the integers, typically the values of a scalar parameter χ on the subclasses $\{\mathcal{A}_n\}_{n\geq 0}$ indexed by size.

PROPOSITION III.3 (Concentration of distribution). *Consider a family of random variables X_n, e.g., values of a scalar parameter χ on the subclass $\mathcal{A}_n$. Assume that the means $\mu_n = \mathbb{E}(X_n)$ and the standard deviations $\sigma_n = \sigma(X_n)$ satisfy the condition*

$$\lim_{n\to+\infty} \frac{\sigma_n}{\mu_n} = 0.$$

Then the distribution of X_n is concentrated *in the sense that, for any $\epsilon > 0$, there holds*

$$\lim_{n\to+\infty} \mathbb{P}\left\{1 - \epsilon \leq \frac{X_n}{\mu_n} \leq 1 + \epsilon\right\} = 1. \tag{11}$$

PROOF. It is a direct consequence of Chebyshev's inequality. □

In probability theory, the concentration property (11) is called *convergence in probability* and is then written more concisely as

$$\frac{X_n}{\mu_n} \xrightarrow{P} 1 \qquad \text{or} \qquad X_n \xrightarrow{P} \mu_n.$$

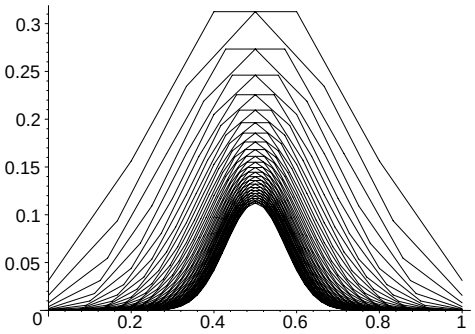


FIGURE 3. Plots of the binomial distributions for $n = 5, \dots, 50$. The horizontal axis is normalized and rescaled to 1, so that the curves display $\{\mathbb{P}(\frac{X_n}{n} = x)\}$, for $x = 0, \frac{1}{n}, \frac{2}{n}, \dots$.

It expresses the fact that values of X_n tend to become closer and closer (in relative terms) to the mean μ_n as n increases. Another figurative way to describe concentration, much used in random combinatorics, is by saying that "X_n/μ_n *tends to 1 with high probability (w.h.p.)*". When this property is satisfied, the expected value is in a strong sense a typical value.

For instance, the binomial distribution *is* concentrated, since the mean of the distribution is $n/2$ and the standard deviation is $\sqrt{n/4}$, a much smaller quantity. Figure 3 illustrates concentration by displaying the graphs (as polygonal lines) associated to the binomial distributions for $n = 5, \dots, 50$. Concentration is also quite perceptible on simulations as n gets large: the table below describes the results of batches of ten (sorted) simulations from the binomial distribution $\{\frac{1}{2^n}\binom{n}{k}\}_{k=0}^{n}$:

$n = 100$	$39, 42, 43, 49, 50, 52, 54, 55, 55, 57$
$n = 1000$	$487, 492, 494, 494, 506, 508, 512, 516, 527, 545$
$n = 10,000$	$4972, 4988, 5000, 5004, 5012, 5017, 5023, 5025, 5034, 5065$
$n = 100,000$	$49798, 49873, 49968, 49980, 49999, 50017, 50029, 50080, 50101, 50284;$

the maximal deviations from the mean observed on such samples are 22% ($n = 10^2$), 9% ($n = 10^3$), 1.3% ($n = 10^4$), and 0.6% ($n = 10^5$). Similarly, the variance computation (10) implies that the number of cycles in a random permutation of large size is concentrated. (At the opposite end of the spectrum, the uniform distributions over $[1\,..\,n]$ are *not* concentrated.)

Moment inequalities are discussed for instance in Billingsley's reference treatise [**18**, p. 74]. They are of great importance in discrete mathematics where they have been put to use in order to show the *existence* of surprising configurations. This field was pioneered by Erdős and is often known as the "probabilistic method" [in combinatorics]; see the book by Alon and Spencer [**3**] for many examples. Moment inequalities can also be used to estimate the probabilities of complex events by reducing the problems to moment estimates for occurrences of simpler configurations—this is one of the bases of the "first and second moment methods", again pioneered by Erdős, which are central in the theory of random graphs [**19, 78**]. Finally, moment inequalities serve to design, analyse, and optimize randomized algorithms, a theme excellently covered in the book by Motwani and Raghavan [**107**].

Finer estimates on distributions form the subject of our Chapter VII dedicated to limit laws. The reader may get a feeling of some of the phenomena at stake when re-examining Figure 3: the visible emergence of a continuous curve (the bell curve) corresponds to a common asymptotic shape for the whole family of distributions (the Gaussian law).

III. 2. Inherited parameters and ordinary multivariate generating functions

Parameters that are *inherited* from substructures can be taken into account by a direct extension of the symbolic method. With a suitable use of the multi-index conventions, it is even the case that the translation rules previously established in Chapters I and II can be copied verbatim. This approach opens the way to a large quantity of multivariate enumeration results that then follow automatically by the symbolic method.

Let us consider a pair $\langle \mathcal{A}, \chi \rangle$, where A is a combinatorial class endowed with its size function $|\cdot|$ and $\chi = (\chi_1, \dots, \chi_d)$ is a d-dimensional (multi)parameter. Write χ_0 for size and z_0 for the variable marking size (previously denoted by z). The key point here is to define an extended multiparameter $\overline{\chi} = (\chi_0, \chi_1, \dots, \chi_d)$, that is, we treat size and parameters on an equal basis. Then the ordinary MGF in (2) assumes an extremely simple and symmetrical form:

$$\begin{aligned} A(\mathbf{z}) &= \sum_{\mathbf{k}} A_{\mathbf{k}} \mathbf{z}^{\mathbf{k}} \\ &= \sum_{\alpha \in \mathcal{A}} \mathbf{z}^{\overline{\chi}(\alpha)}. \end{aligned} \tag{12}$$

There, the indeterminates are the vector $\mathbf{z} = (z_0, z_1, \dots, z_d)$, the indices are $\mathbf{k} = (k_0, k_1, \dots, k_d)$ (where k_0 indexes size, previously denoted by n), and the usual multi-index convention introduced in (1) is in force,

$$\mathbf{z}^{\mathbf{k}} := z_0^{k_0} z_1^{k_1} \cdots {z_d}^{k_d}, \tag{13}$$

but it is now applied to $(d+1)$-dimensional vectors.

Next, we define inherited parameters.

DEFINITION III.3. *Let $\langle \mathcal{A}, \chi \rangle$, $\langle \mathcal{B}, \xi \rangle$, $\langle \mathcal{C}, \zeta \rangle$ be three combinatorial classes endowed with parameters of the same dimension d. The parameter χ is said to be* inherited *in the following cases:*

- *Disjoint union: when $\mathcal{A} = \mathcal{B} + \mathcal{C}$, the parameter χ is inherited from ξ, ζ iff its value is determined by cases from ξ, ζ:*

$$\chi(\omega) = \begin{cases} \xi(\omega) & \text{if } \omega \in \mathcal{B} \\ \zeta(\omega) & \text{if } \omega \in \mathcal{C}. \end{cases}$$

- *Cartesian product: when $\mathcal{A} = \mathcal{B} \times \mathcal{C}$, the parameter χ is inherited from ξ, ζ iff its value is obtained additively from the values of ξ, ζ:*

$$\chi(\langle \beta, \gamma \rangle) = \xi(\beta) + \zeta(\gamma).$$

- *Composite constructions: when $\mathcal{A} = \mathfrak{K}\{B\}$, where $\mathfrak{K}$ is any of $\mathfrak{S}, \mathfrak{C}, \mathfrak{M}, \mathfrak{P}$, the parameter χ is inherited from ξ iff its value is obtained additively from the values of ξ on components; for instance, for sequences:*

$$\chi([\beta_1, \dots, \beta_r]) = \xi(\beta_1) + \cdots + \xi(\beta_r).$$

With a natural extension of the notation used for constructions, one shall write

$$\langle \mathcal{A}, \chi\rangle = \langle \mathcal{B}, \xi\rangle + \langle \mathcal{C}, \zeta\rangle, \quad \langle \mathcal{A}, \chi\rangle = \langle \mathcal{B}, \xi\rangle \times \langle \mathcal{C}, \zeta\rangle, \quad \langle \mathcal{A}, \chi\rangle = \mathfrak{K}\left\{\langle \mathcal{B}, \xi\rangle\right\}.$$

For instance, the class $\mathcal{I}$ of natural numbers, $\mathcal{I} = \mathfrak{S}_{\geq 1}\{\mathcal{Z}\}$ has OGF $I(z) = z/(1-z)$. Let ξ be the parameter that takes the constant value 1 on all elements of $\mathcal{I}$. The ordinary MGF of $\langle \mathcal{I}, \xi\rangle$ is simply

$$I(z, u) = zu + z^2u + z^3u + \cdots = \frac{zu}{1-z}.$$

The class $\mathcal{C}$ of integer compositions is, as seen in Chapter I, specified as the class of all sequences of natural integers: $\mathcal{C} = \mathfrak{S}\{\mathcal{I}\}$, with OGF

$$C(z) = \frac{1}{1 - \frac{z}{1-z}} = \frac{1-z}{1-2z}, \qquad \text{so that} \quad C_n = 2^{n-1}.$$

The constant parameter ξ is unimportant *per se*; however, the parameter χ on $\mathcal{C}$ inherited from $\langle \mathcal{I}, \xi\rangle$ carries some useful information as it represents the number of summands (or parts) that enters a composition. Its ordinary BGF is written $C(z, u)$, or $C(z_0, z_1)$ under the multi-index convention. It turns out (see below, p. 120) that the schemes translating admissible constructions in the univariate case (Chapter I) transport almost verbatim to the multivariate case, so that

$$C(z, u) = \frac{1}{1 - I(z, u)} = \frac{1}{1 - u\frac{z}{1-z}} = \frac{1-z}{1 - z(u+1)}. \tag{14}$$

We have an altogether nontrivial result obtained without any computation, which directly derives from the basic specification $\mathcal{C} = \mathfrak{S}\{\mathcal{I}\}$ relating compositions to integers. This is precisely the spirit of the symbolic method applied to parameters.

THEOREM III.1 (Inherited parameters and ordinary MGFs). *Let $\mathcal{A}$ be a combinatorial class constructed from $\mathcal{B}, \mathcal{C}$, and let χ be a parameter inherited from ξ defined on $\mathcal{B}$ and (as the case may be) from ζ on $\mathcal{C}$. Then the translation rules of admissible constructions stated in Theorem I.1 apply provided the multi-index convention is used. The associated operators on ordinary MGFs are then:*

$$\begin{aligned}
&\textit{Union:} && \mathcal{A} = \mathcal{B} + \mathcal{C} && \Longrightarrow && A(\mathbf{z}) = B(\mathbf{z}) + C(\mathbf{z})\\
&\textit{Product:} && \mathcal{A} = \mathcal{B} \times \mathcal{C} && \Longrightarrow && A(\mathbf{z}) = B(\mathbf{z}) \cdot C(\mathbf{z})\\
&\textit{Sequence:} && \mathcal{A} = \mathfrak{S}\{\mathcal{B}\} && \Longrightarrow && A(\mathbf{z}) = \frac{1}{1 - B(\mathbf{z})}\\
&\textit{Cycle:} && \mathcal{A} = \mathfrak{C}\{\mathcal{B}\} && \Longrightarrow && A(\mathbf{z}) = \sum_{\ell=1}^{\infty} \frac{\varphi(\ell)}{\ell} \log \frac{1}{1 - B(\mathbf{z}^\ell)}.\\
&\textit{Multiset:} && \mathcal{A} = \mathfrak{M}\{\mathcal{B}\} && \Longrightarrow && A(\mathbf{z}) = \exp\left(\sum_{\ell=1}^{\infty} \frac{1}{\ell} B(\mathbf{z}^\ell)\right)\\
&\textit{Powerset:} && \mathcal{A} = \mathfrak{P}\{\mathcal{B}\} && \Longrightarrow && A(\mathbf{z}) = \exp\left(\sum_{\ell=1}^{\infty} \frac{(-1)^{\ell-1}}{\ell} B(\mathbf{z}^\ell)\right)
\end{aligned}$$

PROOF. The verification for sums and products is immediate, given the combinatorial forms of OGFs. For disjoint unions, one has

$$A(\mathbf{z}) = \sum_{\alpha\in\mathcal{A}} \mathbf{z}^{\overline{\chi}(\alpha)} = \sum_{\beta\in\mathcal{B}} \mathbf{z}^{\overline{\xi}(\beta)} + \sum_{\gamma\in\mathcal{C}} \mathbf{z}^{\overline{\zeta}(\gamma)},$$

as results from the fact that inheritance is defined by cases on unions. For cartesian products, one has

$$A(\mathbf{z}) = \sum_{\alpha\in\mathcal{A}} \mathbf{z}^{\overline{\chi}(\alpha)} = \sum_{\beta\in\mathcal{B}} \mathbf{z}^{\overline{\xi}(\beta)} \times \sum_{\gamma\in\mathcal{C}} \mathbf{z}^{\overline{\zeta}(\gamma)},$$

as results from the fact that inheritance is defined additively on products.

The translation of composite constructions are then built up from the union and product schemes, in exactly the same manner as in the proof of Theorem I.1. □

This theorem is shallow. However, its importance devolves from its extremely wide range of combinatorial consequences as well as the ease with which it can be applied. The reader is especially encouraged to study carefully the example that follows as it illustrates in its bare bones version the power of the symbolic method for taking into account combinatorial parameters.

EXAMPLE 2. *Summands in integer compositions.* Let us return to integer compositions, $\mathcal{C}$. The BGF of compositions with χ the scalar parameter equal to the number of summands is

$$C(z_0, z_1) = \frac{1}{1 - I(z_0, z_1)} = \frac{1}{1 - z_1 I(z)} = \frac{1}{1 - z_0 z_1 (1 - z_0)^{-1}}, \tag{15}$$

which, up to notations, is exactly Equation (14) that is now justified. Consider next the double parameter χ where χ_1 is the number of parts equal to 1 and χ_2 the number of parts equal to 2. This is inherited from the corresponding parameter on the class $\mathcal{I}$ of natural numbers, with MGF

$$I(z_0, z_1, z_2) = z_1 z_0 + z_2 z_2 + \frac{z_0^3}{1 - z_0} = \frac{z_0}{1 - z_0} + (z_1 - 1) z_0 + (z_2 - 1) z_0^2. \tag{16}$$

Consequently, the trivariate MGF of $\langle \mathcal{C}, \chi \rangle$ is

$$C(z_0, z_1, z_2) = \frac{1}{1 - I(z_0, z_1, z_2)}. \tag{17}$$

Observe that the marking variables betray their origin. For instance, in (16) and (17), one enumerates compositions through a marking by means of dedicated variables of the configurations to be recorded, while at the same time, the usual rules translating constructions are applied. Much use of this way of envisioning the technique will be made in the remainder of this chapter.

MGFs like (14) or (15) can then be exploited in the usual way through formal power series expansions. For instance, the number of compositions of n with k parts is, by (14),

$$[z^n u^k] \frac{z(1 - z)}{1 + (1 + u) z} = \binom{n}{k} - \binom{n - 1}{k} = \binom{n - 1}{k - 1},$$

a result otherwise obtained in Chapter I by direct combinatorial reasoning (the balls-and-bars model). The number of compositions of n containing k parts equal to 1 is obtained,

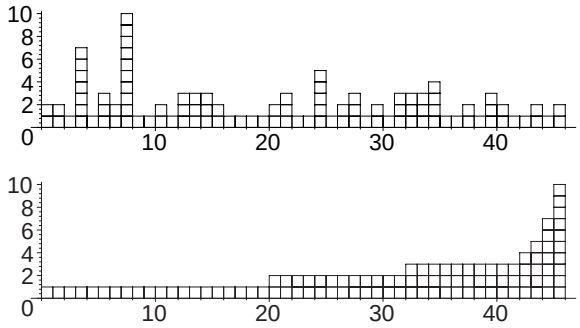


FIGURE 4. A random composition of $n = 100$ represented as a ragged landscape (top); its associated profile $1^{20}2^{12}3^{10}4^{1}5^{1}7^{1}10^{1}$, defined as the partition obtained by sorting the summands (bottom).

upon setting $z_0 = z$, $z_1 = u$ and $z_2 = 1$,

$$[z^n u^k] \frac{1-z}{1 - uz - \frac{z^2}{(1-z)}} = [z^n] \frac{(1-z)^{k+1}}{(1-z-z^2)^k},$$

where the OGF closely resembles a power of the OGF of Fibonacci numbers.

Following the discussion of Section III. 1, such MGFs also carry complete information on moments. For instance, the cumulated value of the number of parts in all compositions of n has OGF

$$\partial_u C(z,u)|_{u=1} = \frac{1-z}{(1-2z)^2},$$

as seen from Section III. 1.2, since cumulated values are obtained via differentiation of a BGF. Therefore, the expected number of parts in a random composition of n is

$$\frac{1}{2^{n-1}} [z^n] \frac{z(1-z)}{(1-2z)^2} = \frac{1}{2}(n+1).$$

What we have shown is a property of random compositions: *On average, a random composition of the integer n has about $n/2$ summands.* A further differentiation will give access to the variance. The standard deviation is found to be $\frac{1}{2}\sqrt{n-1}$, which is of an order (much) smaller than the mean. *The distribution of the number of summands in a random composition satisfies the concentration property as $n \to \infty$.*

In the same vein, the number of parts equal to a fixed number r in compositions is found to have BGF

$$\widehat{C}(z,u) = \left(1 - \left(\frac{z}{1-z} + (u-1)z^r\right)\right)^{-1}.$$

Though expanding this expression explicitly would be cumbersome, one can still pull out the number of r-summands in a random composition of size n. The differentiated form

$$\partial_u \widehat{C}(z,u)\Big|_{u=1} = \frac{z^r(1-z)^2}{(1-2z)^2}.$$

gives by partial fraction expansion

$$\partial_u \widehat{C}(z,u)\Big|_{u=1} = \frac{2^{-r-2}}{(1-2z)^2} + \frac{2^{-r-1} - r2^{-r-2}}{1-2z} + q(z),$$

for a polynomial $q(z)$ that we do not need to make explicit. Another differentiation gives access to the second moment. Consequently, one has (take the nth coefficient and divide by 2^{n-1}): *The number of r summands in a composition of size n has mean*

$$\frac{n}{2^{r+1}} + O(1);$$

the standard deviation is of order $\sqrt{n}$, which ensures concentration of distribution. □

From the point of view of random combinatorics, the example of summands shows that random compositions of large size tend to conform to a global "profile". With high probability, a composition of size n should have about $n/4$ parts equal to 1, $n/8$ parts equal to 2, and so on. Naturally, there are statistically unavoidable fluctuations, and for any finite n, the regularity of this law cannot be perfect: it tends to fade away especially as regards to largest summands that are $\log_2(n) + O(1)$ with high probability. (In this region mean and standard deviation both become of the same order and are $O(1)$, so that concentration no longer holds.) However, such observations *do* tell us a great deal about what a typical random composition must (probably) look like—it should conform to a "logarithmic profile",

$$1^{n/4}\, 2^{n/8}\, 3^{n/16}\, 4^{n/32}\, \cdots .$$

Here are for instance the profiles of two compositions of size $n = 1024$ drawn uniformly at random:

$$1^{250}\, 2^{138}\, 3^{70}\, 4^{29}\, 5^{15}\, 6^{10}\, 7^{4}\, 8^{0}, 9^{1}, \qquad 1^{253}\, 2^{136}\, 3^{68}\, 4^{31}\, 5^{13}\, 6^{8}\, 7^{3}\, 8^{1}\, 9^{1}\, 10^{2}$$

to be compared to the "ideal" profile

$$1^{256}\, 2^{128}\, 3^{64}\, 4^{32}\, 5^{16}\, 6^{8}\, 7^{4}\, 8^{2}\, 9^{1}.$$

It is a striking fact that samples of a very few elements or even just *one* element (this would be ridiculous by the usual standards of statistics) are often sufficient to illustrate asymptotic properties of large random structures. The reason is once more to be attributed to concentration of distributions whose effect is manifest here. Profiles of a similar nature present themselves amongst objects defined by the sequence construction, as we shall see throughout this book. Establishing such general laws is often not difficult but it requires the full power of complex-analytic methods developed in Chapters IV and V.

▷ **4.** *Largest summands in compositions.* For any $\epsilon > 0$, with probability tending to 1 as $n \to \infty$, the largest summand in a random integer composition of size n is almost surely of size in the interval $[(1-\epsilon)\log_2 n, (1+\epsilon)\log_2 n]$. (Hint: use the first second moment methods.) ◁

EXAMPLE 3. *Number of components in abstract schemas I.* Consider now a relation $\mathcal{A} = \mathfrak{K}\{\mathcal{B}\}$, where $\mathfrak{K}$ is any *unlabelled* constructor amongst $\mathfrak{S}, \mathfrak{C}, \mathfrak{M}, \mathfrak{P}$. The parameter "number of components", χ, defined on $\mathcal{A}$ is inherited from the constant parameter ξ equal to 1 on $\mathcal{B}$. The BGF of $\langle \mathcal{B}, \xi \rangle$ is simply

$$B(z, u) = uB(z),$$

with $B(z)$ the OGF of $\mathcal{B}$. The BGF of $\langle \mathcal{A}, \chi \rangle$ is then given by Theorem III.1. Finally, the cumulated quantities of the number of components,

$$\Omega_n := \sum_{\alpha \in \mathcal{A}} \chi(\alpha), \qquad \Omega(z) := \sum_n \Omega_n z^n,$$

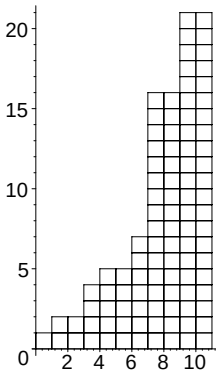


FIGURE 5. A random partition of size $n = 100$ has an aspect rather different from the profile of a random composition of the same size (Figure 4).

are given by the usual differentiation process $\partial_u(\cdot)|_{u=1}$. The easy computations are summarized by the following table:

(18)

$\mathfrak{K}$	MGF $(A(z,u))$	Cumul. OGF $(\Omega(z))$
Sequence:	$\dfrac{1}{1-uB(z)}$	$A(z)\cdot B(z)=\dfrac{B(z)}{(1-B(z))^2}$
Set:	$\exp\left(\displaystyle\sum_{k=1}^{\infty}(-1)^{k-1}\frac{u^k}{k}B(z^k)\right)$	$A(z)\cdot\displaystyle\sum_{k=1}^{\infty}(-1)^{k-1}B(z^k)$
Multiset:	$\exp\left(\displaystyle\sum_{k=1}^{\infty}\frac{u^k}{k}B(z^k)\right)$	$A(z)\cdot\displaystyle\sum_{k=1}^{\infty}B(z^k)$
Cycle:	$\displaystyle\sum_{k=1}^{\infty}\frac{\varphi(k)}{k}\log\frac{1}{1-u^kB(z^k)}$	$\displaystyle\sum_{k=1}^{\infty}\varphi(k)\frac{B(z^k)}{1-B(z^k)}$.

Mean values are then recovered as

$$\mathbb{E}_n(\chi)=\frac{\Omega_n}{A_n},$$

in accordance with the usual formula. □

▷ **5.** *r-Components in abstract schemas I.* Consider unlabelled structures. The BGF of the number of r-components in $\mathcal{A}=\mathfrak{K}\{\mathcal{B}\}$ is given by

$$A(z,u)=(1-B(z)-(u-1)B_rz^r)^{-1},\qquad A(z,u)=A(z)\cdot\left(\frac{1-z^r}{1-uz^r}\right)^{B_r},$$

in the case of sequences ($\mathfrak{K}=\mathfrak{S}$) and multisets ($\mathfrak{K}=\mathfrak{M}$), respectively. ◁

As a next illustration, we discuss the profile of random partitions (Figure 5).

EXAMPLE 4. *The profile of partitions.* Let $\mathcal{P}=\mathfrak{M}\{\mathcal{I}\}$ be the class of all integer partitions. The BGF of $\mathcal{P}$ with u marking the number χ of parts (or summands) is

$$P(z,u)=\prod_{k=1}^{\infty}\frac{1}{1-uz},$$

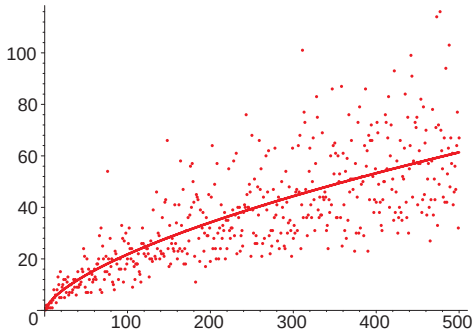


FIGURE 6. The number of parts in random partitions of size $1, \ldots, 500$: exact values of the mean and simulations (circles, one for each value of n).

as results from first principles (see also (18)). The OGF of cumulated values,

$$\Omega(z) = P(z) \cdot \sum_{k=1}^{\infty} \frac{z^k}{1-z^k}, \tag{19}$$

is obtained by logarithmic differentiation. Now, the factor on the right in (19) can be expanded: one has

$$\sum_{k=1}^{\infty} \frac{z^k}{1-z^k} = \sum_{n=1}^{\infty} d(n) z^n,$$

with $d(n)$ the number of divisors of n. Thus, the mean value of χ is

$$\mathbb{E}_n(\chi) = \frac{1}{P_n} \sum_{j=1}^{n} d(j) P_{n-j}. \tag{20}$$

The same technique applies to the number of parts equal to r. The form of BGF

$$\widetilde{P}(z,u) = \frac{1-z^r}{1-uz^r} \cdot P(z),$$

implies that the mean number of r-parts is (apply ∂_u, the set $u=1$)

$$\mathbb{E}_n(\widetilde{\chi}) = \frac{1}{P_n}[z^n] \left(P(z) \cdot \frac{z^r}{1-z^r} \right) = \frac{1}{P_n} \left(P_{n-r} + P_{n-2r} + P_{n-3r} + \cdots \right).$$

From these formulæ and a decent symbolic manipulation package, the means are calculated easily till values of n well in the range of several thousand. □

The comparison between Figures 4 and 5 together with the supporting analysis shows that different combinatorial models may well lead to rather different types of probabilistic behaviours. Figure 6 displays the exact value of the mean number of parts in random partitions of size $n = 1, \ldots, 500$, (as calculated from (20)) accompanied with the observed values of one random sample for each value of n in the range. The mean number of parts is asymptotic to

$$\frac{\sqrt{n} \log n}{\pi \sqrt{2/3}},$$

and the distribution, though it admits a comparatively large standard deviation ($O(\sqrt{n})$), is still concentrated in the technical sense; see [**42**].

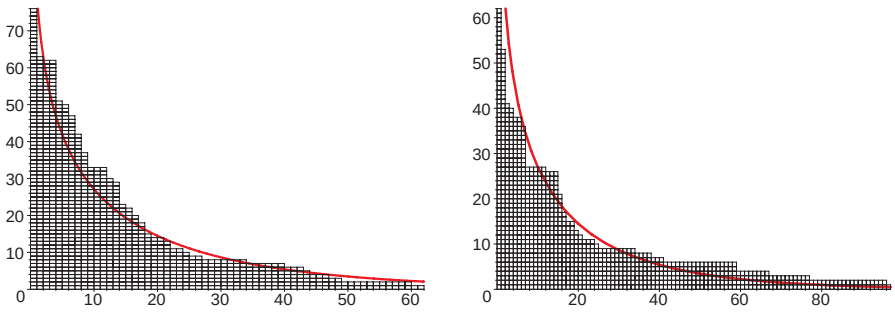


FIGURE 7. Two partitions of $\mathcal{P}_{1000}$ drawn at random, compared to the limiting shape $\Psi(x)$ defined by (21).

In recent years, Vershik and his collaborators [**38, 145**] have shown that most integer partitions tend to conform to a definite profile given (after normalization by $\sqrt{n}$) by the continous plane curve $y = \Psi(x)$ defined implicitly by

$$y = \Psi(x) \qquad \text{iff} \qquad e^{\alpha x} + e^{\alpha y} = 1, \qquad \alpha = \frac{\pi}{\sqrt{6}}. \tag{21}$$

This is illustrated in Figure 7 by two randomly drawn elements of $\mathcal{P}_{1000}$ drawn against the "most likely" limit shape. The theoretical result explains the huge differences that are manifest on simulations between integer compositions and integer partitions.

The last example demonstrates the application of BGFs to estimates regarding the root degree of a tree drawn uniformly at random amongst the class $\mathcal{G}_n$ of general Catalan trees of size n. More "global" tree parameters (e.g., number of leaves and path length) that need a recursive definition will be discussed in Section III. 4 below.

EXAMPLE 5. *Root degree in general Catalan trees.* Consider the parameter χ equal to the degree of the root in a tree. Take the class $\mathcal{G}$ of all plane unlabelled trees, aka Catalan trees. A plane tree is a root to which is appended a sequence of trees,

$$\mathcal{G} = \mathcal{Z} \times \mathfrak{S}\{\mathcal{G}\},$$

where the atomic class $\mathcal{Z}$ is the formed of a single node, so that

$$G(z) = \frac{z}{1 - G(z)}.$$

The bivariate GF with u marking χ is then

$$G(z, u) = \frac{z}{1 - uG(z)}.$$

(To see it from first principles, simply rewrite trees as roots appended to forests

$$\mathcal{G} = \mathcal{Z} \times \mathcal{F}, \qquad \mathcal{F} = \mathfrak{S}\{\mathcal{G}\}.$$

and define ξ on $\mathcal{F}$ as the number of components in the forest: χ on $\mathcal{G}$ is inherited from ζ on $\mathcal{F}$ and the constant weight 0 on the factor Z corresponding to the root. The parameter ζ on $\mathcal{F}$ is given by the usual rules for the number of components in sequences.)

From there, the cumulative GF is found,

$$\Omega(z) = \frac{zG(z)}{(1-G(z))^2}.$$

The recursive relation satisfied by G entails a further simplification,

$$\Omega(z) = \frac{1}{z}G(z)^3 = \left(\frac{1}{z} - 1\right) G(z) - 1.$$

A closed form for the coefficient results, and the mean root degree is found to be

$$\mathbb{E}_n(\chi) = \frac{1}{G_n}\left(G_{n+1} - G_n\right) = 3\frac{n-1}{n+1},$$

which is clearly asymptotic to 3.

A closer analysis reveals that the probability that the root degree equals r is

$$\mathbb{P}_n\{\chi = r\} = \frac{1}{G_n}[z^n] z\, G(z)^r \sim r2^{-r-1}.$$

A random plane tree is thus usually composed of a small number of root subtrees, at least one of which should be accordingly fairly large. □

III. 3. Inherited parameters and exponential multivariate generating functions

The theory of inheritance developed in the last section applies almost *verbatim* to labelled objects. The only difference is that the variable marking size must carry a factorial coefficient. With a suitable use of multi-index conventions, the translation mechanisms developed in the univariate case (Chapter II) remain in vigour.

Let us consider a pair $\langle \mathcal{A}, \chi\rangle$, where A is a labelled combinatorial class endowed with its size function $|\cdot|$ and $\chi = (\chi_1, \ldots, \chi_d)$ is a d-dimensional parameter. Like before, the parameter χ is extended into $\overline{\chi}$ by inserting size as zeroth coordinate and a vector $\mathbf{z} = (z_0, \ldots, z_d)$ of $d+1$ indeterminates is introduced, with z_0 marking size and z_j marking χ_j. Once the multi-index convention of (13) defining $\mathbf{z}^{\mathbf{k}}$ has been brought into the game, the exponential MGF of $\langle \mathcal{A}, \chi\rangle$ (see Definition III.2) can be rephrased as

$$\begin{aligned} A(\mathbf{z}) &= \sum_{\alpha\in\mathcal{A}} \frac{\mathbf{z}^{\mathbf{k}}}{k_0!} \\ &= \sum_{\alpha\in\mathcal{A}} \frac{\mathbf{z}^{\overline{\chi}(\alpha)}}{|\alpha|!}. \end{aligned} \tag{22}$$

In a sense, this MGF is exponential in z (alias z_0) but ordinary in the other variables; only the factorial $k_0!$ is needed to take into account relabelling induced by labelled products.

We only consider parameters that do not depend on the absolute values of labels (but may well depend on the relative order of labels): a parameter is said to be *acceptable* if, for any α, it assumes the same value on any labelled object α and all the order-consistent relabellings of α. A parameter is said to be *inherited* if it is acceptable and it is defined by cases on disjoint unions and determined additively on labelled products—this is Definition III.3 with labelled products replacing cartesian products. In particular, inheritance signifies additivity on components of labelled sequences, sets, and cycles. We can then cut-and-paste (with minor adjustments) the statement of Theorem III.1:

THEOREM III.2 (Inherited parameters and exponential MGFs). *Let $\mathcal{A}$ be a labelled combinatorial class constructed from $\mathcal{B}, \mathcal{C}$, and let χ be a parameter inherited from ξ defined on $\mathcal{B}$ and (as the case may be) from ζ on $\mathcal{C}$. Then the translation rules of admissible*

constructions stated in Theorem II.1 apply provided the multi-index convention (22) *is used. The associated operators on exponential MGFs are then:*

$$\begin{aligned}
&\textit{Union:} && \mathcal{A} = \mathcal{B} + \mathcal{C} && \Longrightarrow && A(\mathbf{z}) = B(\mathbf{z}) + C(\mathbf{z}) \\
&\textit{Product:} && \mathcal{A} = \mathcal{B} \star \mathcal{C} && \Longrightarrow && A(\mathbf{z}) = B(\mathbf{z}) \cdot C(\mathbf{z}) \\
&\textit{Sequence:} && \mathcal{A} = \mathfrak{S}\{\mathcal{B}\} && \Longrightarrow && A(\mathbf{z}) = \frac{1}{1 - B(\mathbf{z})} \\
&\textit{Cycle:} && \mathcal{A} = \mathfrak{C}\{\mathcal{B}\} && \Longrightarrow && A(\mathbf{z}) = \log \frac{1}{1 - B(\mathbf{z})}. \\
&\textit{Set:} && \mathcal{A} = \mathfrak{P}\{\mathcal{B}\} && \Longrightarrow && A(\mathbf{z}) = \exp\big(B(\mathbf{z})\big).
\end{aligned}$$

PROOF. Disjoint unions are treated like in the unlabelled multivariate case. Labelled products result from

$$A(\mathbf{z}) = \sum_{\alpha \in \mathcal{A}} \frac{\mathbf{z}^{\overline{\chi}(\alpha)}}{|\alpha|!} = \sum_{\beta \in \mathcal{B}, \gamma \in \mathcal{C}} \binom{|\beta| + |\gamma|}{|\beta|, |\gamma|} \frac{\mathbf{z}^{\overline{\xi}(\beta)}\, \mathbf{z}^{\overline{\zeta}(\gamma)}}{(|\beta| + |\gamma|)!},$$

and the usual translation of binomial convolutions that reflect labellings by means of products of exponential generating functions (like in the univariate case detailed in Chapter II). The translation for composite constructions is then immediate. □

This theorem can be exploited to determine moments, in a way that entirely parallels its unlabelled counterpart.

EXAMPLE 6. *The profile of permutations.* Let $\mathcal{P}$ be the class of all permutations and χ the number of components. The parameter χ is inherited from the parameter having constant value 1 on all cyclic permutations. Therefore, the exponential BGF is

$$P(z, u) = \exp\left(u \log \frac{1}{1 - z}\right) = (1 - z)^{-u},$$

as was already obtained by an *ad hoc* calculation in (6). We also know (page 115) that the mean number of cycles is the harmonic number H_n and that the distribution is concentrated since the standard deviation is much smaller than the mean.

Let $\widetilde{\chi}$ be the number of cycles of length r. The exponential BGF is

$$\widetilde{P}(z, u) = \exp\left(\log \frac{1}{1 - z} + (u - 1)\frac{z^r}{r}\right).$$

The EGF of cumulated values is then obtained by differentiating and with respect to u and setting $u = 1$:

$$\widetilde{\Omega}(z) = \frac{z^r}{r} \frac{1}{1 - z}.$$

The result is a remarkably simple one: *In a random permutation of size n, the mean number of r-cycles is equal to $\frac{1}{r}$ for any $r \leq n$.*

Thus, the profile of a random permutation, where profile is defined as the ordered sequence of cycle lengths departs significantly from what has been encountered for integer compositions and partitions. This formula sheds a new light on the harmonic number formula for the mean number of cycles. In particular, the mean number of cycles whose size is between $n/2$ and n is $\mathrm{H}_n - \mathrm{H}_{\lfloor n/2 \rfloor}$ a quantity that is approximately $\log 2 \doteq 0.69314$. In other words, we expect a random permutation of size n to have one or a few large cycles. (See the paper by Shepp and Lloyd [**131**] for an original discussion of largest and smallest cycles).

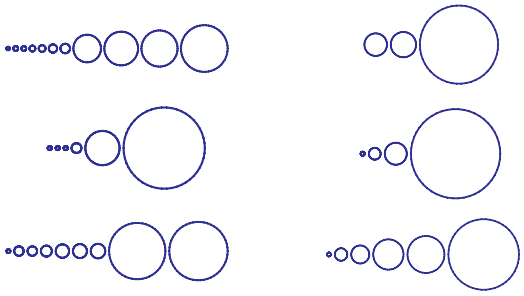


FIGURE 8. The profile of permutations: a rendering of the cycle structure of six random permutations of size 500, where circle areas are drawn in proportion to cyle lengths. Permutations tend to have a few small cycles (of size $O(1)$), a few large ones (of size $\Theta(n)$), and altogether have $\mathrm{H}_n \sim \log n$ cycles on average.

Since formuæ for labelled objects are so simple, one can get more. The BGF of the number of r-cycles is

$$\widetilde{P}(z,u) = \frac{e^{-z^r/r}}{1-z}\, e^{uz^r/r},$$

so that

$$\mathbb{P}\{\overline{\chi} = k\} = \frac{1}{k!\, r^k}[z^{n-kr}]\frac{e^{-z^r/r}}{1-z},$$

where one recognizes in the last factor the EGF of permutations without cycles of length r. From this (and the asymptotics of generalized derangement numbers in Chapter IV),one proves easily that the asymptotic law of the number of r-cycles is Poisson[1] of rate $\frac{1}{r}$. (This interesting property to be established in later chapters constitutes the starting point of [**131**].) □

EXAMPLE 7. *Number of components in abstract schemas II.* Consider labelled structures and the parameter χ equal to the number of components in a construction $\mathcal{A} = \mathfrak{K}\{\mathcal{B}\}$, where $\mathfrak{K}$ is one of $\mathfrak{S}, \mathfrak{C}, \mathfrak{P}$. The exponential BGF $A(z,u)$ and the exponential GF $\Omega(z)$ of cumulated values are given by the following table:

(23)

$\mathfrak{K}$	exp. MGF $(A(z,u))$	Cumul. EGF $(\Omega(z))$
Sequence:	$\dfrac{1}{1-uB(z)}$	$A(z)\cdot B(z) = \dfrac{B(z)}{(1-B(z))^2}$
Set:	$\exp\left(uB(z)\right)$	$A(z)\cdot B(z) = B(z)e^{B(z)}$
Cycle:	$\log\dfrac{1}{1-uB(z)}$	$\dfrac{B(z)}{1-B(z)}$.

Mean values are then easily recovered, and one finds

$$\mathbb{E}_n(\chi) = \frac{\Omega_n}{A_n} = \frac{[z^n]\Omega(z)}{[z^n]A(z)},$$

[1] The Poisson distribution of rate $\lambda > 0$ is supported by the nonnegative integers and determined by

$$\mathbb{P}\{X\} = e^{-\lambda}\frac{\lambda^k}{k!}.$$

by the same formula as in the unlabelled case. □

EXAMPLE 8. *Set partitions.* Set partitions $\mathcal{S}$ are built of blocks, $\mathcal{S} = \mathfrak{P}\{\mathfrak{P}_{\geq 1}\{\mathcal{Z}\}\}$, and the construction is reflected by the EGF equation

$$S(z) = e^{V(z)} \qquad \text{with} \qquad V(z) = e^z - 1.$$

The bivariate EGF with u marking the number of blocks is then

$$S(z, u) = e^{uV(z)} = e^{u(e^z-1)}.$$

Since set partitions are otherwise known to be enumerated by the Stirling partition numbers, one has

$$\sum_{n,k} \left\{ n \atop k \right\} u^k \frac{z^n}{n!} = e^{u(e^z-1)}, \qquad \sum_n \left\{ n \atop k \right\} \frac{z^n}{n!} = \frac{1}{k!}(e^z - 1)^k,$$

which is consistent with earlier calculations of Chapter II.

The EGF of mean values, $\Omega(z)$ is then

$$\Omega(z) = V(z)e^{V(z)} = (e^z - 1)e^{e^z-1}.$$

Due to the simple shape of $V(z)$, this is almost a derivative of $S(z)$:

$$\Omega(z) = \frac{d}{dz}S(z) - S(z).$$

Thus, the mean number of blocks in a random partition of size n is

$$\frac{\Omega_n}{S_n} = \frac{S_{n+1}}{S_n} - 1,$$

a quantity directly expressible in terms of Bell numbers. A delicate computation [**127**] based on the asymptotic expansion of the Bell numbers reveals the expected value and the standard deviation to be respectively asymptotic to

$$\frac{n}{\log n}, \qquad \frac{\sqrt{n}}{\log n}.$$

Similarly the exponential BGF of the number of blocks of size k is

$$e^{e^z + (u-1)\frac{z^k}{k!}},$$

out of which mean and variance can be derived once the asymptotic form of Bell numbers is known. □

EXAMPLE 9. *Root degree in Cayley trees.* For the class $\mathcal{T}$ of non–plane labelled trees (Cayley trees) the basic EGF equation is

$$T(z) = z\, e^{T(z)},$$

since non–planarity is taken into account by a set construction. In that case, the bivariate EGF satisfies $T(z, u) = ze^{uT(z)}$, and we find

$$\Omega(z) = zT(z)e^{T(z)} = (T(z))^2,$$

so that the mean root degree is, by Lagrange inversion,

$$2(1 - \frac{1}{n}) \sim 2.$$

A similar calculation shows that the fraction of trees with root degree k is asymptotically

$$\frac{e^{-1}}{(k-1)!}, \qquad k \geq 1,$$

which is a shifted Poisson law of rate 1. Probabilistic phenomena qualitatively similar to those encountered in plane trees are observed here as the mean root degree is asymptotic to a constant. However a Poisson law eventually reflecting the nonplanarity condition replaces the modified geometric law present in plane trees. □

▷ **6.** *Numbers of components in alignments.* Alignments ($\mathcal{O}$) are sequences of cycles (Chapter II). The expected number of components in a random alignment of $\mathcal{O}_n$ is

$$\frac{[z^n]\log(1-z)^{-1}(1-\log(1-z)^{-1})^{-2}}{[z^n](1-\log(1-z)^{-1})^{-1}}.$$

Methods of Chapter IV imply that the number of components in a random alignment has expectation $\sim n/(e-1)$ and standard deviation $\Theta(\sqrt{n})$. ◁

▷ **7.** *Image cardinality of a random surjection.* The expected cardinality of the image of a random surjection in $\mathcal{R}_n$ (see Chapter II) is

$$\frac{[z^n]e^z(2-e^z)^{-2}}{[z^n](2-e^z)^{-1}}.$$

The number of values whose preimages have cardinality k is obtained by replacing the single exponential factor e^z by $z^k/k!$. Methods of Chapter IV imply that the image cardinality of a random surjection has expectation $n/(2\log 2)$ and standard deviation $\Theta(\sqrt{n})$. ◁

Postscript: Towards a theory of schemas. Let us look back and recapitulate some of the information gathered in pages 120—130 regarding the number of components in composite structures. The classes considered in the table below are compositions of two constructions, either in the unlabelled (**U**) or the labelled (**L**) universe. Each entry contains the BGF for the number of components (e.g., cycles in permutations, parts in integer partitions, and so on), and the asymptotic orders of the mean and standard deviation of the number of components for objects of size n.

Integer partitions, $\mathfrak{M} \circ \mathfrak{S}$ (**U**) $\exp\left(u\frac{z}{1-z}+\frac{u^2}{2}\frac{z^2}{1-z^2}+\cdots\right)$ $\sim \frac{\sqrt{n}\log n}{\pi\sqrt{2/3}}, \quad \Theta(\sqrt{n})$	Integer compositions, $\mathfrak{S} \circ \mathfrak{S}$ (**U**) $\left(1-u\frac{z}{1-z}\right)^{-1}$ $\frac{n}{2}, \quad \Theta(\sqrt{n})$
Set partitions, $\mathfrak{P} \circ \mathfrak{P}$ (**L**) $\exp\left(u\left(e^z-1\right)\right)$ $\sim \frac{n}{\log n} \quad \sim \frac{\sqrt{n}}{\log n}$	Surjections, $\mathfrak{S} \circ \mathfrak{P}$ (**L**) $\left(1-u\left(e^z-1\right)\right)^{-1}$ $\sim \frac{n}{2\log 2}, \quad \Theta(\sqrt{n})$
Permutations, $\mathfrak{P} \circ \mathfrak{C}$ (**L**) $\exp\left(u\log(1-z)^{-1}\right)$ $\sim \log n, \quad \sim \sqrt{\log n}$	Alignments, $\mathfrak{S} \circ \mathfrak{C}$ (**L**) $\left(1-u\log(1-z)^{-1}\right)^{-1}$ $\sim \frac{n}{e-1}, \quad \Theta(\sqrt{n})$

Some obvious facts stand out from the data and call for explanation. First the outer construction appears to play the essential rôle: outer sequence constructs (cf integer compositions, surjections and alignments) tend to dictate a number of components that is $\Theta(n)$

on average, while outer set constructs (cf integer compositions, set partitions, and permutations) are associated with a greater variety of asymptotic regimes. The differences in behaviour are to be assigned to the rather different types of singularity involved: on the one hand sets corresponding algebraically to an $\exp(\cdot)$ operator induce an exponential blow up of singularities; on the other hand sequences expressed algebraically by quasi-inverses $(1-\cdot)^{-1}$ are likely to induce polar singularities. (Recursive structures like trees lead to yet other types of phenomena with a number of components, i.e., the root degree, that is bounded in probability.) Eventually, such facts can be organized into broad analytic schemas, as will be seen in Chapters IV–VII.

▷ **8.** *Balls in bins: occupancy.* There are n balls thrown into m bins in all possible ways (m fixed). The bivariate EGF with z marking the number of balls and u marking the number of bins that contain k balls is

$$\left(e^z + (u-1)\frac{z^k}{k!}\right)^m.$$

Let m and n tend to infinity in such a way that $\frac{n}{m} = \alpha$, a fixed constant. The proportion of bins containing k elements tends (on average and in probability) to the limit

$$e^{-\alpha}\frac{\alpha^k}{k!}.$$

Thus a Poisson law of rate α describes the occupancy of bins in a random allocation. ◁

▷ **9.** *Distinct component sizes in sets.* Take the number of *distinct* block sizes and cycle sizes in set partitions and permutations. The bivariate EGF's are

$$\prod_{n=1}^{\infty}\left(1-u+ue^{z^n/n!}\right),\quad \prod_{n=1}^{\infty}\left(1-u+ue^{z^n/n}\right).$$

Find a comparable OGF for the number of distinct summands in an integer partition. ◁

III. 4. Recursive parameters

In this section, we adapt the general methodology of previous sections in oder to treat parameters that are defined by recursive rules over structures that are themselves recursively specified. Typical applications concern trees and tree-like structures.

Consider a combinatorial class specified recursively

$$\mathcal{Y} = \mathfrak{K}\{\mathcal{Y}\}, \tag{24}$$

where $\mathfrak{K}$ is any composition of basic constructors and atoms. By distinguishing a finite set of configurations $\mathcal{X} \subset \mathcal{Y}$ considered to be "small" size, one can rephrase the specification (24) in the form

$$\mathcal{Y} = \mathcal{X} + \mathcal{V}, \qquad \mathcal{V} = \mathfrak{K}_+\{\mathcal{Y}\}. \tag{25}$$

A certain functional equation will then result for the counting GFs:

$$Y(z) = X(z) + V(z), \qquad V(z) = \Upsilon[Y(z)]. \tag{26}$$

For instance, general plane trees ($\mathcal{G}$) and Cayley trees ($\mathcal{T}$) admit the equivalent specifications

$$\begin{aligned} \mathcal{G} &= \mathcal{Z}\times\mathfrak{S}\{\mathcal{G}\}, & \mathcal{G} &= \mathcal{Z}+\mathcal{Z}\times\mathfrak{S}_{\geq 1}\{\mathcal{G}\}\\ \mathcal{T} &= \mathcal{Z}\star\mathfrak{P}\{\mathcal{T}\}, & \mathcal{T} &= \mathcal{Z}+\mathcal{Z}\star\mathfrak{P}_{\geq 1}\{\mathcal{T}\}. \end{aligned}$$

In other words, the "small" objects of size 1 have been moved out of the original construction. In the case at hand, we individualize *leaves*[2] of trees.

[2] A leaf in a rooted tree is a node without descendents.

First, consider a parameter χ on $\mathcal{Y}$ that is inherited from a parameter β defined on $\mathcal{V}$ and another parameter ξ (the "initial conditions") especially defined on the "small" structures of $\mathcal{X}$, with $\mathcal{X}, \mathcal{V}$ as on the right of (25). Then, with the auxiliary variable u marking χ on $\mathcal{A}$ as well as ξ and β on $\mathcal{X}$ and $\mathcal{V}$, general principles lead to a functional relation,

$$Y_\chi(z,u) = X_\xi(z,u) + \Upsilon[Y_\beta(z,u)]. \tag{27}$$

Here, we have indicated the involved parameters by subscripts for clarity. For instance the parameter χ equal to "root–degree" of a tree is of this structural type, being inherited from $\xi \equiv 0$ on a leaf $\mathcal{Z}$ and $\beta \equiv 1$ on components of $\mathfrak{K}_+$.

What we have done when passing from ξ to χ is to examine the effect of *one* level of recursion. Assume next that χ and β are one and the same. In other words, there is a *unique* parameter χ defined *through recursion* on objects of the recursive class $\mathcal{Y}$, with β that singles out "initial conditions". An instance is now the total number χ of leaves in a tree: it is either defined to be 1, by a special case or else it is inherited additively as the sum of the values obtained from the root subtrees, cf (25). Indeed, if $\tau = \langle \rho, \tau_1, \dots, \tau_r \rangle$ is a tree with root ρ and $r \geq 1$, one has

$$\chi(\tau) = \chi(\tau_1) + \cdots + \chi(\tau_r),$$

with χ coinciding with $\xi \equiv 1$ on atoms. With this identification of χ and β, the bivariate generating function $Y(z,u)$ becomes *implicitly defined* by a *functional equation* of the form

$$Y_\chi(z,u) = X_\xi(z,u) + \Upsilon[Y_\chi(z,u)].$$

Once the mechanism is clear, we may as well drop subscripts indicative of parameters and write

$$Y(z,u) = X(z,u) + \Upsilon[Y(z,u)]. \tag{28}$$

This stands out as a "deformation" of the usual univariate functional equation for the GF of $\mathcal{Y}$, to which it reduces when $u = 1$. With a natural extension of notations, we may even write symbolically a recursive specification for class–parameter pairs,

$$\langle \mathcal{Y}, \chi \rangle = \langle \mathcal{X}, \xi \rangle + \Upsilon[\langle \mathcal{Y}, \chi \rangle],$$

and simply apply the common translation mechanisms to get back (28). Naturally, similar considerations apply to vectorial parameters and/or to collections of mutually recursive combinatorial classes.

EXAMPLE 10. *Leaves in special varieties of trees.* How many leaves does a random tree of some variety have? Can different varieties of trees be somehow distinguished by the proportion of their leaves? Beyond the botany of combinatorics, such considerations are for instance relevant to the analysis of algorithms since tree leaves, having no descendants, can be stored more economically; see [**85**, Sec. 2.3] for a motivation to such questions.

Consider once more the class $\mathcal{G}$ of plane unlabelled trees, $\mathcal{G} = \mathcal{Z} \times \mathfrak{S}\{\mathcal{G}\}$, enumerated by the Catalan numbers: $G_n = \frac{1}{n}\binom{2n-2}{n-1}$. The number $G_{n,k}$ of trees with n nodes and k leaves is to be determined. Let χ be the parameter "number of leaves" and $G(z,u)$ the associated bivariate OGF. In order to individuate leaves, rewrite the original specification of plane trees as

$$\mathcal{G} = \mathcal{Z} + (\mathcal{Z} \times \mathfrak{S}_{\geq 1}\{\mathcal{G}\}).$$

The parameter χ is additive; hence, to the defining relation, there corresponds termwise

$$G(z,u) = zu + \frac{zG(z,u)}{1 - G(z,u)}.$$

The induced quadratic equation can be solved explicitly

$$G(z,u) = \frac{1}{2}\left(1 + (u-1)z - \sqrt{1 - 2(u+1)z + (u-1)^2 z^2}\right).$$

It is however simpler to expand using the Lagrange inversion theorem which provides

$$\begin{aligned} G_{n,k} &= [u^k]\,([z^n]G(z,u)) = [u^k]\left(\frac{1}{n}[y^{n-1}](u + \frac{y}{1-y})^n\right) \\ &= \frac{1}{n}\binom{n}{k}[y^{n-1}]\frac{y^{n-k}}{(1-y)^{n-k}} = \frac{1}{n}\binom{n}{k}\binom{n-2}{k-1}. \end{aligned}$$

These numbers are known as Narayana numbers, see *EIS* **A001263**, and they surface repeatedly in connexion with ballots problems.) The mean number of leaves then derives from the cumulative GF,

$$\Omega(z) = \partial_u G(z,u)|_{u=1} = \frac{1}{2}\,z + \frac{1}{2}\,\frac{z}{\sqrt{1-4z}},$$

so that the mean is $n/2$ exactly for $n \geq 2$. Also, the distribution is concentrated since the standard deviation is easily calculated to be $O(\sqrt{n})$.

In a similar vein, define binary plane trees by the equation,

$$\mathcal{B} = \mathcal{Z} + (\mathcal{B} \times \mathcal{Z}) + (\mathcal{Z} \times B) + (\mathcal{B} \times \mathcal{Z} \times \mathcal{B}), \tag{29}$$

which stresses the distinction between four types of nodes: leaves, left branching, right branching, and binary. Let u_0, u_1, u_2 be variables that mark nodes of degree 0,1,2, respectively. Then the root decomposition (29) gives for the MGF $B = B(z, u_0, u_1, u_2)$ the functional equation

$$B = zu_0 + 2zu_1 + zu_2B^2,$$

by which Lagrange inversion gives

$$B_{n,k_0,k_1,k_2} = \frac{2^{k_1}}{n}\binom{n}{k_0, k_1, k_2},$$

subject to the natural conditions: $k_0 + k_1 + k_2 = n$ and and $k_0 = k_2 + 1$. Specializations and moments can be easily calculated from such an approach [**117**]. In particular, the mean number of nodes of each type is asymptotically:

$$\text{leaves: } \sim \frac{n}{4}, \qquad \text{1-nodes: } \sim \frac{n}{2}, \qquad \text{2-nodes: } \sim \frac{n}{2}.$$

Finally, for Cayley trees, the bivariate EGF with u marking the number of leaves is the solution to

$$T(z,u) = uz + z(e^{T(z,u)} - 1).$$

The distribution is expressed in terms of Stirling partition numbers. The mean number of leaves in a random Cayley tree is found to be asymptotic to ne^{-1}. □

▷ **10.** *Leaves and node-degree profile in simple varieties of trees.* The mean number of nodes of outdegree k in a random Cayley tree of size n is asymptotic to

$$n \cdot e^{-1}\,\frac{1}{k!}.$$

Degrees of nodes are thus approximately given by a Poisson law of rate 1.

More generally, for a family of trees generated by $T(z) = z\phi(T(z))$ with ϕ a power series, the BGF of the number of nodes of degree k satisfies

$$T(z,u) = z\left(\phi(T(z,u)) + (\phi_k u - 1)T(z,u)^k\right),$$

where $\phi_k = [u^k]\phi(u)$. The cumulative GF is

$$\Omega(z) = z\frac{\phi_k T(z)^k}{1 - z\phi'(T(z))} = \phi_k z^2 T(z)^{k-1} T'(z),$$

from which moments can be determined. $\lhd$

$\rhd$ **11.** *Marking in functional graphs.* Consider the class $\mathcal{F}$ of finite mappings discussed in Chapter II:

$$\mathcal{F} = \mathfrak{P}\{\mathcal{K}\},\ \mathcal{K} = \mathfrak{C}\{\mathcal{T}\},\ \mathcal{T} = \{1\} \star \mathfrak{P}\{\mathcal{T}\}.$$

The translation on EGF's is

$$F(z) = e^{K(z)},\ K(z) = \log\frac{1}{1 - T(z)},\ T(z) = e^{T(z)}.$$

Here are bivariate EGF's for (i) the number of components, (ii) the number of maximal trees, (iii) the number of leaves:

$$(i)\ e^{uK(z)}, \qquad (ii)\ \frac{1}{1 - uT(z)}, \qquad (iii)\ \frac{1}{1 - T(z,u)} \quad \text{with} \quad T(z,u) = (u-1)z + ze^{T(z,u)}.$$

The trivariate EGF $F(u_1, u_2, z)$ of functional graphs with u_1 marking components and u_2 marking trees is

$$F(z, u_1, u_2) = \exp(u_1 \log(1 - u_2 T(z))^{-1}) = \frac{1}{(1 - u_2 T(z))^{u_1}}.$$

An explicit expression for the coefficients of the trivariate F involves the Stirling cycle numbers. $\lhd$

We shall stop here these examples that could be multiplied *ad libitum* since such calculations greatly simplify when interpreted in the light of asymptotic analysis. The phenomena observed asymptotically are, for good reasons, especially close to what the classical theory of branching processes provides.

We next turn to finer characteristics of trees, like path length. As a preamble, one needs a simple linear transformation on combinatorial parameters. Let $\mathcal{A}$ be a class equipped with two scalar parameters, χ and ξ, related by

$$\chi(\alpha) = |\alpha| + \xi(\alpha).$$

Then, the combinatorial form of BGFs yields

$$\sum_{\alpha \in \mathcal{A}} z^{|\alpha|} u^{\chi(\alpha)} = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|} u^{|\alpha| + \chi(\alpha)},$$

that is,

$$A_\chi(z, u) = A_\xi(zu, u). \tag{30}$$

This is clearly a general mechanism: *a linear transformation on parameters induces a monomial substitution on the corresponding marking variables in MGFs.* We now put it to use in the analysis of path length in trees.

EXAMPLE 11. *Path length in trees.* Path length is an important "global" characteristic of trees classically defined as the sum of distances of all nodes to the root of the tree. (Distances are measured by the number of edges on the minimal connecting path.) For instance, when a tree is used as a data structure with nodes containing additional informations, path length represents the total cost of accessing all data items when a search is started from the root. For this reason, path length surfaces, under various models, in the analysis of algorithms like tree-sort, quicksort, and so on [**85, 130**].

From the definition of path length,

$$\lambda(\tau) := \sum_{\nu \in \tau} \mathrm{dist}(\nu, \mathrm{root}(\tau)),$$

there immediately results that

$$\lambda(\tau) = \sum_{v \text{ root subtree of } \tau} (\lambda(v) + |v|)\,. \tag{31}$$

(Distribute nodes in their corresponding subtrees: distances to the subtree roots must be corrected by 1; regroup terms.)

From this point on, we specialize the discussion to general plane trees (see Ex. 12 for more): $\mathcal{G} = \mathcal{Z}\mathfrak{S}\{\mathcal{G}\}$. Introduce momentarily the parameter $\mu(\tau) = |\tau| + \lambda(\tau)$. Then, one has from the inductive definition (31) and the general transformation rule (30):

$$G_\lambda(z,u) = \frac{z}{1 - G_\mu(z,u)} \qquad \text{and} \qquad G_\mu(z,u) = G_\lambda(zu,u).$$

In other words, $G(z,u) \equiv G_\lambda(z,u)$ satisfies a nonlinear functional equation of the difference type:

$$G(z,u) = \frac{z}{1 - G(uz,u)}. \tag{32}$$

The generating function $\Omega(z)$ of cumulated values of λ then obtains by differentiation with respect to u upon setting $u = 1$. We find in this way that $\Omega(z) := \partial_u G(z,1)$ satisfies

$$\Omega(z) = \frac{z}{(1 - G(z))^2}\left(zG'(z) + \Omega(z)\right),$$

which is a linear equation that solves to

$$\Omega(z) = z^2 \frac{G'(z)}{(1 - G(z))^2 - z} = \frac{z}{2(1-4z)} - \frac{z}{2\sqrt{1-4z}}$$

where $\delta = 1 - 4z$. Consequently, one has

$$\Omega_n = 2^{2n-1} - \binom{2n-2}{n-1},$$

where the sequence starting 1, 5, 22, 93, 386 for $n \geq 2$ constitutes *EIS* **A000346**. We thus have:

> *The mean path length of a random Catalan tree of size n is asymptotic to $2\sqrt{\pi n^3}$; in short: a branch in a random Catalan tree of size n has expected length of the order of $\sqrt{n}$.*

Under the uniform combinatorial model, trees thus tend to be somewhat imbalanced. □

The imbalance property found for random Catalan trees is a general phenomenon—it applies to binary Catalan and more generally to all simple varieties of trees. Ex. 12 below and Chapter V imply that path length is invariably of order $n\sqrt{n}$ on average in such cases. Height is of typical order $\sqrt{n}$ as shown by Rényi and Szekeres [**120**], de Bruijn, Knuth and Rice [**37**], Kolchin [**90**], as well as Flajolet, and Odlyzko [**52**]. Figure 9 borrowed from [**130**] illustrates this on a simulation. (The contour of the histogram of nodes by levels, once normalized, has been proved to converge to the process known as Brownian excursion.)

▷ **12.** *Path length in simple varieties of trees.* The BGF of path length in a variety of trees generated by $T(z) = z\phi(T(z))$ satisfies

$$T(z,u) = z\phi(T(zu,u)).$$

In particular, the cumulative GF is

$$\Omega(z) \equiv \partial_u \left(T(z,u)\right)_{u=1} = \frac{\phi'(T(z))}{\phi(T(z))}(zT'(z))^2,$$

from which coefficients can be extracted. ◁

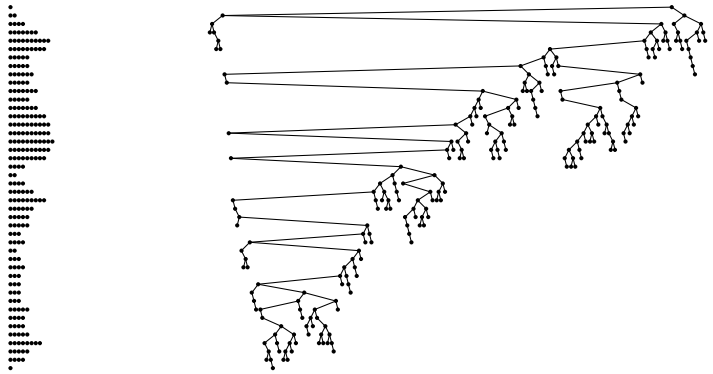


FIGURE 9. A random pruned binary tree of size 256 and its associated level profile: the histogram on the left displays the number of nodes at each level in the tree.

III. 5. "Universal" generating functions and combinatorial models

By a *universal* generating function, we mean a generating function in a number (possibly infinite) of variables that mark a homogeneous collection of characteristics of a combinatorial class. For instance one may be interested in the joint distribution of all the different letters composing words, the number of cycles of all lengths in permutations, and so on. A universal MGF naturally entails very detailed knowledge on the enumerative properties of structures to which it is relative. Universal generating functions, given their expressive power, also make weighted models accessible to calculation, a situation that covers in particular Bernoulli trials and branching processes from classical probability theory.

As a basic example, consider the class of all words $\mathcal{W} = \mathfrak{S}\{\mathcal{A}\}$ over some finite alphabet $\mathcal{A} = \{a_1, \dots, a_r\}$. Let $\chi = (\chi_1, \dots, \chi_r)$, where $\chi_j(w)$ is the number of occurrences of the letter a_j in word w. The MGF of $\mathcal{A}$ with respect to χ is

$$A(z, \mathbf{u}) = zu_1 + zu_2 + \cdots + zu_r,$$

and χ on $\mathcal{W}$ is clearly inherited from χ on $\mathcal{A}$. Thus, by the sequence rule, one has

$$W(z, \mathbf{u}) = \frac{1}{1 - z(u_1 + u_2 + \cdots + u_r)}, \tag{33}$$

which describes all words according to their compositions into letters. In particular, the number of words with n_j occurrences of letter a_j and $n = \sum n_j$ is

$$[u_1^{n_1} u_2^{n_2} \cdots u_r^{n_r}]\,(u_1 + u_2 + \cdots + u^r)^n = \binom{n}{n_1, n_2, \dots, n_r} = \frac{n!}{n_1! n_2! \cdots n_r}.$$

We are back to the usual multinomial coefficients.

▷ **13.** *After Bhaskara Acharya* (*circa 1150*AD). Consider all the numbers formed in decimal with digit 1 used once, with digit 2 used twice,..., with digit 9 used nine times. Such numbers all have 45 digits. Compute their sum S and discover, much to your amazement that S equals

458755596000061532190847692863999999999999999541244403999938467809152307136000000.

This number has a long run of nines (and further nines are hidden!). Is there a simple explanation? This exercise is inspired by the Indian mathematician Bhaskara Acharya who discovered multinomial coefficients near 1150AD; see [**85**, p. 23] for a brief historical note. ◁

Next, consider permutations and the various lengths of their cycles. The MGF where u_1, u_2 mark 1-cycles and 2-cycles respectively is

$$\exp\left(u_1\frac{z}{1}+u_2\frac{z^2}{2}+\frac{z^3}{3}+\cdots\right).$$

By analogy, one is led to considering an MGF in infinitely many variables

$$U(z,\mathbf{u})=\exp\left(u_1\frac{z}{1}+u_2\frac{z^2}{2}+u_3\frac{z^3}{3}+\cdots\right). \tag{34}$$

The MGF expression U has the neat feature that, upon specializing all but a finite number of u_j to 1, we derive all the particular cases of interest with respect to any finite collection of cycles lengths. Mathematically, an object like U in (34) is perfectly well defined: it suffices to consider $\mathbb{K}=\mathbb{C}(u)$ the field of fractions in infinitely many variables—any element of $\mathbb{K}$ involves only finitely many indeterminates; then calculate normally with formal power series of $\mathbb{K}[\![z]\!]$, assuming $\mathbb{K}$ as the coefficient field. Indeed, with the notion of formal convergence[3] defined in the appendix, one can take limits in $\mathbb{K}[\![z]\!]$ and write legitimately

$$\lim_{m\to\infty}\exp\left(\sum_{j=1}^{m}u_j\frac{z^j}{j}\right)=\exp\left(\lim_{m\to\infty}\sum_{j=1}^{m}u_j\frac{z^j}{j}\right)=U.$$

Henceforth, we shall keep in mind that verifications of formal correctness are always possible by returning to basic definitions.

Universal generating functions are often surprisingly simple to expand. For instance, the equivalent form of (34)

$$U(z,\mathbf{u})=e^{u_1 z/1}\cdot e^{u_2 z^2/2}\cdot e^{u_3 z^3/3}\cdots$$

implies immediately that the number of permutations with n_1 cycles of size 1, n_2 of size 2, etc, is

$$\frac{n!}{c_1!\,c_2!\,\cdots c_n!\;1^{c_1}\,2^{c_2}\cdots n^{c_n}}, \tag{35}$$

provided $\sum jc_j=n$. This is a result originally due to Cauchy. Similarly, the EGF of set partitions with u_j marking the number of blocks of size j is

$$\exp\left(u_1\frac{z}{1!}+u_2\frac{z^2}{2!}+u_3\frac{z^3}{3!}+\cdots\right).$$

A formula analogous to (35), with j^{c_j} being replaced by $j!^{c_j}$ follows. Several examples of such "universal" generating functions are presented in Comtet's book; see [**28**], pages 225 and 233.

---

[3] In contrast, the quantity evocative of a generating function of words over an infinite alphabet

$$S\stackrel{!}{=}\left(1-z\sum_{j=1}^{\infty}u_j\right)^{-1}$$

cannot receive a sound definition as a element of the formal domain $\mathbb{K}[\![z]\!]$; for instance, the coefficient of z in the sequence of approximants would not even converge to an element of $\mathbb{K}$ equipped with the discrete topology.

▷ **14.** *Universal GFs for compositions and surjections.* The universal GF's of integer compositions and surjections with u_j marking the number of components of size j are

$$\frac{1}{1-\sum_{j=1}^{\infty} u_j z^j}, \qquad \frac{1}{1-\sum_{j=1}^{\infty} u_j \frac{z^j}{j!}}.$$

The associated counts with $n = \sum_j j n_j$ are given by

$$\binom{n_1+n_2+\cdots}{n_1, n_2, \ldots}, \qquad \frac{n!}{1!^{n_1} 2!^{n_2} \cdots} \binom{n_1+n_2+\cdots}{n_1, n_2, \ldots}.$$

These factored forms derive directly from the multinomial expansion. The symbolic form of the multinomial expansion of powers of a generating function is sometimes expressed in terms of Bell polynomials, themselves nothing but a rephrasing of the multinomial expansion; see Comtet's book [**28**, Sec. 3.3] for a fair treatment of such polynomials. ◁

▷ **15.** *Faà di Bruno's formula.* The formulæ for the successive derivatives of a functional composition $h(z) = f(g(z))$

$$\partial_z h(z) = f'(g(z))g'(z), \quad \partial_z^2 h(z) = f''(g(z))g'(z)^2 + f'(z)g''(z), \ldots,$$

are clearly equivalent to the expansion of a formal power series composition (assume $f(0) = g(0) = 0$):

$$h_1 = f_1 g_1, \quad h_2 = f_2 g_1^2 + 2 f_1 g_2, \ldots .$$

The general form, a mere avatar of the multinomial expansion, is known as Fàa di Bruno's formula [**28**, p. 137]. (Faà di Bruno (1825–1888) was canonized by the Catholic Church in 1988, albeit not for reasons related to his formula.) ◁

▷ **16.** *Relations between symmetric functions.* Symmetric functions may be manipulated by mechanisms that are often reminiscent of the set and multiset construction. They appear in many areas of combinatorial enumeration. Let $X = \{x_i\}_{i=1}^r$ be a collection of formal variables. Define the symmetric functions

$$\prod_i (1 + x_i z) = \sum_n a_n z^n, \quad \prod_i \frac{1}{1-x_i z} = \sum_n b_n z^n, \quad \sum_i \frac{x_i z}{1-x_i z} = \sum_n c_n z^n.$$

The a_n, b_n, c_n, called resp. elementary, monomial, and power symmetric functions are expressible as

$$a_n = \sum_{i_1 < i_2 < \cdots < i_r} x_{i_1} x_{i_2} \cdots x_{i_r}, \quad b_n = \sum_{i_1 \le i_2 \le \cdots \le i_r} x_{i_1} x_{i_2} \cdots x_{i_r}, \quad c_n = \sum_{i=1}^{r} x_i^r.$$

The following relations hold:

$$\begin{aligned} B(z) &= \frac{1}{A(-z)}, & A(z) &= \frac{1}{B(-z)}, \\ C(z) &= z\frac{d}{dz}\log B(z), & B(z) &= \exp\int_0^z C(t)\,\frac{dt}{t}. \end{aligned}$$

Consequently, each of a_n, b_n, c_n is polynomially expressible in terms of any of the other quantities. (The connection coefficients again involve multinomials.) ◁

▷ **17.** *Regular graphs.* A graph is r–regular iff each node has degree exactly equal to r. The number of r–regular graphs of size n is

$$[x_1^r x_2^r \cdots x_n^r] \prod_{1 \le i < j \le n} (1 + x_i x_j).$$

[Gessel [**65**] has shown how to extract explicit expressions from such huge symmetric functions.] ◁

III. 5.1. Word models. The enumeration of words, or "sequences" as they are sometimes also called, constitutes a rich chapter of combinatorial analysis. Applications are to be found in classical probability theory and statistics [**33**] as well as in computer science [**139**] and mathematical models of biology [**149**]. We focus our attention here to problems that involve universal generating functions.

EXAMPLE 12. *Words and records.* Fix an alphabet $\mathcal{A} = \{a_1, \ldots, a_r\}$ and let $\mathcal{W} = \mathfrak{S}\{\mathcal{A}\}$ be the class of all words over $\mathcal{A}$, where $\mathcal{A}$ is naturally ordered by $a_1 < a_2 < \cdots < a_r$. Given a word $w = w_1 \cdots w_n$, a (strict) record is an element w_j that is larger than all preceding elements: $w_j > w_i$ for all $i < j$. (Refer to Figure 12 of Chapter II for a graphical rendering of records in the case of permutations.)

Consider first the subset of $\mathcal{W}$ comprising all words that have the letters $a_{i_1}, \ldots, a_{i_k}$ as successive records, where $i_1 < \cdots < i_k$. The symbolic description of this set is in the form of a product of k terms

$$\left(a_{i_1}(a_1 + \cdots + a_{i_1})^\star\right) \quad \cdots \quad \left(a_{i_k}(a_1 + \cdots + a_{i_k})^\star\right). \tag{36}$$

Consider now MGFs of words where z marks length, v marks the number of records, and each u_j marks the number of occurrences of letter a_j. The MGF associated to the subset described in (36) is then

$$\left(zvu_{i_1}(1 - z(u_1 + \cdots + u_{i_1}))^{-1}\right) \quad \cdots \quad \left(zvu_{i_k}(1 - z(u_1 + \cdots + u_{i_k}))^{-1}\right).$$

Summing over all values of k and of $i_1 < \cdots < i_k$ gives

$$W(z, v, \mathbf{u}) = \prod_{s=1}^{r} \left(1 + zvu_s\,(1 - z(u_1 + \cdots + u_s))^{-1}\right), \tag{37}$$

the rationale being that, for arbitrary quantities y_s, one has

$$\sum_{k=0}^{r} \sum_{1 \le i_1 < \cdots < i_k \le r} y_{i_1} y_{i_2} \cdots y_{i_k} = \prod_{s=1}^{r} (1 + y_s).$$

We shall encounter more applications of (37) below. For the time being let us simply examine the mean number of records in a word of length n over the alphabet $\mathcal{A}$, when all such words are taken equally likely. One should set $u_j \mapsto 1$ (the composition into specific letters is forgotten), so that W assumes the simpler form

$$W(z, v) = \prod_{j=1}^{r} \left(1 + \frac{vz}{1 - jz}\right).$$

Logarithmic differentiation then gives access to the generating function of cumulated values,

$$\Omega(z) \equiv \frac{\partial}{\partial v} W(z, v)\Big|_{v=1} = \frac{z}{1 - rz} \sum_{j=1}^{r} \frac{1}{1 - (j-1)z}.$$

Thus, by partial fraction expansion, the mean number of records in $\mathcal{W}_n$ (whose cardinality is r^n) has value

$$\mathbb{E}_{\mathcal{W}_n}(\#\,\text{records}) = \mathrm{H}_r - \sum_{j=1}^{r-1} \frac{(j/r)^n}{r - j}. \tag{38}$$

There appears the harmonic number H_r, like in the permutation case, but now with a negative correction term which, for fixed r, vanishes exponentially fast with n (this betrays the fact that some letters from the alphabet might be missing). □

EXAMPLE 13. *Weighted word models and Bernoulli trials.* Let $\mathcal{A} = \{a_1, \ldots, a_r\}$ be an alphabet of cardinality r, and let $\Lambda = \{\lambda_1, \ldots, \lambda_r\}$ be a system of numbers called *weights*, where weight λ_j is viewed as attached to letter a_j. Weights may be extended from letters to words multiplicatively by defining the weight $\pi(w)$ of word w as

$$\begin{aligned} \pi(w) &= \lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_n} \qquad \text{if} \quad w = a_{i_1}a_{i_2}\cdots a_{i_n} \\ &= \prod_{j=1}^{r} \lambda_j^{\chi_j(w)}, \end{aligned}$$

where $\chi_j(w)$ is the number of occurrences of letter a_j in w. Finally, the weight of a set is by definition the *sum* of the weights of its elements.

Combinatorially, weights of sets are immediately obtained once the corresponding generating function is known. Indeed, let $\mathcal{S} \subseteq \mathcal{W} = \mathfrak{S}\{\mathcal{A}\}$ have "universal" GF

$$S(z, u_1, \ldots, u_r) = \sum_{w\in S} z^{|w|} u_1^{\chi_1(w)} \cdots u_r^{\chi_r(w)},$$

where $\chi_j(w)$ is the number of occurrences of letter a_j in w. Then one has

$$S(z, \lambda_1, \ldots, \lambda_r) = \sum_{w\in S} z^{|w|}\pi(w),$$

so that extracting the coefficient of z^n gives the total weight of $\mathcal{S}_n = \mathcal{S} \cap \mathcal{W}_n$ under the weight system Λ. In other words, *the GF of a weighted set is obtained by substitution of the numerical values of the weights inside the associated universal MGF.*

In probability theory, Bernoulli trials refer to sequences of independent draws from a fixed distribution with finitely many possible values. One may think of the succession of flippings of a coin or castings of a dice. If any trial has r possible outcomes, then the various possibilities can be described by letters of the r-ary alphabet $\mathcal{A}$. If the probability of the jth outcome is taken to be λ_j, then the Λ-weighted models on words becomes the usual probabilistic model of independent trials. (In this situation, the λ_j's are often written as p_j's.) Observe that, in the probabilistic situation, one must have $\lambda_1 + \cdots + \lambda_r = 1$ with each λ_j satisfying $0 \le \lambda_j \le 1$. The equiprobable case, where each outcome has probability $1/r$ can be obtained by setting $\lambda_j = 1/r$ and it then becomes equivalent to the usual enumerative model. In terms of GFs, the coefficient $[z^n]S(z, \lambda_1, \ldots, \lambda_r)$ then represents the probability that a random word of $\mathcal{W}_n$ belongs to $\mathcal{S}$. Multivariate generating functions and cumulative generating functions then obey properties similar to their usual counterparts.

As an illustration, assume one has a biased coin with probability p for heads (H) and $q = 1 - p$ for tails (T). Consider the event: "in n tosses of the coin, there never appear ℓ contiguous heads. The alphabet is $\mathcal{A} = \{H, T\}$. The language describing the events of interest (with varying n) is, as seen in Chapter I,

$$\mathcal{S} = \mathfrak{S}_{<\ell}\{H\}\mathfrak{S}\{T\mathfrak{S}_{<\ell}\{H\}\}.$$

Its universal GF with u marking heads and v marking tails is then

$$W(z, u, v) = \frac{1 - z^\ell u^\ell}{1 - zu}\left(1 - zv\frac{1 - z^\ell u^\ell}{1 - zu}\right)^{-1}.$$

Thus, the probability of the absence of ℓ-runs amongst a sequence of n random coin tosses is obtained after the substitution $u \to p$, $v \to q$ in the MGF,

$$[z^n]\frac{1-p^\ell z^\ell}{1-z+qp^\ell z^{\ell+1}},$$

leading to an expression which is amenable to numerical or asymptotic analysis. (Fellers' book [**43**, p. 322–326] offers for instance a classical discussion of the problem.)

To conclude the discussion of probabilistic models on words, we come back to the analysis of records. Assume now that the alphabet $\mathcal{A} = \{a_1, \ldots, a_r\}$ has in all generality the probability p_j associated with the letter a_j. The mean number of records is analysed by a process entirely parallel to the derivation of (38): one finds by logarithmic differentiation of (37)

(39)
$$\mathbb{E}_{\mathcal{W}_n}(\#\text{ records}) = [z^n]\Omega(z) \qquad \text{where} \quad \Omega(z) = \frac{z}{1-z}\sum_{j=1}^{r}\frac{p_j}{1-z(p_1+\cdots+p_{j-1})}.$$

The cumulative GF $\Omega(z)$ in (39) has simple poles at the points $1, 1/P_{r-1}, 1/P_{r-2}$, and so on, where $P_s = p_1 + \cdots + p_s$. For asymptotic purposes, only the dominant poles at $z = 1$ counts (see Chapter IV for a systematic discussion), near which

$$\Omega(z) \underset{z\to 1}{\sim} \frac{1}{1-z}\sum_{j=1}^{r}\frac{p_j}{1-P_{j-1}}.$$

Consequently, one has an elegant asymptotic formula generalizing the case of permutations that has a harmonic mean (8):

> *The mean number of records in a random word of length n with non uniform letter probabilities p_j satisfies asymptotically*
>
> $$\mathbb{E}_{\mathcal{W}_n}(\#\text{ records}) \sim \sum_{j=1}^{r}\frac{p_j}{p_j+p_{j+1}+\cdots+p_r}.$$

This relation and similar ones were obtained by Burge [**25**]; analogous ideas may serve to analyse the sorting algorithm *Quicksort* under equal keys [**128**] as well as the hybrid data structures of Bentley and Sedgewick; see [**12, 27**]. □

Similar considerations apply to weighted EGFs of words. For instance, the probability of having attained a complete coupon collection in case a company issues coupon j with probability p_j (with $1 \le j \le r$), is

$$n![z^n]\prod_{j=1}^{r}\left(e^{p_j z}-1\right).$$

The probability that all coupons are different at time n is

$$n![z^n]\prod_{j=1}^{r}\left(1+p_j z\right),$$

which corresponds to the "birthday problem" in the case of nonuniform mating periods. Integral representations comparable to the ones of Chapter II are also available.

III. 5.2. Tree models. We examine here two important universal GFs associated with tree models; these provide valuable informations concerning the degree profile and the level profile of trees, while being tightly coupled with an important class of stochastic processes, the branching processes.

The major classes of trees that we have encountered so far are the unlabelled plane trees and the labelled nonplane trees, prototypes being the general Catalan trees (Chapter I) and the Cayley trees (Chapter II). In both cases, the counting generating functions satisfy a relation of the form

$$Y(z) = z\phi(Y(z)), \tag{40}$$

where the GF is either ordinary (plane unlabelled trees) or exponential (nonplane labelled trees). Corresponding respectively to the two cases, the function ϕ is determined by

$$\phi(w) = \sum_{\omega\in\Omega} u^{\omega}, \qquad \phi(w) = \sum_{\omega\in\Omega} \frac{u^{\omega}}{\omega!}, \tag{41}$$

where $\Omega \subseteq \mathbb{N}$ is the set of allowed node degrees. Meir and Moon in an important paper [**104**] have described some common properties of tree families that are determined by the Axiom (40). (For instance mean path length is of order $n\sqrt{n}$ and height is $O(\sqrt{n})$.) Following these authors, we shall call *simple variety of trees* any class whose counting GF is defined by an equation of type (40). For each of the two cases of (41), we shall write

$$\phi(w) = \sum_{j=0}^{\infty} \phi_j w^j. \tag{42}$$

First we examine the *degree profile* of trees. Such a profile is determined by the collection of parameters χ_j, where $\chi_j(\tau)$ is the number of nodes of outdegree j in τ. The variable u_j will be used to mark χ_j, that is, nodes of outdegree j. The discussion already conducted regarding recursive parameters shows that the GF $Y(z, \mathbf{u})$ satisfies the equation

$$Y(z, \mathbf{u}) = z\Phi(Y(z, \mathbf{u})) \qquad \text{where} \quad \Phi(w) = u_0\phi_0 + u_1\phi_1 w + u_2\phi_2 w^2 + \cdots .$$

Formal Lagrange inversion can then be applied to $Y(z, \mathbf{u})$, to the effect that its coefficients are given by the coefficients of the powers of Φ.

PROPOSITION III.4 (Degree profile of trees). *The number of trees of size n and degree profile $(n_0, n_1, n_2, \ldots)$ in a simple variety of trees defined by the "generator"* (42) *is*

$$Y_{n;n_0,n_1,n_2,\ldots} = \omega_n \cdot \frac{1}{n} \binom{n}{n_0, n_1, n_2, \ldots} \phi_0^{n_0} \phi_1^{n_1} \phi_2^{n_2} \cdots . \tag{43}$$

There, $\omega_n = 1$ in the unlabelled case, whereas $\omega_n = n!$ in the labelled case. The values of the n_j are assumed to satisfy the two consistency conditions: $\sum_j n_j = n$ and $\sum_j j n_j = n - 1$.

PROOF. The consistency conditions translate the fact that the total number of nodes should be n while the total number of edges should equal $n - 1$ (each node of degree j is the originator of j edges). The result follows from Lagrange inversion

$$Y_{n;n_0,n_1,n_2,\ldots} = \omega_n \cdot [u_0^{n_0} u_1^{n_1} u_2^{n_2} \cdots] \left(\frac{1}{n} [w^{n-1}] \Phi(w)^n \right),$$

to which the standard multinomial expansion applies, yielding (43).

For instance, for general Catalan trees ($\phi_j = 1$) and for Cayley trees ($\phi_j = 1/j!$) these formulæ become

$$\frac{1}{n}\binom{n}{n_0, n_1, n_2, \ldots} \qquad \text{and} \qquad \frac{(n-1)!}{0!^{n_0} 1!^{n_1} 2!^{n_2} \cdots}\binom{n}{n_0, n_1, n_2, \ldots}.$$

□

The proof above also shows the logical equivalence between the general tree counting result of Proposition III.4 and the most general case of Lagrange inversion. (This results from the fact that Φ can be specialized to any particular series.) Put otherwise, any direct proof of (43) provides a combinatorial proof of the Lagrange inversion theorem. Such direct derivations have been proposed by Raney [**119**] and are based on simple but cunning surgery performed on lattice path representations of trees (the "conjugation principle" of which a particular case is the "cycle lemma" of Dvoretzky–Motzkin [**40**]).

The next example demonstrates the usefulness of universal generating functions for investigating the profile of trees.

EXAMPLE 14. *Trees and level profile.* Given a rooted tree τ, its *level profile* is defined as the vector $(n_0, n_1, n_2, \ldots)$ where n_j is the number of nodes present at level j (i.e., at distance j from the root) in tree τ. Continuing within the framework of a simple variety of trees, we now define the quantity $Y_{n;n_0,n_1,n_2}$ to be the number of trees with size n and level profile given by the n_j. The corresponding universal GF $Y(z, \mathbf{u})$ with z marking size and u_j marking nodes at level j is expressible in terms of the fundamental "generator" ϕ:

$$Y(z, \mathbf{u}) = z u_0 \phi\left(z u_1 \phi\left(z u_2 \phi\left(z u_3 \phi(\cdots)\right)\right)\right). \tag{44}$$

We may call this a "continued ϕ-form". For instance general Catalan trees have generator $\phi(w) = (1-w)^{-1}$, so that in this case the universal GF is the continued fraction:

$$Y(z, \mathbf{u}) = \cfrac{u_0 z}{1 - \cfrac{u_1 z}{1 - \cfrac{u_2 z}{1 - \cfrac{u_3 z}{\ddots}}}}.$$

In contrast, Cayley trees are generated by $\phi(w) = e^w$, so that

$$Y(z, \mathbf{u}) = z u_0 e^{z u_1 e^{z u_2 e^{z u_3 e^{\cdots}}}},$$

which is a "continued exponential", that is, a tower of exponentials. Expanding such generating functions with respect to $u_0, u_1, \ldots$, in order gives straightforwardly:

PROPOSITION III.5 (Level profile of trees). *The number of trees of size n and level profile $(n_0, n_1, n_2, \ldots)$ in a simple variety of trees defined by the "generator" $\phi(w)$ of* (42) *is*

$$Y_{n;n_0,n_1,n_2,\ldots} = \omega_{n-1} \cdot \phi_{n_1}^{(n_0)} \phi_{n_2}^{(n_1)} \phi_{n_3}^{(n_2)} \cdots \qquad \textit{where} \quad \phi_\nu^{(\mu)} := [w^\nu]\phi(w)^\mu.$$

There, the consistency conditions are $n_0 = 1$ and $\sum_j n_j = n$.

(Note that one must always have $n_0 = 1$ for a single tree; the general formula with $n_0 \neq 1$ gives the level profile of forests.)

For instance, the counts for general Catalan trees and for Cayley trees are respectively

$$\binom{n_0+n_1-1}{n_1}\binom{n_1+n_2-1}{n_2}\binom{n_2+n_3-1}{n_3}\cdots, \quad \frac{(n-1)!}{n_0!n_1!n_2!\cdots} n_0^{n_1} n_1^{n_2} n_2^{n_3}\cdots.$$

The first of these enumerative results is due to Flajolet [**44**] and it places itself within a general combinatorial theory of continued fractions; the second one is due to Rényi and Szekeres [**120**] who developed such a formula in the context of a deep study of the distribution of height in random Cayley trees. □

▷ **18.** *"Continued forms" for path length.* The BGF of path length are obtained from the level profile MGF by means of the substitution $u_j \mapsto q^j$. For general Catalan trees and Cayley trees, this gives

$$G(z,q) = \cfrac{z}{1-\cfrac{zq}{1-\cfrac{zq^2}{\ddots}}}, \qquad T(z,q) = ze^{zqe^{zq^2e^{\cdots}}},$$

where q marks path length. The MGFs are ordinary and exponential respectively. (Combined with differentiation, such MGFs represent an attractive option for mean value analysis.) ◁

It is interesting to compare the counting results provided by universal generating functions. In a way, they contain "all" the information regarding a random object, but in a form that is not necessarily synthetic enough. Thus universal formulæ appear as offering a perspective that complements the analysis of single parameters presented in earlier sections. As we show next, they can also be used to reduce branching processes to combinatorial models.

EXAMPLE 15. *Weighted tree models and branching processes.* Consider the family $\mathcal{G}$ of all general plane trees. Let $\Lambda = (\lambda_0, \lambda_1, \ldots)$ be a system of numeric weights. The weight of a node of outdegree j is taken to be λ_j and the weight of a tree is the product of the individual weights of its nodes:

$$\pi(\tau) = \prod_{j=0}^{\infty} \lambda_j^{\chi_j(\tau)}, \tag{45}$$

with $\chi_j(\tau)$ the number of nodes of degree j in τ. One can view the weighted model of trees as a model in which a tree receives a probability proportional to $\pi(w)$. Precisely, the probability of selecting a particular tree τ under this model is, for a fixed size n

$$\mathbb{P}_{\mathcal{G}_n,\Lambda}(\tau) = \frac{\pi(\tau)}{\sum_{|\tau|=n}\pi(\tau)}. \tag{46}$$

This defines a probability measure over the set $\mathcal{G}_n$ and one can consider events and random variables under this weighted model.

The weighted model defined by (45) and (46) covers any simple variety family of trees: just replace each λ_j by the quantity ϕ_j given by the "generator' (42) of the model. For instance, plane unlabelled unary-binary trees are obtained by $\Lambda = (1,1,1,0,0,\ldots)$, while Cayley trees correspond to $\lambda_j = 1/j!$. Two *equivalence preserving transformations* are then especially important in this context:

(i) Let Λ^* be defined by $\lambda_j^* = c\lambda_j$ for some nonzero constant c. Then the weight corresponding to Λ^* satisfies $\pi^*(\tau) = c^{|\tau|}\pi(w)$. Consequently, the models associated to Λ and Λ^* are equivalent as regards (46).

(ii) Let Λ^{**} be defined by $\lambda_j^{**} = \theta^j \lambda_j$ for some nonzero constant θ. Then the weight corresponding to Λ^{**} satisfies $\pi^{**}(\tau) = c^{|\tau|-1}\pi(w)$, since $\sum_j j\chi_j(\tau) = |\tau| - 1$ for any tree τ. Thus the models Λ^{**} and Λ are again equivalent.

Each transformation has a simple effect on the generator ϕ, namely:

$$\phi(w) \mapsto \phi^*(w) = c\phi(w) \qquad \text{and} \qquad \phi(w) \mapsto \phi^{**}(w) = \phi(\theta w). \tag{47}$$

Once equipped with such equivalence transformations, it becomes possible to describe probabilistically the process that generates trees according to a weighted model. Assume that $\lambda_j \geq 0$ and that the λ_j are summable. Then the normalized quantities

$$p_j = \frac{\lambda_j}{\sum_j \lambda_j}$$

form a probability distribution over $\mathbb{N}$. By the first equivalence-preserving transformation the model induced by the weights p_j is the same as the original model induced by the λ_j.

Such a model defined by nonnegative weights $\{p_j\}$ summing to 1 is nothing but the classical model of *branching processes* (also known as Galton-Watson processes) ; see [**7**]. In effect, a realization T of the branching process is classically defined by the two rules: (i) produce a root node of degree j with probability p_j; (ii) if $j \geq 1$, attach to the root node a collection $T_1, \ldots, T_j$ of independent realizations of the process. This may be viewed as the development of a "family" stemming from a common ancestor where any individual has probability p_j of giving birth to j descendants. Clearly, the probability of obtaining a particular finite tree τ has probability $\pi(\tau)$, where π is given by (45) and the weights are $\lambda_j = p_j$. The generator

$$\phi(w) = \sum_{j=0}^{\infty} p_j w^j$$

is then nothing but the probability generating function of (one-generation) offspring, with the quantity $\mu = \phi'(1)$ being its mean size.

For the record, we recall that branching processes can be classified into three categories depending on the values of μ:

Subcriticality: when $\mu < 1$, the random tree produced is finite with probability 1 and its expected size is also finite.

Criticality: when $\mu = 1$, the random tree produced is finite with probability 1 but its expected size is infinite.

Supercriticality: when $\mu > 1$, the random tree produced is finite with probability strictly less than 1.

From the discussion of equivalence transformations (47), there result that, regarding trees of a *fixed size* n, there is complete equivalence between all branching processes with generators of the form

$$\phi_\theta(w) = \frac{\phi(\theta w)}{\phi(\theta)}.$$

(Such families of related functions are known as "exponential families" in probability theory.) In this way, one may always regard at will the random tree produced by a weighted model of some fixed size n as originating from a branching process of subcritical, critical, or supercritical type conditioned upon the size of the total progeny.

Finally, take a set $\mathcal{S} \subseteq \mathcal{G}$ for which the universal generating function of $\mathcal{S}$ with respect to the degree profile is available,

$$S(z, u_0, u_1, \ldots) = \sum_{\tau \in \mathcal{S}} z^{|\tau|} \left(u_0^{\chi_0(\tau)} u_1^{\chi_1(\tau)} \cdots \right).$$

Then, for a system of weights Λ, one has

$$S(z, \lambda_0, \lambda_1, \ldots) = \sum_{\tau \in \mathcal{S}} \pi(\tau) z^{|\tau|}.$$

Thus, the probability that a weighted tree of size n belongs to $\mathcal{S}$ becomes accessible by extracting the coefficient of z^n. This applies *a fortiori* to branching processes as well. In summary, *the analysis of parameters of trees of size n under either weighted models or branching process models derives from substituting weights or probability values inside the corresponding combinatorial generating functions.* □

The reduction of combinatorial tree models to branching processes has been pursued most notably by the "Russian School": see especially the books by Kolchin [**90, 91**] and references therein. Conversely, symbolic-combinatorial methods may be viewed as a systematic way of obtaining equations relative to characteristics of branching processes. We do not proceed further along these lines as this would take us outside of the scope of the present book.

▷ **19.** *Catalan trees, Cayley trees, and branching processes.* Catalan trees of size n are defined by the weighted model in which $\lambda_j \equiv 1$, but also equivalently by $\widehat{\lambda}_j = c\theta^j$, for any $c > 0$ and $\theta \leq 1$. In particular they coincide with the random tree produced by the critical branching process with offspring probabilities that are geometric: $p_j = 1/2^{j+1}$.

Cayley trees are *a priori* defined by $\lambda_j = 1/j!$. They can be generated by the critical branching process with Poisson probabilities, $p_j = e^{-1}/j!$, and more generally with an arbitrary Poisson distribution $p_j = e^{-\lambda}\lambda^j/j!$. ◁

III. 6. Additional constructions

We discuss here additional constructions already examined in earlier chapters, namely pointing and substitution (Section III. 6.1) as well as order constraints (Section III. 6.2) on the one hand, implicit structures (Section III. 6.3) on the other hand. Given the that basic translation mechanisms can be directly adapted to the multivariate realm, such extensions involve basically no new concept and the methods of Chapters I and II can be recycled. In Section III. 6.4, we revisit the classical principle of inclusion-exclusion under a generating function perspective. In this light, the principle appears as a typically multivariate device well-suited to enumerating objects according the number of occurrences of sub-configurations.

III. 6.1. Pointing and substitution. Let $\langle \mathcal{F}, \chi \rangle$ be a class–parameter pair, where χ is multivariate of dimension $r \geq 1$ and let $F(\mathbf{z})$ be the MGF associated to it in the notations of (12) and (22). In particular $z_0 = z$ marks size, and z_k marks the component j of the multiparameter k. Pick up a variable $x \equiv z_j$ for some j with $0 \leq j \leq r$. Then since

$$x\partial_x (s^a t^b x^f) = f \cdot (s^a t^b x^f),$$

the interpretation of the operator θ_x is immediate; it means "pick up in all possible ways in objects of $\mathcal{F}$ a configuration marked by x and point to it". For instance, if $F(z, u)$ is the BGF of trees where z marks size and u marks leaves, then $\theta_u F(z, u) = u\partial_u F(z, u)$ enumerates trees with one distinguished leaf.

▷ **20.** *Pointing-erasing and the combinatorics of Taylor's formula.* The derivative operator ∂_x corresponds combinatorially to a "pointing-erasing" operation: select in all possible ways an atom marked by x and make it transparent to x-marking (e.g., by replacing it by a neutral object). The operator

$$\mu^k[f](x) := \frac{1}{k!}\partial_x^k f(x),$$

then corresponds to picking up in all possible way a subset of k configurations marked by x and unmarking them. The identity (Taylor's formula)

$$f(x+y) = \sum_{k\geq 0}\left(\frac{1}{k!}\partial_x^k f(x)\right) y^k$$

can then receive a simple combinatorial interpretation: Given a population of individuals ($\mathcal{F}$ enumerated by f), form the bicoloured population of individuals enumerated by $f(x+y)$, where each atom of each object can be repainted either in x-colour or y-colour; this is equivalent to deciding a priori for each individual to repaint k of its atoms from x to y, this for all possible values of $k \geq 0$. Taylor's formula follows. ◁

Similarly, the substitution $x \mapsto S(\mathbf{z})$ in a GF $F(z)$, where $S(\mathbf{z})$ is the MGF of a class $\mathcal{S}$, means attaching an object of type $\mathcal{S}$ to configurations marked by the variable z in $\mathcal{F}$. We refrain from giving detailed definitions (that would be somewhat clumsy and uninformative) as the process is better understood by practice than by long formal developments. Justification in each particular case is normally easily obtained by returning to the combinatorial definition of generating functions as "reduced images" of combinatorial classes.

EXAMPLE 16. *Constrained integer compositions and "slicing".* This example illustrates variations around the substitution scheme. Consider compositions of integers where successive summands have sizes that are constrained to belong to a fixed set $\mathcal{R} \subseteq \mathbb{N}^2$. For instance, the relations

$$\mathcal{R}_1 = \{(x,y) \mid 1 \leq x \leq y\}, \quad \mathcal{R}_2 = \{(x,y) \mid 1 \leq y \leq 2x\},$$

will correspond to weakly increasing summands in the case of $\mathcal{R}_1$ and to summands that can at most double at each stage in the case of $\mathcal{R}_2$. In the "ragged landscape" representation of compositions, this means considering diagrams of unit cells aligned in columns along the horizontal axis, with successive columns obeying the constraint imposed by $\mathcal{R}$.

Let $F(z,u)$ be the BGF of such $\mathcal{R}$–restricted compositions, where z marks total sum and u marks the value of the last summand, that is, the height of the last column. The function $F(z,u)$ satisfies an equation of the form

$$F(z,u) = f(zu) + \left(\mathcal{L}\left[F(z,u)\right]\right)_{u\mapsto zu}, \tag{48}$$

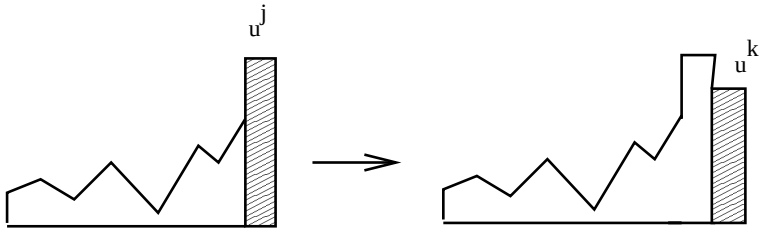


FIGURE 10. The technique of "adding a slice" for enumerating constrained compositions.

where $f(z)$ is the generating function of the one-column objects and $\mathcal{L}$ is a linear operator over formal series in u given by

$$\mathcal{L}[u^j] := \sum_{(j,k)\in\mathcal{R}} u^k. \tag{49}$$

In effect, Equation (48) describes inductively objects as comprising either one column ($f(zu)$) or else being formed by adding a new column to an existing one. In the latter case, the last column added has a size k that must be such that $(j,k) \in \mathcal{R}$, if it was added after a column of size j, and it will contribute $u^k z^k$ to the BGF $F(z,u)$; this is precisely what (49) expresses. In particular, $F(z,1)$ gives back the enumeration of $\mathcal{F}$–objects irrespective of the size of the first column.

For a rule $\mathcal{R}$ that is "simple enough", the basic equation (48) will often involve a substitution. Let us first rederive in this way the enumeration of partitions. We take $\mathcal{R} = \mathcal{R}_1$ and assume that the first column can have any positive size. Compositions into increasing summands are then the same as partitions. Since

$$L[u^j] = u^j + u^{j+1} + u^{j+2} + \cdots = \frac{u^j}{1-u},$$

the function $F(z,u)$ satisfies a functional equation involving a substitution,

$$F(z,u) = \frac{zu}{1-zu} + \frac{1}{1-zu} F(z,zu). \tag{50}$$

This relation iterates: *any linear functional equation of the substitution type*

$$\phi(u) = \alpha(u) + \beta(u)\phi(\sigma(u))$$

is solved formally by

$$\phi(u) = \alpha(u) + \beta(u)\alpha(\sigma(u)) + \beta(u)\beta(\sigma(u))\alpha(\sigma^{\langle 2\rangle}(u)) + \cdots, \tag{51}$$

where $\sigma^{\langle j\rangle}(u)$ designates the jth iterate of u.

Returning to compositions into increasing summands, that is, partitions, the turnkey solution (51) gives, upon iterating on the second argument and with the first argument being treated as a parameter:

$$F(z,u) = \frac{zu}{1-zu} + \frac{z^2u}{(1-zu)(1-z^2u)} + \frac{z^3u}{(1-zu)(1-z^2u)(1-z^3u)} + \cdots. \tag{52}$$

Equivalence with the alternative form

$$F(z,u) = zu + \frac{z^2u^2}{1-z} + \frac{z^3u^3}{(1-z)(1-z^2)} + \frac{z^4u^4}{(1-z)(1-z^2)(1-z^3)}\cdots \tag{53}$$

is then easily verified from (50) upon expanding $F(z,u)$ as a series in u and applying the method of indeterminate coefficients to the form $(1-zu)F(z,u) = zu + F(z,zu)$. The presentation (53) is furthermore consistent with the treatment of partitions given in Chapter I since the quantity $[u^k]F(z,u)$ clearly represents the OGF of partitions whose smallest summand is 1 and whose largest summand is k. (In passing, the equality between (52) and (53) is a shallow but curious identity that is quite typical of the area.)

This same method has been applied in [**54**] to compositions satisfying condition $\mathcal{R}_2$ above. In this case, successive summands are allowed to double at most at each stage. The associated linear operator is

$$\mathcal{L}[u^j] = u + \cdots + u^{2j} = u\frac{1-u^{2j}}{1-u}.$$

For simplicity, it is assumed that the first column has size 1. Thus, F satisfies a functional equation of the substitution type:

$$F(z,u) = zu + \frac{zu}{1-zu}\left(F(z,1) - F(z,z^2u^2)\right).$$

This can be solved by means of the general iteration mechanism (51), treating momentarily $F(z,1)$ as a known quantity: with $a(u) := zu + F(z,1)/(1-zu)$, one has

$$F(z,u) = a(u) - \frac{zu}{1-zu}a(z^2u^2) + \frac{zu}{1-zu}\frac{z^2u^2}{1-z^2u^2}a(z^6u^4) - \cdots.$$

Then, the substitution $u = 1$ in the solution becomes permissible. Upon solving for $F(z,1)$, one eventually gets the somewhat curious GF for compositions satisfying $\mathcal{R}_2$:

$$F(z,1) = \frac{\sum_{j\geq 1}(-1)^{j-1}Q_{j-1}(z)z^{2^{j+1}-j-2}}{\sum_{j\geq 0}(-1)^{j}Q_{j}(z)z^{2^{j+1}-j-2}}$$
$$\text{where} \quad Q_j(z) = (1-z)(1-z^3)(1-z^7)\cdots(1-z^{2^j-1}).$$

The sequence of coefficients starts as $1,1,2,3,5,9,16,28,50$ and is *EIS* **A002572**: it represents for instance the number of possible level profiles of binary trees, or equivalently the number of partitions of 1 into summands of the form $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ (this is related to the number of solutions to Kraft's inequality). See [**54**] for details including very precise asymptotic estimates and Tangora's paper for relations to algebraic topology. □

The reason for presenting this method in some detail is that it is very general. It has been in particular employed to derive a number of original enumerations of polyominos by area, a topic of interest in some branches of statistical mechanics: for instance, the book by Janse van Regsburg [**144**] discusses many applications of such lattice models to polymers and vesicles. See Bousquet-Mélou's review paper [**23**] for a methodological perspective. Some of the origins of the method point to Pólya in the 1930's, see [**114**], and independently to Temperley [**141**, pp. 65–67].

▷ **21.** *Carlitz compositions.* Let $\mathcal{K}$ be the class of compositions such that pairs of adjacent summands are always distinct. These can be generated by the operator $\mathcal{L}[u^j] = u(1-u)^{-1} - u^j$, from which the OGF follows. Alternatively, one may start from Smirnov words (p. 152 below) and effect the substitution $v_j \mapsto z^j$, so that

$$K(z) = \left(1 - \sum_{j=1}^{\infty}\frac{z^j}{1+z^j}\right)^{-1}.$$

For maximal summand $\leq r$, replace ∞ by r in the formula above. (Such compositions have been introduced by Carlitz in 1976; see the paper by Knopfmacher and Prodinger [**82**] for early references and asymptotic properties.) ◁

III. 6.2. Order constraints. We refer in this subsection to the discussion of order constraints in labelled products that has been given in Chapter II. We recall that the modified labelled product

$$\mathcal{A} = (\mathcal{B}^{\square} \star \mathcal{C})$$

only includes the elements of $(\mathcal{B} \star \mathcal{C})$ such that the minimal label lies in the $\mathcal{A}$ component. Once more the univariate rules generalize verbatim for parameters that are inherited and the corresponding exponential MGFs are related by

$$A(z,\mathbf{u}) = \int_0^z (\partial_t B(t,\mathbf{u}))\cdot C(t,\mathbf{u})\,dt.$$

valley:	$\sigma_{i-1} > \sigma_i < \sigma_{i+1}$	leaf node (u_0)
double rise:	$\sigma_{i-1} < \sigma_i < \sigma_{i+1}$	unary right-branching (u_1)
double fall:	$\sigma_{i-1} > \sigma_i > \sigma_{i+1}$	unary left-branching (u_1')
peak:	$\sigma_{i-1} < \sigma_i > \sigma_{i+1}$	binary node (u_2)

FIGURE 11. Local order patters in a permutation and the four types of nodes in the corresponding increasing binary tree.

To illustrate this multivariate extension, we shall consider a quadrivariate statistic on permutations.

EXAMPLE 17. *Local order patterns in permutations.* An element σ_i of a permutation written $\sigma = \sigma_1, \ldots, \sigma_n$ when compared to its immediate neighbours can be categorized into one of four types summarized in the first two columns of Figure 11. The correspondence with binary increasing trees described in Example 16 of Chapter II then shows the following: peaks and valleys correspond to binary nods and leaves, respectively, while double rises and double falls are associated with right-branching and left-branching unary nodes. Let u_0, u_1, u_1', u_2 be markers for the number of nodes of each type, as summarized in Figure 11. Then the exponential MGF of increasing trees under this statistic satisfies

$$\frac{\partial}{\partial z} I(z, \mathbf{u}) = u_0 + (u_1 + u_1') I(z, \mathbf{u}) + u_2 I(z, \mathbf{u})^2.$$

This is solved by separation of variables as

$$I(z, \mathbf{u}) = \frac{\delta}{u_2} \frac{v_1 + \delta \tan(z\delta)}{\delta - v_1 \tan(z\delta)} - \frac{v_1}{u_2}, \tag{54}$$

where the following abbreviations are used:

$$v_1 = \frac{1}{2}(u_1 + u_1'), \qquad \delta = \sqrt{u_0 u_2 - v_1^2}.$$

One has

$$I = u_0 z + u_0 (u_1 + u_1') \frac{z^2}{2!} + u_0 ((u_1 + u_1')^2 + 2u_0 u_2) \frac{z^3}{3!},$$

which agrees with the small cases. This calculation is consistent with what has been found in Chapter II regarding the EGF of all permutations and of alternating permutations,

$$\frac{1}{1-z}, \quad \tan(z),$$

that derive from the substitutions $\{u_0 = u_1 = u_1' = u_2 = 1\}$ and $\{u_0 = u_2 = 1, u_1 = u_1' = 0\}$, respectively. The substitution $\{u_0 = u_1 = u, u_1' = u_2 = 1\}$ gives the BGF of Eulerian numbers (61) derived below by other means.

By specialization of the tetravariate GF, there results that, in a tree of size n the mean number of nodes of nullary, unary, or binary type is asymptotic to $n/3$, with a variance that is $O(n)$, thereby ensuring concentration of distribution. □

A similar analysis yields path length. It is found that a random increasing binary tree of size n has mean path length

$$2n \log n + O(n).$$

Contrary to what the uniform combinatorial model give, such tree tend to be rather well balanced, and a typical branch is only about 38.6% worse than in a perfect binary tee. This

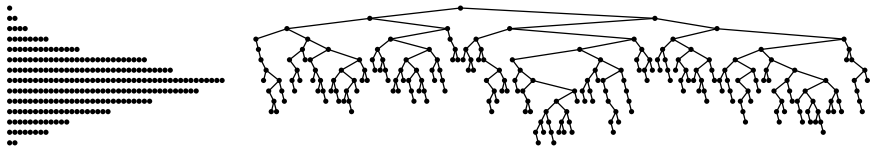


FIGURE 12. The level profile of a random increasing binary tree of size 256. (Compare with Figure 9 for binary trees under the uniform Catalan statistic.)

fact applies to binary search trees and it justifies the performance of such trees to be quite good when applied to random data [**86, 102, 130**] or subjected to randomization [**123**].

III. 6.3. Implicit structures. Here again, we note that equations involving sums and products, either labelled or not, are easily solved just like in the univariate case. The same applies for the sequence construction and for the set construction, especially in the labelled case—refer to the corresponding sections of Chapters I and II. Again, the process is best understood by examples.

Suppose for instance one wants to enumerate connected labelled graphs by the number of nodes (marked by z) and the number of edges (marked by u). The class $\mathcal{K}$ of connected graphs and the class $\mathcal{G}$ of all graphs are related by the set construction,

$$\mathcal{G} = \mathfrak{P}\{\mathcal{K}\},$$

meaning that every graph decomposes uniquely into connected components. The corresponding exponential BGFs then satisfy

$$G(z,u) = e^{K(z,u)} \qquad \text{implying} \qquad K(z,u) = \log G(z,u),$$

since the number of edges in a graph is inherited (additively) from the corresponding numbers in connected components. Now, the number of graphs of size n having k edges is $\binom{n(n-1)/2}{k}$, so that

$$K(z,u) = \log\left(1 + \sum_{n=1}^{\infty} (1+u)^{n(n-1)/2} \frac{z^n}{n!}\right). \tag{55}$$

This formula, which appears as a refinement of the univariate formula of Chapter II, then simply reads: *connected graphs are obtained as components (the* $\log$ *operator) of general graphs, where a general graph is determined by the presence or absence of an edge (corresponding to* $(1+u)$*) between any pair of nodes (the exponent* $n(n-1)/2$*).*

Pulling out information out of the formula (55) is however not obvious due to the alternation of signs in the expansion of $\log(1+w)$ and due to the strongly divergent character of the involved series. As an aside, we note here that the quantity

$$\widehat{K}(z,u) = K\left(\frac{z}{u}, u\right)$$

enumerates connected graphs according to size (marked by z) and excess (marked by u) of the number of edges over the number of nodes. This means that the results of Section 5.3 of Chapter II obtained by Wright's decomposition can be rephrased as the expansion (within $\mathbb{C}(u)[\![z]\!]$):

$$\begin{aligned} \log\left(1+\sum_{n=1}^{\infty}(1+u)^{n(n-1)/2}\frac{z^n u^{-n}}{n!}\right) &= \frac{1}{u}W_{-1}(z)+W_0(z)+\cdots \\ &= \frac{1}{u}\left(T-\frac{1}{2}T^2\right)+\left(\frac{1}{2}\log\frac{1}{1-T}-\frac{1}{2}T-\frac{1}{4}T^2\right)+\cdots, \end{aligned} \tag{56}$$

with $T \equiv T(z)$. See Temperley's early works [**140, 141**] as well as the "giant paper on the giant component" [**77**] and the paper [**55**] for direct derivations that eventually constitute analytic alternatives to Wright's combinatorial approach.

EXAMPLE 18. *Smirnov words.* Following the terminology of Jackson and Goulden [**68**], a Smirnov word is a word that has no consecutive equal letters. Let $\mathcal{W} = \mathfrak{S}\{\mathcal{A}\}$ be the set of words over the alphabet $\mathcal{A} = \{a_1, \ldots, a_r\}$ of cardinality r, and $\mathcal{X}$ be the set of Smirnov words. Let also u_j mark the number of occurrences of the jth letter in a word. One has

$$W(z, \mathbf{u}) = \frac{1}{1-(v_1+\cdots+v_r)} \quad \text{with} \quad v_j = zu_j.$$

Start from a Smirnov word and substitute to any letter a_j that appears in it an arbitrary nonempty sequence of letters a_j. When this operation is done at all places of a Smirnov word, it gives rise to an unconstrained word. Conversely, any word is associated to a unique Smirnov word by collapsing into single letters maximal groups of contiguous equal letters. In other terms, words derive from Smirnov words by a simultaneous substitution that we represent figuratively as

$$\mathcal{W} = \mathcal{S}\big[a_1 \mapsto \mathfrak{S}_{\geq 1}\{a_1\},\ \ldots\ , a_r \mapsto \mathfrak{S}_{\geq 1}\{a_r\}\big].$$

There results the relation

$$W(v_1, \ldots, v_r) = S\left(\frac{v_1}{1-v_1},\ \ldots\ , \frac{v_r}{1-v_r}\right). \tag{57}$$

This relation determines the MGF $S(v_1, \ldots, v_r)$ *implicitly*. Indeed, since the inverse function of $v/(1-v)$ is $v/(1+v)$, one finds

$$S(v_1, \ldots, v_r) = W\left(\frac{v_1}{1+v_1},\ \ldots\ , \frac{v_r}{1+v_r}\right). \tag{58}$$

For instance, if we set $v_j = z$, that is, we "forget" the composition of the words into letters, we get the OGF of Smirnov word counted according to length as

$$\frac{1}{1-r\frac{z}{1+z}} = \frac{1+z}{1-(r-1)z} = 1+\sum_{n\geq 1} r(r-1)^{n-1}z^n.$$

This is consistent with elementary combinatorics since a Smirnov word of length n is determined by the choice of its first letter (r possibilities) followed by a sequence of $n-1$ choices constrained to avoid one letter amongst r (and corresponding to $r-1$ possibilities for each position). The interest of (58) is to apply equally well to the Bernoulli model where letters may receive unequal probabilities and where a direct combinatorial argument does not appear to be easy: it suffices to perform the substitution $v_j \mapsto p_j z$ in this case.

From these developments, one can next build the GF of words that never contain more than m consecutive equal letters. It suffices to effect in (58) the substitution $v_j \mapsto v_j +$

$\cdots + v_j^m$. In particular for the univariate problem (or, equivalently, the case where letters are equiprobable), one finds the OGF

$$\frac{1}{1 - r\dfrac{z\frac{1-z^m}{1-z}}{1 + z\frac{1-z^m}{1-z}}} = \frac{1 - z^{m+1}}{1 - rz + (r-1)z^m}.$$

This extends to an arbitrary alphabet the analysis of single runs and double runs in binary words that was performed in Section 4 of Chapter I. Naturally, this approach applies equally well to nonuniform letter probabilities and to a collection of different run length upperbounds depending on each particular letter. For instance, this topic is pursued in several works of Karlin and coauthors (see, e.g., [**106**]), themselves motivated by biological applications. □

III. 6.4. Inclusion-Exclusion. Inclusion-exclusion is a familiar type of reasoning rooted in elementary mathematics. We re-examine it here in the perspective of multivariate generating functions, where it essentially reduces to a combined use of substitution and implicit definitions.

Let $\mathcal{E}$ be a set endowed with a real or complex valued measure $|\cdot|$ in such a way that, for $A, B \subset \mathcal{E}$, there holds

$$|A \cup B| = |A| + |B| \qquad \text{whenever} \qquad A \cap B = \emptyset.$$

Thus, $|\cdot|$ is an additive measure, typically taken as set cardinality or a discrete probability measure on $\mathcal{E}$. The more general formula

$$|A \cup B| = |A| + |B| - |AB| \qquad \text{where} \quad AB := A \cap B,$$

follows immediately. What is called the *inclusion-exclusion principle* or *sieve formula* is the following multivariate generalization, for an arbitrary family $A_1, \ldots, A_r \subset \mathcal{E}$:

(59)
$$\begin{aligned} |A_1 \cup \cdots \cup A_r| &\equiv \left|\mathcal{E} \setminus (\overline{A}_1 \overline{A}_2 \cdots \overline{A}_r)\right| \qquad \text{where} \quad \overline{A} := \mathcal{E} \setminus A \\ &= \sum_{1 \le i \le r} |A_i| - \sum_{1 \le i_1 < i_2 \le r} |A_{i_1} A_{i_2}| + \cdots + (-1)^{r-1} |A_1 A_2 \cdots A_r|. \end{aligned}$$

The easy proof by induction results from elementary properties of the boolean algebra formed by the subsets of $\mathcal{E}$; see, e.g., [**28**, Ch. IV].) An alternative formulation results from setting $B_j = \overline{A}_j$, $\overline{B}_j = A_j$:

$$(60)\quad |B_1 B_2 \cdots B_r| = |\mathcal{E}| - \sum_{1 \le i \le r} |\overline{B}_i| + \sum_{1 \le i_1 < i_2 \le r} |\overline{B}_{i_1} \overline{B}_{i_2}| - \cdots + (-1)^r |\overline{B}_1 \overline{B}_2 \cdots \overline{B}_r|.$$

In terms of measure, this equality quantifies the set of objects satisfying *exactly* a collection of *simultaneous* conditions (all the B_j) in terms of those that violate *at least some* of the conditions (the members of the $\overline{B}_j$).

Here is a textbook example of an inclusion–exclusion argument, namely, the enumeration of *derangements*. Recall that a derangement is a permutation σ such that $\sigma_i \neq i$, for all i. Fix $\mathcal{E}$ as the set of all permutations of $[1, n]$, take the measure $|\cdot|$ to be set cardinality, and let B_i be the subset of permutations in $\mathcal{E}$ associated to the property $\sigma_i \neq i$. (There are consequently $r = n$ conditions.) Thus, B_i means having no fixed point at i, while $\overline{B}_i$ means having a fixed point at the *distinguished* value i. Then, the left hand side of (60) is the number of permutations that are derangements, that is, D_n. As regards the right hand side, the kth sum comprises itself $\binom{n}{k}$ terms couting possibilities attached to the

choices of indices $i_1 < \cdots < i_k$; each such choice is associated to a factor $\overline{B}_{i_1} \cdots \overline{B}_{i_k}$ that describes all permutations with fixed points at the distinguished points $i_1, \ldots, i_k$ (i.e., $\sigma(i_1) = i_1, \ldots, \sigma_{i_k} = i_k$). Clearly, $|\overline{B}_{i_1} \cdots \overline{B}_{i_k}| = (n-k)!$. Therefore one has

$$D_n = n! - \binom{n}{1}(n-1)! + \binom{n}{2}(n-2)! - \cdots + (-1)^n \binom{n}{n} 0!,$$

which rewrites into the more familiar form

$$\frac{D_n}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!}.$$

This gives an elementary derivation of the derangement numbers already encountered in Chapter II.

The derivation above is perfectly fine but carrying it out on complex examples may represent somewhat of a challenge. In contrast, as we now explain, there exists a parallel approach based on multivariate generating functions, which is technically easy to deal with and has great versatility.

Let us now reexamine derangements in a generating function perspective. Consider the set $\mathcal{P}$ of all permutations and build a superset $\mathcal{Q}$ as follows. The set $\mathcal{Q}$ is comprised of permutations in which an arbitrary number of fixed points—some, maybe none, not necessarily all—have been *distinguished*. (This corresponds to aribitrary products of the $\overline{B}_j$ in the argument above.). For instance $\mathcal{Q}$ contains elements like

$$\underline{1}, 3, 2, \quad 1, 3, 2, \quad \underline{1}, 2, 3, \quad 1, \underline{2}, \underline{3}, \quad \underline{1}, 2, \underline{3}, \quad \underline{1}, \underline{2}, \underline{3},$$

where distinguished fixed points are underlined. Clearly, if one removes the distinguished elements of a $\gamma \in \mathcal{Q}$, what is left constitutes an arbitrary permutation of the remaining elements. One then has

$$\mathcal{Q} \cong \mathcal{U} \star \mathcal{P},$$

where $\mathcal{U}$ denotes the class of urns that are sets of atoms. In particular, the EGF of $\mathcal{Q}$ is $Q(z) = e^z/(1-z)$. What we've just done is enumerating the quantities that appear in (60), but with the signs "wrong", i.e., all positive.

Introduce now the variable v to mark the distinguished fixed points in objects of $\mathcal{Q}$. The exponential BGF is then

$$Q(z, v) = e^{vz} \frac{1}{1-z}.$$

Let $P(z, u)$ be the BGF of permutations where u marks the number of fixed points. (Let us ignore momentarily the fact that $P(z, u)$ is otherwise known.) Permutations with *some* fixed points distinguished are generated by the substitution $u \mapsto 1 + v$ inside $P(z, u)$. In other words one has the fundamental inclusion-exclusion relation

$$Q(z, v) = P(z, 1 + v).$$

This is then easily solved as

$$P(z, u) = Q(z, u - 1),$$

so that knowledge of (the easy) Q gives (the harder) P. For the case at hand, this yields

$$P(z, u) = \frac{e^{(u-1)z}}{1-z}, \qquad P(z, 0) = D(z) = \frac{e^{-z}}{1-z},$$

and, in particular, the EGF of derangements has been retrieved. Note that the sought $P(z, 0)$ comes out as $Q(z, -1)$, so that signs corresponding to the sieve formula (60) have now been put "right", i.e., alternating.

The process employed for derangements is clearly very general. It is a generating function analogue of the inclusion-exclusion principle: counting objects that satisfy a number of *simultaneous* constraints is reduced to counting objets that violate *some* of the constraints at distinguished "places"—the latter is usually a simpler problem The generating function analogue of inclusion exclusion is then simply the substitution $v \mapsto u-1$, if a bivariate GF is sought, or $v \mapsto -1$ in the univariate case.

The book by Goulden and Jackson [**68**, pp. 45–48] describes a useful formalization of the inclusion process operating on MGFs. Conceptually, it combines substitution and implicit definitions. Once again, the *modus operandi* is best grasped through examples.

EXAMPLE 19. *Rises and ascending runs in permutations.* A *rise* in a permutation $\sigma = \sigma_1 \cdots \sigma_n$ is a pair of consecutive elements σ_i, σ_{i+1} satisfying $\sigma_i < \sigma_{i+1}$. The problem is to determine the number $A_{n,k}$ of permutations of size having exactly k rises together with the BGF $A(z,u)$.

Guided by the inclusion-exclusion principle, we tackle the easier problem of enumerating permutations with *distinguished* rises, of which the set is denoted by $\mathcal{B}$. For instance, $\mathcal{B}$ contains elements like

$$2\,1\,\boxed{3\nearrow 4\nearrow 8\nearrow 9\nearrow 11}\,15\,12\,\boxed{5\nearrow 10}\,13\,7\,14,$$

where those rises that are distinguished are represented by arrows. (Note that some rises may *not* be distinguished.) Maximal sequences of adjacent distinguished rises (boxed in the representation) will be called *clusters*. Then, $\mathcal{B}$ can be specified by the sequence constuction applied to atoms ($\mathcal{Z}$) and clusters ($\mathcal{C}$) as

$$\mathcal{B} = \mathfrak{S}\{\mathcal{Z} + \mathcal{C}\}, \qquad \text{where} \quad \mathcal{C} = \mathfrak{P}_{\geq 2}\{\mathcal{Z}\},$$

since a cluster is an ordered sequence, or equivalently a set, having furthermore at least two elements. This gives the EGF of $\mathcal{B}$ as

$$B(z) = \frac{1}{1-(z+(e^z-1-z))} = \frac{1}{2-e^z},$$

which happens to coincide with the EGF of surjections.

For inclusion-exclusion purposes, we need the BGF of $\mathcal{B}$ with v marking the number of distinguished rises. A cluster of size k contains $k-1$ rises, so that

$$B(z,v) = \frac{1}{1-(z+(e^{zv}-1-zv)/v)} = \frac{v}{v+1-e^{zv}}.$$

Now, the usual argument applies: the BGF $A(z,u)$ satisfies $B(z,v) = A(z,1+v)$, so that $A(z,u) = B(z,u-1)$, which yields the particularly simple form

$$A(z,u) = \frac{u-1}{u-e^{z(u-1)}}. \tag{61}$$

In particular, this GF expands as

$$A(z,u) = 1 + z + (u+1)\frac{z^2}{2!} + (u^2+4u+1)\frac{z^3}{3!} + (u^3+11u^2+11u+1)\frac{z^4}{4!} + \cdots.$$

The coefficients $A_{n,k}$ are known as the *Eulerian numbers*. In combinatorial analysis, these numbers are almost as classic as the Stirling numbers. A detailed discussion of their properties is to be found in classical treatises like [**28**] or [**71**]. (From Eq. (61), permutations without rises are enumerated by $B(z,-1) = e^z$, an altogether obvious result.)

Moments derive easily from an expansion of (61) at $u = 1$, which gives

$$A(z,u) = \frac{1}{1-z} + \frac{1}{2}\frac{1}{(1-z)^2}(u-1) + \frac{1}{12}\frac{z^3(2+z)}{(1-z)^3}(u-1)^2 + \cdots .$$

In particular: *the mean of the number of rises in a random permutation of size n is $\frac{1}{2}(n-1)$ and the variance is $\sim \frac{1}{12}n$, ensuring concentration of distribution.*

The same method applies to the enumeration of *ascending runs*: for a fixed parameter ℓ, an ascending run (of order ℓ) is a sequence of consecutive elements $\sigma_i\sigma_{i+1}\cdots\sigma_{i+\ell-1}$ such that $\sigma_i < \sigma_{i+1} < \cdots < \sigma_{i+\ell-1}$. An inclusion-exclusion similar to the one seen above shows that the BGF of the number of ascending runs of order ℓ in permutations is

$$B^{\langle\ell\rangle}(z,u) = \left(\frac{1}{1-(z+(e^{zv}-e_{\ell-1}(zv))/v^{\ell-1})}\right)_{v=u-1}, \qquad e_r(z) := \sum_{j=0}^{r}\frac{z^j}{j!}. \tag{62}$$

(Rises correspond to $\ell = 2$.)

The BGF (62) can be exploited to determine quantitative information on long runs in permutations. First, an expansion at $u = 1$ shows that the mean number of ascending runs is $(n-\ell)/\ell!$ exactly, as soon as $n \geq \ell$. This entails that, if $n = o(\ell!)$, the probability of finding an ascending run of order ℓ tends to 0 as $n \to \infty$. What is used in passing in this argument is the general fact that for a discrete variable X with values in $0, 1, 2, \ldots$, one has (with Iverson's notation)

$$\mathbb{P}(X \geq 1) = \mathbb{E}([\![X \geq 1]\!]) = \mathbb{E}(\min(X,1)) \leq \mathbb{E}(X).$$

An inequality in the converse direction results from the second moment method. In effect, the variance of the number of ascending runs is found to be of the exact form $\alpha_\ell n + \beta_\ell$ where α_ℓ is essentially $1/\ell!$ and β_ℓ is of comparable order. Thus, by Chebyshev's inequalities, concentration of distribution holds as long as ℓ is such that $\ell! = o(n)$. In this case, with high probability (i.e., with probability tending to 1 as n tends to ∞), there are many ascending runs of order ℓ.

What has been found here is a fairly sharp threshold phenomenon:

> *In a random permutation of size n, with high probability, an ascending run of order ℓ is present if $\ell! = o(n)$ but not present if $n = o(\ell!)$. Let $\ell_0(n)$ be the largest integer such that $\ell_0! \leq n$, so that $\ell_0(n) \sim (\log n)/\log\log n$. Then, with high probability, there is no ascending run of length $\ell_0 + 1$ but at least one ascending run (and in fact many) of length $\ell_0 - 1$.*

□

Many variations on the theme of rises and ascending runs are clearly possible. Local order patterns in permutations have been intensely researched, notably by Carlitz in the 1970's. Goulden and Jackson [**68**, Sec. 4.3] offer a general theory of patterns in sequences and permutations. Special permutations patterns associated with binary increasing trees are also studied by Flajolet, Gourdon, and Martínez [**48**] (by combinatorial methods) and Devroye [**39**] (by probabilistic arguments). On another register, the longest ascending run has been found above to be of order $(\log n)/\log\log n$ in probability. The superficially resembling problem of analysing the length of the *longest increasing sequence* in random permutations (elements must be in ascending order but need not be adjacent) has attracted a lot of attention, but is considerably harder. This quantity is $\sim 2\sqrt{n}$ on average and in probability, as shown by a penetrating analysis of the shape of random Young tableaus due to Logan, Shepp, Vershik, and Kerov [**97, 146**] Solving a problem open for over 20 years,

Baik, Deift, and Johansson [**8**] have eventually determined its limiting distribution. (The undemanding survey by Aldous and Diaconis [**2**] discusses some of the background of this problem.)

▷ **22.** *Increasing subsequences in permutations.* This exercise is based on Lifschitz and Pittel's work [**96**] who were amongst the first to provide nontrivial lower bounds by elementary arguments. Let $\lambda(\sigma)$ be the length of the longest increasing subsequence in permutation σ and $\kappa(\sigma)$ the number of increasing subsequences (of all length). Then, the EGF $K(z)$ of cumulated values of κ satisfies

$$K(z) := \sum_{\sigma\in\mathcal{P}} \kappa(\sigma)\frac{z^{|\sigma|}}{|\sigma|!} = \frac{1}{1-z}e^{z/(1-z)}, \qquad [z^n]K(z) = \sum_{k=0}^{n}\binom{n}{k}\frac{1}{k!} \sim \frac{1}{2\sqrt{\pi e}}n^{-1/4}e^{2\sqrt{n}}.$$

Since $2^\lambda \le \kappa$ and the exponential function is convex, the expected length of the longest increasing subsequence has upper bound $c\sqrt{n} + o(n^{1/2})$ with $c = 2/\log 2 \doteq 2.88539$. ◁

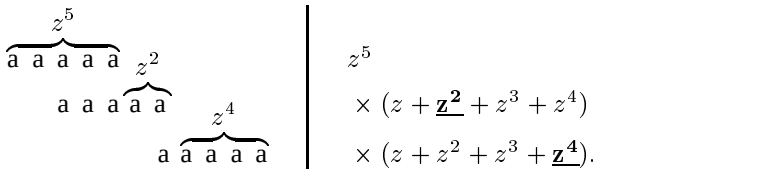
EXAMPLE 20. *Patterns in words.* Take the set of all words $W = \mathfrak{S}\{\mathcal{A}\}$ over a finite alphabet $\mathcal{A} = \{a_1, \dots, a_r\}$. A pattern $\mathfrak{p} = p_1p_2\cdots p_k$, which is particular word of length k has been fixed. What is sought is the BGF $W(z,u)$ of $\mathcal{W}$, where u marks the number of occurrences of pattern $\mathfrak{p}$ inside a word of $\mathcal{W}$. Results of Chapter I already give access to $W(z,0)$, which is the OGF of words not containing the pattern.

In accordance with the inclusion-exclusion principle, one should introduce the class $\mathcal{X}$ of words augmented by distinguishing an arbitrary number of occurrences of $\mathfrak{p}$. Define a *cluster* cluster as a maximal collection of distinguished occurrences that have an overlap. For instance, if $\mathfrak{p} = aaaaa$, a particular word may be give rise to the particular cluster:

```
a b a a a a a a a a a a a a a b a a a a a a a a b b
-------------------------------------------------
      a a a a a
          a a a a a
                a a a a a
```

Then objects of $\mathcal{X}$ decompose as sequences of either arbitrary letters from $\mathcal{A}$ or clusters. Clusters are themselves obtained by repeatedly sliding the pattern, but with the constraint that it should constantly overlap partly with itself.

Let $c(z)$ be the autocorrelation polynomial of $\mathfrak{P}$ as defined in Chapter I, and set $\widehat{c}(z) = c(z) - 1$. A moment's reflection should convince the reader that $z^k\widehat{c}(z)^{s-1}$ when expanded describes all the possibilities for forming clusters of s overlapping occurrences. On the example above, one has $\widehat{c}(z) = 1 + z + z^2 + z^3 + z^4$, and a particular cluster of 3 overlapping occurrences corresponds to one of the terms in $z^k\widehat{c}(z)^2$ as follows:



The OGF of clusters is consequently $C(z) = z^k/(1-\widehat{c}(z))$ since this quantity describes all the ways to write the pattern (z^k) and then slide it so that it should overlap with itself (this is given by $(1-\widehat{c}(z))^{-1}$). By a similar reasoning, the BGF of clusters is $vz^k/(1-v\widehat{c}(z))$, and the BGF of $\mathcal{X}$ with the supplementary variable v marking the number of distinguished occurrences is

$$X(z,v) = \frac{1}{1 - rz - vz^k/(1-v\widehat{c}(z))}.$$

Finally, the usual inclusion-exclusion argument (change v to $u-1$) yields $W(z,u) = X(z,u-1)$. As a result:

> *For a pattern $\mathfrak{p}$ with correlation polynomial $c(z)$ and length k, the BGF of words over an alphabet of cardinality r, where u marks the number of occurrences of $\mathfrak{p}$, is*
>
> $$W(z,u) = \frac{(u-1)c(z) - u}{(1-rz)((u-1)c(z)-u) + (u-1)z^k}.$$

The specialization $u = 0$ gives back the formula already found in Chapter I. The same principles clearly apply to weighted models corresponding to unequal letter probabilities, provided a suitably weighted version of the correlation polynomial is introduced. □

There are a very large number of formulæ related to patterns in strings. For instance, BGFs are known for occurrences of one or several patterns under either Bernoulli or Markov models. We refer globally to Szpankowski's book [**139**], where such questions are treated systematically and in great detail.

▷ **23.** *Moments of number of occurrences.* Observe that the derivatives of $X(z,v)$ at $v=0$ give access to the factorial moments of the number of occurrences of a pattern. Evaluate the mean and variance of the number of occurrences. (Hint: set $z \mapsto z/r$ and expand the rational fractions involved near $z=1$.) ◁

▷ **24.** *Words with fixed repetions.* Let $W^{\langle k\rangle}(z) = [u^k]W(z,u)$ be the OGF of words containing a pattern exactly k times. There exist two functions $\lambda(z), \mu(z)$ such that $W^{\langle k\rangle}(z) = \lambda(z)\mu(z)^k$ for any $k \geq 1$. ◁

▷ **25.** *Patterns in Bernoulli sequences.* Work out the BGF of the number of occurrences of a pattern in a random string with nonuniform letter probabilities $p_j = \mathbb{P}(a_j)$. (Hint: one needs to define a weighted correlation polynomial $c(z)$.) ◁

▷ **26.** *Patterns in binary trees.* Consider the class $\mathcal{B}$ of pruned binary trees. An occurrence of pattern $\mathfrak{t}$ in a tree τ is defined by a node whose "dangling subtree" is isomorphic to $\mathfrak{t}$. Let p be the size of $\mathfrak{t}$. The BGF $B(z,u)$ of class $\mathcal{B}$ where u marks the number of occurrences of $\mathfrak{t}$ is sought.

The OGF of $\mathcal{B}$ is $B(z) = (1-\sqrt{1-4z})/(2z)$. The quantity $uB(zu)$ is the BGF of $\mathcal{B}$ with v marking external nodes. By virtue of the pointing operation, the quantity

$$U_k := \left(\frac{1}{k!}\partial_v^k \left(vB(zv)\right)\right)_{v=1},$$

describes trees with k distinct external nodes distinguished (pointed). The quantity

$$V := \sum U_k u^k (z^p)^k \quad \text{satisfies} \quad V = \left(vB(zv)\right)_{v=1+uz^p},$$

by virtue of Taylor's formula. It is also the BGF of trees with distinguished occurrences of $\mathfrak{t}$. Setting $u \mapsto u-1$ in V then gives back $B(z,u)$ as

$$B(z,u) = \frac{1}{2z}\left(1 - \sqrt{1-4z-4(u-1)z^{p+1}}\right).$$

In particular

$$B(z,0) = \frac{1}{2z}\left(1 - \sqrt{1-4z+4z^{p+1}}\right)$$

gives the OGF of trees *not* containing pattern $\mathfrak{t}$. The method generalizes to any simple variety of trees and it can be used to prove that the factored representation (as a directed acyclic graph) of a random tree of size n has expected size $O(n/\sqrt{\log n})$; see [**57**]. ◁

III. 7. Extremal parameters

Apart from additively inherited parameters already examined at length in this chapter, another important category is that of parameters defined by a maximum rule. Two major cases are the largest component in a combinatorial structure (for instance, the largest cycle of a permutation) and the maximum degree of nesting of constructions in a recursive structure (typically, the height of a tree). In this case, bivariate generating functions are of little help. The standard technique consists in introducing a collection of univariate generating functions defined by imposing a bound on the parameter of interest. Such GF's can then be constructed by the symbolic method in its univariate version.

III. 7.1. Largest components. Consider a construction $\mathcal{B} = \Phi\{\mathcal{A}\}$, where Φ may involve an arbitrary combination of basic constructions, and assume here for simplicity that the construction for $\mathcal{B}$ is a non–recursive one. This corresponds to a relation between generating functions

$$B(z) = \Psi[A(z)],$$

where Ψ is the functional that is the "image" of the combinatorial construction Φ. Elements of $\mathcal{A}$ thus appear as components in an object $\beta \in \mathcal{B}$. Let $\mathcal{B}^{\langle b\rangle}$ denote the subclass of $\mathcal{B}$ formed with objects whose $\mathcal{A}$–components all have a size at most b. The GF of $\mathcal{B}^{\langle b\rangle}$ is obtained by the same process as that of $\mathcal{B}$ itself, safe that $A(z)$ should be replaced by the GF of elements of size at most b. Thus,

$$B^{\langle b\rangle}(z) = \Psi[\mathbf{T}_b A(z)],$$

where the *truncation operator* is defined on series by

$$\mathbf{T}_b f(z) = \sum_{n=0}^{b} f_n z^n \qquad (f(z) = \sum_{n=0}^{\infty} f_n z^n).$$

Several cases of this situation have already been encountered in earlier chapters. For instance, the cycle decomposition of permutations translated by

$$P(z) = \exp\left(\log \frac{1}{1-z}\right)$$

gives more generally the EGF of permutations with longest cycle $\leq b$,

$$P^{\langle b\rangle}(z) = \exp\left(\frac{z}{1} + \frac{z^2}{2} + \cdots + \frac{z^b}{b}\right),$$

which involves the truncated logarithm. Similarly, the EGF of words over an m–ary alphabet

$$W(z) = (e^z)^m$$

leads to the EGF of words such that each letter occurs at most b times:

$$W^{\langle b\rangle}(z) = \left(1 + \frac{z}{1!} + \frac{z^2}{2!} + \cdots + \frac{z^b}{b!}\right)^m,$$

which now involves the truncated exponential. One finds similarly the EGF of set partitions with largest block of size at most b,

$$S^{\langle b\rangle}(z) = \exp\left(\frac{z}{1!} + \frac{z^2}{2!} + \cdots + \frac{z^b}{b!}\right).$$

A slightly less direct example is that of the longest run in a sequence of binary draws. The collection $\mathcal{W}$ of binary strings over the alphabet $\{a, b\}$ admits the decomposition

$$\mathcal{W} = \mathfrak{S}\{a\} \cdot \mathfrak{S}\{b\, \mathfrak{S}\{a\}\},$$

corresponding to a "scansion " dictated by the occurrences of the letter b. The corresponding OGF then appears under the form

$$W(z) = Y(z) \cdot \frac{1}{1 - zY(z)} \qquad \text{where } Y(z) = \frac{1}{1-z}$$

corresponds to $\mathcal{Y} = \mathfrak{S}\{a\}$. Thus, the OGF of strings with at most $k-1$ consecutive occurrences of the letter a obtains upon replacing $Y(z)$ by its truncation:

$$W^{\langle k\rangle}(z) = Y^{\langle k\rangle}(z)\frac{1}{1 - zY^{\langle k\rangle}(z)} \text{ where } Y^{\langle k\rangle}(z) = 1 + z + z^2 + \cdots + z^{k-1},$$

so that

$$W^{\langle k\rangle}(z) = \frac{1 - z^k}{1 - 2z + z^{k+1}}.$$

Such generating functions are thus easy to derive. The asymptotic analysis of their coefficients is however often hard when compared to additive parameters, owing to the need to rely on complex analytic properties of the truncation operator. The bases of a general asymptotic theory have been laid by Gourdon [**70**].

▷ **27.** *Smallest components.* The EGF of permutations with smallest cycle of size $> b$ is

$$\frac{\exp(-\frac{z}{1} - \frac{z^2}{2} - \frac{z^b}{b})}{1-z}.$$

A symbolic theory of *smallest* components in combinatorial structures is easily developed as regards GFs. Elements of the corresponding asymptotic theory are provided by Panario and Richmond in [**112**]. ◁

III. 7.2. Height. The degree of nesting of a recursive construction is a generalization of the notion of height in the simpler case of trees. Consider for instance a recursively defined class

$$\mathcal{B} = \Phi\{\mathcal{B}\},$$

where Φ is a construction. Let $\mathcal{B}^{[h]}$ denote the subclass of $\mathcal{B}$ composed solely of elements whose construction involves at most h applications of Φ. We have by definition

$$\mathcal{B}^{[h+1]} = \Phi\{\mathcal{B}^{[h]}\}.$$

Thus, with Ψ the image functional of construction Φ, the corresponding GF's are defined by a *recurrence*,

$$B^{[h+1]} = \Psi[B^{[h]}].$$

It is usually convenient to start the recurrence with the initial condition $B^{[-1]}(z) = 0$. (This discussion is related to semantics of recursion, p. 16)

Consider for instance general plane trees defined by

$$\mathcal{G} = \mathcal{N} \times \mathfrak{S}\{\mathcal{G}\} \qquad \text{so that} \qquad G(z) = \frac{z}{1 - G(z)}.$$

Define the height of a tree as the number of nodes on its longest branch. Then the set of trees of height $\leq h$ satisfies the recurrence

$$\mathcal{G}^{[0]} = \mathcal{N},\ \mathcal{G}^{[h+1]} = \mathcal{N} \times \mathfrak{S}\{\mathcal{G}^{[h]}\}.$$

Accordingly, the OGF of trees of bounded height satisfies

$$G^{[-1]}(z) = 0,\ G^{[0]}(z) = z,\ G^{[h+1]}(z) = \frac{z}{1 - G^{[h]}(z)}.$$

The recurrence unwinds and one finds

$$G^{[h]}(z) = \cfrac{z}{1 - \cfrac{z}{1 - \cfrac{z}{\ddots \cfrac{}{1 - z}}}},$$

where the number of stages in the fraction equals b. This is the finite form (technically known as a "convergent") of a *continued fraction* expansion. From implied linear recurrences and an analysis based on Mellin transforms, de Bruijn, Knuth, and Rice [**37**] have determined the average height of a general plane tree to be $\sim \sqrt{\pi n}$.

For plane binary trees defined by

$$\mathcal{B} = \mathcal{Z} + \mathcal{B} \times \mathcal{B} \qquad \text{so that} \qquad B(z) = z + (B(z))^2,$$

(size is the number of external nodes), the recurrence is

$$B^{[0]}(z) = z,\ B^{[h+1]}(z) = z + (B^{[h]}(z))^2.$$

In this case, the $B^{[h]}$ are the approximants to a "continuous quadratic form", namely

$$B^{[h]}(z) = z + (z + (z + (\cdots)^2)^2)^2.$$

These are polynomials of degree 2^h for which no closed form expression is known, nor even likely to exist[4]. However, using complex asymptotic methods and singularity analysis, Flajolet and Odlyzko [**52**] have shown that the average height of a binary plane tree is $\sim 2\sqrt{\pi n}$.

For Cayley trees, finally, the defining equation is

$$\mathcal{T} = \{1\} \star \mathfrak{P}\{\mathcal{T}\} \qquad \text{so that} \qquad T(z) = ze^{T(z)}.$$

The EGF of trees of bounded height satisfy the recurrence

$$T^{[0]}(z) = z,\ T^{[h+1]}(z) = ze^{T^{[h]}(z)}.$$

We are now confronted with a "continuous exponential",

$$T^{[h]}(z) = ze^{ze^{ze^{\cdot^{\cdot^{\cdot ze^z}}}}}.$$

The average height was found by Rényi and Szekeres who appealed again to complex asymptotics and found it to be $\sim \sqrt{2\pi n}$.

These examples show that height statistics are closely related to iteration theory. Except in a few cases like general plane trees, normally no algebra is available and one has to resort to complex analytic methods as exposed in forthcoming chapters.

---

[4]These polynomials are exactly the much studied Mandelbrot polynomials whose behaviour in the complex plane gives rise to extraordinary graphics.

▷ **28.** *Height in general Catalan trees.* The model of height in general plane trees can be solved algebraically. The OGF $G^{[h]}(z)$ of trees of height $\leq h$ is of the form

$$z\frac{F_{h+1}(z)}{F_{h+2}(z)},$$

where the F's are the Fibonacci polynomials

$$F_0(z) = 0,\ F_1(z) = 1,\ F_{h+2}(z) = F_{h+1}(z) - zF_h(z).$$

Express the F_h in terms of $G(z)$ itself. Find explicit forms for the distribution of height in trees of $\mathcal{G}_n$ by means of Lagrange inversion. [This treatment is due to De Bruijn, Knuth, and Rice [**37**].] ◁

III. 7.3. Averages and moments. For extremal parameters, the GF of mean values obey a general pattern. Let $\mathcal{F}$ be some combinatorial class with GF $f(z)$. Consider for instance an extremal parameter χ such that $f^{[h]}(z)$ the GF of objects with χ-parameter *at most* h. The GF of objects for which $\chi = k$ *exactly* is equal to

$$f^{[h]}(z) - f^{[h-1]}(z).$$

Thus differencing gives access to the probability distribution of height over $\mathcal{F}$. The generating function of cumulated values (providing mean values after normalization) is then

$$\begin{aligned}\Xi(z) &= \sum_{h=0}^{\infty} h\left[f^{[h]}(z) - f^{[h-1]}(z)\right]\\ &= \sum_{h=0}^{\infty}\left[f(z) - f^{[h]}(z)\right],\end{aligned}$$

as is readily checked by rearranging the second sum, or equivalently using summation by parts.

For maximum component size, the formulæ involve truncated Taylor series. For height, analysis involves in all generality the differences between the fixed point of a functional Φ (the GF $f(z)$) and the approximations to the fixed point ($f^{[h]}(z)$) provided by iteration. This is a common scheme in extremal statistics.

▷ **29.** *Hierarchical partitions.* Let $\varepsilon(z) = e^z - 1$. Find a combinatorial interpretation for

$$\varepsilon(\varepsilon(\cdots(\varepsilon(z)))) \qquad (h \text{ times}).$$

(Such structures show up in statistical classification theory.) ◁

▷ **30.** *Balanced trees.* Balanced structures lead to counting GF's close to the ones obtained for height statistics. The OGF of balanced 2-3 trees of height h counted by the number of leaves satisfies the recurrence

$$Z^{[h+1]}(z) = Z^{[h]}(z^2 + z^3) = (Z^{[h]}(z))^2 + (Z^{[h]}(z))^3.$$

Express it in terms of the iterates of $\sigma(z) = z^2 + z^3$.

Find the OGF of mean values of the number of internal nodes in such trees. ◁

▷ **31.** *Extremal statistics in random mappings.* Find the EGF's relative to the largest cycle, longest branch, and diameter of functional graphs. Do the same for the largest tree, largest component. [Hint: see [**53**] for details.] ◁

▷ **32.** *Deep nodes in trees.* Find the GF of mean values of the number of nodes at maximal depth in a general plane tree and in a Cayley tree. ◁

III. 8. Notes

Multivariate generating functions are a common tool from classical combinatorial analysis. Comtet's book [**28**] is once more an excellent source of examples. A systematization of multivariate generating functions for inherited parameters is given in the book by Jackson and Goulden [**68**].

In contrast generating functions for averages seemed to have received relatively little attention before the advent of digital computers and the analysis of algorithms. Many important techniques are implicit in Knuth's books, especially [**85, 86**]. Wilf discusses related issues in his book [**153**] and the paper [**151**]. Early systems specialized to tree algorithms have been proposed by Flajolet and Steyaert in the beginning 1980's [**45, 59, 60, 138**]; see also Berstel and Reutenauer's work [**15**]. Some of the ideas developed there (viewing generating functions of averages as images of combinatorial structures with multiplicities attached) took their inspiration from the well established treatment of formal power series in noncommutative indeterminates (that can be seen as words with multiplicities attached), see Eilenberg's book [**41**] or the proceedings edited by Berstel [**126**].

The global framework of constructible structures affords a neat structural categorization of parameters of combinatorial objects—additively inherited parameters, recursive parameters, largest components, and height. This approach becomes especially powerful when examined in the light of asymptotic properties of structures. In addition, the principles developed here render the analysis of a large class of combinatorial parameters entirely systematic. This includes complexity measures for a closed class of programmes and data structures. Several computations in this area can then even be automated with the help of computer algebra systems [**56, 158**].

APPENDIX A

Auxiliary Results & Notions

1. Arithmetical functions. A general reference for this section is Apostol's book [**5**].

The function $\varphi(k)$ is the *Euler totient function* and it intervenes in the unlabelled cycle construction. It is defined as the number of integers in $[1, k]$ that are relatively prime to k. Thus, one has $\varphi(p) = p - 1$ if p is a prime. More generally when the prime number decomposition of k is $k = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$, then

$$\varphi(k) = p_1^{\alpha_1 - 1}(p_1 - 1) \cdots p_r^{\alpha_r}(p_r - 1).$$

A number is squarefree if it is not divisible by the square of a prime. The *Moebius function* $\mu(n)$ is defined to be 0 if n is not squarefree and otherwise is $(-1)^r$ if $n = p_1 \cdots p_r$ is a product of r distinct primes.

Many elementary properties of arithmetical functions are easily established by means of *Dirichlet generating functions* (DGF). Let $(a_n)_{n\geq 1}$ be a sequence; its Dirichlet series is formally defined by

$$\alpha(s) = \sum_{n=1}^{\infty} \frac{a_n}{n^s}.$$

In particular, the DGF of the sequence $a_n = 1$ is the Riemann zeta function, $\zeta(s) = \sum_{n\geq 1} n^{-s}$. The fact that every number uniquely decomposes into primes is reflected by Euler's formula,

$$\zeta(s) = \prod_{p\in\mathcal{P}} \left(1 - \frac{1}{p^s}\right)^{-1}, \tag{1}$$

where p ranges over the set $\mathcal{P}$ of all primes. (As observed by Euler, the fact that $\zeta(1) = \infty$ in conjunction with (1) provides a simple analytic proof that there are infinitely many primes!)

Equation (1) implies elementarily that

$$M(s) := \sum_{n\geq 1} \frac{\mu(n)}{n^s} = \prod_{p\in\mathcal{P}} \left(1 - \frac{1}{p^s}\right) = \frac{1}{\zeta(s)}. \tag{2}$$

The coefficients $\mu(n)$ are known as the Moebius coefficient (or Moebius function). They satisfy

$$\mu(n) = (-1)^r \qquad \text{if } n = p_1 p_2 \cdots p_r \text{ for distinct primes } p_j,$$

and $\mu(n) = 0$ whenever n is divisible by a square.

Finally, if $(a_n), (b_n), (c_n)$ have DGF $\alpha(s), \beta(s), \gamma(s)$, then one has the equivalence

$$\alpha(s) = \beta(s)\gamma(s) \quad \Longleftrightarrow \quad a_n = \sum_{d \mid n} b_d c_{n/d}.$$

In particular, taking $c_n = 1$ ($\gamma(s) = \zeta(s)$) and solving for $\beta(s)$ shows (using (2)) the implication

$$a_n = \sum_{d \mid n} b_d \quad \Longleftrightarrow \quad b_n = \sum_{d \mid n} \mu(d) a_{n/d},$$

which is known as *Moebius inversion*. This relation is used in the enumeration of irreducible polynomials (Section I. 6.2).

2. Asymptotic Notations. Let $\mathbb{S}$ be a set and $s_0 \in \mathbb{S}$ a particular element of $\mathbb{S}$. We assume a notion of neighbourhood to exist on $\mathbb{S}$. Examples are $\mathbb{S} = \mathbb{Z}_{>0} \cup \{+\infty\}$ and $s_0 = +\infty$, $\mathbb{S} = \mathbb{R}$ and s_0 any point in $\mathbb{R}$, $\mathbb{S} = \mathbb{C}$ or a subset of $\mathbb{C}$, and so on. Two functions ϕ and g from $\mathbb{S} \setminus \{s_0\}$ to $\mathbb{C}$ are given.

— *$\mathcal{O}$–notation*: write

$$\phi(s) \underset{s \to s_0}{=} \mathcal{O}(g(s))$$

if the ratio $\phi(s)/g(s)$ stays bounded as $s \to s_0$ in $\mathbb{S}$. In other words, there exists a neighbourhood $\mathcal{V}$ of s_0 and a constant $C > 0$ such that

$$|\phi(s)| \leq C\,|g(s)|\,, \qquad s \in \mathcal{V}, \quad s \neq s_0.$$

One also says that *"ϕ is of order at most g, or ϕ is big–Oh of g (as s tends to s_0)"*.

— *$\sim$–notation*: write

$$\phi(s) \underset{s \to s_0}{\sim} g(s)$$

if the ratio $\phi(s)/g(s)$ tends to 1 as $s \to s_0$ in $\mathbb{S}$. One also says that "*ϕ and g are asymptotically equivalent (as s tends to s_0)*".

— *o–notation*: write

$$\phi(s) \underset{s \to s_0}{=} o(g(s))$$

if the ratio $\phi(s)/g(s)$ tends to 0 as $s \to s_0$ in $\mathbb{S}$. In other words, for any (arbitrarily small) $c > 0$, there exists a neighbourhood $\mathcal{V}_c$ of s_0 (depending on c), such that

$$|\phi(s)| \leq c\,|g(s)|\,, \qquad s \in \mathcal{V}_c, \quad s \neq s_0.$$

One also says that *"ϕ is of order smaller than g, or ϕ is little–oh of g (as s tends to s_0)"*.

These notations are due to Bachmann and Landau towards the end of the nineteenth century. See Knuth's note for a historical discussion [**87**, Ch. 4].

Related notations, of which however we only make scanty use, are

— *Ω-notation*: write

$$\phi(s) \underset{s \to s_0}{=} \Omega(g(s))$$

if the ratio $\phi(s)/g(s)$ stays bounded from below in modulus by a nonzero quantity, as $s \to s_0$ in $\mathbb{S}$. One then says that ϕ is of order at least g.

— *Θ-notation*: write

$$\phi(s) \underset{s \to s_0}{=} \Theta(g(s))$$

if $\phi(s) = \mathcal{O}(s)$ and $\phi(s) = \Omega(s)$. One then says that ϕ is of order exactly g.

For instance, one has as $n \to +\infty$ in $\mathbb{Z}_{>0}$:

$$\sin n = o(\log n); \quad \log n = O(\sqrt{n}); \quad \log n = o(\sqrt{n});$$
$$\tbinom{n}{2} = \Omega(n\sqrt{n}); \quad \pi n + \sqrt{n} = \Theta(n).$$

As $x \to 1$ in $\mathbb{R}_{\leq 1}$, one has

$$\sqrt{1-x} = o(1); \quad e^x = O(\sin x); \quad \log x = \Theta(x-1).$$

We take as granted in this book the elementary asymptotic calculus with such notations (see, e.g., [**130**, Ch. 4] for a smooth introduction close to the needs of analytic combinatorics and de Bruijn's classic [**35**] for a beautiful presentation.). We shall retain here the fact that Taylor expansions imply asymptotic expansions; for instance, the convergent expansions for $|u| < 1$,

$$\log(1+u) = \sum_{k=1}^{\infty} \frac{(-1)^k}{k} u^k, \quad \exp(u) = \sum_{k \geq 0} \frac{1}{k!} u^k, \quad (1-u)^\alpha = \sum_{k \geq 0} \binom{k+\alpha-1}{k} u^k,$$

imply (as $u \to 0$)

$$\log(1+u) = u + \mathcal{O}(u^2), \quad \exp(u) = 1 + u + \frac{u^2}{2} + \mathcal{O}(u^3), \quad (1-u)^{1/2} = 1 - \frac{u}{2} + \mathcal{O}(u^2),$$

and, in turn, (as $n \to +\infty$)

$$\log\left(1+\frac{1}{n}\right) = \frac{1}{n} + \mathcal{O}\left(\frac{1}{n^2}\right), \quad \left(1 - \frac{1}{\log n}\right)^{1/2} = 1 - \frac{1}{2\log n} + o\left(\frac{1}{\log n}\right).$$

Two important special expansions are Stirling's formula for factorials and the harmonic number approximation,

$$\begin{array}{rcll} n! & = & n^n e^{-n} \sqrt{2\pi n}\,(1+\epsilon_n)\,, & 0 < \epsilon_n < \frac{1}{12n} \\ \mathrm{H}_n & = & \log n + \gamma + \dfrac{1}{2n} - \dfrac{1}{12n^2} + \eta_n & \eta_n = \mathcal{O}\left(n^{-4}\right), \quad \gamma \doteq 0.57721, \end{array} \tag{3}$$

that are best established as consequences of the Euler–Maclaurin summation formula [**35, 130**].

▷ **1.** *Simplification rules for the asymptotic calculus.* Some of them are

$$\begin{array}{lcll} \mathcal{O}(\lambda f) & \longrightarrow & \mathcal{O}(f) & (\lambda \neq 0) \\ \mathcal{O}(f) \pm \mathcal{O}(g) & \longrightarrow & \mathcal{O}(|f| + |g|) & \\ & \longrightarrow & \mathcal{O}(f) & \text{if } g = \mathcal{O}(f) \\ \mathcal{O}(f \cdot g) & \longrightarrow & O(f)O(g). & \end{array}$$

Similar rules apply for $o(\cdot)$. ◁

▷ **2.** *Harmonics of harmonics.* The harmonic numbers are readily extended to non-integral index by

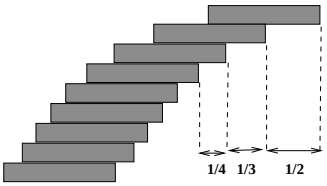
$$\mathrm{H}_x := \sum_{k=1}^{\infty} \left(\frac{1}{k} - \frac{1}{k+x}\right).$$

For instance, $\mathrm{H}_{1/2} = 2 - 2\log 2$. This extension is related to the Gamma function [**150**], and it can be proved that the asymptotic estimate (3), with x replacing n, remains valid as $x \to +\infty$. A typical asymptotic calculation shows that

$$\mathrm{H}_{\mathrm{H}_n} = \log\log n + \gamma + \frac{\gamma + \frac{1}{2}}{\log n} + \mathcal{O}\left(\frac{1}{\log^2 n}\right).$$

What is the shape of an asymptotic expansion of $\mathrm{H}_{\mathrm{H}_{\mathrm{H}_n}}$? ◁

▷ **3.** *Stackings of dominos.* A stock of dominos of length 1cm is given. It is well known that one can stack up dominos in a harmonic mode:



Estimate within 1% the minimal number of dominos needed to achieve a horizontal span of 1m (=100cm). [Hint: about $1.50926\,10^{43}$ dominos!] Set up a scheme to evaluate this integer exactly, and do it! ◁

▷ **4.** *High precision fraud.* Why is it that, to forty decimal places, one finds

$$\begin{aligned} 4\sum_{k=1}^{500,000}\frac{(-1)^{k-1}}{2k-1} &\doteq 3.1415906535897932\underline{40}4626433832\underline{6}9502884197 \\ \pi &\doteq 3.1415926535897932384626433832795028841 97, \end{aligned}$$

with only four "wrong" digits in the first sum? (Hint: consider the simpler problem

$$\frac{1}{9801} \doteq 0.00\,01\,02\,03\,04\,05\,06\,07\,08\,09\,10\,11\,12\,13\,14\,15\,16\,17\,18\,19\,20\,21\,22\,23\,24\,25\cdots.)$$

Many fascinating facts of this kind are to be found in works by Jon and Peter Borwein [**21, 22**]. ◁

3. Cycle construction. The unlabelled cycle construction is introduced in Chapter 1 and is classically obtained within the framework of Pólya theory [**28, 113, 115**]. The derivation given here is based on an elementary use of symbolic methods tha follows [**58**]. It relies on bivariate GF's developed in Chapter III, with z marking size and u marking the number of components. Consider a class $\mathcal{A}$ and the sequence class $\mathcal{S} = \mathfrak{S}_{\geq 1}\{\mathcal{A}\}$. A sequence $\sigma \in \mathcal{S}$ is primitive (or aperiodic) if it is not the repetition of another sequence (e.g., $\alpha\beta\beta\alpha\alpha$ is primitive, but $\alpha\beta\alpha\beta$ is not). The class $\mathcal{PS}$ of primitive sequences is determined implicitly,

$$S(z,u) \equiv \frac{uA(z)}{1-uA(z)} = \sum_{k\geq 1} PS(z^k, u^k),$$

which expresses that every sequence possesses a "root" that is primitive. Moebius inversion then gives

$$PS(z,u) = \sum_{k\geq 1}\mu(k)S(z^k,u^k) = \sum_{k\geq 1}\mu(k)\frac{u^k A(z^k)}{1-u^k A(z^k)}.$$

A cycle is primitive if all of its linear representations are primitive. There is an exact one-to-ℓ correspondence between primitive ℓ-cycles and primitive ℓ-sequences. Thus, the BGF $PC(z,u)$ of primitive cycles is obtained by effecting the transformation $u^\ell \mapsto \frac{1}{\ell}u^\ell$ on $PS(z,u)$, which means

$$PC(z,u) = \int_0^u P(z,v)\,\frac{dv}{v},$$

giving after term-wise integration,

$$PC(z,u) = \sum_{k\geq 1}\frac{\mu(k)}{k}\log\frac{1}{1-u^k A(z^k)}.$$

Finally, cycles can be composed from arbitrary repetitions of primitive cycles, which yields

$$C(z,u) = \sum_{k\geq 1} PC(z^k, u^k).$$

The arithmetical identity $\sum_{d\,|\,k} \mu(d)/d = \varphi(k)/k$ gives eventually

$$C(z,u) = \sum_{k\geq 1} \frac{\varphi(k)}{k} \log \frac{1}{1-u^k A(z^k)}, \tag{4}$$

as was to be proved.

Formula (4) is exactly the one that appears in the translation of the cycle construction in the unlabelled case (Theorem III.1). Upon setting $u = 1$, it gives the univariate version (Theorem I.1).

▷ **5.** *Around the cycle construction.* Similar methods yield the BGFs of multisets of cycles and multisets of aperiodic cycles as

$$\prod_{k\geq 1} \frac{1}{1-u^k A(z^k)} \quad \text{and} \quad \frac{1}{1-uA(z)},$$

respectively [**36**]. (The latter fact corresponds to the property that any word can be written as a decreasing product of Lyndon words; it serves to construct bases of free Lie algebras [**98**, Ch. 5].) ◁

▷ **6.** *Aperiodic words.* An aperiodic word is a primitive sequence of letters. The number of aperiodic words of length n over an m-ary alphabet corresponds to primitive sequences with $A(z) = mz$ and is

$$PW_n^{(m)} = \sum_{d\,|\,n} \mu(d) m^{m/d}.$$

For $m = 2$, the sequence starts as $2, 2, 6, 12, 30, 54, 126, 240, 504, 990$ (*EIS* **A027375**). ◁

4. Formal power series. Formal power series extend the usual operations on polynomials to infinite series of the form

$$f = \sum_{n\geq 0} f_n z^n, \tag{5}$$

where z is a formal indeterminate. The notation $f(z)$ is also employed. Let $\mathbb{K}$ be a field of coefficients (usually $\mathbb{Q}, \mathbb{R}, \mathbb{C}$); the ring of formal power series is denoted by $\mathbb{K}[[z]]$ and it is the set $\mathbb{K}^{\mathbb{N}}$ (of infinite sequences of elements of $\mathbb{K}$) written as infinite power series (5) and endowed with the operations of sum and product,

$$\begin{aligned}
\left(\sum_n f_n z^n\right) + \left(\sum_n g_n z^n\right) &:= \sum_n (f_n + g_n)\, z^n \\
\left(\sum_n f_n z^n\right) \times \left(\sum_n g_n z^n\right) &:= \sum_n \left(\sum_{k=0}^{n} f_k g_{n-k}\right) z^n.
\end{aligned}$$

A topology (known as the formal topology) is put on $\mathbb{K}[[z]]$ by which two series f, g are "close" if they coincide to a large number terms. First, the valuation of a formal power series $f = \sum_n f_n z^n$ is the smallest r such that $f_r \neq 0$ and is denoted by $\mathrm{val}(f)$. (One sets $\mathrm{val}(0) = +\infty$.) Given two power series f and g, their distance $d(f,g)$ is then defined as $2^{-\mathrm{val}(f-g)}$. With this distance (in fact an ultrametric distance), the space of all formal power series is a *complete metric space*. Roughly, the limit of a sequence of series $\{f^{(j)}\}$ exists if, for each n, the coefficient of order n in $f^{(j)}$ eventually stabilizes to a fixed value as $j \to \infty$. In this way convergence can be defined for infinite sums: it suffices that

the general term of the sum should tend to 0 in the formal topology, *i.e.*, the valuation of the general term should tend to ∞. Similarly for infinite products, where $\prod(1+u^{(j)})$ converges as soon as $u^{(j)}$ tends to 0 in the topology of formal power series.

It is then a simple exercise to prove that the sum $Q(f) := \sum_{k\geq 0} f^k$ exists (the sum convergerges in the formal topology) whenever $f_0 = 0$; the quantity then defines the quasi-inverse $(1-f)^{-1}$, with the implied properties with respect to multiplication, namely, $Q(f)(1-f) = 1$. In the same way one defines formally logarithms and exponentials, primitives and derivatives, etc. Also, the composition $f \circ g$ is defined whenever $g_0 = 0$ by substitution of formal power series. More generally, any (possibly infinitary) process on series that involves at each coefficient only finitely many operations is well-defined (and is accordingly a continuous functional in the formal topology).

▷ **7.** *The OGF of permutations.* The ordinary generating function of permutations,

$$P(z) := \sum_{n=0}^{\infty} n! z^n = 1 + z + 2z^2 + 6z^3 + 24z^4 + 120z^5 + 720z^6 + 5040z^7 + \cdots$$

exists as an element of $\mathbb{C}[[z]]$, although the series has radius of convergence 0. The quantity $1/P(z)$ is for instance well-defined (via the quasi-inverse) and one can compute legitimately and effectively $1 - 1/P(z)$ whose coefficients enumerate indecomposable permutations (p. 57). The formal series $P(z)$ can even be made sense of analytically as an asymptotic series (Euler), since

$$\int_0^{\infty} \frac{e^{-t}}{1+tz}\, dt \sim 1 - z + 2!z^2 - 3!z^3 + 4!z^4 - \cdots \qquad (z \to 0+).$$

Thus, the OGF of permutations is also representable as the (formal, divergent) asymptotic series of an integral. ◁

It can be proved that the usual functional properties of analysis extend to formal power series provided they make sense formally.

5. Lagrange Inversion. Lagrange inversion (Lagrange, 1770) relates the coefficients of the inverse of a function to coefficients of the powers of the function itself. It thus establishes a fundamental correspondence between functional composition and standard multiplication of series. Although the proof is technically simple, the result altogether non-elementary.

The inversion problem $z = h(y)$ is solved by the Lagrange series given below. It is assumed that $[y^0]h(z) = 0$, so that inversion is formally well defined and analytically local, and $[y^1]h(y) \neq 0$. The problem is is then conveniently standardized by setting $h(y) = y/\phi(y)$.

THEOREM A.1. *Let $\phi(u) = \sum_{k\geq 0} \phi_k u^k$ be a power series of $\mathbb{C}[[z]]$ with $\phi_0 \neq 0$. Then, the equation $y = z\phi(y)$ admits a unique solution in $\mathbb{C}[[z]]$ whose coefficients are given by (Lagrange form)*

$$y(z) = \sum_{n=1}^{\infty} y_n z^n, \qquad \textit{where} \quad y_n = \frac{1}{n}\left[u^{n-1}\right](\phi(u))^n. \tag{6}$$

Furthermore, one has for $k > 0$ (Bürmann form)

$$y(z)^k = \sum_{n=1}^{\infty} y_n^{(k)} z^n, \qquad \textit{where} \quad y_n = \frac{k}{n}\left[u^{n-k}\right](\phi(u))^n. \tag{7}$$

By linearity, a form equivalent to Burmann's (7), with H an arbitrary function, is

$$[z^n]H(y(z)) = \left[u^{n-1}\right](H'(u)\phi(u)^n).$$

PROOF. The method of indeterminates coefficients provides a system of polynomial equations for $\{y_n\}$ that is seen to have a unique (polynomial) solution:

$$y_1 = \phi_0, \quad y_2 = \phi_0\phi_1, \quad y_3 = \phi_0\phi_1^2 + \phi_0\phi_2, \ \dots \ .$$

Since y_n depends only on the coefficients of $\phi(u)$ till order n, one may assume without loss of generality that ϕ is a polynomial. Then, by general properties of analytic functions, $y(z)$ is analytic at 0 and it maps conformally a neighborhood of 0 into another neighbourhood of 0. Accordingly, the quantity $ny_n = [z^{n-1}]y'(z)$ can be estimated by Cauchy's coefficient formula:

$$\begin{aligned} ny_n &= \frac{1}{2i\pi}\int_{0+} y'(z)\,\frac{dz}{z^n} && \text{(Direct coefficient formula for } y'(z)) \\ &= \frac{1}{2i\pi}\int_{0+} \frac{dy}{(y/\phi(y))^n} && \text{(Change of variable } z \mapsto y) \\ &= [y^{n-1}]\,\phi(y)^n && \text{(Reverse coefficient formula for } \phi(y)^n). \end{aligned} \tag{8}$$

In the context of complex analysis, this useful result appears as nothing but an avatar of the change-of-variable formula. The proof of Bürmann's form is entirely similar. □

There exist instructive (but longer) combinatorial proofs based on what is known as the "cyclic lemma" or "conjugacy principle" [**119**] for Łukasiewicz words. (See also Ex. 36 in Chapter I.) Another classical proof due to Henrici relies on properties of iteration matrices [**28**, p. 144-153]; see also Comtet's book for related formulations [**28**].

Lagrange inversion serves most notably to develop explicit formulæ for simple families of trees (either labelled or not), random mappings, and more generally for problems involving coefficients of powers of some fixed function.

▷ **8.** *Lagrange–BÚrmann inversion for fractional powers.* The formula

$$[z^n]\left(\frac{y(z)}{z}\right)^{\alpha} = \frac{\alpha}{n+\alpha}[u^n]\phi(u)^{n+\alpha}$$

holds for any real or complex exponent α, and hence generalizes Bürmann's form. One can similarly expand $\log(y(z)/z)$. ◁

▷ **9.** *Abel's identity.* By computing in two different ways the coefficient

$$[z^n]e^{(\alpha+\beta)y} = [z^n]e^{\alpha y}\cdot e^{\beta y},$$

where $y = ze^y$ is the Cayley tree function, one derive the *Abel's identity*

$$(\alpha+\beta)(n+\alpha+\beta)^{n-1} = \alpha\beta\sum_{k=1}^{n-1}\binom{n}{k}(k+\alpha)^{k-1}(n-k+\beta)^{n-k-1}.$$

◁

6. Regular languages. Two notions of *regularity* for languages are described in the text (Section I. 4): R-regularity, which means definability by regular specifications, and A-regularity, which corresponds to acceptability by a deterministic finite automaton. We indicate briefly here the reasons why the two notions are equivalent. The arguments are minor adaptations of well known facts in the theory of formal languages, and we refer the reader to one of the many good books on the subject for details.

A-regularity implies S-regularity. This construction is due to Kleene [**81**] whose interest had its origin in the formal expressive power of nerve nets. Let a deterministic automaton $\mathfrak{a}$ be given, with alphabet $\mathcal{A}$, set of states Q, with q_0 and $\overline{Q}$ the initial state and the set of final states respectively. The idea consists in constructing inductively the family

of languages $\mathcal{L}_{i,j}^{(r)}$ of words that connect state q_i to state q_j passing only through states $q_0, \ldots, q_r$ in between q_i and q_j. We initialize the data with $\mathcal{L}_{i,j}^{(-1)}$ to be the singleton set $\{a\}$ if the transition $(q_i \circ a) = q_j$ exists, and the emptyset ($\emptyset$) otherwise. The fundamental recursion

$$\mathcal{L}_{i,j}^{(r)} = \mathcal{L}_{i,j}^{(r-1)} + \mathcal{L}_{i,r}^{(r-1)} \mathfrak{S}\{\mathcal{L}_{r,r}^{(r-1)}\} \mathcal{L}_{r,j}^{(r-1)},$$

incrementally takes into account the possibility of traversing the "new" state q_r. (The unions are clearly disjoint and the segmentation of words according to passages through state q_r is unambiguously defined, hence the validity of the sequence construction.) The language $\mathcal{L}$ accepted by $\mathfrak{a}$ is then given by the regular specification

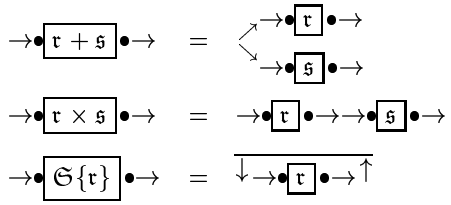
$$\mathcal{L} = \sum_{q_j \in \overline{Q}} \mathcal{L}_{0,j}^{\|Q\|},$$

that describes the set of all words leading from the initial state q_0 to any of the final states while passing freely through any intermediate state of the automaton.

S-regularity implies A-regularity. An object described by a regular specification $\mathfrak{r}$ can be viewed as a word decorated with separators that indicate the way it should be parsed. For instance, an element of $\mathfrak{S}\{a + aa\}$ may be viewed as the word

$$\langle a \,|\, aa \,|\, aa \,|\, a \,|\, aa \rangle,$$

over the enriched alphabet $\mathcal{A} \cup \{\text{`}|\text{'}, \text{`}\langle\text{'}, \text{`}\rangle\text{'}\}$. The extended representations are then recognizable by automata as shown by an inductive construction. We only state the principles informally here. Let $\rightarrow\bullet\boxed{\mathfrak{r}}\bullet\rightarrow$ represent symbolically the automaton recognizing the regular expression $\mathfrak{r}$, with the initial state on the left and the final state(s) on the right. Then, the rules are schematically



The classical theory of formal languages defines the family of *regular languages* as the smallest family containing the finite languages that is closed under set-theoretic union ($\cup$), catenation product ($\cdot$), and the Kleene star operation $\mathcal{L}^\star = \{\epsilon\} \cup \cup(\mathcal{L} \cdot \mathcal{L}) \cup \cdots$. Any regular language is then denoted by a regular expression. The operations are taken in the set-theoretic sense, so that for instance one has the identity[1] $(a \cup aa)^\star = (a)^\star$. It is a standard result of the theory that any regular language is recognizable by a deterministic finite automaton, so that this notion of regularity is indeed equivalent to the two combinatorial notions of A-regularity and S-regularity. (Note: the reduction of regular languages to deterministic automata goes via the construction of nondeterministic automata followed by a reduction, the Rabin-Scott theorem, that usually involves an exponential blow-up in the number of states.)

[1]Union, catenation, and Kleene star resemble sum, cartesian product product, and sequence constructions, respectively. However, there is no systematic correspondence since the set-theoretic operations may be applied ambiguously, in contrast to combinatorial constructions that preserve structure. For instance, in the combinatorial world, one has $\mathfrak{S}\{a\} \neq \mathfrak{S}\{a + aa\}$ and the expression $\mathfrak{S}\{a + aa\}$ denotes structures richer than just words over the letter a (see *coverings* on p. 9).

7. Stirling numbers.. These numbers count amongst the most famous ones of combinatorial analysis. They appear in two kinds:

- the *Stirling cycle number* (also called 'of the first kind') $\begin{bmatrix} n \\ k \end{bmatrix}$ enumerates permutations of size n having k cycles;
- the *Stirling partition number* (also called 'of the second kind') $\left\{ {n \atop k} \right\}$ enumerates partitions of an n-set into k nonempty equivalence classes.

The notations $\begin{bmatrix} n \\ k \end{bmatrix}$ and $\left\{ {n \atop k} \right\}$ proposed by Knuth (himself anticipated by Karamata) are nowadays most widespread; see [**71**].

The most natural way to define Stirling numbers is in terms of the "vertical" EGFs when the value of k is kept fixed:

$$\begin{aligned} \sum_{n\geq 0} \begin{bmatrix} n \\ k \end{bmatrix} \frac{z^n}{n!} &= \frac{1}{k!} \left(\log \frac{1}{1-z} \right)^k \\ \sum_{n\geq 0} \left\{ {n \atop k} \right\} \frac{z^n}{n!} &= \frac{1}{k!} \left(e^z - 1 \right)^k . \end{aligned}$$

From there, the bivariate EGFs follow straightforwardly:

$$\begin{aligned} \sum_{n,k\geq 0} \begin{bmatrix} n \\ k \end{bmatrix} u^k \frac{z^n}{n!} &= \exp\left(u \log \frac{1}{1-z} \right) = (1-z)^{-u} \\ \sum_{n,k\geq 0} \left\{ {n \atop k} \right\} u^k \frac{z^n}{n!} &= \exp\left(u(e^z - 1) \right) . \end{aligned}$$

Stirling numbers and their cognates satisfy a host of algebraic relations. For instance, the differential relations of the EGFs imply the recurrences reminiscent of the binomial recurrence

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n-1 \\ k-1 \end{bmatrix} + (n-1) \begin{bmatrix} n-1 \\ k \end{bmatrix}, \qquad \left\{ {n \atop k} \right\} = \left\{ {n-1 \atop k-1} \right\} + k \left\{ {n-1 \atop k} \right\}.$$

By expanding the powers in the vertical EGF of the Stirling partition numbers or by techniques akin to Lagrange inversion, one finds explicit forms

$$\begin{aligned} \begin{bmatrix} n \\ k \end{bmatrix} &= \sum_{0\leq j\leq h\leq n-k} (-1)^{j+h} \binom{h}{j} \binom{n-1+h}{n-k+h} \binom{2n-k}{n-k-h} \frac{(h-j)^{n-k+h}}{h!} \\ \left\{ {n \atop k} \right\} &= \frac{1}{k!} \sum_{j=0}^{r} \binom{k}{j} (-1)^j (k-j)^n. \end{aligned}$$

Though comforting, these forms are not too useful in general. (The one relative to Stirling cycle numbers was obtained by Schlömilch in 1852 [**28**, p. 216].)

A more important relation is that of the generating polynomials of the $\begin{bmatrix} n \\ r \end{bmatrix}$ for fixed n,

$$P_n(u) \equiv \sum_{r=1}^{n} P_n^{(r)} u^r = u \cdot (u+1) \cdot (u+2) \cdots (u+n-1).$$

This nicely parallels the OGF for the $\left\{ {n \atop r} \right\}$ for fixed r

$$\sum_{n=0}^{\infty} \left\{ {n \atop r} \right\} z^n = \frac{z^r}{(1-z)(1-2z)\cdots(1-kz)}.$$

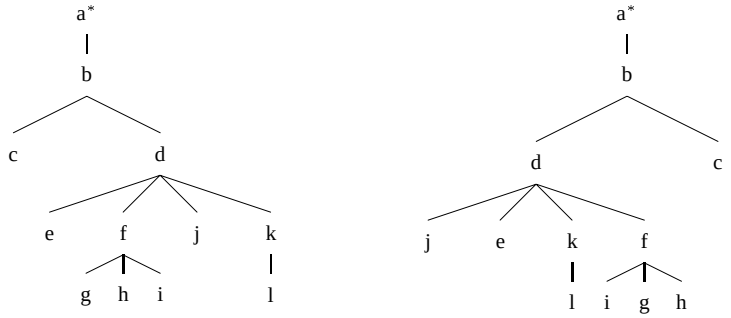
▷ **10.** Schlömilch's formula is established starting from

$$\frac{k!}{n!}\begin{bmatrix} n \\ k \end{bmatrix} = \frac{1}{2i\pi}\oint \log^k \frac{1}{1-z}\,\frac{dz}{z^{n+1}},$$

and performing the change of variable *a la* Lagrange: $z = 1 - e^{-t}$. [**28**, p.216]. ◁

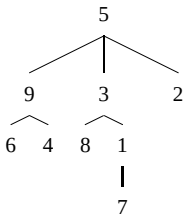
8. Tree concepts. In the abstract graph-theoretic sense, a *forest* is an acyclic (undirected) graph and a *tree* is a forest that consists of just one connected component. A *rooted tree* is a tree in which a specific node is distinguished, the *root*. Rooted trees are drawn with the root either below (the mathematician's and genealogist's convention) or on top (the computer scientist's convention), and in this book, we employ both conventions indifferently. Here are then two planar representations of the same rooted tree

(9)



where the star distinguishes the root. (Tags on nodes, a, b, c, etc, are not part of the tree structure but only meant to discriminate nodes here.) A tree whose nodes are labelled by distinct integers then becomes a *labelled tree*, this in the precise technical sense of Chapter II. Size is defined by the number of nodes (vertices). Here is for instance a labelled tree of size 9:

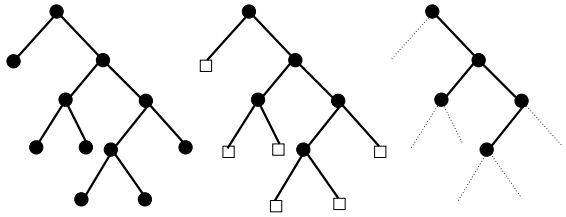
(10)



In a rooted tree, the *outdegree* of a node is the number of its descendants; outdeegree is thus equal to degree (in the graph-theoretic sense, i.e., the number of neighbours) minus 1. Once this convention is clear, one usually abbreviates "outdegree" by "degree" when speaking of rooted trees. A *leaf* is a node without descendant, that is, a node of (out)degree equal to 0. For instance the tree in (10) has 5 leaves. Non-leaf nodes are also called internal nodes.

Many applications from genealogy to computer science require superimposing an additional structure on a graph-theoretic tree. A *plane tree* or *planar tree* is defined as a tree in which subtrees dangling from a common node are ordered between themselves and represented from left to right in order. Thus, the two representations in (9) are equivalent as graph-theoretic trees, but they become distinct objects when regarded as plane trees.

Binary trees play a special role in combinatorics. These are rooted trees in which every nonleaf node has degree 2 exactly as, for instance, in the first two drawings below:



In the second case, the leaves have been distinguished by '□'. The *pruned binary tree* (third representation) is obtained from a regular binary tree by removing the leaves. A binary tree can be fully reconstructed from its pruned version, and a tree of size $2n+1$ always expands a pruned tree of size n.

A few major classes are encountered throughout this book. Here is a summary [2].

General plane trees (Catalan trees)	$\mathcal{G} = \mathcal{Z} \times \mathfrak{S}\{\mathcal{G}\}$	(unlabelled)
Binary trees	$\mathcal{A} = \mathcal{Z} + (\mathcal{Z} \times \mathcal{A} \times \mathcal{A})$	(unlabelled)
Pruned binary trees	$\mathcal{B} = \mathbf{1} + (\mathcal{Z} \times \mathcal{B} \times \mathcal{B})$	(unlabelled)
General nonplane trees (Cayley trees)	$\mathcal{T} = \mathcal{Z} \times \mathfrak{P}\{\mathcal{T}\}$	(labelled)

The corresponding GFs are respectively

$$G(z) = \frac{1-\sqrt{1-4z}}{2}, \quad B(z) = \frac{1-\sqrt{1-4z^2}}{2z}, \quad C(z) = \frac{1-\sqrt{1-4z}}{2z}, \quad T(z) = ze^{T(z)},$$

being respectively of type OGF for the first three and EGF for the last one. The corresponding counts are

$$G_n = \frac{1}{n}\binom{2n-2}{n-1}, \quad A_{2\nu+1} = \frac{1}{\nu+1}\binom{2\nu}{\nu}, \quad B_n = \frac{1}{n+1}\binom{2n}{n}, \quad T_n = n^{n-1}.$$

The common occurrence of the Catalan numbers, C_n ($A_{2\nu+1} = B_\nu = G_{\nu+1} = \mathrm{C}_\nu$)is explained by pruning and by the rotation correspondence described on p. 48.

[2] The term "general" refers to the fact that no degree constraints are imposed.

Bibliography

1. Alfred V. Aho and Margaret J. Corasick, *Efficient string matching: an aid to bibliographic search*, Communications of the ACM **18** (1975), 333–340.
2. David Aldous and Persi Diaconis, *Longest increasing subsequences: from patience sorting to the Baik-Deift-Johansson theorem*, Bull. Amer. Math. Soc. (N.S.) **36** (1999), no. 4, 413–432.
3. Noga Alon and Joel H. Spencer, *The probabilistic method*, John Wiley & Sons Inc., New York, 1992.
4. George E. Andrews, *The theory of partitions*, Encyclopedia of Mathematics and its Applications, vol. 2, Addison–Wesley, 1976.
5. Tom M. Apostol, *Introduction to analytic number theory*, Springer-Verlag, 1976.
6. J. Arney and E. D. Bender, *Random mappings with constraints on coalescence and number of origins*, Pacific Journal of Mathematics **103** (1982), 269–294.
7. Krishna B. Athreya and Peter E. Ney, *Branching processes*, Springer-Verlag, New York, 1972, Die Grundlehren der mathematischen Wissenschaften, Band 196.
8. Jinho Baik, Percy Deift, and Kurt Johansson, *On the distribution of the length of the longest increasing subsequence of random permutations*, Journal of the American Mathematical Society **12** (1999), no. 4, 1119–1178.
9. Edward A. Bender and E. Rodney Canfield, *The asymptotic number of labeled graphs with given degree sequences*, Journal of Combinatorial Theory, Series A **24** (1978), 296–307.
10. Edward A. Bender, E. Rodney Canfield, and Brendan D. McKay, *Asymptotic properties of labeled connected graphs*, Random Structures & Algorithms **3** (1992), no. 2, 183–202.
11. Edward A. Bender and Jay R. Goldman, *Enumerative uses of generating functions*, Indiana University Mathematical Journal (1971), 753–765.
12. Jon Bentley and Robert Sedgewick, *Fast algorithms for sorting and searching strings*, Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM Press, 1997.
13. F. Bergeron, G. Labelle, and P. Leroux, *Combinatorial species and tree-like structures*, Cambridge University Press, Cambridge, 1998, Translated from the 1994 French original by Margaret Readdy, With a foreword by Gian-Carlo Rota.
14. Elwyn R. Berlekamp, *Algebraic coding theory*, Mc Graw-Hill, 1968, Revised edition, 1984.
15. J. Berstel and C. Reutenauer, *Recognizable formal power series on trees*, Theoretical Computer Science **18** (1982), 115–148.
16. Jean Berstel and Dominique Perrin, *Theory of codes*, Academic Press Inc., Orlando, Fla., 1985.
17. Norman Biggs, E. Keith Lloyd, and Robin Wilson, *Graph theory, 1736–1936*, Oxford University Press, 1974.
18. Patrick Billingsley, *Probability and measure*, 2nd ed., John Wiley & Sons, 1986.
19. Béla Bollobás, *Random graphs*, Academic Press, 1985.
20. D. Borwein, S. Rankin, and L. Renner, *Enumeration of injective partial transformations*, Discrete Mathematics **73** (1989), 291–296.
21. Jonathan M. Borwein and Peter B. Borwein, *Strange series and high precision fraud*, American Mathematical Monthly **99** (1992), no. 7, 622–640.
22. Jonathan M. Borwein, Peter B. Borwein, and Karl Dilcher, *Pi, Euler numbers and asymptotic expansions*, American Mathematical Monthly **96** (1989), no. 8, 681–687.
23. Mireille Bousquet-Mélou, *A method for the enumeration of various classes of column-convex polygons*, Discrete Math. **154** (1996), no. 1-3, 1–25.
24. Mireille Bousquet-Mélou and Anthony J. Guttmann, *Enumeration of three-dimensional convex polygons*, Annals of Combinatorics **1** (1997), 27–53.
25. W. H. Burge, *An analysis of binary search trees formed from sequences of nondistinct keys*, JACM **23** (1976), no. 3, 451–454.
26. Noam Chomsky and Marcel Paul Schützenberger, *The algebraic theory of context–free languages*, Computer Programing and Formal Languages (P. Braffort and D. Hirschberg, eds.), North Holland, 1963, pp. 118–161.
27. Julien Clément, Philippe Flajolet, and Brigitte Vallée, *Dynamical sources in information theory: A general analysis of trie structures*, Algorithmica **29** (2001), no. 1/2, 307–369.
28. Louis Comtet, *Advanced combinatorics*, Reidel, Dordrecht, 1974.

29. Robert M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, *On the Lambert W function*, Advances in Computational Mathematics **5** (1996), 329–359.
30. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, New York, 1990.
31. David Cox, John Little, and Donal O'Shea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, 2nd ed., Springer, 1997.
32. H. Davenport, *Multiplicative Number Theory*, revised by H. L. Montgomery, second ed., Springer-Verlag, New York, 1980.
33. F. N. David and D. E. Barton, *Combinatorial chance*, Charles Griffin, London, 1962.
34. N. G. De Bruijn, *On Mahler's partition problem*, Indagationes Math. **10** (1948), 210–220, Reprinted from Koninkl. Nederl. Akademie Wetenschappen, Ser. A.
35. N. G. de Bruijn, *Asymptotic methods in analysis*, Dover, 1981, A reprint of the third North Holland edition, 1970 (first edition, 1958).
36. N. G. De Bruijn and D. A. Klarner, *Multisets of aperiodic cycles*, SIAM Journal on Algebraic and Discrete Methods **3** (1982), 359–368.
37. N. G. De Bruijn, D. E. Knuth, and S. O. Rice, *The average height of planted plane trees*, Graph Theory and Computing (R. C. Read, ed.), Academic Press, 1972, pp. 15–22.
38. A. Dembo, A. Vershik, and O. Zeitouni, *Large deviations for integer partitions*, Markov Processes and Related Fields **6** (2000), no. 2, 147–179.
39. Luc Devroye, *Limit laws for local counters in random binary search trees*, Random Structures & Algorithms **2** (1991), no. 3, 302–315.
40. A. Dvoretzky and Th. Motzkin, *A problem of arrangements*, Duke Mathematical Journal **14** (1947), 305–313.
41. Samuel Eilenberg, *Automata, languages, and machines*, vol. A, Academic Press, 1974.
42. Paul Erdös and Joseph Lehner, *The distribution of the number of summands in the partitions of a positive integer*, Duke Mathematical Journal **8** (1941), 335–345.
43. W. Feller, *An introduction to probability theory and its applications*, vol. 2, John Wiley, 1971.
44. Philippe Flajolet, *Combinatorial aspects of continued fractions*, Discrete Mathematics **32** (1980), 125–161.
45. ———, *Analyse d'algorithmes de manipulation d'arbres et de fichiers*, Cahiers du Bureau Universitaire de Recherche Opérationnelle, vol. 34–35, Université Pierre et Marie Curie, Paris, 1981, 209 pages.
46. ———, *Mathematical methods in the analysis of algorithms and data structures*, Trends in Theoretical Computer Science (Egon Börger, ed.), Computer Science Press, Rockville, Maryland, 1988, (Lecture Notes for *A Graduate Course in Computation Theory*, Udine, 1984), pp. 225–304.
47. Philippe Flajolet, Danièle Gardy, and Loÿs Thimonier, *Birthday paradox, coupon collectors, caching algorithms, and self–organizing search*, Discrete Applied Mathematics **39** (1992), 207–229.
48. Philippe Flajolet, Xavier Gourdon, and Conrado Martínez, *Patterns in random binary search trees*, Random Structures & Algorithms **11** (1997), no. 3, 223–244.
49. Philippe Flajolet, Xavier Gourdon, and Daniel Panario, *The complete analysis of a polynomial factorization algorithm over finite fields*, Journal of Algorithms **40** (2001), no. 1, 37–81.
50. Philippe Flajolet, Yves Guivarc'h, Wojtek Szpankowski, and Brigitte Vallée, *Hidden pattern statistics*, Automata, Languages, and Programming (F. Orejas, P. Spirakis, and J. van Leeuwen, eds.), Lecture Notes in Computer Science, no. 2076, Springer Verlag, 2001, Proceedings of the 28th ICALP Conference, Crete, July 2001., pp. 152–165.
51. Philippe Flajolet, Donald E. Knuth, and Boris Pittel, *The first cycles in an evolving graph*, Discrete Mathematics **75** (1989), 167–215.
52. Philippe Flajolet and Andrew M. Odlyzko, *The average height of binary trees and other simple trees*, Journal of Computer and System Sciences **25** (1982), 171–213.
53. ———, *Random mapping statistics*, Advances in Cryptology (J-J. Quisquater and J. Vandewalle, eds.), Lecture Notes in Computer Science, vol. 434, Springer Verlag, 1990, Proceedings of EUROCRYPT'89, Houtalen, Belgium, April 1989, pp. 329–354.
54. Philippe Flajolet and Helmut Prodinger, *Level number sequences for trees*, Discrete Mathematics **65** (1987), 149–156.
55. Philippe Flajolet, Bruno Salvy, and Gilles Schaeffer, *Airy phenomena and analytic combinatorics of connected graphs*, Preprint, 2002.
56. Philippe Flajolet, Bruno Salvy, and Paul Zimmermann, *Automatic average–case analysis of algorithms*, Theoretical Computer Science **79** (1991), no. 1, 37–109.
57. Philippe Flajolet, Paolo Sipala, and Jean-Marc Steyaert, *Analytic variations on the common subexpression problem*, Automata, Languages, and Programming (M. S. Paterson, ed.), Lecture Notes in Computer Science, vol. 443, 1990, Proceedings of the 17th ICALP Conference, Warwick, July 1990, pp. 220–234.
58. Philippe Flajolet and Michèle Soria, *The cycle construction*, SIAM Journal on Discrete Mathematics **4** (1991), no. 1, 58–60.
59. Philippe Flajolet and Jean-Marc Steyaert, *A complexity calculus for classes of recursive search programs over tree structures*, Proceedings of the 22nd Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1981, pp. 386–393.

60. ———, *A complexity calculus for recursive tree algorithms*, Mathematical Systems Theory **19** (1987), 301–331.
61. Philippe Flajolet, Paul Zimmerman, and Bernard Van Cutsem, *A calculus for the random generation of labelled combinatorial structures*, Theoretical Computer Science **132** (1994), no. 1-2, 1–35.
62. Dominique Foata, *La série génératrice exponentielle dans les problèmes d'énumération*, S.M.S, Montreal University Press, 1974.
63. Dominique Foata, Bodo Lass, and Guo-Niu Han, *Les nombres hyperharmoniques et la fratrie du collectionneur de vignettes*, Seminaire Lotharingien de Combinatoire **47** (2001), Paper B47a.
64. Dominique Foata and Marcel-P. Schützenberger, *Théorie géométrique des polynômes Euleriens*, Lecture Notes in Mathematics, vol. 138, Springer Verlag, 1970.
65. Ira M. Gessel, *Symmetric functions and P–recursiveness*, Journal of Combinatorial Theory, Series A **53** (1990), 257–285.
66. V. Goncharov, *On the field of combinatory analysis*, Soviet Math. Izv., Ser. Math. **8**, 3–48, In Russian.
67. Gaston H. Gonnet, *Expected length of the longest probe sequence in hash code searching*, Journal of the ACM **28** (1981), no. 2, 289–304.
68. Ian P. Goulden and David M. Jackson, *Combinatorial enumeration*, John Wiley, New York, 1983.
69. ———, *Distributions, continued fractions, and the Ehrenfest urn model*, Journal of Combinatorial Theory. Series A **41** (1986), no. 1, 21–31.
70. Xavier Gourdon, *Largest component in random combinatorial structures*, Discrete Mathematics **180** (1998), no. 1-3, 185–209.
71. Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, *Concrete mathematics*, Addison Wesley, 1989.
72. D. H. Greene and D. E. Knuth, *Mathematics for the analysis of algorithms*, Birkhäuser, Boston, 1981.
73. Daniel Hill Greene, *Labelled formal languages and their uses*, Ph.D. thesis, Stanford University, June 1983, Available as Report STAN-CS-83-982.
74. L. J. Guibas and A. M. Odlyzko, *String overlaps, pattern matching, and nontransitive games*, Journal of Combinatorial Theory. Series A **30** (1981), no. 2, 183–208.
75. Laurent Habsieger, Maxime Kazarian, and Sergei Lando, *On the second number of Plutarch*, American Mathematical Monthly **105** (1998), 446–447.
76. Frank Harary and Edgar M. Palmer, *Graphical enumeration*, Academic Press, 1973.
77. Svante Janson, Donald E. Knuth, Tomasz Łuczak, and Boris Pittel, *The birth of the giant component*, Random Structures & Algorithms **4** (1993), no. 3, 233–358.
78. Svante Janson, Tomasz Łuczak, and Andrzej Rucinski, *Random graphs*, Wiley-Interscience, New York, 2000.
79. André Joyal, *Une théorie combinatoire des séries formelles*, Advances in Mathematics **42** (1981), no. 1, 1–82.
80. M. S. Klamkin and D. J. Newman, *Extensions of the birthday surprise*, Journal of Combinatorial Theory **3** (1967), 279–282.
81. S. C. Kleene, *Representation of events in nerve nets and finite automata*, Automata studies, Princeton University Press, Princeton, N. J., 1956, pp. 3–41.
82. Arnold Knopfmacher and Helmut Prodinger, *On Carlitz compositions*, European Journal of Combinatorics **19** (1998), no. 5, 579–589.
83. John Knopfmacher, *Abstract analytic number theory*, Dover, 1990.
84. Donald E. Knuth, *Mathematical analysis of algorithms*, Information Processing 71, North Holland Publishing Company, 1972, Proceedings of IFIP Congress, Ljubljana, 1971, pp. 19–27.
85. ———, *The art of computer programming*, 3rd ed., vol. 1: Fundamental Algorithms, Addison-Wesley, 1997.
86. ———, *The art of computer programming*, 2nd ed., vol. 3: Sorting and Searching, Addison-Wesley, 1998.
87. ———, *Selected papers on analysis of algorithms*, CSLI Publications, Stanford, CA, 2000.
88. Donald E. Knuth, James H. Morris, Jr., and Vaughan R. Pratt, *Fast pattern matching in strings*, SIAM Journal on Computing **6** (1977), no. 2, 323–350.
89. Donald E. Knuth and Ilan Vardi, *Problem 6581 (the asymptotic expansion of $2n$ choose n)*, American Mathematical Monthly **95** (1988), 774.
90. Valentin F. Kolchin, *Random mappings*, Optimization Software Inc., New York, 1986, Translated from *Slučajnye Otobraženija*, Nauka, Moscow, 1984.
91. ———, *Random graphs*, Encyclopedia of Mathematics and its Applications, vol. 53, Cambridge University Press, Cambridge, U.K., 1999.
92. Valentin F. Kolchin, Boris A. Sevastyanov, and Vladimir P. Chistyakov, *Random allocations*, John Wiley and Sons, New York, 1978, Translated from the Russian original *Slučajnye Razmeščeniya*.
93. J. C. Lagarias and A. M. Odlyzko, *Solving low-density subset sum problems*, JACM **32** (1985), no. 1, 229–246.
94. Serge Lang, *Algebra*, Addison-Wesley, Reading, Mass., 1965.
95. ———, *Linear algebra*, Addison-Wesley, Reading, Mass., 1966.
96. V. Lifschitz and B. Pittel, *The number of increasing subsequences of the random permutation*, Journal of Combinatorial Theory, Series A **31** (1981), 1–20.

97. B. F. Logan and L. A. Shepp, *A variational problem for random Young tableaux*, Advances in Mathematics **26** (1977), 206–222.
98. M. Lothaire, *Combinatorics on words*, Encyclopedia of Mathematics and its Applications, vol. 17, Addison–Wesley, 1983.
99. E. Lucas, *Théorie des Nombres*, Gauthier-Villard, Paris, 1891, Reprinted by A. Blanchard, Paris 1961.
100. V. Y. Lum, P. S. T. Yuen, and M. Dodd, *Key to address transformations: A fundamental study based on large existing format files*, Communications of the ACM **14** (1971), 228–239.
101. P. A. MacMahon, *Introduction to combinatory analysis*, Chelsea Publishing Co., New York, 1955, A reprint of the first edition, Cambridge, 1920.
102. Hosam M. Mahmoud, *Evolution of random search trees*, John Wiley, New York, 1992.
103. Conrado Martínez and Xavier Molinero, *A generic approach for the unranking of labeled combinatorial classes*, Random Structures & Algorithms **19** (2001), no. 3-4, 472–497, Analysis of algorithms (Krynica Morska, 2000).
104. A. Meir and J. W. Moon, *On the altitude of nodes in random trees*, Canadian Journal of Mathematics **30** (1978), 997–1015.
105. J. W. Moon, *Counting labelled trees*, Canadian Mathematical Monographs N.1, William Clowes and Sons, 1970.
106. Macdonald Morris, Gabriel Schachtel, and Samuel Karlin, *Exact formulas for multitype run statistics in a random ordering*, SIAM Journal on Discrete Mathematics **6** (1993), no. 1, 70–86.
107. Rajeev Motwani and Prabhakar Raghavan, *Randomized algorithms*, Cambridge University Press, 1995.
108. Donald J. Newman and Lawrence Shepp, *The double dixie cup problem*, American Mathematical Monthly **67** (1960), 58–61.
109. Albert Nijenhuis and Herbert S. Wilf, *Combinatorial algorithms*, second ed., Academic Press, 1978.
110. A. M. Odlyzko, *Periodic oscillations of coefficients of power series that satisfy functional equations*, Advances in Mathematics **44** (1982), 180–205.
111. Richard Otter, *The number of trees*, Annals of Mathematics **49** (1948), no. 3, 583–599.
112. D. Panario and B. Richmond, *Exact largest and smallest size of components*, Algorithmica **31** (2001), no. 3, 413–432.
113. G. Pólya, *Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen*, Acta Mathematica **68** (1937), 145–254.
114. ———, *On the number of certain lattice polygons*, Journal of Combinatorial Theory, Series A **6** (1969), 102–105.
115. G. Pólya and R. C. Read, *Combinatorial enumeration of groups, graphs and chemical compounds*, Springer Verlag, New York, 1987.
116. George Pólya, Robert E. Tarjan, and Donald R. Woods, *Notes on introductory combinatorics*, Progress in Computer Science, Birkhäuser, 1983.
117. Helmut Prodinger, *A note on the distribution of the three types of nodes in uniform binary trees*, Séminaire Lotharingien de Combinatoire **38** (1996), Paper B38b, 5 pages.
118. Andrzej Proskurowski, Frank Ruskey, and Malcolm Smith, *Analysis of algorithms for listing equivalence classes of k-ary strings*, SIAM Journal on Discrete Mathematics **11** (1998), no. 1, 94–109 (electronic).
119. G. N. Raney, *Functional composition patterns and power series reversion*, Transactions of the American Mathematical Society **94** (1960), 441–451.
120. A. Rényi and G. Szekeres, *On the height of trees*, Australian Journal of Mathematics **7** (1967), 497–507.
121. P. Révész, *Strong theorems on coin tossing*, Proceedings of the International Congress of Mathematicians (Helsinki, 1978) (Helsinki), Acad. Sci. Fennica, 1980, pp. 749–754.
122. Gian-Carlo Rota, *Finite operator calculus*, Academic Press, 1975.
123. Salvador Roura and Conrado Martínez, *Randomization of search trees by subtree size*, Algorithms—ESA'96 (Josep Diaz and Maria Serna, eds.), Lecture Notes in Computer Science, no. 1136, 1996, Proceedings of the Fourth European Symposium on Algorithms, Barcelona, September 1996., pp. 91–106.
124. Vladimir N. Sachkov, *Combinatorial methods in discrete mathematics*, Encyclopedia of Mathematics and its Applications, vol. 55, Cambridge University Press, 1996.
125. ———, *Probabilistic methods in combinatorial analysis*, Cambridge University Press, Cambridge, 1997, Translated and adapted from the Russian original edition, Nauka, Moscow, 1978.
126. Arto Salomaa and Matti Soittola, *Automata-theoretic aspects of formal power series*, Springer, Berlin, 1978.
127. Bruno Salvy and John Shackell, *Symbolic asymptotics: multiseries of inverse functions*, Journal of Symbolic Computation **27** (1999), no. 6, 543–563.
128. Robert Sedgewick, *Quicksort with equal keys*, SIAM Journal on Computing **6** (1977), no. 2, 240–267.
129. ———, *Algorithms*, second ed., Addison–Wesley, Reading, Mass., 1988.
130. Robert Sedgewick and Philippe Flajolet, *An introduction to the analysis of algorithms*, Addison-Wesley Publishing Company, 1996.
131. L. A. Shepp and S. P. Lloyd, *Ordered cycle lengths in a random permutation*, Transactions of the American Mathematical Society **121** (1966), 340–357.
132. N. J. A. Sloane, *The on-line encyclopedia of integer sequences*, 2000, Published electronically at `http://www.research.att.com/~njas/sequences/`.

133. N. J. A. Sloane and Simon Plouffe, *The encyclopedia of integer sequences*, Academic Press, 1995.
134. Richard P. Stanley, *Generating functions*, Studies in Combinatorics, M.A.A. Studies in Mathematics, Vol. 17. (G-C. Rota, ed.), The Mathematical Association of America, 1978, pp. 100–141.
135. ———, *Enumerative combinatorics*, vol. I, Wadsworth & Brooks/Cole, 1986.
136. ———, *Hipparchus, Plutarch, Schröder and Hough*, American Mathematical Monthly **104** (1997), 344–350.
137. ———, *Enumerative combinatorics*, vol. II, Cambridge University Press, 1998.
138. Jean-Marc Steyaert, *Structure et complexité des algorithmes*, Doctorat d'état, Université Paris VII, April 1984.
139. Wojciech Szpankowski, *Average-case analysis on algorithms on sequences*, John Wiley, New York, 2001.
140. H. N. V. Temperley, *On the enumeration of the Mayer cluster integrals*, Proc. Phys. Soc. Sect. B. **72** (1959), 1141–1144.
141. ———, *Graph theory and applications*, Ellis Horwood Ltd., Chichester, 1981.
142. Bernard Van Cutsem, *Combinatorial structures and structures for classification*, Comput. Statist. Data Anal. **23** (1996), no. 1, 169–188.
143. J. van Leeuwen (ed.), *Handbook of theoretical computer science*, vol. A: Algorithms and Complexity, North Holland, 1990.
144. E. J. Janse van Rensburg, *The statistical mechanics of interacting walks, polygons, animals and vesicles*, Oxford University Press, Oxford, 2000.
145. A. M. Vershik, *Statistical mechanics of combinatorial partitions, and their limit configurations*, Funktsional'nyĭ Analiz i ego Prilozheniya **30** (1996), no. 2, 19–39.
146. A. M. Vershik and S. V. Kerov, *Asymptotics of the Plancherel measure of the symmetric group and the limiting form of Young tables*, Soviet Mathematical Doklady **18** (1977), 527–531.
147. Jeffrey Scott Vitter and Philippe Flajolet, *Analysis of algorithms and data structures*, Handbook of Theoretical Computer Science (J. van Leeuwen, ed.), vol. A: Algorithms and Complexity, North Holland, 1990, pp. 431–524.
148. J. Vuillemin, *A unifying look at data structures*, Communications of the ACM **23** (1980), no. 4, 229–239.
149. Michael S. Waterman, *Introduction to computational biology*, Chapman & Hall, 1995.
150. E. T. Whittaker and G. N. Watson, *A course of modern analysis*, fourth ed., Cambridge University Press, 1927, Reprinted 1973.
151. Herbert S. Wilf, *Some examples of combinatorial averaging*, American Mathematical Monthly **92** (1985), 250–261.
152. ———, *Combinatorial algorithms: An update*, CBMS–NSF Regional Conference Series, no. 55, Society for Industrial and Applied Mathematics, Philadelphia, 1989.
153. ———, *Generatingfunctionology*, Academic Press, 1990.
154. E. Maitland Wright, *The number of connected sparsely edged graphs*, Journal of Graph Theory **1** (1977), 317–330.
155. ———, *The number of connected sparsely edged graphs. II. Smooth graphs*, Journal of Graph Theory **2** (1978), 299–305.
156. ———, *The number of connected sparsely edged graphs. III. Asymptotic results*, Journal of Graph Theory **4** (1980), 393–407.
157. Robert Alan Wright, Bruce Richmond, Andrew Odlyzko, and Brendan McKay, *Constant time generation of free trees*, SIAM Journal on Computing **15** (1985), no. 2, 540–548.
158. Paul Zimmermann, *Séries génératrices et analyse automatique d'algorithmes*, Ph. d. thesis, École Polytechnique, 1991.

Index

Spectrally Bounded Sequences, Codes and States: Graph Constructions and Entanglement

Matthew G. Parker

Code Theory Group, Inst. for Informatikk, HIB,
University of Bergen, Norway
E-mail: matthew@ii.uib.no,
Web: http://www.ii.uib.no/∼matthew/MattWeb.html

Abstract. A recursive construction is provided for sequence sets which possess good Hamming Distance and low Peak-to-Average Power Ratio (PAR) under any Local Unitary Unimodular Transform. We identify a subset of these sequences that map to binary indicators for linear and nonlinear Factor Graphs, after application of subspace Walsh-Hadamard Transforms. Finally we investigate the quantum PAR_l measure of 'Linear Entanglement' (LE) under any Local Unitary Transform, where optimum LE implies optimum weight hierarchy of an associated linear code.

1 Introduction

Golay Complementary sequences of length 2^n form sequences with Peak-to-Average Power Ratio (PAR) ≤ 2 under the one-dimensional continuous Discrete Fourier Transform (DFT_1^∞) [9]. The upper PAR bound of 2 follows by forming these Complementary Sequences using Rudin-Shapiro construction [25, 26]. This set is the union of certain quadratic cosets of Reed-Muller (RM) $(1, n)$ [5]. Moreover the quadratic coset representatives can be viewed as 'line graphs' in Algebraic Normal Form (ANF) [21]. As these sequences are a subset of $\mathrm{RM}(2, n)$, the Hamming Distance, D, between sequences in the set satisfies $D \geq 2^{n-2}$. The problem of finding error-correcting codes where each codeword also has low PAR has application to Orthogonal Frequency Division Multiplexing (OFDM) communications systems [11]. However the fundamental codeset identified by Davis and Jedwab [5] (DJ sequences) suffers from vanishing rate as n increases, and much higher rates are possible and desirable, where $\mathrm{PAR} \leq O(n)$ [27, 22]. A generalisation of Rudin-Shapiro construction to other starting seeds [16, 17]. allows inclusion of more low PAR quadratic cosets of $\mathrm{RM}(1, n)$ in the code, thereby improving code rate somewhat. Higher degree cosets...etc can also be added, increasing code rate at price of distance, D, which decreases. However these rate improvements are marginal. In this paper we present a construction for much larger codesets of sequences with $\mathrm{PAR} \leq 2^t$, comprising ANFs up to degree u, where $u \leq t$ for $t > 1$, and $u = 2$ for $t = 1$ [19]. These codesets have $\mathrm{PAR} \leq 2^t$ under **all** Linear Unimodular Unitary Transforms (LUUTs), including one and multi-dimensional continuous DFTs. As LUUTs include the Walsh-Hadamard

Transform (WHT) then our construction gives large codesets of Almost-Bent functions [3, 23]. The functions are cryptographically even stronger, as the binary sequences are distant from linear sequences over all alphabets, not just over Z_2. We then describe a mapping of a subset of the bipolar sequences, generated using our construction, to Factor Graphs [12]. By applying tensor products of Hadamard and Identity kernels to our bipolar sequence we transform to a Factor Graph in a Normal Realisation [7] representing a linear or nonlinear error-correcting code. This transformation provides spectral characterisation for Factor Graphs (and Quantum Factor Graphs [15]). Finally we present PAR_l, which is a partial measure of quantum entanglement and measures PAR under **all** Linear Unitary Transforms (LUTs) [17, 18]. We also define 'Linear Entanglement' (LE), and 'Stubborness of Entanglement' (SE), which is a series of parameters related to PAR_l over all sequence subspaces. At least in the bipartite quadratic case, a length 2^n bipolar sequence with optimal LE and SE represents a $[n, k, d]$ binary linear code with optimal weight hierarchy. We conjecture that optimally entangled subsystems represent optimal linear and nonlinear codes - and vice versa. A similar relationship between secrecy and entanglement has recently been highlighted by [4].

2 A Construction For Low PAR Error-Correcting Codes

Joint work with C.Tellambura [19]
PAR is a spectral measure. We must therefore define the transforms over which the spectrum is computed:

2.1 Definitions

Definition 1 $\boldsymbol{L_n}$ *is the infinite set of length* 2^n *complex linear unimodular sequences,* $\boldsymbol{l} = (l_0, l_1, \ldots, l_{2^n-1})$*, where* $|l_i| = |l_j|$*,* $\forall i, j$*,* $\sum_{i=0}^{2^n-1} |l_i|^2 = 1$*, and,*

$$\boldsymbol{l} = \{2^{\frac{-n}{2}}(a_0, b_0) \otimes (a_1, b_1) \otimes \ldots \otimes (a_{n-1}, b_{n-1})\}$$

where $\otimes$ *means 'tensor product'.*

Definition 2 *A* $2^n \times 2^n$ *Linear Unimodular Unitary Transform (LUUT) matrix* $\mathbf{L}$ *has rows taken from* $\boldsymbol{L_n}$ *such that* $\mathbf{L}\mathbf{L}^\dagger = \mathbf{I_{2^n}}$*, where* $\dagger$ *means conjugate transpose, and* $\mathbf{I_{2^n}}$ *is the* $2^n \times 2^n$ *identity matrix.*

Definition 3 $\boldsymbol{G_n}$ *is the infinite set of length* 2^n *complex linear sequences,* $\boldsymbol{l} = (l_0, l_1, \ldots, l_{2^n-1})$*, where* $\sum_{i=0}^{2^n-1} |l_i|^2 = 1$ *and,*

$$\boldsymbol{l} = \{2^{\frac{-n}{2}}(a_0, b_0) \otimes (a_1, b_1) \otimes \ldots \otimes (a_{n-1}, b_{n-1})\}$$

Note that $\boldsymbol{G_n} \supset \boldsymbol{L_n}$*.*

Definition 4 *A* $2^n \times 2^n$ *Linear Unitary Transform (LUT) matrix* $\mathbf{G}$ *has rows taken from* $\boldsymbol{G_n}$ *such that* $\mathbf{G}\mathbf{G}^\dagger = \mathbf{I_{2^n}}$*. LUUTs are a special case of LUT.*

Let s_i be an element of a length 2^n vector, $\boldsymbol{s}$. PAR($\boldsymbol{s}$) is computed by measuring maximum possible correlation of $\boldsymbol{s}$ with **any** length 2^n 'linear' unimodular sequence, $\boldsymbol{l} \in \boldsymbol{L_n}$:

Definition 5 $PAR(\boldsymbol{s}) = 2^n max_{\boldsymbol{l}}(|\boldsymbol{s} \cdot \boldsymbol{l}|^2)$
where $\boldsymbol{l} \in \boldsymbol{L_n}$ *and* $\cdot$ *means 'inner product' [17].*

Let $\boldsymbol{x} = \{x_0, x_1, \ldots, x_{n-1}\}$. Then $p(\boldsymbol{x})$: $Z_2^n \rightarrow Z_2$ has a bipolar representation, $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})} = (s_0, s_1, \ldots, s_{2^n-1})$, where $s_i = (-1)^{p(x_0=i_0, x_1=i_1, \ldots, x_{n-1}=i_{n-1})}$, and $i = \sum_{k=0}^{n-1} i_k 2^k$ is a radix-2 decomposition of i.

2.2 Construction

This paper focuses on a special case of a more general construction. Here, all x_i are two-state binary variables, and the fundamental recursion is based on Walsh-Hadamard Transform (WHT) kernels. The more general construction is presented in [19]. We now present the construction:

$$\begin{array}{l} p(\boldsymbol{x}) = \sum_{j=0}^{L-2} \sum_{l=0}^{t-1} x_{\pi(tj+l)} f_{l,j}(x_{\pi(t(j+1))}, x_{\pi(t(j+1)+1)}, \ldots, x_{\pi(t(j+2)-1)}) \\ + \sum_{j=0}^{L-1} g_j(x_{\pi(tj)}, x_{\pi(tj+1)}, \ldots, x_{\pi(tj+t-1)}) \end{array} \quad (1)$$

where $n = Lt$, π permutes Z_n, and where $f_{l,j} : Z_2^t \rightarrow Z_2$ is such that $f_{\gamma_j} = (f_{0,j}, f_{1,j}, \ldots, f_{t-1,j})$ is an invertible boolean function (permutation polynomial) from $Z_2^t \rightarrow Z_2^t$, governed by the permutation, $i' = \gamma_j(i)$, where $i' = \sum_{l=0}^{t-1} i'_l 2^l$ is a radix-2 decomposition, $i'_l = f_{l,j}(i_0, i_1, \ldots, i_{t-1})$, and each γ_j permutes Z_t. To avoid unnecessary duplications, we exclude the f_{γ_j} where one or more $f_{l,j}$ has a '+1' constant offset, and also the cases where all $f_{l,j}$ are monomials, except when f_{γ_j} is the identity function.

Theorem 1 *[19] The length* $N = 2^n$ *bipolar sequence* $\boldsymbol{s} = (-1)^{\boldsymbol{p}}$ *satisfies* $PAR(s) \leq 2^t$ *under all LUUTs, where* $\boldsymbol{p}$ *is generated using construction (1).*

Proof. (sketch) Let m factor fully as $m = \prod_{i=0}^{F-1} p_i$, p_i not necessarily distinct. A length m vector, $\boldsymbol{l}$, is defined linear if it satisfies $\boldsymbol{l} = \bigotimes_{i=0}^{F-1} \boldsymbol{v_i}$ where length($\boldsymbol{v_i}$) = p_i, and $\sum_{j=0}^{m-1} |l_j|^2 = 1$. Let $\boldsymbol{E_j}$ and $\boldsymbol{A_j}$, $1 \leq j \leq L$, be a series of $N \times N$ and $N \times N^j$ complex matrices, respectively, where $\boldsymbol{A_1} = \boldsymbol{E_1}$ is unitary. Let the rows of $\boldsymbol{A_{j-1}}$, $(\boldsymbol{a_{0,j-1}}, \boldsymbol{a_{1,j-1}}, \ldots, \boldsymbol{a_{N-1,j-1}})$, form a complementary set of N sequences under any $N^{j-1} \times N^{j-1}$ unitary transform with linear unimodular rows. Let $\boldsymbol{l}$ and $\boldsymbol{l_j}$ be normalised linear rows of length N^{j-1} and N, respectively. Let $\boldsymbol{r} = \boldsymbol{A_{j-1}} \boldsymbol{l}$. Let γ permute Z_N. Construct the $N \times N^j$ matrix, $\boldsymbol{A_j}$, such that $\boldsymbol{a_{i,j}} = ((\boldsymbol{a_{\gamma(0),j-1}} | \boldsymbol{a_{\gamma(1),j-1}} | \ldots, | \boldsymbol{a_{\gamma(N-1),j-1}}) \odot (\boldsymbol{e_{i,j}} \otimes \boldsymbol{1}))$ where $\boldsymbol{x} \odot \boldsymbol{y} = (x_0 y_0, x_1 y_1, \ldots, x_{N^j-1} y_{N^j-1})$, $\boldsymbol{1}$ is the length N^{j-1} all-ones vector, $\boldsymbol{e_{i,j}}$ is the ith row of $\boldsymbol{E_j}$, and $'|'$ means concatenation. The rows of $\boldsymbol{A_j}$ form a complementary N-set under any unitary transform if $\boldsymbol{r'} = \boldsymbol{A_j}(\boldsymbol{l_j} \otimes \boldsymbol{l})$ satisfies, $\sum_{k=0}^{N-1} |r'_i|^2 = 1$. This follows if $\sum_{i=0}^{N-1} |\sum_{k=0}^{N-1} (r_{\gamma(k)} e_{i,k} l_k)|^2 = 1$, for $r_k, e_{i,k}$ and l_k elements of

$\boldsymbol{r}$,$\boldsymbol{e_{i,j}}$ and $\boldsymbol{l_j}$, respectively. This is true if $\boldsymbol{E_j}$ is unitary, and if $\boldsymbol{e_{i,j}} \odot \boldsymbol{l_j}$ is unimodular, which follows if $\boldsymbol{e_{i,j}}$ and $\boldsymbol{l_j}$ are unimodular. Construction (1) occurs when successive $\boldsymbol{A_j}$ are recursively generated, where all $\boldsymbol{E_i}$ are $2^t \times 2^t$ WHTs. The γ permutation essentially maps to f_γ, and concatenation is widened to a more general permutation, π, over all linear variables. ∎

Theorem 2 *For a fixed t, let $\boldsymbol{P}$ be the codeset of length 2^n binary sequences of degree μ or less, generated using (1). Then,*

$$\begin{aligned} \frac{|\boldsymbol{P}|}{2^{n+1}} &\leq \frac{(\frac{\Gamma}{t!})^{\frac{n}{t}-1} n!(2^{2^t-t-1})^{\frac{n}{t}}}{2t!}, & \mu = 2 \\ &\leq \frac{((2^t-1)!)^{\frac{n}{t}-1} n!(2^{2^t-t-1})^{\frac{n}{t}}}{2t!}, & \mu \geq 2 \end{aligned} \tag{2}$$

where $\Gamma = \prod_{i=0}^{t-1}(2^t - 2^i) = |GL(t,2)|$. (GL is the General Linear Group). (Only for $t = 1$ is the upper bound exact).

Proof. By counting arguments we can show that, for $\mu = 2$,

$$\frac{|\boldsymbol{P}|}{2^{n+1}} \leq \frac{\prod_{l=1}^{t} \binom{\frac{ln}{t}}{\frac{n}{t}}}{t!} \times \frac{(\frac{n}{t})!^t}{2} \times (\frac{\Gamma}{t!})^{\frac{n}{t}-1} \times (2^{(t/2)})^{\frac{n}{t}}$$

For $\mu \geq 2$, we replace $\frac{\Gamma}{t!}$ with $\frac{(2^t)!}{2^t}$, which is the number of permutations excluding those with a constant offset, '+1'. The Theorem follows. ∎

In Section 2.4 we show how to generate all degree-one permutation polynomials, via an isomorphism to the General Linear Group, where the number of degree-one permutation polynomials is Γ.

2.3 Examples

The $2^n \times 2^n$ Walsh-Hadamard (WHT) and Negahadamard (NHT) Transform matrices are $\bigotimes_{i=0}^{n-1} \boldsymbol{H}$, and $\bigotimes_{i=0}^{n-1} \boldsymbol{N}$, respectively, where $\boldsymbol{H} = \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ and $\boldsymbol{N} = \left(\begin{smallmatrix} 1 & i \\ 1 & -i \end{smallmatrix}\right)$, and $i^2 = -1$. DFT_1^∞ is the set of $2^n \times 2^n$ matrices, the union of whose rows form a subset of $\boldsymbol{L_n}$ such that each row satisfies $a_i = 1$, $b_i = \omega^{ik}$ for some fixed k, and ω is a complex root of unity (see Definition 1). These three transforms are used as 'spot-checks' in the examples to validate the PAR upper-bound.

Example 1 Let γ_j be the identity permutation $\forall j$. Then,
$f_{l,j}(x_{\pi(t(j+1))}, x_{\pi(t(j+1)+1)}, \ldots, x_{\pi(t(j+2)-1)}) = x_{\pi(t(j+1)+l)}$, and (1) becomes,

$$p(\boldsymbol{x}) = \sum_{j=0}^{L-2} \sum_{l=0}^{t-1} x_{\pi(tj+l)} x_{\pi(t(j+1)+l)} + \sum_{j=0}^{L-1} g_j(x_{\pi(tj}, x_{\pi(tj+1)}, \ldots, x_{\pi(tj+t-1)}) \tag{3}$$

When $\deg(g_j) < 2$, $\forall j$, it is well-known that $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$ is Bent (PAR = 1 under the WHT) for L even [14] and (perhaps not known) that $\boldsymbol{s}$ has PAR $= 2^t$

under the WHT for L odd. In general, for any g_j, s has PAR $\leq 2^t$ under all LUUTs. For example, if $L = 4$ and,

$$p(\boldsymbol{x}) = x_0x_3 + x_1x_4 + x_2x_5 + x_3x_6 + x_4x_7 + x_5x_8 + x_6x_9 + x_7x_{10} + x_8x_{11}$$

then $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$ has PAR = 1.0 under the WHT, PAR = 1.0 under the NHT, and PAR = 7.09 under DFT_1^∞. Similarly, let $g_0(x_0, x_1, x_2) = x_1x_2$, $g_1(x_3, x_4, x_5) = x_3x_4x_5$, and $g_2(x_6, x_7, x_8) = 0$. Then $\boldsymbol{s'} = (-1)^{p(\boldsymbol{x})+g_0+g_1+g_2}$ has PAR = 4.0 under the WHT, PAR = 2.0 under the NHT, and PAR = 7.54 under DFT_1^∞. In all cases, PAR ≤ 8.0 under any LUUT.

Example 2, PAR $\leq$ 2.0 Let $t = 1$. Then we have one possible permutation polynomial, namely, $f_\gamma = x$, (we exclude $f_\gamma = x + 1$). From (1) we obtain,

$$p(\boldsymbol{x}) = \textstyle\sum_{j=0}^{L-2} x_{\pi(j)}x_{\pi(j+1)} + c_jx_j + d, \qquad c_j, d \in Z_2 \tag{4}$$

This is exactly the DJ set of binary quadratic cosets of RM$(1, n)$, where $n = L$ [5]. This set has PAR ≤ 2.0 under DFT_1^∞ [5]. Such sequences are Bent for n even [14, 23] and, in [16, 17] it was shown that such a set has PAR = 2.0 under the WHT for n odd, and also, under the NHT, has PAR = 1.0 for $n \neq 2 \bmod 3$ (NegaBent), and PAR = 2.0 for $n = 2 \bmod 3$. More generally the DJ set has PAR ≤ 2.0 under any LUUT [17], and this agrees with Theorem 1. For example, let $p(\boldsymbol{x}) = x_0x_4 + x_4x_1 + x_1x_2 + x_2x_3 + x_1 + 1$. Then $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$ has PAR = 2.0 under the WHT, PAR = 2.0 under the NHT, and PAR = 2.0 under DFT_1^∞. The DJ set, being cosets of $R(2, n)$, forms a codeset with Hamming Distance, $D >= 2^{n-2}$. The rate of the DJ codeset follows $\frac{(\frac{n!}{2})2^{n+1}}{2^{2^n}}$ as n increases. This is their primary drawback as the code rate vanishes rapidly as n increases.

Example 3, PAR $\leq$ 4.0 [5, 22, 17, 23] have all proposed techniques for the inclusion of further quadratic cosets, so as to improve rate at the price of increased PAR. We here propose an improved rate code (although still vanishing), where PAR ≤ 4.0. To achieve this we set $t = 2$ in (1). There are $\frac{(2^t)!}{2^t t!} = 3$ valid permutation polynomials, $f_\gamma = (f_0, f_1)$. These polynomials map from $Z_2^2 \to Z_2^2$, and are taken from the set,

$$f_\gamma(x_0, x_1) \in \{(x_0, x_1), (x_0 + x_1, x_1), (x_0, x_0 + x_1)\}$$

Substituting for $f_{l,j}$ and g_j in (1) gives a large set of polynomials with PAR≤ 4.0 under all LUUTs. We now list, for this construction, the $p(\boldsymbol{x})$ arising from the the 3 invertible polynomial functions, f_γ, for one 'section' of the polynomial, i.e. for $L = 2$, where we fix π to the identity permutation.

$$\begin{aligned}
p(\boldsymbol{x}) &= x_0x_2 + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \mathrm{RM}(1, 4)\\
p(\boldsymbol{x}) &= x_0(x_2 + x_3) + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \mathrm{RM}(1, 4)\\
p(\boldsymbol{x}) &= x_0x_2 + x_1(x_2 + x_3) + c_0x_0x_1 + c_1x_2x_3 + \mathrm{RM}(1, 4)
\end{aligned}$$

where $c_0, c_1 \in Z_2$. The quadratic part of each of these 3 functions is isomorphic to a distinct invertible boolean $t \times t$ matrix, where $t = 2$ (Section 2.4), as the

permutation polynomials form a group which is isomorphic to the General Linear Group, $\mathrm{GL}(t,2)$, where $|\mathrm{GL}(t,2)| = \prod_{i=0}^{t-1}(2^t - 2^i)$ [13]. Two of the 3 quadratic functions are inequivalent under permutation of the four variable indices, e.g.,

$$p(\boldsymbol{x}) = x_0x_2 + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \mathrm{RM}(1,4)$$
$$p(\boldsymbol{x}) = x_0(x_2 + x_3) + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \mathrm{RM}(1,4)$$

An upper bound on $|\boldsymbol{P}|$ is given by Theorem 2, (2). Substituting $t = 2$ into (2),

$$\frac{|\boldsymbol{P}|}{2^{n+1}} < n!2^{\frac{n-4}{2}}3^{\frac{n}{2}-1} \tag{5}$$

An exact enumeration and construction for this set remains open, due to extra 'hidden' symmetries. Computationally we are able to calculate the exact number of quadratic coset leaders for $n = 4, 6, 8, 10$, and these are compared to the upper bound of (5) in Table 1. They are also compared to the number of quadratic coset leaders, $(= \frac{n!}{2})$ in the binary DJ codeset (Example 2). By assigning $t = 2$

Table 1. The Number of Quadratic Coset Leaders for Construction (1) when $t = 2$

n	4	6	8	10
Theorem 2, (5),(2), $\lvert\boldsymbol{P}\rvert/2^{n+1}$	72	12960	4354560	2351462400
Exact Computation	36	9240	4086096	2317593600
$\frac{\text{DJ Code}}{2^{n+1}}$	12	360	20160	1814400
$\log_2(\lvert\boldsymbol{P}\rvert/2^{n+1})$	6.2	13.7	22.1	31.1
$\log_2$(Number of quadratics)	6	15	28	45

we have a construction for a much larger codeset than the DJ codeset and with the same Hamming Distance, $D = 2^{n-2}$, but the price paid is that the PAR is now upper-bounded by 4.0 instead of 2.0. For example, let,
$p(\boldsymbol{x}) = x_0x_2 + x_1x_2 + x_1x_6 + x_2x_5 + x_6x_3 + x_6x_5 + x_5x_4 + x_3x_7 + x_0x_1 + x_5x_3 + x_7 + x_1$
Then $\boldsymbol{s} = (-1)^{\boldsymbol{p}}$ has PAR = 1.0 under the WHT, PAR = 2.0 under the NHT, and PAR = 3.43 under DFT_1^∞.

Example 4, PAR $\leq$ 8.0 Set $t = 3$ in (1). There are now $\frac{(2^t)!}{2^t t!} = 840$ valid permutation polynomials, $f_\gamma = (f_0, f_1, f_2)$. These polynomials map from $Z_2^3 \to Z_2^3$. Moreover, $(2^3-1)(2^3-2)(2^3-2^2)/t! = \frac{168}{6} = 28$ of the polynomials are degree-one permutations leading to quadratic forms, $p(\boldsymbol{x})$, and can be represented by the following 7 permutation polynomials.

$f_\gamma(x_0, x_1, x_2) \in \{$
$(x_0, x_1, x_2), (x_0 + x_2, x_1, x_2), (x_0 + x_2, x_1 + x_2, x_2), (x_0 + x_1 + x_2, x_1, x_2),$
$(x_0 + x_1, x_1 + x_2, x_2), (x_0 + x_1 + x_2, x_1 + x_2, x_2), (x_0 + x_2, x_1 + x_0, x_2 + x_0 + x_1)\}$

Substituting for $f_{l,j}$ and g_j in (1) gives a large set of polynomials with PAR$\leq$ 8.0 under all LUUTs. We now list, for this construction, all quadratic $p(\boldsymbol{x})$ arising

from the 7 inequivalent degree-one permutation polynomials, f_γ, for one 'section' of the polynomial, i.e. for $L = 2$, where π is fixed as the identity permutation.

$$
\begin{aligned}
p(\boldsymbol{x}) &= x_0x_3 + x_1x_4 + x_2x_5 + g(\boldsymbol{x}) \\
p(\boldsymbol{x}) &= x_0x_3 + x_0x_5 + x_1x_4 + x_2x_5 + g(\boldsymbol{x}) \\
p(\boldsymbol{x}) &= x_0x_3 + x_0x_5 + x_1x_4 + x_1x_5 + x_2x_5 + g(\boldsymbol{x}) \\
p(\boldsymbol{x}) &= x_0x_3 + x_0x_4 + x_0x_5 + x_1x_4 + x_2x_5 + g(\boldsymbol{x}) \\
p(\boldsymbol{x}) &= x_0x_3 + x_0x_4 + x_1x_4 + x_1x_5 + x_2x_5 + g(\boldsymbol{x}) \\
p(\boldsymbol{x}) &= x_0x_3 + x_0x_4 + x_0x_5 + x_1x_4 + x_1x_5 + x_2x_5 + g(\boldsymbol{x}) \\
p(\boldsymbol{x}) &= x_0x_3 + x_0x_5 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_2x_5 + g(\boldsymbol{x})
\end{aligned}
$$

where $g(\boldsymbol{x}) = c_0x_0x_1 + c_1x_0x_2 + c_2x_1x_2 + c_3x_0x_1x_2 + c_4x_3x_4 + c_5x_3x_5 + c_6x_4x_5 + c_7x_3x_4x_5 + \mathrm{RM}(1,6)$, and $c_0, c_1, \ldots, c_7 \in Z_2$. An upper bound to $|\boldsymbol{P}|$ can be computed from Theorem 2, (2), and the upper bound is compared to the total number of quadratics in n binary variables in Table 2. As with $t = 2$, an

Table 2. The Number of Quadratic Coset Leaders for Construction (1) when $t = 3$

n	6	9	12	15
Theorem 2, (2), $\log_2(\lvert\boldsymbol{P}\rvert/2^{n+1})$	16.7	33.5	51.7	70.9
$\log_2$(Number of quadratics)	15	36	66	105

exact enumeration and construction for this set remains open, due to extra 'hidden' symmetries. By assigning $t = 3$ we have a construction for a codeset with Hamming Distance, $D \geq 2^{n-2}$ and PAR ≤ 8.0 under all LUUTs.

For $t = 3$ we can also include cubic forms in Construction (1). There are $\frac{5040-168}{6} = 812$ degree 2 permutation polynomials, $f_\gamma = (f_0, f_1, f_2)$, that map from $Z_2^3 \rightarrow Z_2^3$, and lead to cubic forms, $p(\boldsymbol{x})$. This set can be represented by 147 degree 2 permutation polynomials which are inequivalent under variable permutation, and these are listed at [20]. (Along with the 7 inequivalent degree 1 permutation polynomials, this makes a total of 154 inequivalent permutation polynomials for $t = 3$ [10, 28]). Substituting for $f_{l,j}$ and g_j in (1) gives a large set of polynomials with PAR$\leq$ 8.0 under all LUUTs, and Hamming Distance, $D \geq 2^{n-3}$. An upper bound to $|\boldsymbol{P}|$ can be computed from Theorem 2, (2), and the upper bound is compared to the total number of quadratics and cubics in n binary variables in Table 3. Here is an example from this codeset, where ijk, uv

Table 3. The Number of Cubic and Quadratic Coset Leaders for Construction (1) when $t = 3$

n	6	9	12	15
Theorem 2, (2), $\log_2(\lvert\boldsymbol{P}\rvert/2^{n+1})$	23.6	46.3	70.4	95.5
$\log_2$(Number of quadratics and cubics)	35	120	286	560

is short for $x_i x_j x_k + x_u x_v$. Let,

$$p(\boldsymbol{x}) = 034, 035, 045, 135, 145, 234, 235, 245, 367, 368, 378, 567, 568, 69A, 79A, 7AB,$$
$$89A, 345, 9AB, 03, 05, 14, 24, 25, 36, 38, 47, 58, 69, 6A, 6B, 7A, 7B, 89, 8B, 67, 78, AB$$

then $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$ has PAR = 4.0 under the WHT, PAR = 6.625 under the NHT, and PAR = 7.66 under DFT_1^∞. In all cases, PAR ≤ 8.0.

2.4 A Matrix Construction for all Quadratic Codes from (1)

Each degree-one permutation polynomial, f_γ from $Z_2^t \rightarrow Z_2^t$ can be viewed as a $t \times t$ binary adjacency matrix. Let $x = \{x_0, x_1, \ldots, x_{t-1}\}$. We can write,

$$M \Leftrightarrow f_\gamma(x) = (f_0(x), f_1(x), \ldots, f_{t-1}(x)), \qquad M = \{m_{i,l}\}, \deg(f_l(\boldsymbol{x})) = 1, \text{ and}$$
$$m_{i,l} = 1 \qquad \text{if } x_i \in f_l(x) \qquad m_{i,l} = 0 \qquad \text{otherwise}$$

The mapping is an isomorphism from the degree-one permutation polynomials to the General Linear Group, $G = \mathrm{GL}(t, 2)$, of all binary $t \times t$ invertible matrices [13]. To construct all quadratic sequences, $p(\boldsymbol{x})$, for a given n and t we need to construct all degree one permutation polynomials, f_γ. These can, in turn be constructed by generating all members of $G = \mathrm{GL}(t, 2)$, and this is accomplished as follows [1, 2].

Definition 6 *A binary $t \times t$ 'transvection' matrix, X_{ab}, satisfies,*

$$X_{ab} = \{u_{i,j}\}, \text{ where}$$
$$u_{i,j} = 1, \quad i = j, \text{ and } i = a, j = b \qquad u_{i,j} = 0, \quad \text{otherwise}$$

Definition 7 *The Borel subgroup of G over Z_2 is the $t \times t$ upper-triangular binary matrices, B.*

Definition 8 *The Weyl subgroup of G is the $t \times t$ permutation matrices, W.*

Assign a fixed ordering, O, to the $\frac{\binom{t}{2}}{2}$ matrices, X_{ab}, $a < b$. Let $w \in W$ be a permutation of Z_t and its associated $t \times t$ permutation matrix. For each w, form the matrix product, X_w, comprising all X_{ab} which satisfy $a < b = w(a) > w(b)$, where the X_{ab} in X are ordered according to O.

Theorem 3 *[1, 2]*

$$G = X'_w W B \tag{6}$$

where X'_w is any sub-product of X_w that maintains the ordering of the X_{ab} matrices in X_w. This is the 'Bruhat' decomposition.

All quadratic constructions using (1) can be constructed using Theorem 3., where $|\boldsymbol{G}| = \Gamma = \prod_{i=0}^{t-1}(2^t - 2^i)$.

3 Graphical Representations

Joint work with V.Rijmen [18]

We now identify a subset of the length 2^n sequence constructions of (1), where $(-1)^{p(\boldsymbol{x})}$ exhibits a bipolar $\leftrightarrow$ binary equivalence under transform by a tensor product of combinations of $\boldsymbol{H}$ and $\boldsymbol{I}$ 2×2 matrices. The resultant length 2^n binary sequences can be interpreted as indicators for binary linear or nonlinear $[n, k, d]$ error-correcting codes. In such cases, $p(\boldsymbol{x})$ is closely related to a Normal Realisation for the Factor Graph of the associated $[n, k, d]$ code [7]. Let $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$.

Definition 9 *"H acting on i" means the action of the* $2^n \times 2^n$ *transform,* $\mathbf{I} \otimes \ldots \otimes \mathbf{I} \otimes \mathbf{H} \otimes \mathbf{I} \otimes \ldots \otimes \mathbf{I}$ *on* $\boldsymbol{s}$*, where* $\mathbf{H}$ *is preceded by* i $\mathbf{I}$ *matrices, and followed by* $n - i - 1$ $\mathbf{I}$ *matrices. We write this as* $H(i)$*, or* $H(i)[\boldsymbol{s}]$*.*

Definition 10 *Let* $\mathbf{T_C}$*,* $\mathbf{T_{C^\perp}}$ *be integer sets chosen so that* $\mathbf{T_C} \cap \mathbf{T_{C^\perp}} = \emptyset$*, and* $\mathbf{T_C} \cup \mathbf{T_{C^\perp}} = \{0, 1, \ldots, n-1\}$*. This is a bipartite splitting of* $\{0, 1, \ldots, n-1\}$*. Let us also partition the variable set* $\mathbf{x}$ *as* $\mathbf{x} = \mathbf{x_C} \cup \mathbf{x_{C^\perp}}$*, where* $\mathbf{x_C} = \{x_i | i \in \mathbf{T_C}\}$*, and* $\mathbf{x_{C^\perp}} = \{x_i | i \in \mathbf{T_{C^\perp}}\}$*.*

Definition 11 $\kappa_{\mathbf{p}}$ *is the set of all* $s(\mathbf{x})$ *of the form* $s(\mathbf{x}) = (-1)^{p(\mathbf{x})}$*, where* $p(\mathbf{x}) = \sum_k q_k(\mathbf{x_C}) r_k(\mathbf{x_{C^\perp}})$*, where* $\deg(q_k(\mathbf{x_C})) = 1$ $\forall k$*, and where* $x_i \in p(\mathbf{x})$*,* $\forall$ $i \in \{0, 1, \ldots, n-1\}$*. We refer to* $\kappa_{\mathbf{p}}$ *as the set of 'half-linear bipartite bipolar' states.* $\ell_{\mathbf{p}}$ *is the subset of* $\kappa_{\mathbf{p}}$ *where* $\deg(r_k(\mathbf{x_C})) = 1$ $\forall k$*.*

Theorem 4 *[18] Let* $m(\mathbf{x})$ *be a binary ANF. If* $s(\mathbf{x}) \in \kappa_{\mathbf{p}}$*, then the action of* $\prod_{i \in \mathbf{T_C}} H(i)$ *on* $s(\mathbf{x})$ *gives* $s'(\mathbf{x}) = m(\mathbf{x})$*. If* $s(\mathbf{x}) \in \ell_{\mathbf{p}}$*, then the action of* $\prod_{i \in \mathbf{T_{C^\perp}}} H(i)$ *on* $s(\mathbf{x})$ *gives* $s''(\mathbf{x}) = m(\mathbf{x})$*.* $s'(\mathbf{x})$ *(*$s''(\mathbf{x})$*) is the binary indicator for a binary linear or nonlinear* $[n, n - |\mathbf{T}|, d]$ *error correcting code,* $\mathbf{C}$*.*

Theorem 4 is particularly relevant when $p(\boldsymbol{x})$ is constructed using (1), as the 'strongest' members of $\kappa_{\mathbf{p}}$ are generated as a subclass of the construction if $\deg(g_j) < 2$, $\forall j$. (By considering matrices other than $\boldsymbol{H}$ it is conjectured that it is always possible to convert a bipolar sequence, $\boldsymbol{s} = (-1)^{\boldsymbol{p}}$, constructed using (1) to a binary form, even when $\deg(g_j) \geq 2$). If $\boldsymbol{s}$ can be transformed to a binary linear indicator, $\boldsymbol{s'}$, using only tensor products of $\boldsymbol{H}$ and $\boldsymbol{I}$, then we say that $\boldsymbol{s}$ is 'HI-equivalent to' $\boldsymbol{s'}$.

Theorem 5 *[18] The set* $\ell_{\mathbf{p}}$ *is HI-equivalent to the set of* $[n, k, d]$ *binary linear codes.*

3.1 Examples

Example A Let $t = 2$, $L = 3$. Then (1) can generate,

$$p(\boldsymbol{x}) = x_0x_2 + x_1x_3 + x_2x_4 + x_3x_5 + x_2x_5$$

Let $\mathbf{T_C} = \{0, 1, 4, 5\}$ and $\mathbf{T_{C^\perp}} = \{2, 3\}$. Applying $H(0)H(1)H(4)H(5)$ (in any order) to $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$ gives the binary sequence, $\boldsymbol{s'} = m(x) = (x_0 + x_2 + 1)(x_1 +$

$x_3+1)(x_2+x_4+1)(x_2+x_3+x_5+1)$, which is the indicator for a $[6,2,2]$ binary linear code, $\mathbf{C}$. Graphical representations for $\boldsymbol{s}$ and $\boldsymbol{s'}$ are shown in Fig 1, where the graph for $\boldsymbol{s'}$ is a Normal Realisation of a Factor Graph [7]. If, instead, we apply $H(2)H(3)$ (in any order) to $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$, we get the binary sequence, $\boldsymbol{s''} = m(x) = (x_0+x_2+x_4+x_5+1)(x_1+x_3+x_5+1)$, which is the indicator for a $[6,4,2]$ binary linear code, $\mathbf{C}^{\perp}$, the dual of $\mathbf{C}$. Applying $H(0)H(1)H(4)H(5)$ to $\boldsymbol{s'}$, followed by $H(2)H(3)$, gives $\boldsymbol{s''}$. This is the same as applying the WHT to $\boldsymbol{s'}$, and it is known that binary indicators of a linear code code, $\mathbf{C}$, and its dual, $\mathbf{C}^{\perp}$, are related by the WHT [14].

Example B Let $t = 3$, $L = 3$. Then (1) can generate,

$$p(\boldsymbol{x}) = 034, 035, 045, 134, 135, 145, 234, 235, 245, 03, 05, 14, 15, 36, 47, 58$$

Let $\mathbf{T_C} = \{0,1,2,6,7,8\}$ and $\mathbf{T}_{\mathbf{C}^{\perp}} = \{3,4,5\}$. Applying $H(0), H(1), H(2), H(6), H(7), H(8)$ (in any order) to $\boldsymbol{s} = (-1)^{p(\boldsymbol{x})}$ gives,

$$\begin{aligned}&\boldsymbol{s'} = m(x) = \\ &(x_0 + x_3x_4 + x_3x_5 + x_4x_5 + x_3 + x_5 + 1)(x_1 + x_3x_4 + x_3x_5 + x_4x_5 + x_4 + x_5 + 1) \\ &\times(x_2 + x_3x_4 + x_3x_5 + x_4x_5 + 1)(x_3 + x_6 + 1)(x_4 + x_7 + 1)(x_5 + x_7 + 1)\end{aligned}$$

which is the indicator for a $[9,3,3]$ binary nonlinear code, $\mathbf{C}$. Graphical representations for $\boldsymbol{s}$ and $\boldsymbol{s'}$ are shown in Fig 1, where the graph for $\boldsymbol{s'}$ is a Normal Realisation of a **nonlinear** Factor Graph. In this case application of $H(3)H(4)H(5)$ does not produce the dual code, $\mathbf{C}^{\perp}$, but the nonlinear dual could be obtained by nonlocal transform over x_3, x_4, x_5.

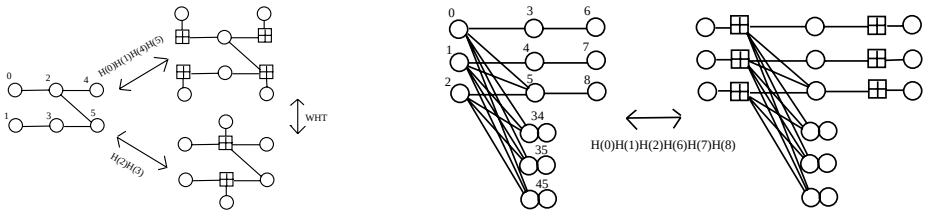


Fig. 1. Bipolar ↔ Factor Graph HI-Equivalence for Examples A and B

Example C The nonlinear $[16,8,6]$ Nordstrom-Robinson binary code is HI-equivalent to a half-linear bipolar bipartite sequence, $(-1)^{p(\boldsymbol{x})}$, where $p(\boldsymbol{x})$ can be constructed using (1), and has ANF comprising 96 cubic and 40 quadratic terms, and where $|T_C| = |T_{C^{\perp}}| = 8$. The quadratic part of $p(\boldsymbol{x})$ is HI-equivalent to a binary linear $[16,8,4]$ code, so we can view the 96 cubic terms of $p(\boldsymbol{x})$ as further 'doping' to increase Hamming Distance, d, from 4 to 6.

3.2 Comments

This section has identified an important subset of $\kappa_{\mathbf{p}}$ as a subset of the construction of (1), where a member of $\kappa_{\mathbf{p}}$ can be transformed to a binary sequence

under selective action of $\boldsymbol{H}$. Conversely, this gives us a way of analysing a Factor Graph, by transforming it back into bipolar sequence form. A natural question to ask is which length 2^n bipolar sequences are transform-equivalent to the best $[n, k, d]$ linear and nonlinear codes? We offer offer the following conjecture,

Conjecture 1 *Optimal linear or nonlinear codes can be constructed from (1) if $L = 2$, and $(-1)^{g_j}$ is, itself, HI-equivalent to an optimal linear or nonlinear code, $\forall j$. But what f_{γ_j} should be chosen?*

In the next section we pose the related question: Which quantum n-qubit states have optimal Linear Entanglement?

4 PAR_l and Quantum 'Linear' Entanglement (LE)

Joint work with V.Rijmen [18]

In previous sections our PAR metric has been measured relative to all LUUTs. Quantum systems require that we compute our PAR metric (now called PAR_l) relative to all LUTs, of which LUUTs are a subset. It is argued in [18] that PAR_l and Linear Entanglement (LE) are good partial measures of quantum entanglement. [1] Let $\boldsymbol{s}$ be a length 2^n bipolar sequence. In the context of quantum systems we interpret (after appropriate normalisation) this sequence as a probability density function of an n-qubit quantum state. Let s_i be an element of $\boldsymbol{s}$. Then $|s_i|^2$ is the probability of measuring the quantum system in state i. We must normalise so that $\sum_{i=0}^{2^n-1} |s_i|^2 = 1$, although normalisation constants are usually omitted in this paper. An n-qubit state, $\boldsymbol{s}$, contains entanglement if $\boldsymbol{s}$ is not a member of $\boldsymbol{G_n}$. The definition of PAR_l is then identical to Definition 5 except that, now, $|l_i|$ does not have to equal $|l_j|$, i.e. $\boldsymbol{l}$ is not necessarily unimodular.

Definition 12 $PAR_l(\boldsymbol{s}) = 2^n max_{\boldsymbol{l}}(|\boldsymbol{s} \cdot \boldsymbol{l}|^2))$

where $\boldsymbol{l}$ is any normalised linear sequence from the set, $\boldsymbol{G_n}$, and $\cdot$ means 'inner product' [17, 18].

Linear Entanglement (LE) is then defined as,

Definition 13 $LE(\boldsymbol{s}) = n - \log_2(PAR_l(\boldsymbol{s}))$

Entanglement and LE are invariant under transformation of $\boldsymbol{s}$ by any LUT. Therefore PAR_l is Local Unitary (LU)-invariant, and two states, $\boldsymbol{s}$ and $\boldsymbol{s'}$, related by a transform from LUT, are LU-equivalent. Code duality under the WHT and the HI-equivalence between $\boldsymbol{s}$ and $\boldsymbol{s'}$, as discussed in Section 3, are special cases of LU-equivalence. One can also view entanglement invariance as a generalisation of code duality.

[1] Quantum information theorists often consider 'mixed-state' entanglement, where entanglement with the environment is unavoidable [24, 8]. This is similar to the analysis of classical communications codes in the context of a corrupting channel. In this paper we only consider a closed (pure) quantum system with no environmental entanglements [6].

4.1 PAR_l for States from $\ell_{\mathbf{p}}$

Theorem 6 *[18] If $\boldsymbol{s} \in \ell_{\mathbf{p}}$, then $\boldsymbol{s}$ is LU equivalent to the indicator for an $[n, k, d]$ binary linear code, and,*

$$PAR_l(\boldsymbol{s}) \geq 2^r, \quad \text{where } r = max(k, n-k)$$

Theorem 6 implies that states, $\boldsymbol{s}$, from $\ell_{\mathbf{p}}$ have a minimum lower bound on PAR_l (upper bound on LE) when the associated $[n, k, d]$ code, $\mathbf{C}$, satisfies $k = \lfloor \frac{n}{2} \rfloor$, with $\mathrm{PAR}_l \geq 2^{\lceil \frac{n}{2} \rceil}$. Here is a stronger result.

Theorem 7 *[18] In (1), let $t = 1$ and $f_{\gamma j}$ be the identity permutation $\forall j$. Using (1), we can generate $s(\mathbf{x}) = (-1)^{p(\boldsymbol{x})}$ for $p(\boldsymbol{x})$ constructed using (4). Then $PAR_l(\boldsymbol{s}) = 2^{\lceil \frac{n}{2} \rceil}$.*

Definition 14 $\quad PA(\boldsymbol{s}) = 2^n max_i(|s_i|^2)$

We now compute PA for any HI transform of a member of $\ell_{\mathbf{p}}$. Let $\mathbf{s} \in \ell_{\mathbf{p}}$. Recalling Definition 10, let $k = |\mathbf{T}_{\mathbf{C}^\perp}|$, $k^\perp = |\mathbf{T}_{\mathbf{C}}|$, and $k + k^\perp = n$. Without loss of generality we renumber integer sets $\mathbf{T}_{\mathbf{C}^\perp}$ and $\mathbf{T}_{\mathbf{C}}$ so that $\mathbf{T}_{\mathbf{C}^\perp} = \{0, 1, \ldots, k-1\}$ and $\mathbf{T}_{\mathbf{C}} = \{k, k+1, \ldots, n-1\}$. Let $\mathbf{t}_{\mathbf{C}^\perp} \subset \mathbf{T}_{\mathbf{C}^\perp}$ and $\mathbf{t}_{\mathbf{C}} \subset \mathbf{T}_{\mathbf{C}}$, where $h = |\mathbf{t}_{\mathbf{C}^\perp}|$ and $h^\perp = |\mathbf{t}_{\mathbf{C}}|$. Let $\mathbf{x}_{\mathbf{t}^\perp} = \{x_i | i \in \mathbf{t}_{\mathbf{C}^\perp}\}$, $\mathbf{x}_{\mathbf{t}} = \{x_i | i \in \mathbf{t}_{\mathbf{C}}\}$, and $\mathbf{x}_* = \mathbf{x}_{\mathbf{t}^\perp} \cup \mathbf{x}_{\mathbf{t}}$. Define $\mathbf{M}$ to be a $k \times k^\perp$ binary matrix where $M_{i,j-k} = 1$ iff $x_i x_j \in p(\mathbf{x})$, and $M_{i,j-k} = 0$ otherwise. Thus $p(\mathbf{x}) = \sum_{i \in \mathbf{T}_{\mathbf{C}^\perp}} x_i (\sum_{j \in \mathbf{T}_{\mathbf{C}}} M_{i,j-k} x_j)$. Let $\mathbf{M}_{\mathbf{t}}$ be a submatrix of $\mathbf{M}$, which comprises only the rows and columns of $\mathbf{M}$ specified by $\mathbf{t}_{\mathbf{C}^\perp}$ and $\mathbf{t}_{\mathbf{C}}$. Let χ_t be the rank of $\mathbf{M}_{\mathbf{t}}$.

Theorem 8 *[18] Let $\boldsymbol{s'}$ be the result of $\prod_{i \in \mathbf{t}_{\mathbf{C}^\perp} \cup \mathbf{t}_{\mathbf{C}}} H(i)$ on $\boldsymbol{s} \in \ell_{\mathbf{p}}$. Then,*

$$PA(\boldsymbol{s'}) = 2^{h + h^\perp - 2\chi_t}$$

Corollary 1 *As $0 \leq \chi_t \leq min(h, h^\perp)$, it follows that, for $\boldsymbol{s} \in \ell_{\mathbf{p}}$, $PA(\boldsymbol{s'}) \geq 2^{|h - h^\perp|}$*

In general, PAR_l must consider $\mathrm{PA}(\boldsymbol{s})$ under all LUTs. $\mathrm{PA}(\boldsymbol{s})$ for $\boldsymbol{s} \in \ell_{\mathbf{p}}$ is easily computed. Let the 'HI multispectra' be the union of the power spectra of $\boldsymbol{s}$ under the action of $\prod_{i \in \boldsymbol{T}} H(i)$, for all possible subsets, $\boldsymbol{T}$, of $\{0, 1, \ldots, n-1\}$.

Theorem 9 *[18] PAR_l of $\boldsymbol{s} \in \ell_{\mathbf{p}}$ is found in the HI multispectra of $\boldsymbol{s}$.*

Theorem 9 means that, for $\boldsymbol{s} \in \boldsymbol{\ell_p}$, we only need compute the 2^n HI transforms to compute PAR_l. If $\mathrm{PA}(\boldsymbol{s})$ is optimally low over the HI multispectra, then $\boldsymbol{s'} = m(\boldsymbol{x})$ is an optimal binary linear code when $\boldsymbol{T} = \boldsymbol{T_C}$ or $\boldsymbol{T} = \boldsymbol{T_{C^\perp}}$.

Definition 15 *The Weight Hierarchy of a linear code* $\mathbf{C}$*, is a series of parameters,* d_j*,* $0 \leq j \leq k$*, representing the smallest blocklength of a linear sub-code of* $\mathbf{C}$ *of dimension* j*, where* $d_k = n$*,* $d_1 = d$*, and* $d_0 = 0$*.*

Theorem 10 *[18] Let* $\boldsymbol{s_c}$ *be the indicator of an* $[n, k, d]$ *binary linear code,* $\mathbf{C}$*. Let* $\mathbf{Q} \subset \{0, 1, \ldots, n-1\}$*. Let,*

$$m_{\mathbf{Q}} = \frac{|\mathbf{Q}| + \log_2(\mu) - n + k}{2}, \quad \text{where } \mu = PA(\boldsymbol{s'_c}) \tag{7}$$

and $\boldsymbol{s'_c} = \prod_{t \in \mathbf{Q}} H(t)[\boldsymbol{s_c}]$*. Then the Weight Hierarchy of* $\mathbf{C}$ *is found from the HI multispectra of* $\boldsymbol{s_c}$*, where* $d_j = min_{\mathbf{Q}|m_Q=j}(|\mathbf{Q}|)$

Quantum measurement projects a system to a subsystem. This allows us to equate a series of quantum measurements with a series of subcodes of $\boldsymbol{C}$. Let the entanglement order of a system be the size (in qubits) of the largest entangled subsystem of the system. A most-destructive series of j single-qubit measurements over some set of possible measurements on $\boldsymbol{s}$ produces a final state $\boldsymbol{s'}$ such that entanglement order($\boldsymbol{s}$) − entanglement order($\boldsymbol{s'}$) is maximised.

Definition 16 *Stubborness of Entanglement (SE) is a series of parameters,* β_j*,* $0 \leq j \leq k'$*, representing smallest possible entanglement order,* β_j*, after* $k' - j$ *most-destructive measurements of an* n*-qubit system, where* $\beta_{k'} = n$*,* $\beta_0 = 0$*.*

Theorem 11 *[18] Let* $\boldsymbol{s} \in \ell_{\mathbf{p}}$ *where* $\boldsymbol{s}$ *is LU equivalent to an optimal or near-optimal binary linear code of dimension* $\leq \frac{n}{2}$*. Then Stubborness of Entanglement is equal to the Weight Hierarchy of the code.*

Corollary 2 *Quantum states from* $\ell_{\mathbf{p}}$ *which have optimum LE and optimum SE are LU-equivalent to binary linear codes with optimum Weight Hierarchy.*

The results of this section suggests the following modification of Conjecture 1.

Conjecture 2 *States with optimal LE can be constructed from (1) if* $L = 2$*, and* $(-1)^{g_j}$ *also has optimal LE,* $\forall j$*. But what* f_{γ_j} *should be chosen?*

5 Discussion and Open Problems

We have highlighted the importance PAR plays (explicitly or implicitly) in current research. We emphasis four areas:
a) Low PAR error-correcting codes for OFDM and CDMA.
b) Highly nonlinear, distinguishable sequence sets for cryptography.
c) Graphical construction primitives for Factor Graphs which represent good error-correcting codes.
d) Classification and quantification of quantum entanglement.
We finish with a list of a few open problems.

- Construction (1) only provides an exact, implementable encoder if the two following sub-problems can be solved:

- Provide algorithms to generate all permutation polynomials, f_γ, of degree $\mu - 1$. $\mu = 0$ is trivial. Section 2.4 provides an answer for $\mu = 1$. But, for $\mu > 1$ the situation is unclear.
- Given an algorithm to generate all permutation polynomials, then construction (1) only generates distinct $p(\boldsymbol{x})$ for $t = 1$. For $t > 1$, the permutation, π, induces extra symmetries which cause many $p(\boldsymbol{x})$ to be generated more than once. This situation is reflected in (2), which is a strict upper bound for $t > 1$. It remains an open problem to provide an algorithm for $t > 1$ which ensures the generated $p(\boldsymbol{x})$ are distinct and form the whole code. Such an algorithm would replace of (2) with an exact expression.

– Construct decoders for the above codes.
– It is considered that successful iteration on a Factor Graph requires few short graph cycles. This is ensured if the graph has a large girth. How does one construct Factor Graphs with low PAR_l and large girth?
– Provide a construction for optimally large sets, $\boldsymbol{P}$, of pure quantum states such that each state satisfies a low upper bound on PAR_l, and where any two members of $\boldsymbol{P}$ are optimally distinguishable. This problem is 'simply' the LUT extension of the problem of low PAR error-correcting codes for OFDM and cryptography.

References

1. Alperin, J.L.,Bell, R.B.: **Groups and Representations,** Graduate Texts in Mathematics, Springer, **162**, pp 39–48, (1995)
2. Brundan, J.: Web Lecture Notes: Math 607, Polynomial representations of GL_n, *http://darkwing.uoregon.edu/∼brundan/teaching.html* pp 29–31, Spring (1999)
3. Canteaut, A.,Carlet, C.,Charpin, P.,Fontaine, C.: Propagation Characteristics and Correlation-Immunity of Highly Nonlinear Boolean Functions. EUROCRYPT 2000, Lecture Notes in Comp. Sci., **1807**, 507–522, (2000)
4. Collins, D.,Popescu, S.: A Classical Analogue of Entanglement *http://xxx.soton.ac.uk/ps/quant-ph/0107082* 16 Jul. 2001
5. Davis, J.A.,Jedwab, J.: Peak-to-mean Power Control in OFDM, Golay Complementary Sequences and Reed-Muller Codes. IEEE Trans. Inform. Theory **45.** No 7, 2397–2417, Nov (1999)
6. Eisert, J.,Briegel, H.J.: Quantification of Multi-Particle Entanglement. *http://xxx.soton.ac.uk/ps/quant-ph/0007081 v2* 29 Aug (2000)
7. Forney, G.D.: Codes on Graphs: Normal Realizations. IEEE Trans. Inform. Theory **47.** No 2, 520–548, Feb, (2001)
8. Fuchs, C.A.,van de Graaf, J.: Cryptographic Distinguishability Measures for Quantum-Mechanical States. IEEE Trans. Inform. Theory **45.** No 4, 1216–1227, May (1999)
9. Golay, M.J.E.: Complementary Series. IRE Trans. Inform. Theory, **IT-7**, pp 82–87, Apr (1961)
10. Harrison, M.A.: The Number of Classes of Invertible Boolean Functions. J. ACM, **10**, 25–28, (1963)

11. Jones, A.E.,Wilkinson, T.A.,Barton, S.K.: Block Coding Scheme for Reduction of Peak to Mean Envelope Power Ratio of Multicarrier Transmission Schemes. Elec. Lett. **30**, 2098–2099, (1994)
12. Kschischang, F.R.,Frey, B.J.,Loeliger, H-A.: Factor Graphs and the Sum-Product Algorithm. IEEE Trans. Inform. Theory **47.** No 1, Jan, (2001)
13. Lidl, L.,Niederreiter, H.: **Introduction to Finite Fields and their Applications** Cambridge Univ Press, pp 361–362, (1986)
14. MacWilliams, F.J.,Sloane, N.J.A.: **The Theory of Error-Correcting Codes** Amsterdam: North-Holland. (1977)
15. Parker, M.G.: Quantum Factor Graphs. Annals of Telecom., July-Aug, pp 472-483, (2001), originally 2nd Int. Symp. on Turbo Codes and Related Topics, Brest, France Sept 4–7, (2000), *http://xxx.soton.ac.uk/ps/quant-ph/0010043*, (2000) *http://www.ii.uib.no/∼matthew/mattweb.html*
16. Parker, M.G.,Tellambura, C.: Generalised Rudin-Shapiro Constructions. *WCC2001, Workshop on Coding and Cryptography, Paris(France)*, Jan 8-12, (2001) *http://www.ii.uib.no/∼matthew/mattweb.html*
17. Parker, M.G.,Tellambura, C.: Golay-Davis-Jedwab Complementary Sequences and Rudin-Shapiro Constructions. Submitted to IEEE Trans. Inform. Theory, *http://www.ii.uib.no/∼matthew/mattweb.html* March (2001)
18. Parker, M.G., Rijmen, V.: The Quantum Entanglement of Binary and Bipolar Sequences. Short version accepted for Discrete Mathematics, Long version at *http://xxx.soton.ac.uk/ps/quant-ph/0107106* or *http://www.ii.uib.no/∼matthew/mattweb.html* Jun. (2001)
19. Parker, M.G.,Tellambura, C.: A Construction for Binary Sequence Sets with Low Peak-to-Average Power Ratio. *Submitted to Int. Symp. Inform. Theory, Laussane, Switzerland, (2002)*, *http://www.ii.uib.no/∼matthew/mattweb.html* October (2001)
20. Inequivalent Invertible Boolean Functions for $t = 3$, *http://www.ii.uib.no/∼matthew/mattweb.html*, (2001)
21. Paterson, K.G.: Generalized Reed-Muller Codes and Power Control in OFDM Modulation. IEEE Trans. Inform. Theory, **46**, No 1, pp. 104-120, Jan. (2000)
22. Paterson, K.G.,Tarokh V.: On the Existence and Construction of Good Codes with Low Peak-to-Average Power Ratios. IEEE Trans. Inform. Theory **46.** No 6, 1974–1987, Sept (2000)
23. Paterson, K.G., Sequences for OFDM and Multi-Code CDMA: Two Problems in Algebriac Coding Theory. Hewlett-Packard Technical Report, HPL-2001-146, (2001)
24. Popescu, S.,Rohrlich, D.: On the Measure of Entanglement for Pure States. Phys. Rev. A **56.** R3319, (1997)
25. Rudin, W.,: Some Theorems on Fourier Coefficients. Proc. Amer. Math. Soc., No 10, pp. 855–859, (1959)
26. Shapiro, H.S.: Extremal Problems for Polynomials. M.S. Thesis, M.I.T., (1951)
27. Shepherd, S.J.,Orriss, J.,Barton, S.K.: Asymptotic Limits in Peak Envelope Power Reduction by Redundant Coding in QPSK Multi-Carrier Modulation. IEEE Trans. Comm., **46**, No 1, 5–10, Jan (1998)
28. Sloane, N.J.A.: The On-Line Encyclopedia of Integer Sequences. $(1, 2, 154, \ldots)$, *http://www.research.att.com/∼njas/sequences/index.html*

Increasing sample sizes do not always increase the power of UMPU-tests for 2×2 tables [1]

By H. Finner and K. Strassburger

Deutsches Diabetes-Forschungsinstitut an der
Heinrich-Heine-Universität Düsseldorf
Abteilung Biometrie und Epidemiologie,
Düsseldorf, Germany

May 2, 2000

Abstract. We consider the uniformly most powerful unbiased (UMPU) one-sided test for the comparison of two proportions based on sample sizes m and n, i.e., the randomized version of Fisher's exact one-sided test. It will be shown that the power function of the one-side UMPU-test based on sample sizes m and n can coincide on the entire parameter space with the power function of the UMPU test based on sample sizes $m+1$ and n for certain levels. A characterization of all such cases with identical power functions is derived. Finally, this characterization is closely related to number theoretical problems concerning Fermat-like binomial equations. Some consequences for Fisher's original exact test will be discussed, too.

1. Introduction

One of the oldest and apparent most basic problems in statistics is the evaluation of a 2×2-table. Originally, our aim was to develope certain optimal multiple decision procedures as for example optimal selection and partitioning procedures in k-sample situations with underlying binomial distributions. We expected that everything should be clarified concerning the evaluation of 2×2-tables. After inspection of uncountable papers on 2×2-tables we learnt that there are more unsolved than solved problems and that some of the problems we had are not even mentioned in the literature. Some of these issues concerning structural properties of Fisher's exact test (Fisher (1934/35), Yates (1934), Irwin (1935)) as well as the corresponding UMPU-tests for 2×2-tables (Tocher (1950)) are discussed and partially solved in Finner & Strassburger (2000). Some of the results for the one-sided UMPU-test derived there will be applied here, too. In the present paper we investigate a surprising phenomenon arising in connection with power considerations for one-sided UMPU-tests for comparing two proportions. It will be shown that increasing sample sizes do not always increase the (unconditional) power of the one-sided UMPU-test.

To set up notation, suppose we have two sets of Bernoulli random variables $X_i,\ i = 1, \ldots, m$ and $Y_i,\ i = 1, \ldots, n$, respectively, with success probabilities p and q, respectively, where $X_1, \ldots, X_m$, $Y_1, \ldots, Y_n$ are independently distributed, m, n are fixed and known and p, q are unknown. We are faced with testing the one-sided hypothesis $H : p \leq q$ versus the one-sided alternative $K : p > q$.

[1] MSC 1991 classification numbers. Primary 62F03, 62C99. Secondary 11D41.
Key words and phrases. diophantine equation, Fermat-like binomial equation, Fisher's exact test, one-sided hypothesis, 2×2 table, uniformly most powerful unbiased test.

Then $X = \sum_{i=1}^{m} X_i$ and $Y = \sum_{i=1}^{n} Y_i$ are independent random variables having a binomial distribution with parameters $m \in I\!N,\ p \in [0,1]$ and $n \in I\!N,\ q \in [0,1]$, respectively. The UMPU-test at level $\alpha \in (0,1)$ for testing H versus K is based on conditioning on $S = X + Y = s$ and is given by

$$\psi(x|s,m,n,\alpha) = \begin{cases} 0, & x < c_{s,m,n,\alpha}, \\ \gamma_{s,m,n,\alpha}, & x = c_{s,m,n,\alpha}, \\ 1, & x > c_{s,m,n,\alpha}, \end{cases} \tag{1.1}$$

where $c = c_{s,m,n,\alpha} \in \{0,\dots,m\}$ and $\gamma = \gamma_{s,m,n,\alpha} \in [0,1)$ are determined such that

$$F(c|s,m,n) - \gamma f(c|s,m,n) = 1 - \alpha.$$

Here

$$f(x|s,m,n) = \begin{cases} \binom{m}{x}\binom{n}{s-x}/\binom{m+n}{s}, & \text{for } \max(0,s-n) \le x \le \min(s,m),\ s,x \in I\!N_0, \\ 0, & \text{otherwise}, \end{cases} \quad m,\ n \in I\!N,$$

denotes the probability mass function (pmf) of the hypergeometric distribution with parameters s,m,n and F denotes the corresponding cumulative mass function (cmf). Let $g(x|m,p) = \binom{m}{x}p^x(1-p)^{m-x}$, $x = 0,\dots,m$, denote the pmf of the binomial distribution with parameters p and m. Setting

$$\beta_1(p|y,m,n,\alpha) = \sum_x \psi(x|x+y,m,n,\alpha)g(x|m,p),$$

the (unconditional) two-dimensional power function of ψ can be calculated by

$$\beta(p,q|m,n,\alpha) = \sum_y \beta_1(p|y,m,n,\alpha)g(y|n,q),\ p,q \in [0,1]. \tag{1.2}$$

It is well known, that the power of the UMPU-test is non-decreasing on the alternative K in the sample sizes m and n, respectively. This is based on the fact, that the UMPU-test, given m and n, is based on the sufficient statistic $(\sum_{i=1}^{m} X_i\ ,\ \sum_{i=1}^{n} Y_i)$. One might expect, that the power function is not only non-decreasing in m and n on K but even strictly increasing for at least some (if not all) parameter points in K. It will be shown, that this expectation turns out to fail, that is, there exist $\alpha \in (0,1)$ and m,n, such that

$$\beta(p,q|m,n,\alpha) = \beta(p,q|m+1,n,\alpha) \qquad \forall p,q \in [0,1]. \tag{1.3}$$

A full characterization of situations where (1.3) is satisfied is given in Section 2. The technical and rather lengthy part of the proof of our main result is deferred to the Appendix. Consequences for the classical exact test of Fisher are discussed in Section 3. Finally, in Section 4 we show that the whole problem results in interesting for the most part unsolved number theoretical problems concerning Fermat-like binomial equations.

2. The main result

Noting that the class of binomial distributions with pmf $g(\cdot|m,p),\ p \in [0,1]$, is complete, we get that (1.3) is equivalent to

$$\beta_1(p|y,m,n,\alpha) = \beta_1(p|y,m+1,n,\alpha) \qquad \forall p \in [0,1],\ y = 0,\dots,n. \tag{2.1}$$

Since $\beta_1(p|y,m,n,\alpha) - \beta_1(p|y,m+1,n,\alpha)$ is a continuous function in p and $\beta(p,p|m,n,\alpha) = \beta(p,p|m+1,n,\alpha)$ for all $p \in [0,1]$, (1.2) yields that (2.1) is equivalent to

$$\beta_1(p|y,m,n,\alpha) = \beta_1(p|y,m+1,n,\alpha) \qquad \forall p \in (0,1),\ y = 0,\ldots,n-1. \tag{2.2}$$

To further characterize the situations where (1.3) holds, we make use of the following Lemma the proof of which is given in the Appendix.

Lemma 1. Let $\alpha \in (0,1)$.

a) $\beta_1(p|0,m,n,\alpha) = \beta_1(p|0,m+1,n,\alpha)$ holds for all $p \in (0,1)$ if and only if there exits an $u \in \{0,\ldots,m-1\}$ such that $F(u|u+1,m,n) = 1-\alpha$.

b) Let $y \in \{1,\ldots,n-1\}$ and suppose there exists an integer $u \in \{0,\ldots,m-1\}$ with $F(u|u+y,m,n) = 1-\alpha$. Then $\beta_1(p|y,m,n,\alpha) = \beta_1(p|y,m+1,n,\alpha)$ holds for all $p \in (0,1)$ if and only if there exits a $v \in \{u+1,\ldots,m-1\}$ such that $F(v|v+y+1,m,n) = 1-\alpha$.

Lemma 1 immediately yields the following characterization of (1.3).

Theorem 2. For $\alpha \in (0,1)$ equation (1.3) holds, if and only if there exists a sequence of integers $0 \le u_1 < \ldots < u_n < m$ such that

$$F(u_i|u_i+i,m,n) = 1-\alpha \ \text{ for all } i = 1,\ldots,n. \tag{2.3}$$

It remains the question whether there exist m,n and a sequence of integers $0 \le u_1 < \ldots < u_n < m$ such that (2.3) is satisfied. In case of $n \in \{1,2\}$ and in case of $m = n,\ n \in I\!N$, we obtain the following answer.

Theorem 3. Let $\alpha \in (0,1)$, $m \in I\!N$.
(a) For $n = 1$, (1.3) holds if and only if $\alpha = \ell/(m+1)$, $\ell \in \{1,\ldots,m\}$.
(b) For $n = 2$, (1.3) holds if and only if there exist $u,v \in \{0,\ldots,m-1\}$ with $\alpha = f(u+1|u+1,m,2) = 1 - f(v|v+2,m,2)$. or, equivalently,

$$m = \frac{(v+1)(v+2)}{2(u+1)} + \frac{u}{2} - 1 \quad \text{and} \quad \alpha = 1 - \frac{(v+1)(v+2)}{(m+1)(m+2)}. \tag{2.4}$$

(c) For $m = n$, (1.3) holds if $\alpha = 1/2$.

Proof. Parts (a) and (b) follow immediately from Theorem 2. In case of $m = n$ and $\alpha = 1/2$ the UMPU-test is given by

$$\psi(x|x+y,m,m,1/2) = \begin{cases} 0, & x < y \\ 1/2, & x = y \\ 1, & x > y \end{cases}. \tag{2.5}$$

This can easily be seen by noting the symmetry of $f(\cdot|s,m,m)$. Hence we also get part (c) from Theorem 2. □

Remark. (a) The set of all solutions (m,u,v) of (2.4) is infinite, since for all $u \in I\!N \cup \{0\}$ the triples $(5u+5, u, 3u+2)$ and $(5u+7, 2u+2, 4u+5)$ belong to this set.

(b) Setting $a = m - u - 1$, $b = v$ and $c = m$, it can easily be seen that the equation $f(u+1|u+1, m, 2) = 1 - f(v|v+2, m, 2)$ is equivalent to

$$(a+1)(a+2) + (b+1)(b+2) = (c+1)(c+2) \quad \text{or} \quad \binom{a+2}{2} + \binom{b+2}{2} = \binom{c+2}{2}.$$

The set of all positive integer solutions for these equations can be found e. g. in Harborth (1988), related references are Khatri (1955), Sierpiǹski (1962) and Fraenkel (1971). At this place we refer to Section 4 of this paper for a more detailed discussion concderning the relationship to number theoretic problems.

We display two examples where the power functions of two different tests coincide. The UMPU tests at level $\alpha = 1/2$ for $(m, n) = (5, 5)$ and $(m, n) = (6, 5)$ are given by

$\psi(x|x+y, 5, 5, 1/2)$

x \ y	0	1	2	3	4	5
5	1	1	1	1	1	$\frac{1}{2}$
4	1	1	1	1	$\frac{1}{2}$	0
3	1	1	1	$\frac{1}{2}$	0	0
2	1	1	$\frac{1}{2}$	0	0	0
1	1	$\frac{1}{2}$	0	0	0	0
0	$\frac{1}{2}$	0	0	0	0	0

$\psi(x|x+y, 6, 5, 1/2)$

x \ y	0	1	2	3	4	5
6	1	1	1	1	1	$\frac{1}{2}$
5	1	1	1	1	$\frac{7}{12}$	$\frac{1}{12}$
4	1	1	1	$\frac{2}{3}$	$\frac{1}{6}$	0
3	1	1	$\frac{3}{4}$	$\frac{1}{4}$	0	0
2	1	$\frac{5}{6}$	$\frac{1}{3}$	0	0	0
1	$\frac{11}{12}$	$\frac{5}{12}$	0	0	0	0
0	$\frac{1}{2}$	0	0	0	0	0

The UMPU tests at level $\alpha = 2/7$ for $(m, n) = (5, 2)$ and $(m, n) = (6, 2)$ are given by

$\psi(x|x+y, 5, 2, 2/7)$

x \ y	0	1	2
5	1	1	$\frac{2}{7}$
4	1	$\frac{1}{2}$	0
3	1	$\frac{1}{4}$	0
2	$\frac{3}{5}$	0	0
1	$\frac{2}{5}$	0	0
0	$\frac{2}{7}$	0	0

$\psi(x|x+y, 6, 2, 2/7)$

x \ y	0	1	2
6	1	1	$\frac{2}{7}$
5	1	$\frac{7}{12}$	$\frac{1}{21}$
4	1	$\frac{1}{3}$	0
3	$\frac{3}{4}$	$\frac{1}{8}$	0
2	$\frac{8}{15}$	0	0
1	$\frac{8}{21}$	0	0
0	$\frac{2}{7}$	0	0

All cases for $3 \le m \le 62$, where (2.4) is fulfilled with corresponding values $\alpha = 1 - \frac{(v+1)(v+2)}{(m+1)(m+2)} = \frac{(m-u)(m-u+1)}{(m+1)(m+2)}$ with $\alpha \le 1/2$ are given in Table 3.

m	α	m	α	m	α	m	α
2	$\frac{1}{2}$	20	$\frac{1}{11}$, $\frac{26}{77}$	39	$\frac{19}{82}$	52	$\frac{35}{477}$, $\frac{176}{477}$
5	$\frac{2}{7}$	22	$\frac{15}{92}$, $\frac{11}{46}$, $\frac{35}{92}$	40	$\frac{40}{287}$, $\frac{11}{41}$, $\frac{100}{287}$	53	$\frac{14}{99}$
7	$\frac{5}{12}$	25	$\frac{40}{117}$	42	$\frac{351}{946}$	54	$\frac{1}{28}$
9	$\frac{2}{11}$	26	$\frac{13}{63}$	43	$\frac{7}{33}$	55	$\frac{187}{532}$
10	$\frac{7}{22}$	27	$\frac{2}{29}$, $\frac{55}{406}$, $\frac{153}{406}$	44	$\frac{1}{23}$	56	$\frac{126}{551}$
12	$\frac{36}{91}$	30	$\frac{171}{496}$	45	$\frac{91}{1081}$, $\frac{378}{1081}$	57	$\frac{171}{1711}$, $\frac{630}{1711}$
14	$\frac{1}{8}$	32	$\frac{70}{187}$	47	$\frac{145}{392}$	58	$\frac{26}{59}$
15	$\frac{45}{136}$	35	$\frac{2}{37}$, $\frac{35}{222}$, $\frac{77}{222}$	48	$\frac{38}{245}$, $\frac{17}{35}$	60	$\frac{351}{1891}$, $\frac{406}{1891}$, $\frac{666}{1891}$
17	$\frac{22}{57}$	36	$\frac{325}{703}$	50	$\frac{155}{442}$	61	$\frac{100}{651}$
19	$\frac{1}{2}$	37	$\frac{92}{247}$	51	$\frac{153}{1378}$	62	$\frac{247}{672}$

Table 3. All values for m, $3 \leq m \leq 62$ and corresponding values of $\alpha \leq 1/2$ for which the assumptions of Theorem 3 (b) are satisfied.

3. Consequences for Fisher's exact test

Fisher's (nonrandomized) exakt Test $\tilde{\psi}$ is given by

$$\tilde{\psi}(x|x+y,m,m,\alpha) = \begin{cases} 1, & \psi(x|x+y,m,m,\alpha) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

The (unconditional) two-dimensional power function of $\tilde{\psi}$ can be calculated by

$$\tilde{\beta}(p,q|m,n,\alpha) = \sum_y \sum_x \tilde{\psi}(x|x+y,m,n,\alpha) g(x|m,p) g(y|n,q), \;\; p,q \in [0,1].$$

The next Theorem yields that there are situations where the power function of Fisher's exact test for sample sizes $(m+1,n)$ is strictly greater than for sample sizes (m,n). Some of these situations are covered by Theorem 2.

Theorem 4. If (1.3) is valid, then

$$\tilde{\psi}(x|x+y,m,n,\alpha) = \tilde{\psi}(x+1|x+y,m+1,n,\alpha), \quad x = 0,\ldots,m, \quad y = 0,\ldots,n-1. \tag{3.2}$$

Moreover, (3.2) implies

$$\tilde{\beta}(p,q|m,n,\alpha) > \tilde{\beta}(p,q|m+1,n,\alpha) \quad \forall p,q \in (0,1).$$

Proof. Using similar arguments as in the beginning of the proof of Lemma 1a,b, Theorem 2 yields that (1.3) implies the existence of integers $0 \leq u_1 < \cdots < u_n < m$, such that

$$\tilde{\psi}(x|x+y,m+1,n,\alpha) = \begin{cases} 0, & \text{for } 0 \leq x \leq u_{y+1}+1, \\ 1, & \text{otherwise,} \end{cases} \tag{3.3}$$

and

$$\tilde{\psi}(x|x+y,m,n,\alpha) = \begin{cases} 0, & \text{for } 0 \leq x \leq u_{y+1}, \\ 1, & \text{otherwise.} \end{cases} \tag{3.4}$$

holds for all $y = 0, \ldots, n-1$. This implies (3.2).

On the other hand, (3.2) implies the existence of integers $0 \leq u_1 < \cdots < u_n < m$, such that (3.3) and (3.4) holds. By noting that $\tilde{\psi}(x+1|x+n, m+1, n, \alpha) = \tilde{\psi}(x|x+n, m, n, \alpha) = 0,\ x = 0, \ldots, m$ and $\tilde{\psi}(0|y, m+1, n, \alpha) = 0,\ y = 0, \ldots, n,$ we conclude that

$$\tilde{\beta}(p, q|m, n, \alpha) - \tilde{\beta}(p, q|m+1, n, \alpha) = \sum_{y=0}^{n-1} g(y|n, q)[H(u_{y+1}+1|m, p) - H(u_{y+1}+2|m+1, p)].$$

Using (A.5) we get

$$\begin{aligned} H(u_{y+1}+1|m, p) - H(u_{y+1}+2|m+1, p) & \\ &= g(u_{y+1}+1|m, p) + H(u_{y+1}+2|m, p) - H(u_{y+1}+2|m+1, p) \\ &= g(u_{y+1}+1|m, p) - pg(u_{y+1}+1|m, p) \\ &= (1-p)g(u_{y+1}+1|m, p). \end{aligned}$$

This completes the proof of Theorem 4. □

There exit situations where (1.3) is not valid but (3.2) holds. Numereous examples for this phenomenon can be constructed using very small values of α, m and n. But there exist some practical relevant cases, too. For instance, (3.2) is fullfilled for $(m, n) = (12, 12)$ and $\alpha = 0.02$. In this case Fisher's exact test is given by

$\widetilde{\psi}(x, x+y, 12, 12, 0.02)$

12	1	1	1	1	1	1	1	1	0	0	0	0	0
11	1	1	1	1	1	1	0	0	0	0	0	0	0
10	1	1	1	1	1	0	0	0	0	0	0	0	0
9	1	1	1	1	0	0	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0	0	0	0	0	0	0
7	1	1	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	10	11	12

and condition (3.2) is satisfied.

The reason for the observed phenomenon is the discreteness of the underlying distributions as well as fixing the level α in advance. It should be noted that the unconditional size of Fisher's exact test for $(m, n) = (13, 12)$ is approximately 0.0077574 which is strictly less than the corresponding unconditional size 0.0114223 for $(m, n) = (12, 12)$, i.e., the loss in power yields a gain with respect to (unconditional) type I errors. The maximum difference between the unconditional power functions over the alternative K is approximately 0.1370881 and is attained in $(p, q) = (0.3846154, 0.0)$. The

maximum conditional size (0.0195629) for $(m,n) = (12,12)$ is attained for $s = 12$, the maximum conditional size (0.0149068) for $(m,n) = (13,12)$ is attained at $s = 20$.

4. Open problems

The question arises whether there exist further parameter configurations $(m,n) \neq (m',n')$ with identical unconditional power functions than given in Theorem 3. The existence of a solution of (2.3) may be described via n urn experiments. Suppose we have an urn with m red and n black balls. The i'th experiment consists in drawing exactly $u_i + i$ balls. Let $A(i,u_i)$ denote the event of drawing at most u_i red balls in the i'th experiment. Then the question is, whether there exist integers $0 \leq u_1 < \cdots < u_n < m$ such that all the events $A(i,u_i),\ i = 1,\ldots,n$ have the same probability.
The attempt to find a set $(\alpha, u_1, \ldots, u_n)$ of solutions of (2.3) for $3 \leq m,n \leq 500$ for $\alpha \in (0,1)$, ($\alpha \neq 1/2$ if $m = n$) by using an algebraic computer package failed, moreover, for $3 \leq m,n \leq 100$ we found that if $F(u|u+i,m,n) = 1-\alpha$ for some i,u, then there exists at most one tuple $(j,v) \neq (i,u)$ with $F(v|v+j,m,n) = 1-\alpha$, except for $(m,n,\alpha) = (85,35,1/2)$ where $F(1|2,85,35) = F(42|60,85,35) = F(83|118,85,35) = 1/2$. For $n = 3$ we found that (2.3) is not solvable for $3 \leq m \leq 66500$ whenever $\alpha \neq 1/2$. These observations lead us to the following conjecture.

Conjecture. Let $\alpha \in (0,1)$ and $m,n \geq 3$. Then (1.3) holds, if and only if $m = n$ and $\alpha = 1/2$.

Finally the whole bag of tricks results in number theoretic problems which also show that the most critical point of our conjecture is the case $n = 3$. For example, the equation $F(u|u+1,m,n) = F(v|v+n,m,n)$ (which is a necessary but not sufficient condition for the validity of (1.3), consider $i = 1,n,\ u_1 = u,\ u_n = v$ in (2.3)) is equivalent to $1 - f(u+1|u+1,m,n) = f(v|v+n,m,n)$. A straight forward calculation shows that the last equation is equivalent to

$$\prod_{i=1}^{n}(m-u-1+i) + \prod_{i=1}^{n}(v+i) = \prod_{i=1}^{n}(m+i).$$

In other words, we are looking for integer solutions a,b,c satisfying the diophantine equation

$$\prod_{i=1}^{n}(a+i) + \prod_{i=1}^{n}(b+i) = \prod_{i=1}^{n}(c+i). \tag{3.5}$$

Harborth (1988) called this type of equation a Fermat-like binomial equation. More precisely, he as well as Wunderlich (1962) for $n = 3,4$, considered the equation

$$\binom{a+n}{n} + \binom{b+n}{n} = \binom{c+n}{n}.$$

Fraenkel (1971) considered a more general class of diophantine equations by replacing i in (3.5) by $i\delta$, δ being a positive integer.

However, (3.5) has at least one solution for each $n \geq 2$, namely $a = b = n-1, c = n$. This has already been mentioned in Fraenkel (1971). By the way, this corresponds to the case $\alpha = 1/2$ and $m = n$. Now the question is, whether there exist further solutions for $n \geq 3$. For $n = 3$ it is

mentioned in Wunderlich (1962) that S. Chowla (in a paper which seems to be unpublished) proved that the number of solutions of (3.5) is infinite. A proof can be found in Sierpiǹski (1962), confer also the remarks in Fraenkel (1971) on this topic. Setting $a = u - v - 1$, $c = u + v - 1$ and $b = 4v - 1$, for $n = 3$ (3.5) is equivalent to $31u^2 - 3v^2 = 1$, or, setting $x = 31u$, we get

$$x^2 - 93v^2 = 31, \tag{3.6}$$

which is a diophantine equation (or Pell equation) of the type $x^2 - Cy^2 = R$. It is well known, that a diophantine equation of this type has either infnitely many solutions or no solution. Since $x = 14 \times 31$, $v = 45$ solves (3.6), we obtain in fact that (3.5) has infinitely many positive integer solutions for $n = 3$.
For $n \geq 4$, the number of further solutions seems to be rather small although Harborth (1988) proved that there exist infinitely many n's with further solutions. Fraenkel (1971) conjectures that the number of solutions is finite for any fixed $n > 3$. Numerical calculations have shown, that there exist only seven solutions (n, a, b, c) for $4 \leq n \leq 500$, $a \leq b$, $1 \leq c \leq 500$ namely $(4, 128, 186, 196)$, $(6, 8, 9, 10)$, $(6, 13, 13, 15)$, $(35, 83, 83, 85)$, $(40, 63, 64, 65)$, $(204, 491, 491, 493)$ and $(273, 440, 441, 442)$. We found no further solution for $n = 4$ for $c \leq 126900$, for $n = 5, 6$ for $c \leq 10000$. As far as we know it is not known whether the set of solutions of (3.5) for fixed $n \geq 4$ is finite.
Some readers may be interested in visiting the WWW-address of Sloane (2000) where sequences of solutuions (which are partially slightly shifted) for $n = 2, 3, 4$ of (3.5) can be found, cf. the sequences A012132, A002311, A020329.

A further question is whether we can conclude that the power functions of the UMPU-tests based on sample sizes (m, n) and $(m + 1, n)$ satisfy $\beta(p, q|m, n, \alpha) \neq \beta(p, q|m + 1, n, \alpha)$ for all $p \neq q$, $p, q \in (0, 1)$, provided that they do not coincide on the entire parameter space. If the power functions coincide on an open subset of the parameter space, then it is immediate that they coincide on the entire parameter space.

It remains the question whether there exit further practically relevant models with testing situations, where the power of optimal tests considered as a function of the sample sizes is not uniformly increasing.

Appendix

Proof of Lemma 1. To prove part a) and b) we make use of the following facts, which are valid for all $m, n \in I\!N$, $x \in \{0, \ldots, m\}$, $y \in \{0, \ldots, n\}$.

$$\psi(x|x + y, m, n, \alpha) \in (0, 1) \Rightarrow \psi(x|x + y, m, n, \alpha) = \frac{\alpha - 1 + F(x|x + y, m, n)}{f(x|x + y, m, n)} \tag{A.1}$$

$$\left.\begin{array}{l} \psi(x - 1|x + y, m, n, \alpha) = 0 \\ \psi(x + 1|x + y, m, n, \alpha) = 1 \end{array}\right\} \Rightarrow \psi(x|x + y, m, n, \alpha) = \frac{\alpha - 1 + F(x|x + y, m, n)}{f(x|x + y, m, n)} \tag{A.2}$$

(A.3) $$\psi(x|x+y,m+1,n,\alpha) \le \psi(x|x+y,m,n,\alpha) \le \psi(x+1|x+1+y,m+1,n,\alpha)$$

(A.4) $$\psi(x|x+y,m,n,\alpha) \le \psi(x+1|x+1+y,m,n,\alpha) \le \psi(x+1|x+y,m,n,\alpha)$$

(A.5) $$H(x|m,p) - H(x|m-1,p) = pg(x-1|m-1,p),$$

where $H(x,m,p) = \sum_{y=x}^{m} g(y|m,p)$ for $x \in \{0,\ldots,m\}$ and $H(x,m,p) = 0$ for $x > m$.

(A.6) $$\frac{g(x|m,p)}{f(x|x+y,m,n,p)} = \frac{g(x+y|m+n,p)}{g(y|n,p)}$$

(A.7) $$\frac{m}{m-x}(1-p)g(x|m-1,p) = g(x|m,p) = \frac{m}{x}pg(x-1|m-1,p) \qquad x \notin \{0,m\}$$

(A.8) $$F(x|x+y,m,n) = (1-\frac{x+y}{m+n})F(x|x+y,m-1,n) + \frac{x+y}{m+n}F(x-1|x-1+y,m-1,n)$$

The monotonicity properties (A.3) and (A.4) can be found in Finner & Strassburger (2000). All remaining properties are easily verified.

Proof of Lemma 1 Part a). Since $\psi(0|0,m,n,\alpha) = \alpha$ for all $m,n \in I\!N$ there exists an integer $u \in \{1,\ldots,m\}$, such that

$$\psi(x|x,m+1,n,\alpha) = \begin{cases} \dfrac{\alpha}{f(x|x,m+1,n)}, & \text{for } 0 \le x \le u+1, \\ 1, & \text{otherwise.} \end{cases}$$

Combining (A.1) with (A.3) we conclude that there exist $r \in [0,1]$, so that

$$\psi(x|x,m,n,\alpha) = \begin{cases} \dfrac{\alpha}{f(x|x,m,n)}, & \text{for } 0 \le x \le u, \\ r, & \text{if } x = u+1, \\ 1, & \text{otherwise.} \end{cases}$$

For $p \in (0,1)$ we get with (A.6)

$$\begin{aligned} \beta_1(p|0,m+1,n,\alpha) &= H(u+2|m+1,p) + \alpha \sum_{x=0}^{u+1} g(x|m+1,p)/f(x|x,m+1,n) \\ &= H(u+2|m+1,p) + \alpha \sum_{x=0}^{u+1} (1-p)^{-n} g(x|m+n+1,p) \\ &= H(u+2|m+1,p) + \alpha(1-p)^{-n}[1 - H(u+2|m+n+1,p)]. \end{aligned}$$

The identity (A.5) yields

(A.9) $$\begin{aligned} \beta_1(p|0,m+1,n,\alpha) &= H(u+2|m,p) + pg(u+1|m,p) \\ &\quad + \alpha(1-p)^{-n}[1 - H(u+2|m+n,p) - pg(u+1|m+n,p)]. \end{aligned}$$

Similarly we get

$$\beta_1(p|0,m,n,\alpha) = H(u+2|m,p) + rg(u+1|m,p) + \alpha(1-p)^{-n}[1 - H(u+1|m+n,p)].$$

Combining this equation with (A.9) we obtain

(A.10) $$\begin{aligned} &\beta_1(p|0,m+1,n,\alpha) - \beta_1(p|0,m,n,\alpha) \\ &\quad = (p-r)g(u+1|m,p) + \alpha(1-p)^{-n}[g(u+1|m+n,p) - pg(u+1|m+n,p)]. \end{aligned}$$

If $u = m$, the r.h.s. of (A.10) is equal to $\alpha(1-p)^{1-n}g(m+1|m+n,p)$, which is positive for all $p \in (0,1)$ so we can reduce our attention to the case $u \leq m-1$. Now the r.h.s. of (A.10) equals $p^{u+1}(1-p)^{m-u-1}[(p-r)\binom{m}{u+1} + \alpha\binom{m+n}{u+1}(1-p)]$. This term equals zero for all $p \in (0,1)$ if and only if

$$p\left[\binom{m}{u+1} - \alpha\binom{m+n}{u+1}\right] = r\binom{m}{u+1} - \alpha\binom{m+n}{u+1} \qquad \forall p \in (0,1).$$

But this can happen if and only if $\binom{m}{u+1}/\binom{m+n}{u+1} = \alpha$ and $r = 1$, or, equivalently $F(u|u+1,m,n) = 1-\alpha$. □

Proof of Lemma 1 part b). Let $y \in \{1,\ldots,n-1\}$. By assumption there exists an $u \in \{0,\ldots,m-1\}$ such that $F(u,u+y,m,n) = 1-\alpha$, hence $\psi(u|u+y,m,n,\alpha) = 0$ and $\psi(u+1|u+y,m,n,\alpha) = 1$. The monotonicity property (A.3) yields $\psi(u|u+y,m+1,n,\alpha) = 0$ and $\psi(u+2|u+y+1,m+1,n,\alpha) = 1$ and by combining (A.3) and (A.4) we get $\psi(u|u+y+1,m+1,n,\alpha) = 0$. Together with (A.2) this yields $\psi(u+1|u+y+1,m+1,n,\alpha) = (\alpha - 1 + F(x|x+y,m+1,n))/f(x|x+y,m+1,n)$. Hence there exist an integer $v \in \{u+1,\ldots,m\}$ such that

$$\psi(x|x+y,m+1,n,\alpha) = \begin{cases} 0, & \text{for } 0 \leq x \leq u, \\ \dfrac{\alpha - 1 + F(x|x+y,m+1,n)}{f(x|x+y,m+1,n)}, & \text{for } u+1 \leq x \leq v+1, \\ 1, & \text{otherwise.} \end{cases}$$

Similarly as in the proof of part a) by making use of (A.3) and (A.1) we conclude that there exists a real number $r \in (0,1]$, such that

$$\psi(x|x+y,m,n,\alpha) = \begin{cases} 0, & \text{for } 0 \leq x \leq u, \\ \dfrac{\alpha - 1 + F(x|x+y,m,n)}{f(x|x+y,m,n)}, & \text{for } u+1 \leq x \leq v, \\ r, & \text{for } x = v+1, \\ 1, & \text{otherwise.} \end{cases}$$

Using (A.6) we get

$$\beta_1(p|y,m,n,\alpha) = H(v+2|m,p) + rg(v+1|m,p) + \sum_{x=u+1}^{v} W(x|y,m,n,p,\alpha) \tag{A.11}$$

and

$$\beta_1(p|y,m+1,n,\alpha) = H(v+2|m+1,p) + \sum_{x=u+1}^{v+1} W(x|y,m+1,n,p,\alpha), \tag{A.12}$$

where W is defined by

$$W(x|y,m,n,p,\alpha) = \frac{g(x+y|m+n,p)}{g(y|n,p)}[\alpha - 1 + F(x|x+y,m,n)].$$

With (A.8) and (A.7) we obtain

$$\begin{aligned} W(x|y,m+1,n,p,\alpha) &= (1 - \frac{x+y}{n+m+1})\frac{g(x+y|m+n+1,p)}{g(y|n,p)}[\alpha - 1 + F(x|x+y,m,n)] \\ &\quad + \frac{x+y}{n+m+1}\frac{g(x+y|m+n+1,p)}{g(y|n,p)}[\alpha - 1 + F(x-1|x-1+y,m,n)] \end{aligned}$$

$$
\begin{aligned}
&= (1-p)\frac{g(x+y|m+n,p)}{g(y|n,p)}[\alpha - 1 + F(x|x+y,m,n)] \\
&\quad + p\frac{g(x-1+y|m+n,p)}{g(y|n,p)}[\alpha - 1 + F(x-1|x-1+y,m,n)] \\
&= (1-p)W(x|y,m,n,p,\alpha) + pW(x-1|y,m,n,p,\alpha),
\end{aligned}
$$

i.e., (A.12) can be written as

$$
\begin{aligned}
\beta_1(p|y,m+1,n,\alpha) &= H(v+2|m+1,p) + \sum_{x=u+1}^{v} W(x|y,m,n,p,\alpha) \\
&\quad + (1-p)W(v+1|y,m,n,p,\alpha) + pW(u|y,m,n,p,\alpha)
\end{aligned} \tag{A.13}
$$

Noting that $W(u|y,m,n,p,\alpha) = 0$ since $F(u|u+y,m,n) = 1-\alpha$, combination of (A.11) and (A.13) and application of (A.5) results in

$$
\begin{aligned}
\beta_1(p|y,m+1,n,\alpha) - \beta_1(p|y,m,n,\alpha) &= (p-r)g(v+1|m,p) \\
&\quad + (1-p)W(v+1|y,m,n,p,\alpha).
\end{aligned} \tag{A.14}
$$

If $v = m$ the r.h.s. of (A.14) is equal to $\alpha\binom{m+n}{m+y+1}p^{m+1}/\binom{n}{y}$, which is positive for all $p \in (0,1)$. So we can reduce attention to the case $v \leq m-1$. Now the r.h.s. of (A.14) equals

$$
g(v+1|m,p)\left[(p-r) + (1-p)\frac{\alpha - 1 + F(v+1|v+1+y,m,n)}{f(v+1|v+1+y,m,n)}\right].
$$

This term equals zero for all $p \in (0,1)$ if and only if

$$
p + (1-p)\frac{\alpha - 1 + F(v+1|v+1+y,m,n)}{f(v+1|v+1+y,m,n)} = r \qquad \forall p \in (0,1).
$$

But this can happen if and only if $[\alpha - 1 + F(v+1|v+1+y,m,n)]/f(v+1|v+1+y,m,n) = 1$ and $r = 1$, or, equivalently $F(v|v+y+1,m,n) = 1-\alpha$. This completes the proof. □

Acknowledgement. We would like to thank Werner Schachinger from the University of Vienna for some helpful hints concerning diophantine equations.

References

Finner, H. & Strassburger, K. (2000). Structural properties of UMPU-tests for 2×2 tables and some implications. *Submitted for publication.*

Fisher, R. A. (1934) Statistical methods for research workers. (originally published 1925, 14th ed. revised and enlarged 1973. *Oliver and Boyd, Edinburgh.*

Fisher, R. A. (1935). The logic of inductive inference. *J. R. Statist. Soc., n. Ser.* **98**, 39-82.

Fraenkel, A. S. (1971). Diophantine equations involving generalized triangular and tetrahedral numbers. *Computers in Number Theory, Proc. Atlas Sympos. No. 2, Oxford 1969,* 99-114.

Harborth, H. (1988). Fermat-like binomial equations. *In: A.N. Philippou et al. (eds.), Applications of Fibonacci Numbers, Kluwer Academic Publ., 1* 1-5.

Irwin, J. O. (1935). Tests of significance for difference between percentages based on small numbers. *Metron* **12**, 83-94.

Khatri, M. N. (1955). Triangular numbers and Phythagorean triangles. *Scripta Math* **21**, 94.

Sierpiǹski, W. (1962). Sur une propriètè des nombres tètraèdraux. *Elem. Math.* **17**, 29-30.

Sloane, N. J. A. (2000). The On-Line Encyclopedia of Integer Sequences. Published electronically at http://www.research.att.com/∼njas/sequences/.

Tocher, K. D. (1950) Extension of the Neyman-Pearson theory of tests to discontinuous variates. *Biometrika* **37**, 130-144.

Wunderlich, M. (1962). Certain properties of pyramidal and figurate numbers. *Math. Comp.* **16**, 482-486.

Yates, F. (1934). Contingency tables involving small numbers and the χ^2-test. *J. R. Statist. Soc. Supp.* **1**, 217-235.

SYMPLECTIC MODELS OF $n-$PARTICLE SYSTEMS

K. Grudzinski* and B.G.Wybourne**

Instytut Fizyki, Uniwersytet Mikołaja Kopernika
ul. Grudziądzka 5/7
87-100 Toruń
Poland

The universe is infinite in all directions, not only above us in the large but also below us in the small
— Emil Wiechert (1896)

Abstract

The dynamical group $\mathcal{S}p(6n,R)$ is used to give a description of the states for $n-$noninteracting particles confined by an isotropic three-dimensional harmonic oscillator potential. The subgroup structure of the dynamical group is used to determine the relevant spins and unitary group representations of the $n-$particle states. This is a necessary precursor to developing model Hamiltonians for describing systems such as quantum dots and nuclei in terms of polynomials in the dynamical group generators.

1. Introduction

The isotropic three-dimensional harmonic oscillator (henceforth we will abbreviate to just $\mathcal{HO}$) for a single particle is one of the few problems whose Schrodinger equation is completely solvable. The complete set of states span a single irreducible representation of the metaplectic group $\mathcal{M}p(6)$ which is the covering group of the symplectic group $\mathcal{S}p(6,R)$[1-8]. Upon the restriction $\mathcal{M}p(6) \rightarrow \mathcal{S}p(6,R)$ the single irreducible representation of $\mathcal{M}p(6)$ decomposes into a pair of irreducible representations which we designate as $\langle s;(0)\rangle$ and $\langle s;(1)\rangle$ [3]. The irreducible representation $\langle s;(0)\rangle$ is spanned by the complete set of *even* parity states and $\langle s;(1)\rangle$ by the odd parity states. Throughout this paper we shall often just write $\mathcal{S}p(N)$ rather than $\mathcal{S}p(N,R)$ with the understanding that we will always be referring to the non-compact symplectic group defined on reals and *not* the compact symplectic group.

For $n-$noninteracting particles the dynamical group is $\mathcal{M}p(6n)$[7,8] and again the complete set of states span a single irreducible representation of $\mathcal{M}p(6n)$. Upon the restriction $\mathcal{M}p(6n) \rightarrow \mathcal{S}p(6n,R)$ the single irreducible representation of $\mathcal{M}p(6n)$ decomposes into a pair of irreducible representations which again we designate as $\langle s;(0)\rangle$ and $\langle s;(1)\rangle$ with the *even* parity states spanning the $\langle s;(0)\rangle$ irreducible representation and $\langle s;(1)\rangle$ by the odd parity states.

The group $\mathcal{M}p(6n)$ has a very rich subgroup structure[4,5,7,8] which we will first outline and then direct our attention to the relevant group-subgroup decompositions leading to a detailed classification of the states and the identification of their spin and unitary $\mathcal{U}(3)$ structure. This should then make it possible to start to develop model Hamiltonians in terms of polynomials in the relevant group generators for $n-$interacting particles in applications associated with quantum dots and symplectic models of nuclei.

2. The substructure of the dynamical group $\mathcal{M}p(6n)$

Let us start by considering the slightly more general case of $n-$noninteracting particles in a $d-$dimensional harmonic oscillator potential. The dynamical group may be formally constructed from the coordinate and momentum operators of the individual particles under the usual Heisenberg commutation relations. Bilinear combinations of these operators are constructed to close under commutation and the associated Lie algebra identified. It is readily found that indeed they close upon the algebra associated with the metaplectic group $\mathcal{M}p(2nd)$ which is the covering group of the non-compact symplectic group $\mathcal{S}p(2nd,R)$. The metaplectic group $\mathcal{M}p(2nd)$ has a very rich subgroup structure[8] as shown in Fig.1. These subgroup structures can be determined by contracting on particle or spatial indices. The diversity of the subgroup structures reflect different ways of separating the spatial and particle number dependencies. Thus the subgroup $\mathcal{O}(d)$ describes the angular momentum states of the system while the subgroup $\mathcal{O}(n)$ gives information on the permutational symmetries of the states via the symmetric group $\mathcal{S}(n)$ which is a subgroup of $\mathcal{O}(n)$.

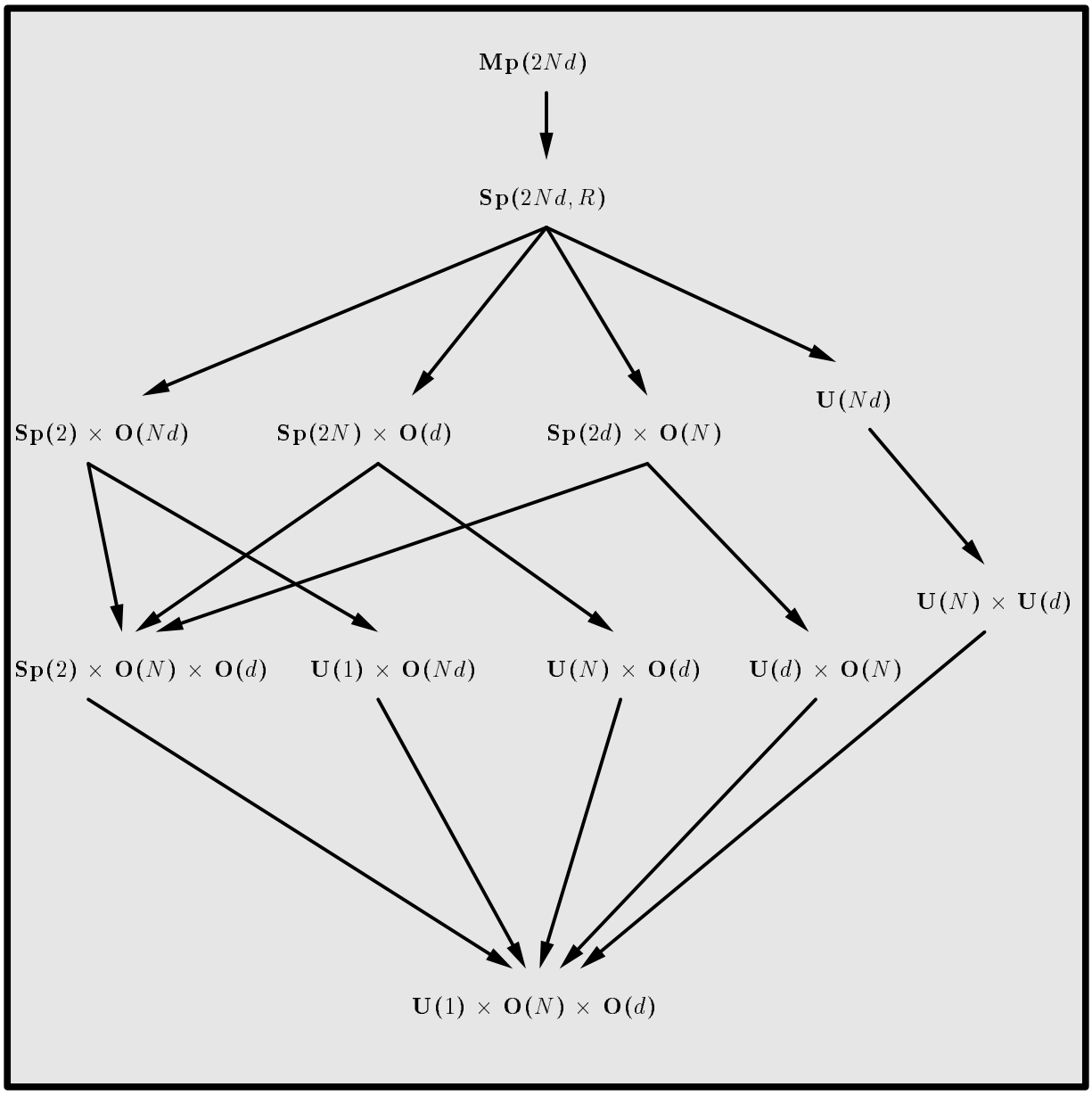


Fig. 1: Group-subgroup structures appropriate to quantum dots.

3. Labelling $\mathcal{S}p(2N, R)$ irreducible representations

The labelling of the irreducible representations of compact Lie groups in terms of partition labels is well established[9]. Here we shall limit ourselves to discussion of the so-called positive discrete unitary irreducible representations of the group $\mathcal{S}p(2n, R)$ and its double covering group, $\mathcal{M}p(2n)$, drawing heavily upon references [2] and [3]. These irreducible representations are all infinite dimensional and are characterised by a *lowest weight* with respect to the ordering of weights of the maximal compact subgroup $U(n)$. There exists a harmonic representation, $\tilde{\Delta}$, associated with the Heisenberg algebra. This is a true, unitary, infinite dimensional irreducible representation of the double covering group $\mathcal{M}p(2n)$ of $\mathcal{S}p(2n, R)$, the so-called *metaplectic group*. This representation is reducible into the sum of two irreducible representations $\tilde{\Delta}_+$ and $\tilde{\Delta}_-$ whose leading weights are $(\frac{1}{2}\frac{1}{2}\ldots\frac{1}{2})$ and $(\frac{3}{2}\frac{1}{2}\ldots\frac{1}{2})$ corresponding to the highest weights of the representations $\varepsilon^{\frac{1}{2}}\{0\}$ and $\varepsilon^{\frac{1}{2}}\{1\}$ which appear in the restriction of $\mathcal{S}p(2n, R)$ to

its maximal compact subgroup $U(n)$.

The tensor powers $\tilde{\Delta}^k$ all decompose into a direct sum of unitary irreducible representations of $\mathcal{M}p(2n)$. All those irreducible representations which derive from $\tilde{\Delta}^k$ for some k will be referred to as *harmonic series representaions*. All those irreducible representations that appear in $\tilde{\Delta}^k$ will be labelled by the symbols $\langle\frac{k}{2}(\lambda)\rangle$. The harmonic series representations appearing in $\tilde{\Delta}^k$ are in one-to-one correspondence with the terms arising in the branching rule appropriate to the restriction from $\mathcal{M}p(2nk)$ to $\mathcal{S}p(2n,R)\times\mathcal{O}(k)$

$$\tilde{\Delta}\rightarrow\sum_{\lambda}\langle\frac{k}{2}(\lambda)\rangle\times[\lambda] \tag{1}$$

where the summation is carried out over all partitions $(\lambda)=(\lambda_1,\lambda_2,\ldots)$ for which the conjugate partition $(\tilde{\lambda})=(\tilde{\lambda}_1,\tilde{\lambda}_2,\ldots)$ satisfies the constraints

$$\tilde{\lambda}_1+\tilde{\lambda}_2\leq k \tag{2a}$$

and

$$\tilde{\lambda}_1\leq n \tag{2b}$$

Irreducible representations of $\mathcal{S}p(2n,R)$ $\langle\frac{1}{2}k(\lambda)\rangle$ satisfying Eq.(2) will be said to be *standard* and we may limit our attention to these irreducible representations of $\mathcal{S}p(2n,R)$.

The value of $\frac{k}{2}$ maybe an integer (k *even*) or a half-odd-integer (k *odd*). In terms of inputting and outputting $\mathcal{S}p(2n,R)$ labelled irreducible representations into SCHUR it is useful to introduce the equivalent notation

$$\langle s\kappa;(\lambda)\rangle\equiv\langle\frac{k}{2}(\lambda)\rangle \tag{3}$$

where

$$\frac{k}{2}=s+\kappa \tag{4}$$

with κ being the integer part of $\frac{k}{2}$ and the residue part is $s=0$ or $\frac{1}{2}$. Thus we have the typical notational equivalences

$$\langle s1;(\lambda)\rangle\equiv\langle\frac{3}{2}(\lambda)\rangle, k=3\quad\langle 1;(\lambda)\rangle\equiv\langle 1(\lambda)\rangle\quad k=2$$

SCHUR accepts irreducible representation labels in the form of lists of $\langle s\kappa;\lambda\rangle$ and standardises the input in accordance with the constraints of Eq.(2) making null all non-standard $\mathcal{S}p(2n,R)$ irreducible representations.

4. Lowest energy states for non-interacting particles in a $\mathcal{HO}$

In the case of n non-interacting spin $\frac{1}{2}$ particles in a three-dimensional isotropic $\mathcal{HO}$ potential the energy of a given state is simply the sum of the one-particle energies (cf. Fig. 2) and hence the lowest energy state associated with a given $\mathcal{S}p(6,R)$ multiplet $[\kappa(\lambda)\rangle$ is, relative to the groundstate energy,

$$w_\lambda\hbar\omega \tag{5}$$

where ω is the oscillator angular frequency and w_λ is the weight of the partition (λ). Representations of $\mathcal{S}p(6,R)$ having different partitions but of the same weight will have the same zero-order energy as given in Eq. (5).

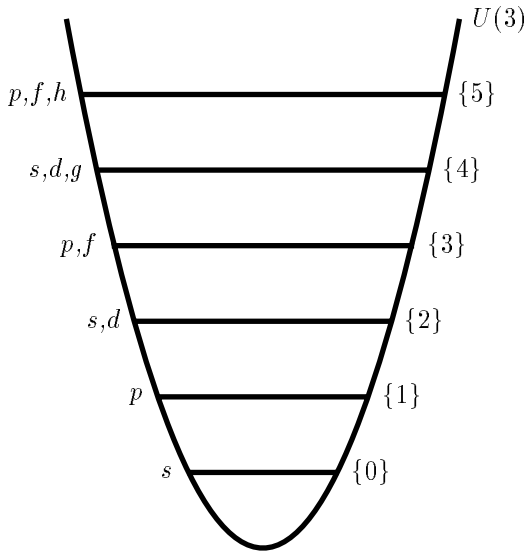


Fig. 2: The states of a single particle in a harmonic oscillator potential.

The states of $n-$particles may be associated with occupations of particles in various of the single particle $\mathcal{U}(3)$ multiplets subject to the Pauli exclusion principle. It is convenient to speak of $n-$particle configurations of the form

$$\{0\}^{m_0}\{1\}^{m_1}\{2\}^{m_2}\dots \tag{6}$$

where the exponents are the occupation numbers for the various $\mathcal{U}(3)$ single particle irreducible representations. The $\mathcal{U}(3)$ states of weight w for $n-$particles may be determined as follows

1. Partition the integer w into n parts allowing zero parts if necessary.
2. Even weight partitions involve even parity states otherwise odd parity states.
3. Replace each part ,i, by $\{i\}$ which then labels the $\mathcal{U}(3)$ irrep for a single particle in the $i-$th harmonic oscillator orbital. A given orbital i can accommodate up to $d_i = (i+1)(i+2)$ particles with spin $\frac{1}{2}$ and hence partitions having parts, i, with a multiplicity exceeding d_i must be discarded.
4. For a given partition containing k distinct non-repeating parts form the $\mathcal{SU}(2)\times\mathcal{U}(3)$ Kronecker product
$$\{\frac{1}{2}\}\times\{i_1\}.\{\frac{1}{2}\}\times\{i_2\}\cdots.\{\frac{1}{2}\}\times\{i_k\} \tag{7}$$
to give a series of $\mathcal{SU}(2)^S\times\mathcal{U}(3)$ multiplets.
5. If the parts i are repeated with a multiplicity m then evaluate the plethysm
$$(\{\frac{1}{2}\}\{i\})\otimes\{1^m\} = \sum_{a=[\frac{m+1}{2}]}^{m} {}^{(2a-m+1)}(\{i\}\otimes\{2^{m-a}1^{2a-m}\}) \tag{8}$$
where the spin multiplicity $(2S+1) = (2a-m+1)$ has been written as a superscript.

For $n=3$ we have for weight 4 the four partitions

$$4+0+0,\quad 3+1+0,\quad 2+2+0,\quad 2+1+1 \tag{9}$$

Applying the above algorithm we find for the first partition a $\mathcal{U}(3)$ multiplet $\{4\}$ with $S=\frac{1}{2}$ corresponding to two particles in the $\{0\}$ orbital and one in the $\{4\}$ orbital.. The second partition gives two $\mathcal{U}(3)$ multiplets, $\{4\}+\{31\}$ with spins $S=\frac{1}{2}$ and $S=\frac{3}{2}$. These are associated with the states arising from the $\mathcal{U}(3)$ configuration $\{0\}^1\{1\}^1\{3\}^1$ The third partition yields the $\mathcal{U}(3)$ multiplet $\{31\}$ with $S=\frac{3}{2}$ and the

$\mathcal{U}(3)$ multiplets $\{4\}+\{31\}+\{2^2\}$ with spin $S=\frac{1}{2}$, corresponding to the configuration $\{0\}^1\{2\}^2$. The fourth partition yields the two $\mathcal{U}(3)$ multiplets $\{31\}+\{2^2\}$ with spin $S=\frac{3}{2}$ and the three $\mathcal{U}(3)$ multiplets $\{4\}+2\{31\}+\{2^2\}+\{21^2\}$ with spin $S=\frac{1}{2}$, corresponding to the configuration $\{1\}^2\{2\}^1$. Thus for spin $S=\frac{3}{2}$ we obtain the $\mathcal{U}(3)$ multiplets $\{4\}+3\{31\}+\{21^2\}$ and for spin $S=\frac{1}{2}$ the $\mathcal{U}(3)$ multiplets $4\{4\}+4\{31\}+2\{2^2\}+\{21^2\}$.

5. The Lowest $\mathcal{U}(3)$ Multiplets

Filling the first k shells with particles will involve a total of

$$N_k=\frac{k(k+1)(k+2)}{3} \tag{10}$$

particles. If $n-N_k$ particles are in the lowest unfilled shell then the weights w_λ of admissible partitions labelling irreducible representations of $\mathcal{U}(3)$ will be given by

$$w_\lambda=k\left[n-\frac{(k+1)(k+2)(k+3)}{12}\right] \tag{11}$$

Thus for 12 particles we would have $w_\lambda=14$.

If the first k shells are fully occupied then the resultant state has spin $S=0$ and belongs to the $\mathcal{U}(3)$ irrep $\{p,p,p\}$ where

$$p=\left[\frac{(k-1)k(k+1)(k+2)}{2}\right] \tag{12}$$

6. Lowest Energy Even Parity 12-particle States

It is desirable to consider a reasonably large number of particles to bring out the main features of the $n-$particle problem. To be specific I shall consider the case of $12-$particles and initially just the even parity states. The lowest states will occur with the first two shells fully occupied and the remaining 4 particles occupying the third shell. We could, in terms of $\mathcal{U}(3)$ multiplets, designate the configuration as

$$\{0\}^2\{1\}^6\{2\}^4 \tag{13}$$

The spin and unitary $\mathcal{U}(3)$ multiplets can be determined by first evaluating the plethysm

$$\{1\}\{2\}\otimes\{1^4\} \tag{14}$$

for the direct product group $\mathcal{SU}(2)\times\mathcal{U}(3)$. This leads to a set of spin $S=2$ states arising from the $\mathcal{U}(3)$ plethysm $\{2\}\otimes\{1^4\}$, a set of $S=1$ states from the $\mathcal{U}(3)$ plethysm $\{2\}\otimes\{21^2\}$ and a set of $S=0$ states from the $\mathcal{U}(3)$ plethysm $\{2\}\otimes\{2^2\}$. The first two filled shells result in a single $S=0$ state transforming under $\mathcal{U}(3)$ as the $\{2^3\}$ and to obtain the final list of $\mathcal{U}(3)$ irreducible representations we must add the partition $\{2^3\}$ to those associated with each of the above plethysms to finally yield the spin S and $\mathcal{U}(3)$ multiplets given in Table 6.1.

$S=2$	$\{653\}$				
$S=1$	$\{83^2\}$	$+\{752\}$	$+\{743\}$	$+\{653\}$	$+\{5^24\}$
$S=0$	$\{842\}$	$+\{743\}$	$+\{6^22\}$	$+\{64^2\}$	

Table 6.1 Spin and $\mathcal{U}(3)$ multiplets for the $\{0\}^2\{1\}^6\{2\}^4$ configuration.

7. Second to Lowest Energy Even Parity 12-particle States

The second to lowest energy even parity 12-particle states all involve $\mathcal{U}(3)$ multiplets labelled by partitions of weight 16. Five configurations arise:-

1. $\{0\}^2\{1\}^5\{2\}^4\{3\}^1$
2. $\{0\}^1\{1\}^6\{2\}^5$
3. $\{0\}^2\{1\}^4\{2\}^6$
4. $\{0\}^2\{1\}^6\{2\}^3\{4\}^1$
5. $\{0\}^2\{1\}^6\{2\}^2\{3\}^2$

Proceeding as before we can systematically determine the various possible spins S and their associated $\mathcal{U}(3)$ multiplets to give:-

$S = 1, 2^2, 3$	$\{952\}$	$+ \{943\}$	$+ \{862\}$	$+ 3\{853\}$	$+ \{84^2\}$
	$+ 2\{763\}$	$+ 3\{754\}$	$+ 2\{6^24\}$	$+ 2\{65^2\}$	
$S = 0, 1^2, 2$	$\{11\ 32\}$	$+ \{10\ 51\}$	$+ 3\{10\ 42\}$	$+ 3\{10\ 3^2\}$	$+ \{961\}$
	$+ 6\{952\}$	$+ 7\{943\}$	$+ \{871\}$	$+ 6\{862\}$	$+ 12\{853\}$
	$+ 5\{84^2\}$	$+ 3\{7^22\}$	$+ 9\{763\}$	$+ 10\{754\}$	$+ 4\{6^24\}$
	$+ 4\{65^2\}$				
$S = 0, 1$	$\{11\ 41\}$	$+ \{11\ 32\}$	$+ \{10\ 51\}$	$+ 4\{10\ 42\}$	$+ 2\{10\ 3^2\}$
	$+ 2\{961\}$	$+ 5\{952\}$	$+ 7\{943\}$	$+ \{871\}$	$+ 6\{862\}$
	$+ 8\{853\}$	$+ 6\{84^2\}$	$+ 2\{7^22\}$	$+ 7\{763\}$	$+ 6\{754\}$
	$+ 4\{6^24\}$	$+ \{65^2\}$			

Table 7.1 Spin and $\mathcal{U}(3)$ multiplets for the $\{0\}^2\{1\}^5\{2\}^4\{3\}^1$ configuration.

$S = 2, 3$	$\{6^24\}$				
$S = 1, 2$	$\{853\}$	$+ \{763\}$	$+ \{754\}$	$+ \{65^2\}$	
$S = 0, 1$	$\{943\}$	$+ \{862\}$	$+ \{853\}$	$+ \{84^2\}$	$+ \{763\}$
	$+ \{754\}$	$+ \{6^24\}$			

Table 7.2 Spin and $\mathcal{U}(3)$ multiplets for the $\{0\}^1\{1\}^6\{2\}^5$ configuration.

$S = 2, 3, 4$	$\{65^2\}$				
$S = 3$	$\{6^24\}$				
$S = 1, 2, 3$	$\{853\}$	$+ \{763\}$	$+ 2\{754\}$	$+ \{6^24\}$	$+ \{65^2\}$
$S = 2$	$\{862\}$	$+ \{853\}$	$+ \{84^2\}$	$+ 2\{763\}$	$+ 2\{754\}$
	$+ 2\{6^24\}$	$+ \{65^2\}$			
$S = 0, 1, 2$	$\{952\}$	$+ \{943\}$	$+ 2\{862\}$	$+ 3\{853\}$	$+ 2\{84^2\}$
	$+ \{7^22\}$	$+ 3\{763\}$	$+ 3\{754\}$	$+ 2\{6^24\}$	$+ \{65^2\}$
$S = 1$	$\{961\}$	$+ 2\{952\}$	$+ 2\{943\}$	$+ \{871\}$	$+ 3\{862\}$
	$+ 5\{853\}$	$+ 2\{84^2\}$	$+ 2\{7^22\}$	$+ 5\{763\}$	$+ 5\{754\}$
	$+ 2\{6^24\}$	$+ 2\{65^2\}$			
$S = 1$	$\{10\ 3^2\}$	$+ \{952\}$	$+ \{943\}$	$+ \{871\}$	$+ \{862\}$
	$+ 3\{853\}$	$+ \{7^22\}$	$+ 2\{763\}$	$+ 2\{754\}$	$+ \{65^2\}$
$S = 0$	$\{10\ 42\}$	$+ \{961\}$	$+ \{952\}$	$+ 2\{943\}$	$+ \{8^2\}$
	$+ \{871\}$	$+ 4\{862\}$	$+ 3\{853\}$	$+ 3\{84^2\}$	$+ 3\{763\}$
	$+ 2\{754\}$	$+ 3\{6^24\}$			

Table 7.3 Spin and $\mathcal{U}(3)$ multiplets for the $\{0\}^2\{1\}^4\{2\}^6$ configuration.

$S = 1, 2$	$\{10\ 3^2\}$	$+ \{952\}$	$+ \{943\}$	$+ 2\{853\}$
	$+ \{763\}$	$+ \{754\}$	$+ \{65^2\}$	
$S = 0, 1$	$\{11\ 32\}$	$+ 2\{10\ 42\}$	$+ \{10\ 3^2\}$	$+ 2\{952\}$
	$+ 3\{943\}$	$+ 2\{862\}$	$+ 3\{853\}$	$+ 2\{84^2\}$
	$+ \{7^22\}$	$+ 2\{763\}$	$+ 2\{754\}$	$+ \{6^24\}$

Table 7.4 Spin and $\mathcal{U}(3)$ multiplets for the $\{0\}^2\{1\}^6\{2\}^3\{4\}$ configuration.

$S=0,1,2$	$\{10\ 42\}$	$+\ \{10\ 3^2\}$	$+\ \{952\}$	$+\ 2\{943\}$
	$+\ 2\{862\}$	$+\ 3\{853\}$	$+\ \{84^2\}$	$+\ 2\{763\}$
	$+\ 2\{754\}$	$+\ \{6^2 4\}$	$+\ \{65^2\}$	
$S=0$	$\{12\ 2^2\}$	$+\ \{11\ 32\}$	$+\ 3\{10\ 42\}$	$+\ 2\{952\}$
	$+\ 2\{943\}$	$+\ 3\{862\}$	$+\ 2\{853\}$	$+\ 3\{84^2\}$
	$+\ 2\{763\}$	$+\ 2\{754\}$	$+\ 2\{6^2 4\}$	
$S=1$	$2\{11\ 32\}$	$+\ 2\{10\ 42\}$	$+\ 2\{10\ 3^2\}$	$+\ 5\{952\}$
	$+\ 4\{943\}$	$+\ 2\{862\}$	$+\ 6\{853\}$	$+\ 2\{84^2\}$
	$+\ 3\{7^2 2\}$	$+\ 4\{763\}$	$+\ 5\{754\}$	$+\ \{6^2 4\}$
	$+\ 2\{65^2\}$			

Table 7.5 Spin and $\mathcal{U}(3)$ multiplets for the $\{0\}^2\{1\}^6\{2\}^2\{3\}^2$ configuration.

In practice there is no difficulty in obtaining the corresponding odd parity 12-particle states. Of course the total number of possible states is infinite and to encompass these we must return to the non-compact groups.

8. Infinite Sets of Even Parity 12-particle States

The complete set of even parity 12-particle states span the infinite dimensional irreducible representation $\langle s;(0)\rangle$ of the non-compact group $\mathcal{S}p(72,R)$. To obtain a description of the states we need to study the decomposition of the irreducible representation $\langle s;(0)\rangle$ as we move through a series of subgroups as portrayed in Fig.1. Any such decomposition involves an infinite set of subgroup irreducible representations and hence to consider manageable problems we need to introduce a cutoff. For simplicity let us consider the restriction $\mathcal{S}p(72,R) \rightarrow \mathcal{S}p(6,R)\times\mathcal{O}(12)$ and furthermore limit our attention to irreducible representations whose labelling partitions (λ) are of weight $w_\lambda \leq 16$. We readily find the decomposition as given in Table 8.1.

$\langle s;(0)\rangle$	$\langle 6;(16\ )\rangle[16\ ]$	$+\ \langle 6;(15\ 1)\rangle[15\ 1]$	$+\ \langle 6;(14\ 2)\rangle[14\ 2]$
	$+\ \langle 6;(14\ 1^2)\rangle[14\ 1^2]$	$+\ \langle 6;(14\ )\rangle[14\ ]$	$+\ \langle 6;(13\ 3)\rangle[13\ 3]$
	$+\ \langle 6;(13\ 21)\rangle[13\ 21]$	$+\ \langle 6;(13\ 1)\rangle[13\ 1]$	$+\ \langle 6;(12\ 4)\rangle[12\ 4]$
	$+\ \langle 6;(12\ 31)\rangle[12\ 31]$	$+\ \langle 6;(12\ 2^2)\rangle[12\ 2^2]$	$+\ \langle 6;(12\ 2)\rangle[12\ 2]$
	$+\ \langle 6;(12\ 1^2)\rangle[12\ 1^2]$	$+\ \langle 6;(12\ )\rangle[12\ ]$	$+\ \langle 6;(11\ 5)\rangle[11\ 5]$
	$+\ \langle 6;(11\ 41)\rangle[11\ 41]$	$+\ \langle 6;(11\ 32)\rangle[11\ 32]$	$+\ \langle 6;(11\ 3)\rangle[11\ 3]$
	$+\ \langle 6;(11\ 21)\rangle[11\ 21]$	$+\ \langle 6;(11\ 1)\rangle[11\ 1]$	$+\ \langle 6;(10\ 6)\rangle[10\ 6]$
	$+\ \langle 6;(10\ 51)\rangle[10\ 51]$	$+\ \langle 6;(10\ 42)\rangle[10\ 42]$	$+\ \langle 6;(10\ 4)\rangle[10\ 4]$
	$+\ \langle 6;(10\ 3^2)\rangle[10\ 3^2]$	$+\ \langle 6;(10\ 31)\rangle[10\ 31]$	$+\ \langle 6;(10\ 2^2)\rangle[10\ 2^2]$
	$+\ \langle 6;(10\ 2)\rangle[10\ 2]$	$+\ \langle 6;(10\ 1^2)\rangle[10\ 1^2]$	$+\ \langle 6;(10\ )\rangle[10\ ]$
	$+\ \langle 6;(97)\rangle[97]$	$+\ \langle 6;(961)\rangle[961]$	$+\ \langle 6;(952)\rangle[952]$
	$+\ \langle 6;(95)\rangle[95]$	$+\ \langle 6;(943)\rangle[943]$	$+\ \langle 6;(941)\rangle[941]$
	$+\ \langle 6;(932)\rangle[932]$	$+\ \langle 6;(93)\rangle[93]$	$+\ \langle 6;(921)\rangle[921]$
	$+\ \langle 6;(91)\rangle[91]$	$+\ \langle 6;(8^2)\rangle[8^2]$	$+\ \langle 6;(871)\rangle[871]$
	$+\ \langle 6;(862)\rangle[862]$	$+\ \langle 6;(86)\rangle[86]$	$+\ \langle 6;(853)\rangle[853]$
	$+\ \langle 6;(851)\rangle[851]$	$+\ \langle 6;(84^2)\rangle[84^2]$	$+\ \langle 6;(842)\rangle[842]$
	$+\ \langle 6;(84)\rangle[84]$	$+\ \langle 6;(83^2)\rangle[83^2]$	$+\ \langle 6;(831)\rangle[831]$
	$+\ \langle 6;(82^2)\rangle[82^2]$	$+\ \langle 6;(82)\rangle[82]$	$+\ \langle 6;(81^2)\rangle[81^2]$
	$+\ \langle 6;(8)\rangle[8]$	$+\ \langle 6;(7^2 2)\rangle[7^2 2]$	$+\ \langle 6;(7^2)\rangle[7^2]$
	$+\ \langle 6;(763)\rangle[763]$	$+\ \langle 6;(761)\rangle[761]$	$+\ \langle 6;(754)\rangle[754]$
	$+\ \langle 6;(752)\rangle[752]$	$+\ \langle 6;(75)\rangle[75]$	$+\ \langle 6;(743)\rangle[743]$
	$+\ \langle 6;(741)\rangle[741]$	$+\ \langle 6;(732)\rangle[732]$	$+\ \langle 6;(73)\rangle[73]$
	$+\ \langle 6;(721)\rangle[721]$	$+\ \langle 6;(71)\rangle[71]$	$+\ \langle 6;(6^2 4)\rangle[6^2 4]$
	$+\ \langle 6;(6^2 2)\rangle[6^2 2]$	$+\ \langle 6;(6^2)\rangle[6^2]$	$+\ \langle 6;(65^2)\rangle[65^2]$
	$+\ \langle 6;(653)\rangle[653]$	$+\ \langle 6;(651)\rangle[651]$	$+\ \langle 6;(64^2)\rangle[64^2]$
	$+\ \langle 6;(642)\rangle[642]$	$+\ \langle 6;(64)\rangle[64]$	$+\ \langle 6;(63^2)\rangle[63^2]$
	$+\ \langle 6;(631)\rangle[631]$	$+\ \langle 6;(62^2)\rangle[62^2]$	$+\ \langle 6;(62)\rangle[62]$
	$+\ \langle 6;(61^2)\rangle[61^2]$	$+\ \langle 6;(6)\rangle[6]$	$+\ \langle 6;(5^2 4)\rangle[5^2 4]$
	$+\ \langle 6;(5^2 2)\rangle[5^2 2]$	$+\ \langle 6;(5^2)\rangle[5^2]$	$+\ \langle 6;(543)\rangle[543]$
	$+\ \langle 6;(541)\rangle[541]$	$+\ \langle 6;(532)\rangle[532]$	$+\ \langle 6;(53)\rangle[53]$
	$+\ \langle 6;(521)\rangle[521]$	$+\ \langle 6;(51)\rangle[51]$	$+\ \langle 6;(4^3)\rangle[4^3]$
	$+\ \langle 6;(4^2 2)\rangle[4^2 2]$	$+\ \langle 6;(4^2)\rangle[4^2]$	$+\ \langle 6;(43^2)\rangle[43^2]$
	$+\ \langle 6;(431)\rangle[431]$	$+\ \langle 6;(42^2)\rangle[42^2]$	$+\ \langle 6;(42)\rangle[42]$
	$+\ \langle 6;(41^2)\rangle[41^2]$	$+\ \langle 6;(4)\rangle[4]$	$+\ \langle 6;(3^2 2)\rangle[3^2 2]$
	$+\ \langle 6;(3^2)\rangle[3^2]$	$+\ \langle 6;(321)\rangle[321]$	$+\ \langle 6;(31)\rangle[31]$
	$+\ \langle 6;(2^3)\rangle[2^3]$	$+\ \langle 6;(2^2)\rangle[2^2]$	$+\ \langle 6;(21^2)\rangle[21^2]$
	$+\ \langle 6;(2)\rangle[2]$	$+\ \langle 6;(1^2)\rangle[1^2]$	$+\ \langle 6;(0)\rangle[0]$

Table 8.1 Decomposition of the irreducible representation $\langle s;(0)\rangle$ of $\mathcal{S}p(72,R)$ under the restriction $\mathcal{S}p(72,R) \to \mathcal{S}p(6,R)\times\mathcal{O}(12)$ (to weight 16).

This is already a considerable list of irreducible representations. The list can be substantially reduced by noting that no partition (λ) of weight $w_\lambda \leq 13$ can yield a Pauli allowed spin state and hence all those members of the list may be removed. Under the restriction $\mathcal{S}p(6,R) \to \mathcal{U}(3)$ for an irreducible representation $\langle 6;(\lambda)\rangle$ the lowest weight $\mathcal{U}(3)$ irreducible representation is necessarily $\{\lambda\}$ and as seen from Sec. 4 partitions into fewer than three parts cannot lead to a Pauli allowed spin state and hence all irreducible representations associated with partitions into fewer than three parts may also be discarded leaving us with the much shorter list shown in Table 8.2.

$\langle s;(0)\rangle$	$\langle 6;(14\ 1^2)\rangle[14\ 1^2]$	$+\ \langle 6;(13\ 21)\rangle[13\ 21]$	$+\ \langle 6;(12\ 31)\rangle[12\ 31]$
	$+\ \langle 6;(12\ 2^2)\rangle[12\ 2^2]$	$+\ \langle 6;(12\ 1^2)\rangle[12\ 1^2]$	$+\ \langle 6;(11\ 41)\rangle[11\ 41]$
	$+\ \langle 6;(11\ 32)\rangle[11\ 32]$	$+\ \langle 6;(11\ 21)\rangle[11\ 21]$	$+\ \langle 6;(10\ 51)\rangle[10\ 51]$
	$+\ \langle 6;(10\ 42)\rangle[10\ 42]$	$+\ \langle 6;(10\ 3^2)\rangle[10\ 3^2]$	$+\ \langle 6;(10\ 31)\rangle[10\ 31]$
	$+\ \langle 6;(10\ 2^2)\rangle[10\ 2^2]$	$+\ \langle 6;(961)\rangle[961]$	$+\ \langle 6;(952)\rangle[952]$
	$+\ \langle 6;(943)\rangle[943]$	$+\ \langle 6;(941)\rangle[941]$	$+\ \langle 6;(932)\rangle[932]$
	$+\ \langle 6;(871)\rangle[871]$	$+\ \langle 6;(862)\rangle[862]$	$+\ \langle 6;(853)\rangle[853]$
	$+\ \langle 6;(851)\rangle[851]$	$+\ \langle 6;(84^2)\rangle[84^2]$	$+\ \langle 6;(842)\rangle[842]$
	$+\ \langle 6;(83^2)\rangle[83^2]$	$+\ \langle 6;(7^22)\rangle[7^22]$	$+\ \langle 6;(763)\rangle[763]$
	$+\ \langle 6;(761)\rangle[761]$	$+\ \langle 6;(754)\rangle[754]$	$+\ \langle 6;(752)\rangle[752]$
	$+\ \langle 6;(743)\rangle[743]$	$+\ \langle 6;(6^24)\rangle[6^24]$	$+\ \langle 6;(6^22)\rangle[6^22]$
	$+\ \langle 6;(65^2)\rangle[65^2]$	$+\ \langle 6;(653)\rangle[653]$	$+\ \langle 6;(64^2)\rangle[64^2]$
	$+\ \langle 6;(5^24)\rangle[5^24]$		

Table 8.2 As for Table 8.1 but with terms of weight < 14 and length < 3 removed.

The $\mathcal{O}(12)$ irreducible representations are all finite dimensional whereas those of $\mathcal{S}p(6,R)$ are all of infinite dimension. The reductions $\mathcal{S}p(6,R) \rightarrow \mathcal{U}(3)$ and $\mathcal{O}(12) \rightarrow \mathcal{S}(12)$ tell us the $\mathcal{U}(3)$ and spin contents respectively.

9. Spin Content of the 12-particle States

The spin content of the states associated with a given irreducible representation $[\lambda]$ of $\mathcal{O}(n)$ is determined by its decomposition under $\mathcal{O}(n) \rightarrow \mathcal{S}(n)$ and seeking out those irreducible representations of $\mathcal{S}(n)$ that are of the form $\{2^r\ 1^s\}$ where $2r+s=n$ and the associated spin is $S=\frac{s}{2}$. These decompositions may be determined systematically[7,8,10,11]. Typically we obtain the spin states shown in Table 9.1 for several $\mathcal{O}(12)$ irreducible representations.

$S=$	0	1	2	3	4
$[5^2\ 4]$		1			
$[64^2]$	1				
$[653]$		1	1		
$[6^2\ 2]$	1				
$[743]$	1	1			
$[752]$		1			
$[83^2]$		1			
$[842]$	1				
$[65^2]$	7	19	14	4	1
$[6^2\ 4]$	16	22	14	5	
$[754]$	26	49	26	5	
$[763]$	27	46	22	3	
$[7^2\ 2]$	7	15	4		
$[84^2]$	20	26	11	1	
$[853]$	33	59	28	4	
$[862]$	24	31	13	1	
$[871]$	3	5	1		
$[943]$	23	35	13	1	
$[952]$	17	30	11	1	
$[961]$	4	5	1		
$[10\ 3^2]$	7	13	5		
$[10\ 42]$	13	15	4		
$[10\ 51]$	2	3	1		
$[11\ 41]$	4	6	1		
$[12\ 2^2]$	1				

Table 9.1 Spin contents of some relevant $\mathcal{O}$(12 irreducible representations.

Note that in going from partitions of weight 14 to 16 the number of possible spin states rapidly increasing in a manner not unlike the Wigner type distribution that arises in the plotting of the distribution of the spacings of consecutive eigenvalues of large random matrices. A similar effect has been observed in other group-subgroup decompositions and merits more study[12-14].

An important, and as yet incompletely solved,problem is to be able to predict those irreducible representations of $\mathcal{O}(n)$ that cannot yield irreducible representations of $\mathcal{S}(n)$ of the form $\{2^r 1^s\}$ without requiring an explicit decomposition. A further problem is to develop a method that will directly yield the multiplicity of a given irreducible representation of the form $\{2^r 1^s\}$ without requiring a complete decomposition under $\mathcal{O}(n) \rightarrow \mathcal{S}(n)$. A key to the evaluation of such decompositions is the evaluation of so-called reduced plethysms[10,11] of the form $\langle 1\rangle \otimes \{\lambda\}$. Hints at a solution come from the observation that if (λ) is a one part partition, say (k), then increasing k in steps of unity results for a certain value of k the multiplicity coefficient of say $\langle \mu_1, \mu_2, \ldots\rangle$ and $\langle \mu_1 + 1, \mu_2, \ldots\rangle$ being equal. Thereafter the multiplicity coefficients of $\langle \mu_1 + x, \mu_2, \ldots\rangle$ are independent of x and are said to be *stabilised*. The coefficients up to the stabilisation point often form identifiable integer sequences[15]. Scharf and Thibon [16] have used such considerations to recently derive a generating function for the multiplicity coefficients that arise in $\langle 1\rangle \otimes \{\lambda\}$.

10. $\mathcal{U}(3)$ Content of the 12-particle States

The $\mathcal{U}(3)$ content of the 12-particle states comes from the decomposition of the irreducible representations of $\mathcal{S}p(6,R)$ under the group reduction $\mathcal{S}p(6,R) \rightarrow \mathcal{U}(3)$ [2,3]. Some relevant decompositions are given below for $\mathcal{U}(3)$ irreducible representations, truncated at weight 18 are given in Table 10.1.

$\langle 6;(5^2\ 4)\rangle$	$\{954\}$	$+\{85^2\}$	$+\{7^24\}$	$+\{765\}$	$+\{754\}$
	$+\{65^2\}$	$+\{5^24\}$			
$\langle 6;(64^2)\rangle$	$\{10\ 4^2\}$	$+\{954\}$	$+2\{864\}$	$+\{84^2\}$	$+\{765\}$
	$+\{754\}$	$+\{6^3\}$	$+\{6^24\}$	$+\{64^2\}$	
$\langle 6;(653)\rangle$	$\{10\ 53\}$	$+\{963\}$	$+\{954\}$	$+\{873\}$	$+2\{864\}$
	$+2\{85^2\}$	$+\{853\}$	$+\{7^24\}$	$+2\{765\}$	$+\{763\}$
	$+\{754\}$	$+\{6^24\}$	$+\{65^2\}$	$+\{653\}$	
$\langle 6;(6^2\ 2)\rangle$	$\{10\ 62\}$	$+\{963\}$	$+\{8^22\}$	$+\{873\}$	$+2\{864\}$
	$+\{862\}$	$+\{765\}$	$+\{763\}$	$+\{6^3\}$	$+\{6^24\}$
	$+\{6^22\}$				
$\langle 6;(743)\rangle$	$\{11\ 43\}$	$+\{10\ 53\}$	$+\{10\ 4^2\}$	$+2\{963\}$	$+2\{954\}$
	$+\{943\}$	$+\{873\}$	$+2\{864\}$	$+\{85^2\}$	$+\{853\}$
	$+\{84^2\}$	$+\{7^24\}$	$+\{765\}$	$+\{763\}$	$+\{754\}$
	$+\{743\}$				
$\langle 6;(752)\rangle$	$\{11\ 52\}$	$+\{10\ 62\}$	$+\{10\ 53\}$	$+2\{972\}$	$+2\{963\}$
	$+2\{954\}$	$+\{952\}$	$+2\{873\}$	$+2\{864\}$	$+\{862\}$
	$+\{85^2\}$	$+\{853\}$	$+2\{7^24\}$	$+\{7^22\}$	$+\{765\}$
	$+\{763\}$	$+\{754\}$	$+\{752\}$		
$\langle 6;(83^2)\rangle$	$\{12\ 3^2\}$	$+\{11\ 43\}$	$+2\{10\ 53\}$	$+\{10\ 3^2\}$	$+\{963\}$
	$+\{954\}$	$+\{943\}$	$+\{873\}$	$+\{85^2\}$	$+\{853\}$
	$+\{83^2\}$				
$\langle 6;(842)\rangle$	$\{12\ 42\}$	$+\{11\ 52\}$	$+\{11\ 43\}$	$+2\{10\ 62\}$	$+2\{10\ 53\}$
	$+2\{10\ 4^2\}$	$+\{10\ 42\}$	$+\{972\}$	$+2\{963\}$	$+2\{954\}$
	$+\{952\}$	$+\{943\}$	$+\{8^22\}$	$+\{873\}$	$+2\{864\}$
	$+\{862\}$	$+\{853\}$	$+\{84^2\}$	$+\{842\}$	
$\langle 6;(65^2)\rangle$	$\{85^2\}$	$+\{765\}$	$+\{65^2\}$		
$\langle 6;(6^2\ 4)\rangle$	$\{864\}$	$+\{765\}$	$+\{6^3\}$	$+\{6^24\}$	
$\langle 6;(754)\rangle$	$\{954\}$	$+\{864\}$	$+\{85^2\}$	$+\{7^24\}$	$+\{765\}$
	$+\{754\}$				
$\langle 6;(763)\rangle$	$\{963\}$	$+\{873\}$	$+\{864\}$	$+\{7^24\}$	$+\{765\}$
	$+\{763\}$				
$\langle 6;(7^2\ 2)\rangle$	$\{972\}$	$+\{873\}$	$+\{7^24\}$	$+\{7^22\}$	
$\langle 6;(84^2)\rangle$	$\{10\ 4^2\}$	$+\{954\}$	$+\{864\}$	$+\{84^2\}$	

Table 10.1 Some $\mathcal{S}p(6,R) \rightarrow \mathcal{U}(3)$ decompositions (to weight 18).

11. Orbital Angular Momentum of 12-particle States

The orbital angular momentum L of the 12-particle states follows from the decomposition of the $\mathcal{U}(3)$ irreducible representations under the restriction $\mathcal{U}(3) \rightarrow \mathcal{SO}(3)$. Considerable simplification arises by recognising that the irreducible representations of $\mathcal{U}(3)$ are irreducible under the restriction $\mathcal{U}(3) \rightarrow \mathcal{SU}(3)$ and for $\mathcal{SU}(3)$ the three part labelling partitions are equivalent to irreducible representations involving partitions into fewer than three parts, indeed

$$\{\lambda_1, \lambda_2, \lambda_3\} \equiv \{\lambda_1 - \lambda_3, \lambda_2 - \lambda_3, 0\} \tag{15}$$

Thus the decomposition of the irreducible representation $\{5^2\ 4\}$ of $\mathcal{U}(3)$ is the same as that of the $\mathcal{SU}(3)$ irreducible representation $\{1\}$. Likewise the decompositions of the $\mathcal{U}(3)$ irreducible representations $\{65^2\}$ and $\{5^2\ 4\}$ are identical with respect to reduction to the subgroup $\mathcal{SO}(3)$. Likewise irreducible representations of $\mathcal{SU}(3)$ that are contragredient to one another i.e.,

$$\{\lambda_1, \lambda_2, \lambda_3\} \quad \text{and} \quad \{\lambda_1 - \lambda_3, \lambda_1 - \lambda_2, 0\} \tag{16}$$

have equivalent decompositions with respect to reduction to the subgroup $\mathcal{SO}(3)$. Some relevant $\mathcal{SU}(3) \rightarrow \mathcal{SO}(3)$ decompositions are given in Table 11.1.

$L=$	0	1	2	3	4	5	6	7	8	9
$\{0\}$	1									
$\{1\}$		1								
$\{2\}$	1		1							
$\{21\}$		1	1							
$\{3\}$		1		1						
$\{31\}$		1	1	1						
$\{4\}$	1		1		1					
$\{41\}$		1	1	1	1					
$\{42\}$	1		2	1	1					
$\{5\}$		1		1		1				
$\{51\}$		1	1	1	1	1				
$\{52\}$		1	1	2	1	1				
$\{6\}$	1		1		1		1			
$\{61\}$		1	1	1	1	1	1			
$\{62\}$	1		2	1	2	1	1			
$\{63\}$		1	1	2	2	1	1			
$\{7\}$		1		1		1		1		
$\{71\}$		1	1	1	1	1	1	1		
$\{72\}$		1	1	2	1	2	1	1		
$\{73\}$		1	1	2	2	2	1	1		
$\{8\}$	1		1		1		1		1	
$\{81\}$		1	1	1	1	1	1	1	1	
$\{82\}$	1		2	1	2	1	2	1	1	
$\{83\}$		1	1	2	2	2	2	1	1	
$\{84\}$	1		2	1	3	2	2	1	1	
$\{93\}$		1	1	2	2	2	2	2	1	1

Table 11.1 Some relevant $\mathcal{SU}(3) \rightarrow \mathcal{SO}(3)$ decompositions.

12. Labelling of the Even Parity 12-particle States

In the preceding I have outlined how one can systematically label the even parity 12-particle states using the group chain

$$\mathcal{S}p(72,R) \supset \mathcal{S}p(6,R) \times \mathcal{O}(12) \supset \mathcal{U}(3) \times \mathcal{S}(12) \supset \mathcal{SO}(3) \times \mathcal{S}(12) \tag{17}$$

The last segment of the chain, $\mathcal{SO}(3) \times \mathcal{S}(12)$, yields the traditional orbital, L, and spin, S, quantum numbers. Specific 12-particle even parity states can be systematically designated by the sequence of irreducible representations associated with the sequence of groups $\mathcal{S}p(72,R)\,\mathcal{S}p(6,R)\,\mathcal{U}(3)\,\mathcal{S}(12)^S\,\mathcal{SO}(3)^L$ leading to the notation

$$|\langle s;(0)\rangle, \langle 6;(\lambda)\rangle \alpha\{\lambda\}\beta_S\gamma_L\ ^{2S+1}L\rangle \tag{18}$$

where α, β_S, γ_L stand for any other numbers required to distinguish the various reduction multiplicities. Usually we will suppress the irreducible representation of $\mathcal{S}p(72,R)$. Using the customary spectroscopic notation for the orbital angular momentum L and the spin multiplicity $2S+1$ as a superscript we may designate the lowest energy even parity 12-particle states of the configuration $\{0\}^2\{1\}^6\{2\}^4$ as shown in Table 12.1.

$\|\langle 6;(5^2\ 4)\rangle\{5^2\ 4\}\ ^3P\rangle$	$\|\langle 6;(64^2)\rangle\{64^2\}\ ^1SD\rangle$	$\|\langle 6;(653)\rangle\{653\}\ ^{5,3}PDF\rangle$
$\|\langle 6;(6^2\ 2)\rangle\{6^2\ 2\}\ ^1SDG\rangle$	$\|\langle 6;(743)\rangle\{743\}\ ^{3,1}PDFG\rangle$	$\|\langle 6;(752)\rangle\{752\}\ ^3PDF_2GH\rangle$
$\|\langle 6;(83^2)\rangle\{83^2\}\ ^3PFH\rangle$	$\|\langle 6;(842)\rangle\{842\}\rangle\ ^1SD_2FG_2HI\rangle$	

Table 12.1 States of the 12-particle configuration $\{0\}^2\{1\}^6\{2\}^4$.

The entries in Table 12.1 may be compared with those given in Table 6.1. Each of the entries in Table 12.1 represent the lowest energy terms of an infinite tower of states with each floor of the tower increasing in energy by $2\hbar\omega$. Each floor of the tower involves several $\mathcal{U}(3)$ multiplets all labelled by partitions of the same weight. Thus in our example the first floor involves $\mathcal{U}(3)$ multiplets of weight 14, those of the second floor weight 16 and so on. Thus all the $\mathcal{U}(3)$ multiplets appearing in Table 6.1 occur on the ground floor of the tower while those in Tables 7.1 to 7.5 occur on the second floor etc. Each floor can involve various values of S and L. All the states associated with a given $\mathcal{S}p(6,R)$ irreducible representation $\langle 6;(\lambda)\rangle$ start from the floor involving partitions of weight w_λ and contribute just the $\mathcal{U}(3)$ multiplet $\{\lambda\}$ to that floor. Going to the next floor can result in the $\mathcal{S}p(6,R)$ irreducible representation contributing several different $\mathcal{U}(3)$ irreducible representations as can be seen from Table 10.1. These $\mathcal{U}(3)$ multiplets will all involve the same spin structure but may involve differing orbital angular momenta as may be seen in the examples shown in Table 12.2.

$\lvert\langle 6;(5^2\ 4)\rangle$	$\{5^2\ 4\}\ ^3P\rangle$	$\lvert\langle 6;(64^2)\rangle$	$\{64^2\}\ ^1SD\rangle$	$\lvert\langle 6;(653)\rangle$	$\{653\}\ ^{5,3}PDF\rangle$
	$\{65^2\}\ ^3P\rangle$		$\{6^2\ 4\}\ ^1SD\rangle$		$\{65^2\}\ ^{5,3}P\rangle$
	$\{754\}\ ^3PDF\rangle$		$\{754\}\ ^1PDF\rangle$		$\{6^2\ 4\}\ ^{5,3}SD\rangle$
			$\{84^2\}\ ^1PDF\rangle$		$\{763\}\ ^{5,3}PDFG\rangle$
					$\{853\}\ ^{5,3}PDF_2GH\rangle$

Table 12.2 Examples of some weight 14 and 16 states.

The odd parity states appear on floors interspacing those of the even parity states. Again successive odd parity floors involve an increase in energy of $2\hbar\omega$. As we ascend the infinite tower we find they become increasingly densely packed with $\mathcal{U}(3)$ multiplets associated with various spins. Each $\mathcal{U}(3)$ multiplet $\{\lambda\}$ appearing on the first floor is the first member of an infinite column of $\mathcal{U}(3)$ multiplets arising from the $\mathcal{S}p(6,R) \rightarrow \mathcal{U}(3)$ reduction of the $\mathcal{S}p(6,R)$ irreducible representation $\langle 6;(\lambda)\rangle$. These columns penetrate each of the successive floors of the same parity. Thus on each floor there will be $\mathcal{U}(3)$ multiplets originating from irreducible representations of $\mathcal{S}p(6,R)$ that started from lower floors, other $\mathcal{U}(3)$ multiplets will be associated with $\mathcal{S}p(6,R)$ irreps that start from that floor (See Fig. 3). Not surprisingly we have infinite sets of infinite dimensional irreducible representations $\langle 6;(\lambda)\rangle$ each starting from the floor whose zero-order energy is $w_\lambda\hbar\omega$.

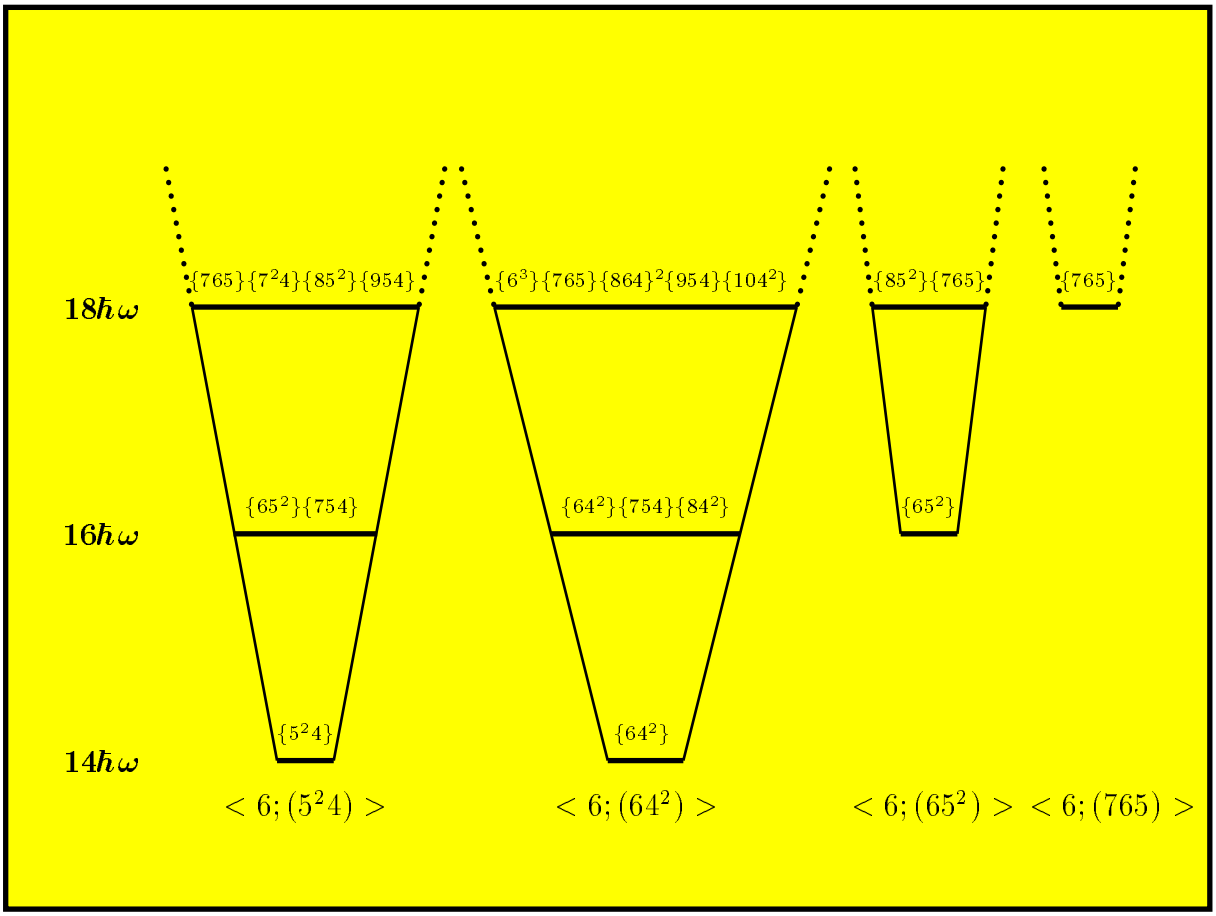


Fig. 3: Some of the infinite $\mathcal{S}p(6,R)$ multiplets for 12 particles showing the $\mathcal{U}(3)$ multiplets for the lowest three zero-order energy levels.

13. An Example

The self-consistency of the picture just outlined can be seen in the following example. We note from Tables 7.1 to 7.5 that the second floor contains the $\mathcal{U}(3)$ irreducible representation $\{853\}$. simply counting the entries in those tables shows that this $\mathcal{U}(3)$ irreducible representation occurs with the spins according to Table 13.1

$S=$	0	1	2	3
	35	63	29	4

Table 13.1 The number of times the $\mathcal{U}(3)$ irreducible representation $\{853\}$ occurs for the four allowed spin values.

At first sight it is tempting to associate all the above entries with the decomposition of the $\mathcal{O}(12)$ irreducible representation into those of $\mathcal{S}(12)$ and thence with the irreducible representation $\langle 6;(853)\rangle$ of $\mathcal{S}p(6,R)$. However, inspection of Table 9.1 shows that the spin content of the [853] irreducible representation of $\mathcal{O}(12)$ produces slightly fewer entries than in Table 14.1. Where have the extra irreducible representations $\{853\}$ of $\mathcal{U}(3)$ come from? The answer is clear if we inspect the entries in Table 10.1 and see that the weight 14 $\mathcal{S}p(6,R)$ irreducible representation can produce weight 16 irreducible representations

of $\mathcal{U}(3)$. Thus the $\mathcal{S}p(6,R)$ irreducible representations

$$\langle 6;(653)\rangle, \quad \langle 6;(743)\rangle, \quad \langle 6;(752)\rangle, \quad \langle 6;(83^2)\rangle, \quad \langle 6;(842)\rangle \tag{19}$$

Inspection of Table 9.1 show that these give precisely the right number of spin multiplicities which when added to those coming from the $\langle 6;(853)\rangle \times [853]$ irreducible representation of $\mathcal{S}p(6,R)\times\mathcal{O}(12)$ reproduce the entries in Table 14.1 which demonstrates the full self-consistency of the non-compact group approach.

14. The Next Steps

In the preceding pages I have outlined how one can consistently establish a non-compact group description of the states of $n-$non-interacting fermions in an isotropic three-dimensional $\mathcal{HO}$. This part of the theory now appears to be fairly complete. The major remaining computational problem is associated with the rapid determination of the $\mathcal{O}(n) \rightarrow \mathcal{S}(n)$ decompositions. Significant progress has been made on this problem and further substantive progress can be expected.

The next step is to investigate model Hamiltonians constructed from polynomials in the group generators[]. A trivial example would be the introduction of a term proportional to $S(S+1)$ which would immediately separate terms according to their spins. If the term is positive then states of lowest spin would lie lowest as indeed the case for many-electron quantum dots[5]. The complete dynamical group $\mathcal{M}p(6n)$ has such a rich subgroup structure and its exploration has hardly begun. This is not surprising as the understanding of the properties of non-compact groups has been a comparatively recent development. In recent years there has been considerable progress in the systematic calculation of the matrix elements of non-compact group generators, a prequisite to undertaking detailed calculations[2,17].

While our discussion has been throughout devoted to three-dimensional systems there is no difficulty in increasing or decreasing the dimension of the system being considered.

Acknowledgements

This work has been assisted by Polish KBN Grant 18/p3/94/07. Much of the work reported herein was done while we were a guests of the Max Planck Astrophysik Institüt in Garching bei München. Part of this work forms the subject of a Master's thesis (KG).

All calculations were done using the C-package SCHUR*

* B. G. Wybourne, **SCHUR** is an interactive C package for calculating properties of Lie groups and symmetric functions. Distributed by: S. Christensen, P. O. Box 16175, Chapel Hill, NC 27516 USA. e-mail: steve@scm.vnet.net . A detailed description can be seen by WEB users at http://scm.vnet.net/Christensen.html

REFERENCES

[1] Wybourne, B. G. : *Classical Groups for Physicists*, Wiley-Interscience, New York 1974.

[2] Rowe, D. J., Wybourne, B. G., Butler, P. H.: *J. Phys. A: Math. Gen.* **18**, 939-53 (1985).

[3] King, R. C. , Wybourne, B. G. : *J. Phys. A: Math. Gen.* **18**, 3113-39 (1985).

[4] Haase, R. W. , Johnson, N. J. : *J. Phys. A: Math. Gen.* **26**, 1663-72 (1993).

[5] Haase, R. W. , Johnson, N. J. : *Phys. Rev. B* **48**,1583-94 (1993).

[6] Wybourne, B. G., : *J. Phys. A: Math. Gen.* **25**, 4389-98 (1992).

[7] Wybourne, B. G., : *Rept. Math. Phys.* **34**, 9-16 (1994).

[8] Grudziński, K. , Wybourne, B. G. : in *Symmetry and Structural Properties of Condensed Matter*, Ed. T. Lulek, W. Florek & S. Wałcerz, World Scientific, Singapore pp 469-83 (1995).

[9] Black, G. R. E., King, R. C., Wybourne, B. G. : *J. Phys. A: Math. Gen.* **16**, 1555-89 (1983).

[10] Salam, M. A., Wybourne, B. G. : *J. Phys. A: Math. Gen.* **22**, 3771-8 (1989).

[11] Scharf, T., Thibon, J-Y, Wybourne, B. G. : *J. Phys. A: Math. Gen.* **26**, 7461-78 (1993).

[12] Wybourne, B. G., : *J. Math. Phys.* **10**, 467-71 (1969),

[13] Cleary, J. G., Wybourne, B. G., : *J. Math. Phys.* **12**, 45-52 (1971),

[14] Hirst, M. G., Wybourne, B. G., :*J. Phys. A: Math. Gen.* **19**, 1545-9 (1986).

[15] Sloane, N. A., Plouffe, S. , : *The Encyclopaedia of Integer Sequences*, Academic Press, New York 1995.

[16] Scharf, T., Thibon, J-Y, (Private communication, September 1995).

[17] Rowe, D. J., : *Rept. Prog. Phys.* **48**, 1419-80 (1985).

Jumping succession rules and their generating functions

Luca Ferrari* Elisa Pergola† Renzo Pinzani†
Simone Rinaldi†

Abstract

We study a generalization of the concept of succession rule, called *jumping succession rule*, where each label is allowed to produce its sons at different levels, according to the production of a fixed succession rule. By means of suitable linear algebraic methods, we obtain simple closed forms for the numerical sequences determined by such rules and give applications concerning classical combinatorial structures. Some open problems are proposed at the end of the paper.

1 Doubled succession rules

Consider a $2 \times n$ rectangle and suppose to tile it using 1×2 domino pieces. Clearly, if one uses vertical pieces only in the tiling, there is exactly one solution to the problem, whereas allowing vertical and horizontal pieces gives F_n possible solutions, where F_n is the n-th Fibonacci number, as it is well-known. These two, very simple enumerative results are clearly related, and it seems obvious that the latter can be derived from the former one, which is completely trivial. Our aim is to develop a general setting to deal with this kind of problems by slightly extending the concept of succession rule and the ECO method.

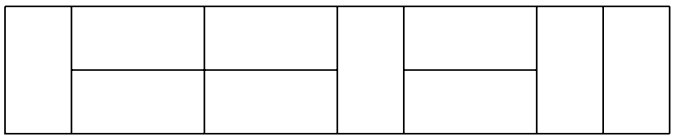


Figure 1: The tiling of a $2 \times n$ rectangle using Fibonacci pieces.

*Dipartimento di Matematica "U. Dini", Viale Morgagni 67/A, 50134 Firenze, Italy ferrari@math.unifi.it

†Dipartimento di Sistemi e Informatica, Via Lombroso 6/17, 50134 Firenze, Italy {elisa,pinzani,rinaldi}@dsi.unifi.it

A *succession rule* Ω is a system constituted by an *axiom* (a), $a \in \mathbb{N}^+ = \mathbb{N} \setminus \{0\}$, and a set of *productions* of the form:

$$(k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)), \qquad k \in M \subset \mathbb{N}^+,$$

where $e_i : \mathbb{N}^+ \to \mathbb{N}^+$, explaining how to derive, for any given label (k), $k \in \mathbb{N}^+$, its *successors* $(e_1(h)), (e_2(h)), \dots, (e_k(h))$. In most of the cases for a succession rule Ω, we use the more compact notation:

$$\begin{cases} (a) \\ (k) \rightsquigarrow (e_1(k))(e_2(k)) \dots (e_k(k)), \end{cases} \tag{1}$$

to mean that there can be infinitely many productions in the system, but at most one for each integer $k \in \mathbb{N}^+$.

The rule Ω can be represented by means of a *generating tree*, that is a rooted tree whose vertices are the labels of Ω; (a) is the label of the root and each node labelled (k) produces k sons labelled $(e_1(k)), \dots, (e_k(k))$, respectively. We refer to [2] for further details and examples. A succession rule Ω defines a sequence of positive integers $\{f_n\}_{n \geq 0}$, being f_n the number of the nodes at level n in the generating tree defined by Ω. By convention the root is at level 0, so $f_0 = 1$. The function $f_\Omega(x) = \sum_{n \geq 0} f_n x^n$ is the *generating function* derived from Ω.

The concept of succession rules was first introduced in [3] by Chung et al. to study reduced Baxter permutations; later, West applied succession rules to the enumeration of permutations with forbidden subsequences [16]. Moreover, they are a fundamental tool used by the ECO method [2], which is a general method for the enumeration of combinatorial objects essentially based on the definition of a recursive construction for a class of objects by means of an operator which performs a "local expansion" on the objects themselves. Let p be a *discriminating parameter* on a class of objects $\mathcal{O}$, that is $p : \mathcal{O} \to \mathbb{N}^+$, such that $|\mathcal{O}_n| = |\{O \in \mathcal{O} : p(O) = n\}|$ is finite. An operator ϑ on the class $\mathcal{O}$ is a function from $\mathcal{O}_n$ to $2^{\mathcal{O}_{n+1}}$, where $2^{\mathcal{O}_{n+1}}$ is the power set of $\mathcal{O}_{n+1}$.

Proposition 1.1 *[2] Let ϑ be an operator on $\mathcal{O}$. If ϑ satisfies the following conditions:*

1. *for each $O' \in \mathcal{O}_{n+1}$, there exists $O \in \mathcal{O}_n$ such that $O' \in \vartheta(O)$,*
2. *for each $O, O' \in \mathcal{O}_n$ with $O \neq O'$, $\vartheta(O) \cap \vartheta(O') = \emptyset$,*

then the family of sets $\mathcal{F}_{n+1} = \{\vartheta(O) : O \in \mathcal{O}_n\}$ is a partition of $\mathcal{O}_{n+1}$.

Once the parameter p is fixed, if we are able to define an operator ϑ which satisfies conditions 1. and 2., then Proposition 1.1 allows us to construct each object $O' \in \mathcal{O}_{n+1}$ from an object $O \in \mathcal{O}_n$, and each object $O' \in \mathcal{O}_{n+1}$ is obtained from exactly one $O \in \mathcal{O}_n$.

The generating tree associated to the couple $(\mathcal{O}, \vartheta)$, is a rooted tree whose vertices are the objects of $\mathcal{O}$. The objects having the same value of the parameter p lie at the same level, and the sons of an object are the objects it produces through ϑ.

A slight generalization of the notion of succession rule is provided by the concept of *coloured succession rules*. Roughly speaking, a rule is said to be coloured when more than one production is allowed for at least one label. The usual notation to indicate a two-coloured rule is the following:

$$\begin{cases} (a) \\ (k) \rightsquigarrow (e_1(k)) \dots (e_t(k))(\overline{e_{t+1}(k)}) \dots (\overline{e_k(k)}), \\ (\overline{k}) \rightsquigarrow (c_1(k)) \dots (c_s(k))(\overline{c_{s+1}(k)}) \dots (\overline{c_k(k)}), \end{cases} \tag{2}$$

For more details about these topics, see [7].

Given a succession rule of the form (1), we define the *rule operator* L_Ω (briefly, L) associated with Ω [7, 8] as:

$$L_\Omega : \mathbb{R}[x] \to \mathbb{R}[x]$$

$$L_\Omega(\mathbf{1}) = x^a;$$

$$L_\Omega(x^k) = x^{e_1(k)} + \ldots + x^{e_k(k)};$$

$$L_\Omega(x^k) = kx^k, \qquad \text{if the label } (k) \text{ is not in the generating tree of } \Omega\ ,$$

and then extending by linearity on $\mathbb{R}[x]$ (considered as a $\mathbb{R}$-vector space). In general, we use the power notation to express the iterated application of L: $L^{n+1}(\mathbf{1}) = L(L^n(\mathbf{1}))$. For any $n \in \mathbb{N}$ we have:

$$f_n = [L^{n+1}(\mathbf{1})]_{x=1} = [DL^n(\mathbf{1})]_{x=1},$$

where D is the derivative operator with respect to the variable x. In [7, 8] many properties of the rule operators are given.

The next definition is the key step in our extension of ECO method.

Given a succession rule Ω as in (1), we call *doubled succession rule* associated with Ω the following expression:

$$\Omega' : \begin{cases} (2a) \\ (2k) \overset{1}{\rightsquigarrow} (2e_1(k)) \dots (2e_k(k)) \\ (2k) \overset{2}{\rightsquigarrow} (2e_1(k)) \dots (2e_k(k)). \end{cases} \tag{3}$$

In order to understand the meaning of this definition we introduce the concept of generating tree associated with Ω', or *doubled generating tree*: it is precisely a rooted labelled tree whose edges can have "length" 1 or 2. The *lengthened level* (briefly, level) of a node N in a doubled generating tree is then defined as follows:

i) if N is the root, then its level is equal to 0;

ii) otherwise, let F be the father of N; in this case, the level of N is equal to the level of F plus the length of the edge from F to N.

In a word, the level of a node N is the sum of the lengths of the edges connecting the root to N. The root of the doubled generating tree is labelled $(2a)$ and every node at level l (labelled $(2k)$) has exactly k sons at level $l+1$ (labelled $(2e_1(k), \dots, (2e_k(k))$, resp.) and k sons at level $l+2$ (with the same labels). We remark that a similar notion has been used in [9, 10]. Anyway, these works deal with specific examples only, without providing a general theory for doubled rules.

At this stage, it is not difficult to see that our starting problem fits into this framework. Indeed, given the (unique) "vertical" tiling of the $2 \times n$ rectangle, we obtain the (unique) "vertical" tiling of the $2 \times (n+1)$ rectangle simply by adding a vertical domino piece on the right; this can be trivially described by the succession rule:

$$\Omega : \begin{cases} (1) \\ (1) \rightsquigarrow (1). \end{cases} \tag{4}$$

On the other hand, if we consider a generic tiling of the $2 \times n$ rectangle by vertical and horizontal dominoes, we can add on the right one vertical domino (so obtaining a tiling for the $2 \times (n+1)$ rectangle) or two horizontal dominoes (in this way obtaining a tiling for the $2 \times (n+2)$ rectangle). Because of the simplicity of this example, it is very easy to show that every tiling of the $2 \times (n+1)$ rectangle derives from exactly one tiling (either of the $2 \times n$ rectangle or of the $2 \times (n-1)$ rectangle). This construction can be described by doubling the succession rule Ω, so obtaining the rule:

$$\Omega' : \begin{cases} (2) \\ (2) \overset{1}{\rightsquigarrow} (2) \\ (2) \overset{2}{\rightsquigarrow} (2). \end{cases} \tag{5}$$

The first levels of its generating tree are represented in Figure 2:

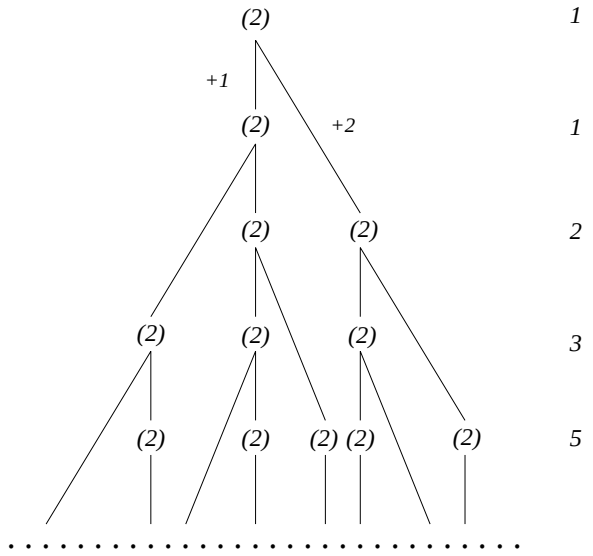


Figure 2: The first levels of the generating tree of the doubled rule (5).

It is immediate to see that the sequence enumerated by the above doubled generating tree is that of Fibonacci numbers: indeed, the number of nodes at each level is the sum of the cardinalities of the two preceding levels.

2 Fibonacci transform

Consider a succession rule Ω of the form (1), and suppose that $(s_n)_{n\geq 0}$ is the numerical sequence determined by Ω. If Ω' is the doubled succession rule associated with Ω, can we determine the sequence $(s'_n)_{n\geq 0}$ related to Ω'? The central result of this section is precisely the solution of this problem.

Before proving our main theorem, we need to state a few definitions. Let L be the rule operator associated with Ω; the series:

$$\sum_{n\geq 0} L^{n+1}(\mathbf{1})t^n \tag{6}$$

is a formal power series in the variables x and t and it is called the *bivariate generating function* of the generating tree determined by Ω. In particular, the sequence of the numbers $[L^{n+1}(\mathbf{1})]_{x=1}$ is precisely the one defined by Ω, and the coefficient of x^k in the polynomial $L^{n+1}(\mathbf{1})$ represents the number of nodes labelled (k) at level n.

If Ω' is the doubled rule associated with Ω, the *normalization* of Ω' is the rule:

$$\widetilde{\Omega}' : \begin{cases} (a) \\ (k) \overset{1}{\rightsquigarrow} (e_1(k)) \dots (e_k(k)) \\ (k) \overset{2}{\rightsquigarrow} (e_1(k)) \dots (e_k(k)) \end{cases} \tag{7}$$

which is obtained by Ω' simply by dividing each label by 2. It is clear that the generating tree defined by $\widetilde{\Omega}'$ loses the "ECO-property", i.e. every node labelled (k) possesses $2k$ sons instead of k; however, Ω' and $\widetilde{\Omega}'$ count the same sequence, and $\widetilde{\Omega}'$ can be better treated in the formalism of rule operators. We remark that systems like $\widetilde{\Omega}'$ are also called *pseudo ECO-systems* [6].

Proposition 2.1 *The bivariate generating function of the generating tree defined by $\widetilde{\Omega}'$ has the form:*

$$\left(\frac{1}{1-tL-t^2L}\right)(L(\mathbf{1})) = \sum_{n\geq 0}(tL+t^2L)^n(L(\mathbf{1})), \tag{8}$$

being $\frac{1}{M}$ the compositional inverse of the operator M.

Proof. Denote by $p_n(x)$ the polynomial such that the coefficient of x^k is the number of nodes labelled (k) at level n of the generating tree of $\widetilde{\Omega}'$. Clearly $p_0(x) = x^a$, $p_1(x) = x^{e_1(a)} + \dots + x^{e_a(a)}$ and, in general, $p_n(x) = L^{n+1}(\mathbf{1})$. Now observe that a node at level n is the son of a node at level $n-1$ or of a node at level $n-2$. Then the following polynomial recurrence holds:

$$p_n(x) = L(p_{n-1}(x)) + L(p_{n-2}(x)). \tag{9}$$

which is valid for every $n \geq 1$ (by defining $p_{-1}(x) = 0$).

According to (9), the generating function $f(x,t) = \sum_{n\geq 0} p_n(x)t^n$ satisfies:

$$f(x,t) = \sum_{n\geq 1} L(p_{n-1}(x))t^n + \sum_{n\geq 2} L(p_{n-2})(x)t^n + L(\mathbf{1})$$

which simplifies into:

$$f(x,t) = (tL + t^2L)(f(x,t)) + L(\mathbf{1})$$

that is

$$(1 - tL - t^2L)(f(x,t)) = L(\mathbf{1}).$$

Therefore $f(x,t)$ is obtained by simply inverting the operator $1 - tL - t^2L$, which is precisely our thesis. □

Theorem 2.1 *The number sequence enumerated by $\widetilde{\Omega'}$ (or by Ω') is the sequence:*

$$s'_n = \sum_{k=0}^{n} \binom{n-k}{k} s_{n-k} = \sum_{k=0}^{n} \binom{k}{n-k} s_k \tag{10}$$

being $(s_n)_{n\geq 0}$ the sequence determined by Ω.

Proof. From Proposition 2.1 we have:

$$s'_n = [[t^n]f(x,t)]_{x=1} = \left[[t^n] \sum_{m\geq 0} (tL + t^2L)^m (L(\mathbf{1})) \right]_{x=1}.$$

Since

$$(tL + t^2L)^m = t^m(1+t)^m L^m = \sum_{k=0}^{m} \binom{m}{k} t^{m+k} L^m,$$

we obtain:

$$[t^n](tL + t^2L)^m = \sum_{k=0}^{n} \binom{n-k}{k} L^{n-k},$$

whence:

$$s'_n = \left[\sum_{k=0}^{n} \binom{n-k}{k} L^{n-k+1}(\mathbf{1})\right]_{x=1} = \sum_{k=0}^{n} \binom{n-k}{k} s_{n-k}. \quad \square$$

The numbers s'_n of (10) count the nodes at level n of the generating tree of Ω'. From a combinatorial view point, each term $\binom{n-k}{k} s_{n-k}$ of the sum in (10) counts the number of the nodes N at level n such that the path from the root to N contains exactly $n-k$ edges of length 2.

We call *Fibonacci transform* of a numerical sequence $(s_n)_{n\geq 0}$ the sequence:

$$s'_n = \sum_{k=0}^{n} \binom{n-k}{k} s_{n-k}. \tag{11}$$

The reason for choosing this name lies in the following

Corollary 2.1 *(Lucas' identity) The Fibonacci transform of the sequence $s_n = 1$, $\forall n \in \mathbb{N}$, is the sequence of Fibonacci numbers.*

Observe that this corollary is also the solution of our starting problem.

We now consider an extension of the ECO method which represents the combinatorial interpretation of doubled succession rules. Let $\mathcal{O}$ be a class of combinatorial objects. A *doubled operator* ϑ is an operator on the class $\mathcal{O}$:

$$\vartheta : \mathcal{O}_n \to 2^{\mathcal{O}_{n+1} \cup \mathcal{O}_{n+2}}.$$

Proposition 2.2 *Let ϑ be a doubled operator on $\mathcal{O}$. If ϑ satisfies the following conditions:*

1. *for each $O' \in \mathcal{O}_n$, there exists $O \in \mathcal{O}_{n-2} \cup \mathcal{O}_{n-1}$ such that $O' \in \vartheta(O)$,*
2. *for each $O, O' \in \mathcal{O}_n \cup \mathcal{O}_{n+1}$ with $O \neq O'$, $\vartheta(O) \cap \vartheta(O') = \emptyset$,*

then the family of sets $\mathcal{F}_{n+2} = \{\vartheta(O) : O \in \mathcal{O}_n \cup \mathcal{O}_{n+1}\} \cap 2^{\mathcal{O}_{n+2}}$ is a partition of $\mathcal{O}_{n+2}$.

Clearly, the generating tree associated to the operator ϑ is a doubled generating tree.

Example 2.1 *Doubled Dyck paths and a combinatorial interpretation of a doubled succession rule.*

On the lattice plane $\mathbb{N}\times\mathbb{N}$, the class $\mathcal{C}$ of *Dyck paths* contains the paths made up of *rise* steps $(1,1)$ and *fall* steps $(1,-1)$, running from $(0,0)$ to $(2n,0)$ (see Fig. 3 (a)). The length of a Dyck path is the number of its steps. It is common knowledge that the number of $2n-$length Dyck paths is the nth *Catalan number* $C_n = \frac{1}{n+1}\binom{2n}{n}$ (for an interesting survey, see [5]).

The last sequence of fall steps in a Dyck path is called its last descent. Let $\mathcal{C}_n$ be the set of Dyck paths having length $2n$, and ϑ the operator defined in [2] such that

$$\vartheta : \mathcal{C}_n \to 2^{\mathcal{C}_{n+1}},$$

which inserts a peak into any point belonging to the last descent of each path.

The succession rule Ω describing this operator on $\mathcal{C}$ is:

$$\Omega : \begin{cases} (1) \\ (h) \rightsquigarrow (2)(3)\dots(h)(h+1). \end{cases} \tag{12}$$

Let us consider the class $\mathcal{CC}$ of lattice paths made up by rise $(1,1)$, fall $(1,-1)$, *double-rise* $(2,2)$ and *double-fall* $(2,-2)$ steps, defined recursively as follows:

i) the empty path belongs to $\mathcal{CC}$;

ii) if C, D are paths in $\mathcal{CC}$, then the path obtained by adding a rise step (resp. a double-rise step) before C and a fall step (resp. a double-fall step) after C and then concatenating with D belongs to $\mathcal{CC}$.

We call these paths *doubled Dyck paths* (see Fig. 3, (b)). In a doubled Dyck path the *last descent* is the last sequence of fall/double-fall steps, and a *peak* (resp. *double peak*) is a rise (resp. double-rise) step followed by a fall (resp. double-fall) step.

The class of doubled Dyck paths is suitably introduced, starting from the class of Dyck paths, with the aim of handling a combinatorial structure whose recursive construction can be defined by means of a doubled operator. Indeed, let us consider the doubled operator ϑ' on $\mathcal{CC}$; if $\mathcal{CC}_n$ denotes the set of paths having length $2n$, then:

$$\vartheta' : \mathcal{CC}_n \to 2^{\mathcal{CC}_{n+1}} \cup 2^{\mathcal{CC}_{n+2}}.$$

The operator ϑ' inserts a peak, or a doubled peak, in each lattice point of the last descent of a doubled Dyck path, clearly excluding those points internal to double-fall steps (see Fig. 4).

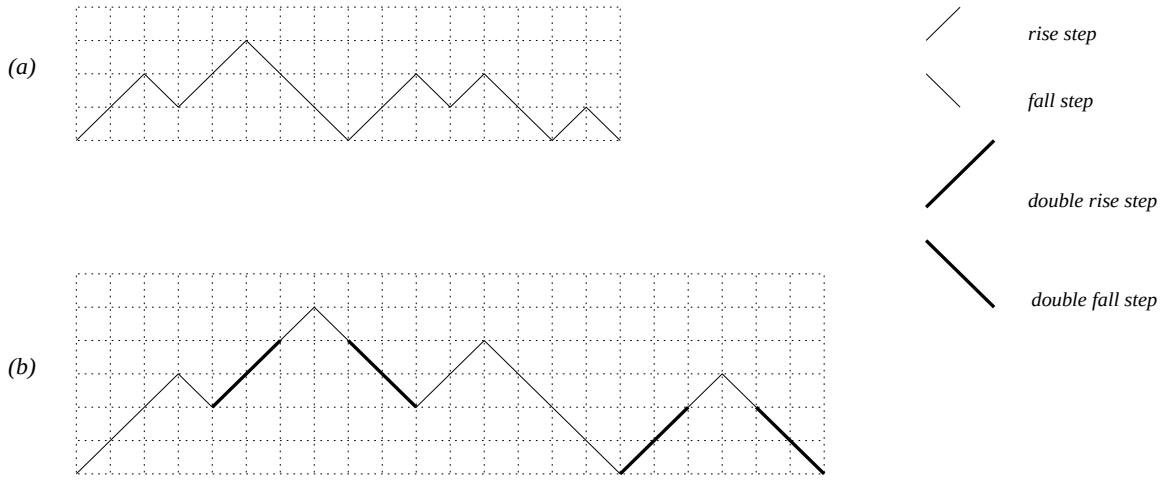


Figure 3: A Dyck path and a doubled Dyck path.

The operator ϑ' satisfies Proposition 2.2, and the doubled generating tree associated with ϑ' (see Fig. 5) determines a doubled succession rule Ω', which is the Fibonacci transform of Ω:

$$\Omega' : \begin{cases} (2) \\ (2h) \overset{1}{\rightsquigarrow} (4)(6)\dots(2h)(2h+2) \\ (2h) \overset{2}{\rightsquigarrow} (4)(6)\dots(2h)(2h+2). \end{cases} \tag{13}$$

Let us have a look at the enumeration of the class $\mathcal{CC}$ according to the path length. Theorem 2.1 ensures us that the number of doubled Dyck paths having length $2n$ is equal to

$$C'_n = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n-k}{k} C_{n-k} = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{k}{n-k} C_k. \tag{14}$$

Equality (14) has a very simple combinatorial interpretation: for any fixed length $2n$, for any $k = 0, \dots, \lfloor \frac{n}{2} \rfloor$, there are exactly $\binom{n-k}{k} C_{n-k}$ paths of $\mathcal{CC}_n$ having k doubled rise step.

Doubled Dyck paths can be represented as *doubled Dyck words*, defined by the unambiguous grammar:

$$S \to xS\overline{x}S|yyS\overline{yy}S|\epsilon,$$

being ϵ the empty word. The generating function is

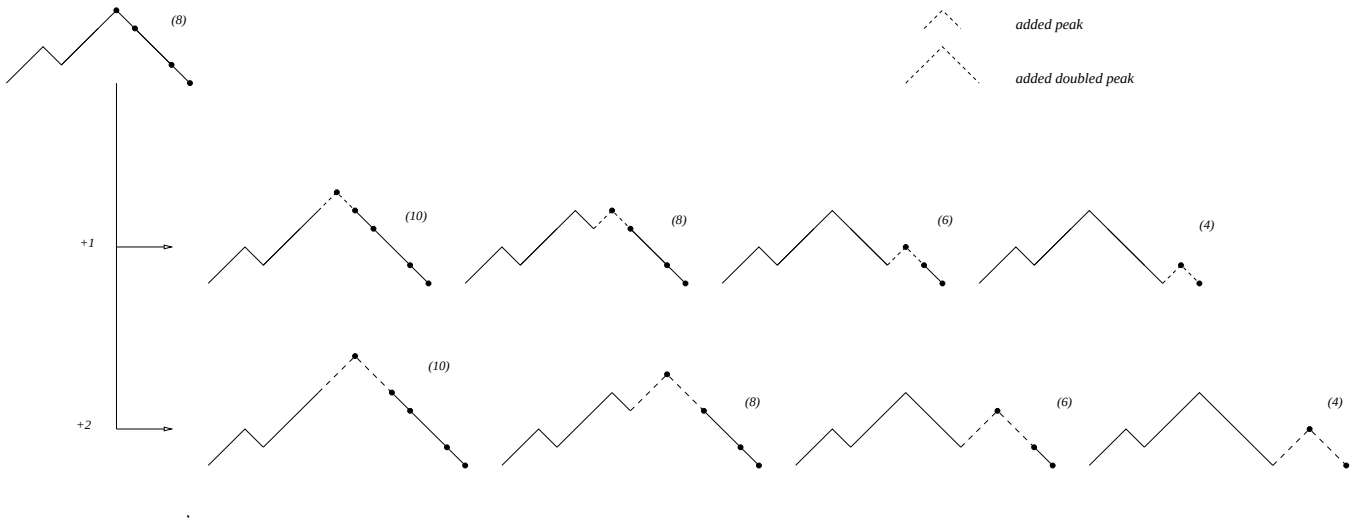


Figure 4: The doubled operator ϑ' on a doubled Dyck path. The marked points denote the sites where the operator performs the transformation.

$$\frac{1-\sqrt{1-4(x^2+x^4)}}{2(x^2+x^4)}$$

defining the numerical sequence $1, 1, 3, 9, 31, 113, 431, 1697, \ldots$, omitting the zeroes (sequence A052709 in [14]).

At the end of this section, we give a result which characterizes the set of generating functions of doubled succession rules. Recall that two rules are said to be *equivalent* when they define the same sequence.

Theorem 2.2 *Let Ω be a succession rule, and Ω' the doubled rule associated with Ω. Then a succession rule Ω'' exists such that Ω'' and Ω' are equivalent.*

Proof. We prove that, given a succession rule (1), the doubled succession rule Ω' associated with Ω, having the form (3), is equivalent to the following coloured rule:

$$\Omega'' : \begin{cases} (\overline{a}) \\ (k+1) \rightsquigarrow (e_1(k)+1)\ldots(e_k(k)+1)(\overline{k}) \\ (\overline{k}) \rightsquigarrow (e_1(k)+1)\ldots(e_k(k)+1). \end{cases} \tag{15}$$

Let L be the rule operator associated with Ω, and M the rule operator associated with Ω'':

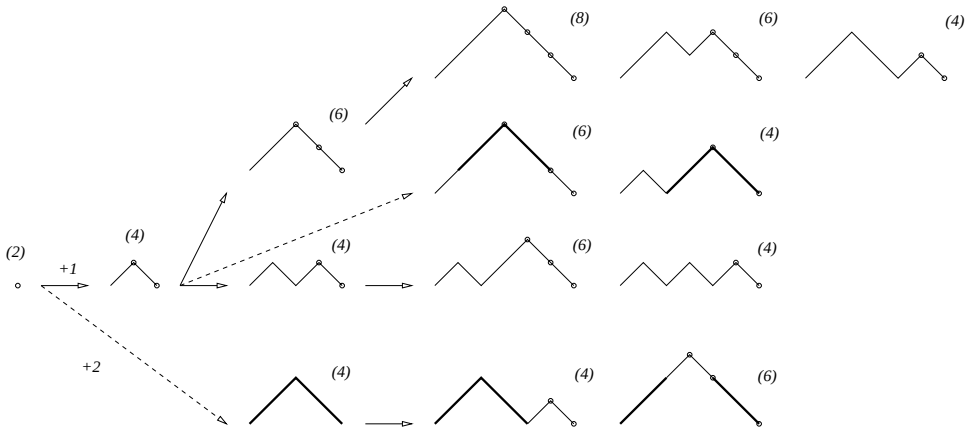


Figure 5: The first levels of the generating tree related to the doubled Catalan operator ϑ'.

$$M : x\mathbb{R}[x] \oplus \mathbb{R}[y] \longrightarrow x\mathbb{R}[x] \oplus \mathbb{R}[y]$$

$$M(\mathbf{1}) = y^a;$$

$$M(y^k) = xL(x^k);$$

$$M(x^{k+1}) = xL(x^k) + y^k.$$

The definition of the rule operator associated with a coloured succession rule can be found in [7]. It is easy to prove the following statements:

1) $M(xp(x)) = xL(p(x)) + p(x)$;

2) $M(p(y)) = xL(p(x))$;

3) $M^n(\mathbf{1}) = x\sum_{k=0}^{n-1} \binom{k-1}{n-k-1} L^k(x^a) + \sum_{k=0}^{n-2} \binom{k-1}{n-k-2} L^k(y^a)$.

As a consequence of these facts we have the desired result:

$$[M^n(\mathbf{1})]_{x=y=1} = \sum_{k=0}^{n-1} \binom{k}{n-k-1} s_k = s'_{n-1}. \quad \square$$

Simple as it is, Theorem 2.2 has a deep meaning from a theoretical viewpoint: in a word, it states that the set of generating functions of doubled succession rules is included into the set of generating functions of succession rules.

For example, we trivially obtain that the doubled rule (5) associated with the rule (4) defines Fibonacci numbers, like the rule:

$$\begin{cases} (1) \\ (1) \rightsquigarrow (2) \\ (2) \rightsquigarrow (1)(2). \end{cases}$$

Moreover, the doubled rule (13), associated with Catalan numbers, is equivalent to the following rule:

$$\begin{cases} (1) \\ (1) \rightsquigarrow (3) \\ (k+1) \rightsquigarrow (3)\dots(k+1)(\overline{k-1}) \\ (\overline{k}) \rightsquigarrow (3)\dots(k+2). \end{cases} \tag{16}$$

3 Jumping succession rules

The idea of doubling a succession rule can be slightly generalized in the following way.

Given the succession rule Ω of the form (1), and $i_1, \dots, i_m \in \mathbb{N}^+$ such that $0 < i_1 < \dots < i_m$, we call *jumping succession rule* of type $(i_1, \dots, i_m)$ associated with Ω the rule:

$$\Omega^{(i_1,\dots,i_m)} : \begin{cases} (ma) \\ (mk) \overset{i_1}{\rightsquigarrow} (me_1(k))\dots(me_k(k)) \\ \dots \\ (mk) \overset{i_m}{\rightsquigarrow} (me_1(k))\dots(me_k(k)). \end{cases} \tag{17}$$

Clearly, a doubled succession rule is a jumping rule of type $(1, 2)$. Following the same philosophy of Section 2, we define the *normalization* of $\Omega^{(i_1,\dots,i_m)}$ as:

$$\widetilde{\Omega}^{(i_1,\dots,i_m)} : \begin{cases} (a) \\ (k) \overset{i_1}{\rightsquigarrow} (e_1(k))\dots(e_k(k)) \\ \dots \\ (k) \overset{i_m}{\rightsquigarrow} (e_1(k))\dots(e_k(k)). \end{cases} \tag{18}$$

The main enumerative results concerning jumping rules can be easily proved following the ideas developed in Section 2.

Proposition 3.1 *The bivariate generating function of the generating tree defined by $\widetilde{\Omega}^{(i_1,\dots,i_m)}$ has the form:*

$$\left(\frac{1}{1-t^{i_1}L-\ldots-t^{i_m}L}\right)\left(\sum_{i=1}^{i_1-1}L^{i+1}(\mathbf{1})t^i\right)$$
$$=\sum_{n\geq 0}(t^{i_1}L+\ldots+t^{i_m}L)^n\left(\sum_{i=1}^{i_1-1}L^{i+1}(\mathbf{1})t^i\right), \tag{19}$$

being L the rule operator associated with Ω.

Theorem 3.1 *If Ω counts the sequence $(s_n)_{n\geq 0}$, then the sequence enumerated by $\Omega^{(i_1,\ldots,i_m)}$ is:*

$$s'_n=\sum_{\alpha=0}^{i_1-1}\sum_{\substack{\mu_1,\ldots,\mu_m\\ \mu_1 i_1+\ldots+\mu_m i_m=n-\alpha}}\binom{\sum_{i=1}^m\mu_i}{\mu_1,\ldots,\mu_m}s_{\left(\sum_{i=1}^m\mu_i+\alpha\right)}, \tag{20}$$

where the expression $\binom{\theta}{\theta_1,\ldots,\theta_t}$, $\theta_1+\ldots+\theta_t=\theta$, denotes the usual multinomial coefficient. We call $(s'_n)_{n\geq 0}$ the Fibonacci transform of type $i_1,\ldots,i_m$ *of $(s_n)_{n\geq 0}$.*

Remark 3.1

1. s'_n is the sum of the number of the nodes at levels $n-i_1,\ldots,n-i_m$ in the "jumping generating tree".

2. If $i_1=1$, the expression for the numbers s'_n counted by $\Omega^{(1,i_2,\ldots,i_m)}$ is a bit more readable:

$$s'_n=\sum_{\substack{\mu_1,\ldots,\mu_m\\ \mu_1+\ldots+\mu_m i_m=n}}\binom{\sum_{i=1}^m\mu_i}{\mu_1,\ldots,\mu_m}s_{\left(\sum_{i=1}^m\mu_i\right)}. \tag{21}$$

3. It is clear that this result applied to $\Omega^{(1,2)}$ coincides with the result obtained for doubled rules, since in this case:

$$s'_n=\sum_{\substack{\mu_1,\mu_2\\ \mu_1+2\mu_2=n}}\binom{\mu_1+\mu_2}{\mu_1,\mu_2}s_{\mu_1+\mu_2}$$
$$=\sum_{\mu_2=0}^{n}\binom{n-\mu_2}{\mu_2}s_{n-\mu_2}. \tag{22}$$

Example 3.1 *Tribonacci numbers.*

Let Ω be

$$\Omega : \begin{cases} (1) \\ (1) \rightsquigarrow (1), \end{cases}$$

the jumping rule $\Omega^{(1,2,3)}$ defines the well-known Tribonacci numbers having $T_0 = 1, T_1 = 1, T_2 = 2$ as initial values. By applying equality (21), we obtain the following remarkable formula:

$$\begin{aligned} T_n &= \sum_{\substack{\mu_1,\mu_2,\mu_3 \\ \mu_1+2\mu_2+3\mu_3=n}} \binom{\mu_1+\mu_2+\mu_3}{\mu_1,\mu_2,\mu_3} \\ &= \binom{n}{n,0,0} + \binom{n-1}{n-2,1,0} + \binom{n-2}{n-3,0,1} \\ &+ \binom{n-2}{n-4,2,0} + \binom{n-3}{n-5,1,1} + \binom{n-3}{n-6,3,0} \\ &+ \binom{n-4}{n-6,0,2} + \binom{n-4}{n-7,2,1} + \binom{n-5,}{n-8,1,2} + \dots \end{aligned}$$

which is the obviuos generalization to Tribonacci numbers of Lucas' identity. This equality was obtained by Shannon in [13] by a direct computation; the interest of our proof lies in the fact that it can be easily generalized to n-bonacci numbers, for every $n \in \mathbb{N}$.

3.1 Scattered succession rules and linear recurrences

We have just studied the generating tree obtained by "repeating" a succession rule Ω at various levels. A step forward could be done by allowing the repetition of Ω "more than one time" at each level.

We say that Ω' is a *scattered succession rule* associated with Ω whenever there exist positive integers $m_1, \dots, m_r, i_1, \dots, i_r$ such that $m = m_1 + \dots + m_r$ and:

$$\Omega' : \begin{cases} (ma) \\ (mk) \overset{i_1}{\rightsquigarrow} (me_1(k))^{m_1} \dots (me_k(k))^{m_1} \\ \dots \\ (mk) \overset{i_r}{\rightsquigarrow} (me_1(k))^{m_r} \dots (me_k(k))^{m_r}. \end{cases}$$

The *normalization* $\widetilde{\Omega'}$ of Ω' is defined in the usual way.

An interesting application of this definition can be obtained by considering the simple rule (4). In fact, the following proposition holds.

Proposition 3.2 *Suppose that the sequence $(a_n)_{n\geq 0}$ is defined by the linear recurrence $a_n = m_1 a_{n-1} + \ldots + m_r a_{n-r}$, $m_1, \ldots, m_r \in \mathbb{N}$ and having the initial values $a_0 = 1$, $a_1 = m_1 a_0$, $a_2 = m_1 a_1 + m_2 a_0$,..., $a_{r-1} = m_1 a_{r-2} + \ldots + m_{r-1} a_0$. Then $(a_n)_{n\geq 0}$ is the sequence determined by the scattered rule Ω' defined by:*

$$\Omega' : \begin{cases} (m) \\ (m) \overset{1}{\rightsquigarrow} (m)^{m_1} \\ \ldots \\ (m) \overset{r}{\rightsquigarrow} (m)^{m_r} \end{cases},$$

with $m = m_1 + \ldots + m_r$.

4 Exploded succession rules

Let Ω be a succession rule of the form (1) and h a positive integer. Consider the following jumping rule:

$$\Omega^{(1,2,\ldots,h)} : \begin{cases} (ha) \\ (hk) \overset{1}{\rightsquigarrow} (he_1(k)) \ldots (he_k(k)) \\ \ldots \\ (hk) \overset{h}{\rightsquigarrow} (he_1(k)) \ldots (he_k(k)). \end{cases} \tag{23}$$

Now let h tend to infinity: clearly the jumping rule $\Omega^{(1,2,\ldots,h)}$ cannot be expressed formally, whereas its normalization $\widetilde{\Omega}^{(1,2,\ldots,h)}$ can. More precisely, we can informally state that

$$\lim_{h\to\infty} \widetilde{\Omega}^{(1,2,\ldots,h)} = \widetilde{\Omega}^{\infty},$$

where

$$\widetilde{\Omega}^{\infty} : \begin{cases} (a) \\ (k) \overset{1}{\rightsquigarrow} (e_1(k)) \ldots (e_k(k)) \\ \ldots \\ (k) \overset{h}{\rightsquigarrow} (e_1(k)) \ldots (e_k(k)) \\ \ldots . \end{cases} \tag{24}$$

Every node possesses an infinite number of sons in the generating tree determined by $\widetilde{\Omega}^{\infty}$. The rule $\widetilde{\Omega}^{\infty}$ is called the *exploded succession rule* associated with Ω.

Next we study the bivariate generating functions and the number sequences given by (24). Quite surprisingly, we get rather simple expressions and closed forms in contrast with the (formal) difficulties when passing from doubled rules to arbitrary jumping rules.

Proposition 4.1 *The bivariate generating function related to* $\widetilde{\Omega}^{\infty}$ *has the form:*

$$\frac{t-1}{1-t-tL}(L(\mathbf{1})) = (t-1)\cdot\sum_{n\geq 0}(1+L)^n t^n(L(\mathbf{1})). \tag{25}$$

Proof. Consider the bivariate generating function of the jumping rule $\widetilde{\Omega}^{(1,2,\ldots,h)}$:

$$\frac{1}{1-tL-t^2L-\ldots-t^hL}(L(\mathbf{1})) = \frac{1}{1-tL(1+t+\ldots+t^{h-1})}(L(\mathbf{1})).$$

By letting h tend to infinity we get:

$$\begin{aligned}\frac{1}{1-tL\cdot\sum_{h\geq 0}t^h}(L(\mathbf{1})) &= \frac{1}{1-tL\frac{1}{1-t}}(L(\mathbf{1}))\\ &= \frac{1-t}{1-t-tL}(L(\mathbf{1}))\end{aligned}$$

which is the desired generating function. □

Theorem 4.1 *The sequence* $(s_n^{\infty})_{n\geq 0}$ *determined by* $\widetilde{\Omega}^{\infty}$ *is:*

$$\begin{aligned} s_0^{\infty} &= 1;\\ s_n^{\infty} &= \sum_{k=0}^{n-1}\binom{n-1}{k}s_{k+1}, \qquad n\geq 1. \end{aligned} \tag{26}$$

We will say that $(s_n^{\infty})_{n\geq 0}$ *is the* exploded Fibonacci transform *of the sequence* $(s_n)_{n\geq 0}$.

Proof. We manipulate the generating function obtained in Proposition 4.1 in the usual way:

$$\begin{aligned} f(x,t) &= (1-t)\sum_{n\geq 0}(1+L)^n t^n(L(\mathbf{1}))\\ &= \sum_{n\geq 0}(1+L)^n(L(\mathbf{1}))t^n - \sum_{n\geq 1}(1+L)^{n-1}(L(\mathbf{1}))t^n\\ &= L(\mathbf{1}) + \sum_{n\geq 1}(1+L)^{n-1}L^2(\mathbf{1})t^n. \end{aligned}$$

Thus, for $n \geq 1$, we have:

$$\begin{aligned} s_n^{\infty} &= [(1+L)^{n-1} L^2(\mathbf{1})]_{x=1} \\ &= \left[\sum_{k=0}^{n-1} \binom{n-1}{k} L^{k+2}(\mathbf{1})\right]_{x=1} \\ &= \sum_{k=0}^{n-1} \binom{n-1}{k} s_{k+1}, \end{aligned}$$

as desired. □

Example 4.1 1. Let

$$\Omega : \begin{cases} (1) \\ (1) \rightsquigarrow (1). \end{cases}$$

We already know that $\Omega^{(1,2)}$ counts the Fibonacci numbers, $\Omega^{(1,2,3)}$ counts the Tribonacci numbers, and so on. Which is the sequence counted by the exploded rule $\widetilde{\Omega}^{\infty}$? By applying theorem 4.1 we get:

$$s_n^{\infty} = \sum_{k=0}^{n-1} \binom{n-1}{k} = 2^{n-1}. \tag{27}$$

The table below shows the first terms of the sequences defined by $\Omega^{(1,2,3,\ldots,h)}$:

$k \backslash n$	0	1	2	3	4	5	6	...
1	1	1	1	1	1	1	1	...
2	1	1	2	3	5	8	13	...
3	1	1	2	4	7	13	24	...
4	1	1	2	4	8	15	29	...
5	1	1	2	4	8	16	31	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
∞	1	1	2	4	8	16	32	...

Thus *the total number of nodes at level n in the generating tree determined by $\widetilde{\Omega}^{\infty}$ is equal to* 2^{n-1}. Now we give a nice combinatorial interpretation of this result. For any fixed n, the set of nodes at level n in the generating tree characterized by $\widetilde{\Omega}^{\infty}$ can be described using the words of length n of the language $\mathcal{L}$ on the alphabet $\Sigma = \{x_1, x_2, x_3, \dots x_n\}$ generated by the regular grammar:

$$S \to x_1 S | x_2^2 S | x_3^3 S | \dots | x_n^n S | \epsilon.$$

Indeed, for any node N at level n, let us consider the path from the root to N. Following such a path, each edge of length i ($i \leq n$) is coded by x_i^i. Thus we obtain a word of $\mathcal{L}$ having length n. For instance, the nodes at level $n = 4$ are coded by the words $x_1x_1x_1x_1$, $x_1x_1x_2x_2$, $x_1x_2x_2x_1$, $x_2x_2x_1x_1$, $x_2x_2x_2x_2$, $x_1x_3x_3x_3$, $x_3x_3x_3x_1$, $x_4x_4x_4x_4$.

Therefore we give another proof of (27) by providing a bijection between n-length words of $\mathcal{L}$, and $(n-1)$-length paths in the discrete plane, running from $(0,0)$ and using *rise* steps $(1,1)$ or *fall* steps $(1,-1)$. Each word $w \in \mathcal{L}$ can be univocally decomposed into blocks:

$$w = B_1 B_2 \dots B_h, \qquad B_i \in \Sigma^+,$$

such that $B_i = x_l^l$, $i = 1, \dots, h$. For example the word $x_2x_2x_2x_2x_1x_3x_3x_3x_1$ is constituted by the blocks x_2x_2, x_2x_2, x_1, $x_3x_3x_3$, x_1. Now we recursively define the function ψ on the words of $\mathcal{L}$ as follows:

$$\psi(\epsilon) = \psi(x_i) = \text{the empty path}, \qquad x_i \in \Sigma;$$

$$\psi(x_i x_j) = \begin{cases} \text{rise step} & \text{if } x_i \text{ and } x_j \text{ belong to the same block,} \\ \text{fall step} & \text{otherwise;} \end{cases}$$

$$\psi(w) = \psi(x_1x_2)\psi(x_2x_3)\dots\psi(x_{n-1}x_n) \qquad \text{being } w = x_1x_2\dots x_n,\ x_i \in \Sigma.$$

It is easy to prove that, for each $n \geq 1$, ψ is a bijection between n-length words in $\mathcal{L}$ and $(n-1)$-length paths. Figure 6 shows the bijection for $n = 4$.

2. By generalizing the above example we can consider:

$$\Omega_a : \begin{cases} (a) \\ (a) \rightsquigarrow (a)^a, \end{cases}$$

defining the sequence $(a^n)_{n\geq 0}$. A simple computation shows that the sequence counted by the exploded succession rule $\widetilde{\Omega}_a^{\infty}$ is precisely $(a(a+1)^{n-1})_{n\geq 0}$.

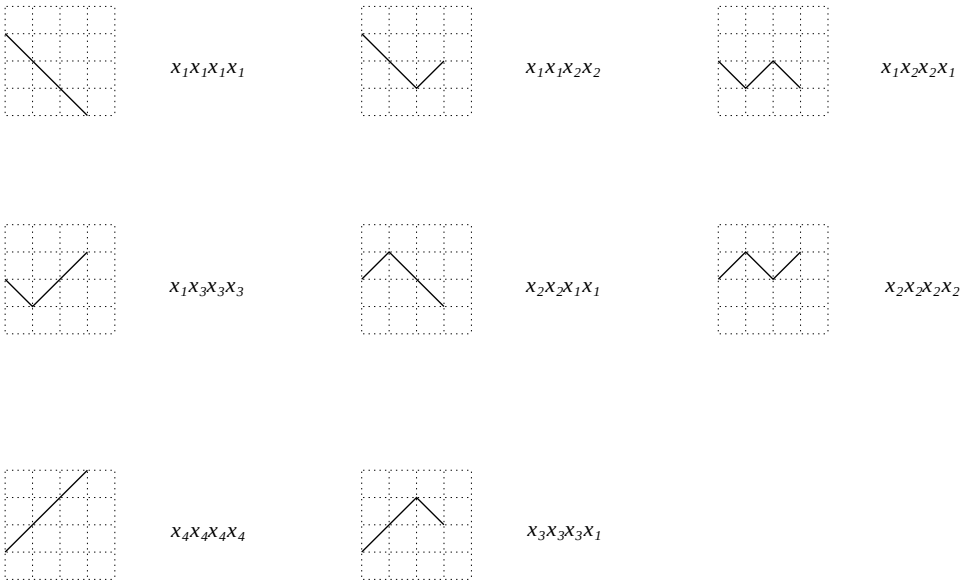


Figure 6: A bijective proof for the infinite Fibonacci transform of the sequence $1, 1, 1, 1, \ldots$.

Example 4.2 Let Ω be the rule (12) defining Catalan numbers. Let us now consider the rules $\Omega^{(1,2,3,\ldots,k)}$, $k \geq 1$; for any fixed k, the rule $\Omega^{(1,2,3,\ldots,k)}$ enumerates the language defined by the unambiguous context-free grammar:

$$S \to x_1 S | x_2^2 S | \ldots | x_k^k S^2 | \epsilon.$$

Then the generating function $f_k(x)$ of the rule $\Omega^{(1,2,3,\ldots,k)}$ is easily determined:

$$f_k(x) = \frac{1 - \sqrt{1 - 4(x + x^2 + \ldots + x^k)}}{2(x + x^2 + \ldots + x^k)}.$$

Letting k tend to infinity we have the generating function $f_\infty(x)$ for the exploded rule $\widetilde{\Omega}^\infty$:

$$f_\infty(x) = \frac{1 - x - \sqrt{1 - 6x + 5x^2}}{2x}.$$

This generating function defines a sequence f_n^∞ which is strictly related to Catalan numbers: the numbers are $1, 1, 3, 10, 36, 137, 543, 2219, 9285, \ldots$, (A002212 in [14]), and count two different structures:

1. f_{n+1}^∞ is the number of 3-coloured Motzkin paths having length n ([15]);
2. f_n^∞ is the number of edge-rooted polyhexes having n hexagons ([11]).

These facts still ask for a combinatorial explanation.

Example 4.3 Let $(B_n)_{n\geq 0}$ be the sequence of Bell numbers; by definition, B_n counts the way to partition an n-set into nonempty subsets. We define the sequence $(\overline{B}_n)_{n\geq 0}$ of *shifted Bell numbers* by setting $\overline{B}_0 = 1$ and $\overline{B}_{n+1} = B_n$ for all $n \in \mathbb{N}$. A succession rule Ω counting these numbers is the following:

$$\Omega : \begin{cases} (\overline{1}) \\ (\overline{1}) \rightsquigarrow (1) \\ (k) \rightsquigarrow (k)^{k-1}(k+1). \end{cases}$$

This is a typical example of a *coloured succession rules*. It is not difficult to extend all the notions defined in this paper to coloured rules. In particular, we can consider the exploded succession rule $\widetilde{\Omega}^{\infty}$; by the usual properties of Bell numbers, we observe that the shifted Bell numbers constitute a "quasi-fixed point" for the infinite Fibonacci transform, since:

$$\overline{B}_n^{\infty} = \sum_{k=0}^{n-1} \binom{n-1}{k} \overline{B}_{k+1} = \sum_{k=0}^{n-1} \binom{n-1}{k} B_k = B_n = \overline{B}_{n+1}. \qquad (28)$$

A result analogous to Theorem 2.2 holds for exploded succession rules.

Theorem 4.2 *Let Ω be a succession rule, and Ω^{∞} the exploded succession rule associated with Ω. Then the succession rule*

$$\Omega' : \begin{cases} (a) \\ (a) \rightsquigarrow (e_1(a)+1)\dots(e_a(a)+1) \\ (k+1) \rightsquigarrow (e_1(k)+1)(e_2(k)+1)\dots(e_k(k)+1)(k+1), \end{cases} \qquad (29)$$

is equivalent to Ω^{∞}.

Example 4.4 Let Ω be the rule defining Fibonacci numbers, having (2) as axiom:

$$\begin{cases} (2) \\ (1) \rightsquigarrow (2), \\ (2) \rightsquigarrow (1)(2) \end{cases}$$

According to Theorem 4.2 the exploded succession rule Ω^∞ associated with Ω is equivalent to the following:

$$\begin{cases} (2) \\ (2) \rightsquigarrow (2)(3), \\ (3) \rightsquigarrow (2)(3)(3), \end{cases}$$

which defines the odd Fibonacci numbers!

Example 4.5 The exploded rule of Catalan numbers, already examined in Example 4.2, is equivalent to the rule:

$$\begin{cases} (1) \\ (1) \rightsquigarrow (3), \\ (k) \rightsquigarrow (3)(4)\dots(k)(k)(k+1), \end{cases}$$

One can go further and iterate the application of the transform defined in Theorem 4.2 to a given succession rule. Let $\mathcal{S}$ be the set of succession rules, and let $T : \mathcal{S} \to \mathcal{S}$ be the operator such that, for any rule Ω, $T(\Omega)$ is the rule defined by (29), equivalent to the exploded succession rule $\widetilde{\Omega}^\infty$ associated with Ω. Now let us define:

$$T^0(\Omega) = \Omega$$

$$T^n(\Omega) = T(T^{n-1}(\Omega)) \qquad n \geq 1$$

Now let Ω be the rule (12) defining Catalan numbers. We easily obtain the following facts, which extend our previous results:

i) for any $n \geq 0$, $T^n(\Omega)$ has the form:

$$\begin{cases} (1) \\ (1) \rightsquigarrow (n+2), \\ (k) \rightsquigarrow (n+2)(n+3)\dots(k-1)(k)^{n+1}(k+1); \end{cases}$$

ii) for any $n \geq 0$, $T^n(\Omega)$ enumerates $(n+2)$-coloured Motzkin paths according to the length of the path.

In a word, the combinatorial meaning of Theorem 4.2 is that exploded succession rules do not enlarge the set of generating functions of succession rules.

5 Further work

1. Given a sequence $(D_n)_{n\geq 0}$, is it possible to find a sequence $(C_n)_{n\geq 0}$ such that $(D_n)_{n\geq 0}$ is its Fibonacci transform? This is simply the problem of inverting a combinatorial sum, and it has been solved, for example, in the classical text [12], where it is classified as a Chebyshev inverse relation. The solution is:

$$C_n = \sum_{k=0}^{n} (-1)^k \left(\binom{n+k-1}{k} - \binom{n+k-1}{k-1} \right) D_{n-k}.$$

   Instead, it would be interesting to know when the sequence $(C_n)_{n\geq 0}$ can be represented by means of a suitable succession rule, since in this case we are able to describe D_n using a doubled succession rule. Of course, these problems can be stated for the Fibonacci transform of any type, but their solution seems much more complicated.

2. If a sequence $(C_n)_{n\geq 0}$ can be described by means of a succession rule, does the same happen for its Fibonacci transform? We have seen that the answer is positive if we allow coloured rules, but the problem remains open if we restrict to non coloured ones. A solution to this question would allow to iterate the Fibonacci transform, as we did in Example 4.5 for the exploded Fibonacci transform.

3. Shifted Bell numbers are a "quasi-fixed point" for the exploded Fibonacci transform. What about the Fibonacci transforms of any other type?

4. Given a double-indexed sequence $\sigma_{n,k}$, we can define, for any sequence $(C_n)_{n\geq 0}$:

$$C_n^{\sigma} = \sum_{k=0}^{n} \sigma_{n-k,k} C_{n-k}.$$

   This is clearly done in analogy with Fibonacci transform. Can we say anything about the sequence $(C_n^{\sigma})_{n\geq 0}$? Is it possible to give a description of this transform in terms of something similar to succession rules, at least when $\sigma_{n,k}$ is a sequence of combinatorial interest (Stirling numbers, etc.)?

References

[1] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy and D. Gouyou-Beauchamps, On generating functions of generating trees, *Proceedings of 11th FPSAC* (1999) 40-52.

[2] E. Barcucci, A. Del Lungo, E. Pergola, R. Pinzani, ECO: a methodology for the Enumeration of Combinatorial Objects, *Journal of Difference Equations and Applications,* Vol.5 (1999) 435-490.

[3] F. R. K. Chung, R. L. Graham, V. E. Hoggatt, M. Kleimann, The number of Baxter permutations, *J. Combin. Theory Ser. A,* 24 (1978) 382-394.

[4] S. Corteel, Problèmes énumératifs issus de l'Informatique, de la Physique et de la Combinatorie, *Phd Thesis, Université de Paris-Sud,* (2000).

[5] E. Deutsch, Dyck path enumeration, *Discrete Mathematics,* 204 (1999) 167-202.

[6] E. Duchi, J. Fedou, C. Garcia, E. Pergola, Eco inversion, *preprint,* available at *http://dsi2.ing.unifi.it/~elisa/elisa.html.*

[7] L. Ferrari, E. Pergola, R. Pinzani, S. Rinaldi, An algebraic characterization of the set of succession rules, *Theor. Comp. Sci.* (to appear).

[8] L. Ferrari, R. Pinzani, A linear operator approach to succession rules, *preprint,* available at *http://www.dsi.unifi.it/~pinzani.*

[9] O. Guibert, Combinatoires des permutations a motifs exclus en liaison avec mots, cartes planaires et tableaux de Young, *Thèse de l'Université de Bordeaux I* (1996).

[10] O. Guibert, E. Pergola, Enumeration of vexillary involutions which are equal to their mirror/complement, *Discrete Mathematics*, 224 (200) 281-287.

[11] F. Harary, R. Read, The enumeration of tree-like polyhexes, *Proc. Edinburgh Math. Soc.,* 17 (1970) 1-13.

[12] J. Riordan, Combinatorial identities, John Wiley & sons, Inc., New York (1968).

[13] A. G. Shannon, Tribonacci numbers and Pascal's Pyramid, *Fibonacci Quarterly,* 15 (1977) 3, 268,275.

[14] N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences, *http://www.research.att.com/~njas/sequences/index.html.*

[15] R.A. Sulanke, Recurrences for moments of generalized Motzkin paths, *Journal of Integer Sequences*, Vol. 3 (2000), Article 00.1.1.

[16] J. West, Generating trees and the Catalan and Schröder numbers, *Discrete Mathematics,* 146 (1995) 247-262.

Extremal problems (and a bit of enumeration) for hypergraphs with linearly ordered vertex sets

Martin Klazar

June 5, 2001

Abstract

A hypergraph $\mathcal{H} = (H_i : \ i \in I)$ with the vertex set $\bigcup_{i\in I} H_i = [n] = \{1, 2, \ldots, n\}$ contains another hypergraph $\mathcal{H}' = (H'_i : \ i \in I')$ with the vertex set $[m]$ $(m \leq n)$ if there is a subsequence $1 \leq v_1 < v_2 < \cdots < v_m \leq n$ of $[n]$ and an injection $f : I' \to I$ such that, for every $r \in [m]$ and $i \in I'$, $r \in H'_i$ implies that $v_r \in H_{f(i)}$. We investigate the extremal functions $H_e(\mathcal{F}, n)$ and $H_i(\mathcal{F}, n)$ defined as the maximum size $e(\mathcal{H}) = |\mathcal{H}| = |I|$, resp. weight $i(\mathcal{H}) = \sum_{i\in I} |H_i|$, of a simple $\mathcal{H}$ with n vertices if $\mathcal{H}$ does not contain $\mathcal{F}$. We determine both functions exactly if $\mathcal{F}$ has only disjoint singleton edges or if $i(\mathcal{F}) \leq 4$ (there are 55 such $\mathcal{F}$). We give enumerative formulas for the numbers of both simple and all $\mathcal{H}$ with $i(\mathcal{H}) = n$ and derive two identities analogous to Dobiǹski's formula for Bell numbers. In the extremal problem we derive, by means of Davenport–Schinzel sequences, two general almost linear bounds. We consider the forbidden 4-path $\mathcal{F}_{42} = 13, 15, 23, 24$ introduced by Füredi and prove that $H_e(\mathcal{F}_{42}, n)$ and $H_i(\mathcal{F}_{42}, n)$ are $O(n \log^2 n \log\log^3 n)$. (Füredi proved in the bipartite graph case the $O(n \log n)$ bound.)

1 Introduction and motivation

Let us begin by stating a typical example of the extremal problems which we shall investigate in our article. If $\mathcal{H}$ is a simple hypergraph with the vertex set $[n] = \{1, 2, \ldots, n\}$ and such that for no four vertices $1 \leq a < b < c < d \leq n$

and for no two distinct edges $A, B \in \mathcal{H}$ the four incidences $a, c \in A$ and $b, d \in B$ occur, what is the maximum number of edges $|\mathcal{H}|$ and what is the maximum weight $\sum_{H\in\mathcal{H}} |H|$. Among other results we prove that the former maximum is $4n-5$ and that the latter is $8n-12$ ($n > 1$). Actually we proved it already in Klazar [15].

A *hypergraph* $\mathcal{H} = (H_i : i \in I)$ is a finite list of finite nonempty subsets H_i of $\mathbf{N} = \{1, 2, \ldots\}$, called *edges*. *Simple* hypergraphs have no repeated edges. The elements of $\bigcup\mathcal{H} = \bigcup_{i\in I} H_i \subset \mathbf{N}$ are called *vertices*. Our hypergraphs have no isolated vertices. Let $\mathcal{H} = (H_i : i \in I)$ and $\mathcal{H}' = (H'_i : i \in I')$ be two hypergraphs. If there exists an *increasing* (with respect to the standard linear ordering of $\mathbf{N}$) injection $F : \bigcup\mathcal{H}' \to \bigcup\mathcal{H}$ and an injection $f : I' \to I$ such that the implication $v \in H'_i \Longrightarrow F(v) \in H_{f(i)}$ holds for every $v \in \bigcup\mathcal{H}'$ and $i \in I'$, we say that $\mathcal{H}$ *contains* $\mathcal{H}'$ and write $\mathcal{H} \supset \mathcal{H}'$. Else we say that $\mathcal{H}$ is $\mathcal{H}'$*-free* and write $\mathcal{H} \not\supset \mathcal{H}'$. The subsets $F(\bigcup\mathcal{H}')$ and $f(I')$ form the $\mathcal{H}'$*-copy* in $\mathcal{H}$. For example, $\mathcal{H}$ is $(\{1\}_1, \{1\}_2)$-free iff its edges are pairwise disjoint, that is, $\mathcal{H}$ is a set partition. The initial example corresponds to $\mathcal{H}'$-freeness for $\mathcal{H}' = (\{1,3\}, \{2,4\})$. If F and f are bijections (remember that F is increasing) and the equivalence $v \in H'_i \Longleftrightarrow F(v) \in H_{f(i)}$ holds for every $v \in \bigcup\mathcal{H}'$ and $i \in I'$, we say that $\mathcal{H}'$ and $\mathcal{H}$ are *isomorphic*.

The *order* $v(\mathcal{H})$ of $\mathcal{H} = (H_i : i \in I)$ is the number of vertices $v(\mathcal{H}) = |\bigcup\mathcal{H}|$, the *size* $e(\mathcal{H})$ is the number of edges $e(\mathcal{H}) = |\mathcal{H}| = |I|$, and the *weight* $i(\mathcal{H})$ is the number of incidences between the vertices and the edges $i(\mathcal{H}) = \sum_{i\in I} |H_i|$. Trivially, $v(\mathcal{H}) \le i(\mathcal{H})$ and $e(\mathcal{H}) \le i(\mathcal{H})$.

We associate with every hypergraph $\mathcal{F}$ (the letter $\mathcal{F}$ is for "forbidden") two (extremal) functions $H_e(\mathcal{F}), H_i(\mathcal{F}) : \mathbf{N} \to \mathbf{N}$ defined by

$$\begin{aligned} H_e(\mathcal{F}, n) &= \max\{e(\mathcal{H}) : \mathcal{H} \not\supset \mathcal{F} \,\&\, \mathcal{H} \text{ is simple } \&\, v(\mathcal{H}) = n\} \\ H_i(\mathcal{F}, n) &= \max\{i(\mathcal{H}) : \mathcal{H} \not\supset \mathcal{F} \,\&\, \mathcal{H} \text{ is simple } \&\, v(\mathcal{H}) = n\}. \end{aligned}$$

It is clear that in the definition $\mathcal{H}$ must be simple. (For $\mathcal{F}$ of the form $(\{1\}_1, \{1\}_2, \ldots, \{1\}_k)$ the simplicity may be dropped but not for any other $\mathcal{F}$.) On the other hand, $\mathcal{F}$ may be any hypergraph, not necessarily simple. In Sections 5 and 6 we work also with the graph version $G_e(\mathcal{F}, n)$ of $H_e(\mathcal{F}, n)$ in which $\mathcal{H}$ runs through graphs ($|E| = 2$ for every $E \in \mathcal{H}$) and with the unordered versions $H^u_e(\mathcal{F}, n)$ and $G^u_e(\mathcal{F}, n)$ in which the vertex injection F is not required to be increasing. Thus for a graph $\mathcal{F}$ the function $G^u_e(\mathcal{F}, n)$ equals to the classical graph extremal function $\mathrm{ex}(\mathcal{F}, n)$. The *reversal* $\overline{\mathcal{F}}$ is obtained from $\mathcal{F}$ by reverting the linear order of $\bigcup\mathcal{F}$. Obviously, $H_e(\overline{\mathcal{F}}, n) =$

$H_e(\mathcal{F},n)$ and $H_i(\overline{\mathcal{F}},n) = H_i(\mathcal{F},n)$. It is also obvious that, for every $n \in \mathbf{N}$ and $\mathcal{F}$, $H_e(\mathcal{F},n) \le 2^n - 1$ and $H_i(\mathcal{F},n) \le n2^{n-1}$ but much better bounds can be given. In the forthcoming sections we investigate the behaviour of $H_e(\mathcal{F},n)$ and $H_i(\mathcal{F},n)$ for various fixed $\mathcal{F}$ and n running through $\mathbf{N} = \{1,2,\ldots\}$. We considered $H_e(\mathcal{F},n)$ and $H_i(\mathcal{F},n)$ implicitly already in [15]. Except this article, as far as we know, our extremal setting is new and was not investigated before. We stress again its two not so usual features: the containment is an ordered one (the vertex injection F is increasing) and $\mathcal{H}$ may have edges of any sizes (even if the forbidden $\mathcal{F}$ is a mere graph).

Before summarizing our results we say few words about our motivation and about connections to other results in extremal set systems theory. In this branch of combinatorics (see, for example, surveys of Frankl [10], Füredi [8], and Tuza [23, 24] or the collection [11]) one is interested in the maximum number of edges in set systems subject to some restrictions. These may restrict intersections of edges or they may exclude some forbidden (sub)configurations. Almost always the underlying universum of vertices is supposed to be unordered. We know of only one systematic study of a class of "ordered" extremal problems (for set systems; we are not speaking here of posets, words etc.), the work of Füredi and Hajnal [9] that deals with simple bipartite graphs with ordered parts. (In [9] the equivalent language of 0-1 matrices is used. So is in Anstee, Ferguson and Sali [1], see also further references thereof, but their extremal problems are "unordered".) One our aim is just to explore the properties of $H_e(\mathcal{F},n)$ and $H_i(\mathcal{F},n)$ and the possibilities which open here. Other aim is to apply results and techniques from the theory of *Davenport–Schinzel sequences* which deals with extremal problems for words; necessary definitions and references will be given in Section 5. Also, we want to extend some results of [9] from 2-element edges to edges of arbitrary cardinality.

In Section 2 we consider hypergraphs $\mathcal{S}_k$ which consist of k disjoint singleton edges. (For the containment of $\mathcal{S}_k$ the order of vertices is irrelevant.) Theorems 2.1 and 2.3 determine $H_e(\mathcal{S}_k,n)$ and $H_i(\mathcal{S}_k,n)$ exactly. We describe all extremal hypergraphs as well. In Theorem 2.2 we prove that if $\mathcal{F} \neq \mathcal{S}_k$ then $H_e(\mathcal{F},n)$ is a strictly increasing function. Trivially, $H_e(\mathcal{F},n) \le H_i(\mathcal{F},n)$ for every $n \in \mathbf{N}$ and $\mathcal{F}$. Theorem 2.4 states that if $\mathcal{F}$ has no two edges of which one completely precedes the other, then for every $n \in \mathbf{N}$ also $H_i(\mathcal{F},n) \le cH_e(\mathcal{F},n)$ where $c > 0$ depends only on $\mathcal{F}$. Theorem 2.5 gives a trivial polynomial upper bound on $H_e(\mathcal{F},n)$. In Section 3 we precisely determine $H_e(\mathcal{F},n)$ and $H_i(\mathcal{F},n)$ for each of the 55 $\mathcal{F}$ with

$1 \le i(\mathcal{F}) \le 4$. Section 4 is enumerative. In Theorem 4.1 we give formulas for the number of hypergraphs, both simple and all, with prescribed numbers of edges of a given cardinality. We use the formulas to calculate the total numbers of hypergraphs, both simple and all, of weight n for $n \le 10$. In Corollary 4.3 two identities similar to Dobiǹski's formula are given. (Dobiǹski's formula deals with set partitions and our identities deal with hypergraphs.) In Section 5 we apply generalized Davenport–Schinzel sequences to obtain two almost linear bounds in which $\alpha(n)$, the inverse Ackermann function, is involved. In Theorem 5.1 we prove that for every fixed set partition $\mathcal{F}$ for every $\mathcal{F}$-free $\mathcal{H}$ the weight $i(\mathcal{H})$ is bounded almost linearly in $e(\mathcal{H})$. An example is given showing that the superlinearity is inevitable. Theorem 5.2 gives an almost linear upper bound on $G_e(\mathcal{F}, n)$ ($\mathcal{H}$ has only two-element edges) in the case that $\mathcal{F}$ is a forest whose components are stars which have all centers smaller than all leaves. An example shows that the superlinearity is again genuine. In Section 6 we investigate the case when $\mathcal{F}$ is a forest such that one part of the bipartition of $\mathcal{F}$ is smaller than the other (it is easy to see that for other $\mathcal{F}$ we have $H_e(\mathcal{F}, n) \gg n^\gamma$, $\gamma > 1$). Theorem 6.2 gives a method for deriving good upper bounds on $H_e(\mathcal{F}, n)$ from those on $G_e(\mathcal{F}, n)$. We give three applications. Theorem 6.3 extends the classical (easy) unordered graph result $\mathrm{ex}(\mathcal{F}, n) \ll n$ if $\mathcal{F}$ is a forest to hypergraphs: $H_e^u(\mathcal{F}, n) \ll n$ for every forest $\mathcal{F}$. Theorem 6.4 extends the almost linear graph bound of Theorem 5.2 to hypergraphs: $H_e(\mathcal{F}, n)$ is almost linear whenever $\mathcal{F}$ is a star forest. In the last Theorem 6.6 we prove the bound $H_e(\mathcal{F}, n) \ll n(\log n)^2(\log\log n)^3$ if $\mathcal{F} = (\{1,3\},\{1,5\},\{2,3\},\{2,4\})$. This forbidden path was investigated first by Füredi who in [7] and [9] proved graph bounds $n \log n \ll G_e(\mathcal{F}, n) \ll n \log n$ (for ordered bipartite graphs). Section 7 contains some open problems.

We need few more definitions. Notation $f(n) \ll g(n)$ is synonymous to the $f(n) = O(g(n))$ notation. If $m, n \in \mathbf{N}$ and $m \le n$, then $[n] = \{1, 2, \ldots, n\}$ and $[m, n] = \{m, m+1, \ldots, n\}$. The *degree* $\deg(v) = \deg_{\mathcal{H}}(v)$ of v in $\mathcal{H} = (H_i : i \in I)$ is the number of the edges H_i containing v. The *simplification* of $\mathcal{H}$ is a simple hypergraph $\mathcal{H}'$ obtained by keeping from each family of repeated edges of $\mathcal{H}$ just one member. The *deletion* of H_j from $\mathcal{H}$ gives the hypergraph $(H_i : i \in I')$ where $I' = I \backslash \{j\}$. The *deletion* of $a \in \bigcup \mathcal{H}$ from $\mathcal{H}$ gives the hypergraph $(H_i \backslash \{a\} : i \in I)$ where we omit $\emptyset$ if $H_i = \{a\}$ (this operation in general destroys simplicity). We may also delete a only from some specified edges. A *(connected) component* $\mathcal{H}_1$ of $\mathcal{H}$ is the minimal subhypergraph $\mathcal{H}_1$ of $\mathcal{H}$ such that every $H \in \mathcal{H} \backslash \mathcal{H}_1$ is disjoint with

every $H_1 \in \mathcal{H}_1$.

2 Singleton hypergraph $\mathcal{S}_k$

In this section $\mathcal{F} = \mathcal{S}_k = (\{1\}, \{2\}, \ldots, \{k\})$. We give exact formulas for $H_e(\mathcal{S}_k, n)$ and $H_i(\mathcal{S}_k, n)$. For $k = 1$ both extremal functions are undefined.

Theorem 2.1 *Let $k \geq 2$ and $\mathcal{S}_k = (\{1\}, \{2\}, \ldots, \{k\})$. Then*

$$H_e(\mathcal{S}_k, n) = \begin{cases} 2^n - 1 & \ldots \quad 1 \leq n < k \\ 2^{k-2} & \ldots \quad n \geq k. \end{cases}$$

In particular, for $k \geq 3$ the function $H_e(\mathcal{S}_k, n)$ has the global maximum $H_e(\mathcal{S}_k, k-1) = 2^{k-1} - 1$.

Proof. The first formula is clear. For $k \geq 2$ and $n \geq k$ we have $H_e(\mathcal{S}_k, n) \geq 2^{k-2}$, because of the hypergraph $([n], X : \emptyset \neq X \subset [k-2])$. We prove by induction on k that for $n \geq k$ also $H_e(\mathcal{S}_k, n) \leq 2^{k-2}$. For $k = 2$ this holds because $H_e(\mathcal{S}_2, n) = 1$ for every $n \in \mathbf{N}$. Let $n \geq k \geq 3$ and $\mathcal{H}$ be simple, $\mathcal{S}_k$-free, and $\bigcup \mathcal{H} = [n]$. We can suppose that (i) $\deg(v) \geq 2$ for every $v \in \bigcup \mathcal{H}$ and (ii) there is an $H \in \mathcal{H}$ with $|H| \geq 2$ and an $a \in H$ such that $H \backslash \{a\} \notin \mathcal{H}$.

If (i) is false, there is a vertex a and an edge H such that $a \in H$ and a lies in no other edge of $\mathcal{H}$. We delete H from $\mathcal{H}$ and obtain a simple hypergraph $\mathcal{H}'$ that must be $\mathcal{S}_{k-1}$-free because any $\mathcal{S}_{k-1}$-copy in $\mathcal{H}'$ can be extended by H and a to $\mathcal{S}_k$-copy in $\mathcal{H}$. By induction, $e(\mathcal{H}) = e(\mathcal{H}') + 1 \leq (2^{(k-1)-1} - 1) + 1 = 2^{k-2}$. Suppose that (ii) is false. Let $a \in \bigcup \mathcal{H}$ be arbitrary and $H \in \mathcal{H}$, $a \in H$, be such that $|H|$ is as small as possible. If $|H| > 1$, we take $b \in H$, $b \neq a$, and the negation of (ii) gives $H \backslash \{b\} \in \mathcal{H}$, contradicting the minimality of $|H|$. Thus $|H| = 1$ and $\{a\} \in \mathcal{H}$. We obtain that $\{a\} \in \mathcal{H}$ for every vertex a of $\mathcal{H}$ but this implies the contradiction $\mathcal{H} \supset \mathcal{S}_k$ $(n \geq k)$.

We can assume that (i) and (ii) hold. Let a and H be as in (ii). Let $H' \in \mathcal{H}$ be such that $a \in H'$, $H' \neq H$, and, if possible, $|H'| = 1$. We define $\mathcal{H}'$ by deleting H' from $\mathcal{H}$ and then a from $\mathcal{H} \backslash \{H'\}$. Some edges may get duplicated and we set $\mathcal{H}''$ to be the simplification of $\mathcal{H}'$. By (i), $v(\mathcal{H}'') = v(\mathcal{H}) - 1 = n - 1 \geq k - 1$. Since any $\mathcal{S}_{k-1}$-copy in $\mathcal{H}''$ can be extended by H' and a to an $\mathcal{S}_k$-copy in $\mathcal{H}$, $\mathcal{H}''$ is $\mathcal{S}_{k-1}$-free. Also, $e(\mathcal{H}') \leq 2e(\mathcal{H}'') - 1$ because, by (ii), $H \backslash \{a\}$ is not duplicated in $\mathcal{H}'$. Notice that $\emptyset \notin \mathcal{H}''$ because

we have deleted $\{a\}$ as H'. By induction (now we use the stronger upper bound on $e(\mathcal{H}'')$),

$$e(\mathcal{H}) = e(\mathcal{H}') + 1 \leq (2e(\mathcal{H}'') - 1) + 1 = 2e(\mathcal{H}'') \leq 2 \cdot 2^{(k-1)-2} = 2^{k-2}.$$

□

$H_e(\mathcal{S}_k, n)$ has the strange feature of being independent of n. We show that the other functions $H_e(\mathcal{F}, n)$ are increasing, as one expects.

Theorem 2.2 *If $\mathcal{F} \neq \mathcal{S}_k$ then $H_e(\mathcal{F}, n) < H_e(\mathcal{F}, n+1)$ for every $n \in \mathbf{N}$.*

Proof. Let $\mathcal{F} \neq \mathcal{S}_k$ and $\bigcup \mathcal{F} = [m]$. We say that $\{u\} \in \mathcal{F}$ is an *isolated singleton* of $\mathcal{F}$ if $\deg(u) = 1$. Let l be the maximum number such that $\{1\}, \{2\}, \ldots, \{l\}$ are isolated singletons of $\mathcal{F}$. Since $\mathcal{F} \neq \mathcal{S}_k$, $0 \leq l < m$. Clearly, any other isolated singleton of $\mathcal{F}$ is preceded by at least $l+1$ vertices.

We proceed by induction on n. The inequality holds for every $n < m-1$ because then $H_e(\mathcal{F}, n) = 2^n - 1$. Let $n \geq m-1$ and let $\mathcal{H}$ attain the value $H_e(\mathcal{F}, n)$. If $a \in H \in \mathcal{H}$ and $\{a\} \notin \mathcal{H}$, we replace H by $\{a\}$. The new hypergraph is simple, $\mathcal{F}$-free, and it has the same number of edges and vertices as $\mathcal{H}$; order does not decrease because else we would have contradiction with the inductive assumption. Repeating the replacements we obtain a simple $\mathcal{F}$-free hypergraph $\mathcal{H}'$ such that $e(\mathcal{H}') = e(\mathcal{H}) = H_e(\mathcal{F}, n)$, $\bigcup \mathcal{H}' = \bigcup \mathcal{H} = [n]$, and $\{a\} \in \mathcal{H}'$ for every $a \in [n]$. We define $\mathcal{H}''$ by inserting in $\mathcal{H}'$, between l and $l+1$, a new singleton edge $\{u\}$. $\mathcal{H}''$ is simple and satisfies $v(\mathcal{H}'') = n+1$ and $e(\mathcal{H}'') = e(\mathcal{H}') + 1 = H_e(\mathcal{F}, n) + 1$. We show that $\mathcal{H}''$ is $\mathcal{F}$-free. This gives $H_e(\mathcal{F}, n+1) \geq e(\mathcal{H}'') > H_e(\mathcal{F}, n)$. The new edge $\{u\}$ would have to participate in every $\mathcal{F}$-copy in $\mathcal{H}''$. It cannot play the role of any of the initial l isolated singletons of $\mathcal{F}$ because $\{1\}, \{2\}, \ldots, \{l\} \in \mathcal{H}'$ and we would have already $\mathcal{F} \subset \mathcal{H}'$. It cannot play the role of any other isolated singleton of $\mathcal{F}$ either because those are preceded in $\mathcal{F}$ by at least $l+1$ vertices but $\{u\}$ is preceded in $\mathcal{H}''$ by only l vertices. Thus $\mathcal{H}'' \not\supset \mathcal{F}$. □

Theorem 2.3 *Let $k \geq 2$ and $\mathcal{S}_k = (\{1\}, \{2\}, \ldots, \{k\})$. Then*

$$H_i(\mathcal{S}_k, n) = \begin{cases} n2^{n-1} & \ldots \quad 1 \leq n < k \\ n + (k-2)2^{k-3} & \ldots \quad k \leq n \leq 2^{k-3}+1 \\ (k-1)n - (k-2) & \ldots \quad n \geq \max(k, 2^{k-3}+1). \end{cases}$$

Note that $H_i(\mathcal{S}_k, k-1) > H_i(\mathcal{S}_k, n)$ for $k \leq n \leq \max(k, 2^{k-2})$ $(k \geq 3)$.

Proof. The formula is clear for $1 \leq n < k$. We suppose that $n \geq k \geq 2$ and that $\mathcal{H}$ is a simple hypergraph with $\bigcup \mathcal{H} = [n]$. Its dual $\mathcal{H}^*$ is defined by

$$\mathcal{H}^* = (H_i^* : \ i \in [n]) \text{ where } H_i^* = \{H \in \mathcal{H} : \ i \in H\}.$$

Thus $e(\mathcal{H}^*) = v(\mathcal{H}) = n$. Let $\Gamma(X) = \Gamma_{\mathcal{H}}(X)$ be for $X \subset [n]$ defined by

$$\Gamma(X) = \left| \bigcup_{i \in X} H_i^* \right| = |\{H \in \mathcal{H} : \ H \cap X \neq \emptyset\}|.$$

By the defect form of P. Hall's theorem (Lovász [16, Problems 7.5 and 13.5]) applied on $\mathcal{H}^*$, $\mathcal{H}$ is $\mathcal{S}_k$-free if and only if

$$\max_{X \subset [n]} |X| - \Gamma(X) \geq n - k + 1.$$

Thus if $\mathcal{H}$ is $\mathcal{S}_k$-free, there exists a set $X \subset [n]$ of cardinality l, $n - k + 2 \leq l \leq n$ ($\Gamma(X) \geq 1$), intersected by only at most $l - n + k - 1$ edges of $\mathcal{H}$. And contrarywise, every such a hypergraph is (trivially) $\mathcal{S}_k$-free. Hence

$$i(\mathcal{H}) \leq (l - n + k - 1)n - (l - n + k - 2) + (n - l)2^{n-l-1} = f(l, k, n)$$

and this bound is attained.

The first difference of $f(l, k, n)$ with respect to l is the increasing function

$$f(l + 1, k, n) - f(l, k, n) = n - 1 - (n - l + 1)2^{n-l-2}.$$

Therefore $f(l, k, n)$ attains its maximum in one of the endpoints $l = n - k + 2$ and $l = n$ (or in both). The corresponding values are $f(n - k + 2, k, n) = n + (k - 2)2^{k-3}$ and $f(n, k, n) = (k - 1)n - (k - 2)$. These values are equal for $n = 2^{k-3} + 1$. For $n < 2^{k-3} + 1$ the former value dominates and for $n > 2^{k-3} + 1$ the latter. We obtain the other two formulas. Maximum weights are attained by $\mathcal{H}_1$ or by $\mathcal{H}_2$ where the edges of $\mathcal{H}_1$, respectively of $\mathcal{H}_2$, are $[n]$ together with all nonempty subsets of some $(k - 2)$-element set $Y \subset [n]$, respectively $[n]$ together with some $k - 2$ distinct $(n - 1)$-element subsets of $[n]$. □

It follows from the proof that $\mathcal{H}_1$ and $\mathcal{H}_2$ are the only types of extremal hypergraphs for $n \geq k$. For $1 \leq n < k$ the maximum weight is attained only by the complete hypergraph. We conclude that the number of simple $\mathcal{S}_k$-free hypergraphs having order n and the maximum weight is 1 if $1 \leq n < k$ and

$\eta_{k,n}\binom{n}{k-2}$ if $n \geq k$, where for $k = 2, 3, 4$ always $\eta_{k,n} = 1$ and for $k \geq 5$ we have $\eta_{k,n} = 1$ if $n \neq 2^{k-3} + 1$ and $\eta_{k,2^{k-3}+1} = 2$.

By means of P. Hall's theorem one can give a quick proof of Theorem 2.1 as well. The number of $\mathcal{H}$ attaining $H_e(\mathcal{S}_k, n)$ is seen to be 1 for $n < k$ and $2^{k-2}\binom{n}{k-2}$ for $n \geq k$.

Two subsets X and Y of $\mathbf{N}$ are *separated* if $\max X < \min Y$ or $\max Y < \min X$. Below we can assume that $e(\mathcal{F}) > 1$ because for $\mathcal{F}$ with just one edge $H_e(\mathcal{F}, n)$ and $H_i(\mathcal{F}, n)$ are easy to determine exactly.

Theorem 2.4 *Suppose that $\mathcal{F}$ has no two separated edges, $p = v(\mathcal{F})$, and $q = e(\mathcal{F}) > 1$. Then for every $n \in \mathbf{N}$*

$$H_i(\mathcal{F}, n) \leq (2p-1)(q-1)H_e(\mathcal{F}, n).$$

Proof. Let $\mathcal{H}$ attain $H_i(\mathcal{F}, n)$. We transform $\mathcal{H}$ in a new hypergraph $\mathcal{H}'$ by keeping all edges with less than p vertices and replacing every edge $H = \{v_1, v_2, \ldots, v_s\}$ of $\mathcal{H}$ with $s \geq p$, where $v_1 < v_2 < \cdots < v_s$, by $t = \lfloor |H|/p \rfloor$ new p-element edges $\{v_1, \ldots, v_p\}$, $\{v_{p+1}, \ldots, v_{2p}\}, \ldots, \{v_{(t-1)p+1}, \ldots, v_{tp}\}$. $\mathcal{H}'$ may not be simple and we set $\mathcal{H}''$ to be the simplification of $\mathcal{H}'$. Two observations: (i) no edge of $\mathcal{H}'$ repeats q or more times and (ii) $\mathcal{H}''$ is $\mathcal{F}$-free. If (i) were false, there would be q distinct edges $H_1, \ldots, H_q$ in $\mathcal{H}$ such that $|\bigcap_{i=1}^{q} H_i| \geq p$. But this implies the contradiction $\mathcal{F} \subset \mathcal{H}$. As for (ii), note that any $\mathcal{F}$-copy in $\mathcal{H}''$ may use from every $H \in \mathcal{H}$ only at most one new edge and so it is an $\mathcal{F}$-copy in $\mathcal{H}$ as well. The observations and the definitions of $\mathcal{H}'$ and $\mathcal{H}''$ imply

$$\begin{aligned} H_i(\mathcal{F}, n) = i(\mathcal{H}) &\leq \frac{(2p-1)i(\mathcal{H}')}{p} \leq \frac{(2p-1)(q-1)i(\mathcal{H}'')}{p} \\ &\leq (2p-1)(q-1)e(\mathcal{H}'') \\ &\leq (2p-1)(q-1)H_e(\mathcal{F}, n). \end{aligned}$$

In the last innocently looking inequality we use Theorem 2.2. □

The same idea gives for $H_e(\mathcal{F}, N)$ a trivial polynomial bound.

Theorem 2.5 *If $\mathcal{F}$ is a hypergraph with $p = v(\mathcal{F})$ and $q = e(\mathcal{F})$, then for every $n \in \mathbf{N}$*

$$H_e(\mathcal{F}, n) \leq (q-1)\binom{n}{p} + \binom{n}{p-1} + \cdots + \binom{n}{1}.$$

Proof. Let $\mathcal{H}$ attain $H_e(\mathcal{F},n)$. We put in $\mathcal{H}'$ every $H\in\mathcal{H}$ with $|H|<p$ and, for every $H\in\mathcal{H}$ with $|H|\geq p$, a p-element subset $H'\subset H$. Since no p-element edge of $\mathcal{H}'$ repeats more than $q-1$ times (else $\mathcal{H}\supset\mathcal{F}$) and other edges do not repeat at all, we have

$$H_e(\mathcal{F},n)=e(\mathcal{H})=e(\mathcal{H}')\leq(q-1)\binom{n}{p}+\binom{n}{p-1}+\cdots+\binom{n}{1}.$$

□

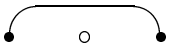
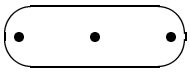
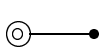
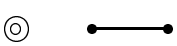
For $\mathcal{F}=([p]_1,[p]_2,\ldots,[p]_q)$ this bound is best possible.

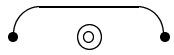
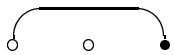
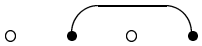
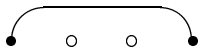
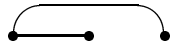
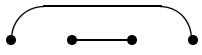
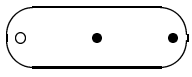
3 One hundred and ten extremal functions

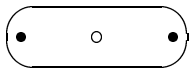
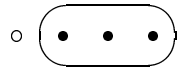
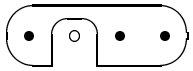
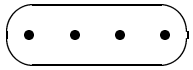
The table below lists extremal functions of the 55 nonempty forbidden $\mathcal{F}$ with $i(\mathcal{F})\leq 4$. In the proofs we refer to $\mathcal{F}$ according to the numbers in column 1. Star indicates that the reversal $\overline{\mathcal{F}}$ is nonisomorphic to $\mathcal{F}$ and is not listed, because it has the same extremal functions. $\mathcal{F}$ are visualized in column 2. Hypergraphs $\mathcal{F}$ with $i(\mathcal{F})=1,2,3$, and 4 occupy lines 1, 2–4, 5–11, and 12–39, respectively. Empty circle $\circ$ denotes a vertex that is a singleton edge and full circle $\bullet$ a vertex that is not a singleton edge. Two-element edges are indicated by arcs and larger edges by ovals. Concentric circles or arcs sharing both endpoints ($\mathcal{F}_{31}$) indicate edge multiplicities. For example, $\mathcal{F}_{18}=(\{1\}_1,\{1\}_2,\{1,2\})$ and $\mathcal{F}_{36}=(\{1,2,3\},\{2\})$. Columns 3 and 4 list functions $H_e(\mathcal{F},n)$ and $H_i(\mathcal{F},n)$. The formulas given hold for all $n\in\mathbf{N}$ if not said otherwise. The extremal functions for hypergraphs $\mathcal{F}_{33}$ and $\mathcal{F}_{34}$ were determined already in [15] but we give the arguments here again for the sake of completeness.

Theorem 3.1

no.	picture of $\mathcal{F}$	$H_e(\mathcal{F},n)$	$H_i(\mathcal{F},n)$
1	$\circ$	not defined	not defined
2	◎	n	n
3	$\circ$ $\quad$ $\circ$	$1,1,\ldots$	n

no.	picture of $\mathcal{F}$	$H_e(\mathcal{F},n)$	$H_i(\mathcal{F},n)$
4		n	n
5		$\left\lfloor \frac{3n}{2} \right\rfloor$	$2n$ $(n>1)$
6*		n	$2n-1$
7		$1,3,2,2,\ldots$	$2n-1$ $(n\neq 2)$
8*		n	$2n-1$
9*		$2n-1$	$3n-2$
10		$2n-1$	$3n-2$
11		$\binom{n+1}{2}$	n^2
12		$2n$ $(n>2)$	$3n$ $(n>2)$
13*		$2n-1$	$\left\lfloor \frac{7(n-1)}{2} \right\rfloor + 1$
14		$n+1$ $(n>1)$	$3n-2$
15*		$n+1$ $(n>1)$	$3n-2$
16		$n+1$ $(n>1)$	$3n-2$
17		$1,3,7,4,4,\ldots$	$3n-2$ $(n\neq 3)$
18*		$2n-1$	$4n-6$ $(n>5)$
19*		$2n-1$	$3n-2$

no.	picture of $\mathcal{F}$	$H_e(\mathcal{F},n)$	$H_i(\mathcal{F},n)$
20		$2n-1$	$3n-2$
21*		$2n-1$	$3n-2$
22*		$2n-1$	$3n-2$
23*		$2n-1$	$3n-2$
24		n	$2n-1$
25*		$4n-5$ $(n>1)$	$8n-12$ $(n>1)$
26*		$4n-5$ $(n>1)$	$8n-12$ $(n>1)$
27		$4n-5$ $(n>1)$	$8n-12$ $(n>1)$
28		$4n-5$ $(n>1)$	$8n-12$ $(n>1)$
29*		$2n-1$	$3n-2$
30		$\left\lfloor \frac{n^2}{4} \right\rfloor + n$	$2\left\lfloor \frac{n^2}{4} \right\rfloor + n$ $(n \neq 3)$
31		$\binom{n+1}{2}$	n^2
32		$2\left\lfloor \frac{(n+1)^2}{4} \right\rfloor - 1$	$5\left\lfloor \frac{(n+1)^2}{4} \right\rfloor - 2n - 2$
33		$4n-5$ $(n>1)$	$8n-12$ $(n>1)$
34		$4n-5$ $(n>1)$	$8n-12$ $(n>1)$
35*		$\binom{n+1}{2}$	n^2

no.	picture of $\mathcal{F}$	$H_e(\mathcal{F},n)$	$H_i(\mathcal{F},n)$
36		$\binom{n+1}{2}$	n^2
37*		n^2-n+1	$\frac{5n^2-9n+6}{2}$
38*		n^2-n+1	$\frac{5n^2-9n+6}{2}$
39		$\frac{n^3+5n}{6}$	$\frac{n^3-n^2+2n}{2}$

Proof. $\mathcal{H}$ is a generic simple $\mathcal{F}$-free hypergraph with $\bigcup\mathcal{H}=[n]$, $n\in\mathbf{N}$. $\mathcal{H}$ is *full* if $\{a\}\in\mathcal{H}$ for every $a\in\bigcup\mathcal{H}$. If $a\in H\in\mathcal{H}$ and $\{a\}\notin\mathcal{H}$, we can replace H by $\{a\}$. (We used this replacement in the proof of Theorem 2.2.) Our hypergraph remains simple and $\mathcal{F}$-free, and its size has not changed (but order might decrease and weight decreases). Repeating this operation, we replace $\mathcal{H}$ by a full $\mathcal{H}'$ with $e(\mathcal{H}')=e(\mathcal{H})$ and $v(\mathcal{H}')=n'\le v(\mathcal{H})=n$. If $e(\mathcal{H}')\le f(n')$ for a nondecreasing function f, we have also $e(\mathcal{H})=e(\mathcal{H}')\le f(n')\le f(n)$. This little trick helps us to obtain upper bounds on $H_e(\mathcal{F},n)$ (it does not help for no. 4, 11, 29–34, and 39 where $\mathcal{F}$ has no singleton) but it does not work for $H_i(\mathcal{F},n)$. When determining $H_e(\mathcal{F},n)$ we assume, without repeating it every time, that $\mathcal{H}$ is replaced by a full $\mathcal{H}'$ with $\bigcup\mathcal{H}'=[n']$.

For obtaining upper bounds on $H_i(\mathcal{F},n)$ we use induction and/or other replacement arguments. To simplify the situation, we get rid of a large edge $H\in\mathcal{H}$ by replacing H by some sets H_i, usually (but not always, see $\mathcal{F}_{30}$) subsets of H. For the resulting hypergraph $\mathcal{H}_0$ one has to check three things: $\mathcal{H}_0$ remains simple ($H_i\notin\mathcal{H}$ for every i), $i(\mathcal{H}_0)\ge i(\mathcal{H})$ ($\sum_i|H_i|\ge|H|$), and $\mathcal{H}_0$ remains $\mathcal{F}$-free (the reason is usually that any $\mathcal{F}$-copy may use at most one of the new edges H_i and therefore, since $H_i\subset H$, $\mathcal{H}_0\supset\mathcal{F}$ implies the contradiction $\mathcal{H}\supset\mathcal{F}$). For each particular replacement these three conditions are easy to check and we leave it to the reader. Repeating the replacements, we eliminate all large edges.

1. No such $\mathcal{H}$ exists. 2. $\mathcal{H}$ is a set partition. 3. $\mathcal{H}$ has one edge. 4. $\mathcal{H}$ has only singleton edges.

5. Recall that $\mathcal{H}'$ is full. Therefore edges with $|H|\ge 2$ must be mutually disjoint and $H_e(\mathcal{F},n)\le n+\lfloor n/2\rfloor$, which is easy to attain. The value $H_i(\mathcal{F},n)=2n$ for $n>1$ is clear; $H_i(\mathcal{F},1)=1$.

6. $\mathcal{H}'$ besides singletons has no other edges and $H_e(\mathcal{F},n) \le n$, which is easy to attain. As for the weight, we have (in $\mathcal{H}$) $\deg(a) \le 2$ for every $a \in [n-1]$, and equality for an a implies $\deg(n) \le 2$. Hence $\deg(a) = 2$ for an $a < n$ implies $i(\mathcal{H}) \le 2n$. Even $i(\mathcal{H}) \le 2n-1$ because $\deg(a) = 2$ for every $a \in [n]$ is impossible ($\mathcal{H}$ is simple). In the other case when $\deg(a) = 1$ for every $a < n$ again $i(\mathcal{H}) \le 2n-1$ because then $\deg(n) \le n$. In both cases $i(\mathcal{H}) \le 2n-1$, attained by $\mathcal{H} = ([n],[n-1])$ and $\mathcal{H} = (\{i,n\},\{n\} : \ i \in [n-1])$.

7. Particular case of Theorems 2.1 and 2.3; $H_i(\mathcal{F},2) = 4$.

8. $H_e(\mathcal{F},n) = n$ for the same reason as in 6. As for $H_i(\mathcal{F},n)$, every component $\mathcal{H}_1$ of $\mathcal{H}$ consists of several edges which pairwise intersect in one common vertex that is their largest vertex. Thus $i(\mathcal{H}_1) \le 2v(\mathcal{H}_1) - 1$ and $H_i(\mathcal{F},n) \le 2n-1$, attained by $\mathcal{H} = (\{i,n\},\{n\} : i \in [n-1])$.

9. In $\mathcal{H}'$, $|H| \le 2$ for every edge and $|H| = 2$ implies $1 \in H$. Hence $H_e(\mathcal{F},n) \le n + (n-1)$, attained by $\mathcal{H} = (\{1\},\{1,i\},\{i\} : i \in [2,n])$. As for $H_i(\mathcal{F},n)$, we eliminate all edges with $|H| \ge 3$ by replacing H by two-element sets $\{a,b\}$ where $a = \min H$ and $a < b \in H$. Since $1 \in H$ for every two-element edge, $H_i(\mathcal{F},n) \le n + 2(n-1)$, attained by the already mentioned $\mathcal{H}$.

10. Use arguments similar to 9. Allowed two-element edges are now $\{i,i+1\}$.

11. $\mathcal{H}$ has only edges of cardinalities 1 and 2. Thus $H_e(\mathcal{F},n) = \binom{n}{1} + \binom{n}{2}$ and $H_i(\mathcal{F},n) = \binom{n}{1} + 2\binom{n}{2}$.

12. If $|H| \ge 3$ for an $H \in \mathcal{H}'$, then $H_1 \notin \mathcal{H}'$ for some $H_1 \subset H$ with $|H_1| = 2$. Replacing H by H_1, we get rid of all edges with three and more vertices. Every vertex is then contained in at most two two-element edges. Therefore $H_e(\mathcal{F},n) \le n + n$, attained for $n > 2$ by $\mathcal{H} = (\{i\},\{i,i+1\} : i \in [n])$ (taken modulo n). For $n = 1,2$ we have $H_e(\mathcal{F},n) = 1,3$. The value $H_i(\mathcal{F},n) = 3n$ for $n > 2$ is clear; for $n = 1,2$ we have $H_i(\mathcal{F},n) = 1,4$.

13. We eliminate from $\mathcal{H}'$ all edges with three and more vertice as in 12. Two-element edges may intersect only in the very last vertex n'. Thus $H_e(\mathcal{F},n) \le n + (n-1)$, attained by $\mathcal{H} = (\{i\},\{n\},\{i,n\} : i \in [n-1])$. As for $H_i(\mathcal{F},n)$, let $n \ge 3$ and v be the first vertex with $\deg(v) \ge 3$ (if v does not exist, $i(\mathcal{H}) \le 2n$). If $\deg(v) = 3$, $i(\mathcal{H}) \le 3n-1$ because $\deg(w) \le 3$ for every $w > v$ and $3n$ cannot be attained. If $\deg(v) > 3$, necessarily $v = n$ and $\deg(w) \le 2$ for $w < n$. Hence $H_i(\mathcal{F},n) \le 2(n-1) + 1 + (n-1) + \lfloor \frac{n-1}{2} \rfloor$, attained by $\mathcal{H} = (\{i,n\},\{2j-1,2j,n\},\{n\} : \ i \in [n-1], j \in [\lfloor \frac{n-1}{2} \rfloor])$ for odd

$n \geq 3$ and the same $\mathcal{H}$ plus $\{n-1\}$ for even $n \geq 4$.

14. Recall that $\mathcal{H}'$ is full. Besides singletons it may have at most one other edge and $H_e(\mathcal{F}, n) \leq n+1$, which is easy to attain; $H_e(\mathcal{F}, 1) = 1$. To determine $H_i(\mathcal{F}, n)$, notice that, in $\mathcal{H}$, $\deg(w) \geq 4$ implies that $\deg(v) = 1$ for every other vertex v. Then $i(\mathcal{H}) \leq 2n-1$. Otherwise $\deg(w) \leq 3$ for every w and $i(\mathcal{H}) \leq 3n$. Since $\deg(w) = \deg(v) = 3$ implies that w and v lie in the same three edges, weights $3n$ and $3n-1$ cannot be attained but $3n-2$ can, by $\mathcal{H} = ([n], [n-1], [2, n])$.

15. $\mathcal{H}'$ has no edges H with $|H| > 2$ and may have only one two-element edge, $\{n'-1, n'\}$. Thus, for $n > 1$, $H_e(\mathcal{F}, n) \leq n+1$, which is easy to attain; $H_e(\mathcal{F}, 1) = 1$. $\mathcal{H} = ([n], [n-1], [2, n])$ shows that $H_i(\mathcal{F}, n) \geq 3n-2$. We prove the opposite inequality by considering $\deg(1)$ in a general $\mathcal{H}$. Case $\deg(1) \geq 4$ is impossible because it implies that $\mathcal{H} \supset \mathcal{F}$. So does $\deg(1) = 3$ if an $H \in \mathcal{H}$ exists with $1 \notin H$. Thus $\deg(1) = 3$ implies that $e(\mathcal{H}) = 3$ and $i(\mathcal{H}) \leq 3n-2$. If $\deg(1) = 2$, we delete the two edges containing 1 from $\mathcal{H}$ and obtain $i(\mathcal{H}) \leq n + (n-1) + (n-1) = 3n-2$ because the resulting hypergraph does not contain $\mathcal{F}_3$. If $\deg(1) = 1$, we proceed by induction on $v(\mathcal{H})$. Let $H \in \mathcal{H}$ with $1 \in H$. If $H_1 = H \backslash \{1\} \notin \mathcal{H}$, we delete 1 (simplicity is preserved) and use induction. If $H_1 \in \mathcal{H}$ and $|H_1| \leq 2$, we delete 1 and H_1 and use induction. If $H_1 \in \mathcal{H}$ and $|H_1| \geq 3$, let $1, u, v$, and w be the first four vertices of H (in this order). If both sets $H_2 = H_1 \backslash \{u\}$ and $H_3 = H_1 \backslash \{v\}$ are edges of $\mathcal{H}$, we have $\mathcal{H} \supset \mathcal{F}$ since $u \in H \cap H_1$, $v \in H_2$, and $w \in H_3$. Hence one of the sets, say H_2, is not an edge and we can again use induction, deleting 1 from $\mathcal{H}$ and u from H_1.

16. $H_e(\mathcal{F}, n)$ is handled similarly to 15. $\mathcal{H} = ([n], [n-1], [2, n])$ shows that $H_i(\mathcal{F}, n) \geq 3n-2$. We prove the opposite inequality. Let $n \geq 3$ and let $u \in [2, n-1]$ have the maximum degree in $\mathcal{H}$ among the vertices in $[2, n-2]$. If $\deg(u) = 1$, then $i(\mathcal{H}) \leq \deg(1) + \deg(n) + n - 2 \leq n + n + n - 2 = 3n-2$. If $\deg(u) \geq 4$, then $\mathcal{H} \supset \mathcal{F}$, which is a contradiction. The same holds if $\deg(u) = 3$ and an edge H exists with $u \notin H$. Thus $\deg(u) = 3$ implies $e(\mathcal{H}) = 3$ and $i(\mathcal{H}) \leq 3n-2$. Let $\deg(u) = 2$. If $\deg(1) \geq 3$ and $\deg(n) \geq 3$, we have again $\mathcal{H} \supset \mathcal{F}$ or $e(\mathcal{H}) = 3$. Thus, say $\deg(1) \leq 2$. If $\deg(1) = 1$, we delete the edge containing 1 and obtain $i(\mathcal{H}) \leq n + 2(n-1) - 1 = 3n-3$ because the rest of $\mathcal{H}$ does not contain $\mathcal{F}_6$. If $\deg(1) = 2$, we delete $H \in \mathcal{H}$ such that $1 \in H$ and $|H| \leq n-1$. The rest again does not contain $\mathcal{F}_6$ and thus $i(\mathcal{H}) \leq n - 1 + 2n - 1 = 3n-2$.

17. Particular case of Theorems 2.1 and 2.3; $H_i(\mathcal{F}, 3) = 12$.

18. In $\mathcal{H}'$, two nonsingleton edges may intersect only in the common last

vertex, which implies that $e(\mathcal{H}_1) \le 2v(\mathcal{H}_1) - 1$ holds for every component $\mathcal{H}_1$ of $\mathcal{H}'$. Hence $H_e(\mathcal{F}, n) \le 2n - 1$, attained by $\mathcal{H} = (\{i\}, \{i, n\}, \{n\} : \ i \in [n-1])$.

As for $H_i(\mathcal{F}, n)$, consider an $\mathcal{H}$ with $\bigcup \mathcal{H} = [n]$. Since $\mathcal{H} \not\supset \mathcal{F}$, $\deg(1) \le 2$. We delete 1 from $\mathcal{H}$ and obtain $\mathcal{H}_1$. $\mathcal{H}_1$ has at most two duplicated edges. Let $H_1 = H_2$ be one of the duplications. If $|H_1| = 1$, we delete H_1 from $\mathcal{H}_1$. If $|H_1| \ge 2$, we delete from H_1 its last vertex. This creates no new duplication (else $\mathcal{H} \supset \mathcal{F}$). In this way we remove from $\mathcal{H}_1$ both possible duplications and obtain a simple $\mathcal{H}_2$ with $\bigcup \mathcal{H}_2 = [2, n]$ and $i(\mathcal{H}) \le 4 + i(\mathcal{H}_2)$. We have the inductive inequality $i(\mathcal{H}) \le 4 + H_i(\mathcal{F}, n-1)$. Note that $\deg(2) \le 2$ and thus for induction we may as well delete 2 instead of 1 and that if one of $\{1\}$, $\{2\}$, and $\{1, 2\}$ is an edge of $\mathcal{H}$, we obtain the streghtening $i(\mathcal{H}) \le 3 + H_i(\mathcal{F}, n-1)$. Note also that $\deg(v) \ge 3$ implies that v is the last vertex of every $H \in \mathcal{H}, v \in H$.

We prove that for $n = 1, 2, 3, 4, 5, 6$ we have $H_i(\mathcal{F}, n) = 1, 4, 8, 11, 15, 18$ and that $H_i(\mathcal{F}, n) = 4n - 6$ for $n \ge 6$. The first two values are trivial. By the inductive inequality, $H_i(\mathcal{F}, 3) \le 4 + 4 = 8$. Weight 8 is attained by $\mathcal{H} = (\{3\}, \{1, 3\}, \{2, 3\}, [3])$. Let $n = 4$ and $\bigcup \mathcal{H} = [4]$. Clearly, $\deg(1), \deg(2) \le 2$. Let first $\deg(3) \ge 3$ and p be the number of edges intersecting both $[2]$ and $[3, 4]$. Clearly, $p \le 2 \cdot 2$. Since no edge can contain both 3 and 4, $\deg(3) + \deg(4) \le p + 2 \le 6$ and $i(\mathcal{H}) = \sum_1^4 \deg(i) \le 2 \cdot 2 + 6 = 10$. If $\deg(3) \le 2$, let p be the number of edges $H \in \mathcal{H}$ such that $4 \in H$ and $H \cap [3] \ne \emptyset$. Then $p \le H_e(\mathcal{F}_5, 3) = 4$, $\deg(4) \le 1 + p \le 5$, and $i(\mathcal{H}) = \sum_1^4 \deg(i) \le 3 \cdot 2 + 5 = 11$. Weight 11 is attained by $\mathcal{H} = (\{4\}, \{i, 4\}, [4] : \ i \in [3])$. Thus $H_i(\mathcal{F}, 4) = 11$. By the inductive inequality, $H_i(\mathcal{F}, 5) \le 4 + 11 = 15$ and weight 15 is attained by $\mathcal{H} = (\{5\}, \{i, 5\}, \{2j-1, 2j, 5\} : \ i \in [4], j \in [2])$.

It remains to show that $H(\mathcal{F}, 6) = 18$ and not $4 + 15 = 19$. $H(\mathcal{F}, 6) \ge 18$ due to $\mathcal{H} = (\{6\}, \{i, 6\}, \{1, 2, 6\}, \{3, 4, 5, 6\} : \ i \in [5])$. We elaborate the argument for $n = 4$. Let $\bigcup \mathcal{H} = [6]$. Clearly, $\deg(1), \deg(2) \le 2$ and $\deg(3) \le 4$. If $\deg(3) = 4$, no edge intersects both $[3]$ and $[4, 6]$ and $i(\mathcal{H}) \le 2H_i(\mathcal{F}, 3) = 16$. If $\deg(3) = 3$, we delete 3 from $\mathcal{H}$. If this creates a duplication, one of $\{1\}$, $\{2\}$, and $\{1, 2\}$ is an edge of $\mathcal{H}$ and by the above remark $i(\mathcal{H}) \le 3 + H_i(\mathcal{F}, 5) = 18$. If no duplication arises, again $i(\mathcal{H}) \le \deg(3) + H_i(\mathcal{F}, 5) = 18$. So $\deg(3) \le 2$. Let $k = \deg(4)$. Let first $k \ge 3$ and p be the number of edges intersecting both $[4]$ and $[5, 6]$ (none of them contains 4). The edges for which 4 is the last vertex contribute by at least $k - 1$ to $\deg(1) + \deg(2) + \deg(3) \le 6$ and thus $k \le 7$ and $p \le 6 - (k-1) = 7 - k$. If $\deg(5) \ge 3$, $\deg(5) + \deg(6) \le p + 2 \le 9 - k$ (no edge contains both 5 and 6) and $i(\mathcal{H}) = \sum_1^6 \deg(i) \le$

$3 \cdot 2 + k + 9 - k = 15$. If $\deg(5) \le 2$, we have $\deg(6) \le 2 + p \le 9 - k$ and $i(\mathcal{H}) \le 4 \cdot 2 + k + 9 - k = 17$. Thus $k = \deg(4) \le 2$ and we have $\deg(i) \le 2$ for every $i \in [4]$. If $\deg(5) \ge 3$ we set again p to be the number of $H \in \mathcal{H}$ intersecting both $[4]$ and $[5,6]$. We have $p \le 4 \cdot 2 = 8$ and $\deg(5) + \deg(6) \le p + 2 \le 10$. Thus $i(\mathcal{H}) = \sum_1^6 \deg(i) \le 4 \cdot 2 + 10 = 18$. If $\deg(5) \le 2$, let p be the number of $H \in \mathcal{H}$ intersecting $[5]$ and containing 6. Then $p \le H_e(\mathcal{F}_5, 5) = 7$ and $\deg(6) \le 1 + p \le 8$. We have again $i(\mathcal{H}) \le 5 \cdot 2 + 8 = 18$. Thus $H_i(\mathcal{F}, 6) = 18$.

Finally, using induction starting at $n = 6$ and the inductive inequality we see that for $n \ge 6$ we have $H_i(\mathcal{F}, n) \le 4n - 6$. The opposite inequality is proved by the hypergraph $\mathcal{H} = (\{i, n-1\}, \{i, n\}, \{n-1\}, \{n\} : i \in [n-2])$.

19. Let v be the first vertex in $\mathcal{H}'$ with $\deg(v) \ge 2$. If $\deg(v) = 2$, $\mathcal{H}'$ has at most one nonsingleton edge and $e(\mathcal{H}') \le n + 1$. If $\deg(v) > 2$, every nonsingleton edge has two vertices and starts in v. Thus $H_e(\mathcal{F}, n) \le 2n - 1$, attained by $\mathcal{H} = (\{1\}, \{1, i\}, \{i\} : i \in [2, n])$. This hypergraph shows that $H_i(\mathcal{F}, n) \ge 3n - 2$. To prove the opposite inequality, we take a general $\mathcal{H}$ and argue as in 15. If $\deg(1) \ge 3$, $|H| \le 2$ for every edge of $\mathcal{H}$ and $|H| = 2$ implies $1 \in H$. Thus $i(\mathcal{H}) \le 3n - 2$. If $\deg(1) = 2$, we delete the two edges containing 1. Since the rest does not contain $\mathcal{F}_4$, we have $i(\mathcal{H}) \le n + (n-1) + (n-1) = 3n - 2$. If $\deg(1) = 1$, let H and H_1 be given by $1 \in H \in \mathcal{H}$, and $H_1 = H \backslash \{1\}$. If $H_1 \notin \mathcal{H}$ or $|H_1| \le 2$, we delete 1 and, if necessary, H_1, and use induction. If H_1 is an edge and $|H_1| \ge 3$, then $H_2 = H_1 \backslash \{u\}$, where $u = \min H_1$, is not an edge (else $\mathcal{H} \supset \mathcal{F}$). We delete 1 from $\mathcal{H}$ and u from H_1 and use induction.

20. In $\mathcal{H}'$, for every two nonsingleton edges $H_1 \ne H_2$ we have $H_1 \le H_2$ or $H_1 \ge H_2$. ($H_1 \le H_2$ means that $x \le y$ for every $x \in H_1, y \in H_2$.) Therefore $\mathcal{H}'$ has at most $n - 1$ such edges. $H_e(\mathcal{F}, n) \le 2n - 1$, attained by $(\{i, i+1\}, \{i\}, \{n\} : i \in [n-1])$. This hypergraph shows also that $H_i(\mathcal{F}, n) \ge 3n - 2$. We prove the opposite inequality by induction. Let $\mathcal{H}$ have $\bigcup \mathcal{H} = [n]$ with $n \ge 3$. If $\deg(v) = 1$ for every $v \in [2, n-1]$ then $i(\mathcal{H}) = \deg(1) + \deg(n) + n - 2 \le 3n - 2$. If $\deg(v) \ge 3$ for some $v \in [2, n-1]$, we split $\mathcal{H}$ into $\mathcal{H}_1$ and $\mathcal{H}_2$ where $\mathcal{H}_1$ takes the edges of $\mathcal{H}$ lying to the left of v, $\mathcal{H}_2$ takes those lying to the right, and if $\{v\} \in \mathcal{H}$ then $\{v\} \in \mathcal{H}_1$; no edge lies on both sides of v because $\mathcal{H} \not\supset \mathcal{F}$. We have $v(\mathcal{H}_1) + v(\mathcal{H}_2) \le n + 1$. If $v(\mathcal{H}_1) + v(\mathcal{H}_2) \le n$, then by induction $i(\mathcal{H}) = i(\mathcal{H}_1) + i(\mathcal{H}_2) \le 3v(\mathcal{H}_1) - 2 + 3v(\mathcal{H}_2) - 2 \le 3n - 4$. If $v(\mathcal{H}_1) + v(\mathcal{H}_2) = n + 1$, we note that $i(\mathcal{H}_2) \le 3v(\mathcal{H}_2) - 3$ because now $v = \min \bigcup \mathcal{H}_2$ and $\{v\} \notin \mathcal{H}_2$. Again by induction $i(\mathcal{H}) = i(\mathcal{H}_1) + i(\mathcal{H}_2) \le 3n - 2$. The last case is if

$\deg(v) = 2$ for some $v \in [2, n-1]$. Let H_1 and H_2 be the edges containing v. If no edge jumps over v we split $\mathcal{H}$ and proceed as before. Else we have, say, $\min H_1 < v < \max H_1$. We delete v from $\mathcal{H}$. Since $H_1 \backslash \{v\} \notin \mathcal{H}$, the only duplication that may arise is when $v = \min H_2$ (case $v = \max H_2$ is similar) and $H_3 = H_2 \backslash \{v\} \in \mathcal{H}$. We cancel this duplication by deleting from H_3 its last vertex. No new duplication then arises ($\mathcal{H} \not\supset \mathcal{F}$) and we have by induction that $i(\mathcal{H}) \leq 3 + 3(n-1) - 2 = 3n - 2$.

21. $H_e(\mathcal{F}, n) \leq 2n - 1$ follows from the fact that, in $\mathcal{H}'$, $|H| \leq 2$ for every edge and $|H| = 2$ implies $1 \in H$. The bound is attained by $\mathcal{H} = (\{1\}, \{i\}, \{1, i\} : \ i \in [2, n])$. This hypergraph shows also that $H_i(\mathcal{F}, n) \geq 3n - 2$. To prove the opposite inequality, consider $\deg(1)$ in a general $\mathcal{H}$. If $\deg(1) = 1$, delete the edge containing 1. The rest does not contain $\mathcal{F}_8$ and thus $i(\mathcal{H}) \leq n + 2(n-1) - 1 = 3n - 3$. If $\deg(1) = 2$, delete an edge H such that $|H| \leq n - 1$ and $1 \in H$. The rest does not contain $\mathcal{F}_{24}$ ($\mathcal{F}_8$ would do here but not in the next argument 22), so $i(\mathcal{H}) \leq n - 1 + 2n - 1 = 3n - 2$. If $\deg(1) \geq 3$, we delete 1 from $\mathcal{H}$. In the resulting hypergraph $\mathcal{H}_0$ only singletons may be duplicated and every component $\mathcal{H}_1$ of $\mathcal{H}_0$ satisfies $i(\mathcal{H}_1) \leq 2v(\mathcal{H}_1)$ since the only intersection in $\mathcal{H}_1$ is the common last vertex v (and $\{v\}$ may be duplicated). Thus $i(\mathcal{H}_0) \leq 2(n-1)$. $(H \backslash \{1\} : \ 1 \in H \in \mathcal{H}, H \neq \{1\})$ is a simple and $\mathcal{F}_{24}$-free, even $\mathcal{F}_8$-free, hypergraph. Hence $\deg(1) \leq 1 + H_e(\mathcal{F}_{24}, n-1) = n$. In total, $i(\mathcal{H}) \leq n + 2(n-1) = 3n - 2$.

22. The arguments are very similar to those in 21.

23. $\mathcal{H}'$ has no edge H with $|H| \geq 3$ and no two-element edge skipping one or more vertices. Again $H_e(\mathcal{F}, n) \leq 2n - 1$, attained by the same hypergraph as in 20. This hypergraph shows also that $H_i(\mathcal{F}, n) \geq 3n - 2$. We prove the opposite inequality by induction on $v(\mathcal{H}) = n$. It is easy to check that $\deg(1) \geq 3$ implies $\mathcal{H} \supset \mathcal{F}$. Let $\deg(1) = 2$. The first case is when $|H| \neq 2$ for both edges containing 1. Deletion of 1 from $\mathcal{H}$ gives then a simple hypergraph and we have $i(\mathcal{H}) \leq 2 + 3(n-1) - 2 = 3n - 3$. If $|H| = 2$ for exactly one of them, we set $H_1 = H$, and if both have two elements, we set H_1 to be the longer one. Deletion of H_1 from $\mathcal{H}$ and 1 from the rest gives a simple hypergraph and $i(\mathcal{H}) \leq 2 + 1 + 3(n-1) - 2 = 3n - 2$. Let now $\deg(1) = 1$ and $1 \in H \in \mathcal{H}$. If $|H| \leq 3$, we delete H and use induction. Let $|H| \geq 4$. If $H_1 = H \backslash \{1\}$ is not an edge, we delete 1 from $\mathcal{H}$ and use induction. If $H_1 \in \mathcal{H}$, let $u = \min H_1$. Clearly, $H_1 \backslash \{u\}$ is not an edge (else $\mathcal{H} \supset \mathcal{F}$). We delete 1 from $\mathcal{H}$ and u from H_1 and use induction.

24. As in 6, $\mathcal{H}'$ has no nonsingleton edge and thus $H_e(\mathcal{F}, n) = n$. As for weights, notice that every component $\mathcal{H}_1$ of $\mathcal{H}$ either consists of at most two

edges or the only intersection in $\mathcal{H}_1$ is one vertex common to all edges. Both cases give bound $i(\mathcal{H}_1) \le 2v(\mathcal{H}_1) - 1$ and thus $H_i(\mathcal{F}, n) \le 2n - 1$, attained by $\mathcal{H} = (\{i, n\}, \{n\} : i \in [n-1])$.

25. We remark that in 25–28 $H_e(\mathcal{F}, 1) = H_i(\mathcal{F}, 1) = 1$. $\mathcal{H}'$ has no edge H with $|H| \ge 4$, every three-element edge must contain 1 and 2, and two-element edges must start in 1 or in 2. Thus, for $n > 1$, $H_e(\mathcal{F}, n) \le n+(n-1)+2(n-2)$, attained by the hypergraph $\mathcal{H}^* = (\{1\}, \{i\}, \{1, i\}, \{2, j\}, \{1, 2, j\} : i \in [2, n], j \in [3, n])$. $\mathcal{H}^*$ shows that, for $n > 1$, $H_i(\mathcal{F}, n) \ge 8n - 12$. To prove the opposite inequality, we consider a general $\mathcal{H}$ with $v(\mathcal{H}) \ge 3$. If $|H \cap [3, n]| \le 1$ for every $H \in \mathcal{H}$, $i(\mathcal{H}) \le i(H^*) = 8n-12$. Let $|H \cap [3, n]| \ge 2$ for an edge H. If $\deg(1), \deg(2) \ge 3$ then $\mathcal{H} \supset \mathcal{F}$. So, say, $\deg(2) \le 2$ (case $\deg(1) \le 2$ is similar). We delete from $\mathcal{H}$ the edges containing 2 and observe that the rest avoids $\mathcal{F}_9$. Hence $i(\mathcal{H}) \le n+n-1+3(n-1)-2 = 5n-6 \le 8n-12$ $(n \ge 2)$.

26. $|H| \le 3$ for every edge of $\mathcal{H}'$, allowed three-element edges are $\{1, b, b+1\}$ $(n-2$ edges) and allowed two-element edges are $\{1, b\}$ and $\{b, b+1\}$ $(2n-3$ edges). Thus $H_e(\mathcal{F}, n) \le 4n - 5$ $(n > 1)$ and it is clear which hypergraph attains this value. We show that the same hypergraph attains also the maximum weight $8n-12$. If $\deg(1) \le 2$, we delete from $\mathcal{H}$ the edges containing 1 and conclude, since the rest avoids $\mathcal{F}_{10}$, that $i(\mathcal{H}) \le n+n-1+3(n-1)-2 = 5n-6 \le 8n-12$ $(n \ge 2)$. Let $\deg(1) \ge 3$. We delete 1 from $\mathcal{H}$. Consider two edges H_1 and H_2 of the resulting $\mathcal{H}_1$. $H_1 = H_2$ implies $|H_1| \le 2$ (else $\mathcal{H} \supset \mathcal{F}$) and no edge of $\mathcal{H}_1$ has higher multiplicity than 2. If $H_1 \ne H_2$ and neither H_i is a singleton, then $H_1 \le H_2$ or $H_2 \le H_1$ (else $\mathcal{H} \supset \mathcal{F}$). Thus $i(\mathcal{H}) = \deg(1) + i(\mathcal{H}_1) \le (1 + n - 2 + n - 1) + 2(n - 1 + 2(n-2)) = 8n - 12$.

27. Similar to 25. Only the interval $[3, n]$ is replaced by $[2, n-1]$.

28. In $\mathcal{H}'$ no edge has more than three elements, three-element edges must consist of consecutive vertices, and two-element edges must be of the form $\{a, a+1\}$ and $\{a, a+2\}$. Again $H_e(\mathcal{F}, n) \le n+2(n-2)+n-1 = 4n-5$, which is attained if we take all decribed edges (and singletons). To prove $H_i(\mathcal{F}, n) \le 8n - 12$, which is attained by the same hypergraph, we show that other edges can be eliminated using induction on n. If an $H \in \mathcal{H}$ exists with $|H| \ge 4$, let $u, v \in H$ be two distinct vertices, none of them the end of H. If $\deg(u) = \deg(v) = 1$, u or v may be deleted from $\mathcal{H}$ (one of these deletions does not create duplication) and induction applies. If $\deg(u) \ge 2$ and $\deg(v) = 1$, we can delete v from $\mathcal{H}$ unless $\deg(u) = 2$ and $H \setminus \{v\} \in \mathcal{H}$. But then we can delete u from $\mathcal{H}$. Similarly if $\deg(u) = 1$ and $\deg(v) \ge 2$. If $\deg(u) \ge 2$ and $\deg(v) \ge 2$, both inequalities must be equalities and u, v lie in

the same two edges. Either of u and v can be deleted and induction applies. Thus we can assume that $|H| \le 3$ for every $H \in \mathcal{H}$. If $H = \{a, b, c\}_< \in \mathcal{H}$ and $c > b + 1$ (case $a < b - 1$ is similar), then $\deg(b) \le 2$ and $\deg(b) = 2$ implies that b and $b + 1$ lie in the same edge. It is easy to see that b can be deleted. We may assume that every edge H with $|H| = 3$ is of the form $H = \{a, a + 1, a + 2\}$. Finally, if $\{a, b\} \in \mathcal{H}$ and $a < b - 2$, it is clear that $b - 2$ and $b - 1$ have degree 1 and lie in the same edge. Either one of them can be deleted. We can assume that $\{a, b\}_< \in \mathcal{H}$ implies $b \le a + 2$.

29. We delete the last vertex from every $H \in \mathcal{H}$, $|H| \ge 2$. The resulting sets are mutually disjoint and lie in $[n-1]$. Thus $H_e(\mathcal{F}, n) \le n+(n-1)$ and $H_i(\mathcal{F}, n) \le n+(n-1)+(n-1)$, attained by $\mathcal{H} = (\{i\}, \{n\}, \{i, n\} : i \in [n-1])$.

30. If $H \in \mathcal{H}$ with $|H| \ge 3$, we replace H by the two-element set of the first two vertices of H. Thus, for bounding $H_e(\mathcal{F}, n)$ from above, we may assume that $|H| \le 2$ for every edge. It is clear that two-element edges form a triangle-free graph on at most n vertices. By a special case of Turán's theorem (see [16, Problem 10.30]), it has at most $\lfloor \frac{n^2}{4} \rfloor$ edges. The value of $H_e(\mathcal{F}, n)$ is attained by $(\{i\}, \{j, k\} : i \in [n], j \in [\lfloor n/2 \rfloor], k \in [\lfloor n/2 \rfloor + 1, n])$. We show that the maximum weight is attained by the same hypergraph with the exception $n = 3$ when $H_i(\mathcal{F}, 3) = 8$ (and not 7). Large edges $H = \{a_1, a_2, \ldots, a_t\}_<$ with $t \ge 4$ are eliminated by the replacement $H \to \{a_1, a_{t-1}\}, \{a_2, a_{t-1}\}, \ldots, \{a_{t-2}, a_{t-1}\}$. If $t = 3$ and $a_3 < n$, we eliminate H by $H \to \{a_2, a_3\}, \{a_2, n\}$. Similarly if $1 < a_1$. Let k be the number of the troublesome edges $\{1, a, n\}$. No two-element edge is incident with any of the as and they form a triangle-free graph on at most $n - k$ vertices. By Turán's theorem, $H_i(\mathcal{F}, n) \le n + 2\lfloor \frac{(n-k)^2}{4} \rfloor + 3k$ where the bound is attained. For $n \ge 4$ this is maximized for $k = 0$ (and $k = 2$ for $n = 4$; $H_i(\mathcal{F}, 4) = 12$ is attained by $(\{i\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\})$ and $(\{i\}, \{1, 4\}, \{1, 2, 4\}, \{1, 3, 4\})$ where $i \in [4]$) and for $n = 3$ by $k = 1$. Indeed, $(\{i\}, \{1, 3\}, \{1, 2, 3\})$ is better than $(\{i\}, \{1, 2\}, \{1, 3\})$ where $i \in [3]$.

31. No two distinct edges of $\mathcal{H}$ intersect in two or more vertices. Hence every $H \in \mathcal{H}$ with $|H| \ge 3$ may be replaced by its two-element subsets; this works for both size and weight. Therefore $H_e(\mathcal{F}, n) = \binom{n}{1} + \binom{n}{2}$ and $H_i(\mathcal{F}, n) = \binom{n}{1} + 2\binom{n}{2}$, as in 11.

32. As for $H_e(\mathcal{F}, n)$, we eliminate from $\mathcal{H}$ all edges with $|H| \ge 4$ by replacing H by the two-set of its first two elements. So $|H| \le 3$ for every $H \in \mathcal{H}$. Let $a + 1$ be the first vertex that is the last point of a two-element edge or the middle point of a three-element edge. $\mathcal{H}$ consists of singletons,

of a bipartite graph with parts $[a]$ and $[a+2,n]$, and of edges of the form $\{b,a+1\}$, $\{a+1,c\}$, and $\{b,a+1,c\}$ where $b\in[a], c\in[a+2,n]$; other edges would create $\mathcal{F}$ or they would contradict the minimality of $a+1$. We see that $H_e(\mathcal{F},n)\le n+2a(n-1-a)+(n-1)$, which is attained and maximized by $a=\lfloor(n-1)/2\rfloor$. The same hypergraph attains the maximum weight because large edges $H=\{a_1,a_2,\dots,a_t\}_<$ can be eliminated by the replacement $H\to\{a_1,a_2\},\{a_1,a_2,a_3\}$ if $t=4,5$ and by $H\to\{a_1,a_2\},\{a_1,a_3\},\dots,\{a_1,a_{t-2}\}$ if $t\ge 6$. Counting the weight instead of size, we obtain the second formula.

33. First, we bound the number of two-element edges in $\mathcal{H}$. Let $\mathcal{H}=\mathcal{G}$ be a ($\mathcal{F}$-free) graph with the vertex set $[n]$. The sets $X_i=\{x\in[i+1,n]: \{i,x\}\in\mathcal{G}\}$, $i\in[n-1]$, are subsets of $[2,n]$ and ($\mathcal{G}\not\supset\mathcal{F}$) $\max X_i\le\min X_{i+1}$. Thus $e(\mathcal{G})=\sum_{i=1}^{n-1}|X_i|\le n-1+n-2=2n-3$. Hence $\mathcal{H}$ has at most $2n-3$ two-element edges. We delete from every $H\in\mathcal{H}$, $|H|\ge 3$, its first and last vertex. If two of the resulting sets intersect, we have two distinct edges $H_1,H_2\in\mathcal{H}$ and five not necessarily distinct vertices $u_1,u_2<v<w_1,w_2$ such that $\{u_1,v,w_1\}\subset H_1$ and $\{u_2,v,w_2\}\subset H_2$. Moreover, we can assume that $u_1\ne u_2$ or $w_1\ne w_2$ because $H_1\ne H_2$. But this implies $\mathcal{H}\supset\mathcal{F}$. Thus the resulting sets, subsets of $[2,n-1]$, are mutually disjoint. We conclude that $e(\mathcal{H})\le n+2n-3+n-2=4n-5$ and $i(\mathcal{H})\le n+2(2n-3)+3(n-2)=8n-12$. These bounds are attained by $\mathcal{H}=(\{1\},\{n\},\{1,n\},\{1,i\},\{i,n\},\{1,i,n\}: i\in[2,n-1])$.

34. It is easily checked that the argument bounding the number of edges with more than 2 elements works here as well. We prove by induction on n that the number of two-element edges is again at most $2n-3$. Let $\mathcal{H}=\mathcal{G}$ be a ($\mathcal{F}$-free) graph with the vertex set $[n]$. If $\deg(1)=1$, we have by induction that $e(\mathcal{G})\le 1+2n-5=2n-4$. For $\deg(1)>1$, if $\{1,n\}\in\mathcal{G}$ let m be the second largest neighbour of 1 and if $\{1,n\}\notin\mathcal{G}$ let m be the largest neighbour of 1. Clearly, $m<n$ and every edge of $\mathcal{G}$, except possibly only $\{1,n\}$, lies either in $[m]$ or in $[m+1,n]$. By induction, $e(\mathcal{G})\le 1+2m-3+2(n-m+1)-3=2n-3$. Thus again $e(\mathcal{H})\le 4n-5$ and $i(\mathcal{H})\le 8n-12$. The extremal hypergraph is, for example, $\mathcal{H}=(\{1\},\{n\},\{1,n\},\{i\},\{1,i\},\{i,i+1\},\{1,i,i+1\}:\ i\in[2,n-1])$.

35. We have $|H|\le 2$ for every $H\in\mathcal{H}'$. Thus $H_e(\mathcal{F},n)=\binom{n}{1}+\binom{n}{2}$. As for the weight, if $H\in\mathcal{H}$ with $|H|\ge 3$, we replace H by the two-element sets $\{a,b\}$ where $a=\min H$ and $a<b\in H$. Thus we may suppose that $|H|\le 2$ for every $H\in\mathcal{H}$ and we conclude that $H_i(\mathcal{F},n)=\binom{n}{1}+2\binom{n}{2}$.

36. Same argument as in 35.

37. In $\mathcal{H}'$, $|H| \le 3$ for every edge and $|H| = 3$ implies $1 \in H$. Thus $H_e(\mathcal{F}, n) = \binom{n}{1} + \binom{n}{2} + \binom{n-1}{2}$. As for the weight, we get rid of all H with $|H| \ge 4$ by the same replacements as in 35. If H with $|H| = 3$ is present, again $1 \in H$. Thus $H_i(\mathcal{F}, n) = \binom{n}{1} + 2\binom{n}{2} + 3\binom{n-1}{2}$.

38. Same argument as in 37. Allowed three-element edges are now $H = \{a, a+1, b\}_<$ and we have again $1 + 2 + \cdots + (n-2) = \binom{n-1}{2}$ of these.

39. Clearly, $H_e(\mathcal{F}, n) = \binom{n}{1} + \binom{n}{2} + \binom{n}{3}$ and $H_i(\mathcal{F}, n) = \binom{n}{1} + 2\binom{n}{2} + 3\binom{n}{3}$. □

We do not have 110 distinct extremal functions and not even close to 78. Hypergraphs $\mathcal{F}$ with $1 \le i(\mathcal{F}) \le 4$ have 28 distinct extremal functions $H_e(\mathcal{F}, n)$ and $H_i(\mathcal{F}, n)$ (included the "undefined function"). Of these 25 differ for infinitely many arguments. The formulas for $H_e(\mathcal{F}, n)$ and $H_i(\mathcal{F}, n)$ hold for $n \ge v(\mathcal{F})$ with the exception of $\mathcal{F}_5, \mathcal{F}_{12}$, $\mathcal{F}_{18}$, and $\mathcal{F}_{30}$ but only the initial values of $H_i(\mathcal{F}_{18}, n)$ caused some troubles. We conclude this section by a nice geometric derivation of the formula for $H_e(\mathcal{F}_{34}, n)$ (crossing pattern) due to Attila Pór. Put the vertices $1, 2, \ldots, n$ in this order clockwise on a circle in the plane and consider the convex hulls $C_i = \mathrm{conv}(H_i)$, $H_i \in \mathcal{H}$. The condition $\mathcal{H} \not\supset \mathcal{F}_{34}$ is equivalent to the condition that the relative interiors of C_i do not intersect. So it is clear that we may have at most $n-2$ edges H with $|H| \ge 3$, maximized by the triangulations, and at most $3n - 6 - (n-3) = 2n - 3$ two-element edges because these form a planar graph with a big outer face. Thus $H_e(\mathcal{F}_{34}, n) \le n - 2 + 2n - 3 + n = 4n - 5$.

4 Enumerative intermezzo

Besides the extremal problems for $\mathcal{F}$-free hypergraphs there is also the enumerative problem to count them. Let

$$h_n^{(v)}(\mathcal{F}) = |\{\mathcal{H} : \mathcal{H} \text{ is simple } \& \ \mathcal{H} \not\supset \mathcal{F} \ \& \ \bigcup \mathcal{H} = [n]\}|$$

be the number of simple nonisomorphic $\mathcal{F}$-free hypergraphs $\mathcal{H}$ with $v(\mathcal{H}) = n$. Let $h_n^{(i,s)}(\mathcal{F})$ and $h_n^{(i)}(\mathcal{F})$ be the analogous counting functions with $i(\mathcal{H}) = n$ in the place of $v(\mathcal{H}) = n$ and with the simplicity of $\mathcal{H}$ dropped in $h_n^{(i)}(\mathcal{F})$. For example, for $\mathcal{F}_2 = (\{1\}_1, \{1\}_2)$ all three counting functions equal to the nth Bell number B_n that counts partitions of $[n]$.

The enumerative problem to determine or to bound, for $\mathcal{F}$ fixed and $n \to \infty$, the three counting functions is already for $i(\mathcal{F}) \leq 4$ much more difficult than the extremal problem. In Klazar [15] we found the ordinary generating functions $F_1(x)$, $F_2(x)$, and $F_3(x)$ of $h_n^{(v)}(\mathcal{F})$, $h_n^{(i,s)}(\mathcal{F})$, and $h_n^{(i)}(\mathcal{F})$, respectively, for the crossing pattern $\mathcal{F}_{34} = (\{1,3\},\{2,4\})$. F_1, F_2, and F_3 are algebraic over $\mathbf{Z}(x)$ of degrees 3, 4, and 4, respectively, and their coefficients grow roughly like $(63.97055\ldots)^n$, $(5.79950\ldots)^n$, and $(6.06688\ldots)^n$ where the bases of the exponentials are algebraic numbers of degrees 4, 15, and 23, respectively. We did not succeed in enumerating $\mathcal{F}_{33}$-free hypergraphs where $\mathcal{F}_{33} = (\{1,4\},\{2,3\})$ and we believe it is a problem that deserves interest.

In this article we drop the condition of $\mathcal{F}$-freeness and we determine the total numbers $h_n^{(v)}$, $h_n^{(i,s)}$, and $h_n^{(i)}$, that is, the number of simple $\mathcal{H}$ with $v(\mathcal{H}) = n$, the number of simple $\mathcal{H}$ with $i(\mathcal{H}) = n$, and the number of all $\mathcal{H}$ with $i(\mathcal{H}) = n$. The numbers $h_n^{(v)}$ have been already investigated before, in the slightly different terminology of set covers, but the remaining two problems seem new. We review the known formulas for $h_n^{(v)}$, derive a new recurrence for them, and then we proceed to $h_n^{(i,s)}$ and $h_n^{(i)}$.

Write s_n for the number of simple set systems on $[n]$, which are (possibly empty) sets of nonempty subsets of $[n]$. Clearly, $s_n = 2^{2^n-1}$ and

$$s_n = 2^{2^n-1} = \sum_{j=0}^{n} \binom{n}{j} h_j^{(v)} \tag{1}$$

because set systems 1-1 correspond to simple $\mathcal{H}$ with $\bigcup \mathcal{H} \subset [n]$. Hence we can easily calculate $h_n^{(v)}$ starting by $h_0^{(v)} = 1$ and continuing by the recurrence

$$h_n^{(v)} = 2^{2^n-1} - \sum_{j=0}^{n-1} \binom{n}{j} h_j^{(v)} \tag{2}$$

given in Hearne and Wagner [13]. Using exponential generating functions $F(x) = \sum_{n\geq 0} s_n x^n/n!$ and $H(x) = \sum_{n\geq 0} h_n^{(v)} x^n/n!$ we invert relation (1) by noting that it amounts to $F(x) = \mathrm{e}^x H(x)$. Thus $H(x) = \mathrm{e}^{-x} F(x)$ and we have the explicit formula

$$h_n^{(v)} = \sum_{j=0}^{n} (-1)^{n-j} \binom{n}{j} 2^{2^j-1} \tag{3}$$

that can be found in Comtet [4, p. 165] and that was derived independently by Macula [17].

We show that for $n \geq 0$ also

$$h_{n+1}^{(v)} = 2 \sum_{0 \leq k,l \leq n} \frac{h_k^{(v)} h_l^{(v)} n!}{(k+l-n)!(n-k)!(n-l)!} - h_n^{(v)}. \tag{4}$$

(The actual summation range is $\max(k,l) \leq n \leq k+l$.) We take a simple $\mathcal{H}$, $\bigcup \mathcal{H} = [n+1]$, and decompose it into $\mathcal{H}_1$ and $\mathcal{H}_2$ where $\mathcal{H}_1$ consists of the sets $H \backslash \{1\}$ such that $1 \in H \in \mathcal{H}$ (we omit the $\emptyset$ if $\{1\} \in \mathcal{H}$) and $\mathcal{H}_2$ consists of the remaining edges of $\mathcal{H}$. We relabel the vertices so that $\bigcup \mathcal{H}_1 = [k]$ and $\bigcup \mathcal{H}_2 = [l]$. It is clear that $\mathcal{H}_1$ and $\mathcal{H}_2$ are simple and that $k, l \leq n$. To invert the decomposition, we first select two simple $\mathcal{H}_1$ and $\mathcal{H}_2$ of order k and l, which can be done in $h_k^{(v)} h_l^{(v)}$ ways. We unite their vertex sets so that n vertices arise. This can be done in $\binom{n}{k+l-n,n-k,n-l}$ ways by choosing, from n vertices, $k+l-n, n-k$, and $n-l$ vertices lying in $\bigcup \mathcal{H}_1 \cap \bigcup \mathcal{H}_2$, only in $\bigcup \mathcal{H}_2$, and only in $\bigcup \mathcal{H}_1$, respectively. We append to every edge in $\mathcal{H}_1$ a new least vertex $1'$ and obtain a simple $\mathcal{H}$ with $n+1$ vertices. Finally, the possible addition of $\{1'\}$ to $\mathcal{H}$ (we always loose edge $\{1\}$ when decomposing) gives two further options, except for $\mathcal{H}_1 = \emptyset$ when $\{1'\}$ must be always added. This explains the factor 2 and the subtraction of $h_n^{(v)}$ in (4).

By means of any of (2), (3), and (4) one finds that

$$(h_n^{(v)})_{n \geq 1} = (1, 5, 109, 32297, 2147321017, 9223372023970362989, \ldots).$$

This quite quickly growing sequence is entry A003465 of Sloane [22].

We turn to counting hypergraphs, both simple and all, by weight. Inspection of the long table in Section 3 reveals that $(h_n^{(i,s)})_{n \geq 1} = (1, 2, 7, 28, \ldots)$ and $(h_n^{(i)})_{n \geq 1} = (1, 3, 10, 41, \ldots)$. What comes next?

Recall that a partition $\lambda = 1^{a_1} 2^{a_2} \ldots l^{a_l}$ of $n \in \mathbf{N}$, where $a_i \geq 0$ are integers and usually $a_l > 0$, is the decomposition $n = 1 + 1 + \cdots + 1 + 2 + \cdots + 2 + \cdots + l + \cdots + l$ with the part i appearing a_i times (parts i with $a_i = 0$ may be ommited). Thus $\sum i a_i = n$. We write briefly $\lambda \vdash n$. If $\mathcal{H}$ has weight n and a_i edges of cardinality i, the maximum edge cardinality being l, then $\lambda = 1^{a_1} 2^{a_2} \ldots l^{a_l} \vdash n$ and we say that $\mathcal{H}$ has *edge type* λ. We derive formulas for numbers of hypergraphs with a given edge type.

Theorem 4.1 *Let $\lambda = 1^{a_1} 2^{a_2} \ldots l^{a_l} \vdash n$ where $a_l > 0$. The number of simple hypergraphs with weight n and edge type λ is*

$$\sum_{j=l}^{n} \binom{\binom{j}{1}}{a_1} \binom{\binom{j}{2}}{a_2} \cdots \binom{\binom{j}{l}}{a_l} \sum_{m=j}^{n} (-1)^{m-j} \binom{m}{j}$$

and the number of all hypergraphs with weight n and edge type λ is

$$\sum_{j=l}^{n}\binom{\binom{j}{1}+a_1-1}{a_1}\binom{\binom{j}{2}+a_2-1}{a_2}\cdots\binom{\binom{j}{l}+a_l-1}{a_l}\sum_{m=j}^{n}(-1)^{m-j}\binom{m}{j}.$$

Proof. Consider the polynomials

$$W_n = W_n(x_1, x_2, \ldots, x_n) = \sum_{\mathcal{H}}\prod_{i=1}^{n} x_i^{e(i,\mathcal{H})}$$

where we sum over all simple $\mathcal{H}$ with $\bigcup\mathcal{H} = [n]$ and $e(i,\mathcal{H})$ is the number of i-element edges in $\mathcal{H}$. Refining (1) we have

$$\prod_{i=1}^{n}(1+x_i)^{\binom{n}{i}} = \sum_{j=0}^{n}\binom{n}{j}W_j$$

where on the left is a polynomial (analogous to W_n) counting simple set systems on $[n]$ according to the edge cardinalities. In terms of exponential generating functions,

$$\sum_{n\geq 0}\prod_{i=1}^{n}(1+x_i)^{\binom{n}{i}}\cdot\frac{y^n}{n!} = \mathrm{e}^y\cdot\sum_{n\geq 0}\frac{W_n y^n}{n!}. \tag{5}$$

Thus, as in (3),

$$W_n(x_1,\ldots,x_n) = \sum_{j=0}^{n}(-1)^{n-j}\binom{n}{j}\prod_{i=1}^{j}(1+x_i)^{\binom{j}{i}}.$$

The number of simple $\mathcal{H}$ with $i(\mathcal{H}) = n$ and edge type $\lambda = 1^{a_1}2^{a_2}\ldots l^{a_l} \vdash n$ is the coefficient at $x_1^{a_1}\ldots x_l^{a_l}$ in $W_l + W_{l+1} + \cdots + W_n$ which is

$$\sum_{m=l}^{n}\sum_{j=0}^{m}(-1)^{m-j}\binom{m}{j}\prod_{i=1}^{l}\binom{\binom{j}{i}}{a_i} = \sum_{j=l}^{n}\prod_{i=1}^{l}\binom{\binom{j}{i}}{a_i}\sum_{m=j}^{n}(-1)^{m-j}\binom{m}{j}.$$

Derivation of the second formula is almost identical, only W_n becomes a power series and $1+x_i$ is replaced by $(1-x_i)^{-1}$ because now any i-element edge may come in arbitrary many copies. □

We give for illustration the distribution of hypergraphs with weight $n = 6$ according to their edge types (the first entry is the number of simple $\mathcal{H}$ and the second, given only if different, is the number of all $\mathcal{H}$):

6^1	$1^1 5^1$	$2^1 4^1$	$1^2 4^1$	3^2	$1^1 2^1 3^1$	$1^3 3^1$	2^3
1	11	41	41, 50	31, 32	239	63, 120	62, 75

$1^2 2^2$	$1^4 2^1$	1^6
198, 264	41, 160	1, 32

Collecting the numbers over all edge types we obtain formulas for $h_n^{(i,s)}$ and $h_n^{(i)}$.

Corollary 4.2 *The numbers of hypergraphs with weight n, simple and all, are ($\lambda = 1^{a_1} 2^{a_2} \dots l^{a_l}$ with $a_l > 0$)*

$$h_n^{(i,s)} = \sum_{\lambda \vdash n} \sum_{j=l}^{n} \prod_{i=1}^{l} \binom{\binom{j}{i}}{a_i} \sum_{m=j}^{n} (-1)^{m-j} \binom{m}{j} \tag{6}$$

$$h_n^{(i)} = \sum_{\lambda \vdash n} \sum_{j=l}^{n} \prod_{i=1}^{l} \binom{\binom{j}{i} + a_i - 1}{a_i} \sum_{m=j}^{n} (-1)^{m-j} \binom{m}{j}. \tag{7}$$

Using (6), (7), and computer algebra system MAPLE we have found that

$$\begin{aligned} (h_n^{(i,s)})_{n \geq 1} &= (1, 2, 7, 28, 134, 729, 4408, 29256, 210710, 1633107, \dots) \\ (h_n^{(i)})_{n \geq 1} &= (1, 3, 10, 41, 192, 1025, 6087, 39754, 282241, 2159916, \dots). \end{aligned}$$

As of May 2001, these sequences were absent in [22].

From the point of view of complexity theory formulas (6) and (7) are inferior compared to those for $h_n^{(v)}$. While any of (2), (3), and (4) needs only polynomially many (in n) operations to turn the input n into the output $h_n^{(v)}$, (6) and (7) require roughly $n^c p(n)$ operations where $p(n) = |\{\lambda : \lambda \vdash n\}|$. Numbers $p(n)$ grow superpolynomially because by the famous Hardy–Ramanujan–Uspensky asymptotics $p(n) \sim (n \cdot 4\sqrt{3})^{-1} \cdot \exp(\pi\sqrt{2n/3})$ if $n \to \infty$. (An elementary proof was given by Erdős [5] who proved that $p(n) \sim cn^{-1} \cdot \exp(\pi\sqrt{2n/3})$ and by Newman [18] who showed that $c = (4\sqrt{3})^{-1}$. A simpler complex-analytical proof was given later by Newman [19]. See also Newman's book [20, chapter 2].) On the other hand, $p(n)$ is subexponential and thus formulas (6) and (7) are nontrivial in the sense that the numbers of operations which they require are substantially smaller than $h_n^{(i,s)}$ and $h_n^{(i)}$ themselves (obviously $h_n^{(i,s)}, h_n^{(i)} > 2^n$ for $n > 3$). A polynomial algorithm generating $h_n^{(i,s)}$ and $h_n^{(i)}$ can be given by means of the recurrence approach that we used to derive (4).

For any rational polynomial $P(m) \in \mathbf{Q}[m]$ we have $\sum_{m=0}^{\infty} P(m)/m! = \mathrm{e} \cdot q$ where $\mathrm{e} = 2.71828\ldots$ is Euler number and $q \in \mathbf{Q}$. This follows simply by expressing $P(m)$ as a $\mathbf{Q}$-linear combination in the basis $\{1, m, m(m-1), m(m-1)(m-2), \ldots\}$. Dobiński's formula ([16, Problems 1.9a and 1.13] and [4, p. 210]) belongs to this family of identities and has $P(m) = m^n$ and $q = B_n$ where B_n is the nth Bell number. Setting in (5), respectively in the analogous equation for all hypergraphs, $y = 1$ and $x_i = x^i$ and comparing coefficients at x^n we obtain two identities of this type.

Corollary 4.3 *For every $n \in \mathbf{N}$ we have the identities ($\lambda = 1^{a_1} 2^{a_2} \ldots m^{a_m}$ with $a_m = 0$ allowed)*

$$\sum_{m=0}^{\infty} \frac{1}{m!} \cdot \sum_{\lambda \vdash n} \prod_{i=1}^{m} \binom{\binom{m}{i}}{a_i} = \mathrm{e} \cdot \sum_{i(\mathcal{H})=n}^{*} \frac{1}{v(\mathcal{H})!}$$

$$\sum_{m=0}^{\infty} \frac{1}{m!} \cdot \sum_{\lambda \vdash n} \prod_{i=1}^{m} \binom{\binom{m}{i} + a_i - 1}{a_i} = \mathrm{e} \cdot \sum_{i(\mathcal{H})=n} \frac{1}{v(\mathcal{H})!}$$

where $\mathrm{e} = 2.71828\ldots$ *and the star indicates that the sum is over simple $\mathcal{H}$ only.*

For $n = 1, 2, 3$, and 4 the factors at e in the first identity are $1, 1, \frac{11}{6}$, and $\frac{25}{8}$ and in the second identity $1, 2, \frac{23}{6}$, and $\frac{89}{8}$.

5 Two applications of Davenport–Schinzel sequences

We begin with reminding a bound from the theory of generalized Davenport–Schinzel sequences. A sequence $v = a_1 a_2 \ldots a_l \in [n]^*$ over the alphabet $[n]$ is *k-sparse* if $a_i = a_j, i < j$, implies $j - i \geq k$. The length of v is denoted $|v|$. If $u, v \in [n]^*$ are two sequences and v has a subsequence that differs from u only by an injective renaming of symbols, we say that v *contains* u. For example, $v = 2131425$ contains $u = 4334$ but v does not contain $u = 2323$. We write $u(k, l)$ to denote the sequence

$$u(k, l) = 12\ldots k12\ldots k\ldots 12\ldots k \in [k]^*$$

with l segments $12\ldots k$. In Klazar [14] we proved that if $v \in [n]^*$ is k-sparse and does not contain $u(k, l)$, where $k \geq 2$ and $l \geq 3$, then for every $n \in \mathbf{N}$

$$|v| \leq n \cdot 2k2^{kl-4}(10k)^{2\alpha(n)^{kl-4}+8\alpha(n)^{kl-5}} \tag{8}$$

where $\alpha(n)$ is the inverse Ackermann function. (If $k = 1$ or $l \leq 2$, it is not difficult to prove that $|v| = O(n)$.)

Recall that $\alpha(n) = \min\{m : \ A(m) \geq n\}$ where $A(n) = F_n(n)$, the Ackermann function, is the diagonal function of the hierarchy of functions $F_i : \mathbf{N} \to \mathbf{N}$, $i \in \mathbf{N}$, starting with $F_1(n) = 2n$ and continuing by the rule $F_{i+1}(n) = F_i(F_i(\ldots F_i(1)\ldots))$ with n iterations of F_i. Thus $F_2(n) = 2^n$ and $F_3(n)$ is the tower function

$$F_3(n) = 2^{2^{\cdot^{\cdot^{\cdot^2}}}} \Big\} n.$$

We write $\beta(k, l, n)$ to denote the factor at n in (8). Thus

$$\beta(k, l, n) = 2k2^{kl-4}(10k)^{2\alpha(n)^{kl-4}+8\alpha(n)^{kl-5}}. \tag{9}$$

We utilize (8) in another approach to bounding $H_i(\mathcal{F}, n)$ from above in terms od $H_e(\mathcal{F}, n)$. Recall that $\mathcal{H}$ is a set partition if it has disjoint edges.

Theorem 5.1 *Suppose that $\mathcal{F}$ is a set partition with $p = v(\mathcal{F}), q = e(\mathcal{F}) > 1$ and $\mathcal{H}$ is a $\mathcal{F}$-free hypergraph with $v(\mathcal{H}) = n$, not necessarily simple. Then*

$$i(\mathcal{H}) < (q-1)n + \beta(q, 2p, e(\mathcal{H})) \cdot e(\mathcal{H})$$

where $\beta(k, l, n)$ is defined in (9).

Proof. Let $\bigcup \mathcal{H} = [n]$ and the edges of $\mathcal{H}$ be $H_1, H_2, \ldots, H_e$ where $e = e(\mathcal{H})$. We set for $i \in [n]$

$$S_i = \{j \in [e] : \ i \in H_j\}$$

and consider the sequence $v = I_1 I_2 \ldots I_n$ where I_i is an arbitrary permutation of S_i. Clearly, $v \in [e]^*$ and $|v| = i(\mathcal{H})$. The sequence v may not be q-sparse, because of the transitions $I_i I_{i+1}$, but it is easy to see that by deleting at most $q-1$ terms from the beginning of every I_i, $i > 1$, one can obtain a q-sparse subsequence w with length $|w| \geq |v| - (q-1)(n-1)$. It is also easy to see that if w (or v) contains $u(q, 2p)$ then $\mathcal{H}$ contains $\mathcal{F}$, which is forbidden. (Note that the subsequence aab in v forces the first a and the b to appear in two distinct segments I_i and thus it gives incidences of H_a and H_b with two distinct vertices.) Hence w does not contain $u(q, 2p)$ and we can apply (8):

$$i(\mathcal{H}) = |v| < (q-1)n + |w| \leq (q-1)n + \beta(q, 2p, e) \cdot e.$$

□

For fixed numbers k, l the function $\beta(k,l,n)$ grows to infinity extremly slowly and for all practical purposes it is bounded. We give an example showing that in the last theorem some unbounded factor at $e(\mathcal{H})$ is necessary.

Hart and Sharir [12] constructed sequences $v \in [n]^*$ which are 2-sparse, do not contain sequence 12121, and have length $|v| \gg n\alpha(n)$. See Sharir and Agarwal [21] for more information. We take such a sequence v, consider the subsequence w of v consisting of the first and last appearances of symbols $i \in [n]$ in v, and decompose v into segments

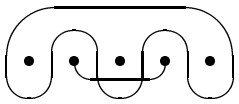
$$v = I_1 I_2 \dots I_m$$

where every I_i ends by a term from w and contains no other term of w. Clearly, $n \leq m = |w| \leq 2n$ (we may assume that v uses every $i \in [n]$). Note that $|I_1| = |I_m| = 1$. If an I_j contains a symbol $a \in [n]$ twice, we have in I_j a subsequence aba, $b \neq a$, because v is 2-sparse. By the definition of segments, the first b appears in v before I_j and the last b after I_j or on its end and v is forced to have the forbidden $bababa$ subsequence. Thus every I_j must be a permutation of a set $S_j \subset [n]$ and we can defined the hypergraph

$$\mathcal{H} = (H_i : \ i \in [n]) \ \text{ where } \ H_i = \{j \in [m] : \ i \in S_j\}.$$

Clearly, $\bigcup \mathcal{H} \subset [m]$ and $n \leq v(\mathcal{H}) \leq 2n$, $e(\mathcal{H}) = n$, and $i(\mathcal{H}) = |v| \gg n\alpha(n)$. It is also clear that $\mathcal{H}$ is $\mathcal{F}_{40}$-free where $\mathcal{F}_{40}$ is the set partition

$$\mathcal{F}_{40} = (\{1,3,5\},\{2,4\}) = \quad .$$



For $\mathcal{F} = \mathcal{F}_{40}$ the factor at $e(\mathcal{H})$ in Theorem 5.1 must be $\gg \alpha(n)$.

Taking in Theorem 5.1 a simple $\mathcal{H}$ with the maximum weight, we obtain as a corollary for every set partition $\mathcal{F}$ ($p = v(\mathcal{F})$ and $q = e(\mathcal{F}) > 1$; case $q = 1$ is trivial) the inequality

$$H_i(\mathcal{F},n) < (q-1)n + \beta(q, 2p, H_e(\mathcal{F},n)) \cdot H_e(\mathcal{F},n).$$

But here Theorem 2.4, when it applies, gives better bound.

In the second application of (8) we obtain an almost linear bound on $H_e(\mathcal{F},n)$ in the case when $\mathcal{F}$ is a *star forest*. These are simple graphs $\mathcal{G}$

which have no two separated edges and such that $\deg(v) = 1$ whenever $v = \max H$, $H \in \mathcal{H}$. Thus every component of a star forest is a star and every centre of a star is smaller then every leaf. We begin with the graph case.

Theorem 5.2 *Let $\mathcal{F}$ be a star forest with $r > 1$ components and p vertices and $G_e(\mathcal{F}, n)$ be the maximum number of edges in a simple graph $\mathcal{G}$ such that $\mathcal{G} \not\supset \mathcal{F}$ and $v(\mathcal{G}) = n$. Then*

$$G_e(\mathcal{F}, n) < (r-1)n + n \cdot \beta(r, 2(p-r+1), n)$$

where $\beta(k, l, n)$ is the almost constant function defined in (9).

Proof. Let $\mathcal{G}$ attain $G_e(\mathcal{F}, n)$ and $\bigcup \mathcal{G} = [n]$. We consider the sequence

$$v = I_1 I_2 \dots I_n \in [n]^*$$

where I_j is any permutation of the set $\{i \in [n] : \{i, j\} \in \mathcal{G}, i < j\}$. As in the previous proof, we select an r-sparse subsequence w of v with length $|w| \geq |v| - (r-1)(n-1)$. It is not hard to see that if w (or v) contains the sequence $u(r, 2(p-r+1))$ then $\mathcal{G} \supset \mathcal{F}$. Thus w does not contain $u(r, 2(p-r+1))$ and we can apply (8):

$$G_e(\mathcal{F}, n) = e(\mathcal{G}) = |v| < (r-1)n + n \cdot \beta(r, 2(p-r+1), n).$$

□

For $r = 1$ component we have $G_e(\mathcal{F}, n) \ll n$. We extend the bound of Theorem 5.2 from graphs to hypergraphs by means of a more generally aplicable technique in the next section. We conclude the present section by an example showing that in general $G_e(\mathcal{F}, n)$ is superlinear for star forests.

Let $v \in [n]^*$ be the same 12121-free sequence as in the previous example for $\mathcal{F}_{40}$ and let

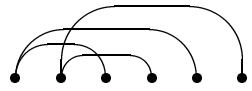
$$v = I_1 I_2 \dots I_m$$

be the same decomposition into segments containing no repeated symbol, $n \leq m \leq 2n$. We rename the symbols in v so that if $i < j$ then the first appearance of j in v precedes that of i. (This does not affect the 12121-freeness.) We define the simple bipartite graph $\mathcal{G}$ with $\bigcup \mathcal{G} = [n + m]$ by

$$\{i, j\} \in \mathcal{G} \iff i \in [n] \ \& \ j \in [n+1, n+m] \ \& \ i \text{ appears in } I_{j-n}.$$

Then $e(\mathcal{G})s = |v| \gg n\alpha(n)$ and $2n \le v(\mathcal{G}) \le 3n$. We show that $\mathcal{G} \not\supset \mathcal{F}_{41}$ where $\mathcal{F}_{41}$ is the star forest

$$\mathcal{F}_{41} = (\{1,3\},\{1,5\},\{2,4\},\{2,6\}) = \quad .$$



Suppose for the contrary that $\mathcal{F}_{41} \subset \mathcal{G}$ and $a_1 < a_2 < \ldots < a_6$ are the vertices of a $\mathcal{F}_{41}$-copy in $\mathcal{G}$. By the definition of $\mathcal{G}$, $z = a_1a_2a_1a_2$ is a subsequence of v, with terms appearing in $I_{a_3-n}, \ldots, I_{a_6-n}$, respectively. But since $a_2 > a_1$, an a_2 must appear in v before z starts and v contains a subsequence of the type 12121, which is forbidden. So $\mathcal{G}$ is $\mathcal{F}_{41}$-free and shows that $G_e(\mathcal{F}_{41}, n) \gg n\alpha(n)$.

6 Orderly bipartite forests

$\mathcal{H}$ is an *orderly bipartite forest* (OBF) if it is a simple graph which has no cycle and such that $\min H < \max H'$ holds for every two edges $H, H' \in \mathcal{H}$. Star forests are OBF. Orderly bipartite forests with some singleton edges (which may repeate) form the largest class of $\mathcal{F}$ for which one can hope for linear or close to linear extremal functions. (Since every OBF with singletons is contained in an OBF without singletons, it is enough to consider only OBF.) We state this simple but important observation as a theorem.

Theorem 6.1 *If the hypergraph $\mathcal{F}$ is not an orderly bipartite forest with singletons, then there is a constant $\gamma > 1$ such that $H_e(\mathcal{F}, n) \gg n^\gamma$.*

Proof. If $\mathcal{F}$ is not an OBF with singletons, then $\mathcal{F}$ has (i) an edge with more than two elements or (ii) two separated two-element edges or (iii) a two-path isomorphic to $(\{1,2\},\{2,3\})$ or (iv) a repeated two-element edge or (v) an even cycle of two-element edges (odd cycles are subsumed in (iii)). In the cases (i)–(iv) it is easy to see that $H_e(\mathcal{F}, n) \gg n^2$ (cf. the results for $\mathcal{F}_{11}$, $\mathcal{F}_{32}$, $\mathcal{F}_{30}$, and $\mathcal{F}_{31}$ in Section 3). An application of the probabilistic method (Erdős [6]) provides an unordered graph that has n vertices, $\gg n^{1+1/k}$ edges, and no even cycle of length k. Thus $H_e(\mathcal{F}, n) \gg n^{1+1/k}$ in case (v) if $\mathcal{F}$ has an even cycle of length k. □

In the unordered case it is well known that $G_e^u(\mathcal{F}, n) = \mathrm{ex}(\mathcal{F}, n) \ll n$ iff $\mathcal{F}$ is a forest, and if $\mathcal{F}$ is not a forest then $\mathrm{ex}(\mathcal{F}, n) \gg n^\gamma$ for some $\gamma > 1$ (by

the aforementioned result). In the ordered case the class OBF enjoys much larger variety of linear and close to linear extremal functions.

We say, for $k \in \mathbf{N}$, that a graph $\mathcal{G}'$ is a *k-blowup* of a graph $\mathcal{G}$ if for every edge coloring $\chi : \mathcal{G}' \to \mathbf{N}$ that uses every color $i \in \mathbf{N}$ at most k times, there exists a subgraph in $\mathcal{G}'$ which is isomorphic to $\mathcal{G}$ and whose edges have totally different colors (no color is repeated on the subgraph). For example, it is not difficult to construct for every OBF $\mathcal{G}$ and $k \in \mathbf{N}$ an OBF $\mathcal{G}'$ that is a k-blowup of $\mathcal{G}$. For $k \in \mathbf{N}$ and a graph $\mathcal{G}$ we write $B(k, \mathcal{G})$ to denote the set of all k-blowups of $\mathcal{G}$. The following theorem shows how to derive bounds for hypergraphs from the graph case.

Theorem 6.2 *Suppose that $\mathcal{F}$ is a graph with $p = v(\mathcal{F})$ and $q = e(\mathcal{F}) > 1$. If $f : \mathbf{N} \to \mathbf{N}$ is a nondecreasing function such that*

$$G_e(B(\tbinom{p}{2}, \mathcal{F}), n) < n \cdot f(n)$$

holds for every $n \in \mathbf{N}$, then

$$H_e(\mathcal{F}, n) < q \cdot G_e(\mathcal{F}, n) \cdot H_e(\mathcal{F}, 2f(n) + 1) \tag{10}$$

holds for every $n \in \mathbf{N}$.

Proof. Let $\mathcal{H}$ attain $H_e(\mathcal{F}, n)$ and $\bigcup \mathcal{H} = [n]$. We put in $\mathcal{H}'$ every edge with more than 1 and less than p vertices and for every $H \in \mathcal{H}$ with $|H| \geq p$ we put in $\mathcal{H}'$ an arbitrary subset $H' \subset H$, $|H'| = p$. No edge of $\mathcal{H}'$ repeats more then $q - 1$ times for else $H \supset \mathcal{F}$. Let $\mathcal{H}''$ be the simplification of $\mathcal{H}'$. So $e(\mathcal{H}) \leq n + (q-1)e(\mathcal{H}'')$. Let $\mathcal{G}$ be the simple graph consisting of all the edges E such that $E \subset H$ for some $H \in \mathcal{H}''$. Observe that if $\mathcal{F}' \in B(\binom{p}{2}, \mathcal{F})$, meaning that $\mathcal{F}'$ is a $\binom{p}{2}$-blowup of $\mathcal{F}$, and $\mathcal{F}' \subset \mathcal{G}$, then $\mathcal{F} \subset \mathcal{H}''$ and thus $\mathcal{F} \subset \mathcal{H}$. (For the edges $E \in \mathcal{G}$ lying in an $\mathcal{F}'$-copy consider the coloring $\chi(E) = H \in \mathcal{H}''$ where $E \subset H$.) Hence $\mathcal{F}' \subset \mathcal{G}$ for no $\mathcal{F}' \in B(\binom{p}{2}, \mathcal{F})$. Let $v(\mathcal{G}) = n'$; $n' \leq n$. We have

$$e(\mathcal{G}) \leq G_e(B(\tbinom{p}{2}, \mathcal{F}), n') < n' \cdot f(n').$$

There exists a vertex $v_0 \in \bigcup \mathcal{G}$ such that

$$d = \deg_{\mathcal{G}}(v_0) < 2f(n') \leq 2f(n).$$

Fix an arbitrary E_0, $v_0 \in E_0 \in \mathcal{G}$. Let $X \subset [n]$ be the union of all $H \in \mathcal{H}''$ with $E_0 \subset H$ and m be the number of such edges in $\mathcal{H}''$. We have the inequalities

$$m \le H_e(\mathcal{F}, |X|) \text{ and } |X| \le d+1.$$

Thus ($H_e(\mathcal{F}, n)$ is increasing by Theorem 2.2)

$$m \le H_e(\mathcal{F}, |X|) \le H_e(\mathcal{F}, d+1) < H_e(\mathcal{F}, 2f(n)+1).$$

We see that the two-element set E_0 is contained in at least one but less than $H_e(\mathcal{F}, 2f(n)+1)$ edges of $\mathcal{H}''$. Deleting those edges we obtain a subhypergraph $\mathcal{H}_1''$ of $\mathcal{H}''$ on which the same argument can be applied. That is, a two-element set E_1 exists such that $E_1 \subset H$ for at least one but less than $H_e(\mathcal{F}, 2f(n)+1)$ edges $H \in \mathcal{H}_1''$ (clearly $E_1 \neq E_0$). Continuing this way until the whole $\mathcal{H}''$ is exhausted, we define a mapping

$$F : \mathcal{H}'' \to \{E : E \subset [n], |E| = 2\}$$

such that

$$F(H) \subset H \text{ and } |F^{-1}(E)| < H_e(\mathcal{F}, 2f(n)+1)$$

holds for every $H \in \mathcal{H}''$ and every $E \subset [n], |E| = 2$. Let $\mathcal{G}'$ be the simple graph $\mathcal{G}' = F(\mathcal{H}'')$. Let $v(\mathcal{G}') = n'$; $n' \le n$.

The containment $\mathcal{F} \subset \mathcal{G}'$ implies, by the definition of $\mathcal{G}'$, that $\mathcal{F} \subset \mathcal{H}''$ and thus $\mathcal{F} \subset \mathcal{H}$, which is forbidden. We have (it is easy to see that $G_e(\mathcal{F}, n)$ is increasing)

$$e(\mathcal{G}') \le G_e(\mathcal{F}, n') \le G_e(\mathcal{F}, n).$$

Putting it all together, we obtain ($G_e(\mathcal{F}, n) \ge n-1$ if $q > 1$)

$$\begin{aligned} H_e(\mathcal{F}, n) = e(\mathcal{H}) &\le n + (q-1) \cdot e(\mathcal{H}'') \\ &< n + (q-1) \cdot H_e(\mathcal{F}, 2f(n)+1) \cdot e(\mathcal{G}') \\ &\le q \cdot H_e(\mathcal{F}, 2f(n)+1) \cdot G_e(\mathcal{F}, n). \end{aligned}$$

□

Recursive inequality (10) is nontrivial only if $f(n) = o(n)$ and thus it has any value only if $\mathcal{F}$ is an OBF (or perhaps if $\mathcal{F}$ is an even cycle). If $\mathcal{F}$ is an OBF and in Theorem 6.2 we replace $B(\binom{p}{2}, \mathcal{F})$ by some subclass $B \subset B(\binom{p}{2}, \mathcal{F}) \cap \text{OBF}$, the number of colors $\binom{p}{2}$ can be replaced by $p-1$.

(Because for $|H| = p$ every p two-element edges $E \subset H$ contain a cycle but now no $\mathcal{F}' \in B$ has a cycle.) Note that the ordering of vertices was not used in the proof (it is crucial only for obtaining linear or close to linear bounds on $G_e(\mathcal{F}, n)$ and $G_e(B, n)$) and therefore Theorem 6.2 holds in the unordered case as well. We make use of this in the first of its three applications.

Theorem 6.3 *Let $\mathcal{F}$ be an unordered forest. Its unordered hypergraph extremal function satisfies*

$$H_e^u(\mathcal{F}, n) \ll n.$$

Proof. Let $v(\mathcal{F}) = p$ and $e(\mathcal{F}) = q > 1$ (case $q = 1$ is trivial). It is not hard to prove that $G_e^u(\mathcal{F}, n) = \mathrm{ex}(\mathcal{F}, n) \leq (q-1)n$ (e.g. Bollobás [2, Exercise 24 in IV.7]). It is also easy to define a large forest $\mathcal{F}'$ with $Q = e(\mathcal{F}') = ((p-1)(q-1)+1)e(\mathcal{F}) = (pq-p-q+2)q \leq pq(q-1)$ edges that is a $(p-1)$-blowup of $\mathcal{F}$. We set $B = \{\mathcal{F}'\}$ and use (10) with the bounds $G_e^u(\mathcal{F}, n) \leq (q-1)n$, $f(n) = Q-1$ (since $G_e^u(B, n) = G_e^u(\mathcal{F}', n) \leq (Q-1)n$), and $H_e^u(\mathcal{F}, n) < 2^n$ (trivial):

$$H_e^u(\mathcal{F}, n) < q(q-1) \cdot n \cdot 2^{2Q-1} = \tbinom{q}{2} 4^{pq(q-1)} \cdot n.$$

□

One can prove this bound also directly without using Theorem 6.2 by adapting the proof of $\mathrm{ex}(\mathcal{F}, n) \leq (q-1)n$ to the hypergraph case.

In the second application of Theorem 6.2 we extend the bound of Theorem 5.2 to hypergraphs.

Theorem 6.4 *Let $\mathcal{F}$ be a star forest with $r > 1$ components, p vertices, and q edges. Let $t = (p-1)(q-1)+1$. Then*

$$H_e(\mathcal{F}, n) \ll n \cdot \beta(r, 2t(p-r)+2, n)^3$$

where $\beta(k, l, n)$ is the almost constant function defined in (9).

Proof. We replace $\mathcal{F}$ by the star forest $\mathcal{F}'$ in which every edge $\{i, j\} \in \mathcal{F}$, $i < j$, is replaced by t edges $\{i, j(1)\}, \ldots, \{i, j(t)\}$ where $i < j(1) < \cdots < j(t)$ and the set $\{j(1), \ldots, j(t)\}$ is slightly blowned up leaf j, that is, $j_1 < j_2$ implies $j_1(a) < j_2(b)$ for all $1 \leq a, b \leq t$ and all leaves j_1, j_2 of $\mathcal{F}$. It is easy to see that $\mathcal{F}' \in B(p-1, \mathcal{F})$.

We set $B = \{\mathcal{F}'\}$ and use (10) with the bounds $G_e(\mathcal{F}, n) \ll n \cdot \beta(r, 2(p-r)+2, n)$ (Theorem 5.2 for $\mathcal{F}$), $f(n) \ll \beta(r, 2t(p-r)+2, n)$ (Theorem 5.2 for $\mathcal{F}'$), and $H_e(\mathcal{F}, n) \ll n^p$ (Theorem 2.5):

$$H_e(\mathcal{F}, n) \ll n \cdot \beta(r, 2t(p-r)+2, n)^{p+1}.$$

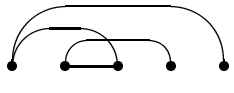
Feeding in (10) this improved upper bound on $H_e(\mathcal{F}, n)$, the second application of Theorem 6.2 gives

$$H_e(\mathcal{F}, n) \ll n \cdot \beta(r, 2t(p-r)+2, n)^2 \cdot \beta(r, 2t(p-r)+2, c\beta(\cdots))^{p+1}$$

where $c > 0$ is a constant. Since $\beta(r, 2t(p-r)+2, c\beta(r, 2t(p-r)+2, n)) < \beta(r, 2t(p-r)+2, n)^{1/(p+1)}$ for every sufficiently large n, we obtain the stated bound. □

By Theorem 2.4, for $H_i(\mathcal{F}, n)$ we have the same bound.

Our third and last application of Theorem 6.2 is to the graph

$\mathcal{F}_{42} = (\{1,3\}, \{1,5\}, \{2,3\}, \{2,4\}) =$  .

It arises from $\mathcal{F}_{41}$ by identifying 3 and 4 but it is more important to note that the starting points of the two edges ending in 3 emanate two noncrossing edges ending to the right of 3. $\mathcal{F}_{42}$ is an OBF but it is not a star forest. $\mathcal{F}_{42}$ in its matrix form

$$\begin{pmatrix} 1 & 1 & \\ 1 & & 1 \end{pmatrix}$$

(configuration C_2 of [9]) was introduced by Füredi [7] in order to prove that every convex n-gon has $O(n \log n)$ diagonals with unit length. (Recently simpler proof was given by Braß and Pach [3].) In [7] he proved that

$$n \log n \ll G_e(\mathcal{F}_{42}, n) \ll n \log n.$$

(The proof was given for ordered bipartite graphs but it works without changes for all ordered graphs.) For our purposes we need somewhat stronger version of the upper bound, which we prove (by the same argument of [7]) in Lemma 6.5. For completeness, we reproduce here the construction proving the lower bound, as given in [9, Construction 3.2].

We define inductively bipartite graphs $\mathcal{G}_n$, $n \in \mathbf{N}$, with parts $[2^n]$ and $[2^n+1, 2^{n+1}]$. $\mathcal{G}_1 = (\{1,3\}, \{2,3\}, \{2,4\})$. Let $A = [2^n]$, $B = [2^n+1, 2^{n+1}]$,

$C = [2^{n+1}+1, 2^{n+1}+2^n]$, and $D = [2^{n+1}+2^n+1, 2^{n+2}]$. $\mathcal{G}_{n+1}$ consists of two copies of $\mathcal{G}_n$, one on the parts A, C and the other on B, D, and a matching between B and C. An easy induction shows that $e(\mathcal{G}_n) = (n+2)2^{n-1}$, $v(\mathcal{G}_n) = 2^{n+1}$, and $\mathcal{G}_n \not\supset \mathcal{F}_{42}$. Thus $G_e(\mathcal{F}_{42}, n) \gg n \log n$.

For $k \in \mathbf{N}$ consider graphs $\mathcal{G}$ with the following structure. $\bigcup \mathcal{G} = [k+1+a+b]$, $a, b \geq k$, and $\mathcal{G}$ has $2k^2+k$ edges: $\{i, k+1\} \in \mathcal{G}$ for $i \in [k]$ and every $i \in [k]$ is connected by k edges to $[k+2, k+1+a]$ and by k edges to $[k+2+a, k+1+a+b]$. Thus $\deg(k+1) = k$ and $\deg(i) = 2k+1$ for $i \in [k]$. We write $\mathcal{F}_{42}(k)$ to denote the set of all such graphs.

Lemma 6.5 *The sets of graphs $\mathcal{F}_{42}(k)$, $k \in \mathbf{N}$, are as defined above.*

1. *$\mathcal{F}_{42}(3k+1) \subset B(k, \mathcal{F}_{42})$. In particular, $\mathcal{F}_{42}(31) \subset B(\binom{5}{2}, \mathcal{F}_{42})$.*

2. *$G_e(\mathcal{F}_{42}(k), n) \ll_k n \log n$.*

Proof. Let $K = 3k+1$ and $\mathcal{G} \in \mathcal{F}_{42}(K)$ be edge colored so that each color appears at most k times. We select two edges $\{i, K+1\}$ and $\{j, K+1\}$, $i < j < K+1$, with different colors. There are K edges $\{i, l\}$ and K edges $\{j, l'\}$ such that $K+1 < l' < l$ holds for every two vertices l' and l. Because in each of the two K-tuples we have at least 4 different colors, we can select vertices l' and l so that the colors of $\{i, K+1\}, \{j, K+1\}, \{j, l'\}$, and $\{i, l\}$ are all different. Since $i < j < K+1 < l' < l$, this subgraph is isomorphic to $\mathcal{F}_{42}$.

Let $n \geq 2$ and $\mathcal{G}$ be a simple graph with $\bigcup \mathcal{G} = [n]$ which contains no $\mathcal{F} \in \mathcal{F}_{42}(k)$. For every fixed $i \in [n]$, we list the endpoints j of $\{i, j\} \in \mathcal{G}$, $i < j$: $i < j_0 < j_1 < \cdots < j_{t-1} \leq n$. Let $s = \lfloor t/(k+1) \rfloor$. We keep only the edges with endpoints $j_{(i-1)(k+1)}$, $i = 1, 2, \ldots, s$. (If $t < k+1$, we keep no edge $\{i, j\}$.) The graph $\mathcal{G}'$ obtained satisfies $e(\mathcal{G}') \geq e(\mathcal{G})/(2k+1) - kn$ and for every two edges $\{i, j\}, \{i, j'\} \in \mathcal{G}'$, $i < j < j'$, there are at least k edges $\{i, l\} \in \mathcal{G}, j < l < j'$, and at least k edges $\{i, l\} \in \mathcal{G}, l > j'$. Now we proceed as in [7]. We say that $\{i, j\} \in \mathcal{G}'$, $i < j$, has *type* (j, m) if there are two edges $\{i, l\}$ and $\{i, l'\}$ of $\mathcal{G}'$ such that $j < l < l'$ and $l - j \leq 2^m < l' - j$. Consider the partition $\mathcal{G}' = \mathcal{G}_1 \cup \mathcal{G}_2$ where $\mathcal{G}_1$ is formed by edges with at least one type and $\mathcal{G}_2$ by edges without type. It follows from the definitions that if k edges of $\mathcal{G}_1$ have the same type, then $\mathcal{F} \subset \mathcal{G}$ for some $\mathcal{F} \in \mathcal{F}_{42}(k)$. The number of types is less than $n(1 + \log_2 n)$. Thus $|\mathcal{G}_1| < kn + kn \log_2 n$. Let $i \in [n]$ and $i < j_0 < j_1 < \cdots < j_{t-1} \leq n$ be the endpoints j, $j > i$, of the edges incident with i which have no type. Let $d_r = j_r - j_{r-1}$, $r \in [t-1]$,

and $D = d_1 + \cdots + d_{t-1} = j_{t-1} - j_0$. If $d_1 \leq D/2$, then $d_1 \leq 2^m < D$ for some $m \in \mathbf{N}_0$ and $\{i, j_0\}$ has type (j_0, m) because of $\{i, j_1\}$ and $\{i, j_{t-1}\}$. Thus $d_1 > D/2$ and $D - d_1 < D/2$. For similar reason $d_2 > (D - d_1)/2$ and $D - d_1 - d_2 < D/4$. In general $1 \leq D - d_1 - \cdots - d_r < D/2^r$ for $r \in [t-2]$. We obtain that $t \leq \lfloor \log_2 D \rfloor + 2 < 3 + \log_2 n$, $|\mathcal{G}_2| < 3n + n \log_2 n$, and $|\mathcal{G}'| = |\mathcal{G}_1| + |\mathcal{G}_2| < (k+3)n \log_2 n$. Therefore

$$e(\mathcal{G}) < kn + (2k+1)e(\mathcal{G}') < (2k+2)(k+3)n \log_2 n.$$

□

Theorem 6.6 *Let* $\mathcal{F}_{42} = (\{1,3\}, \{1,5\}, \{2,3\}, \{2,4\})$. *Then*

$$n \cdot \log n \ll H_e(\mathcal{F}_{42}, n) \ll n \cdot (\log n)^2 \cdot (\log \log n)^3.$$

Proof. The lower bound holds already in the graph case and was proved above. To prove the upper bound, we set $B = \mathcal{F}_{42}(31)$ and apply (10) of Theorem 6.2 with the bounds $f(n) \ll \log n$ (because $G_e(B(\binom{5}{2}, \mathcal{F}_{42}), n) \leq G_e(B, n) \ll n \log n$ by Lemma 6.5), $G_e(\mathcal{F}_{42}, n) \ll n \log n$ ([7] or from the previous bound by $G_e(\mathcal{F}_{42}, n) \leq G_e(B, n)$), and $H_e(\mathcal{F}_{42}, n) \ll n^5$ (Theorem 2.5; we could as well start with the completely trivial bound $H_e(\mathcal{F}_{42}, n) < 2^n$):

$$H_e(\mathcal{F}_{42}, n) \ll n \cdot (\log n)^6.$$

Using this bound, the next application gives

$$H_e(\mathcal{F}_{42}, n) \ll n \cdot (\log n)^2 \cdot (\log \log n)^6.$$

The third application gives

$$\begin{aligned} H_e(\mathcal{F}_{42}, n) &\ll n \cdot (\log n)^2 \cdot (\log \log n)^2 \cdot (\log \log \log n)^6 \\ &\ll n \cdot (\log n)^2 \cdot (\log \log n)^3. \end{aligned}$$

□

By Theorem 2.4,

$$H_i(\mathcal{F}_{42}, n) \ll n \cdot (\log n)^2 \cdot (\log \log n)^3$$

as well.

7 Concluding remarks

We mention possible directions for further research. As for Theorem 2.4, singleton hypergraphs $\mathcal{S}_k$ show that $H_i(\mathcal{F},n) \ll H_e(\mathcal{F},n)$ is not always true. We conjecture that $\mathcal{S}_k$ are the only exceptions: if $\mathcal{F} \neq \mathcal{S}_k$, then $H_i(\mathcal{F},n) < cH_e(\mathcal{F},n)$ holds with a constant $c > 0$ depending only on $\mathcal{F}$ (but cf. Theorem 5.1 and the example with $\mathcal{F}_{40}$). One may try to extend the precise results of Section 3, say to the case when $\mathcal{F}$ is a graph with 3 or 4 edges. One may try to explain the repetitions of functions $H_e(\mathcal{F},n)$ and $H_i(\mathcal{F},n)$, such as for no. 9, 10, and 19–23 or for no. 25–28, 33, and 34.

An interesting question is about the order of magnitude of the hypergraph counting functions considered in Section 4. We can prove that $\log(h_n^{(i,s)})$ and $\log(h_n^{(i)})$ are equal to $n\log n - n\log\log n + O(n)$ but more precise asymptotics are desirable. The ratio $h_n^{(i)}/h_n^{(i,s)}$ seems to tend to a limit lying between 1.2 and 1.3. What is this limit? Which partition $\lambda \vdash n$ maximize the number of (simple or all) hypergraphs with weight n and edge type λ?

As for the class OBF (orderly bipartite forests), we define a hierarchy of three subclasses of OBF

$$\mathrm{LIN} \subset \mathrm{ALIN} \subset \mathrm{CLIN} \subset \mathrm{OBF}.$$

LIN contains $\mathcal{F}$ with linear extremal function: $H_e(\mathcal{F},n) \ll n$. ALIN contains $\mathcal{F}$ with almost linear extremal function: $H_e(\mathcal{F},n) \ll n \cdot f(\alpha(n))$ where $\alpha(n)$ is the inverse Ackermann function and $f(n)$ is primitively recursive. CLIN contains $\mathcal{F}$ with close to linear extremal function: $H_e(\mathcal{F},n) \ll n \cdot (\log n)^c$ where $c > 0$ is a constant depending only on $\mathcal{F}$. (If $\mathcal{F}$ is a hypergraph and $H_e(\mathcal{F},n) \ll n \cdot (\log n)^c$, then by Theorem 6.1 $\mathcal{F}$ must be an OBF.) The first two inclusions are sharp: we have seen that $\mathcal{F}_{41} \in \mathrm{ALIN}\backslash\mathrm{LIN}$ and that $\mathcal{F}_{42} \in \mathrm{CLIN}\backslash\mathrm{ALIN}$. Is it true that CLIN=OBF? If not, what general upper bound can one give for $H_e(\mathcal{F},n)$ if $\mathcal{F}$ is an OBF? As for Theorem 6.6, what is the exact asymptotics of $H_e(\mathcal{F}_{42},n)$?

A basic but difficult question is to determine which OBF lie in LIN, which in ALIN, and which in CLIN. We summarize briefly our knowledge. Here we have proved that the four OBF with $e(\mathcal{F}) \leq 2$, namely $\mathcal{F}_4, \mathcal{F}_{29}, \mathcal{F}_{33}$, and $\mathcal{F}_{34}$, are in LIN. A more general result is given in [15, Theorem 3.3]: for every $k \in \mathbf{N}$ the star forest

$$\mathcal{N}(k) = (\{i, 2k-i+1\}, \{i, 2k+i\} : \ i \in [k])$$

([k] is matched with $[k+1, 2k]$ decreasingly and with $[2k+1, 3k]$ increasingly) is in LIN. (In [15] the linear bound is proved only for the graph case but blowing up the leaves of $\mathcal{N}(k)$ and using Theorem 6.2 we can extend it to hypergraphs.) We have proved here that every star forest is in ALIN; the containment of $\mathcal{F}_{41}$ forces it to be in ALIN\LIN. As for CLIN\ALIN, it contains $\mathcal{F}_{42}$ and some modifications of it but we do not know any large subfamily.

References

[1] R. Anstee, R. Ferguson and A. Sali, Small forbidden configurations II, *Electr. J. Comb.,* **8** (2001), R4, 24 pages.

[2] B. Bollobás, *Modern Graph Theory,* Springer, Berlin, 1998.

[3] P. Brass and J. Pach, The maximum number of times the same distance can occur among the vertices of a convex n-gon is $O(n \log n)$, *J. Comb. Theory, Ser. A,* **94** (2001), 178–179.

[4] L. Comtet, *Advanced Combinatorics,* D. Reidel Publ. Co., Boston, MA, 1974.

[5] P. Erdős, On an elementary proof of some asymptotic formulas in the theory of partitions, *Ann. Math.,* **43** (1942), 437–450.

[6] P. Erdős, Graph theory and probability, *Canadian J. Math.,* **11** (1959), 34–38.

[7] Z. Füredi, The maximum number of unit distances in a convex n-gon, *J. Comb. Theory, Ser. A,* **55** (1990), 316–320.

[8] Z. Füredi, Turán type problems. In: A. D. Keedwell (eds.), *Surveys in Combinatorics, 1991,* Cambridge University Press, Cambridge, UK, 1991; pp. 253–300.

[9] Z. Füredi and A. Hajnal, Davenport–Schinzel theory of matrices, *Discrete Math.,* **103** (1992), 233–251.

[10] P. Frankl, Extremal set systems. In: R. L. Graham, M. Grötschel and L. Lovász (eds.), *Handbook of Combinatorics, Volume 2,* Elsevier, Amsterdam, 1995; pp. 1293–1329.

[11] P. Frank, Z. Füredi, G. Katona, D. Miklós (editors), *Extremal Problems for Finite Sets,* János Bolyai Mathematical Society, Budapest, 1994.

[12] S. Hart and M. Sharir, Nonlinearity of Davenport–Schinzel sequences and of generalized path compression schemes, *Combinatorica,* **6** (1986), 151–177.

[13] T. Hearne and C. Wagner, Minimal covers of finite sets, *Discrete Math.,* **5** (1973), 247–251.

[14] M. Klazar, A general upper bound in extremal theory of sequences, *Commentat. Math. Univ. Carol,* **33** (1992), 737–746.

[15] M. Klazar, Counting pattern-free set partitions II. Noncrossing and other hypergraphs, *Electr. J. Comb.,* **7** (2000), R34, 25 pages.

[16] L. Lovász, *Combinatorial Problems and Exercises,* Akadémiai Kiadó, Budapest, 1993.

[17] A. J. Macula, Covers of a finite set, *Math. Mag.,* **67** (1994), 141–144.

[18] D. J. Newman, The evaluation of the constant in the formula for the number of partitions of n, *Amer. J. Math.,* **73** (1951), 599–601.

[19] D. J. Newman, A simplified proof of the partition formula, *Mich. Math. J.,* **9** (1962), 283–287.

[20] D. J. Newman, *Analytic Number Theory,* Springer, Berlin, 1998.

[21] M. Sharir and P. K. Agarwal, *Davenport–Schinzel Sequences and Their Geometric Applications,* Cambridge University Press, Cambridge, UK, 1995.

[22] N. J. A. Sloane (2000), The On-Line Encyclopedia of Integer Sequences, published electronically at
http://www.research.att.com/~njas/sequences/.

[23] Zs. Tuza, Applications of the set-pair method in extremal hypergraph theory. In: P. Frankl et al. (eds.), *Extremal Problems for Finite Sets,* János Bolyai Mathematical Society, Budapest, 1994; pp. 479–514.

[24] Zs. Tuza, Applications of the set-pair method in extremal problems, II. In: D. Miklós et al. (eds.), *Combinatorics, Paul Erdős is Eighty, Volume 2,* János Bolyai Mathematical Society, Budapest, 1996; pp. 459–490.

Counting even and odd partitions

Martin Klazar

1. Introduction. It is a lovely fact that $[n] = \{1, 2, \ldots, n\}$, $n \geq 1$, has as many subsets X with an even cardinality $|X|$ as those with an odd cardinality, namely 2^{n-1} of both. To prove it, pair every subset X with $X \pm 1$ where $X \pm 1$ is $X \backslash \{1\}$ if $1 \in X$ and $X \cup \{1\}$ if $1 \notin X$. Then $X \mapsto X \pm 1$ is an involution that changes the parity of $|X|$ and the result follows.

More generally, in enumerative combinatorics one often has a family $\mathcal{S}_n$ of objects on $[n]$ such that every object X has a natural *size* $s(X) \in \mathbf{N}_0$ of some kind. Then besides the total number of objects $S_n = |\mathcal{S}_n|$ one can consider also

$$S_n^{\pm} = \sum_{X \in \mathcal{S}_n} (-1)^{s(X)},$$

the surplus of the objects with an even size over those with an odd size. For subsets of $[n]$ and $s(X) = |X|$ we have $S_n^{\pm} = 0$ for every $n \geq 1$ (but $S_0^{\pm} = 1$). In this note we present to the reader four examples of the described situation. We investigate the corresponding numbers $S_n^{\pm}$ by means of generating functions, an analytic continuation argument, and, again, the involution trick. Our first example is a classics but the other three are much less known.

2. Integer partitions. $\mathcal{S}_n$ consists of the partitions X of n into distinct parts, $n = a_1 + a_2 + \cdots + a_k$ where $a_1 > a_2 > \ldots > a_k \geq 1$ are integers, and $s(X) = k$ is just the number of parts.

Theorem 1. (L. Euler, 1748) For integer partitions with distinct parts, $S_n^{\pm} = (-1)^m$ if $n = \frac{1}{2}m(3m \pm 1)$ and $S_n^{\pm} = 0$ else.

This is Euler's celebrated pentagonal identity which can be written equivalently as

$$\prod_{n=1}^{\infty}(1 - x^n) = \sum_{m=-\infty}^{\infty} (-1)^m x^{m(3m+1)/2}.$$

Franklin's famous 1881 proof using the involution trick is reproduced in the book [1] of Andrews or in Hardy and Wright [4].

3. Noncrossing set partitions. A (set) partition of $[n]$ is a collection $X = \{B_1, B_2, \ldots, B_k\}$ of nonempty disjoint subsets of $[n]$, called *blocks*, whose

union is $[n]$. It is *crossing* if there are four numbers $1 \le a < b < c < d \le n$ and two distinct blocks $A, B \in X$ such that $a, c \in A$ and $b, d \in B$. Else X is a *noncrossing* partition. $\mathcal{S}_n$ consists of the noncrossing partitions of $[n]$ and $s(X) = k$ is just the number of blocks. Kreweras [5] proved that $S_n = |\mathcal{S}_n| = \frac{1}{n+1}\binom{2n}{n}$, the nth Catalan number. The survey [10] of Simion contains much information on the combinatorics of noncrossing partitions.

Theorem 2. For noncrossing set partitions, $S_n^{\pm} = (-1)^{m+1}\frac{1}{m+1}\binom{2m}{m}$ if $n = 2m+1$ and $S_n^{\pm} = 0$ if $n = 2m$.

Proof. Let

$$F = F(x,y) = \sum_{n \ge 0} \sum_{X \in \mathcal{S}_n} x^n y^{s(X)} = 1 + xy + x^2(y + y^2) + \cdots.$$

We are interested in

$$G = G(x) = \sum_{n \ge 0} S_n^{\pm} x^n.$$

Clearly, $G(x) = F(x, -1)$. We show that

$$F = 1 + xyF + xF(F-1). \tag{1}$$

The empty X is represented by 1. Now let X be a noncrossing partition of $[n]$, $n \ge 1$, and A, $1 \in A$, be its first block. Either $|A| = 1$ or $|A| > 1$. In the former case, $A = \{1\}$ and after peeling off A we obtain a noncrossing partition whose length and size is by 1 smaller. This is captured by the term xyF. In the latter case, we let a denote the second element of A and decompose X into two partitions X_1 and X_2, where X_1 is induced by X on the interval $[2, a-1]$ and X_2 is induced on $[a, n]$. Both X_i are noncrossing. X_1 may be empty but X_2 is nonempty. Since no block intersects both intervals, $s(X_1) + s(X_2) = s(X)$. This decomposition is captured by the last term $xF(F-1)$.

Setting in (1) $y = -1$ and rearranging, we get the equation $xG^2 - (1 + 2x)G + 1 = 0$. Thus ($G(0) = 1$)

$$G(x) = 1 + \frac{1}{2x}\left(1 - \sqrt{1 + 4x^2}\right).$$

Binomial expansion yields the stated formula for $S_n^{\pm}$. Note that setting in (1) $y = 1$, we recover the result of Kreweras. □

Is there a proof using involutions?

4. All set partitions. Now $\mathcal{S}_n$ consists of all partitions of $[n]$ and $s(X) = k$ is again the number of blocks. The total numbers S_n are the *Bell numbers*

$$1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, 678570, 4213597, \ldots$$

forming sequence A000110 of [11]. They grow superexponentially, $\log S_n = n(\log n - \log\log n + O(1))$. See de Bruijn [2, p. 108] or Lovász [6, Problem 1.9b] for more precise asymptotics. We show that $S_n^{\pm}$ remain superexponential.

Theorem 3. For all set partitions, if $c > 0$ is any constant then $|S_n^{\pm}| > c^n$ for some (in fact, infinitely many) $n \in \mathbf{N}$.

Proof. We begin with the classical expansion (see, for example, Stanley [12, p. 34])

$$G_k(x) = \sum_{n \geq 0} S(n,k)x^n = \frac{x^k}{(1-x)(1-2x)\ldots(1-kx)}$$

where $S(n,k)$, the Stirling number of the second kind, is in our language simply the number of $X \in \mathcal{S}_n$ with $s(X) = k$ blocks. Thus

$$F(x) = \sum_{n \geq 0} S_n^{\pm} x^n = \sum_{k \geq 0} (-1)^k G_k(x) = \sum_{k \geq 0} \frac{(-x)^k}{(1-x)(1-2x)\ldots(1-kx)}.$$

Considering the action of the substitution $x := x/(1-x)$ on this expansion, we obtain the equation

$$F(x) = 1 - \frac{x}{1-x} F(x/(1-x)). \tag{2}$$

Substituting now $x := x/(1+x)$ and solving the resulting equation for $F(x)$, we obtain the second equation

$$F(x) = \frac{1}{x}\Big(1 - F(x/(1+x))\Big). \tag{3}$$

If $|S_n^{\pm}| < c^n$ for all $n \in \mathbf{N}$ for a constant $c > 0$, the series $F(x)$ has radius of convergence $r \geq 1/c > 0$ and defines in the disc $|z| < r$ an analytic function $F(z)$. However, we show that $r > 0$ is contradicted by the equations (2) and (3). Thus $|S_n^{\pm}| < c^n$ holds for no $c > 0$ and our theorem follows.

Suppose, for the contradiction, that $r > 0$. We can assume that $r \le 1$ (certainly $|S_n^{\pm}| \ge 1$ infinitely often). Let $\alpha \in \mathbf{C}$, $|\alpha| = r$, be a singularity of $F(z)$ on the circle of convergence. If $|\alpha/(1-\alpha)| < r$, we use (2) to continue $F(z)$ analytically to a neighborhood of α, which contradicts the definition of α. Clearly, $|\alpha/(1-\alpha)| < r$ is equivalent to $\mathrm{Re}(\alpha) < r^2/2$ and therefore for $\mathrm{Re}(\alpha) < r^2/2$ we have a contradiction. Similarly, if $|\alpha/(1+\alpha)| < r$, which is equivalent to $\mathrm{Re}(\alpha) > -r^2/2$, we use (3) to obtain the same contradiction. (Since $\alpha \ne 1$ in the former case, $\alpha \ne -1$ in the latter case, and always $\alpha \ne 0$, the bad arguments $z = -1, 0, 1$ do not bother us.) For every location of α (2) or (3) leads to a contradiction. (In the strip $|\mathrm{Re}(z)| < r^2/2$ one can use both equations.) Hence $r = 0$. □

The numbers $S_n^{\pm}$, $n \ge 1$,

$$-1, 0, 1, 1, -2, -9, -9, 50, 267, 413, -2180, -17731, -50533, 110176, \ldots$$

form sequence A000587 of [11]. Recently their asymptotics was investigated by Yang [15] (see [11] for more references on them) who mentions that Subbarao and Verma proved that in fact $\limsup \log|S_n^{\pm}|/(n\log n) = 1$. Is $S_n^{\pm}$ zero infinitely often? This question is in [15] atributed to H. S. Wilf. Is $S_n^{\pm}$ ever zero besides $n = 2$?

5. Matchings and crossings. Perhaps the lack of cancelation was caused by the rapid growth of S_n? Our last example shows that $S_n^{\pm}$ can be small even if S_n are superexponential. Now $\mathcal{S}_n$ consists of all partitions X of $[2n]$ into n two-element blocks. We call such X *matchings* and their blocks *edges*. The size $s(X)$ is the number of crossing pairs A, B of the edges of X (we have defined crossing in the second example). It is easy to see that $S_n = (2n-1)!! = 1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2n-1)$: $S_n = (2n-1)S_{n-1}$ because one has $2n-1$ ways to place the end of the new first edge in the spaces of an $X \in \mathcal{S}_{n-1}$. So $\log S_n = n(\log n + O(1))$. But $S_n^{\pm}$ are very small.

Theorem 4. For matchings whose size is measured by the number of crossings, $S_n^{\pm} = 1$ for every $n \in \mathbf{N}$.

Proof. For a matching $X \in \mathcal{S}_n$ the *crucial pair* is the pair of edges $A, B \in X$ such that $\min A + 1 = \min B$ and $\min A$ is as small as possible. Notice that the crucial pair is unique and that every X has it except $X^* = \{\{1,2\},\{3,4\},\ldots,\{2n-1,2n\}\}$. Switching $\min A$ and $\min B$ in X produces

Figure 1: The involution Φ.

the maching X' — see Figure 1. It is clear that A, B remains the crucial pair of X' and that $s(X) - s(X') = \pm 1$ because the set of crossing pairs of X and that of X' differ exactly in the pair A, B. So $\Phi : X \mapsto X'$ is an involution that changes the parity of $s(X)$. It pairs even and odd matchings except X^* and $s(X^*) = 0$ is even. □

A remarkable formula for the generating polynomial counting matchings by crossings was derived by Touchard and Riordan [14, 9] and was later proved bijectively by Penaud [8]:

$$\sum_{X \in \mathcal{S}_n} x^{s(X)} = \frac{1}{(1-x)^n} \sum_{k=-n}^{n} (-1)^k \binom{2n}{n-k} x^{k(k-1)/2}.$$

The reader is invited for an exercise: recover the above formulas for S_n and $S_n^{\pm}$ from the polynomial by setting $x = 1$ and $x = -1$.

6. Concluding remarks. Theorem 2 follows from the equation (1) that is proved in [10, p. 373]. Our derivation is more condensed. The analytic argument proving Theorem 3 seems new. So is perhaps the involution proof of Theorem 4 but the result itself, that $S_n^{\pm} = 1$, was found already by Riordan [9, p. 219]. We conclude by a problem on *connected* matchings. These are matchings X with this property: For every two distinct edges $A, B \in X$ there is a chain of edges $A_0, A_1, \ldots, A_k$ of X such that $A_0 = A$, $A_k = B$, and A_i, A_{i+1} is a crossing pair for every $i = 0, 1, \ldots, k-1$. So both X and X' in Figure 1 are disconnected, having two and three components, respectively. Let $\mathcal{S}_n$ be the set of all connected matchings on $[2n]$ and $s(X)$ be again the number of crossings. It is known and not too difficult to prove, see the articles of Stein [13] and Nijenhuis and Wilf [7], that the numbers $(S_n)_{n \geq 1} = (1, 1, 4, 7, 248, 2830, \ldots)$ (A000699 of [11]) follow the recurrence $S_n = (n-1) \sum_{i=1}^{n-1} S_i S_{n-i}$. (For further results on matchings and crossings

see Flajolet and Noy [3].) Now, as for $S_n^{\pm}$, do we have nice cancelation in the style of Theorems 1, 2, and 4 or do we have rather erratic behaviour as in Theorem 3?

Acknowledgment. I was supported by the project LN00A056 of the Ministry of Education of the Czech Republic. I am grateful to M. Noy for his comments.

References

[1] G. Andrews, *The Theory of Partitions*, Addison-Wesley Pub. Co., Reading, Mass., 1976.

[2] N. G. de Bruijn, *Asymptotic Methods in Analysis*, Dover Publications, New York, 1981.

[3] P. Flajolet and M. Noy, Analytic combinatorics of chord diagrams, *Proceedings of FPSAC'00, Moscow 2000*, (D. Krob, A. A. Mikhalev and A. V. Mikhalev, ed.), Springer, Berlin, 2000, pp. 191–201.

[4] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, Clarendon Press, Oxford, UK, 1979.

[5] G. Kreweras, Sur les partitions non croisées d'un cycle, *Discrete Math.* **1** (1972) 333–350.

[6] L. Lovász, *Combinatorial Problems and Exercises*, Akadémiai Kiadó, Budapest, 1993.

[7] A. Nijenhuis and H. S. Wilf, The enumeration of connected graphs and linked diagrams, *J. Comb. Theory, Ser. A* **27** (1979) 356–359.

[8] J.-G. Penaud, Une preuve bijective d'une formule de Touchard–Riordan, *Discrete Math.* **139** (1995) 347–360.

[9] J. Riordan, The distribution of crossings of chords joining pairs of $2n$ points on a circle, *Math. of Computation* **29** (1975) 215–222.

[10] R. Simion, Noncrossing partitions, *Discrete Math.* **217** (2000) 367–409.

[11] N. J. A. Sloane (2001), The On-Line Encyclopedia of Integer Sequences, published electronically at
http://www.research.att.com/~njas/sequences/

[12] R. P. Stanley, *Enumerative Combinatorics, Volume I*, Wadsworth & Brooks/Cole, Monterey, California, 1986.

[13] P. R. Stein, On a class of linked diagrams, I. Enumeration, *J. Comb. Theory, Ser. A* **24** (1978) 357–366.

[14] J. Touchard, Sur un problème de configurations et sur les fractions continues, *Canad. J. Math.* **4** (1952) 2–25.

[15] Y. Yang, On a multiplicative partition function, *Electr. J. of Comb.* **8** (2001) R 19, 14 pages.

Department of Applied Mathematics (KAM), Charles University, and Institute for Theoretical Computer Science (ITI), Malostranské náměstí 25, 118 00 Praha, Czech Republic
klazar@kam.ms.mff.cuni.cz

The Ehrenfeucht-Mycielski Sequence

K. Sutner

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
sutner@cs.cmu.edu

Abstract. We study the disjunctive binary sequence introduced by Ehrenfeucht and Mycielski in [1]. The match length associated to the bits of the sequence is shown to be a crucial tool in the analysis of the sequence. We show that the match length between two consecutive bits in the sequence differs at most by 1 and give a lower bound for the limiting density of the sequence. Experimental computation in the `automata` package has been very helpful in developing these results.

1 The Ehrenfeucht-Mycielski Sequence

An infinite sequence is *disjunctive* if it contains all finite words as factors. In [1] Ehrenfeucht and Mycielski introduced a method of generating a disjunctive binary sequence based on avoiding repetitions. To construct the Ehrenfeucht-Mycielski (EM) sequence U, start with a single bit 0. Suppose the first n bits $U_n = u_1 u_2 \dots u_n$ have already been chosen. Find the longest suffix v of U_n that appears already in U_{n-1}. Find the last occurrence of v in U_{n-1}, and let b be the first bit following that occurrence of v. Lastly, set $u_{n+1} = \overline{b}$, the complement of b. It is understood that if there is no prior occurrence of any non-empty suffix the last bit in the sequence is flipped. The resulting sequence starts like so:

$$0100110101110001000011110110010100100 1110$$

see also sequence A038219 in Sloane's catalog of integer sequences, [2]. The in the title of their paper the authors ask somewhat tongue-in-cheek how random their sequence is. As a first step towards understanding the properties of U they show that U is indeed disjunctive and conjecture that the limiting density of 1's is $1/2$.

1.1 Preliminary Data

To get a better understanding of U it is natural to generate a few thousand bits of the EM sequence using standard string matching algorithms. In a high-level environment such as Mathematica, see [3], a few lines of code suffice for this. In our work we use an automata theory package built on top of Mathematica that provides a number of tools that are helpful in the analysis of U, see [4]

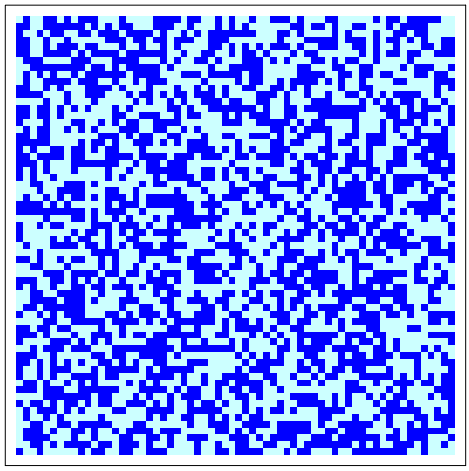


Fig. 1. The first 2^{12} bits of the Ehrenfeucht-Mycielski sequence.

for a recent description of the package. The first 2^{12} bits, in row-major order, are shown in figure 1. The pattern seems surprisingly indistinguishable from a random pattern given the simplicity of the definition of the sequence.

More interesting is a plot of the census function for U: nearly all words of length k appear already among the first 2^k bits of the sequence. Thus, an initial segment of the EM sequence behaves almost like a de Bruijn sequence, see [5]. Define the *cover* $\mathsf{cov}(W)$ of a word W, finite or infinite, to be the set of all its finite factors, and $\mathsf{cov}_k(W) = \mathbf{2}^k \cap \mathsf{cov}(W)$. Here we write $\mathbf{2}$ for the two-symbol alphabet $\{0,1\}$. The census function $C_k(n) = |\mathsf{cov}_k(U_n)|$ for the EM sequence increases initially at a rate of 1, and, after a short transition period, becomes constant at value 2^k. In figure 2, the green line stands for $k = 9$, blue for $k = 10$, and red for $k = 11$.

Another surprising picture emerges when one considers the length of the longest suffix v of $U_n = u_1 u_2 \dots u_n$ that matches with a previous occurrence. We write $\mu(n)$ for the suffix, and $\lambda(n) = |\mu(n)|$ for its length. As with the census function, the match length function λ increases in a very regular fashion. Indeed, in most places the length of the match at position n is $\lfloor \log_2 n \rfloor$. To visualize λ it is best to collapse runs of matches of the same length into a single data point. The plot 3 uses the first 2^{15} bits of the sequence. It is immediate from the definitions that the match length can never increase by more that 1 in a single step. The plot suggests that the match lengths also never drop by more than 1 in a single step, a fact that will be established below. The data also suggest that the match length function is nearly monotonic: once the first match of length k has occurred, all future matches are of length at least $k-2$. If true, this property would imply balance of the EM sequence, see section 3.

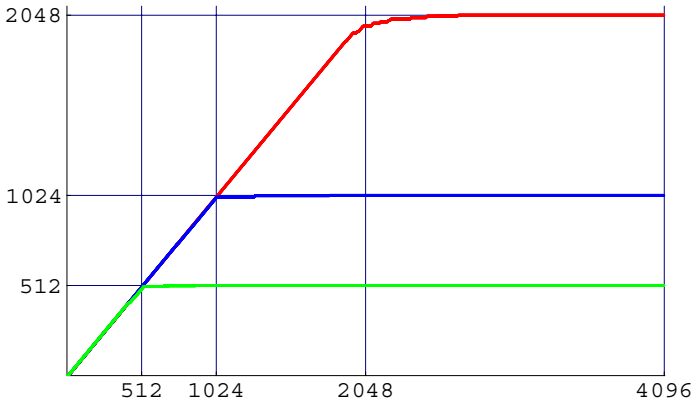


Fig. 2. The census function for the Ehrenfeucht-Mycielski sequence for words of lengths $k = 9, 10, 11$.

1.2 Generating Long Initial Segments

Clearly it would be helpful to test whether the patterns observed in the first few thousands of bits extend to longer initial segments, say, the first few million bits. To generate a million bits one has to resort to faster special purpose algorithms. As far as the complexity of U is concerned, it is clear that the language $\mathsf{pref}(U)$ of all prefixes of U fails to be regular. Hence it follows from the gap theorem in [6] that $\mathsf{pref}(U)$ cannot be context-free. The obvious practical approach is to use a variant of the KMP algorithm. Suppose k was the length of the previous match. We can scan U_n backwards and mark the positions of the nearest matches of length $k-2, k-1, k, k+1$. If no such match appears we have to revise the near-monotonicity conjecture from above. Of course, the scan can be terminated immediately if a match of length $k+1$ appears. If one implements this algorithm in an efficient language such as C++ it is straightforward to generate a few million bits of U.

Much better results can be achieved if one abandons pattern matching entirely and uses an indexing algorithm instead. In essence, it suffices to maintain, for each finite word w of some fixed length at most k, the position of the last occurrence of that word in the prefix so far constructed. This is done in brute-force tables and quite straightforward except at places where the match length function assumes a new maximum. A detailed description of the algorithm can be found in [7]. The reference shows that under the assumption of near-monotonicity discussed in section 1.3 one can generate a bit of the sequence in amortized constant time. Moreover, only linear space is required to construct an initial segment of the sequence, so that a simple laptop computer suffices to generate the first billion bits of the sequence in less than an hour.

As far as importing the bits into `automata` there are two choices. Either one can read the precomputed information from a file. Note, though, that storing

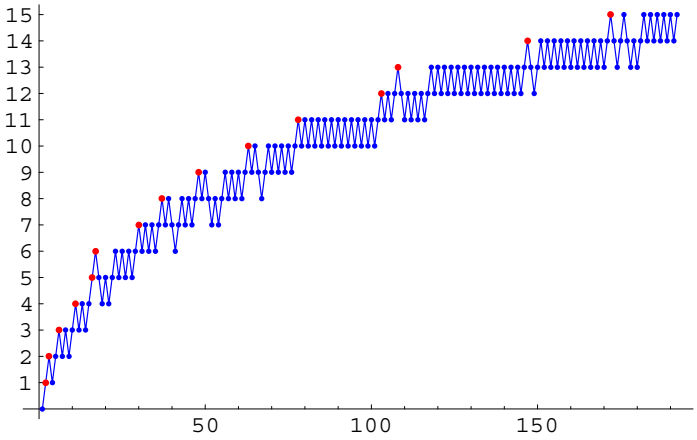


Fig. 3. Changes in the match lengths of the first 2^{15} bits of the Ehrenfeucht-Mycielski sequence.

the first billion bits in the obvious bit-packed format requires 125 million bytes, and there is little hope to decrease this amount of space using data compression: the very definition of the EM sequence foils standard algorithms. For example, the Lemple-Ziv-Welch based `gzip` algorithm produces a "compressed" file of size 159,410 bytes from the first million bits of the EM sequence. The Burrows-Wheeler type `bzip2` algorithm even produces a file of size 165,362 bytes.

The other options exploits the fact that Mathematica offers a communication protocol that allows one to call external programs directly from the kernel. This feature is used in `automata` extensively to speed up crucial algorithms.

1.3 Assorted Conjectures

It is clear from data plots as in the last section that the EM sequence has rather strong regularity properties and is indeed far from random. In their paper [1] Ehrenfeucht and Mycielski ask if their sequence is balanced in the sense that the limiting frequency of 0's and 1's is 1/2. More precisely, for any non-empty word $x \in \mathbf{2}^*$ let $\#_1 x$ be the number of 1's in x. Define the *density* of x to be $\Delta(x) = \frac{\#_1 x}{|x|}$. The following conjecture is from [1]:

Conjecture 1. Balance
In the limit, the density of U_n is 1/2.

Convergence seems to be very rapid. E.g., $\Delta(U_{2000000}) = 1000195/2000000 = 0.5000975$. It is shown in [8] that the density is bounded away from 0, and the argument given below provides a slightly better bound, but the balance conjecture remains open. To show balance, it suffices to establish the following property of the match length function.

Conjecture 2. Near Monotonicity
Any match of length k is followed only by matches of length at least $k-2$.

Near monotonicity implies rapid convergence of the density. We will prove a weaker a monotonicity property, namely that any match of length k is followed only by matches of length at least $k/2$. This suffices to show that the limiting density is bounded away from 0. Another interesting property of U is the rapid growth of the census function, simultaneously for all k.

Conjecture 3. Growth Rate
Any word of length k appears in the first $O(2^k)$ bits of the sequence.

As a matter of fact, a bound of 2^{k+2} appears to suffice, but it is unclear what the growth rate of the number of words that fail to appear already at time 2^{k+1} is. We originally conjectured a bound of 2^{k+1} but had to revise it after Hodsdon computed the first billion bits of the sequence, see [7]. The last two conjectures hold true for the first billion bits of the sequence.

We note in passing another apparent structural property that becomes visible from the data. The plot of the match lengths suggests that they grow in a very regular fashion. It is natural to inquire about the position of the match in U_n, i.e., the position of the nearest occurrence of the suffix v in U_n associated with the next bit. Figure 4 shows the positions of the first 2^{14} matches. The available range of positions for the matches forms a staircase, with a few outliers, and the match positions essentially form square blocks of size 2^k. The outliers are due to the internal dynamics of the sequence, see section 2.2 below, but match positions are very poorly understood at present.

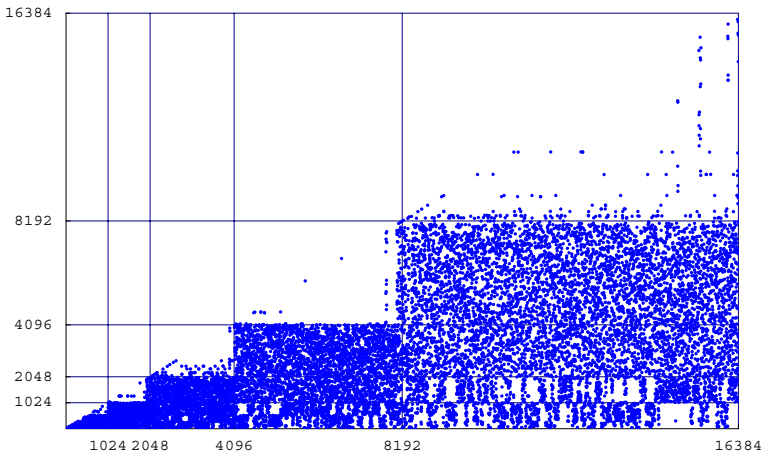


Fig. 4. Match positions in the first 2^{14} bits of the Ehrenfeucht-Mycielski sequence.

2 Recurrence and the Internal Clock

With a view towards computational support, it is convenient to think of the EM sequence as tracing a path in a de Bruijn $\mathcal{B}_k$. We write $\mathcal{B}_k(n)$ for the subgraph of $\mathcal{B}_k$ induced by the edges that lie on the path traced by U_n. Likewise, $\overline{\mathcal{B}}_k(n)$ denotes the complement of $\mathcal{B}_k(n)$, i.e., the subgraph obtained by removing all the edges that lie on the path traced by U_n. We also assume that isolated vertices are removed. It is easy in `automata` to generate and inspect these graphs for a reasonably wide range of parameters. This type of experimental computation turned out to be very helpful in the discovery of some of the results in the next section, and in avoiding dead-ends in the development of some of the proofs.

As a first step towards the analysis of the dynamics of U, from the definition of U we have the following fact.

Proposition 1. *Alternation Principle*
If a vertex u in $\mathcal{B}_k(n)$ appears twice in U_{n-1} it has out-degree 2.

As we will see, the condition for alternation is very nearly the same as having in-degree 2. It is often useful to consider the nodes in $\mathcal{B}_k$ that involve a subword v of length $k-1$. Clearly, there are exactly four such nodes, and they are connected by an alternating path of the form:

$$a\,v \rightarrow v\,b \leftarrow \overline{a}\,v \rightarrow v\,\overline{b} \leftarrow a\,v$$

We will refer to this subgraph as the *zigzag* of v. Since $\mathcal{B}_k$ is the line graph of $\mathcal{B}_{k-1}$, the zigzag of v corresponds to the node v and its 4 incident edges in $\mathcal{B}_{k-1}$. It follows from the last proposition that the path U can not touch a zigzag arbitrarily.

Proposition 2. *No Merge Principle*
The path U can not touch a zigzag in exactly two edges with the same target.

In particular v is a match if, and only if, all the nodes in the zigzag of v have been touched by U.

2.1 The Second Coming

Since we are dealing with a binary sequence one might suspect the initial segments U_{2^k} to be of particular interest, a suspicion borne out by figures 2 and 4. However, it turns out that there are other, natural stages in the construction of the EM sequence associated with the first repetition of the initial segments of the sequence. They determine the point where the census function first deviates from linear growth. First, a simple observation concerning the impossibility of repeated matches. Note that the claim made here is easy to verify using some of the graph algorithms in `automata`.

Proposition 3. *Some initial segment U_n of U traces a simple cycle in $\mathcal{B}_k$, anchored at vertex U_k. Correspondingly, the first match of length k is U_k.*

Proof. Since U is infinite, it must touch some vertex in $\mathcal{B}_k$ twice. But by proposition 2 the first such vertex can only be U_k, the starting point of the cycle. □

The proposition suggests to define $\Lambda(t) = \max\big(\lambda(s) \;\big|\; s \le t\big)$ to be the length of the longest match up to time t. Thus, Λ is monotonically increasing and changes value only at the second occurrence of an initial segment. We write τ_k for the time when U_k is encountered for the second time. Note that we have the upper bound $\tau_k \le 2^k + k - 1$ since the longest simple cycle in $\mathcal{B}_k$ has length 2^k. The fact that initial segments repeat provides an alternative proof of the fact that U is disjunctive, see [1] for the original argument.

Lemma 1. *The Ehrenfeucht-Mycielski sequence U is disjunctive.*

Proof. It follows from the last proposition that every factor of U occurs again in U. Now choose n sufficiently large so that $H = \mathcal{B}_k(n) = \mathcal{B}_k(m)$ for all $m \ge n$. Since every point in H is touched by U at least twice, it must have out-degree 2 by alternation. But the only such graph is $\mathcal{B}_k$ itself. □

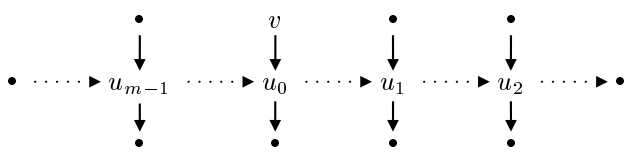
It follows that every word appears infinitely often on U, and we can define τ_i^w, $i \ge 0$, to be the position of the ith occurrence of word w in U. As always, this is interpreted to mean the position of the last bit of w. Define τ_i^k to be $\tau_i^{U_k}$, so $\tau_0^k = k$ and $\tau_1^k = \tau_k$. Also note that $\tau_{k+1} = \tau_2^k + 1$.

Proposition 4. *Any word of length k other than U_k appears exactly once as a match. The initial segment U_k appears exactly twice. Hence, the total number of matches of length k is $2^k + 1$.*

Proof. First suppose $u \in \mathbf{2}^k$ is not an initial segment of U. By lemma 1 $a\,u$ and $\overline{a}\,u$ both appear in U. The first such occurrences will have u as match. Clearly, from then on u cannot appear again as a match. Likewise, by 1 any initial segment $u = U_k$ must occur twice as a match since there are occurrences u, $a\,u$ and $\overline{a}\,u$. As before, u cannot reappear as a match later on in the sequence. □

2.2 Rounds and Irregular Words

Proposition 3 suggests that the construction of U can be naturally decomposed into a sequence of of rounds during which Λ remains constant. We will refer to the interval $R_k = [\tau_k, \tau_{k+1} - 1]$ as the k *principal round*. During R_k, the maximum match function Λ is equal to k, but λ may well drop below k. Up to time $t = \tau_{k+1} - 1$ the EM sequence traces two cycles C_0 and C_1 in $\mathcal{B}_k$, both anchored at $u = U_k$. C_0 is a simple cycle, and the two cycles are edge-disjoint. Note that the complement $\overline{\mathcal{B}}_k(t) = \mathcal{B}_k - C_0 - C_1$ consists only of degree 2 and, possibly, degree 4 points, the latter corresponding to words of length k not yet encountered at time t. The strongly connected components are thus all Eulerian.

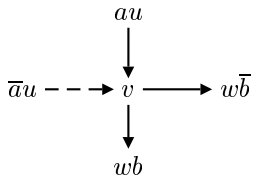


When U later touches one of these components at u_0, by necessity a degree 2 point, we have the following situation: $v = aw$ and $u_0 = wb$ so that the sequence look like $\ldots a\,w\,b \ldots a\,w\,\overline{b} \ldots$ Thus, the first two occurrences of w are preceded by the same bit. Such words will be called *irregular* and we will see shortly that the first three occurrences of any irregular word are of the form $\ldots a\,w\,b \ldots a\,w\,\overline{b} \ldots \overline{a}\,w\,b \ldots$ For the sake of completeness, we distinguish between irregular, regular and initial words. It is easy to see that all words 0^k and 1^k, $k \geq 2$ are irregular. There seem to be few irregular words; for example, there are only 12 irregular words of length 10. It is clear from the definitions that whenever v occurs as a match, all its prefixes must already have occurred as matches. Because of irregular words, the situation for suffixes is slightly more complicated, but we will see that they too occur as matches with a slight delay.

Our interest in irregular words stems from the fact that they are closely connected with changes in match length. Within any principal round, λ can decrease only when an irregular word is encountered for the second time, and will then correspondingly increase when the same word is encountered for the third time, at which point it appears as a match. First, increases in match length.

Lemma 2. *Suppose the match length increases at time t, i.e., $\lambda(t+1) = \lambda(t)+1$, but Λ does not increase at time t. Then $v = \mu(t)$ is irregular and $t = \tau_2^v$. Moreover, at time $s = \tau_1^v$ the match length decreases: $\lambda(s) > \lambda(s+1)$.*

Proof. Set $k = |v|$ and consider the edges incident upon v in $\mathcal{B}_k$ at time t. The dashed edge indicates the last step.



Since the match length increases, both edges (v, wb) and $(v, w\overline{b})$ must already lie on U_t. But that means that the edge (au, v) must appear at least twice on U_t, and v is irregular. Now consider the time $s = \tau_1^v$ of the second appearance. We must have $s > r = \tau_2^k$. But the strongly connected component of v in the residual graph $\overline{\mathcal{B}}_k(r)$ consists only of degree 2 and, possibly, degree 4 points; point v itself is in particular degree 2. As a consequence, U must then trace a closed path in this component that ends at v at time $t = \tau_2^v$. Lastly, the match length at time $s+1$ is k, but must have been larger than k at time s. □

Thus all changes in match length inside of a principal round are associated with irregular words. The lemma suggests the following definition. A *minor round (of order k)* is a pair (r, s) of natural numbers, $r \leq s$, with the property that $\lambda(r-1) \geq k+1$, $\lambda(t) \leq k$ for all t, $r \leq t \leq s$, and $\lambda(s+1) \geq k+1$. Since trivially $\lambda(t+1) \leq \lambda(t)+1$, the last condition is equivalent to $\lambda(s+1) = k+1$.

Note that minor rounds are either disjoint or nested. Moreover, any minor round that starts during a principal round must be contained in that principal

round. We can now show that match length never drops by more than 1 at a time.

Lemma 3. *Let (r, s) be a minor round. Then $\lambda(r-1) = \lambda(r) + 1 = \lambda(s+1)$.*

Proof. From the definition, for any minor round (r, s) we have $\lambda(s+1) - \lambda(r-1) \leq 0$. Now consider the principal round for k. As we have seen, all minor rounds starting before R_k are already finished at time τ_1^k. But if any of the minor rounds during the k principal round had $\lambda(s+1) - \lambda(r-1) < 0$ the match length at the end of R_k would be less than k, contradicting the fact that the match length increases to $k+1$ at the beginning of the next principal round. □

Hence, there cannot be gaps between two consecutive match length values.

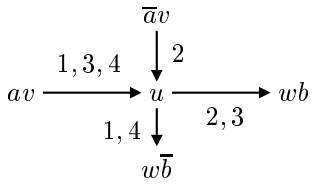
Theorem 1. *No-Gap*
For all n, $\lambda(n) - 1 \leq \lambda(n+1) \leq \lambda(n) + 1$.

2.3 A Lower Bound

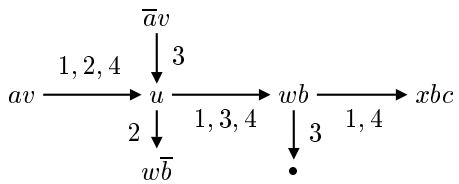
It follows from the last section that for u not an initial segment, $\tau_1^u \in R_k$ implies that u matches at some time $t \in R_k$. We will say that u *matches with delay* at time τ_1^u.

Lemma 4. *Let u be a word, not an initial segment. At time τ_3^u both $0u$ and $1u$ match with delay.*

Proof. First suppose that u is regular. Consider the neighborhood of u in $\mathcal{B}_k$ where $k = |u|$. In the following figure, the edge labels indicate one way U may have passed through u by time τ_3^u. Note that our claim follows trivially if both au and $\overline{a}u$ appear twice on $U_{\tau_3^u}$, so we only need to deal with the asymmetric case.



Since $u\overline{b}$ appears twice, it must match, with delay. But then Both $\overline{a}u\overline{b}$ and $au\overline{b}$ must appear, so $\overline{a}u$ appears twice and must match, with delay. A similar argument covers the remaining case. For u irregular the second encounter entails a third as indicated in the following figure. It suffices to deal with a fourth hit as indicated below.



But then ubc is also irregular, and we must have an occurrence of $\overline{a}ubc$, with delay. □

Lemma 5. *If uab has matched at time t, then both $0u$ and $1u$ match at time t, with delay.*

Proof. From the last lemma, our claim is obvious as long as u is not an initial segment. So suppose $u = U_k$ and consider the first 5 occurrences of u:

$$uabc \ldots xu\overline{a} \ldots \overline{x}ua\overline{b} \ldots zuab\overline{c} \ldots \overline{z}uabc$$

Note that the second occurrence of $\overline{x}uab$ is before the end of round R_{k+2}, so both xu and $\overline{x}u$ must have matched before the end of that round. □

Corollary 1. *If a word u of length k matches at time t, then all words of length at most $\lfloor k/2 \rfloor$ have matched at time t, with delay.*

From the corollary we obtain the lower bound $\tau_k = \Omega(\sqrt{2}^k)$. It follows from an argument in [8] that this yields a lower bound of 0.11 for the asymptotic density of U, a far cry from the observed value of $1/2$.

3 Density and Near Monotonicity

The density of a set $W \subseteq \mathbf{2}^k$ is defined by $\Delta(W) = \frac{1}{|W|} \sum_{x \in W} \Delta(x)$. To keep notation simple, we adopt the convention that a less-than or less-than-or-equal sign in an expression indicates summation or union. E.g., we write $\binom{k}{<p}$ for $\sum_{0 \leq i < p} \binom{k}{i}$. We denote $\mathbf{2}^{k,p}$ the set of words in $\mathbf{2}^k$ of density p/k, i.e., all words containing exactly p many 1's. Thus, $\left|\mathbf{2}^{k,p}\right| = \binom{k}{p}$. Clearly $\Delta(\mathbf{2}^k) = 1/2$ by symmetry. A simple computation shows that, perhaps somewhat counterintuitively, $\Delta(\mathbf{2}^{k,\leq k/2}) = 1/2$. Hence, by monotonicity $\Delta(\mathbf{2}^{k,\leq \varepsilon k}) = 1/2$ for all $1/2 \leq \varepsilon \leq 1$.

Now suppose $W \subseteq \mathbf{2}^k$ is a set of cardinality m. What is the least possible density of W? Clearly, a minimal density set W must have to form $\mathbf{2}^{k,\leq p} \cup A$ where $A \subseteq 2^{k,p+1}$. If m forces $p \geq k/2$, then asymptotically the density of W is $1/2$. Indeed, we will see that $m = \Omega(2^k)$ suffices. Let $0 \leq p \leq k$. From the definition of density we have

$$\Delta(\mathbf{2}^{k,\leq p}) = \frac{\sum_{i \leq p} \binom{k}{i} i/k}{\binom{k}{\leq p}} = 1/2 - \left(4 \frac{\binom{k-1}{<p}}{\binom{k-1}{p}} + 2\right)^{-1}$$

Let $p = \lfloor \varepsilon k \rfloor + c$ where $c \in \mathbb{Z}$ is constant. As long as $1/2 \leq \varepsilon \leq 1$ we obtain density $1/2$ in the limit. However, this is far as one can go.

Lemma 6. *Let $0 \leq \varepsilon < 1/2$ and $p = \lfloor \varepsilon k \rfloor + c$ where $c \in \mathbb{Z}$ is constant. Then $\lim_{k \to \infty} \frac{\binom{k}{<p}}{\binom{k}{p}} = \varepsilon/(1-2\varepsilon)$.*

Proof. For the sake of brevity we write $\gamma = \frac{\binom{k}{<p}}{\binom{k}{p}}$. First note that the density of $\mathbf{2}^{k,\leq \varepsilon k}$ is clearly bounded from above by ε. Since $\Delta(\mathbf{2}^{k,\leq \varepsilon k}) = \frac{\gamma}{2\gamma+1}$ it follows

that $\gamma \leq \frac{\varepsilon}{1-2\varepsilon}$. For the opposite direction we rewrite the individual quotients of binomial coefficients in terms of Pochhammer symbols as $\binom{k}{p-i}/\binom{k}{p} = \frac{(p-i+1)_i}{(k-p+1)_i}$ Hence the limit of $\binom{k}{p-i}/\binom{k}{p}$ as k goes to infinity is $\left(\frac{\varepsilon}{1-\varepsilon}\right)^i$. Now consider a partial sum $\sum_{i=1}^{n} \binom{k}{p-i}/\binom{k}{p} \leq \gamma$ where n is fixed. Then

$$\sum_{i=1}^{n} \frac{\binom{k}{p-i}}{\binom{k}{p}} \quad \longrightarrow \quad \sum_{i=1}^{n} \left(\frac{\varepsilon}{1-\varepsilon}\right)^i = \frac{\varepsilon}{1-2\varepsilon}\left(1 - \left(\frac{\varepsilon}{1-\varepsilon}\right)^n\right)$$

as k goes to infinity. But then $\lim_{k\to\infty} \gamma \geq \frac{\varepsilon}{1-2\varepsilon}$. Thus, in the limit $\gamma = \frac{\varepsilon}{1-2\varepsilon}$. □

Corollary 2. *Let $0 \leq \delta \leq 1/2$. Then $\lim_{k\to\infty} \Delta(\mathbf{2}^{k,\leq \delta k}) = \delta$.*

The definition of density extends naturally to multisets $A, B \subseteq \mathbf{2}^k$ via $\Delta(A+B) = \frac{|A|\Delta(A)+|B|\Delta(B)}{|A+B|}$. Assuming near monotonicity, we can now establish balance of U by calculating the limiting density at times τ_k. Let us say that λ is c-monotonic if $\forall\, t, s\ (\lambda(t+s) \geq \lambda(t) - c)$. Thus, it seems that λ is 2-monotonic, but the argument below works for any constant c.

Theorem 2. *If λ is c-monotonic for some constant c, then the Ehrenfeucht-Mycielski sequence is balanced.*

Proof. Assume otherwise; by symmetry we only have to consider the case where for infinitely many t we have $\Delta(U_t) < \delta_0 < 1/2$. Let $\tau_{k+c} \leq t < \tau_{k+c+1}$ and consider the multiset $W = \mathsf{cov}_k(U_t)$. For t sufficiently large $\Delta(W) < \delta_0$. Since all matches after t have length at least k by our assumption, certainly $\mathbf{2}^k \subseteq W$. Since all words of length $k+c+1$ on U_t are unique, there is a constant bounding the multiplicities of $x \in \mathbf{2}^k$ in W and we can write $W = \mathbf{2}^k + V$ where $\forall\, x \in \mathbf{2}^k\ (V(x) \leq d)$. Let $\delta = \Delta(V)$ and $m = |V|$, so that

$$\delta_0 > \Delta(W) = \frac{2^k \cdot 1/2 + m \cdot \delta}{2^k + m}.$$

It follows that $2^{k-1}(1-2\delta_0) \leq m(\delta_0 - \delta) \leq m$ so that $m = \Omega(2^k)$.

On the other hand, we must have $\delta_0 \geq \Delta(V) \geq \Delta(d \cdot \mathbf{2}^{k,\leq p}) = \Delta(\mathbf{2}^{k,\leq p})$. To see this, note that if for some $x \in \mathbf{2}^k$, $q/k = \Delta(x) < \Delta(\mathbf{2}^k + d \cdot \mathbf{2}^{k,<q})$ then $\mathbf{2}^k + d \cdot \mathbf{2}^{k,\leq q}$ minimizes the density of all multisets with multiplicities bounded by d that include x. From the last corollary we get $p \leq \delta_0 k$. Using Sterling approximation we see that the cardinality m is bounded by $d\binom{k}{\leq \delta_0 k} \leq d + d\delta_0 k\binom{k}{\delta_0 k} \approx d + d\sqrt{\frac{\delta_0 k}{2\pi(1-\delta_0)}}\, 2^{kH(\delta_0)}$ where $H(x) = -x\lg x - (1-x)\lg(1-x)$ is the binary entropy function over the interval $[0,1]$. It is well-known that H is symmetric about $x = 1/2$ and concave, with maximum $H(1/2) = 1$. Hence $2^{H(\delta_0)} < 2$, contradicting our previous lower bound. Hence, the density of W approaches $1/2$, as required. □

4 Conclusion

We have established some regularity properties of the Ehrenfeucht-Mycielski sequence, notably the No-Gap conjecture and a weaker form of Near Monotonicity. A better analysis of the match length function should show that λ is in fact 2-monotonic. Specifically, a study of the de Bruijn graphs $\overline{\mathcal{B}}_k$ in `automata` indicates that the strongly connected component of this graph have special properties that could be exploited to establish this claim. Alas, we are currently unable give a complete proof. The construction of the Ehrenfeucht-Mycielski sequence easily generalizes to arbitrary prefixes: start with a word w, and then attach new bits at the end according to the same rules as for the standard sequence. It seems that all results and conjectures here seem to carry over, mutatis mutandis, to these generalized Ehrenfeucht-Mycielski sequences. In particular, they all appear to have limiting density 1/2.

Source code and Mathematica notebooks used in the writing of this paper can be found at `www.cs.cmu/~sutner`.

References

1. Ehrenfeucht, A., Mycielski, J.: A pseudorandom sequence–how random is it? American Mathematical Monthly **99** (1992) 373–375
2. Sloane, N.J.A.: The on-line encyclopedia of integer sequences. (`www.research.att.com/~njas/sequences`)
3. Wolfram, S.: The Mathematica Book. 4th edn. Wolfram Media, Cambridge UP (1999)
4. Sutner, K.: `automata`, a hybrid system for computational automata theory. In Champarnaud, J.M., Maurel, D., eds.: CIAA 2002, Tours, France (2002) 217–222
5. Golomb, S.W.: Shift Register Sequences. Aegean Park Press, Laguna Hills, CA (1982)
6. Calude, C., Yu, S.: Language-theoretic complexity of disjunctive sequences. Technical Report 007, CDMTCS (1995)
7. Hodsdon, A.: The generalized Ehrenfeucht-Mycielski sequences. Master's thesis, Carnegie Mellon University (2002)
8. McConnell, T.R.: Laws of large numbers for some non-repetitive sequences. `http://barnyard.syr.edu/research.shtml` (2000)

Algebraic Aspects of B-regular Series

Ph. Dumas

Algorithms Project,
INRIA Rocquencourt BP 105,
78153 Le Chesnay Cedex, France

Abstract. This paper concerns power series of an arithmetic nature that arise in the analysis of divide-and-conquer algorithms. Two key notions are studied: that of B-regular sequence and that of Mahlerian sequence with their associated power series. Firstly we emphasize the link between rational series over the alphabet $\{x_0, x_1, \ldots, x_{B-1}\}$ and B-regular series. Secondly we extend the theorem of Christol, Kamae, Mendès France and Rauzy about automatic sequences and algebraic series to B-regular sequences and Mahlerian series. We develop here a constructive theory of B-regular and Mahlerian series. The examples show the ubiquitous character of B-regular series in the study of arithmetic functions related to number representation systems and divide-and-conquer algorithms.

The interest of 2-regular sequences comes from their presence in many problems which touch upon the binary representation of integers or divide-and-conquer algorithms, like sum-of-digits function, number of odd binomial coefficients, Josephus problem, mergesort, Euclidean matching or comparison networks. This explains why we study B-regular sequences that formalize the sequences which are solutions of certain difference equations of the divide-and-conquer type. In other words we want to show that B-regular series (i.e. generating functions of B-regular sequences) are as important in computer science as rational functions are common in mathematics.

Many properties of B-regular sequences like closure properties or growth properties have been etablished by Allouche and Shallit. In particular they showed that there is a link between B-regular sequences and rational series in the sense of formal language theory. The transition from one to another uses the B-ary representation of integers. There is already a long tradition about recognizable sets and automatic sequences.

The link provides us with the well known machinery of rational series and the first part of the paper is devoted to the illustration of its use. For example we introduce the Hankel matrix of a regular series. This is the practical way to find the rank of a regular series, to exhibit minimal recurrence relations or to build up linear representations.

In the second part we compare B-regular series and Mahlerian series. Our goal is to extend the theorem of Christol, Kamae, Mendès France and Rauzy [6], which asserts that q-automatic series with coefficients in the finite field $\mathbb{F}_q$ are exactly algebraic series. To that purpose we introduce a more general notion of Mahlerian series. We prove in particular that B-regular series are Mahlerian series.

The reciprocal is more intricate but most useful. Indeed the theorem of Christol *et alii* is not adequate for theoretical computer science where the sequences have elements that are integer rather than elements of a finite fields. We give a partial answer to this problem, that permits to cover numerous cases of application.

In all the examples we have aimed at making the computations effective.

It is worth noting that we concentrate here on one facet of B-regular sequences, their algebraic closure properties. A complementary point of vue is the study of asymptotic behaviour of these sequences. One will find numerous examples in [9, 10].

1 Rational Series and B-regular Series

The properties of B-regular series come mainly from the properties of rational series in non commutative indeterminates and we build up a catalog where each notion about B-regular series is a translation of the corresponding notion about rational series. In view of the richness of the subject we limit ourselves to the essentials.

Let us begin with an example which gives the flavour of 2-regular series.

Example 1. Let us assume that we want to go from 0 to an integer n by leaps whose lengths are power of 2 and directions are forward or backward. The shortest path has a length w_n which may be defined by the conditions $w_0 = 0$, $w_n = 1$ if $n = 2^k$ and $w_n = 1 + \min(w_{n-2^k}, w_{2^{k+1}-n})$ if $2^k < n < 2^{k+1}$. For example we find $w_{14} = 2$ because $14 = 16 - 2$.

Another way to obtain this sequence (w_n) is to consider the two square matrices

$$A_0 = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

and the row and column matrices

$$\lambda = \begin{pmatrix} 0 & 1 & 1 & 2 \end{pmatrix}, \quad \gamma = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T.$$

If the binary expansion of the integer n is $\epsilon_\ell \cdots \epsilon_1 \epsilon_0$, we have $w_n = \lambda A_{\epsilon_\ell} \cdots A_{\epsilon_1} A_{\epsilon_0} \gamma$. As an illustration

$$w_{14} = \lambda A_1 A_1 A_1 A_0 \gamma = 2.$$

This computation is akin to the definition of recognizable series and indeed B-regular series are merely a translation, as we shall see.

Let the alphabet $\mathcal{X}_{\mathrm{B}}$ be formed of the digits 0, 1, ..., B − 1 used to write the integers in B-ary notation. To avoid confusion between figures and scalars, which lie in a ring $\mathbb{A}$, we represent figures by the indeterminates x_0, x_1, ..., $x_{\mathrm{B}-1}$. We obtain B-regular series by translation of rational series [2].

Definition 1. A formal power series $f(z) \in \mathbb{A}[[z]]$ is a B-regular series if there exists a rational series $S \in \mathbb{A}^{\mathrm{rat}} \langle\langle \mathcal{X}_{\mathrm{B}} \rangle\rangle$ in non-commutating indeterminates, whose support is included in the language $\mathcal{N}$ of integers B-ary expansions,

$$S = \sum_{u \in \mathcal{N}} (S, u)\, u \ ,$$

such that

$$f(z) = \sum_{n \geq 0} (S, \tilde{n})\, z^n \quad ,$$

where $\tilde{n}$ is the B-ary expansion of n.

Linear Representations. In the study of recognizable series, the linear representations come from the use of the division operators that trim a word of its leftmost letter. Classically the divisions are on the left but we favour the right operations, which correspond to the least significant digits. If the alphabet is $\mathcal{X}$ and w is a word, the right division w^{-1} acts on the series S according to the formula

$$Sw^{-1} = \sum_{u \in \mathcal{X}^*} (S, uw)\, u \quad .$$

The division operators give us the section operators S_r, $0 \leq r < \mathrm{B}$, acting on $f(z) = \sum_n f_n\, z^n$ by the formula

$$S_r f(z) = \sum_{n \geq 0} f_{\mathrm{B}n+r}\, z^n \quad .$$

Theorem 2 (Stability theorem). *A formal series is* B*-regular if and only if there exists an* $\mathbb{A}$*-module of finite type which is left stable by the section operators and contains the series.*

We obtain a linear representation of a B-regular series by expressing the section operators with respect to a generating family of that module. Moreover the linear representation permits us to exhibit a rational expression of the series S associated with the B-regular series: if $\Xi = \sum_{0 \leq r < \mathrm{B}} x_r\, A_r$ and $\Xi_+ = \sum_{0 < r < \mathrm{B}} x_r\, A_r$, we have $S = \lambda(I + \Xi_+ \Xi^*)\gamma$. This formula is only a translation of the fact that $\mathcal{N} = \varepsilon + \mathcal{X}_+ \mathcal{X}^*$, where ε is the empty word, $\mathcal{X} = \mathcal{X}_{\mathrm{B}}$ and $\mathcal{X}_+ = \{x_1, \ldots, x_{\mathrm{B}-1}\}$.

Example 2. The complexity of mergesort in the worst case satisfies the divide-and-conquer recurrence

$$T_n = T_{\lfloor n/2 \rfloor} + T_{\lceil n/2 \rceil} + n - 1 \quad ,$$

with the initial conditions $T_0 = T_1 = 0$. The generating series $T(z)$ is 2-regular because the $\mathbb{Z}$-module generated by $T(z)$, $T(z)/z$, $2z/(1-z)^2$, $z(1+z)/(1-z)^2$ and $(1+z)/(1-z)^2$ is left stable by the two section operators S_0 and S_1. With respect to this basis, the matrices of S_0 and S_1 are

$$A_0 = \begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 3 \end{pmatrix}.$$

We take

$$\lambda = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \gamma = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T,$$

because the components of λ are the values at 0 of the series of the basis and γ gives the coordinates of $T(z)$.

Building a linear representation from the section operators gives the relation $\lambda A_0 = \lambda$ because the constant term of a series $g(z)$ is the constant term of $S_0 g(z)$ too. We call such a representation a standard linear representation. We have seen that every B-regular series $f(z)$ hides a rational series $S = \lambda(I + \Xi_+ \Xi^*)\gamma$, but for a standard representation it is simpler to introduce the rational series $R = \lambda\, \Xi^* \gamma$. Both series coincide on language $\mathcal{N} = \varepsilon + \mathcal{X}_+ \mathcal{X}^*$, but the first one extends $f(z)$ by 0 whereas the second one uses the rule $(R, x_0 w) = (R, w)$. Clearly each one determines the other and they have the same rank. By definition this is the rank of the series $f(z)$.

Recurrences. The B-regular series satisfy linear recurrences and the best way to find them is to use their Hankel matrices [5]. For the sake of simplicity, we assume the ring is a field $\mathbb{K}$.

The Hankel matrix of a series $f(z)$ is an infinite matrix whose rows are indexed by the integers and columns are indexed by the words in $\mathcal{X}_{\mathrm{B}}^*$. The columns of the matrix are simply the sequences (f_n), $(f_{\mathrm{B}n})$, $(f_{\mathrm{B}n+1})$, $\ldots$, $(f_{\mathrm{B}n+\mathrm{B}-1})$, $(f_{\mathrm{B}^2 n})$, $\ldots$, if we arrange the words according to their length and lexicographic order.

Definition 3. The Hankel matrix of $f(z) \in \mathbb{K}[[z]]$ is an infinite matrix of type $\mathbb{N} \times \mathcal{X}^*$. The coefficient $H_{n,w}$ of that matrix is $f_{\mathrm{B}^k n + r}$ if w has length k and r is the value of w for radix B.

Clearly a series is B-regular if and only if its Hankel matrix has finite rank. Moreover searching for relations between the columns of the matrix gives us recurrence relations.

Example 3. The van der Corput's sequence associates to an integer n with binary expansion $\epsilon_\ell \ldots \epsilon_0$ the rational number $v_n = \epsilon_0/2 + \epsilon_1/4 + \cdots + \epsilon_\ell/2^{\ell+1}$. It is 2-regular with rank 2 for it satisfies the recurrence

$$v_{2n} = v_n/2, \quad v_{2n+1} = 1/2 + v_n/2 \quad (n \geq 0) \ .$$

Its Hankel matrix begins with

$$\begin{pmatrix}
0 & 0 & 1/2 & 0 & 1/2 & 1/4 & 3/4 \\
1/2 & 1/4 & 3/4 & 1/8 & 5/8 & 3/8 & 7/8 \\
1/4 & 1/8 & 5/8 & 1/16 & 9/16 & 5/16 & 13/16 \\
3/4 & 3/8 & 7/8 & 3/16 & 11/16 & 7/16 & 15/16 \\
1/8 & 1/16 & 9/16 & 1/32 & 17/32 & 9/32 & 25/32 \\
5/8 & 5/16 & 13/16 & 5/32 & 21/32 & 13/32 & 29/32 \\
3/8 & 3/16 & 11/16 & 3/32 & 19/32 & 11/32 & 27/32 \\
7/8 & 7/16 & 15/16 & 7/32 & 23/32 & 15/32 & 31/32
\end{pmatrix} .$$

The two colums with indices ε and x_1 (the first and the third) are independents. Expressing the columns with indices x_0, $x_0 x_1$ and $x_1 x_1$ according to these, we obtain the relations

$$\begin{cases}
v_{2n} &= v_n/2 \ , \\
v_{4n+1} &= -v_n/4 + v_{2n+1} \ , \\
v_{4n+3} &= -v_n/2 + 3\, v_{2n+1}/2 \ ,
\end{cases}$$

which are easy to verify in this case. What we want to emphasize is the shape of these relations and a picture will be clearer than a long comment (see Figure 1).

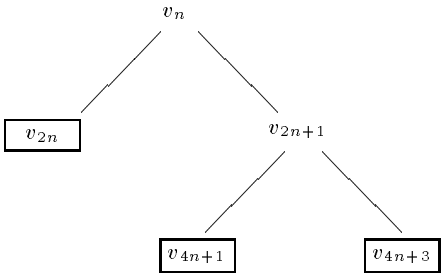


Fig. 1. The leaves of the tree give the shape of the recurrence relations.

This example epitomises the existence of a basis composed with sections $S_w f(z)$, such that the w are the addresses of the internal nodes of a B-ary tree. Furthermore to the leaves of the tree there correspond the recurrence relations; all the recurrences which express linear dependence between the sections are deduced from these [12].

Condensation. If $f(z)$ is B-regular and S is the associated rational series with support in $\mathcal{N} = \varepsilon + \mathcal{X}_+\mathcal{X}^*$, the commutative image [13, p. 147] is a rational series. We call it the condensate of $f(z)$ because it is simply

$$Kf(t) = f_0 + \sum_{l \geq 1} \left(\sum_{\mathrm{B}^{l-1} \leq n < \mathrm{B}^l} f_n \right) t^l \quad .$$

The condensation is useful for regular series just as density is for a regular language.

Example 4. The Taylor series of the logarithm is not B-regular for all B. The condensate of the series

$$\frac{1}{z} \ln \frac{1}{1-z} = \sum_{n \geq 0} \frac{z^n}{n+1}$$

is

$$F(t) = 1 + \sum_{l \geq 1} (H_{\mathrm{B}^l} - H_{\mathrm{B}^{l-1}}) t^l \quad ,$$

with H_n the n-th harmonic number. Using the equality

$$H_{\mathrm{B}^l} - H_{\mathrm{B}^{l-1}} \underset{l \to +\infty}{=} \ln \mathrm{B} + o(1)$$

and the transcendence of $\ln \mathrm{B}$, we see that $F(t)$ is not rational, hence the conclusion.

Closure. The closure properties of rational series show immediately that the set of B-regular series is a module left stable by Hadamard product. Besides, the Cauchy product of two B-regular series is B-regular (assuming that the ring is Noetherian) and a rational function is B-regular if and only if its poles are roots of unity (here we suppose the ring is a field). These properties have been etablished directly by Allouche and Shallit [2], using computation on sequences.

For the sake of simplicity we assume that we use a field in the next theorem.

Theorem 4 (Closure theorem). *A rational function is* B*-regular if and only if its poles are roots of unity. The set of* B*-regular series is closed under*

— *linear combination,*
— *Hadamard product (term by term product),*
— *Cauchy product (function product),*
— *derivation.*

Example 5. Greene and Knuth [11, pp. 25–28] consider the sequence $f(n)$ defined by

$$f(n) = 1 + \min_i \left\{ \frac{i-1}{n} f(i-1) + \frac{n-i}{n} f(n-i) \right\} \ ,$$

which is relative to the search of an integer between 1 and n. The sequence $g(n) = nf(n)$ has second order difference given by

$$\Delta^2 g(n) = \begin{cases} 2 & \text{if } n \text{ is a power of 2} \\ 1 & \text{if } n \text{ is even but not a power of 2} \\ -1 & \text{if } n \text{ odd.} \end{cases}$$

Hence the generating series $g(z)$ is given by

$$g(z) = \frac{1}{(1-z)^2} \left(\frac{1}{1+z} + \sum_{k \geq 0} z^{2^k} \right)$$

and $g(z)$ is 2-regular as sum and product of 2-regular series.

Clearly the subject is not exhausted (we did not speak of Fatou lemma, of properties of coefficients, of decidability questions, *etc*).

2 Mahlerian Series and B-regular Series

As we want to extend the theorem of Christol *et alii* about automatic sequences, we recall at first the subject. Next we establish a general criterion and finally we apply the criterion to four cases:

1. a common case which is very useful because almost all divide-and-conquer recurrences are concerned,
2. the finite field case where we get back the theorem of Christol *et alii*,
3. the modular case, which provides examples where the ring is not an integral domain,
4. the algebraically closed field case, which completes the first case because it permits us to treat more complicated examples.

Let us recall the definition of a B-automatic sequence with values in a set $\mathcal{A}$. First a B-machine is a finite set of states, $\mathcal{S}$, with a distinguished initial state, i, and equipped with transitions $s \mapsto \epsilon.s$ $(0 \leq \epsilon < \mathrm{B})$ from $\mathcal{S}$ into itself. Next we adjoin to this B-machine an application π from $\mathcal{S}$ into $\mathcal{A}$ and so we have a B-automaton. Finally for each integer n, we write its B-ary expansion $\epsilon_\ell \cdots \epsilon_0$ and we compute the state $s = \epsilon_\ell.\cdots.\epsilon_1.\epsilon_0.i$ by going through the automaton from the state i according to the digits of n. The value of the sequence for n is $\pi(s)$.

Clearly the B-automatic sequences with values in a ring are B-regular sequences. The matrices of the transitions, the initial state and the output application provide a linear representation. Conversely a B-regular sequence which takes only a finite number of values is B-automatic.

The theorem under consideration is the next one and has given rise to an extended literature [1, 7].

Theorem 5 (Christol, Kamae, Mendès France, Rauzy). *The generating series of q-automatic sequences with values in the finite field $\mathbb{F}_q$ are exactly the series algebraic over the field $\mathbb{F}_q(z)$ of rational functions.*

This theorem is based on the equality $f(z^q) = f(z)^q$ for a formal series with coefficients in $\mathbb{F}_q$ and this is the reason why algebraic series are in question. In fact the equations which come naturally in light in this situation are Mahlerian equations.

Definition 6. A Mahlerian equation is a functional equation of the form

$$c_0(z)\, f(z) + c_1(z)\, f(z^{\mathrm{B}}) + \cdots + c_N(z)\, f(z^{\mathrm{B}^N}) = b(z) \quad ,$$

where $c_0(z)$, ..., $c_N(z)$ are polynomials. A Mahlerian series is a power series which satisfies a non trivial homogeneous Mahlerian equation.

Our purpose is to extend the theorem to regular series and to separate the radix B and the characteristic m of the ring we use. We show first that every B-regular series is B-mahlerian, at least when the ring is a field. Next we give some criteria which focus on the coefficient $c_0(z)$ and ensure that a solution of the equation is B-regular.

Minimal Equation. Let us assume that the ring is a field $\mathbb{K}$. In this case one can develop an arithmetic for the ring of operators $\mathbb{K}[z, M]$, where M refer to the Mahler operator $f(z) \mapsto f(z^{\mathrm{B}})$. Precisely there is a Euclidean left division, which causes the left ideals to be principal and every Mahlerian series posesses a minimal homogeneous equation [8].

The proof given by Allouche [1] to etablish that a q-automatic series over $\mathbb{F}_q$ is algebraic remains adequate to show that a B-regular series is B-mahlerian. Moreover it often gives a minimal equation for the series if one uses carefully a linear representation of the series. The idea is just to express $f(z)$, $f(z^{\mathrm{B}})$, *etc* in the basis corresponding to the representation and it leads to an effective method of computation.

Example 6. The series $o(z) = \prod_{k \geq 0} \left(1 + 2\, z^{2^k}\right)$ gives the number of odd coefficients in a row of Pascal's triangle [2, ex. 14] [14, seq. 109] [15]. Consequently the complementary series $e(z) = \dfrac{1}{(1-z)^2} - o(z)$ gives the number of even coefficients in a row. This series is 2-regular with rank 3 and a representation is

$$A_0 = \begin{pmatrix} 0 & -2 & -4 \\ 1 & 3 & 4 \\ 0 & 0 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 1 & 3 \end{pmatrix}, \quad \lambda = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}, \quad \gamma = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T.$$

The algorithm gives the equation

$$z^2 e(z) - \left(3\,z^2 - z + 1\right)\left(z^2 + z + 1\right) e(z^2)$$
$$+ \left(3 + 4\,z^2 + 11\,z^4 + 2\,z^8 + 6\,z^6\right) e(z^4) - 2\,\left(2\,z^4 + 1\right)\left(1 + z^4\right)^2 e(z^8) = 0\ .$$

In fact the minimal equation, which is the lcm of the minimal equations for $1/(1-z)^2$ and $o(z)$, is

$$z^2 e(z) - [(1+z^2)^2 + z^2(1+2\,z)]e(z^2) + (1+z^2)^2(1+2\,z^2)e(z^4) = 0\ .$$

Another proof, most in the spirit of this paper, consists in introducing the B-rational operators

$$F = \sum_{k \geq 0} c_k(z)\, M^k \in \mathbb{K}[[z, M]]\ ,$$

which are the images of the rational series S with support in $\mathcal{N} = \varepsilon + \mathcal{X}_+\mathcal{X}^*$ by the anti-morphism which associates to the letter x_r the operator $z^r M$. They are the natural intermediate between the rational series and the B-regular series, since every B-regular series is the value of a rational operator at the series 1. Using the closure properties of rational series and the arithmetic of operators, it is not difficult to prove that every B-rational operator satisfies an equality $QF = P$ where Q and P are two members of $\mathbb{K}[z, M]$ with the constraint $Q \neq 0$ and $\omega_M(Q) = 0$ (Q is a polynomial with respect to z and M and $\omega_M(Q)$ is the valuation of Q according to M). Now if $f(z)$ is a B-regular series it is written $f(z) = F.1$ where F is a rational operator; taking for Q a denominator of F, we have $Qf(z) = P.1$ hence a Mahlerian equation where the second member is a polynomial; it is not difficult to render it homogeneous.

General Criterion. For the rest of the paper we study the converse of the preceding property and we give first a general criterion to ensure that the solutions of a Mahlerian equation are B-regular.

Let us consider a Mahlerian equation

$$c_0(z)f(z) + c_1(z)f(z^{\mathrm{B}}) + \cdots + c_N(z)f(z^{\mathrm{B}^N}) = b(z)$$

where $b(z)$ is a B-regular series. We assume that the ring $\mathbb{A}$ is Noetherian and the coefficient of lowest degree in $c_0(z)$ is inversible in $\mathbb{A}$: we have $c_0(z) = Cz^\gamma g(z)$ with C inversible, γ a non negative integer and $g(0) = 1$. These constraints are normally fulfilled but we need to add the main condition: the set of the sections

$$S_{r_K} \cdots S_{r_1}\left(\frac{1}{g(z^{\mathrm{B}^{K-1}}) \cdots g(z^{\mathrm{B}})g(z)}\right) = S_{r_K}\frac{1}{g}\left(S_{r_{K-1}}\frac{1}{g}\left(\cdots S_{r_1}\left(\frac{1}{g}\right)\right)\right),$$

where $K \geq 0$, $0 \leq r_k < \mathrm{B}$ for $k = 1, \cdots, K$, is contained in a module of finite type. With these hypotheses a solution $f(z)$ of the equation is B-regular.

As we impose a condition only on coefficient c_0 and nothing on $c_1, \ldots, c_N$, there is no hope to find a necessary and sufficient condition. Nevertheless the hypothesis about the set of sections which appears in the criterion is exactly the condition which ensures that the Mahlerian infinite product

$$f(z) = \prod_{k \geq 0} \frac{1}{g(z^{\mathrm{B}^k})}$$

is B-regular.

Common Case. If $g(z) = 1$, the main condition vanishes and we have an easy criterion to recognize a B-regular series. The case contains almost all the divide-and-conquer recurrences and in view of its importance, we extend the result to study vector of series instead of series. This permits us to treat sequences which admits a definition by case according to the residue modulo a power of B, say B^{k+1}, which expresses $\mathrm{B}^{k+1}n + r$ according to the $\mathrm{B}^l n + s$ with $0 \leq l \leq k$. The next assertion uses a natural extension of B-regularity to vector of series.

Theorem 7 (Common case). *We consider a vector of series*

$$F(z) = (\ f_1(z)\ \ \ldots\ f_d(z)\ )^T$$

and we assume the following hypothesis:
— *the ring is Noetherian,*
— *the vector of series satisfies an equation*

$$z^\gamma F(z) + \sum_{k=1}^{N} C_k(z) F(z^{\mathrm{B}^k}) = B(z)$$

where $\gamma \geq 0$, $C_1(z)$, ..., $C_N(z)$ are some square matrices of polynomials and $B(z)$ is a column matrix whose components are B*-regular series.*

With these conditions, the components of $F(z)$ are B*-regular series.*

Example 7. Supowit and Reingold [16] encountered the sequence (C_n) defined by the recurrence

$$\begin{cases} C_{4n} &= a(C_{2n+1} + C_{2n-1}) + b \\ C_{4n+1} &= a(C_{2n+1} + C_{2n}) \\ C_{4n+2} &= a(C_{2n+1} + C_{2n+1}) + b \\ C_{4n+3} &= a(C_{2n+2} + C_{2n+1}) \end{cases}$$

for $n \geq 1$ and the initial conditions $C_0 = C_1 = 0$, $C_2 = b$, $C_3 = ab$, with $a = 1/\sqrt{2}$ and $b = \sqrt{3}$. The number b is only a scale factor and with a division by b we may suppose $b = 1$.

We call $f(z)$ the generating series of (C_n) and we refer to the section $S_w f(z)$ as $f_w(z)$. The recurrence gives us the system

$$\begin{cases} f_{00}(z) &= a(1+z)f_1(z) + 1/(1-z) \\ f_{01}(z) &= af_1(z) + af_0(z) \\ f_{10}(z) &= 2af_1(z) + 1/(1-z) \\ f_{11}(z) &= af_0(z)/z + af_1(z)\ . \end{cases}$$

If we express $f_0(z)$ and $f_1(z)$ with respect to $f_{00}(z)$, $f_{01}(z)$, $f_{10}(z)$ and $f_{11}(z)$ as $f_\epsilon(z) = f_{0\epsilon}(z^2) + z f_{1\epsilon}(z^2)$, we obtain an equation

$$F(z) = a\, C_1(z) F(z^2) + B(z)$$

in which the unknown is the vector $F(z) = (\ f_{00}(z)\ \ f_{01}(z)\ \ f_{10}(z)\ \ f_{11}(z)\ )^T$ and the coefficients are given by

$$C_1(z) = \begin{pmatrix} 0 & 1+z & 0 & z(1+z) \\ 1 & 1 & z & z \\ 0 & 2 & 0 & 2z \\ 1/z & 1 & 1 & z \end{pmatrix}, \quad B(z) = \begin{pmatrix} 1/(1-z) \\ 0 \\ 1/(1-z) \\ 0 \end{pmatrix}.$$

In accordance with our result, we may assert that $F(z)$ and hence $f(z)$ is 2-regular.

Finite Fields and Rings. Let $p(z) \in \mathbb{A}[z]$ be a polynomial such that $p(0) = 1$. We say that T is the period of $p(z)$ if the sequence of coefficients of the formal power series $1/p(z)$ is periodic with period T. The study of the period [4] of

$$g(z^{\mathrm{B}^{K-1}}) \cdots g(z^{\mathrm{B}})g(z)$$

provide us with cases in which we can guarantee that the main condition is satisfied.

Theorem 8 (Finite field). *Let a formal series $f(z)$ have coefficients in the field $\mathbb{F}_q$ with characteristic p and satisfy a Mahlerian equation whose right-hand side is* B*-automatic*

$$c_0(z)\, f(z) + c_1(z)\, f(z^{\mathrm{B}}) + \cdots + c_N(z)\, f(z^{\mathrm{B}^N}) = b(z) \quad .$$

We assume that $c_0(z) = C\, z^{\gamma} g(z)$ with $\gamma \geq 0$, $g(0) = 1$. If p divides B *or if the period T of $g(z)$ and the radix* B *have a common prime divisor, other than the characteristic p, then $f(z)$ is* B*-automatic.*

It is worth noting that $g(z)$ does not matter in the first condition about B. This case extends directly the theorem of Christol, Kamae, Mendès France and Rauzy.

Example 8. The polynomial $g(z) = 1 + z^2 + z^3$, which lies in $\mathbb{F}_2[z]$, is 7-periodic. Hence a formal series $f(z) \in \mathbb{F}_4[[z]]$ which satisfies a Mahlerian equation of the shape

$$z^{1993}(1 + z^2 + z^3)f(z) + c_1(z)f(z^{21}) + c_2(z)f(z^{441}) = 0$$

is 21-regular. (Here $p = 2$, $q = 4$, $T = 7$ and $\mathrm{B} = 21$.)

Starting from these results for the fields $\mathbb{F}_p$, it is not difficult to attain the quotient rings $\mathbb{Z}/(p^a)$. In fact if $g(z)$ has period t modulo p^a, it has period pt modulo p^{a+1}. Next the chinese remainder theorem permits us to consider rings $\mathbb{Z}/(m)$.

Theorem 9 (Modular case). *Let $f(z) \in \mathbb{Z}/(m)[[z]]$ be a formal series which satisfies*

$$c_0(z)f(z) + c_1(z)f(z^{\mathrm{B}}) + \cdots + c_N(z)f(z^{\mathrm{B}^N}) = b(z)$$

with right-hand side $b(z)$ B*-automatic, $c_0(z) = C\, z^{\gamma} g(z)$, C invertible, $\gamma \geq 0$ and $g(0) = 1$. We assume that for every prime divisor p of m, one of the next two conditions is satisfied: i) p divides* B*,or ii) there exists a prime number p' which is different from p and divides both the radix* B *and the period $T(g, p)$ of $g(z)$ reduced modulo p. Then $f(z)$ is* B*-automatic.*

Example 9. Let us consider the integer sequence (u_n) defined by the initial conditions $u_0 = 0$, $u_1 = 1$ and the recurrence relation

$$u_n = u_{n-1} + u_{n-2} + u_{\lfloor n/2 \rfloor} \quad .$$

Clearly u_n is greater than the Fibonacci number F_{n-1} and the generating series

$$u(z) = z + 2\, z^2 + 4\, z^3 + 8\, z^4 + 14\, z^5 + 26\, z^6 + 44\, z^7 + 78\, z^8 + \cdots$$

is not 2-regular because its coefficients grow too rapidly. Nevertheless it is 2-regular when we reduce it modulo every integer. It suffices to look at the primary numbers p^a. If $p = 2$ the result is immediatly obtained for p equals B. Otherwise it suffices to remark that the period of $1 - z - z^2$ modulo an odd prime is even, because the Mahlerian equation which is to be considered is

$$(1 - z - z^2)u(z) - (1 + z)u(z^2) = z \ .$$

Example 10. A B-ary partition is an integer partition in which the parts are power of B. As an illustration there are nine 3-partitions of 16, namely 1^{16}, $1^{13}3$, $1^{10}3^2$, $1^7 3^3$, $1^4 3^4$, 13^5, $1^7 9$, $1^4 3\,9$, $13^2 9$ (we use the classical notation: $13^2 9$ refers to $1 + 3 + 3 + 9$). The generating function of the number of B-ary partition is [3, p. 161]

$$p(z) = \prod_{k=0}^{+\infty} \frac{1}{1 - z^{\mathrm{B}^k}}$$

and it satisfies the Mahlerian equation

$$(1 - z)p(z) = p(z^{\mathrm{B}}) \ .$$

Because the period of $g(z) = 1 - z$ is 1 modulo every integer, we cannot use the second condition of our theorem, but the first one shows that $p(z)$ is B-regular if we reduce it modulo m and every prime divisor of m divides B. As an example the number of binary partition reduced modulo 8 may be defined by the 2-automaton

$$A_0 = \begin{pmatrix} 0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0 \\ 0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0 \\ 0&1&0&0&0&1&1 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0 \end{pmatrix},$$

$$\lambda = (\ 1\ \ 1\ \ 0\ \ 4\ \ 2\ \ 0\ \ 6\ ), \quad \gamma = (\ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ )^T .$$

Algebraically Closed Field. Finally we apply our criterion to algebraically closed fields. Here the trick to obtain the main condition is to impose that

$$S_{r_K} \cdots S_{r_1} \left(\frac{1}{g(z^{\mathrm{B}^{K-1}}) \cdots g(z^{\mathrm{B}}) g(z)} \right)$$

have poles in a finite set with bounded multiplicities. This guarantees that they lie in a vector space of finite dimension. We obtain the following theorem.

Theorem 10 (Algebraically closed field). *Let $f(z)$ be a formal series with coefficients in an algebraically closed field. We assume that $f(z)$ satisfies a Mahlerian equation*

$$c_0(z)f(z) + c_1(z)f(z^{\mathrm{B}}) + \cdots + c_N(z)f(z^{\mathrm{B}^N}) = b(z)$$

in which $b(z)$ is B*-regular, $c_0(z) = C\, z^\gamma g(z)$ with $C \neq 0$, $\gamma \geq 0$ and $g(0) = 1$. If all the roots of $g(z)$ are roots of unity with an order (in the sense of group theory) which is not prime relative to* B*, then $f(z)$ is* B*-regular.*

Example 11. Let us consider the integer sequence (u_n) defined by $u_0 = 0$, $u_1 = 1$ and the recurrence

$$u_n = u_{n-1} - u_{n-2} + u_{\lfloor n/3 \rfloor} \quad (n \geq 2) \ .$$

Its generating function $u(z)$ is the solution of

$$(1 - z + z^2)u(z) - u(z^3) = z \ .$$

The roots of $1 - z + z^2$ are the primitive 6-th roots of unity, hence $u(z)$ is 3-regular. Besides its rank is 3. Moreover it is 3-automatic according to the equality

$$u(z) = (1 + z) \sum_{k,l \geq 0} (-1)^l z^{3^k(3l+1)} \ .$$

Acknowledgement. This work was (partially) supported by the ESPRIT Basic Research Action Nr. 7141 (ALCOM II).

References

1. J.-P. Allouche. Automates finis en théorie des nombres. *Expositiones Mathematicae*, 5:239–266, 1987.
2. J.-P. Allouche and J. Shallit. The ring of k-regular sequences. *Theoretical Computer Science*, 98:163–197, 1992.
3. G. E. Andrews. *The Theory of Partitions*, volume 2 of *Encyclopedia of Mathematics and its Applications*. Addison–Wesley, 1976.
4. E. R. Berlekamp. *Algebraic Coding Theory*. Mc Graw-Hill, revised 1984 edition, 1968.
5. J. Berstel and Ch. Reutenauer. *Rational series and their languages*, volume 12 of *EATCS monographs on theoretical computer science*. Springer, 1988.
6. G. Christol, T. Kamae, M. Mendès France, and G. Rauzy. Suites algébriques, automates et substitutions. *Bulletin de la Société Mathématique de France*, 108:401–419, 1980.
7. M. Dekking, M. Mendès France, and A. Van der Poorten. Folds! *Mathematical Intelligencer*, 4:130–138, 173–181, 190–195, 1982.
8. Philippe Dumas. *Récurrences Mahlériennes, suites automatiques, et études asymptotiques*. Doctorat de mathématiques, Université de Bordeaux I, 1993.
9. Philippe Flajolet and Mordecai Golin. Exact asymptotics of divide–and–conquer recurrences. Proceedings of ICALP'93, Lund., July 1993. This volume.
10. Philippe Flajolet, Peter Grabner, Peter Kirschenhofer, Helmut Prodinger, and Robert Tichy. Mellin transforms and asymptotics: Digital sums, July 1991. 23 pages. INRIA Research Report. Accepted for publication in *Theoretical Computer Science*.
11. D. H. Greene and D. E. Knuth. *Mathematics for the analysis of algorithms*. Birkhauser, Boston, 1981.
12. Ch. Reutenauer. Séries rationnelles et algèbres syntactiques. Master's thesis, Université Pierre et Marie Curie (Paris VI), 1980.
13. A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer, Berlin, 1978.
14. N. J. A. Sloane. *A Handbook of Integer Sequences*. Academic Press, 1973.
15. Kenneth B. Stolarsky. Power and exponential sums of digital sums related to binomial coefficients. *SIAM Journal on Applied Mathematics*, 32(4):717–730, 1977.
16. K. J. Supowit and E. M. Reingold. Divide and conquer heuristics for minimum weighted Euclidean matching. *SIAM Journal on Computing*, 12(1):118–143, February 1983.

BOLTZMANN SAMPLERS FOR THE RANDOM GENERATION OF COMBINATORIAL STRUCTURES

PHILIPPE DUCHON, PHILIPPE FLAJOLET, GUY LOUCHARD, GILLES SCHAEFFER

ABSTRACT. This article proposes a surprisingly simple framework for the random generation of combinatorial configurations based on what we call *Boltzmann models*. The idea is to perform random generation of possibly complex structured objects by placing an appropriate measure spread over the whole of a combinatorial class—an object receives a probability essentially proportional to an exponential of its size. As demonstrated here, the resulting algorithms based on real-arithmetic operations often operate in linear time. They can be implemented easily, be analysed mathematically with great precision, and, when suitably tuned, tend to be very efficient in practice.

1. INTRODUCTION

In this study, *Boltzmann models* are introduced as a framework for the random generation of structured combinatorial configurations, like words, trees, permutations, constrained graphs, and so on. A Boltzmann model relative to a combinatorial class $\mathcal{C}$ depends on a *real-valued* (continuous) control parameter $x > 0$ and places an appropriate measure that is spread over the whole of $\mathcal{C}$: This measure is essentially proportional to $x^{|\omega|}$ for an object $\omega \in \mathcal{C}$ of size $|\omega|$. Random objects under a Boltzmann model then have a fluctuating size, but objects with the same size invariably occur with the same probability. In particular, a *Boltzmann sampler* (i.e., a random generator that produces objects distributed according to a Boltzmann model) draws *uniformly* at random an object of size n, when the size of its output is conditioned to be the fixed value n.

As we demonstrate, Boltzmann samplers can be derived systematically (and simply) for classes that are specified in terms of a basic collection of general-purpose combinatorial constructions. These constructions are precisely the ones that surface recurrently in modern theories of combinatorial analysis [4, 28, 30, 60, 61] and in systematic approaches to random generation of combinatorial structures [29, 51]. As a consequence, one obtains with surprising ease Boltzmann samplers covering an extremely wide range of combinatorial types.

In most of the combinatorial literature so far, fixed-size generation has been the standard paradigm for the random generation of combinatorial structures, and a vast literature exists on the subject. There, either specific bijections are exploited or general combinatorial decompositions are put to use in order to generate objects at random based on counting possibilities—the latter approach has come to be known as the "recursive method" originating with Nijenhuis and Wilf [51], then systematized and extended by Flajolet, Zimmermann, and Van Cutsem in [29].

Date: Version of January 1,2003. Submitted to *Combinatorics, Probability, and Computing.*

In contrast, the basic principle of Boltzmann sampling is to *relax* the constraint of generating objects of a strictly fixed size, and prefer to draw objects with a randomly varying size. As we shall see, normally, one can then *tune* the value of the control parameter x in order to favour objects of a size in the vicinity of a target value n. (A "tolerance" of, say, a few percents on size of the object produced is likely to cater for many practical simulation needs.) If the tuning mentioned above is not sufficient, one can always pile up a rejection method to restrict further the size of the element drawn. In this way, Boltzmann samplers may be employed for approximate-size as well as fixed-size random generation.

We propose Boltzmann samplers as an attractive alternative to standard combinatorial generators based on the recursive method and implemented in packages like Combstruct (under the computer algebra system MAPLE) and CS (under MUPAD). The algorithms underlying the recursive necessitate a preprocessing phase where tables of integer constants are set up, then they appeal to a boustrophedonic strategy in order to draw a random object of size n. In the abstract, the *integer-arithmetic* complexities attached to the recursive method and measured by the number of (large) *integer-arithmetic* operations are as follows:

(1)

Preproc. memory	*Preproc. time*	*Time per generation*
$O(n)$	$O(n^2)$ or $O(n^{1+\varepsilon})$	$O(n \log n)$

The integer-based algorithms require the costly maintenance of large tables of constants (in number $O(n)$). In fact, they effect arithmetic operations over large multiprecision integers, which themselves have size $O(n)$ (in the unlabelled case) or $O(n \log n)$ (in the labelled case); see [29]. Consequently, the overall Boolean complexities involve an extra factor of $O(n)$ at least, leading to a cost measured in elementary operations that is quadratic or worse. (The integer-arithmetic time of the preprocessing phase could in principle be decreased from $O(n^2)$ to $O(n^{1+\varepsilon})$ thanks to the recent work of van der Hoeven [65], but this does not affect our basic conclusions.) An alternative, initiated by Denise, Dutour, and Zimmermann [12, 13], consists in treating integers as real numbers and approximating them using real arithmetics ("floating-point" implementations), possibly supplementing the technique by adaptive precision routines. In the case of real-based algorithms, the Boolean as well as practical complexities improve, and they become fairly well represented by the data of Equation (1), but the memory and time of the preprocessing phase remains fairly large, while the time per generation remains inherently superlinear.

As we propose to show, Boltzmann algorithms can well be competitive when compared to combinatorial methods: Boltzmann samplers only necessitate a small *fixed* number of low precision real constants that are normally easy to compute while their complexity is always linear in the size of the object drawn. Accordingly, uniform random generation of objects with sizes in the range of millions is becoming a possibility, whenever the Boltzmann framework is applicable. The price to be paid is an occasional loss of certainty in the exact size of the object generated, typically, a *tolerance* on sizes of a few percents should be granted; refer to Figure 10 in the concluding section. The table that summarizes the complexities of Boltzmann generators, measured in *real-arithmetic* operations is then:

(2)

Preproc. memory	*Preproc. time*	*Time per generation*
$O(1)$	"small"	with tolerance : $O(n)$

The vague qualifier "small" refers to the fact that practical implementations will be based on floating point approximations to exact real number arithmetics, in which case, typically, the preprocessing time is likely to be a small power of $\log n$. (That this preprocessing is practically feasible and of a very low complexity should at least transpire from the various examples given, but a systematic discussion would carry us too far away from our main objectives[1].)

As regards random generation, the ideas presented here draw their origins from many sources. First the recursive method of [29, 51] served as a key conceptual guide for delineating the types of objects that are systematically amenable to Boltzmann sampling. Ideas from a statistical physics point of view on combinatorics, of which great use was made by Vershik and his collaborators [10, 67], then provided crucial insight regarding the new class of algorithms for random generation that is presented here. Another important ingredient is the collection of rejection algorithms developed by Duchon, Louchard, and Schaeffer for certain types of trees, polyominos, and planar maps [17, 45, 56]. There are also similarities with the technique of "shifting the mean" (see Greene and Knuth's book [33, p. 78–80]) as well as the theory of large deviations [11] and "exponential families" of probability theory—we have benefited from discussions with Alain Denise on these aspects. Finally, the principles of analytic combinatorics (see [28]) provide essential clues for deciding situations in which the algorithms are likely to be efficient. Further connections are discussed at the end of the next section.

Plan of this study. Boltzmann models and samplers are introduced in Section 2. Boltzmann models exist in two varieties: the ordinary and the exponential models. Ordinary models serve for combinatorial classes that are "unlabelled", the corresponding samplers being developed in Section 3, where basic construction rules are described. Section 4 proceeds in a parallel way with exponential models and "labelled" classes. Some of the complexity issues raised by Boltzmann sampling are examined in Section 5. There it is shown that, at least in the idealized sense of *exact real-number computations*, a Boltzmann sampler suitably equipped with a fixed (and small) number of driving constants operates in time that is *linear* in the (fluctuating) size of the object it produces.

Sections 2 to 5 develop Boltzmann samplers that operate *freely* under the sole effect of the defining parameter x. We examine next the way the control parameter x can be *tuned* to attain objects at or near a target value: this is the subject of Section 6, where rejection is introduced and a technique based on the pointing transformation is developed. Section 7 describes two types of situation where the basic Boltzmann samplers turn out to be *optimized* by assigning a critical value to the control parameter x. Section 8 offers a few concluding remarks.

An extended abstract summarizing several of the results described here has been presented at the ICALP'2002 Conference in Malaga [18].

2. Boltzmann models and samplers

We consider a class $\mathcal{C}$ of combinatorial objects of sorts, with $|\cdot|$ the size function mapping $\mathcal{C}$ to $\mathbb{Z}_{\geq 0}$. By $\mathcal{C}_n$ is meant the subclass of $\mathcal{C}$ comprising all the objects in $\mathcal{C}$ having size n, and each $\mathcal{C}_n$ is assumed to be finite. One may think of binary

[1]The primary goal of this article is on practical algorithmic design, *not* analysis of algorithms, although a fair amount of analysis, by necessity, enters into the discussion.

words (with size defined as length), permutations, graphs and trees of various types (with size defined as number of vertices), and so on. Any set $\mathcal{C}$ endowed with a size function and satisfying the finiteness axiom will henceforth be called a *combinatorial class*.

The *uniform probability distribution* over $\mathcal{C}_n$ assigns to each $\gamma \in \mathcal{C}_n$ the probability

$$\mathbb{P}_{\mathcal{C}_n}\{\gamma\} = 1/C_n,$$

with $C_n := \mathrm{card}(\mathcal{C}_n)$. *Exact-size* random generation means the process of drawing **uniformly** at random from the class $\mathcal{C}_n$. We also consider (see Sections 6 and 7 for a description of various strategies) random generation from "neighbouring classes", $\mathcal{C}_N$ where N may not be totally under control, but should still be in the vicinity of n, namely, in some interval $(1-\varepsilon)n \le N \le (1+\varepsilon)n$, for some "*tolerance*" factor $\varepsilon > 0$; this is called *approximate-size* (uniform) random generation. It must be stressed that, even under approximate-size random generation, *two objects of the same size are invariably drawn with the same probability.*

Definition 1. *The* Boltzmann models *of parameter x exist in two varieties, the ordinary version and the exponential version. They assign to any object $\gamma \in \mathcal{C}$ the following probability:*

$$\textit{Ordinary/Unlabelled case:} \qquad \mathbb{P}_x(\gamma) = \frac{1}{C(x)} \cdot x^{|\gamma|} \quad \textit{with} \quad C(x) = \sum_{\gamma \in \mathcal{C}} x^{|\gamma|},$$

$$\textit{Exponential/Labelled case:} \qquad \mathbb{P}_x(\gamma) = \frac{1}{\widehat{C}(x)} \cdot \frac{x^{|\gamma|}}{|\gamma|!} \quad \textit{with} \quad \widehat{C}(x) = \sum_{\gamma \in \mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!}.$$

A Boltzmann sampler (or generator) $\Gamma C(x)$ *for a class $\mathcal{C}$ is a process that produces objects from C according to the corresponding Boltzmann model, either ordinary or exponential.*

The normalization coefficients are nothing but the values at x of the counting generating functions, respectively of ordinary type (OGF) for C and exponential type (EGF) for $\widehat{C}$:

$$C(z) = \sum_{n \ge 0} C_n z^n, \qquad \widehat{C}(z) = \sum_{n \ge 0} C_n \frac{z^n}{n!}.$$

Coherent values of x defined to be such that $0 < x < \rho_C$ (or $\rho_{\widehat{C}}$), with ρ_f the radius of convergence of f are to be considered. The quantity ρ_f is referred to as the "critical" or "singular" value. (In the particular case when the generating function $C(x)$ still converges at ρ_C, one may also use the limit value $x = \rho_C$ to define a valid Boltzmann model; see Section 7 for uses of this technique.)

For reasons which will become apparent, we have introduced two categories of models, the ordinary and exponential ones. Exponential Boltzmann models are appropriate for handling *labelled* combinatorial structures while ordinary models correspond to *unlabelled* structures of combinatorial theory[2]. In the unlabelled universe, all elementary components of objects ("atoms") are indistinguishable, while in the labelled universe, they are all distinguished from one another by bearing a distinctive mark, say one of the integers between 1 and n if the object considered

[2]This terminology is standard in combinatorial enumeration and graph theory; see, e.g., the books of Bergeron et al., Goulden–Jackson, Harary–Palmer, Stanley, and Wilf [4, 30, 34, 60, 61, 69] or the preprints by Flajolet & Sedgewick [28].

has size n. Permutations written as sequences of distinct integers are typical labelled objects while words over a binary alphabet appear as typical unlabelled objects made of "anonymous" letters, say $\{a, b\}$ for a binary alphabet.

For instance, consider the (unlabelled) class $\mathcal{W}$ of all binary words, $\mathcal{W} = \{a, b\}^\star$. There are $W_n = 2^n$ words of length n and the OGF is $W(z) = (1 - 2z)^{-1}$. The probability assigned by the ordinary Botzmann model to any word w is $x^{|w|}(1-2x)$. There, the coherent values of x are all the positive values less than the critical value $\rho_W = \frac{1}{2}$. The probability that a word of length n is selected is $(2x)^n(1 - 2x)$, so that the Boltzmann model of binary words is logically equivalent to the following process: draw a random variable N according to the geometric distribution of parameter $2x$; if the value $N = n$ is obtained, draw uniformly at random any of the possible words of size n. For the labelled case, consider the class K of all cyclic permutations, $\mathcal{K} = \{[1], [1\,2], [1\,2\,3], [1, 3, 2], \ldots\}$. There are $K_n = (n - 1)!$ cyclic permutations of size n over the canonical set of "labels" $\{1, \ldots, n\}$. The EGF is

$$\widehat{K}(z) = \sum_{n\geq 1}(n-1)!\frac{z^n}{n!} = \sum_{n\geq 1}\frac{z^n}{n} = \log\frac{1}{1-z}. \tag{3}$$

The probability of drawing a cyclic permutation of some fixed size n is then,

$$\frac{1}{\log(1-x)^{-1}}\frac{x^n}{n}, \tag{4}$$

a quantity defined for $0 < x < \rho_{\widehat{K}} = 1$. (This is known as the "logarithmic series distribution"; see Section 4). Like in the case of binary words, the Boltzmann model can thus be realized by first selecting size according to the logarithmic series distribution, and then by drawing uniformly at random a cyclic permutation of the chosen size. We are precisely going to *revert* this process and show that, in many cases, it is of advantage to draw *directly* from a Boltzmann model, (Sections 3 to 5), and from there derive random generators that are efficient for a given range of sizes (Sections 6 and 7).

The size of the resulting object under a Boltzmann model is a random variable denoted throughout by N. By construction, the probability of drawing an object of size n is, under the model of index x,

$$\mathbb{P}_x(N = n) = \frac{C_n x^n}{C(x)}, \quad \text{or} \quad \mathbb{P}_x(N = n) = \frac{C_n x^n}{n!\widehat{C}(x)}, \tag{5}$$

for the ordinary and exponential model, respectively. The law is well quantified by the following lemma. (See, e.g., Huang's book [37] for similar calculations from the statistical mechanics angle.)

Proposition 1. *The random size of the object produced under the ordinary Boltzmann model of parameter x has first and second moments satisfying*

$$\mathbb{E}_x(N) = x\frac{C'(x)}{C(x)}, \qquad \mathbb{E}_x(N^2) = \frac{x^2C''(x) + xC'(x)}{C(x)}. \tag{6}$$

The same expressions are valid, but with $\widehat{C}$ replacing C, in the case of the exponential Boltzmann model. In both cases, the expected size $\mathbb{E}_x(N)$ is an increasing function of x.

Proof. Under the ordinary Boltzmann model, the probability generating function of N is

$$\sum_n \mathbb{P}_x(N=n)z^n = \frac{C(xz)}{C(x)},$$

by virtue of (5). The result then immediately follows by differentiation upon setting $z=1$:

$$\mathbb{E}_x(N) = \left(\frac{\partial}{\partial z}\frac{C(xz)}{C(x)}\right)_{z=1}, \qquad \mathbb{E}_x\left(N(N-1)\right) = \left(\frac{\partial^2}{\partial z^2}\frac{C(xz)}{C(x)}\right)_{z=1}.$$

The very same calculation applies to exponential Boltzmann models, but with the EGF $\widehat{C}$ then replacing the OGF C.

The mean size $\mathbb{E}_x(N)$ is always a strictly increasing function of x as soon as the class $\mathcal{C}$ contains at least two elements of different sizes. Indeed one verifies by a trite calculation the identity

$$x\frac{d}{dx}\mathbb{E}_x(N) = \mathbb{V}_x(N),$$

where $\mathbb{V}$ denote the variance operator. Since the variance of a nondegenerate random variable is always strictly positive the derivative of $\mathbb{E}_x(N)$ is positive and $\mathbb{E}_x(N)$ is increasing. (This property is in fact a special case of Hadamard's convexity theorem.) ∎

For instance, in the case of binary words, the coherent choice $x = 0.4$ leads to a size with mean value 4 and standard deviation about 4.47; for $x = 0.49505$, the mean and standard deviation of size become respectively 100 and 100.5. For cyclic permutations, we determine similarly that the choice $x = 0.99846$ leads to an object of mean size equal to 100, while the standard deviation is somewhat higher than for words, being equal to 234. In general, the distribution of random sizes under a Boltzmann model, as given by Formula (5), strongly depends on the family under consideration. Figure 1 illustrates three widely differing profiles: for set partitions, the distribution is "bumpy", so that a choice of the appropriate x will most likely generate an object close to the desired size; for surjections (whose behaviour is analogous to the one of binary words), the distribution becomes fairly "flat" as x nears the critical value; for trees, it is "peaked" at the origin, so that very small objects are generated with high probability. It is precisely the purpose of later sections (Sections 6 and 7) to recognize and exploit the "physics" of these distributions in order to deduce efficient samplers for exact and approximate size random generation.

Relation to other fields. The term "Boltzmann model" comes from the great statistical physicist Ludwig Boltzmann whose works (together with those of Gibbs and Maxwell) led to enunciate the following principle: *Statistical mechanical configurations of energy equal to E in a system have a probability*[3] *of occurrence proportional to $e^{-\beta E}$.* If one identifies size of a combinatorial configuration with energy of a thermodynamical system and sets $x = e^{-\beta}$, then what we term the ordinary Boltzmann models become the usual model of statistical mechanics. The counting generating function in the combinatorial world then coincides with the normalization constant in the statistical mechanics world where it is known as the *partition*

[3]Distributions of the type $e^{-\beta E}$ play an important rôle in the study of point processes and they tend to be known to probabilists under the name of "Gibbs measures".

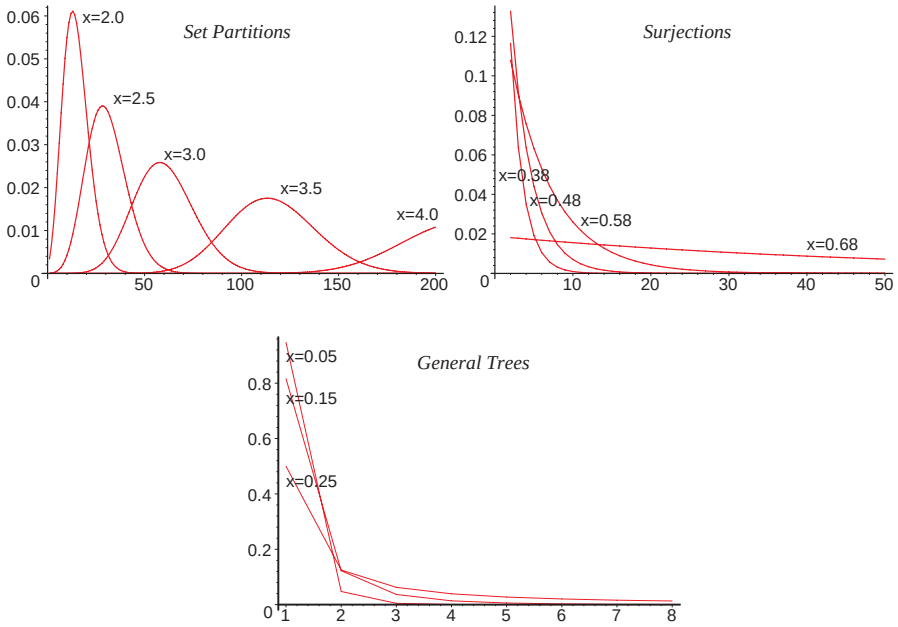


FIGURE 1. Size distributions under Boltzmann models for various values of parameter x. From top to bottom: the "bumpy" type of set partitions (Example 5), the "flat" type of surjections (Example 6), and the "peaked" type of general trees. (Example 2).

function—the *Zustandsumme* often denoted by Z. (Note: In statistical mechanics, $\beta = 1/(kT)$ is an inverse temperature. Thus situations where $x \to 0$ formally correspond to low temperatures or "freezing" and give more weight to small structures, while $x \to \rho^-$ corresponds to high temperatures or "melting", that is, to larger sizes of the combinatorial configurations being generated.)

Exponential weights of the Boltzmann type are naturally essential to the simulated annealing approach to combinatorial optimization. In the latter area, for instance, Fill and Huber [22] have shown the possibility of drawing at random independent sets of graphs according to a Boltzmann distribution, at least for certain values of the control parameter $x = e^{-\beta}$. Closer to us, Compton [7, 8] has made an implicit use of what we call Boltzmann models for the analysis of 0–1 laws and limit laws in logic; see also the account by Burris [6]. Vershik has initiated in a series of papers (see [67] and references therein) a programme that can be described in our terms as first developing the probabilistic study of combinatorial objects under a Boltzmann model and then "returning" to fixed size statistics by means of Tauberian arguments of sorts. (A similar description can be applied to Compton's approach; see especially the work [50] for recent developments in this direction.) As these examples indicate, the general idea of Boltzmann models is certainly not new, and, in this work, we may at best claim originality for aspects related to the fast random generation of combinatorial structures.

Construction		*Generator*
Singleton	$\mathcal{C} = \{\omega\}$	$\Gamma C(x) = \omega$
Union	$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma C(x) = \left(\text{Bern}\left(\frac{A(x)}{A(x)+B(x)}\right) \longrightarrow \Gamma A(x) \mid \Gamma B(x) \right)$
Product	$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$\Gamma C(x) = \big(\Gamma A(x); \Gamma B(x)\big)$
Sequence	$\mathcal{C} = \mathfrak{S}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Geom}(A(x)) \Longrightarrow \Gamma A(x)\big)$

FIGURE 2. The inductive rules for ordinary Boltzmann samplers.

3. Ordinary Boltzmann Generators

In this section and the next one, we develop a collection of rules by which one can assemble Boltzmann generators from simpler ones. The combinatorial classes considered are built by means of a small set of constructions that have wide expressive power. The language in which classes are specified is in essence the same as the one underlying the recursive method [29]: it includes the constructions of union, product, sequence, and, in the labelled case treated in the next section, the additional set and cycle constructions. For each allowable class, a Boltzmann sampler can be derived in an entirely systematic (and even automatic) manner.

A *combinatorial construction* builds a new class $\mathcal{C}$ from structurally simpler classes $\mathcal{A}, \mathcal{B}$, in such a way that $\mathcal{C}_n$ is determined from smaller objects, that is, from elements of $\{\mathcal{A}_j\}_{j=0}^n, \{\mathcal{B}_j\}_{j=0}^n$. The unlabelled constructions considered here are disjoint *union* $(+)$, cartesian *product* $(\times)$, and *sequence* formation $(\mathfrak{S})$. We define these in turn and concurrently build the corresponding Boltzmann sampler ΓC for the composite class $\mathcal{C}$, given random generators $\Gamma A, \Gamma B$ for the ingredients and assuming the values of intervening generating functions $A(x), B(x)$ at x to be real numbers which are *known exactly*.

Finite Sets. Clearly if $\mathcal{C}$ is finite (and in practice small), one can generate a random element of $\mathcal{C}$ by selecting it according to the finite probability distribution defined by the Boltzmann model: If $\mathcal{F} = \{\omega_1, \ldots, \omega_r\}$, then one selects f_j with probability proportional to $z^{|f_j|}$. Thus, drawing from a finite set is equivalent to a finite probabilistic switch. Drawing from a singleton set is then a deterministic procedure which directly outputs the object in question. In particular, in what follows, we make use of the singleton classes, $\mathbf{1}$ and $\mathcal{Z}$, formed respectively of one element of size 0 (analogous to the empty word of formal language theory) and of one element of size 1 that can be viewed as a generic "atom" out of which complex combinatorial structures are formed.

Disjoint union. Write $\mathcal{C} = \mathcal{A} + \mathcal{B}$ if $\mathcal{C}$ is the union of disjoint copies of $\mathcal{A}$ and $\mathcal{B}$, with size on $\mathcal{C}$ inherited from $\mathcal{A}, \mathcal{B}$. By disjointness, one has $C_n = A_n + B_n$, so that

$$C(z) = A(z) + B(z). \tag{7}$$

Consider a random element of $\mathcal{C}$ under the Boltzmann model of index x. Then, the probability that this random element is some $\alpha \in \mathcal{A}$ is

$$\mathbb{P}_{\mathcal{C},x}(\alpha) \equiv \frac{x^{|\alpha|}}{C(x)} = \frac{x^{|\alpha|}}{A(x)} \cdot \left(\frac{A(x)}{C(x)}\right).$$

The Boltzmann model corresponding to $C(x)$ is then a mixture of the models associated to $A(x)$ and $B(x)$, the probability of selecting a particular γ in $\mathcal{C}$ being

$$\mathbb{P}_{\mathcal{C},x}(\gamma \in \mathcal{A}) = \frac{A(x)}{C(x)}, \qquad \mathbb{P}_{\mathcal{C},x}(\gamma \in \mathcal{B}) = \frac{A(x)}{C(x)}.$$

Given a generator for a Bernoulli variable $\mathrm{Bern}(p)$ defined by

$$\mathrm{Bern}(p) = 1 \text{ with probability } p; \quad \mathrm{Bern}(p) = 0 \text{ with probability } 1-p,$$

two Boltzmann samplers $\Gamma A(x), \Gamma B(x)$, and the values of the OGFs $A(x), B(x)$, a Boltzmann sampler ΓC for class $\mathcal{C} = \mathcal{A} + \mathcal{B}$ is simply obtained by the procedure:

```
function ΓC(x : real);      {generates C = A + B}
let p_A := A(x)/(A(x) + B(x));
if Bern(p_A) then return(ΓA(x)) else return(ΓB(x)) fi; end.
```

We abbreviate this construction as

$$\left(\mathrm{Bern}\left(\frac{A(x)}{C(x)}\right) \longrightarrow \Gamma A(x) \mid \Gamma B(x) \right), \tag{8}$$

where $(X \longrightarrow f \mid g)$ is a shorthand notation for: "if the random variable X is 1, then execute f, else execute g". More generally, if X ranges over a finite set with r elements endowed with a probability measure, $p_1, \ldots, p_r$, we shall use the extended notation

$$(\mathrm{Bern}(p_1, \ldots, p_{r-1}) \longrightarrow f_1 \mid \ldots \mid f_r) \tag{9}$$

to represent the corresponding r-fold probabilistic switch.

Cartesian Product. Write $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ if $\mathcal{C}$ is the set of ordered pairs from $\mathcal{A}$ and $\mathcal{B}$, and size on $\mathcal{C}$ is inherited additively from $\mathcal{A}, \mathcal{B}$. Generating functions satisfy

$$C(z) = A(z) \cdot B(z) \quad \text{since} \quad C(z) = \sum_{\langle \alpha, \beta \rangle \in \mathcal{A} \times \mathcal{B}} z^{|\alpha|+|\beta|}. \tag{10}$$

A random element of $\gamma \in \mathcal{C}$ with $\gamma = (\alpha, \beta)$ then has probability

$$\mathbb{P}_{\mathcal{C},x}(\gamma) \equiv \frac{x^{|\gamma|}}{C(x)} = \frac{x^{|\alpha|}}{A(x)} \cdot \frac{x^{|\beta|}}{B(x)}.$$

It is thus obtained by forming a pair $\langle \alpha, \beta \rangle$ with α, β drawn *independently* from the Boltzmann models $\Gamma A(x), \Gamma B(x)$:

```
function ΓC(x : real);    {generates C = A × B}
return(⟨ΓA(x), ΓB(x)⟩) {independent calls}.
```

We shall abbreviate this schema as

$$\Gamma C(x) = \big(\Gamma A(x); \Gamma B(x)\big),$$

which can be read either as functionally producing a pair, or as sequential execution of the two procedures. We shall also use the natural extension $(f_1; \ldots; f_r)$ when r-tuples are involved.

Sequences. Write $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ if $\mathcal{C}$ is composed of all the finite sequences of elements of $\mathcal{A}$ (with size of a sequence additively inherited from sizes of components). The sequence class $\mathcal{C}$ is also the solution to the symbolic equation $\mathcal{C} = \mathbf{1} + \mathcal{A} \times \mathcal{C}$

(with **1** the empty sequence), which only involves unions and products and is reflected by the relation between OGFs: $C = 1 + AC$. Consequently,

$$C(z) = \frac{1}{1 - A(z)}. \tag{11}$$

This gives rise to two logically equivalent designs for a ΓC sampler:

(i) the recursive sampler,

```
function ΓC(x : real);   {generates C = 𝔖(A)}
if Bern(A(x)) then return(ΓA(x), ΓC(x))  {recursive call}
else return(1).
```

(ii) the geometric sampler,

```
function ΓC(x : real);   {generates C = 𝔖(A)}
draw k according to Geom(A(x));
return the k-tuple ⟨ΓA(x), ..., ΓA(x)⟩  {k independent calls}.
```

The recursive sampler for sequences is built from first principles (union and product rules). It might in principle loop for ever. However, by design, it repeatedly draws a Bernoulli random variable till the value 0 is attained. Thus, the number of components generated is a geometric random variable with rate $A(x)$, where, we recall, X is geometric of rate λ if

$$\mathbb{P}(X = k) = (1 - \lambda)\lambda^k.$$

For coherence to be satisfied, we must have $A(x) < 1$. Then, the recursive sampler halts with probability 1 since the expected number of recursive calls is finite and equal to $(1 - A(x))^{-1}$. This discussion justifies the geometric generator, which unwinds the recursion of the basic recursive sampler using a generator $\mathrm{Geom}(\lambda)$ for the geometric variable of parameter λ.

In what follows, we use the notation,

$$\big(Y \Longrightarrow f\big) \tag{12}$$

to mean: the random variable Y is drawn; if the value $Y = y$ is returned, then y *independent calls*, $f_1, \ldots, f_y$ are launched. The scheme giving the sequence sampler for $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ is then simply:

$$\Gamma C(x) = \left(\mathrm{Geom}(A(x)) \Longrightarrow \Gamma(x)\right).$$

Recursive classes. As suggested by the sequence construction, recursively defined classes admit generators that call themselves recursively. A specification by means of constructors is "well-founded" if it builds objects from smaller ones. An equivalent condition, when no recursion is involved, is that the sequence (and, for exponential Boltzmann models below, set, and cycle) operations are never applied to classes that contain objects of size 0. For recursive structures this is a testable property akin to "properness" in the theory of context-free grammars. (A context-free grammar is proper if the empty word is not generated with infinite multiplicity.) This well-foundedness condition also guarantees that the equations defining generating function equations are well-posed and contracting in the space of formal power series endowed with the standard metric, $\mathrm{dist}(f, g) = 2^{-\mathrm{val}(f-g)}$; accordingly, iteration provides a geometrically converging approximation scheme that makes it possible to determine generating function values for all coherent values of x (by analyticity and dominated convergence). See [27, 29] for a detailed discussion of this topic and the corresponding decision procedures.

Theorem 1. *Define as* specifiable *an unlabelled class that can be* finitely *specified (in a possibly recursive way) from finite sets by means of disjoint unions, cartesian products, and the sequence construction. Let C be an unlabelled specifiable class and x be a coherent parameter in $(0, \rho_C)$. Assume as given an oracle that provides the finite collection of* exact *values at a coherent value x of the generating functions intervening in a specification of a class $\mathcal{C}$. Then, the Boltzmann generator $\Gamma C(x)$ assembled from the definition of $\mathcal{C}$ by means of the four rules summarized in Figure 2 has a complexity measured in the number of $(+, -, \times, \div)$ real-arithmetic operations that is* linear *in the size of its output object.*

Proof. For a coherent value of size, the expectation of size is finite, so that, in particular, size is finite with probability 1. Given a specification Σ for $\mathcal{C}$, each object ω admits a unique parse tree (or syntax tree) $\tau[\omega]$ relative to Σ. For well-founded specifications, this parse tree τ is of a size linear in the size of the object produced. We shall see later (Lemma 1 in Section 5) that in the real-arithmetic model a Bernoulli choice can be effected with complexity $O(1)$ and a geometric random variable which assumes value k can be generated at cost $O(k+1)$. From this fact, the total cost of a Boltzmann sampler is of the form

$$O\left(\sum_{\nu \in \tau[\omega]} (\deg(\nu) + 1)\right),$$

where the summation ranges over all the nodes ν of tree τ, and $\deg(\nu)$ is the outdegree of node ν. Since, for any tree τ, one has $\sum_\nu 1 = |\tau|$ and $\sum_\nu \deg(\nu) = |\tau| - 1$, the total cost is linear in the size of τ, hence linear in the size of ω. The statement follows. ■

Given this proposition, one can *compile* automatically specifications of combinatorial classes into Boltzmann samplers. The only piece of auxiliary data required is a table of constants representing the values of the ordinary generating functions associated with the subclasses that intervene in a specification. These are in finite number and computable.

In the examples that follow, we enlarge the expressivity of the specification language by allowing constructions of the form

$$\mathfrak{S}_\Omega(\mathcal{A}) = \left\{ \langle \alpha_1, \ldots, \alpha_r \rangle \;\middle|\; \alpha_j \in \mathcal{A},\ r \in \Omega \right\}, \tag{13}$$

where $\Omega \subset \mathbb{N}$ is either a finite or a cofinite subset of the integers. If Ω is finite, this construction reduces to a disjunction of finitely many cases and the corresponding sampler is obtained by Bernoulli trials. If Ω is cofinite, we may assume without loss of generality that $\Omega = \{n \geq m_0\}$ for some $m_0 \in \mathbb{N}$, in which case, the construction $\mathfrak{S}_{\geq m_0}(\mathcal{A})$ reduces to $\mathcal{A}^{m_0} \times \mathfrak{S}(\mathcal{A})$.

EXAMPLE 1. *Words without long runs.* Consider the collection $\mathcal{R}$ of all binary words over the alphabet $\mathcal{A} = \{a, b\}$ that never have more than m consecutive occurrences of any letter (such consecutive sequences are also called "runs" and intervene at many places in statistics, coding theory, and genetics). Here we regard m as a fixed quantity. It is not *a priori* obvious how to generate a random word in $\mathcal{R}$ of length n: a brutal rejection method based on generating random unconstrained words and filtering out those that satisfy the condition $\mathcal{R}$ will not work in polynomial time since the constrained words have an exponentially small probability. On

the other hand, any word decomposes into a sequence of alternations also called its *core*, of the form

$$(aa\cdots a\,|\,bb\cdots b)\,(aa\cdots a\,|\,bb\cdots b)\,\cdots(aa\cdots a\,|\,bb\cdots b), \tag{14}$$

possibly prefixed with a header of b's and postfixed with a trailer of a's. In symbols, the set $\mathcal{W}$ of all words is expressible by a regular expression written in our notation

$$\mathcal{W} = \mathfrak{S}(b) \times \mathfrak{S}\,(a\mathfrak{S}(a)b\mathfrak{S}(b)) \times \mathfrak{S}(a).$$

The decomposition was customized to serve for $\mathcal{R}$: simply replace any internal $a\mathfrak{S}(a)$ by $\mathfrak{S}_{1\,..\,m}(a)$ and any $b\mathfrak{S}(b)$ by $\mathfrak{S}_{1\,..\,m}(b)$, where $\mathfrak{S}_{1\,..\,m}$ means a sequence of between 1 and m elements, and adapt accordingly the header and trailer:

$$\mathcal{R} = \mathfrak{S}_{\le m}(b) \times \mathfrak{S}\,(\mathfrak{S}_{1\,..\,m}(a)\mathfrak{S}_{1\,..\,m}(b)) \times \mathfrak{S}_{\le m}(a).$$

The composition rules given above give rise to a generator for $\mathcal{R}$ that has the following form: two generators that produce sequences of a's or b's according to a truncated geometric law; a generator for the product $\mathcal{C} := (\mathfrak{S}_{1\,..\,m}(a)\mathfrak{S}_{1\,..\,m}(b))$ that is built according to the product rule; a generator for the sequence $\mathcal{D} := \mathfrak{S}(\mathcal{C})$ constructed according to the sequence rule. The generator finally assembled *automatically* is:

$$\begin{array}{lcl}
\Gamma R(x) & = & (X \Longrightarrow b); \Gamma\mathrm{Core}(x); (X' \Longrightarrow a) \\
\Gamma\mathrm{Core}(x) & = & \Big(\mathrm{Geom}\,\big(\frac{x^2(1-x^m)^2}{(1-x)^2}\big) \Longrightarrow ((Y \Longrightarrow a); (Y' \Longrightarrow b))\Big) \\
 & & X, X' \in \underset{\le m}{\mathrm{Geom}}(x), \quad Y, Y' \in \underset{1\,..\,m}{\mathrm{Geom}}(x).
\end{array}$$

Observe that a table of only a small number of real-valued constants rationally related to x and including

$$c_1 = x, \quad c_2 = C(x) = x^2(1-x^m)^2(1-x)^{-2},$$

needs to be precomputed in order to implement the algorithm. □

Here are three runs of the sampler $\Gamma R(x)$ for $m = 4$ produced with the coherent value $x = 0.5$ (the critical value is $\rho_R \doteq 0.51879$), of respective lengths 124 (truncated), 23, and 35, with the coding $a =$□, $b =$■:

With this value of the parameter, the mean size of a random word produced is about 27. The distribution turns out to be of the "flat" type, like for Surjections in Figure 1. We shall see later in Section 7 that one can design optimized samplers for such types of distributions. The technique applies to any language composed of words with excluded patterns, meaning words that are constrained *not* to contain any of a finite set of words as factor. (For such a language, one can specifically construct a finite automaton by way of the Aho–Corasick construction [1], then write the automaton as a linear system of equations relating specifications, and finally compile the set of equations into a recursive Boltzmann sampler.) More generally, the method applies to any regular language: it suffices to convert a description of the language into a deterministic finite automaton and apply the recursive construction of a sampler, or alternatively to obtain an unambiguous regular expression and derive from it a nonrecursive sampler based on the geometric law.

The next set of examples is relative to structures that satisfy nonlinear recursive descriptions of the context-free type.

EXAMPLE 2. *Rooted plane trees.* Take the class $\mathcal{B}$ of binary trees defined by the recursive specification

$$\mathcal{B} = \mathcal{Z} + (\mathcal{Z} \times \mathcal{B} \times \mathcal{B}),$$

where $\mathcal{Z}$ is the class comprising the generic node. The generator ΓZ is deterministic and consists simply of the instruction "output a node" (since $\mathcal{Z}$ is finite and in fact has only one element). The Boltzmann generator ΓB calls ΓZ (and halts) with probability $x/B(x)$ where $B(x)$ is the OGF of binary trees,

$$B(x) = \frac{1 - \sqrt{1 - 4x^2}}{2x}.$$

With the complementary probability corresponding to the strict binary case, it will make a call to ΓZ and two recursive calls to itself. In shorthand notation, the recursive sampler is

$$\Gamma B(x) = \left(\mathrm{Bern}\left(\tfrac{x}{B(x)}\right) \longrightarrow \mathcal{Z} \;\middle|\; (\mathcal{Z}; \Gamma B(x); \Gamma B(x)) \right).$$

In other words: *the Boltzmann generator for binary trees as constructed automatically from the composition rules produces a random sample of the branching process with probabilities* $(\frac{x}{B(x)}, \frac{xB(x)^2}{B(x)})$. Note that the generator is defined for $x < 1/2$ (the radius of convergence of $B(x)$), in which case the branching process is subcritical, so that the algorithm halts in finite expected time, as it should. Only two constants are needed for implementation, namely x and the quadratic irrational $\frac{x}{B(x)}$.

Unbalanced 2-3 trees in which only external nodes contribute to size are similarly produced by $\mathcal{U} = \mathcal{Z} + \mathcal{U}^2 + \mathcal{U}^3$. Figure 3 displays such a tree for the value of the parameter x set at the critical value $\rho_U = \frac{5}{27}$. (This critical value can be determined by methods exposed in Section 7.) In this case, the branching probabilities for a nullary, binary, and ternary node are found to be respectively

$$p_0 = \frac{5}{9}, \qquad p_2 = \frac{1}{3}, \qquad p_3 = \frac{1}{9},$$

and these three constants are the only ones required by the algorithm. A typical run of 30 Boltzmann samplings produces trees with total number of nodes equal to

$$(15)\quad 3, 6, 1, 1, 6, 7, 33, 1, 1, 1, 9, 1, 1, 3, 1, 3, 169, 1881, 1, 54, 6, 1, 1, 3, 3746, 1, 1, 1, 1, 1,$$

which empirically gives an indication of the distribution of sizes (it turns out to be of the peaked type, like in Figure 1, bottom). We shall see later in Section 7 that one can actually characterize the profile of this distribution (it decays like $n^{-3/2}$) and put to good use some of its features.

Unary-binary trees (also known as Motzkin trees) are defined by $\mathcal{V} = \mathcal{Z}(\mathbf{1} + \mathcal{V} + \mathcal{V}^2)$. General plane trees, $\mathcal{G}$, where all degrees of nodes are allowed, can be specified by the grammar

$$\mathcal{G} = \mathcal{Z} \times \mathfrak{S}(\mathcal{G}),$$

with OGF $G(z) = (1 - \sqrt{1 - 4z})/2$. Accordingly, the automatically produced sampler is

$$\Gamma G(x) = \left(\mathcal{Z}; (\mathrm{Geom}(G(x)) \Longrightarrow \Gamma G(x))\right),$$

which corresponds to the well-known fact that such trees are equivalent to trees of a branching process where the offspring distribution is geometric. □

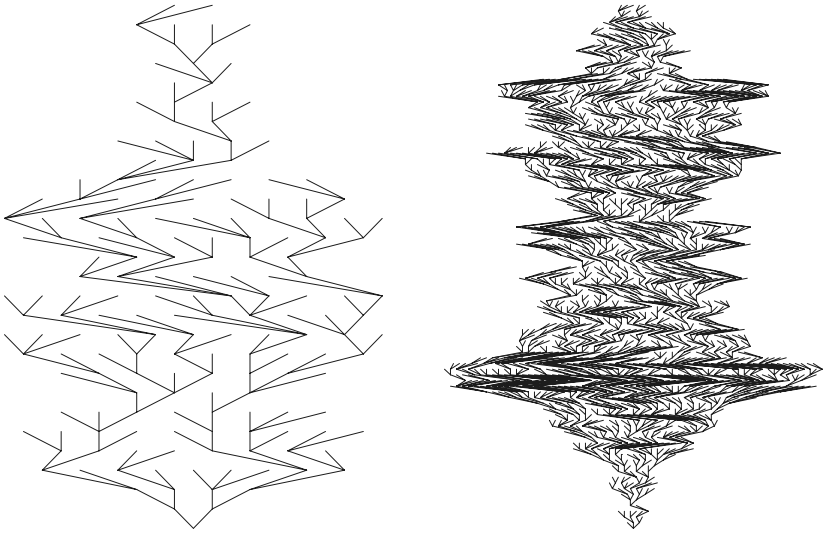


FIGURE 3. Random unbalanced 2–3 trees of 173 and 2522 nodes (in total) produced by a critical Boltzmann sampler.

EXAMPLE 3. *Secondary structures.* This example is inspired by works of Waterman *et al.*, themselves motivated by the problem of enumerating secondary RNA structures [36, 62]. To fix ideas, consider rooted binary trees where edges contain 2 or 3 atoms and leaves ("loops") contain 4 or 5 atoms. A specification is $\mathcal{W} = (\mathcal{Z}^4 + \mathcal{Z}^5) + (\mathcal{Z}^2 + \mathcal{Z}^3)^2 \times (\mathcal{W} \times \mathcal{W})$. A Bernoulli switch will decide whether to halt or not, two independent recursive calls being made in case it is decided to continue, with the algorithm being sugared with suitable Bernoulli draws. Here is the complete code:

$$\begin{aligned}
\Gamma A(x) &= \left(\mathrm{Bern}(\tfrac{x^4}{x^4+x^5}) \longrightarrow Z^4 \;\middle|\; Z^5\right)\\
\Gamma B(x) &= \left(\mathrm{Bern}(\tfrac{x^2}{x^2+x^3}) \longrightarrow Z^2 \;\middle|\; Z^3\right)\\
&\quad \text{let } p = (x^4+x^5)/W(x) = \tfrac{1}{2}(1+\sqrt{1-4x^8(1+x)^3});\\
\Gamma W(x) &= \left(\mathrm{Bern}(p) \longrightarrow \Gamma A(x) \;\middle|\; \Gamma B(x); \Gamma W(x); \Gamma B(x); \Gamma W(x)\right).
\end{aligned}$$

The method is clearly universal for this entire class of problems. □

EXAMPLE 4. *Noncrossing graphs.* Consider graphs which, for size n, have vertices at the nth roots of unity, $v_k = e^{2ik\pi/n}$, and are *connected* and *noncrossing* in the sense that no two edges are allowed to meet in the interior of the unit circle; see Figure 4 for a random instance. The generating function of such graphs has been first determined by Domb and Barret [15] motivated by the investigation of certain perturbative expansions of statistical physics. Their derivation is not based on methods conducive to Boltzmann sampling, though. On the other hand, the planar structure of such configurations entails a neat decomposition, which is described in [24]. At the top level, consider the graph as rooted at vertex v_0. Let v_i and v_j be two consecutive neighbours of v_0; the subgraph induced on the vertex set $\{v_i, v_{i+1}, \ldots, v_j\}$ is either a connected graph of $\mathcal{D}$ or is formed of two disjoint components containing v_i and v_j respectively. Also, if v_ℓ is the first neighbour of v_0 and

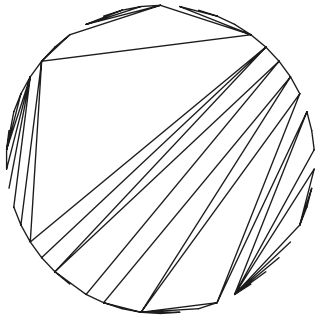


FIGURE 4. A random connected noncrossing graph of size 50.

v_m is the last neighbour, there are two connected components on $\{v_1, \ldots, v_\ell\}$ and on $\{v_m, \ldots, v_{n-1}\}$ respectively. The grammar for connected noncrossing graphs is then a transcription of this simple decomposition, although its detail is complicated as care must be exercised to avoid double counting of vertices. The class of all such connected noncrossing graphs is denoted by $\mathcal{X}$ and the grammar is:

$$\mathcal{X} = \mathcal{Z} + \mathcal{Z} \times \mathcal{E}, \quad \mathcal{E} = \mathcal{X} \times \mathfrak{S}(\mathcal{E} + \mathcal{X} \times (1 + \mathcal{E})) \times \mathcal{X}.$$

One finds that $E(z) = -1 + X(z)/z$ while $X(z)$ is a branch of the algebraic function defined implicitly by

$$X^3 + X^2 - 3zX + 2z^2 = 0,$$

and the critical value (the upper limit of all coherent values) is $\rho_X = \frac{1}{18}\sqrt{3} \doteq 0.09622$. The Boltzmann sampler compiled from the specification is then of the global form

$$\begin{array}{lcl} \Gamma X(x) & = & \left(\mathrm{Bern}\left(\frac{x}{X(x)}\right) \longrightarrow \mathcal{Z} \mid \mathcal{Z}; \Gamma E(x)\right) \\ \Gamma E(x) & = & (\Gamma X(x); (\mathrm{Geom}(E(x) + X(x)(1 + E(x))) \Longrightarrow ((\cdots))); \Gamma X(x)). \end{array}$$

The algorithm needs the parameter x, the cubic quantity $y = X(x)$ and a small number of quantities that are all rationally expressed in terms of x and y. For instance, the coherent choice $x = 0.095$ which is close to the critical value ρ_X, leads to $X(x) \doteq 0.11658$. There is then a probability of about $\frac{1}{7000}$ to attain a graph of size exactly 50; one such graph drawn uniformly at random is represented in Figure 4. □

In the last three cases (trees, secondary structures, and noncrossing graphs), the profile of the Boltzmann distribution resembles that of general trees in Figure 1. Optimized algorithms adapted to such tree-like profiles are discussed in Sections 6 and 7, where it is shown that random generation can be achieved in linear time provided a fixed nonzero tolerance on size is allowed. The method applies to any class that can be described unambiguously by a context-free grammar.

4. Exponential Boltzmann Generators

We consider here *labelled structures* in the precise technical sense of combinatorial theory [4, 28, 30, 34, 60, 61, 69]. A labelled object of size n is then composed of n distinguishable atoms, each bearing a distinctive label that is an integer in the interval $[1, n]$. For instance, the class $\mathcal{K}$ of labelled circular graphs, where cycles

Construction		*Generator*
Singleton	$\mathcal{C} = \{\omega\}$	$\Gamma C(x) = \omega$
Union	$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma C(x) = \left(\text{Bern}\left(\frac{\widehat{A}(x)}{\widehat{A}(x)+\widehat{B}(x)}\right) \longrightarrow \Gamma A(x) \mid \Gamma B(x)\right)$
Product	$\mathcal{C} = \mathcal{A} \star \mathcal{B}$	$\Gamma C(x) = \big(\Gamma A(x); \Gamma B(x)\big)$
Sequence	$\mathcal{C} = \mathfrak{S}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Geom}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big)$
Set	$\mathcal{C} = \mathfrak{P}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Pois}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big)$
Cycle	$\mathcal{C} = \mathfrak{C}(\mathcal{A})$	$\Gamma C(x) = \big(\text{Loga}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big)$

FIGURE 5. The inductive rules for exponential Boltzmann samplers.

are oriented in some conventional manner (say, positively) is

$$\mathcal{K} = \left\{ \textcircled{1}, \quad \begin{array}{c}\textcircled{1}\\ \textcircled{2}\end{array}, \quad \begin{array}{c}\textcircled{1}\\ \textcircled{2}\ \textcircled{3}\end{array}, \quad \begin{array}{c}\textcircled{1}\\ \textcircled{3}\ \textcircled{2}\end{array}, \ldots \right\}.$$

Clearly, there are $K_n = (n-1)!$ labelled objects of size $n \geq 1$, and the corresponding exponential generating function $\widehat{K}(z)$ has been determined in (3). In what follows, we focus on generating the "shape" of labelled objects—for instance, the shape of an n-cyclic graph would be a cycle with n anonymous dots placed on it. The reason for doing so is that labels can then always be obtained by superimposing a random permutation[4] on the unlabelled nodes. Note however, that the unlabelled (ordinary) and labelled (exponential) Boltzmann models assign rather different probabilities to objects: in the unlabelled case, there would be only $k_n \equiv 1$ object of size n, with OGF $k(x) = x/(1-x)$ so that the distribution of component sizes is geometric, while in the labelled case, the logarithmic series distribution (4) occurs.

Labelled combinatorial classes can be subjected to the *labelled product* defined as follows: if $\mathcal{A}$ and $\mathcal{B}$ are labelled classes, the product $\mathcal{C} = \mathcal{A} \star \mathcal{B}$ is obtained by forming all ordered pairs $\langle \alpha, \beta \rangle$ with $\alpha \in \mathcal{A}$ and $\beta \in \mathcal{B}$ and relabelling them in all possible order-consistent ways. Straight from the definition, one has a binomial convolution $C_n = \sum_{k=0}^{n} \binom{n}{k} A_k B_{n-k}$, where the binomial takes care of relabellings. In terms of exponential generating functions, this becomes

$$\widehat{C}(z) = \widehat{A}(z) \cdot \widehat{B}(z).$$

Like in the ordinary case, we proceed by assembling Boltzmann generators for structured objects from simpler ones.

Disjoint union. The unlabelled construction carries over verbatim to this case to the effect that, for labelled classes $\mathcal{A}, \mathcal{B}, \mathcal{C}$ satisfying $\mathcal{C} = \mathcal{A} + \mathcal{B}$, EGFs are related by $\widehat{C}(z) = \widehat{A}(z) + \widehat{B}(z)$ and the exponential Boltzmann sampler for C is

$$\Gamma C(x) = \left(\text{Bern}\left(\tfrac{\widehat{A}(x)}{\widehat{A}(x)+\widehat{B}(x)}\right) \longrightarrow \Gamma A(x) \ \mid \ \Gamma B(x)\right).$$

Labelled product. The cartesian product construction adapts to this case with minor modifications: to produce an element from $\mathcal{C} = \mathcal{A} \star \mathcal{B}$, simply produce a pair

[4]Drawing a random permutation of $[1, n]$ only necessitates $O(n)$ real operations [39, p. 145].

by the cartesian product rule using values $\widehat{A}(x), \widehat{B}(x)$:

$$\Gamma C(x) = (\Gamma A(x); \Gamma B(x))\,.$$

Complete by a randomly chosen relabelling if actual values of the labels are needed.

Sequences. In the labelled universe, $\mathcal{C}$ is the sequence class of $\mathcal{A}$, written $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ iff it is composed of all the sequences of elements from A up to order-consistent relabellings. Then, the EGF relation

$$\widehat{C}(x) = \sum_{k\geq 0} \widehat{A}(x)^k = \frac{1}{1-\widehat{A}(x)}$$

holds, and either of the two constructions of the generator ΓC from ΓA given in Section 3 is applicable. In particular, the nonrecursive generator is

$$\Gamma C(x) = \big(\,\mathrm{Geom}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big),$$

where the stenographic convention of (12) is employed.

Sets. This is a new construction that we did not consider in the unlabelled case. The class $\mathcal{C}$ is the set-class of $\mathcal{A}$, written $\mathcal{C} = \mathfrak{P}(\mathcal{A})$ ($\mathfrak{P}$ is reminiscent of "powerset") if $\mathcal{C}$ is the quotient of sequences, $\mathcal{C} = \mathfrak{S}(\mathcal{A})/\equiv$, by the relation $\equiv$ that declares two sequences as equivalent if one derives from the other by an arbitrary permutation of the components. It is then easily seen that the EGFs are related by

$$\widehat{C}(x) = \sum_{k\geq 0} \frac{1}{k!}\widehat{A}(x)^k = e^{\widehat{A}(x)},$$

where the factor $1/k!$ "kills" the order present in k–sequences.

The Poisson law of rate λ is classically defined by

$$\mathbb{P}(X=k) = e^{-\lambda}\frac{\lambda^k}{k!}.$$

On the other hand, under the exponential Boltzmann, the probability for a set in $\mathcal{C}$ to have k components in $\mathcal{A}$ is

$$\frac{1}{\widehat{C}(x)}\frac{1}{k!}\widehat{A}(x)^k = e^{-\widehat{A}(x)}\frac{\widehat{A}(x)^k}{k!},$$

that is, a Poisson law of rate $\widehat{A}(x)$. This gives rise to a simple algorithm for generating sets (analogous to the geometric algorithm for sequences):

$$\Gamma C(x) = \big(\,\mathrm{Pois}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big).$$

Cycles. This construction, written $\mathcal{C} = \mathfrak{C}(\mathcal{A})$, is defined like sets but with two sequences being identified if one is a cyclic shift of the other. The EGFs satisfy

$$\widehat{C}(x) = \sum_{k\geq 1} \frac{1}{k}\widehat{A}(x)^k = \log\frac{1}{1-\widehat{A}(x)},$$

where the factor $1/k$ "converts" k–sequences into k–cycles. The log-law of rate $\lambda < 1$, an "integral" of the geometric law also known as the logarithmic series distribution, is the law of a variable X such that

$$\mathbb{P}(X=k) = \frac{1}{\log(1-\lambda)^{-1}}\frac{\lambda^k}{k}.$$

(This is the same as in Equation (4); the distribution occurs in statistical ecology and economy and forms the subject of Chapter 7 of [38].) Then cycles under the exponential Boltzmann model can be drawn like in the case of sets upon replacing the Poisson law by the log-law:

$$\Gamma C(x) = \big(\operatorname{Loga}(\widehat{A}(x)) \Longrightarrow \Gamma A(x)\big).$$

These constructions are summarized in Figure 5.

For reasons identical to the ones that justify Theorem 1, one has:

Theorem 2. *Define as* specifiable *a labelled class that can be* finitely *specified (in a possibly recursive way) from finite sets by means of disjoint unions, cartesian products, as well as the sequence, set and cycle constructions. Let $\mathcal{C}$ be a labelled specifiable class and x be a coherent parameter in $(0, \rho_C)$. Assume as given an oracle that provides the finite collection of* exact *values at a coherent value x of the generating functions intervening in a specification of a class $\mathcal{C}$. Then, the Boltzmann generator $\Gamma C(x)$ assembled from the definition of $\mathcal{C}$ by means of the six rules of Figure 5 has a complexity measured in the number of $(+, -, \times, \div)$ real-arithmetic operations that is* linear *in the size of its output object.*

(We also allow constructions $\mathfrak{S}_\Omega, \mathfrak{P}_\Omega, \mathfrak{C}_\Omega$ as in (13); in this case, the random variable of geometric, Poisson, or logarithmic type should be conditioned to assume its values in the set Ω.)

Like in the unlabelled case, Boltzmann samplers can be compiled automatically from combinatorial specifications. There is here added expressivity in the language of specifications, thanks to the inclusion of the Set and Cycle constructions. In the examples that follow, we omit the hat-marker "$\widehat{f}$", whenever the exponential/labelled character of the model is clear from the context.

Example 5. *Set partitions.* A set partition of size n is a partition of the integer interval $[1, n]$ into a certain number of nonempty classes, also called blocks, the blocks being by definition unordered between themselves. Let $\mathfrak{P}_{\geq 1}$ represent the powerset construction where the number of components is constrained to be ≥ 1. (This modified construction is easily subjected to random generation by using a truncated Poisson variable K, where K is conditioned to be ≥ 1.) The labelled class of all set partitions is then definable as $\mathcal{S} = \mathfrak{P}(\mathfrak{P}_{\geq 1}(\mathcal{Z}))$, where $\mathcal{Z}$ consists of a single labelled atom, $\mathcal{Z} = \{1\}$. Observe that the EGF of $\mathcal{S}$ is the well-known generating function of the Bell numbers, $S(z) = e^{e^z - 1}$. By the composition rules, one gets a random generator as follows: *Choose the number K of blocks as Poisson($e^x - 1$). Draw K independent copies $X_1, X_2, \ldots, X_K$ from the Poisson law of rate x, but each conditioned to be at least 1.* In symbols:

$$\Gamma S(x) = \left(\operatorname{Pois}(e^x - 1) \Longrightarrow \left(\operatorname*{Pois}_{\geq 1}(x) \Longrightarrow \mathcal{Z}\right)\right).$$

What this generates is in reality the "shape" of a set partition (the number of blocks (K) and the block sizes (X_j)), with the "correct" distribution. To complete the task, it suffices to transport this structure on a random permutation of the integers between 1 and N, where $N = X_1 + \cdots + X_K$.

The process distinctly differs from the classical algorithm of Nijenhuis and Wilf [51] that requires tables of large integers. It is related to a continuous model devised by Vershik [67] that can be interpreted as generating random set partitions based on

$$S(x) = e^{x/1!} \cdot e^{x^2/2!} \cdot e^{x^3/3!} \cdots,$$

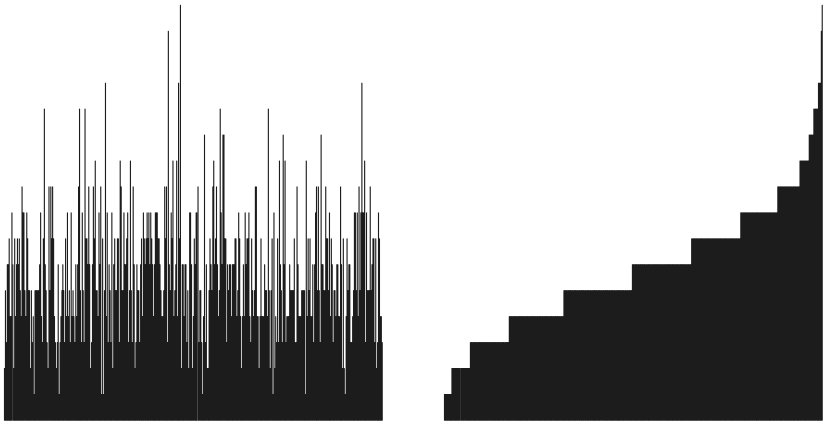


FIGURE 6. A random partition obtained by the Boltzmann parameter of parameter $x = 6$, here of size $n = 2356$ and comprised of 409 blocks: (left) the successive block sizes generated; (right) the block sizes in sorted order.

i.e., by ordered block lengths, as a potentially infinite sequence of Poisson variables of parameters $x/1!$, $x^2/2!$, and so on. □

Figure 6 represents a random set partition produced by the Boltzmann model of parameter $x = 6$. This particular object has size $n = 2356$, the expected size being $\mathbb{E}_x(N) = 2420$ for this value of the parameter. The closeness between the observed size and its mean value agrees with the concentration that is perceptible on Figure 1. In addition, the Boltzmann model immediately provides a simple heuristic model of partitions of large size. Objects of size "near" n, are produced by the value x_n defined by $x_n e^{x_n} = n$, that is, $x_n \approx \log n - \log\log n$. Then, the number of blocks is expected to be about $e^{x_n} \approx n/(\log n)$. This number being large, and individual blocks being generated by independent Poisson variables of parameter x_n, we expect, for large n, the sorted profile of blocks to converge to the histogram of the Poisson distribution of rate x_n (Figure 6, right). As shown by Vershik [67], this heuristic model is indeed a valid asymptotic model of partitions of large sizes.

EXAMPLE 6. *Random surjections (or ordered set partitions).* These may be defined as functions from $[1, n]$ to $[1, n]$ such that the image of f is an initial segment of $[1, n]$ (i.e., there are no "gaps"). One has for the class $\mathcal{Q}$ of surjections $\mathcal{Q} = \mathfrak{S}(\mathfrak{P}_{\geq 1}(\mathcal{Z}))$. Thus a random generator for $\mathcal{Q}$ is:

$$\Gamma Q(x) = \left(\mathrm{Geom}(e^x - 1) \Longrightarrow \left(\underset{\geq 1}{\mathrm{Pois}}(x) \Longrightarrow \mathcal{Z}\right)\right).$$

In words: first choose a number of components given by a geometric law and then launch a number of Poisson generators conditioned to be at most 1. □

Set partitions find themselves attached to a compound (Poisson∘Poisson) process, whereas surjections are generated by a compound (Geometric∘Poisson) process (with suitable dependencies on parameters). This reflects the basic combinatorial opposition between freedom and order (for blocks). Here are two more examples.

EXAMPLE 7. *Cycles in permutations.* This corresponds to $\mathcal{P} = \mathfrak{P}(\mathfrak{C}_{\geq 1}(\mathcal{Z}))$ and is obtained by a (Poisson∘Log) process:

$$\Gamma P(x) = \left(\mathrm{Pois}(\log(1-x)^{-1}) \Longrightarrow (\mathrm{Loga}(x) \Longrightarrow \mathcal{Z})\right).$$

This example is loosely related to the Shepp–Lloyd model [57] that generates permutations by ordered cycle lengths, as a potentially infinite sequence of Poisson variables of parameters $x/1$, $x^2/2$, and so on. The interest of this construction is to give rise to a number of useful particularizations. For instance derangements (permutations such that $\sigma(x) \neq x$) are produced by $\mathcal{P} = \mathfrak{P}(\mathfrak{C}_{\geq 2}(\mathcal{Z}))$; involutions (permutations such that $\sigma \circ \sigma(x) = x$) are given by $\mathcal{P} = \mathfrak{P}(\mathfrak{C}_{1\,..\,2}(\mathcal{Z}))$. □

EXAMPLE 8. *Assemblies of filaments.* Imagine assemblies of linear filaments floating freely in a liquid. We may model these as sets of sequences, $\mathcal{F} = \mathfrak{P}(\mathfrak{S}_{\geq 1}(\mathcal{Z}))$. The EGF is $\exp\left(\frac{z}{1-z}\right)$. The random generation algorithm is a compound of the form (Poisson∘Geometric), with appropriate parameters:

$$\Gamma F(x) = \left(\mathrm{Pois}\left(\tfrac{x}{1-x}\right) \Longrightarrow \left(\underset{\geq 1}{\mathrm{Geom}}(x) \Longrightarrow \mathcal{Z}\right)\right).$$

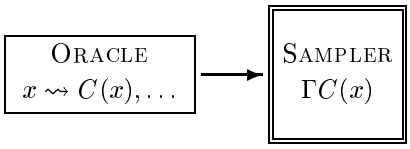
The corresponding counting sequence, $1, 1, 3, 13, 73, 501, \ldots$, appears as A000262 in Sloane's encyclopedia [58]. This example is closely related to linear forests and posets as described in Burris' book (see [6], pp. 23–24 and Ch. 4). □

At this stage, it may be of interest to note that many classical probabilistic distributions of probability theory can be retrieved as (size distributions of) Boltzmann models associated to simple combinatorial games. Consider an unbounded supply of distinguishable (i.e., labelled) balls. View an urn as an unordered finite collection of balls ($\mathfrak{P}(\mathcal{Z})$) and a stack as an ordered collection of balls ($\mathfrak{S}(\mathcal{Z})$). The geometric and Poisson distributions arise as the size distributions of the stack and the urn. If, by an exclusion principle, an urn is only allowed to contain 0 or 1 ball ($\mathbf{1} + \mathcal{Z}$), then the family of all basic Bernoulli distributions results. If m urns or stacks are considered, then the distributions are Poisson or negative binomial, respectively, and, with exclusion, one gets in this way the binomial distributions corresponding to m trials. If balls and urns are taken to be indistinguishable, one obtains automatically Vershik's model of integer partitions [67], which is an infinite product of geometric distributions of exponentially decaying rates. (The recent work by Milenković and Compton [50] discusses exact and asymptotic transforms associated to several such distributions.) For similar reasons, the two classical models of random graphs due to Erdős and Rényi are related to one another by "Boltzmannization". A large number of examples along similar lines could clearly be listed.

5. The realization of Boltzmann samplers

In this section, we make explicit the way Boltzmann sampling can be implemented and sketch a discussion of the main complexity issues involved. Broadly speaking, samplers can be realized under two types of computational models corresponding to computations carried out over the set $\mathbb{R}$ of real numbers or the set $\mathbb{S} = \{0,1\}^{\mathbb{N}}$ of infinite-length binary strings. (In the latter case, only finite prefixes are ever used.) These are the *real-arithmetic model*, $\mathbb{R}$, which is the one considered here and the *bit string model* (or Boolean model), $\mathbb{S}$, whose algorithms will be described in a future publication. The "ideal" real-domain model $\mathbb{R}$ comprises the elementary operations $+, -, \times, \div$ each taken to have *unit cost*.

By definition, a Boltzmann sampler requires as input the value of the control parameter x that defines the Boltzmann model of use. As seen in previous sections, it also needs the finite collection of values at x of the generating functions that intervene in a specification, in order to drive Bernoulli, geometric, Poisson, and logarithmic generators. We assume these values to be provided by what we call the (generating function) "*oracle*":



Such constants need only be precomputed *once*; they can be provided by a multi-precision package or a computer algebra system used as coroutine. We take here these constants as given, noting that the corresponding power series expansions at 0 are computable in low polynomial complexity (this is, e.g., encapsulated in the Maple package Combstruct; see [27, 29] for the underlying principles) so that values of the generating functions of constructible classes strictly inside their disc of convergence are computable real numbers of low polynomial time complexity.

There remains to specify fully generators for the probabilistic laws Geom(λ), Pois(λ), Loga(λ), as well as the Bernoulli generator Bern(p), where the latter outputs 1 with probability p and 0 otherwise. What is assumed here is a random generator 'uniform()' that produces at unit cost a random variable uniformly distributed over the real interval $(0, 1)$.

Bernoulli generator. The Bernoulli generator is simply

$$\text{Bern}(p) := \textsf{if } \text{uniform}() \le p \textsf{ then } \text{return}(1) \textsf{ else } \text{return}(0) \textsf{ fi}.$$

This generator serves in particular to draw from unions of classes.

Geometric, Poisson, and Logarithmic generators. For the remaining laws, we let p_k be the probability that a random variable with the desired distribution has value k, namely,

$$\text{Geom}(\lambda) : p_k = (1-\lambda)\lambda^k; \ \text{Pois}(\lambda) : p_k = e^{-\lambda}\frac{\lambda^k}{k!}; \ \text{Loga}(\lambda) : p_k = \frac{1}{\log(1-\lambda)^{-1}}\frac{\lambda^k}{k}.$$

The general scheme that goes well with real-arithmetic models is the *sequential algorithm*:

$U := \text{uniform}(); \ S := 0; \ k := 0;$
$\textsf{while } U < S \textsf{ do } S := S + p_k; \ k := k + 1; \textsf{ od};$
$\text{return}(k).$

This scheme is nothing but a straightforward implementation based on *inversion* of distribution functions (see [14, Sec. 2.1] or [39, Sec. 3.4.1]). For the three distributions under consideration, the probabilities p_k can themselves be computed recurrently on the fly as follows:

(16)

Geom(λ)	Pois(λ)	Loga(λ)
$p_0 = (1-\lambda)$	$p_0 = e^{-\lambda}$	$p_1 = 1/\left(\log(1-\lambda)^{-1}\right)$
$p_{k+1} = \lambda p_k$	$p_{k+1} = \lambda p_k \frac{1}{k+1}$	$p_{k+1} = \lambda p_k \frac{k}{k+1}$.

(Such principles also apply to constructions modified by a constraint on the number of components; e.g., to generate a $\text{Pois}_{\ge 1}(\lambda)$ random variable, initialize the generator with $p_1 = (e^{\lambda} - 1)^{-1}$ and $k = 1$.)

Observe that the transcendental values in (16) (like $e^{-\lambda}$) are in the present context already provided by the oracle. For instance, if one has to generate sets corresponding to $\mathcal{C} = \mathfrak{P}(A)$, then the generator for sets, $\mathrm{Pois}(A(x)) \Longrightarrow \Gamma A(x)$, requires the knowledge of $e^{-A(x)}$ which is none other than $1/C(x)$. Under the model that has unit cost for the four elementary real-arithmetic operations, the sequential generators thus have a useful property:

Lemma 1. *For either of the geometric, Poisson, or logarithmic generators, a random variable with outcome k is drawn with a number of real-arithmetic operations which is $O(k+1)$.*

This lemma completes te justification of Theorems 1 and 2.

In practice, one may realize approximately a Boltzmann sampler by truncating real numbers to some fixed precision, say using floating point numbers represented on 64 bits or 128 bits. The resulting samplers operate in time that is linear in the size of the object produced, though they may fail (by lack of digits in values of generating functions, i.e., by insufficient accuracy in parameter values) in a small number of cases, and accordingly must deviate (slightly) from uniformity. Pragmatically, such samplers are likely to suffice for many simulations.

A sensitivity analysis of truncated Boltzmann samplers would be feasible, though rather heavy to carry out. One could even correct perfectly the lack of uniformity by appealing to an adaptive precision strategy based on guaranteed multiprecision floating point arithmetic—-e.g., double the accuracy of computations when more digits are needed. In case of floating-point implementations of the recursive method, such ideas are discussed in Zimmermann's survey [71], and the reader may get a feeling of the type of analysis involved by referring to the works of Denise, Dutour, and Zimmermann [12, 13]. In a companion paper, we shall explore another route and describe purely discrete Boltzmann samplers which are solely based on binary coin flips in the style of Knuth and Yao's work [40] and have the additional feature of "automatically" detecting when accuracy is insufficient.

6. Exact-size and approximate-size sampling

Our primary objective in this article is the fast random generation of objects of some large size. In this section and the next one, we consider two types of constraints on size.

- *Exact-size* random sampling, where objects of $\mathcal{C}$ should be drawn *uniformly* at random from the subclass $\mathcal{C}_n$ of objects of size exactly n.
- *Approximate-size* random sampling, where objects should be drawn with a size in an interval of the form $I(n,\varepsilon) = [n(1-\varepsilon), n(1+\varepsilon)]$, for some quantity $\varepsilon \geq 0$ called the (relative) *tolerance*. In applications, one is likely to consider cases where ε is a small fixed number, like 0.05, corresponding to an uncertainty on sizes of $\pm 5\%$. Though size may fluctuate (within limits), sampling is still *unbiased* in the sense that two objects with the same size are drawn with equal likelihood.

The conditions of exact and approximate-size sampling are automatically satisfied if one filters the output a Boltzmann generator by retaining only the elements that obey the desired size constraint. (As a matter of fact, we have liberally made use of this feature in previous examples, e.g., when selecting the trees of Figure 3 to be large enough.) Such a filtering is simply achieved by a *rejection* technique.

The main question then becomes: "When and how can the rejection strategy be reasonably efficient?".

The major conclusion of this section is that in many cases, including all the examples seen so far, approximate-size sampling is achievable in linear time under the (exact) real-arithmetic model. In addition, the constants appear to be not too large if a "reasonable" tolerance on size is accepted. Precisely, we develop analyses and optimizations corresponding to the three common types of distributions exemplified in Figure 1.

— For size distributions that are "bumpy", the straight rejection strategy succeeds with high probability in one trial, hence the linear-time complexity of approximate-size Boltzmann sampling results (Section 6.1).
— For size distributions that are "flat", the straight rejection strategy succeeds in $O(1)$ trials on average, a fact that again ensures linear-time complexity when a nonzero tolerance on size is allowed (Section 6.2).
— For size distributions that are "peaked" (at the origin), the technique of *pointing* may be used to transform automatically specifications into equivalent ones of the flat type (Section 6.3).

6.1. **Size-control and rejection samplers.** The basic *rejection sampler* denoted by $\mu C(x; n, \varepsilon)$ uses a Boltzmann generator $\Gamma C(x)$ for the class $\mathcal{C}$ and is described as follows, for any x with $0 < x < \rho_C$, n a target size and $\varepsilon \geq 0$ a relative tolerance:

function $\mu C(x; n, \varepsilon)$;
{Returns an object of $\mathcal{C}$ in $I(n, \varepsilon) := [n(1-\varepsilon), n(1+\varepsilon)]$}
repeat $\gamma := \Gamma C(x)$ until $|\gamma| \in I(n, \varepsilon)$;
return(γ); end.

The rejection sampler μC depends on a parameter x that one may choose arbitrarily amongst all coherent values. It simply tries repeatedly until an object of satisfactory size is produced. The case $\varepsilon = 0$ then gives exact-size sampling.

The outcome of a basic Boltzmann sampler has a random size N whose distribution is described by Proposition 1. One has

$$\mathbb{E}_x(N) = \nu_1(x), \qquad \mathbb{E}_x(N^2) = \nu_2(x), \qquad \mathbb{E}_x(N^2) - \mathbb{E}_x(N)^2 = \sigma(x)^2,$$

where σ represents standard deviation, with

$$\nu_1(x) := x\frac{C'(x)}{C(x)}, \qquad \nu_2(x) := x^2\frac{C''(x)}{C(x)} + x\frac{C'(x)}{C(x)}, \qquad \sigma(x) = \sqrt{\nu_2(x) - \nu_1(x)}.$$

If x stays bounded away from the critical value ρ_C, then $\nu_1(x)$ remains bounded, so that the object drawn is likely to have a small size (on average and in probability). Thus, values of x approaching the critical value $\rho \equiv \rho_C$ have to be considered. Introduce the *mean value condition* as

$$\text{Mean Value Condition}: \quad \lim_{x \to \rho^-} \nu_1(x) = +\infty. \tag{17}$$

(This condition is satisfied in particular when $C(\rho^-) = +\infty$.) Then a "natural tuning" for the rejection sampler consists in adopting as control parameter x the value x_n that satisfies

$$x_n \quad \text{is the root in } (0, \rho) \text{ of } \quad n = x\frac{C'(x)}{C(x)}, \tag{18}$$

which is uniquely determined. One then has:

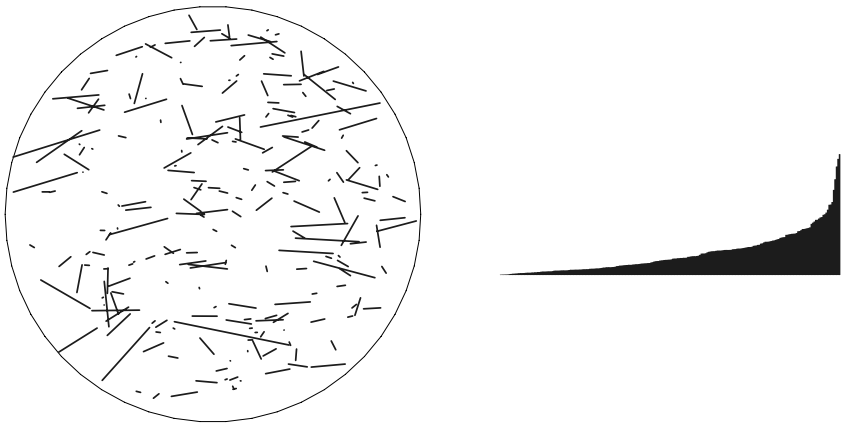


FIGURE 7. A random assembly of filaments of size $n = 46299$ produced by the exponential Boltzmann sampler tuned to $x_{50000} \doteq 0.9952$ (left) and its filaments presented in increasing order of lengths (right).

Theorem 3. *Let $\mathcal{C}$ be a combinatorial class and ε a* fixed *(relative) tolerance on size. Assume the Mean Value Condition* (17) *and the following variance condition*

$$\text{Variance Condition}: \quad \lim_{x\to\rho^-} \frac{\sigma(x)}{\nu_1(x)} = 0. \tag{19}$$

Then, the rejection sampler $\mu C(x_n; n, \varepsilon)$ equipped with the value $x = x_n$ implicitly determined by (18) *succeeds in one trial with probability tending to 1 as $n \to \infty$. In particular, if $\mathcal{C}$ is specifiable, then the overall cost of approximate-size sampling is $O(n)$ on average.*

Proof. This is a direct consequence of Chebyshev's inequalities. ∎

The mean and variance conditions *are* satisfied by the class $\mathcal{S}$ of set partitions (Example 5, observe concentration on Figure 1, top) and the class $\mathcal{F}$ of assemblies of filaments (Example 8 and Figure 7). In effect, for set partitions, $\mathcal{S}$, the exponential generating function is entire, which corresponds to $\rho = +\infty$. One finds

$$\nu_1(x) = xe^x, \qquad \sigma(x)^2 = x(x+1)e^x,$$

while x_n determined implicitly by the equation $x_n e^{x_n} = n$ satisfies $x_n \sim \log n - \log\log n$. These quantities are most easily interpreted when expressed in terms of n itself:

$$\nu_1(x_n) = n, \qquad \sigma(x_n) \sim \sqrt{n \log n}.$$

For assemblies of filaments, $\mathcal{F}$, one finds $\rho = 1$ and $\nu_1(x) = \frac{x}{(1-x)^2}$, so that x_n has value

$$x_n = 1 + \frac{1}{2n} - \frac{\sqrt{1+4n}}{2n} \sim 1 - \frac{1}{\sqrt{n}}.$$

and $\sigma(x_n) \sim \sqrt{2n}$. Here is, for various values of n, a table of the sizes of objects drawn in batches of 10 runs and the interval in which sizes are found to lie:

n	x_n	N (batch of 10 runs)	$N_{\min}$–$N_{\max}$
50	0.85857	$61, 80, 62, 13, 32, 65, 21, 34, 67, 16$	$13-80$
500	0.95527	$647, 426, 323, 752, 599, 457, 505, 318, 358, 424$	$318-752$
5,000	0.98585	$4575, 4311, 4419, 4257, 4035, 4067, 4187, 4984, 4543, 5035$	$4035-5035$

The fact that concentration of distribution improves with larger values of n is perceptible on such data. This feature in turn implies sampling in linear time, as soon as a positive tolerance on size is granted.

Exact-size sampling. The previous discussion suggests investigating conditions under which exact-size generation is still reasonably efficient. The smooth aspect of the "bumpy" curves associated with set partitions suggests the possibility that, in such cases, there exist a local limit distribution for sizes, as $x \to \rho$, implying an expected cost of $O(n\sigma(x_n))$ for exact-size sampling. It turns out that a sufficient set of *complex-analytic* conditions can be stated by borrowing results from the theory of *admissibility*, an area originally developed for the purpose of estimating asymptotically Taylor coefficients of entire functions. This theory was started in an important paper of Hayman [35] and is lucidly exposed in Odlyzko's survey [52, Sec. 12]. A function is said to be H-admissible if, in addition to the mean value condition (17) and the variance condition (19), it satisfies the following two properties:

- There exists a function $\delta(x)$ defined for $x < \rho$ with $0 < \delta(x) < \pi$ such that, for $|\theta| < \delta(x)$ as $x \to \rho^-$,
$$f(xe^{i\theta}) \sim f(x)e^{ia\theta - \frac{1}{2}b\theta^2}, \qquad a = \nu_1(x), \quad b = \sigma^2(x).$$
- Uniformly as $x \to \rho^-$, for $\delta(x) \le |\theta| \le \pi$,
$$f(xe^{i\theta}) = o\left(\frac{f(x)}{\sigma(x)}\right).$$

These conditions are the minimal ones that guarantee the applicability of the saddle-point method to Cauchy coefficient integrals. They imply in particular knowledge of the asymptotic form of the coefficients of f, namely,

$$f_n \equiv [z^n]f(z) \sim \frac{f(x_n)}{\sqrt{2\pi}x_n^n\sigma(x_n)}, \qquad n \to \infty.$$

We state:

Theorem 4. *Consider a class $\mathcal{C}$ whose generating function $f(z)$ satisfies the complex-analytic conditions of H-admissibility. Then exact size rejection sampling base on $\mu C(x_n; n, 0)$ succeeds in a mean number of trials that is asymptotic to*

$$\sqrt{2\pi}\sigma(x_n).$$

In particular, if $\mathcal{C}$ is specifiable, then the overall cost of exact-size sampling is $O(n\sigma(x_n))$ on average.

Proof. This is a direct adaptation of one of Hayman's estimates, see Theorem I of [35] (specialized in the notations of [35] as $r \to x_n$, $n \mapsto m$),

$$\frac{f_m x_n^m}{f(x_n)} \sim \frac{1}{\sqrt{2\pi}\sigma(x_n)} \exp\left(-\frac{(m-n)^2}{2\sigma(x_n)^2} + o(1)\right),$$

uniformly for all m as $x_n \to \rho$. This last equation means generally that the distribution of size values m is asymptotically normal as $x_n \to \rho^-$, that is, as $n \to \infty$. The specialization $m = n$ gives the statement. ∎

Hayman admissibility is easily checked to be satisfied by the EGFs of set partitions and assemblies of filaments. There results that exact size sampling has the following costs:

$$\text{Set partitions}: \quad O(n^{3/2}\sqrt{\log n}); \qquad \text{Assemblies}: \quad O(n^{3/2}).$$

Another result of Hayman states that, under H-admissibility, standard deviation is smaller than the mean, $\sigma(x_n) = o(n)$ (see Corollary I of [35]), so that exact-size generation by Boltzmann rejection is necessarily *subquadratic* ($o(n^2)$).

The usefulness of Hayman's conditions devolves from a rich set of closure properties: under mild restrictions, admissible functions are closed under sum ($f + g$), product (fg), and exponentiation (e^f). An informally stated consequence is then: *For classes whose generating function is "dominated" by an exponential, i.e., the "principal" construction is of the set type, approximate-size generation is of linear time complexity and exact-size generation is of subquadratic complexity.* Here are a few more examples.

- Statistical classification theory superimposes a tree structure on objects based on a similarity measure (e.g., the number of common phenotypes or genes). In this context, the value of a proposed classification tree may be assessed by comparing it to a random classification tree (structural properties should be substantially different in order for the classification to be likely to make sense). Such comparisons in turn benefit from random generation algorithms, a point originally made by Van Cutsem and collaborators [63, 64]. For instance, *hierarchies* are labelled objects determined by

  $$\mathcal{H} = \mathcal{Z} + \mathfrak{P}_{\geq 2}(\mathcal{H}),$$

  and they correspond to Schröder's systems of combinatorial theory [9, p. 223–224]. Hierarchies with a bounded depth of nesting are of interest in this context, and their EGFs

  $$e^z - 1, \quad z + e^{e^z - 1} - e^z, \quad e^{z + e^{e^z - 1} - e^z} - 1 - e^{e^z - 1} + e^z, \quad \dots,$$

  are all admissible, hence amenable to the conclusions of Theorem 4.
- Similar comments apply to labelled trees (Cayley trees, $\mathcal{T} = \mathcal{Z} \star \mathfrak{P}(\mathcal{T})$) of bounded height, with the sequence of EGFs starting as

  $$z, \quad ze^z, \quad ze^{ze^z}, \quad ze^{ze^{ze^z}}, \quad , \dots,$$

  and to "superpartitions" obtained by iterating the construction $\mathfrak{P}_{\geq 1}$:

  $$e^{e^z - 1} - 1, \quad e^{e^{e^z - 1} - 1} - 1, \quad e^{e^{e^{e^z - 1} - 1} - 1} - 1,$$

  where, e.g., the number sequence $(1, 3, 12, 60, 358, \dots)$ associated to the second case is A000258 of Sloane's *EIS* [58]. Related structures are of interest in finite model theory; see [68] for an introduction.
- Admissibility also covers generating functions of the type $e^{P(z)}$, with P a polynomial with nonnegative coefficients. This includes permutations with sizes of cycles constrained to be in a finite set Ω, for instance involutions

($\mathcal{I} = \mathfrak{P}(\mathfrak{C}_{1,2}(\mathcal{Z}))$, the solutions of $\sigma^d = Id$ in the symmetric group, and permutations whose longest cycle is at most some fixed value m.

The conditions of Theorem 3 are not satisfied by words without long runs (Example 1), surjections (Example 6, observe the lack of concentration on Figure 1, middle), and permutations (Example 7), although they fail by little, since the mean and standard deviation, $\nu_1(x)$ and $\sigma(x)$, happen to be of the same order of magnitude. They fail more dramatically for binary trees (Example 2 and Figure 1, bottom), secondary structures (Example 3), and noncrossing graphs (Example 4), where the ratio $\sigma(x)/\nu_1(x)$ now tends to infinity, in which case sizes produced by Boltzmann models exhibit a high dispersion. As discussed in the next two subsections and in Section 7, such situations can however be dealt with.

6.2. **Singularity types and rejection samplers.** It is possible to discuss at a fair level of generality cases where rejection sampling is efficient. The discussion is fundamentally based on the types of singularities that the generating functions exhibit. This is an otherwise well-researched topic as it is central to asymptotic enumeration [26, 28, 52].

Definition 2. *A function $f(z)$ analytic at 0 and a with finite radius of analyticity $\rho > 0$ is said to be* Δ–singular *if it satisfies the two conditions:*

(i) The function admits ρ as its only singularity on $|z| = \rho$ and it is continuable in a domain

$$\Delta(r, \theta) = \{z \mid z \neq \rho,\ |z| < r,\ \arg(z - \rho) \notin (-\theta, \theta)\},$$

for some $r > \rho$ and some θ satisfying $0 < \theta < \frac{\pi}{2}$.

(ii) For z tending to ρ in the Δ domain, $f(z)$ satisfies a singular expansion of the form

$$f(z) \underset{z\to\rho}{\sim} P(z) + c_0(1 - z/\rho)^{-\alpha} + o((1 - z/\rho)^{-\alpha}), \qquad \alpha \in \mathbb{R} \setminus \{0, -1, -2, \ldots\},$$

where $P(z)$ is a polynomial. The quantity $-\alpha$ is called the singular exponent *of $f(z)$.*

For reasons argued in [27], all the generation functions associated with specifiable models in the sense of this article are either entire or, else, they have dominant singularities which are isolated, hence they satisfy continuation conditions similar to (i). Condition (ii) is also granted in a large number of cases. Here, words without long runs, surjections, and permutations (Examples 1, 6, and 7) have generating functions with a polar singularity, corresponding to the singular exponent -1. Trees, secondary structures, and noncrossing graphs (Example 2, 3, and 4), which are recursively defined have singular exponent $\frac{1}{2}$; see [24, 49] and Section 8 below. Many properties go along with the conditions of Definition 2. Most notably, the counting sequence associated with a generating function $f(z)$ that is Δ-singular systematically obeys an asymptotic law:

$$[z^n]f(z) \sim \frac{c_0}{\Gamma(\alpha)}\rho^{-n}n^{\alpha-1}, \qquad (n \to \infty). \tag{20}$$

(This results from the singularity analysis theory exposed in [26, 28, 52].)

Returning to random generation, one has:

Theorem 5. *Let $\mathcal{C}$ be a combinatorial class such that its generating function is Δ-singular with an exponent $-\alpha < 0$. Then the rejection sampler $\mu C(x_n; n, \varepsilon)$ corresponding to a fixed tolerance $\varepsilon > 0$ succeeds in a number of trials whose expected value is asymptotic to the constant*

$$\frac{1}{\xi_\alpha(\varepsilon)}, \quad \textit{where} \quad \xi_\alpha(\varepsilon) = \frac{\alpha^\alpha}{\Gamma(\alpha)} \int_{-\varepsilon}^{\varepsilon} (1+s)^{\alpha-1} e^{-\alpha(1+s)}\, ds.$$

If $\mathcal{C}$ is specifiable, approximate-size Boltzmann sampling based on $\mu C(x_n; n, \varepsilon)$ has cost that is $O(n)$; exact-size sampling has cost $O(n^2)$.

Here is a table of numerical values of the expected number of trials ($1/\xi_{\alpha(\varepsilon)}$) for various values of the singular exponent $-\alpha$ and tolerance ε:

(21)

	$\varepsilon = 0.2$	$\varepsilon = 0.1$	$\varepsilon = 0.05$	$\varepsilon = 0.01$
$-\alpha = -2$	4.619	9.236	18.47	92.36
$-\alpha = -\frac{3}{2}$	5.387	10.80	21.61	108.0
$-\alpha = -1$	6.750	13.56	27.17	135.9
$-\alpha = -\frac{1}{2}$	9.236	20.61	41.30	206.6

For instance a tolerance of $\pm 10\%$ is likely to necessitate about 10 trials when $-\alpha$ is -2 or $-\frac{3}{2}$, while this number doubles for the singular exponent $-\frac{1}{2}$.

Proof. The rejection sampler used with the value x has a probability of success in one trial equal to

$$\mathbb{P}_x(|N/n - 1| \le \varepsilon),$$

which is to be estimated. The inverse of this quantity gives the expected number of trials.

Functions that are Δ-singular are closed under differentiation, since, by elementary complex analysis, asymptotic expansions valid in sectors can be subjected to differentiation [54, p. 9]. Consequently, one has

$$\nu_1(x) \underset{x \to \rho^-}{\sim} \frac{\alpha x/\rho}{1 - x/\rho} \to \infty,$$

which verifies the mean value condition, whereas a similar calculation shows $\sigma(x)$ to be of the same order as $\nu_1(x)$ and the variance condition is not satisfied. The strong form of coefficient estimates in (20) then entails

$$\mathbb{P}_x(N = m) \sim \frac{1}{\Gamma(\alpha)} \frac{m^{\alpha-1} |x/\rho|^m}{(1 - x/\rho)^{-\alpha}}, \tag{22}$$

for $x \to \rho^-$ and $m \to \infty$.

Tune now the rejection sampler at the value $x = x_n$, so that $\nu_1(x_n) = n$. One has

$$x_n \sim \rho \left(1 - \frac{\alpha}{n}\right).$$

Then, setting $m = t\nu_1(x_n) = tn$ transforms the estimate (22) into

$$\begin{aligned} \mathbb{P}_x(N = \lfloor tn \rfloor) &\sim \frac{1}{\Gamma(\alpha)} \frac{t^{\alpha-1} e^{tn \log(1-(\alpha/n))}}{\alpha^{-\alpha} n} \\ &\sim \frac{1}{n\Gamma(\alpha)} \alpha^\alpha t^{\alpha-1} e^{-\alpha t}, \end{aligned} \tag{23}$$

uniformly for t in a compact subinterval of $(0, \infty)$. This is exactly a *local limit law* for Boltzmann sizes in the form of a Gamma distribution [21, p. 47].

Cumulating the estimates in the formula above, one finds (by Euler-Maclaurin summation),

$$\mathbb{P}_{x_n}(|N/n-1|\le\varepsilon) \;\sim\; \frac{\alpha^\alpha}{\Gamma(\alpha)}\int_{-\varepsilon}^{\varepsilon}(1+s)^{\alpha-1}e^{-\alpha(1+s)}ds \tag{24}$$

which gives the value $\xi_\alpha(\varepsilon)$ of the statement. Linearity for the cumulated size then follows from the moderate dispersion of sizes induced by the relation $\sigma(x) = \Theta(\nu_1(x))$.

The argument adapts when ε is allowed to tend to 0. In this case, as seen directly from (23), the success probability of a single trial is asymptotic to

$$2\frac{(\alpha e)^\alpha}{\Gamma(\alpha)}\varepsilon,$$

with the inverse of this quantity giving the mean number of trials. In particular, if the target size lies in a fixed-width window around n ($\varepsilon = O(1/n)$), which covers exact-size random sampling, then a random generation necessitates $O(n)$ trials, corresponding to an overall complexity that is $O(n^2)$ under the real-arithmetic model. ■

Given the polar singularity involved, Theorem 5 applies directly to words without long runs (Ex. 1), surjections (Ex. 6), and cycles-in-permutations (Ex. 7).

EXAMPLE 9. *Mappings with degree constraints.* By a mapping of size n is meant here a function from $[1,n]$ into $[1,n]$. (Obviously, there are n^n of these.) We fix a finite set Ω and restrict attention to degree-constrained mappings f such that for each x in the domain, the cardinality of $f^{(-1)}(x)$ lies in Ω. (In the combinatorics literature, such mappings are surveyed in [2, 25].) For instance, in a finite field, a non-zero element has either 0 or 2 predecessors under the mapping $f; x \mapsto x^2$, so that (neglecting one exceptional value) a quadratic function may be regarded as an element of the set of mappings constrained by $\Omega = \{0, 2\}$. Mappings are of interest in computational number theory as well as in cryptography [55], and the eighth Fermat number, $F_8 = 2^{2^8}+1$ was first factored by Brent and Pollard [5] in 1981 by means of an algorithm that precisely exploits statistical properties of degree-constrained mappings.

As is well known, a mapping can be represented as a directed graph (Figure 8) where each vertex has outdegree equal to 1, while, by the degree constraint, indegrees must lie in Ω. Then the graph of a mapping is made of components, where each component is made of a unique cycle on which trees are grafted (see, e.g., [4] for this classical construction). With $\mathfrak{P}_\Omega$ representing the set construction with a number of elements constrained to lie in Ω, the class $\mathcal{M}$ of Ω–constrained mappings is

$$\mathcal{M} = \mathfrak{P}(\mathfrak{C}(\mathcal{U})), \quad \mathcal{U} = \mathcal{Z} \star \mathfrak{P}_{\Omega-1}(\mathcal{T}), \quad \mathcal{T} = \mathcal{Z} \star \mathfrak{P}_\Omega(T).$$

There $\mathcal{T}$ is the class of rooted labelled trees with outdegrees in Ω, $\mathcal{U}$ is the class of trees grafted on a cycle, which are such that their root degree must lie in $\Omega - 1$.

Let $\phi(y) := \sum_{\omega\in\Omega} y^\omega/\omega!$. The EGF of trees, T, is implicitly defined by $T = z\phi(T)$ and one has $U = z\phi'(T)$. It has been first established by Meir and Moon [49] that the EGF $T(z)$ has systematically a singularity of the square-root type (corresponding to "failure" in the implicit function theorem, see also Lemma 3 below). Precisely, one has $T(z) \sim \tau - c\sqrt{1-z/\rho}$ as $z \to \rho$, where $\rho \equiv \rho_T$ is given by

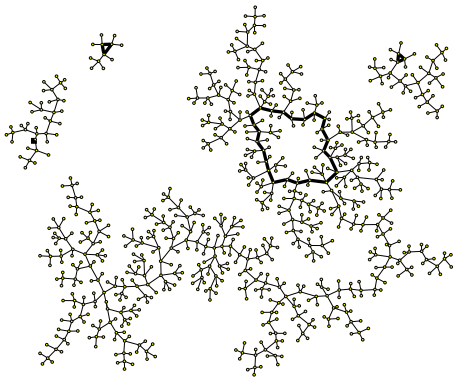


FIGURE 8. A random ternary map ($\Omega = \{0, 3\}$) of size 846 produced by Boltzmann sampling.

$\rho = \tau/\phi(\tau)$ and τ is the positive root of $\phi(\tau) - \tau\phi'(\tau) = 0$. There results that the EGF of constrained mappings satisfies as $z \to \rho$,

$$M(z) \sim \frac{1}{1 - \rho\phi'(\tau - c\sqrt{1 - z/\rho})} \sim \frac{d}{\sqrt{1 - z/\rho}},$$

for some $d > 0$. In view of this last expansion, Theorem 5 directly applies. Approximate-size random generation of Ω-constrained mappings is thus achievable in linear time. □

6.3. **The pointing operator.** In this section we further enlarge the types of structures amenable to fast Boltzmann sampling. As a byproduct, we are able to lift the restriction $-\alpha < 0$ in Theorem 5, thus bringing in its scope trees, secondary structures, and noncrossing graphs (Examples 2, 3, and 4) whose singularity is known [24, 49] to be of the square–root type, i.e., $\alpha = \frac{1}{2}$.

Given a combinatorial class $\mathcal{C}$, we define the class

$$\mathcal{C}^\bullet \;=\; \{(\gamma, i) \mid \gamma \in \mathcal{C},\, i \in \{1, \ldots, |\gamma|\}\}, \quad \text{equivalently,} \quad \mathcal{C}_n^\bullet \simeq \mathcal{C}_n \times \{1, \ldots, n\},$$

of *pointed* objects. Pointing is for instance studied systematically in [4, Sec. 2.1]. Objects in $\mathcal{C}^\bullet$ may be viewed as standard objects of $\mathcal{C}$ with one of the atoms distinguished by the mark "$\bullet$". From the definition, one has $|\mathcal{C}_n^\bullet| = n|\mathcal{C}_n|$, and the GF of the class $\mathcal{C}^\bullet$ is

$$C^\bullet(z) \;=\; z\frac{d}{dz}C(z),$$

regardless of the type of $C(x)$ (ordinary or exponential). Pointing can then be viewed as a combinatorial lifting of the usual operation of taking derivatives in elementary calculus. Since any non-pointed object of $\mathcal{C}$ gives rise to exactly n pointed objects, random sampling can be equally well be performed on $\mathcal{C}_n$ or $\mathcal{C}_n^\bullet$: it suffices to "forget" the pointer in an object produced by a sampler of $C_n^\bullet$ to obtain an object of $\mathcal{C}_n$. (Only the distributions of sizes under ΓC and $\Gamma C^\bullet$ are different.)

The pointing operator $\bullet$ is related to an operator studied systematically by Greene [32] (his "box" operation) and it plays a central rôle in the recursive method

(where it has been used under the name of "Theta operator"). For Boltzmann sampling, pointing can be used in conjunction with the previously defined operators $+, \times$ and $\star, \mathfrak{S}, \mathfrak{P}, \mathfrak{C}$ in either the labelled or unlabelled universe.

Lemma 2. *Let $\mathcal{C}$ be a specifiable unlabelled or labelled class (in the sense of Theorem 1 or 2). Then the class $\mathcal{C}^\bullet$ is also specifiable, i.e., it admits a specification without the pointing operator $\bullet$.*

Proof. First, for a finite class $\mathcal{C}$, the class $\mathcal{C}^\bullet$ is also finite and can be represented (and sampled) explicitly. Next, the pointing operator admits composition rules with all the other operators; in the labelled case, one has

$$(25)\quad \left\{ \begin{array}{lcllcl} (\mathcal{A}+\mathcal{B})^\bullet &=& \mathcal{A}^\bullet + \mathcal{B}^\bullet, & (\mathcal{A}\star\mathcal{B})^\bullet &=& \mathcal{A}^\bullet\star\mathcal{B} + \mathcal{A}\star\mathcal{B}^\bullet, \\ (\mathfrak{S}\mathcal{A})^\bullet &=& \mathfrak{S}\mathcal{A}\star\mathcal{A}^\bullet\star\mathfrak{S}\mathcal{A}, & (\mathfrak{C}\mathcal{A})^\bullet &=& \mathcal{A}^\bullet\star\mathfrak{S}\mathcal{A}, \\ (\mathfrak{P}\mathcal{A})^\bullet &=& \mathcal{A}^\bullet\star\mathfrak{P}\mathcal{A}. & & & \end{array} \right.$$

In the unlabelled case, the first three rules apply, upon changing the labelled product "$\star$" into the cartesian product "$\times$". These rules are a combinatorial analogue of the usual differentiation rules, and have a simple interpretation: e.g., pointing at a sequence ($(\mathfrak{S}\mathcal{A})^\bullet$) implies pointing at a component ($\mathcal{A}^\bullet$), which breaks the chain and individuates a left ($\mathfrak{S}\mathcal{A}$) and a right ($\mathfrak{S}\mathcal{A}$) subsequence.

Consider now a specification of the class $\mathcal{C} = \mathcal{F}_1$ in the form of a system,

$$\mathcal{S} \;=\; \{\mathcal{F}_i \;=\; \Phi_i(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m),\; i = 1, \ldots, m\},$$

where $\mathcal{F}_i$ are auxiliary classes and the Φ_i are functional terms involving finite classes and the standard operators (without pointing). Then, one can build a specification of the class $\mathcal{C}^\bullet$ in the form of a derived system,

$$\mathcal{S}' = \mathcal{S} \;\cup\; \{\mathcal{F}_i^\bullet \;=\; \Psi_i(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m, \mathcal{F}_1^\bullet, \ldots, \mathcal{F}_m^\bullet),\; i = 1, \ldots, m\},$$

where the functionals Ψ_i do not involve the pointing operator "$\bullet$": Ψ_i is obtained from $\Phi_i^\bullet$ by application of the derivation rules until the pointing operator is applied to variables only. In the derived specification, each $\mathcal{F}_i^\bullet$ is treated as a new variable, thereby leading to a complete elimination of the pointing operator within constructions. ■

Our interest for pointing lies in the following two observations.

- If a class $\mathcal{C}$ has a generating function $C(z)$ that is Δ-analytic with exponent $-\alpha$, then the generating function $zC'(z)$ of the class $\mathcal{C}^\bullet$ is also Δ-analytic and has an exponent $-\alpha - 1$, which is smaller.
- Uniform sampling in $\mathcal{C}_n$ is equivalent to uniform sampling in $\mathcal{C}_n^\bullet$. As a consequence, the sampler $\mu\mathcal{C}^\bullet(x; n, \varepsilon)$ is a correct approximate-size sampler for the class $\mathcal{C}$ (upon removing the mark).

Let $\mu\mathcal{C}^{\bullet k}(x; n, \varepsilon)$ denote the rejection sampler of the class derived from $\mathcal{C}$ by k successive applications of the pointing operator. The last two observations immediately lead to an extension of Theorem 5:

Theorem 6. *Let $\mathcal{C}$ be a combinatorial class such that its generating function is Δ-singular with* any *exponent $-\alpha \neq \{0, 1, \ldots\}$. Let $\alpha_+ = \max(0, \lceil -\alpha \rceil)$ the integral positive part of $-\alpha$, and $\alpha_0 = \alpha + \alpha_+$ its fractional part. Then the rejection sampler $\mu\mathcal{C}^{\bullet\alpha_+}(x_n; n, \varepsilon)$ corresponding to a fixed tolerance $\varepsilon > 0$ succeeds in a number of trials whose mean is asymptotic to the constant $\frac{1}{\xi_{\alpha_0}(\varepsilon)}$. In particular, if $\mathcal{C}$ is specifiable, the total cost of the rejection sampler $\mu\mathcal{C}^{\bullet\alpha_+}(x_n; n, \varepsilon)$ is $O(n)$ on average.*

As an illustration of Theorem 6, we examine the internal workings of the algorithm that results for the class $\mathcal{B}$ of binary trees taken here for convenience as

$$\mathcal{B} = \mathcal{Z} + (\mathcal{B} \times \mathcal{B}),$$

so that only external nodes contribute to size. The pointed class satisfies

$$\mathcal{B}^\bullet = \mathcal{Z}^\bullet + (\mathcal{B}^\bullet \times \mathcal{B}) + (\mathcal{B} \times \mathcal{B}^\bullet),$$

which completely defines it in terms of $\mathcal{B}$ and itself. Accordingly, the Boltzmann samplers for $\mathcal{B}$ and $\mathcal{B}^\bullet$ are defined by the system of simultaneous equations

$$\begin{cases} \Gamma B(x) & = & \big(\mathrm{Bern}\,(p_0) \longrightarrow \mathcal{Z} \;\big|\; (\Gamma B(x); \Gamma B(x))\big) \\ \Gamma B^\bullet(x) & = & \big(\mathrm{Bern}\,(p_1, p_2) \longrightarrow \mathcal{Z}^\bullet \;\big|\; (\Gamma B^\bullet(x); \Gamma B(x)) \;\big|\; (\Gamma B(x); \Gamma B^\bullet(x))\big) \end{cases}$$

where

$$p_0 = \frac{2x}{1-\sqrt{1-4x}}, \quad p_1 = \sqrt{1-4x}, \quad p_2 = \frac{1}{2} - \frac{1}{2}\sqrt{1-4x},$$

and the notation (9) for probabilistic switches is employed.

Random generation of a tree of size near n is achieved by a call to $\Gamma B^\bullet(x_n)$. For large n, the quantity x_n is very close to the critical value $\rho = \frac{1}{4}$. Thus, $\Gamma B^\bullet$ generates a terminal node with a small probability (since $p_1 \approx 0$), and, with high probability, $\Gamma B^\bullet(x_n)$ triggers a long sequence of calls to ΓB, which itself produces each time a near-critical tree (since $p_0 \approx \frac{1}{2}$) . In particular, the "danger" of generating small trees is automatically controlled by $\Gamma B^\bullet$. Observe that a sampler formally equivalent to $\Gamma B^\bullet(x)$ (by recursion removal) is then as follows: generate a long random branch (with randomly chosen $(\frac{1}{2}, \frac{1}{2})$ left or right branchings) and attach to it a collection of (near) critical trees[5]. For instance, here are the sizes observed in runs of 20 calls, one relative to ΓB equipped with the value $x_{500} = 0.2499997495$, the other to $\Gamma B^\bullet$ equipped with $x'_{500} = 0.2497497497$:

$$2, 1, 4, 5, 4, 1, 1, 1, 1, 1, 1, 1, 56, 1, 1, 7, 2, 1, 2, 2$$
$$831, 6, 76, 120, 1, 532, 15, 7, 11, 68, 99, 45, 1176, 12, 94, 81, 784, 3393, 21, 493.$$

(See also (15) for more extensive data that are similar to the first line.) While the parameters are chosen in each case such that the resulting object has expected size $n = 500$, it is clear that the $\Gamma B^\bullet$ sampler gets a better shot at the target.

Pointing also constitutes a valuable optimization whenever structures are driven by a cycle construction. Define a function f to be logarithmic if it is continuable in a Δ–domain and satisfies

$$f(z) = c \log \frac{1}{1 - z/\rho} + O(1), \qquad z \to \rho.$$

This may somehow be regarded as the limit case $\alpha \to 0$ of a singular exponent $-\alpha$. As the table (21) suggests, the efficiency of rejection deteriorates in this case: singularity analysis may be used to verify that $\sigma(x_n) = n\sqrt{\log n}$, so that approximate-size is of superlinear complexity, namely $O(n\sqrt{\log n})$. This problem is readily fixed by pointing. If $\mathcal{C} = \mathfrak{C}(\mathcal{A})$, then the transformation rules of (25) imply that we can alternatively generate a sequence, which is amenable to straight rejection sampling in linear time, since its generating function now has a polar-like singularity (with exponent $-\alpha = -1$). For instance, the class $\mathcal{K}$ of connected mappings is defined by

$$\{\mathcal{K} = \mathfrak{C}(\mathcal{T}), \quad \mathcal{T} = \mathcal{Z} \star \mathfrak{P}(\mathcal{T})\}.$$

[5]This construction is akin to the "size-biased" Galton–Watson process exposed in [47]. It is interesting to note that we are here led naturally to it by a systematic use of formal transformations.

The derived specification for $\mathcal{K}^\bullet$ is then

$$\{\mathcal{K}^\bullet = \mathcal{T}^\bullet \star \mathfrak{S}(\mathcal{T}),\ \mathcal{T} = \mathcal{Z} \star \mathfrak{P}(\mathcal{T}),\ \mathcal{T}^\bullet = \mathcal{Z}^\bullet \star \mathfrak{P}(\mathcal{T}) + \mathcal{Z} \star \mathfrak{P}(\mathcal{T}) \star \mathcal{T}^\bullet\},$$

with nonterminals $\mathcal{K}^\bullet, \mathcal{T}, \mathcal{T}^\bullet$. The generator $\Gamma\mathcal{K}^\bullet$ then achieves linear time sampling for any fixed tolerance $\varepsilon > 0$. (Figure 8 has been similarly produced by pointing.)

This technique applies to plane trees and variants thereof (Example 2), secondary structures (Example 3), and noncrossing graphs (Example 4). It also applies to all the simple families of labelled nonplane trees, $\mathcal{T} = \mathcal{Z} \star \mathfrak{P}_\Omega(\mathcal{T})$ defined by restrictions on node degrees (Example 9). In all these cases, linear-time approximate-size sampling is granted by Theorem 6.

7. Singular Boltzmann samplers.

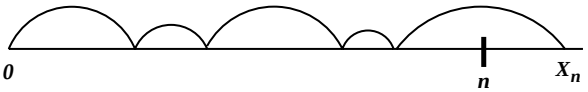
We now discuss two *infinite* categories of models, where it is possible to place oneself right at the singularity $x = \rho_C$ in order to develop rejection samplers from Boltzmann models. These "singular" rejection generators are freed from the necessity to adapt the control variable x to the target size n, thus making available implementations that only need a *fixed* set of constants to be determined once and for all, this independently of the value of n.

7.1. **Singular samplers for sequences.** The first type of singular generator we present is dedicated to the sequence construction: define a sequence construction to be *supercritical* if $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ and $\rho_A > \rho_C$ (so that $A(\rho_A^-) > 1$)). Put otherwise, the generating function of components $A(x)$ should cross the value 1 before it becomes singular. The generating function of $\mathcal{C}$ and $\mathcal{A}$ satisfy $C(z) = 1/(1 - A(z))$, so that the supercriticality condition implies that $A(\rho_C) = 1$ and the (dominant) singularity ρ_C of $C(x)$ is a pole. (This notion of supercriticality is borrowed from Soria [59] who showed it to be determinant in the probabilistic properties of sequences.)

Literally taken, the Boltzmann sampler ΓC of Section 3 taken with $x = \rho_C$ loops forever and generates objects of infinite size, as it produces a number of components equal to a "Geom(1)". This prevents us from using the rejection algorithm $\mu\mathcal{C}(x; n, \varepsilon)$ with $x = \rho$. However, one may adapt the idea by halting execution as soon as the target size has been attained. Precisely, the early-interrupt *singular sequence sampler* is defined as follows:

```
function σC(ρ; n); {Early-interrupt singular sequence sampler}
i := 0; repeat i := i + 1; γ_i := ΓA(ρ) until |(γ_1, ..., γ_i)| > n;
return((γ_1, ..., γ_i)); end.
```

The principle of the algorithm can be depicted as "leapfrogging" over n:



The *singular early-interrupt* sampler determined by the choice $x = \rho_C$ has excellent probabilistic and complexity-theoretic characteristics summarized in the following statement. There, we assume without loss of generality that $A(z)$ is aperiodic in the sense that the quantity $d := \gcd\{n \mid A_n \neq 0\}$ satisfies $d = 1$. (If $d \geq 2$, a linear change of the size functions brings us back to the aperiodic case.)

Theorem 7. *Consider a sequence construction, $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ that is supercritical and aperiodic. Then the early-interrupt singular sequence generator, $\sigma C(\rho_C; n)$ is a valid sampler for $\mathcal{C}$. It produces an object of size $n + O(1)$ in one trial with*

high probability. For a specifiable class $\mathcal{A}$, exact-size random generation in $\mathcal{C}$ is achievable from this generator by rejection in expected time $O(n)$.

Proof. Let X_n denote the random variable giving the size of the output of the early-interrupt singular sequence generator with tarket size n. The analysis of X_n can be treated by classical renewal theory [20, Sec. XIII.10], but we opt for a direct approach based on generating functions, which integrates smoothly within our general formalism.

The bivariate (probability) generating function with variable z marking the target size n and variable u marking the size X_n of the actually generated object is

$$F(z,u) := \sum_{n\geq 1}\sum_{m\geq n} \mathbb{P}(X_n = m)\, z^n u^m.$$

A trial decomposes into a sequence of samples of $\Gamma\mathcal{A}(\rho)$ ending by a sample that brings the total over n, which implies

$$F(z,u) = \frac{1}{1-A(\rho z u)}\mathcal{L}[A(\rho z u)] = \frac{z}{1-z}\frac{A(\rho u)-A(\rho z u)}{1-A(\rho z u)}.$$

There $\mathcal{L}[f(z)] := z(f(1)-f(z))/(1-z)$ is a linear operator, and, e.g.,

$$\mathcal{L}\Big[\frac{1}{1-zu}\Big] = z(u+u^2+\cdots)+z^2(u^2+u^3+\cdots)+z^3(u^3+u^4+\cdots)+\cdots,$$

so that all powers of the form $z^n u^\ell$ with $\ell \geq n$ are produced.

One checks that $F(z,1) = z/(1-z)$, as should be. Next the expected size $\mathbb{E}(X_n)$ of the output is given by the coefficient of z^n in

$$\begin{aligned}\frac{\partial}{\partial u}F(z,u)\big|_{u=1} &= \frac{z}{1-z}\frac{\rho A'(\rho)}{1-A(\rho z)}\\ &= \frac{z}{(1-z)^2}+\frac{\rho A''(\rho)}{2A'(\rho)}\cdot\frac{z}{1-z}+O(1) \qquad (z\to 1).\end{aligned}$$

This expansion at the polar singularity 1 then yields the expected "overshoot":

$$\mathbb{E}(X_n - n) \;=\; [z^n]\frac{\partial}{\partial u}F(z,u)\big|_{u=1} \;-\; n \;=\; \frac{\rho A''(\rho)}{2A'(\rho)}+O(1/n).$$

The second moment of the expected size of the output is similarly obtained via two successive differentiations. A simple computation then shows the variance of the overshoot to satisfy

$$\mathbb{E}((X_n-n)^2) - \mathbb{E}(X_n-n)^2 \;=\; O(1).$$

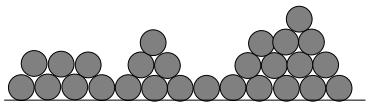
As a matter of fact, the discrete distribution of the overshoot is described by

$$\begin{aligned}\mathbb{P}(X_n - n \;=\; m) &= [z^n u^{n+m}]F(z,u) \;=\; [z^n u^m]\frac{z}{u-z}\left(1-\frac{1-A(\rho u)}{1-A(\rho z)}\right),\\ &= [z^{n+m}]\frac{1}{1-A(\rho z)} - \sum_{\ell=0}^{m-1}[z^{n+\ell}]\frac{1}{1-A(\rho z)}[u^{m-\ell}]A(\rho u),\\ &= \left(\frac{1}{\rho A'(\rho)}+O(1/n)\right)\left(1-\sum_{\ell=0}^{m-1}\mathbb{P}(N \;=\; \ell)\right)\\ &= \frac{\mathbb{P}(N\geq m)}{\mathbb{E}(N)}+O(1/n).\end{aligned}$$

where N denotes the random size of an element of $\mathcal{A}$ under the Boltzmann model of parameter ρ and the two last estimates hold for $n \to \infty$ uniformly in m. The distribution of N has exponential tails (since $\rho \equiv \rho_C$ lies strictly within the disc of convergence of $A(z)$), and thus the probability of a large overshoot decays geometrically fast. This proves that exact size n is attained in $O(1)$ trials. ∎

This theorem applies to "cores"of words without long runs (Equation (14) from Example 1) and it can be adapted to yield a generator of the full set $\mathcal{R}$. It applies to surjections (Example 6), for which exact-size generation becomes possible in linear time. It also provides a global setting for a variety of *ad hoc* algorithms developed by Louchard [43, 44, 46] in the context of efficient generation of certain types (directed, convex) of random planar diagrams known as "animals" and "polyominos".

EXAMPLE 10. *Coin fountains ($\mathcal{O}$).* A fountain is formed by starting with a row of coins, then stacking additional coins on top so that each new coin touches two in the previous row, for instance,



These configurations have been enumerated by Odlyzko and Wilf [53] and the counting sequence starts as (A005169 of [58])

$$1, 1, 1, 2, 3, 5, 9, 15, 26, 45, 78, 135, 234, 406, 704, \ldots$$

They correspond to Dyck paths (equivalently, Bernoulli excursions) taken according to area but disregarding length. A decomposition by slices taken at an angle of $\frac{2}{3}\pi$ (on the example, this gives 1,2,2,2,1,2,3,1,1,2,3,3,4) is then expressed by an infinite specification (not *a priori* covered by the standard paradigm):

$$\mathfrak{S}(\mathcal{Z}\mathfrak{S}(\mathcal{Z}^2\mathfrak{S}(\mathcal{Z}^3\mathfrak{S}(\cdots)))).$$

The OGF is consequently given by the continued fraction (see also [23]),

$$O(z) = \cfrac{1}{1 - \cfrac{z}{1 - \cfrac{z^2}{1 - \cfrac{z^3}{\cdots}}}}.$$

At top level, the singular Boltzmann sampler of Theorem 7 applies (write $\mathcal{O} = \mathfrak{S}(\mathcal{Q})$ and $O(z) = (1 - Q(z))^{-1}$), this even though $\mathcal{O}$ is not finitely specifiable. The root ρ of $Q(z) = 1$ is easily found to 50D,

$$\rho \doteq 0.57614876914275660229786857371993878235472466311 89,$$

see [53] for a transcendental equation satisfied by ρ that involves the q–exponential. The objects of $\mathcal{Q}$ needed are then with high probability of size at most $O(\log n)$ (by general properties of largest components in sequences [31]), so that they can be generated by whichever subexponential method is convenient (e.g., MAPLE's Combstruct) to the effect that the overall (theoretical and practical) complexity remains $O(n)$.

Precisely, the implementation runs like this. First define a family of finitely specifiable approximants to $\mathcal{Q}$, as follows:

$$\mathcal{Q}^{[j]} := \mathcal{Z}\mathfrak{S}(\mathcal{Z}^2\mathfrak{S}(\mathcal{Z}^3\mathfrak{S}(\cdots\mathcal{Z}^{j-1}\mathfrak{S}(\mathcal{Z}^j)\cdots))).$$

At any given time, the program operates with the class $Q^{[d]}$ of depth d: $Q^{[d]}(z)$ and $Q(z)$ coincide till terms of order $\nu(d) = \binom{d+1}{2} - 1$. The corresponding counts till $\nu(d)$ are assumed to be available, together with the corresponding exact-size samplers for $\mathcal{Q}^{[d]}$. (It is proves especially convenient here to appeal to algorithms based on the recursive method as provided by Combstruct.) In this way, one "knows" how to sample from Q till size $\nu(d)$, and from knowledge of the precise value of ρ, one also "knows" whenever a $\mathcal{Q}$ component of size larger than $\nu(d)$ might be required. (If so, adaptively increase the value of d and resume execution.) For instance, taking $d = 4$ (with $\nu = 9$) already suffices in 92% of the cases to produce an element of ΓQ, while $d = 20$ (and $\nu = 104$) suffices with probability about $1 - 2 \cdot 10^{-19}$ and is thus likely to cater for all simulation needs one might ever have.

The resulting implementation constants are reasonably *low*, so that random generation in the range of millions becomes feasible thanks to the singular Boltzmann generator. Here is for instance a fragment of a random fountain of size 100,004 ($n = 10^5$) obtained in this way (in only about a trillion clock cycles under Maple):

Dutour et al. [19] have previously employed an adaptation of the recursive method, but it is limited to sizes perhaps in the order of a few hundreds. □

Example 11. *Weighted Dyck paths and adsorbing staircase walks.* In [48], Martin and Randall examine (under the name of adsorbing walks) the generation of Dyck paths of length $2n$, where a path receives a weight proportional to λ^k if it hits the horizontal axis k times. Their Markov chain based algorithm has a high polynomial time complexity, perhaps as much as $O(n^{10})$, if not beyond. In contrast, for $\lambda > 2$, a Boltzmann sampler based on supercritical sequences has a complexity that is $O(n)$, this even when exact-size random generation is required. Precisely, let $\mathcal{D}$ be the class of Dyck paths defined by the grammar $\mathcal{D} = \mathbf{1} + \nearrow \mathcal{D} \searrow \mathcal{D}$ with OGF $D(z) = (1 - \sqrt{1-4z})/(2z)$ (with z marking size taken here to be half-length). One needs to generate objects from the weighted class $\mathcal{E} := \mathfrak{S}(\nearrow \mathcal{D} \searrow)$, viewed as weighted sequences of "arches" with OGF $(1 - z\lambda D(z))^{-1}$, where the coefficient

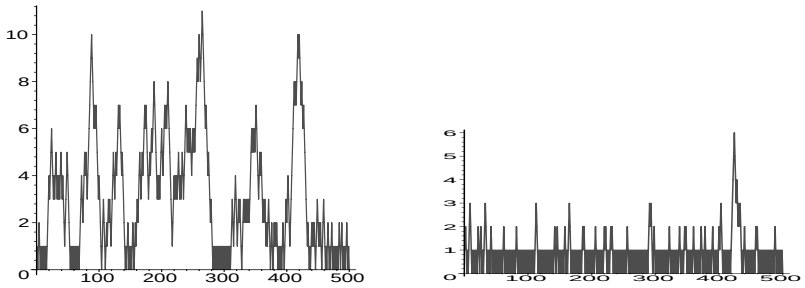


Figure 9. Weighted Dyck paths of length 500 corresponding to $\lambda = 2.1$ (left) and $\lambda = 3.1$ (right).

λ takes the proper weighting into account. The sequence is supercritical as soon as $\lambda > 2$, and the singular value of the Boltzmann parameter is found to be at $\rho = (\lambda - 1)/\lambda^2$. Then, the linear time generator is, for $\lambda > 2$:

let $\rho := \frac{\lambda-1}{\lambda^2}$, $D_k = \frac{1}{k+1}\binom{2k}{k}$;
repeat $S := 0$; repeat
generate K according to the distribution $\left\{\frac{\lambda-1}{\lambda} D_k \rho^k\right\}_{k=0}^{\infty}$;
$S := S + 2K + 2$; draw at random from $\nearrow \mathcal{D}_K \searrow$; {e.g., in linear time}
until $S \geq 2n$; until $S = 2n$.

There, the last successful run should be returned. (The case where $\lambda \leq 2$ is easily treated in linear time by direct combinatorics.) Figure 9 displays two such paths of length 500 (higher values of λ increase the number of contacts). □

The book by van Rensburg [66] describes models similar to the last two ones (in the context of critical phenomena in polymers and vesicles), a number of which are amenable to efficient Boltzmann sampling as they correspond to combinatorial classes that are specifiable.

7.2. **Singular samplers for recursive structures.** Recursive structures tend to conform to a universal complex-analytic pattern corresponding to a square-root singularity, that is, a singular exponent $-\alpha = 1/2$. This specific behaviour may be exploited, resulting in another variety of singular samplers.

In the statement below, a recursive class $\mathcal{C}$ is defined as the component $\mathcal{C} = \mathcal{F}_1$ of a system of mutually dependent equations,

$$\{\mathcal{F}_1 = \Psi_1(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m), \ldots, \mathcal{F}_m = \Psi_m(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m)\}$$

where the Ψ's are *any* functional term involving any of the basic constructors previously defined ('+', '×' or '⋆', and $\mathfrak{S}, \mathfrak{P}, \mathfrak{C}$; pointing is not allowed here). The system is said to be *irreducible* if the dependency graph between the $\mathcal{F}_j$ is strongly connected (every nonterminal $\mathcal{F}_j$ depends on any other $\mathcal{F}_k$). A class $\mathcal{F}$ is said to be of *lattice type* if the index set of the nonzero coefficients of $F(z)$ is contained in an arithmetic progression of some ratio d, with $d \geq 2$. (The terminology is borrowed from classical probability theory.) For instance, the class of "complete" binary trees ($\mathcal{F} = \mathcal{Z} + \mathcal{Z}\mathcal{F}^2$) only has objects of size $n = 1, 3, 5, 7, \ldots$, and is consequently lattice of ratio 2. Any lattice class is equivalent to a nonlattice one, upon redefining size via a linear transformation.

Lemma 3. *Consider a combinatorial class $\mathcal{C}$ defined by a recursive specification that is irreducible and non-lattice. Then $C(z)$ has a unique dominant singularity which is algebraic and of the square-root type, that is, with singular exponent $-\alpha = 1/2$ in the notations of Section 6.2.*

Proof (sketch). The $F_j(x)$ are implicitly defined by an image system $\mathbf{F} = \Psi[\mathbf{F}]$. The Jacobian matrix of Ψ,

$$\mathbf{J}(z) := \left(\frac{\partial}{\partial F_i} \Psi_j(\mathbf{F})\right)_{i,j}$$

is at least defined near the origin. Let $\lambda(z)$ be the spectral radius of $\mathbf{J}(z)$. For small enough positive x, the matrix $\mathbf{J}(x)$ is Perron–Frobenius by irreducibility. A local analysis of the Drmota–Lalley–Woods type [16, 41, 70] based on "failure" of the implicit function theorem in its analytic version establishes the following: each F_j has a singularity at ρ which is determined as the smallest positive root

of $\det \mathbf{J}(x) = 1$, and the behaviour of F_j there is of the square-root type in a Δ-domain. The non-lattice assumption implies that each F_j satisfies $|F(z)| < F(|z|)$ for any z satisfying $0 < |z| < \rho$ and $z \notin \mathbb{R}_{>0}$; by domination properties of analytic functions with positive coefficients and matrices with complex entries, this implies that $\lambda(z) < \lambda(|z|)$, whence the fact that each F_j is analytic on $|z| = \rho$, $z \neq \rho$. ∎

In view of Lemma 3, $C(z)$ is Δ-singular with an expansion of the form

$$C(x) \;=\; C(\rho) - c_0(1 - z/\rho)^{1/2} + O(1 - z/\rho), \tag{26}$$

where $C(\rho) > 0$ and $c_0 > 0$. Singularity analysis then implies that the coefficients are asymptotically given by

$$[z^n]C(z) = \frac{c_0}{2\sqrt{\pi}} \rho^{-n} n^{-3/2} \left(1 + O(n^{-1})\right). \tag{27}$$

(For details see [28, Ch. 8] and reference therein.) Consequently, the distribution of sizes at the critical value $x = \rho$ is of the form $\mathbb{P}(N = n) \propto n^{-3/2}$, which means that it has heavy tails. In particular, the expectation of size $\mathbb{E}(N)$ is infinite (this fact is well-known in the special case of critical branching processes). Such an observation precludes the use of straight-rejection Boltzmann sampling.

The idea of an early interruption discussed in the previous section may be adapted and extended. Consider in all generality a Boltzmann sampler $\Gamma C(x)$ built according to the design principles already exposed and let m be a *ceiling* (i.e., an uppperbound) imposed on the size of the required objects. It is possible to build a modification $\Gamma C^{<m}(x)$ of $\Gamma C(x)$ as follows: maintain a running count, implemented as a global counter K, of the number of atoms produced at any given time during a partial execution of sampling by $\Gamma C(x)$; the counter is regularly incremented as long as $K \leq m$ each time an atom is produced; however, as soon as K exceeds m, execution is interrupted and the "undefined" symbol $\perp$ is returned. Then, rejection can be piled on top of this sampler, which corresponds to the scheme:

> **function** $\nu C(x; n, \varepsilon)$; {Ceiled rejection sampler}
> **repeat** $\gamma := \Gamma C^{<m}(x; n(1+\varepsilon))$ **until** $(\gamma \neq \perp) \wedge (|\gamma| \geq n(1-\varepsilon))$;
> return(γ); **end**.

This ceiling technique optimizes *any* Boltzmann sampler for *any* value of x. The choice of the singular value $x = \rho$ makes the algorithm well-behaved for recursive classes.

Theorem 8. *Let $\mathcal{C}$ be a combinatorial class given by a recursive specification that is irreducible and aperiodic. Then the singular ceiled rejection sampler $\nu\mathcal{C}(\rho; n, \varepsilon)$, corresponding to a fixed tolerance $\varepsilon > 0$ succeeds in a number of trials whose expected value grows like $n^{1/2}/\zeta(\varepsilon)$ for a positive constant $\zeta(\varepsilon)$ given by* (30) *below.*

Moreover the cumulated size T_n of the generated and rejected objects during the call of $\nu\mathcal{C}(\rho; n, \varepsilon)$ satisfies as $n \to \infty$

$$\mathbb{E}(T_n) \sim \frac{n}{\varepsilon}\left((1-\varepsilon)^{1/2} + (1+\varepsilon)^{1/2}\right) \tag{28}$$

with its variance, $\sigma^2 = \mathbb{E}(T_n^2) - \mathbb{E}(T_n)^2$, being

$$\sigma^2 \sim \mathbb{E}(T_n)^2 + \frac{n^2}{\varepsilon}\left(\frac{1}{3}(1-\varepsilon)^{3/2} + (1+\varepsilon)^{3/2}\right). \tag{29}$$

Under these conditions, approximate-size sampling and exact-size sampling are of average-case complexity respectively $O(n)$ and $O(n^2)$.

Proof. Let $C(x)$ be the generating function of $\mathcal{C}$, and let $C^{<n_1}(x)C^{>n_2}(x), C^{[n_1,n_2]}(x)$ be the generating function for the subclass of those objects with size respectively strictly less than $n_1 = (1-\varepsilon)n$, strictly greater than $n_2 = (1+\varepsilon)n$, and between n_1 and n_2. The coefficients of $C(z)$ are known from Equation (27), so that $\Gamma C(\rho)$ produces sizes according to

$$\mathbb{P}(N = k) \;\sim\; \frac{c_0}{2C(\rho)\sqrt{\pi}}\, k^{-3/2}.$$

For any $\varepsilon > 0$, the probability that a single trial (one execution of the repeat loop) of the ceiled rejection sampler $\nu C(\rho; n, \varepsilon)$ succeeds is obtained by summing over all values of k in the interval $[n(1-\varepsilon), n(1+\varepsilon)]$. This probability thus decays like $\zeta(\varepsilon)n^{-1/2}$ where

$$\zeta(\varepsilon) \;=\; \frac{c_0}{5C(\rho)\sqrt{\pi}}\,((1+\varepsilon)^{5/2} - (1-\varepsilon)^{5/2}). \tag{30}$$

The expected number of trials follows.

Next, the probability generating function of the interruptive singular Boltzmann sampler targeted at $[n_1, n_2]$ is

$$F(u) \;=\; \sum_k \mathbb{P}(T_n = k)\, u^k.$$

From the decomposition of a call to $\nu\mathcal{C}$ into a sequence of unsuccessful trials (contributing to T_n) followed by a final successful trial (not contributing to T_n),

$$F(u) = \left(1 - \frac{1}{C(\rho)}\left(C^{<n_1}(\rho u) + C^{>n_2}(\rho)u^{n_2}\right)\right)^{-1} \frac{C^{[n_1,n_2]}(\rho)}{C(\rho)}.$$

(This is the cost *in addition* to the size of the last successful output, and it is assumed that the generation of objects with size larger than n_2 is interrupted at size n_2.) The moments of the cost are then given by

$$\mathbb{E}(T_n) = \frac{\partial}{\partial u}F(u)\big|_{u=1}, \qquad \mathbb{E}(T_n^2) = \frac{(u\partial)^2}{\partial u^2}F(u)\big|_{u=1}.$$

Taking partial derivatives, then specializing to $u = 1$, and observing that $C(x) - C^{<n_1}(x) - C^{>n_2}(x) = C^{[n_1,n_2]}(x)$, we get

$$\begin{aligned}
\mathbb{E}(T_n) &= \frac{\rho C'^{<n_1}(\rho) + n_2 C^{>n_2}(\rho)}{C^{[n_1,n_2]}(\rho)},\\
\mathbb{E}(T_n^2) &= \frac{\rho^2 C''^{<n_1}(\rho) + n_2(n_2-1)C^{>n_2}(\rho)}{C^{[n_1,n_2]}(\rho)} + 2\,\mathbb{E}(T_n)^2 \;+\; \mathbb{E}(T_n).
\end{aligned}$$

The asymptotic expression for the coefficients of $C(x)$ as given in (27) yields, by direct Euler-MacLaurin summation:

$$\begin{array}{llll}
\rho C'^{<n_1}(\rho) \;\sim\; 2c_0 n_1^{1/2}, & \rho^2 C''^{<n_1}(\rho) \;\sim\; \frac{2c_0}{3} n_1^{3/2},\\
C^{>n_2}(\rho) \;\sim\; 2c_0 n_2^{-1/2}, & C^{[n_1,n_2]}(\rho) \;\sim\; 2c_0\varepsilon n^{-1/2}.
\end{array} \tag{31}$$

The estimates (31) combine with the exact expressions of $\mathbb{E}(T_n)$ and $\mathbb{E}(T_n^2)$ to give the values stated in (28) and (29).

For a relative tolerance $\varepsilon = \varepsilon_n$ depending on n and tending to zero, the estimates become $\mathbb{E}(T_n) \sim \frac{2n}{\varepsilon}$ and $\sigma \sim \mathbb{E}(T_n)$, which implies the quadratic cost of exact-size sampling. ∎

The singular ceiled rejection sampler thus provides linear-time approximate-size random generation for all the simple varieties of trees of Example 2, including binary trees, unary-binary trees, 2-3 trees, and so on, for secondary structures (Example 3), and for noncrossing graphs (Example 4). In all these cases, exact-size is also achievable in quadratic time. The method does not require the pointing transformations of Section 6.3 and only necessitates a fixed number of constants, themselves independent of the target value n. The technique is akin to the "Florentine algorithm" invented by Barcucci–Pinzani–Sprugnoli [3] to generate prefixes of Motzkin words and some directed plane animals. The cost analysis given above is related to Louchard's work [45].

Note. Let $\mathcal{T}$ be a class of trees determined by restricting the degrees of nodes to lie in a finite set Ω, that is, $\mathcal{T} = \mathfrak{S}_\Omega(\mathcal{T})$ or $\mathcal{T} = \mathfrak{P}_\Omega(\mathcal{T})$, depending on whether the trees are embedded in the plane or not. The corresponding generating function satisfies $T(z) = z\phi(T(z))$ (see Example 9). For such trees, exact-size sampling can be performed in time $O(n^{3/2})$, which improves on the general bound $O(n^2)$ of Theorem 8. Indeed, in order to generate a tree of size n, it suffices to generate a Łukasiewicz code of length n, with steps in $\Omega - 1$. By Raney's conjugacy principle [42, Ch. 11] (also known as Dvoretzky and Motzkin's cycle lemma), this task itself reduces to generating at random a planar path of length n with steps in $\Omega - 1$ and with final altitude -1. When one places oneself right at the singular value ρ (for $T(z)$), the latter task is equivalent to sampling from n independent random variables, having support $\Omega - 1$ and probability generating function $\psi(z) = \phi(\rho z)/(z\phi(\rho))$, conditioned to sum to the value -1. Rejection (on the final value of the n–sum) achieves this in $O(n^{1/2})$ trials, by virtue of the local limit theorem for sums of discrete random variables. In this way, trees from any finitely generated family of trees can be sampled in total time $O(n^{3/2})$; equivalently, the technique makes it possible to sample from any branching process (with finitely supported offspring distribution) conditioned upon the size of the total progeny being n, this again in time $O(n^{3/2})$.

8. Conclusions

As shown here, combinatorial decompositions allow for random generation in low polynomial time. In particular, approximate-size random generation can often be effected in linear time, using algorithms that suitably exploit the "physics" of random combinatorial structures. Given the large number of combinatorial decompositions that have been gathered over the past two decades (see, e.g., [4, 28, 30]) we thus estimate to well over a hundred the number of classical combinatorial structures that are amenable to efficient Boltzmann sampling. In contrast with the recursive method [13, 29, 51], memory requirements are kept to a minimum since only a table of constants of size $O(1)$ is required.

For the reader's convenience, we gather in Figure 10 the best strategies that have been developed for each of the eleven pilot examples of this article. Naturally, a few of the basic cases are beaten by special-purpose combinatorial generators—this happens for permutations ($\mathcal{P}$), binary trees ($\mathcal{B}$), or mappings ($\mathcal{M}$) and Cayley trees ($\mathcal{T}$), where the counting sequences admit of a product form and specific bijections may be exploited to achieve exact-size sampling in linear time [51]. In such cases, however, the same complexity estimates continue to hold when Boltzmann sampling is applied to a large number of related classes, whereas dedicated combinatorial generators based on bijections generally break down. For instance, Boltzmann

	Structures		*Approx. size*	*Exact size*
1.	Runs	$\mathcal{R}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)
2.	Trees	$\mathcal{B}$	$O(n)$ (point.; sing. ceil.)	$O(n^2)$ (point.; sing. ceil.); $O(n^{3/2})$
3.	Secondary S.	$\mathcal{W}$	$O(n)$ (point.; sing. ceil.)	$O(n^2)$ (point.; sing. ceil.)
4.	Noncrossing G.	$\mathcal{X}$	$O(n)$ (point.; sing. ceil.)	$O(n^2)$ (point.; sing. ceil.)
5.	Set Part.	$\mathcal{S}$	$O(n)$ (reject.)	$O(n^{3/2}\sqrt{\log n})$ (reject.)
6.	Surjections	$\mathcal{Q}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)
7.	Permutations	$\mathcal{P}$	$O(n)$ (reject.)	$O(n^2)$ (reject.)
8.	Filaments	$\mathcal{F}$	$O(n)$ (reject.)	$O(n^{3/2})$ (reject.)
9.	Mappings	$\mathcal{M}$	$O(n)$ (point.)	$O(n^2)$ (point.; sing. ceil.)
10.	Fountains	$\mathcal{O}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)
11.	Weighted Dyck	$\mathcal{E}$	$O(n)$ (reject.)	$O(n)$ (sing. seq.)

FIGURE 10. The best strategies of the paper for Boltzmann sampling: rejection (Section 6.1, 6.2), pointing (Section 6.3), singular sequence (Section 7.1), and singular ceiled (Section 7.2).

algorithms for permutations can be adapted to obtain derangements ($\mathfrak{P}(\mathfrak{C}_{\geq 2}(\mathcal{Z}))$ and the like) and involutions ($\mathfrak{P}(\mathfrak{C}_{1,2}(\mathcal{Z}))$ and related structures); the branching process algorithms deduced automatically for binary trees apply equally well to unbalanced 2–3 trees ($\mathcal{U} = \mathcal{Z} + \mathcal{U}^2 + \mathcal{U}^3$) and to other families of trees defined by degree restrictions; random mappings satisfying various constraints then become amenable to Boltzmann sampling, and so on.

This article has shown that combinatorial samplers can be *compiled automatically* from formal specifications ("grammars") describing combinatorial models. The process is an efficient one as the program size of the sampler is derived by a single-pass linear-time formal transformation. A general-purpose implementation would most conveniently be developed on top of MAPLE's Combstruct, as many functionalities are already available there. As matter of fact, a prototype has been developed by Marni Mishna; together with other experiments, it confirms the ease of implementation and the practical efficiency of Boltzmann sampling for the random generation of many different types of combinatorial structures.

In forthcoming works, we propose to demonstrate the versatility of Boltzmann sampling for a number of simulation needs including:

— the extension of the set of allowed constructions, e.g., in the unlabelled case, sampling for multisets ($\mathfrak{M}$, repetitions are allowed), powersets ($\mathfrak{P}$, no repetitions allowed), cycles ($\mathfrak{C}$), and the substitution operation;

— multivariate extensions, meaning the sampling of configurations according to a constraint on size *and* on an auxiliary parameter (e.g., words of some length containing an unusual number of occurrences of a designated pattern);

— the realization of Boltzmann samplers using only discrete sources of randomness and basic logical operations in the style of Knuth and Yao's fundamental study [40]—nearly linear boolean (bit level) complexity still seems to be achievable in many cases of practical interest.

Acknowledgements: The authors are grateful to Alain Denise, Bernard Ycart, Brigitte Vallée, Jim Fill, Marni Mishna, Paul Zimmermann, and Philippe Robert for bibliographical suggestions, programming ideas, as well as encouragements and architectural remarks.

This work was supported in part by the ALCOM–FT Project IST-1999-14186 of the European Union. *Merci* also to Gilles Kahn and INRIA for backing the ALCOPHYS Action under which some of these ideas were hatched and to CNRS (GDR ALP and Department STIC) for its sustained support of the French Group ALÉA. *Grazie mille* finally to Alberto del Lungo and Renzo Pinzani for kindly offering an occasion to expose an early form of these ideas at the GASCOM meeting, Sienna, November 2001.

REFERENCES

[1] AHO, A. V., AND CORASICK, M. J. Efficient string matching: an aid to bibliographic search. *Communications of the ACM 18* (1975), 333–340.

[2] ARNEY, J., AND BENDER, E. D. Random mappings with constraints on coalescence and number of origins. *Pacific Journal of Mathematics 103* (1982), 269–294.

[3] BARCUCCI, E., PINZANI, R., AND SPRUGNOLI, R. The random generation of directed animals. *Theoretical Computer Science 127*, 2 (1994), 333–350.

[4] BERGERON, F., LABELLE, G., AND LEROUX, P. *Combinatorial species and tree-like structures.* Cambridge University Press, Cambridge, 1998.

[5] BRENT, R. P., AND POLLARD, J. M. Factorization of the eighth Fermat number. *Mathematics of Computation 36* (1981), 627–630.

[6] BURRIS, S. N. *Number theoretic density and logical limit laws*, vol. 86 of *Mathematical Surveys and Monographs.* American Mathematical Society, Providence, RI, 2001.

[7] COMPTON, K. J. A logical approach to asymptotic combinatorics. I. First order properties. *Advances in Mathematics 65* (1987), 65–96.

[8] COMPTON, K. J. A logical approach to asymptotic combinatorics. II. Second–order properties. *Journal of Combinatorial Theory,* Series A *50* (1987), 110–131.

[9] COMTET, L. *Advanced Combinatorics.* Reidel, Dordrecht, 1974.

[10] DEMBO, A., VERSHIK, A., AND ZEITOUNI, O. Large deviations for integer partitions. *Markov Processes and Related Fields 6*, 2 (2000), 147–179.

[11] DEN HOLLANDER, F. *Large deviations.* American Mathematical Society, Providence, RI, 2000.

[12] DENISE, A., DUTOUR, I., AND ZIMMERMANN, P. CS: a MuPAD package for counting and randomly generating combinatorial structures. In *Proceedings of 10th Conference on Formal Power Series and Algebraic Combinatorics (FPSAC'98)* (1998), pp. 195–204.

[13] DENISE, A., AND ZIMMERMANN, P. Uniform random generation of decomposable structures using floating-point arithmetic. *Theoretical Computer Science 218*, 2 (1999), 233–248.

[14] DEVROYE, L. *Non-Uniform Random Variate Generation.* Springer Verlag, 1986.

[15] DOMB, C., AND BARRETT, A. Enumeration of ladder graphs. *Discrete Mathematics 9* (1974), 341–358.

[16] DRMOTA, M. Systems of functional equations. *Random Structures & Algorithms 10*, 1–2 (1997), 103–124.

[17] DUCHON, P. Relaxed random generation of trees. *Algorithms Seminar*, 05-11-01, 2001.

[18] DUCHON, P., FLAJOLET, P., LOUCHARD, G., AND SCHAEFFER, G. Random sampling from Boltzmann principles. In *Automata, Languages, and Programming* (2002), P. Widmayer et al., Ed., no. 2380 in Lecture Notes in Computer Science, Springer Verlag, pp. 501–513.

[19] DUTOUR, I., AND FÉDOU, J.-M. Object grammars and random generation. *Discrete Mathematics and Theoretical Computer Science 2* (1998), 47–61.

[20] FELLER, W. *An Introduction to Probability Theory and its Applications*, third ed., vol. 1. John Wiley, 1968.

[21] FELLER, W. *An Introduction to Probability Theory and Its Applications*, vol. 2. John Wiley, 1971.

[22] FILL, J. A., AND HUBER, M. The randomness recycler: A new technique for perfect sampling. In *Proceeding of the 41th Annual IEEE Symposium on Foundations of Computer Science* (2000), pp. 503–511.

[23] FLAJOLET, P. Combinatorial aspects of continued fractions. *Discrete Mathematics 32* (1980), 125–161.

[24] FLAJOLET, P., AND NOY, M. Analytic combinatorics of non-crossing configurations. *Discrete Mathematics 204*, 1-3 (1999), 203–229. (Selected papers in honor of Henry W. Gould).

[25] Flajolet, P., and Odlyzko, A. M. Random mapping statistics. In *Advances in Cryptology* (1990), J.-J. Quisquater and J. Vandewalle, Eds., vol. 434 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 329–354. Proceedings of Eurocrypt'89, Houtalen, Belgium, April 1989.

[26] Flajolet, P., and Odlyzko, A. M. Singularity analysis of generating functions. *SIAM Journal on Algebraic and Discrete Methods 3*, 2 (1990), 216–240.

[27] Flajolet, P., Salvy, B., and Zimmermann, P. Automatic average–case analysis of algorithms. *Theoretical Computer Science 79*, 1 (Feb. 1991), 37–109.

[28] Flajolet, P., and Sedgewick, R. *Analytic Combinatorics*. 2001. Book in preparation: Individual chapters are available as INRIA Research Reports 1888, 2026, 2376, 2956, 3162, 4103 and electronically under http://algo.inria.fr/flajolet/Publications/books.html.

[29] Flajolet, P., Zimmerman, P., and Van Cutsem, B. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science 132*, 1-2 (1994), 1–35.

[30] Goulden, I. P., and Jackson, D. M. *Combinatorial Enumeration*. John Wiley, New York, 1983.

[31] Gourdon, X. Largest component in random combinatorial structures. *Discrete Mathematics 180*, 1-3 (1998), 185–209.

[32] Greene, D. H. *Labelled formal languages and their uses*. PhD thesis, Stanford University, June 1983. Available as Report STAN-CS-83-982.

[33] Greene, D. H., and Knuth, D. E. *Mathematics for the analysis of algorithms*. Birkhäuser, Boston, 1981.

[34] Harary, F., and Palmer, E. M. *Graphical Enumeration*. Academic Press, 1973.

[35] Hayman, W. K. A generalization of Stirling's formula. *Journal für die reine und angewandte Mathematik 196* (1956), 67–95.

[36] Howell, J. A., Smith, T. F., and Waterman, M. S. Computation of generating functions for biological molecules. *SIAM Journal on Applied Mathematics 39*, 1 (1980), 119–133.

[37] Huang, K. *Statistical Mechanics*, 2nd ed. John Wiley, 1987.

[38] Johnson, N. L., and Kotz, S. *Discrete Distributions*. John Wiley, 1969.

[39] Knuth, D. E. *The Art of Computer Programming*, 3rd ed., vol. 2: Seminumerical Algorithms. Addison-Wesley, 1998.

[40] Knuth, D. E., and Yao, A. C. The complexity of nonuniform random number generation. In *Algorithms and complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1976)*. Academic Press, New York, 1976, pp. 357–428.

[41] Lalley, S. P. Finite range random walk on free groups and homogeneous trees. *Ann. Probab. 21*, 4 (1993), 2087–2130.

[42] Lothaire, M. *Combinatorics on Words*, vol. 17 of *Encyclopedia of Mathematics and its Applications*. Addison–Wesley, 1983.

[43] Louchard, G. Probabilistic analysis of some (un)directed animals. *Theoretical Computer Science 159*, 1 (1996), 65–79.

[44] Louchard, G. Probabilistic analysis of column-convex and directed diagonally-convex animals. *Random Structures & Algorithms 11*, 2 (1997), 151–178.

[45] Louchard, G. Asymptotic properties of some underdiagonal walks generation algorithms. *Theoretical Computer Science 218*, 2 (1999), 249–262.

[46] Louchard, G. Probabilistic analysis of column-convex and directed diagonally-convex animals. II. Trajectories and shapes. *Random Structures & Algorithms 15*, 1 (1999), 1–23.

[47] Lyons, R., Pemantle, R., and Peres, Y. Conceptual proofs of $L \log L$ criteria for mean behavior of branching processes. *The Annals of Probability 23*, 3 (1995), 1125–1138.

[48] Martin, R., and Randall, D. Sampling adsorbing staircase walks using a new Markov chain decomposition method. In *Proc. of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2000)* (2000), pp. 492–502.

[49] Meir, A., and Moon, J. W. On the altitude of nodes in random trees. *Canadian Journal of Mathematics 30* (1978), 997–1015.

[50] Milenković, O., and Compton, K. J. Probabilistic transforms for combinatorial urn models. Preprint, 2002.

[51] Nijenhuis, A., and Wilf, H. S. *Combinatorial Algorithms*, second ed. Academic Press, 1978.

[52] Odlyzko, A. M. Asymptotic enumeration methods. In *Handbook of Combinatorics*, R. Graham, M. Grötschel, and L. Lovász, Eds., vol. II. Elsevier, Amsterdam, 1995, pp. 1063–1229.

[53] Odlyzko, A. M., and Wilf, H. S. The editor's corner: n coins in a fountain. *American Matematical Monthly 95* (1988), 840–843.
[54] Olver, F. W. J. *Asymptotics and Special Functions*. Academic Press, 1974.
[55] Quisquater, J.-J., and Delescaille, J.-P. How easy is collision search? Application to DES. In *Proceedings of EUROCRYPT'89* (1990), vol. 434 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 429–434.
[56] Schaeffer, G. Random sampling of large planar maps and convex polyhedra. In *Proceedings of the thirty-first annual ACM symposium on theory of computing (STOC'99)* (Atlanta, Georgia, May 1999), ACM press, pp. 760–769.
[57] Shepp, L. A., and Lloyd, S. P. Ordered cycle lengths in a random permutation. *Transactions of the American Mathematical Society 121* (1966), 340–357.
[58] Sloane, N. J. A. *The On-Line Encyclopedia of Integer Sequences*. 2000. Published electronically at `http://www.research.att.com/~njas/sequences/`.
[59] Soria-Cousineau, M. *Méthodes d'analyse pour les constructions combinatoires et les algorithmes*. Doctorat ès sciences, Université de Paris–Sud, Orsay, July 1990.
[60] Stanley, R. P. *Enumerative Combinatorics*, vol. I. Wadsworth & Brooks/Cole, 1986.
[61] Stanley, R. P. *Enumerative Combinatorics*, vol. II. Cambridge University Press, 1998.
[62] Stein, P. R., and Waterman, M. S. On some new sequences generalizing the Catalan and Motzkin numbers. *Discrete Mathematics 26*, 3 (1979), 261–272.
[63] Van Cutsem, B. Combinatorial structures and structures for classification. *Computational Statistics & Data Analysis 23*, 1 (1996), 169–188.
[64] Van Cutsem, B., and Ycart, B. Indexed dendrograms on random dissimilarities. *Journal of Classification 15*, 1 (1998), 93–127.
[65] Van der Hoeven, J. Relax, but don't be too lazy. Preprint, 2001. 65 pages. Available at `http://www.math.u-psud.fr/~vdhoeven/`.
[66] van Rensburg, E. J. J. *The statistical mechanics of interacting walks, polygons, animals and vesicles*. Oxford University Press, Oxford, 2000.
[67] Vershik, A. M. Statistical mechanics of combinatorial partitions, and their limit configurations. *Funktsional'nyĭ Analiz i ego Prilozheniya 30*, 2 (1996), 19–39.
[68] Weiermann, A. Zero-one law characterizations of ε_0. In *Mathematics and Computer Science II: Algorithms, Trees, Combinatorics and Probabilities* (Basel, 2002), B. Chauvin, P. Flajolet, D. Gardy, and A. Mokkadem, Eds., Trends in Mathematics, Birkhäuser Verlag, pp. 527–539.
[69] Wilf, H. S. *Generatingfunctionology*. Academic Press, 1990.
[70] Woods, A. R. Coloring rules for finite trees, and probabilities of monadic second order sentences. *Random Structures Algorithms 10*, 4 (1997), 453–485.
[71] Zimmermann, P. Arithmétique en précision arbitraire. Research Report 4272, Institut National de Recherche en Informatique et en Automatique, Sept. 2001. 22 pages.

P. Duchon.: LaBRI, Université de Bordeaux I, 351 Cours de la Libération, F-33405 Talence Cedex, France; duchon@labri.fr

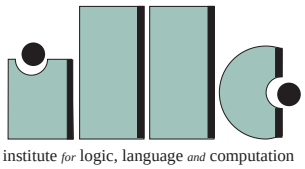
P. Flajolet: Algorithms Project, INRIA-Rocquencourt, F-78153 Le Chesnay, France; Philippe.Flajolet@inria.fr.

G. Louchard, Université Libre de Bruxelles, Département d'informatique, Boulevard du Triomphe, B-1050 Bruxelles, Belgique; louchard@ulb.ac.be

G. Schaeffer: Adage Group, LORIA – CNRS, 615 rue du jardin Botanique, F-54000 Villers-les-Nancy, France. Gilles.Schaeffer@loria.fr

Computations
in
Propositional Logic

ILLC Dissertation Series 1996-1



For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-5256090
fax: +31-20-5255101
e-mail: illc@fwi.uva.nl

Computations
in
Propositional Logic

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof.dr P.W.M. de Meijer
ten overstaan van een door het college van dekanen ingestelde
commissie in het openbaar te verdedigen in de
Aula der Universiteit
op dinsdag 12 maart 1996 te 15.00 uur

door

Alex Hendriks

geboren te Deventer.

Promotor: Prof.dr. G.R. Renardel de Lavalette
Faculteit der Wiskunde en Natuurwetenschappen
Rijksuniversiteit Groningen
Blauwborgje 3
9747 AC Groningen

Co-promotor: Dr. D.H.J. de Jongh
Faculteit der Wiskunde, Informatica, Natuurkunde en Sterrenkunde
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

The investigations were supported by the Mathematical Research Foundation (SMC), which is subsidized by the Netherlands Organization for Scientific Research (NWO).

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Hendriks, Alex

Computations in propositional logic / Alex Hendriks. -
Amsterdam : Institute for Logic, Language and
Computation, Universiteit van Amsterdam. - Ill. - (ILLC
dissertation series ; 1996-01)
Proefschrift Universiteit van Amsterdam- - Met index, lit.
opg. - Met samenvatting in het Nederlands.
ISBN 90-74795-44-7
NUGI 855
Trefw.: propositielogica / intuïtionisme / modale logica.

Contents

Dankwoord

Graag zou ik hier iedereen, met naam en toenaam, bedanken die op de een of andere manier heeft bijgedragen aan het totstandkomen van dit proefschrift. Iets wat helaas niet gaat. Maar temidden van de velen in wie ik mijn leermeesters, voorbeelden en toeverlaten herken, zijn er bij deze gelegenheid enkele die beslist niet ongenoemd kunnen blijven.

Allereerst wil ik hier uitdrukkelijk mijn promotor Gerard Renardel de Lavalette en mijn co-promotor Dick de Jongh bedanken. Zonder hun aanmoediging was ik er nooit aan begonnen, zonder hun steun was het nooit afgekomen, zonder hun kritiek was het nooit wat geworden en zonder hun vriendschap was het allemaal ook minder de moeite waard geweest.

Dick heeft mij bovendien de afgelopen vier jaar als medebewoner op zijn kamer geduld. Een privilege waaraan de overige bewoners op de gang in de loop der tijd nog meer glans wisten te geven. Ik dank hierbij met name professor Troelstra, Andreja Prijatelj en Paul van Ulsen voor zowel de stimulerende als de amusante gesprekken die ik met hen mocht voeren.

De vakgroep toegepaste logica van de faculteit voor Wijsbegeerte van de Rijksuniversiteit van Utrecht ben ik erkentelijk voor het jaar (1991) dat ik in Utrecht heb mogen werken om de grondslag te leggen voor dit proefschrift.

Met Henk van Riemsdijk, John Tromp en Jan Zwanenburg heb ik de afgelopen jaren korte of langere tijd mogen samenwerken en het enthousiasme kunnen delen over de mogelijkheden die computers bieden voor onderzoek in de logica.

Volodya Shavrukov wil ik hier nogmaals bedanken voor zijn kritiek waarmee hij mij behoed heeft voor een ernstig misverstand over de structuur van de exacte modellen in de modale logica.

De leden van mijn promotiecommissie, professor Van Benthem, professor Kamp, dr. Rodenburg, professor Troelstra en dr. Visser, ben ik zeer erkentelijk voor hun commentaar en suggesties voor verbeteringen.

Dankbaar ben ik verder voor het geduld en de welwillendheid waarmee vrienden, familieleden en mijn collega's bij het GAK de afgelopen jaren mijn verhalen over exotische propositielogica's hebben aangehoord.

Dankbaar ben ik ook mijn schoonouders voor hun vertrouwen en steun waardoor ik mijn studie heb kunnen afmaken.

Meer nog dan al deze dank verdienen mijn vrouw Hannie, mijn dochters Maartje en Aletta en onze pleegzoons van de afgelopen jaren, Rob en Gabor. Zij hebben niet alleen het geduld opgebracht om mij aan het werk te laten, zij hebben mij er ook dikwijls uit losgerukt om samen met hen nieuwe energie op te doen.

Dit proefschrift draag ik op aan mijn ouders, Jan Hendriks en Aleida Hendriks-Velders. Hun betrokkenheid was mij ook bij het totstandkomen van dit proefschrift tot steun. Ik ben blij dat ik mijn vader het resultaat heb kunnen laten zien.

Amsterdam Lex Hendriks
Januari, 1996.

Chapter 1

General Introduction

The main topic of this thesis is the representation of fragments of intuitionistic and modal propositional logic by (usually finite) structures, called *exact models*. One of the reasons for the interest in properties of fragments with such a finite representation is the possibility of designing computer programs to decide derivability within the fragment. In general, this kind of program, based on checking the validity of formulas in a model, is much more efficient than traditional theorem proving. This efficiency of 'model checking versus theorem proving' has in recent years attracted the attention of researchers in artificial intelligence and knowledge representation [HV 91].

For the benefits of model checking we have to pay a price. Theorem provers are 'general purpose' tools, accepting any formula in the language of the logic (obviously with certain practical limitations). However, finite representations such as exact models exist only when we cut down the expressive power of the language.

In this thesis we focus our attention on *finite fragments* of propositional logics, languages with restrictions on the use of atomic subformulas and connectives, that have a finite *Lindenbaum algebra* or *diagram* as we prefer to call it here.

The structure of these finite diagrams can be calculated and studied using efficient computer programs based on model checking. Knowledge of the structure of diagrams can be used in constructing new finite complete models.

The history of this kind of research into the structure of finite fragments can be traced back to the calculation of diagrams by Skolem [Skolem 13] and Lindenbaum's suggestion to use (equivalence classes of) formulas in semantics [Mostowski 65]. The discovery of the lattice of the one-variable fragment of intuitionistic propositional logic by Rieger [Rieger 49] and its rediscovery by Nishimura [Nishimura 60] proves that, although perhaps not always very prominent, the interest in the subject remained in the years after.

After the introduction of semantic tableaux by Beth [Beth 55], Kanger and Hintikka, and the invention of Kripke semantics for modal as well as intuitionistic logic by Kripke [Kripke 65] and others, a more systematic investigation of the semantical fine structure of fragments seemed possible.

As Beth pointed out in [Beth 55], the strongly mechanical character of his semantic tableau procedures suggests the possibility of constructing a *logical machine*. In 1955 Beth imagined this futuristic 'computer' to display its results using a crossbreed of a traffic light and a telegraph. A *red* light would announce a proof, to be produced on a strip of paper, and a *yellow* light would announce the machine to print a finite counter-example.

The construction of the logical machine would of course depend on the logic used. Only in those cases where the derivability of a formula from a finite set of formulas is decidable, one may expect the machine always to halt after a finite amount of time.

In the case of predicate logic, adding a green light to the machine announcing the construction of an infinite tableau would make the implementation of the specifications impossible (if on every input the machine has to switch on one of the lights after a finite period of time).

Nowadays, at a time where there are probably more computers than traffic lights around, it is no problem to implement Beth's logical machines as computer programs. Of course, if we want the machine to halt on every input, such a computer program is only possible for decidable logics, such as most propositional logics.

In the early sixties, as soon as computers came within the reach of university scientists, Beth stimulated his students De Jongh and Kamp to develop computer programs deciding derivability and compute diagrams in intuitionistic propositional logic (**IpL**).

In 1963 De Jongh and Kamp succeeded in making the computer decide correctly whether a formula was derivable in **IpL**. If not, the program produced the description of a Kripke model that served as a counter-example.

However, the program was too time-consuming to be of any practical use in studying diagrams. In the late seventies, this line of research was picked up by the author [Hendriks 80] who wrote Algol68 programs that could decide derivability in **IpL** and compute small diagrams. Improved results were obtained by Van Riemsdijk [Riemsdijk 85] with Pascal programs.

These computer programs, using algorithms based on the semantic tableaux method, realized the kind of logical machine that Beth envisaged 25 years earlier.

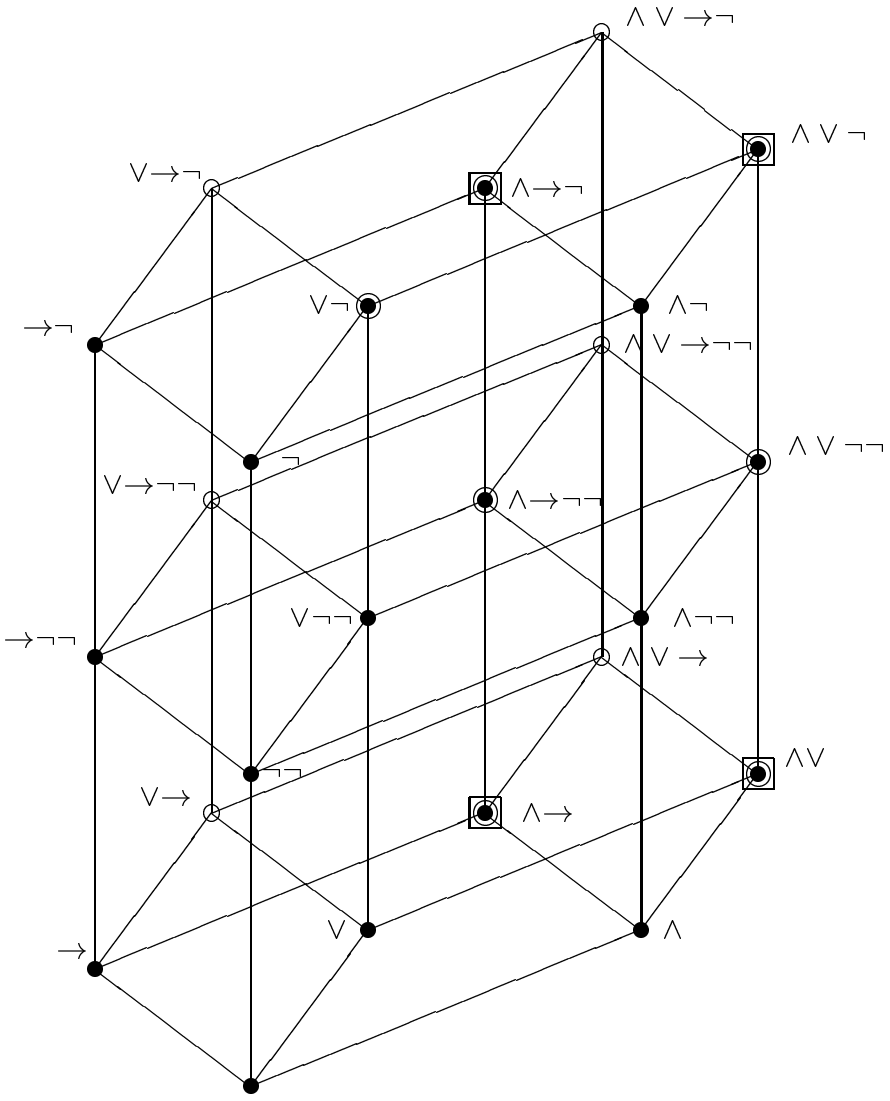
By that time the history of the subject had also made a detour in algebra. Investigating the algebraic structure of diagrams of $[\rightarrow]$ fragments of **IpL**, Diego proved that all diagrams of $[\rightarrow]$ with a finite number of propositional variables are finite [Diego 66].

Independently, Urquhart gave a simpler prove in [Urquhart 74] and in 1975, De Bruijn proved the same result for all diagrams of $[\wedge, \rightarrow]$ [Bruijn 75a]. In this proof De Bruijn introduced the notion of an *exact model* . An exact model is a part of the Lindenbaum algebra, such that the lattice of upward closed subsets of the exact model (ordered by inclusion) is isomorphic to the entire Lindenbaum algebra.

Let us write $[\wedge, \rightarrow]^n$ for the **IpL** fragment with formulas generated from the atomic formulas $\{p_1, \ldots, p_n\}$ and the connectives $\wedge$ and $\rightarrow$. To construct the exact model of the fragment $[\wedge, \rightarrow]^3$, De Bruijn used a computer. He also published a

computer program that used this exact model in deciding derivability in the fragment [Bruijn 75b].

In 1987 De Jongh, Renardel de Lavalette and the author started working on computer aided research into the structure of diagrams of **IpL**. With help from Van Riemsdijk and Tromp new computer programs were developed based on exact models. The present work reflects the results of the research that started in this group.



1. FIGURE. *The lattice of fragments in* **IpL**.

De Bruijn's concept of exact model was reformulated in the more familiar context of Kripke models, simplified and generalized to compute exact models, not only of $[\wedge, \rightarrow, \neg]$ fragments (see [JHR 91]) but also those of other fragments of **IpL** [Hendriks 93]. As a result we are now able to construct a finite complete Kripke model for every finite fragment in the lattice of fragments depicted in figure 1. Each

node in this diagram stands for a certain set of connectives and hence for an infinity of fragments, one for each number of atoms. Note that the double negation ($\neg\neg$) is treated as a primitive operator.

In figure 1 the fragments with an infinite diagram (at least for more than one propositional variable) are denoted by an open circle, the others by a closed circle. Whenever the fragment (again over a finite set of atoms) has an exact model, the closed circle of the fragment is surrounded by an open circle. Fragments with an exact Kripke model are marked by a square.

As can be seen from the lattice, every fragment considered here with a finite diagram is a subfragment of a fragment with an exact Kripke model.

The fragments in the lattice above are obtained by simply deleting one or more of the usual connectives (on top of the restriction to a finite set of atoms). For several reasons, restricting the use of connectives in a more sophisticated way is an interesting alternative. Observe for example that the interplay of implication and disjunction cannot be studied in finite fragments, since in every finite fragment either one of them will be absent. Nor is there a non-trivial sequence of these 'simple' finite fragments which has $\mathbf{IpL}^n$ as its union.

If we turn to modal logic, the situation is even worse. By simply deleting connectives we will not, in general, obtain finite fragments if the modal operators are still available, and without them we are left with fragments of classical propositional logic, **CpL**.

In modal logic, there are some well-known results concerning formulas with a restricted *modal depth*, as the nesting of modal operators is usually called (see in particular [Fine 74]). In the context of an attempt to characterize formulas in provability logic using sets of worlds of a certain type in a Kripke model, these results inspired the introduction of the notion of *semantic type*. The notion of semantic type turned out to be a versatile tool also in intuitionistic propositional logic.

Just as the restriction of modal depth in modal logic, the restriction of nesting of implications in **IpL** fragments with a finite number of propositional variables yields fragments with finite diagrams that have exact Kripke models. The structure of these exact models will be studied[1] in Chapter 4. Intuitionistic propositional logic can be obtained as the limit of a sequence of fragments with an increasing nesting of implication and an increasing number of propositional variables.

The problem in provability logic, **L**, that brought us on the trail of the semantic types was the computation of the *exactly provable* formulas[2] in the fragment $\mathbf{L}_1^1$ (see Chapter 5). According to Solovay's theorem on provability interpretations for formulas of **L** [Solovay 76], the theorems of **L** are those modal formulas that are provable in Peano arithmetic (**PA**) under arbitrary arithmetical interpretations (interpreting $\Box$ as the formalized provability predicate in **PA**). If we fix the arithmetical interpretation of one or more of the propositional variables, the interpreted formulas true in

[1] Some of the research was done in cooperation with Zwanenburg [Zwanenburg 94].

[2] The term 'exact' in 'exactly provable' has no relation to its use in 'exact models'.

PA form an *interpretable theory*:

$$\{\phi(p_1,\ldots,p_n) \mid \vdash_{\mathbf{PA}} \phi^*(A_1,\ldots,A_n)\}$$

for certain arithmetic sentences $A_1,\ldots,A_n$.

There are only finitely many interpretable theories in the fragment $\mathbf{L}^n_m$, with n atoms and a nesting of the provability operator less or equal m.

If we introduce the relation $\phi \vdash\!\!\!\sim \psi$ for $\phi \wedge \Box\phi \vdash \psi$, then we can reformulate the condition, found by V. Shavrukov, that is both necessary and sufficient for a ϕ in **L** to be the axiom of an interpretable theory [Shavrukov 93]:

$$\text{for all } \psi,\chi \quad \phi \vdash\!\!\!\sim \Box\psi \vee \Box\chi \quad \Rightarrow \quad \phi \vdash\!\!\!\sim \psi \text{ or } \phi \vdash\!\!\!\sim \chi.$$

A ϕ which is the axiom of an interpretable theory in the sense that there is an interpretation such that:

$$\phi \vdash\!\!\!\sim \psi \quad \Leftrightarrow \quad \vdash_{PA} \psi^*$$

is called an *exactly provable* formula. The strong disjunction property above turns out to have a characterization in semantical terms by means of which it is possible to calculate the exactly provable formulas in the fragment $\mathbf{L}^1_1$.

The results based on these ideas were first published in [HJ 96].

1.1 Outline of the thesis

Each chapter starts with a short introduction and a preliminary section. The preliminaries common to all chapters can be found in section 1.2.

Chapter 2 is an introduction into the theory of semantic types and exact models. Some related results, as from [Fine 74], [Jankov 68] and [De Jongh 70], are presented in this framework.

Chapter 3 is an overview of finite fragments of **IpL** with a restricted set of connectives. For fragments with an exact model the construction of the exact model is given. Part of these results were published in [JHR 91] and most of them can also be found in [Hendriks 93]. In this chapter these results are presented for the first time within the framework of semantic types introduced in the previous chapter.

Chapter 4 deals with the structure of exact Kripke models for fragments of **IpL** with restricted nesting of implication.

Chapter 5 applies some of the techniques of the previous chapters to the computation of exactly provable formulas in provability logic. This chapter is a revised version of [HJ 96], with an emphasis on the contributions of the present author, viz. the introduction of semantic types and the computation of the exactly provable formulas with no nesting of the provability operator.

Chapter 6 describes a family of theorem testers based on semantic tableaux, including a tester for **IpL** and testers for several modal logics and contains a description of the algorithms to compute diagrams and exact models.

Appendix A contains some of the more important parts of the computer programs (in the programming language C), that are described in this thesis.

Appendix B is a collection of examples of the output of the computer programs that calculated diagrams of fragments and the exactly provable formulas in $\mathbf{L}_1^1$.

Appendix C contains tables of the number of equivalence classes computed for several fragments of **IpL**.

1.2 General preliminaries

The language of classical propositional logic (**CpL**), intuitionistic propositional logic (**IpL**) and modal propositional logic used in the consecutive chapters consists of the constants $\bot$ and $\top$ and an infinite stock of propositional variables $\{p_1, p_2, \ldots\}$, also called atomic formulas (or simply atoms), together with the usual propositional connectives $\{\wedge, \vee, \rightarrow, \neg\}$ (and sometimes $\neg\neg$). In the case of modal logic also $\Box$ and $\Diamond$ are included. We will, in the case of classical (modal) logic, most of the time treat $\vee, \rightarrow$ and $\Diamond$ as defined from $\wedge, \neg$ and $\Box$ (but sometimes $\vee, \rightarrow$ and $\Box$ defined from $\wedge, \neg$ and $\Diamond$). On the other hand, in **IpL** we will take $\neg\phi$ to be defined as $\phi{\rightarrow}\bot$.

To avoid a plethora of parentheses, in writing our formulas, we define the order in which the connectives take preference above each other as:

$$\Diamond \qquad \Box \qquad \neg \qquad \wedge \qquad \vee \qquad \rightarrow.$$

Hence, $\neg\Box\neg p{\rightarrow}q \wedge r \vee s$ is equivalent to $\neg(\Box(\neg p)){\rightarrow}((q \wedge r) \vee s)$.

The derivability relation for a logic L will be denoted by $\vdash_L$ or by $\vdash$ if the choice of the logic is obvious from the context. Formulas ϕ and ψ are called *equivalent* in the logic L, $\phi \equiv_L \psi$ (or $\phi \equiv \psi$ if L is obvious), if they are interderivable: $\phi \vdash_L \psi$ and $\psi \vdash_L \phi$. The derivability relations of the logics treated in the sequel are assumed to be defined by the set of rules and axioms below. Let T and T' be sets of formulas and ϕ, ψ and χ be formulas. We will define $\vdash$ as a relation between sets of formulas and formulas, but we will write $T, \phi \vdash \psi$, where more formally $T \cup \{\phi\} \vdash \psi$ is meant.

The rules for intuitionistic propositional logic are:

1. $\phi \in T \quad \Rightarrow \quad T \vdash \phi$
2. $T, \psi \vdash \phi$ and $T' \vdash \psi \quad \Rightarrow \quad T \cup T' \vdash \phi$
3. $T \vdash \phi$ and $T \vdash \psi \quad \Leftrightarrow \quad T \vdash \phi \wedge \psi$
4. $T, \phi \vdash \chi$ and $T, \psi \vdash \chi \quad \Leftrightarrow \quad T, \phi \vee \psi \vdash \chi$
5. $T, \phi \vdash \psi \quad \Leftrightarrow \quad T \vdash \phi{\rightarrow}\psi$
6. $\bot \in T \quad \Rightarrow \quad T \vdash \phi$

In the case of classical logic we add the axiom:

7. $\neg\neg\phi \vdash \phi$

In the case of the classical modal logic **K** we add also:

8. $\vdash \phi \quad \Rightarrow \quad \vdash \Box\phi$
9. $\Box(\phi\rightarrow\psi) \vdash \Box\phi\rightarrow\Box\psi$

For classical and intuitionistic propositional logic, alternative axioms and rules can also be found for example in [TD 82] and for (other) modal logics one may consult [HC 84].

A *fragment* is a sublanguage of a logic, obtained by restricting the set of atoms or the application of connectives (or both). In this thesis we will often restrict the language to a finite set of atoms $p_1, \ldots, p_n$. Let F and G be fragments of a logic L. Then G is called a *subfragment* of F, if every formula of G is a formula of F (which will be denoted as $G \subseteq F$). The *diagram*, $Diag(F)$, of a fragment F, is the set of equivalence classes in F ordered by $\vdash$.

Let $\langle W, \leq\rangle$ be an ordered set (more traditionally: a partially ordered set or poset). A subset $X \subseteq W$ will be called a *closed* subset of W if for all v and w in W, $v \in X$ and $v \leq w$ implies $w \in X$. The set of closed subsets of W will be denoted by $\mathcal{P}^*(W)$.

The Kripke model theory used here is fairly standard and can be found, for example, in [Benthem 83]. A *Kripke frame* is a tuple $\langle W, R\rangle$, with a *domain* W (the set of *worlds* or *nodes*) and $R \subseteq W^2$ a binary relation on W. The relation R is called an *accessibility relation*. If the accessibility relation is known to be irreflexive, we will often use $<$ for R. If R is reflexive, we will use $\leq$ and $l < k$ (or $k > l$) will be used as a shorthand for $l \leq k$ and $l \neq k$. lRk will sometimes also be written as $k\breve{R}l$. If k and l are nodes in W and kRl, then l is called a *successor* of k and k is called a *predecessor* of l. A node l is a *direct successor* of k, kR_1l (or $k <_1 l$), if $k \neq l$ and for all m such that kRm and mRl either $m = k$ or $m = l$. A node k is the *root* of a Kripke model K if k is the only node in K that has at most itself as a predecessor. A node k is a *terminal* node if k has at most itself as a successor. So a terminal node has only itself as a successor or no successors at all.

A *Kripke model* $K = \langle W, R, atom\rangle$ is a Kripke frame $\langle W, R\rangle$ with a valuation *atom*, mapping nodes of W to sets of propositional variables. As usual we will define $k \Vdash \phi$, the *forcing* of a formula ϕ by a node k in a Kripke model K.

The Kripke models defined above will be used in (classical) modal logic. For **CpL** and **IpL** we will define special classes of Kripke models.

1.2.0.1. Definition. *Let $K = \langle W, R, atom\rangle$ be a Kripke model.*
K is a **CpL** Kripke model *if R is the identity relation.*
K is an intuitionistic Kripke model **IpL** *Kripke model for short) if*

1. *R is reflexive, transitive and anti-symmetric;*
2. *atom is order preserving.*

We will use $atom^n$ for the restriction of *atom* to $\{p_1, \ldots, p_n\}$ (nowhere a $q \notin \{p_1, \ldots, p_n\}$ is forced). Note that $K = \langle W, R, atom^n\rangle$ is again a Kripke model, which will be called an *n-model*.

In the sequel we will write $k \in K$ to express that a world k is a node of Kripke model K, instead of the more pedantic $K = \langle W, R, atom\rangle$ and $k \in W$. If $K = \langle W, R, atom\rangle$ and $V \subseteq W$ then $L = \langle V, R{\restriction}V, atom{\restriction}V\rangle$ is called a *submodel*. The fact that L is a submodel of K will be written as $L \subseteq K$.

Let us first recall the truth definition of classical propositional logic, **CpL**, in terms of Kripke semantics.

1.2.0.2. DEFINITION. *Let $K = \langle W, R, atom\rangle$ be a Kripke model and $k \in K$. Define $k \Vdash \phi$, the node k forces the formula ϕ, inductively as:*

- $k \Vdash p \quad \Leftrightarrow \quad p \in atom(k)$ *(p atomic);*
- $k \Vdash \psi \wedge \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *and* $k \Vdash \chi$*;*
- $k \Vdash \psi \vee \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *or* $k \Vdash \chi$*;*
- $k \Vdash \psi{\rightarrow}\chi \quad \Leftrightarrow \quad k \nVdash \psi$ *or* $k \Vdash \chi$*;*
- $k \Vdash \neg\psi \quad \Leftrightarrow \quad k \nVdash \psi$ *(i.e not* $k \Vdash \psi$*).*

We will say that K models *ϕ ($K \Vdash \phi$) (or ϕ holds in K) if for all $k \in K$ it is true that $k \Vdash \phi$.*

To obtain the Kripke semantics for modal logic we need to add rules for the modal operators to the rules defined for **CpL**.

1.2.0.3. DEFINITION. *Let $K = \langle W, R, atom\rangle$ be a Kripke model and $k \in K$.*

- $k \Vdash \Box\psi \quad \Leftrightarrow \quad \forall l \in K(if\ kRl\ then\ l \Vdash \psi)$*;*
- $k \Vdash \Diamond\psi \quad \Leftrightarrow \quad \exists l \in K(kRl\ and\ l \Vdash \psi)$*;*

Again *K models ϕ* ($K \Vdash \phi$) (or ϕ holds in K) if for all $k \in K$ it is true that $k \Vdash \phi$.

Next we define the forcing relation on intuitionistic Kripke models. Note that in the Kripke semantics of **IpL** implication and negation have non-local behaviour, like the modal operators.

1.2.0.4. DEFINITION. *Let $K = \langle W, R, atom\rangle$ be an intuitionistic Kripke model and let $k \in K$. Define $k \Vdash \phi$, inductively as:*

- $k \Vdash p \quad \Leftrightarrow \quad p \in atom(k)$ *(p atomic);*
- $k \Vdash \psi \wedge \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *and* $k \Vdash \chi$*;*
- $k \Vdash \psi \vee \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *or* $k \Vdash \chi$*;*
- $k \Vdash \psi{\rightarrow}\chi \quad \Leftrightarrow \quad \forall l \in K(if\ kRl\ then\ l \nVdash \psi\ or\ l \Vdash \chi)$*;*
- $k \Vdash \neg\psi \quad \Leftrightarrow \quad \forall l \in K(if\ kRl\ then\ l \nVdash \psi)$*.*

And as above, *K models ϕ* ($K \Vdash \phi$) (or ϕ holds in K) if for all $k \in K$ it is true that $k \Vdash \phi$.

Let $\mathcal{M}$ be a class of Kripke models. A formula ψ is a *local $\mathcal{M}$-consequence* of a formula ϕ, $\phi \models_{\mathcal{M}} \psi$, if for every $K \in \mathcal{M}$ and every $k \in K$ such that $k \Vdash \phi$ it is true that $k \Vdash \psi$. A formula ψ is a *global $\mathcal{M}$-consequence* of a formula ϕ, $\phi \Vdash_{\mathcal{M}} \psi$, if for every $K \in \mathcal{M}$ such that $K \Vdash \phi$ it is true that $K \Vdash \psi$.

A propositional logic L is said to be *sound* for $\mathcal{M}$ if for all L-formulas ϕ and ψ: $\phi \vdash_L \psi$ implies $\phi \models_{\mathcal{M}} \psi$. L is said to be *complete* for $\mathcal{M}$, if for all L-formulas ϕ and

ψ such that $\phi \models_{\mathcal{M}} \psi$ it is true that $\phi \vdash_L \psi$. If we restrict our attention to formulas in a logic L with all atomic subformulas in the set $\{p_1, \ldots, p_n\}$, we obtain the fragment L^n. If L is sound and complete for a class of Kripke models $\mathcal{M}$, one can, usually with almost the same proof, obtain also a theorem stating that L^n is sound and complete for the n-models in $\mathcal{M}$ (i.e. L is n-complete for $\mathcal{M}$). The following well known soundness and completeness theorems are stated here as facts[3].

1.2.0.5. Facts.

1. **K** *is sound and complete for the class of finite Kripke models.*
2. **CpL** *is sound and complete for the class of finite* **CpL** *Kripke models.*
3. **IpL** *is sound and complete for the class of finite* **IpL** *Kripke models.*

For the proofs of these facts we refer to [HC 84] and [TD 88].

If $L \subseteq K$ then L is a *generated submodel* of K if the domain of L is a closed subset of K. As a notation for the generated submodel above a node we will use $\uparrow k = \{l \mid kRl\}$. For the smallest generated submodel including node k, we will use the notation $\underline{\uparrow} k = \{l \mid kRl \text{ or } l = k\}$. If the accessibility relation is known to be reflexive, then of course $\uparrow k = \underline{\uparrow} k$ for all nodes $k \in K$. Occasionally we will use $\downarrow k$ for the set of nodes below node k, hence $\downarrow k = \{l \mid lRk\}$.

1.2.0.6. Definition. *Let K be a Kripke model. The nodes $k_1, \ldots, k_n \in K$ form a* cycle *in K of length n if $k_n R k_1$ and $1 \leq i < n$ implies $k_i R k_{i+1}$.*

K is called anticyclic *if K contains no cycles of length more than 1.*

In a finite anticyclic Kripke model K, we define the *depth* of a node $k \in K$ as usual.

1.2.0.7. Definition. *If K is a finite anticyclic Kripke model and k is a node of K, then $\delta(k)$, the* depth *of k is defined as*

$$\delta(k) = \begin{cases} 0 & \text{if } kRl \text{ implies } k = l \\ \max\{\delta(l) \mid l \neq k \text{ and } kRl\} + 1 & \text{otherwise.} \end{cases}$$

Most of the models in this thesis will be Kripke models. The next definition however introduces a more abstract notion of model. This allows us to call a finite representation of a fragment a *model* even if it is not a Kripke model. As we will see in Chapter 3, not all exact models are exact Kripke models.

[3] If we designate a proposition as a fact, we will not give a proof, either because it can be found elsewhere, or because it is rather trivial.

1.2.0.8. DEFINITION. *A structure $M = \langle W, \preceq, \omega \rangle$ is called a* model *for fragment F if $\langle W, \preceq \rangle$ is an ordered set and $\omega : F \mapsto \mathcal{P}^*(W)$, such that for all formulas $\phi, \psi \in F$:*

$$\phi \vdash \psi \quad \Rightarrow \quad \omega(\phi) \subseteq \omega(\psi).$$

M is called a classical *model if $\preceq$ is the identity relation (and hence all subsets of W are closed: $\mathcal{P}^*(W) = \mathcal{P}(W)$).*

A model M is complete *for F if for all ϕ and ψ in F*

$$\omega(\phi) \subseteq \omega(\psi) \quad \Rightarrow \quad \phi \vdash \psi.$$

A model M is exact *for F if M is complete for F and ω is surjective.*

Note that the definition the valuation of these abstract models does not require recursion on the length of the formula ϕ. The valuation ω may be any kind of mapping from formulas into closed subsets of W, as long as ω is monotone in the order of derivability.

A Kripke model corresponds to a model $\langle W, \preceq, \omega \rangle$ in the sense of the definition above. In classical models the relation $\preceq$ will be the identity, and in intuitionistic models $\preceq$ coincides with R. In both cases the function $[\![\phi]\!] = \{k \in K \mid k \Vdash \phi\}$ will map formulas onto (closed) subsets of K, in such a way, that $\phi \vdash \psi \quad \Rightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!]$.

Suppose $M = \langle W, \preceq, \omega \rangle$ is a model for fragment F. For $w \in W$ and $\phi \in F$ we define $w \Vdash \phi$, in a natural way, by

$$w \Vdash \phi \quad \Leftrightarrow \quad w \in \omega(\phi).$$

Obviously if $\phi \vdash \psi$ and $w \Vdash \phi$ we may infer that $w \Vdash \psi$.

This definition includes models with $W \subseteq F$, $\preceq$ the restriction of $\dashv$, the converse of $\vdash$, to W and ω defined as $\omega(\phi) = \{\psi \in W \mid \psi \vdash \phi\}$.

Note that if $M = \langle W, \preceq, \omega \rangle$ is an exact model of fragment F, then

$$\langle \mathcal{P}^*(W), \subseteq \rangle \cong Diag(F).$$

As it is our intention to use Kripke semantics as a general framework for the semantics of **CpL**, **IpL** as well as modal logics, it may be worthwhile to define forcing of a formula by a node in a Kripke model with respect to a general language containing the languages of these logics.

For example, in our computer programs for calculating diagrams of fragments from exact models there is only a small difference between modal logic and intuitionistic logic, as will be pointed out in Chapter 2. The rules for calculating the set of worlds in the exact model that force a certain formula ϕ can easily be explained using the generalized language and its Kripke semantics.

This generalized language can be defined as a language of propositional modal logic, with the connectives $\Diamond, \sim, \wedge, \vee, \Rightarrow$ and constants $\bot$ and $\top$.

The definition of $k \Vdash \phi$, a node k forcing a formula ϕ, is as definition 1.2.0.2 and 1.2.0.3 for $\Diamond, \wedge$ and $\vee$. The definition below of forcing for $\sim$ and $\Rightarrow$ reveals that $\sim$ is the classical negation and $\Rightarrow$ the intuitionistic implication:

1.2.0.9. Definition.

1. $k \Vdash \psi \Rightarrow \chi \quad \Leftrightarrow \quad \forall l \in K(\mathit{if}\, kRl\ \mathit{then}\, l \nVdash \psi\ \mathit{or}\, l \Vdash \chi)$;
2. $k \Vdash \sim \psi \quad \Leftrightarrow \quad k \nVdash \psi$;

where $k \nVdash \psi$ *is shorthand for* not $k \Vdash \psi$.

We can turn our generalized language into the language of classical modal propositional logic, by removing $\Rightarrow$, defining $\neg$ as $\sim$ and defining $\phi{\rightarrow}\psi$ and $\Box$ as usual as $\neg\phi \vee \psi$ and $\neg\Diamond\neg$. For the language of **CpL** we have to remove $\Diamond$ and $\Box$ from the language of classical modal logic. Likewise we can define the language of **IpL** by removing $\Diamond$ and $\sim$ from the generalized language, defining $\rightarrow$ as $\Rightarrow$ and defining $\neg\phi = \phi{\rightarrow}\bot$.

Chapter 2

Semantic Types and Exact Models

2.1 Introduction

In this chapter we will introduce the notion of the *semantic type* of a world in a Kripke model and explain the relation between semantic types, type formulas and exact models. Within this framework we will restate some proofs of related classical theorems about **CpL**, **K** and **IpL**. In the next chapters we will use semantic types to obtain exact models of finite fragments of **IpL** and calculate axioms of interpreted theories of provability logic.

A semantic type, in some fragment F of modal or intuitionistic propositional logic, is an abstract representation of a node in a Kripke model. The idea is that the semantic type of a node k in a Kripke model contains exactly the information that determine which formulas are forced in k, i.e. if a node l has the same semantic type as k in F, then k and l force the same formulas in F. We will write $Th_F(k)$ for the F-theory of k, defined by $Th_F(k) = \{\phi \in F \mid k \Vdash \phi\}$. In analogy to the situation in model theory, cf. [CK 73], $Th_F(k)$ could be called the *type* of k (in the language of F). If F is finite and closed under $\wedge$, there is formula $\phi_F(k)$, unique up to equivalence, which is a conjunction of representatives of every equivalence class in $Th_F(k)$. Such a formula $\phi_F(k)$ is an axiom for $Th_F(k)$ and is written as $\phi_F(k) \equiv \bigwedge Th_F(k)$. Here we will adopt the terminology of modal logic and call the formula $\phi_F(k)$ (or more precisely its equivalence class) the *F-type* of k (or the *type formula* of k in F). In modal logic (especially in provability logic) types are also known as the *character* of k (in [Bernardi 75] and [GG 90] or as an *atom* in [Bellissima 84]). The term *type* was used in [Shavrukov 93].

For the relation between types and exact Kripke models, recall definition 1.2.0.8. A Kripke model K is an exact Kripke model for a fragment F if K has the following properties:

1. K is F-complete, i.e. for all $\phi, \psi \in F$:

$$\phi \vdash \psi \quad \Leftrightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!].$$

2. Every closed subset of K is F-definable, i.e. for all closed $X \subseteq K$ there is a $\phi \in F$ such that:
$$X = [\![\phi]\!].$$

Observe that by property 2, for every k in an exact Kripke model there is a type formula for k in F.

For a finite representation of the fragment F, the types in the exact Kripke model of F would be sufficient. Recall the generalized notion of model from definition 1.2.0.8. The set of types T (ordered by derivability) is an *exact model* if we add the mapping ω, defined by

$$\omega(\phi) = \{\psi \in T \mid \psi \vdash \phi\}.$$

If a fragment F has an exact model, the type formulas in an exact model for F can often be derived from some *normal form* for the formulas in F. But such an exact model with a set of formulas as its universe, is less useful for our purpose than an exact Kripke model. In the calculation of $\omega(\phi)$ in a general exact model, one uses the derivability relation, instead of deciding the derivability of ψ from ϕ by model checking, as in an exact Kripke model.

On the other hand, the general exact model almost gives us an exact Kripke model. We only have to construct a Kripke model K such that for each type ϕ in the general exact model there is a $k \in K$ such that ϕ is the type of k (and such that this mapping of types on worlds of K is 1–1). The core question for this step in the construction of an exact Kripke model is: 'which kind of world does realize type ϕ'.

The answer to this question is a semantic equivalent to the type formula of a node k, and will be called the *semantic type* of k in F. The semantic type is an abstract representation of the node, in such a way that identical semantic types in F are equivalent in F (i.e. have the same F-theory).

To represent the essential information about a node k in a Kripke model we have to know:

a. which atoms hold in k;
b. what happens in the successor nodes.

If we use $\tau_F(k)$ for the semantic type of k in K, the general format of a semantic type is

$$\tau_F(k) = \langle atom^n(k), T \rangle$$

where T is a set of semantic types of successors[1] of k in K.

Of course, the semantic types of F have to fulfill the condition:

$$\tau_F(k) = \tau_F(l) \quad \Leftrightarrow \quad Th_F(k) = Th_F(l)$$

[1] As we will see in Chapter 3, sometimes the information about a subset of the successors of k is sufficient.

for all $k \in K$ and $l \in L$ where K and L are Kripke models used in the semantics of F. If there is a type formula for each node k in F, a $\phi_F(k) \in F$ that is an axiom[2] for $Th_F(k)$, the condition above is equivalent to:

$$\tau_F(k) = \tau_F(l) \quad \Leftrightarrow \quad \phi_F(k) = \phi_F(l).$$

Of course the distinction between the semantic types in F and the $\phi_F(k)$ is just a matter of point of view.

Note that with the restrictions that apply to the fragment F, regarding the atoms used and the applicability of connectives, the information we need in the semantic type of k in K need not be a full description of the submodel of K generated by k. Otherwise we would not have gained much in switching from Kripke models to semantic types.

The approach in this thesis to the construction of the exact model of a fragment F will be to find a minimal set of semantic types that is *complete* for F. First we will define what kind of objects the semantic types for F are (in some class of Kripke models). Next we will define a set T of these semantic types such that for each ϕ and ψ in F with $\phi \nvdash \psi$ there is a type $t \in T$ available, such that if for a node k in a Kripke model K, $\tau_F(k) = t$, then $k \Vdash \phi$ and $k \nVdash \psi$.

If we prove T to be minimal, the construction of our exact Kripke model is almost complete because the semantic types usually carry in them a natural accessibility relation. However in modal logic this order relation is not unique, the semantic types may be ordered in various alternative ways to obtain an exact Kripke model.

Turning to intuitionistic propositional logic, a fragment F will have a finite exact model iff $Diag(F)$ is isomorphic to a set of closed subsets ordered by inclusion, as was pointed out in the preliminary section of the introduction. Hence, $Diag(F)$ is a finite distributive lattice. Let us use $\phi \oplus \psi$ for the representative of the equivalence class in $Diag(F)$ that is the join of the classes represented by ϕ and ψ. Note that if $\vee$ is one of the connectives of F then $\phi \oplus \psi \equiv \phi \vee \psi$.

2.1.0.1. Definition. *An equivalence class ϕ will be called* join-irreducible *(or*irreducible *for short) in $Diag(F)$ if ϕ is not the bottom element of $Diag(F)$ and for all $\psi, \chi \in F$ we have*

$$\phi \vdash \psi \oplus \chi \quad \Rightarrow \quad \phi \vdash \psi \text{ or } \phi \vdash \chi.$$

Let us denote the set of join-irreducible classes in $Diag(F)$ as $I(F)$. Then $I(F)$ may be regarded as an ordered set (with $\dashv$, the reverse of $\vdash$, as its order relation[3]) and hence the set of closed subsets $\mathcal{P}^*(I(F))$ is defined. According to Birkhoff's representation theorem $Diag(F)$ will be isomorphic to $\mathcal{P}^*(I(F))$ ordered by inclusion.

[2] As is the case if F is finite.

[3] This is more convenient in case we want to turn $I(F)$ into a Kripke model.

2.1.0.2. THEOREM. (Birkhoff) *Any finite distributive lattice is isomorphic to the lattice of the closed sets of its join-irreducible elements.*

Proof. A proof can be found in [DP 90]. ⊣

Hence $I(F)$ will be a set of *types* that can be used for an exact model. However in the intuitionistic case the order of the exact model, given by $\dashv$, will in general not be the identity relation. (In classical logic, where we deal with Boolean algebras instead of Heyting algebras, the different atoms of the algebra exclude each other: if ϕ and ψ are irreducible, then $\phi \vdash \psi \quad \Rightarrow \quad \phi \equiv \psi$.)

In intuitionistic propositional logic the general notion of exact model is closer to that of an exact Kripke model than in classical modal logic. It is not difficult to turn an exact model for F into a Kripke model by stipulating the valuation $atom(\phi) = \{p \text{ atomic} \mid \phi \vdash p\}$.

However, these notions do not coincide, as we cannot prove in general for the resulting Kripke model that the node corresponding to type ϕ does indeed force the formula ϕ. As we will see in Chapter 3 the fragment $[\vee, \neg]^n$ has for each n an exact model, which is, for $n > 1$, not an exact Kripke model.

As the order in an exact Kripke model of **IpL** is induced by the derivability relation, the exact Kripke model of a fragment F, if it exists, is unique (that is, isomorphic to the set of irreducibles $I(F)$ ordered by derivability).

If, as in the case of the $[\vee, \neg]^n$ fragments, an exact Kripke model is out of reach, but we do have a minimal complete set of semantic types for the fragment F at hand, we can at least try to find a minimal finite model realizing all of the types in this set. The resulting model is of course complete for the fragment F and will be called a *universal model* for F.

2.2 Preliminaries

In this section we will introduce some useful notations for semantic types, introduce the important notion of *bisimulation* between Kripke models and define a layered variant of this relation. These layered bisimulations are related to the *model equivalence* in [Fine 74] and play a major rôle in the semantics of fragments with restricted nesting of modal operators or restricted nesting of implication.

We will take the liberty of using R for the accessibility relation even in those cases where we are dealing with more than one model. Usually it is clear from the context which model the relation belongs to.

2.2.0.1. DEFINITION. *A relation S between two Kripke models K and L is said to be a* bisimulation *iff for all $k \in K$ and $l \in L$ such that kSl:*

1. $atom(k) = atom(l)$;
2. $\forall k' \breve{R} k \; \exists l' \breve{R} l \; (k' S l')$;
3. $\forall l' \breve{R} l \; \exists k' \breve{R} k \; (k' S l')$.

A bisimulation relation which is a function is called a p-morphism. *If a p-morphism from K to L is surjective, it is often called a* reduction *from K to L (also known as pseudo-epimorphism).*

We will use the notation $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow l$ to denote that k and l bisimulate each other, that is, there exists a non-empty bisimulation S such that kSl. That $\leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow$ is an equivalence relation between nodes in Kripke models is obvious.

2.2.0.2. THEOREM. (Bisimulation Theorem) *If $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow l$ and $k \Vdash \phi$ then $l \Vdash \phi$.*

Proof. For propositional formulas ϕ, both in modal logic and intuitionistic logic, the theorem is easily proved by induction on the length of ϕ. Note that we could use the general language from the general preliminaries to prove this theorem for both logics at once. ⊣

An *n-bisimulation* is a bisimulation between two n-models (and hence with the first condition of definition 2.2.0.1 changed into: $atom^n(k) = atom^n(l)$). If k and l n-bisimulate each other we will write $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow^n l$.

Spelling out the proof of the bisimulation theorem will reveal that it can easily be transformed into a proof that if all propositional variables of ϕ are in $\{p_1, \ldots, p_n\}$, then $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow^n l \;\Rightarrow\; k \Vdash \phi$ then $l \Vdash \phi$.

Layered bisimulations also known as *bounded bisimulations* or n, m-bisimulations, will be defined by induction on m.

2.2.0.3. DEFINITION. *A relation S between two Kripke models K and L is said to be an* $n, 0$-bisimulation *iff for all $k \in K$ and $l \in L$ such that kSl it is true that $atom^n(k) = atom^n(l)$.*

A relation S between two Kripke models K and L is said to be an $n, m+1$-bisimulation *iff there is a n, m-bisimulation S' such that, for all $k \in K$ and $l \in L$ with kSl,*

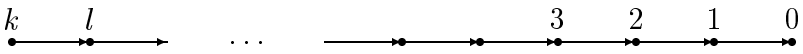
1. $atom^n(k) = atom^n(l)$;
2. $\forall k' \breve{R} k \; \exists l' \breve{R} l \; (k' S' l')$;
3. $\forall l' \breve{R} l \exists k' \breve{R} k \; (k' S' l)'$.

We will write $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow^n_m l$ if there exists an n, m-bisimulation between k and l.

In the section on modal logic in this chapter, we will prove an n, m-bisimulation theorem for fragments of modal logic with atomic formulas in $\{p_1, \ldots, p_n\}$ and modal degree less than or equal to m. In Chapter 4 a similar theorem is proved for $\mathbf{IpL}^n_m$, the fragment with atomic formulas in $\{p_1, \ldots, p_n\}$ and the nesting of implication bounded by m. Fragments with this kind of restriction on the nesting of one of the connectives are called *layered fragments*.

It will be clear from these definitions that $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow l$ implies $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow^n l$, which implies $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow^n_m l$ for each m. Our notation may suggest that $k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow^n l$ if $\forall m \; (k \leftrightarrows\!\!\!\!\!\!\rightarrow\!\!\!\!\!\!\leftarrow^n_m l)$. In general this is not true, as the following counter-example shows.

2.2.0.4. EXAMPLE. *In the Kripke model below the accessibility relation is irreflexive. At the right hand side of l there is a copy of the natural numbers in descending order. No atoms are forced in the nodes of this model. We have $k \overleftarrow{\mathcal{Z}}^n_m l$ for each n and m, but not $k \overleftarrow{\mathcal{Z}}^n l$.*



2. FIGURE. *A counter-example against* $\forall m\ (k \overleftarrow{\mathcal{Z}}^n_m l) \quad \Rightarrow \quad k \overleftarrow{\mathcal{Z}}^n l$.

In case k and l are nodes in finite Kripke models, $\forall m\ (k \overleftarrow{\mathcal{Z}}^n_m l)$ does imply $k \overleftarrow{\mathcal{Z}}^n l$.

2.2.0.5. DEFINITION. *A Kripke model K is called* locally finite *if for every node $k \in K$ the set $\uparrow k = \{l \in K \mid kRl\}$ is finite.*

2.2.0.6. THEOREM. *For nodes k and l in locally finite Kripke models:*

$$\forall m\ (k \overleftarrow{\mathcal{Z}}^n_m l) \quad \Leftrightarrow \quad k \overleftarrow{\mathcal{Z}}^n l.$$

Proof. Assume $k \overleftarrow{\mathcal{Z}}^n_m l$ for all m. We will prove that the relation S between the (finite) models of k and l defined as $k'Sl' \quad \Leftrightarrow \quad \forall m(k' \overleftarrow{\mathcal{Z}}^n_m l')$ is a bisimulation.

That $atom^n(k) = atom^n(l)$, is an immediate consequence of the definition of n, m-bisimulation. As the other two conditions for a bisimulation are symmetric, we only prove the first.

Suppose kRk' and let $l_1, \ldots, l_r$ be an enumeration of the successors of l. We will prove that there is an $i \leq r$ such that $k' \overleftarrow{\mathcal{Z}}^n_m l_i$ for all m.

For every $i \leq r$ such that not $\forall m\ (k' \overleftarrow{\mathcal{Z}}^n_m l_i)$ there is a least m, say m_i, with not $k' \overleftarrow{\mathcal{Z}}^n_{m_i} l_i$.

If not $\exists l' \breve{R} l\ \forall m\ (k' \overleftarrow{\mathcal{Z}}^n_m l')$, then let $M = \max\{m_i \mid m_i = \min\{m \mid \text{not } k' \overleftarrow{\mathcal{Z}}^n_m l_i\}\}$. Hence for no l_i it will be true that $k' \overleftarrow{\mathcal{Z}}^n_M l_i$, for it is easy to prove from the definition of n, m-bisimulation, that $k' \overleftarrow{\mathcal{Z}}^n_M l_i$ would imply $k' \overleftarrow{\mathcal{Z}}^n_m l_i$ for all $m \leq M$.

By assumption we know $k \overleftarrow{\mathcal{Z}}^n_{M+1} l$ and hence for some $l' \breve{R} l$ it should be true that $k' \overleftarrow{\mathcal{Z}}^n_M l'$. From this contradiction we infer that for some $l_i \breve{R} l$ it is true that $\forall m\ (k' \overleftarrow{\mathcal{Z}}^n_m l_i)$. $\dashv$

2.2.0.7. COROLLARY. *For nodes k and l in finite Kripke models:*

$$\forall m\ (k \overleftarrow{\mathcal{Z}}^n_m l) \quad \Leftrightarrow \quad k \overleftarrow{\mathcal{Z}}^n l.$$

As stated in the introduction of this chapter, the semantic type of a node k in a Kripke model K will in general be of the form $\tau(k) = \langle atom(k), T \rangle$, where T a set of types of successors of k. To point out the separate parts of a type we also define the projections $j_0(t)$ and $j_1(t)$ for a tuple t.

2.2.0.8. DEFINITION. *Let t be a tuple, $t = \langle P, Q\rangle$. Define $j_0(t) = P$ and $j_1(t) = Q$.*

In layered fragments (with restricted nesting of modal operators or implication) we will introduce hierarchies of types, $\{T_m \mid m \in I\!N\}$. If $t \in T_0$ then $j_1(t) = \emptyset$ holds, while for $t \in T_{m+1}$ we will have $j_1(t) \subseteq T_m$.

If $\tau_F(k) = t$, then k is said to *realize* the type t.

2.3 Types in CpL

The main point of introducing (semantic) types and exact models in classical propositional logic, **CpL**, is to illustrate the concepts defined in the introduction of this chapter. Some facts about **CpL** and its types that appear in this section are also useful in the next sections and chapters.

Recall that $\mathbf{CpL}^n$ is the fragment of **CpL** formulas of which the atomic subformulas belong to the set $\{p_1, \ldots, p_n\}$. By the n-completeness theorem a $\mathbf{CpL}^n$ formula ϕ is derivable in **CpL** iff ϕ is valid in all finite n-models K.

2.3.0.1. DEFINITION. *Let $Q \subseteq \{p_1, \ldots, p_n\}$ be a finite set of atoms. Define:*

$$\phi^n_Q = \bigwedge Q \wedge \bigwedge\{\neg q \mid q \in \{p_1, \ldots, p_n\} \setminus Q\}.$$

The definition of the formulas ϕ^n_Q will also be useful in later chapters.

2.3.0.2. DEFINITION. *The* type $\phi^n_{\mathbf{CpL}}(k)$ *of a world k in an n-model K is the formula $\phi^n_{atom^n(k)}$.*

Only in this section we will write $\phi^n(k)$ for $\phi^n_{\mathbf{CpL}}(k)$. In other fragments where the **CpL** type of a node is used it will be necessary to distinguish the type $\phi^n_{\mathbf{CpL}}(k)$ from the type $\phi^n(k)$ in the fragment at hand.

If k is a world in a **CpL** model, let us write $Th^n(k)$ for the $\mathbf{CpL}^n$ theory of k, defined by $Th^n(k) = \{\phi \in \mathbf{CpL}^n \mid k \Vdash \phi\}$.

2.3.0.3. LEMMA. *Let k be a node in an n-model and let $\phi^n(k)$ be the type of k in $\mathbf{CpL}^n$. Then*

1. *ϕ is an irreducible formula in $\mathbf{CpL}^n$ (see definition 2.1.0.1) iff for all formulas $\psi \in \mathbf{CpL}^n$:*
$$\phi \nvdash \psi \quad \Leftrightarrow \quad \phi \vdash \neg\psi.$$
2. *$\phi^n(k)$ is irreducible, i.e.:*
$$\forall \psi, \chi \in \mathbf{CpL}^n(\phi^n(k) \vdash \psi \vee \chi \quad \Rightarrow \quad \phi^n(k) \vdash \psi \text{ or } \phi^n(k) \vdash \chi).$$
3. *if a $\mathbf{CpL}^n$ formula ϕ is irreducible (in $\mathbf{CpL}^n$), then ϕ is equivalent to a type of $\mathbf{CpL}^n$;*
4. *the Lindenbaum algebra of $\mathbf{CpL}^n$ is a finite Boolean algebra with the types of $\mathbf{CpL}^n$ as its atoms;*

5. *if l is a node in an n-model K, then*

$$l \Vdash \phi^n(k) \quad \Leftrightarrow \quad atom(l) = atom(k).$$

6. *$\phi^n(k)$ is an axiom for $Th^n(k)$.*

Proof. 1: Let $\phi \in \mathbf{CpL}^n$ be irreducible. Then from $\phi \vdash \psi \vee \neg\psi$ infer that $\phi \vdash \psi$ or $\phi \vdash \neg\psi$. For the other direction, let $\phi \vdash \psi \vee \chi$. If $\phi \nvdash \psi$, then by assumption, $\phi \vdash \neg\psi$. Hence we would have $\phi \vdash \chi$.

2: With a simple induction on the length of formula $\psi \in \mathbf{CpL}^n$ prove that $\phi^n(k) \vdash \psi$ or $\phi^n(k) \vdash \neg\psi$.

3: Let $\phi \in \mathbf{CpL}^n$ be irreducible. According to definition 2.1.0.1 $\phi \nVdash \bot$. Hence, for some node k in a $\mathbf{CpL}$ n-model, we have $k \Vdash \phi$. Now note that obviously $k \Vdash \phi^n(k)$, hence $\phi^n(k) \nvdash \neg\phi$ and $\phi \nvdash \neg\phi^n(k)$. As both ϕ and $\phi^n(k)$ are irreducible, this implies $\phi \equiv \phi^n(k)$.

4: To prove $\phi^n(k)$ to be an atom in $Diag(\mathbf{CpL}^n)$, assume that $\phi \in \mathbf{CpL}^n$, $\phi \not\equiv \bot$ and $\phi \vdash \phi^n(k)$. As $\phi^n(k)$ is irreducible, use 1 to infer from $\phi^n(k) \nvdash \neg\phi$ that $\phi^n(k) \equiv \phi$.

5: By definition, if $atom^n(k) = atom^n(l)$, then $\phi^n(k) = \phi^n(l)$. For the other direction, assume that $l \Vdash \phi^n(k)$. Then both $\phi^n(k) \nvdash \neg\phi^n(l)$ and $\phi^n(l) \nvdash \neg\phi^n(k)$. From the irreducibility of $\phi^n(k)$ and $\phi^n(l)$ infer, with 2, that $\phi^n(k) \equiv \phi^n(l)$ and hence, by definition, $atom^n(k) = atom^n(l)$.

6: As $\phi^n(k)$ is irreducible, use 1 to prove that $k \Vdash \phi$ implies $\phi^n(k) \vdash \phi$. On the other hand, as $k \Vdash \phi^n(k)$, from $\phi^n(k) \vdash \phi$ infer that $\phi \in Th^n(k)$. ⊣

2.3.0.4. Corollary. *Every formula in $\mathbf{CpL}^n$ is equivalent to a disjunction of irreducible formulas in $\mathbf{Cpl}^n$.*

Proof. Obvious, as $Diag(\mathbf{CpL}^n)$ is a Boolean algebra with the irreducible formulas as its atoms. ⊣

2.3.0.5. Theorem. *Let A^n be the set of types (irreducible formulas) of $\mathbf{CpL}^n$. Then A^n is an exact model of $\mathbf{CpL}^n$.*

Proof. As every formula in $\mathbf{CpL}^n$ is equivalent to a disjunction of irreducible formulas, according to corollary 2.3.0.4, there is a unique correspondence between the subsets of A^n and the equivalence classes of $\mathbf{CpL}^n$. ⊣

According to lemma 2.3.0.3, the type of a world k in an n-model is determined by the set $atom^n(k)$.

2.3.0.6. Definition. *Let k be a node in a $\mathbf{CpL}$ model. Then $\tau^n(k)$, the* semantic type *of k in $\mathbf{CpL}^n$ is defined by*

$$\tau^n(k) = \langle atom^n(k), \emptyset \rangle.$$

The following fact justifies our choice for the definition of semantic type in $\mathbf{CpL}^n$. It is a simple consequence of the definition of semantic type in $\mathbf{CpL}^n$ and lemma 2.3.0.3.

2.3.0.7. FACT. *Let k and l be nodes in* **CpL** *models. Then*

$$\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad \phi^n(k) \equiv \phi^n(l) \quad \Leftrightarrow \quad Th^n(k) = Th^n(l).$$

For K a **CpL** model, let K^τ be the set of n-types in K. K^τ may be treated as a **CpL** model, with $atom^n(\tau^n(k)) = atom^n(k)$. According to the facts above the models K and K^τ force the same $\mathbf{CpL}^n$ formulas. Of course the application of this reduction to K^τ would yield K^τ itself and hence we call K^τ *n-irreducible*.
Let $Exm(\mathbf{CpL}^n)$ be the set of all $\mathbf{CpL}^n$ types, i.e.

$$Exm(\mathbf{CpL}^n) = \{\langle Q, \emptyset\rangle \mid Q \subseteq \{p_1, \ldots, p_n\}\}.$$

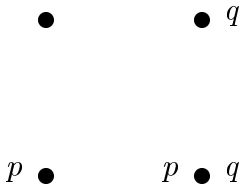
To make $Exm(\mathbf{CpL}^n)$ into a $\mathbf{CpL}^n$ Kripke model, use j_1 as the $atom^n$. If we use k_Q to denote the world in $Exm(\mathbf{CpL}^n)$ corresponding to the type $\langle Q, \emptyset\rangle$, then obviously $atom^n(k_Q) = Q$.

Clearly every k_Q corresponds to the type ϕ^n_Q defined above. Note that all subsets of $Exm(\mathbf{CpL}^n)$ are *closed*, as the accessibility relation is empty. Every subset $X \subseteq Exm(\mathbf{CpL}^n)$ corresponds to the disjunction of the ϕ^n_Q such that $k_Q \in X$, which proves that $Exm(\mathbf{CpL}^n)$ is an *exact Kripke model* of $\mathbf{CpL}^n$.

The following facts summarize these conclusions.

2.3.0.8. FACTS. *Let $Exm(\mathbf{CpL}^n)$ be the model defined above.*

1. *$Exm(\mathbf{CpL}^n)$ is (isomorphic to) the exact model of $\mathbf{CpL}^n$;*
2. *$Exm(\mathbf{CpL}^n)$ is an exact Kripke model of $\mathbf{CpL}^n$;*
3. *if K a* **CpL** *n-model that is an exact Kripke model of $\mathbf{CpL}^n$, then K is isomorphic to $Exm(\mathbf{CpL}^n)$;*
4. *$Exm(\mathbf{CpL}^n)$ has 2^n nodes and the Lindenbaum algebra of $\mathbf{CpL}^n$ has 2^{2^n} equivalence classes.*



3. FIGURE. *The exact Kripke model of $\mathbf{CpL}^2$.*

If K has the same set of worlds as $Exm(\mathbf{CpL}^n)$, but a non-empty accessibility relation (hence K is not a **CpL** Kripke model), then K is a universal model for $\mathbf{CpL}^n$. As every subset of nodes in K corresponds uniquely to an equivalence class in $\mathbf{CpL}^n$, K is also an exact Kripke model (where the set of 'closed subsets' in the

definition of exact model is taken to be the set of all subsets[4]). Hence there are (up to isomorphism) $2^{2^{2n}}$ exact Kripke models of $\mathbf{CpL}^n$.

Note that $Exm(\mathbf{CpL}^n)$ would not have been a model if we had restricted the definition of a $\mathbf{CpL}^n$ model to single worlds k (or singleton sets), as is usual.

2.4 Types in modal logic

In this section we will introduce fragments of modal logic with restricted nesting of the box operator. Our logical framework will be the system $\mathbf{K}$, the rules and axioms of which were given in the general preliminary section of the introduction.

Recall the standard definition of *modal depth* of a formula, also known as *modal degree*, which we here prefer to call the level of *box nesting* in analogy with the level of nesting of the implication in $\mathbf{IpL}$ that will be used later on.

2.4.0.1. Definition. *The level of* box nesting *of a* $\mathbf{K}$ *formula is denoted by the inductively defined function* $\beta(\phi)$*:*

p *atom*: $\beta(p) = 0$;
$\phi = \psi \circ \chi$: $\beta(\phi) = max\{\beta(\psi), \beta(\chi)\}$ *if* $\circ \in \{\wedge, \vee, \rightarrow\}$;
$\phi = \neg\psi$: $\beta(\phi) = \beta(\psi)$;
$\phi = \Box\psi$: $\beta(\phi) = \beta(\psi) + 1$.

The fragment $\mathbf{K}^n_m$ will be the fragment with $\{p_1, \ldots, p_n\}$ as its set of propositional variables and the nesting of the box operator restricted by the condition $\beta(\phi) \leq m$.

2.4.0.2. Fact. *The Lindenbaum algebra of* $\mathbf{K}^n_m$ *is a finite Boolean algebra and the Lindenbaum algebra of* $\mathbf{K}^n$ *is an infinite Boolean algebra.*

If L is an extension of $\mathbf{K}$, that is, if L can be derived by adding axioms to $\mathbf{K}$ and L^n_m is defined like $\mathbf{K}^n_m$ above, the above fact is also true for L.

As finite Boolean algebras are *atomic*, both the diagrams of $\mathbf{K}^n_m$ and of L^n_m are generated by their atoms. As in the case of $\mathbf{CpL}$, treated in the previous section, these atoms can be proved to be *irreducible* (see definition 2.1.0.1).

Clearly the set of irreducible formulas (or their equivalence classes to be precise) in $\mathbf{K}^n_m$ is an exact model according to definition 1.2.0.8. Every formula in $\mathbf{K}^n_m$ is equivalent to a disjunction of irreducible formulas and in this way we have a 1–1 correspondence between formulas and sets of irreducible formulas.

The Lindenbaum algebra of $\mathbf{K}^n$ is also an atomic Boolean algebra, but this is not the case for the L^n of arbitrary extensions L of $\mathbf{K}$ (see [Bellissima 84]). The set of irreducible formulas in $\mathbf{K}^n$ is not an exact model, as there are infinite sets of irreducibles that do not correspond to a formula in $\mathbf{K}^n$.

Define $Th^n_m(k) = \{\psi \in \mathbf{K}^n_m \mid k \Vdash \psi\}$. From the fact that $\mathbf{K}^n_m$ is finite, we may conclude that $Th^n_m(k)$ is a finite theory and hence we can define a formula $\phi^n_m(k)$ in $\mathbf{K}^n_m$ as $\phi^n_m(k) = \bigwedge Th^n_m(k)$.

[4]Recall that in classical (modal) Kripke models the order of the general model of definition 1.2.0.8 has nothing to do with the accessibility relation in the Kripke model.

This formula $\phi^n_m(k)$ will be recognized as the *type* of k in $\mathbf{K}^n_m$ and for every l in a Kripke model L we have $l \Vdash \phi^n_m(k)$ iff $Th^n_m(k) = Th^n_m(l)$.

We will give a more explicit definition of the types of $\mathbf{K}^n_m$ in the sequel. First we try to find the semantic types in $\mathbf{K}^n_m$ and a characterization of the exact Kripke model of $\mathbf{K}^n_m$. To do so we will rephrase a theorem in [Fine 74] using the layered bisimulations introduced in the general preliminaries.

2.4.0.3. DEFINITION. *Nodes k and l in Kripke models are called n,m-*equivalent*, $k \equiv^n_m l$, if for all $\phi \in \mathbf{K}^n_m$*

$$k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$$

(and hence $Th^n_m(k) = Th^n_m(l)$).

2.4.0.4. THEOREM. *Nodes k and l, in Kripke models K and L respectively, are n,m-equivalent iff $k \leftrightarrows^n_m l$.*

Proof. By induction on m. For $m = 0$ note that $k \leftrightarrows^n_0 l$ iff $atom^n(k) = atom^n(l)$, and that $Th^n_0(k)$ is the set of $\mathbf{CpL}^n$ formulas forced by k. Hence also: $Th^n_0(k) = Th^n_0(l) \quad \Leftrightarrow \quad atom^n(k) = atom^n(l)$.

Now assume the theorem proved for m. Let $k \leftrightarrows^n_{m+1} l$. We will prove $k \Vdash \phi$ to be equivalent with $l \Vdash \phi$ for all $\phi \in \mathbf{K}^n_{m+1}$ by showing $k \Vdash \phi$ implies $l \Vdash \phi$. We use induction on the length of ϕ.

The cases in which ϕ is atomic, a conjunction or a negation are obvious. So let $\phi = \Box\psi$ and $k \Vdash \Box\psi$. Note that as $\phi \in \mathbf{K}^n_{m+1}$ we know that $\psi \in \mathbf{K}^n_m$. Let $l_1 \in L$ be such that lRl_1. From $k \leftrightarrows^n_{m+1} l$ we infer that there is a $k_1 \in K$ such that kRk_1 and $k_1 \leftrightarrows^n_m l_1$. As $k_1 \Vdash \psi$, by our first induction hypothesis also $l_1 \Vdash \psi$. Which proves $l \Vdash \Box\psi$.

For the other direction, assume $k \equiv^n_{m+1} l$ and kRk_1. We have to prove the existence of an $l_1 \breve{R} l$ such that $l_1 \leftrightarrows^n_m k_1$, which, according to our induction hypothesis, is equivalent to $l_1 \equiv^n_m k_1$.

Now let $\phi^n_m(k_1)$ be the type of k_1 in $\mathbf{K}^n_m$ (as pointed out above, $\phi^n_m(k_1) = \bigwedge Th^n_m(k_1)$). As $k \Vdash \Diamond\phi^n_m(k_1)$ and $k \equiv^n_{m+1} l$ we will have also $l \Vdash \Diamond\phi^n_m(k_1)$ and for some $l_1 \breve{R} l$ it must be true that $l_1 \Vdash \phi^n_m(k_1)$. As observed above this implies that $k_1 \equiv^n_m l_1$.

By interchanging the rôles of k and l the n,m-bisimulation condition in the other direction is proved in the same way. ⊣

In [Fine 74] Fine only proved one direction of this theorem, i.e.

$$k \leftrightarrows^n_m l \quad \Rightarrow \quad Th^n_m(k) = Th^n_m(l).$$

Fine did not use layered bisimulation, but *m-equivalence*, a notion that is easily proved equivalent with our notion of n,m-bisimulation[5]. The analogy with results of

[5]That is k and l are m-equivalent according to the definition in [Fine 74], iff k and l n,m-bisimulate each other.

Fraïssé and Ehrenfeucht for first order theories, that was mentioned in Fine's article, will be taken up in the last section of this chapter.

As a simple corollary of theorem 2.4.0.4, for each n and m $k \leftrightarrows^n_m l$ will be equivalent to $k \equiv^n_m l$ (or $Th^n_m(k) = Th^n_m(l)$). In general $\forall m(k \equiv^n_m l)$ does not imply $k \leftrightarrows^n l$, as example 2.2.0.4 provides us with a counter-example.

The semantic types that we will define for $\mathbf{K}^n_m$ are quite natural characterizations of the equivalence classes of the n, m-bisimulations.

2.4.0.5. Definition. *Let k be a node in a finite n-model. Then the* semantic n, m-type *of k (in $\mathbf{K}$), $\tau^n_m(k)$, is defined by:*

- $\tau^n_0(k) = \langle atom^n(k), \emptyset \rangle$;
- $\tau^n_{m+1}(k) = \langle atom^n(k), \{ \tau^n_m(l) \mid kRl \} \rangle$.

The set of all semantic n, m-types $\tau^n_m(k)$ is written T^n_m.

This definition is justified by the following lemma.

2.4.0.6. Lemma. *If k, l are nodes in finite Kripke models then*

$$\tau^n_m(k) = \tau^n_m(l) \quad \Leftrightarrow \quad k \equiv^n_m l.$$

Proof. We will apply theorem 2.4.0.4 and prove $\tau^n_m(k) = \tau^n_m(l) \quad \Leftrightarrow \quad k \leftrightarrows^n_m l$. We will proceed by introducing a relation $\sim^n_m$ between K and L, defined as $k \sim^n_m l \quad \Leftrightarrow \quad \tau^n_m(k) = \tau^n_m(l)$, and prove $\sim^n_m$ to be an n, m-bisimulation, using induction on m.

By the definition of the semantic n, m-types, $k \sim^n_m l$ implies $atom^n(k) = atom^n(l)$. This proves the case that $m = 0$ and the first condition for an n, m-bisimulation in general. To prove the other conditions for an $n, m+1$-bisimulation, assume $k \sim^n_{m+1} l$ and kRk'. From $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$ it follows that $\tau^n_m(k') \in j_1(\tau^n_{m+1}(l))$ and hence there is an $l' \breve{R} l$ such that $\tau^n_m(k') = \tau^n_m(l')$. As by definition $\tau^n_m(k') = \tau^n_m(l') \quad \Leftrightarrow \quad k' \sim^n_m l$, this proves the first condition of n, m-bisimulation. The second condition is proved in the same way, interchanging the rôles of k and l and k' and l'.

For the proof of the other direction we will also use induction on m. The case $m = 0$ is again trivial, so suppose $k \leftrightarrows^n_{m+1} l$. Then obviously it will be true that $atom^n(k) = atom^n(l)$. To prove that also $j_1(\tau^n_{m+1}(k)) = j_1(\tau^n_{m+1}(l))$, let kRk'. By the definition of $\leftrightarrows^n_{m+1}$ there should be an $l' \breve{R} l$ such that $k' \leftrightarrows^n_m l'$. Using the induction hypothesis, it follows that $\tau^n_m(k') = \tau^n_m(l')$. Which proves $j_1(\tau^n_{m+1}(k)) \subseteq j_1(\tau^n_{m+1}(l))$. For the other direction of the inclusion, interchange the rôles of k and l. ⊣

In [Bellissima 84], $\phi^n_m(k)$, the $\mathbf{K}^n_m$ *type* of k (n, m-types for short), is defined as follows.[6] (Recall definition 2.3.0.1 for $\phi^n_{\mathbf{CpL}}(k)$).

[6] This definition is essentially the same as that of an $m, \vec{p}$-type in [Shavrukov 93].

2.4.0.7. DEFINITION. *Let k be a node in a finite n-model. Define $\phi^n_m(k)$, the $\mathbf{K}^n_m$* type *of k inductively as:*

- $\phi^n_0(k) = \phi^n_{\mathbf{CpL}}(k)$;
- $\phi^n_{m+1}(k) = \phi^n_{\mathbf{CpL}}(k) \wedge \bigwedge\{\Diamond\phi^n_m(l) \mid kRl\} \wedge \Box\bigvee\{\phi^n_m(l) \mid kRl\}$.

Let A^n_m be the set of (equivalence classes of) n, m-types.

2.4.0.8. FACT. *If k is a a node in a finite n-model, then for all m*

$$k \Vdash \phi^n_m(k).$$

The proof of this fact is obvious.

The next lemma shows that $\phi^n_m(k)$ indeed is an axiom for $Th^n_m(k)$ (using theorem 2.4.0.6) and hence is a type.

2.4.0.9. LEMMA. *Let k and l be nodes in finite n-models. If $k \Vdash \phi^n_m(l)$, then $\tau^n_m(k) = \tau^n_m(l)$.*

Proof. We will use induction on m. If $m = 0$, then $k \Vdash \phi^n_{\mathbf{CpL}}(l)$ implies $atom^n(k) = atom^n(l)$ and hence $\tau^n_0(k) = \tau^n_0(l)$. So assume $k \Vdash \phi^n_{m+1}(l)$. Then $k \Vdash \phi^n_{\mathbf{CpL}}(l)$ and we may infer that $atom^n(k) = atom^n(l)$. To prove that also $j_1(\tau^n_{m+1}(k)) = j_1(\tau^n_{m+1}(l))$, we show $j_1(\tau^n_{m+1}(k)) \subseteq j_1(\tau^n_{m+1}(l))$.

Let kRr and hence $\tau^n_m(r) \in j_1(\tau^n_{m+1}(k))$. From $k \Vdash \phi^n_{m+1}(l)$, by definition 2.4.0.7, infer that $r \Vdash \bigvee\{\phi^n_m(s) \mid lRs\}$. So, for some $s\breve{R}l$ we have $r \Vdash \phi^n_m(s)$ and, by the induction hypothesis, $\tau^n_m(r) = \tau^n_m(s)$. Which proves $\tau^n_m(r) \in j_1(\tau^n_{m+1}(l))$.

To prove $j_1(\tau^n_{m+1}(l)) \subseteq j_1(\tau^n_{m+1}(k))$, let lRs and hence $\tau^n_m(s) \in j_1(\tau^n_{m+1}(l))$. As $k \Vdash \phi^n_{m+1}(l)$, by definition 2.4.0.7, we may infer that $k \Vdash \Diamond\phi^n_m(s)$. Hence, for some $r\breve{R}k$, $r \Vdash \phi^n_m(s)$. By the induction hypothesis, this implies $\tau^n_m(r) = \tau^n_m(s)$, which proves $\tau^n(s) \in j_1(\tau^n_{m+1}(k))$. ⊣

The lemmas 2.4.0.6 and 2.4.0.9 combine into:

2.4.0.10. THEOREM. *The set T^n_m of semantic n, m-types corresponds exactly to the set A^n_m of types in $\mathbf{K}^n_m$, in the sense that:*

$$\forall l \in K(\, l \Vdash \phi^n_m(k) \quad \Leftrightarrow \quad \tau^n_m(l) = \tau^n_m(k)).$$

Lemma 2.4.0.10 immediately leads to:

2.4.0.11. COROLLARY. *If K is a Kripke model such that each n, m-type occurs exactly once in K, then the subsets of K correspond exactly to the equivalence classes of $\mathbf{K}^n_m$. That is, the following function $[\![.]\!]$ is an isomorphism:*

$$[\![\phi]\!] = \{k \in K \mid k \Vdash \phi\}.$$

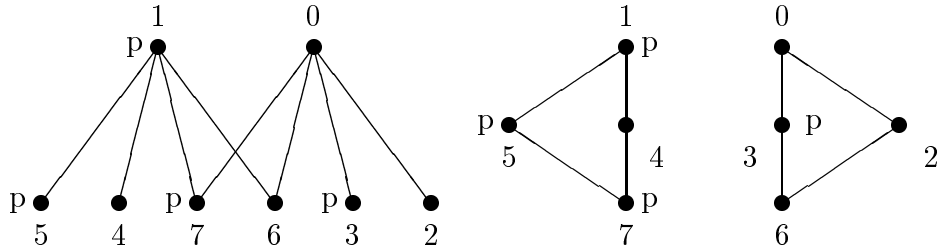
It can be proved that such an *exact model*, in which each subset corresponds to a formula and vice versa, exists for each $\mathbf{K}^n_m$.

2.4.0.12. THEOREM. *For each n and m there exists an exact Kripke-model K for $\mathbf{K}^n_m$, i.e., for each $U \subseteq K$, there is a formula ϕ in L^n_m such that $\{k \in K \mid k \Vdash \phi\} = U$, and K is n,m-complete, in the sense that for all $\phi, \psi \in L^n_m$, $\{k \in K \mid k \Vdash \phi\} = \{k \in K \mid k \Vdash \psi\}$ iff $\vdash \phi \leftrightarrow \psi$.*

Proof. We apply the so-called Henkin method to the (up to equivalence) finite set of formulas in $\mathbf{K}^n_m$, which is closed under taking subformulas. This gives one a Kripke-model with the maximal consistent sets as its worlds, with $\Gamma R \Delta$ defined by: for each $\Box\gamma \in \Gamma$, γ is an element of Δ and $\Gamma \Vdash p_i$ by: $p_i \in \Gamma$. The maximal consistent sets can be replaced by their conjunctions which are exactly the irreducible elements of $\mathbf{K}^n_m$. So a subset of the model will correspond to a disjunction of irreducibles, i.e. an arbitrary formula of $\mathbf{K}^n_m$. Obviously, non-equivalent formulas are forced on different subsets of the model. ⊣

The Henkin construction above also works for fragments L^n_m, where L is an extension of $\mathbf{K}$. The result in each case is called the *canonical exact model* . But the frame of the resulting model is not necessarily a frame for the logic L.

Unlike the exact models of fragments of intuitionistic propositional logic (see [JHR 91], [Hendriks 93]), not all the exact models of L^n_m are necessarily isomorphic.



4. FIGURE. *Two exact models for* $\mathbf{K}^1_1$.

The formulas in the exact models of $\mathbf{K}^1_1$:

0. $\neg p \wedge \Box\bot$	4. $\neg p \wedge \Diamond p \wedge \Box p$
1. $p \wedge \Box\bot$	5. $p \wedge \Diamond p \wedge \Box p$
2. $\neg p \wedge \Diamond\neg p \wedge \Box\neg p$	6. $\neg p \wedge \Diamond p \wedge \Diamond\neg p$
3. $p \wedge \Diamond\neg p \wedge \Box\neg p$	7. $p \wedge \Diamond p \wedge \Diamond\neg p$

The accessibility relation defined in a canonical exact model corresponds to the relation between irreducible elements α and β of L^n_m defined as:

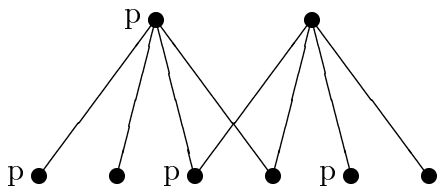
$$\alpha R \beta \quad \Leftrightarrow \quad \alpha, \Diamond\beta \nvdash \bot.$$

It is often possible to restrict this relation. For example in such a way that the Kripke exact model belongs to a certain subclass of the class of Kripke models (the reflexive models, well-founded models and so on). Note that in this way the completeness theorems for $\mathbf{K}$ and some of its extensions can be proved (see for example [HC 84]).

For the infinite fragment $\mathbf{K}^n$ there is no exact model in which all subsets determine a formula, but there is a (infinite) model which is n-complete. We will give the construction of such a model and call it ExK^n. Our ExK^n is comparable to the n-complete model given in [Grigolia 83] and [Rybakov 89] for provability logic[7].

2.4.0.13. Definition. *ExK^n with its R and $\Vdash$ is defined as the union of inductively defined ExK^n_m for $m \in \omega$.*

- $ExK^n_0 = \mathcal{P}(\{p_1, \cdots, p_n\})$, *the elements of ExK^n_0 are all R-incomparable, and $Q \Vdash p \Leftrightarrow p \in Q$;*
- $ExK^n_{m+1} = \{\langle Q, X\rangle \mid Q \subseteq \{p_1, \cdots, p_n\}, X \subseteq \bigcup_{i\leq m} ExK^n_i, X \cap ExK^n_m \neq \emptyset\}$, $\langle Q, X\rangle R\, Y \Leftrightarrow Y \in X$, *and* $\langle Q, X\rangle \Vdash p \Leftrightarrow p \in Q$;
- $ExK^n = \bigcup_{i\in\omega} ExK^n_i$.



5. Figure. *The model $ExK^1_0 \cup ExK^1_1$.*

From this picture it can be calculated that ExK^1_2 will have 504 nodes.
It is obvious from the construction that each n, m-type will be realized by some $k \in ExK^n$. This ensures the n-completeness of ExK^n.

The above definition is such that each node in ExK^n_i is the root of a finite reverse well-founded (and hence irreflexive) Kripke model.

2.4.0.14. Fact. **K** *is complete for finite reverse well-founded Kripke models.*

Let us use the completeness of **K** for finite and reverse well-founded Kripke models to define semantic types in **K**.

2.4.0.15. Definition. *For a node k in a finite reverse well-founded Kripke model define the semantic type in* **K***:*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid kRl\}\rangle.$$

As we are dealing with finite reverse well-founded models we may use $\delta(k)$, the depth of node k, to show that definition 2.4.0.15 is sound. Obviously $\tau^n(k) \notin j_1(\tau^n(k))$.

Observe that in ExK^n all semantic n-types of nodes in finite, reverse well-founded Kripke models are realized.

The definition of semantic types for $\mathbf{K}^n$ would not be of any use without the following lemma, stating its relation with bisimulation.

[7]Similar constructions for fragments in **K4Grz** and **S4** may be found in [Shehtman 78].

2.4.0.16. LEMMA. *If k and l are nodes in finite, irreflexive and reverse well-founded models, then:*

$$\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad k \leftrightarrows^n l.$$

Proof. Define $d = \max\{\delta(k), \delta(l)\}$. We will proceed by induction on d. In case $d = 0$, both k and l are terminal nodes, and then the lemma is trivial as both sides of the equivalence sign are equivalent to $atom^n(k) = atom^n(l)$.

So suppose $d > 0$. If $k \leftrightarrows^n l$ then trivially $atom^n(k) = atom^n(l)$. To prove $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$, assume kRk'. As k and l bisimulate each other there is an $l'\breve{R}l$ such that $k' \leftrightarrows l'$. The maximum of the depth of k' and l' is less than d and hence, by the induction hypothesis, $\tau^n(k') = \tau^n(l')$. So $\tau^n(k') \in j_1(\tau^n(l))$, which proves $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$. As the proof of $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$ is similar, we may conclude that $\tau^n(k) = \tau^n(l)$.

If $\tau^n(k) = \tau^n(l)$ then again trivially $atom^n(k) = atom^n(l)$. Assume kRk'. Then $\tau^n(k') \in j_1(\tau^n(l))$ and so there is an $l'\breve{R}l$ such that $\tau^n(l') = \tau^n(k')$. By applying the induction hypothesis infer that $k' \leftrightarrows^n l'$. As the other condition for bisimulation is proved likewise, we conclude that $k \leftrightarrows^n l$. ⊣

In general, to be able to construct a universal model (a *minimal* complete model) from the semantic types of nodes in M-models, there should not be too many models in M.

For example, $\mathbf{K}^n$ is complete for the class of finite n-models, but also for the subclass of finite reverse well-founded n-models. Assume that we would have defined a semantic type $\tau^n(k)$ for nodes k in finite n-models (in general), such that lemma 2.4.0.16 holds. Then clearly the set of these new semantic types would contain too many semantic types to be the universe of a *minimal* complete model for $\mathbf{K}^n$.

To prove that ExK^n is a universal model for $\mathbf{K}^n$ (and hence that the class of finite reverse well-founded models is small enough) we will define a type $\phi^n(k)$ (in $\mathbf{K}^n$) for every node in ExK^n, in such a way that $[\![\phi^n(k)]\!] = \{k\}$.

The definition of these types seems to belong to modal logic folklore (see for example [Bellissima 84]) and is very similar to the definition of the n, m-types above.

2.4.0.17. DEFINITION. *Let k be a node in a finite reverse well-founded Kripke model. Define $\phi^n(k)$, the* type *of k in $\mathbf{K}^n$, by:*

$$\phi^n(k) = \phi^n_{\mathbf{CpL}}(k) \wedge \{\Diamond\phi(l) \mid kRl\} \wedge \Box\bigvee\{\phi(l) \mid kRl\}.$$

Define $\bigwedge\emptyset = \bot$ and $\bigvee\emptyset = \top$, and observe that if k is a terminal node, $\phi^n(k) = \phi^n_{\mathbf{CpL}} \wedge \Box\bot$. That the types we defined for $\mathbf{K}^n$ correspond exactly to the semantic types of $\mathbf{K}^n$ is a corollary of the next theorem.

2.4.0.18. LEMMA. *If k and l are nodes in finite reverse well-founded Kripke models, then $k \Vdash \phi^n(l)$ implies $\tau^n(k) = \tau^n(l)$.*

Proof. We will use lemma 2.4.0.16 and prove $k \Vdash \phi^n(l)$ implies $k \leftrightarrows^n l$. We will use induction on $\delta(k)$, the depth of k. Note that, as $\phi^n(l)$ implies $\phi^n_{\mathbf{CpL}}(l)$, we may infer

that $atom^n(k) = atom^n(l)$. Now assume $k \Vdash \phi^n(l)$. In case $\delta(k) = 0$,we know that k is a terminal node and $k \Vdash \Box\bot$. Note that l will also be a terminal node. For lRl' would imply $k \Vdash \Diamond\phi^n(l')$ which would make $Th^n(k)$ inconsistent. For terminal nodes $atom^n(k) = atom^n(l)$ implies $k \leftrightarrows^n l$.

So let $\delta(k) > 0$. If $k'\breve{R}k$, then $\phi^n(l) \nvdash \Box\bot$ and hence $k' \Vdash \bigvee\phi^n(l_i)$ (where the l_i are the successors of l). Hence, for some $l'\breve{R}l$, $k' \Vdash \phi^n(l')$. Using the induction hypothesis we may conclude that $k' \leftrightarrows^n l'$.

Now let lRl'. Then $k \Vdash \Diamond\phi^n(l')$ and hence for some $k'\breve{R}k$ we have $k' \Vdash \phi^n(l')$. Again by the induction hypothesis we conclude $k' \leftrightarrows^n l'$. Which proves $k \leftrightarrows^n l$. ⊣

2.4.0.19. Theorem. *If k and l are nodes in finite reverse well-founded Kripke models, then*

$$k \Vdash \phi^n(l) \quad \Leftrightarrow \quad \tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) = Th^n(l).$$

Proof. By lemma 2.4.0.16 $\tau^n(k) = \tau^n(l)$ is equivalent with $k \leftrightarrows^n l$ and (by the bisimulation theorem) hence implies $Th^n(k) = Th^n(l)$. As $\phi^n(l) \in Th^n(l)$, from $Th^n(k) = Th^n(l)$ we may infer that $k \Vdash \phi^n(l)$. On the other hand, by lemma 2.4.0.18, $k \Vdash \phi^n(l)$ implies $\tau^n(k) = \tau^n(l)$. ⊣

2.5 Types and reductions in IpL

In the semantics of **IpL** we confine our attention mainly to finite, transitive, reflexive and anti-symmetric Kripke models (the finite **IpL** models).

2.5.0.1. Definition. *Let k be a node in a finite* **IpL** *model. The* semantic type *of k in* **IpL**, *$\tau^n(k)$, is defined by induction on $\delta(k)$, the depth of k.*

$$\begin{aligned}\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l \text{ and if } atom^n(k) = atom^n(l) \\ \text{then } \exists k' > k(\tau^n(k') \neq \tau^n(l) \wedge \tau^n(k') \notin j_1(\tau^n(l)))\}\rangle.\end{aligned}$$

Define the order of semantic types in **IpL** *as:*

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t).$$

Observe that, as a special case of this definition, we have $\tau^n(k) = \langle atom^n(k), \emptyset\rangle$ if $\delta(k) = 0$. Definition 2.5.0.1 is rather complex in comparison to definition 2.4.0.15, due to the fact that the accessibility relation is reflexive in this case.

2.5.0.2. Lemma. *Let k and l be nodes in a finite* **IpL** *model and $k < l$. Then*

1. *if $atom^n(k) \neq atom^n(l)$ then $\tau^n(l) \in j_1(\tau^n(k))$;*
2. *$j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$;*
3. *$\tau^n(k) \neq \tau^n(l) \quad \Leftrightarrow \quad \tau^n(l) \in j_1(\tau^n(k))$;*
4. *$\tau^n(k) \preceq \tau^n(l)$.*

Proof. 1: This is a simple consequence of definition 2.5.0.1.

2: Let $l < l'$ in such a way that $\tau^n(l') \in j_1(\tau^n(l))$. As obviously $k < l'$, if $atom^n(k) \neq atom^n(l')$ then $\tau^n(l') \in j_1(\tau^n(k))$. Now suppose that $atom^n(k) = atom^n(l')$. From $k < l < l'$ infer that $atom^n(l) = atom^n(l')$. From definition 2.5.0.1, infer that $\exists k' > l(\tau^n(k') \neq \tau^n(l') \wedge \tau^n(k') \notin j_1(\tau^n(l')))$. As $k < l$ also $\exists k' > k(\tau^n(k') \neq \tau^n(l') \wedge \tau^n(k') \notin j_1(\tau^n(l')))$ and from definition 2.5.0.1 infer that $\tau^n(l') \in j_1(\tau^n(k))$, which proves $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$.

3: Obviously $\tau^n(k) \notin j_1(\tau^n(k))$, from which the $\Leftarrow$ part follows trivially.

To prove the $\Rightarrow$ part, suppose that $\tau^n(l) \notin j_1(\tau^n(k))$. From the first part of the lemma we may conclude that $atom^n(k) = atom^n(l)$. According to definition 2.5.0.1, for every $k' > k$ it will be the case that $\tau^n(k') = \tau^n(l)$ or $\tau^n(k') \in j_1(\tau^n(l))$. Hence, if $k' > k$ and $\tau^n(k') \in j_1(\tau^n(k))$ then, by the assumption that $\tau^n(l) \notin j_1(\tau^n(k))$, $\tau^n(k') \neq \tau^n(l)$ and so we may conclude that $\tau^n(k') \in j_1(\tau^n(l))$. Which proves $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$. In combination with the second part of the lemma, we conclude that $j_1(\tau^n(k)) = j_1(\tau^n(l))$ and hence $\tau^n(k) = \tau^n(l)$.

4: Observe that from $\tau^n(k) \neq \tau^n(l) \Leftrightarrow \tau^n(l) \in j_1(\tau^n(k))$ we may infer that $\tau^n(k) = \tau^n(l)$ or $\tau^n(l) \in j_1(\tau^n(k))$ and hence $\tau^n(k) \preceq \tau^n(l)$. ⊣

To prove that the semantic types introduced above do indeed satisfy the condition that $\tau^n(k) = \tau^n(l)$ implies $Th^n(k) = Th^n(l)$, we will use a theorem stating in effect that the semantic types are equivalence classes for n-bisimulation.

2.5.0.3. Theorem. *For nodes k and l in finite* **IpL** *models, we have*

$$\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad k \leftrightarrows^n l.$$

Proof. $\Rightarrow$: We will prove that the relation $k \sim^n l$, defined as $\tau^n(k) = \tau^n(l)$, is an n-bisimulation. It is trivial that the first condition for bisimulation, $atom^n(k) = atom^n(l)$, will apply. As the two remaining conditions are symmetric, we will prove only the first.

Suppose we know that $\tau^n(k) = \tau^n(l)$ and $k \leq k'$. We have to show that there is an $l' \geq l$ such that $\tau^n(k') = \tau^n(l')$. In case we have $\tau^n(k') = \tau^n(k)$ of course $l' = l$ will do. So assume that $\tau^n(k') \neq \tau^n(k)$. Using lemma 2.5.0.2, infer that $\tau^n(k') \in j_1(\tau^n(k))$ and hence, as $\tau^n(k) = \tau^n(l)$, $\tau^n(k') \in j_1(\tau^n(l))$. So, for some $l' > l$ we have $\tau^n(k') = \tau^n(l')$.

$\Leftarrow$: Let $k \leftrightarrows^n l$ and define $d = \max\{\delta(k), \delta(l)\}$. With induction on d we will prove $\tau^n(k) = \tau^n(l)$. Note that from $k \leftrightarrows^n l$ we may infer that $atom^n(k) = atom^n(l)$. We will prove $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$. The proof of $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$ is essentially the same, interchanging the rôles of k and l. As $atom^n(k) = atom^n(l)$, we may conclude that $\tau^n(k) = \tau^n(l)$.

Suppose that $k < k_1$ and $\tau^n(k_1) \in j_1(\tau^n(k))$. As $k \leftrightarrows^n l$, there is a $l_1 \geq l$ with $k_1 \leftrightarrows^n l_1$. Assume that $\tau^n(l_1) = \tau^n(l)$. Using the, already proved, first part of the theorem, then $l \leftrightarrows^n l_1$ and hence also $k_1 \leftrightarrows^n k$. From $k_1 \leftrightarrows^n k$ we may infer that $atom^n(k_1) = atom^n(k)$. As $\tau^n(k_1) \in j_1(\tau^n(k))$, there is, according to definition 2.5.0.1, a $k_2 > k$ such that $\tau^n(k_2) \neq \tau^n(k_1)$ and $\tau^n(k_2) \notin j_1(\tau^n(k_1))$. The fact that $k_1 \leftrightarrows^n k$ implies that there is a $k_3 \geq k_1$ with $k_3 \leftrightarrows^n k_2$. Note that both

$k < k_2$ and $k < k_3$. Hence by the induction hypothesis we have $\tau^n(k_2) = \tau^n(k_3)$. So, $\tau^n(k_3) \neq \tau^n(k_1)$ and $\tau^n(k_3) \notin j_1(\tau^n(k_1))$, contradicting lemma reflemt4.3 applied to $k_1 \leq k_3$.

From this contradiction we infer that $\tau^n(l_1) \neq \tau^n(l)$. As $l \leq l_1$, again by lemma 2.5.0.2.3, we conclude that $\tau^n(l_1) \in j_1(\tau^n(l))$. Hence, we have $l < l_1$ and $k < k_1$. By the induction hypothesis we infer from $k_1 \leftrightarrows^n l_1$ that $\tau^n(k_1) = \tau^n(l_1)$ and so $\tau^n(k_1) \in j_1(\tau^n(l))$, what had to proved. ⊣

2.5.0.4. Corollary. *Let k be a node in a finite* **IpL** *model and $k < l$. Then*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l \wedge \neg(k \leftrightarrows^n l)\}\rangle.$$

Proof. We prove that for $k < l$ we have $\tau^n(l) \in j_1(\tau^n(k))$ iff $\neg(k \leftrightarrows^n l)$. By lemma 2.5.0.2.3, If $k < l$, then $\tau^n(l) \in j_1(\tau^n(k))$ is equivalent to $\tau^n(k) \neq \tau^n(l)$. Now apply theorem 2.5.0.3. ⊣

Let us write $Th^n(k)$ for the $\mathbf{IpL}^n$ theory of a node k in a finite **IpL** model. Hence, $Th^n(k) = \{\phi \in \mathbf{IpL}^n \mid k \Vdash \phi\}$.

2.5.0.5. Lemma. *Let k and l be nodes in finite* **IpL** *models. If $\tau^n(k) \preceq \tau^n(l)$ then $Th^n(k) \subseteq Th^n(l)$.*

Proof. Let $\tau^n(k) \preceq \tau^n(l)$. If $\tau^n(k) = \tau^n(l)$, then by the bisimulation theorem, theorem 2.2.0.2, $Th^n(k) = Th^n(l)$. On the other hand, if $\tau^n(k) \neq \tau^n(l)$ then there is a $k' > k$ such that $\tau^n(k') = \tau^n(l)$ and hence $Th^n(k') = Th^n(l)$. In an **IpL** model, from $k' > k$ infer $Th^n(k) \subseteq Th^n(k')$. ⊣

By ordering the semantic types in an **IpL** model K we will construct a new model, K^τ, a *maximal reduction*[8] of K.

2.5.0.6. Definition. *Let K be a finite* **IpL** *model. Define K^τ,* the maximal reduction *of K, by:*

$$K^\tau = \langle \{\tau^n(k) \mid k \in K\}, \preceq, j_0\rangle.$$

If K and K^τ are isomorphic, K is called an irreducible *model.*

The proofs of the following facts are straightforward.

2.5.0.7. Facts. *Let K be a finite* **IpL** *Kripke model.*

1. *K^τ is a Kripke model (note that $atom^n(\tau^n(k)) = j_0(\tau^n(k))$);*
2. *τ^n is a reduction from K to K^τ;*
3. *K^τ is irreducible.*

[8]The reader familiar with [Hendriks 93] will recognise the analogy with the γ-reduction introduced there.

As in modal logic, the semantic types in $\mathbf{IpL}^n$ correspond to formula types. The definition of the n-type of a node k in a finite (**IpL**) Kripke model K is a result of a theorem of de Jongh (in [De Jongh 68], [De Jongh 70] and [JC 95]).

2.5.0.8. Definition. *Let k be a node in a finite irreducible* **IpL** *model. Define both $\theta^n(k)$ and $\chi^n(k)$ inductively over $\delta(k)$, the depth of k.*
Let

1. $Newatom^n(k) = \{q \in \{p_1, \ldots, p_n\} \mid k \nVdash q \text{ and } \forall l > k(l \Vdash q)\}$,
2. $\Psi^n(k) = \bigvee\{\chi^n(l) \mid k <_1 l\}$,
3. $\Phi^n(k) = \bigvee\{\theta^n(l) \mid k <_1 l\}$.

Then for

$\delta(k) = 0$: $\quad \theta^n(k) = \phi^n_{CpL}(k)$; $\chi^n(k) = \neg\theta^n(k)$,
$\delta(k) > 0$: $\quad \theta^n(k) = \bigwedge atom^n(k) \wedge (\bigvee Newatom^n(k) \vee \Psi^n(k) \rightarrow \Phi^n(k))$;
$\qquad\qquad\quad \chi^n(k) = \theta^n(k) \rightarrow \Phi^n(k)$.

2.5.0.9. Theorem. (Jankov/De Jongh) *If k and l are nodes in irreducible finite* **IpL** *n-models then:*

1. $l \Vdash \theta^n(k) \quad \Leftrightarrow \quad k \leq l$,
2. $l \nVdash \chi^n(k) \quad \Leftrightarrow \quad l \leq k$.

Proof. We will prove 1 and 2 simultaneously by induction on the depth of k. In case $\delta(k) = 0$, both 1 and 2 are obvious. Assume the lemma for $\delta(k) \leq m$ and let $\delta(k) = m + 1$.

1. $\Rightarrow$: Let $l \Vdash \theta^n(k)$. If $l \Vdash \Phi^n(k)$ then for some h such that $k \leq_1 h$ we have $l \Vdash \theta^n(h)$. By the induction hypothesis this would imply $k \leq_1 h \leq l$ and hence $k \leq l$.

On the other hand, we will show that $l \nVdash \Phi^n(k)$ implies $k = l$. From $l \Vdash \theta^n(k)$ we may conclude that $l \Vdash \bigwedge atom^n(k)$. As $l \nVdash \Phi^n(k)$, we also may conclude $l \nVdash \bigvee Newatom^n(k)$ and $l \nVdash \Psi^n(k)$. By the induction hypothesis we infer that $l \leq h$ for all h such that $k \leq_1 h$. So if $q \in atom^n(l) \setminus atom^n(k)$, then we would have $q \in Newatom^n(k)$, contradicting $l \nVdash \bigvee Newatom^n(k)$. Hence $atom^n(l) = atom^n(k)$.

To prove that l also has the same successors as k, let g have a minimal depth such that $l \leq g$ and for all h with $k \leq_1 h$, $h \not\leq h$. From the induction hypothesis it follows that $g \nVdash \Phi^n(k)$. As g is a successsor of l, we have also $g \Vdash \theta^n(k)$. In the same way as we proved for l, we may prove for g that $atom^n(g) = atom^n(k)$ and $g \leq h$ for all h such that $k \leq_1 h$. For g there is no proper successor g' which is not a successor of k. Otherwise g' would be a successor of l with $\delta(g') < \delta(g)$, $l \leq g$ and for all h with $k \leq_1 h$, $h \not\leq h$, contradicting the minimality of (the depth) of g. From the irreducibility of the model conclude $g = k$ and hence $l \leq g$ implies $k \leq g$. Again by the irreducibility of the model infer $k = l$.

1. $\Leftarrow$: We first prove that $k \Vdash \theta^n(k)$. As obviously $k \Vdash \bigwedge atom^n(k)$ we still have to prove $k \Vdash \bigvee Newatom^n(k) \vee \Psi^n(k) \rightarrow \Phi^n(k)$. Observe that if $k <_1 h$ then by our induction hypothesis $k \nVdash \chi^n(h)$. Hence we may conclude that $k \nVdash \Psi^n(k)$. Of course, by the definition of $Newatom^n(k)$, also $k \nVdash \bigvee Newatom^n(k)$. So, if we assume $k \leq g$ with $g \Vdash \bigvee Newatom^n(k) \vee \Psi^n(k)$, then $k < g$. Hence, $g \Vdash \theta^n(h)$ for some h such

that $k <_1 h$ and hence $g \Vdash \Phi^n(k)$ So infer that $k \Vdash \theta^n(k)$ and apply lemma 2.5.0.5 to conclude that if $k \leq l$, then $l \Vdash \theta^n(k)$.

2. $\Rightarrow$: Assume that $l \nVdash \chi^n(k)$. Then for some g such that $l \leq g$, $g \Vdash \theta^n(k)$ and $g \nVdash \Phi^n(k)$. From the first part of this proof infer that $k \leq g$ and $h \not\leq g$ for all h such that $k <_1 h$. Hence conclude that $= k$, which proves $l \leq k$.

2. $\Leftarrow$: As k is a node in an irreducible model, we have $\tau^n(k) \neq \tau^n(h)$ for all h such that $k <_1 h$ and, by induction hypothesis, $k \nVdash \theta^n(h)$. Hence, as $k \Vdash \theta^n(k)$ it should be true that $k \nVdash \Psi^n(k)$. A fortiori, for $l \leq k$ it is true that $l \nVdash \chi^n(k)$. $\dashv$

Recall that τ^n is a reduction from K to K^τ, as defined in 2.5.0.6.

2.5.0.10. Definition. *For k a node in a finite* **IpL** *model define*

$$\phi^n(k) = \theta^n(\tau^n(k))$$

and

$$\psi^n(k) = \chi^n(\tau^n(k)).$$

2.5.0.11. Theorem. *If k and l are nodes in finite* **IpL** *n-models then:*

1. $l \Vdash \phi^n(k) \quad \Leftrightarrow \quad \tau^n(k) \preceq \tau^n(l)$,
2. $l \nVdash \psi^n(k) \quad \Leftrightarrow \quad \tau^n(l) \preceq \tau^n(k)$.

Proof. Assume $k \in K$ and $l \in L$ and let, in K^τ, $k' = \tau^n(k)$ and $l' = \tau^n(l)$. Use lemma 2.5.0.5 to conclude that $Th^n(k) = Th^n(k')$ and $Th^n(l) = Th^n(l')$.
1: Observe that $l \Vdash \phi^n(k) \quad \Leftrightarrow \quad l' \Vdash \phi^n(k')$ and hence $l \Vdash \phi^n(k) \quad \Leftrightarrow \quad k' \leq l'$.
2: Likewise, from $l \nVdash \psi^n(k) \quad \Leftrightarrow \quad l' \nVdash \psi^n(k')$ conclude $l \nVdash \psi^n(k) \quad \Leftrightarrow \quad l' \leq k'$. $\dashv$

2.5.0.12. Corollary. *If k is a node in a finite n-model and ψ is an* **IpL***-formula, then:*

$$k \Vdash \psi \quad \Leftrightarrow \quad \phi^n(k) \vdash \psi.$$

Let $K^\phi = \langle \{\phi^n(k) \mid k \in K\}, \vdash \rangle$, then it is easily verified that K^ϕ, with the obvious valuation $\phi^n(k) \Vdash p \quad \Leftrightarrow \quad \phi^n(k) \vdash p$, is a Kripke model. Recall that by definition 2.5.0.6, K^τ is the maximal reduction of K. Now we are ready to state another important (and easy to prove) corollary from theorem 2.5.0.9.

2.5.0.13. Corollary. *The model K^τ is isomorphic to the model K^ϕ.*

Readers familiar with [Jankov 68] may wonder why Jankov's name has been connected to theorem 2.5.0.9. The following corollary about finite *frames* presents what is usualy known as Jankov's theorem.

Let us call a reflexive, transitive and anti-symmetric frame an **IpL** frame for short. To define bisimulation between frames, we simply use definition 2.2.0.1 leaving out the condition on the atoms forced. Likewise we may define $k \leftrightarrows l$ between nodes k and l in frames in the obvious way.

2.5.0.14. COROLLARY. *For every finite rooted* **IpL** *frame* $\uparrow k$ *there is a formula* ψ_k *such that for any finite* **IpL** *frame* F *we have:* $F \nVdash \psi_k$ *iff for some* $l \in F$ *it is true that* $k \overleftrightarrows l$.

Proof. Define a valuation on $\uparrow k$ on the set of atoms $\{p_i \mid 1 \leq i \leq |\uparrow k|\}$, in such a way that there is a 1–1 mapping $\sigma : \uparrow k \mapsto \{p_i \mid 1 \leq i \leq |\uparrow k|\}$ and for $l \in \uparrow k$ we have $l \Vdash \sigma(m) \quad \Leftrightarrow \quad l \leq m$. For the formula ψ_k in the corollary take $\psi^n(k)$ (assuming $|\uparrow k| = n$). If $F \nVdash \psi^n(k)$ then for some model K based on F we will have for some $l' \in F$ that $l' \nVdash \psi^n(k)$. Now apply the theorem to infer that for some $l \leq l'$ we will have (in K) that $\tau^n(k) = \tau^n(l)$ and hence, by theorem 2.5.0.3, also $k \overleftrightarrows l$. $\dashv$

The Jankov theorem was independently proved by De Jongh in his dissertation [De Jongh 68]. In modal logic Fine in [Fine 85] introduced *subframe formulas* for finite transitive frames for which he proved the modal analogue of the Jankov theorem, apparently without being aware of theorems in intuitionistic propositional logic proved by Jankov and De Jongh.

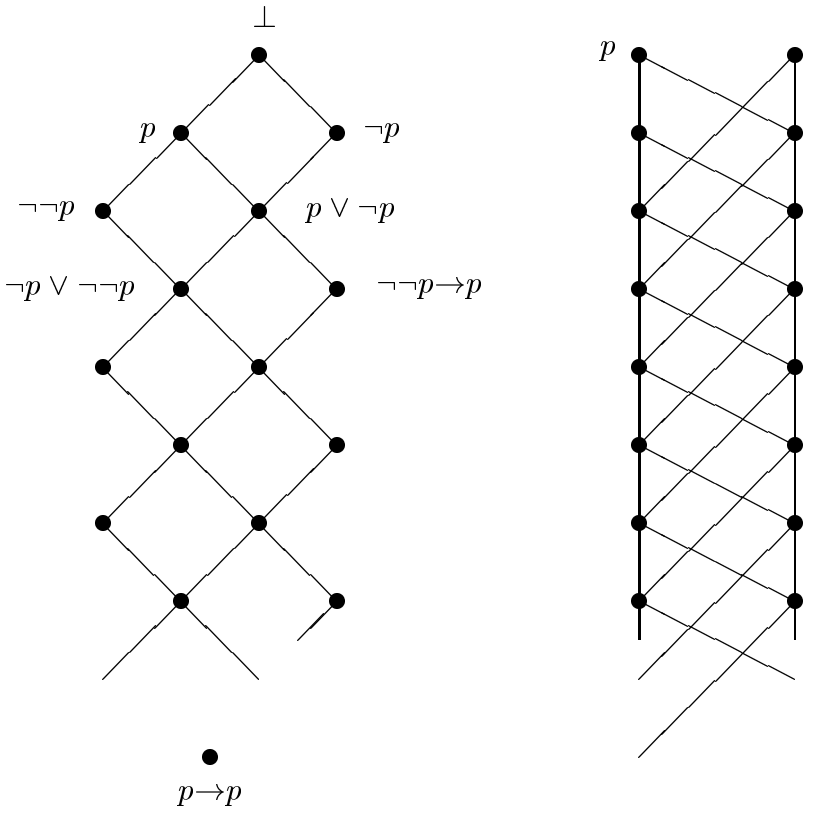
Obviously there are infinitely many types (and semantic types) in fragments $\mathbf{IpL}^n$ if $n \geq 1$.

The diagram of the fragment $\mathbf{IpL}^1$, see figure 6, is known as the Rieger-Nishimura lattice (see [Nishimura 60]) and the ordered set of all non-derivable irreducible types is the exact Kripke model of this fragment (the set of all elements will correspond to $\top$).

Note that all semantic types in $\mathbf{IpL}^1$ are realized in this model. Hence it will be complete for $\mathbf{IpL}^1$. As every semantic type corresponds to an irreducible formula (its type) every finite closed set of irreducible formulas corresponds to an $\mathbf{IpL}^1$ formula (the disjunction of the set of irreducibles). As the set of all elements is the only infinite closed subset of the model and is assigned to $\top$, this proves the model to be the exact model of $\mathbf{IpL}^1$.

For $n > 1$ the fragment $\mathbf{IpL}^n$ will not have an exact model as the diagram of $\mathbf{IpL}^n$ is not a complete distributive lattice for $n > 1$. For example, let $\{\phi_n(p) \mid n \in \mathbb{N}\}$ be a set of representatives of the irreducible equivalence classes in $\mathbf{IpL}^1$ and q an atomic formula. Then $\{q \wedge \phi^n(p) \mid \quad \nvdash \phi_n(p)\}$ is a closed set of irreducible formulas, that does not correspond to a formula in $\mathbf{IpL}^2$.

2.5.0.15. FACT. *For* $n > 1$ *the fragment* IpL^n *does not have an exact model.*



6. FIGURE. *The diagram*[9] *of* **IpL**[1] *(left) and its exact Kripke model (right).*

2.6 Calculations in exact models

As explained in the introduction of this chapter and illustrated by the examples of exact models in the previous sections, the proof of the construction of an exact Kripke model K for some fragment F is accompanied by a mapping $[\![.]\!]$ of formulas in F to (closed) subsets of K. A finite exact Kripke model K and its mapping $[\![.]\!]$ together provide us with a decision method for formulas in F. The restriction to closed subsets is necessary only in the case of fragments of **IpL**. In dealing with classical propositional logics (**CpL** or modal systems extending **K**) all subsets of K will be considered to be closed. So in topological terms, in classical logic we use the discrete topology and in intuitionistic logic the topology of upwardly closed subsets induced by the order of K. Recall the definition of the interior operation (rephrased in the context of Kripke models):

2.6.0.1. DEFINITION. *Let K be a Kripke model and $X \subseteq K$. Then X°, the* interior *of X is defined as:*

$$X^\circ = \bigcup\{Y \subseteq X \mid Y \text{ is closed}\}.$$

[9] Or its dual, according to our definition of $Diag(F)$ in Chapter 1.

In addition the following definition turns out to be useful.

2.6.0.2. DEFINITION. *Let K be a Kripke model and $X \subseteq K$. Then $X^\bullet$, the* predecessor set *of X is defined as:*

$$X^\bullet = \{k \in K \mid \exists l \in X(kRl)\}.$$

It is easily verified that, writing $\overline{X}$ for the complement of set X, in **IpL** models the interior can be calculated as:

$$X^\circ = X \setminus \overline{X}^\bullet.$$

2.6.0.3. FACTS. *Let K be a finite Kripke model for fragment F and let $[\![\phi]\!] = \{k \in K \mid k \Vdash \phi\}$. For all formulas ϕ and ψ of F (and as far as the connectives are applicable in F):*

1. $[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!]$;
2. $[\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!]$;
3. $[\![\phi{\rightarrow}\psi]\!] = ((K \setminus [\![\phi]\!]) \cup [\![\psi]\!])^\circ$;
4. $[\![\neg\phi]\!] = (K \setminus [\![\phi]\!])^\circ$;
5. $[\![\Diamond\phi]\!] = [\![\phi]\!]^\bullet$;
6. $\vdash \phi \quad \Rightarrow \quad [\![\phi]\!] = K$;
7. $\phi \vdash \psi \quad \Rightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!]$.

All of these facts can be proved by writing out the definitions and using well-known facts about Kripke semantics.

In case K is an exact Kripke model for the fragment F, the implications in the last two facts above can be changed into equivalences.

Hence $[\![\phi]\!]$ can be calculated using set theoretic and topological operations on the exact model.

A computer program to calculate $[\![\phi]\!]$ on an exact Kripke model will need the relevant information about the exact model to calculate the set operations and the predecessor sets. Clearly this can be done in linear time, which makes testing of formulas using exact models such an efficient decision procedure.

For exact models of the fragments of $\mathbf{CpL}^n$ we do not need predecessor sets and the calculation of $[\![\phi]\!]$ is very much like constructing a truth table for ϕ.

The testing of formulas by calculations in an exact Kripke model of a fragment F can be used to calculate the diagram of F and all its subfragments. Let G be a subfragment of F and let K_F be a finite exact model of F. To calculate the diagram of G the algorithm *mkDiag* is given the $[\![p]\!]$ of all atomic formulas in G. These atomic formulas are taken as the representatives of their equivalence classes and the start of a list of elements of the diagram to be constructed. From this list (of formulas representing equivalence classes already found) the algorithm systematically picks one or two representatives to make a new formula according to the connectives available in G. Such a new formula ϕ is taken as a candidate representative of a class not yet in the list. Using the rules explained above $[\![\phi]\!]$ is calculated and compared with the sets corresponding to the classes already found. If ϕ does represent an equivalence class not yet in the list, ϕ is added to the list of representatives and

$[\![\phi]\!]$ to the list of sets corresponding to the representatives. As the diagram of G is finite, this procedure terminates. In fact, by testing for each representative ϕ both $[\![\phi]\!] \subseteq [\![\psi]\!]$ and $[\![\psi]\!] \subseteq [\![\phi]\!]$ the algorithm does not only determine whether ϕ represents a new class or not, but also keeps track of the relations in the diagram.

2.7 Games and bisimulations

Let us finish this chapter with the introduction of *Ehrenfeucht* games and their relation to (layered) bisimulations and to semantic types in general. In the next chapter we will occasionally use this kind of game to decide the equivalence of nodes in Kripke models for formulas in certain fragments of **IpL**. For an application of Ehrenfeucht games to second-order and intensional logic see [Doets 87].

In the present context an *Ehrenfeucht game* is a game with two Kripke models, played by two players (player I and player II). At the start player I makes a choice between the two models, by pointing to a world in one of the models. After this start of the game, each of the players in turn will point to a world in the player's model. If a player has chosen world l as the previous move, the l' for the present move has to fulfill the condition that $l \leq l'$. If player I made a move by choosing world k, the world l in the move of player II will also have to meet the condition that $atom(k) = atom(l)$.

The game is finished is one of the players is unable to come up with a satisfactory world. A player that cannot make a valid move in turn has lost.

The idea behind this kind of game is simple. Player II will win the game if able to simulate each of the moves of player I. As player I may choose models first, a *winning strategy* for player II is only possible if there is a simulation relation between the models.

We will use $G(K, L)$ for the Ehrenfeucht game with models K and L defined by the rules above. For player II having a winning strategy we introduce the notation $\models G(K, L)$.

2.7.0.1. Fact. *Let $G(K, L)$ be an Ehrenfeucht game for Kripke models K and L. Then $\models G(K, L)$ iff there exists a bisimulation S between K and L and S is full ($dom(S) = K$ and $ran(S) = L$).*

This fact is a simple consequence of the similarity between the definition of an Ehrenfeucht game above and the definition of a bisimulation in the preliminaries of this chapter.

In the sequel the Ehrenfeucht games all will be played on finite n-models. Our first (simple) refinement of this general scheme of Ehrenfeucht games will be the introduction of two *starting worlds*.

In a game $G(K, L, \langle k, l\rangle)$ with starting worlds $k \in K$ and $l \in L$ (and K and L finite n-models) the first move of each player has to be either k or l. As an easy corollary of fact 2.7.0.1 we have $\models G(K, L, \langle k, l\rangle)$ iff $k \leftrightarrows^n l$.

Other refinements of the scheme of Ehrenfeucht games will be introduced in Chapter 3.

Chapter 3

Exact Models in IpL

3.1 Introduction

In this chapter we will describe all non-trivial finite fragments of intuitionistic propositional logic with atoms in some finite set $\{p_1, \ldots, p_n\}$ and connectives in the set $\{\wedge, \vee, \rightarrow, \neg, \neg\neg\}$. Each of these fragments will be denoted by the number of atoms and the set of connectives used, like $[\wedge, \vee]^2$ for the fragment with two atoms and conjunction and disjunction as its only connectives. Not included are the descriptions of the trivial fragments $[\neg]^n$, $[\neg\neg]^n$ and the fragment with n atomic formulas and no connectives.

Our main task in this chapter will be to show how to construct exact Kripke models for fragments of $\mathbf{IpL}^n$, using the notion of semantic type introduced in Chapter 2. In some cases (i.e. the fragments $[\vee, \neg]^n$ and fragments with $\neg\neg$ without $\neg$) there exists an exact model, but no exact Kripke model. For these fragments we will show to construct, via a *completion* of the exact model, a *universal model* that can be used to calculate the diagram (and subdiagrams). Such a completion of the exact model will be called a *Kripke completion*.

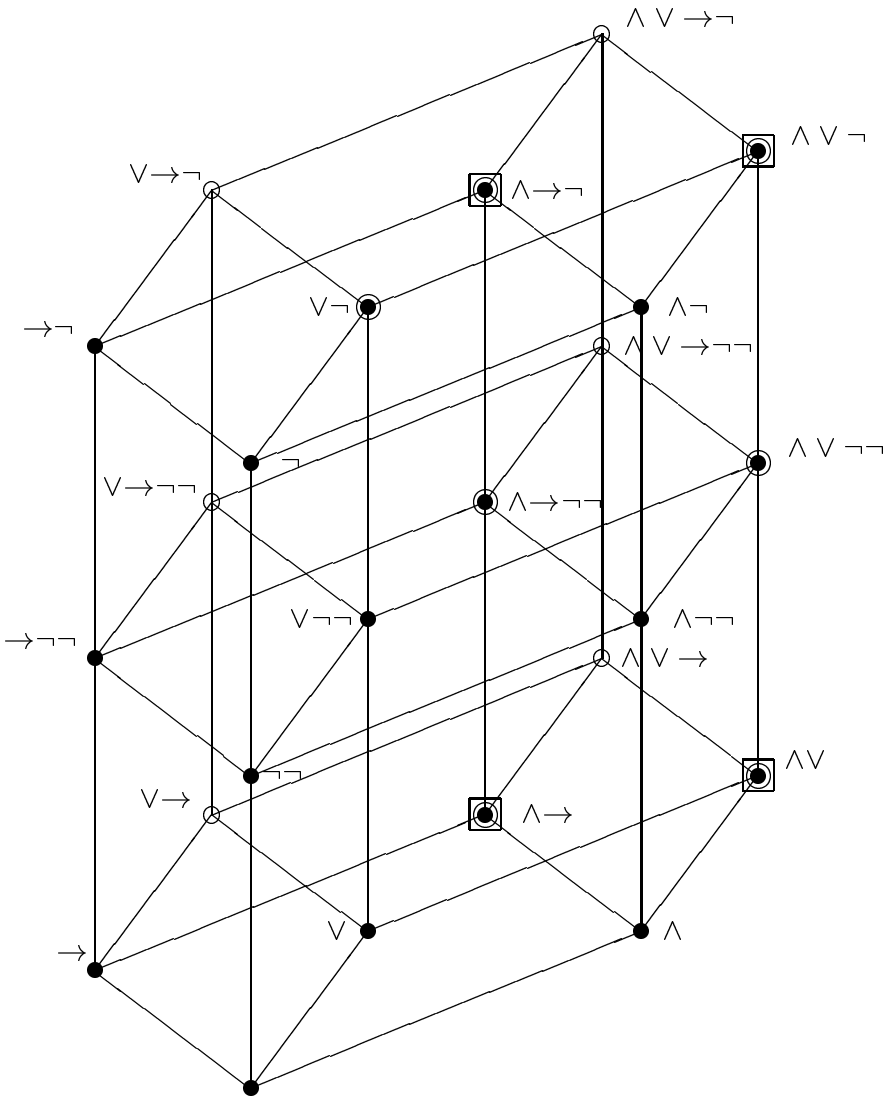
In the sequel we will define, for each of the fragments F in $\mathbf{IpL}^n$ with an exact model, semantic types $\tau_F(k)$ and corresponding types $\phi_F(k)$ for nodes k in a Kripke model. As it should be clear from the context which is the fragment in question, we will most often drop the index F.

The fragments of **IpL** with connectives in the set $\{\wedge, \vee, \rightarrow, \neg, \neg\neg\}$ can be pictured as a lattice (using the inclusion of fragments defined in the preliminaries of the introduction).

The lattice of fragments in the picture below (which was also given in the general introduction in Chapter 1) provides us with an overview of the (non-trivial) fragments that can be obtained by restricting the set of connectives.

Recall that fragments with an infinite diagram (at least in case of more than one propositional variable) are denoted by an open circle. Finite fragments (in $\mathbf{IpL}^n$) are pictured as closed circles and fragments with an exact model have a closed circle

surrounded by an additional open circle. Fragments with an exact Kripke model are marked by a square.



7. FIGURE. *The lattice of fragments in* **IpL**.

As was pointed out in Chapter 2, the diagrams of fragments with an exact Kripke model can be calculated very efficiently if the model is given. The same is true for the subfragments of fragments with an exact Kripke model (just restrict the calculations of formulas and sets according to the restrictions in the subfragment).

As observed in the introduction of Chapter 2, a fragment of **IpL** has a finite exact model iff its diagram is a finite, distributive lattice.

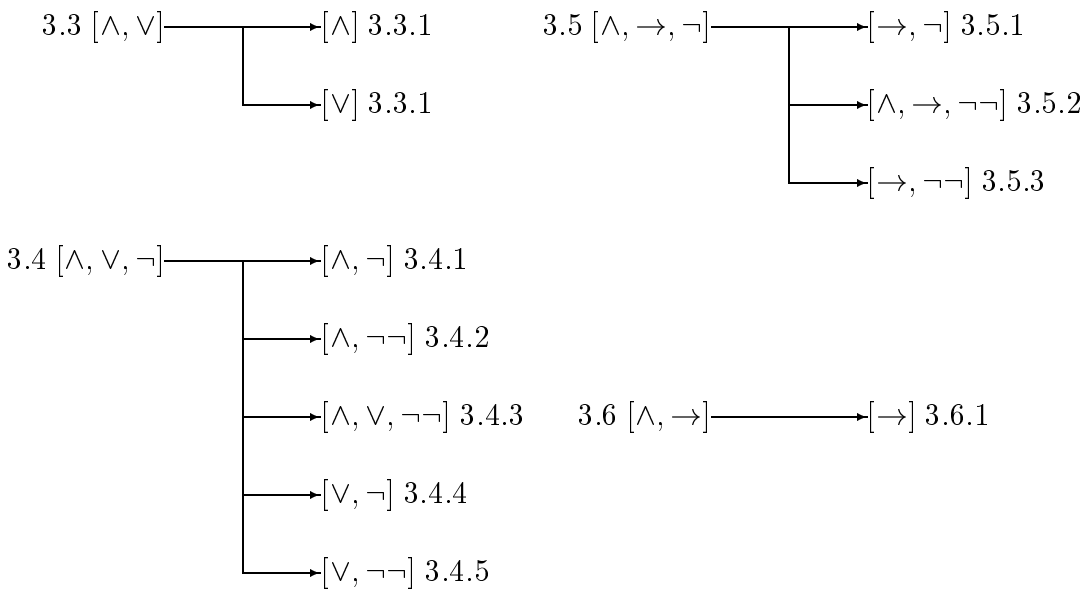
This criterion may be necessary and sufficient for the existence of an exact model for a fragment of $\mathbf{IpL}^n$, but it does not reveal how to obtain an exact model for a particular fragment.

If we knew the irreducible formulas in a fragment F, we could order them with $\dashv$ to obtain $Exm(F)$, the exact model of F. But determining the irreducible formulas in

F may be far from easy. For example in fragments not containing the disjunction (like the $[\wedge, \rightarrow, \neg]$ fragments) it is not immediately clear which formulas are irreducible in the lattice of $Diag(F)$.

In the sequel of this chapter we will construct exact models for fragments F by defining an appropriate semantic type and a (straightforward) ordering.

Except for the preliminaries, this chapter has four sections. In the first subsection of each section we describe one of the fragments with an exact Kripke model. The other subsections deal with subfragments that do not have an exact Kripke model of their own.



8. FIGURE. *The structure of this chapter.*

The general structure of a subsection about fragment F is first to define a class of Kripke models $\mathcal{M}$, for which the fragment is complete. We then define semantic types $\tau_F(k)$ and type formulas $\phi_F(k)$ for the nodes k in the Kripke models in $\mathcal{M}$. In general, this set of semantic types in F can be turned into a (minimal) complete model for F. This *universal* model for F will contain an exact model, if such a model exists for F. In case F has an exact Kripke model, the exact Kripke model and the universal model will coincide.

3.2 Preliminaries

If F is a fragment of **IpL** with a finite exact model, the elements in such a model correspond to the irreducible formulas in $Diag(F)$, as observed in Chapter 2. For

IpL fragments F this implies that if an exact model exists, it is unique (up to isomorphism).

To prove this, we will show that the order in the exact model is determined by the derivability relation. Let ϕ and ψ be irreducible formulas in F and let k_ϕ and k_ψ denote the corresponding nodes in an exact model $Exm(F)$. If ω is the correspondence between formulas and closed subsets in $Exm(F)$ then clearly:

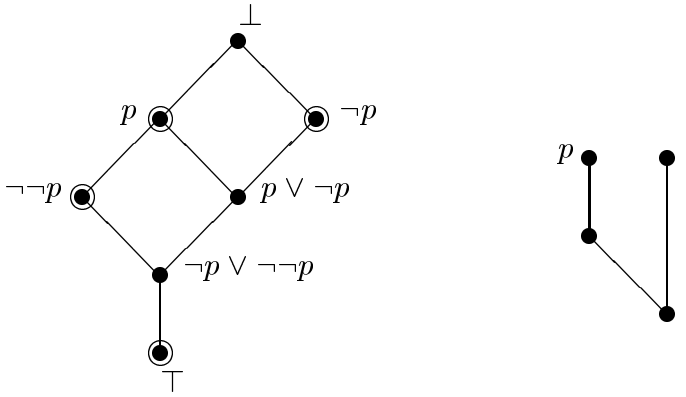
$$\phi \vdash \psi \quad \Leftrightarrow \quad \omega(\phi) \subseteq \omega(\psi) \quad \Leftrightarrow \quad k_\psi \leq k_\phi.$$

A fortiori this is true if F has an exact Kripke model.

3.2.0.1. Fact. *If F is a fragment in* **IpL** *and F has an exact (Kripke) model, then this model is unique up to isomorphism.*

Because of this fact we will in the sequel, when dealing with exact models in **IpL** fragments, simply write 'the' exact (Kripke) model instead of 'an' exact (Kripke) model.

As an example of the relationship between the diagram and the exact (Kripke) model of a fragment, figure 9 shows the diagram and the exact Kripke model of the fragment $[\wedge, \vee, \neg]^1$, where the irreducible elements in the diagram are marked with an extra circle.



9. Figure. *The diagram of $[\wedge, \vee, \neg]^1$ (left) and its exact Kripke model (right).*

In case $\vee$ is in the **IpL** fragment F, $\vee$ will naturally act as the join in the diagram of F. Hence the irreducibles in F will be the $\vee$-irreducible formulas (i.e. those formulas ϕ in F such that for all ψ and χ in F, $\phi \vdash \psi \vee \chi$ implies $\phi \vdash \psi$ or $\phi \vdash \chi$).

To characterize the $\vee$-irreducible formulas in **IpL** we will use the *Aczel slash* (see for example [TD 88]).

3.2.0.2. Definition. (Aczel slash) *Let Γ be a set of* **IpL** *formulas. For an* **IpL** *formula ϕ define $\Gamma \mid \phi$ inductively as:*

1. $\Gamma \mid p \quad \Leftrightarrow \quad \Gamma \vdash p$ *for p atomic or $p = \bot$;*
2. $\Gamma \mid \phi \wedge \psi \quad \Leftrightarrow \quad \Gamma \mid \phi$ *and* $\Gamma \mid \psi$;
3. $\Gamma \mid \phi \vee \psi \quad \Leftrightarrow \quad \Gamma \mid \phi$ *or* $\Gamma \mid \psi$;
4. $\Gamma \mid \phi \to \psi \quad \Leftrightarrow \quad \Gamma \vdash \phi \to \psi$ *and* $(\Gamma \mid \phi \quad \Rightarrow \quad \Gamma \mid \psi)$.

3.2.0.3. FACTS. *Let Γ be a set of* **IpL** *formulas and let ϕ and ψ be* **IpL** *formulas.*

1. *([Kleene 62]) If $\phi \not\equiv \bot$ then ϕ is $\vee$-irreducible iff $\phi \mid \phi$.*
2. *If $\Gamma \mid \phi$ then $\Gamma \vdash \phi$.*
3. *If $\Gamma \vdash \phi \rightarrow \psi$ and $\Gamma \nvdash \phi$ then $\Gamma \mid \phi \rightarrow \psi$.*
4. *All formulas in $[\wedge, \rightarrow, \neg]$ are either equivalent to $\bot$ or $\vee$-irreducible.*
5. *All formulas $\neg\phi$ not equivalent to $\bot$ are $\vee$-irreducible.*

Especially the last two of the above facts will be useful in this chapter.

In the rest of this chapter the Kripke models used will be **IpL** models (reflexive, transitive and anti-symmetric). In particular, if we mention n-models in this section we mean **IpL** n-models.

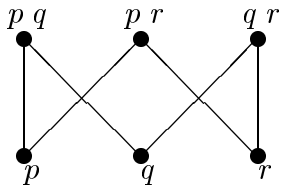
3.3 The $[\wedge, \vee]$ fragments

The structure of the $[\wedge, \vee]^n$ fragments is relatively well known (see [DP 90] for example):

3.3.0.1. FACTS. *Let $\vdash_c$ be the derivability relation in* **CpL**.

1. *The $[\wedge, \vee]^n$ fragments in* **IpL** *and* **CpL** *coincide. For formulas ϕ and ψ in $[\wedge, \vee]$:*
$$\phi \vdash \psi \quad \Leftrightarrow \quad \phi \vdash_c \psi.$$
2. *The diagram of $[\wedge, \vee]^n$ is isomorphic to the free distributive lattice over n generators.*
3. *Each $\phi \in [\wedge, \vee]^n$ is equivalent to a finite disjunction of $[\wedge]^n$ formulas.*
4. *All $[\wedge]^n$ formulas are $\vee$-irreducible.*
5. *The diagram of $[\wedge]^n$ is dual to the diagram of $[\vee]^n$.*
6. *For all $\phi \in [\wedge, \vee]^n$ we have $\bigwedge\{p_1, \ldots, p_n\} \vdash \phi$.*

From 3, it follows that the diagram of $[\wedge]^n$ is almost the exact model of $[\wedge, \vee]^n$. Almost, as the empty set does not correspond to a formula in $[\wedge, \vee]^n$. On the other hand, the conjunction of all atoms in $[\wedge, \vee]^n$ is the bottom element of the diagram. By removing this bottom element from the diagram of $[\wedge]^n$ we get the exact model of $[\wedge, \vee]^n$ (where the empty subset of the exact model corresponds to the bottom of the diagram).



10. FIGURE. *The exact Kripke model of $[\wedge, \vee]^3$.*

The model above has 18 closed subsets, from which we may infer that the diagram of $[\wedge,\vee]^3$ has 18 elements.

Note that for a node k in a Kripke model K the formula $\bigwedge atom^n(k)$ will be the $[\wedge,\vee]^n$-type of k, an axiom of $Th^n(k)$, the $[\wedge,\vee]^n$ theory of k.

3.3.0.2. Definition. *Let k be a node in a Kripke model. The semantic type of k in $[\wedge,\vee]^n$, $\tau^n(k)$ is defined as:*

$$\tau^n(k) = \langle atom^n(k), \emptyset\rangle.$$

If t and t' are semantic types in $[\wedge,\vee]^n$, define:

$$t \preceq t' \quad \Leftrightarrow \quad j_0(t) \subseteq j_0(t').$$

The type formula of k in $[\wedge,\vee]^n$, $\phi^n(k)$ is defined as:

$$\phi^n(k) = \bigwedge j_0(\tau^n(k)).$$

The following lemma states that the above defined types are indeed semantic types in $[\wedge,\vee]^n$ as described in Chapter 2.

3.3.0.3. Lemma. *If k and l are nodes in Kripke models, then*

$$l \Vdash \phi^n(k) \quad \Leftrightarrow \quad \tau(k) \preceq \tau(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad \phi^n(l) \vdash \phi^n(k).$$

Proof. Obvious. ⊣

It is also obvious that if k and l are nodes in an **IpL** Kripke model K, then $k \leq l$ implies $\tau(k) \preceq \tau(l)$.

Note that the type $\langle\{p_1,\ldots,p_n\},\emptyset\rangle$ is a special one in $[\wedge,\vee]^n$ in that a node with such a type will force all formulas in the fragment. We will encounter such *bottom types* again in the sequel and they will be disregarded in the construction of the exact Kripke model (or the universal model in some cases). The reason has been stated above already, for including such a type would prevent the empty set in the exact model to correspond to the bottom of the diagram.

3.3.0.4. Theorem. *The set of types in $[\wedge,\vee]^n$, with exception of the bottom type, i.e. $\langle\{p_1,\ldots,p_n\},\emptyset\rangle$, ordered by $\preceq$ and taking $atom^n(t) = j_0(t)$ for a type t, is the exact Kripke model of $[\wedge,\vee]^n$.*

Proof. From the lemma above and the observations following it, it should be clear that in the intended model each t realizes its own type, (i.e. $\tau^n(t) = t$). Moreover, we have $t \preceq t'$ iff for all formulas in $[\wedge,\vee]^n$ it is true that $t \Vdash \phi \quad \Rightarrow \quad t' \Vdash \phi$. Hence $[\![\phi]\!] = \{t \mid t \Vdash \phi\}$ is a $1-1$ correspondence between closed subsets of the model and formulas in $[\wedge,\vee]^n$. ⊣

In general the exact Kripke model of $[\wedge,\vee]^n$ will have $2^n - 2$ nodes (as there are $2^n - 2$ nonempty proper subsets of a set of n elements).

Obviously the types in $[\wedge, \vee]$ are just sets of atoms if we disregard the general format of semantic types. Hence the exact Kripke model above is isomorphic to the set of proper nonempty subsets of $\{p_1, \ldots, p_n\}$, ordered by inclusion.

As the characteristic functions of closed sets in the exact Kripke model of $[\wedge, \vee]^n$ are the monotonic functions into $\{0, 1\}$, theorem 3.3.0.4 establishes the correspondence between formulas of $[\wedge, \vee]^n$ and monotonic functions of $2^n \mapsto 2$. The problem of determining the number $D(n)$ of these functions (for each n) goes back to Dedekind and is known in a different, but equivalent, form as the Sperner problem (see [Kleitman 69], [Kisielewicz 88]).

In [Sloane 73] there is a table[1] (nr. 1 439) for $D(n)$:

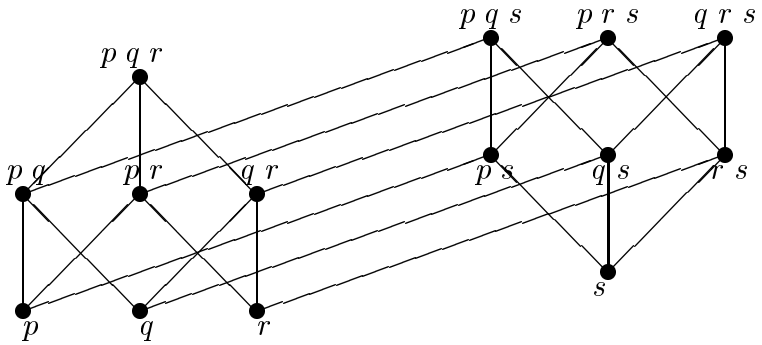
n	$D(n)$
1	1
2	4
3	18
4	166
5	7 579
6	7 828 352
7	2 414 682 040 996

Although there is no simple formula known to calculate the number $D(n)$ there is a simple construction for the exact model of $[\wedge, \vee]^{n+1}$ from the exact model of $[\wedge, \vee]^n$.

Let E^n be the exact model of $[\wedge, \vee]^n$. To obtain the exact model E^{n+1}, take a copy of E^n, denoted as E^n_{n+1}, and connect every $k \in E^n$ with its twin in $k' \in E^n_{n+1}$ (hence $k < k'$). Now change the valuation on E^n_{n+1}, so that in every node $l \in E^n_{n+1}$ also the atom p_{n+1} is forced. Next add a new root below E^n_{n+1} where only p_{n+1} is forced and add a new node k above all nodes in E^n in such a way that $atom(k) = \{p_1, \ldots, p_n\}$.

Clearly the new model exactly realizes all types in $[\wedge, \vee]^{n+1}$, but for the bottom type (where all atoms in the fragment would be forced).

The procedure is illustrated in the figure below, where the exact Kripke model of $[\wedge, \vee]^4$ is constructed from the exact Kripke model of $[\wedge, \vee]^3$.



11. FIGURE. *The exact Kripke model of* $[\wedge, \vee]^4$.

[1] It is convenient to define $D(0) = 0$.

Note that from this construction it simply follows that the exact Kripke model of $[\wedge,\vee]^n$ is the *n-dimensional hypercube* without its top and bottom elements.

3.3.1 $[\wedge]$ and $[\vee]$ fragments

The diagram of $[\wedge]^n$, and dually $[\vee]^n$, is of course isomorphic to the powerset of the nonempty subsets of a set of n elements, ordered by inclusion. Hence the diagram of $[\wedge]^n$ (or $[\vee]^n$) will be isomorphic to the n-dimensional hypercube without its bottom element and have $2^n - 1$ elements.

3.4 The $[\wedge,\vee,\neg]$ fragments

Let us start the treatment of the $[\wedge,\vee,\neg]$ fragments by defining an Ehrenfeucht game for this fragment (see definition 2.7).

3.4.0.1. Definition. *Let K and L be finite Kripke models, $k \in K$ and $l \in L$. The* Ehrenfeucht game *for $[\wedge,\vee,\neg]^n$ with starting worlds k and l, $G^n(K,L,\langle k,l\rangle)$, is a game between two players, I and II, who each make exactly one move, in turn.*

Player I starts by choosing a terminal node m_I above either k or l. Player II replies by choosing a terminal node m_{II} above k, if $l \leq_L m_I$, or above l, if $k \leq_K m_I$.

Player II has won the game if $atom^n(k) = atom^n(l)$ and $atom^n(m_I) = atom^n(m_{II})$.

$\models G^n(K,L,\langle k,l\rangle)$ will denote that there is a winning strategy for player II in the game $G^n(K,L,\langle k,l\rangle)$.

Let $Th^n(k)$ denote the $[\wedge,\vee,\neg]^n$ theory of node k. For finite Kripke models K and L (and $k \in K$, $l \in L$), we have the following theorem.

3.4.0.2. Theorem. $\models G^n(K,L,\langle k,l\rangle) \quad \Leftrightarrow \quad Th^n(k) = Th^n(l)$

Proof. $\Rightarrow$: By induction on the length of $\phi \in [\wedge,\vee,\neg]^n$ we will prove that $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$. The cases where ϕ is either atomic, a conjunction or a disjunction are trivial. Assume $\phi = \neg\psi$ and $k \Vdash \phi$. Then for no terminal node m above k it will be true that $m \Vdash \psi$. Suppose m_I is a terminal node such that $l \leq m_I$. If $m_I \Vdash \psi$ then, as II has a winning strategy for the game $G(K,L,\langle k,l\rangle)$, there is a terminal node $m_{II} \geq k$ such that $atom^n(m_I) = atom^n(m_{II})$. Which would imply $m_{II} \Vdash \psi$, a contradiction. This proves that for no terminal node $m_I \geq l$ $m_I \Vdash \psi$, and hence $l \Vdash \neg\psi$.

$\Leftarrow$: Note that $Th^n(k) = Th^n(l)$ implies $atom^n(k) = atom^n(l)$. Suppose player I chooses m_I (say in K, above k). Recall the definition of $\phi^n_{\mathbf{CpL}}(m_I)$ (definition 2.3.0.2). Then $k \nVdash \neg\phi^n_{\mathbf{CpL}}(m_I)$ and, as $\neg\phi_{\mathbf{CpL}}(m_I)$ is equivalent to a formula in $[\wedge,\vee,\neg]^n$ and $Th^n(k) = Th^n(l)$, $l \nVdash \neg\phi^n_{\mathbf{CpL}}(m_I)$. So, for some terminal node $m_{II} \geq l$, $m_{II} \Vdash \phi^n_{\mathbf{CpL}}(m_I)$, which implies $atom^n(m_I) = atom^n(m_{II})$. Hence there is a winning strategy for II in the game $G(K,L,\langle k,l\rangle)$. $\dashv$

The proof of the theorem above contains both a suggestion for the definition of $\tau^n(k)$, the semantic type in $[\wedge, \vee, \neg]^n$, and of $\phi^n(k)$, the type in $[\wedge, \vee, \neg]^n$ (i.e. an axiom for $Th^n(k)$). Recall the definition of $\phi^n_{\mathbf{CpL}}(k)$ from definition 2.3.0.1.

3.4.0.3. DEFINITION. *Let k be a node in a finite Kripke model and let $Ter(k)$ denote the set of terminal nodes above k:*

$$Ter(k) = \{m \geq k \mid m \text{ is a terminal node}\}.$$

Define:

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset \rangle & \text{if} \quad \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\} \rangle & \text{otherwise.} \end{cases}$$

For semantic types t and t' in $[\wedge, \vee, \neg]^n$ define:

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t))$$

$$\phi^n(k) = \begin{cases} \phi^n_{\mathbf{CpL}}(k) & \text{if } \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \bigwedge j_0(\tau^n(k)) \wedge \\ \neg\neg \bigvee \{\phi^n_{\mathbf{CpL}}(l) \mid \tau^n(l) \in j_1(\tau^n(k))\} & \text{otherwise.} \end{cases}$$

Observe that in particular $\tau^n(k) = \langle atom^n(k), \emptyset \rangle$ if k is a terminal node.

The next lemma shows we are on the right track with these characterizations of the $[\wedge, \vee, \neg]^n$ theory of a node in a Kripke model.

But let us first state as a fact the following simple consequence of the definition of a semantic $[\wedge, \vee, \neg]^n$ type.

3.4.0.4. FACT. *If $k \in K$ and $l \in L$ are nodes in finite Kripke models and $\tau^n(k) = \tau^n(l)$, then $\models G(K, L, \langle k, l \rangle)$.*

The next lemma, in combination with theorem 3.4.0.2, has as a consequence that for nodes $k \in K$ and $l \in L$ in finite Kripke models K and L also $\models G(K, L, \langle k, l \rangle)$ implies $\tau^n(k) = \tau^n(l)$.

3.4.0.5. LEMMA. *Let k and l be nodes in finite Kripke models. Then the following statements are equivalent:*

1. $l \Vdash \phi^n(k)$;
2. $\tau^n(k) \preceq \tau^n(l)$;
3. $Th^n(k) \subseteq Th^n(l)$;
4. $\phi^n(l) \vdash \phi^n(k)$.

Proof. We will prove $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$.

$1 \Rightarrow 2$: Assume $l \Vdash \phi^n(k)$. If $\phi^n(k) = \phi^n_{\mathbf{CpL}}(k)$, then clearly for all $m \geq l$ we have $atom^n(m) = atom^n(k)$ and hence $\tau^n(l) = \langle atom^n(k), \emptyset \rangle = \tau^n(k)$. On the other hand, if $\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\} \rangle$, then for $l' \in Ter(l)$ we can

prove that $\tau^n(l') \in j_1(\tau^n(k))$. Let $l' \in Ter(l)$. Then we have, by the definition of $\phi^n(k)$, that $l' \Vdash \bigvee\{\phi^n_{\mathbf{CpL}}(m) \mid \tau^n(m) \in j_1(\tau^n(k))\}$. Hence $l' \Vdash \phi^n_{\mathbf{CpL}}(m)$ for some $m \in Ter(k)$ and, as above, infer that $\tau^n(l') = \tau^n(m)$. Note that either $\tau^n(l) = \tau^n(l')$ for some $l' \in Ter(l)$ and thus $\tau^n(l') \in j_1(\tau^n(k))$, or $j_1(\tau^n(l)) \neq \emptyset$ and we proved $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. In both cases we may conclude $\tau^n(k) \preceq \tau^n(l)$ as trivially $atom^n(k) \subseteq atom^n(l)$ holds if $l \Vdash \phi^n(k)$.

$2 \Rightarrow 3$: Assume $\tau^n(k) \preceq \tau^n(l)$. Note that if for all $m > k$ we have $atom^n(m) = atom^n(k)$, then from $\tau^n(k) \preceq \tau^n(l)$ we may infer that $\tau^n(k) = \tau^n(l) = \langle atom^n(k), \emptyset\rangle$ and hence by fact 3.4.0.4 $\models G(K, L', \langle k, l\rangle)$. Which proves $Th^n(k) = Th^n(l)$, using theorem 3.4.0.2.

So assume there is a $k' \in Ter(k)$ with $atom^n(k') \neq atom^n(k)$. Let $l \in L$ and let L' be the model constructed from L by adding a new node l_0 with $atom^n(l_0) = atom^n(k)$ and placed below l and all terminal nodes above k. Note that such a construction of L' as a finite Kripke model is possible as $atom^n(k) \subseteq atom^n(l)$, which we may infer from the assumption. Also from the assumption that $\tau^n(k) \preceq \tau^n(l)$ we may conclude that $\tau^n(l_0) = \tau^n(k)$. By fact 3.4.0.4 this implies $\models G(K, L', \langle k, l_0\rangle)$ and hence by theorem 3.4.0.2, $Th^n(k) = Th^n(l_0)$. From the construction of L' infer that as a consequence we have $Th^n(k) \subseteq Th^n(l)$.

$3 \Rightarrow 4$: Note that from the two previous steps we may conclude that $\phi^n(m)$ is an axiom of $Th^n(m)$ (for any node m in a finite Kripke model). For suppose $\phi \in Th^n(k)$ and $l \Vdash \phi^n(k)$. Then by combining the first and the second part of this proof we have $Th^n(k) \subseteq Th^n(l)$ and hence $l \Vdash \phi$. Which, by the completeness theorem, proves $\phi^n(k) \vdash \phi$.
From the fact that $\phi^n(m)$ is an axiom for $Th^n(m)$ one easily proves that the inclusion of the theories $Th^n(k) \subseteq Th^n(l)$ implies the interderivability of their axioms: $\phi^n(l) \vdash \phi^n(k)$.

$4 \Rightarrow 1$: As $\phi^n(l)$ is the axiom of $Th^n(l)$, we know that $l \Vdash \phi^n(l)$. And hence from $\phi^n(l) \vdash \phi^n(k)$ we infer $l \Vdash \phi^n(k)$. ⊣

From the definition of semantic types in $[\wedge, \vee, \neg]^n$ it is clear that there are only finitely many of these types. It is also easy to prove that all tuples of the form $\langle S, T\rangle$ such that:

1. T is a set of types $\langle U, \emptyset\rangle$, where $U \subseteq \{p_1, \ldots, p_n\}$,
2. if $T \neq \emptyset$ then $S \subseteq \bigcap\{j_0(t) \mid t \in T\}$,
3. if $T = \{\langle U, \emptyset\rangle\}$ then $S \neq U$,

are types in $[\wedge, \vee, \neg]^n$.

As each semantic type of $[\wedge, \vee, \neg]$ can be realized in a rooted **IpL** model with depth less than two, we have established the following fact.

3.4.0.6. Fact. $[\wedge, \vee, \neg]$ *is complete for rooted* **IpL** *models of depth less than two.*

An intermediate logic is a *conservative extension* of a fragment of **IpL** if for any two formulas ϕ and ψ in the fragment ψ is a consequence of ϕ in the intermediate logic iff $\phi \vdash_{\mathbf{IpL}} \psi$. The intermediate logic **IpL** + $((p{\to}(((q{\to}r){\to}q){\to}q)){\to}p){\to}p$, complete for models of depth less than two, can be proved to be a maximal conservative

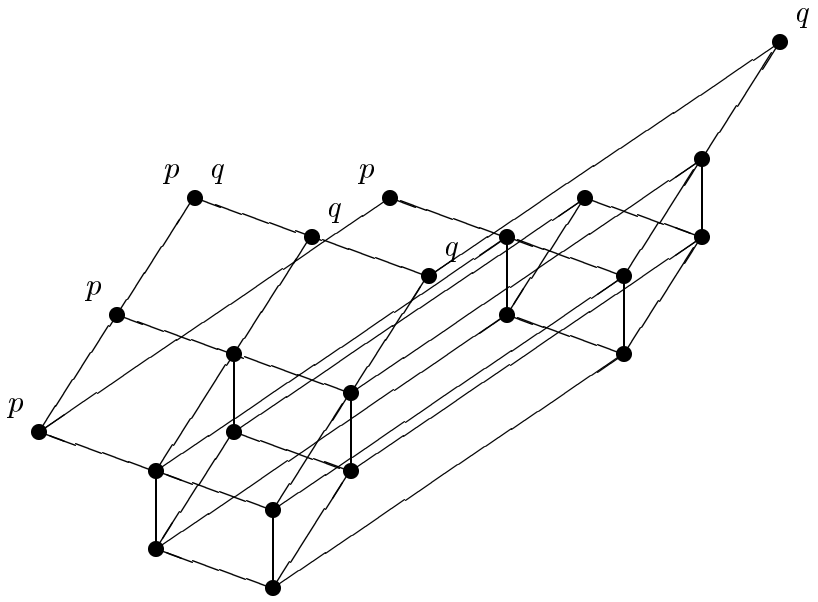
extension of $[\wedge, \vee, \neg]$. But it is not unique. A. Chagrov announced a proof for the existence of continuum of maximal conservative extensions of $[\wedge, \vee, \neg]^n$ for each $n > 1$.

Ordering the types in $[\wedge, \vee, \neg]^n$, putting $\langle S, T\rangle \preceq \langle S', T'\rangle$ if $S \subseteq S'$ and $T' \subseteq T$ will yield a Kripke model $Exm([\wedge, \vee, \neg]^n)$ (with $atom^n(t) = j_0(t)$).

Note that as $Exm([\wedge, \vee, \neg]^n)$ realizes all semantic types in $Exm([\wedge, \vee, \neg]^n)$, it is a complete Kripke model for this fragment. As a consequence of the above lemma also $[\![\phi^n(k)]\!] = \uparrow k$. Hence every closed subset of $Exm([\wedge, \vee, \neg]^n)$ can be obtained as the valuation of a formula in $[\wedge, \vee, \neg]^n$. Which proves the following theorem.

3.4.0.7. THEOREM. *The model $Exm([\wedge, \vee, \neg]^n)$ defined above is the exact Kripke model of $[\wedge, \vee, \neg]^n$.*

As an example, in figure 12 we give the exact Kripke model of $[\wedge, \vee, \neg]^2$.



12. FIGURE. *The exact Kripke model of $[\wedge, \vee, \neg]^2$.*

As all types in $[\wedge, \vee, \neg]^n$ are formulas in $[\wedge, \neg]^n$, we have the following corollary.

3.4.0.8. COROLLARY. (The $[\wedge, \neg]$ normal form) *In $[\wedge, \vee, \neg]^n$ each formula is equivalent to a disjunction of formulas in $[\wedge, \neg]^n$.*

Note that as each formula in $[\wedge, \neg]^n$ which is not equivalent to $\bot$ is irreducible (use fact 3.2.0.3.4), each of these formulas will be (equivalent to) a type in $[\wedge, \vee, \neg]^n$. As a result, we may state the following fact.

3.4.0.9. FACT. *By leaving out $\bot$, the diagram of $[\wedge, \neg]^n$ becomes the exact Kripke model of $[\wedge, \vee, \neg]^n$.*

We will use the exact Kripke model of $[\wedge,\vee,\neg]^n$ in the proof of the characterization of the $[\wedge,\vee,\neg]$ formulas in **IpL**. First we introduce the *terminal reduction* of a rooted Kripke model.

3.4.0.10. DEFINITION. *For a finite Kripke model K with root k, the submodel $(\uparrow k)^T$, with domain $\{k\} \cup Ter(k)$ and the accessibility relation and valuation inherited from K is called the* terminal reduction *of K.*

Obviously, for a node k in a finite Kripke model K, the semantic type (in $[\wedge,\vee,\neg]^n$) of k in K and in $(\uparrow k)^T$, the terminal reduction of the submodel $\uparrow k$, are the same.

Hence, a rooted Kripke model and its terminal reduction force the same $[\wedge,\vee,\neg]$ formulas (have the same $[\wedge,\vee,\neg]$ theory).

3.4.0.11. THEOREM. *An* **IpL** *formula ϕ is equivalent to a $[\wedge,\vee,\neg]$ formula iff for every node k in a finite Kripke model:*

$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^T \Vdash \phi.$$

Proof. As observed above, the node k in a finite Kripke model K and the root of the terminal reduction $(\uparrow k)^T$, have the same semantic type in $[\wedge,\vee,\neg]^n$. By lemma 3.4.0.5 this implies that k and $(\uparrow k)^T$ force the same $[\wedge,\vee,\neg]$ formulas. Which proves one direction of the theorem.

For the other direction, assume ϕ is a formula in $\mathbf{IpL}^n$ and for every k in a finite Kripke model it is true that $k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^T \Vdash \phi$. Let χ be the $[\wedge,\vee,\neg]$ formula with $[\![\chi]\!] = [\![\phi]\!]$ in $Exm([\wedge,\vee,\neg]^n)$.

We will show that ϕ is equivalent to χ by showing (for k a node in a finite Kripke model) $k \Vdash \phi \quad \Leftrightarrow \quad k \Vdash \chi$. We first use the assumption that $k \Vdash \phi$ is equivalent to $(\uparrow k)^T \Vdash \phi$. The root k in $(\uparrow k)^T$ clearly bisimulates the node $\tau^n(k)$ in the terminal reduction $(\uparrow \tau^n(k))^T$ of the submodel $\uparrow \tau^n(k)$ in the exact Kripke model $Exm([\wedge,\vee,\neg]^n$. Hence, $(\uparrow k)^T \Vdash \phi$ is equivalent to $(\uparrow \tau^n(k))^T \Vdash \phi$. Which, by the assumption about ϕ is equivalent to $\tau^n(k) \Vdash \phi$ (in the exact Kripke model) and by definition of χ also to $\tau^n(k) \Vdash \chi$. As χ is a $[\wedge,\vee,\neg]^n$ formula, $\tau^n(k) \Vdash \chi \quad \Leftrightarrow \quad k \Vdash \chi$, which proves $k \Vdash \phi$ to be equivalent to $k \Vdash \chi$. ⊣

3.4.1 The $[\wedge,\neg]$ fragments

We will prove that $[\wedge,\neg]$ is complete for models based on the simple frame of two connected worlds (and which will be called **2**). We will prove, that, as a consequence, the **IpL** fragment $[\wedge,\neg]$ is in fact the same as the $[\wedge,\neg]$ fragment of the three valued Heyting logic $\mathbf{H}_3$. The construction of the exact models for $\mathbf{H}_3^n$, the fragments of $\mathbf{H}_3$ with atoms restricted to the set $\{p_1,\ldots,p_n\}$, serves as an example to show that the technique of semantic types is also applicable in intermediate logics.

First however, we will compute the number of equivalence classes in $[\wedge,\neg]^n$, using the $Exm([\wedge,\vee,\neg]^n)$ from the previous subsection. Recall from fact 3.4.0.9 that the model $Exm([\wedge,\vee,\neg]^n)$ is isomorphic to the diagram of $[\wedge,\neg]^n$, without the bottom element.

3.4.1.12. THEOREM.

$$|Diag([\wedge, \neg]^n) = \sum_{k=0}^{n} \binom{n}{k} (2^{2^k} - 1) + 1.$$

Proof. If $S \subseteq \{p_1, \ldots, p_n\}$ and $|S| = k$, then there are 2^{n-k} sets U, in such a way that $S \subseteq U \subseteq \{p_1, \ldots, p_n\}$. Excluding the combination of S and $\langle S, \emptyset \rangle$, this implies that there are $2^{2^{n-k}} - 1$ semantic types t in $[\wedge, \vee, \neg]^n)$ with $j_0(t) = S$. As a consequence, we have

$$|Diag([\wedge, \neg]^n) = \sum_{k=0}^{n} \binom{n}{k} (2^{2^{n-k}} - 1) + 1.$$

Now use $\binom{n}{n-k} = \binom{n}{k}$ to obtain the formula in the theorem. ⊣

Let us first prove that $[\wedge, \neg]$ is complete for **2**-models, that is for models based on the frame **2**. In fact the theorem we will prove in the sequel is somewhat stronger and states that $[\wedge, \neg]^n$ is complete for the n-models based on **2**.

As a bridge between **IpL** models and **2**-models we first define *terminal models.*

3.4.1.13. DEFINITION. *If K is a finite* **IpL** *model K and $k, l \in K$ we call $\langle k, l \rangle$ a* terminal submodel *if l is a terminal node in K and $k < l$.*

Obviously a terminal submodel defined in K is a Kripke model in its own right as a submodel[2] of K.

3.4.1.14. LEMMA. *Let ϕ be a $[\wedge, \neg]$ formula, k a node in a finite* **IpL** *model K and $\langle k, l \rangle$ a terminal submodel in K. If $K \Vdash \phi$ then $\langle k, l \rangle \Vdash \phi$.*

Proof. By induction on the length of ϕ. If ϕ is atomic or a conjunction the proof is obvious. Note that in case $\phi = \neg\psi$, we may infer from $k \Vdash \neg\psi$ that the terminal node l will not force ψ. But the terminal node l above k in K and l in $\langle k, l \rangle$ force the same formulas. Hence $\langle k, l \rangle \Vdash \neg\psi$. ⊣

3.4.1.15. LEMMA. *Let ϕ be a $[\wedge, \neg]$ formula, k a node in a finite* **IpL** *model K. If $k \nVdash \phi$ then for some terminal submodel $\langle k, l \rangle$ in K, $\langle k, l \rangle \nVdash \phi$.*

Proof. By induction on the length of ϕ. The atomic and conjunction cases are easy. In case $\phi = \neg\psi$, we may infer from $k \nVdash \neg\psi$ that some terminal node $l \geq k$ must force ψ. Now any terminal model with this l will meet the condition from the lemma. ⊣

3.4.1.16. THEOREM. *The* **IpL** *fragment $[\wedge, \neg]$ is complete for* **2***-models.*

[2] Although not necessarily a generated submodel, as there may be an $m \in \uparrow k$ such that $m \notin \langle k, l \rangle$.

Proof. By combining lemma 3.4.1.14 and 3.4.1.15. ⊣

Let us briefly introduce the three valued Heyting logic, $\mathbf{H}_3$. The most concise definition of $\mathbf{H}_3$ would be: $\mathbf{H}_3$ is the logic of the **2**-models. If we use $\Vdash_2$ for forcing in **2**-models, $\phi \Vdash_2 \psi$ if for all k in a **2**-model $k \Vdash_2 \phi$ implies $k \Vdash_2 \psi$, and $\vdash_3$ for derivability in $\mathbf{H}_3$, then $\mathbf{H}_3$ being the logic of **2**-models comes down to:

$$\phi \vdash_3 \psi \quad \Leftrightarrow \quad \phi \Vdash_2 \psi.$$

An alternative, and more traditional, definition of $\mathbf{H}_3$ introduces $\vdash_3$ by truth tables for the connectives:

$\wedge$	f	$*$	t
f	f	f	f
$*$	f	$*$	$*$
t	f	$*$	t

$\vee$	f	$*$	t
f	f	$*$	t
$*$	$*$	$*$	t
t	t	t	t

$\rightarrow$	f	$*$	t
f	t	t	t
$*$	f	t	t
t	f	$*$	t

	$\neg$
f	t
$*$	f
t	f

It is left to the reader to check that these matrices correspond to the behavior of the connectives, according to the definition of forcing in **IpL** models, on the following **2**-model. Here the set $\{0,1\}$ represents the truth value **t**, $\{0\}$ the value $*$ and the empty set corresponds to the value **f**.

There are several alternative axiomatizations of the three valued Heyting logic.

3.4.1.17. Fact. $\mathbf{H}_3$ *can be axiomatized by adding one of the following formulas as an axiom to the axioms of* **IpL**.

1. $(p \leftrightarrow q) \vee (p \leftrightarrow r) \vee (p \leftrightarrow s) \vee (q \leftrightarrow r) \vee (q \leftrightarrow s) \vee (r \leftrightarrow s)$;
2. $p \vee (p \rightarrow q) \vee \neg q$;
3. $(((p \rightarrow (((q \rightarrow r) \rightarrow q) \rightarrow q)) \rightarrow p) \rightarrow p) \wedge ((p \rightarrow q) \vee (q \rightarrow p))$;
4. $((p \rightarrow q) \rightarrow r) \rightarrow (((s \rightarrow p) \rightarrow r) \rightarrow r)$.

The first of these axioms is Gödels formula expressing that there are only three truth values [Gödel 32]. The second is a simplified version of Hosoi's $p \vee \neg p \vee (p \rightarrow q) \vee (q \rightarrow r)$ in [Hosoi 66]. The first conjunct of 3 is the (once) iterated Peirce formula which is true exactly in the frames of depth less than two ([Gabbay 81]). The second conjunct is Dummett's axiom for the intermediate logic **LC**, the logic of linearly ordered frames[Gabbay 81]. In combination these formulas axiomatize the logic of linearly ordered frames of depth less than two, that is the frame **2** (and its subframe with only one world).

Formula 4 stems from Thomas [Thomas 62]. More details can be found in [Troelstra 65].

The definition of semantic types in $\mathbf{H}_3^n$ will not come as a surprise.

3.4.1.18. DEFINITION. *Let k be a node in a* **2***-model. The* semantic type *of k in $\mathbf{H}_3^n$ is defined by:*

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset \rangle & \text{if} \quad \forall l > k . atom^n(k) = atom^n(l) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid k < l\} \rangle & \text{otherwise.} \end{cases}$$

The order of semantic types t and t' in $\mathbf{H}_3^n$ is defined by

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t).$$

Define the type, *$\phi^n(k)$, of k in $\mathbf{H}_3^n$ by:*

$$\phi^n(k) = \begin{cases} \phi^n_{\mathbf{CpL}}(k) & \text{if} \quad j_1(\tau^n(k)) = \emptyset \\ \\ \bigwedge j_0(\tau^n(k)) \wedge \\ \bigwedge \{\neg\neg p \mid p \in j_0(\tau^n(l))\} \wedge \\ \bigwedge \{\neg p \mid p \in \{p_1, \ldots, p_n\} \setminus j_0(\tau^n(l))\} \wedge \\ \bigwedge \{p \leftrightarrow q \mid p, q \in j_0(\tau^n(l)) \setminus j_0(\tau^n(k))\} & \text{if} \quad j_1(\tau^n(k)) = \{\tau^n(l)\}. \end{cases}$$

Observe that in particular $\tau^n(k) = \langle atom^n(k), \emptyset \rangle$ if k is a terminal node. Moreover, if t is a semantic type in $\mathbf{H}_3^n$ and $j_1(t) \neq \emptyset$, then $j_1(t) = \{t'\}$ and $j_1(t') = \emptyset$.

Observe also, that if $k \leq l$ in a **2**-model, then $\tau^n(k) \preceq \tau^n(l)$.

As can be verified easily, the definition of $\phi^n(k)$ assures that $k \Vdash \phi^n(k)$ for k a node in a **2**-model. Note that if k is a terminal node then $\phi^n(k) \equiv \phi^n_{\mathbf{CpL}}(k)$.

Now we are ready to prove, like we did in lemma 3.4.0.5 for $[\wedge, \vee, \neg]^n$, that the types and semantic types introduced for $\mathbf{H}_3^n$ behave like one would expect. In the sequel of this subsection we will use $Th^n(k)$ for the theory of formulas in $\mathbf{H}_3^n$ forced by k.

3.4.1.19. LEMMA. *Let k and l be nodes in* **2***-models. Then the following statements are equivalent:*

1. *$l \Vdash \phi^n(k)$;*
2. *$\tau^n(k) \preceq \tau^n(l)$;*
3. *$Th^n(k) \subseteq Th^n(l)$;*
4. *$\phi^n(l) \vdash \phi^n(k)$.*

Proof. We will prove $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$.

$1 \Rightarrow 2$: We have to prove that either $\tau^n(k) = \tau^n(l)$ or l is a terminal node and $j_1(\tau^n(k)) = \{\tau^n(l)\}$.

In case l and k are both terminal nodes $\phi^n(k)$ is a $\mathbf{CpL}^n$ type and obviously $l \Vdash \phi^n(k)$ implies $\tau^n(k) = \tau^n(l)$. If l is a terminal node and k is not, let k' be the terminal node above k. We will prove $\tau^n(l) = \tau^n(k')$. For $p \in atom^n(k')$ infer from the definition of the type of k that $\phi^n(k) \vdash \neg\neg p$. As $l \Vdash \phi^n(k)$ this assures us that $p \in atom^n(l)$. Hence $atom^n(k') \subseteq atom^n(l)$. Likewise, if $p \notin atom^n(k')$ then $\phi^n(k) \vdash \neg p$ and hence $p \notin atom^n(l)$. Which proves $atom^n(k') = atom^n(l)$ and, as both are terminal nodes, $\tau^n(k) = \tau^n(l)$. Note that if k is a terminal node

$l \Vdash \phi^n_{\mathbf{CpL}}(k)$ obviously implies that $j_1(\tau^n(l)) = \emptyset$ and hence $\tau^n(k) = \tau^n(l)$. So suppose both $j_1(\tau^n(l)) \neq \emptyset$ and $j_1(\tau^n(k)) \neq \emptyset$. Then there is an $l' > l$ such that $l' \Vdash \phi^n(k)$. Again we may infer that $\tau^n(k') = \tau^n(l')$ for the terminal node k' above k. Clearly $atom^n(k) \subseteq atom^n(l)$ and to prove $atom^n(l)$ to be a subset of $atom^n(k)$, let $p \in atom^n(l)$. Now either $p \in atom^n(k)$, or $p \in atom^n(k') \setminus atom^n(k)$ or p is not in $atom^n(k')$. In the first case we are ready and in the third case $\phi^n(k) \vdash \neg p$, contradicting $p \in atom^n(l)$. If $p \in atom^n(k') \setminus atom^n(k)$, note that, as $j_1(\tau^n(l)) \neq \emptyset$ there is a $q \in atom^n(l') \setminus atom^n(l)$. As $atom^n(l') = atom^n(k')$ we will have $\phi^n(k) \vdash p \rightarrow q$, contradicting $p \in atom^n(l)$.

2 $\Rightarrow$ 3: Assume $\tau^n(k) \preceq \tau^n(l)$. If $j_1(\tau^n(k)) = \emptyset$, then obviously $k \not\leftrightarrows l$ and $Th^n(k) = Th^n(l)$. On the other hand, if $j_1(\tau^n(k)) = \{\tau^n(k')\}$, then either $\tau^n(l) = \tau^n(k')$ or $j_1(\tau^n(l)) = \{\tau^n(l')\}$, $\tau^n(l') = \tau^n(k')$ and $atom^n(k) = atom^n(l)$. As k' is a terminal node, from $\tau^n(l) = \tau^n(k')$ we may conclude $l \not\leftrightarrows k'$ and hence $Th^n(k) \subset Th^n(k') = Th^n(l)$. In case $j_1(\tau^n(l)) = \{\tau^n(l')\}$, we conclude from $\tau^n(l') = \tau^n(k')$, that $l' \not\leftrightarrows k'$. As also $atom^n(k) = atom^n(l)$, we may infer that $k \not\leftrightarrows l$ and hence $Th^n(k) = Th^n(l)$.

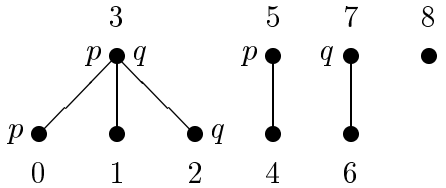
3 $\Rightarrow$ 4: As in the proof of theorem 3.4.0.5 we conclude from the previous steps that in general $\phi^n(m)$ is an axiom of $Th^n(m)$. Hence from $Th^n(k) \subseteq Th^n(l)$ we conclude that $\phi^n(l) \vdash \phi^n(k)$.

4 $\Rightarrow$ 1: As observed earlier $l \Vdash \phi^n(l)$ is a simple consequence of definition 3.4.1.18. Hence trivially, $\phi^n(l) \vdash \phi^n(k)$ implies $l \Vdash \phi^n(k)$. $\dashv$

Now define $Exm(\mathbf{H}^n_3)$ as the ordered set of semantic types in $\mathbf{H}^n_3$. Obviously $Exm(\mathbf{H}^n_3)$ is a Kripke model if we take $atom^n(t) = j_0(t)$ as its valuation. Note that $Exm(\mathbf{H}^n_3)$ will again be a **2**-model.

As $Exm(\mathbf{H}^n_3)$ realizes all semantic types in $\mathbf{H}^n_3$ with lemma 3.4.1.19 one easily proves that the model is complete for $\mathbf{H}^n_3$. Moreover, for every node $k \in Exm(\mathbf{H}^n_3)$ we have a type formula $\phi^n(k)$ such that $[\![\phi^n(k)]\!] = \uparrow k$. As closed subsets in $Exm(\mathbf{H}^n_3)$ correspond to disjunctions of these type formulas we may conclude that $Exm(\mathbf{H}^n_3)$ is the exact Kripke model of $\mathbf{H}^n_3$.

3.4.1.20. Theorem. *The model $Exm(\mathbf{H}_3)$ defined above is the exact Kripke model of $\mathbf{H}^n_3$.*



13. Figure. *The exact Kripke model of $\mathbf{H}^2_3$.*

The irreducible formulas in $\mathbf{H}^2_3$ are:

0. $p \wedge \neg\neg q$	3. $p \wedge q$	6. $\neg p \wedge \neg\neg q$
1. $\neg\neg p \wedge (p \leftrightarrow q)$	4. $\neg\neg p \wedge \neg q$	7. $\neg p \wedge q$
2. $\neg\neg p \wedge q$	5. $p \wedge \neg q$	8. $\neg p \wedge \neg q$

Note that, in contrast to in **IpL**, not all $[\wedge, \neg]^2$ formulas are irreducible in $\mathbf{H}_3$.

For example (as can be proved using the exact Kripke model of $\mathbf{H}_3^2$): $\neg\neg p \wedge \neg\neg q \equiv (\neg\neg p \wedge (p \leftrightarrow q)) \vee (p \wedge \neg\neg q) \vee (\neg\neg p \wedge q)$.

The model above has been used to calculate the diagram of $\mathbf{H}_3^2$. A listing of all 162 equivalence classes can be found in appendix B.2.

From the structure of the exact Kripke model of $\mathbf{H}_3^n$ one can calculate the number of elements in $Exm(\mathbf{H}_3)$ as $\sum_{k=0}^{n} 2^k.\binom{n}{k}$ and the number of classes in $Diag(\mathbf{H}_3^n)$ as:

$$\prod_{k=0}^{n} (2^{2^k-1}+1)^{\binom{n}{k}}.$$

3.4.2 The $[\wedge, \neg\neg]$ fragments

The $[\wedge, \neg\neg]$ fragments have rather simple and regular diagrams. The expressive power of these fragments is too limited to be of very much interest, but each $[\wedge, \neg\neg]^n$ fragment 'almost' has an exact model. For formulas in $[\wedge, \neg\neg]^n$ we have an obvious normal form.

3.4.2.21. Fact. (The $[\wedge, \neg\neg]^n$ normal form) *Each formula in $[\wedge, \neg\neg]^n$ is equivalent to a formula of the form $\bigwedge P \wedge \bigwedge\{\neg\neg q \mid q \in Q\}$ where both P and Q are subsets of $\{p_1, \ldots, p_n\}$, $P \cup Q \neq \emptyset$ and $P \cap Q = \emptyset$.*

To characterize the semantic types for $[\wedge, \neg\neg]^n$, let us introduce a special type of **IpL** models, *n-maximal models* .

3.4.2.22. Definition. *A finite n-model K is called n-maximal if each $k \in K$ forces at least $n-1$ atoms.*

3.4.2.23. Theorem. *If ϕ and ψ formulas in $[\wedge, \neg\neg]^n$ such that $\phi \nvdash \psi$ then there is a node k in an n-maximal model such that $k \Vdash \phi$ and $k \nVdash \psi$.*

Proof. Let $\bigwedge P \wedge \bigwedge\{\neg\neg q \mid q \in Q\}$ be the normal form of ϕ and $\bigwedge R \wedge \bigwedge\{\neg\neg q \mid q \in S\}$ the normal form of ψ. From $\phi \nvdash \psi$ we may infer that either there is an $r \in R$ such that $r \notin P$ or there is an $s \in S$ such that $s \notin P \cup Q$.

In the first case, let $atom^n(k)$ contain all atoms but r and $k < l$ such that $atom^n(l) = \{p_1, \ldots, p_n\}$. Obviously $k \Vdash \phi$ but $k \nVdash \psi$.

In the second case, let k be a node forcing all atoms in $\{p_1, \ldots, p_n\}$ except s. Let k have two successors l_0 and l_1, with $atom^n(l_0) = \{p_1, \ldots, p_n\}$ and $atom^n(l_1) = atom^n(k)$. Again k will force ϕ but not ψ. ⊣

3.4.2.24. Corollary. *The fragment $[\wedge, \neg\neg]^n$ is complete for n-maximal models.*

Recall the definition of $Ter(k)$ from definition 3.4.0.3 as the set of all terminal nodes above the node k. The proof of theorem 3.4.2.23 motivates the following definition of semantic type (in $[\wedge, \neg\neg]^n$) for a node in an n-maximal model.

3.4.2.25. Definition. *Let k be a node in an n-maximal model. Then $\tau^n(k)$, the semantic type of k in $[\wedge, \neg\neg]^n$ is defined by:*

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset \rangle & \text{if} \quad \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\} \rangle & \text{otherwise.} \end{cases}$$

For semantic types t and t' in $[\wedge, \neg\neg]^n$ define:

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

3.4.2.26. Definition. *A node k in an n-maximal model is called a* proper *node, if $j_1(\tau^n(k)) \neq \emptyset$.*

Inspection of the proof of the theorem 3.4.2.23 reveals the following fact.

3.4.2.27. Fact. *If ϕ and ψ formulas in $[\wedge, \neg\neg]^n$ such that $\phi \nvdash \psi$ then there is a proper node k in an n-maximal model such that $k \Vdash \phi$ and $k \nVdash \psi$.*

Note that the ordered set of semantic types of proper nodes for $[\wedge, \neg\neg]^n$, n disjoint **2**-models, is not a Kripke model realizing all the semantic types in $[\wedge, \neg\neg]^n$. By adding terminal nodes with semantic type $\langle Q, \emptyset \rangle$ where $|Q| \geq n - 1$, the ordered set of types becomes an n-maximal model.

3.4.2.28. Definition. *The model $Umod([\wedge, \neg\neg]^n)$ is the ordered set of semantic types in $[\wedge, \neg\neg]^n$.*

To prove that $Umod([\wedge, \neg\neg]^n)$ is a universal model for $[\wedge, \neg\neg]^n$ we will show that it is the Kripke completion of the exact model of the fragment $[\wedge, \neg\neg, \top]^n$, that is the fragment $[\wedge, \neg\neg]^n$ with the formula $\top$ added.

In the proof we will need the (formula) types in $[\wedge, \neg\neg]^n$.

3.4.2.29. Definition. *If k a proper node in an n-maximal Kripke model, then $\phi^n(k)$, the* type *of k in $[\wedge, \neg\neg]^n$, is defined as:*

$$\phi^n(k) = \bigwedge j_0(\tau^n(k)) \wedge \bigwedge \{\neg\neg q \mid q \in \bigcap \{j_0(t) \mid t \in j_1(\tau^n(k))\}\}.$$

3.4.2.30. Theorem. *The model $Umod([\wedge, \neg\neg]^n)$ defined above is a universal model for $[\wedge, \neg\neg]^n$ and the ordered set of semantic types in $[\wedge, \neg\neg]^n$ is an exact model for $[\wedge, \neg\neg, \top]^n$.*

Proof. $Umod([\wedge, \neg\neg]^n)$ clearly is complete for $[\wedge, \neg\neg]^n$ and minimal in realizing all the semantic types of proper nodes in $[\wedge, \neg\neg]^n$.

Hence our main task will be to prove that we realy need all semantic types in $[\wedge, \neg\neg]^n$. First note that if k is a node in the intended universal model with a semantic type in $[\wedge, \neg\neg]^n$, then $k \Vdash \phi^n(k)$. From the definition of $\phi^n(k)$ it is also

clear that in $Umod([\wedge, \neg\neg]^n)$ it is true that $l \Vdash \phi^n(k)$ iff $k \leq l$. Hence, for k such that $\tau^n(k)$ is a semantic type in $[\wedge, \neg\neg]^n$ we have $[\![\phi^n(k)]\!] = \uparrow k$.

Suppose $k_1, \ldots, k_m$ is a close subset in the submodel of the proper nodes in $Umod([\wedge, \neg\neg]^n)$. Let ϕ be the formula

$$\phi = \bigwedge \bigcap_{i=1}^m j_0(\tau^n(k_i)) \wedge \bigwedge \{\neg\neg p \mid p \in \bigcap_{i=1}^m \bigcap \{j_0(t) \mid t \in j_1(\tau^n(k_i))\}\}.$$

Note that if $k_1, \ldots, k_m$ is the set of all proper nodes in $Umod([\wedge, \neg\neg]^n)$, then $\phi = \top$ (which is not a $[\wedge, \neg\neg]^n$ formula).

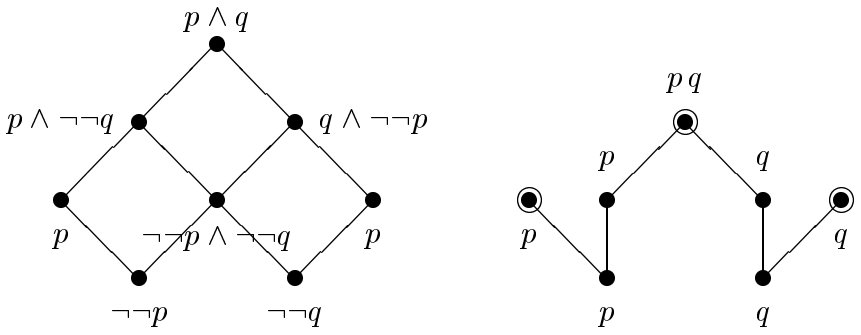
To prove that $[\![\phi]\!] = \uparrow\{k_1, \ldots, k_n\}$, let $k \in \{k_1, \ldots, k_n\}$. Then $\bigcap_{i=1}^m j_0(\tau^n(k_i)) \subseteq j_0(\tau^n(k))$ and $\bigcap_{i=1}^m \bigcap \{j_0(t) \mid t \in j_1(\tau^n(k_i))\} \subseteq \bigcap \{j_0(t) \mid t \in j_1(\tau^n(k))\}$. From which we may infer that $k \Vdash \phi$.

On the other hand, if $k \Vdash \phi$, suppose k is a terminal node in $Umod([\wedge, \neg\neg]^n)$. If $atom^n(k) = \{p_1, \ldots, p_n\}$, then clearly $k_i > k$ for all k_i. If $|atom^n(k)| = n - 1$, then there is exactly one l in $Umod([\wedge, \neg\neg]^n)$ such that $l < k$. Let $q \in \{p_1, \ldots, p_n\} \setminus atom^n(k)$, then $l \in \{k_1, \ldots, k_m\}$ iff $\phi \nvdash \neg\neg q$. Hence, from $k \Vdash \phi$ we conclude $\phi \nvdash \neg\neg q$ and hence $k \in \uparrow\{k_1, \ldots, k_m\}$. In case k is a proper node of $Umod([\wedge, \neg\neg]^n)$, for $q \in \{p_1, \ldots, p_n\} \setminus atom^n(k)$ we have $\phi \nvdash q \quad \Leftrightarrow \quad k \in \{k_1, \ldots, k_m\}$. From $k \Vdash \phi$, we infer that $\phi \nvdash q$ and hence $k \in \{k_1, \ldots, k_m\}$.

Hence, we proved that the ordered set of semantic types in $[\wedge, \neg\neg]^n$ is an exact model for $[\wedge, \neg\neg, \top]^n$. ⊣

3.4.2.31. COROLLARY. *The exact model of $[\wedge, \neg\neg, \top]^n$ is isomorphic with n disjoint copies of* **2** *(disregarding the valuation in $Umod([\wedge, \neg\neg]^n)$).*

Hence the fragment $[\wedge, \neg\neg, \top]^n$ has $3^n - 1$ equivalence classes.



14. FIGURE. *The diagram of $[\wedge, \neg\neg]^2$ and the Kripke completion of its exact model (with the added terminal nodes encircled).*

3.4.3 The $[\wedge, \vee, \neg\neg]$ fragments

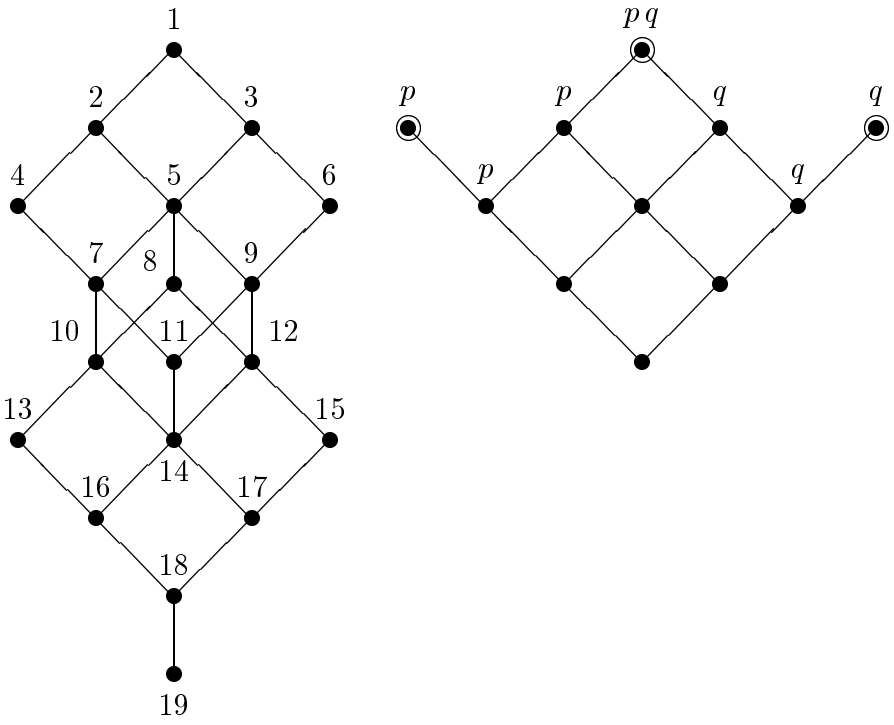
The construction of the exact model of $[\wedge, \vee, \neg\neg]^n$ from the irreducible formulas in this fragment is rather straightforward.

3.4.3.32. Lemma. *The irreducible formulas in* $[\wedge, \vee, \neg\neg]^n$ *are (modulo logical equivalence) of the form* $\bigwedge Q \wedge \neg\neg\psi$*, where* Q *is some subset of* $\{p_1, \ldots, p_n\}$ *and* ψ *is some formula in* $[\wedge, \vee]^n$*.*

Proof. If $\phi = \bigwedge Q \wedge \neg\neg\psi$ and $\phi \nvdash \bot$, then ϕ is $\vee$-irreducible by 3.2.0.3.4. If $\phi \in [\wedge, \vee, \neg\neg]^n$, it is not difficult to prove ϕ to be equivalent to a disjunction of formulas of the form $\bigwedge Q \wedge \neg\neg\psi$. Hence, if ϕ is irreducible, ϕ is equivalent to a formula of the form $\bigwedge Q \wedge \neg\neg\psi$. ⊣

3.4.3.33. Corollary. (The $[\wedge, \vee, \neg\neg]^n$ normal form) *Every formula in* $[\wedge, \vee, \neg\neg]$ *is equivalent to a disjunction of formulas of the form* $\bigwedge Q \wedge \neg\neg\psi$ *where* Q *is some subset of* $\{p_1, \ldots, p_n\}$ *and* ψ *is a formula in* $[\wedge, \vee]^n$*.*

With exception of the fragment with only one atom, these exact models are not exact Kripke models, as can be seen from the example of $[\wedge, \vee, \neg\neg]^2$.



15. Figure. *The diagram of* $[\wedge, \vee, \neg\neg]^2$ *and the Kripke completion of its exact model (the encircled nodes have been added).*

The formulas in the diagram of $[\wedge, \vee, \neg\neg]^2$:

1. $p \wedge q$	8. $\neg\neg(p \wedge q)$	15. $\neg\neg q$
2. $p \wedge \neg\neg q$	9. $(p \vee q) \wedge \neg\neg q$	16. $\neg\neg p \vee q$
3. $\neg\neg p \wedge q$	10. $\neg\neg p \wedge (p \vee \neg\neg q)$	17. $p \vee \neg\neg q$
4. p	11. $p \vee q$	18. $\neg\neg p \vee \neg\neg q$
5. $\neg\neg(p \wedge q) \wedge (p \vee q)$	12. $(\neg\neg p \vee q) \wedge \neg\neg q$	19. $\neg\neg(p \vee q)$
6. q	13. $\neg\neg p$	
7. $\neg\neg p \wedge (p \vee q)$	14. $(p \vee \neg\neg q) \wedge (\neg\neg p \vee q)$	

To find the semantic types in $[\wedge, \vee, \neg\neg]^n$ the normal form of the irreducible formulas suggests the following definitions.

3.4.3.34. Definition. *A finite IpL model K is called a* proper $[\wedge, \vee, \neg\neg]^n$ model *if for no terminal node l it is true that $atom^n(l) = \emptyset$ and for every $k \in K$ which is not a terminal node, there is a terminal $l > k$ with $atom^n(l) = \{p_1, \ldots, p_n\}$.*

3.4.3.35. Definition. *For k a node in a proper $[\wedge, \vee, \neg\neg]^n$ model define $\tau^n(k)$, the* semantic type *of k in $[\wedge, \vee, \neg\neg]^n$, as:*

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset \rangle & \text{if} \quad \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\} \rangle & \text{otherwise} \end{cases}$$

For semantic types t and t' in $[\wedge, \neg\neg]^n$ define:

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

Define $\phi^n(k)$, the type *of k in $[\wedge, \vee, \neg\neg]^n$, as:*

$$\phi^n(k) = \bigwedge j_0(\tau^n(k)) \wedge \neg\neg \bigvee \{\bigwedge j_0(t) \mid t \in j_1(\tau^n(k))\}$$

Observe that for each k in a proper $[\wedge, \vee, \neg\neg]^n$ model which is not a terminal node, we have $k \Vdash \phi^n(k)$.

To prove that each irreducible formula of $[\wedge, \vee, \neg\neg]^n$ is a $\phi^n(k)$ for some non-terminal node k in a proper $[\wedge, \vee, \neg\neg]^n$ model, first note that in the normal form of irreducible formulas in $[\wedge, \vee, \neg\neg]^n$ the part in the scope of $\neg\neg$ is a formula of $[\wedge, \vee]$ and hence needs for its realization terminal nodes l with $atom^n(l) \neq \emptyset$. A second observation we need is that for $\psi \in [\wedge, \vee]^n$ it is always true that $\neg\neg(\psi \vee \bigwedge\{p_1, \ldots, p_n\}) \equiv \neg\neg\psi$.

So we may infer that $[\wedge, \vee, \neg\neg]^n$ is complete for proper $[\wedge, \vee, \neg\neg]^n$ models, as every irreducible formula in the fragment can be realized in one of these models.

3.4.3.36. Fact. *The fragment $[\wedge, \vee, \neg\neg]^n$ is complete for proper $[\wedge, \vee, \neg\neg]^n$ models.*

The ordered set of semantic types in $[\wedge, \vee, \neg\neg]^n$ will provide us with an exact model of $[\wedge, \vee, \neg\neg]^n$.

3.4.3.37. Definition. *Let* $\mathrm{Umod}([\wedge, \vee, \neg\neg]^n)$ *be the Kripke model constructed from the ordered set of semantic types $\langle Q, T \rangle$, such that $Q \subseteq \{p_1, \ldots, p_n\}$ and T a set of semantic types $\langle U, \emptyset \rangle$, such that $\langle \{p_1, \ldots, p_n\}\, \emptyset \rangle \in T$. The valuation in* $\mathrm{Umod}([\wedge, \vee, \neg\neg]^n)$ *is defined by $atom^n(t) = j_0(t)$.*

Obviously this model is a proper $[\wedge,\vee,\neg\neg]^n$ model realizing all semantic types in the fragment and hence it is complete for $[\wedge,\vee,\neg\neg]^n$. However, as an exact model $Umod([\wedge,\vee,\neg\neg]^n)$ is too large. More precisely we do not need the terminal nodes in this model, as observed earlier. Note that the semantic type $\langle\{p_1,\ldots,p_n\},\emptyset\rangle$, corresponding to the type $\bigwedge\{p_1,\ldots,p_n\}$, will only be realized in the model as the type of a terminal node. However as this type acts as a bottom element in the diagram of the fragment, it is not needed in the exact model as it will correspond to the empty set.

3.4.3.38. Theorem. *The model* $\mathrm{Umod}([\wedge,\vee,\neg\neg]^n)$ *without its terminal nodes is the exact model of* $[\wedge,\vee,\neg\neg]^n$.

Proof. As we have seen, every non-terminal element $\langle Q,T\rangle$ in $Umod([\wedge,\vee,\neg\neg]^n)$ corresponds to a type $\bigwedge Q\wedge\neg\neg\bigvee\{\bigwedge j_0(t)\mid t\in T\}$ and every irreducible formula in $[\wedge,\vee,\neg\neg]^n$ is equivalent to such a type. The only thing we still have to prove is that different semantic types indeed have different type formulas. This we may infer from the fact that for t and t' semantic types in $[\wedge,\vee,\neg\neg]^n$ and ϕ_t and $\phi_{t'}$ the corresponding type formulas it is true that $t\preceq t' \;\Leftrightarrow\; \phi_{t'}\vdash\phi_t$. The proof of this last fact is straightforward from the definitions and is left to the industrious reader. ⊣

3.4.4 The $[\vee,\neg]$ fragments

As was already mentioned before, a fragment $[\vee,\neg]^n$ has an exact model. For $n>1$ this is not an exact Kripke model, as we will see.

The fragment $[\vee,\neg]^n$ is a subfragment of $[\wedge,\vee,\neg]^n$, for which we already defined semantic types and constructed an exact Kripke model. As the irreducible formulas in $[\vee,\neg]^n$ are also irreducible in $[\wedge,\vee,\neg]^n$ we have already met the (semantic) types in $[\vee,\neg]^n$: those types in $[\wedge,\vee,\neg]^n$ which are equivalent to a formula in $[\vee,\neg]^n$ (and the corresponding semantic types).

Of course the only irreducible formulas in $[\vee,\neg]^n$ are the atomic formulas and the negations. Note also that conjunctions of negations are equivalent to negations of disjunctions (i.e. $\neg p\wedge\neg q\equiv\neg(p\vee q)$), and thus are part of the fragment.

Recall that semantic types in $[\wedge,\vee,\neg]^n$ are of the form:

$$\tau^n(k)=\begin{cases}\langle atom^n(k),\emptyset\rangle & \text{if}\quad \forall l>k.\,atom^n(l)=atom^n(k)\\ \langle atom^n(k),\{\tau^n(l)\mid l\in Ter(k)\}\rangle & \text{otherwise}\end{cases}$$

The corresponding type $\phi^n(k)$ was defined as:

$$\phi^n(k)=\begin{cases}\phi^n_{CpL}(k) & \text{if } \forall l>k.\,atom^n(l)=atom^n(k)\\ \bigwedge j_0(\tau^n(k))\wedge & \\ \neg\neg\bigvee\{\phi^n_{CpL}(l)\mid \tau^n(l)\in j_1(\tau^n(k))\} & \text{otherwise}\end{cases}$$

If a formula type $\phi^n(k)$ in $[\wedge,\vee,\neg]^n$ corresponds to a formula in $[\vee,\neg]^n$ then

1. $atom^n(k)$ is either empty or a singleton;
2. k if $atom^n(k) \neq \emptyset$ and $n > 1$ then k is not a terminal node.

Note that in the first case, where $\phi^n(k)$ is equivalent to an atomic formula p, we have $\bigvee\{\phi^n_{CpL}(l) \mid \tau^n(l) \in j_1(\tau^n(k))\} \equiv p$. Hence, for all $Q \in \{p_1, \ldots, p_n\}$ such that $p \in Q$, there is a $t \in j_1(\tau^n(k))$ with $j_0(t) = Q$. Otherwise, we would have $k \Vdash \neg\phi^n_Q$, which contradicts $\phi^n(k) \equiv p$, as $p \nvdash \neg\phi^n_Q$.

As it is not difficult to see that every formula in $[\wedge, \vee, \neg]^1$ is equivalent to a formula in $[\vee, \neg]^1$ (see figure 9), the two fragments have the same diagram. In the sequel of this subsection we will assume $n > 1$, without making this exception explicit every time we should.
The observations above inspired the definition of a semantic type in $[\vee, \neg]^n$.

3.4.4.39. Definition. *Let k be a node in a finite IpL model. Then k is a* proper $[\vee, \neg]^n$ node *if $atom^n(k) = \emptyset$ or $atom^n(k) = \{p\}$ for some $p \in \{p_1, \ldots, p_n\}$ and for every $Q \subseteq \{p_1, \ldots, p_n\}$ such that $p \in Q$ there is an $l \in Ter(k)$ with $atom^n(l) = Q$.*

If k is a proper $[\vee, \neg]^n$ node, then $\tau^n(k)$, the semantic type *of k in $[\wedge, \vee, \neg]^n$ is the semantic type of k in $[\vee, \neg]^n$ and $\phi^n(k)$, the type formula of k in $[\wedge, \vee, \neg]^n$, is the type formula of k in $[\vee, \neg]^n$.*

Let $Th^n(k)$ in the sequel of this subsection be the $[\vee, \neg]^n$ theory of k, $Th^n(k) = \{\phi \in [\vee, \neg]^n \mid k \Vdash \phi\}$.

To see that $\phi^n(k)$ is a formula in $[\vee, \neg]^n$ note that either $atom^n(k) = \{p\}$ for some atom p or else $\phi^n(k)$ is a negation.

3.4.4.40. Lemma. *If k and l are nodes in finite Kripke models and k and l have semantic types in $[\vee, \neg]^n$, then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k)$$

Proof. The lemma is an application of lemma 3.4.0.5, in the special case where k and l have semantic types in $[\vee, \neg]^n$ and the theories $Th^n(k), Th^n(l)$ and the formula $\phi^n(k)$ are in $[\vee, \neg]^n$. ⊣

By ordering the semantic types in $[\vee, \neg]^n$, adding the terminal nodes which have no type in $[\vee, \neg]^n$, we get a Kripke completion of the exact model of $[\vee, \neg]^n$ as we will see.

3.4.4.41. Definition. *Let* $\mathrm{Umod}([\vee, \neg]^n) = \langle T, \preceq, j_0\rangle$ *be the Kripke model constructed from the set T of both semantic types in $[\vee, \neg]^n$ and types of the form $\langle Q, \emptyset\rangle$, where Q a nonempty subset of $\{p_1, \ldots, p_n\}$. The order relation between these types is defined as*

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

The valuation in $\mathrm{Umod}([\vee, \neg]^n)$ *is defined by $atom^n(t) = j_0(t)$.*

From the definitions and observations above the following fact is a simple consequence.

3.4.4.42. FACT. *The model* $\mathrm{Umod}([\vee,\neg]^n)$ *realizes each semantic type in* $[\vee,\neg]^n$.

3.4.4.43. THEOREM. *The model* $\mathrm{Umod}([\vee,\neg]^n)$ *is a universal model for the fragment* $[\vee,\neg]^n$ *and the ordered set of semantic types in* $[\vee,\neg]^n$ *is the exact model of* $[\vee,\neg]^n$.

Proof. The irreducible formula classes of $[\vee,\neg]^n$ correspond exactly to the semantic types in $[\vee,\neg]^n$. If X is an upwardly closed subset of types $\{\tau^n(k_1),\ldots,\tau^n(k_m)\}$, this X will correspond to the formula $\bigvee\{\phi^n(k_1),\ldots,\phi^n(k_m)\}$ (where $\bigvee\emptyset=\bot$).

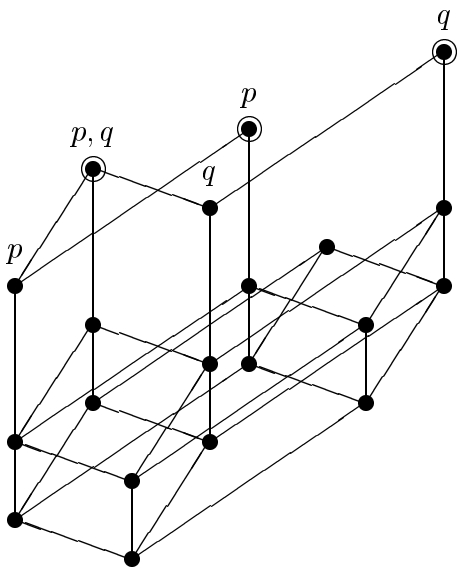
To realize the semantic types in $[\vee,\neg]^n$ one obviously needs the terminal nodes forcing non-empty sets of atoms. As these are the only elements added in $Umod([\vee,\neg]^n)$, this model is a minimal Kripke completion of the exact model. ⊣

The structure of the exact model of $[\vee,\neg]^n$ might also be described as that of the ordered set of all non-contradictory negations where the atomic formulas are added.

The set of non-contradictory negations is isomorphic to the diagram of the classical diagram $[\vee,\neg]^n_{CpL}$ without the tautology and hence also with the 2^n-dimensional hypercube without a top.

Hence the universal model of $[\vee,\neg]^n$ will be an n-dimensional hypercube without a top, where at n corners there have been added nodes that force just one atom and which have been connected to the 2^n-1 terminal nodes outside the exact model.

As an illustration, figure 16 shows the universal model of $[\vee,\neg]^2$.



16. FIGURE. *The universal model of* $[\vee,\neg]^2$. *(The encircled nodes have been added to the exact model.)*

Using this model one can compute the diagram of $[\vee,\neg]^2$ which has 385 equivalent classes.

3.4.5 The $[\vee, \neg\neg]$ fragments

With exception of $[\vee, \neg\neg]^1$, the fragments $[\vee, \neg\neg]^n$ will not have an exact model. Note that the minimal elements in the diagram of $[\vee, \neg\neg]^n$ are the atomic formulas and for $n > 1$ there will not be a bottom in $Diag([\vee, \neg\neg]^n)$. Hence the diagram of $[\vee, \neg\neg]^n$ is not a lattice.

But by adding $\bot$ to $[\vee, \neg\neg]^n$ we will have a fragment with an exact Kripke model as we will see.

The fragment $[\vee, \neg\neg]^1$ has a simple diagram (two classes, p and $\neg\neg p$) and has an exact model (with $\neg\neg(p)$ as its only element and p corresponding to the empty set) which is not an exact Kripke model. In the sequel of this subsection we will assume $n > 1$.

There is a simple normal form in $[\vee, \neg\neg]^n$ which we will use in the construction of this exact Kripke model of $[\vee, \neg\neg, \bot]^n$.

3.4.5.44. Fact. (The $[\vee, \neg\neg]^n$ normal form) *Every formula in $[\vee, \neg\neg]^n$ is equivalent to a disjunction of formulas that are either atomic or of the form $\neg\neg\psi$, where ψ is a disjunction of atomic formulas.*

This fact can be straightforwardly proved by induction on the length of the formula.

To characterize the semantic types in $[\vee, \neg\neg]^n$ we will introduce a special type of *IpL* models, as we did previously for $[\wedge, \neg\neg]^n$.

3.4.5.45. Definition. *A finite n-model K is called* n-minimal *if each node in K forces at most one atom and each terminal node forces at least one atom.*

3.4.5.46. Theorem. *If ϕ and ψ are formulas in $[\vee, \neg\neg]^n$ such that $\phi \nvdash \psi$ then there is a node k in an n-minimal model such that $k \Vdash \phi$ and $k \nVdash \psi$.*

Proof. Let $\bigvee P \vee \bigvee\neg\neg\bigvee Q_i$ be the normal form of ϕ and $\bigvee R \vee \bigvee\neg\neg\bigvee S_j$ the normal form of ψ (where P, Q_i, R and the S_j are subsets of $\{p_1, \ldots, p_n\}$). As $\phi \nvdash \psi$ we have either some $p \in P$ which is not an element of R or some Q_i which is not a subset of any of the S_j.

In the first case a simple model of one node k such that $atom^n(k) = \{p\}$ is sufficient as a counter-example n-minimal model.

In the second case, let K be the n-minimal model with a root k_0 such that $atom^n(k_0) = \emptyset$ and terminal nodes k_q for every $q \in Q_i$. Then $k_0 \Vdash \neg\neg\bigvee Q_i$ but $k_0 \nVdash \bigvee R$ and also for every S_j, as Q_i is not a subset of S_j, we will have $k_0 \nVdash \neg\neg\bigvee S_j$. Hence $k_0 \Vdash \phi$ and $k_0 \nVdash \psi$ as required. $\dashv$

3.4.5.47. Corollary. *The fragment $[\vee, \neg\neg]^n$ is complete for n-minimal models.*

As we may confine our attention to n-minimal models in constructing an exact Kripke model for $[\vee, \neg\neg, \bot]^n$ we will use them in the definition of semantic types in $[\vee, \neg\neg]^n$.

3.4.5.48. Definition. *Let k be a node in an n-minimal model. Then $\tau^n(k)$, the semantic type of k in $[\wedge, \vee, \neg]^n$ is the* semantic type *of k in $[\vee, \neg\neg]^n$ and $\phi^n(k)$, the*

type formula of k in $[\wedge,\vee,\neg]^n$ is the type *of k in $[\vee,\neg\neg]^n$. The order of the semantic types in $[\vee,\neg\neg]^n$ is defined by:*

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

Note that for types in $[\vee,\neg\neg]^n$ we will have $\tau^n(k) \preceq \tau^n(l)$ if $atom^n(k) = atom^n(l)$ or $atom^n(k) = \emptyset$ and $\langle atom^n(l), \emptyset\rangle \in j_1(\tau^n(k))$.

In the definition of $\phi^n(k)$, the type of k in $[\vee,\neg\neg]^n$ one will recognize the normal form of the irreducible elements in the fragment.

Note that as k is a node in an n-minimal model $\phi^n(k)$ is either equivalent to an atomic formula or to $\neg\neg\bigvee Q$ where Q is the set of atoms forced in the terminal nodes above k. Hence modulo equivalence $\phi^n(k)$ is indeed a formula in $[\vee,\neg\neg]^n$.

Let in this subsection $Th^n(k)$ be the notation for the formulas in $[\vee,\neg\neg]^n$ forced by k.

3.4.5.49. Lemma. *If k and l are nodes in n-minimal models then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k).$$

Proof. That $\tau^n(k) \preceq \tau^n(l)$ implies $Th^n(k) \subseteq Th^n(l)$ is a straightforward application of lemma 3.4.0.5, on n-minimal models. As $k \Vdash \phi^n(k)$, obviously, $Th^n(k) \subseteq Th^n(l)$ implies $l \Vdash \phi^n(k)$

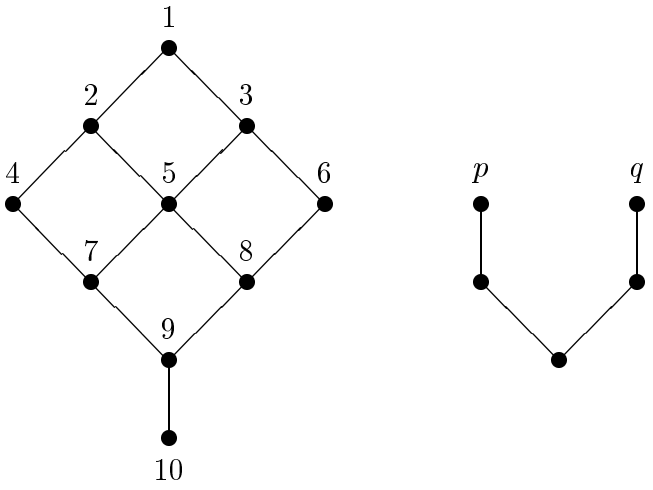
To prove that $l \Vdash \phi^n(k)$ implies $\tau^n(k) \preceq \tau^n(l)$, let $l \Vdash \phi^n(k)$. As a simple consequence we have $j_0(\tau^n(k)) \subseteq j_0(\tau^n(l))$. Hence, if $atom^n(k) = \{q\}$, then also $atom^n(l) = \{q\}$, as both k and l are nodes in n-minimal models. As a consequence, $atom^n(k) = \{q\}$ implies $\tau^n(k) = \tau^n(l)$. So, assume $atom^n(k) = \emptyset$, by the definition of an n-minimal model, k cannot be a terminal node. Observe, that $\phi^n(k)$ is a negation, equivalent to $\neg\neg\bigvee\{\phi^n(m) \mid m \in Ter(k)\}$.

If $atom^n(l) = \{q\}$, then, as l is a node in an n-minimal model, $\tau^n(l) = \langle\{q\},\emptyset\rangle$ and from $l \Vdash \phi^n(k)$ we may conclude $l \Vdash \bigvee\{\phi^n(m) \mid m \in Ter(k)\}$ and hence $l \Vdash \phi^n(m)$, for some $m \in Ter(k)$. This proves that $\tau^n(l) \in j_1(\tau^n(k))$.

On the other hand, if $atom^n(l) = \emptyset$, then l cannot be (bisimular to) a terminal node in an n-minimal model. Hence $j_0(\tau^n(l)) \neq \emptyset$. To prove that $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$, and hence $\tau^n(k) \preceq \tau^n(l)$, let $m \in Ter(l)$. Suppose $atom^n(m) = \{q\}$, then, as $m \Vdash \phi^n(k)$, $\phi^n(k) \nvdash \neg q$. From the definition of $\phi^n(k)$ infer that there is a $k' \in Ter(k)$ with $atom^n(k') = \{q\}$. Hence, $\tau^n(m) = \tau^n(k') \in j_1(\tau^n(k))$. So, we have $atom^n(k) = atom^n(l)$ and $\emptyset \neq j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. Which, by definition, implies $\tau^n(k) \preceq \tau^n(l)$. ⊣

By ordering the semantic types in $[\vee,\neg\neg]^n$ we construct a n-minimal model $Exm([\vee,\neg\neg,\bot]^n)$ which consists of n terminal nodes, each forcing one of the atoms in $\{p_1,\ldots,p_n\}$ and non-terminal nodes that force no atomic formulas but are characterized by their set of terminal nodes.

Note that for $n > 1$ we added $\bot$ to the fragment to have a formula corresponding to the empty set in the exact model.



17. FIGURE. *The diagram of $[\vee, \neg\neg, \bot]^2$ and its exact Kripke model.*

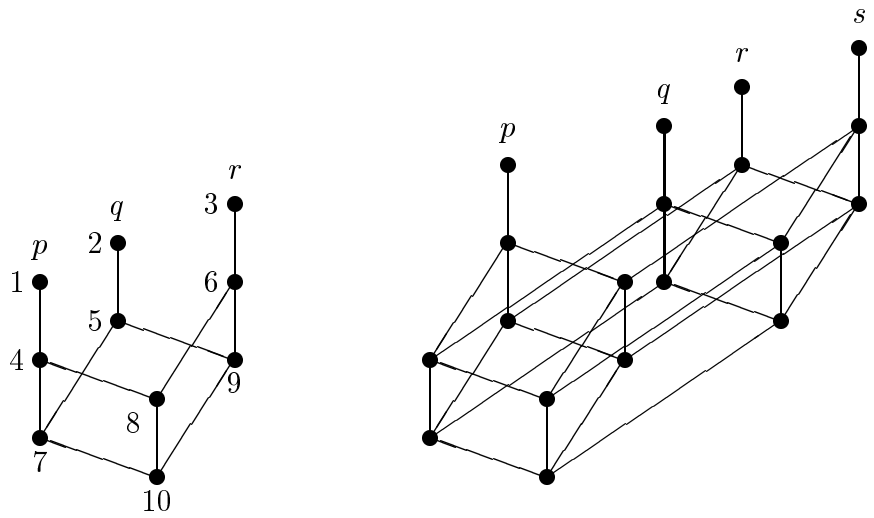
The formulas in the diagram of $[\vee, \neg\neg, \bot]^2$:

1. $\bot$	3. q	5. $p \vee q$	7. $\neg\neg p \vee q$	9. $\neg\neg p \vee \neg\neg q$
2. p	4. $\neg\neg p$	6. $\neg\neg q$	8. $p \vee \neg\neg q$	10. $\neg\neg(p \vee q)$

3.4.5.50. THEOREM. *The model $Exm([\vee, \neg\neg, \bot]^n)$ is the exact Kripke model of the fragment $[\vee, \neg\neg, \bot]^n$.*

Proof. Because $Exm([\vee, \neg\neg, \bot]^n)$ realizes all semantic types in $[\vee, \neg\neg]^n)$, the model is complete for the fragment. As we have seen, every semantic type corresponds to a type formula in $[\vee, \neg\neg, \bot]^n)$ and hence every non-empty closed subset corresponds exactly to a disjunction of these type formulas in $[\vee, \neg\neg, \bot]^n$. For the empty set the formula $\bot$ was added. $\dashv$

The construction of $Exm([\vee, \neg\neg, \bot]^n)$ shows that the non-terminal nodes correspond to non-empty subsets of $\{p_1, \ldots, p_n\}$ ordered by inclusion. Hence $Exm([\vee, \neg\neg, \bot]^n)$ is isomorphic to the the n-dimensional hypercube without a top, where the n maximal elements are connected to terminal nodes each forcing one of the atoms in $\{p_1, \ldots, p_n\}$.



18. FIGURE. *The exact Kripke models of* $[\vee, \neg\neg, \bot]^3$ *and* $[\vee, \neg\neg, \bot]^4$

As an example, we give the type formulas of $[\vee, \neg\neg, \bot]^3$.

1. p
2. q
3. r
4. $\neg\neg p$
5. $\neg\neg q$
6. $\neg\neg r$
7. $\neg\neg(p \vee q)$
8. $\neg\neg(p \vee r)$
9. $\neg\neg(q \vee r)$
10. $\neg\neg(p \vee q \vee r)$

Recall from subsection 3.3 that $D(k)$ is the k-th Dedekind number.

3.4.5.51. THEOREM.

$$|Diag([\vee, \neg\neg, \bot]^n)| = \sum_{k=0}^{n} \binom{n}{k} (D(k) + 1).$$

Proof. Observe that every formula in $[\vee, \neg\neg, \bot]^n$ is equivalent to a disjunction of atoms and formulas of the form $\neg\neg\bigvee R$ (where $\bigvee\emptyset = \bot$). Let $Q \subseteq \{p_1, \ldots, p_n\}$ and $|Q| = k$. It is not difficult to see that the set of formulas of the form $\neg\neg\bigvee R$, with $R \neq \emptyset$ and $R \subseteq \{p_1, \ldots, p_n\}$, ordered by $\vdash$, is isomorphic to $Diag([\vee]^n)$ (or equivalently $Diag([\wedge])$). Hence, the number of equivalence classes in $[\vee, \neg\neg, \bot]^n$ with a representative of the form $\bigvee Q \vee \neg\neg\bigvee R$ with $R \setminus Q \neq \emptyset$ equals $D(n-k)$.

As there are $\binom{n}{k}$ subsets $Q \subseteq \{p_1, \ldots, p_n\}$ with $|Q| = k$, we have, taking in account the cases where $R = \emptyset$:

$$|Diag([\vee, \neg\neg, \bot]^n)| = \sum_{k=0}^{n} \binom{n}{k} (D(n-k) + 1).$$

Now use $\binom{n}{k} = \binom{n}{n-k}$ to obtain the formula of the theorem. ⊣

3.5 The $[\wedge, \to, \neg]$ fragments

The diagrams of $[\wedge, \to, \neg]$ fragments have been studied by De Bruijn using exact models in [Bruijn 75a] as a special case of $[\wedge, \to]$ fragments[3]. Briefly stated the $[\wedge, \to, \neg]^n$ fragment is like a $[\wedge, \to]^{n+1}$ fragment, where one of the atoms is treated as $\bot$ (and hence $p_{n+1} \to \bigwedge\{p_1, \ldots, p_n\}$ will be true).

Using semantic types we may start with the $[\wedge, \to, \neg]$ fragments as the more 'natural' fragment.

Note that it is not trivial that the diagram of $[\wedge, \to, \neg]^n$ is a finite distributive lattice. Diego proved in [Diego 66] that the $[\wedge, \to]^n$ fragments are finite (and from the proof one could also infer that the diagram would be distributive). Using the above cited enbedding of $[\wedge, \to, \neg]^n$ into $[\wedge, \to]^{n+1}$ this implies that also $[\wedge, \to, \neg]^n$ will have a finite diagram.

As for the lattice operations in the diagram of $[\wedge, \to, \neg]^n$, it will be obvious how $\wedge$ will act as $\cap$, but for $\cup$ there is no simple operation in $[\wedge, \to, \neg]^n$ as $\vee$ is not definable in terms of $\{\wedge, \to, \neg\}$. Hence in the following subsections the reader should be cautious in not taking the irreducible formulas in (subfragments of) $[\wedge, \to, \neg]^n$ as $\vee$-irreducible formulas.

To define the semantic types in $[\wedge, \to, \neg]$ fragments we will restrict our models to the $\cap$-*independent models*.

3.5.0.1. Definition. *Let K be a finite* **IpL***-model model and $k \in K$. k is* $\cap$-independent *if k is a* terminal node *or:*

$$atom(k) \neq \bigcap\{atom(l) \mid k < l\}.$$

A finite **IpL***-model K is* $\cap$-independent *if every node $k \in K$ is $\cap$-independent.*

Observe that, in a $\cap$-independent n-model, $k < l$ implies $atom^n(k) \neq atom^n(l)$.

As is not difficult to see, $\cap$-independentness is preserved under taking submodels.

3.5.0.2. Definition. *Let K be a finite* **IpL***-model. The* $\cap$-independent reduction *of K is the model $K^\cap$, with the $\cap$-independent nodes of K as its worlds and its accessibility relation inherited from K.*

Note that, as $\cap$-independentness is preserved by taking submodels, the $\cap$-independent reduction is an $\cap$-independent model.

3.5.0.3. Theorem. *Let K be a finite* **IpL***-model and $k \in K$, then for all formulas $\phi \in [\wedge, \to, \neg]$:*

$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^\cap \Vdash \phi.$$

Proof. By induction on $\delta(k)$. If $\delta(k) = 0$, then the theorem is trivial. So assume $\delta(k) = m + 1$ and apply induction on the length of ϕ. The only interesting case is $\phi = \psi \to \chi$ (including negation as a special case).

[3] This was also the approach in [Hendriks 93]

For the proof in the $\Rightarrow$-direction, let $k \Vdash \psi{\rightarrow}\chi$. To prove $(\uparrow k)^{\cap} \Vdash \psi{\rightarrow}\chi$, let $l \in (\uparrow k)^{\cap}$ and $(\uparrow l)^{\cap} \Vdash \psi$. By the induction hypothesis, $l \Vdash \psi$. As $k \leq l$, we conclude $l \Vdash \chi$ and, again by the induction hypothesis, $(\uparrow l)^{\cap} \Vdash \chi$. Which proves $(\uparrow k)^{\cap} \Vdash \psi{\rightarrow}\chi$

For the proof in the $\Leftarrow$-direction, let $(\uparrow k)^{\cap} \Vdash \psi{\rightarrow}\chi$. To prove $k \Vdash \psi{\rightarrow}\chi$, let $k \leq l$ and $l \Vdash \psi$. By the induction hypothesis, $(\uparrow l)^{\cap} \Vdash \psi$ and as obviously $(\uparrow l)^{\cap} \subseteq (\uparrow k)^{\cap}$, $(\uparrow l)^{\cap} \Vdash \chi$ and, again by the induction hypothesis, $l \Vdash \chi$. Which proves $k \Vdash \psi{\rightarrow}\chi$. $\dashv$

The converse of theorem 3.5.0.3 also holds, as theorem 3.5.0.24 below proves.

3.5.0.4. Theorem. *The $[\wedge, \rightarrow, \neg]$ fragment is complete for $\cap$-independent* **IpL** *models.*

Proof. Obvious using theorem 3.5.0.3. $\dashv$

Now we are ready to define the semantic types for $[\wedge, \rightarrow, \neg]^n$ of nodes in $\cap$-independent **IpL**-models.

3.5.0.5. Definition. *For k a node in a finite $\cap$-independent n-model, we define $\tau^n(k)$, the* semantic type *of k in $[\wedge, \rightarrow, \neg]^n$ as:*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l\}\rangle.$$

If t and t' are semantic types in $[\wedge, \rightarrow, \neg]^n$ then define $t \preceq t'$ if $t = t'$ or $t' \in j_1(t)$.

The definition is sound, as in $\cap$-independent n-models $k < l$ implies $atom^n(k) \neq atom^n(l)$ and hence, it is excluded that $\tau^n(k) \in j_1(\tau^n(k))$.

The following simple lemma proves that semantic types in $[\wedge, \rightarrow, \neg]^n$ indeed behave as expected.

3.5.0.6. Lemma. *For nodes k and l in finite $\cap$-independent n-models, define $k \sim l$ if $\tau^n(k) = \tau^n(l)$. Then $\sim$ is a bisimulation.*

Proof. That $k \sim l$ implies $atom^n(k) = atom^n(l)$ is trivial. As the other conditions in definition 2.2.0.1 are symmetric, we only prove one of them. Let $k < k'$ then, by definition, $\tau^n(k') \in j_1(\tau^n(k))$. From $j_1(\tau^n(k)) = j_1(\tau^n(l))$ infer that there is a $l' > l$ such that $\tau^n(k') = \tau^n(l')$. $\dashv$

3.5.0.7. Corollary. *If k and l are nodes in finite $\cap$-independent n-models then $\tau^n(k) \preceq \tau^n(l)$ implies $Th^n(k) \subseteq Th^n(l)$.*

Proof. If $\tau^n(k) = \tau^n(l)$ then we have $k \overleftrightarrow{\sim}^n l$ and hence $Th^n(k) = Th^n(l)$. Otherwise, from $\tau^n(l) \in j_1(\tau^n(k))$ infer that there is a $k' > k$ such that $\tau^n(k') = \tau^n(l)$ and hence $Th^n(k) \subseteq Th^n(k') = Th^n(l)$. $\dashv$

We are almost ready now to introduce the exact Kripke model of $[\wedge, \rightarrow, \neg]^n$ as the ordered set of semantic types in $[\wedge, \rightarrow, \neg]^n$.

First however we have to prove that there are only finitely many semantic types in $[\wedge, \rightarrow, \neg]^n$. To do so we use the following lemma.

3.5.0.8. LEMMA. *If t is a semantic type in $[\wedge, \to, \neg]^n$, then there is a node k in a finite $\cap$-independent n-model such that $\tau^n(k) = t$ and $\delta(k) \leq n - |j_0(t)|$.*

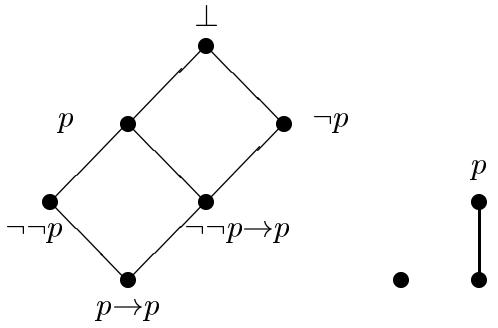
Proof. By a simple induction on $d = n - |j_0(t)|$. For $d = 0$ observe that $j_0(t) = \{p_1, \ldots, p_n\}$. As t has to be a semantic type of a node in an $\cap$-independent n-model, we may infer that $j_1(t) = \emptyset$.

If $d > 0$ then by induction hypothesis every $t' \in j_1(t)$ is realizable by a node $k_{t'}$ with depth at most $d - 1$ (again using the fact that t must be a type of a node in an $\cap$-independent n-model). Of course if a node k with $atom^n(k) = j_0(t)$ is put below all of these $k_{t'}$ (with $t' \in j_1(t)$), then $\tau^n(k) = t$ and $\delta(k) \leq d$. ⊣

3.5.0.9. COROLLARY. *There are finitely many semantic types in $[\wedge, \to, \neg]^n$.*

Proof. Note that there are only finitely many $\cap$-independent n-models with depth less or equal than n and every semantic type in $[\wedge, \to, \neg]^n$ can be realized in one of these models. ⊣

3.5.0.10. DEFINITION. *If T is the set of semantic types in $[\wedge, \to, \neg]^n$ then define $Exm([\wedge, \to, \neg]^n) = \langle T, \preceq, j_0 \rangle$.*



19. FIGURE. *The diagram of $[\wedge, \to, \neg]^1$ and the model $Exm([\wedge, \to, \neg]^1)$.*

Note that we somewhat prematurely baptized the model defined as an exact model, but we will prove this claim in due course. First there are some simple facts to be arrested. If ϕ a formula in $[\wedge, \to, \neg]^n$, then $[\![\phi]\!]$ will be the valuation of ϕ in $Exm([\wedge, \to, \neg]^n)$ (hence $[\![\phi]\!] = \{k \in Exm([\wedge, \to, \neg]^n) \mid k \Vdash \phi\}$).

3.5.0.11. FACTS.

1. *$Exm([\wedge, \to, \neg]^n)$ is a finite $\cap$-independent n-model;*
2. *if t a semantic type in $[\wedge, \to, \neg]^n$ then $\tau^n(t) = t$ in $Exm([\wedge, \to, \neg]^n)$;*
3. *$Exm([\wedge, \to, \neg]^n)$ is complete for $[\wedge, \to, \neg]^n$: if ϕ and ψ in $[\wedge, \to, \neg]^n$ we have*

$$\phi \vdash \psi \quad \Leftrightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!].$$

The first two of these facts are established by a close inspection of the construction of $Exm([\wedge,\rightarrow,\neg]^n)$ from the semantic types in $[\wedge,\rightarrow,\neg]^n$. The last one is a simple consequence of the lemmas above.

To prove $Exm([\wedge,\rightarrow,\neg]^n)$ to be the exact model of $[\wedge,\rightarrow,\neg]^n$, we need a formula in $[\wedge,\rightarrow,\neg]^n$ for every closed subset of $Exm([\wedge,\rightarrow,\neg]^n)$. Unfortunately there is no known simple construction for such a formula, which is independent of the construction of the exact model. However, there is an easy way out.

3.5.0.12. Lemma. *The diagram of* $[\wedge,\rightarrow,\neg]^n$ *is finite.*

Proof. Observe that the equivalence class of a $[\wedge,\rightarrow,\neg]^n$ formula ϕ corresponds to $[\![\phi]\!]$ in $Exm([\wedge,\rightarrow,\neg]^n)$. That is $\phi \equiv \psi$ iff $[\![\phi]\!] = [\![\psi]\!]$. As $Exm([\wedge,\rightarrow,\neg]^n)$ is finite, there are only finitely many equivalence classes in $[\wedge,\rightarrow,\neg]^n$. $\dashv$

De Bruijn proved the finiteness of $Diag([\wedge,\rightarrow,\neg]^n)$ in [Bruijn 75a]. Diego and Urquhart independently proved that $Diag([\rightarrow]^n)$ is finite (see [Diego 66] and [Urquhart 74]), from which the finiteness of $Diag([\wedge,\rightarrow,\neg]^n)$ is a simple corollary. Observe, that using $p \wedge q{\rightarrow}r \wedge s \equiv (p{\rightarrow}(q{\rightarrow}r)) \wedge (p{\rightarrow}(q{\rightarrow}s))$, we can prove that every formula in $[\wedge,\rightarrow,\neg]^n$ is a conjunction of formulas in $[\rightarrow,\neg]^n$. As obviously $|Diag([\rightarrow,\neg]^n)| \leq |Diag([\rightarrow]^{n+1})|$, the finiteness of $Diag([\rightarrow]^n)$ (for each n) implies that $Diag([\wedge,\rightarrow,\neg]^n)$ is finite.

3.5.0.13. Corollary. *For every node* k *in a finite* $\cap$*-independent* n*-model there are, up to equivalence, only finitely many formulas of* $[\wedge,\rightarrow,\neg]^n$ *in* $Th^n(k)$.

The corollary justifies the following definitions.

3.5.0.14. Definition. *For a node* k *in a finite* $\cap$*-independent* n*-model define,* $\phi^n(k)$*, the* type *of* k *in* $[\wedge,\rightarrow,\neg]^n$ *as*

$$\phi^n(k) = \bigwedge Th^n(k).$$

As stated earlier, this is an easy way out and we will return to the construction of type formulas in the sequel. Obviously $\phi^n(k)$ is an axiom for $Th^n(k)$ and $\tau^n(k) \preceq \tau^n(l)$ then $l \Vdash \phi^n(k)$.

3.5.0.15. Lemma. *Let* k *and* l *be nodes in finite* $\cap$*-independent* n*-models. If* $l \Vdash \phi^n(k)$ *then* $\tau^n(k) \preceq \tau^n(l)$.

Proof. Assume $l \Vdash \phi^n(k)$. To prove $\tau^n(k) \preceq \tau^n(l)$ we will use induction on $\delta(l)$, the depth of l. If $\delta(l) = 0$ then $k \nVdash \neg\phi^n_{\mathbf{CpL}}(l)$ and hence there is a $k' > k$ such that $k' \Vdash \phi^n_{\mathbf{CpL}}(l)$. In a $\cap$-independent n-model we may infer that k' has to be a terminal node and hence $\tau^n(k') = \tau^n(l)$. Which proves $\tau^n(k) \preceq \tau^n(l)$.

If $\delta(l) > 0$, let $l' > l$. Then $l' \Vdash \phi^n(k)$ and $\delta(l') < \delta(l)$. According to the induction hypothesis we will have $\tau^n(k) \preceq \tau^n(l')$. This proves that $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. As obviously the assumption implies that $atom^n(k) \subseteq atom^n(l)$, this proves $\tau^n(k) \preceq \tau^n(l)$. $\dashv$

3.5.0.16. COROLLARY. *If k and l are nodes in finite $\cap$-independent n-models then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k).$$

Proof. Corollary 3.5.0.7 takes care of $\tau^n(k) \preceq \tau^n(l) \quad \Rightarrow \quad Th^n(k) \subseteq Th^n(l)$. As $\phi^n(k)$ is the axiom of $Th^n(k)$, obviously $Th^n(k) \subseteq Th^n(l)$ implies $l \Vdash \phi^n(k)$. Finally, $l \Vdash \phi^n(k) \quad \Rightarrow \quad \tau^n(k) \preceq \tau^n(l)$ by lemma 3.5.0.15. $\dashv$

With corollary 3.5.0.16 we are ready to prove that $Exm([\wedge, \to, \neg]^n)$ is indeed the exact Kripke model we were looking for.

3.5.0.17. THEOREM. *The model $Exm([\wedge, \to, \neg]^n)$ defined above is the exact Kripke model of $[\wedge, \to, \neg]^n$.*

Proof. As noted before, $Exm([\wedge, \to, \neg]^n)$ is complete for $[\wedge, \to, \neg]^n$ and we have to prove that every closed subset X in this model corresponds to a formula in $[\wedge, \to, \neg]^n$.

To do so we apply essentially use the same trick that was used to define the types of nodes in $[\wedge, \to, \neg]^n$. Let $\phi^n(X) = \bigwedge\bigcap\{Th^n(k) \mid k \in X\}$. Then clearly, by definition, it will be true that $X \subseteq [\![\phi^n(X)]\!]$.

To prove the inclusion in the other direction, suppose that $k \Vdash \phi^n(X)$. With induction on $\delta(k)$, the depth of k, we will prove that $k \in X$. If $\delta(k) = 0$ then k is a terminal node and apparently it is the case that for some $l \in X$ we had $l \nVdash \neg\phi^n_{\mathbf{CpL}}(k)$. As otherwise $\phi^n(X)$ would imply $\neg\phi^n_{\mathbf{CpL}}(k)$. Hence for some $l \in X$ there is a $l' > l$ with $l' \Vdash \phi^n_{\mathbf{CpL}}(k)$. As $Exm([\wedge, \to, \neg]^n)$ is a $\cap$-independent n-model, this l' has to be a terminal node. As the semantic types in $Exm([\wedge, \to, \neg]^n)$ are unique, we conclude that $k = l'$. From $l \in X$ and $l < k$ infer that $k \in X$ as X is a closed subset of $Exm([\wedge, \to, \neg]^n)$.

If $\delta(k) > 0$ then for $k' > k$ we conclude from $k' \Vdash \phi^n(X)$ and the induction hypothesis that $k' \in X$. As $Exm([\wedge, \to, \neg]^n)$ is a $\cap$-independent n-model, there is a $q \in atom^n(k) \setminus \bigcap\{atom^n(l) \mid k < l\}$. Note that for $k < l$ we have $l \Vdash \phi^n(k) \to q$ but $k \nVdash \phi^n(k) \to q$. Now suppose that $l \in X$ and $l \Vdash \phi^n(k)$ then by lemma 3.5.0.15 we have $k \leq l$. Hence $(\phi^n(k) \to q) \in X$ iff $k \notin X$. As $k \Vdash \phi^n(X)$ infer that $k \in X$. $\dashv$

The model $Exm([\wedge, \to, \neg]^n)$ can stagewise be constructed as the minimal $\cap$-independent n-model realizing all semantic types in $[\wedge, \to, \neg]^n$. Let us define the $n + 1$ stages E^n_i needed in the construction. Recall that $\mathcal{P}^*(X)$ is the set of closed subsets in X.

3.5.0.18. DEFINITION. *Define E^n_0 as the set of 2^n terminal nodes with semantic type $\langle Q, \emptyset\rangle$ such that $Q \subseteq \{p_1, \ldots, p_n\}$.*

Now inductively define:

$$E^n_{m+1} = E^n_m \cup \{\langle Q, S\rangle \mid S \in \mathcal{P}^*(E^n_m) \text{ and } Q \subset \bigcap\{j_0(t) \mid t \in S\} \neq Q\}.$$

The order in E^n_m is the order of types in $[\wedge, \to, \neg]^n$.

Note that the construction of E^n_m is only possible for $m \leq n$.

3.5.0.19. Facts. *From the construction of E^n_n the following facts are obvious:*

1. *E^n_n is a finite $\cap$-independent n-model;*
2. *Every semantic type of $[\wedge, \rightarrow, \neg]^n$ is realized in E^n_n exactly once;*
3. *$E^n_n = Exm([\wedge, \rightarrow, \neg]^n)$.*

Let us return to the type formulas in $[\wedge, \rightarrow, \neg]^n$. Recall the definition of $\phi^n(k)$ in definition 3.5.0.14.

3.5.0.20. Definition. *Let k be a node in a finite $\cap$-independent n-model and $X \subseteq \{p_1, \ldots, p_n\}$. Define:*

1. $Newatom^n(k) = \{q \mid q \in \bigcap\{atom^n(l) \mid k < l\} \setminus atom^n(k)\}$;
2. $\Delta X = \bigwedge\{p \rightarrow q \mid p, q \in X\}$;
3. $\psi^n(k) = \begin{cases} \neg\phi^n_{\mathbf{CpL}}(k) & \text{if } \delta(k) = 0 \\ \phi^n(k) \rightarrow q, \text{ where } q \in Newatom^n(k) & \text{otherwise.} \end{cases}$

The proper definition of $\psi^n(k)$ of course requires a choice of $q \in Newatom^n(k)$. As this choice will not make any difference in the sequel, one may take for example the p_i with the least i such that $p_i \in Newatom^n(k)$.

3.5.0.21. Lemma. *If k and l are nodes in finite $\cap$-independent n-models then:*

$$l \nVdash \psi^n(k) \quad \Leftrightarrow \quad \tau^n(l) \preceq \tau^n(k).$$

Proof. If k is a terminal node, the lemma is rather trivial. So, assume $\delta(k) > 0$. To prove $l \nVdash \psi^n(k) \Rightarrow \tau^n(l) \preceq \tau^n(k)$, let $l \nVdash \psi^n(k)$. As $\psi^n(k) = \phi^n(k) \rightarrow q$, this implies, for some $l' \geq l$, that $l' \Vdash \phi^n(k)$ and $l' \nVdash q$, where $q \in Newatom^n(k)$. According to corollary 3.5.0.16, $l' \Vdash \phi^n(k)$ implies $\tau^n(k) \preceq \tau^n(l')$. In finite $\cap$-independent models, it is not difficult to prove that if $\tau^n(k) \prec \tau^n(m)$ (i.e. $\tau^n(k) \preceq \tau^n(m)$ but $\tau^n(k) \neq \tau^n(m)$), then $m \Vdash q$, for $q \in Newatom^n(k)$. As $l' \nVdash q$ and obviously from $l \leq t'$ we may conclude that $\tau^n(l) \preceq \tau^n(l')$, we have $\tau^n(l) \preceq \tau^n(l') = \tau^n(k)$.

To prove $\tau^n(l) \preceq \tau^n(k) \Rightarrow l \nVdash \psi^n(k)$, observe that by definition $k \nVdash q$. So, if $\tau^n(l) \preceq \tau^n(k)$, then $l \Vdash \psi^n(k)$ would imply, by corollary 3.5.0.16, that $k \Vdash \psi^n(k)$. As $k \Vdash \phi^n(k)$, we would have $k \Vdash q$, a contradiction. Hence, we conclude $l \nVdash \psi^n(k)$. $\dashv$

We are now ready for a characterization of $\phi^n(k)$, the type of k in $[\wedge, \rightarrow, \neg]^n$. An analogous characterization was used, as a definition, in [De Jongh 68] (also in [De Jongh 70], [De Jongh 80] and [JHR 91]). We will use the exact model of $[\wedge, \rightarrow, \neg]^n$ in the characterization. Note that as for every semantic type in a finite $\cap$-independent n-model, there is a node in the exact model with the same semantic type, theorem 3.5.0.23 is more generally applicable.

3.5.0.22. DEFINITION. *If k is a node in $Exm([\wedge, \to, \neg]^n)$ and $q \in Newatom^n(k)$ then define:*

$$\Phi^n(k) = \begin{cases} \phi^n_{\mathbf{CpL}}(k) & \text{if } \delta(k) = 0 \\ \\ \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge \\ \quad \bigwedge\{\psi^n(l) \to q \mid k <_1 l\} \wedge \\ \quad \bigwedge\{\psi^n(m) \mid not(m \leq k) \text{ and} \\ \qquad \bigcap\{atom^n(l) \mid k < l\} \subseteq atom^n(m)\} & \text{if } \delta(k) > 0. \end{cases}$$

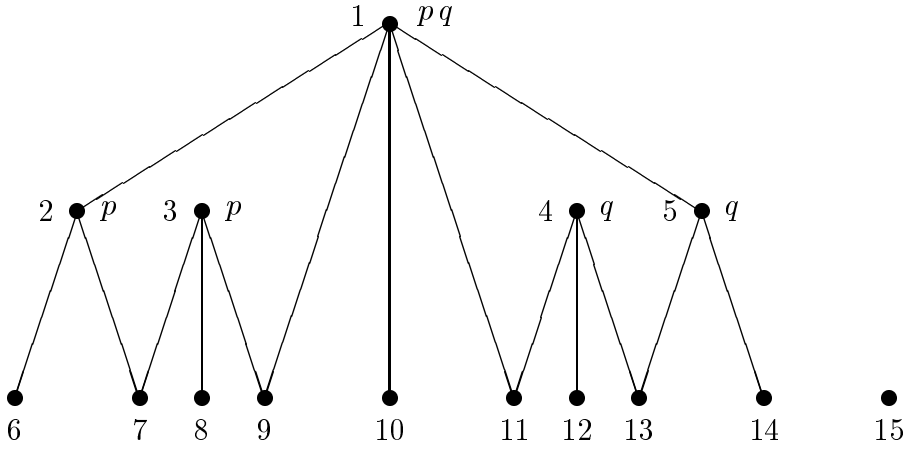
3.5.0.23. THEOREM. *If k is a node in $Exm([\wedge, \to, \neg]^n)$ then $\phi^n(k) \equiv \Phi^n(k)$.*

Proof. If k is a terminal node, then it is rather obvious that $\phi^n(k) = \phi^n_{\mathbf{CpL}}(k)$ and hence the theorem is true by definition. So assume $\delta(k) > 0$.

To prove $\phi^n(k) \vdash \Phi^n(k)$ we show that $k \Vdash \Phi^n(k)$. That $k \Vdash \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge$ is rather obvious. For $k < l$ we have $l \Vdash q$ and $k \nVdash \psi^n(l)$ by lemma 3.5.0.21. According to the same lemma $k \Vdash \psi^n(m)$ if not $m \leq k$, which proves that k will also force the last of the conjunctions in $\Phi^n(k)$.

For the proof of the other direction, assume $l \Vdash \Phi^n(k)$. We will show that as a consequence $k \leq l$ and hence $l \Vdash \phi^n(k)$. As $Exm([\wedge, \to, \neg]^n)$ is the exact model of $[\wedge, \to, \neg]^m$, this proves $\Phi^n(k) \vdash \phi^n(k)$.

Suppose $Newatom^n(k) \subseteq atom^n(l)$. Then, using the last part in the conjunction of $\Phi^n(k)$, not $k \leq l$ implies $\Phi^n(k) \vdash \psi^n(l)$. As $l \nVdash \psi^n(l)$, infer that $k \leq l$ and hence $l \Vdash \phi^n(k)$. If $Newatom^n(k)$ is not a subset of $atom^n(l)$, then $l \nVdash q$ for every $q \in Newatom^n(k)$ (because $l \Vdash \Delta Newatom^n(k)$). Hence if $k <_1 k'$ then, using the third conjunct in $\Phi^n(k)$, we have $l \nVdash \psi^n(k')$. By lemma 3.5.0.21 this implies that $l \leq k'$. Hence $atom^n(l)$ will be included in $atom^n(k) \cup Newatom^n(k)$. From $atom^n(k) \subseteq atom^n(k')$ and $Newatom^n(k) \cap atom^n(l) = \emptyset$ infer that $atom^n(l) = atom^n(k)$ and hence $\tau^n(k) = \tau^n(l)$. As semantic types are unique in $Exm([\wedge, \to, \neg]^n)$, we conclude $k = l$ and trivially $l \Vdash \phi^n(k)$. ⊣

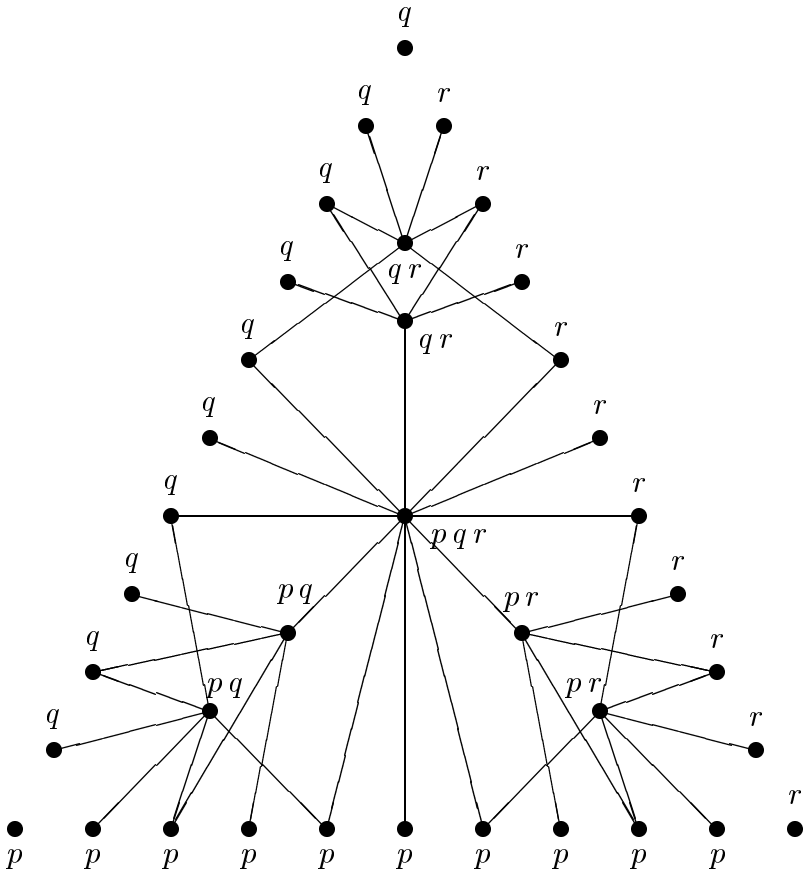


20. FIGURE. *The exact Kripke model of $[\wedge, \to, \neg]^2$.*

This model has 2 134 upwards closed subsets, corresponding to the 2 134 equivalence classes of $[\wedge, \to, \neg]^2$.

The type formulas in $[\wedge, \to, \neg]^2$ are:

1. $p \wedge q$
2. $p \wedge \neg\neg q$
3. $p \wedge \neg q$
4. $q \wedge \neg p$
5. $q \wedge \neg\neg p$
6. $\neg\neg q \wedge ((p \to q) \to p)$
7. $(\neg\neg q \to p) \wedge ((\neg\neg q \to q) \to p)$
8. $\neg(p \to q)$
9. $(q \to p) \wedge (\neg q \to p) \wedge (\neg\neg q \to q)$
10. $(p \leftrightarrow q) \wedge \neg\neg p$
11. $(p \to q) \wedge (\neg p \to q) \wedge (\neg\neg p \to p)$
12. $\neg(q \to p)$
13. $(\neg\neg p \to q) \wedge ((\neg\neg p \to p) \to q)$
14. $\neg\neg p \wedge ((q \to p) \to q)$
15. $\neg p \wedge \neg q$



21. Figure. *Part of the model* $Exm([\wedge, \to, \neg]^3)$. *The* 6 386 *nodes* k *with* $atom^n(k) = \emptyset$ *have been omitted. The order in the model is from the outside inwards.*

We may now use the exact model of the fragment $[\wedge, \to, \neg]^n$ to prove the converse of theorem 3.5.0.3. This was suggested first by Albert Visser.

3.5.0.24. Theorem. *If* ϕ *is an* **IpL** *formula such that for every node* k *in a finite Kripke model:*

$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap} \Vdash \phi$$

then ϕ is equivalent to a formula in $[\wedge, \to, \neg]^n$.

Proof. Let ϕ be a formula in $\mathbf{IpL}^n$ with the property that for every Kripke model K and every node $k \in K$, $k \Vdash \phi \Leftrightarrow (\uparrow k)^\cap \Vdash \phi$. Let $\chi \in [\wedge, \to, \neg]^n$ be the formula with $[\![\chi]\!] = [\![\phi]\!]$ in $Exm([\wedge, \to, \neg]^n)$. For a node k in a finite $\cap$-independent model we have, using lemma 3.5.0.6: $k \Vdash \phi \Leftrightarrow k \Vdash \chi$.

As χ is a formula in $[\wedge, \to, \neg]^n$, by theorem 3.5.0.3, $k \Vdash \chi \Leftrightarrow (\uparrow k)^\cap \Vdash \chi$. Hence we have:

$$k \Vdash \phi \Leftrightarrow (\uparrow k)^\cap \Vdash \phi \Leftrightarrow (\uparrow k)^\cap \Vdash \chi \Leftrightarrow k \Vdash \chi.$$

Which proves $\phi \equiv \chi$. ⊣

3.5.1 The $[\to, \neg]$ fragments

To calculate the diagram of $[\to, \neg]^n$ we have to use the exact Kripke model of $[\wedge, \to, \neg]^n$, as $Diag([\to, \neg]^n)$ for $n > 1$ is not a lattice and hence does not have an exact model of its own.

3.5.1.25. Lemma. *Every formula in $[\wedge, \to, \neg]^n$ is equivalent to a conjunction of formulas in $[\to, \neg]^n$*

Proof. We proceed by induction on the length of ϕ. Only the cases in which ϕ is a negation or an implication are non-trivial. If $\phi = \neg\psi$, then according to the induction hypothesis ψ is a conjunction of formulas in $[\to, \neg]^n$. Now apply the **IpL** theorem $\neg(A \wedge B) \equiv A \to \neg B$ to show that ϕ is equivalent to a formula in $[\to, \neg]^n$.

In the case that $\phi = \psi \to \chi$, we use the induction hypothesis first to infer that ϕ is equivalent to a conjunction of formulas of the form $\psi \to \upsilon_i$, where $\chi = \bigwedge \upsilon_i$ and every υ_i is a formula in $[\to, \neg]^n$. Again applying both the induction hypothesis and the theorem $A \wedge B \to C \equiv A \to (b \to C)$, we conclude that ϕ is equivalent to a conjunction of formulas in $[\to, \neg]^n$. ⊣

3.5.1.26. Lemma. *An* **IpL** *formula ϕ is equivalent to a formula in $[\to, \neg]^n$ iff $\phi \equiv \neg\psi$ or $\phi \equiv \psi \to p$, for some $\psi \in [\wedge, \to, \neg]^n$ and $p \in \{p_1, \ldots, p_n\}$.*

Proof. That every formula $\phi \in [\to, \neg]^n$ is equivalent to either a negation or a formula $\psi \to p$ with $\psi \in [\wedge, \to, \neg]^n$ and $p \in \{p_1, \ldots, p_n\}$ can easily be proved by induction on the length of ϕ. If $\phi = \psi \to \chi$, note that by the induction hypothesis $\chi \equiv \upsilon \to p$ and hence $\phi \equiv \psi \wedge \upsilon \to p$.

For the other direction of the lemma, note that if $\phi \equiv \psi \to p$ with $\phi \in [\wedge, \to, \neg]^n$, then ψ is, according to lemma 3.5.1.25 equivalent to a conjunction of formulas in $[\to, \neg]^n$. Now apply $A \wedge B \to C \equiv A \to (B \to C)$. ⊣

For the calculation of the number of classes in $Diag([\to, \neg]^n)$, it is more convenient to work with the complement of $[\![\phi]\!]$ in $Exm([\wedge, \to, \neg]^n)$.

3.5.1.27. Definition. *Let $[\![\phi]\!]$ be the valuation of formulas in $Exm([\wedge,\rightarrow,\neg]^n)$. Define $\alpha^n(\phi) = Exm([\wedge,\rightarrow,\neg]^n) \setminus [\![\phi]\!]$.*

3.5.1.28. Lemma. *Let ϕ and ψ be formulas in $[\wedge,\rightarrow,\neg]^n$. Then*

1. $\alpha^n(\phi) \subseteq \alpha^n(\psi) \quad \Leftrightarrow \quad \psi \vdash \phi$;
2. $\alpha^n(\phi \wedge \psi) = \alpha^n(\phi) \cup \alpha^n(\psi)$;
3. $\alpha^n(\phi{\rightarrow}\psi) = {\downarrow}(\alpha^n(\psi) \setminus \alpha^n(\phi))$;
4. $\alpha^n(\neg\phi) = {\downarrow}(Exm([\wedge,\rightarrow,\neg]^n) \setminus \alpha^n\phi) = {\downarrow}[\![\phi]\!]$.

Proof. The proofs of the first two propositions in the lemma are straightforward. The last part of the lemma is a simple corollary of the third.

For proof of the third statement in the lemma, observe that by the definition of α^n: $k \in \alpha^n(\phi{\rightarrow}\psi)$ iff for some $l \geq k$ both $l \Vdash \phi$ and $l \nVdash \psi$. Hence $k \in \alpha^n(\phi{\rightarrow}\psi)$ iff for some $l \geq k$ we have $l \in \alpha^n(\phi) \setminus \alpha^n(\psi)$. But the latter is equivalent to $k \in {\downarrow}(\alpha^n(\psi) \setminus \alpha^n(\phi))$. ⊣

3.5.1.29. Definition. *For a formula ϕ in $[\wedge,\rightarrow,\neg]^n$ we define $ucv^n(\phi)$,* the upper carrier of *ϕ, as the set of maximal elements in $\alpha^n(\phi)$.*

The upper carrier valuation was introduced in [Bruijn 75a]. Using the dual of our exact models, De Bruijn, called it the *lower carrier valuation.* Observe that $\alpha^n(\phi) = {\downarrow}ucv^n(\phi)$ and $ucv^n(\phi)$ is the smallest subset in $Exm([\wedge,\rightarrow,\neg]^n)$ with this property.

3.5.1.30. Lemma. *For $\phi \in [\wedge,\rightarrow,\neg]^n$ let $An^n(\phi)$ be the set of equivalence classes in $[\wedge,\rightarrow,\neg]^n$ that have a representative of the form $\psi{\rightarrow}\phi$, with $\psi \in [\wedge,\rightarrow,\neg]^n$. Then*

$$|An^n(\phi)| = |\mathcal{P}(ucv^n(\phi))| = 2^{|ucv^n(\phi)|}.$$

Proof. As $\alpha^n(\psi{\rightarrow}\phi) = {\downarrow}(\alpha^n(\psi) \setminus \alpha^n(\phi)) = {\downarrow}(ucv^n(\phi) \setminus \alpha^n(\psi))$, every $\psi{\rightarrow}\phi \in [\wedge,\rightarrow,\neg]^n$ corresponds to a subset in $ucv^n(\phi)$.

For every subset $X \subset ucv^n(\phi)$ there is a formula $\psi \in [\wedge,\rightarrow,\neg]^n$ such that $\alpha^n(\psi) = {\downarrow}(ucv^n(\phi) \setminus X)$, because $Exm([\wedge,\rightarrow,\neg]^n)$ is the exact model of $[\wedge,\rightarrow,\neg]^n$. Infer that $\alpha^n(\psi{\rightarrow}\phi) = {\downarrow}X$ and hence every subset of $ucv^n(\phi)$ corresponds to an equivalence class representable by a formula of the form $\psi{\rightarrow}\phi$. ⊣

The following theorem is a simple generalization of the technique used in [Bruijn 75a] to calculate the number of equivalence classes in $[\rightarrow]^3$.

3.5.1.31. Theorem. *Let $N(n,0) = 0$ and $N(n,k) = 2^{|\bigcap\{ucv^n(p_i)|i\leq k\}|}$ for $k > 0$. Moreover, let $M(n,0) = 2^{|ucv^n(\bot)|}$ and $M(n,k) = 2^{|ucv^n(\bot)\cap\bigcap\{ucv^n(p_i|i\leq k\}|}$. Then:*

$$|Diag([\rightarrow,\neg]^n)| = M(n,0) + \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}(N(n,k) - M(n,k)).$$

Proof. According to lemma 3.5.1.26 and lemma 3.5.1.30 every formula in $[\rightarrow,\neg]^n$ corresponds exactly to a subset of $ucv^n(\bot)$ or $ucv^n(p)$ for some $p \in \{p_1,\ldots,p_n\}$. In

general these upper carrier valuations are not disjoint. So, in order to count their subsets, we have to use the rule $|\mathcal{P}(A) \cup \mathcal{P}(B)| = |\mathcal{P}(A)| + |\mathcal{P}(B)| - |\mathcal{P}(A \cap B)|$. So:

$$\begin{aligned} |Diag([\to, \neg]^n)| \quad = \quad & |\mathcal{P}(ucv^n(\bot))| + \\ & |\bigcup\{\mathcal{P}(ucv^n(p_i)) \mid i \leq n\}| - \\ & |\mathcal{P}(ucv^n(\bot)) \cap \bigcup\{\mathcal{P}(ucv^n(p_i)) \mid i \leq n\}| \end{aligned}$$

Using the symmetry in $Exm([\wedge, \to, \neg]^n)$, we have

$$|\bigcup\{\mathcal{P}(ucv^n(p_i)) \mid i \leq n\}| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} N(n,k)$$

and

$$|\mathcal{P}(ucv^n(\bot)) \cap \bigcup\{\mathcal{P}(ucv^n(p_i)) \mid 1 \leq i \leq n\}| = \sum_{k=1}^{n} (-1)^{k} \binom{n}{k} M(n,k).$$

From which the equation follows. ⊣

3.5.1.32. Corollary. *The number of elements in $[\to, \neg]^2$ is:*

$$2^4 + 2(2^8 - 2^2) - (2^2 - 2) = 518.$$

Proof. In $Exm([\wedge, \to, \neg]^2)$ (see figure 20) we have $ucv^2(\bot) = \{1, 3, 4, 15\}$, $ucv^2(p) = \{4, 5, 6, 7, 8, 9, 10, 15\}$ and $ucv^2(q) = \{2, 3, 10, 11, 12, 13, 14, 15\}$. So we can calculate $ucv^2(\bot) \cap ucv^2(p) = \{4, 15\}$, $ucv^2(p) \cap ucv^2(q) = \{10, 15\}$ and $ucv^2(\bot) \cap ucv^2(p) \cap ucv^2(q) = \{15\}$. According to theorem 3.5.1.31, then $|Diag([\to, \neg]^2| = 2^4 + 2(2^8 - 2^2) - (2^2 - 2)$ ⊣

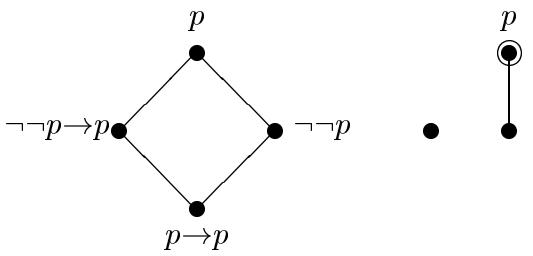
The 518 elements of the diagram of $[\to, \neg]^2$ have been calculated using the model $Exm([\wedge, \to, \neg]^n)$. They are listed in appendix B.1.

Applying the method of theorem 3.5.1.31 on $Exm([\wedge, \to, \neg]^3)$, Renardel de Lavalette calculated the cardinality of $Diag([\to, \neg]^3)$.

3.5.1.33. Fact. $|Diag([\to, \neg]^3)| = 3.2^{2^{148}} - 546$

3.5.2 The $[\wedge, \to, \neg\neg]$ fragments

The fragment $[\wedge, \to, \neg\neg]^n$ does have an exact model which is not an exact Kripke model. This is even true if $n = 1$, see figure 22, where the irreducible classes in the diagram correspond to the formulas $\neg\neg p$ and $\neg\neg p \to p$. The exact model needs a Kripke completion to force $\neg\neg p$ in the appropriate node.



22. FIGURE. *The diagram of* $[\wedge, \to, \neg\neg]^1$ *and its universal model. (The encircled node has been added.)*

As we will see, for each n there is universal model for $[\wedge, \to, \neg\neg]^n$ that is a simple Kripke extension of the exact model of $[\wedge, \to, \neg\neg]^n$.

As $[\wedge, \to, \neg\neg]^n$ is a subfragment of $[\wedge, \to, \neg]^n$, the following fact is a simple consequence of theorem 3.5.0.4.

3.5.2.34. FACT. *The fragment* $[\wedge, \to, \neg\neg]^n$ *is complete for finite* $\cap$*-independent* n*-models.*

Obviously, a node k in a finite $\cap$-independent n-model with $atom^n(k) = \{p_1, \ldots, p_n\}$ will force every formula in $[\wedge, \to, \neg\neg]^n$.

3.5.2.35. DEFINITION. *A finite* $\cap$*-independent* n*-model* K *is a* proper $[\wedge, \to, \neg\neg]^n$ model *if for every* $k \in K$ *with* $\delta(k) > 0$*, there is a* $l > k$ *such that* $atom^n(l) = \{p_1, \ldots, p_n\}$

In this subsection $Th^n(k)$ will denote the set of formulas in $[\wedge, \to, \neg\neg]^n$ forced by k.

3.5.2.36. LEMMA. *The fragment* $[\wedge, \to, \neg\neg]^n$ *is complete for proper* $[\wedge, \to, \neg\neg]^n$ *models.*

Proof. We will prove that for $\phi, \psi \in [\wedge, \to, \neg\neg]^n$ such that $\phi \nvdash \psi$, there is a k in a proper $[\wedge, \to, \neg\neg]^n$ model with $k \Vdash \phi$ and $k \nVdash \psi$.

Let $\phi, \psi \in [\wedge, \to, \neg\neg]^n$ and $\phi \nvdash \psi$. According to fact 3.5.2.34, there is a k in a $\cap$-independent n-model with $k \Vdash \phi$ and $k \nVdash \psi$. By induction on the depth of k we will prove that we can extend the submodel $\uparrow k$ to a proper $[\wedge, \to, \neg\neg]^n$ model, without changing the $[\wedge, \to, \neg\neg]^n$ theory of k.

If $\delta(k) = 0$, then $\uparrow k$ is already a proper $[\wedge, \to, \neg\neg]^n$ model. For the induction step, add a terminal node k_n to $\uparrow k$, such that $atom^n(k_n) = \{p_1, \ldots, p_n\}$ and for all $l \geq k$ with $l \notin Ter(k)$, $l > k_n$. Using induction on the length of formula $\chi \in [\wedge, \to, \neg\neg]^n$ it is straightforward to prove that $k \Vdash \chi$ iff k forces χ in the extended model. From which we conclude that the $[\wedge, \to, \neg\neg]^n$ theory of k in both models is the same. $\dashv$

The semantic types in $[\wedge, \to, \neg\neg]^n$ will be defined as the semantic types of $[\wedge, \to, \neg]^n$ restricted to proper $[\wedge, \to, \neg\neg]^n$ models.

3.5.2.37. DEFINITION. *Let* k *be a node in a proper* $[\wedge, \to, \neg\neg]^n$ *model then* $\tau^n(k)$*, the* semantic type *of* k *in* $[\wedge, \to, \neg\neg]^n$ *is defined by:*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l\}\rangle.$$

If t and t' are semantic types in $[\wedge, \to, \neg]^n$ then define $t \preceq t'$ if $t = t'$ or $t' \in j_1(t)$.

The proofs of the following facts are the same as in section 3.5.

3.5.2.38. Facts. *Let k and l be nodes in proper $[\wedge, \to, \neg\neg]^n$ models.*

1. $\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad k \leftrightarrows^n l$;
2. $\tau^n(k) \preceq \tau^n(l) \quad \Rightarrow \quad Th^n(k) \subseteq Th^n(l)$.

Let us now define the model $Umod([\wedge, \to, \neg\neg]^n)$, that will be proved in the sequel to be the universal model for $[\wedge, \to, \neg\neg]^n$.

3.5.2.39. Definition. *We define $Umod([\wedge, \to, \neg\neg]^n) = \langle T, \preceq, j_0 \rangle$, where T is the set of semantic types in $[\wedge, \to, \neg\neg]^n$.*

3.5.2.40. Facts.

1. *$Umod([\wedge, \to, \neg\neg]^n)$ is a proper $[\wedge, \to, \neg\neg]^n$ model;*
2. *if t a semantic type in $[\wedge, \to, \neg\neg]^n$ then $\tau^n(t) = t$ in $Umod([\wedge, \to, \neg\neg]^n)$;*
3. *$Umod([\wedge, \to, \neg\neg]^n)$ is complete for $[\wedge, \to, \neg\neg]^n$: if ϕ and ψ in $[\wedge, \to, \neg\neg]^n$ we have*

$$\phi \vdash \psi \quad \Leftrightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!].$$

As proper $[\wedge, \to, \neg\neg]^n$ models are finite $\cap$-independent n-models, we may use $Newatom^n(k)$, $\Delta Newatom^n(k)$ as defined in definition 3.5.0.20. Observe that, as $Diag([\wedge, \to, \neg\neg]^n)$ is obviously finite, the following definition is allowed.

3.5.2.41. Definition. *For a node k in a proper $[\wedge, \to, \neg\neg]^n$ model define $\phi^n(k)$, the* type *of k in $[\wedge, \to, \neg\neg]^n$ as:*

$$\phi^n(k) = \bigwedge Th^n(k).$$

3.5.2.42. Lemma. *Let k and l be nodes in proper $[\wedge, \to, \neg\neg]^n$ models. If $l \Vdash \phi^n(k)$ then $\tau^n(k) \preceq \tau^n(l)$ or $atom^n(l) = \{p_1, \ldots, p_n\}$.*

Proof. Assume $l \Vdash \phi^n(k)$ and $atom^n(l) \neq \{p_1, \ldots, p_n\}$. To prove $\tau^n(k) \preceq \tau^n(l)$ we will use induction on $\delta(l)$, the depth of l. If $\delta(l) = 0$ then $k \nVdash \bigwedge atom^n(l) \wedge \Delta Newatom^n(l) \to \bigwedge\{p_1, \ldots, p_n\}$ and hence there is a $k' > k$ such that $k' \Vdash \bigwedge atom^n(l) \wedge \Delta Newatom^n(l)$ and $k' \nVdash \bigwedge\{p_1, \ldots, p_n\}$. In a proper $[\wedge, \to, \neg\neg]^n$ model we may infer that k' has to be a terminal node and hence $\tau^n(k') = \tau^n(l)$. Which proves $\tau^n(k) \preceq \tau^n(l)$.

If $\delta(l) > 0$, let $l' > l$. Then $l' \Vdash \phi^n(k)$ and $\delta(l') < \delta(l)$. According to the induction hypothesis we will have $\tau^n(k) \preceq \tau^n(l')$. This proves that $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. As obviously the assumption implies that $atom^n(k) \subseteq atom^n(l)$, we may infer that $\tau^n(k) \preceq \tau^n(l)$. $\dashv$

3.5.2.43. Corollary. *Let k and l be nodes in proper $[\wedge, \to, \neg\neg]^n$ models, then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k).$$

To prove $Umod([\wedge,\rightarrow,\neg\neg]^n)$ to be an universal model for $[\wedge,\rightarrow,\neg\neg]^n$ we will use the next lemma. We will write k_n to refer to the node in $Umod([\wedge,\rightarrow,\neg\neg]^n)$ with $atom^n(k_n)=\{p_1,\ldots,p_n\}$.

3.5.2.44. Lemma. *Let X be a closed subset in $Umod([\wedge,\rightarrow,\neg\neg]^n)$, containing k_n. Define $\phi^n(X)=\bigwedge\bigcap\{Th^n(l)\mid l\in X\}$. Then for every node k in $Umod([\wedge,\rightarrow,\neg\neg]^n)$:*

$$k\in X \quad\Leftrightarrow\quad k\Vdash\phi^n(X).$$

Proof. That $k\in X$ implies $k\Vdash\phi^n(X)$ is clear from the definition of $\phi^n(X)$. For the other direction, like in theorem 3.5.0.17 we proceed by induction over $\delta(k)$, the depth of k.

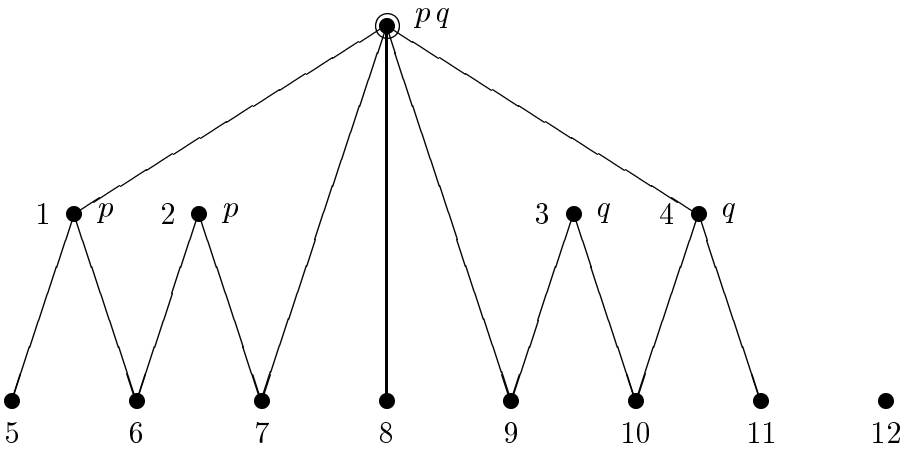
So assume $k\Vdash\phi^n(X)$. If $\delta(k)=0$, then $\Delta Newatom^n(k)\vdash p{\rightarrow}q$ for every p and q in $\{p_1,\ldots,p_n\}$. Either $k=k_n$ and $k\in X$ by definition, or $\phi^n(X)\nvdash\bigwedge atom^n(k)\wedge\Delta Newatom^n(k){\rightarrow}\bigwedge\{p_1,\ldots,p_n\}$. Suppose $k\neq k_n$. Then for some $l\in X$ there is a $l'\geq l$ such that $l'\Vdash\phi^n(X)\nvdash\bigwedge atom^n(k)\wedge\Delta Newatom^n(k)$ and $l'\nVdash\bigwedge\{p_1,\ldots,p_n\}$. As $Umod([\wedge,\rightarrow,\neg\neg]^n)$ is a proper $[\wedge,\rightarrow,\neg\neg]^n$ model, this implies that l' is a terminal node and $atom^n(k)=atom^n(l)$. As semantic types are unique in $Umod([\wedge,\rightarrow,\neg\neg]^n)$, infer that $k=l'$. The set X was supposed to be closed, so, from $l\leq k$ we conclude that $k\in X$.

If $\delta(k)>0$ then for $k'>k$ we conclude from $k'\Vdash\phi^n(X)$ and the induction hypothesis that $k'\in X$. As $Umod([\wedge,\rightarrow,\neg\neg]^n)$ is a proper $[\wedge,\rightarrow,\neg\neg]^n$ model, there is a $q\in atom^n(k)\setminus\bigcap\{atom^n(l)\mid k<l\}$. Note that for $k<l$ we have $l\Vdash\phi^n(k){\rightarrow}q$ but $k\nVdash\phi^n(k){\rightarrow}q$. Now suppose that $l\in X$ and $l\Vdash\phi^n(k)$ then by lemma 3.5.2.42 we have $k\leq l$. Hence $(\phi^n(k){\rightarrow}q)\in X$ iff $k\notin X$. As $k\Vdash\phi^n(X)$ infer that $k\in X$. $\dashv$

3.5.2.45. Theorem. *$Umod([\wedge,\rightarrow,\neg\neg]^n)$ is a universal model for $[\wedge,\rightarrow,\neg\neg]^n)$.*

Proof. $Umod([\wedge,\rightarrow,\neg\neg]^n)$ is a complete model for $[\wedge,\rightarrow,\neg\neg]^n$, according to fact 3. By lemma 3.5.2.44 we have an exact correspondence between equivalence classes in $[\wedge,\rightarrow,\neg\neg]^n$ and closed subsets of $Umod([\wedge,\rightarrow,\neg\neg]^n)$ that include the node k_n. Deleting k_n from $Umod([\wedge,\rightarrow,\neg\neg]^n)$ and assigning the empty set to $\bigwedge\{p_1,\ldots,p_n\}$, we have an exact correspondence between members of $Diag()$ and closed subsets in the resulting model. Clearly $Umod([\wedge,\rightarrow,\neg\neg]^n)$ is a minimal complete model for $[\wedge,\rightarrow,\neg\neg]^n$. $\dashv$

3.5.2.46. Corollary. *The exact model of $[\wedge,\rightarrow,\neg\neg]^n$ is (isomorphic to) $Umod([\wedge,\rightarrow,\neg\neg]^n)$, after deleting k_n.*



23. FIGURE. *The model Umod*$([\wedge, \to, \neg\neg]^n)$. *The encircled node has been added to the exact model of* $[\wedge, \to, \neg\neg]^2$.

The exact model has 676 upward closed subsets, corresponding to the 676 equivalence classes of $[\wedge, \to, \neg\neg]^2$.

The type formulas in $[\wedge, \to, \neg\neg]^2$ are:

1. $p \wedge \neg\neg q$	5. $\neg\neg q \wedge ((p \to q) \to p)$	9. $(q \to \neg\neg p) \to (p \wedge q)$
2. $p \wedge (\neg\neg q \to q)$	6. $(\neg\neg q \to (p \wedge q)) \to p)$	10. $(\neg\neg p \to (p \wedge q)) \to q)$
3. $q \wedge (\neg\neg p \to p)$	7. $(p \to \neg\neg q) \to (p \wedge q)$	11. $\neg\neg p \wedge ((q \to p) \to q)$
4. $\neg\neg p \wedge q$	8. $(p \to q) \wedge (q \to p) \wedge \neg\neg p$	12. $((q \to p) \to \neg\neg p) \to (p \wedge q)$

3.5.3 The $[\to, \neg\neg]$ fragments

The diagram of $[\to, \neg\neg]^n$ is not a lattice (it does not have a bottom element) if $n > 1$. For $n = 1$ we have, of course, $Diag([\to, \neg\neg]^1) \cong Diag([\wedge, \to, \neg\neg]^1)$ (see figure 22).

To calculate the diagram of $[\to, \neg]^n$ we will use the universal model of $[\wedge, \to, \neg\neg]^n$.

3.5.3.47. LEMMA. *Every formula in* $[\wedge, \to, \neg\neg]^n$ *is equivalent to a conjunction of formulas in* $[\to, \neg\neg]^n$

Proof. The proof is much like that of lemma 3.5.1.25. We proceed by induction on the length of ϕ. Only the cases in which ϕ is a double negation or an implication are non-trivial. If $\phi = \neg\neg\psi$, then according to the induction hypothesis ψ is a conjunction of formulas in $[\to, \neg\neg]^n$. Now apply the **IpL** theorem $\neg\neg(A \wedge B) \equiv \neg(A \to \neg B)$ to show that ϕ is equivalent to a formula in $[\to, \neg]^n$.

In the case that $\phi = \psi \to \chi$, the proof runs like in 3.5.1.25. ⊣

3.5.3.48. LEMMA. *An* **IpL** *formula* ϕ *is equivalent to a formula in* $[\to, \neg\neg]^n$ *iff* $\phi \equiv \psi \to p$ *for some* $\psi \in [\wedge, \to, \neg\neg]^n$ *and* $p \in \{p_1, \ldots, p_n\}$.

Proof. The proof is essentially the same as that of lemma 3.5.1.26. Observe that double negations can be treated as implications, using $\neg\neg\phi \equiv (\neg\neg\phi \to \phi) \to \phi$. ⊣

As in subsection 3.5.1, for the calculation of the number of classes in $Diag([\rightarrow,\neg\neg]^n)$ it is more convenient to work with the complement of $[\![\phi]\!]$ in $Umod([\wedge,\rightarrow,\neg\neg]^n)$.

3.5.3.49. Definition. *Let $[\![\phi]\!]$ be the valuation of formulas in $Umod([\wedge,\rightarrow,\neg\neg]^n)$. Define $\alpha^n(\phi) = Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus [\![\phi]\!]$.*

The proof of the following facts is exactly the same as for lemma 3.5.1.28.

3.5.3.50. Facts. *Let ϕ and ψ be formulas in $[\wedge,\rightarrow,\neg\neg]^n$. Then*

1. $\alpha^n(\phi) \subseteq \alpha^n(\psi) \quad \Leftrightarrow \quad \psi \vdash \phi$;
2. $\alpha^n(\phi \wedge \psi) = \alpha^n(\phi) \cup \alpha^n(\psi)$;
3. $\alpha^n(\phi{\rightarrow}\psi) = {\downarrow}(\alpha^n(\psi) \setminus \alpha^n(\phi))$;
4. $\alpha^n(\neg\neg\phi) = {\downarrow}Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus ({\downarrow}Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus dar\alpha^n(\phi)) = {\downarrow}Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus {\downarrow}[\![\phi]\!]$.

3.5.3.51. Definition. *For a formula ϕ in $[\wedge,\rightarrow,\neg\neg]^n$ we define $ucv^n(\phi)$,* the upper carrier of *ϕ, as the set of maximal elements in $\alpha^n(\phi)$.*

Observe that the element k_n in $Umod([\wedge,\rightarrow,\neg\neg]^n)$, with $atom^n(k_n) = \{p_1,\ldots,p_n\}$, is in $[\![\phi]\!]$ for every $\phi \in [\wedge,\rightarrow,\neg\neg]^n$ and hence in no $\alpha^n(\phi)$ or $ucv^n(\phi)$.

3.5.3.52. Lemma. *For $\phi \in [\wedge,\rightarrow,\neg\neg]^n$ let $An^n(\phi)$ be the set of equivalence classes in $[\wedge,\rightarrow,\neg\neg]^n$ that have a representative of the form $\psi{\rightarrow}\phi$ with $\psi \in [\wedge,\rightarrow,\neg\neg]^n$. Then*

$$|An^n(\phi)| = |\mathcal{P}(ucv^n(\phi))| = 2^{|ucv^n(\phi)|}.$$

Proof. The proof is essentially the same as for lemma 3.5.1.30. ⊣

3.5.3.53. Theorem.

$$|Diag([\rightarrow,\neg\neg]^n)| = \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}N(n,k).$$

where $N(n,k) = 2^{|\bigcap\{ucv^n(p_i)|i\leq k\}|}$.

Proof. The proof is a simplified version of the proof of theorem 3.5.1.31, using the symmetry in $Umod([\wedge,\rightarrow,\neg\neg]^n)$. ⊣

3.5.3.54. Corollary. *The number of elements in $[\rightarrow,\neg\neg]^2$ is:*

$$2.2^7 - 2^2 = 252.$$

Proof. In $Umod([\wedge,\rightarrow,\neg\neg]^n$ (see figure 23) we have $ucv^2(p) = \{3,4,5,6,7,8,12\}$ $ucv^2(q) = \{1,2,8,9,10,11,12\}$. So we have $ucv^2(p) \cap ucv^2(q) = \{8,12\}$. According to theorem 3.5.3.53, then $|Diag([\rightarrow,\neg]^2| = 2.2^7 - 2^2$ ⊣

Applying the method of theorem 3.5.1.31 on $Exm([\wedge,\rightarrow,\neg]^3)$, Renardel de Lavalette calculated the cardinality of $Diag([\rightarrow,\neg\neg]^3)$.

3.5.3.55. Fact. $|Diag([\rightarrow,\neg\neg]^3)| = 3.2^{689} - 380$

3.6 The $[\wedge, \rightarrow]$ fragments

The $[\wedge, \rightarrow]^n$ fragments of **IpL** are much like the $[\wedge, \rightarrow, \neg]^n$ fragments[4].

As we will see, the only difference between the semantic types in $[\wedge, \rightarrow, \neg]^n$ and $[\wedge, \rightarrow]^n$ is that in the later the semantic types with $j_0(k) = \{p_1, \ldots, p_n\}$ are redundant. Observe that a node with such a semantic type (i.e. where all the atoms hold), forces all formulas in $[\wedge, \rightarrow]^n$.

3.6.0.1. Definition. *A finite $\cap$-independent n-model K is a* proper $[\wedge, \rightarrow]^n$ model *if for no $k \in K$ we have $atom^n(k) = \{p_1, \ldots, p_n\}$.*

Any $\cap$-independent n-model can easily be turned into a proper $[\wedge, \rightarrow]^n$ model.

3.6.0.2. Definition. *Let K be a $\cap$-independent n-model. Then K^- is the model resulting from K after leaving out all nodes k with $atom^n(k) = \{p_1, \ldots, p_n\}$.*

3.6.0.3. Lemma. *Let K be a finite $\cap$-independent n-model and $k \in K^-$. Then for all $\phi \in [\wedge, \rightarrow]^n$:*

$$k \Vdash_K \phi \quad \Leftrightarrow \quad k \Vdash_{K^-} \phi.$$

Proof. Let us use $\Vdash'$ for forcing in K^- in contrast to $\Vdash$ for forcing in K. We proceed by induction on the length of ϕ. The cases where ϕ is either atomic or a conjunction are trivial. So let $\phi = \psi \rightarrow \chi$. Suppose $k \Vdash \phi$ and let $l \in K^-$ such that $k \leq l$ and $l \Vdash' \psi$. Using the induction hypothesis we conclude that $l \Vdash \psi$ and hence $l \Vdash \chi$. Again by the induction hypothesis, we infer that $l \Vdash' \chi$. Which proves $\forall l \geq k(l \Vdash' \psi \quad \Rightarrow \quad l \Vdash' \chi)$, i.e. $k \Vdash' \phi$.

Now suppose $k \Vdash' \phi$ and $l \in K$ with both $k \leq l$ and $l \Vdash \psi$. If $atom^n(l) = \{p_1, \ldots, p_n\}$ then l forces all formulas of $[\wedge, \rightarrow]^n$ and hence also $l \Vdash \chi$. Otherwise, we have $l \in K^-$. By the induction hypothesis $l \Vdash' \psi$ and, as $k \Vdash' \phi$, also $l \Vdash' \chi$. Again with the induction hypothesis, we conclude $l \Vdash \chi$. Which proves $k \Vdash \phi$. $\dashv$

The following theorem justifies our definition of proper $[\wedge, \rightarrow]^n$ models.

3.6.0.4. Theorem. *The fragment $[\wedge, \rightarrow]^n$ is complete for proper $[\wedge, \rightarrow]^n$ models.*

Proof. Let ϕ and ψ be formulas in $[\wedge, \rightarrow]^n$, such that $\phi \nvdash \psi$. As $[\wedge, \rightarrow]^n$ is a subfragment of $[\wedge, \rightarrow, \neg]^n$, by application of theorem 3.5.0.4, there is a node k in a $\cap$-independent n-model K with $k \Vdash \phi$ and $k \nVdash \psi$. From $k \nVdash \psi$ infer that $k \in K^-$. As K^- is a proper $[\wedge, \rightarrow]^n$ model and according to lemma 3.6.0.2 $k \Vdash \phi$ and $k \Vdash \psi$ in K^-. $\dashv$

3.6.0.5. Definition. *For a node k in a proper $[\wedge, \rightarrow]^n$ model define the* semantic type *of k in $[\wedge, \rightarrow]^n$ as:*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l\}\rangle.$$

[4] As an alternative notation of $[\wedge, \rightarrow, \neg]^n$ we could have taken $[\wedge, \rightarrow, \bot]^n$.

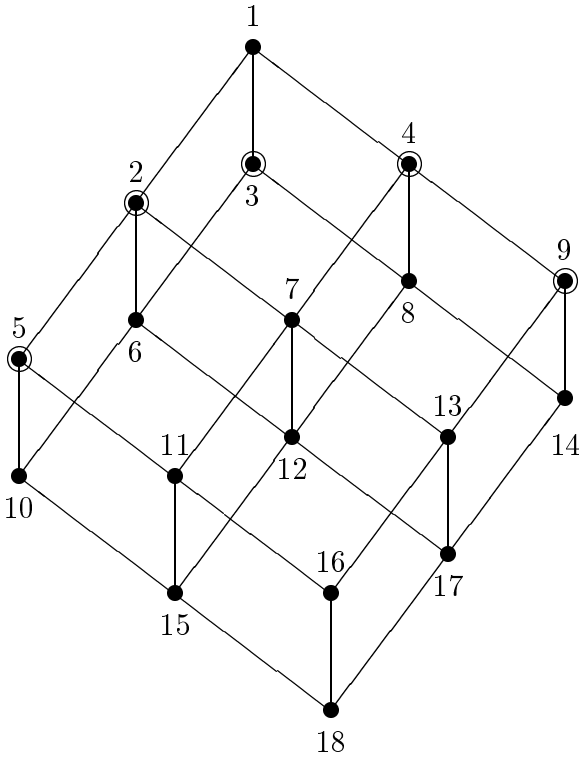
Semantic types in $[\wedge,\to]^n$ are a special case of semantic types in $[\wedge,\to,\neg]^n$ and they are ordered in the same way. Obviously there are only finitely many semantic types in $[\wedge,\to]^n$.

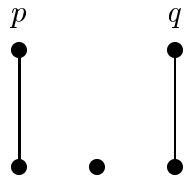
For the proof of the following facts one only has to modify slightly the corresponding proofs in section 3.5. Obviously $Th^n(k)$ in this section means the theory of node k in $[\wedge,\to]^n$.

3.6.0.6. Facts. *Let k and l be nodes in proper $[\wedge,\to]^n$ models.*

1. $\tau^n(k)=\tau^n(l) \quad\Leftrightarrow\quad k \overleftrightarrows{}^n l$;
2. $\tau^n(k)\preceq\tau^n(l) \quad\Rightarrow\quad Th^n(k)\subseteq Th^n(l)$.

3.6.0.7. Definition. *If T is the set of semantic types in $[\wedge,\to]^n$, then define $Exm([\wedge,\to]^n)=\langle T,\preceq,j_0\rangle$.*





24. Figure. *The fragment $[\wedge,\to]^2$ and the model $Exm([\wedge,\to]^n)$.*

The formulas in $Diag([\wedge,\to]^2)$:

1. $p\wedge q$
2. p
3. $(p\to q)\wedge(q\to p)$
4. q
5. $(p\to q)\to p$
6. $((p\to q)\to q)\to p$
7. $((p\to q)\to q)\wedge((q\to p)\to p)$
8. $((q\to p)\to p)\to q$
9. $(q\to p)\to q$
10. $q\to p$
11. $(p\to q)\to q$
12. $(((p\to q)\to p)\to p)\wedge(((q\to p)\to q)\to q)$
13. $(q\to p)\to p$
14. $p\to q$
15. $((q\to p)\to q)\to q$
16. $((p\to q)\wedge(q\to p))\to p$
17. $((p\to q)\to p)\to p$
18. $p\to p$

As in section 3.5 the following facts are rather simple consequences of the definition of $Exm([\wedge,\to]^n)$.

3.6.0.8. Facts.

1. *$Exm([\wedge,\to]^n)$ is a proper $[\wedge,\to]^n$ model;*
2. *for t a semantic type in $[\wedge,\to]^n$ we have $\tau^n(t)=t$ in $Exm([\wedge,\to]^n)$;*
3. *$Exm([\wedge,\to]^n)$ is complete for $[\wedge,\to]^n$: for ϕ and ψ in $[\wedge,\to]^n$ we have*

$$\phi\vdash\psi \quad\Leftrightarrow\quad [\![\phi]\!]\subseteq[\![\psi]\!].$$

As $Diag([\wedge,\to]^n)$ is finite, the following definition is allowed.

3.6.0.9. Definition. *For a node k in a proper $[\wedge,\to]^n$ model define $\phi^n(k)$, the* type *of k in $[\wedge,\to]^n$ as*

$$\phi^n(k)=\bigwedge Th^n(k).$$

3.6.0.10. Lemma. *Let k and l be nodes in proper $[\wedge,\to]^n$ models. If $l\Vdash\phi^n(k)$ then $\tau^n(k)\preceq\tau^n(l)$.*

Proof. Assume $l\Vdash\phi^n(k)$. We will use induction on $\delta(l)$, the depth of l. If $\delta(l)=0$ then l is a terminal node. We may conclude that the formula $\bigwedge atom^n(l)\to\bigwedge\{p\to\bigwedge\{p_1,\ldots,p_n\}\mid p\in\{p_1,\ldots,p_n\}\setminus atom^n(l)\}$ does not belong to $Th^n(k)$. Hence for some terminal node k' with $k\leq k'$ we have $atom^n(k')=atom^n(l)$, which proves $\tau^n(l)=\tau^n(k)$.

If $\delta(l)>0$, let $l'>l$. Then $l'\Vdash\phi^n(k)$ and $\delta(l')<\delta(l)$. According to the induction hypothesis we will have $\tau^n(k)\preceq\tau^n(l')$. This proves that $j_1(\tau^n(l))\subseteq j_1(\tau^n(k))$. As the assumption implies that $atom^n(k)\subseteq atom^n(l)$, this proves $\tau^n(k)\preceq\tau^n(l)$. ⊣

3.6.0.11. Corollary. *If k and l are nodes in proper $[\wedge,\to]^n$ models then:*

$$\tau^n(k)\preceq\tau^n(l)\quad\Leftrightarrow\quad Th^n(k)\subseteq Th^n(l)\quad\Leftrightarrow\quad l\Vdash\phi^n(k).$$

3.6.0.12. Theorem. *The model $Exm([\wedge,\to]^n)$ defined above is the exact Kripke model of $[\wedge,\to]^n$.*

Proof. $Exm([\wedge,\to]^n)$ is complete for $[\wedge,\to]^n$ according to fact 3.6.0.8. We still have to prove that every closed subset X in this model corresponds to a formula in $[\wedge,\to]^n$. The proof is very much like that of theorem 3.5.0.17.

Let $\phi^n(X) = \bigwedge\bigcap\{Th^n(k) \mid k \in X\}$. Then clearly, by definition, it will be true that $X \subseteq [\![\phi^n(X)]\!]$.

To prove the inclusion in the other direction, suppose that $k \Vdash \phi^n(X)$. With induction on $\delta(k)$, the depth of k, we will prove that $k \in X$. If $\delta(k) = 0$ then k is a terminal node and apparently it is the case that for some $l \in X$ we had $l \nVdash \Phi(k)$, where $\Phi(k) = \bigwedge atom^n(k) \to \bigwedge\{p \to \bigwedge\{p_1, \ldots, p_n\} \mid p \in \{p_1, \ldots, p_n\} \setminus atom^n(k)\}$.

As in the proof of lemma 3.6.0.10 infer that for some $l \in X$ there is a terminal node $l' > l$ with $atom^n(l) = atom^n(k)$. As the semantic types in $Exm([\wedge, \to]^n)$ are unique, we conclude that $k = l'$. From $l \in X$ and $l < k$ infer that $k \in X$ as X is a closed subset of $Exm([\wedge, \to]^n)$.

If $\delta(k) > 0$ then for $k' > k$ we conclude from $k' \Vdash \phi^n(X)$ and the induction hypothesis that $k' \in X$. As $Exm([\wedge, \to]^n)$ is a proper $[\wedge, \to]^n$ model, there is a $q \in atom^n(k) \setminus \bigcap\{atom^n(l) \mid k < l\}$. Note that for $k < l$ we have $l \Vdash \phi^n(k) \to q$ but $k \nVdash \phi^n(k) \to q$. Now suppose that $l \in X$ and $l \Vdash \phi^n(k)$ then by lemma 3.6.0.10 we have $k \leq l$. Hence $(\phi^n(k) \to q) \in X$ iff $k \notin X$. As $k \Vdash \phi^n(X)$ infer that $k \in X$. $\dashv$

Like $Exm([\wedge, \to, \neg]^n)$, the model $Exm([\wedge, \to]^n)$ can stagewise be constructed as the minimal proper $[\wedge, \to]^n$ model realizing all semantic types in $[\wedge, \to]^n$. Let us define the $n+1$ stages E^n_i needed in the construction. Recall that $\mathcal{P}^*(X)$ is the set of closed subsets in X.

3.6.0.13. Definition. *Define E^n_0 as the set of 2^n terminal nodes with semantic type $\langle Q, \emptyset\rangle$ such that $Q \subset \{p_1, \ldots, p_n\} \neq Q$.*

Now inductively define:

$$E^n_{m+1} = E^n_m \cup \{\langle Q, S\rangle \mid S \in \mathcal{P}^*(E^n_m) \text{ and } Q \subset \bigcap\{j_0(t) \mid t \in S\} \neq Q\}.$$

Where the order in E^n_m is the order of types in $[\wedge, \to]^n$.

Note that the construction of E^n_m is only possible for $m \leq n$. From the construction of E^n_n the following facts are obvious:

3.6.0.14. Facts.

1. *E^n_n is a proper $[\wedge, \to]^n$ model;*
2. *Every semantic type of $[\wedge, \to]^n$ is realized in E^n_n exactly once;*
3. *$E^n_n = Exm([\wedge, \to]^n)$.*

Let us return to the type formulas in $[\wedge, \to]^n$.

3.6.0.15. Definition. *Let k be a node in a proper $[\wedge, \to]^n$ model and $X \subset \{p_1, \ldots, p_n\}$. Define:*

1. *$Newatom^n(k) = \{q \mid q \in \bigcap\{atom^n(l) \mid k < l\} \setminus atom^n(k)\}$;*
2. *$\Delta X = \bigwedge\{p \to q \mid p, q \in X\}$;*
3. *$\psi^n(k) = \phi^n(k) \to q$, where $q \in Newatom^n(k)$;*
4. $$\psi^n(k) = \begin{cases} \phi^n(k) \to \bigwedge\{p_1, \ldots, p_n\} & \text{if } \delta(k) = 0 \\ \phi^n(k) \to q, \text{ where } q \in Newatom^n(k) & \text{otherwise.} \end{cases}$$

The proper definition of $\psi^n(k)$ of course requires a choice of $q \in Newatom^n(k)$. As this choice will not make any difference in the sequel one may take for example the p_i with the least i such that $p_i \in Newatom^n(k)$. If k is a terminal node, by defining $\bigcap\emptyset = \{p_1,\ldots,p_n\}$, we have $Newatom^n(k) = \{p_1,\ldots,p_n\} \setminus atom^n(k)$.

3.6.0.16. LEMMA. *If k and l nodes in proper $[\wedge,\to]^n$ models then:*

$$l \nVdash \psi^n(k) \quad \Leftrightarrow \quad \tau^n(l) \preceq \tau^n(k).$$

Proof. If k is a terminal node, the lemma is rather trivial. So, assume $\delta(k) > 0$. To prove $l \nVdash \psi^n(k) \;\Rightarrow\; \tau^n(l) \preceq \tau^n(k)$, let $l \nVdash \psi^n(k)$. As $\psi^n(k) = \phi^n(k)\to q$, this implies, for some $l' \geq l$, that $l' \Vdash \phi^n(k)$ and $l' \nVdash q$, where $q \in Newatom^n(k)$. According to corollary 3.5.0.16, $l' \Vdash \phi^n(k)$ implies $\tau^n(k) \preceq \tau^n(l')$. In finite $\cap$-independent models, it is not difficult to prove that if $\tau^n(k) \prec \tau^n(m)$ (i.e. $\tau^n(k) \preceq \tau^n(m)$ but $\tau^n(k) \neq \tau^n(m)$), then $m \Vdash q$, for $q \in Newatom^n(k)$. As $l' \nVdash q$ and obviously from $l \leq t'$ we may conclude that $\tau^n(l) \preceq \tau^n(l')$, we have $\tau^n(l) \preceq \tau^n(l') = \tau^n(k)$.

To prove $\tau^n(l) \preceq \tau^n(k) \;\Rightarrow\; l \nVdash \psi^n(k)$, observe that by definition $k \nVdash q$. So, if $\tau^n(l) \preceq \tau^n(k)$, then $l \Vdash \psi^n(k)$ would imply, by corollary 3.5.0.16, that $k \Vdash \psi^n(k)$. As $k \Vdash \phi^n(k)$, we would have $k \Vdash q$, a contradiction. Hence, we conclude $l \nVdash \psi^n(k)$. $\dashv$

We are now ready for a characterization of $\phi^n(k)$, the type of k in $[\wedge,\to,\neg]^n$. An analogous characterization was used, as a definition, in [De Jongh 68] (also in [De Jongh 70], [De Jongh 80] and [JHR 91]). We will use the exact model of $[\to,\neg]^n$ in the characterization.

3.6.0.17. DEFINITION. *If k is a node in $Exm([\wedge,\to]^n)$ and $q \in Newatom^n(k)$ then define:*

$$\Phi^n(k) = \begin{cases} \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) & \text{if } \delta(k) = 0 \\ \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge \\ \quad \bigwedge\{\psi^n(l)\to q \mid k <_1 l\} \wedge \\ \quad \bigwedge\{\psi^n(m) \mid not\,(m \leq k) \text{ and} \\ \qquad \bigcap\{atom^n(l) \mid k < l\} \subseteq atom^n(m)\} & \text{if } \delta(k) > 0. \end{cases}$$

3.6.0.18. THEOREM. *If k is a node in $Exm([\wedge,\to]^n)$ then $\phi^n(k) \equiv \Phi^n(k)$.*

Proof. If k is a terminal node, first observe that trivially $k \Vdash \Phi^n(k)$ as $Exm([\wedge,\to]^n)$ is a proper $[\wedge,\to]^n$ model. If $l \in Exm([\wedge,\to]^n)$ and $l \Vdash \Phi^n(k)$ then clearly $atom^n(k) \subseteq atom^n(l)$ and, again because $Exm([\wedge,\to]^n)$ is a proper $[\wedge,\to]^n$ model, for no $l' \geq l$ it will be true that $l' \Vdash p$ for some $p \notin atom^n(k)$. Hence l has to be a terminal node with $atom^n(l) = atom^n(k)$, which proves $\tau^n(k) = \tau^n(l)$, so $k = l$.

So assume $\delta(k) > 0$. To prove $\phi^n(k) \vdash \Phi^n(k)$ we show that $k \Vdash \Phi^n(k)$. That $k \Vdash \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge$ is rather obvious. For $k < l$ we have $l \Vdash q$ and $k \nVdash \psi^n(l)$ by lemma 3.6.0.16. According to the same lemma $k \Vdash \psi^n(m)$ if not $m \leq k$, which proves that k will also force the last of the conjunctions in $\Phi^n(k)$.

For the proof of the other direction, assume $l \Vdash \Phi^n(k)$. We will show that as a consequence $k \leq l$ and hence $l \Vdash \phi^n(k)$. As $Exm([\wedge,\rightarrow]^n)$ is the exact model of $[\wedge,\rightarrow]^m$, this proves $\Phi^n(k) \vdash \phi^n(k)$.

Suppose $Newatom^n(k) \subseteq atom^n(l)$. Then, using the last part in the conjunction of $\Phi^n(k)$, not $k \leq l$ implies $\Phi^n(k) \vdash \psi^n(l)$. As $l \nVdash \psi^n(l)$, infer that $k \leq l$ and hence $l \Vdash \phi^n(k)$.

If $Newatom^n(k)$ is not a subset of $atom^n(l)$, then $l \nVdash q$ for every $q \in Newatom^n(k)$ (because $l \Vdash \Delta Newatom^n(k)$). Hence if $k <_1 k'$ then, using the third conjunction in $\Phi^n(k)$, we have $l \nVdash \psi^n(k')$. By lemma 3.6.0.16 this implies that $l \leq k'$. Hence $atom^n(l)$ will be included in $atom^n(k) \cup Newatom^n(k)$. From $atom^n(k) \subseteq atom^n(k')$ and $Newatom^n(k) \cap atom^n(l) = \emptyset$ infer that $atom^n(l) = atom^n(k)$ and hence $\tau^n(k) = \tau^n(l)$. As semantic types are unique in $Exm([\wedge,\rightarrow]^n)$, we conclude $k = l$ and $l \Vdash \phi^n(k)$. ⊣

We may now use the exact model of the fragment $[\wedge,\rightarrow]^n$ to prove a characterization of the $[\wedge,\rightarrow]$ formulas in **IpL**. Recall the definition of $K^{\text{-}}$ from definition 3.6.0.2.

3.6.0.19. Theorem. *If ϕ is an* **IpL** *formula, then ϕ is equivalent to a $[\wedge,\rightarrow]$ formula if for every node k in a finite Kripke model:*

$$k \Vdash \phi \quad \Leftrightarrow \quad ((\uparrow k)^{\cap})^{\text{-}} \Vdash \phi.$$

Proof. For $\phi \in [\wedge,\rightarrow]$ we have by theorem 3.5.0.3 that $k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap}) \Vdash \phi$. Using theorem 3.6.0.3 we may infer that $k \Vdash \phi \quad \Leftrightarrow \quad ((\uparrow k)^{\cap})^{\text{-}} \Vdash \phi$.

To prove the other direction, let ϕ be a formula in $\mathbf{IpL}^n$ with the property that for every finite Kripke model K and every node $k \in K$, $k \Vdash \phi \quad \Leftrightarrow \quad ((\uparrow k)^{\cap})^{\text{-}} \Vdash \phi$. Let $\chi \in [\wedge,\rightarrow,\neg]^n$ be the formula with $[\![\chi]\!] = [\![\phi]\!]$ in $Exm([\wedge,\rightarrow]^n)$. For a node k in a proper $[\wedge,\rightarrow]$ model we have, using fact 3.6.0.6.1: $k \Vdash \phi \quad \Leftrightarrow \quad k \Vdash \chi$. Hence we have:

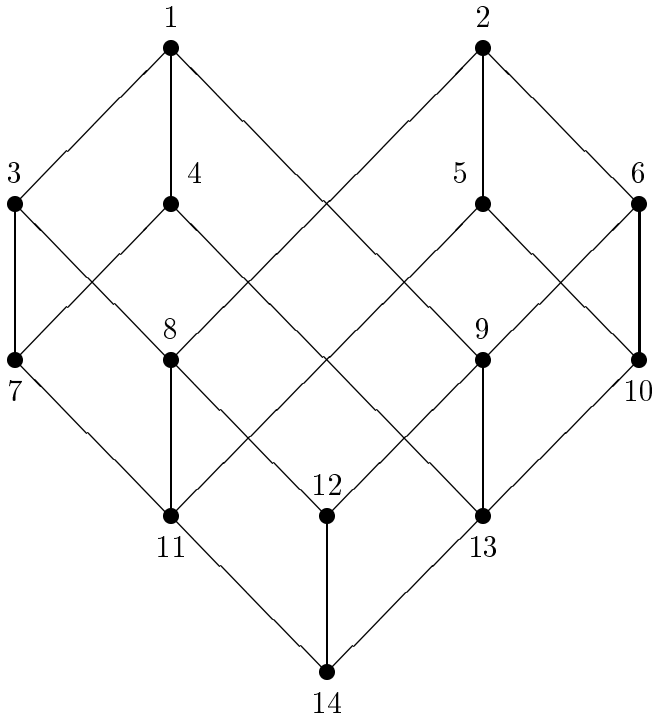
$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap} \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap} \Vdash \chi \quad \Leftrightarrow \quad k \Vdash \chi.$$

Which proves $\phi \equiv \chi$. ⊣

3.6.1 The $[\rightarrow]$ fragments

The $[\rightarrow]$ fragments are the most expressive fragments in **IpL** with only one connective. For example $[\rightarrow]^3$ has 25 165 802 equivalence classes, whereas the fragments with three atoms and exactly one of the other connectives in $\{\wedge,\vee,\neg,\neg\neg\}$ all have less then 10 classes.

To calculate the diagram of $[\rightarrow]^n$ we have to use the exact Kripke model of $[\wedge,\rightarrow]^n$, for example, as $Diag([\rightarrow]^n)$ for $n > 1$ is not a lattice and hence does not have an exact model of its own.



25. FIGURE. *The diagram of* $[\to]^2$.

The formulas in $Diag([\to]^2)$:

1. p	6. $(q\to p)\to q$	11. $((q\to p)\to q)\to q$
2. q	7. $q\to p$	12. $(((p\to q)\to q)\to p)\to p$
3. $(p\to q)\to p$	8. $(p\to q)\to q$	13. $((p\to q)\to p)\to p$
4. $((p\to q)\to q)\to p$	9. $(q\to p)\to p$	14. $p\to p$
5. $((q\to p)\to p)\to q$	10. $p\to q$	

To calculate the number of classes in $Diag([\to]^n)$ we will proceed much like in subsection 3.5.1. The proofs of the following lemma's, preparing for theorem 3.6.1.26, are omitted, as they are essentially the same as for the lemma's 3.5.1.25 up to 3.5.1.30.

3.6.1.20. LEMMA. *Every formula in $[\wedge,\to]^n$ is equivalent to a conjunction of formulas in $[\to]^n$*

3.6.1.21. LEMMA. *An* **IpL** *formula ϕ is equivalent to a formula in $[\to]^n$ iff $\phi\equiv\psi\to p$ for some $\psi\in[\wedge,\to]^n$ and $p\in\{p_1,\ldots,p_n\}$.*

As in subsection 3.5.1, in the calculation of the number of equivalence classes in $[\wedge,\to]^n$ it is more convenient to work with the dual of $[\![\phi]\!]$ in $Exm([\wedge,\to]^n)$.

3.6.1.22. DEFINITION. *Let $[\![\phi]\!]$ be the valuation of formulas in $Exm([\wedge,\to]^n)$. Define $\alpha^n(\phi)=Exm([\wedge,\to]^n)\setminus[\![\phi]\!]$.*

3.6.1.23. Lemma. *Let ϕ and ψ be formulas in $[\wedge,\rightarrow]^n$. Then*

1. $\alpha^n(\phi) \subseteq \alpha^n(\psi) \quad \Leftrightarrow \quad \psi \vdash \phi$;
2. $\alpha^n(\phi \wedge \psi) = \alpha^n(\phi) \cup \alpha^n(\psi)$;
3. $\alpha^n(\phi{\rightarrow}\psi) = {\downarrow}(\alpha^n(\psi) \setminus \alpha^n(\phi))$.

3.6.1.24. Definition. *For a formula ϕ in $[\wedge,\rightarrow]^n$ we define $ucv^n(\phi)$,* the upper carrier of *ϕ, as the set of maximal elements in $\alpha^n(\phi)$.*

3.6.1.25. Lemma. *For $\phi \in [\wedge,\rightarrow]^n$ let $An^n(\phi)$ be the set of equivalence classes in $[\wedge,\rightarrow]^n$ that have a representative of the form $\psi{\rightarrow}\phi$, with $\psi \in [\wedge,\rightarrow]^n$. Then*

$$|An^n(\phi)| = |\mathcal{P}(ucv^n(\phi))| = 2^{|ucv^n(\phi)|}.$$

3.6.1.26. Theorem. *The number of equivalence classes in $[\rightarrow]^n$ is:*

$$\sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}N(n,k)$$

where $N(n,k) = 2^{|\bigcap\{ucv^n(p_i)|i\leq k\}|}$.

Proof. As in the case of theorem 3.5.1.31, we have to calculate the number of different subsets in the $ucv^n(p_i)$, a union of non-disjunct subsets. The summation above uses the symmetry in $Exm([\wedge,\rightarrow]^n)$. ⊣

3.6.1.27. Corollary. *The number of elements in $[\rightarrow]^3$ is:*

$$3.2^{23} - 3.2^3 + 1.2 = 25\,165\,802.$$

Proof. Use $Exm([\wedge,\rightarrow]^3)$ and determine $ucv^3(p)$, $ucv^3(q)$ and $ucv^3(r)$ and their intersections, to calculate $N(3,1) = 23$, $N(3,2) = 3$ and $N(3,3) = 1$. The corollary is a result of the substitution of these values in the formula of theorem 3.6.1.26. ⊣

Applying theorem 3.6.1.26 on $Exm([\wedge,\rightarrow]^4)$, Renardel de Lavalette calculated the cardinality of $Diag([\rightarrow]^4)$.

3.6.1.28. Fact. $|Diag([\rightarrow]^4)| = 2^{623\,662\,965\,552\,393} - 50\,331\,618$

Chapter 4

Restricted nesting of implication in IpL

4.1 Introduction

In Chapter 2 we introduced fragments of modal logic with restricted nesting of $\Box$ and showed how in the hierarchy of fragments $\mathbf{K}_m^n$ the types and semantic types of nodes in finite Kripke models could be defined. Semantic types were used to construct exact Kripke models of the fragments $\mathbf{K}_m^n$.

In **IpL** we will introduce a similar stratification of fragments $\mathbf{IpL}_m^n$ to obtain exact Kripke models. With the exception of $\mathbf{IpL}^1$ (and $\mathbf{IpL}^0$, which is the trivial fragment of the classes $\top$ and $\bot$), an exact model for $\mathbf{IpL}^n$ cannot exist (fact 2.5.0.15 in Chapter 2).

In the sequel we will show that restricting the nesting of implication to a maximum of m and confining the propositional variables to the $\{p_1, \ldots, p_n\}$ yields fragments $\mathbf{IpL}_m^n$ with a finite exact Kripke model.

4.2 Preliminaries

4.2.0.1. Definition. *The level of* nesting of the implication, $\mu(\phi)$, *of an* **IpL** *formula* ϕ *is defined inductively as:*

- $\mu(p) = 0$ *if* p *is an atomic formula;*
- $\mu(\psi \circ \chi) = \max\{\mu(\psi), \mu(\chi)\}$ *if* $\circ \in \{\wedge, \vee\}$*;*
- $\mu(\neg\psi) = \mu(\psi) + 1$*;*
- $\mu(\psi \to \chi) = \max\{\mu(\psi), \mu(\chi)\} + 1$.

The fragment $\mathbf{IpL}_m^n$ is defined as the fragment of **IpL** formulas ϕ with propositional variables restricted to $\{p_1, \ldots, p_n\}$, such that $\mu(\phi) \leq m$.

4.3 Semantic types in $\mathbf{IpL}^n_m$

The definition of a semantic type in $\mathbf{IpL}^n_m$ much resembles the definition in modal logic in Chapter 2.

4.3.0.1. Definition. *Let K be a finite* **IpL** *Kripke model. For $k \in K$ define inductively:*

1. $\tau^n_0(k) = \langle atom^n(k), \emptyset\rangle$;;
2. $\tau^n_{m+1}(k) = \langle atom^n(k), \{\tau^n_m(l) \mid k \leq l\}\rangle$.

As in previous chapters, we define $Th^n_m(k) = \{\phi \in \mathbf{IpL}^n_m \mid k \Vdash \phi\}$.

4.3.0.2. Theorem. *Let K and L be finite* **IpL** *Kripke models. If $k \in K$ and $l \in L$ then:*

$$\tau^n_m(k) = \tau^n_m(l) \quad \Leftrightarrow \quad Th^n_m(k) = Th^n_m(l).$$

Proof. By induction on m. $\Rightarrow$: For $m = 0$ the proof is obvious as we have $atom^n(k) = atom^n(l)$ iff for all $\phi \in \mathbf{IpL}^n_0$: $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$. (Note that the fragment $\mathbf{IpL}^n_0$ is the fragment $[\wedge, \vee]^n$ in the previous chapter).

Assume the theorem to be true for m and let $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$. To prove $Th^n_{m+1}(k) = Th^n_{m+1}(l)$ we will show for all $\phi \in \mathbf{IpL}^n_{m+1}$ we have $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \psi$.

Apply induction on the length of ϕ. In case ϕ is atomic, a conjunction or a disjunction the proof that $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$ is straightforward. So let $\phi = \psi{\rightarrow}\chi$ and assume $k \Vdash \psi{\rightarrow}\chi$. Note that both ψ and χ will be formulas in $\mathbf{IpL}^n_m$.

Let $l \leq h$ and $h \Vdash \psi$. As by definition $\tau^n_m(h) \in j_1(\tau^n_{m+1}(l))$ and $j_1(\tau^n_{m+1}(k)) = j_1(\tau^n_{m+1}(l))$, for some h' such that $k \leq h'$ we have $\tau^n_m(h) = \tau^n_m(h')$. From the first induction hypothesis ($\tau^n_m(k) = \tau^n_m(l) \Rightarrow Th^n_m(k) = Th^n_m(k)$) infer that $h' \Vdash \psi$. As $k \Vdash \psi{\rightarrow}\chi$ and $k \leq h'$ also $h' \Vdash \chi$. Again by the first induction hypothesis we may conclude that $h \Vdash \chi$. From which we conclude $l \Vdash \psi{\rightarrow}\chi$.

By interchanging the role of k and l this proof can also be used to prove the other direction: $l \Vdash \phi \quad \Rightarrow \quad k \Vdash \phi$.

As the case that ϕ is a negation is treated likewise, this completes the proof that for all $\phi \in \mathbf{IpL}^n_{m+1}$ we have $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \psi$.

$\Leftarrow$: Assume for all $\phi \in \mathbf{IpL}^n_{m+1}$ that $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$. We will again apply induction on m to prove $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$. Obviously $atom^n(k) = atom^n(l)$ and hence the case that $m = 0$ is simple.

For the induction step, assume $k \leq h$. So we may infer that $\tau^n_m(h) \in j_1(\tau^n_{m+1}(k))$. We will prove that for some h' such that $l \leq h'$ it is true that $\tau^n_m(h) = \tau^n_m(h')$. In this way we show that $j_1(\tau^n_{m+1}(k)) \subseteq j_1(\tau^n_{m+1}(l))$. As the proof for the inclusion in the other direction is in fact the same (interchanging k and l) and as $atom^n(k) = atom^n(l)$, this proves $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$.

As L is a finite model, let $\{l_0, \ldots, l_r\}$ be the finite set of successors of l (including l itself). For each l_i such that $\tau^n_m(l_i) \neq \tau^n_m(h)$, there is, by the induction hypothesis, some formula $\phi_i \in \mathbf{IpL}^n_m$ such that $h \Vdash \phi_i$ or $l \Vdash \phi_i$ but not both.

Define $\Phi = \bigwedge\{\phi_i \mid h \Vdash \phi_i\}$ and $\Psi = \bigvee\{\phi_i \mid h \nVdash \phi_i\}$. Clearly $\Phi{\rightarrow}\Psi \in \mathbf{IpL}^n_{m+1}$. If $\tau^n_m(h)$ would be different from all $\tau^n_m(l_i)$ then we would have $l \Vdash \Phi{\rightarrow}\Psi$. So by our

assumption that l and k force the same $\mathbf{IpL}^n_m$ formulas, also $k \Vdash \Phi{\rightarrow}\Psi$. But this would imply $h \Vdash \Phi{\rightarrow}\Psi$. As obviously $h \Vdash \Phi$ and $h \nVdash \Psi$ this is a contradiction. Hence we may conclude that for some l_i will have the same n,m-type as h. ⊣

4.3.0.3. Definition. *Define the order* $\preceq$ *between* n,m*-types as:*

1. $\tau^n_0(k) \preceq \tau^n_0(l)$ *if* $atom^n(k) \subseteq atom^n(l)$*;*
2. $\tau^n_{m+1}(k) \preceq \tau^n_{m+1}(l)$ *if* $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$ *or* $\tau^n_m(l) \in j_1(\tau^n_{m+1}(k))$.

4.3.0.4. Corollary. *Let* K *and* L *be finite* **IpL** *models such that* $k \in K$ *and* $l \in L$. *Then* $\tau^n_m(k) \preceq \tau^n_m(l)$ *implies* $Th^n_m(k) \subseteq Th^n_m(l)$.

Proof. Let k' be a new node, having k and l (and hence their successors) as its successors. Moreover let $atom^n(k') = atom^n(k)$. Note that as $\tau^n_m(k) \preceq \tau^n_m(l)$ also $atom^n(k) \subseteq atom^n(l)$ and hence we may take $\uparrow k'$ as a new Kripke model. Obviously $\tau^n_{m+1}(k') = \tau^n_{m+1}(k)$ and $Th^n_{m+1}(k') \subseteq Th^n_{m+1}(l)$. Theorem 4.3.0.2 assures us that $Th^n_{m+1}(k') = Th^n_{m+1}(k)$. ⊣

Before defining the type formulas $\phi^n_m(k)$ in $\mathbf{IpL}^n_m$ let us draw some conclusions from this theorem for the structure of the exact Kripke model of $\mathbf{IpL}^n_m$ and give an example.

Let T^n_m be the set of n,m-types in $\mathbf{IpL}^n_m$ and let $Th^n_m(k) = \{\phi \in \mathbf{IpL}^n_m \mid k \Vdash \phi\}$. Obviously T^n_m is finite.

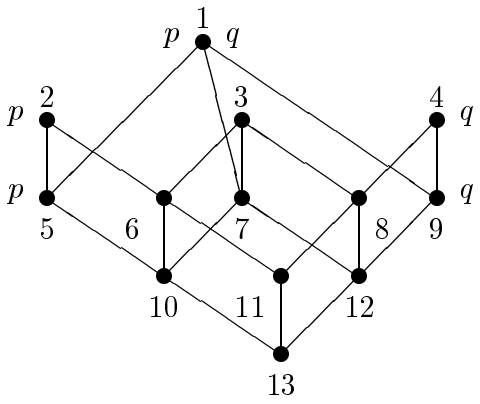
4.3.0.5. Definition. *Define* $Exm(\mathbf{IpL}^n_m)$ *as the Kripke model with* T^n_m *as its domain,* $\preceq$ *as its accessibility relation and* $atom^n(t) = j_0(t)$ *as its valuation.*

4.3.0.6. Theorem. $Exm(\mathbf{IpL}^n_m)$ *is the exact Kripke model of* $\mathbf{IpL}^n_m$.

Proof. Obviously $Exm(\mathbf{IpL}^n_m)$ is a finite **IpL** n-model. By induction on m is easily proved that if $t \in T^n_m$ is an n,m-type, in $Exm(\mathbf{IpL}^n_m)$ we have $\tau^n_m(t) = t$. Hence $Exm(\mathbf{IpL}^n_m)$ is a model realizing exactly all n,m-types in $\mathbf{IpL}^n_m$. ⊣

4.3.0.7. Corollary. *The exact Kripke model of* $\mathbf{IpL}^n_m$ *is unique up to isomorphism.*

Proof. If M is some exact Kripke model of $\mathbf{IpL}^n_m$ the mapping $\tau^n_m : M \mapsto Exm(bfIpL^n_m)$ is an isomorphism. ⊣



26. FIGURE. *$Exm(\mathbf{IpL}_1^2)$, the exact Kripke model of $\mathbf{IpL}_1^2$.*

The irreducible formulas in the exact model of $\mathbf{IpL}_1^2$ are:

1. $p \wedge q$	5. p	9. q	13. $p \to p$
2. $p \wedge \neg q$	6. $\neg q$	10. $q \to p$	
3. $\neg p \wedge \neg q$	7. $p \leftrightarrow q$	11. $\neg(p \wedge q)$	
4. $\neg p \wedge q$	8. $\neg p$	12. $p \to q$	

The exact model of $\mathbf{IpL}_1^2$ was first constructed by Zwanenburg, using the subset of $\wedge$-irreducible formulas in the set of $\vee$-irreducible formulas of $\mathbf{IpL}_1^2$ as a 'skeleton' [Zwanenburg 94].

The exact model can be used to calculate the 98 equivalence classes in the diagram of $\mathbf{IpL}_1^2$, as listed in appendix B.3.

The exact model of $\mathbf{IpL}_2^2$ has 718 elements.

4.4 The n, m-types in IpL

As in case of modal logic we will introduce formulas $\phi_m^n(k)$ for the n, m-type of a node k in a finite Kripke model. We first will define the $\phi_m^n(k)$ and then prove that such a formula is indeed an axiom of $Th_m^n(k)$, the theory of n, m-formulas forced by the node k.

4.4.0.1. DEFINITION. *For a node k in a finite* **IpL** *model, define $\phi_m^n(k)$ inductively as:*

1. $\phi_0^n(k) = \bigwedge atom^n(k)$;
2. $\phi_{m+1}^n(k) =$
   $\bigwedge\{\phi_m^n(l) \to \bigvee\{\phi_m^n(h) \mid k \leq h \text{ and } \phi_m^n(l) \nvdash \phi_m^n(h)\} \mid \tau_{m+1}^n(k) \not\preceq \tau_{m+1}^n(l)\}$.

4.4.0.2. LEMMA. *Let K and L be finite* **IpL** *Kripke models. Assume that $k \in K$ and $l \in L$. Then $\tau_m^n(k) \preceq \tau_m^n(l)$ implies $Th_m^n(k) \subseteq Th_m^n(l)$*

Proof. The case $m = 0$ is obvious. For $m > 0$, by the definition of $\preceq$ we have $\tau_m^n(k) \preceq \tau_m^n$. So apply corollary 4.3.0.4. $\dashv$

4.4.0.3. THEOREM. *Let K and L be finite* **IpL** *Kripke models such that $k \in K$ and $l \in L$. Then:*

$$l \Vdash \phi^n_m(k) \quad \Leftrightarrow \quad \tau^n_m(k) \preceq \tau^n_m(l).$$

Proof. By induction on m. The case $m = 0$ is simple.

So assume $l \Vdash \phi^n_{m+1}(k)$ and let $\tau^n_{m+1}(k) \not\preceq \tau^n_{m+1}(l)$. From the induction hypothesis we know that $l \Vdash \phi^n_m(l)$. By definition of $\phi^n_{m+1}(k)$, infer that $l \Vdash \bigvee\{\phi^n_m(h) \mid k \leq h \text{ and } \phi^n_m(l) \nvdash \phi^n_m(h)\}$.

Hence for some h such that $k \leq h$ we have $l \nvdash \phi^n_m(h)$ and $l \Vdash \phi^n_m(h)$. This is a contradiction as $l \Vdash \phi^n_m(h)$ implies $\phi^n_m(l) \vdash \phi^n_m(h)$. To prove this, take g a node in some **IpL** Kripke model such that $g \Vdash \phi^n_m(l)$. By induction hypothesis $\tau^n_m(l) \preceq \tau^n_m(g)$. So, by the previous lemma, conclude that $g \Vdash \phi^n_m(h)$. Hence, by the completeness theorem for **IpL**, it follows that $\phi^n_m(l) \vdash \phi^n_m(h)$. So $l \Vdash \phi^n_{m+1}(k)$ implies $\tau^n_{m+1}(k) \preceq \tau^n_{m+1}(l)$.

For the other direction, assume that $\tau^n_{m+1} \preceq \tau^n_{m+1}(l)$. We will use the previous lemma and show $k \Vdash \phi^n_{m+1}(k)$ to prove that $l \Vdash \phi^n_{m+1}(k)$.

Let $k \leq h$. If $h \Vdash \phi^n_m(l)$ then, by induction hypothesis, we know $\tau^n_m(l) \preceq \tau^n_m(h)$. Assume $\phi^n_m(l) \Vdash \phi^n_m(h)$. Then $l \Vdash \phi^n_m(h)$ and so, again by induction hypothesis, $\tau^n_m(h) \preceq \tau^n_m(l)$. Hence we would have $Th^n_m(h) = Th^n_m(l)$ and by the theorem in the previous section, $\tau^n_m(h) = \tau^n_m(l)$. Obviously this implies $\tau^n_m(k) \preceq \tau^n_m(l)$.

Hence $h \Vdash \phi^n_m(h)$ and $\phi^n_m(l) \vdash \phi^n_m(h)$. So, for $\tau^n_{m+1}(k) \not\preceq \tau^n_{m+1}(l)$ we have:

$$k \Vdash \phi^n_m(l) \rightarrow \bigvee\{\phi^n_m(k) \mid k \leq h \text{ and } \phi^n_m(l) \nvdash \phi^n_m(k)\}.$$

Which we had to prove. ⊣

4.4.0.4. COROLLARY. *For a node k in a finite* **IpL** *Kripke model K the formula $\phi^n_m(k)$ is an axiom of the theory $Th^n_m(k)$.*

Proof. Let $\psi \in Th^n_m(k)$ and assume for some node l in a finite **IpL** Kripke model L that $l \Vdash \phi^n_m(k)$. By the theorem above we have $\tau^n_m(k) \preceq \tau^n_m(l)$ and so $l \Vdash \psi$. ⊣

4.4.0.5. COROLLARY. *If $\phi^n_m(k) \equiv \phi^n_m(l)$ then $\tau^n_m(k) = \tau^n_m(l)$.*

Proof. Obvious, as $Th^n_m(k) = Th^n_m(l)$. ⊣

Chapter 5

Exactly provable L formulas

5.1 Introduction

In this chapter we will study the exactly provable formulas in fragments of provability logic **L** (**GL** in [Boolos 93], **PRL** in [Smoryński 85]). According to Solovay's theorem [Solovay 76] on provability interpretations the theorems of the provability logic **L** are precisely those modal formulas that are provable in **PA** under arbitrary arithmetical interpretations (interpreting $\Box$ as the formalized provability predicate in **PA**). The logic **L** is also known to be the logic of the diagonizable algebras, recently also called Magari algebras. Here, we are concerned with the finitely generated Magari algebras that are embeddable in the Magari algebra of Peano Arithmetic. Shavrukov [Shavrukov 93] characterized these subalgebras, which are recursively enumerable, as having the so-called strong disjunction property.

In the context of the present work the terminology of propositional theories (i.e. sets of propositional modal formulas closed under modus ponens and necessitation) is more convenient.

Let us introduce a new derivability relation to distinguish between the usual modal theories (closed under modus ponens) and the modal theories that are in addition closed under necessitation.

5.1.0.1. Definition. *Let ϕ and ψ be modal formulas. Define:*

$$\phi \vdash \psi \quad \Leftrightarrow \quad \phi \wedge \Box\phi \vdash \psi.$$

A propositional theory in **L** will here be a set of propositional formulas closed under $\vdash$. Rephrased in the terminology of propositional theories, we study those theories T over **L** in a finite number of propositional variables that are (faithfully) interpretable in **PA**. Theories correspond to τ-filters in the free Magari algebras and interpretability to embeddability as a subalgebra. Interpretable theories T in $p_1, \cdots, p_n$ are those propositional theories in $p_1, \cdots, p_n$ for which there is a sequence of arithmetical sentences $A_1, \cdots, A_n$ such that an **L** formula ψ is an $\vdash$ consequence of T iff ψ^* is a theorem of **PA** in the arithmetical interpretation $*$ in which the atomic formula p_i is

interpreted as A_i (see e.g. [Solovay 76], [Boolos 93] or [Smoryński 85]). Written out: T axiomatizes an arithmetically interpreted theory:

$$\{\psi \mid T \vdash \psi\} = \{\psi \mid \vdash_{PA} \psi^*(A_1, \cdots, A_n)\}.$$

The faithfully interpretable propositional theories T in $\mathbf{L}^n$ (i.e., **L** restricted to the language of $p_1, \cdots, p_n$) are according to Shavrukov the consistent recursively enumerable (r.e.) theories that satisfy *the strong disjunction property*: $T \vdash \Box\psi \vee \Box\chi$ implies $T \vdash \psi$ or $T \vdash \chi$. (Parenthetically: interpretable theories in infinitely many propositional variables need not be r.e.) The strong disjunction property may be thought of as being composed out of the simple disjunction property: $T \vdash \Box\psi \vee \Box\chi$ implies $T \vdash \Box\psi$ or $T \vdash \Box\chi$, and *ω-consistency*: $T \vdash \Box\psi$ implies $T \vdash \psi$.

An older concept to which this can be related is the concept of *exact provability* introduced in [De Jongh 82] (see also [JC 95]): in the terminology used here[1] a formula can be defined to be exactly provable if it axiomatizes an interpretable theory. That means that an exactly provable formula of **L** is a formula ϕ which axiomatizes an arithmetically interpreted propositional theory:

$$\{\psi \mid \phi \vdash \psi\} = \{\psi \mid \vdash_{PA} \psi^*(A_1, \cdots, A_n)\}.$$

One of the objects of our research is to get an overview of exactly provable formulas of low complexity aided by computerized calculations. For that purpose the semantic characterizations in terms of Kripke-models and (semantic) types developed in the previous chapters will be applied to interpretable theories and exactly provable formulas. It turns out that an important role is played by *maximal exactly provable formulas*, i.e. exactly provable formulas that are not implied by any other exactly provable formula, and, more in general, by *maximal theories with the* strong disjunction *property*. The characterizations of these concepts discussed in this chapter make heavy use of the relationship between exactly provable formulas in provability logic and sets of finite *types* of modal formulas as introduced in Chapter 2.

This chapter is built up as follows. After a preliminary section 5.2, characterizations of interpretable theories and exactly provable formulas are given in section 5.3. Maximal exactly provable formulas are discussed in section 5.4. In the last section 5.5, it is shown how the theory was applied to calculate the 62 exactly provable formulas in one propositional variable of modal complexity 1, and the 8 maximal ones among them.

5.2 Preliminaries

The *provability logic* **L** is the modal propositional logic with as its axioms the ones of classical propositional logic as well as all formulas of the forms $\Box(\phi\rightarrow\psi)\rightarrow(\Box\phi\rightarrow\Box\psi)$ and $\Box(\Box\phi\rightarrow\phi)\rightarrow\Box\phi$, and the inference rules modus ponens and necessitation. As

[1] In [HJ 96] exactly provable formulas were called *exact* formulas. In the present context this terminology might suggest a connection with exact models which does not exist.

usual $\Diamond\psi$ is defined as $\neg\Box\neg\psi$, and we will use the abbreviation $\boxdot\phi$ for the formula $\phi \wedge \Box\phi$. Note that in $\mathbf{L}$ $\phi \vdash \psi$ is equivalent to $\boxdot\phi \vdash \psi$.

We say '*ϕ is interderivable with ψ*' and write $\phi \equiv \psi$ for the conjunction of $\phi \vdash \psi$ and $\psi \vdash \phi$. Note that this implies that always $\phi \equiv \boxdot\phi$. We reserve the terminology '*ϕ is equivalent to ψ*' for $\vdash \phi \leftrightarrow \psi$.

Propositional theories in $\mathbf{L}^n$ will here be sets of propositional formulas closed under $\vdash$. Such a propositional theory T is called *consistent* if $T \nvdash \bot$.

By its completeness theorem, $\mathbf{L}$ is the logic of all finite, transitive and irreflexive Kripke-models (a proof can be found in [Boolos 93] and [Smoryński 85]).

Recall from Chapter 2 the definition of $\beta(\phi)$, the *modal degree* of a formula ϕ. The definition of semantic types and type formulas in $\mathbf{L}^n_m$ will be essentially the same as in $\mathbf{K}^n_m$ (see Chapter 2).

5.2.0.1. Definition. *Let k be a node in a finite, transitive and irreflexive Kripke model. Then, $\tau^n_m(k)$, the n,m-*type *of k (in $\mathbf{L}$) is defined by:*

- $\tau^n_0(k) = \langle atom^n(k), \emptyset \rangle$;
- $\tau^n_{m+1}(k) = \langle atom^n(k), \{\, \tau^n_m(l) \mid kRl\}\rangle$.

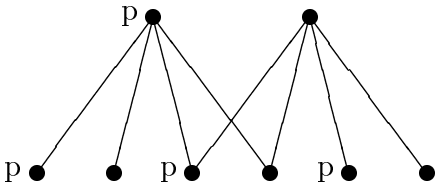
The set of all such n,m-types is written T^n_m. Define $\phi^n_m(k)$, the $\mathbf{L}^n_m$ type *of k inductively as:*

- $\phi^n_0(k) = \phi^n_{\mathbf{CpL}}(k)$
- $\phi^n_{m+1}(k) = \phi^n_{\mathbf{CpL}}(k) \wedge \bigwedge\{\Diamond\phi^n_m(l) \mid kRl\} \wedge \Box\bigvee\{\phi^n_m(l) \mid kRl\}$

One easily verifies that the $n, m+1$-type of k, $\tau^n_{m+1}(k)$ uniquely determines the n,m-type of k. If t is an $n, m+1$-type, let us write $t\restriction m$ for the corresponding n,m-type. The following fact will be useful in the sequel of this chapter.

5.2.0.2. Fact. *Let k be a node in a finite, transitive and irreflexive Kripke model. If $m \leq l$ then $\tau^n_m(k) = \tau^n_l(k)\restriction m$.*

As in that chapter was done for $\mathbf{K}$, the fragment $\mathbf{L}^n_m$ will be the fragment of formulas ϕ in $\mathbf{L}^n$ such that $\beta(\phi) \leq m$. It can be proved that there exists an exact model



for each $\mathbf{L}^n_m$.

27. Figure. *The construction of ExL^1_0 and ExL^1_1.*

For the infinite fragment $\mathbf{L}^n$ there is no such exact model, but as in case of $\mathbf{K}$ in Chapter 2, there is a canonical (infinite) model ExL^n which is n-complete. It gives considerable insight into the free Magari algebra over n generators.

It is convenient to us to execute most of our constructions inside this model. Many of these constructions are applicable more generally, however.

Let us write $\Box^0\phi = \phi$ and $\Box^{n+1}\phi = \Box\Box^n\phi$. The following facts about the nodes of ExL^n will be useful in the sequel.

5.2.0.3. Facts.

1. $\delta(k) = m \quad \Leftrightarrow \quad k \Vdash \neg\Box^m\bot \wedge \Box^{m+1}\bot$;
2. *If* $\delta(k), \delta(l) \leq m$ *and* $\tau^n_m(k) = \tau^n_m(l)$, *then* $k = l$;
3. *If* $\delta(k) = m$ *and* $l \Vdash \phi^n_m(k) \wedge \neg\Box^m\bot \wedge \Box^{m+1}\bot$, *then* $k = l$.

These facts suggest a kind of normal form for the irreducible formulas corresponding to the elements of ExL^n.

5.2.0.4. Definition. *Let* $k \in ExL^n$ *and assume* $\delta(k) = m$.
Then $\phi^n(k) = \phi^n_m(k) \wedge \neg\Box^m\bot \wedge \Box^{m+1}\bot$.

From the n-completeness of ExL^n we conclude that for $k \in ExL^n$ the $\phi^n(k)$ are the irreducible elements in $\mathbf{L}^n$.

5.3 Exactly provable formulas in $\mathbf{L}^n$

As stated in the introduction, Shavrukov's theorem in [Shavrukov 93] gives a characterization of the exactly provable formulas in **L**.

5.3.0.1. Fact. *A formula* $\phi \in \mathbf{L}$ *is exactly provable iff* ϕ *is not a contradiction and has the* strong disjunction property (*is* s.d.):

$$\forall \psi, \chi \in L\,(\phi \vdash \Box\psi \vee \Box\chi \quad \Rightarrow \quad \phi \vdash \psi \quad \textit{or} \quad \phi \vdash \chi).$$

The property in this fact is called *steady* by Shavrukov [Shavrukov 93]. Whether a formula $\phi \in \mathbf{L}^n_m$ is steady or not, Shavrukov [Shavrukov 93] also proved, depends only on its behavior with regard to other formulas in $\mathbf{L}^n_m$:

5.3.0.2. Fact. *A formula* $\phi \in \mathbf{L}^n_m$ *is exactly provable iff* ϕ *is not a contradiction and is s.d. for formulas in* $\mathbf{L}^n_m$:

$$\forall \psi, \chi \in \mathbf{L}^n_m(\phi \vdash \Box\psi \vee \Box\chi \quad \Rightarrow \quad \phi \vdash \psi \quad \textit{or} \quad \phi \vdash \chi)$$

For a simple proof of this last fact see [Zambella 94]. We will transform this characterization of exact provability into a semantic one. This characterization does not work if we are not only interested in exactly provable formulas, but want to study interpretable theories in general (see [HJ 96]).

5.3.0.3. Definition. $\omega^n(\phi) = \{k \in ExL^n \mid k \Vdash \boxdot\phi\}$.
If T *is a propositional theory in* $\mathbf{L}^n$, *then* $\omega^n(T) = \{k \in ExL^n \mid k \Vdash T\}$.

Obviously $\omega^n(\phi)$ and $\omega^n(T)$ will always be *closed upwards* in the sense that, if e.g. $k \in \omega^n(\phi)$ and $k < l$, then $l \in \omega^n(\phi)$.

5.3.0.4. Theorem. *A formula* $\phi \in \mathbf{L}^n$ *is exactly provable iff* $\omega^n(\phi)$ *is non-empty and* downwards directed, *i.e.* $\forall k, l \in \omega^n(\phi)\, \exists h \in \omega^n(\phi)\, (h < k \,\&\, h < l)$.

Proof. $\Rightarrow$: Let ϕ be an exactly provable formula in $\mathbf{L}^n$. As ϕ unequals the contradiction by definition, we have $\omega^n(\phi) \neq \emptyset$ by the completeness of ExL^n. To prove the second condition, let $k, l \in \omega^n(\phi)$, and $\phi^n(k), \phi^n(l)$ be (representatives of) the irreducible classes in $\mathbf{L}^n$ corresponding to k and l. Assume that, if $h \in \omega^n(\phi)$ and $h < k$, then $h \not< l$. Then, again by the completeness of ExL^n, we would have $\phi \vdash \Diamond\phi^n(k) \rightarrow \Box\neg\phi^n(l)$, or equivalently $\phi \vdash \Box\neg\phi^n(k) \vee \Box\neg\phi^n(l)$. As ϕ is supposed to be exactly provable, ϕ would either prove $\neg\phi^n(k)$ or $\neg\phi^n(l)$, in contradiction with the assumption that $k, l \in \omega^n(\phi)$. Hence, there should be an $h \in \omega^n(\phi)$ such that $h < k$ and $h < l$.
$\Leftarrow$: Let $\psi, \chi \in \mathbf{L}^n$ and $\phi \vdash \Box\psi \vee \Box\chi$, and assume there are $k, l \in \omega^n(\phi)$ such that $k \nVdash \psi$ and $l \nVdash \chi$. By the last condition of the theorem, there is an $h \in \omega^n(\phi)$ such that $h < k$ and $h < l$. As we would then have $h \nVdash \Box\phi \vee \Box\psi$, we obtain a contradiction. Hence, we proved that $\phi \vdash \psi$ or $\phi \vdash \chi$. $\dashv$

By the completeness of $\mathbf{L}$ non-interderivable ϕ and ψ give rise to distinct $\omega^n(\phi)$ and $\omega^n(\psi)$. This is in general not so for theories. An example is the theory axiomatized by p on the one hand, and the theory T_1 axiomatized by $\Box^m\bot\rightarrow p$ for each m, on the other. The sets $\omega^1(p)$ and $\omega^1(T_1)$ are the same, consisting of all nodes that together with all their successors force p, but clearly the theories are not: p is not a consequence of T_1. Similarly, the theory $T_2 = \Box^m\bot\rightarrow\Box p \vee \Box\neg p$ for each m can be shown to have the strong disjunction property. But not all pairs of nodes in $\omega^1(T_2)$ have a common predecessor in ExL^1, because $\omega^1(T_2)$ consists of those nodes that together with all their successors force p and those nodes that together with all their successors don't force p. This shows that the semantic characterization of exact provability does not generalize to interpretability of non-finitely axiomatizable theories, at least if one doesn't freely use infinite models. For a restricted class of theories that does respect the characterization see [HJ 96].

Note that the $\omega^n(\phi)$ of an exactly provable $\phi \in \mathbf{L}^n$ is infinite by the conditions of the characterization. On the other hand there is a simple correspondence between such an infinite set and a finite set of n, m-types in ExL^n:

5.3.0.5. Definition. *Let ϕ be an $\mathbf{L}^n$ formula.*
Then $T^n_m(\phi) = \{\tau^n_m(k) \mid k \in ExL^n, k \Vdash \boxdot\phi\}$.

5.3.0.6. Lemma. *Let ϕ and ψ be $\mathbf{L}^n_m$ formulas.*
Then $T^n_m(\phi) = T^n_m(\psi)$ *iff* $\phi \equiv \psi$.

Proof. For the non-trivial direction, from left to right, let $T^n_m(\phi) = T^n_m(\psi)$, and assume $k \Vdash \boxdot\phi$. Then $\tau^n_m(k) \in T^n_m(\phi) = T^n_m(\psi)$. Hence $\tau^n_m(k) = \tau^n_m(k')$ for some k' that forces $\boxdot\psi$. So, $k' \Vdash \psi$ and, since $\psi \in \mathbf{L}^n_m$, $k \Vdash \psi$. The rest is evident. $\dashv$

5.3.0.7. Lemma. *Let ϕ be an $\mathbf{L}^n_{m+k}$ formula.*
Then $T^n_m(\phi) = \{t\restriction m \mid t \in T^n_{m+k}(\phi)\}$.

Proof. Obvious, considering fact 5.2.0.2. $\dashv$

5.3.0.8. LEMMA. *For each* $\mathbf{L}^n$ *formula* ϕ *and each* m *there is a finite upwardly closed subset* K *of* $\omega^n(\phi)$ *such that the elements of* K *exactly realize* $T^n_m(\phi)$*, i.e.,* $T^n_m(\phi)=\{\tau^n_m(k)|\ k\in K\}$.

Proof. Just take any finite subset of $\omega^n(\phi)$ such that its elements exactly realize $T^n_m(\phi)$. The upward closure of this set will do, because its elements also force $\boxdot\phi$. $\dashv$

To find the sets of n,m-types suitable for exactly provable formulas ϕ, we have to translate the conditions on the $\omega^n(\phi)$ of exactly provable ϕ into conditions on the underlying set of n,m-types. For example, for a finite $T^n_m(\phi)$ to correspond to an infinite $\omega^n(\phi)$, it is necessary that some type in $T^n_m(\phi)$ can have itself as a successor. To describe this kind of reflexivity we introduce the notion of a *reflexive type*.

5.3.0.9. DEFINITION. *A type* $t\in T^n_{m+1}$ *is called* reflexive *if* $t\restriction m\in j_1(t)$.

The following theorem is related to lemma 5.13 of [Shavrukov 93].

5.3.0.10. THEOREM. *A formula* $\phi\in\mathbf{L}^n_m$ *with* $m>0$ *is exactly provable iff there is a type* $t\in T^n_m(\phi)$ *such that* $j_1(t)=T^n_{m-1}(\phi)$*, which, of course, makes* t *a reflexive type.*

Proof. $\Rightarrow$: Let $\phi\in\mathbf{L}^n_m$ be an exactly provable formula. Note that $T^n_{m-1}(\phi)$ is a finite set of types. Let $K\subset\omega^n(\phi)$ be finite and closed upwards such that $\{\tau^n_m(k)\,|\,k\in K\}=T^n_m(\phi)$, as guaranteed to exist by lemma 5.3.0.8. According to theorem 5.3.0.4 we can find an $h\in\omega^n(\phi)$ below all of the elements of K. By lemma 5.3.0.7 this h must have a type as required.
$\Leftarrow$: Assume ϕ and t to fulfill the conditions given. As $T^n_m(\phi)\neq\emptyset$, also $\omega^n(\phi)\neq\emptyset$. Suppose $k,l\in\omega^n(\phi)$. Let K be a finite upwardly closed subset of $\omega^n(\phi)$ such that $k,l\in K$ and $\{\tau^n_{m-1}(k')\,|\,k'\in K\}=T^n_{m-1}(\phi)$ (compare lemma 5.3.0.8). Consider a world h just below this K such that $\{p\in P^n\,|\,h\Vdash p\}=j_0(t)$. It will be clear that $\tau^n_m(h)=t$ and (since ϕ is assumed to be an $\mathbf{L}^n_m$ formula) this proves $h\in\omega^n(\phi)$. Of course, $h<k$ and $h<l$, so the conditions of theorem 5.3.0.4 apply to $\omega^n(\phi)$. $\dashv$

The theory developed in this chapter and in Chapter 2 has enabled us to calculate the exactly provable formulas in $\mathbf{L}^1_1$. This will be explained in more detail in the last section. It will be shown that already in this very first small fragment there are 62 non-interderivable members. It turned out that it was worthwile to single out the 8 maximal elements of these 62.

5.4 Maximal exactly provable formulas

This section will be devoted to maximal exactly provable formulas. First we will have to sharpen our semantic characterization of exactly provable formulas. Let us exploit the relationship between irreducible formulas and semantic types to write $\phi^n_m(t)$ for the $\phi^n_m(k)$ with $\tau^n_m(k)=t$.

5.4.0.1. DEFINITION. *Let* C *be a set of* n,m*-types.*
Then $\phi^n_m(C)=\bigvee\{\phi^n_m(t)\,|\,t\in C\}$.

Recall that $T^n_m(\phi) = \{\tau^n_m(k) \mid k \Vdash \Box\phi\}$.

5.4.0.2. LEMMA. *If $\phi \in \mathbf{L}^n_m$, then $\phi \equiv \phi^n_m(T^n_m(\phi))$.*

Proof. Immediate from lemma 5.3.0.6 as soon as one realizes that $T^n_m(\phi^n_m(T^n_m(\phi))) = T^n_m(\phi)$. ⊣

5.4.0.3. LEMMA. *If $C \subseteq T^n_m$ $(m > 0)$, then $C = T^n_m(\phi)$ for an exactly provable formula $\phi \in \mathbf{L}^n_m$ iff*

1. *There is a finite upwards closed $K \subseteq ExL^n$ such that $C = \{\tau^n_m(k) \mid k \in K\}$ (we will call C* upwards closed realizable*)*
2. *There is $t \in C$ such that $\forall t' \in C\,(t' \restriction (m-1) \in j_1(t))$. Such a type t will be called an* enveloping type *for C.*

Moreover, in that case $\phi \equiv \phi^n_m(C)$.

Proof. $\Rightarrow$: If $\phi \in \mathbf{L}^n_m$ is exactly provable, then $T^n_m(\phi)$ will have the required property 1 by the definition of $T^n_m(\phi)$, property 2 by theorem 5.3.0.10, and satisfies the final requirement by lemma 5.4.0.2.
$\Leftarrow$: We prove that $\phi^n_m(C)$ is an exactly provable formula. To apply theorem 5.3.0.10 to $\phi^n_m(C)$, we have to find an appropriate reflexive n,m-type. By the assumption on C, there is an n,m-type $t \in C$ such that $\forall t' \in C\,(t' \restriction (m-1) \in j_1(t))$. Let K be the upwardly closed realization of the types in C as assumed in the first conditon of this lemma. Note that K realizes precisely the $n,m-1$-types in $\{t' \restriction (m-1) \mid t' \in C\}$ (compare lemma 5.3.0.7). Let k be a (new) root immediately below K such that k forces exactly the elements of $j_0(t)$. Then $\tau^n_m(k) = t$. So, $k \Vdash \Box\phi^n_m(C)$ and, hence, t is a member of $T^n_m(\phi^n_m(C)) \subseteq C$ and a type appropriate for the application of theorem 5.3.0.10. ⊣

We will prove that the maximal exactly provable formulas in $\mathbf{L}^n$ correspond to what we will call *tail models* in ExL^n. Clearly this is a result that is, to a large extent, bound to the particular model ExL^n.

5.4.0.4. DEFINITION. *$K \subset ExL^n$ is called a* tail model *iff:*

1. *K is closed upwards;*
2. *there is an $m \in \omega$ such that $\{k \in K \mid \delta(k) \geq m\}$ is linearly ordered by $<$ and all nodes of this set force the same atoms.*

If $k \in ExL^n$, then we write $\underline{\uparrow}k\downarrow$ for the tail model consisting of $\underline{\uparrow}k$ and a tail descending from k with the forcing of the atoms as in k.

Our definition of tail model slightly differs from the one in [Visser 84] in that Visser's tail models are equipped with a minimal (infinite-depth) element.

5.4.0.5. LEMMA. *If $\phi \in \mathbf{L}^n_m$, $k \in \omega^n(\phi)$ and k has a reflexive n,m-type, then $\underline{\uparrow}k\downarrow \subseteq \omega^n(\phi)$.*

Proof. First note that all elements of the tail have the same n,m-type as k. Hence, all these nodes force ϕ, and consequently $\boxdot\phi$. $\dashv$

5.4.0.6. Lemma. *If $K \subset ExL^n$ is a tail model, then $K = \omega^n(\phi)$ for some ϕ in $\mathbf{L}^n$.*

Proof. Let K be $\uparrow\!\!\downarrow k\downarrow$, k having depth m, and let θ be the conjunction of the propositional variables and negations of propositional variables as they are forced on k. Then $K = \omega^n(\phi)$ for ϕ defined as the conjunction of $\Box^{m+1}\bot \to \bigvee\{\phi^n(k') \mid k \leq k'\}$ and $\neg\Box^{m+1}\bot \to \theta \wedge \Diamond\phi^n(k)$. $\dashv$

5.4.0.7. Lemma. *If $\phi \in \mathbf{L}^n$ and $\omega^n(\phi)$ is a tail model, then ϕ is maximal exactly provable.*

Proof. Assume $\omega^n(\phi)$ is a tail model and $\phi \in \mathbf{L}^n_m$. Since $\omega^n(\phi)$ is infinite and $T^n_{m-1}(\phi)$ finite it is obvious that the tail has to contain elements appropriate for an application of theorem 5.3.0.10. This shows that ϕ has to be exactly provable. Assume ψ to be an exactly provable formula such that $\psi \vdash \phi$, i.e., such that $\omega^n(\psi) \subseteq \omega^n(\phi)$. Then, because $\omega^n(\psi)$ is non-empty and downwards directed it has to contain the tail elements from a certain node downwards, and, because it it is closed upwards it has to contain all other elements of $\omega^n(\phi)$, which means that ϕ and ψ are interderivable. Hence, ϕ is maximal exactly provable. $\dashv$

5.4.0.8. Lemma. *If $\phi \in \mathbf{L}^n$, then there exists a formula $\psi \in \mathbf{L}^n$ such that $\omega^n(\psi) \subseteq \omega^n(\phi)$ and $\omega^n(\psi)$ is a tail model.*

Proof. Let $\phi \in \mathbf{L}^n$ and assume $t \in T^n_m(\phi)$ is a reflexive type with the properties guaranteed to exist by theorem 5.3.0.10. Now, take as in the proof of lemma 5.4.0.3($\Leftarrow$) $k \in \omega^n(\phi)$ with n,m-type t such that $\tau^n_m(\uparrow\!\!\downarrow k) = T^n_m(\phi)$. By lemma 5.4.0.5, $\uparrow\!\!\downarrow k\downarrow \subseteq \omega^n(\phi)$. By lemma 5.4.0.6, there exists a ψ with $\omega^n(\psi) = \uparrow\!\!\downarrow k\downarrow \subseteq \omega^n(\phi)$. $\dashv$

From lemma 5.4.0.8 it follows immediately that any $\mathbf{L}^n$ formula ϕ is determined uniquely (up to interderivability) by the maximal exactly provable $\mathbf{L}^n$ formulas that imply it. Certainly this does not generalize to interpretable theories. The s.d. theory T_1 axiomatized by $\Box^n\bot \to p$ for each n that was introduced after theorem 5.3.0.4 provides a counter-example. Its only maximal s.d. extension is the one axiomatized by p. Also, lemma 5.4.0.8 does not, in general imply that ϕ is equivalent to a finite disjunction of maximal exactly provable formulas (each preceded by $\boxdot$), although that may very well be the case. A counter-example is provided by the formula $\top$.

5.4.0.9. Theorem. *If $\phi \in \mathbf{L}^n$, then ϕ is maximal exactly provable in $\mathbf{L}^n$ iff $\omega^n(\phi)$ is a tail model in ExL^n.*

Proof. The direction from right to left follows from lemma 5.4.0.7. The other direction from 5.4.0.8 using the simple fact that, if one tail model is part of another, they have to be equal. $\dashv$

From lemma 5.4.0.6 and theorem 5.4.0.9 it is clear that there is a one-one correspondence between maximal exactly provable formulas and tail models.

Also from theorem 5.4.0.9, it follows that maximal exactly provable formulas in p cannot be symmetric with regard to p and $\neg p$ as the tail is always asymmetric. We follow with some additional properties and problems concerning maximal exactly provable formulas.

5.4.0.10. THEOREM. *If a formula $\phi \in \mathbf{L}^n_m$ is maximal exactly provable, then there is precisely one reflexive type t in $T^n_m(\phi)$. Moreover, $T^n_{m-1}(\phi) = j_1(t)$.*

Proof. The last part follows immediately from theorem 5.3.0.10. Assume $\phi \in \mathbf{L}^n_m$ with $m > 0$ is maximal exactly provable. Assume s and s' to be two distinct n, m-types in $T^n_m(\phi)$. If k and k' in $\omega^n(\phi)$ realize s and s' respectively, then, by lemma 5.4.0.5, $\underline{\uparrow} k \downarrow$ and $\underline{\uparrow} k' \downarrow$ are two distinct tail models within $\omega^n(\phi)$. This contradicts the fact that $\omega^n(\phi)$ is a tail model. ⊣

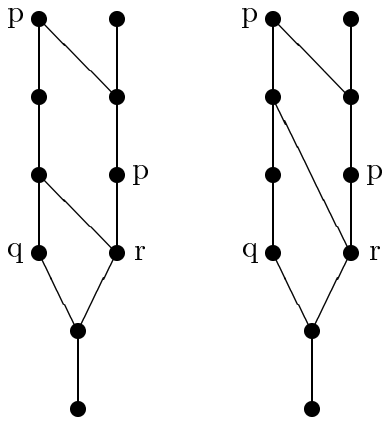
Examples of non-maximal exactly provable $\mathbf{L}^1_1$ formulas with exactly one reflexive $1, 1$-type will be given in the table in the last section.

5.4.0.11. DEFINITION. *An exactly provable $\mathbf{L}^n_m$ formula ϕ is called* n, m-maximal exactly provable *iff, for all exactly provable $\psi \in \mathbf{L}^n_m$ such that $\psi \vdash \phi$, $\psi \equiv \phi$.*

It will turn out in the last section that the $1, 1$-maximal exactly provable formulas in $\mathbf{L}^1$ are maximal exactly provable. In general, however, not all the n, m-maximal exactly provable formulas in $\mathbf{L}^n_m$ are maximal exactly provable. To construct counter-examples the following insight derived from lemma 5.4.0.3 and the fact that, by lemma 5.3.0.6, $\mathbf{L}^n_m$ formulas are, up to $\equiv$, determined by their n, m-types was used.

5.4.0.12. FACT. *The m-maximal exactly provable $\mathbf{L}^n_m$-formulas are the ones with a set of types C that contains exactly one reflexive n, m-type t and for which C is minimal upwardly closed realizable, in the sense that, C is upwardly closed realizable, but this is not the case for any proper subset of C containing t.*

The simplest counter-example we found uses a set of $3, 2$-types with exactly one minimal enveloping type in the sense of the previous fact. Such a set of $3, 2$-types will correspond to a $3, 2$-maximal exactly provable formula. The following two models, both built using only this set of types, show there is a real choice in ordering it.



28. FIGURE. *Two models built from the set of* $3,2$*-types corresponding to a* $3,2$*-maximal exactly provable formula.*

The models above can be extended to tail models corresponding to different maximal exactly provable $\mathbf{L}_3^3$ formulas. From both of these formulas the $3,2$-maximal exactly provable formula corresponding to the set of $3,2$-types is derivable. Hence this $3,2$-maximal exactly provable formula is clearly not maximal exactly provable.

A further conjecture is that the set of n,m-types of an arbitrary exactly provable $\mathbf{L}_m^n$ formula ϕ is the union of the sets of types of the n,m-maximal exactly provable $\mathbf{L}_m^n$ formulas from which ϕ is derivable. That such a union always is the set of types of an exactly provable formula if a common enveloping type is present, follows immediately from the next lemma.

5.4.0.13. LEMMA. *If* C *is the union of sets* $C_1,\cdots,C_k$ *of* n,m *types corresponding to exactly provable* $\mathbf{L}_m^n$ *formulas* $\phi_1,\cdots,\phi_k$ *with an enveloping type* t *for all of* C*, then there exists a* $\phi\in\mathbf{L}_m^n$ *such that* $T_m^n(\phi)=C$*.*

Proof. It suffices to note that, if $K_1,\cdots,K_k$ are upwards closed realizations of $C_1,\cdots,C_k$, then $K_1\cup\cdots\cup K_k$ is an upwardly closed realization of C, and then to apply lemma 5.4.0.3. ⊣

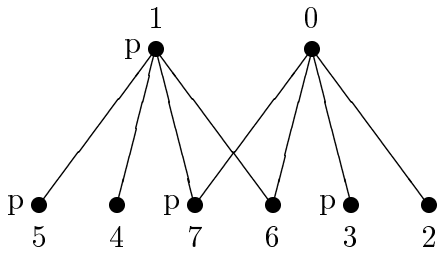
It is certainly not true that any union of types of n,m-maximal exactly provable formulas is the set of n,m-types of some exactly provable $\mathbf{L}_m^n$ formula. A counter-example is provided by the sets of types belonging to p and to $\neg p$, both $0,1$-maximal exactly provable formulas, which cannot be combined to an exactly provable formula, even for $m=1$. A common enveloping type is needed, and is obviously not available for p and $\neg p$ (see section 5.5).

5.5 Calculating exactly provable formulas

In this section the calculation of the exactly provable formulas in $\mathbf{L}_1^1$ will be discussed. It will be shown that already in this very first small fragment there are 62 non-interderivable members with 8 maximal elements. Of the next fragment $\mathbf{L}_2^1$ even the

cardinality of the set of maximal exactly provable elements has eluded us so far. The fragment $\mathbf{L}_3^1$ is definitely too large to attack in this manner.

To calculate the exactly provable formulas in $\mathbf{L}_1^1$ we use sets of $1,1$-types. The $1,1$-types can be ordered into an exact Kripke model $Exm(\mathbf{L}_1^1)$:



29. FIGURE. *An exact Kripke model of* $\mathbf{L}_1^1$.

This exact model corresponds to the first two layers (ExL_0^1 and ExL_1^1) in the construction of ExL^1. In ordering the types into an exact model other choices could have been made, resulting in different models. In fact, in the calculation of exactly provable formulas the choice of the exact model is arbitrary. In the sequel we will denote the $1,1$-types by their number in the exact model above.
In the previous section we proved that $\phi \in L_1^1$ is exactly provable iff

1. $T_1^1(\phi)$ is upwards closed realizable;
2. there is a $t \in T_1^1(\phi)$ such that $\forall t' \in T_1^1(\phi)(t' \restriction 0 \in j_1(t))$.

These criteria are easily translated into a test on a set of $1,1$-types C. The first condition of this test requires C to be upwards closed realizable (in ExL^1 not necessarily in the model above) and the second condition demands an enveloping type in C. Let $\phi \in L_1^1$ and $C = T_1^1(\phi)$. Then ϕ is exactly provable iff

1. if $2 \in C$ or $3 \in C$, then $0 \in C$
   if $4 \in \ \in C$ or $5 \in C$, then $1 \in C$
   if $6 \in C$ or $7 \in C$, then $C \cap \{0,2,4\} \neq \emptyset$ and $C \cap \{1,3,5\} \neq \emptyset$;
2. $6 \in C$ or $7 \in C$ or $C = \{0,2\}$ or $C = \{1,5\}$.

The sets of $1,1$-types corresponding to exactly provable formulas in $\mathbf{L}^1$ can be found in applying the above test to the 255 non-empty subsets of T_1^1. We prefer however to calculate the exactly provable formulas together with their corresponding sets of $1,1$-types. To do so, the exact model above will be used to calculate all $\mathbf{L}_1^1$ formulas in the following manner.

Our computer program generates a list of formulas and sets. It starts with the formulas $\bot$ and p and the sets $[\![\bot]\!] = \emptyset$ and $[\![p]\!] = \{1,3,5,7\}$ (where $[\![\phi]\!] = \{k \in Exm(\mathbf{L}_1^1) \mid k \Vdash \phi\}$).

The list of formulas ϕ and sets $[\![\phi]\!]$ is extended by systematically applying the connectives $(\neg, \wedge, \vee, \rightarrow, \Box)$ and the corresponding set operations, adding a pair consisting of a formula and its set only if the set does not yet occur in the list. In this way we ensure that no two distinct interderivable formulas will occur in the list. Note that this computation of the Lindenbaum algebra of an exact Kripke model is similar to the calculation described in Chapter 2.

In generating the list of formulas and sets the test defined above is applied to distinguish the exactly provable formulas in $\mathbf{L}_1^1$.

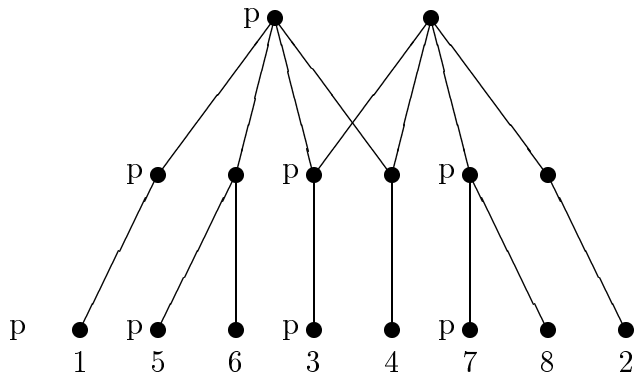
The exactly provable formulas in $\mathbf{L}_1^1$ have been listed in appendix B.4.

To find the $1,1$-maximal exactly provable formulas ϕ in the list, one has to look for the minimal sets $T_1^1(\phi)$ (i.e. those that do not occur as a proper subset of some $T_1^1(\psi)$ in the list).

The sets of types of this kind are:

1. $\{1,5\}$	3. $\{0,1,7\}$	5. $\{1,4,7\}$	7. $\{0,3,7\}$
2. $\{0,2\}$	4. $\{0,1,6\}$	6. $\{1,4,6\}$	8. $\{0,3,6\}$

It turns out that each of these $1,1$-maximal exactly provable formulas is maximal exactly provable. The corresponding tail models can be found, using the model below, by extending the submodels $\underline{\uparrow}k$ downward with a tail of copies of k for each of the numbered elements.



30. FIGURE. *Extending ExL_1^1 to find tail models.*

We will give these maximal exactly provable formulas in $\mathbf{L}_1^1$ a more informative form:

1. p
2. $\neg p$
3. $(\Box p \to \Box\bot) \wedge (\Box\neg p \to \Box\bot) \wedge (\neg p \to \Box\neg p)$
4. $(\Box p \to \Box\bot) \wedge (\Box\neg p \to \Box\bot) \wedge (p \to \Box p)$
5. $p \leftrightarrow \Diamond\neg p \vee \Box\bot$
6. $p \leftrightarrow \Box\neg p$
7. $p \leftrightarrow \neg\Box p$
8. $p \leftrightarrow \Box\neg p \vee \neg\Box\bot$

Formulas 1 and 2 correspond to *provable* and *refutable* sentences in **PA**. Formulas 6 and 7 can be (faithfully) interpreted by *Gödel-sentences* and their duals in **PA**. Similarly, formulas 3 and 4 correspond to *Rosser-sentences* and their duals in **PA**. The only small surprise is formed by formula 8 and its dual 5. It is easy to see that 8 is interderivable with $p \leftrightarrow \Box\Box\bot \wedge \neg\Box\bot$ and thus, of course, 5 with $p \leftrightarrow (\Box\Box\bot \to \Box\bot)$.

These two formulas are not $\mathbf{L}_1^1$, but can apparently interderivably be given as such. Note also that, by the fixed point theorem of $\mathbf{L}$ (see e.g., [Smoryński 85], [Boolos 93]), there is no surprise in the fact that in the equivalences of 5 and 8 the p in the right hand side can be eliminated in favor of the $\bot$, but only in the fact that by using p instead of $\bot$ one can push down the complexity.

Chapter 6

A family of propositional testers

6.1 Introduction

The common origin of the theorem testers treated here is the semantic tableau method introduced by Beth in 1955 [Beth 55]. Beth defined semantic tableaux both for classical and intuitionistic (predicate) logic. Restricting these methods to propositional formulas yields decision procedures for the classical propositional logic (**CpL**) and the intuitionistic propositional logic (**IpL**). By appropriately changing the rules, the semantic tableau method can also be used in modal logic.

Algorithms to decide for a given logic L and a given formula A whether $\vdash_L A$ are called *formula testers* here, whereas the (usual) term *theorem prover* is reserved for algorithms that produce a proof (for example in natural deduction style) if the given formula is a theorem. In [Hendriks 80] for example, a theorem prover is given, based on the tableau method for **CpL**.

6.2 Preliminaries

A *tableau* is an ordered set of sequents $L \bullet R$, where L and R are structures of sequences of formulas. A *tableau method* defines what shall be considered as a sequent and gives a set of rules to derive new sequents from a given sequent (thus defining the order of the tableau). In those cases where application of a rule results in more than one sequent, the tableau is said to *branch* into subtableaux.

A sequent $L \bullet R$ is *closed* if $L \cap R \neq \emptyset$. Here we used the intersection of L and R as if they were sets. In the sequel we will treat L and R as sets if in the context there is no risk of confusion. Let us write $\#X$ for the number of elements in X and $Sub(X)$ for the set of subformulas of formulas in X.

The rules of a tableau method resemble a system of derivation rules for sequents in a system of sequent calculus. Treatment of a sequent results in a finite directed acyclic graph. If all terminal sequents are closed, the resulting tableau is a proof, but upside down, as the tableau method started with the conclusion of the proof and

the closed sequents correspond to axioms.

All formula testers presented here are based on a tableau method. In the rest of this chapter we will use the following convention of writing:

p	an atomic formula
A, B, C	formulas
K, M, N, S, T, U	sequences of formulas, not containing duplicates
L, R	ordered pairs of sequences of formulas

To test whether or not the formula A is derivable, one starts with a sequent $\bullet A$ and applies the rules, until all sequents are either closed or no rule can be applied. If enough sequents close, the tableau is said to *close* and A is derivable. Otherwise the tableau is said to stay *open* and A is not derivable. In the description of the tableaux algorithms we will use rewriting rules on socalled *split sequents* $L \bullet R$, where L and R are finite sequences of finite sequences of formulas, separated by additional symbols (like ; and ,). If $L \bullet R$ is a split sequent, $A, L \bullet R$ is the split sequent where A is added on the left hand side of the leftmost sequence in L. Also we will write $L \bullet R, A$ for adding A to the right of R. However, we will assume that before adding a formula A to a sequence X in a split sequent, a check is performed whether A is already an element of X. So, if $A \in X$ then A, X and X, A are equal to X.

6.3 CpLtest: a CpL tester

The simplest member of our family is *CpLtest*, a formula tester for **CpL**.

The split sequents of *CpLtest* are of the form $M; N \bullet S; T$. To test whether A is derivable, one starts with the split sequent $;\bullet A;$. Hence A is a formula in S, the sequence of righthand side formulas to be treated. The *CpLtest* rules below are applied to a split sequent by treating the leftmost formula of S or the rightmost formula of N. If $N = S = \emptyset$ treatment stops. In treating a formula A subformulas of A are placed in S or in N. A formula in S (N) is placed in sequence T (M) to facilitate recognition of a closure, i.e. a formula A occurring both in L and in R.

The rules of the tester *CpLtest* are:

$$(pR)\ \frac{L \bullet p, R}{L \bullet R, p} \qquad (pL)\ \frac{L, p \bullet ; T}{p, L \bullet ; T}$$

$$(\neg R)\ \frac{L \bullet \neg A, R}{L, A \bullet R, \neg A} \qquad (\neg L)\ \frac{L, \neg A \bullet ; T}{\neg A, L \bullet A; T}$$

$$(\wedge R)\ \frac{L \bullet A \wedge B, R}{L \bullet A, R, A \wedge B \quad L \bullet B, R, A \wedge B} \qquad (\wedge L)\ \frac{L, A \wedge B \bullet ; T}{A \wedge B, L, A, B \bullet ; T}$$

$$(\vee R)\ \frac{L \bullet A \vee B, R}{L \bullet A, B, R, A \vee B} \qquad (\vee L)\ \frac{L, A \vee B \bullet ; T}{A \vee B, L, A \bullet ; T \quad A \vee B, L, B \bullet ; T}$$

$$(\rightarrow R)\ \frac{L \bullet A \rightarrow B, R}{L, A \bullet B, R, A \rightarrow B} \qquad (\rightarrow L)\ \frac{L, A \rightarrow B \bullet ; T}{A \rightarrow B, L \bullet A; T \quad A \rightarrow B, L, B \bullet ; T}$$

None of the *CpLtest* rules is applicable to a closed split sequent.

Note that all L-rules require that $S = \emptyset$. So for each split sequent at most one rule is applicable and hence the algorithm *CpLtest* is deterministic. We define measures of complexity that will strictly decrease with each application of a *CpLtest* rule.

6.3.0.1. Definition. *Let X be a set of split sequents.*

1. $\gamma(p) = 0$;
2. $\gamma(\neg A) = \gamma(A) + 1$;
3. $\gamma(A \circ B) = \gamma(A) + \gamma(B) + 2 \qquad$ if $\circ \in \{\wedge, \vee, \rightarrow\}$;
4. $\mu(M; N \bullet S; T) = \Sigma\{\gamma(A) + 1 \mid A \in N\} + \Sigma\{\gamma(A) + 1 \mid A \in S\}$;
5. $\eta(M; N \bullet S; T) = \#Sub(N) + \#Sub(S)$;
6. $\sigma(X) = \Sigma\{2^{\eta(L \bullet R)} \times \mu(L \bullet R) \mid L \bullet R \in X\}$.

6.3.0.2. Lemma. *If $L \bullet R$ is a split sequent then $\mu(L \bullet R) \geq 0$. If $L \bullet R$ is a split sequent derived from split sequent $L' \bullet R'$ by application of one of the* CpLtest *rules, then*

$$\mu(L \bullet R) < \mu(L' \bullet R')$$

If X is a set of split sequents then $\sigma(X) \geq 0$. If X' is a set of split sequents derived from X by application of one of the CpLtest *rules (replacing the split sequent treated by the result(s) of the application of the* CpLtest *rule), then*

$$\sigma(X') < \sigma(X)$$

Proof. By checking the rules. ⊣

The measure $\sigma(X)$ provides us with an upper bound to the number of steps it may take *CpLtest* to treat all split sequents in X (and the resulting sequents and so on) until no rule of *CpLtest* is applicable (hence $N = S = \emptyset$).

6.3.0.3. Definition. *A split sequent $L \bullet R$ is* open *if it is not closed and no* CpLtest *rule is applicable. A split sequent $L \bullet R$ is* closing *if it is closed or if a* CpLtest *rule is applicable and the resulting split sequent(s) are closing. We will write $L \stackrel{\bullet}{-} R$ if $L \bullet R$ is closing.*

Note that the definition of *closing* is sound because the algorithm is terminating.

6.3.0.4. Lemma. *$L \stackrel{\bullet}{-} R$ is closing iff $L \vdash \bigvee R$.*

Proof. If $L \bullet R$ is closed then of course $L \vdash \bigvee R$. If $L \bullet R$ is open, define a model M by taking $M \models p \quad \Leftrightarrow \quad p \in L$ for atomic formulas p. Using the fact that all formulas in N and S in the split sequent are treated by one of the rules, one proves $M \models \bigwedge L$ and $M \not\models \bigvee R$.

As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

By checking the *CpLtest* rules, observe that they correspond to equivalent statements about the derivability relation of **CpL** as stated in lemma 6.3.0.4. For example for the $\vee L$-rule one can prove

$$A \vee B \vdash C \quad \Leftrightarrow \quad A \vee B, A \vdash C \quad \text{and} \quad A \vee B, B \vdash C$$

in **CpL**. ⊣

We now present *CPLtest* as a pseudo-code program, called *Ctest*. In the pseudo-code language the notation of the sequence operation A, X introduced earlier, will be replaced by $\langle A, X\rangle$, writing A for $\langle A, \emptyset\rangle$ and $\langle A, B, X\rangle$ for $\langle A, \langle B, X\rangle\rangle$. $Ctest(,,\phi,)$, the program *Ctest*, with as its input the formula ϕ, will return the value **true** if $\vdash_{CpL} \phi$ and the value **false** otherwise.

$Ctest(M, N, S, T :$ **sequence of** *formula*$)$: **bool**
  if $S \neq \emptyset$
  then let $S = \langle A, S'\rangle$
    if $A \in M \cup N$ **then true**
    else in case A
      atomic : $Ctest(M, N, S', \langle A, T\rangle)$
      $\neg B$: $Ctest(M, \langle B, N\rangle, S', \langle A, T\rangle)$
      $B \wedge C$: **if** $Ctest(M, N, \langle B, S'\rangle, \langle A, T\rangle)$
        then $Ctest(M, N, \langle C, S'\rangle, \langle A, T\rangle)$
        else false
      $B \vee C$: $Ctest(M, N, \langle B, C, S'\rangle, \langle A, T\rangle)$
      $B \to C$: $Ctest(M, \langle B, N\rangle, \langle C, S'\rangle, \langle A, T\rangle)$
  else if $N \neq \emptyset$
    then let $N = \langle A, N'\rangle$
      if $A \in T$ **then true**
      else in case A
        atomic : $Ctest(\langle A, M\rangle, N', , T)$
        $\neg B$: $Ctest(\langle A, M\rangle, N', B, T)$
        $B \wedge C$: $Ctest(\langle A, M\rangle, \langle A, B, N'\rangle, , T)$
        $B \vee C$: **if** $Ctest(\langle A, M\rangle, \langle B, N\rangle, , T)$
          then $Ctest(\langle A, M\rangle, \langle C, N\rangle, , T)$
          else false
        $B \to C$: **if** $Ctest(\langle A, M\rangle, N, B, T)$
          then $Ctest(\langle A, M\rangle, \langle C, N\rangle, , T)$
          else false
    else false

To calculate an upper bound to the amount of time needed to calculate $Ctest(,,\phi,)$, we can make use of the measure σ defined above, as $\sigma(\{;\bullet\phi;\})$ is an upper bound to the number of calls to the *Ctest* procedure.

6.3.0.5. FACT. *Let $|\phi|$ be the length of formula ϕ, i.e. the number of atoms and connectives in ϕ. Then*

1. $\gamma(\phi) < |\phi|$;
2. $\mu(;\bullet\phi;) \leq |\phi|$;
3. $\eta(;\bullet\phi;) < |\phi|$;
4. $\sigma(;\bullet\phi;) < |\phi|.2^{|\phi|}$.

Next we need an upper bound to the time it takes to respond to a call of *Ctest*. In the worst case the procedure *Ctest* involves the following steps:

1. determine whether $S = \emptyset$ and $N = \emptyset$,
2. splitting a sequence X as $\langle A, X'\rangle$,
3. determine whether a formula is in $M \cup N$ or T,
4. decompose a formula A into its principal subformulas,
5. concatenating a formula A and a sequence X into $\langle X, A\rangle$, which should result in the sequence X if A is already a member of X.

We assume that placing a (new) call to *Ctest* will take a small constant amount of time. Let us assume that X is the largest sequence and D is the longest formula in the *Ctest* call we are dealing with.

The first step will only take a small constant amount of time, as will the second step if sequences are represented as linked lists for example.

To determine equality of two formulas A and B will cost, at the most, $\min\{|A|, |B|\}$ steps. Hence, as an upper bound for the third step we can use $\#X \times |D|$.

Step four can be done in a number of steps linear in the length of the formula treated. Step five may occur thrice and each time we may use $\#X \times |D|$ as an upper bound.

From the rules of *Ctest* it is clear that the original ϕ from the input is the longest formula appearing in any of the consecutive calls to *Ctest*. Hence in the formulas above we can replace D by ϕ. Also, from the rules of *Ctest* we can find as an upper bound for the largest sequence of subformulas in the input formula ϕ. Hence $\#X \leq |\phi|$. As a result we have found an upper bound

$$4.|\phi|^2 + c_1.|\phi| + c_2$$

for the time needed to answer one call to *Ctest* as part of the calculation of $Ctest(,,\phi,)$, c_1 and c_2 being (implementation dependent) constants.

By combining the two upper bounds calculated above, we found the upper bound to the amount of time needed in the calculation of $Ctest(,,\phi,)$ as a whole to be of the order $4.|\phi|^3.2^{|\phi|}$.

As for the upper bound to the space needed in calculating $Ctest(,,\phi,)$, note that in the worst case a call to *Ctest* is replaced by two other calls plus a command to process the results. As the number of calls is $\sigma(\{;;\bullet\phi;\})$ and a call will take $2\#X \times |D|$ on the stack at the most (as each occurence of a subformula of ϕ will occur at most twice in the split sequent) an upper bound for the stack is $2.|\phi|^3.2^{|\phi|}$. We assume that to calculate a call to *Ctest* one needs to keep the initial sequences in the memory. As we also have to produce (at most) three new sequences and need some space for (at most) three formulas, a fair upper bound for the space needed in one call is $4.\#X \times |D| + 3.|D|$ or $4.|\phi|^2 + 3.|\phi|$. Hence the order of space needed to calculate $Ctest(,,\phi,)$ is $2.|\phi|^3.2^{|\phi|}$.

6.4 IpLtest: an IpL tester

The split sequents of *IpLtest* are of the form $K;M;N \circ S;T;U$ or $K;M;N \odot S;T;U$. As in the previous section, we will use the abbreviations L and R in describing the rules of *IpLtest*. Here $L = K;M;N$ and $R = S;T;U$. The notation $L \bullet R$ will be used to denote either $L \circ R$ or $L \odot R$.

Testing the derivability of formula A starts with the split sequent $;; \circ A;;$. Formulas to be treated are placed in N or S, those already treated are placed in K or U (and kept to facilitate the recognition of closure of a sequent). In *IpLtest* we have to take special care of implications and negations. Treatment on the righthand side (i.e. if implications or negations appear in S) is postponed; the implications and negations are placed in T. Formulas in T will only be treated if everything else fails.

On the lefthand side implications and negations may have to be treated more than once. After being treated, implications and negations are not moved from N to K, but to M. If $N = S = \emptyset$, then *IpLtest* may try all formulas in M again (by the RL-rule). To avoid *IpLtest* to go on with repeating the formulas in M indefinitely, there is a mechanism to keep track of the changes in the set of atoms on the lefthand side of the split sequent. We will write $L \circ R$ if there have not been introduced new atoms on the lefthand side since the last treatment of a formula in T. After the introduction of a 'new' atomic formula on the lefthand side, the split sequent is written as $L \odot R$. $L \odot R$ becomes $L' \circ R'$ via the RL-rule.

As in case of *CpLtest* a split sequent $L \bullet R$ is *closed* if $L \cap R \neq \emptyset$. As before we will assume that no *IpLtest* rules are applicable to a closed split sequent. Let p be an atomic formula. The rules of *IpLtest* are:

$$(pR)\ \frac{L \bullet p, R}{L \bullet R, p} \qquad (pL1)\ \frac{L, p \odot ;T;U}{p, L \odot ;T;U}$$

$$((pL2)\ \frac{L, p \circ ;T;U}{L \circ ;T;U} \quad p \in L \qquad (pL3)\ \frac{L, p \circ ;T;U}{p, L \odot ;T;U} \quad p \notin L$$

$$(\neg R)\ \frac{L \bullet \neg A, S;T;U}{L \bullet S;T, \neg A;U} \qquad (\neg L)\ \frac{K;M;N, \neg A \bullet ;T;U}{K;\neg A, M;N \bullet A;T;U}$$

$$(\wedge R)\ \frac{L \bullet A \wedge B, R}{L \bullet A, R, A \wedge B \quad L \bullet B, R, A \wedge B} \qquad (\wedge L)\ \frac{L, A \wedge B \bullet ; T; U}{A \wedge B, L, A, B \bullet ; T; U}$$

$$(\vee R)\ \frac{L \bullet A \vee B, R}{L \bullet A, B, R, A \vee B} \quad (\vee L)\ \frac{L, A \vee B \bullet ; T; U}{A \vee B, L, A \bullet ; T; U \quad A \vee B, L, B \bullet ; T; U}$$

$$(\rightarrow R)\ \frac{L \bullet A{\rightarrow}B, S; T; U}{L \bullet S; T, A{\rightarrow}B; U}$$

$$(\rightarrow L)\ \frac{K; M; N, A{\rightarrow}B \bullet ; T; U}{K; A{\rightarrow}B, M; N \bullet A; T; U \quad K; A{\rightarrow}B, M; N, B \bullet ; T; U}$$

$$\mathbf{RL}\ \frac{K; M; \odot ; T; U}{; K; M \circ ; T; U}$$

$$\neg\mathbf{RR}\ \frac{K; M; \circ ; \neg A, T; U}{K; M; A \circ ; ; \quad K; M; \circ ; T; U}$$

$$\rightarrow\mathbf{RR}\ \frac{K; M; \circ ; A{\rightarrow}B, T; U}{K; M; A \circ B; ; \quad K; M; \circ ; T; U}$$

IpLtest, like *CpLtest*, has an L- and an R-rule for each of the connectives. $\neg R$ and $\rightarrow R$ postpone the treatment of negations and implications until all other rules but $\neg$RR or $\rightarrow$RR have failed. The RR-rules are special in that they may decrease the number of formulas in the split sequent. The RL-rule enforces treatment of all implications and negations on the lefthand side of the split sequent. The RL-rule causes the sequence M (the implications and negations to be repeated) to make up the new sequence N (of formulas to be treated).

Note that for each split sequent at most one of the *IpLtest* rules is applicable. Hence the algorithm *IpLtest* is deterministic. To prove *IpLtest* to terminate on every split sequent we again define a measure of complexity on split sequents, as we did for *CpLtest*. This time however the definition is more complex.

6.4.0.1. Definition. *Let p be an atomic formula A an* **IpL** *formula, $L \bullet R$ a split sequent (such that $L = K; M; N$ and $R = S; T; U$) and X a set of split sequents.*

1. $\gamma(p) = 0$;
2. $\gamma(\neg A) = \gamma(A) + 2$;
3. $\gamma(A * B) = \gamma(A) + \gamma(B) + 3$ if $* \in \{\wedge, \vee\}$;
4. $\gamma(A{\rightarrow}B) = \gamma(A) + \gamma(B) + 4$;
5. $\delta(L \circ R) = \Sigma\{\gamma(A) + 2 \mid A \in N\} + \Sigma\{\gamma(A) + 2 \mid A \in S\} +$
   $\Sigma\{\gamma(A) + 1 \mid A \in T\}$;
6. $\delta(L \odot R) = \delta(L \circ R) + 1$;
7. $\lambda(L \bullet R) = \Sigma\{\gamma(A) + 2 \mid (A = B{\rightarrow}C$ or $A = \neg B)$ and $A \in Sub(L \cup R)\}$;
8. $\eta(L \bullet R) = \#Sub(M \cup N) + \#Sub(S) + \#Sub(T)$;

9. $n(L \bullet R) = \#\{p \text{ atomic} \mid p \in Sub(L \cup R)\}$;
10. $m(L \bullet R) = \#\{p \text{ atomic} \mid p \in K\}$;
11. $\mu(L \bullet R) = (n(L \bullet R) - m(L \bullet R)).\lambda(L \bullet R) + \delta(L \bullet R)$;
12. $\sigma(X) = \Sigma\{2^{\eta(L \bullet R)} \times \mu(L \bullet R) \mid L \bullet R \in X\}$.

6.4.0.2. Lemma. *If $L \bullet R$ a split sequent then $\mu(L \bullet R) \geq 0$. If $L \bullet R$ is derived from split sequent $L' \bullet R'$ by application of one of the* IpLtest *rules, then*

$$\mu(L \bullet R) < \mu(L' \bullet R')$$

If X a set of split sequents then $\sigma(X) \geq 0$. If X' a set of split sequents derived from X by application of one of the IpLtest *rules (replacing the split sequent treated by the result(s) of the application of the* IpLtest *rule) then*

$$\sigma(X') < \sigma(X)$$

Proof. By checking the rules. In most cases application of a rule wille decrease the δ of the split sequent. Only the RL-rule increases the δ. However, for a given split sequent the $pL3$-rule can only be applied $n - m$-times (as n is the total number of atoms in the split sequent and m the number of atoms in K). Hence, also the RL-rule can only be applied $n - m$ times. The number λ, as defined above, is an upper bound on the increase of δ by an application of the RL-rule. ⊣

6.4.0.3. Definition. *A split sequent $L \bullet R$ is closing ($L \stackrel{\bullet}{-} R$) if*

1. *$L \bullet R$ is closed (i.e. $L \cap R \neq \emptyset$);*
2. *one of the RR-rules is applicable and one of the resulting split sequents is closing;*
3. *one of the other rules is applicable and its resulting split sequent(s) is (are) closing.*

To prove *IpLtest* to be sound and complete we will prove

$$L \stackrel{\bullet}{-} R \quad \Leftrightarrow \quad L \vdash \bigvee R$$

In order to do so, we need the following definition and some facts.

6.4.0.4. Definition. *A split sequent $L \bullet R$ is* reduced *if it is not closed and no other rules but the RR-rules are applicable. If $L \bullet R$ a split sequent that is not closing, application of the* IpLtest *rules, with the exception of the RR-rules, will result in one or more reduced split sequents that will be called* reductions *of $L \bullet R$.*

Note that a split sequent is reduced if not closed and $N = S = \emptyset$.

6.4.0.5. Fact. *If $L \bullet R$ is a reduced split sequent then:*

1. $A \wedge B \in L \quad \Rightarrow \quad A \in L$ *and* $B \in L$;
2. $A \vee B \in L \quad \Rightarrow \quad A \in L$ *or* $B \in L$;
3. $\neg A \in L \quad \Rightarrow \quad A \notin L$;
4. $A \rightarrow B \in L \quad \Rightarrow \quad A \notin L$ *or* $B \in L$;
5. $A \wedge B \in R \quad \Rightarrow \quad A \in R$ *or* $B \in R$;
6. $A \vee B \in R \quad \Rightarrow \quad A \in R$ *and* $B \in R$.

The truth of this fact can be established by observation of the *IpLtest* rules. No rule changes the monotone increase of the set of formulas L and only with the RR-rules do formulas disappear from R. Note that a reduced split sequent is always of the form $K; M; \circ; T; U$.

6.4.0.6. LEMMA. *$L \stackrel{\bullet}{-} R$ implies $L \vdash \bigvee R$ (in* **IpL***).*

Proof. For a closed split sequent the lemma is obvious. As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

According to the *IpLtest* rules $L \stackrel{\bullet}{-} A \wedge B, R$ iff both $L \stackrel{\bullet}{-} A, R, A \wedge B$ and $L \stackrel{\bullet}{-} B, R, A \wedge B$. By induction hypothesis we may infer $L \vdash A \vee (A \wedge B) \vee \bigvee R$ and $L \vdash B \vee (A \wedge B) \vee \bigvee R$. Hence in **IpL** one can derive $L \vdash (A \wedge B) \vee \bigvee R$.

All *IpLtest* rules can be treated in the same way. For the RR-rules observe that only one of the consequents of the rules has to be closing.

For the $RR{\rightarrow}$-rule for example: if $L, A \stackrel{\bullet}{-} B$ then by induction hypothesis $L, A \vdash B$ and hence $L \vdash A{\rightarrow}B$. Otherwise if $L \stackrel{\bullet}{-} R$ and hence $L \vdash \bigvee R$, then of course also $L \vdash A{\rightarrow}B \vee \bigvee R$. For the $RR\neg$-rule, in case $R = \emptyset$, note that $\bigvee \emptyset = \bot$. ⊣

To prove $L \bullet R$ is not closing implies $L \nvdash \bigvee R$, we will extract from the non-closing tableau a Kripke model K forcing all formulas in L and none of those in R. In the definition of the Kripke model we will make use of the concept of the leftmost non-closing reduction of a split sequent. In finding this reduction one chooses to follow the leftmost non-closing conclusion of each *IpLtest* rule.

6.4.0.7. DEFINITION. *Let $L \bullet R$ be a non-closing split sequent. The Kripke model K* associated with *$L \bullet R$ is defined as the ordered set of (leftmost) non-closing reduced split sequents:*

1. *the leftmost non-closing reduction of $L \bullet R$ is the root of K;*
2. *if $k_l \in K$ corresponds to the split sequent $L'\circ; A{\rightarrow}B, T; U$ and $T \neq \emptyset$, then the leftmost non-closing reductions of $L', A \circ B; ;$ and $L'\circ; T; U$ are nodes of K, say respectively k_m and k_n, such that $k_l \leq k_m$ and $k_l \leq k_n$;*
3. *if $k_l \in K$ corresponds to the split sequent $L'\circ; A{\rightarrow}B, T; U$ and $T = \emptyset$, then the leftmost non-closing reduction of $L', A \circ B; ;$ is a node of K, say k_m, such that $k_l \leq k_m$;*
4. *if $k_l \in K$ corresponds to the split sequent $L'\circ; \neg A, T; U$ and $T \neq \emptyset$, then the leftmost non-closing reductions of $L', A\circ; ;$ and $L'\circ; T; U$ are nodes of K, say respectively k_m and k_n, such that $k_l \leq k_m$ and $k_l \leq k_n$;*
5. *if $k_l \in K$ corresponds to the split sequent $L'\circ; \neg A, T; U$ and $T = \emptyset$ then the leftmost non-closing reduction of $L', A\circ; ;$ is a node of K, say k_m, such that $k_l \leq k_m$;*
6. *the order relation $\leq$ is reflexive and transitive;*
7. *if $k_l \in K$ is the node corresponding to $L' \circ R'$, then $k_l \Vdash p$ for atomic formulas p iff $p \in L$.*

6.4.0.8. LEMMA. *If $L \odot R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L' \circ R'$ is the leftmost non-closing reduction of $L \bullet R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *IpLtest* rules (all except the RR-rules are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L' \circ R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction or a disjunction are obvious (using fact 6.4.0.5).

Let $A = B \to C$ and $A \in L'$. Let $k_l \leq k_m$ and $k_m \in K$ correspond to a reduced split sequent $L'' \circ R''$ and $k_m \Vdash B$. As formula A has been treated in the derivation of $L'' \circ R''$, there is a $k_n \in K$, $k_m \leq k_n$ and $k_n \nVdash B$ or $k_n \Vdash C$ such that k_m and k_n force the same atoms. This is due to the fact that the RL-rule would have been applied between the sequents of k_n and k_m if there was a difference in the atoms forced. By a simple lemma on Kripke models k_m and k_n force the same formulas and hence $k_m \Vdash C$, which proves $k_l \Vdash B \to C$.

Let $A = B \to C$ and $A \in R'$. Note that $A \in T$ and the split sequent $L', B \circ C;;$ (appearing after one or more applications of an RR-rule) will not be closing. Hence, if k_m corresponds to the leftmost non-closing reduction of $L', B \circ C;;$, by the induction hypothesis $k_m \nVdash A$. As we have $k_l \leq k_m$ we infer that $k_l \nVdash A$. ⊣

6.4.0.9. THEOREM. *A split sequent $L \odot R$ is closing, using the* IpLtest *rules, iff $L \vdash \bigvee R$.*

Proof. By combining the previous two lemmas. ⊣

The following pseudo-code program, *Itest* is an implementation of the *IpLtest* algorithm. For the language conventions see the *Ctest* program in the previous section. To test whether a formula ϕ is a theorem of **IpL**, one calls the program $Itest(,,,\phi,,,\mathbf{false})$, where the input corresponds to the initial sequent $;; \circ \phi;;$ for *IpLtest*.

```
Itest(K, M, N, S, T, U : sequence of formula, d : bool) : bool
  if S ≠ ∅
  then let S = ⟨A, S'⟩
       if A ∈ K ∪ M ∪ N then true
       else in case A
            atomic : Itest(K, M, N, S', T, ⟨A, U⟩, d)
            ¬B     : Itest(K, M, N, S', ⟨A, T⟩, U, d)
            B ∧ C  : if Itest(K, M, N, ⟨B, S'⟩, T, ⟨A, U⟩, d)
                     then Itest(K, M, N, ⟨C, S'⟩, T, ⟨A, U⟩, d)
                     else false
            B ∨ C  : Itest(K, M, N, ⟨B, C, S'⟩, T, ⟨A, U⟩, d)
            B→C    : Itest(K, M, N, S', ⟨A, T⟩, U, d)
  else if N ≠ ∅
       then let N = ⟨A, N'⟩
            if A ∈ T ∪ U then true
            else in case A
                 atomic : if A ∉ K
                          then Itest(⟨A, K⟩, M, N', , T, U, true)
                          else Itest(K, M, N', , T, U, d)
                 ¬B     : Itest(K, ⟨A, M⟩, N', B, T, U, d)
                 B ∧ C  : Itest(⟨A, K⟩, M, ⟨A, B, N'⟩, , T, U, d)
                 B ∨ C  : if Itest(⟨A, K⟩, M, ⟨B, N'⟩, , T, U, d)
                          then Itest(⟨A, K⟩, M⟨C, N'⟩, , T, U, d)
                          else false
                 B→C    : if Itest(K, ⟨A, M⟩, N', B, T, U, d)
                          then Itest(K, ⟨A, M⟩, ⟨C, N'⟩, , T, U, d)
                          else false
       else if d then Itest(K, , M, , T, U, false)
            else if T ≠ ∅
                 then let T = ⟨A, T'⟩
                      in case A
                      ¬B     : if Itest(K, M, B, , , , d) then true
                               else Itest(K, M, , , T', U, d)
                      B→C    : if Itest(K, M, B, C, , , d) then true
                               else Itest(K, M, , , T', U, d)
                 else false
```

6.5 Ktest: a tester for K

In this section and the following we will introduce tableaux testers for modal propositional logic.

The first tester to be described is *Ktest*, a tester for the modal logic **K**, that will act as the minimal system for the modal logics in this section. The axioms of

K are those of classical propositional logic **CpL** plus $\Box(A{\to}B){\to}(\Box A{\to}\Box B)$ and necessitation ($\vdash A \quad \Rightarrow \quad \vdash \Box A$) as an extra derivation rule.

Split sequents of *Ktest* are of the form $K;M;N \bullet S;T;U$. The rules for *Ktest* are the rules of *CpLtest*, with rules added to deal with $\Box$ and $\Diamond$:

$$(pR)\ \frac{L \bullet p,R}{L \bullet R,p} \qquad (pL)\ \frac{L,p \bullet ;T;U}{p,L \bullet ;T;U}$$

$$(\neg R)\ \frac{L \bullet \neg A,R}{L,A \bullet R,\neg A} \qquad (\neg L)\ \frac{L,\neg A \bullet ;T;U}{\neg A,L \bullet A;T;U}$$

$$(\wedge R)\ \frac{L \bullet A\wedge B,R}{L \bullet A,R,A\wedge B \quad L \bullet B,R,A\wedge B} \quad (\wedge L)\ \frac{L,A\wedge B \bullet ;T;U}{A\wedge B,L,A,B \bullet ;T;U}$$

$$(\vee R)\ \frac{L \bullet A\vee B,R}{L \bullet A,B,R,A\vee B} \quad (\vee L)\ \frac{L,A\vee B \bullet ;T;U}{A\vee B,L,A \bullet ;T \quad A\vee B,L,B \bullet ;T;U}$$

$$(\to R)\ \frac{L \bullet A{\to}B,R}{L,A \bullet B,R,A{\to}B} \quad (\to L)\ \frac{L,A{\to}B \bullet ;T;U}{A{\to}B,L \bullet A;T \quad A{\to}B,L,B \bullet ;T;U}$$

$$(\Box R)\ \frac{L \bullet \Box A,S;T;U}{L \bullet S;T,\Box A;U} \qquad (\Box L)\ \frac{K;M;N,\Box A \bullet ;T;U}{K;\Box A,M;N \bullet ;T;U}$$

$$(\Diamond R)\ \frac{K;M;N \bullet \Diamond A,S;T;U}{K;\Box\neg A,M;N \bullet S;T;U,\Diamond A} \qquad (\Diamond L)\ \frac{L,\Diamond A \bullet ;T;U}{\Diamond A,L \bullet ;T,\Box\neg A;U}$$

The NW-rule

$$\frac{K;M;\ \bullet ;\Box A,T;U}{;;M^* \bullet A;;\quad K;M;\ \bullet ;T;U} \text{ where } M^* = \{B \mid \Box B \in M\}$$

The NW-rule (the *new world rule*) plays the same role as the RR-rules in *IpLtest*. If the algorithm is regarded as a method of systematically constructing a Kripke model (that is a counterexample to the formula tested and failure of which proves that it is a theorem is true) this rule forces the introduction of a new world in the model construction.

The rules in *Ktest* for the possibility operator differ from the other rules (in *Ktest*, *IpLtest* or *CpLtest*), as in treating the formula $\Diamond A$, we not only use the subformula A, but in the result of the $\Diamond$-rules there appears a formula $\Box\neg A$. This is best understood as treating $\Diamond$ as an abbreviation of $\neg\Box\neg$. In this way we somewhat restricted the

number of rules. From the rules above it can be proved that we get an equivalent system by adding a new $\Diamond$NW-rule:

$$\frac{K; M, \Diamond A; \bullet ; ; U}{; ; M^*, A \bullet ; ; \quad K; M; \bullet ; ; U} \text{ where } M^* = \{B \mid \Box B \in M\}$$

and replacing the $\Diamond$ rules above by:

$$(\Diamond R)\ \frac{L \bullet \Diamond A, S; T; U}{L \bullet S; T, \Diamond A; U} \qquad (\Diamond L)\ \frac{K; M; N, \Diamond A \bullet ; T; U}{K; \Diamond A, M; N \bullet ; T; U}$$

Define a *Ktest* split sequent $L \bullet R$ to be *closed* if $L \cap R \neq \emptyset$. None of the *Ktest* rules is applicable to a closed split sequent.

The rules of *Ktest* are named according to the kind of formula treated and its position. Hence we have a pL- and a pR-rule, an $\rightarrow L$- and an $\rightarrow R$-rule and so on.

Note that at most one rule is applicable to any split sequent and hence the algorithm *Ktest* is deterministic. To prove the algorithm *Ktest* to terminate on each split sequent, we define a measure of complexity on a set X of split sequents, $\sigma(X)$ that will strictly decrease with each application of a *Ktest* rule on a member of X. Application of a *Ktest* rule to X has as its result a new set of split sequents X', where the split sequent treated in X is replaced by the result from the application of the *Ktest* rule.

6.5.0.1. Definition. *Let p be an atomic formula, A a* **K** *formula $L \bullet R$ a split sequent (such that $L = M; N$ and $R = S; T$) and X a set of split sequents.*

1. $\gamma(p) = 0$;
2. $\gamma(\neg A) = \gamma(A) + 1$;
3. $\gamma(A \circ B) = \gamma(A) + \gamma(B) + 2 \qquad$ if $\circ \in \{\wedge, \vee, \rightarrow\}$;
4. $\gamma(\Box A) = \gamma(A) + 1$;
5. $\gamma(\Diamond A) = \gamma(A) + 3$;
6. $\mu(L \bullet R) = \Sigma\{\gamma(A) + 1 \mid A \in N\} + \Sigma\{\gamma(A) \mid A \in M\} +$
   $\Sigma\{\gamma(A) + 1 \mid A \in S\} + \Sigma\{\gamma(A) \mid A \in T\}$;
7. $\eta(L \bullet R) = \#Sub(M) + \#Sub(N) + \#Sub(S) + \#Sub(T)$;
8. $\sigma(X) = \Sigma\{2^{\eta(L \bullet R)} \times \mu(L \bullet R) \mid L \bullet R \in X\}$.

6.5.0.2. Lemma. *If $L \bullet R$ a split sequent then $\mu(L \bullet R) \geq 0$. If $L \bullet R$ is a split sequent derived from split sequent $L' \bullet R'$ by application of one of the* Ktest *rules, then*

$$\mu(L \bullet R) < \mu(L' \bullet R')$$

If X a set of split sequents then $\sigma(X) \geq 0$. If X' a set of split sequents derived from X by application of one of the Ktest *rules (replacing the split sequent treated by the result(s) of the application of the* Ktest *rule) then*

$$\sigma(X') < \sigma(X)$$

Proof. By simply checking the rules. ⊣

6.5.0.3. Definition. *A split sequent $L \bullet R$ is closing ($L \stackrel{\bullet}{-} R$) if*

1. *$L \bullet R$ is closed;*
2. *the NW-rule is applicable and one of the resulting split sequents is closing;*
3. *the $\wedge R$-, $\vee L$- or $\rightarrow L$-rule is applicable and both the resulting split sequents are closing;*
4. *one of the other rules is applicable and its resulting split sequent is closing.*

To prove *Ktest* to be sound and complete we will prove

$$L \stackrel{\bullet}{-} R \quad \Leftrightarrow \quad L \vdash \bigvee R$$

But to do so we need the following definition and facts.

6.5.0.4. Definition. *A split sequent $L \bullet R$ is* reduced *if it is not closed and $N = S = \emptyset$ (no other rules but the NW-rule are applicable). If $L \bullet R$ is a split sequent that is not closing, application of the* Ktest *rules, with the exception of the NW-rule, will result in one or more reduced split sequents that will be called* reductions *of $L \bullet R$.*

A fortiori a split sequent is reduced if it is not closed and no *Ktest* rule applies to it.

6.5.0.5. Fact. *If $L \bullet R$ is a reduced split sequent then:*

1. $A \wedge B \in L \quad \Rightarrow \quad A \in L$ *and* $B \in L$;
2. $A \vee B \in L \quad \Rightarrow \quad A \in L$ *or* $B \in L$;
3. $\neg A \in L \quad \Rightarrow \quad A \in R$;
4. $A \rightarrow B \in L \quad \Rightarrow \quad A \in R$ *or* $B \in L$;
5. $A \wedge B \in R \quad \Rightarrow \quad A \in R$ *or* $B \in R$;
6. $A \vee B \in R \quad \Rightarrow \quad A \in R$ *and* $B \in R$;
7. $\neg A \in R \quad \Rightarrow \quad A \in L$;
8. $A \rightarrow B \in R \quad \Rightarrow \quad A \in L$ *and* $B \in R$.

The truth of this fact can be established by observation of the *Ktest* rules. Observe that only the NW-rule changes the monotonic increase of the sets of formulas L and R.

6.5.0.6. Lemma. *If a split sequent $L \bullet R$ is closing (by the* Ktest *rules) then $L \vdash \bigvee R$ (in* **K***).*

Proof. For a closed split sequent the lemma is obvious. As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

According to the *Ktest* rules $L \stackrel{\bullet}{-} A \wedge B, R$ iff both $L \stackrel{\bullet}{-} A, R, A \wedge B$ and $L \stackrel{\bullet}{-} B, R, A \wedge B$. By the induction hypothesis we may infer $L \vdash A \vee A \wedge B \vee \bigvee R$ and $L \vdash B \vee A \wedge B \vee \bigvee R$. Hence in **K** one can derive $L \vdash A \wedge B \vee \bigvee R$.

All *Ktest* rules can be treated in the same way. For the NW-rule observe that only one of the consequents of the rule has to be closing.

As M contains only boxed formulas, from $M^* \vdash A$ infer (by necessitation) that $M \vdash \Box A$. Hence we may conclude that $K, M \vdash \Box A \vee \bigvee T \vee \bigvee U$. If on the other hand it should be the case that $K, M \vdash \bigvee T \vee \bigvee U$ then obviously we would have $K, M \vdash \Box A \vee \bigvee T \vee \bigvee U$. $\dashv$

To prove that if $L \bullet R$ is not closing, then $L \nvdash \bigvee R$ we will extract from the non-closing tableau a Kripke model K forcing all formulas in L and none of those in R. In the definition of the Kripke model we will make use of the concept of the leftmost non-closing reduction of a split sequent as we did for *IpLtest*. In finding this reduction one chooses to follow the leftmost non-closing conclusion of each *Ktest* rule for which there is a choice.

6.5.0.7. Definition. *Let $L \bullet R$ be a non-closing split sequent. The Kripke model K associated with $L \bullet R$ is defined as a set of (leftmost) non-closing reduced split sequents, with an irreflexive relation $<$:*

1. *the leftmost non-closing reduction of $L \bullet R$ is the root of K;*
2. *if $k_l \in K$ corresponds to the split sequent $K; M; \bullet; T; U$ and $T = \{\Box A_1, \ldots \Box A_t\} \neq \emptyset$ then the leftmost non-closing reductions of $; ; M^* \bullet A_i; ;$ (where $M^* = \{B \mid \Box B \in M\}$ and $\Box A_i \in T$) are nodes of K, say respectively $l_1, \ldots, l_t$, such that for all i such that $1 \leq i \leq t$: $k_l < l_i$;*
3. *if $k_l \in K$ corresponds to the split sequent $K; M; \bullet; ; U$ (hence $T = \emptyset$) then k_l is a terminal node in K.*
4. *if $k_l \in K$ is the node corresponding to $L' \bullet R'$, then $k_l \Vdash p$ for atomic formulas p iff $p \in L$.*

6.5.0.8. Lemma. *If $L \bullet R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L' \bullet R'$ is the leftmost non-closing reduction of $L \bullet R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *Ktest* rules (all except the NW-rule are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L' \bullet R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction, a disjunction or an implication are obvious (using fact 6.5.0.5).

Let $A = \Box B$ and $A \in L'$. As $L' \bullet R'$ is reduced A will be a member of M. Let $k_l < k_m$ and $k_m \in K$ correspond to a reduced split sequent $L'' \bullet R''$. Then $L'' \bullet R''$ will be a result of the application of the NW-rule and hence we will have $B \in L''$. By induction hypothesis conclude that $k_m \Vdash B$. Which proves that $k_l \Vdash \Box B$. Note that if k_l is a terminal node then, as K has been defined in such a way that it is irreflexive, trivially $k_l \Vdash \Box B$.

If $A = \Box B$ and $A \in R'$ then A will be in T and application of the NW-rule (and reduction) will result in a node k_m corresponding to a reduced split sequent $L'' \bullet R''$

such that $k_l < k_m$ and $B \in R''$. Using the induction hypothesis we conclude $k_m \nVdash B$ and hence $k_l \nVdash \Box B$. ⊣

6.5.0.9. THEOREM. *A split sequent $L \bullet R$ is closing, using the* Ktest *rules, iff $L \vdash \bigvee R$.*

Proof. By combining the previous two lemmas. ⊣

Like we did previously for *CpLtest* and *IpLtest*, we will give a pseudo-code translation of the algorithm *KMtest*.

$KMtest(K, M, N, S, T, U : \textbf{sequence of } formula) : \textbf{bool}$
  if $S \neq \emptyset$
  then let $S = \langle A, S' \rangle$
    if $A \in J \cup K \cup M \cup N$ **then true**
    else in case A
      atomic : $KMtest(K, M, N, S', T, \langle A, U \rangle)$
      $\neg B$: $KMtest(K, M, \langle B, N \rangle, S', T, U)$
      $B \wedge C$: **if** $KMtest(K, M, N, \langle B, S' \rangle, T, \langle A, U \rangle)$
        then $KMtest(K, M, N, \langle C, S' \rangle, T, \langle A, U \rangle)$
        else false
      $B \vee C$: $KMtest(K, M, N, \langle B, C, S' \rangle, T, \langle A, U \rangle)$
      $B \rightarrow C$: $KMtest(K, M, \langle B, N \rangle, \langle C, S' \rangle, T, \langle A, U \rangle)$
      $\Box B$: $KMtest(K, M, N, S', \langle A, T \rangle, U)$
      $\Diamond B$: $KMtest(K, M, N, S', \langle \neg B, T \rangle, U)$
  else if $N \neq \emptyset$
    then let $N = \langle A, N' \rangle$
      if $A \in T \cup U$ **then true**
      else in case A
        atomic $KMtest(\langle A, K \rangle, M, N', , T, U)$
        $\neg B$: $KMtest(\langle A, K \rangle, M, N', B, T, U)$
        $B \wedge C$: $KMtest(\langle A, K \rangle, M, \langle A, B, N' \rangle, , T, U)$
        $B \vee C$: **if** $KMtest(\langle A, K \rangle, M, \langle B, N \rangle, , T, U)$
          then $KMtest(\langle A, K \rangle, M \langle C, N \rangle, , T, U)$
          else false
        $B \rightarrow C$: **if** $KMtest(\langle A, K \rangle, M, N, B, T, U)$
          then $KMtest(\langle A, K \rangle, M, \langle C, N \rangle, , T, U)$
          else false
        $\Box B$: $KMtest(K, \langle A, M \rangle, N, , T, U)$
        $\Diamond B$: $KMtest(K, M, N, , \langle T, A \rangle, U)$
    else if $T \neq \emptyset$
      then let $T = \langle \Box A, T' \rangle$ **and** $M^* = \{B \mid \Box B \in M\}$
        if $KMtest(, , M^*, A, ,)$ **then true**
        else $KMtest(K, M, , , T, U)$
      else false

6.6 Other testers for modal propositional logic

Testers for other modal propositional logics can be derived from *Ktest* by changing some of the rules (mainly the NW-rule). In this section we will indicate for several modal logics how a tester algorithm may be obtained.

6.6.1 Ttest: a T tester

The modal logic **T** has as its axioms and rules those of **K** plus the axiom $\Box\phi\rightarrow\phi$. **T** is complete for finite reflexive Kripke models (a proof can be found in [HC 84]).

To obtain *Ttest*, a tester for the modal logic **T**, we only have to change the $\Box$L-rule in *Ktest*.

6.6.1.1. Definition. *The tester* Ttest *has the same rules as* Ktest*, except for the* $\Box L$*-rule that is replaced by:*

$$(\mathbf{T}\Box L)\ \frac{K;M;N,\Box A\bullet;T;U}{K;\Box A,M;N,A\bullet;T;U}.$$

6.6.1.2. Theorem. *A split sequent* $L\bullet R$ *is closing, using the* Ttest *rules, iff* $L\vdash\bigvee R$.

Proof. The proof is essentially as for theorem 6.5.0.9, using amended versions of lemma 6.5.0.6 and lemma 6.5.0.8.

As for lemma 6.5.0.6, note that in **T**, using $\Box A\vdash A$, from $L,\Box A,A\vdash\bigvee R$ we may infer $L,\Box A\vdash\bigvee R$.

To prove an amended version of lemma 6.5.0.8, we have to change the definition of an associated Kripke model, definition 6.5.0.7, in such a way that the resulting model is always reflexive. Note that the change in the $\Box L$-rule reflects the axiom $\Box\phi\rightarrow\phi$ of **T**. If $L\bullet R$ is a split sequent and $\Box A\in L$, then in the leftmost non-closing reduction of $L\bullet R$, by the $\Box L$-rule, we will have $A\in L$. This is exactly what we need to change the proof of lemma 6.5.0.8 to apply to **T**. ⊣

6.6.2 K4test: a K4 tester

The modal logic **K4** has as its axioms and rules those of **K** plus the axiom $\Box\phi\rightarrow\Box\Box\phi$. A proof that **K4** is complete for finite transitive Kripke models can be found in [HC 84]. For the definition of *K4test*, a tester for the modal logic **K4**, we will extend the split sequents of *Ktest*. A split sequent of *K4test* is of the form $K;M;N\ (w;W)\ S;T;U$, where $K;M;N\bullet S;T;U$ is a split sequent of *Ktest*, w a *world*, a tuple $\langle X,Y\rangle$, with X and Y sets of formulas, and W a sequence of worlds.

The algorithm of *K4test*, given below, is obtained by amending the rules of *Ktest* for the split sequents of *K4test*, changing the NW-rule and adding a new rule restricting the applicability of the *K4test* rules.

The K4NW-rule is

$$\frac{K;M;\ (w,W)\ ;\Box A,T;U}{;M;M^*\ (w';W,w)\ A;;\quad K;M;\ (w;W)\ ;T;U}$$

where $M^* = \{B \mid \Box B \in M\}$.

This rule reflects the transitivity of the frames where the axiom $\Box\phi{\rightarrow}\Box\Box\phi$ is valid, by repeating all boxed formulas that have appeared on the left-hand side of the reduced split sequent.

The new rule of non-applicability declares that for a sequent $L(w;W)R$ with $w \in W$ no rule of *K4test* is applicable. In particular this may be the result of the K4NW-rule, if the world $w' = \langle M \cup M^*, \{A\}\rangle$ already occures in the list W, w of worlds that appeared above this split sequents in its construction from the starting split sequents, using the *K4test*-rules.

For the rules of *K4test* we use the same conventions as for *Ktest* and we will abbreviate $K;M;N\ (w;W)\ S;T;U$ by $L\ (w;W)\ R$.

$$(pR)\ \frac{L\ (w;W)\ p,R}{L\ (w;W)\ R,p} \qquad (pL)\ \frac{L,p\ (w;W)\ ;T;U}{p,L\ (w;W)\ ;T;U}$$

$$(\neg R)\ \frac{L\ (w;W)\ \neg A,R}{L,A\ (w;W)\ R,\neg A} \qquad (\neg L)\ \frac{L,\neg A\ (w;W)\ ;T;U}{\neg A,L\ (w;W)\ A;T;U}$$

$$(\wedge R)\ \frac{L\ (w;W)\ A\wedge B,R}{L\ (w;W)\ A,R,A\wedge B \quad L\ (w;W)\ B,R,A\wedge B}$$

$$(\wedge L)\ \frac{L,A\wedge B\ (w;W)\ ;T;U}{A\wedge B,L,A,B\ (w;W)\ ;T;U} \qquad (\vee R)\ \frac{L\ (w;W)\ A\vee B,R}{L\ (w;W)\ A,B,R,A\vee B}$$

$$(\vee L)\ \frac{L,A\vee B\ (w;W)\ ;T;U}{A\vee B,L,A\ (w;W)\ ;T \quad A\vee B,L,B\ (w;W)\ ;T;U}$$

$$(\rightarrow R)\ \frac{L\ (w;W)\ A{\rightarrow}B,R}{L,A\ (w;W)\ B,R,A{\rightarrow}B}$$

$$(\rightarrow L)\ \frac{L,A{\rightarrow}B\ (w;W)\ ;T;U}{A{\rightarrow}B,L\ (w;W)\ A;T \quad A{\rightarrow}B,L,B\ (w;W)\ ;T;U}$$

$$(\Box R)\ \frac{L\ (w;W)\ \Box A,S;T;U}{L\ (w;W)\ S;T,\Box A;U} \qquad (\Box L)\ \frac{K;M;N,\Box A\ (w;W)\ ;T;U}{K;\Box A,M;N\ (w;W)\ ;T;U}$$

$$(\Diamond R)\ \frac{K;M;N\ (w;W)\ \Diamond A,S;T;U}{K;\Box\neg A,M;N\ (w;W)\ S;T,\Box A;U} \qquad (\Diamond L)\ \frac{L,\Diamond A\ (w;W)\ ;T;U}{L\ (w;W)\ ;T,\Box\neg A;U}$$

The K4NW-rule

$$\frac{K;M;\ (w;W)\ ;\Box A,T;U}{;M;M^*\ (\langle M\cup M^*,\{A\}\rangle;W,w)\ A;;\quad K;M;\ (w;W)\ ;T;U}$$

where $M^*=\{B\mid \Box B\in M\}$.

Note that the top sequent of the K4NW-rule will be called closing if one of the resulting split sequents is closing.

6.6.2.3. Lemma. *The algorithm* K4test *is deterministic and terminates on the input of any split sequent.*

Proof. To see that *K4test* is deterministic, it can be verified that for each split sequent at most one rule of *K4test* is applicable.

To prove that *K4test* terminates on every split sequent, we can define a measure of complexity, $\sigma(X)$, on a set X of split sequents, like we did for *Ktest* in definition 6.5.0.1. We will not spell out this definition here, but the only difference with the one for *Ktest* will be a contribution for the $(w;W)$ part in the split sequent.

Let the initial sequent be $L(w;W)R$. The worlds that may appear in the application of the *K4test* rules to this sequent (and its resulting sequents) are tuples $\langle M,A\rangle$, where $M\cup A$ is a set of subformulas in the initial sequent $L(w;W)R$. If m is the number of these world-like tuples that may be made out of $L(w;W)R$ and n the number of worlds in W in the initial sequent, then for every split sequent $L'(w';W')R'$ that may be developed out of $L(w,W)R$ we have measure

$$\upsilon(L'(w',W')R')=m+n-\#W'+1$$

that is strictly decreasing after each non-closing application of the K4NW-rule.

Taking this υ into account, one can construct a strictly decreasing measure of complexity on a set of split sequents, as in definition 6.5.0.1. ⊣

To prove the counterpart of theorem 6.5.0.9 for *K4test*, we proceed as in section 6.5.

6.6.2.4. Lemma. *If a split sequent $L(w;W)R$ is closing (by the* K4test *rules) then $L\vdash\bigvee R$ (in* **K4***).*

Proof. For a closed split sequent the lemma is obvious. As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

All *K4test* rules can be treated as in the proof of lemma 6.5.0.6, except for the rule K4NW.

If $K;M;\ (w;W)\ ;T;U$ is closing then, by the induction hypothesis, $K,M\vdash\bigvee T\vee\bigvee U$. Then obviously also $K,M\vdash\Box A\vee\bigvee T\vee\bigvee U$.

On the other hand, if $;M;M^*\ (\langle M\cup M^*,\{A\}\rangle;W,w)\ A;;$ is closing, then, by the induction hypothesis, $M,M^*\vdash A$. Applying the necessitation rule, infer that $M\vdash\Box A$, as $M=\{\Box B\mid B\in M^*\}$ and by the **K4** axiom $M\vdash\Box\bigwedge M$. ⊣

For the proof of the following lemma, we will, as in section 6.5, associate a Kripke model to a non-closing split sequent $L(w;W)R$.

6.6.2.5. DEFINITION. *Let $L(w;W)R$ be a non-closing split sequent.* The Kripke model K associated with $L(w;W)R$ *is defined as a set of (leftmost) non-closing reduced split sequents, with a transitive relation ρ:*

1. *the leftmost non-closing reduction of $L(w;W)R$ is the root of K;*
2. *if $k_l \in K$ corresponds to the split sequent $K;M;(w;W);T;U$ and $T = \{\Box A_1, \ldots \Box A_t\} \neq \emptyset$ then the leftmost non-closing reductions of $;M;M^* \bullet A_i;;$ (where $M^* = \{B \mid \Box B \in M\}$ and $\Box A_i \in T$) are nodes of K, say respectively $l_1, \ldots, l_t$, such that for all i such that $1 \leq i \leq t$: $k_l \rho l_i$;*
3. *if $L(w;W)R$ is non-closing because of $w \in W$ and w was introduced in W by application of the K4NW-rule to $L'(w;W')R'$, then, if $L'(w;W')R$ corresponds to k_l and $L(w;W)R$ corresponds to k_m, we identify k_l and k_m.*
4. *if $k_l \in K$ corresponds to the split sequent $K;M;(w;W);;U$ (hence $T = \emptyset$) then k_l is a terminal node in K.*
5. *if $k_l \in K$ is the node corresponding to $L'(w;W)R'$, then $k_l \Vdash p$ for atomic formulas p iff $p \in L$.*

6.6.2.6. LEMMA. *If $L(w;W)R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L'(w;W)R'$ is the leftmost non-closing reduction of $L(w;W)R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *Ktest* rules (all except the K4NW-rule are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L'(w;W)R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction, a disjunction or an implication are obvious (using fact 6.5.0.5).

Let $A = \Box B$, $A \in L'$ and (as $L'(w;W)R'$ is reduced) $L' = K';M';$. Observe that A will be a member of M' and application of the K4NW-rule will result in a leftmost split sequent containing both A and B. Repeated applications of the K4NW-rule hereafter will result in (leftmost) spliting sequents with the same property.

Let $k_l \rho k_m$ and let $k_m \in K$ correspond to a reduced split sequent $L''(w'', W'')R''$. Now either $w' \in W'$ or $L''(w'', W'')R''$ is the result of (repeated) application of the K4NW-rule. From the observation above infer that in either case $B \in L''$. By induction hypothesis conclude that $k_m \Vdash B$. Which proves that $k_l \Vdash \Box B$.

Note that if no *K4test* rule is applicable for $L'(w;W)R'$ and $w \notin W$, then k_l is an irreflexive terminal node and trivially $k_l \Vdash \Box B$.

If $A = \Box B$ and $A \in R'$ then A will be in T and application of the K4NW-rule (and reduction) will result in a node k_m corresponding to a reduced split sequent $L''(w'';W'')R''$ such that $k_l < k_m$ and $B \in R''$. Using the induction hypothesis we conclude $k_m \nVdash B$ and hence $k_l \nVdash \Box B$. $\dashv$

6.6.2.7. THEOREM. *A split sequent $L(w;W)R$ is closing, using the* K4test *rules, iff* $L \vdash \bigvee R$.

Proof. By combining the previous two lemmas. ⊣

For the differences between *K4test* and *Ktest* one may compare the pseudo-code of *KMtest* with following pseudo-code program, *K4Mtest*, for the algorithm *K4test*.

$K4Mtest(K, M, N, S, T, U$: **sequence of** *formula*
w : *world*, W : **sequence of** *world*): **bool**
if $S \neq \emptyset$
then let $S = \langle A, S' \rangle$
if $A \in J \cup K \cup M \cup N$ **then true**
else in case A
atomic : $K4Mtest(K, M, N, S', T, \langle A, U \rangle, w, W)$
$\neg B$: $K4Mtest(K, M, \langle B, N \rangle, S', T, U, w, W)$
$B \wedge C$: **if** $K4Mtest(K, M, N, \langle B, S' \rangle, T, \langle A, U \rangle, w, W)$
then $K4Mtest(K, M, N, \langle C, S' \rangle, T, \langle A, U \rangle, w, W)$
else false
$B \vee C$: $K4Mtest(K, M, N, \langle B, C, S' \rangle, T, \langle A, U \rangle, w, W)$
$B \rightarrow C$: $KMtest(K, M, \langle B, N \rangle, \langle C, S' \rangle, T, \langle A, U \rangle, w, W)$
$\Box B$: $K4Mtest(K, M, N, S', \langle A, T \rangle, U, w, W)$
$\Diamond B$: $K4Mtest(K, M, N, S', \langle \neg B, T \rangle, U, w, W)$
else if $N \neq \emptyset$
then let $N = \langle A, N' \rangle$
if $A \in T \cup U$ **then true**
else in case A
atomic $K4Mtest(\langle A, K \rangle, M, N', , T, U, w, W)$
$\neg B$: $K4Mtest(\langle A, K \rangle, M, N', B, T, U, w, W)$
$B \wedge C$: $K4Mtest(\langle A, K \rangle, M, \langle A, B, N' \rangle, , T, U, w, W)$
$B \vee C$: **if** $K4Mtest(\langle A, K \rangle, M, \langle B, N \rangle, , T, U, w, W)$
then $K4Mtest(\langle A, K \rangle, M \langle C, N \rangle, , T, U, w, W)$
else false
$B \rightarrow C$: **if** $K4Mtest(\langle A, K \rangle, M, N, B, T, U, w, W)$
then $K4Mtest(\langle A, K \rangle, M, \langle C, N \rangle, , T, U, w, W)$
else false
$\Box B$: $K4Mtest(K, \langle A, M \rangle, N, , T, U, w, W)$
$\Diamond B$: $K4Mtest(K, M, N, , \langle T, A \rangle, U, w, W)$
else if $T \neq \emptyset$
then let $T = \langle \Box A, T' \rangle$ **and** $M^* = \{B \mid \Box B \in M\}$
and $w' = \langle M \cup M^*, \{A\} \rangle$
if $w' \in W \cup \{w'\}$ **then** $K4Mtest(K, M, , , T, U, w, W)$
else $K4Mtest(, M, M^*, A, , , w', \langle w, W \rangle)$
else false

6.6.3 S4test: an S4 tester

The modal logic **S4** has as its axioms and rules those of **K4** plus the axiom of **T**, $\Box\phi\rightarrow\phi$. A proof that **S4** is complete for finite reflexive and transitive Kripke models can be found in [HC 84]. The tester *S4test* is obtained by replacing the $\Box$L-rule in *K4test* by the Ttest-rule defined above.

6.6.3.8. Definition. *The tester* S4test *has the same rules as* K4test*, except for the* $\Box L$*-rule that is replaced by the* $\mathbf{T}\Box L$*-rule.*

6.6.3.9. Theorem. *A split sequent* $L(w;W)R$ *is closing, using the* S4test *rules, iff* $L \vdash \bigvee R$.

Proof. The proof is essentially as for theorem 6.6.2.7. The definition of the Kripke model associated with a non-closing sequent has to be changed in such a way that the model is always reflexive. Note that the change in the $\Box L$-rule reflects the addition of the **T** axiom and the reflexivity of the associated models. ⊣

6.6.4 Ltest: an L tester

The modal logic **L** has as its axioms and rules those of **K**, plus the *Löb axiom* $\Box(\Box A\rightarrow A)\rightarrow\Box A$. As in **L** the theorem $\Box A \vdash \Box\Box A$ is derivable, **L** is an extension of **K4**. A proof that **L** is complete for finite, transitive reverse well-founded Kripke models can be found in [Smoryński 85] and [Boolos 93]. The split sequents of the **L** tester *Ltest* will be of the same form as those for **K**.

6.6.4.10. Definition. *The tester* Ltest *has the same rules as* Ktest*, except for the NW-rule that is replaced by the* LNW-rule*:*

$$\frac{K;M;\ \bullet\ ;\Box A,T;U}{;M;M^*,\Box A\ \bullet\ A;;\quad K;M;\ (w';W.w)\ ;T;U}$$

where $M^* = \{B \mid \Box B \in M\}$.

6.6.4.11. Lemma. *The algorithm* Ltest *is deterministic and terminates on the input of any split sequent.*

Proof. The proof is essentially the same as for *K4test* in lemma 6.6.2.3

6.6.4.12. Lemma. *If a split sequent* $L\bullet R$ *is closing (by the* Ltest *rules) then* $L \vdash \bigvee R$ *(in* **L***).*

Proof. For a closed split sequent $L \bullet R$ the lemma is obvious.

As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

All *Ltest* rules can be treated as in the proof of lemma 6.5.0.6, except for the rule LNW, for which we can proceed as in the proof of lemma 6.6.2.4. ⊣

For *Ltest* we define the Kripke model associated with a non-closing split sequent as in definition 6.6.2.5, omitting the looping back rule 3.

6.6.4.13. LEMMA. *If $L \bullet R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L' \bullet R'$ is the leftmost non-closing reduction of $L \bullet R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *Ktest* rules (all except the LNW-rule are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L' \bullet R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction, a disjunction or an implication are obvious (using fact 6.5.0.5).

Let $A = \Box B$, $A \in L'$ and (as $L' \bullet R'$ is reduced) $L' = K'; M';$. Observe that, as in case of the K4NW-rule, A will be a member of M' and application of the LNW-rule will result in a leftmost split sequent containing both A and B. Repeated applications of the LNW-rule hereafter will result in (leftmost) spliting sequents with the same property.

Hence, for the proof of this lemma we can proceed as in the proof of lemma 6.6.2.4 (omitting the case where $w' \in W'$). $\dashv$

The pseudo-code program *LMtest* for the algorithm *Ltest* only differs slightly from the program *K4Mtest* in subsection 6.6.2.

$LMtest(K, M, N, S, T, U$: **sequence of** *formula*): **bool**
  if $S \neq \emptyset$
  then let $S = \langle A, S' \rangle$
    if $A \in J \cup K \cup M \cup N$ **then true**
    else in case A
      atomic : $LMtest(K, M, N, S', T, \langle A, U \rangle)$
      $\neg B$: $LMtest(K, M, \langle B, N \rangle, S', T, U)$
      $B \wedge C$: **if** $LMtest(K, M, N, \langle B, S' \rangle, T, \langle A, U \rangle)$
        then $LMtest(K, M, N, \langle C, S' \rangle, T, \langle A, U \rangle)$
        else false
      $B \vee C$: $LMtest(K, M, N, \langle B, C, S' \rangle, T, \langle A, U \rangle)$
      $B \rightarrow C$: $KMtest(K, M, \langle B, N \rangle, \langle C, S' \rangle, T, \langle A, U \rangle)$
      $\Box B$: $LMtest(K, M, N, S', \langle A, T \rangle, U)$
      $\Diamond B$: $LMtest(K, M, N, S', \langle \neg B, T \rangle, U)$
  else if $N \neq \emptyset$
    then let $N = \langle A, N' \rangle$
      if $A \in T \cup U$ **then true**
      else in case A
        atomic $LMtest(\langle A, K \rangle, M, N', , T, U)$
        $\neg B$: $LMtest(\langle A, K \rangle, M, N', B, T, U)$
        $B \wedge C$: $LMtest(\langle A, K \rangle, M, \langle A, B, N' \rangle, , T, U,)$
        $B \vee C$: **if** $LMtest(\langle A, K \rangle, M, \langle B, N \rangle, , T, U)$
          then $LMtest(\langle A, K \rangle, M \langle C, N \rangle, , T, U)$
          else false
        $B \rightarrow C$: **if** $LMtest(\langle A, K \rangle, M, N, B, T, U)$
          then $LMtest(\langle A, K \rangle, M, \langle C, N \rangle, , T, U)$
          else false
        $\Box B$: $LMtest(K, \langle A, M \rangle, N, , T, U)$
        $\Diamond B$: $LMtest(K, M, N, , \langle T, A \rangle, U)$
    else if $T \neq \emptyset$
      then let $T = \langle \Box A, T' \rangle$ **and** $M^* = \{B \mid \Box B \in M\}$
        if $LMtest(, M, \langle \Box A, M^* \rangle, A, ,)$ **then true**
        else $LMtest(K, M, , , T, U)$
      else false

Appendix A

Computer programs

A.1 Preliminaries

The computer programs in this appendix, written in the programming language C, all make use of a module that contains the types, functions and procedures that are common to the mkDiag program described in section 2.6 and the family of testers treated in Chapter 6. Parts of this module, supporting the understanding of the C-programs in the sequel, are listed below.

In the computer programs in this appendix formulas are represented by (pointers to) structures of the form:

```
struct formType
   { char            type;          /* -,&,|,:,L,M else the atom */
     struct formType *an;
     struct formType *co;
     unsigned        treated : 1;
     unsigned        revisit : 1;
   };

typedef struct formType *formula;
```

The *type* of a formula is denoted by its main connective. The list of possible connectives `-, &, |, :, L, M` corresponds with the list $\neg, \wedge, \vee, \rightarrow, \Box, \Diamond$. If the type character is not in this list, the formula is assumed to be atomic.

If a formula is not atomic, the main subformula(s) is (are) represented in the structure by a pointer to this (these) formula(s). The flags *'treated'* and *'revisit'* are used in the programs to mark the formulas as treated or as to be revisited.

Lists of formulas are represented by simple linked lists:

```
struct flistType
    { formula             element;
      struct flistType    *next;
    };

typedef struct flistType *formlist;
```

The utility module defines procedures for making and printing formulas (either on screen or in a file). For example the procedures *mkatom, mkNegation, mkConjunction* and *mkNecessarily*, to make atomic formulas, negations, conjunctions and necessitations are defined as:

```
formula mkAtom( char c )
{ formula form    = newForm();
  form->type      = findAtom( c );
  form->an        = NULL;
  form->co        = NULL;
  form->treated   = 1;
  return form;
}

formula mkNegation( formula x )
{ formula form = newForm();
  form->type   = '-';
  form->an     = x;
  return form;
}

formula mkConjunction( formula x, formula y )
{ formula form = newForm();
  form->type   = '&';
  form->an     = x;
  form->co     = y;
  return form;
}

formula mkNecessarily( formula x )
{ formula form = newForm();
  form->type   = 'L';
  form->an     = x;
  return form;
}
```

The function `newForm` allocates for a pointer the memory to be used to store the apointed formula structure. The function `findAtom` assigns a numeric character to the type of an atomic formula. This is not really needed for the programs described in this appendix.

A.2 The mkDiag program

A description of the *mkDiag* program can be found in section 2.6. The program makes use of a representation of a *Kripke model* K and an **IpL** *fragment* F to compute the equivalence classes in the fragment in the theory of the model. Hence two formulas ϕ and ψ are equivalent if

$$K \Vdash \phi \leftrightarrow \psi.$$

In the program, the **IpL** fragment F is given by the values of the constants `NEG, DNEG, CON, DIS, IMP` and `MaxMu`, corresponding to the connectives

$\neg, \neg\neg, \wedge, \vee, \rightarrow$ and the maximum level of nesting of the implication in F. The value of a connective-constant will be one or zero, depending on whether the corresponding connective is or is not in F.

The information on the Kripke model K is encoded in the constants `NE, NEM, ALL0, ALL1` and in the functions:

```
eset      comp( eset x);
eset      neg( eset x);
```

The constants `NE, NEM, ALL0` and `ALL1` all are involved in the representation of sets of worlds in K. The constants `NE` and `NEM` are related: `NE` $=$ `NEM` $+ 1$. A subset in K is called an `eset` (element set) in the program and is represented by an array of `NE` integers (`r[0]` to `r[NEM]`), each a binary encoding of a part of the model K. For $0 \leq i \leq$ `NEM` we have `ALL0` as an upper bound, $0 \leq$ `r[i]` $\leq$ `ALL0`. For the last part we have $0 \leq$ `r[NEM]` $\leq$ `ALL1`. In general it will be the case that `ALL1` $\leq$ `ALL0`.

With this information `comp(s)`, the complement of an `eset s`, can simply be calculated.

The *order* in K (the accessibility relation) is encoded in the function `neg`, computing the complement of the predecessor set as defined in definition 2.6.0.2. In section 2.6 it has been explained how the interior of a set in an **IpL** model can be calculated using complements and predecessor sets.

Apart from these procedures and those in the utility module (as described in the previous section), the program makes use of the following procedures:

```
void      classTest( char s, unsigned an, unsigned co, unsigned mu );
unsigned  noSet( eset x );
eset      meet( eset x, eset y );
eset      join( eset x, eset y );
unsigned  Inc( eset x, eset y);
void      fprintSet( FILE *f, eset x );
void      fprintVal( void );
```

The procedure `classTest` makes a new formula, computes the set of worlds in K where this formula is forced and tests (using the function `noSet`) whether or not this set already exists (in the list of formulas and sets `E`). The functions `meet` and `join` compute the meet and join of two sets and `Inc(x, y)` checks whether or not the set `x` is a subset of `y`. The procedure `fprintSet` prints a set (in a readable format) into a (text) file.

The result of the program `mkdiag` is an array `E` of pairs of sets and formulas defined as:

```
struct { eset       set;
         formula    form;
         unsigned   mu;
       } E[Dnr];
```

where `mu` can be used to calculate the nesting of the implication in the formulas and `Dnr` is the maximal number of classes `E` can contain. The number of classes in `E` is denoted by the variable `Emax`.

The output of the program is

1. a text file recording the equivalence classes found and their corresponding subsets in the model;
2. a file with formulas. Depending on one of the run-time parameters for the program, either the formulas in $Diag(F)$ or the representatives of the join-irreducible classes in the diagram are printed in this file;
3. a file describing the order of either the diagram or the set of join-irreducible elements in the diagram (again depending on a run-time parameter).

The last two files are made by the procedure `fprintVal` (not reprinted here).

The procedure *main* below is the main routine in the program *mkDiag*. Its listing is followed by the listings of the most important procedures used in *main* (i.e. *classTest, noSet, meet, join* and *Inc*).

```
main()
{ unsigned i, j, mu;
  char c;

  DiagramStart = 1 - NEG;
  if ( init() )
  { for ( mu=0; mu <= MaxMu; mu++ )
    { printf( "-------------\n mu= %d \n-------------\n", mu );
      fprintf( out, "-------------\n mu= %d \n-------------\n", mu );
      for ( i=DiagramStart; i <= Emax; i++)
      if ( E[i].mu == mu )
      { printf( "%5d  ", i );
        printForm( E[i].form );
        printf( "\n" );
        fprintf(out, "%5d  ", i );
        fprintForm( out, E[i].form );
        fprintSet( out, E[i].set );
      }

      for ( j = 2; j <= Emax; j++ )
      { if ( NEG && E[j].mu == mu - 1 ) classTest('-', j, i, mu);
        if ( DNEG && E[j].mu == mu - 2 ) classTest('d', j, i, mu);
        for ( i = 2; i < j && Emax < Dnr; i++ )
        { if ( E[j].mu == mu )
          { if ( !Inc(E[i].set, E[j].set) )
            { if ( IMP && E[i].mu < mu ) classTest( ':', i, j, mu );
              if ( !Inc(E[j].set, E[i].set) )
              { if ( CON ) classTest( '&', i, j, mu );
                if ( DIS ) classTest( '|', i, j, mu );
              }
            }
          }
          else
          { if ( IMP )
            { if ( E[i].mu == mu-1 && !Inc(E[i].set, E[j].set))
                classTest( ':', i, j, mu );
              if ( !Inc(E[j].set, E[i].set)
                    && ( E[i].mu == mu || E[j].mu == mu-1 ) )
```

```
                classTest( ':', j, i, mu );
              }
            }
          }
        }
    }
    fclose(out);
    fprintVal();
  }
}

void classTest( char s, unsigned an, unsigned co, unsigned mu )
{ unsigned counter, i, n = Emax + 1;
  eset nset;
  void *oldheaptop = getHeapTop();
/* may be the form made is not needed, so remember HeapTop */
  unsigned buz = 1, more, less;
  formula form, fan = E[an].form, fco = E[co].form;

  if ( n == Dnr )
  { printf( "there are too much classes: MaxHeap is too small\n" );
    fprintf( out,
   "there are too much classes: MaxHeap is too small\n" );
    fclose(out);
    exit(4);
  }

  switch ( s )
  { case '-'    :   form = mkNegation( fan );
                    nset = neg( E[an].set ); break;
    case 'd'    :   form = mkNegation(mkNegation( fan ) );
                    nset = neg(neg( E[an].set )); break;
    case '&'    :   form = mkConjunction( fan, fco );
                    nset = meet( E[an].set, E[co].set ); break;
    case '|'    :   form = mkDisjunction( fan, fco );
                    nset = join( E[an].set, E[co].set ); break;
    case ':'    :   form = mkImplication( fan, fco );
                    nset = neg(comp(join( comp(E[an].set),
                    E[co].set))); break;
  }

  if ( noSet(nset) )
  { E[n].form   = form;
    E[n].set    = nset;
    E[n].mu     = mu;
    Emax = n;
    printf( "%5d  ", n );
    printForm( form );
    printf( "\n" );
    fprintf(out, "%5d  ", n );
    fprintForm(out, form );
    fprintSet(out, nset );
    fprintf(out, "\n" );
  }
```

```
  else /* we don't need form anymore */
    setHeapTop( oldheaptop );
}

unsigned   noSet( eset x )
{ unsigned i, j, res=1;

  for (i=0; i<NE && res; i++) res = x.r[i] == 0;
  if ( res ) return E[0].mu > MaxMu;
  res = 1;
  for (i=0; i<NEM && res; i++) res = x.r[i] == ALL0;
  if ( res && x.r[NEM] == ALL1 ) return E[1].mu <= MaxMu ? 0 : 1;
  res = 0;
  for ( i=2; i <= Emax && !res; i++ )
  { res = 1;
    for ( j=0; j < NE && res ; j++) res = (E[i].set.r[j] == x.r[j]);
  }
  return !res;
}

eset  meet( eset x, eset y )
{ eset res;
  unsigned i;

  for (i=0; i < NE; i++) res.r[i] = x.r[i] & y.r[i];
  return res;
}

eset  join( eset x, eset y )
{ eset res;
  unsigned i;

  for (i=0; i < NE; i++) res.r[i] = x.r[i] | y.r[i];
  return res;
}

unsigned Inc( eset x, eset y)
{ unsigned i;

  for (i=0; i<NE && x.r[i] == (x.r[i] & y.r[i]); i++);
  if (i<NE) return 0;
  else return 1;
}
```

A.3 A simple CpL tester

In chapter 6 we calculated the complexity of the algorithm *Ctest*. For comparison we specify an algorithm *Cval* based on truth tables and calculate its complexity. This algorithm assumes a representation of atoms p_i such that calculating i from p_i is simple (i.e. linear in the size of p_i).

For natural numbers i and N, $i \in N$ means that if N is taken as a binary number representing some subset S of $\{0, \ldots, n-1\}$, that $i \in S$.

We will assume that the indices of the atoms in a formula to be tested form some sequence $\{0, \ldots, n-1\}$.

Global $N : I\!N$
$Cval(A : formula)$: **bool**
  Calculate n the number of atoms in A
  $N = 0$
  while $N < 2^n$ **and** $SubVal(A)$
    $N := N + 1$
  if $N < 2^n$ **then false else true**

$SubVal(A : formula)$: **bool**
  in case A

p_i	:	**if** $i \in N$ **then true**
		else false
$\neg B$	:	**if** $SubVal(B)$ **then false**
		else true
$B \wedge C$	:	**if** $SubVal(B)$ **then** $SubVal(C)$
		else false
$B \vee C$	:	**if** $SubVal(B)$ **then true**
		else $SubVal(C)$
$B \rightarrow C$	:	**if** $SubVal(C)$ **then true**
		else if $SubVal(C)$ **then false**
		else true

To calculate $Cval(\phi)$ for some formula ϕ note that:

1. the main part of $Cval$ needs storage for ϕ and three numbers;
2. the number of atoms in ϕ can be calculated in time and space both linear in $|\phi|$;
3. in $SubVal$ the formula is to be split in its principal subformulas, which takes time in the order of $|\phi|$;
4. $SubVal$ needs space to store three formulas;
5. the number of atoms in ϕ is at most $|\phi|$ and hence there will be at most $2^{|\phi|}$ calls to $SubVal$ (which is also an upperbound of the number of items on stack;

Disregarding the small constants this amounts in an upper bound on the time needed to calculate $Cval(\phi)$ of order $|\phi|.2^{|\phi|}$. Likewise the upper bound on the amount of space needed is of the order $3.|\phi|.2^{|\phi|}$.

A.4 The IpLtest program

The *IpLtest* program is a rather straightforward implementation of the algorithm *IpLtest* (and the pseudo-code program *Itest*) in Chapter 6.

Many of the utilities used in this program do exactly what one would expect them to do.

For example `copy` does make a copy of a formula and `notDisjunct` checks whether or not two formula lists have a common formula. Both `putRight` and `putLeft` add a formula to a formula list, but in the procedure `putLeft`, if the added formula is atomic and does not occurs in the list as an already treated formula, the global flag `LeftChange` is set (compare the rule *pL*3 in the definition of *IpLtest* in Chapter 6. Note that we use a variable `oldvalue` to keep the previous value of `LeftChange` in store if needed.

The procedure `markRevisit` marks the formulas in a list as to be revisited and the function `untreatedFormula` takes an untreated formula out of a formula list (taking value `NULL` if there is no such formula in the list). In the same way `revisitLeftside` has as its result the list of all formulas marked to be revisited, out of a given formula list. And the function `revisitRightside` takes a formula marked to be revisited out of a formula list (again, with value `NULL` if there is no such formula in the list).

The main procedure, `refutable`, has as its input two lists of formulas, `left` and `right` and returns the value zero iff none of the formulas in the list `right` is a consequence (in **IpL**) of the formulas in the list `left`.

If necessary the program writes error messages to an output file (for which a pointer `out` is used).

```
/* FUNCTIONS */
int refutable( formlist left, formlist right )
{formula  form, cform;
 formlist flist;
 int oldval = LeftChange, res;

 if ( notDisjunct( left, right ) ) res = 0;
 else
 { if ( form = untreatedFormula( right ), form )
   { form->treated = 1;
     switch( form->type )
     { case '-' : form->revisit = 1;
                  res = refutable( left, right );
                  form->revisit = 0;
                  break;
       case '&' : res = refutable(left, putRight(form->an,right))
                  ? 1
                  : refutable(left, putRight(form->co,right));
                  break;
       case '|' : res = refutable(left,
                  putRight(form->an,
                  putRight(form->co, right)));
                  break;
       case '=' : res = refutable(left,
                  putRight(
```

```
                  mkConjunction(
                     mkImplication(form->an, form->co),
                     mkImplication(form->co, form->an))
                  right) );
               break;
    case ':' : form->revisit = 1;
               res = refutable( left, right );
               form->revisit = 0;
               break;
  }
  form->treated = 0;
}
else
{ if ( form = untreatedFormula( left ), form )
  { form->treated = 1;
    switch( form->type )
    { case '-' : form->revisit = 1;
                 res = refutable(left,
                 putRight(copy( form->an ),right));
                 form->revisit = 0;
                 break;
      case '&' : res = refutable(putLeft(form->an,
                 putLeft(form->co, left)), right);
                 break;
      case '|' : res = refutable(putLeft(form->an, left), right)
                 ? 1
                 : ( LeftChange = oldval,
                     refutable(putLeft(form->co, left), right)
                   );
                 break;
      case '=' : res = refutable(
                          putLeft(
                             mkConjunction(
                                mkImplication(form->an, form->co),
                                mkImplication(form->co, form->an)),
                             left),
                          right );
                        break;
      case ':' : if ( refutable(putLeft(form->co,left), right) )
                   res = 1;
                 else
                 { LeftChange = oldval;
                   form->revisit = 1;
                   res = refutable(left,
                                      putRight(copy( form->an),
                                   right));
                   form->revisit = 0;
                 }
                 form->revisit = 0;
                 break;
    }
    form->treated = 0;
  }
  else
```

```
    { flist = revisitLeftside( left );
      if ( flist )
      { LeftChange = 0;
        res = refutable( left, right );
        LeftChange = 1;
        markRevisit( flist );
      }
      else
      { form = revisitRightside( right );
        if ( form )
        { form->revisit = 0;
          switch( form->type )
          { case '-' : if ( refutable(putLeft(form->an, left), NULL) )
                       { LeftChange = oldval;
                         res = refutable(left, right );
                       }
                       else res = 0;
                       break;
            case ':' : if ( refutable(putLeft(form->an,left),
                                putRight(form->co,NULL)) )
                       { LeftChange = oldval;
                         res = refutable(left, right);
                       }
                       else res = 0;
                       break;
          }
          form->revisit = 1;
        }
        else res = 1;
      }
    }
  }
 }
 LeftChange = oldval;
 return res;
}
```

A.5 Testers for modal logic

The modal testers described in Chapter 6 have been implemented in one module. Depending on the setting of the constants `T, K4, S5, L` and `Grz` the module is compiled as a tester for **K**, **T**, **K4**, **L**, **S4**, **S5** or Grzegorczyk's logic **K4Grz** (the last two not treated in this thesis).

```
#define T  0  /* Lp->p                                  */
#define K4 0  /* Lp->LLp  S4 == T + K4                  */
#define S5 0  /* S5 == 1  =>  S4 == 1                   */
#define L  0  /* L(Lp->p)->Lp, L == 1   =>  K4 == 1     */
#define Grz 0 /* L(L(p->Lp)->p)->p, Grz == 1 => K4 == 1 */
```

Many of the procedures in the program for the modal testers are the same (at least in principle) as in the *IpLtest* program above. Some noteworthy exceptions are

`put`, simply adding a formula to a formula list, and the procedures dealing with (lists of) worlds. Both for worlds and lists of worlds we use pointers:

```
struct worldType
       { formlist       wleft;
         formlist       wright;
       };

typedef struct worldType  *world;

struct wlistType
       { world               element;
         struct wlistType    *next;
       };

typedef struct wlistType  *worldlist;
```

The structure of worlds and their rôle in the algorithm *K4test* has been explained in subsection 6.6.2.

The procedures `newWorld` and `newWorldList` allocate memory needed to store the data of appointed `world` structures and `worldlist` structures. To add worlds to a list of worlds, there is a procedure `addWorld` and to find out whether a given world is in a given list of worlds, there is a procedure `memberworld`.

Again, the main procedure for the program for the formula tester(s) is `refutable(left, right, worlds)`, returning the value 1 iff there is a Kripke model starting with the list of worlds `worlds`, forcing all the formulas in the list `left` and no formula in the list `right`.

```
formlist revisitLeftside( formlist x )
{ formlist res = NULL;
  formula  an, el;

  if ( LeftChange)
  { while ( x )
    { el = x->element;
      if ( el->revisit )
      { an = el->an;
        if (!member(an, res))      /* K */
          res = add( copy(an), res );
#if K4 == 1
        if (!member(el, res))
          res = add( copy(el), res );
#endif
#if K5 == 1
        else res = put(mkPossibly(el), res);
#endif
      }
      x = x->next;
    }
  }
  return res;
}
```

```
formula revisitRightside( formlist x )
{ formula res = NULL;
  while ( !res && x )
  { if ( (x->element)->revisit ) res = x->element;
    else x = x->next;
  }
  return res;
}

formlist addModal( formula f, formlist x)
{ formula g;
  formlist l, y;

  l = x;
  y = add(f->an, NULL);
  while ( l )
  { g = l->element;
    if ( g->revisit && g != f ) y = add(g, y);
    l = l->next;
  }
  return y;
}

int refutable( formlist left, formlist right, worldlist worlds )
{ formula     form,
              cform;
  formlist    flist,
              glist;
  int         oldval = LeftChange,
              putback,
              res = 1;
  world       nwworld;
  worldlist   nwworlds; /* wlist to debug */

  if ( notDisjunct( left, right ) ) res = 0;
  else
  { if ( form = untreatedFormula( right ), form )
    { form->treated = 1;
      switch( form->type )
      { case '-' : res = refutable( put(form->an, left), right, worlds );
                   break;
        case 'L' : form->revisit = 1;
                   res = refutable( left, right, worlds );
                   form->revisit = 0;
                   break;
        case 'M' : res = refutable(
                           put( mkNecessarily(mkNegation(form->an)), left ),
                                right, worlds );
                   break;
        case '&' : res = refutable(left, put(form->an, right), worlds)
                   ? 1
                   : refutable(left, put(form->co, right), worlds);
                   break;
        case '|' : res = refutable(left, put(form->an,
```

```
                                          put(form->co, right)), worlds);
                    break;
          case '=' : res = refutable( left, put(
                                          mkConjunction(
                                             mkImplication(
                                                form->an,
                                                form->co),
                                             mkImplication(
                                                form->co,
                                                form->an)),
                                           right), worlds );
                    break;
          case ':' : res = refutable( put(form->an, left),
                                      put(form->co, right), worlds );
                    break;
        }
        form->treated = 0;
      }
      else
      { if ( form = untreatedFormula( left ), form )
        { form->treated = 1;
          switch( form->type )
          { case '-' : res = refutable(left, put(copy( form->an ), right)
                                           , worlds);
                       break;
#if T == 1
            case 'L' : LeftChange = 1;
                       form->revisit = 1;
                       res = refutable(put(form->an, left), right, worlds);
                       form->revisit = 0;
                       break;
#else
            case 'L' : LeftChange = 1;
                       form->revisit = 1;
                       res = refutable(left, right, worlds);
                       form->revisit = 0;
                       break;
#endif
            case 'M' : res = refutable(
                                left,
                                put(mkNecessarily(mkNegation(form->an)),
                                    right), worlds );
                       break;
            case '&' : res = refutable( put(
                                          form->an,
                                          put(form->co, left)),
                                       right, worlds );
                       break;
            case '|' : res = refutable( put(form->an, left), right, worlds )
                       ? 1
                       : ( LeftChange = oldval,
                       refutable( put(form->co, left), right, worlds ));
                       break;
            case '=' : res = refutable( put(
```

```
                                        mkImplication(form->an, form->co),
                                        put(
                                          mkImplication(form->co, form->an),
                                          left)),
                                      right, worlds);
                    break;
          case ':' : if ( refutable(put(form->co, left), right, worlds ))
                    res = 1;
                    else
                    { LeftChange = oldval;
                      /* form->revisit = 1; */
                      res = refutable(left, put(copy(form->an), right),
                                                worlds);
                    }
                    break;
          }
          form->treated = 0;
      }
      else
      { if (LeftChange)
        { form = revisitRightside( right );
          if (form)
          { form->revisit = 0;
#if T == 1
            res = member(form->an, right);
            /* if res then the branch will stay open */
            if (!res)
            {
#endif
              flist = revisitLeftside( left );
              LeftChange = 0;
              glist = put(form->an, NULL);
              #if S5 == 1
              glist = addModal(form, right);
#endif
#if L == 1
              if (!member(form, flist)) flist = add(copy(form), flist);
#endif
#if K4 == 0 || L == 1
              res = refutable(flist, glist, worlds);
#else
              nwworld = newWorld();
              nwworld->wleft = flist;
              nwworld->wright = glist;
              res = memberworld(nwworld, worlds);
              /* if res then the branch will stay open */
              if (!res)
              { nwworlds = addWorld(nwworld, worlds);
                res = refutable(flist, glist, nwworlds);
              }
#endif
#if Grz == 1
              else res = 0;
#endif
```

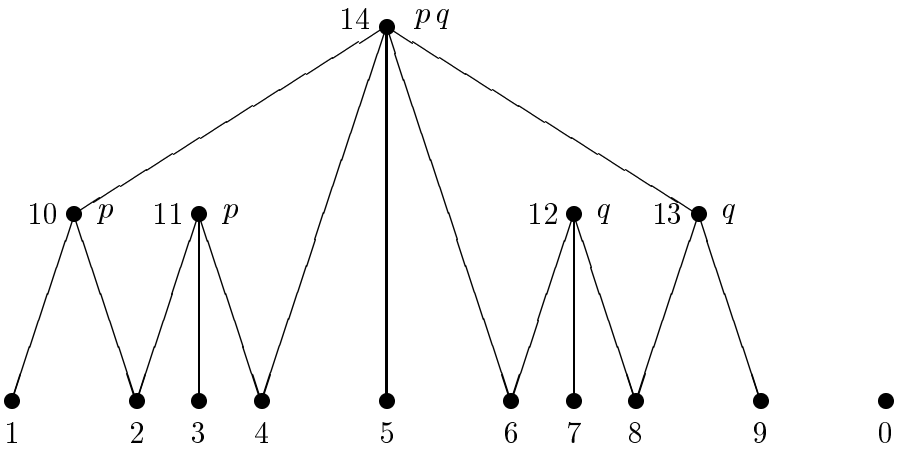
```
#if T == 1
            }
#endif
            if (res)
            { LeftChange = oldval;
              res = refutable(left, right, worlds);
            }
            else res = 0;
            LeftChange = 1;
            form->revisit = 1;
          }
          else res = 1;
        }
        else res = 1;
      }
    }
  }
  LeftChange = oldval;
  return res;
}
```

Appendix B

Output of computer programs

B.1 The diagram of the IpL fragment $[\rightarrow, \neg]^2$

The fragment $[\rightarrow, \neg]^n$ was treated in subsection 3.5.1. The diagram of $[\rightarrow, \neg]^2$, listed below, was computed using the exact Kripke model of $[\wedge, \rightarrow, \neg]^2$ (compare figure 20):



31. FIGURE. *The exact model of* $[\wedge, \rightarrow, \neg]^2$.

0	$\neg(p\rightarrow p)$	$\{\}$
1	$(p\rightarrow p)$	$\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$
2	p	$\{10,11,14\}$
3	q	$\{12,13,14\}$
4	$\neg p$	$\{0,7,12\}$
5	$\neg q$	$\{0,3,11\}$
6	$(p\rightarrow q)$	$\{0,5,6,7,8,9,12,13,14\}$
7	$(q\rightarrow p)$	$\{0,1,2,3,4,5,10,11,14\}$
8	$(q\rightarrow\neg p)$	$\{0,3,7,11,12\}$
9	$\neg\neg p$	$\{1,2,3,4,5,9,10,11,13,14\}$
10	$(\neg p\rightarrow q)$	$\{1,2,3,4,5,6,8,9,10,11,12,13,14\}$
11,	$\neg\neg q$	$\{1,5,6,7,8,9,10,12,13,14\}$
12,	$(\neg q\rightarrow p)$	$\{1,2,4,5,6,7,8,9,10,11,12,13,14\}$
13,	$(\neg p\rightarrow\neg q)$	$\{0,1,2,3,4,5,9,10,11,13,14\}$
14	$(\neg q\rightarrow\neg p)$	$\{0,1,5,6,7,8,9,10,12,13,14\}$

15	$\neg(p\to q)$	$\{3,11\}$
16	$((p\to q)\to p)$	$\{1,2,3,4,10,11,14\}$
17	$((p\to q)\to q)$	$\{1,2,3,4,10,11,12,13,14\}$
18	$\neg(q\to p)$	$\{7,12\}$
19	$((q\to p)\to p)$	$\{6,7,8,9,10,11,12,13,14\}$
20	$((q\to p)\to q)$	$\{6,7,8,9,12,13,14\}$
21	$\neg(q\to\neg p)$	$\{1,5,9,10,13,14\}$
22	$((q\to\neg p)\to p)$	$\{1,2,4,5,9,10,11,13,14\}$
23	$((q\to\neg p)\to q)$	$\{1,5,6,8,9,10,12,13,14\}$
24	$(\neg q\to\neg\neg p)$	$\{1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$
25	$((q\to p)\to\neg(p\to q))$	$\{3,7,11,12\}$
26	$((q\to p)\to((p\to q)\to p))$	$\{1,2,3,4,6,7,8,9,10,11,12,13,14\}$
27	$(\neg\neg p\to p)$	$\{0,6,7,10,11,12,14\}$
28	$(\neg\neg p\to q)$	$\{0,6,7,8,12,13,14\}$
29	$(\neg\neg p\to(q\to p))$	$\{0,1,2,3,4,5,6,7,10,11,12,14\}$
30	$\neg(\neg p\to q)$	$\{0\}$
31	$((\neg p\to q)\to p)$	$\{0,10,11,14\}$
32	$((\neg p\to q)\to q)$	$\{0,7,12,13,14\}$
33	$(\neg\neg q\to p)$	$\{0,2,3,4,10,11,14\}$
34	$(\neg\neg q\to q)$	$\{0,3,4,11,12,13,14\}$
35	$(\neg\neg q\to(p\to q))$	$\{0,3,4,5,6,7,8,9,11,12,13,14\}$
36	$(\neg\neg q\to(\neg p\to q))$	$\{0,1,2,3,4,5,6,8,9,10,11,12,13,14\}$
37	$((\neg q\to p)\to p)$	$\{0,3,10,11,14\}$
38	$((\neg q\to p)\to q)$	$\{0,12,13,14\}$
39	$(\neg\neg p\to(\neg q\to p))$	$\{0,1,2,4,5,6,7,8,9,10,11,12,13,14\}$
40	$((\neg p\to\neg q)\to p)$	$\{6,7,10,11,12,14\}$
41	$((\neg p\to\neg q)\to q)$	$\{6,7,8,12,13,14\}$
42	$((\neg q\to\neg p)\to p)$	$\{2,3,4,10,11,14\}$
43	$((\neg q\to\neg p)\to q)$	$\{3,4,11,12,13,14\}$
44	$(((p\to q)\to p)\to p)$	$\{0,5,6,7,8,9,10,11,12,13,14\}$
45	$(\neg\neg p\to((p\to q)\to p))$	$\{0,1,2,3,4,6,7,10,11,12,14\}$
46	$((\neg p\to q)\to((p\to q)\to p))$	$\{0,1,2,3,4,10,11,14\}$
47	$((\neg p\to\neg q)\to((p\to q)\to p))$	$\{1,2,3,4,6,7,10,11,12,14\}$
48	$(((p\to q)\to q)\to p)$	$\{0,5,10,11,14\}$
49	$(\neg\neg p\to((p\to q)\to q))$	$\{0,1,2,3,4,6,7,8,10,11,12,13,14\}$
50	$((\neg p\to q)\to((p\to q)\to q))$	$\{0,1,2,3,4,7,10,11,12,13,14\}$
51	$(\neg\neg q\to((p\to q)\to q))$	$\{0,1,2,3,4,10,11,12,13,14\}$
52	$((\neg p\to\neg q)\to((p\to q)\to q))$	$\{1,2,3,4,6,7,8,10,11,12,13,14\}$
53	$(((q\to p)\to p)\to q)$	$\{0,5,12,13,14\}$
54	$(\neg\neg p\to((q\to p)\to p))$	$\{0,6,7,8,9,10,11,12,13,14\}$
55	$(\neg\neg q\to((q\to p)\to p))$	$\{0,2,3,4,6,7,8,9,10,11,12,13,14\}$
56	$((\neg q\to p)\to((q\to p)\to p))$	$\{0,3,6,7,8,9,10,11,12,13,14\}$
57	$((\neg q\to\neg p)\to((q\to p)\to p))$	$\{2,3,4,6,7,8,9,10,11,12,13,14\}$
58	$(((q\to p)\to p)\to((p\to q)\to q))$	$\{0,1,2,3,4,5,10,11,12,13,14\}$
59	$(\neg\neg p\to((q\to p)\to q))$	$\{0,6,7,8,9,12,13,14\}$
60	$(\neg\neg q\to((q\to p)\to q))$	$\{0,3,4,6,7,8,9,11,12,13,14\}$
61	$((\neg q\to\neg p)\to((q\to p)\to q))$	$\{3,4,6,7,8,9,11,12,13,14\}$
62	$(\neg(q\to\neg p)\to p)$	$\{0,2,3,4,6,7,10,11,12,14\}$
63	$(\neg(q\to\neg p)\to q)$	$\{0,3,4,6,7,8,11,12,13,14\}$
64	$((\neg p\to q)\to\neg(q\to\neg p))$	$\{0,1,5,9,10,13,14\}$
65	$(((q\to\neg p)\to p)\to p)$	$\{0,3,6,7,10,11,12,14\}$
66	$((\neg p\to q)\to((q\to\neg p)\to p))$	$\{0,1,2,4,5,9,10,11,13,14\}$
67	$(((q\to\neg p)\to q)\to q)$	$\{0,3,4,7,11,12,13,14\}$
68	$((\neg q\to p)\to((q\to\neg p)\to q))$	$\{0,1,5,6,8,9,10,12,13,14\}$
69	$(\neg\neg p\to((q\to p)\to((p\to q)\to p)))$	$\{0,1,2,3,4,6,7,8,9,10,11,12,13,14\}$
70	$(((p\to q)\to q)\to(\neg\neg p\to p))$	$\{0,5,6,7,10,11,12,14\}$
71	$(((q\to p)\to p)\to(\neg\neg p\to q))$	$\{0,5,6,7,8,12,13,14\}$
72	$(((q\to p)\to q)\to(\neg\neg p\to q))$	$\{0,1,2,3,4,5,6,7,8,10,11,12,13,14\}$
73	$(((q\to p)\to p)\to((\neg p\to q)\to q))$	$\{0,5,7,12,13,14\}$
74	$(((q\to p)\to q)\to((\neg p\to q)\to q))$	$\{0,1,2,3,4,5,7,10,11,12,13,14\}$
75	$(((p\to q)\to p)\to(\neg\neg q\to p))$	$\{0,2,3,4,5,6,7,8,9,10,11,12,13,14\}$
76	$(((p\to q)\to q)\to(\neg\neg q\to p))$	$\{0,2,3,4,5,10,11,14\}$
77	$(((q\to p)\to p)\to(\neg\neg q\to q))$	$\{0,3,4,5,11,12,13,14\}$
78	$(((p\to q)\to p)\to((\neg q\to p)\to p))$	$\{0,3,5,6,7,8,9,10,11,12,13,14\}$
79	$(((p\to q)\to q)\to((\neg q\to p)\to p))$	$\{0,3,5,10,11,14\}$
80	$((\neg q\to\neg p)\to((\neg p\to\neg q)\to p))$	$\{2,3,4,6,7,10,11,12,14\}$
81	$((\neg q\to\neg p)\to((\neg p\to\neg q)\to q))$	$\{3,4,6,7,8,11,12,13,14\}$
82	$(\neg(q\to\neg p)\to(((p\to q)\to q)\to p))$	$\{0,2,3,4,5,6,7,10,11,12,14\}$

83	$(((q\to\neg p)\to p)\to(((p\to q)\to q)\to p))$	$\{0,3,5,6,7,10,11,12,14\}$
84	$(\neg(q\to\neg p)\to(((q\to p)\to p)\to q))$	$\{0,3,4,5,6,7,8,11,12,13,14\}$
85	$(((q\to\neg p)\to q)\to(((q\to p)\to p)\to q))$	$\{0,3,4,5,7,11,12,13,14\}$
86	$((\neg\neg p\to p)\to q)$	$\{5,8,9,12,13,14\}$
87	$((\neg\neg p\to p)\to((p\to q)\to q))$	$\{1,2,3,4,5,8,9,10,11,12,13,14\}$
88	$((\neg\neg p\to p)\to((q\to p)\to q))$	$\{5,6,7,8,9,12,13,14\}$
89	$((\neg\neg p\to q)\to p)$	$\{1,2,3,4,5,10,11,14\}$
90	$((\neg\neg p\to q)\to q)$	$\{1,2,3,4,5,9,10,11,12,13,14\}$
91	$((\neg\neg p\to(q\to p))\to p)$	$\{9,10,11,13,14\}$
92	$((\neg\neg p\to(q\to p))\to q)$	$\{8,9,12,13,14\}$
93	$((\neg\neg p\to(q\to p))\to((p\to q)\to p))$	$\{1,2,3,4,9,10,11,13,14\}$
94	$((\neg\neg p\to(q\to p))\to((p\to q)\to q))$	$\{1,2,3,4,8,9,10,11,12,13,14\}$
95	$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p))$	$\{0,9,10,11,13,14\}$
96	$((\neg\neg p\to p)\to((\neg p\to q)\to q))$	$\{0,5,7,8,9,12,13,14\}$
97	$((\neg\neg p\to q)\to((\neg p\to q)\to q))$	$\{0,1,2,3,4,5,7,9,10,11,12,13,14\}$
98	$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to q))$	$\{0,7,8,9,12,13,14\}$
99	$((\neg\neg q\to p)\to p)$	$\{1,5,6,7,8,9,10,11,12,13,14\}$
100	$((\neg\neg q\to p)\to(\neg\neg p\to p))$	$\{0,1,5,6,7,8,9,10,11,12,13,14\}$
101	$((\neg\neg p\to(q\to p))\to(\neg\neg q\to p))$	$\{0,2,3,4,9,10,11,13,14\}$
102	$((\neg\neg q\to q)\to p)$	$\{1,2,5,10,11,14\}$
103	$((\neg\neg q\to q)\to((q\to p)\to p))$	$\{1,2,5,6,7,8,9,10,11,12,13,14\}$
104	$((\neg\neg p\to p)\to(\neg\neg q\to q))$	$\{0,3,4,5,8,9,11,12,13,14\}$
105	$((\neg\neg q\to q)\to(\neg\neg p\to p))$	$\{0,1,2,5,6,7,10,11,12,14\}$
106	$((\neg\neg p\to q)\to(\neg\neg q\to q))$	$\{0,1,2,3,4,5,9,10,11,12,13,14\}$
107	$((\neg\neg p\to(q\to p))\to(\neg\neg q\to q))$	$\{0,3,4,8,9,11,12,13,14\}$
108	$((\neg\neg q\to q)\to((\neg p\to q)\to p))$	$\{0,1,2,5,10,11,14\}$
109	$((\neg\neg q\to(p\to q))\to p)$	$\{1,2,10,11,14\}$
11, 0	$((\neg\neg q\to(p\to q))\to q)$	$\{1,10,12,13,14\}$
11, 1	$((\neg\neg q\to(p\to q))\to((q\to p)\to p))$	$\{1,2,6,7,8,9,10,11,12,13,14\}$
11, 2	$((\neg\neg q\to(p\to q))\to((q\to p)\to q))$	$\{1,6,7,8,9,10,12,13,14\}$
11, 3	$((\neg\neg q\to(p\to q))\to(\neg\neg p\to p))$	$\{0,1,2,6,7,10,11,12,14\}$
11, 4	$((\neg\neg q\to(p\to q))\to(\neg\neg p\to q))$	$\{0,1,6,7,8,10,12,13,14\}$
11, 5	$((\neg\neg q\to(p\to q))\to((\neg p\to q)\to p))$	$\{0,1,2,10,11,14\}$
11, 6	$((\neg\neg q\to(p\to q))\to((\neg p\to q)\to q))$	$\{0,1,7,10,12,13,14\}$
11, 7	$((\neg\neg q\to(\neg p\to q))\to q)$	$\{7,12,13,14\}$
11, 8	$((\neg\neg q\to(\neg p\to q))\to((p\to q)\to q))$	$\{1,2,3,4,7,10,11,12,13,14\}$
11, 9	$((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p))$	$\{0,3,9,10,11,13,14\}$
12, 0	$((\neg\neg q\to p)\to((\neg q\to p)\to p))$	$\{0,1,3,5,6,7,8,9,10,11,12,13,14\}$
12, 1	$((\neg\neg q\to q)\to((\neg q\to p)\to p))$	$\{0,1,2,3,5,10,11,14\}$
12, 2	$((\neg\neg q\to(p\to q))\to((\neg q\to p)\to p))$	$\{0,1,2,3,10,11,14\}$
12, 3	$((\neg\neg p\to p)\to((\neg q\to p)\to q))$	$\{0,5,8,9,12,13,14\}$
12, 4	$((\neg\neg p\to(q\to p))\to((\neg q\to p)\to q))$	$\{0,8,9,12,13,14\}$
12, 5	$((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q))$	$\{0,1,10,12,13,14\}$
12, 6	$((\neg\neg p\to(\neg q\to p))\to p)$	$\{3,10,11,14\}$
12, 7	$((\neg\neg p\to(\neg q\to p))\to((q\to p)\to p))$	$\{3,6,7,8,9,10,11,12,13,14\}$
12, 8	$(((\neg p\to\neg q)\to p)\to((p\to q)\to q))$	$\{0,1,2,3,4,5,8,9,10,11,12,13,14\}$
12, 9	$((\neg\neg p\to q)\to((\neg p\to\neg q)\to p))$	$\{1,2,3,4,5,6,7,10,11,12,14\}$
13, 0	$((\neg\neg q\to q)\to((\neg p\to\neg q)\to p))$	$\{1,2,5,6,7,10,11,12,14\}$
13, 1	$((\neg\neg q\to(p\to q))\to((\neg p\to\neg q)\to p))$	$\{1,2,6,7,10,11,12,14\}$
13, 2	$((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p))$	$\{3,6,7,10,11,12,14\}$
13, 3	$((\neg\neg q\to(p\to q))\to((\neg p\to\neg q)\to q))$	$\{1,6,7,8,10,12,13,14\}$
13, 4	$((\neg\neg p\to(q\to p))\to((\neg q\to\neg p)\to p))$	$\{2,3,4,9,10,11,13,14\}$
13, 5	$(((\neg q\to\neg p)\to q)\to((q\to p)\to p))$	$\{0,1,2,5,6,7,8,9,10,11,12,13,14\}$
13, 6	$((\neg\neg p\to p)\to((\neg q\to\neg p)\to q))$	$\{3,4,5,8,9,11,12,13,14\}$
13, 7	$((\neg\neg p\to(q\to p))\to((\neg q\to\neg p)\to q))$	$\{3,4,8,9,11,12,13,14\}$
13, 8	$(((\neg p\to q)\to p)\to((\neg q\to\neg p)\to q))$	$\{3,4,5,6,7,8,9,11,12,13,14\}$
13, 9	$((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,7,11,12,13,14\}$
140	$((\neg\neg p\to((p\to q)\to p))\to p)$	$\{5,9,10,11,13,14\}$
141	$((\neg\neg p\to((p\to q)\to p))\to((q\to p)\to p))$	$\{5,6,7,8,9,10,11,12,13,14\}$
142	$((\neg\neg p\to((p\to q)\to p))\to((\neg p\to q)\to p))$	$\{0,5,9,10,11,13,14\}$
143	$((\neg\neg p\to((p\to q)\to p))\to(\neg\neg q\to p))$	$\{0,2,3,4,5,9,10,11,13,14\}$
144	$((\neg\neg p\to((p\to q)\to p))\to((\neg q\to p)\to p))$	$\{0,3,5,9,10,11,13,14\}$
145	$((\neg\neg p\to((p\to q)\to p))\to((\neg q\to\neg p)\to p))$	$\{2,3,4,5,9,10,11,13,14\}$
146	$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to p)))$	$\{0,1,2,3,4,9,10,11,13,14\}$
147	$(((\neg p\to q)\to((p\to q)\to p))\to((\neg q\to\neg p)\to p))$	$\{2,3,4,5,6,7,8,9,10,11,12,13,14\}$
148	$((\neg\neg p\to((p\to q)\to q))\to p)$	$\{5,10,11,14\}$
149	$((\neg\neg p\to((p\to q)\to q))\to q)$	$\{5,9,12,13,14\}$
150	$((\neg\neg p\to((p\to q)\to q))\to((\neg p\to q)\to q))$	$\{0,5,7,9,12,13,14\}$

151 $((\neg\neg p\to((p\to q)\to q))\to(\neg\neg q\to q))$ $\{0,3,4,5,9,11,12,13,14\}$
152 $((\neg\neg p\to((p\to q)\to q))\to((\neg q\to p)\to q))$ $\{0,5,9,12,13,14\}$
153 $((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p))$ $\{5,6,7,10,11,12,14\}$
154 $((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to p))$ $\{2,3,4,5,10,11,14\}$
155 $((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to q))$ $\{3,4,5,9,11,12,13,14\}$
156 $(((\neg p\to q)\to((p\to q)\to q))\to q)$ $\{5,6,8,9,12,13,14\}$
157 $((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))$ $\{0,1,2,3,4,5,7,8,9,10,11,12,13,14\}$
158 $((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))$ $\{0,1,2,3,4,7,8,9,10,11,12,13,14\}$
159 $(((\neg p\to q)\to((p\to q)\to q))\to(\neg\neg q\to q))$ $\{0,3,4,5,6,8,9,11,12,13,14\}$
160 $(((\neg p\to q)\to((p\to q)\to q))\to((\neg q\to p)\to q))$ $\{0,5,6,8,9,12,13,14\}$
161 $(((\neg p\to q)\to((p\to q)\to q))\to((\neg q\to\neg p)\to q))$ $\{3,4,5,6,8,9,11,12,13,14\}$
162 $((\neg\neg p\to(q\to p))\to(\neg\neg q\to((p\to q)\to q)))$ $\{0,1,2,3,4,8,9,10,11,12,13,14\}$
163 $((\neg\neg q\to(p\to q))\to(((q\to p)\to p)\to q))$ $\{0,1,5,10,12,13,14\}$
164 $((\neg\neg p\to((q\to p)\to p))\to q)$ $\{5,12,13,14\}$
165 $((\neg\neg p\to((q\to p)\to p))\to((p\to q)\to q))$ $\{1,2,3,4,5,10,11,12,13,14\}$
166 $((\neg\neg q\to(p\to q))\to(\neg\neg p\to((q\to p)\to p)))$ $\{0,1,2,6,7,8,9,10,11,12,13,14\}$
167 $((\neg\neg p\to((q\to p)\to p))\to((\neg p\to\neg q)\to q))$ $\{5,6,7,8,12,13,14\}$
168 $((\neg\neg p\to((q\to p)\to p))\to((\neg q\to\neg p)\to q))$ $\{3,4,5,11,12,13,14\}$
169 $((\neg\neg p\to((q\to p)\to p))\to((\neg p\to\neg q)\to((p\to q)\to q)))$ $\{1,2,3,4,5,6,7,8,10,11,12,13,14\}$
170 $((\neg\neg q\to((q\to p)\to p))\to p)$ $\{1,5,10,11,14\}$
171 $((\neg\neg q\to((q\to p)\to p))\to(\neg\neg p\to p))$ $\{0,1,5,6,7,10,11,12,14\}$
172 $((\neg\neg q\to((q\to p)\to p))\to((\neg p\to q)\to p))$ $\{0,1,5,10,11,14\}$
173 $((\neg\neg q\to((q\to p)\to p))\to((\neg q\to p)\to p))$ $\{0,1,3,5,10,11,14\}$
174 $((\neg\neg q\to((q\to p)\to p))\to((\neg p\to\neg q)\to p))$ $\{1,5,6,7,10,11,12,14\}$
175 $(((\neg q\to p)\to((q\to p)\to p))\to p)$ $\{1,2,4,5,10,11,14\}$
176 $(((\neg q\to p)\to((q\to p)\to p))\to(\neg\neg p\to p))$ $\{0,1,2,4,5,6,7,10,11,12,14\}$
177 $(((\neg q\to p)\to((q\to p)\to p))\to((\neg p\to q)\to p))$ $\{0,1,2,4,5,10,11,14\}$
178 $((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))$ $\{0,1,2,3,5,6,7,8,9,10,11,12,13,14\}$
179 $((\neg\neg q\to(p\to q))\to((\neg q\to p)\to((q\to p)\to p)))$ $\{0,1,2,3,6,7,8,9,10,11,12,13,14\}$
180 $(((\neg q\to p)\to((q\to p)\to p))\to((\neg p\to\neg q)\to p))$ $\{1,2,4,5,6,7,10,11,12,14\}$
181 $((\neg\neg q\to(p\to q))\to(\neg\neg p\to((q\to p)\to q)))$ $\{0,1,6,7,8,9,10,12,13,14\}$
182 $((\neg\neg q\to((q\to p)\to q))\to q)$ $\{1,5,10,12,13,14\}$
183 $((\neg\neg q\to((q\to p)\to q))\to(\neg\neg p\to q))$ $\{0,1,5,6,7,8,10,12,13,14\}$
184 $((\neg\neg q\to((q\to p)\to q))\to((\neg p\to q)\to q))$ $\{0,1,5,7,10,12,13,14\}$
185 $((\neg\neg q\to((q\to p)\to q))\to((\neg p\to\neg q)\to q))$ $\{1,5,6,7,8,10,12,13,14\}$
186 $((\neg(q\to\neg p)\to p)\to p)$ $\{1,5,9,10,11,13,14\}$
187 $((\neg(q\to\neg p)\to p)\to((\neg p\to q)\to p))$ $\{0,1,5,9,10,11,13,14\}$
188 $((\neg(q\to\neg p)\to p)\to((\neg q\to p)\to p))$ $\{0,1,3,5,9,10,11,13,14\}$
189 $((\neg(q\to\neg p)\to q)\to q)$ $\{1,5,9,10,12,13,14\}$
190 $((\neg(q\to\neg p)\to q)\to((\neg p\to q)\to q))$ $\{0,1,5,7,9,10,12,13,14\}$
191 $((\neg(q\to\neg p)\to q)\to((\neg q\to p)\to q))$ $\{0,1,5,9,10,12,13,14\}$
192 $((\neg\neg q\to q)\to(((q\to\neg p)\to p)\to p))$ $\{0,1,2,3,5,6,7,10,11,12,14\}$
193 $((\neg\neg q\to(p\to q))\to(((q\to\neg p)\to p)\to p))$ $\{0,1,2,3,6,7,10,11,12,14\}$
194 $((\neg\neg q\to((q\to p)\to p))\to(((q\to\neg p)\to p)\to p))$ $\{0,1,3,5,6,7,10,11,12,14\}$
195 $((\neg\neg p\to p)\to(((q\to\neg p)\to q)\to q))$ $\{0,3,4,5,7,8,9,11,12,13,14\}$
196 $((\neg\neg p\to(q\to p))\to(((q\to\neg p)\to q)\to q))$ $\{0,3,4,7,8,9,11,12,13,14\}$
197 $((\neg\neg p\to((p\to q)\to q))\to(((q\to\neg p)\to q)\to q))$ $\{0,3,4,5,7,9,11,12,13,14\}$
198 $((((q\to p)\to p)\to(\neg\neg p\to q))\to q)$ $\{1,2,3,4,9,10,11,12,13,14\}$
199 $((((q\to p)\to p)\to(\neg\neg p\to q))\to((\neg p\to q)\to q))$ $\{0,1,2,3,4,7,9,10,11,12,13,14\}$
200 $((((q\to p)\to p)\to(\neg\neg p\to q))\to(\neg\neg q\to q))$ $\{0,1,2,3,4,9,10,11,12,13,14\}$
201 $((((q\to p)\to q)\to(\neg\neg p\to q))\to q)$ $\{9,12,13,14\}$
202 $((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg p\to q)\to q))$ $\{0,7,9,12,13,14\}$
203 $((((q\to p)\to q)\to(\neg\neg p\to q))\to(\neg\neg q\to q))$ $\{0,3,4,9,11,12,13,14\}$
204 $((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to p)\to q))$ $\{0,9,12,13,14\}$
205 $((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to\neg p)\to q))$ $\{3,4,9,11,12,13,14\}$
206 $((((q\to p)\to q)\to(\neg\neg p\to q))\to(((q\to\neg p)\to q)\to q))$ $\{0,3,4,7,9,11,12,13,14\}$
207 $((((q\to p)\to p)\to((\neg p\to q)\to q))\to q)$ $\{1,2,3,4,6,8,9,10,11,12,13,14\}$
208 $((((q\to p)\to p)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$ $\{0,1,2,3,4,6,8,9,10,11,12,13,14\}$
209 $((((q\to p)\to q)\to((\neg p\to q)\to q))\to q)$ $\{6,8,9,12,13,14\}$
210 $((((q\to p)\to q)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$ $\{0,3,4,6,8,9,11,12,13,14\}$
211, $((((q\to p)\to q)\to((\neg p\to q)\to q))\to((\neg q\to p)\to q))$ $\{0,6,8,9,12,13,14\}$
212, $((((q\to p)\to q)\to((\neg p\to q)\to q))\to((\neg q\to\neg p)\to q))$ $\{3,4,6,8,9,11,12,13,14\}$
213, $((((p\to q)\to p)\to(\neg\neg q\to p))\to p)$ $\{1,10,11,14\}$
214 $((((p\to q)\to p)\to(\neg\neg q\to p))\to((q\to p)\to p))$ $\{1,6,7,8,9,10,11,12,13,14\}$
215 $((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p))$ $\{0,1,6,7,10,11,12,14\}$
216 $((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to q)\to p))$ $\{0,1,10,11,14\}$
217 $((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg q\to p)\to p))$ $\{0,1,3,10,11,14\}$
218 $((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to\neg q)\to p))$ $\{1,6,7,10,11,12,14\}$

219	$((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,1,6,7,8,9,10,11,12,13,14\}$
220	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg q\to p)\to((q\to p)\to p)))$	$\{0,1,3,6,7,8,9,10,11,12,13,14\}$
221	$((((p\to q)\to p)\to(\neg\neg q\to p))\to(((q\to\neg p)\to p)\to p))$	$\{0,1,3,6,7,10,11,12,14\}$
222	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to p)$	$\{1,2,4,10,11,14\}$
223	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to((q\to p)\to p))$	$\{1,2,4,6,7,8,9,10,11,12,13,14\}$
224	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,1,2,4,6,7,10,11,12,14\}$
225	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to((\neg p\to q)\to p))$	$\{0,1,2,4,10,11,14\}$
226	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{1,2,4,6,7,10,11,12,14\}$
227	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,1,2,4,6,7,8,9,10,11,12,13,14\}$
228	$((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to((\neg p\to\neg q)\to p)))$	$\{2,3,4,5,6,7,10,11,12,14\}$
229	$((\neg\neg p\to((q\to p)\to p))\to((\neg q\to\neg p)\to((\neg p\to\neg q)\to q)))$	$\{3,4,5,6,7,8,11,12,13,14\}$
230	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to p)$	$\{1,9,10,11,13,14\}$
231	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to((\neg p\to q)\to p))$	$\{0,1,9,10,11,13,14\}$
232	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to((\neg q\to p)\to p))$	$\{0,1,3,9,10,11,13,14\}$
233	$((((q\to\neg p)\to p)\to(((p\to q)\to q)\to p))\to p)$	$\{1,2,4,9,10,11,13,14\}$
234	$((((q\to\neg p)\to p)\to(((p\to q)\to q)\to p))\to((\neg p\to q)\to p))$	$\{0,1,2,4,9,10,11,13,14\}$
235	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to q)$	$\{1,9,10,12,13,14\}$
236	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to((\neg p\to q)\to q))$	$\{0,1,7,9,10,12,13,14\}$
237	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to((\neg q\to p)\to q))$	$\{0,1,9,10,12,13,14\}$
238	$((((q\to\neg p)\to q)\to(((q\to p)\to p)\to q))\to q)$	$\{1,6,8,9,10,12,13,14\}$
239	$((((q\to\neg p)\to q)\to(((q\to p)\to p)\to q))\to((\neg q\to p)\to q))$	$\{0,1,6,8,9,10,12,13,14\}$
240	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to p)\to q))$	$\{1,5,8,9,10,12,13,14\}$
241	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to p)\to q))$	$\{5,7,8,9,12,13,14\}$
242	$(((\neg q\to\neg p)\to((q\to p)\to q))\to((\neg\neg p\to p)\to q))$	$\{0,1,5,8,9,10,12,13,14\}$
243	$(((\neg q\to p)\to((q\to\neg p)\to q))\to((\neg\neg p\to p)\to q))$	$\{3,4,5,7,8,9,11,12,13,14\}$
244	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to p)\to((p\to q)\to q)))$	$\{1,2,3,4,5,7,8,9,10,11,12,13,14\}$
245	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to q)\to q))$	$\{1,2,3,4,5,7,9,10,11,12,13,14\}$
246	$((\neg\neg q\to q)\to((\neg\neg p\to(q\to p))\to p))$	$\{1,2,5,9,10,11,13,14\}$
247	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to p))$	$\{1,2,9,10,11,13,14\}$
248	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg p\to(q\to p))\to p))$	$\{3,9,10,11,13,14\}$
249	$(((\neg q\to\neg p)\to q)\to((\neg\neg p\to(q\to p))\to p))$	$\{0,1,2,5,9,10,11,13,14\}$
250	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to q))$	$\{1,8,9,10,12,13,14\}$
251	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to(q\to p))\to q))$	$\{7,8,9,12,13,14\}$
252	$(((\neg q\to p)\to((q\to\neg p)\to q))\to((\neg\neg p\to(q\to p))\to q))$	$\{3,4,7,8,9,11,12,13,14\}$
253	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to(q\to p))\to((p\to q)\to q)))$	$\{1,2,3,4,7,8,9,10,11,12,13,14\}$
254	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p)))$	$\{0,1,2,9,10,11,13,14\}$
255	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to p)\to((\neg p\to q)\to q)))$	$\{0,1,5,7,8,9,10,12,13,14\}$
256	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg p\to q)\to q)))$	$\{0,1,7,8,9,10,12,13,14\}$
257	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to p)\to p))$	$\{1,3,5,6,7,8,9,10,11,12,13,14\}$
258	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to q)\to p))$	$\{1,2,3,5,10,11,14\}$
259	$(((\neg p\to q)\to((q\to\neg p)\to p))\to((\neg\neg q\to q)\to p))$	$\{1,2,3,5,6,7,10,11,12,14\}$
260	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to q)\to((q\to p)\to p)))$	$\{1,2,3,5,6,7,8,9,10,11,12,13,14\}$
261	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to(p\to q))\to p))$	$\{1,2,3,10,11,14\}$
262	$(((\neg p\to q)\to((q\to\neg p)\to p))\to((\neg\neg q\to(p\to q))\to p))$	$\{1,2,3,6,7,10,11,12,14\}$
263	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to q))$	$\{1,7,10,12,13,14\}$
264	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to(p\to q))\to((q\to p)\to p)))$	$\{1,2,3,6,7,8,9,10,11,12,13,14\}$
265	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{5,7,9,12,13,14\}$
266	$((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{5,7,12,13,14\}$
267	$((\neg\neg p\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,2,3,4,5,7,10,11,12,13,14\}$
268	$((\neg\neg q\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,5,7,10,12,13,14\}$
269	$((\neg(q\to\neg p)\to q)\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,5,7,9,10,12,13,14\}$
270	$((((q\to p)\to p)\to(\neg\neg p\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,2,3,4,7,9,10,11,12,13,14\}$
271	$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{7,9,12,13,14\}$
272	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,7,9,10,12,13,14\}$
273	$((\neg\neg q\to q)\to((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p)))$	$\{0,1,2,3,5,9,10,11,13,14\}$
274	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p)))$	$\{0,1,2,3,9,10,11,13,14\}$
275	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg q\to p)\to q)))$	$\{0,1,8,9,10,12,13,14\}$
276	$((\neg\neg p\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{3,5,9,10,11,13,14\}$
277	$(((\neg p\to q)\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{3,5,6,7,8,9,10,11,12,13,14\}$
278	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{3,5,10,11,14\}$
279	$((\neg\neg q\to((q\to p)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,5,10,11,14\}$
280	$((\neg(q\to\neg p)\to p)\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,5,9,10,11,13,14\}$
281	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,10,11,14\}$
282	$((((p\to q)\to q)\to(\neg\neg q\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,6,7,8,9,10,11,12,13,14\}$
283	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,9,10,11,13,14\}$
284	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p)))$	$\{3,5,6,7,10,11,12,14\}$
285	$((\neg\neg q\to((q\to p)\to p))\to((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p)))$	$\{1,3,5,6,7,10,11,12,14\}$
286	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg\neg p\to(\neg q\to p))\to$	

	$((\neg p\to\neg q)\to p)))$	$\{1,3,6,7,10,11,12,14\}$
287	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q)))$	$\{3,4,5,7,9,11,12,13,14\}$
288	$((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q)))$	$\{3,4,5,7,11,12,13,14\}$
289	$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg\neg q\to(\neg p\to q))\to$ $((\neg q\to\neg p)\to q)))$	$\{3,4,7,9,11,12,13,14\}$
290	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to((\neg\neg p\to p)\to q)))$	$\{1,5,7,8,9,10,12,13,14\}$
291	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to q)\to((\neg\neg p\to(q\to p))\to p)))$	$\{1,2,3,5,9,10,11,13,14\}$
292	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to p)))$	$\{1,2,3,9,10,11,13,14\}$
293	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to q)))$	$\{1,7,8,9,10,12,13,14\}$
294	$(((\neg\neg p\to p)\to q)\to q)$	$\{0,1,2,3,4,6,7,10,11,12,13,14\}$
295	$(((\neg\neg p\to p)\to q)\to(((q\to p)\to p)\to q))$	$\{0,1,2,3,4,5,6,7,10,11,12,13,14\}$
296	$(((\neg\neg p\to p)\to((p\to q)\to q))\to q)$	$\{0,6,7,12,13,14\}$
297	$(((\neg\neg p\to p)\to((p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,3,4,6,7,11,12,13,14\}$
298	$(((\neg\neg p\to p)\to((p\to q)\to q))\to(((q\to p)\to p)\to q))$	$\{0,5,6,7,12,13,14\}$
299	$(((\neg\neg p\to p)\to((p\to q)\to q))\to(((q\to p)\to p)\to(\neg\neg q\to q)))$	$\{0,3,4,5,6,7,11,12,13,14\}$
300	$(((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))$	$\{0,1,2,3,4,5,6,7,9,10,11,12,13,14\}$
301	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to q)$	$\{1,2,3,4,6,10,11,12,13,14\}$
302	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,1,2,3,4,6,10,11,12,13,14\}$
303	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to q))$	$\{0,1,2,3,4,5,6,10,11,12,13,14\}$
304	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to((\neg\neg p\to q)\to q))$	$\{1,2,3,4,5,6,9,10,11,12,13,14\}$
305	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to q)$	$\{6,8,12,13,14\}$
306	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((p\to q)\to q))$	$\{1,2,3,4,6,8,10,11,12,13,14\}$
307	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,3,4,6,8,11,12,13,14\}$
308	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg q\to p)\to q))$	$\{0,6,8,12,13,14\}$
309	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,6,8,11,12,13,14\}$
310	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(\neg\neg q\to((p\to q)\to q)))$	$\{0,1,2,3,4,6,8,10,11,12,13,14\}$
311,	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to q))$	$\{0,5,6,8,12,13,14\}$
312,	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to((p\to q)\to q)))$	$\{0,1,2,3,4,5,6,8,10,11,12,13,14\}$
313,	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to(\neg\neg q\to q)))$	$\{0,3,4,5,6,8,11,12,13,14\}$
314	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to q))\to q)$	$\{1,2,3,4,5,6,10,11,12,13,14\}$
315	$(((\neg\neg q\to q)\to p)\to p)$	$\{0,3,4,6,7,8,9,10,11,12,13,14\}$
316	$(((\neg\neg q\to q)\to p)\to((p\to q)\to p)\to p))$	$\{0,3,4,5,6,7,8,9,10,11,12,13,14\}$
317	$(((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))$	$\{0,1,3,4,5,6,7,8,9,10,11,12,13,14\}$
318	$(((\neg\neg q\to q)\to((q\to p)\to p))\to p)$	$\{0,3,4,10,11,14\}$
319	$(((\neg\neg q\to q)\to((q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,3,4,6,7,10,11,12,14\}$
320	$(((\neg\neg q\to q)\to((q\to p)\to p))\to(((p\to q)\to q)\to p))$	$\{0,3,4,5,10,11,14\}$
321	$(((\neg\neg q\to q)\to((q\to p)\to p))\to(((p\to q)\to q)\to(\neg\neg p\to p)))$	$\{0,3,4,5,6,7,10,11,12,14\}$
322	$(((\neg\neg q\to q)\to((q\to p)\to p))\to((\neg\neg p\to(q\to p))\to p))$	$\{0,3,4,9,10,11,13,14\}$
323	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to q)$	$\{1,6,7,10,12,13,14\}$
324	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((p\to q)\to q))$	$\{1,2,3,4,6,7,10,11,12,13,14\}$
325	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((\neg p\to q)\to q))$	$\{0,1,6,7,10,12,13,14\}$
326	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to(((q\to p)\to p)\to q))$	$\{0,1,5,6,7,10,12,13,14\}$
327	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((\neg\neg p\to q)\to q))$	$\{1,2,3,4,5,6,7,9,10,11,12,13,14\}$
328	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to p)$	$\{3,4,9,10,11,13,14\}$
329	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((q\to p)\to p))$	$\{3,4,6,7,8,9,10,11,12,13,14\}$
330	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to(((p\to q)\to q)\to p))$	$\{0,3,4,5,9,10,11,13,14\}$
331	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((\neg\neg q\to p)\to p))$	$\{1,3,4,5,6,7,8,9,10,11,12,13,14\}$
332	$(((\neg\neg p\to p)\to((p\to q)\to q))\to((\neg\neg p\to q)\to(\neg\neg q\to q)))$	$\{0,1,2,3,4,5,6,9,10,11,12,13,14\}$
333	$(((\neg\neg p\to(q\to p))\to(\neg\neg q\to q))\to q)$	$\{1,5,6,7,10,12,13,14\}$
334	$(((\neg\neg p\to(q\to p))\to(\neg\neg q\to q))\to((p\to q)\to q))$	$\{1,2,3,4,5,6,7,10,11,12,13,14\}$
335	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to q))$	$\{1,6,8,10,12,13,14\}$
336	$(((\neg\neg q\to(p\to q))\to(\neg\neg p\to p))\to p)$	$\{3,4,5,9,10,11,13,14\}$
337	$(((\neg\neg q\to(p\to q))\to(\neg\neg p\to p))\to((q\to p)\to p))$	$\{3,4,5,6,7,8,9,10,11,12,13,14\}$
338	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to p)$	$\{2,4,10,11,14\}$
339	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((q\to p)\to p))$	$\{2,4,6,7,8,9,10,11,12,13,14\}$
340	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,2,4,6,7,10,11,12,14\}$
341	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg p\to q)\to p))$	$\{0,2,4,10,11,14\}$
342	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{2,4,6,7,10,11,12,14\}$
343	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(((p\to q)\to p)\to p))$	$\{0,2,4,5,6,7,8,9,10,11,12,13,14\}$
344	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(((p\to q)\to q)\to p))$	$\{0,2,4,5,10,11,14\}$
345	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,2,4,6,7,8,9,10,11,12,13,14\}$
346	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(((p\to q)\to q)\to(\neg\neg p\to p)))$	$\{0,2,4,5,6,7,10,11,12,14\}$
347	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to(q\to p))\to p))$	$\{2,4,9,10,11,13,14\}$
348	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p)))$	$\{0,2,4,9,10,11,13,14\}$
349	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to p)$	$\{4,6,7,8,9,10,11,12,13,14\}$
350	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,4,6,7,8,9,10,11,12,13,14\}$
351	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to(((p\to q)\to p)\to p))$	$\{0,4,5,6,7,8,9,10,11,12,13,14\}$
352	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to((\neg\neg q\to p)\to p))$	$\{1,4,5,6,7,8,9,10,11,12,13,14\}$

353 $(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to((\neg\neg q\to p)\to(\neg\neg p\to p)))$ $\{0,1,4,5,6,7,8,9,10,11,12,13,14\}$
354 $(((\neg\neg q\to(p\to q))\to((\neg q\to p)\to p))\to p)$ $\{4,5,6,7,8,9,10,11,12,13,14\}$
355 $(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q)))$ $\{0,1,6,8,10,12,13,14\}$
356 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to q)$ $\{6,7,12,13,14\}$
357 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((\neg q\to\neg p)\to q))$ $\{3,4,6,7,11,12,13,14\}$
358 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to p)$ $\{3,4,10,11,14\}$
359 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg p\to\neg q)\to p))$ $\{3,4,6,7,10,11,12,14\}$
360 $(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to p))\to p))$ $\{2,4,5,9,10,11,13,14\}$
361 $(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to p))\to((q\to p)\to p)))$ $\{2,4,5,6,7,8,9,10,11,12,13,14\}$
362 $(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to p))\to((\neg p\to q)\to p)))$ $\{0,2,4,5,9,10,11,13,14\}$
363 $(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to p))$ $\{2,4,5,10,11,14\}$
364 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to p))$ $\{3,4,5,10,11,14\}$
365 $(((\neg\neg p\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$ $\{0,5,6,7,9,12,13,14\}$
366 $(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$ $\{1,5,6,7,9,10,12,13,14\}$
367 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$ $\{5,6,7,9,12,13,14\}$
368 $(((\neg\neg p\to p)\to((\neg q\to\neg p)\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$ $\{0,1,5,6,7,9,10,12,13,14\}$
369 $(((\neg\neg p\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to(\neg\neg q\to q)))$ $\{0,3,4,5,6,7,9,11,12,13,14\}$
370 $(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p)))$ $\{2,4,5,6,7,10,11,12,14\}$
371 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p)))$ $\{3,4,5,6,7,10,11,12,14\}$
372 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,5,6,7,9,11,12,13,14\}$
373 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to q)$ $\{6,12,13,14\}$
374 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to(\neg\neg q\to q))$ $\{0,3,4,6,11,12,13,14\}$
375 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg q\to p)\to q))$ $\{0,6,12,13,14\}$
376 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg q\to\neg p)\to q))$ $\{3,4,6,11,12,13,14\}$
377 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to(((q\to p)\to p)\to q))$ $\{0,5,6,12,13,14\}$
378 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to(((q\to p)\to p)\to(\neg\neg q\to q)))$ $\{0,3,4,5,6,11,12,13,14\}$
379 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(p\to q))\to q))$ $\{1,6,10,12,13,14\}$
380 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q)))$ $\{0,1,6,10,12,13,14\}$
381 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to((p\to q)\to q))\to q))$ $\{5,6,9,12,13,14\}$
382 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to((p\to q)\to q))\to(\neg\neg q\to q)))$ $\{0,3,4,5,6,9,11,12,13,14\}$
383 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to((p\to q)\to q))\to((\neg q\to p)\to q)))$ $\{0,5,6,9,12,13,14\}$
384 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,5,6,9,11,12,13,14\}$
385 $(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to q)$ $\{5,6,12,13,14\}$
386 $(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg q\to\neg p)\to q))$ $\{3,4,5,6,11,12,13,14\}$
387 $(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(p\to q))\to q))$ $\{1,5,6,10,12,13,14\}$
388 $(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(\neg p\to q))\to q))$ $\{5,6,7,12,13,14\}$
389 $(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q)))$ $\{0,1,5,6,10,12,13,14\}$
390 $(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,5,6,7,11,12,13,14\}$
391 $(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to(((q\to p)\to p)\to q)))$ $\{0,1,5,6,8,10,12,13,14\}$
392 $(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg p\to((q\to p)\to p))\to q))$ $\{5,6,8,12,13,14\}$
393 $(((\neg\neg p\to((p\to q)\to q))\to((\neg p\to q)\to q))\to((\neg\neg p\to((q\to p)\to p))\to q))$ $\{1,2,3,4,5,6,8,10,11,12,13,14\}$
394 $(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg p\to((q\to p)\to p))\to((\neg q\to\neg p)\to q)))$ $\{3,4,5,6,8,11,12,13,14\}$
395 $(((\neg\neg q\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$ $\{0,1,3,4,5,10,11,14\}$
396 $(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$ $\{1,3,4,5,9,10,11,13,14\}$
397 $(((\neg\neg q\to q)\to((\neg p\to\neg q)\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$ $\{0,1,3,4,5,9,10,11,13,14\}$
398 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$ $\{1,3,4,5,10,11,14\}$
399 $(((\neg\neg q\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to(\neg\neg p\to p)))$ $\{0,1,3,4,5,6,7,10,11,12,14\}$
400 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to((\neg p\to\neg q)\to p)))$ $\{1,3,4,5,6,7,10,11,12,14\}$
401 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to p)$ $\{4,10,11,14\}$

402 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to(\neg\neg p\to p))$ $\{0,4,6,7,10,11,12,14\}$
403 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg p\to q)\to p))$ $\{0,4,10,11,14\}$
404 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg p\to\neg q)\to p))$ $\{4,6,7,10,11,12,14\}$
405 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to(((p\to q)\to q)\to p))$ $\{0,4,5,10,11,14\}$
406 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to(((p\to q)\to q)\to$
$(\neg\neg p\to p)))$ $\{0,4,5,6,7,10,11,12,14\}$
407 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg\neg p\to(q\to p))\to p))$ $\{4,9,10,11,13,14\}$
408 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg\neg p\to(q\to p))\to$
$((\neg p\to q)\to p)))$ $\{0,4,9,10,11,13,14\}$
409 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg\neg p\to$
$((p\to q)\to p))\to p))$ $\{4,5,9,10,11,13,14\}$
410 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg\neg p\to((p\to q)\to p))\to((\neg p\to q)\to p)))$ $\{0,4,5,9,10,11,13,14\}$
411, $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg\neg p\to((p\to q)\to q))\to p))$ $\{4,5,10,11,14\}$
412, $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p)))$ $\{4,5,6,7,10,11,12,14\}$
413, $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg\neg q\to((q\to p)\to p))\to p))$ $\{1,4,5,10,11,14\}$
414 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg\neg q\to((q\to p)\to p))\to(\neg\neg p\to p)))$ $\{0,1,4,5,6,7,10,11,12,14\}$
415 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg\neg q\to((q\to p)\to p))\to((\neg p\to q)\to p)))$ $\{0,1,4,5,10,11,14\}$
416 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg\neg q\to((q\to p)\to p))\to((\neg p\to\neg q)\to p)))$ $\{1,4,5,6,7,10,11,12,14\}$
417 $(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to$
$((q\to p)\to q))\to q))$ $\{1,5,6,8,10,12,13,14\}$
418 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg(q\to\neg p)\to p)\to p))$ $\{1,4,5,9,10,11,13,14\}$
419 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((\neg(q\to\neg p)\to p)\to((\neg p\to q)\to p)))$ $\{0,1,4,5,9,10,11,13,14\}$
420 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to$
$((\neg(q\to\neg p)\to q)\to q))$ $\{1,5,6,9,10,12,13,14\}$
421 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to$
$((\neg(q\to\neg p)\to q)\to((\neg q\to p)\to q)))$ $\{0,1,5,6,9,10,12,13,14\}$
422 $(((\neg\neg p\to p)\to q)\to((((q\to p)\to p)\to(\neg\neg p\to q))\to q))$ $\{0,1,2,3,4,6,7,9,10,11,12,13,14\}$
423 $(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to((((q\to p)\to p)\to(\neg\neg p\to q))\to q))$ $\{1,2,3,4,6,9,10,11,12,13,14\}$
424 $(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((((q\to p)\to p)\to(\neg\neg p\to q))\to q))$ $\{1,2,3,4,6,7,9,10,11,12,13,14\}$
425 $(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to((((q\to p)\to p)\to$
$(\neg\neg p\to q))\to(\neg\neg q\to q)))$ $\{0,1,2,3,4,6,9,10,11,12,13,14\}$
426 $(((\neg\neg p\to p)\to((p\to q)\to q))\to((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{0,6,7,9,12,13,14\}$
427 $(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{1,6,7,9,10,12,13,14\}$
428 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((((q\to p)\to q)\to$
$(\neg\neg p\to q))\to q))$ $\{6,7,9,12,13,14\}$
429 $(((\neg\neg p\to p)\to((\neg q\to\neg p)\to q))\to((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{0,1,6,7,9,10,12,13,14\}$
430 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to$
$((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{6,9,12,13,14\}$
431 $(((\neg\neg p\to p)\to(((q\to\neg p)\to q)\to q))\to((((q\to p)\to q)\to$
$(\neg\neg p\to q))\to q))$ $\{1,6,9,10,12,13,14\}$
432 $(((\neg\neg p\to p)\to((p\to q)\to q))\to((((q\to p)\to q)\to$
$(\neg\neg p\to q))\to(\neg\neg q\to q)))$ $\{0,3,4,6,7,9,11,12,13,14\}$
433 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to$
$((((q\to p)\to q)\to(\neg\neg p\to q))\to(\neg\neg q\to q)))$ $\{0,3,4,6,9,11,12,13,14\}$
434 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to$
$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to p)\to q)))$ $\{0,6,9,12,13,14\}$
435 $(((\neg\neg p\to p)\to(((q\to\neg p)\to q)\to q))\to$
$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to p)\to q)))$ $\{0,1,6,9,10,12,13,14\}$
436 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to$
$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,6,7,9,11,12,13,14\}$
437 $(((\neg\neg p\to p)\to((\neg p\to q)\to(p\to q)\to q)))\to$
$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,6,9,11,12,13,14\}$
438 $(((\neg\neg q\to q)\to p)\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{0,1,3,4,6,7,8,9,10,11,12,13,14\}$
439 $(((\neg\neg q\to q)\to((q\to p)\to p))\to(((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{0,1,3,4,10,11,14\}$
440 $(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,3,4,9,10,11,13,14\}$
441 $(((\neg\neg q\to q)\to((\neg p\to q)\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,3,4,6,7,8,9,10,11,12,13,14\}$
442 $(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,4,6,7,8,9,10,11,12,13,14\}$
443 $(((\neg\neg q\to q)\to((\neg p\to\neg q)\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{0,1,3,4,9,10,11,13,14\}$
444 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((((p\to q)\to p)\to$

$(\neg\neg q\to p))\to p))$ $\{1,3,4,10,11,14\}$

445 $(((\neg\neg q\to q)\to((\neg q\to p)\to$
$((q\to p)\to p)))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,4,10,11,14\}$

446 $(((\neg\neg q\to q)\to(((q\to\neg p)\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,4,9,10,11,13,14\}$

447 $(((\neg\neg q\to q)\to((q\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p)))$ $\{0,1,3,4,6,7,10,11,12,14\}$

448 $(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p)))$ $\{0,1,4,6,7,8,9,10,11,12,13,14\}$

449 $(((\neg\neg q\to q)\to((\neg q\to p)\to$
$((q\to p)\to p)))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p)))$ $\{0,1,4,6,7,10,11,12,14\}$

450 $(((\neg\neg q\to q)\to((\neg q\to p)\to$
$((q\to p)\to p)))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to q)\to p)))$ $\{0,1,4,10,11,14\}$

451 $(((\neg\neg q\to q)\to(((q\to\neg p)\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to q)\to p)))$ $\{0,1,4,9,10,11,13,14\}$

452 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to\neg q)\to p)))$ $\{1,3,4,6,7,10,11,12,14\}$

453 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to\neg q)\to p)))$ $\{1,4,6,7,10,11,12,14\}$

454 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to q)$ $\{8,12,13,14\}$

455 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((p\to q)\to q))$ $\{1,2,3,4,8,10,11,12,13,14\}$

456 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg p\to q)\to q))$ $\{0,7,8,12,13,14\}$

457 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(\neg\neg q\to q))$ $\{0,3,4,8,11,12,13,14\}$

458 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg q\to p)\to q))$ $\{0,8,12,13,14\}$

459 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg q\to\neg p)\to q))$ $\{3,4,8,11,12,13,14\}$

460 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg p\to q)\to((p\to q)\to q)))$ $\{0,1,2,3,4,7,8,10,11,12,13,14\}$

461 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(\neg\neg q\to((p\to q)\to q)))$ $\{0,1,2,3,4,8,10,11,12,13,14\}$

462 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(((q\to p)\to p)\to q))$ $\{0,5,8,12,13,14\}$

463 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to p)\to((p\to q)\to q)))$ $\{0,1,2,3,4,5,8,10,11,12,13,14\}$

464 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(((q\to\neg p)\to q)\to q))$ $\{0,3,4,7,8,11,12,13,14\}$

465 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to p)\to((\neg p\to q)\to q)))$ $\{0,5,7,8,12,13,14\}$

466 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to q)\to((\neg p\to q)\to q)))$ $\{0,1,2,3,4,5,7,8,10,11,12,13,14\}$

467 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to p)\to(\neg\neg q\to q)))$ $\{0,3,4,5,8,11,12,13,14\}$

468 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to\neg p)\to q)\to(((q\to p)\to p)\to q)))$ $\{0,3,4,5,7,8,11,12,13,14\}$

469 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to q))$ $\{1,8,10,12,13,14\}$

470 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(p\to q))\to((\neg p\to q)\to q)))$ $\{0,1,7,8,10,12,13,14\}$

471 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$ $\{7,8,12,13,14\}$

472 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(\neg p\to q))\to((p\to q)\to q)))$ $\{1,2,3,4,7,8,10,11,12,13,14\}$

473 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q)))$ $\{0,1,8,10,12,13,14\}$

474 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,7,8,11,12,13,14\}$

475 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(p\to q))\to(((q\to p)\to p)\to q)))$ $\{0,1,5,8,10,12,13,14\}$

476 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to q))$ $\{5,8,12,13,14\}$

477 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to((p\to q)\to q)))$ $\{1,2,3,4,5,8,10,11,12,13,14\}$

478 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to((\neg q\to\neg p)\to q)))$ $\{3,4,5,8,11,12,13,14\}$

479 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to((q\to p)\to q))\to q))$ $\{1,5,8,10,12,13,14\}$

480 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to((q\to p)\to q))\to((\neg p\to q)\to q)))$ $\{0,1,5,7,8,10,12,13,14\}$

481 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to q)))$ $\{1,7,8,10,12,13,14\}$

482 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to q)))$ $\{5,7,8,12,13,14\}$

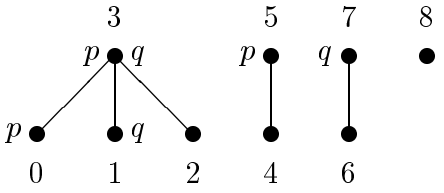
483 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q)))$ $\{1,2,3,4,5,7,8,10,11,12,13,14\}$

484 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$

	$((\neg\neg q\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q)))$	$\{1,5,7,8,10,12,13,14\}$
485	$((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$ $((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q))))$	$\{3,4,5,7,8,11,12,13,14\}$
486	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to p)$	$\{2,10,11,14\}$
487	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((q\to p)\to p))$	$\{2,6,7,8,9,10,11,12,13,14\}$
488	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,2,6,7,10,11,12,14\}$
489	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg p\to q)\to p))$	$\{0,2,10,11,14\}$
490	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg q\to p)\to p))$	$\{0,2,3,10,11,14\}$
491	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{2,6,7,10,11,12,14\}$
492	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((p\to q)\to p)\to p))$	$\{0,2,5,6,7,8,9,10,11,12,13,14\}$
493	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((p\to q)\to q)\to p))$	$\{0,2,5,10,11,14\}$
494	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,2,6,7,8,9,10,11,12,13,14\}$
495	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg q\to p)\to((q\to p)\to p)))$	$\{0,2,3,6,7,8,9,10,11,12,13,14\}$
496	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((q\to\neg p)\to p)\to p))$	$\{0,2,3,6,7,10,11,12,14\}$
497	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((p\to q)\to q)\to(\neg\neg p\to p)))$	$\{0,2,5,6,7,10,11,12,14\}$
498	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $(((p\to q)\to p)\to((\neg q\to p)\to p)))$	$\{0,2,3,5,6,7,8,9,10,11,12,13,14\}$
499	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $(((p\to q)\to q)\to((\neg q\to p)\to p)))$	$\{0,2,3,5,10,11,14\}$
500	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $(((q\to\neg p)\to p)\to(((p\to q)\to q)\to p)))$	$\{0,2,3,5,6,7,10,11,12,14\}$
501	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg\neg p\to(q\to p))\to p))$	$\{2,9,10,11,13,14\}$
502	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p)))$	$\{0,2,9,10,11,13,14\}$
503	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p)))$	$\{0,2,3,9,10,11,13,14\}$
504	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{2,3,10,11,14\}$
505	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to(\neg q\to p))\to((q\to p)\to p)))$	$\{2,3,6,7,8,9,10,11,12,13,14\}$
506	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p)))$	$\{2,3,6,7,10,11,12,14\}$
507	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to p))\to p))$	$\{2,5,9,10,11,13,14\}$
508	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to p))\to((q\to p)\to p)))$	$\{2,5,6,7,8,9,10,11,12,13,14\}$
509	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to p))\to((\neg p\to q)\to p)))$	$\{0,2,5,9,10,11,13,14\}$
510	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to p))\to((\neg q\to p)\to p)))$	$\{0,2,3,5,9,10,11,13,14\}$
511,	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to q))\to p))$	$\{2,5,10,11,14\}$
512,	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p)))$	$\{2,5,6,7,10,11,12,14\}$
513,	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to(\neg q\to p))\to((\neg\neg p\to(q\to p))\to p)))$	$\{2,3,9,10,11,13,14\}$
514	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p)))$	$\{2,3,5,9,10,11,13,14\}$
515	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $(((\neg p\to q)\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p)))$	$\{2,3,5,6,7,8,9,10,11,12,13,14\}$
516	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to p)))$	$\{2,3,5,10,11,14\}$
517	$((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$ $((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p))))$	$\{2,3,5,6,7,10,11,12,14\}$

B.2 The diagram of $\mathbf{H}_3^2$

The logic $\mathbf{H}_3$ was introduced in subsection 3.4.1. In the computation of the diagram of the fragment $\mathbf{H}_3^2$ the exact model of this fragment was used:



32. FIGURE. *The exact model of* $\mathbf{H}_3^2$.

Listed are the formulas in $\mathbf{H}_3^2$ and their valuations in this model.

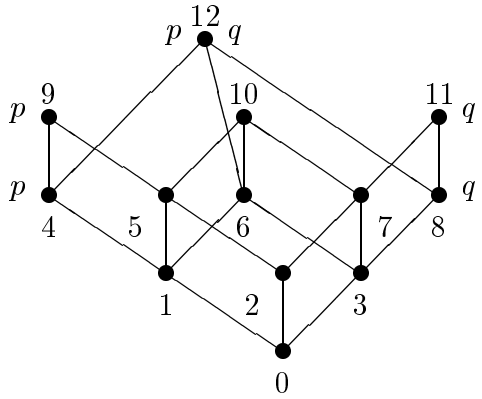
0	$(p \wedge \neg p)$	$\{\}$
1	$(p \to p)$	$\{0,1,2,3,4,5,6,7,8\}$
2	p	$\{0,3,5\}$
3	q	$\{1,3,7\}$
4	$(p \wedge q)$	$\{3\}$
5	$(p \vee q)$	$\{0,1,3,5,7\}$
6	$\neg p$	$\{6,7,8\}$
7,	$\neg q$	$\{4,5,8\}$
8	$(p \to q)$	$\{1,2,3,6,7,8\}$
9	$(q \to p)$	$\{0,2,3,4,5,8\}$
10	$\neg(p \wedge q)$	$\{4,5,6,7,8\}$
11	$\neg(p \vee q)$	$\{8\}$
12	$((p \vee q) \to (p \wedge q))$	$\{2,3,8\}$
13	$(p \vee \neg p)$	$\{0,3,5,6,7,8\}$
14	$(q \wedge \neg p)$	$\{7\}$
15	$(q \vee \neg p)$	$\{1,3,6,7,8\}$
16	$((p \wedge q) \vee \neg p)$	$\{3,6,7,8\}$
17	$((p \vee q) \vee \neg p)$	$\{0,1,3,5,6,7,8\}$
18	$(p \wedge \neg q)$	$\{5\}$
19	$(p \vee \neg q)$	$\{0,3,4,5,8\}$
20	$(q \vee \neg q)$	$\{1,3,4,5,7,8\}$
21	$((p \wedge q) \vee \neg q)$	$\{3,4,5,8\}$
22	$((p \vee q) \vee \neg q)$	$\{0,1,3,4,5,7,8\}$
23	$(p \vee (p \to q))$	$\{0,1,2,3,5,6,7,8\}$
24	$(\neg q \vee (p \to q))$	$\{1,2,3,4,5,6,7,8\}$
25	$(q \vee (q \to p))$	$\{0,1,2,3,4,5,7,8\}$
26	$(\neg p \vee (q \to p))$	$\{0,2,3,4,5,6,7,8\}$
27	$(p \vee \neg(p \wedge q))$	$\{0,3,4,5,6,7,8\}$
28	$(q \vee \neg(p \wedge q))$	$\{1,3,4,5,6,7,8\}$
29	$((p \wedge q) \vee \neg(p \wedge q))$	$\{3,4,5,6,7,8\}$
30	$((p \vee q) \wedge \neg(p \wedge q))$	$\{5,7\}$
31	$((p \vee q) \vee \neg(p \wedge q))$	$\{0,1,3,4,5,6,7,8\}$
32	$(p \vee \neg(p \vee q))$	$\{0,3,5,8\}$
33	$(q \vee \neg(p \vee q))$	$\{1,3,7,8\}$
34	$((p \wedge q) \vee \neg(p \vee q))$	$\{3,8\}$
35	$((p \vee q) \vee \neg(p \vee q))$	$\{0,1,3,5,7,8\}$
36	$(p \vee ((p \vee q) \to (p \wedge q)))$	$\{0,2,3,5,8\}$
37	$(q \vee ((p \vee q) \to (p \wedge q)))$	$\{1,2,3,7,8\}$
38	$((p \vee q) \vee ((p \vee q) \to (p \wedge q)))$	$\{0,1,2,3,5,7,8\}$
39	$(\neg p \vee ((p \vee q) \to (p \wedge q)))$	$\{2,3,6,7,8\}$
40	$(\neg q \vee ((p \vee q) \to (p \wedge q)))$	$\{2,3,4,5,8\}$
41	$(\neg(p \wedge q) \vee ((p \vee q) \to (p \wedge q)))$	$\{2,3,4,5,6,7,8\}$
42	$(q \wedge (p \vee \neg p))$	$\{3,7\}$
43	$((p \vee q) \wedge (p \vee \neg p))$	$\{0,3,5,7\}$
44	$(\neg q \wedge (p \vee \neg p))$	$\{5,8\}$
45	$(\neg(p \wedge q) \wedge (p \vee \neg p))$	$\{5,6,7,8\}$

46	$(((p \vee q) \rightarrow (p \wedge q)) \vee (p \vee \neg p))$	$\{0,2,3,5,6,7,8\}$
47	$(\neg q \vee (q \wedge \neg p))$	$\{4,5,7,8\}$
48	$((q \rightarrow p) \vee (q \wedge \neg p))$	$\{0,2,3,4,5,7,8\}$
49	$(\neg(p \vee q) \vee (q \wedge \neg p))$	$\{7,8\}$
50	$(((p \vee q) \rightarrow (p \wedge q)) \vee (q \wedge \neg p))$	$\{2,3,7,8\}$
51	$(q \vee (p \wedge \neg q))$	$\{1,3,5,7\}$
52	$((p \wedge q) \vee (p \wedge \neg q))$	$\{3,5\}$
53	$((p \rightarrow q) \vee (p \wedge \neg q))$	$\{1,2,3,5,6,7,8\}$
54	$(((p \vee q) \rightarrow (p \wedge q)) \vee (p \wedge \neg q))$	$\{2,3,5,8\}$
55	$((q \vee \neg p) \vee (p \wedge \neg q))$	$\{1,3,5,6,7,8\}$
56	$(((p \wedge q) \vee \neg p) \vee (p \wedge \neg q))$	$\{3,5,6,7,8\}$
57	$((q \wedge \neg p) \vee (p \vee \neg q))$	$\{0,3,4,5,7,8\}$
58	$(((p \vee q) \rightarrow (p \wedge q)) \vee (q \vee \neg q))$	$\{1,2,3,4,5,7,8\}$
59	$((p \vee \neg p) \wedge (q \vee \neg q))$	$\{3,5,7,8\}$
60	$(((p \wedge q) \vee \neg p) \wedge (q \vee \neg q))$	$\{3,7,8\}$
61	$(((p \vee q) \vee \neg p) \wedge (q \vee \neg q))$	$\{1,3,5,7,8\}$
62	$((p \vee \neg p) \wedge ((p \wedge q) \vee \neg q))$	$\{3,5,8\}$
63	$((q \wedge \neg p) \vee ((p \wedge q) \vee \neg q))$	$\{3,4,5,7,8\}$
64	$((p \vee \neg p) \wedge ((p \vee q) \vee \neg q))$	$\{0,3,5,7,8\}$
65	$((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q)))$	$\{3,5,7\}$
66	$(\neg(p \vee q) \vee ((p \vee q) \wedge \neg(p \wedge q)))$	$\{5,7,8\}$
67	$(((p \vee q) \rightarrow (p \wedge q)) \vee ((p \vee q) \wedge \neg(p \wedge q)))$	$\{2,3,5,7,8\}$
68	$((q \wedge \neg p) \vee (p \vee ((p \vee q) \rightarrow (p \wedge q))))$	$\{0,2,3,5,7,8\}$
69	$((p \wedge \neg q) \vee (q \vee ((p \vee q) \rightarrow (p \wedge q))))$	$\{1,2,3,5,7,8\}$
7, 0	$((p \wedge \neg q) \vee (\neg p \vee ((p \vee q) \rightarrow (p \wedge q))))$	$\{2,3,5,6,7,8\}$
7, 1	$((q \wedge \neg p) \vee (\neg q \vee ((p \vee q) \rightarrow (p \wedge q))))$	$\{2,3,4,5,7,8\}$
7, 2	$\neg\neg p$	$\{0,1,2,3,4,5\}$
7, 3	$(\neg p \rightarrow q)$	$\{0,1,2,3,4,5,7\}$
7, 4	$\neg\neg q$	$\{0,1,2,3,6,7\}$
7, 5	$(\neg q \rightarrow p)$	$\{0,1,2,3,5,6,7\}$
7, 6	$(\neg p \rightarrow \neg q)$	$\{0,1,2,3,4,5,8\}$
7, 7	$(\neg q \rightarrow \neg p)$	$\{0,1,2,3,6,7,8\}$
7, 8	$\neg(p \rightarrow q)$	$\{4,5\}$
7, 9	$((p \rightarrow q) \rightarrow p)$	$\{0,3,4,5\}$
80	$((p \rightarrow q) \rightarrow q)$	$\{0,1,3,4,5,7\}$
81	$\neg(q \rightarrow p)$	$\{6,7\}$
82	$((q \rightarrow p) \rightarrow p)$	$\{0,1,3,5,6,7\}$
83	$((q \rightarrow p) \rightarrow q)$	$\{1,3,6,7\}$
84	$\neg\neg(p \wedge q)$	$\{0,1,2,3\}$
85	$(\neg(p \wedge q) \rightarrow p)$	$\{0,1,2,3,5\}$
86	$(\neg(p \wedge q) \rightarrow q)$	$\{0,1,2,3,7\}$
87	$(\neg(p \wedge q) \rightarrow (p \vee q))$	$\{0,1,2,3,5,7\}$
88	$\neg\neg(p \vee q)$	$\{0,1,2,3,4,5,6,7\}$
89	$(\neg(p \wedge q) \rightarrow \neg(p \vee q))$	$\{0,1,2,3,8\}$
90	$\neg((p \vee q) \rightarrow (p \wedge q))$	$\{4,5,6,7\}$
91	$(((p \vee q) \rightarrow (p \wedge q)) \rightarrow p)$	$\{0,1,3,4,5,6,7\}$
92	$((p \vee \neg p) \rightarrow q)$	$\{1,2,3,7\}$
93	$((p \vee \neg p) \rightarrow (p \wedge q))$	$\{1,2,3\}$
94	$((p \vee \neg p) \rightarrow ((p \vee q) \rightarrow (p \wedge q)))$	$\{1,2,3,8\}$
95	$((p \rightarrow q) \rightarrow (q \wedge \neg p))$	$\{4,5,7\}$
96	$((q \vee \neg p) \rightarrow p)$	$\{0,2,3,4,5\}$
97	$(((p \vee q) \vee \neg p) \rightarrow (p \wedge q))$	$\{2,3\}$
98	$((q \rightarrow p) \rightarrow (p \wedge \neg q))$	$\{5,6,7\}$
99	$((p \vee \neg q) \rightarrow q)$	$\{1,2,3,6,7\}$
100	$((q \vee \neg q) \rightarrow p)$	$\{0,2,3,5\}$
101	$((q \vee \neg q) \rightarrow (p \wedge q))$	$\{0,2,3\}$
102	$((q \vee \neg q) \rightarrow ((p \vee q) \rightarrow (p \wedge q)))$	$\{0,2,3,8\}$
103	$((q \vee \neg q) \rightarrow ((p \wedge q) \vee \neg p))$	$\{0,2,3,6,7,8\}$
104	$((p \vee \neg p) \rightarrow ((p \wedge q) \vee \neg q))$	$\{1,2,3,4,5,8\}$
105	$((\neg q \vee (p \rightarrow q)) \rightarrow q)$	$\{0,1,3,7\}$
106	$((\neg q \vee (p \rightarrow q)) \rightarrow (p \wedge q))$	$\{0,3\}$
107	$((\neg q \vee (p \rightarrow q)) \rightarrow (q \vee \neg p))$	$\{0,1,3,6,7,8\}$
108	$((\neg q \vee (p \rightarrow q)) \rightarrow ((p \wedge q) \vee \neg p))$	$\{0,3,6,7,8\}$
109	$((\neg p \vee (q \rightarrow p)) \rightarrow p)$	$\{0,1,3,5\}$
110	$((\neg p \vee (q \rightarrow p)) \rightarrow (p \wedge q))$	$\{1,3\}$
111	$((\neg p \vee (q \rightarrow p)) \rightarrow (p \vee \neg q))$	$\{0,1,3,4,5,8\}$
112	$((\neg p \vee (q \rightarrow p)) \rightarrow ((p \wedge q) \vee \neg q))$	$\{1,3,4,5,8\}$
113	$(\neg(p \wedge q) \rightarrow (p \vee \neg(p \vee q)))$	$\{0,1,2,3,5,8\}$

114	$((\neg p \vee (q \to p)) \to (p \vee \neg(p \vee q)))$	$\{0,1,3,5,8\}$
115	$(\neg(p \wedge q) \to (q \vee \neg(p \vee q)))$	$\{0,1,2,3,7,8\}$
116	$((\neg q \vee (p \to q)) \to (q \vee \neg(p \vee q)))$	$\{0,1,3,7,8\}$
117	$((\neg q \vee (p \to q)) \to ((p \wedge q) \vee \neg(p \vee q)))$	$\{0,3,8\}$
118	$((\neg p \vee (q \to p)) \to ((p \wedge q) \vee \neg(p \vee q)))$	$\{1,3,8\}$
119	$((\neg p \vee ((p \vee q) \to (p \wedge q))) \to p)$	$\{0,1,3,4,5\}$
120	$((\neg q \vee ((p \vee q) \to (p \wedge q))) \to q)$	$\{0,1,3,6,7\}$
121	$((\neg(p \wedge q) \vee ((p \vee q) \to (p \wedge q))) \to (p \wedge q))$	$\{0,1,3\}$
122	$((\neg(p \wedge q) \vee ((p \vee q) \to (p \wedge q))) \to ((p \wedge q) \vee \neg(p \vee q)))$	$\{0,1,3,8\}$
123	$((p \to q) \to (q \wedge (p \vee \neg p)))$	$\{0,3,4,5,7\}$
124	$((q \vee \neg p) \to (q \wedge (p \vee \neg p)))$	$\{0,2,3,4,5,7\}$
125	$(((p \vee q) \vee \neg p) \to (q \wedge (p \vee \neg p)))$	$\{2,3,7\}$
126	$((q \vee \neg q) \to (q \wedge (p \vee \neg p)))$	$\{0,2,3,6,7\}$
127	$(((p \vee q) \vee \neg q) \to (q \wedge (p \vee \neg p)))$	$\{2,3,6,7\}$
128	$((\neg q \vee (p \to q)) \to (q \wedge (p \vee \neg p)))$	$\{0,3,7\}$
129	$((q \vee (q \to p)) \to (q \wedge (p \vee \neg p)))$	$\{3,6,7\}$
130	$((q \vee \neg(p \wedge q)) \to (q \wedge (p \vee \neg p)))$	$\{0,2,3,7\}$
131	$((q \vee \neg(p \vee q)) \to (q \wedge (p \vee \neg p)))$	$\{0,2,3,4,5,6,7\}$
132	$((q \vee ((p \vee q) \to (p \wedge q))) \to (q \wedge (p \vee \neg p)))$	$\{0,3,4,5,6,7\}$
133	$((q \vee \neg q) \to ((p \vee q) \wedge (p \vee \neg p)))$	$\{0,2,3,5,6,7\}$
134	$((q \vee (q \to p)) \to ((p \vee q) \wedge (p \vee \neg p)))$	$\{0,3,5,6,7\}$
135	$((q \vee \neg(p \wedge q)) \to ((p \vee q) \wedge (p \vee \neg p)))$	$\{0,2,3,5,7\}$
136	$((\neg q \vee (p \to q)) \to (((p \vee q) \to (p \wedge q)) \vee (q \wedge \neg p)))$	$\{0,2,3,7,8\}$
137	$((q \to p) \to (q \vee (p \wedge \neg q)))$	$\{1,3,5,6,7\}$
138	$((p \vee \neg p) \to (q \vee (p \wedge \neg q)))$	$\{1,2,3,4,5,7\}$
139	$((p \vee \neg q) \to (q \vee (p \wedge \neg q)))$	$\{1,2,3,5,6,7\}$
140	$((p \vee (p \to q)) \to (q \vee (p \wedge \neg q)))$	$\{1,3,4,5,7\}$
141	$((p \vee \neg(p \wedge q)) \to (q \vee (p \wedge \neg q)))$	$\{1,2,3,5,7\}$
142	$((p \vee \neg(p \vee q)) \to (q \vee (p \wedge \neg q)))$	$\{1,2,3,4,5,6,7\}$
143	$((p \vee ((p \vee q) \to (p \wedge q))) \to (q \vee (p \wedge \neg q)))$	$\{1,3,4,5,6,7\}$
144	$((p \vee \neg p) \to ((p \wedge q) \vee (p \wedge \neg q)))$	$\{1,2,3,4,5\}$
145	$(((p \vee q) \vee \neg p) \to ((p \wedge q) \vee (p \wedge \neg q)))$	$\{2,3,4,5\}$
146	$(((p \vee q) \vee \neg q) \to ((p \wedge q) \vee (p \wedge \neg q)))$	$\{2,3,5\}$
147	$((p \vee (p \to q)) \to ((p \wedge q) \vee (p \wedge \neg q)))$	$\{3,4,5\}$
148	$((\neg p \vee (q \to p)) \to ((p \wedge q) \vee (p \wedge \neg q)))$	$\{1,3,5\}$
149	$((p \vee \neg(p \wedge q)) \to ((p \wedge q) \vee (p \wedge \neg q)))$	$\{1,2,3,5\}$
150	$((((p \vee q) \to (p \wedge q)) \vee (p \vee \neg p)) \to ((p \wedge q) \vee (p \wedge \neg q)))$	$\{1,3,4,5\}$
151	$((\neg p \vee (q \to p)) \to (((p \vee q) \to (p \wedge q)) \vee (p \wedge \neg q)))$	$\{1,2,3,5,8\}$
152	$((((p \vee q) \to (p \wedge q)) \vee (q \vee \neg q)) \to (q \wedge (p \vee \neg p)))$	$\{0,3,6,7\}$
153	$((\neg q \vee (p \to q)) \to (((p \wedge q) \vee \neg p) \wedge (q \vee \neg q)))$	$\{0,3,7,8\}$
154	$((\neg p \vee (q \to p)) \to ((p \vee \neg p) \wedge ((p \wedge q) \vee \neg q)))$	$\{1,3,5,8\}$
155	$(((p \vee q) \vee \neg p) \to ((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q))))$	$\{2,3,4,5,7\}$
156	$(((p \vee q) \vee \neg q) \to ((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q))))$	$\{2,3,5,6,7\}$
157	$((p \vee (p \to q)) \to ((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q))))$	$\{3,4,5,7\}$
158	$((q \vee (q \to p)) \to ((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q))))$	$\{3,5,6,7\}$
159	$(((p \vee q) \vee \neg(p \wedge q)) \to ((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q))))$	$\{2,3,5,7\}$
160	$(((p \vee q) \vee \neg(p \vee q)) \to ((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q))))$	$\{2,3,4,5,6,7\}$
161	$(((p \vee q) \vee ((p \vee q) \to (p \wedge q))) \to ((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q))))$	$\{3,4,5,6,7\}$

B.3 The diagram of the fragment $\mathbf{IpL}_1^2$

The fragment $\mathbf{IpL}_m^n$ with restricted nesting of implication was treated in Chapter 4. The diagram of $\mathbf{IpL}_1^2$, listed below, was computed using the exact Kripke model of the fragment (compare figure 26):



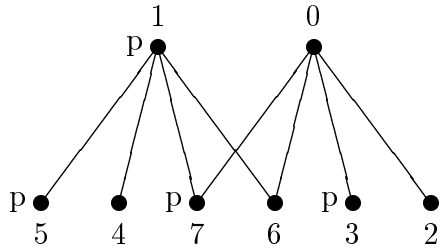
33. FIGURE. *The exact model of* $\mathbf{IpL}_1^2$.

	– –	
	mu = 0	
	– –	
2	p	$\{4, 9, 12\}$
3	q	$\{8, 11, 12\}$
4	$(p \wedge q)$	$\{12\}$
5	$(p \vee q)$	$\{4, 8, 9, 11, 12\}$
	– –	
	mu = 1	
	– –	
0	$(p \wedge \neg p)$	$\{\}$
1	$(p \rightarrow p)$	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
6	$\neg p$	$\{7, 10, 11\}$
7	$\neg q$	$\{5, 9, 10\}$
8	$(p \rightarrow q)$	$\{3, 6, 7, 8, 10, 11, 12\}$
9	$(q \rightarrow p)$	$\{1, 4, 5, 6, 9, 10, 12\}$
10	$\neg(p \wedge q)$	$\{2, 5, 7, 9, 10, 11\}$
11	$\neg(p \vee q)$	$\{10\}$
12	$((p \vee q) \rightarrow (p \wedge q))$	$\{6, 10, 12\}$
13	$(p \vee \neg p)$	$\{4, 7, 9, 10, 11, 12\}$
14	$(q \wedge \neg p)$	$\{11\}$
15	$(q \vee \neg p)$	$\{7, 8, 10, 11, 12\}$
16	$((p \wedge q) \vee \neg p)$	$\{7, 10, 11, 12\}$
17	$((p \vee q) \vee \neg p)$	$\{4, 7, 8, 9, 10, 11, 12\}$
18	$(p \wedge \neg q)$	$\{9\}$
19	$(p \vee \neg q)$	$\{4, 5, 9, 10, 12\}$
20	$(q \vee \neg q)$	$\{5, 8, 9, 10, 11, 12\}$
21	$((p \wedge q) \vee \neg q)$	$\{5, 9, 10, 12\}$
22	$((p \vee q) \vee \neg q)$	$\{4, 5, 8, 9, 10, 11, 12\}$
23	$(\neg p \vee \neg q)$	$\{5, 7, 9, 10, 11\}$
24	$(p \vee (p \rightarrow q))$	$\{3, 4, 6, 7, 8, 9, 10, 11, 12\}$
25	$(\neg q \vee (p \rightarrow q))$	$\{3, 5, 6, 7, 8, 9, 10, 11, 12\}$
26	$(q \vee (q \rightarrow p))$	$\{1, 4, 5, 6, 8, 9, 10, 11, 12\}$
27	$(\neg p \vee (q \rightarrow p))$	$\{1, 4, 5, 6, 7, 9, 10, 11, 12\}$
28	$((p \rightarrow q) \vee (q \rightarrow p))$	$\{1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
29	$(p \vee \neg(p \wedge q))$	$\{2, 4, 5, 7, 9, 10, 11, 12\}$
30	$(q \vee \neg(p \wedge q))$	$\{2, 5, 7, 8, 9, 10, 11, 12\}$
31	$((p \wedge q) \vee \neg(p \wedge q))$	$\{2, 5, 7, 9, 10, 11, 12\}$
32	$((p \vee q) \wedge \neg(p \wedge q))$	$\{9, 11\}$
33	$((p \vee q) \vee \neg(p \wedge q))$	$\{2, 4, 5, 7, 8, 9, 10, 11, 12\}$

34	$((p\to q)\vee\neg(p\wedge q))$	$\{2,3,5,6,7,8,9,10,11,12\}$
35	$((q\to p)\vee\neg(p\wedge q))$	$\{1,2,4,5,6,7,9,10,11,12\}$
36	$(p\vee\neg(p\vee q))$	$\{4,9,10,12\}$
37	$(q\vee\neg(p\vee q))$	$\{8,10,11,12\}$
38	$((p\wedge q)\vee\neg(p\vee q))$	$\{10,12\}$
39	$((p\vee q)\vee\neg(p\vee q))$	$\{4,8,9,10,11,12\}$
40	$(p\vee((p\vee q)\to(p\wedge q)))$	$\{4,6,9,10,12\}$
41	$(q\vee((p\vee q)\to(p\wedge q)))$	$\{6,8,10,11,12\}$
42	$((p\vee q)\vee((p\vee q)\to(p\wedge q)))$	$\{4,6,8,9,10,11,12\}$
43	$(\neg p\vee((p\vee q)\to(p\wedge q)))$	$\{6,7,10,11,12\}$
44	$(\neg q\vee((p\vee q)\to(p\wedge q)))$	$\{5,6,9,10,12\}$
45	$(\neg(p\wedge q)\vee((p\vee q)\to(p\wedge q)))$	$\{2,5,6,7,9,10,11,12\}$
46	$(q\wedge(p\vee\neg p))$	$\{11,12\}$
47	$((p\vee q)\wedge(p\vee\neg p))$	$\{4,9,11,12\}$
48	$(\neg q\wedge(p\vee\neg p))$	$\{9,10\}$
49	$(\neg q\vee(p\vee\neg p))$	$\{4,5,7,9,10,11,12\}$
50	$(\neg(p\wedge q)\wedge(p\vee\neg p))$	$\{7,9,10,11\}$
51	$(((p\vee q)\to(p\wedge q))\vee(p\vee\neg p))$	$\{4,6,7,9,10,11,12\}$
52	$(\neg q\vee(q\wedge\neg p))$	$\{5,9,10,11\}$
53	$((q\to p)\vee(q\wedge\neg p))$	$\{1,4,5,6,9,10,11,12\}$
54	$(\neg(p\vee q)\vee(q\wedge\neg p))$	$\{10,11\}$
55	$(((p\vee q)\to(p\wedge q))\vee(q\wedge\neg p))$	$\{6,10,11,12\}$
56	$(\neg q\vee(q\vee\neg p))$	$\{5,7,8,9,10,11,12\}$
57	$((q\to p)\vee(q\vee\neg p))$	$\{1,4,5,6,7,8,9,10,11,12\}$
58	$(((p\vee q)\to(p\wedge q))\vee(q\vee\neg p))$	$\{6,7,8,10,11,12\}$
59	$(\neg q\vee((p\wedge q)\vee\neg p))$	$\{5,7,9,10,11,12\}$
60	$(\neg q\vee((p\vee q)\vee\neg p))$	$\{4,5,7,8,9,10,11,12\}$
61	$(((p\vee q)\to(p\wedge q))\vee((p\vee q)\vee\neg p))$	$\{4,6,7,8,9,10,11,12\}$
62	$(q\vee(p\wedge\neg q))$	$\{8,9,11,12\}$
63	$((p\wedge q)\vee(p\wedge\neg q))$	$\{9,12\}$
64	$((p\to q)\vee(p\wedge\neg q))$	$\{3,6,7,8,9,10,11,12\}$
65	$(((p\vee q)\to(p\wedge q))\vee(p\wedge\neg q))$	$\{6,9,10,12\}$
66	$((q\vee\neg p)\vee(p\wedge\neg q))$	$\{7,8,9,10,11,12\}$
67	$(((p\wedge q)\vee\neg p)\vee(p\wedge\neg q))$	$\{7,9,10,11,12\}$
68	$((p\to q)\vee(p\vee\neg q))$	$\{3,4,5,6,7,8,9,10,11,12\}$
69	$(((p\vee q)\to(p\wedge q))\vee(p\vee\neg q))$	$\{4,5,6,9,10,12\}$
70	$((q\wedge\neg p)\vee(p\vee\neg q))$	$\{4,5,9,10,11,12\}$
71	$(((p\vee q)\to(p\wedge q))\vee(q\vee\neg q))$	$\{5,6,8,9,10,11,12\}$
72	$((p\vee\neg p)\wedge(q\vee\neg q))$	$\{9,10,11,12\}$
73	$(((p\wedge q)\vee\neg p)\wedge(q\vee\neg q))$	$\{10,11,12\}$
74	$(((p\vee q)\vee\neg p)\wedge(q\vee\neg q))$	$\{8,9,10,11,12\}$
75	$((p\vee\neg p)\wedge((p\wedge q)\vee\neg q))$	$\{9,10,12\}$
76	$((q\wedge\neg p)\vee((p\wedge q)\vee\neg q))$	$\{5,9,10,11,12\}$
77	$(((p\vee q)\to(p\wedge q))\vee((p\vee q)\vee\neg q))$	$\{4,5,6,8,9,10,11,12\}$
78	$((p\vee\neg p)\wedge((p\vee q)\vee\neg q))$	$\{4,9,10,11,12\}$
79	$(((p\vee q)\to(p\wedge q))\vee(\neg p\vee\neg q))$	$\{5,6,7,9,10,11,12\}$
80	$(\neg(p\wedge q)\vee(p\vee(p\to q)))$	$\{2,3,4,5,6,7,8,9,10,11,12\}$
81	$(\neg(p\wedge q)\vee(q\vee(q\to p)))$	$\{1,2,4,5,6,7,8,9,10,11,12\}$
82	$(\neg(p\wedge q)\vee((p\to q)\vee(q\to p)))$	$\{1,2,3,4,5,6,7,8,9,10,11,12\}$
83	$(((p\vee q)\to(p\wedge q))\vee(p\vee\neg(p\wedge q)))$	$\{2,4,5,6,7,9,10,11,12\}$
84	$(((p\vee q)\to(p\wedge q))\vee(q\vee\neg(p\wedge q)))$	$\{2,5,6,7,8,9,10,11,12\}$
85	$((p\vee q)\wedge((p\wedge q)\vee\neg(p\wedge q)))$	$\{9,11,12\}$
86	$(\neg(p\vee q)\vee((p\vee q)\wedge\neg(p\wedge q)))$	$\{9,10,11\}$
87	$(((p\vee q)\to(p\wedge q))\vee((p\vee q)\wedge\neg(p\wedge q)))$	$\{6,9,10,11,12\}$
88	$(((p\vee q)\to(p\wedge q))\vee((p\vee q)\vee\neg(p\wedge q)))$	$\{2,4,5,6,7,8,9,10,11,12\}$
89	$((q\wedge\neg p)\vee(p\vee((p\vee q)\to(p\wedge q))))$	$\{4,6,9,10,11,12\}$
90	$((\neg p\vee\neg q)\vee(p\vee((p\vee q)\to(p\wedge q))))$	$\{4,5,6,7,9,10,11,12\}$
91	$((p\wedge\neg q)\vee(q\vee((p\vee q)\to(p\wedge q))))$	$\{6,8,9,10,11,12\}$
92	$((\neg p\vee\neg q)\vee(q\vee((p\vee q)\to(p\wedge q))))$	$\{5,6,7,8,9,10,11,12\}$
93	$((\neg p\vee\neg q)\vee((p\vee q)\vee((p\vee q)\to(p\wedge q))))$	$\{4,5,6,7,8,9,10,11,12\}$
94	$((p\wedge\neg q)\vee(\neg p\vee((p\vee q)\to(p\wedge q))))$	$\{6,7,9,10,11,12\}$
95	$((q\wedge\neg p)\vee(\neg q\vee((p\vee q)\to(p\wedge q))))$	$\{5,6,9,10,11,12\}$
96	$((\neg q\vee((p\vee q)\to(p\wedge q)))\vee((p\vee q)\wedge(p\vee\neg p)))$	$\{4,5,6,9,10,11,12\}$
97	$((q\vee((p\vee q)\to(p\wedge q)))\vee(\neg(p\wedge q)\wedge(p\vee\neg p)))$	$\{6,7,8,9,10,11,12\}$

B.4 The exactly provable formulas in $\mathbf{L}_1^1$

The exactly provable formulas in $\mathbf{L}_1^1$ where computed using the exact model:



34. FIGURE. *An exact model of* L_1^1.

An explanation of the calculation of the list below can be found in section 5.5

For each formula the corresponding set of $1,1$-types is given on the right. Note that for the bracketing the priority of $\wedge$ is higher than $\vee$ and $\rightarrow$. Likewise $\vee$ has a higher priority than $\rightarrow$.

1: $p \rightarrow p$	{0, 1, 2, 3, 4, 5, 6, 7}
2: $p \rightarrow \Box\bot$	{0, 1, 2, 4, 6}
3: $p \vee \Box\bot$	{0, 1, 3, 5, 7}
4: $p \rightarrow \Box p$	{0, 1, 2, 4, 5, 6}
5: $p \wedge \Box p$	{1, 5}
6: $p \vee \Box p$	{0, 1, 3, 4, 5, 7}
7: $p \rightarrow \Box\neg p$	{0, 1, 2, 3, 4, 6}
8: $p \vee \Box\neg p$	{0, 1, 2, 3, 5, 7}
9: $\neg p \wedge \Box\neg p$	{0, 2}
10: $\Box\bot \vee \neg\Box p$	{0, 1, 2, 3, 6, 7}
11: $\Box\neg p \rightarrow \Box p$	{0, 1, 4, 5, 6, 7}
12: $p \wedge \Box p \rightarrow \Box\bot$	{0, 1, 2, 3, 4, 6, 7}
13: $p \wedge \Box\neg p \rightarrow \Box\bot$	{0, 1, 2, 4, 5, 6, 7}
14: $\Box\neg p \rightarrow p \wedge \Box\bot$	{1, 4, 5, 6, 7}
15: $\Box p \rightarrow (p \vee \Box\bot)$	{0, 1, 2, 3, 5, 6, 7}
16: $\Box\neg p \rightarrow (p \vee \Box\bot)$	{0, 1, 3, 4, 5, 6, 7}
17: $\Box p \rightarrow \neg p \wedge \Box\bot$	{0, 2, 3, 6, 7}
18: $\Box\neg p \vee (p \rightarrow \Box p)$	{0, 1, 2, 3, 4, 5, 6}
19: $\Box\neg p \vee p \wedge \neg\Box p$	{0, 1, 2, 3, 7}
20: $\Box\neg p \vee (p \vee \Box p)$	{0, 1, 2, 3, 4, 5, 7}
21: $(p \vee \Box p) \rightarrow \Box\neg p$	{0, 1, 2, 3, 6}
22: $(p \vee \Box\bot) \wedge \neg(p \wedge \Box p)$	{0, 3, 7}
23: $\Box p \vee p \wedge \neg\Box\neg p$	{0, 1, 4, 5, 7}
24: $(p \vee \Box\neg p) \rightarrow \Box p$	{0, 1, 4, 5, 6}
25: $\neg(p \wedge \Box p) \wedge (p \vee \Box\neg p)$	{0, 2, 3, 7}
26: $p \wedge \Box\bot \vee \neg(p \vee \Box\neg p)$	{1, 4, 6}
27: $p \wedge \Box p \vee \neg(p \vee \Box\neg p)$	{1, 4, 5, 6}
28: $\Box\bot \vee \neg(p \vee \Box p)$	{0, 1, 2, 6}
29: $\Box\bot \vee p \wedge \neg\Box p$	{0, 1, 3, 7}
30: $p \wedge (\Box p \vee \Box\neg p) \rightarrow \Box\bot$	{0, 1, 2, 4, 6, 7}
31: $(\Box p \vee \Box\neg p) \rightarrow (p \vee \Box\bot)$	{0, 1, 3, 5, 6, 7}
32: $\Box\bot \vee \neg(p \vee \Box\neg p)$	{0, 1, 4, 6}
33: $\Box\bot \vee p \wedge \neg\Box\neg p$	{0, 1, 5, 7}
34: $\Box\bot \vee \neg(\Box p \vee \Box\neg p)$	{0, 1, 6, 7}
35: $(p \vee \Box p) \wedge (p \wedge \Box p \rightarrow \Box\bot)$	{0, 1, 3, 4, 7}
36: $\Box\bot \vee \neg(\Box\neg p \vee p \wedge \Box p)$	{0, 1, 4, 6, 7}
37: $(p \vee \Box\neg p) \wedge (p \wedge \Box\neg p \rightarrow \Box\bot)$	{0, 1, 2, 5, 7}
38: $\Box\bot \vee \neg(\Box p \vee p \wedge \Box\neg p)$	{0, 1, 2, 6, 7}
39: $(p \vee \Box p) \wedge (\Box\neg p \rightarrow p \wedge \Box\bot)$	{1, 4, 5, 7}
40: $p \wedge \Box\bot \vee \neg(\Box\neg p \vee p \wedge \Box p)$	{1, 4, 6, 7}
41: $(p \vee \Box p) \rightarrow \Box p \wedge (p \vee \Box\bot)$	{0, 1, 2, 5, 6}
42: $(\Box p \vee \Box\neg p) \rightarrow \Box p \wedge (p \vee \Box\bot)$	{0, 1, 5, 6, 7}

43: $\Box\bot \vee (\Box p \to p) \wedge \neg(p \wedge \Box\neg p)$	$\{0, 1, 2, 5, 6, 7\}$
44: $(p \vee \Box\neg p) \to \Box\neg p \wedge (p \vee \Box\bot)$	$\{0, 1, 3, 4, 6\}$
45: $\Box\bot \vee \neg\Box p \wedge (\Box\neg p \to p)$	$\{0, 1, 3, 6, 7\}$
46: $\Box\bot \vee \neg(p \wedge \Box p) \wedge (\Box\neg p \to p)$	$\{0, 1, 3, 4, 6, 7\}$
47: $(p \vee \Box p) \to \Box\neg p \wedge \neg(p \wedge \Box\bot)$	$\{0, 2, 3, 6\}$
48: $(\Box p \vee \Box\neg p) \to (p \vee \Box\bot) \wedge \neg(p \wedge \Box p)$	$\{0, 3, 6, 7\}$
49: $(p \vee \Box p) \to (\Box\neg p \vee p \wedge \Box p)$	$\{0, 1, 2, 3, 5, 6\}$
50: $(p \vee \Box\neg p) \to (\Box p \vee p \wedge \Box\neg p)$	$\{0, 1, 3, 4, 5, 6\}$
51: $\Box\neg p \vee (p \vee \Box p) \wedge \neg(p \wedge \Box p)$	$\{0, 1, 2, 3, 4, 7\}$
52: $\Box\bot \vee p \wedge \neg(\Box p \vee \Box\neg p)$	$\{0, 1, 7\}$
53: $(p \vee \Box\neg p) \wedge (p \wedge (\Box p \vee \Box\neg p) \to \Box\bot)$	$\{0, 1, 2, 7\}$
54: $\Box p \vee (p \vee \Box\neg p) \wedge \neg(p \wedge \Box\neg p)$	$\{0, 1, 2, 4, 5, 7\}$
55: $\Box\bot \vee \neg(\Box\neg p \vee (p \vee \Box p))$	$\{0, 1, 6\}$
56: $\Box\neg p \wedge (p \vee \Box\bot) \vee \neg(\Box\neg p \vee (p \vee \Box p))$	$\{0, 1, 3, 6\}$
57: $(p \vee \Box p) \wedge (p \wedge (\Box p \vee \Box\neg p) \to \Box\bot)$	$\{0, 1, 4, 7\}$
58: $\Box p \wedge (p \vee \Box\bot) \vee \neg(\Box\neg p \vee (p \vee \Box p))$	$\{0, 1, 5, 6\}$
59: $(\Box\neg p \vee (p \vee \Box p)) \wedge (p \wedge (\Box p \vee \Box\neg p) \to \Box\bot)$	$\{0, 1, 2, 4, 7\}$
60: $(p \vee \Box p) \wedge (p \wedge \Box\bot \vee \neg(\Box\neg p \vee p \wedge \Box p))$	$\{1, 4, 7\}$
61: $(\Box\neg p \vee (p \vee \Box p)) \to (p \vee \Box\bot) \wedge (\Box p \vee \Box\neg p)$	$\{0, 1, 3, 5, 6\}$
62: $(p \to \Box\neg p \wedge \neg\Box\bot) \wedge ((\Box p \vee \Box\neg p) \to (p \vee \Box\bot))$	$\{0, 3, 6\}$

Appendix C

Table of fragments in IpL

For each fragment F of **IpL** in the table below, the number of equivalence classes of F^1, F^2, F^3 and F^4 have been calculated (if possible). In some cases only a lower bound of the number of elements in the diagram could be given.

fragment	$n=1$	$n=2$	$n=3$	$n=4$
$[\wedge]$	1	3	7	15
$[\vee]$	1	3	7	15
$[\wedge, \vee]$	1	4	18	166
$[\neg]$	3	6	9	12
$[\neg\neg]$	2	4	6	8
$[\wedge, \neg]$	5	23	311	66 659
$[\vee, \neg]$	7	385	$> 2^{70}$	
$[\wedge, \vee, \neg]$	7	626	$> 2^{70}$	
$[\wedge, \neg\neg]$	2	8	26	80
$[\vee, \neg\neg]$	2	9	40	281
$[\wedge, \vee, \neg\neg]$	2	19	1 889	
$[\rightarrow]$	2	14	25 165 802	$2^{623\,662\,965\,552\,393} - 50\,331\,618$
$[\wedge, \rightarrow]$	2	18	623 662 965 552 330	
$[\vee, \rightarrow]$	2	∞	∞	∞
$[\wedge, \vee, \rightarrow]$	2	∞	∞	∞
$[\rightarrow, \neg]$	6	518	$3 \times 2^{2\,148} - 546$	
$[\wedge, \rightarrow, \neg]$	6	2 134	**G**	
$[\vee, \rightarrow, \neg]$	∞	∞	∞	∞
$[\wedge, \vee, \rightarrow, \neg]$	∞	∞	∞	∞
$[\rightarrow, \neg\neg]$	4	252	$3 \times 2^{689} - 380$	
$[\wedge, \rightarrow, \neg\neg]$	4	676	$> 2^{6\,383}$	
$[\vee, \rightarrow, \neg\neg]$	5	∞	∞	∞
$[\wedge, \vee, \rightarrow, \neg\neg]$	5	∞	∞	∞

Most of the numbers for F^1, F^2 and F^3 in the tables above can also be found in [JHR 91]. Exceptions are $|Diag([\rightarrow, \neg]^3)|$ and $Diag(|[\rightarrow, \neg\neg]^3)|$, which have been calculated by G. Renardel, and $|Diag([\wedge, \vee, \neg\neg]^3)|$ which has been computed by one of the programs of the author.

The number **G** (which approximately equals $2^{6\,385}$ and has 1 923 digits) was calculated by G. Renardel using a Mathematica program. The outcome of the program:

238535109048049239085364641333913374702561529971090162796061247075003268850 81606333743261028514058270740859585578518573169722287063435154816477455100 67300534461520514807499786875488139392344486567996485245232543943372913882 20916098391913867598073806389545947608903608155768791241781137739941904366 21566924758229992740577301231317143465014885728610629366990425960927253785 72868491272712012675687555199920889937803673124068400811155686757146749638 65974534352524036934304177304456570282321152101220432826978038593549587195 61298318118789345879838234751135199907500279761720979890850319554898038571 28812638789025679933332870594905076897115219091126975780798055001237127554 39620520944274159250068847435305296595661994571943613671505170184414276367 35090517116806133764983542543661798774036708731768419238640888313490745738 62390086523146350166015737176712305161901811444146115001322492027910006472 49183864045852362977616094414762999993096165017734442678516227452375506290 08736460451314686250737873370044479270305241560590241813818217796310418774 11331313443793531657329975493044087486558347732766093260445537422311746173 17797093528190201831176870004473919803016312262407857458418463883914748132 67681770574727454215143982420330885951746991049085873043780462197178577860 18041842769826515608721303793816082124571771381482585476984496988963204119 18886962749800874605124857776935934360695173952772313454078280989807879332 34903875965383757843614426540104617022254343668201611284496543949479906553 24258197494826420483488034937460587870807503895520346988328053802689058378 51783083939857171884062118390901441082675261493352504742093907983048354586 38410994314017753050581581202543786977979384559899009623073296359349660827 33645869281440647431698737494310390629154854362968976529657537647190442245 17143990072975037663714964421877031340844819255400302311404035671823066622 71616684332089066751295339296672239443402484581279801991756 6

Bibliography

[BKK 80] J. Barwise, H.J. Keisler, K. Kunen (eds.), *The Kleene Symposium*, North-Holland, Amsterdam (1980).

[Bellissima 84] F. Bellissima, 'Atoms in modal algebras', *Zeitschr. f. math. Logik und Grundlagen d. Math.*, **30**, 303–312 (1984).

[Benthem 83] J.F.A.K. van Benthem, *Modal Logic and Classical Logic*, Napoli (1983).

[Bernardi 75] C. Bernardi, 'The fixed-point theorem for diagonalizable algebras', *Studia Logica*, **34**, 239–251, (1975).

[Beth 55] E.W. Beth, 'Semantic entailment and formal derivability', *Med. Kon. Ned. Akad. Wet.* Vol. **18**, no 13, Amsterdam (1955).

[Boolos 93] G. Boolos, *The Logic of Provability*, Cambridge University Press (1993).

[Bruijn 75a] N.G. de Bruijn, 'Exact finite models for minimal propositional calculus over a finite alphabet', *Technological University Eindhoven, Report* 75–WSK–02, (1975).

[Bruijn 75b] N.G. de Bruijn, 'An Algol program for deciding derivability in minimal propositional calculus with implication and conjunction propositional calculus over a three letter alphabet', *Technological University Eindhoven, Report* 75–WSK–06 (1975).

[CD 65] J.N. Crossley and M.A.E. Dummett (eds.), *Formal Systems and Recursive Functions*, North-Holland, Amsterdam (1965).

[CK 73] C.C. Chang and H.J. Keisler, *Model Theory*, Amsterdam (1973).

[Curry 77] H.B. Curry, *Foundations of Mathematical Logic*, second edition, New York (1977).

[Diego 66] A. Diego, *Sur les Algèbres de Hilbert*, Gauthier-Vilars, Paris (1966).

[Doets 87] K. Doets, *Completeness and Definability*, Ph.D. Thesis, University of Amsterdam (1987).

[DP 90] B.A. Davey and H.A. Priestly, *Introduction to Lattices and Order*, Cambridge University Press, Cambridge (1990).

[Ershov 89] Yu.L. Ershov (ed.), *Matematicheskaya Logika i Algoritmicheskie*

Problemy, Trudy Instituta Matematiki, vol. 12, Nauka, Novosibirsk, 120–138 (1989).

[Fine 74] K. Fine, *Logics containing K4. Part I*, J.S.L. **39**, 1, 31–42, (1974).

[Fine 85] K. Fine, *Logics containing K4. Part II*, J.S.L. **50**, 3, 619–651, (1985).

[Fitting 69] M.C. Fitting, *Intuitionistic Logic, Model Theory and Forcing*, North-Holland, Amsterdam (1969).

[Gabbay 81] D.M. Gabbay, *Semantical Investigations in Heyting's intuitionistic Logic*, Reidel, Dordrecht (1981).

[GG 90] Z. Gleit and W. Goldfarb, 'Characters and fixed points in provability logic', *Notre Dame Journal of Formal Logic* **31**, 26–36 (1990).

[Gödel 32] K. Gödel, 'Zum intuitionistischen Aussagenkalkül', *Anzeiger der Akademie der Wissenschaften in Wien*, **69**, 65–66 (1932).

[Grigolia 83] R.Sh. Grigolia, 'Finitely generated free Magari algebras' (in Russian), *Logiko-Metodologicheskie Issledovaniya*, Metsniereba, Tbilisi, 135–149 (1983).

[HC 84] G.E. Hughes and M.J. Cresswell, *A Companion to Modal Logic*, Methuen, London (1984).

[Hendriks 80] L. Hendriks, *Logische automaten, beslissen en bewijzen met de computer*, Master's Thesis (in Dutch), Department of Mathematics, University of Amsterdam (1980).

[Hendriks 93] L. Hendriks, 'Inventory of fragments and exact models in intuitionistic propositional logic', *ILLC Prepublication Series*, ML–93–11 (1993).

[HJ 96] L. Hendriks and D.H.J. de Jongh, 'Finitely generated Magari algebras and arithmetic', in *Logic and algebra, Proceedings of the Magari memorial conference, Siena 1994*, to appear (1996).

[Hosoi 66] T. Hosoi, 'The axiomatization of the intermediate propositional systems S_n of Gödel', *Journal of the Faculty of Science of the University of Tokio*, **13**, 183–187 (1966).

[HV 91] J. Halpern and M. Vardi, 'Model checking vs. theorem proving: a manifesto', in *Principles of Knowledge Representation and Reasoning: Proceedings of the* 2^{nd} *International Conference* (1991).

[Jankov 68] V.A. Jankov, 'Constructing a sequence of strongly independent superintuitionistic propositional calculi', *Sov. Math. Doc.* **9**, 806–807 (1966).

[JC 95] D.H.J. de Jongh and L.A. Chagrova, 'The decidability of dependency in intuitionistic propositional logic', *JSL* **60**, 498–504 (1995).

[JHR 91] D.H.J. de Jongh, L. Hendriks, G.R. Renardel de Lavalette, 'Computations in fragments of intuitionistic propositional logic', *Journal of Automated Reasoning* **7**, 537–561 (1991).

[De Jongh 68] D.H.J. de Jongh, *Investigations on the intuitionistic propositional calculus*, Ph.D. Thesis, University of Wisconsin, Madison (1968).

[De Jongh 70] D.H.J. de Jongh, 'A characterization of the intuitionistic proposi-

tional calculus' in [KMV 70], 211–217 (1970).

[De Jongh 80] D.H.J. de Jongh, 'A class of intuitionistic connectives' in: [BKK 80], 103–111 (1980).

[De Jongh 82] D.H.J. de Jongh, 'Formulas of one propositional variable in intuitionistic arithmetic', in [TD 82], 51–64 (1982).

[JT 66] D. de Jongh and A.S. Troelstra, 'On the connection of partially ordered sets with some pseudo-Boolean algebras', *Indag. Math.* **28** no. 3, 317–329 (1966).

[JV 95] D.H.J. de Jongh and A. Visser, 'Embeddings of Heyting algebras', *ILLC Research Report* ML-95-06 (1995) (revised edition of ML-93-14).

[KMV 70] A. Kino, J. Myhill, J. Vesley (eds.), *Intuitionism and Proof Theory*, North-Holland, Amsterdam (1970).

[Kisielewicz 88] A. Kisielewicz, 'A solution of Dedekind's problem on the number of isotone Boolean functions', *J. reine angew. Math.* **386**, 139–144 (1988).

[Kleene 62] S.C. Kleene, 'Disjunction and existence under implications in elementary intuitionistic formalisms', *JSL* **27**, 11–18 (1962).

[Kleitman 69] D. Kleitman, 'On Dedekind's problem: the number of monotone Boolean functions', *Proc. of the AMS* **21** 677–682 (1969).

[Kneale 75] W. Kneale and M. Kneale, *The Development of Logic*, Oxford University Press, London (1975).

[Kripke 65] S.A. Kripke, 'Semantical analysis of intuitionistic logic I', in [CD 65] 92–130 (1965).

[Mostowski 65] A. Mostowski, 'Thirty years of foundational studies', *Acta Phil. Fennica* **XVII** 1–180 (1965).

[Nishimura 60] I. Nishimura, 'On formulas of one variable in intuitionistic propositional logic', *JSL*,**25**, 327–331 (1960).

[Renardel 89] G.R. Renardel de Lavalette, 'Interpolation in fragments of intuitionistic propositional logic', *JSL* **54**, 1419–1430 (1989).

[Rieger 49] N.S. Rieger, 'On the lattice theory of Brouwerian propositional logic', *Acta Fac. Rerum Nat. Univ. Carolinae*, **189**, 3–40 (1949).

[Riemsdijk 85] H. van Riemsdijk, *Het genereren van diagrammen*, Master's Thesis (in Dutch), Department of Mathematics, University of Amsterdam (1985).

[De Rijke 93] M. de Rijke, *Extending Modal Logic*, Ph.D. Thesis, University of Amsterdam (1993).

[Rodenburg 86] P.H. Rodenburg, *Intuitionistic Correspondence Theory*, Ph.D. Thesis, University of Amsterdam (1986).

[Rybakov 89] V.V. Rybakov, 'On admissibility of inference rules in the modal system G' (in Russian), [Ershov 89] (1989).

[Shehtman 78] V.B. Shehtman, 'Rieger-Nishimura lattices', *Soviet Math. Dokl.*, **19/4**, 1014–1018 (1978).

[Shavrukov 93] V. Yu. Shavrukov, 'Subalgebras of diagonalizable algebras of

theories containing arithmetic', *Dissertationes Mathematicae (Rozprawy Matematyczne)* **CCCXXIII**, Warszawa (1993).

[Skolem 13] Th. Skolem, 'Om konstitusjonen av den identiske kalkyls grupper', *Proc. Scan. Math. Con.*, Kristiania, 149–163 (1913).

[Sloane 73] N.J.A. Sloane, *A Handbook of Integer Sequences*, New York (1973).

[Smoryński 85] C. Smoryński, *Self-Reference and Modal Logic*, Universitext, Springer-Verlag, 1985.

[Smullyan 68] R. Smullyan, *First Order Logic*, Springer, New York (1968).

[Solovay 76] R. Solovay, 'Provability interpretations of modal logic', *Israel Journal of Mathematics* **25** 287–304 (1976).

[TD 82] A.S. Troelstra and D. van Dalen (eds.), *The L.E.J. Brouwer Centenary Symposium*, North-Holland, Amsterdam (1982).

[TD 88] A. S. Troelstra and D. van Dalen, *Constructivism in Mathematics, an Introduction* (two volumes), North-Holland, Amsterdam (1988).

[Thijsse 92] E.G.C. Thijsse, *Partial Logic and Knowledge Representation*, Ph.D. Thesis, Tilburg University (1992).

[Thomas 62] I. Thomas, 'Finite limitations on Dummett's *LC*', *Notre Dame Journal of Formal Logic*, **III/3**, 170–174 (1962).

[Troelstra 65] A. Troelstra, 'Finite logics', *Indag. Math.* **27**, 141–152 (1965).

[Urquhart 74] A. Urquhart, 'Implicational formulas in intuitionistic logic', *JSL* **39**, 661–664 (1974).

[Visser 84] A. Visser, 'The provability logics of recursively enumerable theories extending Peano arithmetic at arbitrary theories extending Peano arithmetic', *Journal of Philosophical Logic* **13** 97–113 (1984).

[VBJR 95] A. Visser, J.F.A.K.van Benthem, D.H.J. de Jongh, G.R. Renardel de Lavalette, 'NNIL, A study in intuitionistic propositional logic', in: *Modal Logics and Process Algebra - a bisimulation perspective*, eds. A. Ponse, M. de Rijke, Y. Venema, CSLI, Stanford, 289–326 (1995).

[Visser 85] A. Visser, 'Evaluation, provably deductive equivalence in Heyting's arithmetic of substitution instances of propositional formulas', *Logic Group Preprint Series*, 4, University of Utrecht, Utrecht (1985).

[Zambella 94] D. Zambella, 'Shavrukov's theorem on the subalgebras of diagonalizable algebras for theories containing $I\Delta_0 + EXP$', *Notre Dame Journal of Formal Logic* **35** 147–157 (1994).

[Zwanenburg 94] J. Zwanenburg, *Skeletons in intuitionistic propositional logic*, Master's Thesis, University of Groningen (1994).

List of symbols

Index

Samenvatting

Dit proefschrift doet verslag van een onderzoek naar de semantiek van de intuïtionistische en de modale propositielogica. Dit onderzoek is voor een belangrijk deel geïnspireerd en mogelijk gemaakt door het experimenteren met computerprogramma's.

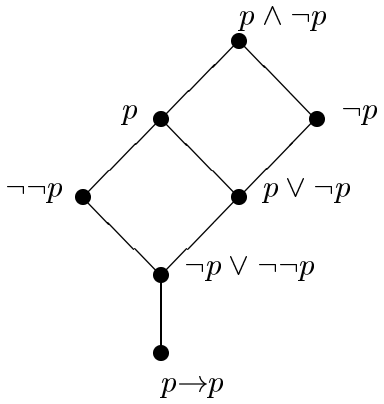
De oudste van deze computerprogramma's zijn zogenaamde *stellingtesters*, programma's waarmee kan worden uitgerekend of uit een bewering A de bewering B logisch volgt. Daarbij wordt alleen gebruik gemaakt van de *vorm* van de beweringen A en B. De computer hoeft dan geen verstand te hebben van sterrenkunde, om uit de bewering 'De Maan is niet van groene kaas' af te leiden: 'Als de Maan van groene kaas is, dan draait Venus om de Aarde'. In *Hoofdstuk 6* worden diverse programma's beschreven om, voor verschillende logische systemen, te berekenen of B uit A volgt. De belangrijkste onderdelen van deze programma's zijn opgenomen in *Appendix A*.

Door de formele taal van de propositielogica, waarin de beweringen kunnen worden geformuleerd, voldoende te beperken krijgt men een zogenaamd *fragment* waarin slechts eindig veel logisch verschillende beweringen mogelijk zijn. Voorbeelden van de beperkingen die men kan opleggen zijn het toelaten van slechts eindig veel basisbeweringen en het verbieden van een of meerdere van de connectieven (voegwoorden) uit de rij 'en' ($\wedge$), 'of' ($\vee$), 'als ... dan' ($\rightarrow$), 'niet' ($\neg$), 'mogelijk' ($\Diamond$) en 'noodzakelijk' ($\Box$). Daarbij maakt het ook nogal wat verschil welke logische afleidingsregels men in het fragment toelaat. Zo heeft $[\wedge, \vee, \rightarrow, \neg]^1_{\mathbf{CpL}}$, het fragment uit de klassieke propositielogica logica met precies één basisbewering en met als connectieven $\wedge, \vee, \rightarrow$ en $\neg$, vier echt verschillende beweringen (A, $\neg A$, $A \wedge \neg A$ en $A \rightarrow A$). Maar het fragment $[\wedge, \vee, \rightarrow, \neg]^1_{\mathbf{IpL}}$ in de intuïtionistische propositielogica, **IpL**, telt oneindig veel verschillende beweringen. Dit geldt voor alle fragmenten in **IpL** die zowel $\vee$ als $\rightarrow$ bevatten.

Als er maar eindig veel verschillende beweringen in een fragment zijn, kunnen we, in principe, alle echt verschillende beweringen uit het fragment berekenen, met behulp van een computerprogramma dat kan uitmaken of een bewering A gelijkwaardig is met de bewering B. Ook de onderlinge relaties tussen deze beweringen (wat volgt er uit wat) kunnen we op die manier in kaart brengen. Zo'n kaart van een fragment,

met daarop alle beweringen uit het fragment en hun onderlinge relaties, noemen we in dit proefschrift een *diagram*.

Hieronder is een voorbeeld van zo'n diagram getekend, in dit geval van het fragment $[\wedge, \vee, \neg]^1$ in de intuïtionistische propositielogica, met basisbewering p:



In dit voorbeeld kan het diagram nog met de hand worden berekend. Voor diagrammen met meer dan twintig beweringen is dat al haast niet meer doenlijk en moet bijvoorbeeld een beroep gedaan worden op een van de eerder genoemde stellingtesters. Uit de eerste experimenten met het berekenen van diagrammen met deze stellingtesters, eind jaren zeventig en begin jaren tachtig, bleek al snel dat zo alleen 'kleine' fragmenten (met hooguit zo'n honderd echt verschillende beweringen) in redelijke tijd in kaart te brengen zijn.

Exacte modellen

Gelukkig bestaat er ook een alternatief voor de stellingtesters, namelijk programma's die gebruik maken van *exacte Kripke-modellen*. Kripke-modellen zijn in de intuïtionistische en modale logica bekende hulpmiddelen om bijvoorbeeld situaties (en hun onderlinge relaties) mee te beschrijven waarin een bepaalde bewering A geldt en de bewering B juist niet. Dat geeft dan een tegenvoorbeeld tegen de bewering dat B uit A volgt.

Een exact Kripke-model van een fragment beschrijft precies alle tegenvoorbeelden die we nodig hebben om voor een fragment uit te maken voor welke beweringen geldt dat B uit A volgt. Elke bewering uit het fragment heeft in het exacte Kripke-model een gebied waar deze bewering geldig is. Als het gebied waar A geldig bevat is in het gebied waar B geldt, dan is B blijkbaar een logisch gevolg van A.

Het berekenen van diagrammen van fragmenten met behulp van exacte modellen gaat vele malen sneller dan met behulp van de eerder genoemde stellingtesters. Lang niet alle fragmenten hebben echter een exact Kripke-model (de situatie in **IpL** is weergegeven in figuur 1 in hoofdstuk 1). Daar staat tegenover dat we veel fragmenten kunnen beschouwen als onderdeel van een fragment dat wel een exact model heeft. Voorbeelden van de berekeningen van diagrammen met behulp van exacte modellen zijn opgenomen in *Appendix B*.

Hoofdstuk 3 van dit proefschrift is gewijd aan de berekening van de diagrammen van de eindige fragmenten in de intuïtionistische propositie logica. Daarbij wordt niet alleen gebruik gemaakt van exacte Kripke-modellen, bij de fragmenten die zich daarvoor lenen wordt ook aangegeven hoe deze exacte modellen kunnen worden geconstrueerd. Zoals uit de tabel in *Appendix C* blijkt worden de diagrammen van eindige fragmenten van **IpL** al bij een klein aantal basisbeweringen in het algemeen al snel astronomisch groot. Het werkelijk laten berekenen van de formules die bij de verschillende beweringen uit de fragmenten horen is in dat geval praktisch uitgesloten en het inzicht in de structuur van de exacte Kripke-modellen is dan vooral van theoretisch belang.

In de modale logica levert, ook met een eindig aantal basisbeweringen, het beperken van de gebruikte voegwoorden in het algemeen nog geen eindige fragmenten op. Een bekende ingreep om toch te komen tot eindige diagrammen is het beperken van de mate waarin het 'mogelijk' en 'noodzakelijk' in een bewering gestapeld voorkomen. Bij een grens van één zou bijvoorbeeld de bewering $\Box\Box A$ (het is noodzakelijk dat het noodzakelijk is dat A') niet meer tot het fragment horen.

In *Hoofdstuk 4* van dit proefschrift wordt iets soortgelijks gedaan voor de intuïtionistische propositielogica. Door het beperken van de stapeling van $\rightarrow$ leidt het samenspel van 'of' ($\vee$) en 'als ... dan' ($\rightarrow$) ook in **IpL** niet langer tot oneindig veel verschillende beweringen. Aangetoond wordt hoe voor deze fragmenten met beperkte stapeling van de implicatie exacte Kripke-modellen geconstrueerd kunnen worden.

Semantische typen

Om de exacte modellen voor fragmenten van propositielogica's te kunnen berekenen is nader onderzocht welke situaties en relaties nodig zijn om alle gewenste tegenvoorbeelden in een Kripke-model te kunnen weergeven. Wat maakt, met andere woorden, een bewering geldig in een bepaalde situatie in een Kripke-model? Het antwoord op deze vraag hangt af van de logica en van het fragment binnen die logica waarmee we werken. In het algemeen kunnen we een volledig beeld geven van een situatie met behulp van een opsomming van de basisbeweringen die er gelden, samen met een overzicht van de andere situaties die vanuit deze situatie 'denkbaar' zijn[1]. De combinatie van deze opsommingen noemen we een *semantisch type*.

Situaties die voor een bepaald fragment van een propositielogica hetzelfde semantische type hebben, gedragen zich logisch gezien eender en er gelden dezelfde beweringen uit het fragment. Het opsporen van de semantische typen voor een bepaald fragment blijkt een heel geschikte methode om een Kripke-model te maken waarin alle voor een fragment nodige tegenvoorbeelden voorhanden zijn. Vaak is zo'n model te groot om een mooi exact Kripke-model te zijn, maar als basis voor een computerprogramma om een diagram mee te berekenen voldoet het prima.

[1] Wat 'denkbaar' is, welke relaties de situaties in een model kunnen hebben, hangt van de logica in kwestie af.

In *Hoofdstuk 2* van dit proefschrift wordt de theorie over de semantische typen uiteengezet en in verband gebracht met een aantal reeds bekende resultaten over modellen en beweringen uit de klassieke, de intuïtionistische en de modale propositielogica.

Formele rekenkunde

In *Hoofdstuk 5* van het proefschrift wordt de theorie van de semantische typen toegepast op een probleem uit de formele rekenkunde, de Peano-rekenkunde **PA**. In de rekenkundige taal zelf kunnen we de bewering formuleren dat een rekenkundige zin bewijsbaar is. Als A een rekenkundige bewering is, dan wordt de rekenkundige bewering 'bewijsbaar A' ook wel geschreven als $\Box A$. De regels die voor deze vorm van 'bewijsbaarheid' gelden vormen een bijzondere modale propositielogica, de bewijsbaarheidslogica **L**.

Nemen we voor een basisbewering p in de bewijsbaarheidslogica een bepaalde rekenkundige zin (bijvoorbeeld '7 heeft 64 verschillende delers'), dan noemen we de verzameling beweringen die we kunnen maken in het fragment van **L** met één basisbewering en die geldig zijn in de rekenkunde als we voor de basisbewering een rekenkundige zin nemen, de $\mathbf{L}^1$-*theorie* van die rekenkundige zin.

Een $\mathbf{L}^1$-theorie heeft als axioma de bewering A, als A zelf een bewering uit de theorie is en alle andere beweringen in de theorie logische gevolgen zijn van A.

Zelfs bij een beperking van het fragment van **L** waarbij alleen beweringen worden toelaten waarin $\Box$ maar één keer gestapeld mag voorkomen (de stapelgrens in dit fragment is dus 2), was tot voor kort niet bekend hoeveel verschillende axioma's voor $\mathbf{L}_2^1$-theorieën er zijn.

Zoals in *Hoofdstuk 5* wordt aangetoond (en uiteindelijk met de computer kon worden berekend) zijn er precies 62 verschillende axioma's voor dit soort theorieën.

Net als bij het berekenen van het aantal verschillende beweringen in de eindige fragmenten van **IpL** is zo'n getal als uitkomst uiteindelijk niet het belangrijkste. Wat telt is dat we zoveel inzicht hebben gekregen in de structuur van fragmenten van propositielogica's dat we computerprogramma's kunnen maken om dergelijke berekeningen uit te voeren.

Titles in the ILLC Dissertation Series:

ILLC DS-1993-1: **Paul Dekker**
Transsentential Meditations; Ups and downs in dynamic semantics

ILLC DS-1993-2: **Harry Buhrman**
Resource Bounded Reductions

ILLC DS-1993-3: **Rineke Verbrugge**
Efficient Metamathematics

ILLC DS-1993-4: **Maarten de Rijke**
Extending Modal Logic

ILLC DS-1993-5: **Herman Hendriks**
Studied Flexibility

ILLC DS-1993-6: **John Tromp**
Aspects of Algorithms and Complexity

ILLC DS-1994-1: **Harold Schellinx**
The Noble Art of Linear Decorating

ILLC DS-1994-2: **Jan Willem Cornelis Koorn**
Generating Uniform User-Interfaces for Interactive Programming Environments

ILLC DS-1994-3: **Nicoline Johanna Drost**
Process Theory and Equation Solving

ILLC DS-1994-4: **Jan Jaspars**
Calculi for Constructive Communication, a Study of the Dynamics of Partial States

ILLC DS-1994-5: **Arie van Deursen**
Executable Language Definitions, Case Studies and Origin Tracking Techniques

ILLC DS-1994-6: **Domenico Zambella**
Chapters on Bounded Arithmetic & on Provability Logic

ILLC DS-1994-7: **V. Yu. Shavrukov**
Adventures in Diagonalizable Algebras

ILLC DS-1994-8: **Makoto Kanazawa**
Learnable Classes of Categorial Grammars

ILLC DS-1994-9: **Wan Fokkink**
Clocks, Trees and Stars in Process Theory

ILLC DS-1994-10: **Zhisheng Huang**
Logics for Agents with Bounded Rationality

ILLC DS-1995-1: **Jacob Brunekreef**
On Modular Algebraic Protocol Specification

ILLC DS-1995-2: **Andreja Prijatelj**
Investigating Bounded Contraction

ILLC DS-1995-3: **Maarten Marx**
Algebraic Relativization and Arrow Logic

ILLC DS-1995-4: **Dejuan Wang**
Study on the Formal Semantics of Pictures

ILLC DS-1995-5: **Frank Tip**
Generation of Program Analysis Tools

ILLC DS-1995-6: **Jos van Wamel**
Verification Techniques for Elementary Data Types and Retransmission Protocols

ILLC DS-1995-7: **Sandro Etalle**
Transformation and Analysis of (Constraint) Logic Programs

ILLC DS-1995-8: **Natasha Kurtonina**
Frames and Labels. A Modal Analysis of Categorial Inference

ILLC DS-1995-9: **G.J. Veltink**
Tools for PSF

ILLC DS-1995-10: **Giovanna Ceparello**
(to be announced)

ILLC DS-1995-11: **W.P.M. Meyer Viol**
Instantial Logic. An Investigation into Reasoning with Instances

ILLC DS-1995-12: **Szabolcs Mikulás**
Taming Logics

ILLC DS-1995-13: **Marianne Kalsbeek**
Metalogics for Logic Programming

ILLC DS-1995-14: **Rens Bod**
Enriching Linguistics with Statistics: Performance Models of Natural Language

ILLC DS-1995-15: **Marten Trautwein**
Computational Pitfalls in Tractable Grammatical Formalisms

ILLC DS-1995-16: **Sophie Fischer**
The Solution Sets of Local Search Problems

ILLC DS-1995-17: **Michiel Leezenberg**
Contexts of Metaphor

ILLC DS-1995-18: **Willem Groeneveld**
Logical Investigations into Dynamic Semantics

ILLC DS-1995-19: **Erik Aarts**
Investigations in Logic, Language and Computation

ILLC DS-1995-20: **Natasha Alechina**
Modal Quantifiers

ILLC DS-1996-1: **Lex Hendriks**
Computations in Propositional Logic

Algebraic succession rules and Lattice paths with an infinite set of jumps

Cyril Banderier [a,*], Jean-Marc Fédou [b], Christine Garcia [b], Donatella Merlini [c]

[a] *LIPN, Univ. Paris 13, 93 430 Villetaneuse (France)*

[b] *I3S, URA 1376 du CNRS Sophia-Antipolis (France)*

[c] *DSI, Università degli Studi di Firenze, Via Lombroso 6/17, 50134 Firenze (Italy)*

Abstract

Whereas walks on $\mathbb{N}$ with a finite set of jumps were the subject of numerous studies, walks with an infinite number of jumps remain quite rarely studied, at least from a combinatorial point of view. A reason is that even for relatively well structured models, the classical approach with context-free grammars fails as we deal with rewriting rules over an infinite alphabet. However, several classes of such walks offer a surprising structure: in this article, we show that one can make explicit the generating functions of the number of walks (with respect to their length) between two fixed points. We also give several theorems on their nature (rational, algebraic). In fact, we mostly deal with succession rules of the type

$$(k) \rightsquigarrow (0)^{e_k}(1)^{e_{k-1}} \dots (k-1)^{e_1}(k)^{e_0} \dots (k+a)^{e_{-a}},$$

for which we show that the associated generating function $F(z)$ is algebraic if the generating function $E(z)$ of the e_k's is rational (via a new combinatorial argument: a decomposition of the paths which leads to an algebraic equation satisfied by the noncommutative generating function). Via an analytical argument (the kernel method), we also show a stronger result: if $E(z)$ is algebraic, then $F(z)$ is algebraic. When $a = 1$, this leads to remarkably simple formulae which can also be proved with a Riordan array approach. This generalises all the previously known results.

We end with some examples of recent problems in combinatorics or theoretical computer science which lead to such rules.

* Corresponding author.

Email addresses: `Cyril.Banderier@inria.fr` (Cyril Banderier), `fedou@unice.fr` (Jean-Marc Fédou), `cgarcia@unice.fr` (Christine Garcia), `merlini@dsi.unifi.it` (Donatella Merlini).

URLs: `http://algo.inria.fr/banderier` (Cyril Banderier), `http://deptinfo.unice.fr/~fedou` (Jean-Marc Fédou), `http://www.dsi.unifi.it/~merlini` (Donatella Merlini).

22 January 2003

1 Introduction

A considerable number of problems from computer science deals with a sum of independent identical distributed random variables $\Sigma_n = X_1 + X_2 + \ldots + X_n$ (where each of the X_i's assumes integer values). We consider here the following model of random walks: the walk starts (at time 0) from a point Σ_0 of $\mathbb{Z}$ and at time n, one makes a jump $X_n \in \mathbb{Z}$; so the new position is given by the recurrence $\Sigma_n = \Sigma_{n-1} + X_n$ where, when $\Sigma_{n-1} = k$, the jump X_n is constrained to belong to a fixed set $\mathcal{P}_k$ (that is, the possible jumps depend on the position of the walk).

These "walks on $\mathbb{Z}$" are homogeneous in time (that is to say, the set of jumps when one is at position k is independent from the time). When the positions Σ_n's are constrained to be nonnegative, we talk about "walks on $\mathbb{N}$". The probabilistic model under consideration here is the uniform distribution on all paths of length n.

When the sets $\mathcal{P}_k$'s are equal to a fixed set $\mathcal{P}$ (the simplest interesting case being $\mathcal{P} = \{-1, +1\}$), the corresponding walks have been deeply studied both in combinatorics (Dyck paths,...) and in probability theory (coin flipping, ...). We refer to [4] for enumerative and analytical studies of such "walks on $\mathbb{N}$ with a finite set of jumps". When the sets $\mathcal{P}_k$'s are unbounded, both enumeration and asymptotics become cumbersome: contrary to the previous case, the walks are not space-homogeneous (the set of available jumps depends on the position) and it is not possible to generate them by context-free grammars (which are classically defined for finite alphabet only). However, if the sets $\mathcal{P}_k$'s have a "combinatorial" shape, it is reasonable to hope that the generating function associated to the corresponding walk would have some nice properties. We show here that this hope is legitimate and we present several classes of such walks, for which we are able to give the nature of their generating function.

Our results have potential impacts on the theory of generating trees, the enumeration and generation of combinatorial objects (general classes of lattice paths, constrained permutations, ...) and on the study of rewriting rules on an infinite alphabet.

A definition of the generating function associated to the walk is given in Section 2 where we also present the generating tree and Riordan array viewpoints. In Section 3, we give several theorems related to the nature of the generating functions associated to some walks (which deeply generalise previously known results from [1–8,15,17–22]). Then, we give some asymptotic results. In Section 4, we give some examples of problems in which some of the new classes of walks that we study in this article appear.

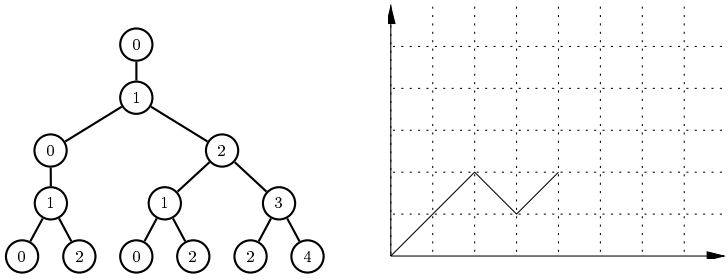


Fig. 1. The generating tree of the walk on $\mathbb{N}$ with jumps $\mathcal{P} = \{+1, -1\}$ starting in 0 (and up to length $n = 4$). Each branch corresponds to a path. The branch $(0, 1, 2, 1, 2)$ corresponds to the path drawn on the lattice.

2 Lattice paths, generating trees, succession rules and their generating functions

In combinatorics, it is classical to represent a particular walk as a path in a two dimensional lattice. Thus the drawing corresponds to the walk of length n linking the points $((0, \Sigma_0), (1, \Sigma_1), \ldots, (n, \Sigma_n))$. It is also convenient to represent all the walks of length $\leq n$ as a tree of height n, where the root (at level 0 by convention) is labelled with the starting point of the walks and where the label of each node at level n encodes a possible position of the walk (see Figure 1).

Let $f_{n,k}$ be the number of walks on $\mathbb{N}$ of length n going from the starting point to k (or, equivalently, the number of nodes with label k at level n in the tree). We want to find the bivariate generating function

$$F(z, u) = \sum_{n \geq 0} f_n(u) z^n = \sum_{k \in \mathbb{N}} F_k(z) u^k = \sum_{k \in \mathbb{N}, n \geq 0} f_{n,k} u^k z^n \,. \tag{1}$$

where u encodes the final altitude of the walk (the label in the tree), z the length of the walk (the level in the tree), and where $f_n(u)$ is a Laurent polynomial (that is, a polynomial with finitely many monomials of negative and positive degree).

2.1 Generating trees and succession rules

The concept of generating trees has been used from various points of view and was introduced in the literature by Chung, Graham, Hoggatt and Kleiman [10] to examine the reduced Baxter permutations.

We define here a generating tree as a rooted labelled tree with the property that if two nodes have the same label then, for any integer ℓ, they have exactly

the same number of children with label ℓ. For readability, we often write the labels in parentheses. Thus, a generating tree is fully defined by:
1) the label of the root (that we also call "axiom");
2) a set of rules $\{(k) \rightsquigarrow \mathcal{M}_k\}_{k\in\mathbb{N}}$ explaining how to derive from the label of a parent the labels of its children. ($\mathcal{M}_k$ is a multiset [1] of labels.)

Point 2) defines what we call a *succession rule*. The multisets $\mathcal{M}_k$ are directly related to the multisets $\mathcal{P}_k$ (the allowed jumps introduced in Section 1) via the relation $\mathcal{M}_k := \{k + x, x \in \mathcal{P}_k\}$. For example, Figure 1 illustrates the upper part of the generating tree which corresponds to the set of rules $\{(k) \rightsquigarrow (k-1)(k+1)\}_{k\in\mathbb{N}}$ with 0 as label of the root. That is, one has in this case $\mathcal{P}_k = \{-1, +1\}$ and $\mathcal{M}_k = \{k-1, k+1\}$. In what follows, instead of writing

$$((0), \{(k) \rightsquigarrow (k-1)(k+1)\}_{k\in\mathbb{N}}) \ ,$$

we use the more readable notation

$$[(0),\ (k) \rightsquigarrow (k-1)(k+1)\,] \, ,$$

or alternatively

$$\begin{cases} (0) \\ (k) \rightsquigarrow (k-1)(k+1) \, . \end{cases}$$

Note that we only consider nonnegative walks, thus when a rule gives a negative label, we simply ignore this label. In the above case, when $k = 0$ the rule is thus $(0) \rightsquigarrow (1)$ and not $(0) \rightsquigarrow (-1)(1)$. If a label is repeated, we directly write $(k)^n$ instead of $(k)\ldots(k)$ (n occurrences). This corresponds to walks with multiplicities, or if one wants, to distinguish two occurrences of the same label in a succession rule by colouring them in two different colours.

The method of generating trees was also successfully used by West [25], Dulucq, Gire, and Guibert [12–14], for the enumeration of permutations with forbidden sequences (see Fig. 2). In fact, the kind of rewriting rules under consideration here were intensively studied partly because they are useful to solve some cases of the following famous conjecture:

Conjecture 1 (Stanley–Wilf) *For any given pattern, there exists a constant C such that there are asymptotically $O(C^n)$ permutations of length n avoiding this pattern.*

[1] Multisets are sets in which repetitions are allowed. E.g., for multisets, one has $\{1, 1, 2\} \cup \{1, 2\} = \{1, 1, 1, 2, 2\}$.

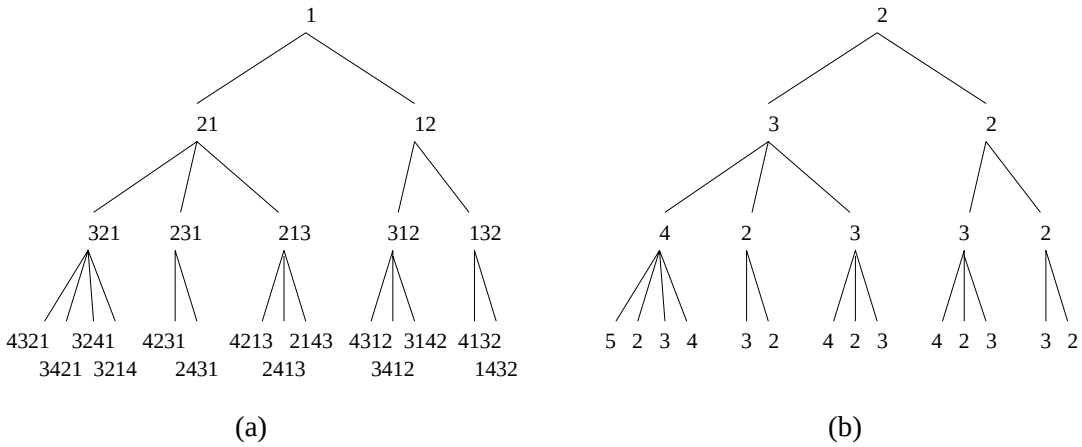


Fig. 2. The generating tree of 123-avoiding permutations. (a) Nodes labelled by the permutations. (b) Nodes labelled by the numbers of children. It can be proved that the right tree corresponds to the rule $[(2),(k) \rightsquigarrow (2)\dots(k+1)]$.

This conjecture shows that to forbid a pattern is a strong constraint (permutations with a forbidden pattern are of density zero in the whole set of permutations). For any fixed pattern, the algebraicity of the generating function of permutations avoiding this pattern would be a proof of this conjecture. However, it is not possible to solve all the cases by this approach (as some patterns lead to non-D-finite [2] generating functions).

These last years, the concept of generating tree has been intensively exploited by Barcucci, Del Lungo, Ferrari, Pergola, Pinzani, and Rinaldi [6,7,17,18] in relation with the ECO method (ECO stands for enumeration of combinatorial objects) which allows the enumeration and recursive construction of various classes of combinatorial objects. In fact, the succession rule approach has several equivalent interpretations, ECO systems, discrete random walks, infinite automata or Riordan arrays (see later). For all these problems, it is interesting to classify the rules according to the nature (rational, algebraic, transcendental) of the corresponding generating function $F(z,u)$. This program has been proposed by Pinzani and *al.* [6,7,17,18] in the area of ECO systems (the so called "ECO systems" are the generating trees where each integer has exactly k successors). A classical and easy result is that finite succession rules have rational generating function since they correspond to a regular language. Another result (proved in [3]) is that every finite transformation of the succession rule

$$(k) \rightsquigarrow (1)(2)\dots(k)(k+1)$$

leads to an algebraic generating function. In the same paper are also described succession rules leading to exponential generating functions having a

[2] A series $F(z)$ is said to be *holonomic*, or *D-finite*, if it satisfies a linear differential equation with polynomial coefficients in z. Equivalently, its coefficients f_n satisfy a linear recurrence relation with polynomial coefficients in n.

nice closed-form formula which have been more extensively studied by Corteel in [11]. Our paper is principally devoted to the study of succession rules having algebraic generating function.

In a first step, our approach is closely related with Schützenberger's methodology, which consists in finding first a bijection between the objects and the words of an algebraic language and then a non ambiguous grammar for the language. Taking the commutative image leads to an algebraic system for the generating function. For a succession rule, we define its noncommutative formal power series using the infinite alphabet of positive integers. We use a new operation $\oplus$ which allows us to get a non ambiguous decomposition of the formal power series associated to the generating tree. We deduce algebraic equation by taking the commutative image of the formal power series. This method allows us to get an algebraic decomposition of the general succession rule

$$(k) \rightsquigarrow (1)\dots(k-1)(k)^{e_0}\dots(k+a)^{e_{-a}},$$

for any finite sequence (e_i), and more generally for the succession rule

$$(k) \rightsquigarrow (1)^{e_{k-1}}\dots(k-1)^{e_1}(k)^{e_0}\dots(k+a)^{e_{-a}},$$

for any sequence $(e_i)_{i=-a}^{\infty}$ proving thereby that the generating function of the generating tree is *algebraic* when the sequence (e_i) is *rational*. This gives a combinatorial proof for a generalisation of the results of [3].

In a second step, we give some analytical proofs (based on the kernel method) of the algebraicity of the generating function associated to the generating tree when the sequence (e_i) in the succession rule is *algebraic*.

2.2 Noncommutative generating functions for succession rules

It is convenient to see a generating tree (defined in the previous subsection) as the infinite tree constructed with a root labelled by the axiom and where each node labelled k has sons labelled according to the succession rule.

For a generating tree $\mathcal{T}$, we define the language $\mathcal{L}$ as the set of words over $\mathbb{N}$, beginning by the axiom r and in which each letter (k) is followed by a letter (if any) which belongs to the multiset $\mathcal{M}_k$. Each word w of $\mathcal{L}$ corresponds to at least one branch[3] of $\mathcal{T}$. For each word $w \in \mathcal{L}$, let $m(w)$ be the number of branches in the generating tree $\mathcal{T}$ corresponding to the word w. We denote by S the noncommutative formal power series $S = \sum_{w\in\mathcal{L}} m(w)w$.

[3] By branch of the infinite tree $\mathcal{T}$, we mean any sequence of labels corresponding to a branch of any finite subtree of $\mathcal{T}$. Figure 1 gives an example.

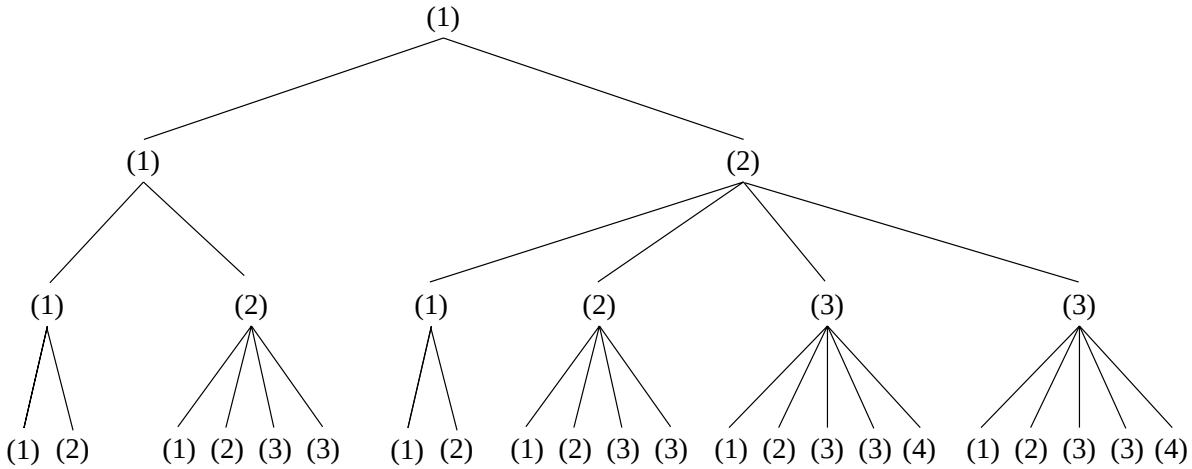


Fig. 3. Truncated generating tree of $[(1),(k) \rightsquigarrow (1)(2)(3)(3)(4)\ldots(k)(k+1)]$. The associated generating function is $F(z,1) = z + 2z^2 + 6z^3 + 22z^4 + \ldots$ and the corresponding noncommutative GF is $S = 1+11+12+111+112+121+122+2.123+\ldots$

By construction, the generating tree $\mathcal{T}$ and the noncommutative formal power series S have the same generating function

$$F(z,1) = \sum_{n\in\mathbb{N}} f_n(1) z^n = \sum_{n\in\mathbb{N}} \left(\sum_{w\in\mathcal{L}, |w|=n+1} m(w) \right) z^n \, .$$

We use standard external product and concatenation over the noncommutative formal power series: For any real x and for any word v, one has

$$xS := \sum_{w\in\mathcal{L}} (xm(w))w \qquad \text{and} \qquad v.S := \sum_{w\in\mathcal{L}} m(w)(v.w).$$

We now define the "shift" operation (that we write $\oplus$) as follows:

Definition 2 *For $i \in \mathbb{N}$, we define $i^{\oplus} := i+1$. By extension if $w = w_1 \ldots w_n$ is a word with n letters, then $w^{\oplus} := w_1^{\oplus} \ldots w_n^{\oplus}$ and $S^{\oplus} := \sum_{w\in\mathcal{L}} m(w) w^{\oplus}$.*

Clearly, the generating functions associated to S and $S^{\oplus}$ are equal.

2.3 Riordan arrays

We introduce now the concept of *matrix associated to a generating tree*: this is an infinite matrix $\{d_{n,k}\}_{n,k\in\mathbb{N}}$ where $d_{n,k}$ is the number of nodes at level n with label $k+r$, where r is the label of the root. For example, the matrix associated to the generating tree of the Figure 1 (walk with jumps $+1, -1$) is

the following:

$n\backslash k$	0	1	2	3	4
0	1				
1	0	1			
2	1	0	1		
3	0	2	0	1	
4	2	0	3	0	1

Many such matrices can be studied from a *Riordan array* viewpoint. In fact, the concept of a Riordan array provides a remarkable characterisation of many lower triangular arrays that arise in combinatorics and algorithm analysis. The theory has been introduced in the literature in 1991 by Shapiro, Getu, Woan, and Woodson [23]. Riordan arrays are a powerful tool in the study of many counting problems having a flavour of Lagrange inversion [19].

A Riordan array is an infinite lower triangular array $\{d_{n,k}\}_{n,k\in\mathbb{N}}$, defined by a pair of formal power series $(d(z), h(z))$, such that the k-th column is given by $d(z)(zh(z))^k$, i.e.:

$$d_{n,k} = [z^n]d(z)(zh(z))^k, \qquad n, k \geq 0.$$

From this definition, one has $d_{n,k} = 0$ for $k > n$. The bivariate generating function for the Riordan array is:

$$\sum_{n,k\geq 0} d_{n,k}u^k z^n = \frac{d(z)}{1 - uzh(z)}.$$

In what follows, we always assume that $d(0) \neq 0$; if we also have $h(0) \neq 0$ then the Riordan array is said to be *proper*; in the proper-case the diagonal elements $d_{n,n}$ are different from zero for all $n \in \mathbb{N}$. The most simple example is the Pascal triangle for which one has

$$\binom{n}{k} = [z^n]\frac{1}{1-z}\left(\frac{z}{1-z}\right)^k,$$

where we recognise the proper Riordan array with $d(z) = h(z) = 1/(1-z)$. Proper Riordan arrays are characterised by the existence of a sequence $A = (a_i)_{i\in\mathbb{N}}$ with $a_0 \neq 0$, called the *A-sequence*, such that every element $d_{n+1,k+1}$ can be expressed as a linear combination, with coefficients in A, of the elements in the preceding row, starting from the preceding column:

$$d_{n+1,k+1} = a_0 d_{n,k} + a_1 d_{n,k+1} + a_2 d_{n,k+2} + \ldots$$

It can be proved that $h(z) = A(zh(z))$, $A(z)$ being the generating function for the sequence A. For example, for the Pascal triangle one has: $A(z) = 1+z$ and the previous relation reduces to the well-known recurrence relation for binomial coefficients. The A-sequence doesn't characterise completely $(d(z), h(z))$ because $d(z)$ is independent of $A(z)$. But it can be proved that there exists a unique sequence $Z = (z_0, z_1, z_2, \ldots)$, such that every element in column 0 can be expressed as a linear combination of all the elements of the preceding row:

$$d_{n+1,0} = z_0 d_{n,0} + z_1 d_{n,1} + z_2 d_{n,2} + \ldots$$

This property has been recently studied in [19], where it is proved that $d(z) = d(0)/(1 - zZ(zh(z)))$, $Z(z)$ being the generating function for Z. Thus the triple $(d(0), Z(z), A(z))$ characterises every proper Riordan array. We use these claims in Theorem 10.

3 Generating functions of succession rules

This section contains the main results of our article. We give several theorems, making explicit the generating functions associated to different kind of rules ("rational" exponents: Theorem 4, "algebraic" exponents: Theorem 9, ...).

3.1 Lattice paths and generating trees

Consider a function $e(k, i)$ which is going from $\mathbb{N}^2$ to $\mathbb{N}$. We now fix an integer $a > 0$ (a corresponds to the largest positive possible jump; so we restrict here our attention to functions such that $e(k, i) = 0$ for any k as soon as $i > a$). Then the walks with an infinite set of jumps under consideration here are of the following kind:

$$\begin{cases} (r) \\ (k) \rightsquigarrow (0)^{e(k,0)}(1)^{e(k,1)} \ldots (k-1)^{e(k,k-1)}(k)^{e(k,k)} \ldots (k+a)^{e(k,k+a)} \end{cases} \tag{2}$$

where the exponent $e(k, i)$ is the multiplicity of the jumps from k to i and where r is the starting position of the walk (or equivalently, the root of the associated generating tree). In what follows, we often (but not always) consider the case for which $e(k, i) = e_{k-i}$ (where $(e_k)_{k\in\mathbb{Z}}$ is a fixed sequence).

If the sequence $(e(k, i))_k$ (for a fixed i) is ultimately 0, then the situation covers the case of walks with a finite set of jumps [4]. If the sequence is ultimately 1, then this covers the case of "factorial rules" which are of great interests for

the generation of combinatorial objects [8] and for which it was proved in [3] that the associated generating functions are algebraic.

We still note $f_{n,k}$ the number of walks on $\mathbb{N}$ of length n going from the starting point to k and we want to find the bivariate generating function $F(z,u) = \sum_{n,k\geq 0} f_{n,k} u^k z^n$. These random walks on $\mathbb{N}$ can equivalently be seen as lattice paths, generating trees, and also as Riordan arrays (when $a = 1$).

In Tables 1 and 2 (see at the end of this article), we give a list of succession rules with simple combinatorial patterns, the reference to famous numbers or combinatorial problems they refer to, the generating function $F(z,1)$, and the numbers identifying the corresponding sequences in the *On-Line Encyclopedia of Integer Sequences* `http://www.research.att.com/~njas/sequences/`; ECS stands for the *Encyclopedia of Combinatorial Structures*, a database reachable via `http://algo.inria.fr/encyclopedia/`.

3.2 Succession rules: "rational" exponents

In this section, we study succession rules having the following general form

$$[(1), (k) \rightsquigarrow (1)^{e_{k-1}} \dots (k-1)^{e_1} (k)^{e_0} \dots (k+a)^{e_{-a}}] .$$

"Rational" exponents means here that the generating function $E(z)$ of the e_k's (which are nonnegative integers) is rational.

Using decomposition of paths, we prove the algebraicity of the associated generating function $F(z,1)$, first when the sequence $(e_i)_{i>0}$ is constant equal to one (Theorem 3), and then when the sequence (e_i) follows a linear recurrence, that is when the e_i's are coefficients of a rational generating function (Theorem 4).

Theorem 3 *The noncommutative generating function S associated to the generating tree*

$$[(1), (k) \rightsquigarrow (1) \dots (k-1)(k)^{e_0} \dots (k+a)^{e_{-a}}]$$

satisfies the following equation

$$S = (1) + (1) \sum_{i=0}^{a} e_{-i} S^{i\oplus} \prod_{j=0}^{i-1} (\epsilon + S^{j\oplus}),$$

where $S^{i\oplus} = (S^{(i-1)\oplus})^{\oplus}$ and $S^{0\oplus} = S$.

Consequently, the (commutative) generating function $F(z,1)$ of the generating

tree is algebraic and satisfies

$$F(z,1) = z + zF(z,1)\sum_{i=0}^{a} e_{-i}(1+F(z,1))^i\,.$$

Example: For the generating tree associated to $[(1),(k) \rightsquigarrow (1)\dots(k+1)]$, this gives $S = (1) + S + S^{\oplus}(\epsilon + S)$.

Remark: The algebraicity of $F(z,1)$ for such generating trees was first proved analytically in [3] (via a proof which is leading to a neat closed-form formula). We give here a *combinatorial* proof of this algebraicity (via a neat decomposition of the tree).

Proof. The proof is deduced from the recursive decomposition of the paths in the generating tree. We need to define $({}_rS)$ as the formal sum of the paths in the generating tree obtained by replacing the axiom by r:

$$[(r),(k) \rightsquigarrow (1)\dots(k-1)(k)^{e_0}\dots(k+a)^{e_{-a}}]\,.$$

We can write recursively $({}_rS)$ using the following non ambiguous decomposition (see Fig. 4). Let $w \neq r$ be a non trivial path of $({}_rS)$, then w can be written $w = r.u$

- if each letter of u is $\geq r$ then $u = v^{(r-1)\oplus}$ where v is a path of the generating tree,
- if not, let m the first letter $< r$ in u, so u can be written $v_1^{(r-1)\oplus}v_2$ where v_1 is a path of the generating tree and v_2 is a path of $({}_mS)$, v_2 being the longest suffix of u beginning by m.

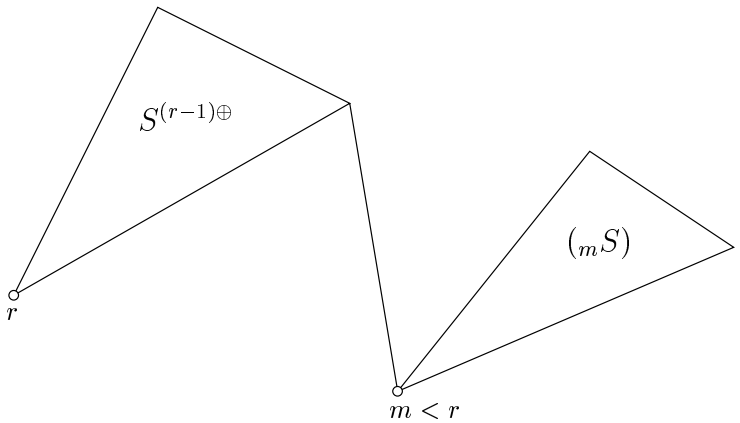


Fig. 4. Decomposition of $({}_rS)$.

One has $({}_rS) = S^{(r-1)\oplus}(\epsilon + \sum_{m=1}^{r-1}({}_mS))$. It is easy to see that

$$({}_{r+1}S) = S^{r\oplus}\prod_{m=0}^{r-1}(\epsilon + S^{m\oplus})\,.$$

The equality $S = (1) + (1)\sum_{r=0}^{a} e_{-r}({}_{r+1}S)$ concludes the proof. □

The algebraic equation satisfied by the algebraic generating function given in the small catalogue of ECO-systems of [3] can be deduced from the previous theorem. For instance, this is the case of Motzkin numbers, Schröder numbers and ternary trees. For the general case, that we consider now, the difficulty is to deal with the e_i jumps from (k) to $(k-i)$.

Theorem 4 *Consider $E(z) = \sum_{i\geq -a} e_i z^i$. The rationality of $E(z)$ implies the algebraicity of the generating function $F(z,1)$ of the generating tree*

$$[(1),(k) \rightsquigarrow (1)^{e_{k-1}}\dots(k-1)^{e_1}(k)^{e_0}\dots(k+a)^{e_{-a}}]\,.$$

Proof. We begin by giving the different equations obtained from the recursive decomposition of the paths in the generating tree. As in the proof of Theorem 3, we need to define $({}_rS_i)$ as the formal sum of the paths ending by i in the following generating tree

$$[(r),(k) \rightsquigarrow (1)^{e_{k-1}}\dots(k-1)^{e_1}(k)^{e_0}\dots(k+a)^{e_{-a}}]\,.$$

We write (S_i) for $({}_1S_i)$. Applying the same non ambiguous decomposition as in Theorem 3 and considering the last letter of each factor (see Fig. 5), we get

$$({}_{r+1}S_i) = (S_{i-r})^{r\oplus} + \sum_{m=1}^{r}\sum_{j\geq 1} e_{j+r-m}(S_j)^{r\oplus}({}_mS_i). \tag{3}$$

Let ${}_rF_i(z)$ be the generating functions of $({}_rS_i)$ (paths beginning in r and ending in i). By convention, ${}_rF_i = 0$ for $i \leq 0$ or $r \leq 0$. One has $F_i = ({}_1F_i)$ (as 1 is the root of the generating tree). Let $G_i := \sum_{j\geq 1} e_{i+j-1}F_j$ for $1 \leq i \leq p$ and $H_n(z) := \sum_{n=i_1+\dots+i_h} G_{i_1}(z)\dots G_{i_h}(z)$ for $n \geq 0$ with the convention that $H_0(z) := 1$. Note that H_n is a polynomial in $G_1,\dots,G_a$. From Equation (3), one gets

$$\begin{aligned}{}_{r+1}F_i(z) &= F_{i-r}(z) + \sum_{m=1}^{r}\sum_{j\geq 1} e_{j+r-m}\,F_j(z)\;({}_mF_i(z))\\ &= F_{i-r}(z) + \sum_{m=1}^{r} G_{r-m+1}(z)\;({}_mF_i(z))\,.\end{aligned}$$

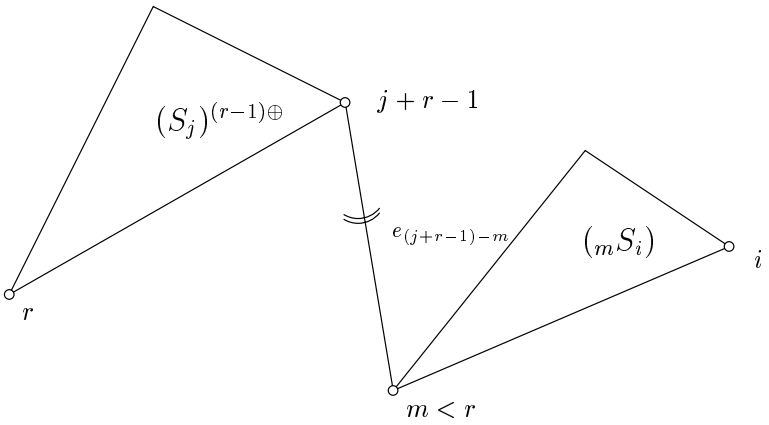


Fig. 5. Decomposition of $({}_rS_i)$.

For $k \geq 2$, decomposing F_k according to the first positive jump, gives

$$F_k = z \sum_{m=0}^{a} e_{-m}({}_{m+1}F_k).$$

Using the fact that

$$\begin{aligned}
({}_{r+1}F_i) &= F_{i-r} + \sum_{m=1}^{r} G_{r-m+1} \sum_{j=0}^{m-1} H_{m-1-j} F_{i-j} \\
&= F_{i-r} + \sum_{m=0}^{r-1} F_{i-m} \sum_{j=0}^{r-m-1} G_{r-m-j} H_j \\
&= F_{i-r} + \sum_{m=0}^{r-1} F_{i-m} H_{r-m} \\
&= \sum_{m=0}^{r} H_{r-m} F_{i-m}\,,
\end{aligned}$$

one has

$$\begin{aligned}
F_k &= z \sum_{m=0}^{a} e_{-m} \sum_{i=0}^{m} H_{m-i} F_{k-i} \\
&= z \sum_{j=0}^{a} F_{k-j} \sum_{i=j}^{a} e_{-i} H_{i-j} \\
&= z F_k \sum_{i=0}^{a} e_{-i} H_i + z \sum_{j=1}^{a} F_{k-j} \sum_{i=j}^{a} e_{-i} H_{i-j}.
\end{aligned}$$

For $k=1$, one gets $F_1 = z + z\sum_{i=0}^{a} e_{-i}({}_{i+1}F_1)$. Let $b_j = \sum_{i=j}^{a} e_{-i}H_{i-j}$, one has

$$\begin{cases} F_1 = \frac{z}{1-zb_0} \\ F_k = \frac{z}{1-zb_0}\sum_{i=1}^{a} zb_i\, F_{k-i}\,, \qquad \text{for} \quad k>1. \end{cases}$$

Let M be the following a by a matrix (whose entries are rational functions in $G_1,\ldots,G_a$),

$$M := \begin{pmatrix} \frac{zb_1}{1-zb_0} & \frac{zb_2}{1-zb_0} & \cdots & \frac{zb_{a-1}}{1-zb_0} & \frac{zb_a}{1-zb_0} \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} F_k \\ F_{k-1} \\ \vdots \\ F_{k-a+1} \end{pmatrix} = M \begin{pmatrix} F_{k-1} \\ F_{k-2} \\ \vdots \\ F_{k-a} \end{pmatrix} = M^{k-1} \begin{pmatrix} F_1 = \frac{z}{1-zb_0} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Before to go on, one needs the following lemma.

Lemma 5 *The rationality of the sequence (e_i) implies the algebraicity of the sequence (G_i).*

Proof. If the sequence $(e_i)_{i\geq -a}$ is rational, then the sequence $(e_k)_{k\geq 1}$ is also rational and there exist two polynomials P and Q such that $\sum_{k\geq 1} e_k z^{k-1} = \frac{P(z)}{Q(z)}$, with $Q(0)\neq 0$. Thus one has $\sum_{k\geq 1} e_k M^{k-1} = P(M)Q(M)^{-1}$, because $Q(M)$ is invertible. Indeed, decomposing $Q(z)$ in $\mathbb{C}$ leads to $Q(z) = c\prod_{i=1}^{deg(Q)}(z-\rho_i)$, so that $\mathrm{Det}(Q(M)) = c\prod_{i=1}^{deg(Q)}\mathrm{Det}(M-\rho_i I)$, which is obviously nonzero by computing,

$$\mathrm{Det}(M-\rho I) = (-1)^{a+1}(-\rho^a + \frac{z}{1-zb_0}\sum_{m=1}^{a} b_m\rho^{a-m}).$$

Thus we can write an algebraic system of a equations for $G_1, \ldots, G_a$,

$$\begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ G_a \end{pmatrix} = \sum_k e_k \begin{pmatrix} F_k \\ F_{k-1} \\ \vdots \\ F_{k-(a-1)} \end{pmatrix} = \sum_k e_k M^{k-1} \begin{pmatrix} \frac{z}{1-zb_0} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \frac{P(M)}{Q(M)} \begin{pmatrix} \frac{z}{1-zb_0} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

The Jacobian of this system is equal to the identity for $z = 0$ so the G_i's are algebraic functions of z. □

Moreover F_k is algebraic for all $k \geq 1$:

$$\begin{pmatrix} F_k \\ F_{k-1} \\ \vdots \\ F_{k-a+1} \end{pmatrix} = M^{k-1} \begin{pmatrix} \frac{z}{1-zb_0} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Finally, this leads to

$$\sum_{k\geq 1} \begin{pmatrix} F_k \\ F_{k-1} \\ \vdots \\ F_{k-a+1} \end{pmatrix} = (M-1)^{-1} \begin{pmatrix} \frac{z}{1-zb_0} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

taking the first entry gives that $F(z,1) = \sum_{k\geq 1} F_k(z)$ is algebraic. □

Remark: In fact Lemma 5 can be extended. Indeed, if $E(z)$ is algebraic, then the G_i's are still algebraic. For this, let P the bivariate polynomial such that $P(E,z) = 0$. Now, consider the first and the third member in last formula in the proof of the Lemma. Multiplying them by adequate monomials $E(M)^i M^j$ and summing over adequate values of (i,j) allows to get $P(E(M),M)$ in the third member. As this is equal to 0, one thus gets a system of a equations for the G_i's (with algebraic coefficients). The rest of the proof still implies the algebraicity of $F(z,1)$. We don't push the proof in this direction because, in Theorem 9 hereafter, we give another proof which leads to a neat closed-form formula for $F(z,u)$.

Theorem 4 above allows us to generalise a result from [3] concerning finite transformations of $(k) \rightsquigarrow (1)\dots(k-1)(k)(k+1)$. A finite transformation of a rule consists in adding a fixed integer to one (resp. all) succession rule(s). The noncommutative formal power series approach allows us to interpret finite transformations and show that they do not change the algebraicity of the generating function. Moreover, the property of algebraicity does not depend on the choice of the axiom.

Theorem 6 *Consider $E(z) = \sum_{i \geq -a} e_i z^i$. If $E(z)$ is algebraic, then all "finite transformations" (as defined above) of the succession rule*

$$(k) \rightsquigarrow (1)^{e_{k-1}} \dots (k-1)^{e_1} (k)^{e_0} \dots (k+a)^{e_{-a}}$$

lead to an algebraic associated generating function $F(z,1)$. More generally, all finite transformations of the succession rule $(k) \rightsquigarrow \mathcal{M}_k$ lead to an algebraic associated generating function $F(z,u)$ as soon as the original bivariate generating function is algebraic.

Proof. For any fixed nonnegative integer c, let $\mathcal{T}$, $\mathcal{T}'$, and $\mathcal{T}''$ be the following the generating trees:

$$\mathcal{T} = \begin{cases} (r) \\ (k) \rightsquigarrow \mathcal{M}_k\,, \end{cases}$$

$$\mathcal{T}' = \begin{cases} (r) \\ (k) \rightsquigarrow \mathcal{M}_k \cup (c)\,, \end{cases}$$

$$\mathcal{T}'' = \begin{cases} (r) \\ (k_0) \rightsquigarrow \mathcal{M}_{k_0} \cup (c) \\ (k) \rightsquigarrow \mathcal{M}_k\,, \qquad \text{for} \quad k \neq k_0\,. \end{cases}$$

Thus, $\mathcal{T}'$ and $\mathcal{T}''$ are finite transformations of $\mathcal{T}$. Let S, S', and S'' (resp. F, F', and F'') be the formal sum of paths (resp. the commutative generating functions) associated to $\mathcal{T}, \mathcal{T}'$, and $\mathcal{T}''$. As in the proofs of Theorem 3 and Theorem 4, let (S_k) be the formal sum of paths ending by k and $({}_cS)$ be the formal sum of the paths in the generating tree $[(c), (k) \rightsquigarrow (1)^{e_{k-1}} \dots (k-1)^{e_1} (k)^{e_0} \dots (k+a)^{e_{-a}}]$, that is the original generating tree $\mathcal{T}$ where the axiom r has been replaced by c.

As $S'' = S + S_{k_0}.({}_cS_{k_0})^*({}_cS)$, this gives $F''(z,u) = F(z,u) + \dfrac{F_{k_0}(z) \quad {}_cF(z,u)}{1 - {}_cF_{k_0}(z)}$ where the right member involves only functions which are known to be algebraic, thus F'' is also algebraic. Similarly, the relation $S' = S.({}_cS)^*$, gives the algebraicity of F'. □

By duality, similar results hold if you *remove* a label from one (or all) rule(s). Note also that there is no difficulty to apply the same kind of proofs to other transformations like $[(r),(k_0) \rightsquigarrow \mathcal{M}_{k_0},(k) \rightsquigarrow \mathcal{M}_k \cup (c)]$.

3.4 Succession rules: polynomial exponents and no negative bounded jump allowed

Theorem 7 *For any constant $B \geq 0$, the generating tree*

$$[(r),(k) \rightsquigarrow (0)^{e(k,0)} \dots (B)^{e(k,B)} \quad (k)^{e_0} \dots (k+a)^{e_{-a}}]$$

(where $e(k,0),\dots,e(k,B)$ are polynomial in k, $e(k,i) = 0$ for $B < i < k$ and $e(k,i) = e_{k-i}$, some fixed constants, for $i \geq k$) has a rational generating function $F(z,u)$.

Proof. First, we illustrate the general case by the following example:

$$\begin{cases} (0) \\ (k) \rightsquigarrow (0)^{k^2}(2)^{3k-1}(3)(k)(k+1)^2(k+3)^5\,, \end{cases} \tag{4}$$

for which $B = 3$, the polynomials in k are $e(k,0) = k^2, e(k,1) = 0, e(k,2) = 3k-1, e(k,3) = 1$, and the fixed constants are $e_0 = 1$, $e_{-1} = 2$, $e_{-2} = 0$, $e_{-3} = 5$.

The part $(k) \rightsquigarrow (0)^{k^2}$ implies a transformation $u^k \rightsquigarrow k^2u^0$. The part $(k) \rightsquigarrow (2)^{3k-1}$ implies a transformation $u^k \rightsquigarrow (3k-1)u^2$. The part $(k) \rightsquigarrow (3)$ implies a transformation $u^k \rightsquigarrow u^3$. It is possible to perform all these transformations using the derivation[4], evaluation in $u = 1$ and multiplication by a monomial: in the first case, the multiplicity k^2 is obtained by $\partial(u\partial(u^k))$ and then evaluating in $u = 1$; for the second case, the multiplicity $3k-1$ is obtained by taking $\partial(u^{3k})/u$ and then evaluating in $u = 1$; for the third case simply evaluate in $u = 1$ and multiply by u^3. The part $(k) \rightsquigarrow (k)(k+1)^2(k+3)^5$ gives $u^k \rightsquigarrow P(u)u^k$ where $P(u) = 1 + 2u + 5u^3$. All these transformations are in fact linear, so to act on u^k or a polynomial in u (like $f_n(u)$) is the same.

[4] We denote the derivation with respect to u by ∂_u or by ∂ or $'$ when there is no ambiguity. We also write abusively $\partial_u F(z,1)$ for $(\partial_u F)(z,1)$.

Finally, evaluating $\partial(u\partial f_n(u))$ in $u=1$ gives $f_n''(1)+f_n'(1)$ and evaluating $u^2\partial_u f_n(u^3)/u$ in $u=1$ gives $u^2(3f_n'(1)-f_n(1))$, so these trivial simplifications gives the following recurrence:

$$f_{n+1}(u) = P(u)f_n(u) + u^0(f_n''(1)+f_n'(1)) + u^2(3f_n'(1)-f_n(1)) + u^3 f_n(1)\,.$$

Multiplying by z^{n+1} and summing for $n \geq 0$ leads to the functional equation

$$(1-zP(u))F(z,u) = 1 + z(u^3-1)F(z,1) + z(3u^2+1)\partial_u F(z,1) + z\partial_u^2 F(z,1)\,.$$

Taking the first 2 derivatives and instantiating in $u=1$ gives a rational system of full rank, hence $F(z,u)$ is rational:

$$F(z,u) = \frac{u^3(22z^2-112z^3-z) + u^2(480z^3-60z^2) + 528z^3 - 250z^2 + 31z - 1}{(1-zP(u))(872z^3-212z^2+30z-1)}\,.$$

For the general case, one has the following functional equation

$$(1-zP(u))F(z,u) = u^r + z\sum_{i=0}^{d} t_i(u)\partial_u^i F(z,1)$$

(d is the largest degree of the polynomials $e(k,i)$, and the t_i's are some Laurent polynomials which can be made explicit). Taking the first d derivatives and instantiating in $u=1$ gives a system (for $m=0,\ldots,d$):

$$\partial_u^m u^r + \left(\sum_{i=0}^{m-1}\left(z\partial_u^m t_i(1) + z\binom{m}{i}\partial_u^{m-i}P(1)\right)\partial_u^i F(z,1)\right)$$
$$+\left(z\partial_u^m t_i(1) - (1-zP(1))\right)\partial_u^m F(z,1) + z\sum_{i=m+1}^{d}\partial_u^m t_i(1)\partial_u^i F(z,1) = 0\,.$$

This gives a matricial equation $M.\overrightarrow{F} = \overrightarrow{v}$ where $\overrightarrow{v} = (u^r,0,\ldots,0)^T$ and $\overrightarrow{F} = (\partial_u^0 F(z,1),\ldots,\partial_u^d F(z,1))^T$. The coefficients of the main diagonal of M are $-1+z\ldots$ (as they are the coefficients of the $\partial_u^m F(z,1)$ summand) and all the other coefficient of M are monomials in z of degree 1. Thus, one has $[z^0]\det M = \pm 1$ and then $\det M \neq 0$. Consequently, this system is of full rank. Solving it gives rational expressions for the $\partial_u^i F(z,1)$ and for $F(z,u)$. $\square$

3.5 Succession rules: polynomial exponents and negative jumps allowed

We now give a generalisation of a result of [3] which was giving the algebraicity of "factorial rules": we allow here initial multiplicities which are not space-homogeneous.

Theorem 8 *For any constant $B \geq 0$, the generating tree*

$$[(r),(k) \rightsquigarrow (0)^{e(k,0)} \dots (B)^{e(k,B)} (B+1) \dots (k-b-1)(k-b)^{e_b} \dots (k+a)^{e_{-a}}]$$

(where $e(k,0), \dots, e(k,B)$ are polynomial in k, $e(k,i) = 1$ for $B < i < k-b$ and $e(k,i) = e_{k-i}$, some fixed constants, for $i \geq k-b$) has an algebraic generating function $F(z,u)$.

Proof. We illustrate the general case by the following example:

$$[(0),(k) \rightsquigarrow (0)^{k^2} (2)^{3k^5-2} (6)(7) \dots (k-5)(k-4)^2 (k-2)^3 (k)(k+3)^2 (k+23)]\,,$$

for which $B = 5, b = 4, a = 23$, the polynomials in k are $e(k,0) = k^2$, $e(k,2) = 3k^5 - 2$, $e(k,1) = e(k,3) = e(k,4) = e(k,5) = 0$ and the fixed constants are $e_4 = 2$, $e_2 = 3$, $e_0 = 1$, $e_{-3} = 2$, $e_{-23} = 1$. One sets $P(u) = 2u^{-4} + 3u^{-2} + 1 + 2u^3 + u^{23}$, the recurrence is

$$f_{n+1}(u) = P(u) f_n(u) - \{u^{<0}\} P(u) f_n(u) + \sum_{i=0}^{5} t_i(u) \partial_u^i f_n(1)\,,$$

where $\{u^{<0}\}$ stands for the sum of the monomials in u with a negative degree. Multiplying by z^{n+1} and summing for $n \geq 0$ leads to the functional equation

$$(1 - zP(u))F(z,u) = 1 - z \sum_{k=0}^{4-1} r_k(u) F_k(z) + z \sum_{i=0}^{5} t_i(u) \partial_u^i F(z,1)\,, \qquad (5)$$

where $r_k(u) := \{u^{<0}\} P(u) u^k$ and $t_i(u)$ are (Laurent) polynomials which can be made explicit.

One can use the kernel method (we refer to [4,9] for recent applications of this method) to solve this equation. We call $1 - zP(u)$ the *kernel* of the equation. Solving $1 - zP(u) = 0$ with respect to u gives 4 roots $u_1(z)$, $u_2(z)$, $u_3(z)$ and $u_4(z)$ which are Puiseux series in $z^{1/4}$ and which tend to zero in 0. There are also 23 others roots which behave like $z^{-1/23}$ around 0, so we call $u_1, \dots, u_4$ the *small* roots of the kernel. Plugging the 4 small roots of the kernel in Equation 5 and considering the 6 other equations obtained by taking the first 5 derivatives of Equation 5 (and then setting $u = 1$) gives a system of full rank with 10 equations with 10 unknown univariate generating functions, which are thus all algebraic, and then one has a formula for $F(z,u)$, involving the u_i, which implies its algebraicity. For the general case, simply replace 4 by b and 5 by d in Equation 5. Then, one can argue as in Theorem 7 above, with a new matricial equation $M.\overrightarrow{F} = \overrightarrow{v}$; looking at the valuation in z of each entries in M (some of them involves the small roots u_i's, but at most a product of b of them) gives $\det M \neq 0$ and thus a system of full rank, so $F(z,u)$ can be expressed as a rational function in z, u, and the small roots u_i's. As these roots are algebraic, $F(z,u)$ is algebraic. □

3.6 Succession rules: "algebraic" exponents

Consider now the case where, for each i, the exponent $e(k,i)$ of the rule (2) is a constant (that is, $e(k,i) := e_{k-i}$ for a fixed sequence $(e_k)_{k\in\mathbb{Z}}$). "Algebraic" exponents means here that the generating function of the e_k's (which are nonnegative integers) is algebraic. How far can we relate the behaviour of the walk

$$[(0),(k) \rightsquigarrow (0)^{e_k} \dots (k-1)^{e_1}(k)^{e_0}(k+1)^{e_{-1}} \dots (k+a)^{e_{-a}}] \tag{6}$$

to the generating function of the exponents $E(u) = \sum_{i\geq -a} e_i u^i$? We give here a first element of answer:

Theorem 9 *Consider the generating tree*

$$[(0),(k) \rightsquigarrow (0)^{e_k}(1)^{e_{k-1}} \dots (k-1)^{e_1}(k)^{e_0} \dots (k+a)^{e_{-a}}]\,. \tag{7}$$

For $a = 1$, one has

$$F(z,u) = \frac{F_0(z)}{1 - u\, e_{-1} z\, F_0(z)} \qquad \text{with} \qquad F_0(z) = \frac{1}{e_{-1} z}\, E^{<-1>}(\frac{1}{z})$$

where $E^{<-1>}$ is the compositional inverse of $E(u)$ and where e_{-1} is the multiplicity of the $+1$ jump. More generally, for $a \geq 1$, the generating function $F(z,u)$ is expressed in terms of the a solutions $u_1(z), \dots, u_a(z)$ of $1 - zE(u) = 0$ which satisfy $u_k(z) \sim e^{2ik\pi/a} e_{-a}^{1/a} z^{1/a}$ for $z \sim 0$:

$$F(z,u) = F_0(z) \prod_{i=1}^{a} \frac{1}{1 - u u_i(z)} = \sum_{k\geq 0} F_0(z) \left(\sum_{i_1+\dots+i_a=k} u_1^{i_1} \dots u_a^{i_a} \right) u^k\,.$$

One has

$$F_0(z) = \frac{(-1)^{a+1}}{z e_{-a}} \prod_{i=1}^{a} u_i(z) \qquad \text{and} \qquad F(z,1) = \frac{-1}{z e_{-a}} \prod_{i=1}^{a} \frac{1}{1 - \frac{1}{u_i(z)}}\,.$$

Consequently, if the generating function of the exponents $E(u)$ is algebraic then the bivariate generating function $F(z,u)$ is algebraic.

Proof. For $a = 1$, the first identity reflects the combinatorial decomposition (one to one correspondence, in fact) "a walk from 0 to $k+1$" is "a walk from 0 to k" then followed by a jump $+1$ then followed by "a walk from $k+1$ to $k+1$ never going below $k+1$". The generating function of these last walks is clearly $F_0(z)$, thus one has $F_{k+1}(z) = F_k(z) e_{-1} z F_0(z) = F_0(z)(z e_{-1} F_0(z))^{k+1}$.

For the walks corresponding to the rule (7), the set of jumps is given by $E(1/u)$; if one reverses the time direction, one gets a new walk where the

set of available jumps is given by $E(u)$. Define $\widetilde{F}(z,u)$ as the corresponding generating function (one starts at altitude 0), one has:

$$\widetilde{f}_{n+1}(u) = \{u^{\geq 0}\}E(u)\widetilde{f}_n(u), \qquad \widetilde{f}_0(u) = 1$$

where $\{u^{\geq 0}\}$ stands for the sum of all monomials in u with a nonnegative degree. Multiplying by z^{n+1} and summing for $n \geq 0$ gives

$$\widetilde{F}(z,u) = \widetilde{f}_0(u) + zE(u)\widetilde{F}(z,u) - z\{u^{-1}\}\frac{e_{-1}}{u}\widetilde{F}(z,u)\,,$$

that one rewrites as the following functional equation

$$(1 - zE(u))\widetilde{F}(z,u) = 1 - z\frac{e_{-1}}{u}\widetilde{F}_0(z)\,.$$

Then solving the "kernel" $1 - zE(u) = 0$ with respect to u gives a series $u_1(z) = E^{<-1>}(1/z)$, which is algebraic as the compositional inverse of an invertible algebraic function is algebraic (simply plug the inverse in the polynomial equation $\Phi(E(u), u) = 0$ satisfied by $E(u)$ to check this fact). Note that E is invertible because $a \geq 1$ implies $E'(0) \neq 0$.

If one then evaluates the above functional equation at $u = u_1(z)$, one gets $0 = 1 - z\frac{e_{-1}}{u_1}\widetilde{F}_0(z)$ and thus $\widetilde{F}_0(z) = \frac{u_1}{e_{-1}z}$. As one has $\widetilde{F}_0(z) = F_0(z)$ (a walk from 0 to 0 from left to right is still a walk from 0 to 0 from right to left), one gets the result from the theorem. Note that if one sets $\widetilde{f}_0(u) = \frac{1}{1-u}$, $\widetilde{F}_0$ enumerates walks from anywhere to 0, so $\widetilde{F}_0(z) = \frac{u_1/(ze_{-1})}{1-u_1} = F(z,1)$, which is coherent with the theorem (case $a = 1$).

For $a \geq 1$, one sets $P(u) := \sum_{i=-a}^{-1} e_i u^i$; one has

$$(1 - zE(u))\widetilde{F}(z,u) = \widetilde{f}_0(u) - z\{u^{<0}\}P(u)\widetilde{F}(z,u)\,.$$

This is rewritten as

$$(1 - zE(u))\widetilde{F}(z,u) = \widetilde{f}_0(u) - z\sum_{k=0}^{a-1} r_k(u)\widetilde{F}_k(z)\,. \tag{8}$$

where $r_k(u) := \{u^{<0}\}P(u)u^k$ is a Laurent polynomial with monomials of degree going from -1 down to $k - a$.

$E(u)$ being algebraic, there exists a bivariate polynomial $P \in \mathbb{Q}[E,u]$ such that $P(E,u) = 0$. Now, as one has the kernel equation $1 - zE(u) = 0$, it means that the roots $u_i(z)$ of the kernel are algebraic and satisfy $P(\frac{1}{z}, u_i(z)) = 0$. The classical theory of Newton polygon then gives the Puiseux expansion of these roots. Among these roots, the kernel equation $1 - zE(u) = 0$ has a

roots $u_1(z), \ldots, u_a(z)$ which are Puiseux series in $z^{1/a}$ and which tend to 0 when z tends to 0. When $\widetilde{f}_0(u) = 1$, plugging these roots in the functional equation shows that they correspond to the a roots of the polynomial $u^a - zu^a \sum_{k=0}^{a-1} r_k(u) F_k(z)$, whose leading term is u^a and whose constant term is $-ze_{-a}\widetilde{F}_0(z)$. This gives $\widetilde{F}_0(z) = \frac{-\prod_{i=1}^{a} u_i}{-ze_{-a}}$. When $\widetilde{f}_0(u) = \frac{1}{1-u}$, this gives a system of a equations for a unknowns (the $\widetilde{F}_k$'s). Solving it for $\widetilde{F}_0$ gives $F(z,1)$. Solving the $\widetilde{F}_0$ for $\widetilde{f}_0(u) = u^k$ gives the $F_k(z)$. This concludes Theorem 9. $\square$

Remark: as D-finite functions are not necessarily closed under compositional inverse, it is not true that if $E(u)$ is D-finite, then $F(z,1)$ or $F_0(z)$ (and *a fortiori* $F(z,u)$) are D-finite, even in the case $a = 1$.

For $a = 1$, the Riordan arrays approach that we presented in Subsection 2.3 also gives the algebraicity of $F(z,u)$. In fact, a theorem from [22] says:

Theorem 10 *If $(a_j)_{j\in\mathbb{N}}$ and $(z_j)_{j\in\mathbb{N}}$ are two nonnegative integer sequences, with $a_0 \neq 0$, then the matrix associated to the generating tree*

$$\begin{cases} (r) \\ (k) \rightsquigarrow (r)^{z_{k-r}}(r+1)^{a_{k-r}}(r+2)^{a_{k-r-1}} \ldots (k+1)^{a_0} \end{cases} \tag{9}$$

is a proper Riordan Array D defined by the triple (d_0, A, Z), such that

$$d_0 = 1, \quad A = (a_0, a_1, a_2, \ldots), \quad Z = (z_0, z_1, z_2, \ldots).$$

Accordingly, this gives

$$F(z,u) = \frac{d(z)}{1 - uzh(z)} \text{ where } h(z) = A(zh(z)) \text{ and } d(z) = 1/(1 - zZ(zh(z))) .$$

For $a > 1$, the matrix associated (see Section 2) to the rule (6) is called a *horizontally stretched Riordan array*. The algebraicity of the corresponding generating function $F(z,u)$ then depends on the algebraicity of $A(z) = \sum_{k\geq 0} a_k z^k$ and $F_0(z), \ldots, F_{a-1}(z)$ (the generating functions of the first a columns of the matrix). However, while the theory of Riordan arrays has been intensively studied, the theory of stretched Riordan arrays, from a generating function point of view, is still in progress.

We end with a last application of the kernel method.

Theorem 11 *Consider the succession rule (6) when the e_i's are ultimately constants (say, equal to a constant C after rank b):*

$$[(0), (k) \rightsquigarrow (0)^C \ldots (k-b-1)^C (k-b)^{e_b} \ldots (k)^{e_0} \ldots (k+a)^{e_{-a}}] .$$

Then $F(z,u)$ is algebraic and satisfies

$$F(z,u) = \frac{\prod_{i=0}^{b} u - u_i(z)}{K(z,u)},$$

where the u_i's and K are defined as below.

Proof. One has the recurrence $f_{n+1}(u) = C\frac{f_n(u)-f_n(1)}{u-1} + P(u)f_n(u)$ this leads to the functional equation

$$\left(1 - zP(u) - z\frac{C}{u-1}\right) F(z,u) \quad =$$

$$1 - \frac{zC}{u-1}F(z,1) - z\sum_{k=0}^{b-1}\{u^{<0}\}P(u)u^k F_k(z)\,, \tag{10}$$

where $P(u) = \sum_{i=1}^{b}(e_i - C)\frac{1}{u^i} + \sum_{i=0}^{a} e_{-i}u^i$. Define the kernel K as $K(u,z) = u^b(1-u)(1 - zP(u) - \frac{zC}{u-1})$. It has b roots $u_1(z), \ldots, u_b(z)$ which are Puiseux series in $z^{1/b}$ and which tend to 0 in 0 and one root $u_0(z)$ which tends to 1 in 0. These are exactly the $b+1$ roots of the right hand part of (10) (once multiplied by $(1-u)u^b$). So $F(z,u) = \frac{\prod_{i=0}^{b} u-u_i(z)}{K(z,u)}$, where the u_i's are the $b+1$ small roots of the kernel. □

3.7 Asymptotics

Given a particular rule for Theorems 7, 8, 9, 10 or 11, it is possible to find an asymptotic expansion for the number of walks. It is not really possible to merge all these results in a single one, as the rules are too unconstrained. However, for the algebraic case, a kind of universality holds for the behaviour of the roots of the kernel. This leads to following theorem, which has to be adapted case by case for rules of Theorems 8 and 9 (and is easily applied to rules of Theorem 11).

Theorem 12 *The number of walks of length n for the "factorial" rule*

$$[(0),(k) \rightsquigarrow (0)(1)\ldots(k-b-1)(k-b)^{e_b}\ldots(k)^{e_0}\ldots(k+a)^{e_{-a}}]$$

(where $e(k,i) = 1$ for $0 \le i < k-b$ and $e(k,i) = e_{k-i}$, some fixed constants, for $i \ge k-b$) has the following asymptotics $A\frac{\rho^{-n}}{\sqrt{2\pi n^3}}$, where A and ρ are algebraic constants depending on the finite multiset of jumps $\mathcal{P} = \{-b,\ldots,+a\}$.

Proof. See [2] for a proof and applications to the limit laws of final altitude and number of factors. The approach is similar to the one used for walks with a finite number of jumps but there are some complications due to the fact that the kernel is now of the kind $1 - z\phi(u)$ where $\phi(u)$ is not unimodal. One can however establish that the real positive root u_0 now dominates and has a square-root behaviour. □

This result is the first step towards limit laws of several parameters (like final altitude, local time, ...). It would be interesting (but much more difficult) to get the asymptotics of parameter like height and area. Note that for these parameters, it is possible to get closed-form formulae (particularly in the case $a = 1$ for the area and for any value of a for the height, via the kernel method, see [1])... but this is another story!

3.8 Algebraic equations

In Theorems 3 and 4, we gave a direct way to obtain an algebraic equation satisfied by $F(z, 1)$ when the generating function of the exponents is rational. For the other theorems, as the algebraic generating function $F(z, u)$ is expressed in terms of the roots of the kernel, it is possible to get an algebraic equation for $F(z, u)$ via resultant or Gröbner bases computations. Note that a more efficient way, the so-called Platypus algorithm, is presented in [4]. It also relies on an exploitation of the roots of the kernel.

3.9 Variations...

As a first variation, it is possible to play with the root r of the tree (the starting point of the paths). We gave above results mostly for $r = 0$ or $r = 1$, but it is also possible to follow our proofs for other values of r.

As a next variation, it is also possible to remove the non-negativity constraint. In this case, the walks are on $\mathbb{Z}$ and thus one gets directly $F(z, u) = \frac{1}{1-zE(u^{-1})}$. If one then considers walks ending at a given altitude k, it is possible to get a closed-form formula for their generating function $F_k(z)$ (which is algebraic), via residue computation and a conjugacy principle (simply follow the same proofs as in [2,4] and get Spitzer-like formulae and closed-form formulae still involving the roots of the kernel).

As a third variation, in Rule (2), it also possible to consider exponents $e(k, i)$ from $\mathbb{Z}^2$ to $\mathbb{Z}$, even if the combinatorial meaning of a "negative multiplicity" is not clear... It is also possible to consider the case $e(k, i) = e_i$ (instead of $e(k, i) = e_{k-i}$ as we did in this article). The sequences increase very quickly, it is then then natural to look for exponential generating functions. Some nice formulae were given in [1,3] and combinatorial proofs were given in [11].

As a last variation, it is also possible (see [1]) to reconsider all the above results for walks of higher Markovian order, that is for walks for which $f_{n+1}(u)$ depends not only on $f_n(u)$ but also on $f_{n-1}(u)$, and on finitely many other f_n's. Here again, our approaches are still working. For example, with a Markovian random walk of order 2 (positions one step before are involved with multiplicities encoded by $E(u)$, and positions two steps before are involved with multiplicities encoded by $E_2(u)$), formulae roughly involves something like $\frac{A}{1-zE(u)-z^2E_2(u)}$ instead of $\frac{A}{1-zE(u)}$. There are already combinatorial interests for such walks, see [18].

4 Examples

We now give a series of examples from combinatorics or computer science in which succession rules studied in Section 3 appear.

EXAMPLE 1. *Fully directed compact animals.*

They are also called *Diagonally directed convex polyominoes* (see [16] and Fig. 6). They are known to be counted according to their number of diagonals by $\frac{1}{2n+1}\binom{3n}{n}$ which corresponds to the the generating tree $[(1),(k) \rightsquigarrow (1)^{k+1}(2)^{k}\dots(k-1)^{3}(k)^{2}(k+1)]$. □

EXAMPLE 2. *A new generating tree for Catalan numbers.*

From [24] (see the exercise on Catalan numbers pp.221-247), the generating tree

$$[(1),(k) \rightsquigarrow (1)^{2^{k-2}}(2)^{2^{k-3}}\dots(k-2)^{2}(k-1)(k+1)]$$

generates the partition $\{B_1,\dots,B_p\}$ of $[n]$ such that the numbers $1,2,\dots,n$ are arranged in order around a circle, then the convex hulls of the blocks $B_1,\dots,B_p$ are pairwise disjoint. Indeed, let k be the number of isolated points around 1. The 2^{k-1} successors of this configuration are obtained by taking all the subset of $\{\alpha_1=1,\alpha_2,\dots,\alpha_k,n+1\}$ containing $n+1$. □

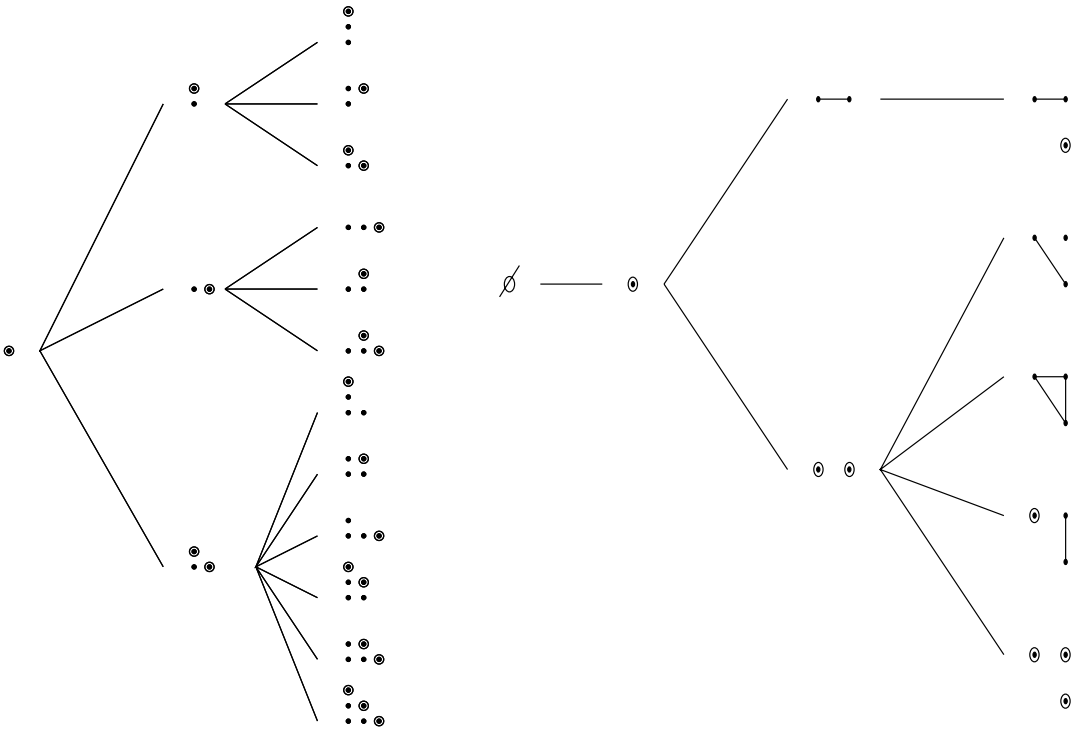


Fig. 6. Generating trees for (fully directed compact) animals and Catalan blocks.

EXAMPLE 3. *Two families of rules leading to an algebraic generating function.*

For the rule $[(0),(k) \rightsquigarrow (0)^{e_k}(1)^{e_{k-1}}\dots(k-1)^{e_1}(k)^{e_0}(k+1)]$, where e_k for $k \geq 0$ is the number of t-ary trees with k nodes, $F(z,u)$ satisfies a algebraic equation of degree t. E.g., for $t=3$, one has:
$1-(3+(4-3u)z)F(z,u)-(-3+(6u-7)z+(-3u^2+8u-3)z^2)F(z,u)^2-(1+(3-3u)z+(3u^2-7u+3)z^2+(-u^3+4u^2-3u+1)z^3)F(z,u)^3=0.$

For the rule $[(0),(k) \rightsquigarrow (0)^{c+k}(1)^{c+k-1}\dots(k-2)^{c+2}(k-1)^{c+1}(k)^c(k+1)]$, $F(z,u)$ satisfies an algebraic equation of degree 3:
$((1-2u)z^2+(c-(c+1)+2u^2))F^3+((u-2)z+(-c-2+4u-2u^2)z^2)F^2+(1+(2-2u)z)F=1.$

Similar examples for $a>1$ lead to expressions which are perhaps a bit large to be written here *in extenso*. However, the reader interested by such examples can have a look at `http://algo.inria.fr/banderier/Papers/dm03.mws`. This is a Maple worksheet where we get the equations for $F(z,u)$, plot the roots of the kernel, give the asymptotics for different kind of walks. □

EXAMPLE 4. *Tennis ball problem.*

Let $s \geq 2$ be an integer and consider the following problem known as *the s-tennis ball problem*. At the first turn one is given balls numbered 1 to s. One throws one of them out of the window onto the lawn. At the second turn balls numbered $s+1$ through $2s$ are brought in and now one throws out on the lawn any of the $2s-1$ remained. Then balls $2s+1$ through $3s$ are brought in and one throws out one of the $3s-2$ available balls. The game continues for n turns. At this point, one picks up the n balls in the lawn and consider the ordered sequence $B=(b_1,b_2,\dots,b_n)$ with $b_1<b_2<\dots<b_n$. This sequence is called a *tennis ball s-sequence* and the first question is: how many tennis ball s-sequences of length n exist? The second question is: what is the sum of all the balls in all the possible s-sequences of length n ? Obviously, if we answer to both these questions, we also know the average sum of the balls in an s-sequence of length n. The general case $s \geq 1$ has been studied in [21] from a generating function viewpoint. In fact, the authors consider an infinite tree with root 0 and with s children. Each $(n+1)$-length path in this tree corresponds to an s-sequence of length n. This infinite tree is isomorphic to the generating tree with specification $[(1),(k) \rightsquigarrow (1)\dots(k+s-2)(k+s-1)]$.

By using this result the authors find that the number of tennis ball s-sequences of length n are counted by T_{n+1}, where $T_n=\frac{1}{1+(s-1)n}\binom{sn}{n}$ (the number of s-ary trees with n-nodes) and the cumulative sum of all the balls thrown onto the

lawn in n turn is

$$\Sigma_n = \frac{1}{2}(sn^2 + (3s-1)n + 2s)T_{n+1} - \frac{1}{2}\sum_{k=0}^{n+1}\binom{sk}{k}\binom{s(n+1-k)}{n+1-k}.$$

□

EXAMPLE 5. A new succession rule for $(4,2)$-tennis ball problem.

The problem of balls on the lawn admits many other variants. For example, one could be supplied with s balls at each turn but now throw out t balls at a time with $t < s$. The general (s,t) case is an open problem while the $(4,2)$ case has been treated in [21], where the authors study the problem by introducing a bilabelled generating tree technique. Anyway, recently Merlini and Sprugnoli found that the problem can be expressed by the rule (6) with $e_i = i+3$ and $a = 2$, namely:

$$[(0),(k) \rightsquigarrow (0)^{k+3}(1)^{k+2}(2)^{k+1}\ldots(k+2)] \tag{11}$$

In fact, if we don't care of the order of the balls thrown away, so that the configuration $(1,4)$, $(5,8)$, $(2,10)$ is considered to be the same as $(1,2)$, $(4,5)$, $(8,10)$, it can be proved that the number of $(4,2)$-sequences of length $2n$ in which the last-but-one element is $2n+k-1$ corresponds to the number of nodes with label k at level n in the generating tree of Figure 7 (for example, the possible sequences of length 2 are $(1,2)$, $(1,3)$, $(1,4)$, $(2,3)$, $(2,4)$ and $(3,4)$). □

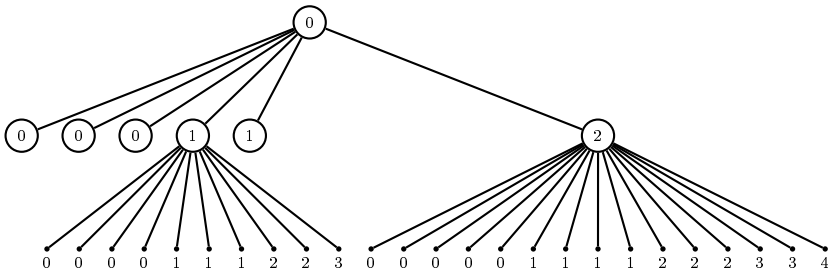


Fig. 7. The partial generating tree for the specification (11).

EXAMPLE 6. *Printers.*

In [20] the authors present a combinatorial model for studying the characteristics of job scheduling in a slow device, for example a printer in a local network. The policy usually adopted by spooling systems is called *First Come First Served* (FCFS) and can be realised by queueing the processes according to their arrival time and by using a FIFO algorithm. A job (printing a file) consists in a finite number of *actions* (printing-out a single page). Each action takes constant time to be performed (a *time slot*). If we fix n time slots, and suppose that at the end of the period the queue becomes empty, while it was

never empty before, the successive states of the jobs queue can be described by a combinatorial structure called *labelled 1-histograms*. A 1-*histogram* of length n is a histogram whose last column only contains 1 cell and, whenever a column is composed by k cells, then the next column contains at least $k-1$ cells. It is at all obvious that a 1-histogram corresponds to a path in the generating tree produced by the specification $[(1),(k) \rightsquigarrow (1)\dots(k+1)]$. A *labelled* 1-*histograms* of length n is a 1-histogram in which we label each cell according to some rules (see [20] for the details). Figure 8 illustrates the possible schedules for two particular 1-histograms of length 3: the first one, for example, corresponds to i) a first job which consists in printing two pages and a second job, which starts at time slot 2, and corresponds to printing a page at time slot 3, and ii) three different jobs which consist in printing a single page, the first at time slot 1, the second at time slot 2 and the third at time slot 3, after queueing at time slot 2. It can be proved that the number of schedules of length n with k jobs request at the first time slot corresponds to the number of nodes at level n having label $k+1$ in the generating tree with specification:

$$[(1),(k) \rightsquigarrow (1)^2 \dots (k)^2(k+1)]\,.$$

This gives that the number S_n of possible schedules corresponds to the n^{th} small Schröder number, that is, the generating function for S_n is $(1-3z-\sqrt{1-6z+z^2})/(4z)$. □

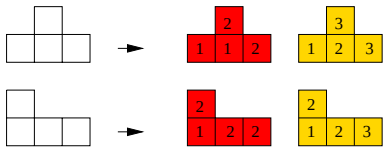


Fig. 8. The schedules corresponding to two particular 1-histograms.

Acknowledgements. Cyril Banderier's work was partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT), by the INRIA postdoctoral programme and by the Max-Planck Institut. He is now supported by the CNRS. During their invitations in Florence, the three French authors also benefited of pasta. Finally, the four authors are grateful to anonymous referees (from the FPSAC'02 conference) for their constructive comments on former versions [5,15] of this article.

Rule	EIS description	Generating Function $F(z,u)$
$(0),(k) \rightsquigarrow (0)^k(k+1)$	F_0, $F(z,1)$: powers of 2	$\dfrac{1-2z-z^2}{1-(u+2)z-2uz^2}$
$(0),(k) \rightsquigarrow (0)^{2k}(k+1)$	$F(z,1)$:A001333 continued fraction convergents to $\sqrt{2}$ F_0: A052542 (ECS)	$\dfrac{1-2z+z^2}{1-(u+2)z+(2u-1)z^2+uz^3}$
$(0),(k) \rightsquigarrow (0)^{3k}(k+1)$	$F(z,1)$: A026150 (ECS)	$\dfrac{1-2z+z^2}{1-(u+2)z+(2u-2)z^2+2uz^3}$
$(0),(k) \rightsquigarrow (0)^{4k}(k+1)$	$F(z,1)$: A046717 half of 3^n	$\dfrac{1-2z+z^2}{1-(u+2)z+(2u-3)z^2+3uz^3}$
$(0),(k) \rightsquigarrow (0)^k(k+1)(k+2)$	$F(z,1)$: A001075 and F_0: A005320 Pell's equation	$\dfrac{1-4z+4z^2}{1-(4+u+u^2)z+(4u^2+u-1)z^2-\ldots}$
$(1),(k) \rightsquigarrow (0)(1)^2(k)(k+2)^2(k+3)^5$	6^n and A003464 $(6^n-1)/5$	$\dfrac{(4u-1)z-u}{(1-6z)((2u^2+1)z-1)}$
$(0),(k) \rightsquigarrow (0)^{k^2}(2)^{3k-1}(3)(k)(k+1)^2(k+3)^5$		see Theorem 7

Table 1
Some succession rules leading to rational generating functions. The generating functions $F(z,1)$ and $F_0(z)$ are defined as in Equation 1.

Rule	EIS description	Generating Function $F(z,u)$
$(1),(k) \rightsquigarrow (1)\dots(k+s-2)(k+s-1)$	$F(z,1)$: s-ary trees	See also Ex. 4
$(1),(k) \rightsquigarrow (1)^2\dots(k)^2(k+1)$	$F(z,1)$: A001003 Schröder's second problem	$\frac{u}{2}\frac{1-(2u+1)z-\sqrt{1-6z-z^2}}{(1-u)z+(u^2+u)z^2}$ (see also Ex. 6)
$(0),(k) \rightsquigarrow (0)^{k^2}(2)^{3k-1}(3)(k-1)(k)(k+1)^2(k+3)^5$		see Theorem 8
$(0),(k) \rightsquigarrow (0)^k(1)^{k-1}\dots(k-1)^1(k)^0(k+1)$	A036765 $F(z,1)$: rooted trees with a degree constraint	equation of degree 3
$(0),(k) \rightsquigarrow (0)^{k+2}(1)^{k+1}\dots(k-1)^3(k)^2(k+1)$	F_0: A006013 A046648 noncrossing trees on a circle $F(z,1)$: A001764 ternary trees	equation of degree 3 (see Ex. 1 for a variant)
$(0),(k) \rightsquigarrow (0)^{k+3}\dots(k-1)^4(k)^3(k+1)^2(k+2)$	$F(z,1)$: A066357 planar trees with root parity constraint	equation of degree 4 (see also Ex. 5)
$(0),(k) \rightsquigarrow (0)^{C_k}\dots(k-1)^{C_1}(k)^{C_0}(k+1)$ (where C_k is the k-th Catalan number)	F_0: A006318 large Schröder numbers	$\frac{1}{2}\frac{3-(4u+1)z-\sqrt{1-6z-z^2}}{1-3uz+(2u^2+u)z^2}$
$(0),(k) \rightsquigarrow (0)^{C_k}\dots(k-1)^{C_1}(k+1)$	F_0: A052705 (ECS)	$\frac{1}{2}\frac{3-(4u+2)z-\sqrt{1-4z-4z^2}}{1-(3u+2)z+(2u^2-2u+1)z^2}$
$(0),(k) \rightsquigarrow (0)^{T_k}\dots(k-1)^{T_1}(k)^{T_0}(k+1)\}$ (where T_k is the k-th tri-Catalan number)	F_0: A054727 noncrossing forests of rooted trees	equation of degree 3 (see Ex. 3)

Table 2
Some succession rules leading to algebraic generating functions.

References

[1] Cyril Banderier. *Combinatoire analytique des chemins et des cartes*. PhD thesis, Université de Paris 6, 2001.

[2] Cyril Banderier. Limit laws for basic parameters of lattice paths with unbounded jumps. In *Mathematics and computer science (Versailles, 2002)*, pages 33–47, Basel, 2002. Birkhäuser.

[3] Cyril Banderier, Mireille Bousquet-Mélou, Alain Denise, Philippe Flajolet, Danièle Gardy, and Dominique Gouyou-Beauchamps. Generating functions for generating trees. *Discrete Mathematics*, 246(1-3):29–55, 2002. Formal power series and algebraic combinatorics (Barcelona, 1999).

[4] Cyril Banderier and Philippe Flajolet. Basic analytic combinatorics of directed lattice paths. *Theoretical Computer Science*, 281(1-2):37–80, 2002. Selected papers in honour of Maurice Nivat.

[5] Cyril Banderier and Donatella Merlini. Lattice paths with an infinite set of jumps. In *Formal power series and algebraic combinatorics (Melbourne, 2002)*, July 2002.

[6] Elena Barcucci, Alberto Del Lungo, Elisa Pergola, and Renzo Pinzani. A methodology for plane tree enumeration. *Discrete Mathematics*, 180(1-3):45–64, 1998.

[7] Elena Barcucci, Alberto Del Lungo, Elisa Pergola, and Renzo Pinzani. ECO: a methodology for the enumeration of combinatorial objects. *Journal of Difference Equations and Applications*, 5(4-5):435–490, 1999.

[8] Elena Barcucci, Alberto Del Lungo, Elisa Pergola, and Renzo Pinzani. From Motzkin to Catalan permutations. *Discrete Mathematics*, 217(1-3):33–49, 2000. Formal power series and algebraic combinatorics (Vienna, 1997).

[9] Mireille Bousquet-Mélou. Walks on the slit plane: other approaches. *Advances in Applied Mathematics*, 27(2-3):243–288, 2001.

[10] F. R. K. Chung, R. L. Graham, V. E. Hoggatt, and M. Kleiman. The number of Baxter permutations. *Journal of Combinatorial Theory, Series A*, 24:382–394, 1978.

[11] Sylvie Corteel. Séries génératrices exponentielles pour les eco-systèmes signés. In *Proceedings of the 12-th International Conference on Formal Power Series and Algebraic Combinatorics*. Springer, June 2000.

[12] S. Dulucq, S. Gire, and O. Guibert. A combinatorial proof of J. West's conjecture. *Discrete Mathematics*, 187(1-3):71–96, 1998.

[13] S. Dulucq, S. Gire, O. Guibert, and J. West. Énumération de permutations à motifs exclus. In *Séminaire Lotharingien de Combinatoire (Gerolfingen, 1993)*, volume 1993/34 of *Prépubl. Inst. Rech. Math. Av.*, pages 19–28. Univ. Louis Pasteur, Strasbourg, 1993.

[14] S. Dulucq, S. Gire, and J. West. Permutations with forbidden subsequences and nonseparable planar maps. *Discrete Mathematics*, 153(1-3):85–103, 1996. Proceedings of the 5th Conference on Formal Power Series and Algebraic Combinatorics (Florence, 1993).

[15] Jean-Marc Fédou and Christine Garcia. Algebraic succession rules. In *Formal power series and algebraic combinatorics (Melbourne, 2002)*, July 2002.

[16] Svjetlan Feretić. A q-enumeration of directed diagonally convex polyominoes. *Discrete Mathematics*, 246(1-3):99–109, 2002. Formal power series and algebraic combinatorics (Barcelona, 1999).

[17] Luca Ferrari, Elisa Pergola, Renzo Pinzani, and Simone Rinaldi. An algebraic characterization of the set of succession rules. *Theoretical Computer Science*, 281(1-2):351–367, 2002. Selected papers in honour of Maurice Nivat.

[18] Luca Ferrari and Renzo Pinzani. A linear operator approach to succession rules. *Linear Algebra and its Applications*, 348:231–246, 2002.

[19] D. Merlini, D. G. Rogers, R. Sprugnoli, and M. C. Verri. On some alternative characterizations of Riordan arrays. *Canadian Journal of Mathematics*, 49(2):301–320, 1997.

[20] D. Merlini, R. Sprugnoli, and M. C. Verri. Waiting patterns for a printer. In *FUN with algorithms 2, E. Lodi, L. Pagli, N. Santoro, Editors*, Carleton Scientific, pages 183–198, 2001. Extended version to appear in Discrete Applied Mathematics.

[21] D. Merlini, R. Sprugnoli, and M. C. Verri. The tennis ball problem. *Journal of Combinatorial Theory. Series A*, 99(2):307–344, 2002.

[22] D. Merlini and M. C. Verri. Generating trees and proper Riordan Arrays. *Discrete Mathematics*, 218:167–183, 2000.

[23] L. W. Shapiro, S. Getu, W.-J. Woan, and L. Woodson. The Riordan group. *Discrete Applied Mathematics*, 34:229–239, 1991.

[24] Richard P. Stanley. *Enumerative combinatorics. Vol. 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999. With a foreword by Gian-Carlo Rota and appendix 1 by Sergey Fomin.

[25] Julian West. *Permutations with forbidden subsequences and stack-sortable permutations*. PhD thesis, MIT, Cambridge, MA, 1990.

DIVISIBILITY OF AN F-L TYPE CONVOLUTION

Michael Wiemann and Curtis Cooper
Department of Mathematics and Computer Science
Central Missouri State University
Warrensburg, MO 64093-5045
email: mwiemann@home.com and cnc8851@cmsu2.cmsu.edu

1. Motivation

Sometimes when working on one problem, another problem and solution are found. The divisibility result in this paper is a consequence of attempts to prove some conjectures of Melham [9] related to the sum

$$L_1 L_3 \cdots L_{2m+1} \sum_{k=1}^{n} F_{2k}^{2m+1},$$

where m is a nonnegative integer and n is a positive integer. Here, we use the usual notation for Fibonacci and Lucas numbers, i.e.

$$F_0 = 0, \quad F_1 = 1, \quad \text{and} \quad F_n = F_{n-1} + F_{n-2}, \quad \text{for} \quad n \geq 2$$

and

$$L_0 = 2, \quad L_1 = 1, \quad \text{and} \quad L_n = L_{n-1} + L_{n-2}, \quad \text{for} \quad n \geq 2.$$

When $m = 2$, Melham found that

$$L_1 L_3 L_5 \sum_{k=1}^{n} F_{2k}^5 = 4F_{2n+1}^5 - 15F_{2n+1}^3 + 25F_{2n+1} - 14.$$

To prove this result we will use the identity

$$F_m^5 = \frac{1}{25}\Big(F_{5m} - 5(-1)^m F_{3m} + 10F_m \Big)$$

(proved using Binet's formula), a result by Melham [9] that if m is an odd integer

$$L_m \sum_{k=1}^{n} F_{2mk} = F_{m(2n+1)} - F_m$$

(proved using Binet's formula and summing the resulting geometric series), and the well-known identities [6]

$$F_{5n} = 25F_n^5 + 25(-1)^n F_n^3 + 5F_n \quad \text{and} \quad F_{3n} = 5F_n^3 + 3(-1)^n F_n.$$

Substituting these in turn into our sum we obtain

$$\begin{aligned}
L_1L_3L_5 \sum_{k=1}^{n} F_{2k}^5 &= L_1L_3L_5 \sum_{k=1}^{n} \frac{1}{25}\left(F_{10k} - 5F_{6k} + 10F_{2k}\right) \\
&= \frac{1}{25} L_1L_3L_5 \left(\sum_{k=1}^{n} F_{10k} - 5\sum_{k=1}^{n} F_{6k} + 10\sum_{k=1}^{n} F_{2k} \right) \\
&= \frac{1}{25}\big(L_1L_3(F_{10n+5} - F_5) - 5L_1L_5(F_{6n+3} - F_3) + 10L_3L_5(F_{2n+1} - F_1)\big) \\
&= \frac{1}{25}\big(L_1L_3F_{10n+5} - L_1L_3F_5 - 5L_1L_5F_{6n+3} + 5L_1F_3L_5 \\
&\quad + 10L_3L_5F_{2n+1} - 10F_1L_3L_5\big) \\
&= \frac{1}{25}\big(L_1L_3(25F_{2n+1}^5 - 25F_{2n+1}^3 + 5F_{2n+1}) - L_1L_3F_5 \\
&\quad - 5L_1L_5(5F_{2n+1}^3 - 3F_{2n+1}) + 5L_1F_3L_5 + 10L_3L_5(F_{2n+1}) - 10F_1L_3L_5\big) \\
&= (L_1L_3)F_{2n+1}^5 - (L_1L_3 + L_1L_5)F_{2n+1}^3 \\
&\quad + \frac{L_1L_3 + 3L_1L_5 + 2L_3L_5}{5} F_{2n+1} - \frac{L_1L_3F_5 - 5L_1F_3L_5 + 10F_1L_3L_5}{25} \\
&= 4F_{2n+1}^5 - 15F_{2n+1}^3 + 25F_{2n+1} - 14.
\end{aligned}$$

In the last step, we note that

$$25 \big| L_1L_3F_5 - 5L_1F_3L_5 + 10F_1L_3L_5. \tag{1}$$

Here, | means divides. This paper will generalize (1).

2. History and Result

Divisibility of Fibonacci and Lucas numbers has been the topic of much research in the mathematical literature. Some well-known divisibility properties of Fibonacci

numbers and Lucas numbers can be found in [3]. For example,

$$F_n|F_m \text{ if and only if } m = kn;$$

$$L_n|F_m \text{ if and only if } m = 2kn, \quad n > 1;$$

$$\text{and} \quad L_n|L_m \text{ if and only if } m = (2k-1)n, \quad n > 1.$$

In [8], Matijasevič proved that

$$F_m^2 \big| F_{mr} \text{ if and only if } F_m \big| r.$$

Later, Hoggatt and Bicknell-Johnson [5] extended these results. In [4], Hoggatt and Bergum discovered a number of interesting results. For example, they proved that

$$n = 2 \cdot 3^k \text{ and } k \geq 1 \text{ implies } n|L_n.$$

They also showed that

$$p \text{ is an odd prime and } p|F_n \text{ implies } p^k|F_{np^{k-1}} \text{ for all } k \geq 1.$$

A corollary to this last result is the fact that

$$5^k \big| F_{5^k} \text{ for } k \geq 1.$$

In this paper we will prove the following theorem.

Theorem. Let n be a nonnegative integer. Then

$$5^n \left| L_1 L_3 \cdots L_{2n+1} \sum_{i=0}^{n} \binom{2n+1}{n-i} (-1)^{n-i} \frac{F_{2i+1}}{L_{2i+1}}. \right. \tag{2}$$

3. Lemmas

To prove our theorem we will need several lemmas. Some of these lemmas involve the quantity

$$a_{pj} = (-1)^j \sum_{k=j}^{p} (-1)^k 2^{p-k} \binom{p+1}{k+1} \binom{k}{j}, \tag{3}$$

where p and j are positive integers and $1 \leq j \leq p$. If we list the first few values of a_{pj} we have

$$
\begin{array}{cccccccccccc}
1 & & & & & & & & & & & \\
4 & 1 & & & & & & & & & & \\
11 & 5 & 1 & & & & & & & & & \\
26 & 16 & 6 & 1 & & & & & & & & \\
57 & 42 & 22 & 7 & 1 & & & & & & & \\
120 & 99 & 64 & 29 & 8 & 1 & & & & & & \\
247 & 219 & 163 & 93 & 37 & 9 & 1 & & & & & \\
502 & 466 & 382 & 256 & 130 & 46 & 10 & 1 & & & & \\
1013 & 968 & 848 & 638 & 386 & 176 & 56 & 11 & 1 & & & \\
2036 & 1981 & 1816 & 1486 & 1024 & 562 & 232 & 67 & 12 & 1 & & \\
4083 & 4017 & 3797 & 3302 & 2510 & 1586 & 794 & 299 & 79 & 13 & 1 & .
\end{array}
$$

This array is part of the sequence A008949 and can be found in [10]. Another notation we will use is $\left\langle \right\rangle$. This will denote an Eulerian number [2].

Lemma 1. Let p be a positive integer. Then

$$a_{p1} = \left\langle \begin{matrix} p+1 \\ 1 \end{matrix} \right\rangle.$$

Lemma 2. Let p and j be positive integers and let $1 \leq j \leq p$. Then

$$a_{pj} = \sum_{0 \leq i \leq p-j} \binom{p+1}{i}.$$

Lemma 3. Let n and k be positive integers with $n > k$. Then

$$\sum_{i=0}^{n} \binom{2n+1}{i} (-1)^i (2n-2i+1)^{2k+1} = 0.$$

Lemma 4. Let p and j be positive integers and $1 \leq j \leq p+1$. Then

$$a_{p+1,j} - \binom{p+1}{j} = 2a_{pj}.$$

Here we adopt the convention that $a_{p,p+1} = 0$.

Lemma 5. Let k and p be positive integers with $p \geq 2k$. Then

$$\sum_{j=1}^{p} (-1)^j a_{pj} j^{2k} = 0.$$

4. Proof of Lemma 1

The proof is by induction on p.

Base Step. Since

$$\begin{aligned} a_{11} &= (-1)^1 \sum_{k=1}^{1} (-1)^k 2^{1-k} \binom{2}{k+1} \binom{k}{1} \\ &= (-1)^1 (-1)^1 2^{1-1} \binom{2}{2} \binom{1}{1} = 1 \end{aligned}$$

and

$$\left\langle {2 \atop 1} \right\rangle = 1,$$

the result is true for $p = 1$.

Induction Step. Assume the result is true for some positive integer p. Then by properties of binomial coefficients, the induction hypothesis, and a recurrence relation for Eulerian numbers, we have

$$\begin{aligned} a_{p+1,1} &= -\sum_{k=1}^{p+1} (-1)^k 2^{p+1-k} \binom{p+2}{k+1} \binom{k}{1} \\ &= -\sum_{k=1}^{p} (-1)^k 2^{p+1-k} \binom{p+1}{k+1} \binom{k}{1} - \sum_{k=1}^{p+1} (-1)^k 2^{p+1-k} \binom{p+1}{k} \binom{k}{1} \\ &= -2 \sum_{k=1}^{p} (-1)^k 2^{p-k} \binom{p+1}{k+1} \binom{k}{1} - \sum_{k=1}^{p+1} (-1)^k 2^{p+1-k} (p+1) \binom{p}{k-1} \\ &= -2 \sum_{k=1}^{p} (-1)^k 2^{p-k} \binom{p+1}{k+1} \binom{k}{1} + (p+1) \sum_{k=0}^{p} (-1)^k 2^{p-k} \binom{p}{k} \\ &= 2a_{p1} + (p+1)(2-1)^p = 2a_{p1} + (p+1) \cdot 1 \\ &= 2 \left\langle {p+1 \atop 1} \right\rangle + (p+1) \left\langle {p+1 \atop 0} \right\rangle = \left\langle {p+2 \atop 1} \right\rangle. \end{aligned}$$

Thus, the result is true for $p+1$. By induction, the result is true for all positive integers p.

5. Proof of Lemma 2

We will prove this result in 3 parts. Let

$$c_{pj} = \sum_{0 \le i \le p-j} \binom{p+1}{i}.$$

First we will show that for any positive integer p,

$$a_{pp} = c_{pp}.$$

This follows since

$$\begin{aligned} a_{pp} &= (-1)^p \sum_{k=p}^{p} (-1)^k 2^{p-k} \binom{p+1}{k+1} \binom{k}{p} \\ &= (-1)^p (-1)^p 2^{p-p} \binom{p+1}{p+1} \binom{p}{p} = 1 \end{aligned}$$

and

$$c_{pp} = \sum_{0 \le i \le p-p} \binom{p+1}{i} = \binom{p+1}{0} = 1.$$

Second we will show that for any positive integer p,

$$a_{p1} = c_{p1}.$$

By Lemma 1

$$a_{p1} = \left\langle {p+1 \atop 1} \right\rangle.$$

By a property of Eulerian numbers

$$c_{p1} = \sum_{0 \le i \le p-1} \binom{p+1}{i} = 2^{p+1} - p - 2 = \left\langle {p+1 \atop 1} \right\rangle.$$

Third we will show that for $p \geq 2$ and $2 \leq j \leq p$,

$$a_{p+1,j} = a_{pj} + a_{p,j-1}$$

and

$$c_{p+1,j} = c_{pj} + c_{p,j-1}.$$

We see that

$$\begin{aligned} c_{p+1,j} &= \sum_{0 \leq i \leq p+1-j} \binom{p+2}{i} = \sum_{0 \leq i \leq p+1-j} \binom{p+1}{i} + \sum_{1 \leq i \leq p+1-j} \binom{p+1}{i-1} \\ &= \sum_{0 \leq i \leq p-(j-1)} \binom{p+1}{i} + \sum_{0 \leq i \leq p-j} \binom{p+1}{i} = c_{p,j-1} + c_{pj}. \end{aligned}$$

We also see (using several binomial coefficient identities and rearranging terms in the sums) that

$$\begin{aligned} a_{p+1,j} &= (-1)^j \sum_{k=j}^{p+1} (-1)^k 2^{p+1-k} \binom{p+2}{k+1} \binom{k}{j} \\ &= 2^{p+1-j} \binom{p+2}{j+1} \binom{j}{j} + (-1)^j \sum_{k=j+1}^{p} (-1)^k 2^{p+1-k} \binom{p+2}{k+1} \binom{k}{j} \\ &\quad + (-1)^j (-1)^{p+1} \binom{p+2}{p+2} \binom{p+1}{j} \\ &= 2^{p+1-j} \binom{p+1}{j} \binom{j-1}{j-1} + 2^{p+1-j} \binom{p+1}{j+1} \binom{j}{j} \\ &\quad + (-1)^j \sum_{k=j+1}^{p} (-1)^k 2^{p+1-k} \left[\binom{p+1}{k} \binom{k-1}{j} + \binom{p+1}{k} \binom{k-1}{j-1} + \binom{p+1}{k+1} \binom{k}{j} \right] \\ &\quad + (-1)^j (-1)^{p+1} \binom{p+1}{p+1} \binom{p}{j-1} + (-1)^j (-1)^{p+1} \binom{p+1}{p+1} \binom{p}{j} \\ &= 2^{p+1-j} \binom{p+1}{j} \binom{j-1}{j-1} + (-1)^j \sum_{k=j+1}^{p} (-1)^k 2^{p+1-k} \binom{p+1}{k} \binom{k-1}{j-1} \\ &\quad + (-1)^j (-1)^{p+1} \binom{p+1}{p+1} \binom{p}{j-1} + 2^{p+1-j} \binom{p+1}{j+1} \binom{j}{j} \\ &\quad + (-1)^j \sum_{k=j+1}^{p} (-1)^k 2^{p+1-k} \left[\binom{p+1}{k} \binom{k-1}{j} + \binom{p+1}{k+1} \binom{k}{j} \right] \\ &\quad + (-1)^j (-1)^{p+1} \binom{p+1}{p+1} \binom{p}{j} \end{aligned}$$

$$
\begin{aligned}
&= 2^{p+1-j}\binom{p+1}{j}\binom{j-1}{j-1} + (-1)^{j-1}\sum_{k=j}^{p-1}(-1)^k 2^{p-k}\binom{p+1}{k+1}\binom{k}{j-1} \\
&\quad + (-1)^j(-1)^{p+1}\binom{p+1}{p+1}\binom{p}{j-1} + 2^{p+1-j}\binom{p+1}{j+1}\binom{j}{j} \\
&\quad + (-1)^j\sum_{k=j+1}^{p}(-1)^k 2^{p+1-k}\left[\binom{p+1}{k}\binom{k-1}{j} + \binom{p+1}{k+1}\binom{k}{j}\right] \\
&\quad + (-1)^j(-1)^{p+1}\binom{p+1}{p+1}\binom{p}{j} \\
&= (-1)^{j-1}\sum_{k=j-1}^{p}(-1)^k 2^{p-k}\binom{p+1}{k+1}\binom{k}{j-1} + 2^{p+1-j}\binom{p+1}{j+1}\binom{j}{j} - 2^{p-j}\binom{p+1}{j+1}\binom{j}{j} \\
&\quad + (-1)^j\sum_{k=j+2}^{p}(-1)^k 2^{p+1-k}\binom{p+1}{k}\binom{k-1}{j} + (-1)^j\sum_{k=j+1}^{p-1}(-1)^k 2^{p+1-k}\binom{p+1}{k+1}\binom{k}{j} \\
&\quad + (-1)^j(-1)^p 2\binom{p+1}{p+1}\binom{p}{j} + (-1)^j(-1)^{p+1}\binom{p+1}{p+1}\binom{p}{j} \\
&= (-1)^{j-1}\sum_{k=j-1}^{p}(-1)^k 2^{p-k}\binom{p+1}{k+1}\binom{k}{j-1} + 2^{p-j}\binom{p+1}{j+1}\binom{j}{j} \\
&\quad + (-1)^j\sum_{k=j+1}^{p-1}(-1)^{k+1} 2^{p-k}\binom{p+1}{k+1}\binom{k}{j} \\
&\quad + (-1)^j\sum_{k=j+1}^{p-1}(-1)^k 2^{p+1-k}\binom{p+1}{k+1}\binom{k}{j} + (-1)^j(-1)^p\binom{p+1}{p+1}\binom{p}{j} \\
&= (-1)^{j-1}\sum_{k=j-1}^{p}(-1)^k 2^{p-k}\binom{p+1}{k+1}\binom{k}{j-1} + 2^{p-j}\binom{p+1}{j+1}\binom{j}{j} \\
&\quad + (-1)^j\sum_{k=j+1}^{p-1}(-1)^k 2^{p-k}\binom{p+1}{k+1}\binom{k}{j} + (-1)^j(-1)^p\binom{p+1}{p+1}\binom{p}{j} \\
&= (-1)^{j-1}\sum_{k=j-1}^{p}(-1)^k 2^{p-k}\binom{p+1}{k+1}\binom{k}{j-1} + (-1)^j\sum_{k=j}^{p}(-1)^k 2^{p-k}\binom{p+1}{k+1}\binom{k}{j} \\
&= a_{p,j-1} + a_{pj}.
\end{aligned}
$$

Thus, by the 3 parts, the two arrays are identical. Therefore, the proof of Lemma 2 is complete.

6. Proof of Lemma 3

Let

$$f(i) = (2n - 2i + 1)^{2k+1}$$

and let $\triangle$ denote the forward-difference operator. Then

$$\begin{aligned}\triangle^{2n+1} f(0) &= \sum_{i=0}^{2n+1} \binom{2n+1}{i} (-1)^i (2n - 2i + 1)^{2k+1} \\ &= 2\sum_{i=0}^{n} \binom{2n+1}{i} (-1)^i (2n - 2i + 1)^{2k+1}.\end{aligned}$$

But since f is a polynomial in i of degree $2k + 1$ and $n > k$,

$$\triangle^{2n+1} f(0) = 0.$$

Therefore,

$$\sum_{i=0}^{n} \binom{2n+1}{i} (-1)^i (2n - 2i + 1)^{2k+1} = 0.$$

7. Proof of Lemma 4

Let p and j be positive integers and $1 \le j \le p + 1$. By Lemma 2

$$a_{pj} = \sum_{0 \le i \le p-j} \binom{p+1}{i}.$$

Also, assume $a_{p,p+1} = 0$. Thus,

$$\begin{aligned}a_{p+1,j} - \binom{p+1}{j} &= \sum_{0 \le i \le p+1-j} \binom{p+2}{i} - \binom{p+1}{j} \\ &= \binom{p+2}{0} + \sum_{1 \le i \le p+1-j} \binom{p+2}{i} - \binom{p+1}{j} \\ &= \binom{p+1}{0} + \sum_{1 \le i \le p+1-j} \left(\binom{p+1}{i} + \binom{p+1}{i-1} \right) - \binom{p+1}{j} \\ &= \binom{p+1}{0} + \sum_{1 \le i \le p+1-j} \binom{p+1}{i} - \binom{p+1}{p+1-j} + \sum_{1 \le i \le p+1-j} \binom{p+1}{i-1} \\ &= \binom{p+1}{0} + \sum_{1 \le i \le p-j} \binom{p+1}{i} + \sum_{0 \le i \le p-j} \binom{p+1}{i} \\ &= 2 \sum_{0 \le i \le p-j} \binom{p+1}{i} = 2a_{pj}.\end{aligned}$$

8. Proof of Lemma 5

The proof is by induction on p.

Base Step.

We will show that Lemma 5 is true for $p = 2k$. We will do this by solving a sequence of recurrence relations by the perturbation method. Let m be a nonnegative integer. Consider the recurrence relation

$$x_{-1} = 0, \quad \text{and} \quad x_n = n^m - x_{n-1} \quad \text{for} \quad n \geq 0.$$

Let $P_m(n)$ be the solution of this recurrence relation. To describe the solutions to these recurrences we need the following notation. Let $C(n)$ denote a statement which is either true or false, depending on n. Then using APL notation [2] we define

$$[C(n)] = \begin{cases} 1, & \text{if } C(n) \text{ is true} \\ 0, & \text{if } C(n) \text{ is false.} \end{cases}$$

The first 3 recurrence relations and their solutions can be found in Problem 21 of Chapter 2 of [2]. The solutions for $m = 0, 1$ and 2 are

$$\begin{aligned} P_0(n) &= 1 - [n \text{ is odd}] \\ P_1(n) &= \frac{1}{2}n + \frac{1}{2}[n \text{ is odd}] \\ \text{and} \quad P_2(n) &= \frac{1}{2}n^2 + \frac{1}{2}n. \end{aligned} \tag{4}$$

In using the perturbation method to find the solutions for $m \geq 3$, we obtain the relation

$$P_m(n) = \frac{1}{2}\left((n+1)^m - \sum_{i=1}^{m} \binom{m}{i} P_{m-i}(n)\right). \tag{5}$$

Using this relation, we can compute $P_m(n)$ for $m = 3, 4, \ldots, 12$.

$$\begin{aligned}
P_3(n) &= \frac{1}{2}n^3 + \frac{3}{4}n^2 - \frac{1}{4}[n \text{ is odd}] \\
P_4(n) &= \frac{1}{2}n^4 + n^3 - \frac{1}{2}n \\
P_5(n) &= \frac{1}{2}n^5 + \frac{5}{4}n^4 - \frac{5}{4}n^2 + \frac{1}{2}[n \text{ is odd}] \\
P_6(n) &= \frac{1}{2}n^6 + \frac{3}{2}n^5 - \frac{5}{2}n^3 + \frac{3}{2}n \\
P_7(n) &= \frac{1}{2}n^7 + \frac{7}{4}n^6 - \frac{35}{8}n^4 + \frac{21}{4}n^2 - \frac{17}{8}[n \text{ is odd}] \\
P_8(n) &= \frac{1}{2}n^8 + 2n^7 - 7n^5 + 14n^3 - \frac{17}{2}n \\
P_9(n) &= \frac{1}{2}n^9 + \frac{9}{4}n^8 - \frac{21}{2}n^6 + \frac{63}{2}n^4 - \frac{153}{4}n^2 + \frac{31}{2}[n \text{ is odd}] \\
P_{10}(n) &= \frac{1}{2}n^{10} + \frac{5}{2}n^9 - 15n^7 + 63n^5 - \frac{255}{2}n^3 + \frac{155}{2}n \\
P_{11}(n) &= \frac{1}{2}n^{11} + \frac{11}{4}n^{10} - \frac{165}{8}n^8 + \frac{231}{2}n^6 - \frac{2805}{8}n^4 + \frac{1705}{4}n^2 - \frac{691}{4}[n \text{ is odd}] \\
P_{12}(n) &= \frac{1}{2}n^{12} + 3n^{11} - \frac{55}{2}n^9 + 198n^7 - \frac{1683}{2}n^5 + 1705n^3 - \frac{2073}{2}n.
\end{aligned}$$

Each $P_m(n)$ is a polynomial of degree m plus possibly a term involving $[n \text{ is odd}]$. If we let b_m denote the coefficient in front of the term $[n \text{ is odd}]$ in $P_m(n)$, then we have the table of elements

m	0	1	2	3	4	5	6	7	8	9	10	11	12	$\cdots$
b_m	-1	$1/2$	0	$-1/4$	0	$1/2$	0	$-17/8$	0	$31/2$	0	$-691/4$	0	$\cdots$

By (4) and (5), the values of the b_ms satisfy the conditions $b_0 = -1$ and for $m \geq 1$,

$$b_m = -\frac{1}{2}\sum_{i=0}^{m-1}\binom{m}{i}b_i.$$

Using generating functions, it can be shown that

$$\sum_{k=0}^{\infty} b_k \frac{x^k}{k!} = \frac{-2}{e^x + 1}.$$

Since

$$\frac{-2}{e^x + 1} + 1 = \frac{e^x - 1}{e^x + 1}$$

is an odd function it follows that the even subscripted bs are 0, i.e. $b_{2k} = 0$ for $k \geq 1$. Therefore, $P_{2k}(n)$ for $k \geq 1$ is a polynomial of degree $2k$, i.e. it contains no term [n is odd].

It should be noted that the Genocchi numbers [1] are defined by

$$\frac{2x}{e^x + 1} = \sum_{k=0}^{\infty} G_k \frac{x^k}{k!}.$$

Therefore, for $n \geq 0$

$$b_n = -\frac{1}{n+1} G_{n+1}.$$

Now, using Lemma 2 on the first equality we have

$$\begin{aligned}
\sum_{j=1}^{2k} (-1)^j a_{2k,j} j^{2k} &= \sum_{j=1}^{2k} (-1)^j \sum_{i=0}^{2k-j} \binom{2k+1}{i} j^{2k} \\
&= \sum_{i=0}^{2k-1} \binom{2k+1}{i} \sum_{j=1}^{2k-i} (-1)^j j^{2k} \\
&= \sum_{i=0}^{2k-1} \binom{2k+1}{i} \sum_{j=0}^{2k-i} (-1)^j j^{2k} \\
&= \sum_{i=0}^{2k+1} \binom{2k+1}{i} \sum_{j=0}^{2k-i} (-1)^j j^{2k} \\
&= \sum_{i=0}^{2k+1} \binom{2k+1}{2k+1-i} \sum_{j=0}^{2k-(2k+1-i)} (-1)^j j^{2k} \\
&= \sum_{i=0}^{2k+1} \binom{2k+1}{i} (-1)^{i+1} \left(\sum_{j=0}^{i-1} (-1)^j j^{2k} (-1)^{i+1} \right) \\
&= \sum_{i=0}^{2k+1} \binom{2k+1}{i} (-1)^{i+1} P_{2k}(-1+i).
\end{aligned}$$

But since the last sum is $-\triangle^{2k+1} P_{2k}(-1)$ and P_{2k} is a polynomial of degree $2k$, it follows that the above sum is 0. This completes the proof of the base step.

Induction Step. Next, we will show that if the formula is true for some $p \geq 2k$, then it is true for $p+1$. Suppose that the formula is true for some $p \geq 2k$. We will use the fact that

$$\sum_{j=0}^{p+1}(-1)^{j+1}\binom{p+1}{j}j^{2k} = 0.$$

This can be seen by noting that if $Q(j) = j^{2k}$, then

$$\sum_{j=0}^{p+1}(-1)^{j+1}\binom{p+1}{j}j^{2k} = -\triangle^{p+1}Q(0) = 0$$

since Q is a polynomial in j of degree $2k$ and $p+1 > 2k$. Hence,

$$\begin{aligned}
&\sum_{j=1}^{p+1}(-1)^j a_{p+1,j}j^{2k} \\
&= \sum_{j=1}^{p+1}(-1)^j a_{p+1,j}j^{2k} + \sum_{j=0}^{p+1}(-1)^{j+1}\binom{p+1}{j}j^{2k} \\
&= \sum_{j=1}^{p+1}(-1)^j a_{p+1,j}j^{2k} + \sum_{j=1}^{p+1}(-1)^{j+1}\binom{p+1}{j}j^{2k} \\
&= \sum_{j=1}^{p+1}(-1)^j\left(a_{p+1,j} - \binom{p+1}{j}\right)j^{2k} \\
&= \sum_{j=1}^{p}(-1)^j 2a_{pj}j^{2k} = 2\left(\sum_{j=1}^{p}(-1)^j a_{pj}j^{2k}\right).
\end{aligned}$$

The next to last equality follows from Lemma 4. But the last expression is 0 by our induction hypothesis. Therefore, the result is true for $p+1$. This completes the proof of the induction step.

Thus, by induction, Lemma 5 is proved.

9. Proof of the Theorem

We begin the proof of (2) by noting that if

$$(x-1)^{2n+1}\Big|(x+1)(x^3+1)\cdots(x^{2n+1}+1)\sum_{i=0}^{n}\binom{2n+1}{n-i}(-1)^{n-i}\frac{x^{2i+1}-1}{x^{2i+1}+1} \qquad (6)$$

is true, then (2) is true. Suppose (6) is true and substitute α/β for x in (6), where

$$\alpha = \frac{1+\sqrt{5}}{2} \quad \text{and} \quad \beta = \frac{1-\sqrt{5}}{2}.$$

Using the fact that $\alpha - \beta = \sqrt{5}$ and multiplying (6) by β^{n^2}, (6) becomes

$$5^n \Big| (\alpha+\beta)(\alpha^3+\beta^3)\cdots(\alpha^{2n+1}+\beta^{2n+1}) \sum_{i=0}^{n} \binom{2n+1}{n-i} (-1)^{n-i} \frac{\alpha^{2i+1}-\beta^{2i+1}}{\sqrt{5}(\alpha^{2i+1}+\beta^{2i+1})}.$$

But this last result, by the use of Binet's formula [3], i.e.

$$F_n = \frac{\alpha^n - \beta^n}{\sqrt{5}} \quad \text{and} \quad L_n = \alpha^n + \beta^n,$$

is (2) .

Let

$$f(x) = (x+1)(x^3+1)\cdots(x^{2n+1}+1)$$

and

$$g(x) = \sum_{i=0}^{n} \binom{2n+1}{n-i} (-1)^{n-i} \frac{x^{2i+1}-1}{x^{2i+1}+1}.$$

Now, if D denotes the derivative operator, then by applying the product rule j times we obtain the formula

$$D^j f(x)g(x) = \sum_{i=0}^{j} \binom{j}{i} D^i f(x) D^{j-i} g(x). \tag{7}$$

Proving (6) would be equivalent to showing that

$$D^j f(1)g(1) = 0 \text{ for } j = 0, 1, \ldots, 2n. \tag{8}$$

But by (7) we can prove (8) if we can show that

$$g(1) = Dg(1) = D^2 g(1) = \cdots = D^{2n} g(1) = 0. \tag{9}$$

Simplifying $g(x)$ we have

$$\begin{aligned} g(x) &= \sum_{i=0}^{n} \binom{2n+1}{n-i} (-1)^{n-i} \frac{x^{2i+1}-1}{x^{2i+1}+1} \\ &= \sum_{i=0}^{n} \binom{2n+1}{n-i} (-1)^{n-i} \left(1 - \frac{2}{x^{2i+1}+1}\right). \end{aligned} \tag{10}$$

First of all, it is clear that $g(1) = 0$. To compute the pth derivative of $g(x)$ where $1 \leq p \leq 2n$, we need to find the pth derivative of

$$\frac{1}{x^{2i+1}+1}.$$

Using a result in [7],

$$D^p \left[\frac{1}{x^{2i+1}+1}\right] = \sum_{k=1}^{p} (-1)^k \binom{p+1}{k+1} \frac{1}{(x^{2i+1}+1)^{k+1}} D^p \left[(x^{2i+1}+1)^k\right].$$

We now need the notation for falling factorials [2], i.e.

$$x^{\underline{p}} = x(x-1)\cdots(x-p+1)$$

and the binomial theorem

$$(x^{2i+1}+1)^k = \sum_{j=0}^{k} \binom{k}{j} x^{(2i+1)j}.$$

Thus,

$$\begin{aligned} D^p \left[\sum_{j=0}^{k} \binom{k}{j} x^{(2i+1)j}\right] &= \sum_{j=0}^{k} \binom{k}{j} D^p x^{(2i+1)j} \\ &= \sum_{j=0}^{k} \binom{k}{j} [(2i+1)j][(2i+1)j-1]\cdots[(2i+1)j-p+1] x^{(2i+1)j-p} \\ &= \sum_{j=0}^{k} \binom{k}{j} [(2i+1)j]^{\underline{p}} x^{(2i+1)j-p}. \end{aligned}$$

It follows that

$$D^p\left[\frac{1}{x^{2i+1}+1}\right]\Bigg|_{x=1} = \sum_{k=1}^{p}(-1)^k\binom{p+1}{k+1}2^{-k-1}\sum_{j=0}^{k}\binom{k}{j}[(2i+1)j]^{\underline{p}}. \tag{11}$$

Next, we will study (11) with $2i+1$ replaced by m, i.e.

$$\sum_{k=1}^{p}(-1)^k\binom{p+1}{k+1}2^{-k-1}\sum_{j=0}^{k}\binom{k}{j}(jm)^{\underline{p}}.$$

Using the fact that $p \geq 1$, so we have no term when $j = 0$, we wish to investigate the sum

$$\sum_{k=1}^{p}(-1)^k\binom{p+1}{k+1}2^{-k-1}\sum_{j=1}^{k}\binom{k}{j}(jm)^{\underline{p}}. \tag{12}$$

By changing the order of summation, it follows that (12) becomes

$$\begin{aligned}&\sum_{j=1}^{p}(jm)^{\underline{p}}\sum_{k=j}^{p}(-1)^k\binom{p+1}{k+1}\binom{k}{j}2^{-k-1}\\ &= \frac{1}{2^{p+1}}\sum_{j=1}^{p}(jm)^{\underline{p}}\sum_{k=j}^{p}(-1)^k2^{p-k}\binom{p+1}{k+1}\binom{k}{j}.\end{aligned}$$

We want to show that the above polynomial in m only contains odd terms, i.e. there are only terms of odd degree in the polynomial. The first few such polynomials are

$$\begin{aligned}&\frac{1}{4}(-m),\\ &\frac{1}{8}(2m),\\ &\frac{1}{16}(2m^3-8m),\\ &\frac{1}{32}(-24m^3+48m),\\ &\frac{1}{64}(-16m^5+280m^3-384m),\\ \text{and}\quad &\frac{1}{128}(480m^5-3600m^3+3840m),\end{aligned}$$

for $p = 1, 2, 3, 4, 5$, and 6, respectively. Now, by (3) we have that the polynomial is

$$D^p\left[\frac{1}{x^m+1}\right]\Bigg|_{x=1} = \frac{1}{2^{p+1}}\sum_{j=1}^{p}(-1)^j a_{pj}(jm)^{\underline{p}}.$$

Next, we recall the Stirling numbers of the first kind. They are denoted by

$$s(n,k)$$

and count the number of ways to arrange n objects into k cycles [1,2]. A property of Stirling numbers of the first kind is

$$s(n,n-k) = \sum_{0\le i_1<\cdots<i_k\le n-1} i_1\cdots i_k.$$

Thus, we have that

$$x^{\underline{p}} = x(x-1)\cdots(x-p+1) = \sum_{j=0}^{p}(-1)^j s(p,p-j)x^{p-j}.$$

It follows that

$$(jm)^{\underline{p}} = \sum_{k=0}^{p}(-1)^k = s(p,p-k)(jm)^{p-k} = \sum_{k=0}^{p}(-1)^k s(p,p-k)j^{p-k}m^{p-k}. \quad (13)$$

Hence, by using (13) and changing the order of summation, the polynomial in m is

$$\begin{aligned}
&D^p\left[\frac{1}{x^m+1}\right]\Bigg|_{x=1}\\
&= \frac{1}{2^{p+1}}\sum_{j=1}^{p}(-1)^j a_{pj}(jm)^{\underline{p}}\\
&= \frac{1}{2^{p+1}}\sum_{j=1}^{p}(-1)^j a_{pj}\sum_{k=0}^{p}(-1)^k s(p,p-k)j^{p-k}m^{p-k}\\
&= \frac{1}{2^{p+1}}\sum_{k=0}^{p}(-1)^k s(p,p-k)m^{p-k}\sum_{j=1}^{p}(-1)^j a_{pj}j^{p-k}\\
&= \frac{1}{2^{p+1}}\sum_{k=0}^{p}(-1)^{p-k} s(p,k)m^{k}\sum_{j=1}^{p}(-1)^j a_{pj}j^{k}.
\end{aligned}$$

Therefore, for $p \geq 1$ we have by (7) that

$$\begin{aligned}
D^p g(1) &= \sum_{i=0}^{n} \binom{2n+1}{n-i} (-1)^{n-i} D^p \left(1 - \frac{2}{g_{2i+1}(x)}\right)\Bigg|_{x=1} \\
&= \sum_{i=0}^{n} \binom{2n+1}{i} (-1)^{i} D^p \left(1 - \frac{2}{g_{2n-2i+1}(x)}\right)\Bigg|_{x=1} \\
&= -2\sum_{i=0}^{n} \binom{2n+1}{i} (-1)^{i} D^p \left(\frac{1}{g_{2n-2i+1}(x)}\right)\Bigg|_{x=1} \\
&= -2\sum_{i=0}^{n} \binom{2n+1}{i} (-1)^{i} \frac{1}{2^{p+1}} \sum_{k=0}^{p} (-1)^{p-k} s(p,k) (2n-2i+1)^k \sum_{j=1}^{p} (-1)^j a_{pj} j^k \\
&= \frac{-2}{2^{p+1}} \sum_{k=0}^{p} (-1)^{p-k} s(p,k) \sum_{j=1}^{p} (-1)^j a_{pj} j^k \sum_{i=0}^{n} \binom{2n+1}{i} (-1)^i (2n-2i+1)^k.
\end{aligned}$$

To finish the proof of the Theorem we will prove that the last expression is 0. To do this we will isolate the term when $k = 0$ and the two sums when $0 < 2k+1 \leq p$ and $0 < 2k \leq p$. The term and the two sums are listed below.

$$\begin{aligned}
&\frac{-2}{2^{p+1}} (-1)^p s(p,0) \sum_{j=1}^{p} (-1)^j a_{pj} \sum_{i=0}^{n} \binom{2n+1}{i} (-1)^i \\
&+ \frac{-2}{2^{p+1}} \sum_{0<2k+1\leq p} (-1)^{p-2k-1} s(p,2k+1) \sum_{j=1}^{p} (-1)^j a_{pj} j^{2k+1} \\
&\quad \left(\sum_{i=0}^{n} \binom{2n+1}{i} (-1)^i (2n-2i+1)^{2k+1}\right) \\
&+ \frac{-2}{2^{p+1}} \sum_{0<2k\leq p} (-1)^{p-2k} s(p,2k) \left(\sum_{j=1}^{p} (-1)^j a_{pj} j^{2k}\right) \\
&\quad \sum_{i=0}^{n} \binom{2n+1}{i} (-1)^i (2n-2i+1)^{2k}.
\end{aligned}$$

The term when $k = 0$ is 0 since $s(p,0) = 0$ for $p \geq 1$. Since $1 \leq p \leq 2n$ and $2k+1 \leq p$, it follows that $k < n$. Thus by Lemma 3 the first sum is 0. Lemma 5 proves that the second sum is 0.

Summarizing, we have just shown that the term and the two sums are 0. Thus, for $1 \leq p \leq 2n$ we have $D^p g(1) = 0$. Since $g(1) = 0$ we have proved that (6) is true. Therefore, the Theorem is proved.

10. Further Questions

First of all, we could study the polynomial P_m in Lemma 5. Is there an explicit formula for P_m? Second, in studying (2) we came across the conjecture that

$$(x+1)^n \left| (x+1)(x^3+1)\cdots(x^{2n+1}+1) \sum_{i=0}^{n} \binom{2n+1}{n-i} (-1)^{n-i} \frac{x^{2i+1}-1}{x^{2i+1}+1}. \right.$$

Finally, we could again study Melham's sum

$$L_1 L_3 \cdots L_{2m+1} \sum_{k=1}^{n} F_{2k}^{2m+1},$$

where m is a nonnegative integer and n is a positive integer.

References

1. L. Comtet. *Advanced Combinatorics.* Dordrecht: D. Reidel, 1974.
2. R. L. Graham, D. E. Knuth, & O. Patashnik. *Concrete Mathematics.* Reading, Mass.: Addison-Wesley, 1994.
3. V. E. Hoggatt, Jr. *Fibonacci and Lucas Numbers.* Boston: Houghton Mifflin, 1969.
4. V. E. Hoggatt, Jr. & G. E. Bergum. "Divisibility and Congruence Relations." *The Fibonacci Quarterly* **12.2** (1974): 189–195.
5. V. E. Hoggatt, Jr. & Marjorie Bicknell-Johnson. "Divisibility by Fibonacci and Lucas Squares." *The Fibonacci Quarterly* **15.1** (1977): 3–8.
6. D. Jennings. "Some Polynomial Identities for the Fibonacci and Lucas Numbers." *The Fibonacci Quarterly* **31.2** (1993): 134–137.
7. R. A. Leslie. "How Not to Repeatedly Differentiate a Reciprocal." *American Mathematical Monthly* **98** (1991): 732–735.
8. Y. V. Matijasevič. "Enumerable Sets are Diophantine." *Proc. of the Academy of Sciences of the USSR* **11** (1970): No. 2.

9. R. S. Melham. private communication.

10. N. J. A. Sloane. *On-Line Encyclopedia of Integer Sequences.* Published electronically at http://www.research.att.com/∼njas/sequences.

AMS Classification Numbers: 11B39.

COUNTING OCCURRENCES OF A PATTERN OF TYPE $(1,2)$ OR $(2,1)$ IN PERMUTATIONS

ANDERS CLAESSON AND TOUFIK MANSOUR

ABSTRACT. Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. Claesson presented a complete solution for the number of permutations avoiding any single pattern of type $(1,2)$ or $(2,1)$. For eight of these twelve patterns the answer is given by the Bell numbers. For the remaining four the answer is given by the Catalan numbers.

With respect to being equidistributed there are three different classes of patterns of type $(1,2)$ or $(2,1)$. We present a recursion for the number of permutations containing exactly one occurrence of a pattern of the first or the second of the aforementioned classes, and we also find an ordinary generating function for these numbers. We prove these results both combinatorially and analytically. Finally, we give the distribution of any pattern of the third class in the form of a continued fraction, and we also give explicit formulas for the number of permutations containing exactly r occurrences of a pattern of the third class when $r \in \{1,2,3\}$.

1. INTRODUCTION AND PRELIMINARIES

Let $[n] = \{1, 2, \ldots, n\}$ and denote by $\mathcal{S}_n$ the set of permutations of $[n]$. We shall view permutations in $\mathcal{S}_n$ as words with n distinct letters in $[n]$.

Classically, a pattern is a permutation $\sigma \in \mathcal{S}_k$, and an occurrence of σ in a permutation $\pi = a_1 a_2 \cdots a_n \in \mathcal{S}_n$ is a subword of π that is order equivalent to σ. For example, an occurrence of 132 is a subword $a_i a_j a_k$ $(1 \le i < j < k \le n)$ of π such that $a_i < a_k < a_j$. We denote by $s^r_\sigma(n)$ the number of permutations in $\mathcal{S}_n$ that contain exactly r occurrences of the pattern σ.

In the last decade much attention has been paid to the problem of finding the numbers $s^r_\sigma(n)$ for a fixed $r \ge 0$ and a given pattern σ (see [1, 2, 4, 6, 7, 8, 11, 13, 14, 16, 17, 18, 19, 20, 21]). Most of the authors consider only the case $r = 0$, thus studying permutations *avoiding* a given pattern. Only a few papers consider the case $r > 0$, usually restricting themselves to patterns of length 3. Using two simple involutions (*reverse* and *complement*) on $\mathcal{S}_n$ it is immediate that with respect to being equidistributed, the six patterns of length three fall into the two classes $\{123, 321\}$ and $\{132, 213, 231, 312\}$. Noonan [15] proved that $s^1_{123}(n) = \frac{3}{n}\binom{2n}{n-3}$. A general approach to the problem was suggested by Noonan and Zeilberger [16]; they gave another proof of Noonan's result, and conjectured that

$$s^2_{123}(n) = \frac{59n^2 + 117n + 100}{2n(2n-1)(n+5)} \binom{2n}{n-4}$$

and $s^1_{132}(n) = \binom{2n-3}{n-3}$. The latter conjecture was proved by Bóna in [7]. A conjecture of Noonan and Zeilberger states that $s^r_\sigma(n)$ is P-recursive in n for any r and σ. It was proved by Bóna [5] for $\sigma = 132$.

Mansour and Vainshtein [14] suggested a new approach to this problem in the case $\sigma = 132$, which allows one to get an explicit expression for $s^r_{132}(n)$ for any given r.

Date: October 8, 2002.

More precisely, they presented an algorithm that computes the generating function $\sum_{n\geq 0} s^r_{132}(n)x^n$ for any $r \geq 0$. To get the result for a given r, the algorithm performs certain routine checks for each element of the symmetric group S_{2r}. The algorithm has been implemented in C, and yields explicit results for $1 \leq r \leq 6$.

In [3] Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. The motivation for Babson and Steingrímsson in introducing these patterns was the study of Mahonian permutation statistics. Two examples of (generalized) patterns are 1-32 and 13-2. An occurrence of 1-32 in a permutation $\pi = a_1 a_2 \cdots a_n$ is a subword $a_i a_j a_{j+1}$ of π such that $a_i < a_{j+1} < a_j$. Similarly, an occurrence of 13-2 is a subword $a_i a_{i+1} a_j$ of π such that $a_i < a_j < a_{i+1}$. More generally, if $xyz \in \mathcal{S}_3$ and $\pi = a_1 a_2 \cdots a_n \in \mathcal{S}_n$, then we define

$$(x\text{-}yz)\,\pi \;=\; |\{a_i a_j a_{j+1} : \mathrm{proj}(a_i a_j a_{j+1}) = xyz, 1 \leq i < j < n\}|,$$

where $\mathrm{proj}(x_1x_2x_3)(i) = |\{j \in \{1,2,3\} : x_j \leq x_i\}|$ for $i \in \{1,2,3\}$ and $x_1, x_2, x_3 \in [n]$. For instance, $\mathrm{proj}(127) = \mathrm{proj}(138) = \mathrm{proj}(238) = 123$, and

$$(1\text{-}23)\,491273865 = |\{127, 138, 238\}| = 3.$$

Similarly, we also define $(xy\text{-}z)\,\pi = (z\text{-}yx)\,\pi^r$, where π^r denotes the reverse of π, that is, π read backwards.

For any word (finite sequence of letters), w, we denote by $|w|$ the length of w, that is, the number of letters in w. A pattern $\sigma = \sigma_1\text{-}\sigma_2\text{-}\cdots\text{-}\sigma_k$ containing exactly $k-1$ dashes is said to be of type $(|\sigma_1|, |\sigma_2|, \ldots, |\sigma_k|)$. For example, the pattern 142-5-367 is of type $(3,1,3)$, and any classical pattern of length k is of type $(\underbrace{1,1,\ldots,1}_{k})$.

In [11] Elizalde and Noy presented the following theorem regarding the distribution of the number of occurrences of any pattern of type (3).

Theorem 1 (Elizalde and Noy [11]). *Let $h(x) = \sqrt{(x-1)(x+3)}$. Then*

$$\sum_{\pi\in\mathcal{S}} x^{(123)\pi} \frac{t^{|\pi|}}{|\pi|!} = \frac{2h(x)e^{\frac{1}{2}(h(x)-x+1)t}}{h(x)+x+1+(h(x)-x-1)e^{h(x)t}},$$

$$\sum_{\pi\in\mathcal{S}} x^{(213)\pi} \frac{t^{|\pi|}}{|\pi|!} = \frac{1}{1-\int_0^t e^{(x-1)z^2/2}dz}.$$

The easy proof of the following proposition can be found in [9].

Proposition 2 (Claesson [9]). *With respect to being equidistributed, the twelve patterns of type $(1,2)$ or $(2,1)$ fall into the three classes*

$$\{\,1\text{-}23,\ 3\text{-}21,\ 12\text{-}3,\ 32\text{-}1\,\},$$
$$\{\,1\text{-}32,\ 3\text{-}12,\ 21\text{-}3,\ 23\text{-}1\,\},$$
$$\{\,2\text{-}13,\ 2\text{-}31,\ 13\text{-}2,\ 31\text{-}2\,\}.$$

In the subsequent discussion we refer to the classes of the proposition above (in the order that they appear) as Class 1, 2 and 3 respectively.

Claesson [9] also gave a solution for the number of permutations avoiding any pattern of the type $(1,2)$ or $(2,1)$ as follows.

Proposition 3 (Claesson [9]). *Let $n \in \mathbb{N}$. We have*

$$|\mathcal{S}_n(\sigma)| = \begin{cases} B_n & \text{if } \sigma \in \{1\text{-}23, 3\text{-}21, 12\text{-}3, 32\text{-}1, 1\text{-}32, 3\text{-}12, 21\text{-}3, 23\text{-}1\}, \\ C_n & \text{if } \sigma \in \{2\text{-}13, 2\text{-}31, 13\text{-}2, 31\text{-}2\}, \end{cases}$$

where B_n and C_n are the nth Bell and Catalan numbers, respectively.

In particular, since B_n is not P-recursive in n, this result implies that for generalized patterns the conjecture that $s_\sigma^r(n)$ is P-recursive in n is false for $r = 0$ and, for example, $\sigma = 1\text{-}23$.

This paper is organized as follows. In Section 2 we find a recursion for the number of permutations containing exactly one occurrence of a pattern of Class 1, and we also find an ordinary generating function for these numbers. We prove these results both combinatorially and analytically. Similar results are also obtained for patterns of Class 2. In Section 3 we give the distribution of any pattern of Class 3 in the form of a continued fraction, and we also give explicit formulas for the number of permutations containing exactly r occurrences of a pattern of Class 3 when $r \in \{1, 2, 3\}$.

2. Counting occurrences of a pattern of Class 1 or 2

Theorem 4. *Let $u_1(n)$ be the number of permutations of length n containing exactly one occurrence of the pattern $1\text{-}23$ and let B_n be the nth Bell number. The numbers $u_1(n)$ satisfy the recurrence*

$$u_1(n+2) = 2u_1(n+1) + \sum_{k=0}^{n-1} \binom{n}{k} \big[u_1(k+1) + B_{k+1}\big],$$

whenever $n \geq -1$, with the initial condition $u_1(0) = 0$.

Proof. Each permutation $\pi \in \mathcal{S}_{n+2}^1(1\text{-}23)$ contains a unique subword abc such that $a < b < c$ and bc is a segment of π. Let x be the last letter of π and define the sets $\mathcal{T}$, $\mathcal{T}'$, and $\mathcal{T}''$ by

$$\pi \in \begin{cases} \mathcal{T} & \text{if } x = 2, \\ \mathcal{T}' & \text{if } x \neq 2 \text{ and } a = 1, \\ \mathcal{T}'' & \text{if } x \neq 2 \text{ and } a \neq 1. \end{cases}$$

Then $\mathcal{S}_{n+2}^1(1\text{-}23)$ is the disjoint union of $\mathcal{T}$, $\mathcal{T}'$, and $\mathcal{T}''$, so

$$u_1(n+2) = |\mathcal{T}| + |\mathcal{T}'| + |\mathcal{T}''|.$$

Since removing/adding a trailing 2 from/to a permutation does not affect the number of hits of $1\text{-}23$, we immediately get

$$|\mathcal{T}| = u_1(n+1).$$

For the cardinality of $\mathcal{T}'$ we observe that if $x \neq 2$ and $a = 1$ then $b = 2$: If the letter 2 precedes the letter 1 then every hit of $1\text{-}23$ with $a = 1$ would cause an additional hit of $1\text{-}23$ with $a = 2$ contradicting the uniqueness of the hit of $1\text{-}23$; if 1 precedes 2 then $a = 1$ and $b = 2$. Thus we can factor any permutation $\pi \in \mathcal{T}'$ uniquely in the form $\pi = \sigma 2 \tau$, where σ is $(1\text{-}23)$-avoiding, the letter 1 is included in σ, and τ is nonempty and (12)-avoiding. Owing to Proposition 3 we have showed

$$|\mathcal{T}'| = \sum_{k=0}^{n-1} \binom{n}{k} B_{k+1}.$$

Suppose $\pi \in \mathcal{T}''$. Since $x \neq 2$ and $a \neq 1$ we can factor π uniquely in the form $\pi = \sigma 1 \tau$, where σ contains exactly one occurrence of $1\text{-}23$, the letter 2 is included in σ, and τ is nonempty and (12)-avoiding. Consequently,

$$|\mathcal{T}''| = \sum_{k=0}^{n} \binom{n}{k} u_1(k+1),$$

which completes the proof. □

Example 5. Let us consider all permutations of length 5 that contain exactly one occurrence of 1-23, and give a small illustration of the proof of Theorem 12. If $\mathcal{T}$, $\mathcal{T}'$ and $\mathcal{T}''$ are defined as above then

$$\begin{aligned}
\mathcal{T} &= \underline{1354}|2\ \ \underline{1435}|2\ \ \underline{1453}|2\ \ \underline{1534}|2\ \ \underline{4135}|2\ \ \underline{5134}|2\ \ \underline{3451}|2\\
\mathcal{T}' &= \begin{array}{l}\underline{1}|\underline{2543}\ \ \underline{13}|\underline{254}\ \ \underline{14}|\underline{253}\ \ \underline{143}|\underline{25}\ \ \underline{15}|\underline{243}\ \ \underline{153}|\underline{24}\\ \underline{154}|\underline{23}\ \ \underline{31}|\underline{254}\ \ \underline{314}|\underline{25}\ \ \underline{315}|\underline{24}\ \ \underline{341}|\underline{25}\ \ \underline{351}|\underline{24}\\ \underline{41}|\underline{253}\ \ \underline{413}|\underline{25}\ \ \underline{415}|\underline{23}\ \ \underline{431}|\underline{25}\ \ \underline{451}|\underline{23}\ \ \underline{51}|\underline{243}\\ \underline{513}|\underline{24}\ \ \underline{514}|\underline{23}\ \ \underline{531}|\underline{24}\ \ \underline{541}|\underline{23}\end{array}\\
\mathcal{T}'' &= \begin{array}{l}\underline{234}|15\ \ \underline{235}|14\ \ \underline{2354}|1\ \ \underline{2435}|1\ \ \underline{245}|13\\ \underline{2453}|1\ \ \underline{2534}|1\ \ \underline{3452}|1\ \ \underline{4235}|1\ \ \underline{5234}|1\end{array}
\end{aligned}$$

where the underlined subword is the unique hit of 1-23, and the bar indicates how the permutation is factored in the proof of Theorem 12.

Theorem 6. *Let $v_1(n)$ be the number of permutations of length n containing exactly one occurrence of the pattern 1-32 and let B_n be the nth Bell number. The numbers $v_1(n)$ satisfy the recurrence*

$$v_1(n+1) = v_1(n) + \sum_{k=1}^{n-1}\left[\binom{n}{k}v_1(k) + \binom{n-1}{k-1}B_k\right],$$

whenever $n \geq 0$, with the initial condition $v_1(0) = 0$.

Proof. Each permutation $\pi \in \mathcal{S}^1_{n+2}(1\text{-}32)$ contains a unique subword acb such that $a < b < c$ and cb is a segment of π. Define the sets $\mathcal{T}$ and $\mathcal{T}'$ by

$$\pi \in \begin{cases}\mathcal{T} & \text{if } a = 1,\\ \mathcal{T}' & \text{if } a \neq 1.\end{cases}$$

Then $\mathcal{S}^1_{n+2}(1\text{-}32)$ is the disjoint union of $\mathcal{T}$ and $\mathcal{T}'$, so

$$v_1(n+2) = |\mathcal{T}| + |\mathcal{T}'|.$$

For the cardinality of $\mathcal{T}$ we observe that if $a = 1$ then $b = 2$: If the letter 2 precedes the letter 1 or 12 is a segment of π then every hit of 1-23 with $a = 1$ would cause an additional hit of 1-32 with $a = 2$ contradicting the uniqueness of the hit of 1-23; if 1 precedes 2 then $a = 1$ and $b = 2$. Thus we can factor π uniquely in the form $\pi = \sigma x 2 \tau$, where σx is (1-32)-avoiding, the letter 1 is included in σ, and τ is nonempty and (12)-avoiding. Let $\mathcal{R}_n$ be the set of (1-32)-avoiding permutations of $[n]$ that do not end with the letter 1. Since the letter 1 cannot be the last letter of a hit of 1-32, we have, by Proposition 3, that $|\mathcal{S}^0_n(1\text{-}32) \setminus \mathcal{R}_n| = B_{n-1}$. Consequently, $|\mathcal{R}_n| = B_n - B_{n-1}$ and

$$\begin{aligned}
|\mathcal{T}| &= \sum_{k=1}^{n}\binom{n-1}{k-1}|\mathcal{R}_k|\\
&= \sum_{k=1}^{n}\binom{n-1}{k-1}(B_k - B_{k-1})\\
&= \sum_{k=1}^{n-1}\binom{n-1}{k-1}B_k.
\end{aligned}$$

For the last identity we have used the familiar recurrence relation $B_{n+1} = \sum_{k=0}^{n}\binom{n}{k}B_k$.

Suppose $\pi \in \mathcal{T}'$. Since $a \neq 1$ we can factor π uniquely in the form $\pi = \sigma 1 \tau$, where σ contains exactly one occurrence of 1-32, and τ is nonempty and (12)-avoiding. Accordingly,

$$|\mathcal{T}''| = \sum_{k=0}^{n} \binom{n}{k} v_1(k),$$

which completes the proof. □

Let σ be a pattern of Class 1 or 2. Using combinatorial reasoning we have found a recursion for the number of permutations containing exactly one occurrence of the pattern σ (Theorem 4 and 6). More generally, given $r \geq 0$, we would like to find a recursion for the number of permutations containing exactly r occurrence of the pattern σ. Using a more general and analytic approach we will now demonstrate how this (at least in principle) can be achieved.

Let $S_\sigma^r(x)$ be the generating function $S_\sigma^r(x) = \sum_n s_\sigma^r(n) x^n$. To find functional relations for $S_\sigma^r(x)$ the following lemma will turn out to be useful.

Lemma 7. *If $\{a_n\}$ is a sequence of numbers and $A(x) = \sum_{n\geq 0} a_n x^n$ is its ordinary generating function, then, for any $d \geq 0$,*

$$\sum_{n\geq 0} \left[\sum_{j=0}^{n} \binom{n}{j} a_{j+d} \right] x^n = \frac{(1-x)^{d-1}}{x^d} \left[A\Big(\frac{x}{1-x}\Big) - \sum_{j=0}^{d-1} a_j \Big(\frac{x}{1-x}\Big)^j \right].$$

Proof. It is plain that

$$\sum_{n\geq 0} \Big[\sum_{j=0}^{n} \binom{n}{j} a_j \Big] x^n = \frac{1}{1-x} A\Big(\frac{x}{1-x}\Big).$$

See for example [12, p 192]. On the other hand,

$$\sum_{n\geq 0} a_{n+d} x^n = \frac{1}{x^d} \Big[A(x) - \sum_{j=0}^{d-1} a_j x^j \Big].$$

Combining these two identities we get the desired result. □

Define $\mathcal{S}_n^r(\sigma)$ to be the set of permutations $\pi \in S_n$ such that $(\sigma)\pi = r$. Let $s_\sigma^r(n) = |\mathcal{S}_n^r(\sigma)|$ for $r \geq 0$ and $s_\sigma^r(n) = 0$ for $r < 0$. Given $b_1, b_2, \ldots, b_k \in \mathbb{N}$, we also define

$$s_\sigma^r(n; b_1, b_2, \ldots, b_k) = \#\{a_1 a_2 \cdots a_n \in \mathcal{S}_n^r(\sigma) \mid a_1 a_2 \cdots a_k = b_1 b_2 \cdots b_k\}.$$

As a direct consequence of the above definitions, we have

$$s_\sigma^r(n) = \sum_{j=1}^{n} s_\sigma^r(n; j). \tag{1}$$

We start by considering patterns that belong to Class 1 and we use 12-3 as a representative of this class. Let us define

$$\begin{aligned} u_r(n; b_1, \ldots, b_k) &= s_{12\text{-}3}^r(n; b_1, \ldots, b_k), \\ u_r(n) &= s_{12\text{-}3}^r(n), \\ U_r(x) &= S_{12\text{-}3}^r(x). \end{aligned}$$

Lemma 8. *Let $n \geq 1$. We have $u_r(n; n-1) = u_r(n; n) = u_r(n-1)$ and*

$$u_r(n; i) = \sum_{j=1}^{i-1} u_r(n-1; j) + \sum_{j=0}^{n-i-1} u_{r-j}(n-1; n-1-j),$$

whenever $1 \leq i \leq n-2$.

Proof. If $a_1 a_2 \cdots a_n$ is any permutation of $[n]$ then

$$(12\text{-}3)a_1 a_2 \cdots a_n = (12\text{-}3)a_2 a_3 \cdots a_n + \begin{cases} n - a_2 & \text{if } a_1 < a_2, \\ 0 & \text{if } a_1 > a_2. \end{cases}$$

Hence,

$$\begin{aligned} u_r(n;i) &= \sum_{j=1}^{i-1} u_r(n;i,j) + \sum_{j=i+1}^{n} u_r(n;i,j) \\ &= \sum_{j=1}^{i-1} u_r(n-1;j) + \sum_{j=i+1}^{n} u_{r-n+j}(n-1;j-1) \\ &= \sum_{j=1}^{i-1} u_r(n-1;j) + \sum_{j=0}^{n-i-1} u_{r-j}(n-1;n-1-j). \end{aligned}$$

For $i = n-1$ or $i = n$ it is easy to see that $u_r(n;i) = u_r(n-1)$. □

Using Lemma 8 we quickly generate the numbers $u_r(n)$; the first few of these numbers are given in Table 1. Given $r \in \mathbb{N}$ we can also use Lemma 8 to find a

$n\backslash r$	0	1	2	3	4	5	6
0	1						
1	1						
2	2						
3	5	1					
4	15	7	1	1			
5	52	39	13	12	2	1	1
6	203	211	112	103	41	24	17
7	877	1168	843	811	492	337	238
8	4140	6728	6089	6273	4851	3798	2956
9	21147	40561	43887	48806	44291	38795	33343
10	115975	256297	321357	386041	394154	379611	355182

TABLE 1. The number of permutations of length n containing exactly r occurrences of the pattern 12-3.

functional relation determining $U_r(x)$. Here we present such functional relations for $r = 0, 1, 2$ and also explicit formulas for $r = 0, 1$.

Equation 1 tells us how to compute $u_r(n)$ if we are given the numbers $u_r(n;i)$. For the case $r = 0$ Lemma 9, below, tells us how to do the converse.

Lemma 9. *If* $1 \leq i \leq n-2$ *then*

$$u_0(n;i) = \sum_{j=0}^{i-1} \binom{i-1}{j} u_0(n-2-j).$$

Proof. For $n = 1$ the identity is trivially true. Assume the identity is true for $n = m$. We have

$$\begin{aligned} u_0(m+1;i) &= \sum_{j=1}^{i-1} u_0(m;j) + u_0(m-1) && \text{by Lemma 8} \\ &= \sum_{j=1}^{i-1}\sum_{k=0}^{j-1} \binom{j-1}{k} u_0(m-2-k) + u_0(m-1) && \text{by the induction hypothesis} \\ &= \sum_{j=1}^{i-1}\sum_{k=j-1}^{i-2} \binom{k}{j-1} u_0(m-1-j). \end{aligned}$$

Using the familiar equality $\binom{1}{k} + \binom{2}{k} + \cdots + \binom{n}{k} = \binom{n+1}{k+1}$ we then get

$$u_0(m+1;i) = \sum_{j=1}^{i-1} \binom{i-1}{j} u_0(m-1-j).$$

Thus the identity is true for $n = m+1$ and by the principle of induction the desired identity is true for all $n \geq 1$. □

The following proposition is a direct consequence of Proposition 3. However, we give a different proof. The proof is intended to illustrate the general approach. It is advisable to read this proof before reading the proof of Theorem 4′ below.

Proposition 10. *The ordinary generating function for the number of* (12–3)*-avoiding permutations of length* n *is*

$$U_0(x) = \sum_{k\geq 0} \frac{x^k}{(1-x)(1-2x)\cdots(1-kx)}.$$

Proof. We have

$$\begin{aligned} u_0(n) &= \sum_{k=1}^{n} u_0(n;k) && \text{by Equation 1} \\ &= 2u_0(n-1) + \sum_{i=1}^{n-2}\sum_{j=0}^{i-1} \binom{i-1}{j} u_0(n-2-j) && \text{by Lemma 8 and 9} \\ &= u_0(n-1) + \sum_{i=0}^{n-2} \binom{n-2}{i} u_0(n-1-i) && \text{by } \sum_{i=k}^{n}\binom{i}{k} = \binom{n+1}{k+1} \\ &= u_0(n-1) + \sum_{i=0}^{n-2} \binom{n-2}{i} u_0(i+1). \end{aligned}$$

Therefore, by Lemma 7, we have

$$U_0(x) = xU_0(x) + 1 - x + xU_0\left(\frac{x}{1-x}\right),$$

which is equivalent to

$$U_0(x) = 1 + \frac{x}{1-x} U_0\left(\frac{x}{1-x}\right).$$

An infinite number of applications of this identity concludes the proof. □

We now derive a formula for $U_1(x)$ that is somewhat similar to the one for $U_0(x)$. The following lemma is a first step in this direction.

Lemma 11. *If $1 \leq i \leq n-2$ then*

$$u_1(n;i) = \sum_{j=0}^{i-1} \binom{i-1}{j} u_1(n-2-j) + u_0(n;i).$$

Proof. For $n = 1$ the identity is trivially true. Assume the identity is true for $n = m$. Lemma 8 and the induction hypothesis imply

$$\begin{aligned} u_1(m+1;i) &= \sum_{j=1}^{i-1} u_1(m;j) + u_1(m-1) + u_0(m-1) \\ &= \sum_{j=0}^{i-1} \binom{j-1}{k} u_1(m-1-j) + \sum_{j=1}^{i-1} u_0(m;j) + u_0(m-1). \end{aligned}$$

In addition, Lemma 9 implies

$$\begin{aligned} u_0(m+1;i) &= \sum_{j=1}^{i-1} \sum_{k=0}^{j-1} \binom{j-1}{k} u_0(n-2-k) + u_0(n-1) \\ &= \sum_{j=0}^{i-1} \binom{i-1}{j} u_0(n-1-j) \\ &= \sum_{j=1}^{i-1} u_0(m;j) + u_0(m-1). \end{aligned}$$

Thus the identity is true for $n = m+1$ and by the principle of induction the desired identity is true for all $n \geq 1$. □

Next, we rediscover Theorem 4.

Theorem 4′. *Let $u_1(n)$ be the number of permutations of length n containing exactly one occurrence of the pattern 12-3 and let B_n be the nth Bell number. The numbers $u_1(n)$ satisfy the recurrence*

$$u_1(n+2) = 2u_1(n+1) + \sum_{k=0}^{n-1} \binom{n}{k} \big[u_1(k+1) + B_{k+1}\big],$$

whenever $n \geq -1$, with the initial condition $u_1(0) = 0$.

Proof. Similarly to the proof of Proposition 10, we use Equation 1, Lemma 8, 9, and 11 to get

$$\begin{aligned} u_1(n) &= 2u_1(n-1) + \sum_{i=1}^{n-2} \left[\sum_{j=0}^{i-1} \binom{i-1}{j} u_1(n-2-j) + u_0(n;i) \right] \\ &= 2u_1(n-1) + \sum_{i=1}^{n-2} \sum_{j=0}^{i-1} \binom{i-1}{j} \big(u_1(n-2-j) + u_0(n-2-j)\big) \\ &= u_1(n-1) - u_0(n-1) + \sum_{i=0}^{n-2} \binom{n-2}{i} \big(u_1(i+1) + u_0(i+1)\big) \\ &= 2u_1(n-1) + \sum_{i=0}^{n-3} \binom{n-2}{i} \big(u_1(i+1) + u_0(i+1)\big). \end{aligned}$$

□

Corollary 12. *The ordinary generating function, $U_1(x)$, for the number of permutations of length n containing exactly one occurrence of the pattern* 12-3 *satisfies the functional equation*

$$U_1(x) = \frac{x}{1-x}\Big(U_1\Big(\frac{x}{1-x}\Big) + U_0\Big(\frac{x}{1-x}\Big) - U_0(x)\Big).$$

Proof. The result follows from Theorem 4 together with Lemma 7. □

Corollary 13. *The ordinary generating function for the number of permutations of length n containing exactly one occurrence of the pattern* 12-3 *is*

$$U_1(x) = \sum_{n\geq 1} \frac{x}{1-nx} \sum_{k\geq 0} \frac{kx^{k+n}}{(1-x)(1-2x)\cdots(1-(k+n)x)}.$$

Proof. We simply apply Corollary 12 an infinite number of times and in each step we perform some rather tedious algebraic manipulations. □

Theorem 14. *The ordinary generating function, $U_2(x)$, for the number of permutations of length n containing exactly two occurrences of the pattern* 12-3 *satisfies the functional equation*

$$\begin{aligned} U_2(x) = \frac{x}{(1-x)^2(1-2x)}\Big(& U_2\Big(\frac{x}{1-x}\Big) - (1-x)U_2(x) + \\ & U_1\Big(\frac{x}{1-x}\Big) - (1-x)^2U_1(x) + \\ & U_0\Big(\frac{x}{1-x}\Big) - (1-x)^2U_0(x) \Big). \end{aligned}$$

Proof. The proof is similar to the proofs of Lemma 11, Theorem 4' and Corollary 12, and we only sketch it here.

Lemma 8 yields

$$\begin{aligned} u_2(n;n) &= u_2(n-1) \\ u_2(n;n-1) &= u_2(n-1) \\ u_2(n;n-2) &= u_2(n-1) - u_2(n-2) + u_1(n-2) \end{aligned}$$

and, by means of induction,

$$u_2(n;i) = u_1(n;i) + u_0(n;i) - u_0(n-1;i) + \sum_{j=0}^{i-1} \binom{i-1}{j} u_2(n-2-j),$$

whenever $1 \leq i \leq n-3$. Therefore, $u_2(0) = u_2(1) = u_2(2) = 0$ and

$$\begin{aligned} u_2(n) = {} & 3u_2(n-1) - u_2(n-2) + u_1(n-2) + \\ & \sum_{i=1}^{n-3} \binom{n-3}{i} (u_2(n-1-i) + u_1(n-1-i) + u_0(n-1-i) - u_0(n-2-i)). \end{aligned}$$

whenever $n \geq 3$. Thus, the result follows from Lemma 7. □

We now turn our attention to patterns that belong to Class 2 and we use 23-1 as a representative of this class. The results found below regarding the 23-1 pattern are very similar to the ones previously found for the 12-3 pattern, and so are the proofs; therefore we choose to omit most of the proofs. However, we give the necessary lemmas from which the reader may construct her/his own proofs.

Define

$$\begin{aligned} v_r(n; b_1, \ldots, b_k) &= s^r_{23\text{-}1}(n; b_1, \ldots, b_k), \\ v_r(n) &= s^r_{23\text{-}1}(n), \\ V_r(x) &= S^r_{23\text{-}1}(x). \end{aligned}$$

If $a_1 a_2 \cdots a_n$ is any permutation of $[n]$ then

$$(23\text{-}1)a_1 a_2 \cdots a_n = (23\text{-}1)a_2 a_3 \cdots a_n + \begin{cases} a_1 - 1 & \text{if } a_1 < a_2, \\ 0 & \text{if } a_1 > a_2. \end{cases}$$

Lemma 15. *Let $n \geq 1$. We have $v_r(n;1) = v_r(n;n) = v_r(n-1)$ and*

$$v_r(n;i) = \sum_{j=1}^{i-1} v_r(n-1;j) + \sum_{j=i}^{n-1} v_{r-i+1}(n-1;j),$$

whenever $2 \leq i \leq n-1$.

Using Lemma 15 we quickly generate the numbers $v_r(n)$; the first few of these numbers are given in Table 2.

$n\backslash r$	0	1	2	3	4	5	6
0	1						
1	1						
2	2						
3	5	1					
4	15	6	3				
5	52	32	23	10	3		
6	203	171	152	98	62	22	11
7	877	944	984	791	624	392	240
8	4140	5444	6460	6082	5513	4302	3328
9	21147	32919	43626	46508	46880	41979	36774
10	115975	208816	304939	360376	396545	393476	377610

TABLE 2. The number of permutations of length n containing exactly r occurrences of the pattern 23-1.

Lemma 16. *If $2 \leq i \leq n-1$ then*

$$v_0(n;i) = \sum_{j=0}^{i-2} \binom{i-2}{j} v_0(n-2-j).$$

Proposition 17. *The ordinary generating function for the number of (23-1)-avoiding permutations of length n is*

$$V_0(x) = \sum_{k \geq 0} \frac{x^k}{(1-x)(1-2x)\cdots(1-kx)}.$$

Lemma 18. *If $2 \leq i \leq n-1$ then*

$$v_1(n;i) = \sum_{j=0}^{i-2} \binom{i-2}{j} v_1(n-2-j) + v_0(n;i-1) - v_0(n-1,i-1).$$

Theorem 6′. *Let $v_1(n)$ be the number of permutations of length n containing exactly one occurrence of the pattern* 23-1 *and let B_n be the nth Bell number. The numbers $v_1(n)$ satisfy the recurrence*

$$v_1(n+1) = v_1(n) + \sum_{k=1}^{n-1} \left[\binom{n}{k} v_1(k) + \binom{n-1}{k-1} B_k \right],$$

whenever $n \geq 0$, with the initial condition $v_1(0) = 0$.

Corollary 19. *The ordinary generating function for the number of permutations of length n containing exactly one occurrence of the pattern* 23-1 *satisfies the functional equation*

$$V_1(x) = \frac{x}{1-x} V_1\Big(\frac{x}{1-x}\Big) + x\bigg(V_0\Big(\frac{x}{1-x}\Big) - V_0(x)\bigg).$$

Corollary 20. *The ordinary generating function for the number of permutations of length n containing exactly one occurrence of the pattern* 23-1 *is*

$$V_1(x) = \sum_{n\geq 1} \frac{x}{1-(n-1)x} \sum_{k\geq 0} \frac{k x^{k+n}}{(1-x)(1-2x)\cdots(1-(k+n)x)}.$$

Theorem 21. *The ordinary generating function, $V_2(x)$, for the number of permutations of length n containing exactly two occurrences of the pattern* 23-1 *satisfies the functional equation*

$$V_2(x) = \frac{x}{1-x}\bigg(V_2\Big(\frac{x}{1-x}\Big) + (1-2x)V_1\Big(\frac{x}{1-x}\Big) + (1-3x+x^2)V_0\Big(\frac{x}{1-x}\Big)\bigg) - x + x^2$$

Proof. By Lemma 5

$$\begin{aligned}
v_2(n;n) &= v_2(n-1)\\
v_2(n;1) &= v_2(n-1)\\
v_2(n;2) &= v_2(n-2) + v_1(n-1) - v_1(n-2)\\
v_2(n;3) &= v_2(n-2) + v_2(n-3) + v_1(n-2) - v_1(n-3) +\\
&\qquad + v_0(n-1) - v_0(n-2) - v_0(n-3)
\end{aligned}$$

and, by means of induction,

$$v_2(n;i) = \sum_{j=0}^{i-2} \binom{i-2}{j} v_2(n-2-j) + v_1(n;i-1) + v_1(n-1;i-1) - v_0(n-1;i-2)$$

for $n-1 \geq i \geq 4$. Thus $v_2(0) = v_2(1) = v_2(2) = 0$ and for all $n \geq 3$

$$\begin{aligned}
v_2(n) = v_2(n-1) &+ \sum_{j=0}^{n-2} \binom{n-2}{j} v_2(n-1-j) +\\
&+ \sum_{j=0}^{n-3} \binom{n-3}{j} \big(v_1(n-1-j) - v_1(n-2-j)\big) +\\
&+ \sum_{j=0}^{n-4} \binom{n-4}{j} \big(v_0(n-1-j) - v_0(n-2-j) - v_0(n-3-j)\big).
\end{aligned}$$

The result now follows from Lemma 7. □

3. Counting occurrences of a pattern of Class 3

We choose 2-13 as our representative for Class 3 and we define $w_r(n)$ as the number of permutations of length n containing exactly r occurrences of the pattern 2-13. We could apply the analytic approach from the previous section to the problem of determining $w_r(n)$. However, a result by Clarke, Steingrímsson and Zeng [10, Corollary 11] provides us with a better option.

Theorem 22. *The following Stieltjes continued fraction expansion holds*

$$\sum_{\pi\in\mathcal{S}} x^{1+(12)\pi} y^{(21)\pi} p^{(2\text{-}31)\pi} q^{(31\text{-}2)\pi} t^{|\pi|} = \cfrac{1}{1-\cfrac{x[1]_{p,q}t}{1-\cfrac{y[1]_{p,q}t}{1-\cfrac{x[2]_{p,q}t}{1-\cfrac{y[2]_{p,q}t}{\ddots}}}}}$$

where $[n]_{p,q} = q^{n-1} + pq^{n-2} + \cdots + p^{n-2}q + p^{n-1}$.

Proof. In [10, Corollary 11] Clarke, Steingrímsson and Zeng derived the following continued fraction expansion

$$\sum_{\pi\in\mathcal{S}} y^{\operatorname{des}\pi} p^{\operatorname{Res}\pi} q^{\operatorname{Ddif}\pi} t^{|\pi|} = \cfrac{1}{1-\cfrac{[1]_p t}{1-\cfrac{yq[1]_p t}{1-\cfrac{q[2]_p t}{1-\cfrac{yq^2[2]_p t}{\ddots}}}}}$$

where $[n]_p = 1 + p + \cdots + p^{n-1}$. We refer the reader to [10] for the definitions of Ddif and Res. However, given these definitions, it is easy to see that Res $=$ (2-31) and Ddif $=$ (21) $+$ (2-31) $+$ (31-2). Moreover, des $=$ (21) and $|\pi| = 1 + (12)\pi + (21)\pi$. Thus, substituting $y(xq)^{-1}$ for y, pq^{-1} for p, and xt for t, we get the desired result. □

The following corollary is an immediate consequence of Theorem 22.

Corollary 23. *The bivariate ordinary generating function for the distribution of occurrences of the pattern* 2-13 *admits the Stieltjes continued fraction expansion*

$$\sum_{\pi\in\mathcal{S}} p^{(2\text{-}13)\pi} t^{|\pi|} = \cfrac{1}{1-\cfrac{[1]_p t}{1-\cfrac{[1]_p t}{1-\cfrac{[2]_p t}{1-\cfrac{[2]_p t}{\ddots}}}}}$$

where $[n]_p = 1 + p + \cdots + p^{n-1}$

Using Corollary 23 we quickly generate the numbers $w_r(n)$; the first few of these numbers are given in Table 3.

Corollary 24. *The number of* (2-13)*-avoiding permutations of length* n *is*

$$w_0(n) = \frac{1}{n+1}\binom{2n}{n}.$$

$n\backslash r$	0	1	2	3	4	5	6
0	1						
1	1						
2	2						
3	5	1					
4	14	8	2				
5	42	45	25	7	1		
6	132	220	198	112	44	12	2
7	429	1001	1274	1092	700	352	140
8	1430	4368	7280	8400	7460	5392	3262
9	4862	18564	38556	56100	63648	59670	47802
10	16796	77520	193800	341088	470934	541044	535990

TABLE 3. The number of permutations of length n containing exactly r occurrences of the pattern 2-13.

Proof. This result is explicitly stated in Proposition 3, but it also follows from Corollary 23 by putting $p = 0$. □

Corollary 25. *The number of permutations of length n containing exactly one occurrence of the pattern 2-13 is*

$$w_1(n) = \binom{2n}{n-3}.$$

Proof. For $m > 0$ let

$$W(p,t;m) = \cfrac{1}{1-\cfrac{[m]_p t}{1-\cfrac{[m]_p t}{1-\cfrac{[m+1]_p t}{1-\cfrac{[m+1]_p t}{\ddots}}}}}$$

Note that

$$W(p,t;m) = \cfrac{1}{1-\cfrac{[m]_p t}{1-[m]_p t W(p,t;m+1).}}$$

Assume $m > 1$. Differentiating $W(p,t;m)$ with respect to p and evaluating the result at $p = 0$ we get

$$D_p W(p,t;m)\big|_{p=0} = tC(t)^3 + t^2C(t)^5 + t^2C(t)^4 D_p W(p,t;m+1)\big|_{p=0}$$

where $C(t) = W(0,t,1)$ is the generating function for the Catalan numbers. Applying this identity an infinite number of times we get

$$D_p W(p,t,m)\big|_{p=0} = tC(t)^3 + t^2C(t)^5 + t^3C(t)^7 + \cdots = \frac{tC(t)^3}{1-tC(t)^2}.$$

On the other hand, $D_p W(p,t;1)\big|_{p=0} = t^2C(t)^4 D_p W(p,t;2)\big|_{p=0}$. Combining these two identities we get

$$D_p W(p,t;1)\big|_{p=0} = \frac{t^3C(t)^7}{1-tC(t)^2}.$$

Since $\sum_{n\geq 0} w_1(n)t^n = D_pW(p,t;1)\big|_{p=0}$ the proof is completed on extracting coefficients in the last identity. □

The proofs of the following two corollaries are similar to the proof of Corollary 25 and are omitted.

Corollary 26. *The number of permutations of length n containing exactly two occurrences of the pattern* 2-13 *is*

$$w_2(n) = \frac{n(n-3)}{2(n+4)}\binom{2n}{n-3}.$$

Corollary 27. *The number of permutations of length n containing exactly three occurrences of the pattern* 2-13 *is*

$$w_3(n) = \frac{1}{3}\binom{n+2}{2}\binom{2n}{n-5}.$$

As a concluding remark we note that there are many questions left to answer. What is, for example, the formula for $w_k(n)$ in general? What are the combinatorial explanations of $ns^1_{1-2-3}(n) = 3s^1_{2-13}(n)$ and

$$(n+3)(n+2)(n+1)s^1_{2-13}(n) = 2n(2n-1)(2n-2)s^1_{2-1-3}(n)?$$

In addition, Corollary 25 obviously is in need of a combinatorial proof.

References

[1] N. Alon and E. Friedgut. On the number of permutations avoiding a given pattern. *J. Combin. Theory Ser. A*, 89(1):133–140, 2000.

[2] M. D. Atkinson. Restricted permutations. *Discrete Math.*, 195(1-3):27–38, 1999.

[3] E. Babson and E. Steingrímsson. Generalized permutation patterns and a classification of the Mahonian statistics. *Sém. Lothar. Combin.*, 44:Art. B44b, 18 pp. (electronic), 2000.

[4] M. Bóna. Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A*, 80(2):257–272, 1997.

[5] M. Bóna. The number of permutations with exactly r 132-subsequences is P-recursive in the size! *Adv. in Appl. Math.*, 18(4):510–522, 1997.

[6] M. Bóna. Permutations avoiding certain patterns: the case of length 4 and some generalizations. *Discrete Math.*, 175(1-3):55–67, 1997.

[7] M. Bóna. Permutations with one or two 132-subsequences. *Discrete Math.*, 181(1-3):267–274, 1998.

[8] T. Chow and J. West. Forbidden subsequences and Chebyshev polynomials. *Discrete Math.*, 204(1-3):119–128, 1999.

[9] A. Claesson. Generalized pattern avoidance. *European J. Combin.*, 22(7):961–971, 2001.

[10] R.J. Clarke, E. Steingrímsson, and J. Zeng. New Euler-Mahonian statistics on permutations and words. *Adv. in Appl. Math.*, 18(3):237–270, 1997.

[11] S. Elizalde and M. Noy. Enumeration of subwords in permutations. In *Formal power series and algebraic combinatorics (Tempe, 2001)*, pages 179–189. Arizona State University, 2001.

[12] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete mathematics*. Addison-Wesley Publishing Company, Reading, MA, second edition, 1994.

[13] T. Mansour. Permutations containing and avoiding certain patterns. In *Formal power series and algebraic combinatorics (Moscow, 2000)*, pages 704–708. Springer, Berlin, 2000.

[14] T. Mansour and A. Vainshtein. Counting occurrences of 132 in a permutation. *To appear in: Adv. Appl. Math.*, 2001.

[15] J. Noonan. The number of permutations containing exactly one increasing subsequence of length three. *Discrete Math.*, 152(1-3):307–313, 1996.

[16] J. Noonan and D. Zeilberger. The enumeration of permutations with a prescribed number of "forbidden" patterns. *Adv. in Appl. Math.*, 17(4):381–407, 1996.

[17] A. Robertson. Permutations containing and avoiding 123 and 132 patterns. *Discrete Math. Theor. Comput. Sci.*, 3(4):151–154 (electronic), 1999.

[18] R. Simion and F. W. Schmidt. Restricted permutations. *European J. Combin.*, 6(4):383–406, 1985.

[19] Z. Stankova. Forbidden subsequences. *Discrete Math.*, 132(1-3):291–316, 1994.

[20] Z. Stankova. Classification of forbidden subsequences of length 4. *European J. Combin.*, 17(5):501–517, 1996.

[21] J. West. Generating trees and the Catalan and Schröder numbers. *Discrete Math.*, 146(1-3):247–262, 1995.

Matematik, Chalmers tekniska högskola och Göteborgs universitet, S-412 96 Göteborg, Sweden

E-mail address: claesson@math.chalmers.se

LaBRI, Univerité Bordeaux I, 351 cours de la Libération, 33405 Talence Cedex, France

E-mail address: toufik@labri.fr

COUNTING DISTINCT STRINGS

Dennis Moore

School of Computing
Curtin University of Technology

W. F. Smyth

Department of Computer Science & Systems
McMaster University
tel. 1-905-525-9140 ext. 23436
e-mail: smyth@mcmaster.ca

School of Computing
Curtin University of Technology

Dianne Miller

Department of Computer Science & Systems
McMaster University

KEYWORDS

string, word, algorithm, testing, distinct

ABSTRACT

This paper discusses how to count and generate strings that are "distinct" in two senses: p-distinct and b-distinct. Two strings x on alphabet A and x' on alphabet A' are said to be p-distinct iff they represent distinct "patterns"; that is, iff there exists no one-one mapping from A to A' that transforms x into x'. Thus aab and baa are p-distinct while aab and ddc are p-equivalent. On the other hand, x and x' are said to be b-distinct iff they give rise to distinct border (failure function) arrays: thus aab with border array 010 is b-distinct from aba with border array 001. The number of p-distinct (respectively, b-distinct) strings of length n formed using exactly k different letters is the $[k, n]$ entry in an infinite p' (respectively, b') array. Column sums $p[n]$ and $b[n]$ in these arrays give the number of distinct strings of length n. We present algorithms to compute, in constant time per string, all p-distinct (respectively, b-distinct) strings of length n formed using exactly k letters, and we also show how to compute all elements $p'[k, n]$ and $b'[k, n]$. These ideas and results have application to the efficient generation of appropriate test data sets for many string algorithms.

1 INTRODUCTION

When is a string "distinct" from another? The answer to this question depends on how we intend to process the string. For some purposes we might choose to regard $x = abbcc$ and $x' = bccaa$ as distinct; if, however, we regard the letters of the alphabet as interchangeable, so that x and x' can be seen as conforming to the same "pattern", then we might prefer to think of x as being equivalent to x' in a well-defined sense. This would be true, for example, if we were generating test data for an algorithm which recognized no ordering of the alphabet (say, an algorithm to compute all repetitions [1] in a string): in this case, if the algorithm executed correctly on input x, it would do so also on input x'.

To make this idea precise, let

$$x = x[1]x[2]\cdots x[n] = x[1..n], \quad x' = x'[1]x'[2]\cdots x'[n] = x'[1..n]$$

denote arbitrary finite strings of length $|x| = n \geq 1$. We say that x is *p-equivalent* to x' if and only if, for all integers i and j satisfying $1 \leq i \leq j \leq n$,

$$x[i] = x[j] \quad \Leftrightarrow \quad x'[i] = x'[j].$$

Clearly p-equivalence is an equivalence relation, breaking down the strings of length n into equivalence classes. Strings that are not p-equivalent are said to be *p-distinct*.

Another interpretation of "distinctness" is possible. Recall that a string x is said to have *border* u if and only if u is a proper prefix and suffix of x. For example, $x = abaabaab$ has borders $u = \epsilon$ (the empty string), ab and $abaab$, of lengths 0, 2 and 5, respectively. The *border array* $\beta_n = \beta[1..n]$ corresponding to $x_n = x[1..n]$ is a string defined on the integer alphabet $\{0, 1, \ldots, n-1\}$ in which, for every integer $j \in 1..n$, $\beta[j]$ is the length of the longest border of $x_j = x[1..j]$. ($\beta[j]$ is also referred to as the "failure function" of x_j [2].)

We say that two strings are *b-equivalent* if and only if they give rise to identical border arrays. Strings that are not b-equivalent are said to be *b-distinct*. Thus, for example, even though $x_5 = ababb$ and $x'_5 = ababc$ are p-distinct, we find that they are nevertheless b-equivalent since both correspond to the border array $\beta_5 = 00120$. On the other hand, x_5 and $x''_5 = abacb$ are b-distinct since they give rise to distinct border arrays 00120 and 00100, respectively. It is clear then that each distinct valid border array determines an equivalence class of b-equivalent strings. Observe that two b-distinct strings are necessarily also p-distinct (so that p-equivalent strings are necessarily also b-equivalent); as we have just seen, the converse is not true.

In this paper we consider the two kinds of distinctness described above; for each, and for all positive integers k and n, we show how to

* generate (in only constant time per string) all distinct strings of length n formed using exactly k letters;
* count the number of all such strings.

In particular, we shall see that the number of p-distinct patterns of length n formed using exactly k letters is $\left\{ {n \atop k} \right\}$, a Stirling number of the second kind, a fact apparently not previously observed. We shall see therefore (equation (2.5)) that the total number of p-distinct strings of length n using at most k letters is reduced by an asymptotic factor of $1/k!$ from the number of such strings that are distinct in the ordinary sense. Moreover, the computation of b-distinct patterns leads to a sequence of integers that is apparently new, and that represents a decline, by a further exponential factor, from the number of p-distinct patterns (Theorem 3.3(f) and equation (3.1)). Algorithms for generating distinct strings have been implemented in a software package for the testing of string algorithms [3].

2 DISTINCT PATTERNS

In this section we discuss p-distinct strings: how to count them and how to generate them. In order to do so, it is convenient to identify a unique representative of each p-distinct equivalence class. We therefore introduce a countably infinite *standard alphabet*

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_k, \ldots.\}, \qquad \ldots(2.1)$$

with subalphabets $\Lambda_k = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ for every integer $k \geq 1$. We suppose the letters of Λ to be naturally ordered according to $\lambda_1 < \lambda_2 < \cdots < \lambda_k < \cdots$. Then, given any string $x = x[1..n]$ on any alphabet A, we define the *p-canonical* string x^* corresponding to x to be the lexicographically least string on Λ that is p-equivalent to x. It is clear that x^* satisfies the following property:

(P) For every positive integer j, the first occurrence (if any) of λ_j in x^* precedes the first occurrence of λ_{j+1}.

We first concern ourselves with the problem of counting the number $p'[k,n]$ of p-canonical strings x^* of length n formed using exactly the letters of Λ_k. We imagine these values to be laid out in an infinite two-dimensional array called the p' array.

Theorem 2.1 For any positive integers n and k:
(a) $p'[1,n] = 1$;
(b) if $k > n$, $p'[k,n] = 0$;
(c) $p'[k,k] = 1$;
(d) if $k \geq 2$ and $n \geq 2$, $p'[k,n] = p'[k-1,n-1] + kp'[k,n-1]$.

Proof (a) For $k = 1$, the only p-canonical string is $x^* = \lambda_1^n$.
(b) By property (P), no p-canonical string x^* can contain a letter λ_k, $k > n$.

(c) Again by property (P), there exists exactly one p-canonical string of length k formed using exactly k distinct letters: $x^* = \lambda_1\lambda_2\cdots\lambda_k$.

(d) Let $\pi_1 = p'[k-1, n-1]$ denote the number of distinct p-canonical strings of length $n-1$ that include exactly the $k-1$ letters of Λ_{k-1}. Denote these strings by

$$S_1 = \{x_1, x_2, \ldots, x_{\pi_1}\}.$$

Then for every integer i satisfying $1 \le i \le \pi_1$, each string

$$x_i\lambda_k \qquad \ldots(2.2)$$

is distinct and p-canonical.

Similarly, let $\pi_2 = p'[k, n-1]$ denote the number of distinct p-canonical strings of length $n-1$ on exactly k distinct letters Λ_k. Denote these strings by

$$S_2 = \{y_1, y_2, \ldots, y_{\pi_2}\}.$$

Then for every integer i satisfying $1 \le i \le \pi_2$, the k strings

$$\{y_i\lambda_1, y_i\lambda_2, \ldots, y_i\lambda_k\} \qquad \ldots(2.3)$$

must all be distinct and p-canonical. Further, since the distinct final letter occurs at least twice in each string, each of these strings is distinct from any of the strings (2.2). Thus $p'[k, n] \ge p'[k-1, n-1] + kp'[k, n-1]$.

Suppose now that x^* is a p-canonical string of length n formed using exactly the letters Λ_k. Let $x^* = y^*\lambda_i$. If λ_i occurs in y^*, then $y^* \in S_2$ and therefore x^* is one of the strings (2.3). Otherwise, by property (P), λ_k cannot occur in y^* either, and so $i = k$, $y^* \in S_1$, and x^* is one of the strings (2.2). We conclude that $p'[k, n] \le p'[k-1, n-1] + kp'[k, n-1]$, and so the result is proved. □

The recurrence relation of Theorem 2.1(d) is well-known; with the initial values specified by Theorem 2.1(a)-(c), it defines the Stirling numbers $\left\{{n \atop k}\right\}$ of the second kind [4, 5]. Hence

$$p'[k, n] = \left\{{n \atop k}\right\} \qquad \ldots(2.4)$$

for all positive integers n and k. In fact, as we illustrate with an example, the correspondence between classical Stirling numbers and our $p'[k, n]$ values can be made in another way. A common definition [5] of $\left\{{n \atop k}\right\}$ is the number of ways that a set S of n elements can be decomposed into k nonempty nonintersecting subsets whose union is S. To see how this definition corresponds to $p'[k, n]$, consider the case $n = 4$, $k = 2$. If

we write down the seven strings counted by $p'[2,4]$ and collect into $k=2$ subsets the *indices* of identical letters in these strings, we find that each pair of subsets is a unique (because each string is distinct) decomposition of $\{1,2,3,4\}$ into nonempty (because each of the k letters occurs) nonintersecting (because each position contains exactly one letter) subsets:

1234	
$aaab$	$\{1,2,3\}\ \{4\}$
$aaba$	$\{1,2,4\}\ \{3\}$
$aabb$	$\{1,2\}\ \{3,4\}$
$abaa$	$\{1,3,4\}\ \{2\}$
$abab$	$\{1,3\}\ \{2,4\}$
$abba$	$\{1,4\}\ \{2,3\}$
$abbb$	$\{1\}\ \{2,3,4\}$

The unions of the pairs of sets in the righthand column exhaust all the possible ways of forming $S=\{1,2,3,4\}$ from $k=2$ nonempty nonintersecting subsets.

Theorem 2.1(d) provides an iterative method of computing $p'[k,n]$ and various formulæ for direct computation are available in the literature [6]. Observe that, for any fixed value of k, the partial column sum $\sum_{i=1}^{k} p'[i,n]$ is the number of p-distinct strings of length n formed from at most k letters. Since for n large with respect to k almost all of these strings contain exactly k letters, it follows that

$$\lim_{n\to\infty}\left(\sum_{i=1}^{k} p'[i,n]\Big/\frac{k^n}{k!}\right)=1. \qquad \ldots(2.5)$$

In the usual meaning of distinctness in strings, the number of distinct strings of length n formed from at most k letters is k^n. Thus (2.5) tells us that using p-distinct strings on an alphabet of fixed size k reduces the number of strings that need to be generated by an asymptotic factor of $1/k!$. Of particular interest is the case

$$p[n]\equiv\sum_{i=1}^{n} p'[i,n],$$

the number of p-distinct strings of length n, known in the literature as Bell numbers [7]. These numbers also can be computed directly or iteratively in various ways [6, 8], in particular using

$$p[n]=\sum_{j=0}^{n-1}\binom{n-1}{j}p[j], \qquad \ldots(2.6)$$

$p[0] \equiv 1$, that avoids any reference to the p' values. The first few Bell numbers are $p[1] = 1$, $p[2] = 2$, $p[3] = 5$, $p[4] = 15$, $p[5] = 52$, $p[6] = 203$. By contrast, there are 46,656 distinct (in the ordinary sense) strings of length 6 on an alphabet of 6 letters.

We conclude this section with a discussion of the generation of p-canonical strings. It is clear from the proof of Theorem 2.1(d) that, in order to generate all the strings counted by $p'[k, n]$, we

* append λ_k to the strings counted by $p'[k-1, n-1]$;
* append $\lambda_1, \lambda_2, \ldots, \lambda_k$ to the strings counted by $p'[k, n-1]$.

This observation gives rise to straightforward recursive algorithms to generate either *all* the p-canonical strings x^* counted by $p'[k, n]$ or else *pseudorandom* strings x^*. The generation of each pseudorandom string will necessarily require $\Theta(n)$ time, but the generation of all p-canonical strings of length n can actually be accomplished in constant time per string by making use of a rooted tree structure T_n of height n, as described below.

The nodes of T_n may be thought of as pairs (λ, k), where λ is a letter of Λ and k is the number of distinct letters λ found in the nodes which lie on the path to the current node from the root. T_1 consists of the single root node $(\lambda_1, 1)$, and for every integer $n \geq 2$, T_n is formed by adding the following children to every leaf node (λ, k) of T_{n-1}:

$$(\lambda_1, k), (\lambda_2, k), \ldots, (\lambda_k, k), (\lambda_{k+1}, k+1).$$

It is easy to see that T_n has exactly $p[n]$ leaf nodes and that the letters found on the paths to these nodes from the root give exactly the $p[n]$ p-canonical strings x^* of length n. Thus the generation of these strings x^* is accomplished simply by generating T_n. Observe that, for every integer $n \geq 2$, T_n is formed from T_{n-1} by appending $p[n]$ leaf nodes, a task requiring $\Theta(p[n])$ time. Since by (2.6) $p[n] \geq 2p[n-1]$, it follows that T_n can be constructed in $\Theta(p[n])$ time.

Theorem 2.2 For every positive integer n, all $p[n]$ p-canonical strings of length n can be computed in $\Theta(p[n])$ time and represented in $\Theta(p[n])$ space. □

We may establish a similar result for the generation of all p-canonical strings counted by $p'[k, n]$. In this case we generate only the subtree of T_n whose paths of length n terminate at a vertex whose label is (λ, k) for any letter λ; these paths represent exactly the $p'[k, n]$ p-canonical strings of length n which contain exactly k letters. Thus in this case only the nodes on these paths need to be computed, and so we have

Theorem 2.3 For all positive integers k and $n \geq k$, all $p'[k, n]$ p-canonical strings of length n formed using exactly k letters can be computed in $O(kp'[k, n])$ time and represented in $O(kp'[k, n])$ space.

Proof The recurrence relation of Theorem 2.1(d) implies that, in order to compute the strings counted by $p'[k,n]$, k diagonal entries

$$p'[1,n-j-k+1]=1,\ p'[2,n-j-k+2],\ldots,\ p'[k,n-j]$$

need to be computed for every integer $j = n-k, n-k-1, \ldots, 0$. Thus for $j = n-k$, the k elements

$$p'[1,1]=1,\ p'[2,2]=1,\ldots,\ p'[k,k]=1$$

in the main diagonal of the p' array are computed, while for $j < n-k$ the elements in the diagonal distance $n-k-j$ above the main diagonal are computed. For every valid integer j, let

$$D_{k,n-j} = \sum_{i=0}^{k-1} p'[k-i, n-i-j]$$

denote the sum of the terms in the $(n-k-j)^{\text{th}}$ diagonal. Observe that, since $p'[k,n-j]$ is the largest element in its diagonal, $kp'[k,n-j] \geq D_{k,n-j}$, with equality if and only if $j = n-k$. Further, it follows from the recurrence relation that

$$p'[k,n-j] > kp'[k,n-j-1] \geq D_{k,n-j-1},$$

provided $n-j > 1$. Hence

$$\begin{aligned}\sum_{j=0}^{n-k} D_{k,n-j} &\leq kp'[k,n] + p'[k,n](1+1/k+\cdots+1/k^{n-k-1}) \\ &\leq (k+2)p'[k,n],\end{aligned}$$

and the result follows. □

We remark finally that the tree T_n may be traversed in various ways corresponding to various orderings of the p-canonical strings. For example, preorder traversal of T_n (or any subtree of it generated by $p'[k,n]$) yields the strings in lexicographic order; so also does postorder traversal if the empty letter is assumed to sort largest. In fact, if each string of T_n can be discarded after generation, then the strings determined by T_n can actually be generated using only $\Theta(n)$ storage, corresponding to either preorder or postorder traversal of T_n. Since by (2.6) $p[n] \geq 2^{n-1}$, this reduces the storage requirement to $O(\log p[n])$.

3 DISTINCT BORDER ARRAYS

In this section we consider how to generate and how to count b-distinct strings. We begin with a series of lemmas that show how b-distinct strings of length $n+1$ can be derived from those of length n.

Among any class of b-equivalent strings, it will again be convenient to identify one *b-canonical* string x^* as a representative of its class: as with p-canonical strings, we choose this string to be the lexicographically least among those strings on the standard alphabet that are in the class. Every class of b-equivalent strings on Λ is of infinite cardinality, but we can simplify matters without loss of generality by restricting such classes only to strings that are also p-canonical. Then, for example, the class of p-canonical b-equivalent strings on Λ corresponding to $\beta_5 = 00100$ is

$$S_5 = \{\lambda_1\lambda_2\lambda_1\lambda_3\lambda_2, \lambda_1\lambda_2\lambda_1\lambda_3\lambda_3, \lambda_1\lambda_2\lambda_1\lambda_3\lambda_4\},$$

with b-canonical element $x_5^* = \lambda_1\lambda_2\lambda_1\lambda_3\lambda_2$.

In order to establish a recurrence to compute a b-canonical string $x_{n+1}^* = x^*[1..n+1]$ from a b-canonical string $x_n^* = x^*[1..n]$, we need to understand how β_{n+1} is computed from β_n. Let $\beta^i[n]$, $i \geq 1$, denote $\beta[\beta^{i-1}[n]]$, where $\beta^0[n] \equiv n$. We state without proof a lemma on which the standard failure function algorithm [2] is based:

Lemma 3.1 Let β_n denote the border array of some string x_n of length $n \geq 1$, and let $k < n$ be the least integer such that $\beta^k[n] = 0$. Then

(a) the borders of x_n are exactly $x_{\beta^i[n]} = x[1..\beta^i[n]]$ for integers $i \in 1..k$;

(b) for any string x_{n+1} with proper prefix x_n, $\beta_{n+1} = \beta_n\beta[n+1]$, where $\beta[n+1] \in \{0, \beta[n]+1, \beta^2[n]+1, \ldots, \beta^k[n]+1\}$. □

This result describes the values that may possibly be assumed by $\beta[n+1]$, given $\beta_n = \beta[1..n]$. We now prove a much stronger result, that the set of values *actually* assumed by $\beta[n+1]$ is independent of the underlying string x_n.

Lemma 3.2 For $n \geq 1$, the values assumed by $\beta[n+1]$ depend only on β_n and the size of the alphabet.

Proof Suppose that there exist two strings x_n and y_n, both defined on alphabets of size α, both with border array β_n. Suppose further that for some letter λ and some integer m, $x_{n+1} = x_n\lambda$ has border array $\beta_{n+1} = \beta_n m$, but that there exists no letter μ such that $y_{n+1} = y_n\mu$ has β_{n+1}. Then $\beta[n+1] = m$ is one of the values specified in Lemma 3.1(b).

First consider the case $m = \beta^i[n] + 1$ for some integer $i \in 1..k$. Since $\beta^i[n] = m-1$, it follows that

$$y[1..m-1] = y[n+2-m..n].$$

Since $\beta^i[n+1] \neq m$, we observe that setting $y[n+1] = y[m]$ implies

$$y[1..m'] = y[n+2-m'..n]$$

for some $m' > m$. But this means that

$$y[1..m'-1] = y[n+2-m'..n],$$

so that $\beta[n] = m'-1 > m-1$, a contradiction. Thus the lemma holds for every $m = \beta^i[n]+1$.

Now suppose that $m = 0$. Then every one of the α possible choices $y[n+1] = \mu$ yields a unique value $\beta[n+1] \neq 0$, while at least one choice $x[n+1] = \lambda$ gives rise to $\beta[n+1] = 0$. Hence there exists $m' > 0$ such that $y[n+1]$ yields $\beta[n+1] = m'$ while $x[n+1]$ does not yield $\beta[n+1] = m'$, in contradiction to the previous case.

We conclude that β_{n+1} is a border array of some x_{n+1} if and only if it is a border array of some y_{n+1}. □

This fundamental result raises the possibility, discussed below, that β_{n+1} can be computed from β_n without reference to any specific string. We can use the result immediately, however, to show that every b-canonical string x^*_{n+1} must have a b-canonical string as a prefix:

Lemma 3.3 For $n \geq 1$, every b-canonical string $x^*_{n+1} = x^*_n\lambda$, where x^*_n is also b-canonical and λ is some letter of the standard alphabet.

Proof Suppose $x^*_{n+1} = x_n\lambda$ with associated border array β_{n+1}, where x_n is a string of length n that is not b-canonical. Suppose that x_n has border array β_n. Then there exists a string $y_n < x_n$ with border array β_n. Hence by Lemma 3.2 there also exists $y_{n+1} = y_n\lambda'$ with border array β_{n+1}, where $y_{n+1} < x^*_{n+1}$. But then x^*_{n+1} is not b-canonical, a contradiction. □

It is thus clear that *all* of the b-canonical strings x^*_{n+1} can be formed from b-canonical strings x^*_n — no other strings need be considered. This foreshadows a tree structure similar to that of Section 2, where strings x^*_{n+1} are children of strings x^*_n. The next lemma provides more exact information about how to generate distinct border arrays β_{n+1} from a given β_n, and also about the form of the associated b-canonical strings x^*_{n+1}.

Lemma 3.4 Suppose a border array β_n corresponds to a b-canonical string x^*_n on the standard alphabet Λ. Then β_n gives rise to exactly κ distinct border arrays β_{n+1} if and only if $x^*_n\lambda_\kappa$ is a b-canonical string that corresponds to $\beta^{(0)}_{n+1} = \beta_n 0$.

Proof Suppose first that $x_{n+1} = x_n^* \lambda_\kappa$ is b-canonical and has only the empty border. Then, since every b-canonical string corresponding to a given border array must be lexicographically least, it follows that there exists no λ_i, $i < \kappa$, such that $x_n^* \lambda_i$ has only the empty border; that is, for every $i \in 1..\kappa - 1$, every $x_n^* \lambda_i$ has a distinct nonempty border.

Now suppose that for some integer $i > \kappa$, the b-canonical string $x_n^* \lambda_i$ has a longest border of length $m > 0$, so that $\beta_{n+1} = \beta_n m$. (Note that in fact, since $m \geq i > \kappa \geq 2$, $m \geq 3$.) It follows from Lemma 3.3 that x_n^* has a b-canonical prefix $x_m^* = x_{m-1}^* \lambda_i$ for some b-canonical string x_{m-1}^*. Moreover, since $x_n^* \lambda_\kappa$ has only the empty border, it follows that the string $x_{m-1}^* \lambda_\kappa$ also has only the empty border. Then for some positive integer $\kappa' \leq \kappa$, $x_{m-1}^* \lambda_{\kappa'}$ is a b-canonical string with only the empty border while $x_{m-1}^* \lambda_i$, $i > \kappa'$, is a b-canonical string with a nonempty border. In other words, we have reduced an instance of a problem for finite positive integers n and κ to an instance of exactly the same problem for finite positive integers $m-1$ and κ'. This reduction can therefore be continued indefinitely, an impossibility which persuades us that there exists no $i > \kappa$ such that $x_n^* \lambda_i$ has a nonempty border. Thus there are exactly κ distinct border arrays β_{n+1}, and sufficiency is proved.

To prove necessity, suppose that there exist exactly κ distinct border arrays β_{n+1}. But then one of them must be $\beta_n 0$ and, as we have just seen, must correspond to $x_n^* \lambda_\kappa$. □

It is noteworthy that Lemma 3.4 does not necessarily hold on a finite alphabet Λ_k; in other words, it holds only if the alphabet is sufficiently large. For example, on the alphabet $\Lambda_3 = \{\lambda_1, \lambda_2, \lambda_3\}$, the b-canonical string $x_7^* = \lambda_1 \lambda_2 \lambda_1 \lambda_3 \lambda_1 \lambda_2 \lambda_1$ has border array $\beta_7 = 0010123$, but there is no $x_8^* = x_7^* \lambda$ on Λ_3 with border array $\beta_8 = 00101230$.

Lemmas 3.2-3.4 suggest an algorithm for generating all b-canonical strings of length n: for every integer $j = 1, 2, \ldots, n-1$, append to each b-canonical string x_j^* single standard letters $\lambda_1, \lambda_2, \ldots$, until for some integer $\kappa \geq 2$, $x_j^* \lambda_\kappa$ has only the empty border. Then the strings $x_j^* \lambda_1, x_j^* \lambda_2, \ldots, x_j^* \lambda_\kappa$ will be exactly the b-canonical strings derived from x_j^*.

To implement this algorithm, we generate a rooted tree T_n', similar to the tree employed in Section 2. Here each node of T_n' is a pair (λ, β), where $\lambda \in \Lambda$ and β denotes the border array entry for λ in the string defined by the labels in the nodes on the path from the root of T_n' to the current node. Thus T_1' consists of the root node $(\lambda_1, 0)$, and for every integer $n \geq 2$, T_n' is formed by adding the children

$$(\lambda_1, \beta_1), (\lambda_2, \beta_2), \ldots, (\lambda_\kappa, 0)$$

to every leaf node of T'_{n-1}. Hence each node of T'_n determines a b-canonical string together with its border array. Denoting by $b[n]$ the number of b-canonical strings of length exactly n, we see that T'_n has exactly $b[n]$ leaf nodes. Thus all $b[n]$ b-canonical strings (and their corresponding border arrays) can be represented simply by appending $b[n]$ children to the leaf nodes of T'_{n-1}, a task requiring $\Theta(b[n])$ time since the border array element contained in each new child can be computed in amortized constant time using the standard failure function algorithm [2]. Since by Lemma 3.4 every non-leaf node of T'_n, $n > 0$, has at least two children, it follows that the number of nodes in each level of T'_n exceeds the number of nodes in all previous levels, hence that T'_{n-1} contains fewer than $b[n]$ nodes, and so can be constructed in $O(b[n])$ time. We have then the analogue to Theorem 2.2:

Theorem 3.1 For every positive integer n, all $b[n]$ b-canonical strings of length n can be computed in $\Theta(b[n])$ time and represented in $\Theta(b[n])$ space. □

We remark that trivial modification to the algorithm outlined above yields an algorithm to compute all the b-canonical strings of length n defined on Λ_k: in computing the children of each node, it is necessary only, as indicated above, to ensure that every child $(\lambda_\kappa, 0) = (\lambda_{k+1}, 0)$ is omitted from the tree. Note also that it is straightforward, using the tree T'_n, to compute b-canonical strings that are "random" in the sense that, at each step, a child x^*_j of x^*_{j-1} is pseudorandomly selected.

It is clear from Lemma 3.4 that there always exist at least two border arrays $\beta^{(0)}_{n+1} = \beta_n 0$ and $\beta^{(m+1)}_{n+1} = \beta_n(m+1)$, where $m = \beta[n]$. The next result shows how to determine whether or not there exists $\beta^{(i)}_{n+1}$, $1 \le i \le m$, and so provides a basis for an algorithm which, given all distinct border arrays β_n, computes all distinct border arrays β_{n+1} without any knowledge of x^*_n. Thus Theorem 3.2 establishes the interesting and nonobvious fact that distinct border arrays of length n can be computed by constructing a tree T''_n whose nodes contain border array elements only. In fact, as observed by a referee, T''_n can like T'_n be constructed in $\Theta(b[n])$ time, but only at a cost of introducing an extra pointer into each node i. Thus no storage is saved using T''_n and it turns out that the algorithm for its construction is considerably more complicated than the one given above for T'_n. The algorithm is therefore not described here in detail. In the following theorem, the notation $j' \to j$ is used to mean that $\beta^i[j'] = j$ for some $i > 0$.

Theorem 3.2 Let $m = \beta[n] \ge 1$. For every integer $i \in 1..m$, there exists a valid border array $\beta^{(i)}_{n+1} = \beta_n i$ if and only if the following conditions all hold:
(a) $\beta[m+1] \not\to i$;
(b) $\beta[m] \to i - 1$;
(c) there exists no integer $i' \to i$ such that $\beta^{(i')}_{n+1} = \beta_n i'$ is valid.

Proof To prove the necessity of the three conditions, suppose first that $\beta_n i$ is a valid

border array. Then there exists a b-canonical string $x^*_{n+1} = x^*_n\lambda$ with a longest border $x^*_i = x^*[1..i]$, where x^*_n has a longest border $x^*_m = x^*[1..m]$, $m \geq i$. Thus $\lambda \equiv x^*[n+1] = x^*[i]$ while $\lambda \neq x^*[m+1]$, since otherwise it would follow that x^*_{n+1} would have a longest border x^*_{m+1}. We conclude that $x^*[m+1] \neq x^*[i]$, from which (a) follows.

To prove (b), observe first that for $i = 1$, (b) is true. Suppose therefore that $i > 1$. But then the fact that $\lambda = x^*[i]$ leads to the conclusion that $x^*[n] = x^*[m] = x^*[i-1]$, hence that $\beta[m] \to i-1$.

To prove (c), suppose on the contrary that for some $i' \to i$, $\beta_n i'$ is a valid border array. But then in order to form a border x^*_i of x^*_{n+1}, a longer border $x^*_{i'}$ is necessarily formed, contradicting the assumption that $\beta_n i$ is a valid border array. Thus (c) also must be true.

To prove sufficiency, suppose that (a), (b) and (c) all hold. Since $\beta[m] \to i-1$, we may choose $\lambda = x^*[i]$ to ensure that x^*_{n+1} has a border of length at least i. Since $\beta[m+1] \not\to i$, we are assured that $x^*[m+1] \neq x^*[i]$, hence that x^*_{n+1} does not have a border of length m. Since by (c) i is a leaf node in B_{n+1}, we are further assured that x^*_{n+1} has no border longer than i. Thus $\beta^{(i)}_{n+1} = \beta_n i$ is a valid border array, as required. □

We turn now to consideration of a b' array analogous to the p' array of Section 2: for positive integers k and n, $b'[k,n]$ denotes the number of b-canonical strings of length n formed using exactly the k standard letters of Λ_k. Then the already-defined quantities $b[n]$ are the column sums in the b' array:

$$b[n] = \sum_{k \geq 1} b'[k,n].$$

As we shall see below (Theorem 3.3(a)), all terms in the n^{th} column of the b' array are zero for $k > \lceil \log_2(n+1) \rceil$; that is, the k^{th} letter of the alphabet does not appear in b-canonical strings of length $n < 2^{k-1}$. For $k \leq \lceil \log_2(n+1) \rceil$, computation of the elements $b'[k,n]$ requires generation of a tree T'''_n in which each node takes the form of a triple (λ, β, i), where as in Section 2 the additional term i counts the number of distinct letters in the b-canonical string represented by the path from the root. Using T'''_n a straightforward algorithm allows $b'[k,n]$ to be computed in $O(b[n])$ time.

In general, it appears to be much more difficult to find well-known expressions for the elements of the b' array than for those of the p' array. However, the following theorem provides enough information to allow useful upper bounds to be stated on $b'[k,n]$ and $b[n]$. It also illustrates the difficulty of expressing these values in closed form.

Theorem 3.3 Given positive integers k and n:

(a) $b'[k,n] = 0$, $k > \lceil \log_2(n+1) \rceil$.

(b) $b'[1,n] = b'[k,2^{k-1}] = 1$.

(c) $b'[2,n] = p'[2,n] = 2^{n-1} - 1$.

(d) Let $\hat{b}[k,n]$ denote the number of strings counted by $b'[k,n]$ which contain λ_k only in position n. Then

$$\hat{b}[3,n] = 2^{n-3} - 2^{\lceil n/2 \rceil - 2} - 2^{n-4} \sum_{j=0}^{\lfloor n/2 \rfloor - 2} \hat{b}[3,j+2]/2^{2j}$$

for every $n \geq 2$.

(e) Let $\tilde{b}[k,n] = b'[k,n] - \hat{b}[k,n]$. Then for every $k \geq 3$ and $n \geq 3$,

$$\tilde{b}[k,n] \geq 2b'[k,n-1].$$

(f) For every nonnegative integer j,

$$b'[k,2^{k-1}+j] \leq p'[k,k+j],$$

with equality holding for $1 \leq k \leq 2$.

Proof (a) The proof is by induction. Observe that the result holds for $n = 1$. We suppose then that it holds for every n satisfying $2^{k-1} \leq n \leq 2^k - 1$ for some positive integer k, and we show that therefore it must hold for values n' satisfying $2^k \leq n' \leq 2^{k+1} - 1$.

By the definition of the b' array, the inductive assumption is equivalent to supposing that over the range of values n, at most k letters $\lambda_1, \lambda_2, \ldots, \lambda_k$ (in ascending order) are required in order to form the b-canonical string x_n corresponding to every border array β_n. Thus the letter λ_{k+1} does not occur in any position less than 2^k of any b-canonical string $x^*_{n'}$, $n' \geq 2^k$.

We need to show that for every n' satisfying $2^k \leq n' \leq 2^{k+1} - 1$, no b-canonical string $x^*_{n'}$ contains λ_{k+2}. Suppose on the contrary that some such $x^*_{n'}$ contains λ_{k+2} as its final letter: $x^*_{n'} = x^*_{n'-1}\lambda_{k+2}$. This can occur only if each of the strings

$$\{x^*_{n'-1}\lambda_1, x^*_{n'-1}\lambda_2, \ldots, x^*_{n'-1}\lambda_{k+1}\}$$

is b-canonical and has a nonempty border. In particular, let $x^*_{n'} = x^*_{n'-1}\lambda_{k+1}$, and let j denote the position of the first occurrence of λ_{k+1} in $x^*_{n'}$. By the inductive hypothesis, $j \geq 2^k$, and so the length of the longest border of $x^*_{n'}$ must exceed $n'/2$. But this implies that $x^*_{n'}[j - (n' - \beta[n'])] = \lambda_{k+1}$,

contradicting the assumption that j is the first occurrence of λ_{k+1}. We conclude that $x^*_{n'-1}\lambda_{k+1}$ cannot have a nonempty border, hence by Lemma 3.4 that no b-canonical string $x^*_{n'}$ contains λ_{k+2}, as required.

(b) $b'[1,n] = 1$ corresponding to the strings λ_1^n, while $b'[k, 2^{k-1}] = 1$ corresponding to the strings

$$\{\lambda_1, \lambda_1\lambda_2, \lambda_1\lambda_2\lambda_1\lambda_3, \lambda_1\lambda_2\lambda_1\lambda_3\lambda_1\lambda_2\lambda_1\lambda_4, \ldots\}.$$

(c) Follows from the observation that for $n = 2$ every p-canonical string is also b-canonical.

(d) To improve readability we make the substitution $\{a, b, c\} \leftarrow \{\lambda_1, \lambda_2, \lambda_3\}$. Then observe that every b-canonical string $x^*_{n-1} = ab * a$ gives rise to a b-canonical string $x^*_n = x^*_{n-1}c$. (Here $ab * a$ denotes a string with prefix ab, suffix a, and zero or more "don't-care" letters in between.) There are 2^{n-4} such b-canonical strings.

For any integer $j \geq 0$, let y_j denote a substring of length j on $\{a, b\}$. Then observe further that every b-canonical string $x^*_{n-1} = ay_1b * ay_1$ gives rise to a b-canonical string $x^*_n = x^*_{n-1}c$: there are $2(2^{n-6})$ such strings.

Next consider $x^*_{n-1} = ay_2b * ay_2$ giving rise to $x^*_n = x^*_{n-1}c$. Here y_2 can take the values aa, ab and bb, but not ba, since the string $ab * a$ has already been counted. Thus in this case there are $(2^2 - 1)2^{n-8}$ new distinct b-canonical strings. Similarly for $x^*_{n-1} = ay_3b * ay_3$: here y_3 omits the values baa and bba, again since $ab*a$ has already been omitted. Thus we count $(2^4-2)2^{n-10}$ new distinct strings.

We see in general that corresponding to each $x^*_{n-1} = ay_jb * ay_j$, there are

$$(2^j - \hat{b}[3, j+2])2^{n-2j-4}$$

distinct b-canonical strings which give rise to $x^*_n = x^*_{n-1}c$. Thus

$$\hat{b}[3,n] = \sum_{j=0}^{\lfloor n/2 \rfloor - 2} (2^j - \hat{b}[3, j+2])2^{n-2j-4},$$

a sum which after simplification reduces to the form given in the statement of the theorem.

(e) Observe that the b-canonical strings counted by $\tilde{b}[k,n]$ include at least the strings $x^*_{n-1}\lambda_1$ and $x^*_{n-1}\lambda_2$, where x^*_{n-1} is any b-canonical string counted by $b'[k, n-1]$.

(f) A consequence of (a) and the fact that every b-canonical string is also p-canonical. □

These results provide us with some capability to estimate the size of the entries in the b' array. It appears from Theorem 3.3(d) that exact computation of these entries is in general rather complicated. Theorem 3.3(f) shows that, for every fixed $k \geq 3$, the entries $b'[k,n]$ are asymptotically less, by a factor exponential in k, than the corresponding entries $p'[k,n]$. This result can easily be applied to yield an upper bound on $b[n]$ expressed in terms of entries in the p' array: for every positive integer n,

$$b[n] \leq \sum_{k=1}^{k^*} p'[k, n - 2^{k-1} + k], \qquad \ldots (3.1)$$

where $k^* = \lceil \log_2(n+1) \rceil$. Note that by reducing the value of k^*, we can also use (3.1) to bound the partial column sums in the b' array.

We conclude by displaying some of the smaller values in the b' array:

Non-Zero Elements $b'[k,n]$, $n \leq 10$

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
1	1	1	1	1	1	1	1	1	1	1	
2		1	3	7	15	31	63	127	255	511	
3				1	2	6	12	27	54	114	$(\hat{b}[3,n])$
					2	9	34	107	316	883	$(\tilde{b}[3,n])$
4								1	2	7	$(\hat{b}[4,n])$
									2	9	$(\tilde{b}[4,n])$
$b[n]$	1	2	4	9	20	47	110	263	630	1525	

Table 3.1

The values in this array satisfy an interesting recurrence relation that we put forward as a

Conjecture $\tilde{b}[k,n] = \sum_{j=k}^{k^*} \{b'[j, n-1] + b'[j, n-2]\}$.

4 CONCLUSION

In this paper we have shown how "distinct" strings of length n formed using exactly k letters can be efficiently computed and counted, according to two definitions of distinctness. Both of these definitions lead to algorithms that are considerably more economical than the computation or counting of $\Theta(k^n)$ strings.

REFERENCES

[**1**] Michael G. Main & Richard J. Lorentz, **An $O(n \log n)$ algorithm for finding all repetitions in a string**, *J. Algs. 5* (1984) 422-432.
[**2**] J. H. Morris & V. R. Pratt, *A Linear Pattern Matching Algorithm*, Tech. Report No. 40, Computing Center, University of California, Berkeley (1970).
[**3**] Yin Li, *A Windows-Based String Algorithm Testing System*, Undergraduate Computer Science Project, Department of Computer Science & Systems, McMaster University (1996).
[**4**] Donald E. Knuth, *The Art of Computer Programming I — Fundamental Algorithms*, Addison-Wesley (1968).
[**5**] George Pólya, Robert E. Tarjan & Donald R. Woods, *Notes on Introductory Combinatorics*, Birkhäuser (1983).
[**6**] John Riordan, *Combinatorial Identities*, John Wiley (1968).
[**7**] N. J. A. Sloane & Simon Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press (1995). See also `http://www.research.att.com/~ njas/sequences/`.
[**8**] A. P. Prudnikov, Yu. A. Brychkov & O. L. Marichev, *Integrals and Series I* (transl. N. M. Queen) Gordon & Breach (1986).

ACKNOWLEDGEMENTS

The work of the second author was supported in part by Grant No. A8180 of the Natural Sciences & Engineering Research Council of Canada and by Grant No. GO-12778 of the Medical Research Council of Canada. The authors also express their gratitude to an anonymous referee whose comments have substantially improved the quality of the paper.

Generating functions for generating trees

Cyril Banderier Mireille Bousquet-Mélou* Alain Denise
Philippe Flajolet Danièle Gardy Dominique Gouyou-Beauchamps

To appear in Discrete Math. (Submitted in November 1999. Typos fixed in Sep. 2000). This work supplants a preleminary draft "On Generating Functions of Generating Trees" dated Nov. 1998 which appeared in the Proceedings of the FPSAC'99 Barcelona Conference).

Abstract

Certain families of combinatorial objects admit recursive descriptions in terms of generating trees: each node of the tree corresponds to an object, and the branch leading to the node encodes the choices made in the construction of the object. Generating trees lead to a fast computation of enumeration sequences (sometimes, to explicit formulae as well) and provide efficient random generation algorithms. We investigate the links between the structural properties of the rewriting rules defining such trees and the rationality, algebraicity, or transcendence of the corresponding generating function.

1 Introduction

Only the simplest combinatorial structures — like binary strings, permutations, or pure involutions (*i.e.*, involutions with no fixed point) — admit product decompositions. In that case, the set Ω_n of objects of size n is isomorphic to a product set: $\Omega_n \cong [1, e_1] \times [1, e_2] \times \cdots \times [1, e_n]$. Two properties result from such a strong decomposability property: (i) enumeration is easy, since the cardinality of Ω_n is $e_1 e_2 \cdots e_n$; (ii) random generation is efficient since it reduces to a sequence of random independent draws from intervals. A simple infinite tree, called a *uniform generating tree* is determined by the e_i: the root has degree e_1, each of its e_1 descendents has degree e_2, and so on. This tree describes the sequence of all possible choices and the objects of size n are then in natural correspondence with the branches of length n, or equivalently with the nodes of generation n in the tree. The generating tree is thus fully described by its root degree (e_1) and by rewriting rules, here of the special form,

$$(e_i) \rightsquigarrow (e_{i+1})\,(e_{i+1})\,\cdots(e_{i+1}) \equiv (e_{i+1})^{e_i},$$

where the power notation is used to express repetitions. For instance binary strings, permutations, and pure involutions are determined by

$$\begin{array}{ll} \mathcal{S}: & [(2),\ (2) \rightsquigarrow (2)\,(2)] \\ \mathcal{P}: & [(1),\ \{(k) \rightsquigarrow (k+1)^k\}_{k\geq 1}] \\ \mathcal{I}: & [(1),\ \{(2k-1) \rightsquigarrow (2k+1)^{2k-1}\}_{k\geq 1}]. \end{array}$$

*Corresponding author.

A powerful generalization of this idea consists in considering unconstrained *generating trees* where any set of rules

$$\Sigma = [(s_0),\ \{(k) \rightsquigarrow (e_{1,k})\,(e_{2,k})\,\cdots\,(e_{k,k})\}] \tag{1}$$

is allowed. Here, the *axiom* (s_0) specifies the degree of the root, while the *productions* $e_{i,k}$ list the degrees of the k descendents of a node labeled k. Following Barcucci, Del Lungo, Pergola and Pinzani, we call Σ an *ECO-system* (ECO stands for "Enumerating Combinatorial Objects"). Obviously, much more leeway is available and there is hope to describe a much wider class of structures than those corresponding to product forms and uniform generating trees.

The idea of generating trees has surfaced occasionally in the literature. West introduced it in the context of enumeration of permutations with forbidden subsequences [27, 28]; this idea has been further exploited in closely related problems [6, 5, 12, 13]. A major contribution in this area is due to Barcucci, Del Lungo, Pergola, and Pinzani [4, 3] who showed that a fairly large number of classical combinatorial structures can be described by generating trees.

A form equivalent to generating trees is well worth noting at this stage. Consider the *walks* on the integer half-line that start at point (s_0) and such that the only allowable transitions are those specified by Σ (the steps corresponding to transitions with multiplicities being labeled). Then, the walks of length n are in bijective correspondence with the nodes of generation n in the tree. These walks are constrained by the consistency requirement of trees, namely, that the number of outgoing edges from point k on the half-line has to be exactly k.

EXAMPLE 1. *123-avoiding permutations*
The method of "local expansion" sometimes gives good results in the enumeration of permutations avoiding specified patterns. Consider for example the set $\mathfrak{S}_n(123)$ of permutations of length n that *avoid the pattern* 123: there exist no integers $i < j < k$ such that $\sigma(i) < \sigma(j) < \sigma(k)$. For instance, $\sigma = 4213$ belongs to $\mathfrak{S}_4(123)$ but $\sigma = 1324$ does not, as $\sigma(1) < \sigma(3) < \sigma(4)$.

Observe that if $\tau \in \mathfrak{S}_{n+1}(123)$, then the permutation σ obtained by erasing the entry $n+1$ from τ belongs to $\mathfrak{S}_n(123)$. Conversely, for every $\sigma \in \mathfrak{S}_n(123)$, insert the value $n+1$ in each place that gives an element of $\mathfrak{S}_{n+1}(123)$ (this is the local expansion). For example, the permutation $\sigma = 213$ gives 4213, 2413 and 2143, by insertion of 4 in first, second and third place respectively. The permutation 2134, resulting from the insertion of 4 in the last place, does not belong to $\mathfrak{S}_4(123)$. This process can be described by a tree whose nodes are the permutations avoiding 123: the root is 1, and the children of any node σ are the permutations derived as above. Figure 1(a) presents the first four levels of this tree.

Let us now label the nodes by their number of children: we obtain the tree of Figure 1(b). It can be proved that the k children of any node labeled k are labeled respectively $k+1, 2, 3, \ldots, k$ (see [27]). Thus the tree we have constructed is the generating tree obtained from the following rewriting rules:

$$[(2),\ \{(k) \rightsquigarrow (2)(3)\ldots(k-1)(k)(k+1)\}_{k\geq 2}].$$

The interpretation of this system in terms of paths implies that 123-avoiding permutations are equinumerous with "walks with returns" on the half-line, themselves isomorphic to Łukasiewicz codes of plane trees (see, *e.g.*, [26, p. 31–35]). We thus recover a classic result [18]: 123-avoiding permutations are counted by Catalan numbers; more precisely, $|\mathfrak{S}_n(123)| = \binom{2n}{n}/(n+1)$. □

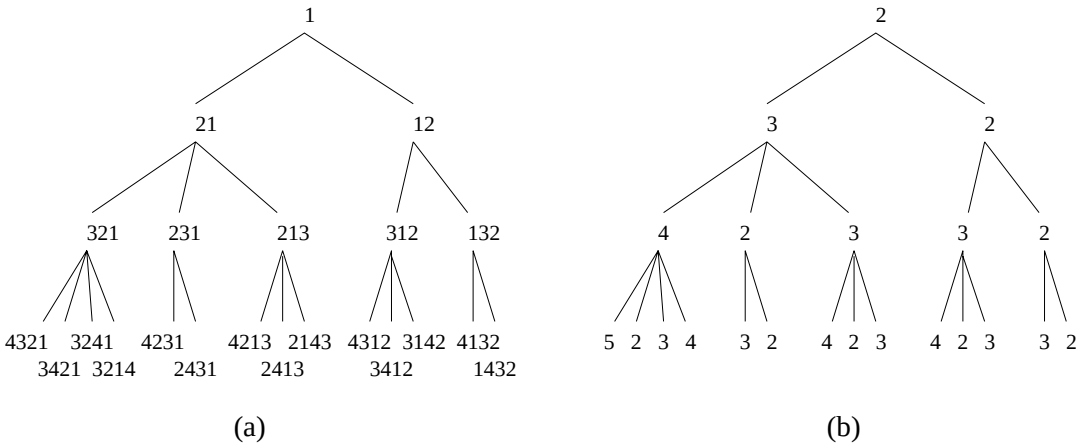


Figure 1: The generating tree of 123-avoiding permutations. (a) Nodes labeled by the permutations. (b) Nodes labeled by the numbers of children.

We shall see below that (certain) generating trees correspond to enumeration sequences of relatively low computational complexity and provide fast random generation algorithms. Hence, there is an obvious interest in delineating as precisely as possible which combinatorial classes admit a generating tree specification. Generating functions condense structural information in a simple analytic entity. We can thus wonder what kind of generating function can be obtained through generating trees. More precisely, we study in this paper the connections between the *structural* properties of the rewriting rules and the *algebraic* properties of the corresponding generating function.

We shall prove several conjectures that were presented to us by Pinzani and his coauthors in March 1998. Our main results can be roughly described as follows.

— *Rational systems.* Systems satisfying strong regularity conditions lead to rational generating functions (Section 2). This covers systems that have a finite number of allowed degrees, as well as systems like $(2.a)$, $(2.b)$, $(2.c)$ and $(2.d)$ below where the labels are constant except for a fixed number of labels that depend linearly and uniformly on k.

— *Algebraic systems.* Systems of a *factorial* form, *i.e.*, where a finite modification of the set $\{1, \ldots, k\}$ is reachable from k, lead to algebraic generating functions (Section 3). This includes in particular cases $(2.f)$ and $(2.g)$.

— *Transcendental systems.* One possible reason for a system to give a transcendental series is the fact that its coefficients grow too fast, so that its radius of convergence is zero. This is the case for System $(2.h)$ below. Transcendental generating functions are also associated with systems that are too "irregular". An example is System $(2.e)$. We shall also discuss the holonomy of transcendental systems (Section 4).

EXAMPLE 2. *A zoo of rewriting systems*
Here is a list of examples recurring throughout this paper.

$$[(3), \{(k) \rightsquigarrow (3)^{k-3}(k+1)(k+2)(k+9)\}] \tag{2.a}$$
$$[(3), \{(k) \rightsquigarrow (3)^{k-1}(3k+6)\}] \tag{2.b}$$
$$[(2), \{(k) \rightsquigarrow (2)^{k-2}(2+(k \bmod 2))(k+1)\}] \tag{2.c}$$
$$[(2), \{(k) \rightsquigarrow (2)^{k-2}(3-(k \bmod 2))(k+1)\}] \tag{2.d}$$
$$[(2), \{(k) \rightsquigarrow (2)^{k-2}(3-[\exists p : k = 2^p])(k+1)\}] \tag{2.e}$$
$$[(2), \{(k) \rightsquigarrow (2)(3)\dots(k-1)(k)(k+1)\}] \tag{2.f}$$
$$[(1), \{(k) \rightsquigarrow (1)(2)\dots(k-1)(k+1)\}] \tag{2.g}$$
$$[(2), \{(k) \rightsquigarrow (2)(3)(k+2)^{k-2}\}] \tag{2.h}$$

(In (2.*e*), we make use of Iverson's brackets: $[P]$ equals 1 if P is true, 0 otherwise.) □

Notations. From now on, we adopt functional notations for rewriting rules: systems will be of the form

$$[(s_0), \quad \{(k) \rightsquigarrow (e_1(k))\,(e_2(k))\dots(e_k(k))\}]$$

where s_0 is a constant and each e_i is a function of k. Moreover, we assume that all the values appearing in the generating tree are positive: each node has at least one descendent.

In the generating tree, let f_n be the number of nodes at level n and s_n the sum of the labels of these nodes. By convention, the root is at level 0, so that $f_0 = 1$. In terms of walks, f_n is the number of walks of length n. The generating function associated with the system is

$$F(z) = \sum_{n \geq 0} f_n z^n.$$

Remark that $s_n = f_{n+1}$, and that the sequence $(f_n)_n$ is nondecreasing.

Now let $f_{n,k}$ be the number of nodes at level n having label k (or the number of walks of length n ending at position k). The following generating functions will be also of interest:

$$F(z,u) = \sum_{n,k \geq 0} f_{n,k} z^n u^k \qquad \text{and} \quad F_k(z) = \sum_{n \geq 0} f_{n,k} z^n.$$

We have $F(z) = F(z,1) = \sum_{k \geq 1} F_k(z)$. Furthermore, the F_k's satisfy the relation

$$F_k(z) = [k = s_0] + z \sum_{j \geq 1} \pi_{j,k} F_j(z), \tag{2}$$

where $\pi_{j,k} = |\{i \leq j : e_i(j) = k\}|$ denotes the number of one-step transitions from j to k. This is equivalent to the following recurrence for the numbers $f_{n,k}$,

$$f_{0,k} = [k = s_0] \qquad \text{and} \quad f_{n+1,k} = \sum_{j \geq 1} \pi_{j,k} f_{n,j}, \tag{3}$$

that results from tracing all the paths that lead to k in $n+1$ steps.

Counting and random generation. The recurrence (3) gives rise to an algorithm that computes the successive rows of the matrix $(f_{n,k})$ by "forward propagation": to compute the $(n+1)$th row, propagate the contribution $f_{n,j}$ to $f_{n+1,e_i(j)}$ for all pairs (i,j) such that $i \leq j$. Assume the system is *linearly bounded*: this means that the labels of the nodes that can be reached in m steps are bounded by a linear function of m. (All the systems given in Example 2, except for (2.*b*), are linearly bounded; more generally, systems where forward jumps are bounded by a constant are linearly bounded.) Clearly, the forward propagation algorithm provides a counting algorithm of arithmetic complexity that is at most cubic.

For a linearly bounded system, uniform random generation can also be achieved in polynomial time, as shown in [2]. We present here the general principle.

Let $g_{n,k}$ be the number of walks of length n that start from label k. These numbers are determined by the recurrence $g_{n,k} = \sum_i g_{n-1,e_i(k)}$, that traces all the possible continuations of a path given its initial step. Obviously, $f_n = g_{n,s_0}$, with s_0 the axiom of the system. As above, the $g_{n,k}$ can be determined in time $O(n^3)$ and $O(n^2)$ storage. Random generation is then achieved as follows: In order to generate a walk of length n starting from state k, pick up a transition i with probability $g_{n-1,e_i(k)}/g_{n,k}$, and generate recursively a walk of length $n-1$ starting from state $e_i(k)$. The cost of a single random generation is then $O(n^2)$ if a sequential search is used over the $O(n)$ possibilities of each of the n random drawings; the time complexity goes down to $O(n \log n)$ if binary search is used, but at the expense of an increase in storage complexity of $O(n^3)$ (arising from $O(n^2)$ arrays of size $O(n)$ that binary search requires).

2 Rational systems

We give in this section three main criteria (and a variation on one of them) implying that the generating function of a given ECO-system is rational.

Our first and simplest criterion applies to systems in which the functions e_i are uniformly bounded.

Proposition 1 *If finitely many labels appear in the tree, then $F(z)$ is rational.*

Proof. Only a finite number of F_k's are nonzero, and they are related by linear equations like Equation (2) above. ∎

EXAMPLE 3. *The Fibonacci numbers*
The system $[(1),\ \{(k) \rightsquigarrow (k)^{k-1}((k \bmod 2)+1)\}]$ can be also written as $[(1),\ \{(1) \rightsquigarrow (2),\ (2) \rightsquigarrow (1)(2)\}]$. Hence the only labels that occur in the tree are 1 and 2. Eq. (2) gives $F_1(z) = 1 + zF_2(z)$ and $F_2(z) = z(F_1(z) + F_2(z))$. Finally,

$$F(z) = \frac{1}{1-z-z^2} = \sum_{n \geq 0} f_n z^n = 1 + z + 2z^2 + 3z^3 + 5z^4 + \cdots,$$

the well-known Fibonacci generating function. □

None of the systems of Example 2 satisfy the assumptions of Proposition 1. However, the following criterion can be applied to systems (2.*a*) and (2.*b*).

Proposition 2 *Let $\sigma(k) = e_1(k) + e_2(k) + \cdots + e_k(k)$. If σ is an affine function of k, say $\sigma(k) = \alpha k + \beta$, then the series $F(z)$ is rational. More precisely:*

$$F(z) \;=\; \frac{1 + (s_0 - \alpha)z}{1 - \alpha z - \beta z^2}.$$

Proof. Let $n \geq 0$ and let $k_1, k_2, \ldots k_{f_n}$ denote the labels of the f_n nodes at level n. Then

$$\begin{aligned} f_{n+2} \;=\; s_{n+1} \;&=\; (\alpha k_1 + \beta) + (\alpha k_2 + \beta) + \cdots + (\alpha k_{f_n} + \beta) \\ &=\; \alpha s_n + \beta f_n \;=\; \alpha f_{n+1} + \beta f_n. \end{aligned}$$

We know that $f_0 = 1$ and $f_1 = s_0$. The result follows. ∎

EXAMPLE 4. *Bisection of Fibonacci sequence*
The system $[(2),\ \{(k) \rightsquigarrow (2)^{k-1}(k+1)\}]$ gives $F(z) = \frac{1-z}{1-3z+z^2} = 1 + 2z + 5z^2 + \cdots$, the generating function for Fibonacci numbers of even index. (Changing the axiom to $(s_0) = (3)$ leads to the other half of the Fibonacci sequence.) Some other systems, like

$$\begin{aligned} &[(2),\ \{(k) \rightsquigarrow (1)^{k-1}(2k)\}], \\ &[(2),\ \{(k) \rightsquigarrow (2)^{k-2}(3 - (k \bmod 2))(k + (k \bmod 2))\}], \\ &[(2),\ \{(k) \rightsquigarrow (2)^{k-2}(3 - [k \text{ is prime}])(k + [k \text{ is prime}])\}], \end{aligned}$$

lead to the same function $F(z)$ since $\sigma(k) = 3k - 1$ and $s_0 = 2$. However, the generating trees are different, as are the bivariate functions $F(z, u)$. □

EXAMPLE 5. *Prime numbers and rational generating functions*
Amazingly, it is possible to construct a generating tree whose set of labels is the set of prime numbers but that has a rational generating function $F(z)$. This is a bit unexpected, as prime numbers are usually thought "too irregular" to be associated with rational generating functions. For $n \geq 1$, let p_n denote the nth prime; hence $(p_1, p_2, p_3, \ldots) = (2, 3, 5, \ldots)$. Assume for the moment that the Goldbach conjecture is true: every even number larger than 3 is the sum of two primes. Remember that, according to Bertrand's postulate, $p_{n+1} < 2p_n$ for all n (see, *e.g.*, [23, p. 140]).

For $n \geq 1$, the number $2p_n - p_{n+1} + 3$ is an even number larger than 3. Let q_n and r_n be two primes such that $2p_n - p_{n+1} + 3 = q_n + r_n$. In particular, $q_1 = r_1 = 2$. Consider the system

$$[(2), \{(p_n) \rightsquigarrow (p_{n+1})(q_n)(r_n)(2)^{p_n - 3}\}].$$

It satisfies the criterion of Proposition 2, with $\sigma(k) = 4k - 3$. Hence, the generating function of the associated generating tree is

$$F(z) = \frac{1 - 2z}{1 - 4z + 3z^2} = \frac{1}{2}\left[\frac{1}{1 - z} + \frac{1}{1 - 3z}\right].$$

Consequently, the number of nodes at level n is simply $f_n = (1 + 3^n)/2$. This can be checked on the first few levels of the tree drawn in Figure 2.

Now, one can object that the Goldbach conjecture is not proved; however, it is known that every even number is the sum of at most six primes [22], and a similar example can be constructed using this result.

□

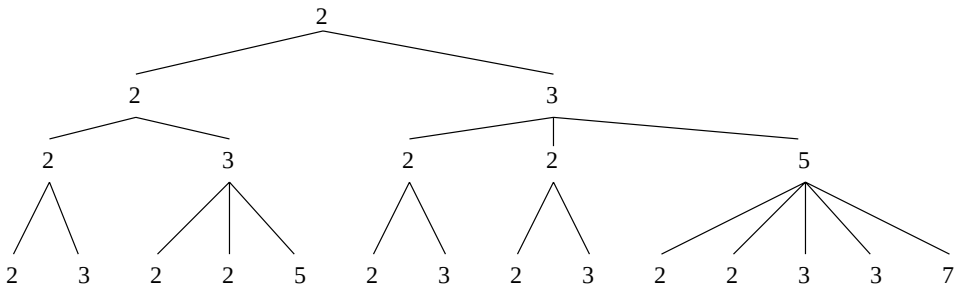


Figure 2: A generating tree with prime labels and rational generating function.

Proposition 2 can be adapted to apply to systems that "almost" satisfy the criterion of Proposition 2, like System (2.c) or (2.d). Let us consider a system of the form

$$(s_0), \quad (k) \rightsquigarrow e_1^{[0]}(k), \ldots, e_k^{[0]}(k) \quad \text{if } k \text{ is even},$$
$$(k) \rightsquigarrow e_1^{[1]}(k), \ldots, e_k^{[1]}(k) \quad \text{if } k \text{ is odd}.$$

Assume, moreover, that:

(i) the corresponding functions σ_0 and σ_1 are affine and have the same leading coefficient α, say $\sigma_0(k) = \alpha k + \beta_0$ and $\sigma_1(k) = \alpha k + \beta_1$;

(ii) exactly m odd labels occur in the right-hand side of each rule, for some $m \geq 0$.

Proposition 3 *If a system satisfies properties* (i) *and* (ii) *above, then*

$$F(z) = \frac{1 + (s_0 - \alpha)z + (s_1 - \alpha s_0 - \beta_0)z^2}{1 - \alpha z - \beta_0 z^2 - m(\beta_1 - \beta_0)z^3}.$$

Of course, if $\beta_0 = \beta_1$, we recover the generating function of Proposition 2.

Proof. The proof is similar to that of Proposition 2. The only new ingredient is the fact that, for $n \geq 1$, the number of nodes of odd label at level n is mf_{n-1}. ∎

System (2.c) satisfies properties (i) and (ii) above with $\alpha = 3$, $\beta_0 = -1$, $\beta_1 = 0$, $m = 1$, $s_0 = 2$ and $s_1 = 5$. Consequently, its generating function is $F(z) = \frac{1-z}{1-3z+z^2-z^3}$. System (2.$d$), although very close to (2.c), does not satisfy property (ii) above, so that Proposition 3 does not apply. However, another minor variation on the argument of Proposition 2, based on the fact that the number o_n of odd labels at level n satisfies $o_n = 2(f_{n-1} - o_{n-1})$, proves the rationality of $F(z)$.

Alternatively, rationality follows from the last criterion of this section, which is of a different nature. We consider systems $[(s_0), \{(k) \rightsquigarrow (e_1(k))(e_2(k)) \ldots (e_k(k))\}]$ that can be written as

$$[(s_0), \{(k) \rightsquigarrow (c_1(k))(c_2(k)) \ldots (c_{k-m}(k))(k + a_1)(k + a_2) \ldots (k + a_m)\}] \tag{4}$$

where $1 \leq a_1 \leq a_2 \leq \cdots \leq a_m$ and the functions c_i are uniformly bounded. Let $C = \max_{i,k}\{s_0, c_i(k)\}$.

Proposition 4 *Consider the system* (4), *and let $\pi_{j,k} = |\{i \leq j : e_i(j) = k\}|$. If all the series*

$$\sum_{j \geq 1} \pi_{j,k}\ t^j$$

for $k \leq C$ are rational, then so is the series $F(z)$.

Proof. We form an infinite system of equations defining the series $F_k(z)$ by writing Eq. (2) for all $k \geq 1$. In particular, for $k > C$, we obtain

$$F_k(z) = z \sum_{\ell=1}^{m} F_{k-a_\ell}(z),$$

with $F_j(z) = 0$ if $j \leq 0$. This part of the system is easy to solve in terms of $F_1, \ldots, F_C$. Indeed, for $k \in \mathbb{Z}$:

$$F_k(z) = \sum_{i=1}^{C} P_{i,k}(z) F_i(z) \tag{5}$$

where the $P_{i,k}$ are polynomials in z defined by the following recurrence: for all $i \leq C$,

$$P_{i,k}(z) = \begin{cases} 0 & \text{if } k \leq 0, \\ [k = i] & \text{if } 0 < k \leq C, \\ z \sum_{\ell=1}^{m} P_{i,k-a_\ell}(z) & \text{if } k > C. \end{cases} \tag{6}$$

Using (5), we find

$$F(z) = \sum_{k \geq 1} F_k(z) = \sum_{i=1}^{C} \left[F_i(z) \sum_{k \geq 1} P_{i,k}(z) \right].$$

According to (6), for all $i \leq C$, the series $\sum_{k \geq 1} P_{i,k}(z) t^k$ is a rational function of z and t, of denominator $1 - z \sum_\ell t^{a_\ell}$. At $t = 1$, it is rational in z. Hence, to prove the rationality of $F(z)$, it suffices to prove the rationality of the $F_i(z)$, for $i \leq C$.

Let us go back to the C first equations of our system; using (5), we find, for $k \leq C$:

$$F_k(z) = [k = s_0] + z \sum_{i=1}^{C} \left[F_i(z) \sum_{j \geq 1} P_{i,j}(z) \pi_{j,k} \right].$$

Again, $\sum_{j \geq 1} P_{i,j}(z) \pi_{j,k} t^j$ is a rational function of z and t (the Hadamard product of two rational series is rational). Thus the series $F_k(z)$, for $k \leq C$, satisfy a linear system with rational coefficients: they are rational themselves, as well as $F(z)$. ∎

Examples (2.*a*), (2.*c*), (2.*d*) and (2.*e*) have the form (4). The above proposition implies that the first three have a rational generating function. System (2.*e*) will be discussed in Section 4, and proved to have a transcendental generating function.

3 Factorial walks and algebraic systems

In this section, we consider systems that are of a *factorial form*. By this, we mean informally that the set of successors of (k) is a finite modification of the integer interval $\{1, 2, \ldots, k\}$. As was detailed in the introduction, ECO-systems can be rephrased in terms of walks over the integer half-line. We thus consider the problem of enumerating walks over the integer half-line such that the set of allowed moves from point k is a finite modification of the integer interval $[0, k]$. We shall mostly study modifications around the point k (although some examples where the interval is modified around 0 as well are given at the end of the

section). Precisely, a *factorial walk* is defined by a finite (multi)set $A \subset \mathbb{Z}$ and a finite set $B \subset \mathbb{N}^+$, where $\mathbb{N}^+ = \{1, 2, 3, \dots\}$, specifying respectively the *allowed supplementary jumps* (possibly labeled) and the *forbidden backward jumps*. In other words, the possible moves from k are given by the rule:

$$(k) \rightsquigarrow [0, k-1] \setminus (k - B) \ \cup \ (k + A). \tag{7}$$

Observe that these walk models are not necessarily ECO-systems, first because we allow labels to be zero – but a simple translation can take us back to a model with positive labels – and second because we do not require (k) to have exactly k successors.

We say that an ECO-system is factorial if a shift of indices transforms it into a factorial walk. Hence the rules of a factorial ECO-system are of the form

$$(k + r) \rightsquigarrow [r, k + r - 1] \setminus (k + r - B) \ \cup \ (k + r + A),$$

that is,

$$(k) \rightsquigarrow [r, k-1] \setminus (k - B) \ \cup \ (k + A) \quad \text{for } k \geq r \geq 1. \tag{8}$$

The generating function $F(z)$ for such an ECO-system, taken with axiom (s_0), equals the generating function for the walk model (7), taken with axiom $(s_0 - r)$. However, remember that the rewriting rules defining a generating tree have to obey the additional condition that a node labeled k has exactly k successors. Taking $k = r$ in (8), this implies that $r = |A|$. Taking $k > r + \max B$, this implies that $r + |B| = |A|$, so that finally $B = \emptyset$. Hence, strictly speaking, either one has a "fake" factorial ECO-system (that is some of its initial rules are not of the factorial type), either one has a "real" factorial ECO-system and then it is given by rules of the form

$$(k) \rightsquigarrow [r, k-1] \ \cup \ (k + A) \quad \text{for } k \geq r \geq 1,$$

where A is a multiset of integers of cardinality r. For instance, Systems $(2.f)$ and $(2.g)$ are factorial. We shall prove that all factorial walks have an algebraic generating function. The result naturally applies to factorial ECO-systems.

We consider again the generating function $F(z, u) = \sum_{n,k \geq 0} f_{n,k} z^n u^k$, where $f_{n,k}$ is the number of walks of length n ending at point k. We also denote by $F_k(z)$ the coefficient of u^k in this series, and by $f_n(u)$ the coefficient of z^n. The first ingredient of the proof is a linear operator M, acting on formal power series in u, that encodes the possible moves. More precisely, for all $n \geq 0$, we will have:

$$M[f_n](u) = f_{n+1}(u).$$

The operator M is constructed step by step as follows.

— The set of moves from k to all the positions $0, 1, \dots, k-1$ is described by the operator L_0 that maps u^k to $u^0 + u^1 + \cdots + u^{k-1} = (1 - u^k)/(1 - u)$. As L_0 is a linear operator, we have, for any series $g(u)$:

$$L_0[g](u) = \frac{g(1) - g(u)}{1 - u}.$$

— The fact that transitions near k are modified, with those of type $k+\alpha$ (with $\alpha \in A$) allowed and those of type $k-\beta$ (with $\beta \in B$) forbidden, is expressed by a Laurent polynomial

$$P(u) = \sum_{k=-b}^{a} p_k u^k = A(u) - B(u) \quad \text{with} \quad A(u) = \sum_{\alpha \in A} u^\alpha \quad \text{and} \quad B(u) = \sum_{\beta \in B} u^{-\beta}.$$

The degree of P is $a := \max A$, the largest forward jump and $b := \max(0, -B, -A)$ is largest forbidden backward jump or the largest supplementary backward jumps (we take $b = 0$ if the set B is empty).

The operator

$$L[g](u) := L_0[g](u) + P(u)g(u)$$

describes the extension of a walk by one step.

— Finally, the operator M is given by

$$M[g](u) = L[g](u) - \{u^{<0}\}L[g](u),$$

where $\{u^{<0}\}h(u)$ is the sum of all the monomials in $h(u)$ having a negative exponent. Hence M is nothing but L stripped of the negative exponent monomials, which correspond to walks ending on the nonpositive half-line. Observe that, for any series $g(u)$, the only part of $L[g](u)$ that is likely to contain monomials with negative exponents is $P(u)g(u)$. Consequently,

$$M[g](u) = L[g](u) - \{u^{<0}\}[P(u)g(u)]$$

and if $g(u) = \sum_k g_k u^k$, then

$$\{u^{<0}\}[P(u)g(u)] = \sum_{i=1}^{b} \sum_{k=0}^{i-1} g_k p_{-i}\, u^{k-i} = \sum_{k=0}^{b-1} g_k r_k(u). \tag{9}$$

Assume for simplicity that the initial point of the walk is 0; other cases follow the same argument. The linear relation $f_{n+1}(u) = M[f_n](u)$, together with $f_0(u) = 1$, yields

$$\begin{aligned} F(z,u) &= 1 + zM[F](z,u) \\ &= 1 + z\left(\frac{F(z,1) - F(z,u)}{1-u} + P(u)F(z,u) + \{u^{<0}\}[P(u)F(z,u)]\right). \end{aligned}$$

Thanks to (9), we can write

$$\{u^{<0}\}[P(u)F(z,u)] = \sum_{k=0}^{b-1} r_k(u) F_k(z),$$

where $r_k(u)$ is a Laurent polynomials (defined by Equation 9) whose exponents belong to $[k-b, -1]$. Thus, $F(z,u)$ satisfies the following functional equation:

$$F(z,u)\left(1 + \frac{z}{1-u} - zP(u)\right) = 1 + \frac{zF(z,1)}{1-u} + z\sum_{k=0}^{b-1} r_k(u) F_k(z). \tag{10}$$

Let us take an example. The moves

$$(k) \rightsquigarrow (0)(1)\cdots(k-5)(k-3)(k-1)(k)(k+7)(k+9),$$

lead to $A(u) = u^0 + u^7 + u^9$ and $B(u) = u^{-4} + u^{-2}$. Moreover,

$$\{u^{<0}\}[B(u)F(z,u)] = (u^{-2}+u^{-4})F_0(z) + (u^{-1}+u^{-3})F_1(z) + u^{-2}F_2(z) + u^{-1}F_3(z),$$

so that the functional equation defining $F(z,u)$ is

$$F(z,u)\left(1 + \frac{z}{1-u} - z(1+u^7+u^9-u^{-4}-u^{-2})\right) =$$

$$1 + \frac{zF(z,1)}{1-u} + z(u^{-2}+u^{-4})F_0(z) + z(u^{-1}+u^{-3})F_1(z) + zu^{-2}F_2(z) + zu^{-1}F_3(z).$$

The second ingredient of the proof, sometimes called the *kernel method*, seems to belong to the "mathematical folklore" since the 1970's. It has been used in various combinatorial problems [10, 18, 20] and in probabilities [14]. See also [8, 9, 21] for more recent and systematic applications. This method consists in cancelling the left-hand side of the fundamental functional equation (10) by coupling z and u, so that the coefficient of the (unknown) quantity $F(z,u)$ is zero. This constraint defines u as one of the branches of an algebraic function of z. Each branch that can be substituted analytically into the functional equation yields a linear relation between the unknown series $F(z,1)$ and $F_k(z)$, $0 \le k < b$. If enough branches can be substituted analytically, we obtain a system of linear equations, whose solution gives $F(z,1)$ and the $F_k(z)$ as algebraic functions. From there, an expression for $F(z,u)$ also results in the form of a bivariate algebraic function.

Let us multiply Eq. (10) by $u^b(1-u)$ to obtain an equation with polynomial coefficients (remind that we take $b = 0$ if the set B of forbidden backward steps is empty). The new equation reads $K(z,u)F(z,u) = R(z,u)$, where $K(z,u)$ is the *kernel* of the equation:

$$\begin{aligned} K(z,u) &= u^b(1-u)\left(1 + \frac{z}{1-u} - zP(u)\right), \\ &= u^b(1-u) + zu^b - z(1-u)\sum_{\alpha\in A} u^{\alpha+b} + z(1-u)\sum_{\beta\in B} u^{b-\beta}. \end{aligned} \tag{11}$$

This polynomial has degree $a+b+1$ in u, and hence, admits $a+b+1$ solutions, which are algebraic functions of z. The classical theory of algebraic functions and the Newton polygon construction enable us to expand the solutions near any point as Puiseux series (that is, series involving fractional exponents; see [11]). The $a+b+1$ solutions, expanded around 0, can be classified as follows:

— the "unit" branch, denoted by u_0, is a power series in z with constant term 1;

— b "small" branches, denoted by $u_1, \ldots, u_b$, are power series in $z^{1/b}$ whose first nonzero term is $\zeta z^{1/b}$, with $\zeta^b + 1 = 0$;

— a "large" branches, denoted by $v_1, \ldots, v_a$, are Laurent series in $z^{1/a}$ whose first nonzero term is $\zeta z^{-1/a}$, with $\zeta^a + 1 = 0$.

In particular, all the roots are distinct. (It is not difficult to check "by hand" the existence of these solutions: for instance, plugging $z = t^b$ and $u = tw(t)$ in $K(z,u) = 0$ confirms the existence of the b small branches.) Note that there are exactly $b+1$ finite branches: the unit branch u_0 and the b small branches $u_1, \ldots, u_b$. As $F(z,u)$ is a series in z with *polynomial* coefficients in u, these $b+1$ series u_i, having no negative exponents, can be substituted for u in $F(z,u)$. More specifically, let us replace u by u_i in (10): the right-hand side of the equation vanishes, giving a linear equation relating the $b+1$ unknown series $F(z,1)$ and $F_k(z)$, $0 \leq k < b$. Hence the $b+1$ finite branches give a set of $b+1$ linear equations relating the $b+1$ unknown series. One could solve directly this system, but the following argument is more elegant.

The right-hand side of (10), once multiplied by $u^b(1-u)$, is

$$R(z,u) = u^b(1-u)\left(1 + \frac{z}{1-u}F(z,1) + z\sum_{k=0}^{b-1} r_k(u)F_k(z)\right).$$

By construction, it is a *polynomial* in u of degree $b+1$ and leading coefficient -1. Hence, it admits $b+1$ roots, which depend on z. Replacing u by the series $u_0, u_1, \ldots, u_b$ in Eq. (10) shows that these series are exactly the $b+1$ roots of R, so that

$$R(z,u) = -\prod_{i=0}^{b}(u - u_i).$$

Let $p_a := [u^a]P(u)$ be the multiplicity of the largest forward jump. Then the coefficient of u^{a+b+1} in $K(z,u)$ is $p_a z$, and we can write

$$K(z,u) = p_a z \prod_{i=0}^{b}(u - u_i)\prod_{i=1}^{a}(u - v_i).$$

Finally, as $K(z,u)F(z,u) = R(z,u)$, we obtain

$$F(z,u) = \frac{-\prod_{i=0}^{b}(u-u_i)}{u^b(1-u) + zu^b - zu^b(1-u)P(u)} = -\frac{1}{p_a z\prod_{i=1}^{a}(u-v_i)}. \tag{12}$$

We have thus proved the following result.

Proposition 5 *The generating function $F(z,u)$ for factorial walks defined by* (7) *and starting from* 0 *is algebraic; it is given by* (12)*, where $u_0, \ldots, u_b$ (resp. $v_1, \ldots, v_a$) are the finite (resp. infinite) solutions at $z = 0$ of the equation $K(z,u) = 0$ and the kernel K is defined by* (11)*. In particular, the generating function for all walks, irrespective of their endpoint, is*

$$F(z,1) = -\frac{1}{z}\prod_{i=0}^{b}(1-u_i),$$

and the generating function for excursions*, i.e., walks ending at* 0*, is, for $b < 0$:*

$$F(z,0) = \frac{(-1)^b}{z}\prod_{i=0}^{b} u_i,$$

(for $b = 0$, the relation becomes $F(z,0) = \frac{(-1)^b}{1+z-p_0 z}\prod_{i=0}^{b} u_i$.)

These results could be derived by a detour via multivariate linear recurrences, and the present treatment is closely related to [9, 21]; however, our results were obtained independently in March 1998 [1].

The asymptotic behaviour of the number of n-step walks can be established via singularity analysis or saddle point methods. The series u_i have "in general" a square root singularity, yielding an asymptotic behaviour of the form $A\mu^n n^{-3/2}$. We plan to develop this study in a forthcoming paper.

EXAMPLE 6. *Catalan numbers*
This is the simplest factorial walk, $(k) \rightsquigarrow (0)(1)\dots(k)(k+1)$, which corresponds to the ECO-system $(2.f)$. The operator M is given by

$$M[f](u) = \frac{f(1) - f(u)}{1-u} + (1+u)f(u).$$

The kernel is $K(z,u) = 1 - u + z - z(1-u)(1+u) = 1 - u + zu^2$, hence $u_0(z) = \frac{1-\sqrt{1-4z}}{2z}$, so that

$$F(z,1) = -\frac{1-u_0}{z} = \frac{1-2z-\sqrt{1-4z}}{2z^2} = \sum_{n\geq 1} \binom{2n}{n} \frac{z^{n-1}}{n+1},$$

the generating function of the Catalan numbers (sequence **M1459**[1]). This result could be expected, given the obvious relation between these walks and Łukasiewicz codes. □

EXAMPLE 7. *Motzkin numbers*
This example, due to Pinzani and his co-authors, is derived from the previous one by forbidding "forward" jumps of length zero. The rule is then

$$(k) \rightsquigarrow (0)\cdots(k-1)(k+1).$$

The operator M is

$$M[f](u) = \frac{f(1) - f(u)}{1-u} + uf(u).$$

The kernel is $K(z,u) = 1 - u + z - zu(1-u) = 1 + z - u(1+z) + zu^2$, leading to

$$F(z,1) \quad = \quad \frac{1-z-\sqrt{1-2z-3z^2}}{2z^2} = 1 + z + 2z^2 + 4z^3 + 9z^4 + 21z^5 + O(z^6),$$

the generating function for Motzkin numbers (sequence **M1184**). □

EXAMPLE 8. *Schröder numbers*
This example is also due to the Florentine group. The rule is $(k) \rightsquigarrow (0)\dots(k-1)(k)(k+1)^2$. From Proposition 5, we derive

$$F(z,1) = \frac{1-3z-\sqrt{1-6z+z^2}}{4z^2} = 1 + 3z + 11z^2 + 45z^3 + 197z^4 + O(z^5).$$

The coefficients are the Schröder numbers (**M2898**: Schröder's second problem). We give in Table 1 at the end of the paper a generalization of Catalan and Schröder numbers, corresponding to the rule $(k) \rightsquigarrow (0)\dots(k-1)(k)(k+1)^m$. This generalized rule has recently been shown to describe a set of permutations avoiding certain patterns [19]. □

[1]The numbers **Mxxxx** are identifiers of the sequences in *The Encyclopedia of Integer Sequences* [24].

The above examples were all quadratic. However, it is clear from our treatment that algebraic functions of arbitrary degree can be obtained: it suffices that the set of "exceptions" around k have a span greater than 1. Let us start with a family of ECO-systems where supplementary forward jumps of length larger than one are allowed.

EXAMPLE 9. *Ternary trees, dissections of a polygon, and m-ary trees*
The ECO-system with axiom $(s_0) = (3)$ and rule

$$(k) \rightsquigarrow (3)(4)\cdots(k)(k+1)(k+2)$$

is equivalent to the walk

$$(k) \rightsquigarrow (0)(1)\cdots(k)(k+1)(k+2).$$

The kernel is $K(z,u) = 1 - u + zu^3$, and the generating function

$$F(z,1) = \sum_{n\geq 1} \binom{3n}{n} \frac{z^{n-1}}{2n+1}$$

counts ternary trees (**M2926**).

More generally, the system with axiom (m) and rewriting rules

$$(k) \rightsquigarrow (m)\cdots(k)(k+1)(k+2)\cdots(k+m-1)$$

yields the m-Catalan numbers, $\binom{mn}{n}/((m-1)n+1)$, that count m-ary trees. The kernel is $1 - u + zu^m$ and the generating function $F(z,1)$ satisfies $F(z,1) = (1 + zF(z,1))^m$. In particular, the 4-Catalan numbers $\binom{4n}{n}/(3n+1)$ appear in [24] (sequence **M3587**) and count dissections of a polygon.

□

In the above examples, all backward jumps are allowed. In other words, each of these examples corresponds to an ECO-system. Let us now give an example where backward jumps of length 1 are forbidden.

EXAMPLE 10.
Consider the following modification of the Motzkin rule:

$$(k) \rightsquigarrow (0)\cdots(k-2)(k+1).$$

The kernel is now $K(z,u) = u(1-u) + zu - z(1-u)(u^2-1)$, and, according to (12), the series $F(z) = F(z,1)$ is given by $F(z) = 1/[z(v_1 - 1)]$, where v_1 satisfies $K(z,v_1) = 0$ and is infinite at $z = 0$. Denoting $G = zF(z)$, we find that the algebraic equation defining G is:

$$G = z\,\frac{1 + 2G + G^2 + G^3}{1+G}.$$

□

So far, we have only dealt with walks for which the set of allowed moves was obtained by modifying the interval $[0,k]$ around k. One can also modify this interval around 0: we shall see – in examples – that the generating function remains algebraic. However, it is interesting to note that in these examples, the kernel method does not immediately provide enough equations between the "unknown functions" to solve the functional equation.

Let us first explain how we modify the interval $[0,k]$ around 0. The walks we wish to count are still specified by a multiset A of allowed supplementary jumps and a set B of forbidden backward jumps. But, in addition, we forbid backward jumps to end up in C, where C is a given finite subset of $\mathbb{N}$. In other words, the possible moves from k are given by the rule

$$(k) \rightsquigarrow [0,k-1] \setminus (C \cup (k-B)) \ \cup \ (k+A).$$

Again, we can write a functional equation defining $F(z,u)$:

$$F(z,u) = 1 + z\left(\frac{F(z,1)-F(z,u)}{1-u} + P(u)F(z,u) + \sum_{k=0}^{b-1} r_k(u)F_k(z) - \sum_{\gamma\in C} u^\gamma G_\gamma(z)\right), \quad (13)$$

where, as above,

$$P(u) = \sum_{\alpha\in A} u^\alpha - \sum_{\beta\in B} u^{-\beta} \quad \text{and} \quad r_k(u) = \sum_{\beta>k,\ \beta\in B} u^{k-\beta},$$

the new terms in the equations being

$$G_\gamma(z) = F(z,1) - \sum_{k=0}^{\gamma} F_k(z) - \sum_{\beta\in B} F_{\beta+\gamma}(z).$$

Observe that the first three terms are the same as in the case $C=\emptyset$. The equation, once multiplied by $u^b(1-u)$, reads $K(z,u)F(z,u) = R(z,u)$ where $K(z,u)$ is given by (11) and

$$R(z,u) = u^b(1-u)\left(1 + \frac{zF(z,1)}{1-u} + z\sum_{k=0}^{b-1} r_k(u)F_k(z) - z\sum_{\gamma\in C} u^\gamma G_\gamma(z)\right).$$

The kernel is not modified by the introduction of C. As above, it has degree $a+b+1$ in u, and admits $b+1$ finite roots $u_0,\dots,u_b$ around $z=0$. However, $R(z,u)$ now involves $b+1+|C|$ unknown functions, namely $F(z,1)$, the $F_k(z)$, $0\le k<b$ and the $G_\gamma(z)$, $\gamma\in C$. The degree of R in u is no longer $b+1$ but $b+c+1$, where $c=\max C$. The $b+1$ roots of K that can be substituted for u in Eq. (13) provide $b+1$ linear equations between the $b+|C|+1$ unknown functions. Additional equations will be obtained by extracting the coefficient of u^j from Eq. (13), for some values of j. In general, we have:

$$F_j(z) = [j=0] + z\sum_{\alpha\in A} F_{j-\alpha}(z) + z[j\notin C]\left(F(z,1) - \sum_{k=0}^{j} F_k(z) - \sum_{\beta\in B} F_{j+\beta}(z)\right). \quad (14)$$

It is possible to construct a finite subset $S\subset\mathbb{N}$ such that the combination of the $b+1$ equations obtained via the kernel method and the equations (14) written for $j\in S$ determines all unknown functions as algebraic functions of z – more precisely, as rational functions of z and the roots $u_0,\dots,u_b$ of the kernel. However, this is a long development, and these classes of walks play a marginal role in the context of ECO-systems. For these reasons, we shall merely give two examples. The details on the general procedure for constructing the set S can be found in [7].

EXAMPLE 11.
This example is obtained by modifying the Motzkin rule of Example 7 around the point 0. Take $A = C = \{1\}$ and $B = \emptyset$. The rewriting rule is

$$(k) \rightsquigarrow (0)(2)(3)\cdots(k-1)(k+1).$$

The functional equation reads

$$(1-u+z-zu(1-u))F(z,u) = 1-u+zF(z,1)-zu(1-u)G_1(z), \tag{15}$$

with $G_1(z) = F(z,1) - F_0(z) - F_1(z)$. The kernel has a *unique* finite root at $z = 0$:

$$u_0 = \frac{1+z-\sqrt{1-2z-3z^2}}{2z},$$

whereas the right-hand side of Eq. (15) contains *two* unknown functions. Writing Eq. (14) for $j = 0$ and $j = 1$ yields

$$F_0(z) = 1 + z(F(z,1) - F_0(z)) \quad \text{and} \quad F_1(z) = zF_0(z).$$

These two equations allow us to express F_0 and F_1, and hence G_1, in terms of $F(z,1)$:

$$G_1(z) = (1-z)F(z,1) - 1.$$

This equation relates the two unknown functions of Eq. (15). We replace $G_1(z)$ by the above expression in (15), so that only one unknown function, namely $F(z,1)$, is left. The kernel method finally gives:

$$F(z,1) = \frac{3-3z^2-2z^3-(1+z)\sqrt{1-2z-3z^2}}{2(1-z-z^2+z^3+z^4)} = 1+z+2z^2+3z^3+6z^4+12z^5+O(z^6).$$

□

EXAMPLE 12.
Let us choose $A = \{1\}$, $B = \{2\}$ et $C = \{2\}$. The rewriting rule is now:

$$(k) \to (0)(1)(3)(4)(5)\ldots(k-3)(k-1)(k+1).$$

The functional equation reads

$$\left[u^2(1-u) + zu^2 - zu^3(1-u) + z(1-u)\right] F(z,u)$$

$$= u^2(1-u) + zu^2F(z,1) + z(1-u)\left[F_0(z) + uF_1(z)\right] - zu^4(1-u)G_2(z), \tag{16}$$

with $G_2(z) = F(z,1) - F_0(z) - F_1(z) - F_2(z) - F_4(z)$. Only three roots, u_0, u_1, u_2 can be substituted for u in the kernel, while the right-hand side of the equation contains four unknown functions, $F(z,1), F_0(z), F_1(z)$ and $G_2(z)$. Writing (14) for $j = 0, 1$ and 2 yields

$$\begin{aligned} F_0(z) &= 1 + z\left[F(z,1) - F_0(z) - F_2(z)\right], \\ F_1(z) &= zF_0(z) + z\left[F(z,1) - F_0(z) - F_1(z) - F_3(z)\right], \\ F_2(z) &= zF_1(z). \end{aligned}$$

The second equation is not of much use but, by combining the first and third one, we find

$$F_0(z) = \frac{1 + z\left[F(z,1) - zF_1(z)\right]}{1+z}.$$

Replacing $F_0(z)$ by this expression in (16) gives:

$$\left[u^2(1-u) + zu^2 - zu^3(1-u) + z(1-u)\right] F(z,u) = u^2(1-u) + \frac{z(1-u)}{1+z}$$

$$+zF(z,1)\left[u^2 + \frac{z(1-u)}{1+z}\right] + z(1-u)F_1(z)\left[u - \frac{z^2}{1+z}\right] - zu^4(1-u)G_2(z). \qquad (17)$$

We are left with three unknown functions, related by three linear equations obtained by cancelling the kernel. Solving these equations would give $F(z,1)$ as an enormous rational function of z, u_0, u_1 and u_2, symmetric in the u_i. This implies that $F(z,1)$ can also be written as a rational function of z and $v \equiv v_1$, the fourth and last root of the kernel. In particular, $F(z,1)$ is algebraic of degree at most 4.

In order to obtain directly an expression of $F(z,1)$ in terms of z and v, we can proceed as follows. Let $R'(z,u)$ denote the right-hand side of Eq. (17). Then $R'(z,u)$ is a polynomial in u of degree 5, and three of its roots are u_0, u_1, u_2. Consequently, as a polynomial in u, the kernel $K(z,u)$ divides $(u-v)R'(z,u)$.

Let us evaluate $(u-v)R'(z,u)$ modulo $K(z,u)$: we obtain a polynomial of degree 3 in u, whose coefficients depend on $z, v, F(z,1), F_1(z)$ and $G_2(z)$. This polynomial has to be zero: this gives a system of four (dependent) equations relating the three unknown functions $F(z,1), F_1(z)$ and $G_2(z)$. Solving the first three of these equations yields

$$\begin{aligned} F(z,1) &= \frac{1+z+z^2-(z+1)zv+(z+1)zv^2-z^2v^3}{1-z^2-z(1-z^2)v+z^3v^3} \\ &= 1+z+2z^2+3z^3+6z^4+11z^5+23z^6+47z^7+101x^8+O(z^9). \end{aligned}$$

Eliminating v between this expression and $K(z,v)=0$ gives a quartic equation satisfied by $F(z,1)$. □

4 Transcendental systems

4.1 Transcendence

The radius of convergence of an algebraic series is always positive. Hence, one possible reason for a system to give a transcendental series is the fact that its coefficients grow too fast, so that its radius of convergence is zero. This is the case for System (2.h), as proved by the following proposition.

Proposition 6 *Let b be a nonnegative integer. For $k \geq 1$, let $m(k) = |\{i : \ e_i(k) \geq k-b\}|$. Assume that:*

1. *for all k, there exists a forward jump from k (i.e., $e_i(k) > k$ for some i),*
2. *the sequence $(m(k))_k$ is nondecreasing and tends to infinity.*

Then the (ordinary) generating function of the system has radius of convergence 0.

Proof. Let s_0 be the axiom of the system. Let us denote by h_n the product $m(s_0+b)m(s_0+2b)\cdots m(s_0+nb)$. Let us prove that the generating tree contains at least h_n nodes at level $n(b+1)$. At level nb, take a node v labeled k, with $k \geq s_0+nb$. Such a node exists thanks to the first assumption. By definition of $m(k)$, this node v has $m(k)$ sons whose label is at least $k-b$. As m is non decreasing, v has at least $m(s_0+nb)$ sons of label at least $s_0+(n-1)b$. Iterating this procedure shows that, at level $nb+i$, at least $m(s_0+(n-i+1)b)\cdots m(s_0+nb)$ descendents of v have a label larger than or equal to $s_0+(n-i)b$, for $0<i\leq n$. In particular, for $i=n$, we obtain at level $n(b+1)$ at least h_n descendents of v whose label is at least s_0.

Hence $f_{n(b+1)} \geq h_n$. But as $h_n/h_{n-1} = m(s_0+nb)$ goes to infinity with n, the series $\sum_n h_n z^{n(b+1)}$ has radius of convergence 0, and the same is true for $F(z)=\sum_n f_n z^n$. ∎

In particular, this proposition implies that *the generating function of any ECO-system in which the length of backward jumps is bounded has radius of convergence* 0. Many examples of this type will be given in the next subsection, in which we shall study whether the corresponding generating function is holonomic or not. The following example, in which backward jumps are not bounded, was suggested by Nantel Bergeron.

EXAMPLE 13. *A fake factorial walk*
Consider the system with axiom (1) and rewriting rules $\{(k) \rightsquigarrow (2)(4)\cdots(2k)\}$. Proposition 6 applies with $b=0$ and $m(k)=1+\lfloor k/2 \rfloor$. Note that the radius of convergence of $F(z)$ is zero although *all* the functions e_i are bounded, and indeed constant: $e_i(k)=2i$ for all $k \geq i$. The series $F(z)$ is of course transcendental. Note, however, that $F(z,u)$ satisfies a functional equation that is at first sight reminiscent of the equations studied in Section 3:

$$F(z,u) = u + zu^2\,\frac{F(z,1)-F(z,u^2)}{1-u^2}.$$

□

The following example shows that Proposition 6 is not far from optimal: an ECO-system in which all functions e_i grow linearly can have a finite radius of convergence.

EXAMPLE 14.
The system with axiom (1) and rules $(k) \rightsquigarrow (\lceil k/2 \rceil)^{k-1}(k+1)$ leads to a generating function with a positive radius of convergence.
Let us start from the recursion defining the numbers $f_{n,k}$. We have $f_{0,1}=1$ and for $n \geq 1$,

$$f_{n+1,k} = f_{n,k-1} + (2k-1)f_{n,2k} + (2k-2)f_{n,2k-1}.$$

The largest label occurring at level n in the tree is $n+1$. Let us introduce the numbers $g_{n,k} = f_{n,n-k+1}$, for $k \leq n$. The above recursion can be rewritten as:

$$g_{n+1,k} = g_{n,k} + (2n-2k+3)g_{n,2k-n-3} + (2n-2k+2)g_{n,2k-n-2}. \tag{18}$$

We have $g_{n,k}=0$ for $k<0$. Hence Eq. (18) implies that for $k \geq 0$, the sequence $(g_{n,k})_n$ is nondecreasing and reaches a constant value $g(k)$ as soon as $n \geq 2k-1$ (see Table 1).

Going back to the number f_n of nodes at level n, we have

$$f_n = \sum_{k=0}^{n} g_{n,k} \leq \sum_{k=0}^{n} g(k).$$

n k	1	2	3	4	5	6
0	**1**					
1	**0**	**1**				
2	1	**0**	**1**			
3	0	**3**	**0**	**1**		
4	3	3	**3**	**0**	**1**	
5	3	9	**7**	**3**	**0**	**1**

n k	0	1	2	3	4	5
0	**1**					
1	**1**	**0**				
2	**1**	**0**	1			
3	**1**	**0**	**3**	0		
4	**1**	**0**	**3**	3	3	
5	**1**	**0**	**3**	**7**	9	3

Table 1: The numbers $f_{n,k}$ and $g_{n,k}$. Observe the convergence of the coefficients.

But

$$\sum_{n\geq 0} z^n \sum_{k=0}^{n} g(k) = \frac{1}{1-z} \sum_{k=0}^{n} g(k) z^k,$$

and hence it suffices to prove that the generating function for the numbers $g(k)$ has a finite radius of convergence, that is, that these numbers grow at most exponentially.

Writing (18) for $n+1 = 2k-i$, for $1 \leq i \leq k$, we obtain:

$$g_{2k-i,k} = g_{2k-i-1,k} + (2k-2i+1) g_{2k-i-1,i-2} + (2k-2i) g_{2k-i-1,i-1}.$$

Iterating this formula for i between 1 and k yields

$$\begin{aligned} g(k) &= g_{2k-1,k} = \sum_{i=1}^{k} [(2k-2i+1) g_{2k-i-1,i-2} + (2k-2i) g_{2k-i-1,i-1}] \\ &\leq \sum_{i=1}^{k} [(2k-2i+1) g(i-2) + (2k-2i) g(i-1)] = \sum_{i=0}^{k-2} (4k-4i-5) g(i). \end{aligned}$$

This inequality, together with the fact that $g(0) = 1$, implies that for all $k \geq 0$, $g(k) \leq \widetilde{g}(k)$, where the sequence $\widetilde{g}(k)$ is defined by $\widetilde{g}(0) = 1$ and $\widetilde{g}(k) = \sum_{i=0}^{k-2} (4k-4i-5)\widetilde{g}(i)$ for $k > 0$. But the series $\sum_k \widetilde{g}(k) z^k$ is rational, equal to $(1-z)^2/(1-2z-2z^2-z^3)$, and has a finite radius of convergence. Consequently, the numbers $\widetilde{g}(k)$ and $g(k)$ grow at most exponentially. □

Algebraic generating functions are strongly constrained in their algebraic structure (by a polynomial equation) as well as in their analytic structure (in terms of singularities and asymptotic behaviour). In particular, they have a finite number of singularities, which are algebraic numbers, and they admit local asymptotic expansions that involve only rational exponents. *A contrario*, a generating function that has infinitely many singularities (*e.g.*, a natural boundary) or that involves a transcendental element (*e.g.*, a logarithm) in a local asymptotic expansion is by necessity transcendental; see [16] for a discussion of such transcendence criteria. In the case of generating trees, this means that the presence of a condition involving a transcendental element is expected to lead to a transcendental generating function. This is the case in the following example.

EXAMPLE 15. *A Fredholm system*
We examine System (2.*e*), in which the rules are irregular at powers of 2:

$$(s_0) = (2), \qquad (k) \rightsquigarrow (2)^{k-2}(3 - [\exists p : k = 2^p])(k+1), \quad k \geq 2.$$

This example will involve the Fredholm series $h(z) := \sum_{p\geq 1} z^{2^p}$, which is well-known to admit the unit circle as a natural boundary. (This can be seen by way of the functional equation $h(z) = z^2 + h(z^2)$, from which there results that $h(z)$ is infinite at all iterated square-roots of unity.) According to Eq. (2), we have, for $k > 3$, $F_k(z) = zF_{k-1}(z)$, so that

$$F_k(z) = z^{k-3}F_3(z) \quad \text{for } k \geq 3.$$

Now, writing Eq. (2) for $k = 2$ gives

$$\begin{aligned} F_2(z) &= 1 + z\sum_{k\geq 3}(k-2)F_k(z) + z\sum_{p\geq 1} F_{2^p}(z) \\ &= 1 + \frac{z}{(1-z)^2}F_3(z) + zF_2(z) + F_3(z)\left(\frac{h(z)}{z^2} - 1\right) \\ &= 1 + zF_2(z) + F_3(z)\left(\frac{z}{(1-z)^2} + \frac{h(z)}{z^2} - 1\right). \end{aligned}$$

For $k = 3$, we obtain:

$$\begin{aligned} F_3(z) &= zF_2(z) + z\sum_{k\geq 3,\ k\neq 2^p} F_k(z) \\ &= zF_2(z) + F_3(z)\left(\frac{1}{1-z} - \frac{h(z)}{z^2}\right). \end{aligned}$$

Solving for $F_2(z)$ and $F_3(z)$, then summing ($F(z) = F_2(z) + F_3(z)/(1-z)$), we obtain:

$$F(z) = \frac{(1-z)^2 h(z)}{(1-2z)(1-z)^2 h(z) - z^4} = 1 + 2z + 5z^2 + 14z^3 + 39z^4 + 108z^5 + O(x^6).$$

The functions $h(z)$ and $F(z)$ are rationally related, so that $F(z)$ is itself transcendental. The series h has radius 1, but the denominator of F vanishes before z reaches 1 – actually, before z reaches $1/2$. Hence the radius of F is the smallest root of its denominator. Its value is easily determined numerically and found to be about 0.360102. □

4.2 Holonomy

In the transcendental case, one can also discuss the *holonomic* character of the generating function $F(z)$.

A series is said to be *holonomic*, or *D-finite* [25], if it satisfies a linear differential equation with polynomial coefficients in z. Equivalently, its coefficients f_n satisfy a linear recurrence relation with polynomial coefficients in n. Consequently, given a sequence f_n, the ordinary generating function $\sum_n f_n z^n$ is holonomic if and only if the exponential generating function $\sum_n f_n z^n/n!$ is holonomic. The set of holonomic series has nice closure properties: the sum or product of two of them is still holonomic, and the substitution of an algebraic series into an holonomic one gives an holonomic series. Holonomic series include algebraic series, and have a finite number of singularities. This implies that Example 15, for which $F(z)$ has a natural boundary, is not holonomic.

We study below five ECO-systems that, at first sight, do not look to be very different. In particular, for each of them, forward and backward jumps are bounded. Consequently,

Proposition 6 implies that the corresponding ordinary generating function has radius of convergence zero. However, we shall see that the first three systems have an holonomic generating function, while the last two have not. We have no general criterion that would allow us to distinguish between systems leading to holonomic generating functions and those leading to nonholonomic generating functions.

Among the systems with bounded jumps, those for which $e_i(k) - k$ belongs to $\{-1, 0, 1\}$ for all $i \leq k$ have a nice property: the generating function for the corresponding *excursions* (walks starting and ending at level 0) can be written as the following continued fraction [15]:

$$\cfrac{1}{1 - b_0 z - \cfrac{a_1 c_0 z^2}{1 - b_1 z - \cfrac{a_2 c_1 z^2}{1 - b_2 z - \cfrac{a_3 c_2 z^2}{\cdots}}}},$$

where the coefficients a_k, b_k and c_k are the multiplicities appearing in the rules, which read $(k) \rightsquigarrow (k-1)^{a_k}(k)^{b_k}(k+1)^{c_k}$.

EXAMPLE 16. *Arrangements*
The system $(k) \rightsquigarrow (k)(k+1)^{k-1}$ with axiom $(s_0) = (2)$ generates a sequence that starts with $1, 2, 5, 16, 65, 326$ (**M1497**). It is not hard to see that the triangular array $f_{n,k+2}$ is given by the arrangement numbers $k!\binom{n}{k}$, so that the *exponential* generating function (EGF) of the sequence is

$$\widetilde{F}(z, u) = \sum_{n \geq 0, k \geq 2} f_{n,k} u^k \frac{z^n}{n!} = \frac{u^2 e^z}{1 - uz}.$$

This system satisfies the conditions of Proposition 6 with $b = 0$ and $m(k) = k$. Accordingly, one has $f_n \sim e\, n!$, so that the *ordinary* generating function $F(z)$ has radius of convergence 0 and cannot be algebraic. However, $\widetilde{F}(z, 1) = e^z/(1 - z)$ is holonomic, and so is $F(z)$. □

EXAMPLE 17. *Involutions and Hermite polynomials*
The system $(k) \rightsquigarrow (k-1)^{k-1}(k+1)$ with axiom $(s_0) = (1)$ generates a sequence that starts with $1, 1, 2, 4, 10, 26, 76$ (**M1221**). These numbers count involutions: more precisely, one easily derives from the recursion satisfied by the coefficients $f_{n,k}$ that $f_{n,k}$ is the number of involutions on n points, $k - 1$ of which are fixed. Proposition 6 applies with $b = 1$ and $m(k) = k$.

The corresponding EGF is

$$\widetilde{F}(z, u) = \sum_{n \geq 0, k \geq 1} f_{n,k} u^k \frac{z^n}{n!} = u \exp\left(zu + \frac{z^2}{2}\right), \tag{19}$$

and its value at $u = 1$ is holonomic.

The polynomials $f_n(u) = \sum_k f_{n,k} u^k$ counting involutions on n points are in fact closely related to the Hermite polynomials, defined by:

$$\sum_{n \geq 0} H_n(x) \frac{t^n}{n!} = \exp\left(xt - \frac{t^2}{2}\right).$$

Indeed, comparing the above identity with (19) shows that $f_n(u) = u\ i^n H_n(-iu)$. □

EXAMPLE 18. *Partial permutations and Laguerre polynomials*
The rewriting rule $(k) \rightsquigarrow (k+1)^{k-1}(k+2)$, taken with axiom (2), generates a sequence that starts with $1, 2, 7, 34, 209, \ldots$ (**M1795**). From the recursion satisfied by the coefficients $f_{n,k}$, we derive that $f_{n,n+k}$ is the number of partial injections of $\{1, 2, \ldots, n\}$ into itself in which $k-2$ points are unmatched. From this, we obtain:

$$\widetilde{F}(z,u) = \frac{u^2}{1-uz} \exp\left(\frac{u^2 z}{1-uz}\right) = u^2 \sum_{n\geq 0} L_n(-u) \frac{(uz)^n}{n!}$$

where $L_n(u)$ is the nth Laguerre polynomial. Again, $\widetilde{F}(z,1)$ is holonomic. □

The next two systems, as announced, lead to nonholonomic generating functions.

EXAMPLE 19. *Set partitions and Stirling polynomials*
Let us consider the system $[(1),(k) \rightsquigarrow (k)^{k-1}(k+1)]$. From the recursion satisfied by the coefficients $f_{n,k}$, we derive that $f_{n,k+1}$ is equal to the Stirling number of the second kind $\left\{ {n \atop k} \right\}$, which counts partitions of n objects into k nonempty subsets. The corresponding EGF is

$$\widetilde{F}(z,u) = u \exp\left(u(\exp z - 1)\right).$$

At $u = 1$, this generating function specializes to

$$\widetilde{F}(z,1) = \exp(\exp(z)-1)) = \sum_{n\geq 0} B_n \frac{z^n}{n!} = 1 + z + 2\frac{z^2}{2!} + 5\frac{z^3}{3!} + 15\frac{z^4}{4!} + 52\frac{z^5}{5!} + 203\frac{z^6}{6!} + \ldots$$

This is the exponential generating function of the Bell numbers (**M1484**). It is known that $\log B_n = n\log n - n\log\log n + O(n)$ (see [20]), and this cannot be the asymptotic behaviour of the logarithm of the coefficients of an holonomic series (see [29] for admissible types). Hence, $\widetilde{F}(z,1)$, as well as $F(z,1)$, is nonholonomic. □

EXAMPLE 20. *Bessel numbers*
We study the system with axiom (2) and rewriting rules

$$(2) \rightsquigarrow (2)(3), \quad (k) \rightsquigarrow (k-1)(k)^{k-2}(k+1), \quad k \geq 3. \tag{20}$$

We shift the labels by 2 to obtain a walk model with axiom (0) and rules

$$(0) \rightsquigarrow (0)(1), \quad (k) \rightsquigarrow (k-1)(k)^{k}(k+1), \quad k \geq 1.$$

The corresponding bivariate generating function $F(z,u)$ satisfies the functional differential equation

$$F(z,u)\left(1 - z(u+u^{-1})\right) = 1 + z(1-u^{-1})F(z,0) + zu\frac{\partial F}{\partial u}(z,u),$$

which is certainly not obvious to solve. However, as observed in [15], it is easy to obtain a continued fraction expansion of the excursion generating function:

$$F(z,0) = 1+z+2z^2+4z^3+9z^4+\cdots = \cfrac{1}{1-z-\cfrac{z^2}{1-z-\cfrac{z^2}{1-2z-\cfrac{z^2}{1-3z-\ddots}}}} = \frac{1}{1-z-z^2B(z)},$$

where $B(z) = \sum_n B_n^* z^n = 1 + z + 2z^2 + 5z^3 + 14z^4 + 43z^5 + 143z^6 + \cdots$ is the generating function of Bessel numbers (**M1462**) and counts non-overlapping partitions [17]. As $F(z,0)$ itself, the series $B(z)$ has radius of convergence zero. The fast increase of B_n^* entails

$$[z^n]F(z,0) \sim B_{n-2}^*.$$

From [17], we know that $\log B_n^* = n \log n - n \log\log n + O(n)$. Again, this prevents $F(z,0)$ from being holonomic.

In order to prove that $F(z,1)$ itself is nonholonomic, we are going to prove that its coefficients f_n have the same asymptotic behaviour as the coefficients of $F(z,0)$. Clearly,

$$[z^n]F(z,0) = f_{n,0} \leq \sum_k f_{n,k} = f_n.$$

To find an upper bound for f_n, we compare the system (20) (denoted Σ_1 below) to the system Σ_2 with axiom (2) and rule $(k) \rightsquigarrow (k)^{k-1}(k+1)$. This system generates a tree with counting sequence g_n. The form of the rules implies that the (unlabeled) tree associated with Σ_1 is a subtree of the tree associated with Σ_2. Hence $f_n \leq g_n$. Comparing Σ_2 to the system studied in the previous example shows that g_n is the Bell number B_{n+1}, the logarithm of which is also known to be $n \log n - n \log\log n + O(n)$ (see [20]). Hence $\log f_n = n \log n - n \log\log n + O(n)$, and this prevents the series $F(z,1)$ from being holonomic. □

Axiom	System	Name	Id.	Generating Function
	Rational OGF			**OGF**
(1)	$(k) \rightsquigarrow (k)^{k-1}((k \bmod 2)+1)$	Ex. 3: Fibonacci	M0692	$\frac{1}{1-z-z^2}$
(2)	$(k) \rightsquigarrow (2)^{k-1}(k+1)$	Ex. 4: even Fibonacci	M1439	$\frac{1-z}{1-3z+z^2}$
(3)	$(k) \rightsquigarrow (2)^{k-1}(k+1)$	Ex. 4: odd Fibonacci	M2741	$\frac{1}{1-3z+z^2}$
	Algebraic OGF			**OGF**
(1)	$(k) \rightsquigarrow (1)\cdots(k-1)(k+1)$	Ex. 7: Motzkin numbers	M1184	$\frac{1-z-\sqrt{1-2z-3z^2}}{2z^2}$
(2)	$(k) \rightsquigarrow (2)\cdots(k)(k+1)$	Ex. 6: Catalan numbers	M1459	$\frac{1-2z-\sqrt{1-4z}}{2z^2}$
(3)	$(k) \rightsquigarrow (3)\cdots(k)(k+1)^2$	Ex. 8: Schröder numbers	M2898	$\frac{1-3z-\sqrt{1-6z+z^2}}{4z^2}$
(4)	$(k) \rightsquigarrow (4)\cdots(k)(k+1)^3$	—	M3556	$\frac{1-4z-\sqrt{1-8z+4z^2}}{6z^2}$
(m)	$(k) \rightsquigarrow (m)\cdots(k)(k+1)^{m-1}$	—	—	$\frac{1-mz-\sqrt{1-2mz+(m-2)^2z^2}}{2(m-1)z^2}$
(3)	$(k) \rightsquigarrow (3)\cdots(k+2)$	Ex. 9: Ternary trees	M2926	$F=(1+zF)^3$
(4)	$(k) \rightsquigarrow (4)\cdots(k+3)$	Ex. 9: Dissections of a polygon	M3587	$F=(1+zF)^4$
(m)	$(k) \rightsquigarrow (m)\cdots(k+m-1)$	Ex. 9: m-ary trees		$F=(1+zF)^m$
	Holonomic transcendental OGF			**EGF**
(1)	$(k) \rightsquigarrow (k+1)^k$	Permutations	M1675	$1/(1-z)$
(2)	$(k) \rightsquigarrow (k)(k+1)^{k-1}$	Ex. 16: Arrangements	M1497	$e^z/(1-z)$
(1)	$(k) \rightsquigarrow (k-1)^{k-1}(k+1)$	Ex. 17: Involutions	M1221	$e^{z+\frac{1}{2}z^2}$
(2)	$(k) \rightsquigarrow (k+1)^{k-1}(k+2)$	Ex. 18: Partial permutations	M1795	$e^{z/(1-z)}/(1-z)$
(2)	$(k) \rightsquigarrow (k-1)^{k-2}(k)(k+1)$	Switchboard problem	M1461	$e^{2z+\frac{1}{2}z^2}$
(2)	$(k) \rightsquigarrow (k-1)^{k-2}(k+1)^2$	Bicolored involutions	M1648	e^{2z+z^2}
	Nonholonomic OGF			**EGF**
(1)	$(k) \rightsquigarrow (k)^{k-1}(k+1)$	Ex. 19: Bell numbers	M1484	e^{e^z-1}
(2)	$(k) \rightsquigarrow (k)^{k-2}(k+1)^2$	Bicolored partitions	M1662	$e^{2(e^z-1)}$
(2)	$(k) \rightsquigarrow (k-1)(k)^{k-2}(k+1)$	Ex. 20: Bessel numbers	M1462	—

Table 2: Some ECO-systems of combinatorial interest.

A small catalog of ECO-systems
To conclude, we present in Table 2 a small catalog of ECO-systems that lead to sequences of combinatorial interest. Several examples are detailed in the paper; others are due to West [27, 28] or Barcucci, Del Lungo, Pergola, Pinzani [4, 6, 5, 3], or are folklore. Each of them is an instance of application of our criteria.

Acknowledgements. We thank Elisa Pergola and Renzo Pinzani who presented us the problem we deal with in this paper. We are also very grateful for helpful discussions with Jean-Paul Allouche.

References

[1] C. Banderier. Combinatoire analytique : application aux marches aléatoires. *Mémoire de DEA, Université Paris VII*, 1998.

[2] E. Barcucci, A. Del Lungo, and E. Pergola. Random generation of trees and other combinatorial objects. *Theoretical Computer Science*, 218(2):219–232, 1999.

[3] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. A methodology for plane tree enumeration. *Discrete Mathematics*, 180(1-3):45–64, 1998.

[4] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. ECO: a methodology for the enumeration of combinatorial objects. *Journal of Difference Equations and Applications*, 5:435–490, 1999.

[5] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. From Motzkin to Catalan permutations. *Discrete Mathematics*, 217(1–3):33–49, 2000.

[6] E. Barcucci, A. Del Lungo, E. Pergola, and R. Pinzani. Permutations avoiding an increasing number of length increasing forbidden subsequences. *Discrete Mathematics and Theoretical Computer Science*, 4(1):31–44, 2000.

[7] M. Bousquet-Mélou. Un modèle un peu plus général de marches sur $\mathbb{N}$. Manuscript, December 1998.

[8] M. Bousquet-Mélou. Multi-statistic enumeration of two-stack sortable permutations. *Electronic Journal of Combinatorics*, 5:R21, 1998.

[9] M. Bousquet-Mélou and M. Petkovšek. Linear recurrences with constant coefficients: the multivariate case. *Discrete Mathematics*, to appear.

[10] R. Cori and J. Richard. Énumération des graphes planaires à l'aide des séries formelles en variables non commutatives. *Discrete Mathematics*, 2:115–162, 1972.

[11] J. Dieudonné. *Infinitesimal calculus*. Hermann, Paris, 1971. Appendic 3.

[12] S. Dulucq, S. Gire, and O. Guibert. A combinatorial proof of J. West's conjecture. *Discrete Mathematics*, 187(1-3):71–96, 1998.

[13] S. Dulucq, S. Gire, and J. West. Permutations with forbidden subsequences and nonseparable planar maps. *Discrete Mathematics*, 153(1-3):85–103, 1996.

[14] G. Fayolle and R. Iasnogorodski. Solutions of functional equations arising in the analysis of two-server queueing models. In *Performance of computer systems*, pages 289–303. North-Holland, 1979.

[15] P. Flajolet. Combinatorial aspects of continued fractions. *Discrete Mathematics*, 32:125–161, 1980.

[16] P. Flajolet. Analytic models and ambiguity of context–free languages. *Theoretical Computer Science*, 49:283–309, 1987.

[17] P. Flajolet and R. Schott. Non–overlapping partitions, continued fractions, Bessel functions and a divergent series. *European Journal of Combinatorics*, 11:421–432, 1990.

[18] D. E. Knuth. *The Art of Computer Programming. Vol.* 1*: Fundamental Algorithms.* Addison-Wesley, 1968. Exercises 4 and 11, section 2.2.1.

[19] D. Kremer. Permutations with forbidden subsequences and a generalized Schröder number. *Discrete Mathematics*, 218:121–130, 2000.

[20] A. M. Odlyzko. Asymptotic enumeration methods. In Graham, Grötschel, and Lovász, editors, *Handbook of combinatorics*, volume 2, pages 1063–1229. Elsevier, Amsterdam, 1995. section 14.4.

[21] M. Petkovšek. The irrational chess knight. In *Proceedings of FPSAC'98*, pages 513–522, Toronto, 1998.

[22] O. Ramaré. On Šnirel'man's constant. *Annali della Scuola Normale Superiore di Pisa. Classe di Scienze. Serie IV*, 22(4):645–706, 1995.

[23] P. Ribenboim. *The little book of big primes.* Springer-Verlag, New York, 1991.

[24] N. J. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences.* Academic Press Inc., New York, 1995.

[25] R. P. Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1:175–188, 1980.

[26] R. P. Stanley. *Enumerative combinatorics, Vol.* 2, volume 62 of *Cambridge Studies in Advanced Mathematics.* Cambridge University Press, 1999.

[27] J. West. Generating trees and the Catalan and Schröder numbers. *Discrete Mathematics*, 146:247–262, 1995.

[28] J. West. Generating trees and forbidden subsequences. *Discrete Mathematics*, 157:363–374, 1996.

[29] J. Wimp and D. Zeilberger. Resurrecting the asymptotics of linear recurrences. *Journal of Mathematical Analysis and Applications*, 111:162–176, 1985.

Cyril Banderier, Philippe Flajolet
Projet Algorithmes
INRIA Rocquencourt
F-78153 Le Chesnay
France
Cyril.Banderier@inria.fr
Philippe.Flajolet@inria.fr

Mireille Bousquet-Mélou
LaBRI, Université Bordeaux 1
351 cours de la Libération
F-33405 Talence Cedex
France
bousquet@labri.u-bordeaux.fr

Alain Denise, Dominique Gouyou-Beauchamps
LRI, Bâtiment 490
Université Paris-Sud XI
F-91405 Orsay Cedex
France
Alain.Denise@lri.fr
dgb@lri.fr

Danièle Gardy
Université de Versailles/Saint-Quentin
Laboratoire PRISM
45, avenue des États-Unis
F-78035 Versailles Cedex
France
Daniele.Gardy@prism.uvsq.fr

ON DIRICHLET SERIES FOR SUMS OF SQUARES

JONATHAN MICHAEL BORWEIN AND KWOK-KWONG STEPHEN CHOI

ABSTRACT. In [14], Hardy and Wright recorded elegant closed forms for the generating functions of the divisor functions $\sigma_k(n)$ and $\sigma_k^2(n)$ in the terms of Riemann Zeta function $\zeta(s)$ only. In this paper, we explore other arithmetical functions enjoying this remarkable property. In Theorem 2.1 below, we are able to generalize the above result and prove that if f_i and g_i are completely multiplicative, then we have

$$\sum_{n=1}^{\infty} \frac{(f_1 * g_1)(n) \cdot (f_2 * g_2)(n)}{n^s} = \frac{L_{f_1 f_2}(s) L_{g_1 g_2}(s) L_{f_1 g_2}(s) L_{g_1 f_2}(s)}{L_{f_1 f_2 g_1 g_2}(2s)}$$

where $L_f(s) := \sum_{n=1}^{\infty} f(n) n^{-s}$ is the Dirichlet series corresponding to f. Let $r_N(n)$ be the number of solutions of $x_1^2 + \cdots + x_N^2 = n$ and $r_{2,P}(n)$ be the number of solutions of $x^2 + Py^2 = n$. One of the applications of Theorem 2.1 is to obtain closed forms, in terms of $\zeta(s)$ and Dirichlet L-functions, for the generating functions of $r_N(n), r_N^2(n), r_{2,P}(n)$ and $r_{2,P}(n)^2$ for certain N and P. We also use these generating functions to obtain asymptotic estimates of the average values for each function for which we obtain a Dirichlet series.

1. INTRODUCTION

Let σ_k denote the sum of kth powers of the divisors of n. It is also quite usual to write d for σ_0 and τ for σ_1. There is a beautiful formula for the generating functions of $\sigma_k(n)$ (see Theorem 291 in Chapter XVII of [14])

$$\sum_{n=1}^{\infty} \frac{\sigma_k(n)}{n^s} = \zeta(s)\zeta(s-k), \qquad \Re(s) > \max\{1, k+1\} \tag{1.1}$$

which is in terms of only the Riemann Zeta function $\zeta(s)$. Following Hardy and Wright, by standard techniques, one can prove the following remarkable identity due to Ramanujan (see [21]) (also see Theorem 305 in Chapter XVII of [14])

$$\sum_{n=1}^{\infty} \frac{\sigma_a(n)\sigma_b(n)}{n^s} = \frac{\zeta(s)\zeta(s-a)\zeta(s-b)\zeta(s-a-b)}{\zeta(2s-a-b)} \tag{1.2}$$

for $\Re(s) > \max\{1, a+1, b+1, a+b+1\}$. In this paper, we identify other arithmetical functions enjoying similarly explicit representations. In Theorem 2.1 of §2 below,

Date: February 6, 2002.

1991 *Mathematics Subject Classification.* Primary 11M41, 11E25.

Key words and phrases. Dirichlet Series, Sums of Squares, Closed Forms, Binary Quadratic Forms, Disjoint Discriminants, L-functions.

Research supported by NSERC and by the Canada Research Chair Programme.

CECM Preprint 01:167.

we are able to generalize the above result and prove that if f_i and g_i are completely multiplicative, then we have

$$\sum_{n=1}^{\infty} \frac{(f_1 * g_1)(n) \cdot (f_2 * g_2)(n)}{n^s} = \frac{L_{f_1 f_2}(s) L_{g_1 g_2}(s) L_{f_1 g_2}(s) L_{g_1 f_2}(s)}{L_{f_1 f_2 g_1 g_2}(2s)}$$

where $L_f(s) := \sum_{n=1}^{\infty} f(n) n^{-s}$ is the Dirichlet series corresponding to f. As we shall see, this result recovers Hardy and Wright's formulae (1.1) and (1.2) immediately.

More generally, for certain classes of Dirichlet series, $\sum_{n=1}^{\infty} A(n) n^{-s}$, our Theorem 2.1 can be applied to obtain closed forms for the series $\sum_{n=1}^{\infty} A^2(n) n^{-s}$. In particular, if the generating function $L_f(s)$ of an arithmetic function f is expressible as a sum of products of two L-functions:

$$L_f(s) = \sum_{\chi_1, \chi_2} a(\chi_1, \chi_2) L_{\chi_1}(s) L_{\chi_2}(s)$$

for certain coefficients $a(\chi_1, \chi_2)$ and Dirichlet characters χ_i, then we are able to find a simple closed form (in term of L-functions) for the generating function $L_f^2(s) := \sum_{n=1}^{\infty} f^2(n) n^{-s}$.

One of our central applications is to the study of the number of representations as a sum of squares. Let $r_N(n)$ be the number of solutions to $x_1^2 + x_2^2 + \cdots + x_N^2 = n$ (counting permutations and signs). Hardy and Wright record a classical closed form, due to Lorenz, of the generating function for $r_2(n)$ in the terms of $\zeta(s)$ and a Dirichlet L-function, namely,

$$\sum_{n=1}^{\infty} \frac{r_2(n)}{n^s} = 4\zeta(s) L_{-4}(s)$$

where $L_\mu(s) = \sum_{n=1}^{\infty} \left(\frac{\mu}{n}\right) n^{-s}$ is the *primitive L-function* corresponding to the *Kronecker symbol* $\left(\frac{\mu}{n}\right)$. Define

$$\mathcal{L}_N(s) := \sum_{n=1}^{\infty} \frac{r_N(n)}{n^s} \qquad \text{and} \qquad \mathcal{R}_N(s) := \sum_{n=1}^{\infty} \frac{r_N^2(n)}{n^s}.$$

Simple closed forms for $\mathcal{L}_N(s)$ are known for $N = 2, 4, 6$ and 8; indeed the corresponding $q-$series were known to Jacobi. The entity $\mathcal{L}_3(s)$ in particular is still shrouded in mystery, as a series relevant to the study of lattice sums in the physical sciences. Lately there has appeared a connection between $\mathcal{L}_3$ and a modern theta-cubed identity of G. Andrews [1] which we list in (6.7), R. Crandall [6] and p.301 of [3]. In §3, we shall obtain simple closed forms for $\mathcal{R}_N(s)$ for these N from the corresponding $\mathcal{L}_N(s)$, via Theorem 2.1. Since the generating functions are accessible, by an elementary convolution argument, see §3 below, we are also able to deduce

$$\sum_{n \le x} r_N^2(n) = W_N x^{N-1} + O(x^{N-2})$$

for $N = 6, 8$ and for $N = 4$ with an error term $O(x^2 \log^5 x)$ where

$$W_N := \frac{1}{(N-1)(1-2^{-N})} \frac{\pi^N}{\Gamma^2(\frac{1}{2}N)} \frac{\zeta(N-1)}{\zeta(N)}, \qquad (N \ge 3). \tag{1.3}$$

This technique can be adjusted to handle all $N \geq 2$ except $N = 3$, see Theorem 3.3, and so to establish all but the most difficult case of the following general conjecture due to Wagon:

Wagon's Conjecture. *For* $N \geq 3, \sum_{n \leq x} r_N^2(n) \sim W_N x^{N-1}$ *as* $x \longrightarrow \infty$.

Now from (3.14) below, one has $\sum_{n \leq x} r_2^2(n) \sim 4x \log x$ so that Wagon's conjecture holds only for $N \geq 3$. This conjecture motivated our interest in such explicit series representations. Recently, it has been proved by Crandall and Wagon in [8]. In fact, they show that

$$\lim_{x \to \infty} x^{1-N} \sum_{n \leq x} r_N^2(n) = W_N,$$

with various rates of convergence (those authors found the $N = 3$ case especially difficult, with relevant computations revealing very slow convergence to the above limit). In their treatment of the Wagon conjecture and related matters, they needed to evaluate the following Dirichlet series

$$\sum_{n=1}^{\infty} \frac{\phi(n)\sigma_0(n^2)}{n^s}$$

and we have established, by an easier version of what follows, that it is

$$\sum_{n=1}^{\infty} \frac{\phi(n)\sigma_0(n^2)}{n^s} = \zeta^3(s-1) \prod_p \left(1 - \frac{3}{p^s} - \frac{1}{p^{2s-2}} + \frac{4}{p^{2s-1}} - \frac{1}{p^{3s-2}}\right)$$

where the product is over all primes. A word is in order concerning the importance of first- and second-order summatories. In a theoretical work [7] and a computational one [8] it is explained that the Wagon conjecture implies that *sums of three squares have positive density*. This interesting research connection is what inspired Wagon to posit his computationally motivated conjecture. Though it is known that the density of the set $S = \{x^2 + y^2 + z^2\}$ is exactly $5/6$ due to Landau (e.g [18] or [11]), there are intriguing signal-processing and analytic notions that lead more easily at least to positivity of said density. Briefly, the summatory connection runs as follows: from the Cauchy-Schwarz inequality we know

$$\#\{n < x; n \in S\} > \frac{\left(\sum_{n<x} r_3(n)\right)^2}{\sum_{n<x} r_3^2(n)},$$

so the Wagon conjecture even gives an explicit numerical lower bound on the density of S. Of course, the density for sums of more than 3 squares is likewise positive, and boundable, yet the Lagrange theorem that sums of four squares comprise *all* nonnegative integers dominates in the last analysis. Still, the signal-processing and computational notions of Crandall and Wagon forge an attractive link between these L-series of our current interest and additive number theory.

In §4 and §5, we similarly study the number of representations by a binary quadratic forms. Let $r_{2,P}(n)$ be the number of solutions of the binary quadratic form $x^2 + Py^2 = n$. Define

$$\mathcal{L}_{2,P}(s) := \sum_{n=1}^{\infty} \frac{r_{2,P}(n)}{n^s} \qquad \text{and} \qquad \mathcal{R}_{2,P}(s) := \sum_{n=1}^{\infty} \frac{r_{2,P}(n)^2}{n^s}.$$

The closed forms of $\mathcal{L}_{2,P}(s)$ has been studied by a number of people, particular by Glasser, Zucker and Robertson (see [10] and [23]). In finding the exact evaluation of lattice sums, they are interested in expressing a multiple sum, such as the generating functions of $r_{2,P}(n)$, as a product of simple sums. As a result, plenty of closed forms of Dirichlet series $\sum_{(n,m)\neq(0,0)}(am^2+bmn+cn^2)^{-s}$ in terms of L-functions have been found. One of the most interesting cases is when the binary quadratic forms have *disjoint discriminants*, i.e, have only one form per genus. Then there are simple closed forms for the corresponding $\mathcal{L}_{2,P}(s)$ (see (4.1) below). By applying Theorem 2.1, we obtain closed forms for $\mathcal{R}_{2,P}(s)$ and from this we also deduce asymptotic estimates for $r_{2,P}(n)$ and $r_{2,P}(n)^2$.

In the last section, we shall discuss $\mathcal{L}_N(s)$ for some other less tractable cases. In particular, we collect some representations of the generating function for $r_3(n), r_N(n)$, and discuss $r_{12}(n)$ and $r_{24}(n)$.

Throughout, our notation is consistent with that in [14, 15] and [16]. We should also remark that we were lead to the structures exhibited herein by a significant amount of numeric and symbolic computation: leading to knowledge of the formulae for $\mathcal{R}_2, \mathcal{R}_4, \mathcal{R}_8, \mathcal{R}_{2,2}$ and $\mathcal{R}_{2,3}$ before finding our general results. And indeed R. Crandall triggered our interest by transmitting his formula for $\mathcal{R}_4$.

2. Basic Results

Let $\sigma(f)$ be the *abscissa* of absolute convergence of the Dirichlet series

$$L_f(s) := \sum_{n=1}^{\infty} f(n)n^{-s}.$$

For any two arithmetic functions f and g, define

$$f * g(n) := \sum_{d|n} f(d)g(n/d)$$

to be the *convolution* of f and g.

Theorem 2.1. *Suppose f_1, f_2 and g_1, g_2 are completely multiplicative arithmetic functions. Then for $\Re(s) \geq \max\{\sigma(f_i), \sigma(g_i)\}$, we have*

$$\sum_{n=1}^{\infty} \frac{(f_1 * g_1)(n) \cdot (f_2 * g_2)(n)}{n^s} = \frac{L_{f_1f_2}(s)L_{g_1g_2}(s)L_{f_1g_2}(s)L_{g_1f_2}(s)}{L_{f_1f_2g_1g_2}(2s)}. \tag{2.1}$$

Proof. Since $(f_1 * g_1)(n) \cdot (f_2 * g_2)(n)$ is multiplicative, we only need to consider its values at the prime powers. For any prime p and any $l \geq 0$,

$$(f_i * g_i)(p^l) = \sum_{d|p^l} f_i(d)g_i(p^l/d) = \frac{f_i(p)^{l+1} - g_i(p)^{l+1}}{f_i(p) - g_i(p)},$$

as each of f_1, f_2, g_1, g_2 is completely multiplicative. We intend above that if both $f_i(p)$ and $g_i(p)$ are zero, then

$$(f_i * g_i)(p^l) = \begin{cases} 1 & \text{if } l = 0; \\ 0 & \text{if } l \geq 1. \end{cases}$$

Thus, we have

$$\begin{aligned}\Sigma_p &:= \sum_{l=0}^{\infty}(f_1*g_1)(p^l)(f_2*g_2)(p^l)p^{-ls}\\ &= \sum_{l=0}^{\infty}\frac{(f_1(p)^{l+1}-g_1(p)^{l+1})(f_2(p)^{l+1}-g_2(p)^{l+1})}{(f_1(p)-g_1(p))(f_2(p)-g_2(p))}p^{-ls}\\ &= \frac{\sum_{l=0}^{\infty}\left\{(f_1f_2)(p)^{l+1}p^{-ls}+(g_1g_2)(p)^{l+1}p^{-ls}-(f_1g_2)(p)^{l+1}p^{-ls}-(g_1f_2)(p)^{l+1}p^{-ls}\right\}}{(f_1(p)-g_1(p))(f_2(p)-g_2(p))}.\end{aligned}$$

On summing up all the geometric series, we arrive at

$$\begin{aligned}\Sigma_p &:= \frac{\frac{(f_1f_2)(p)}{1-(f_1f_2)(p)p^{-s}}+\frac{(g_1g_2)(p)}{1-(g_1g_2)(p)p^{-s}}-\frac{(f_1g_2)(p)}{1-(f_1g_2)(p)p^{-s}}-\frac{(g_1f_2)(p)}{1-(g_1f_2)(p)p^{-s}}}{(f_1(p)-g_1(p))(f_2(p)-g_2(p))}\\ &= \frac{1-(f_1f_2g_1g_2)(p)p^{-2s}}{(1-(f_1f_2)(p)p^{-s})(1-(g_1g_2)(p)p^{-s})(1-(f_1g_2)(p)p^{-s})(1-(g_1f_2)(p)p^{-s})}.\end{aligned}$$

In view of the Euler product form for a Dirichlet series, we have

$$\begin{aligned}\sum_{n=1}^{\infty}\frac{(f_1*g_1)(n)\cdot(f_2*g_2)(n)}{n^s} &= \prod_p\left\{\sum_{l=0}^{\infty}\frac{(f_1*g_1)(p^l)(f_2*g_2)(p^l)}{p^{ls}}\right\}\\ &= \frac{L_{f_1f_2}(s)L_{g_1g_2}(s)L_{f_1g_2}(s)L_{g_1f_2}(s)}{L_{f_1f_2g_1g_2}(2s)}.\end{aligned}$$

This proves our theorem. □

A first easy application of Theorem 2.1 is to evaluate the Dirichlet series $\sum_{n=1}^{\infty}\sigma_k(n)n^{-s}$ and $\sum_{n=1}^{\infty}\sigma_a(n)\sigma_b(n)n^{-s}$. If we let $f_1(n):=n^k$, $f_2(n):=\delta(n)$ and $g_1(n)=g_2(n):=1$ where $\delta(n)$ is 1 if $n=1$ and 0 otherwise, then

$$L_{f_1f_2}(s)=L_{g_1f_2}(s)=L_{f_1f_2g_1g_2}(s)=1,$$

$$L_{f_1g_2}(s)=\zeta(s-k),\qquad L_{g_1g_2}(s)=\zeta(s).$$

Thus Theorem 2.1 recovers the identity (1.1)

Similarly, if we let $f_1(n):=n^a$, $f_2(n):=n^b$ and $g_1(n)=g_2(n):=1$, then

$$L_{f_1f_2}(s)=L_{f_1f_2g_1g_2}(s)=\zeta(s-(a+b)),\qquad L_{g_1g_2}(s)=\zeta(s),$$

$$L_{f_1g_2}(s)=\zeta(s-a),\qquad L_{f_2g_1}(s)=\zeta(s-b).$$

and Theorem 2.1 gives (1.2).

In particular, for any real λ,

$$\sum_{n=1}^{\infty}\sigma_\lambda^2(n)n^{-s}=\frac{\zeta(s-2\lambda)\zeta(s-\lambda)^2\zeta(s)}{\zeta(2(s-\lambda))}. \tag{2.2}$$

We shall discuss more elaborate applications of Theorem 2.1 in the latter sections. Before doing this, we give the following example here to explain why Theorem 2.1 cannot in general be extended nicely to higher order.

We are interested in obtaining the generating functions for the *kth moment* of $r_2(n)$. For any $n\geq 1$ and $|x|<1$, in view of

$$\sum_{l=0}^{\infty}lx^l=x(1-x)^{-2}$$

and

$$x\frac{d}{dx}\sum_{l=0}^{\infty} l^n x^l = \sum_{l=0}^{\infty} l^{n+1}x^l \tag{2.3}$$

it is immediate that

$$\sum_{l=0}^{\infty} l^n x^l = \frac{xE_n(x)}{(1-x)^{n+1}}, \qquad n = 1, 2, \cdots \tag{2.4}$$

for a certain polynomial $E_n(x)$ of degree $n-1$. $E_n(x)$ is known as the nth *Euler polynomial* [4] and it is easy to see that (2.3) implies the recursion

$$E_{n+1}(x) = (1+nx)E_n(x) + x(1-x)E_n'(x).$$

Explicitly, the first few Euler polynomials are $E_1(x) = 1, E_2(x) = 1+x, E_3(x) = 1+4x+x^2$ and $E_4(x) = 1+11x+11x^2+x^3$. Equation (2.4) enables us to obtain the generating functions for the higher moments of $r_2(n)$ as follows: for $\mu \equiv 0$ or $1 \pmod 4$, we let $\left(\frac{\mu}{n}\right)$ be the *Jacobi-Legendre-Kronecker symbol* and again consider

$$L_\mu(s) := \sum_{n=1}^{\infty} \left(\frac{\mu}{n}\right) n^{-s}$$

the L-function corresponding to $\left(\frac{\mu}{n}\right)$. It is known (e.g. p. 291 in [3]) that

$$\sum_{n=1}^{\infty} \frac{r_2(n)}{n^s} = 4\zeta(s)L_{-4}(s) = \sum_{n=1}^{\infty} \frac{4(1 * \left(\frac{-4}{n}\right))(n)}{n^s}$$

and $r_2(n) = 4(1 * \left(\frac{-4}{n}\right))(n)$ for any $n \geq 1$. A simple calculation shows that for any $l \geq 0$,

$$\left(1 * \left(\frac{-4}{n}\right)\right)(p^l) = \begin{cases} 1 & \text{if } p = 2; \\ l+1 & \text{if } p \geq 3 \text{ and } \left(\frac{-1}{p}\right) = 1; \\ \frac{(-1)^l+1}{2} & \text{if } p \geq 3 \text{ and } \left(\frac{-1}{p}\right) = -1. \end{cases}$$

We now have

$$\begin{aligned}
\sum_{n=1}^{\infty} \frac{r_2^N(n)}{n^s} &= 4^N \sum_{n=1}^{\infty} \frac{\left\{(1 * \left(\frac{-4}{n}\right))(n)\right\}^N}{n^s} \\
&= 4^N \prod_p \sum_{l=0}^{\infty} \frac{\left\{(1 * \left(\frac{-4}{n}\right))(p^l)\right\}^N}{p^{ls}} \\
&= \frac{4^N}{1-2^{-s}} \left\{ \prod_{(\frac{-1}{p})=-1} \sum_{l=0}^{\infty} \left(\frac{(-1)^l+1}{2}\right)^N p^{-ls} \right\} \left\{ \prod_{(\frac{-1}{p})=1} \sum_{l=0}^{\infty} (l+1)^N p^{-ls} \right\} \\
&= \frac{4^N}{1-2^{-s}} \prod_{(\frac{-1}{p})=-1} \frac{1}{1-p^{-2s}} \prod_{(\frac{-1}{p})=1} \frac{E_N(p^{-s})}{(1-p^{-s})^{N+1}}
\end{aligned}$$

on using (2.4). [Here $\prod_p$ denotes the infinite product over all primes.] Firstly, when $N = 2$, we have most pleasingly,

$$\begin{aligned}\sum_{n=1}^{\infty}\frac{r_2^2(n)}{n^s} &= \frac{16}{1-2^{-s}}\prod_{(\frac{-1}{p})=-1}\frac{1}{1-p^{-2s}}\prod_{(\frac{-1}{p})=1}\frac{1+p^{-s}}{(1-p^{-s})^3}\\ &= \frac{16}{1+2^{-s}}\left\{\frac{1}{1-2^{-s}}\prod_{(\frac{-1}{p})=-1}\frac{1}{1-p^{-2s}}\prod_{(\frac{-1}{p})=1}\frac{1}{(1-p^{-s})^2}\right\}^2\prod_p(1-p^{-2s})\\ &= \frac{\left(4\zeta(s)L_{-4}(s)\right)^2}{(1+2^{-s})\zeta(2s)}. \qquad (2.5)\end{aligned}$$

However, when $N \geq 3$, the generating functions cannot be expressed in terms of L-functions as completely as in formula (2.5). For example, when $N = 3$

$$\sum_{n=1}^{\infty}\frac{r_2^3(n)}{n^s} = \frac{64}{1-2^{-s}}\prod_{(\frac{-1}{p})=-1}\frac{1}{1-p^{-2s}}\prod_{(\frac{-1}{p})=1}\frac{1+4p^{-s}+p^{-2s}}{(1-p^{-s})^4},$$

and when $N = 4$

$$\sum_{n=1}^{\infty}\frac{r_2^4(n)}{n^s} = \frac{256}{1-2^{-s}}\prod_{(\frac{-1}{p})=-1}\frac{1}{1-p^{-2s}}\prod_{(\frac{-1}{p})=1}\frac{1+11p^{-s}+11p^{-2s}+p^{-3s}}{(1-p^{-s})^5}.$$

This helps explain why our Theorem 2.1 has no 'closed-form' extension to higher order. For the detailed asymptotic estimate of the generating function of the kth moment of $r_2(n)$, we refer the reader to [5].

3. Sums of a Small Even Number of Squares

In view of Theorem 2.1, whenever a Dirichlet series is expressible as a sum of two-fold products of L-functions:

$$L_f(s) = \sum_{\chi_1,\chi_2} a(\chi_1,\chi_2)L_{\chi_1}(s)L_{\chi_2}(s),$$

we are able to provide a closed form (in terms of L-functions) of the Dirichlet series $L_{f^2}(s) = \sum_{n=1}^{\infty} f^2(n)n^{-s}$, on using (2.1).

In particular, let $r_N(n)$ be the number of solutions to $x_1^2 + x_2^2 + \cdots + x_N^2 = n$ (counting permutations and signs) and let

$$\mathcal{L}_N(s) := \sum_{n=1}^{\infty} r_N(n)n^{-s}, \qquad \mathcal{R}_N(s) := \sum_{n=1}^{\infty} r_N^2(n)n^{-s}$$

be the Dirichlet series corresponding to $r_N(n)$ and $r_N^2(n)$. Closed forms are obtainable for $\mathcal{L}_N(s)$ for certain even N from the explicit formulae known for $r_N(n)$. For example, we have

$$\mathcal{L}_2(s) = 4\zeta(s)L_{-4}(s), \qquad (3.1)$$

$$\mathcal{L}_4(s) = 8(1-4^{1-s})\zeta(s)\zeta(s-1), \qquad (3.2)$$

$$\mathcal{L}_6(s) = 16\zeta(s-2)L_{-4}(s) - 4\zeta(s)L_{-4}(s-2), \qquad (3.3)$$

$$\mathcal{L}_8(s) = 16(1-2^{1-s}+4^{2-s})\zeta(s)\zeta(s-3). \qquad (3.4)$$

The derivation of (3.1) and (3.3) from the formulas for $r_2(n)$ and $r_6(n)$ (e.g. §91 in [20]) is immediate if we write those formulas in the form

$$\begin{aligned} r_2(n) &= 4\sum_{\substack{m,d\geq 1\\ md=n}}\chi(d)\\ r_6(n) &= 16\sum_{\substack{m,d\geq 1\\ md=n}}\chi(m)d^2-4\sum_{\substack{m,d\geq 1\\ md=n}}\chi(d)d^2 \end{aligned}$$

where χ denotes the non-principal character modulo 4. For derivation of (3.2) and (3.4) from the formulas for $r_4(n)$ and $r_8(n)$ (e.g. §91 in [20]) is immediate if we write those formulas in the form

$$\begin{aligned} r_4(n) &= 8\sigma_1(n)-32\sigma_1(n/4)\\ r_8(n) &= 16\sigma_3(n)-32\sigma_3(n/2)+256\sigma_3(n/4) \end{aligned}$$

where it is understood that $\sigma_k(n)=0$ if n is not a positive integer.

In this section, we shall demonstrate how to use our Theorem 2.1 to obtain counterpart closed forms for $\mathcal{R}_N(s)$ from the above expressions for $\mathcal{L}_N(s)$.

Let us start with $\mathcal{R}_2(s)$. It has already been shown in (2.5) that

$$\mathcal{R}_2(s)=\sum_{n=1}^{\infty}\frac{r_2^2(n)}{n^s}=\frac{(4\zeta(s)L_{-4}(s))^2}{(1+2^{-s})\zeta(2s)}$$

but it can also be deduced directly from our Theorem 2.1 and (3.1) by taking $f_1(n)=f_2(n)=1$ and $g_1(n)=g_2(n)=\left(\frac{-4}{n}\right)$.

We shall consider $\mathcal{R}_4(s)$ and $\mathcal{R}_8(s)$ later. For $\mathcal{R}_6(s)$, we first write

$$\begin{aligned} \mathcal{L}_6(s) &= 16\zeta(s-2)L_{-4}(s)-4\zeta(s)L_{-4}(s-2)\\ &= 16\sum_{n=1}^{\infty}\left(\sum_{d|n}d^2\left(\frac{-4}{n/d}\right)\right)n^{-s}-4\sum_{n=1}^{\infty}\left(\sum_{d|n}d^2\left(\frac{-4}{d}\right)\right)n^{-s}\\ &= \sum_{n=1}^{\infty}\left(16(f_1*g_1)(n)-4(f_2*g_2)(n)\right)n^{-s} \end{aligned}$$

where $f_1(n)=n^2$, $g_1(n)=\left(\frac{-4}{n}\right)$, $f_2(n)=1$ and $g_2(n)=\left(\frac{-4}{n}\right)n^2$. It follows from our Theorem 2.1 and (3.3) that

$$\begin{aligned} \mathcal{R}_6(s) &= \sum_{n=1}^{\infty}\left(16(f_1*g_1)(n)-4(f_2*g_2)(n)\right)^2n^{-s}\\ &= 16^2\sum_{n=1}^{\infty}(f_1*g_1)^2(n)n^{-s}-128\sum_{n=1}^{\infty}(f_1*g_1)(n)(f_2*g_2)(n)n^{-s}\\ &\qquad +16\sum_{n=1}^{\infty}(f_2*g_2)^2(n)n^{-s}\\ &= 16^2\frac{L_{f_1^2}(s)L_{g_1^2}(s)L_{f_1g_1}(s)^2}{L_{f_1^2g_1^2}(2s)}-128\frac{L_{f_1f_2}(s)L_{g_1g_2}(s)L_{f_1g_2}(s)L_{g_1f_2}(s)}{L_{f_1f_2g_1g_2}(2s)}\\ &\qquad +16\frac{L_{f_2^2}(s)L_{g_2^2}(s)L_{f_2g_2}(s)^2}{L_{f_2^2g_2^2}(2s)}. \end{aligned}\tag{3.5}$$

It remains to evaluate the component L-functions and they are

$$L_{f_1^2}(s) = \zeta(s-4), \qquad L_{g_1^2}(s) = (1-2^{-s})\zeta(s),$$

$$L_{f_2^2}(s) = \zeta(s), \qquad L_{g_2^2}(s) = (1-16\cdot 2^{-s})\zeta(s-4),$$

$$L_{f_1g_1}(s) = L_{-4}(s-2), \qquad L_{f_1f_2}(s) = \zeta(s-2), \qquad L_{g_1g_2}(s) = (1-4\cdot 2^{-s})\zeta(s-2),$$

$$L_{f_1g_2}(s) = L_{-4}(s-4), \qquad L_{g_1f_2}(s) = L_{-4}(s), \qquad L_{f_2g_2}(s) = L_{-4}(s-2),$$

$$L_{f_1^2g_1^2}(s) = L_{f_2^2g_2^2}(s) = L_{f_1f_2g_1g_2}(s) = (1-16\cdot 2^{-s})\zeta(s-4).$$

Now from (3.5), we have

$$\mathcal{R}_6(s) = 16\frac{(17-32\cdot 2^{-s})}{(1-16\cdot 2^{-2s})}\frac{\zeta(s-4)L_{-4}^2(s-2)\zeta(s)}{\zeta(2s-4)} - \frac{128}{(1+4\cdot 2^{-s})}\frac{L_{-4}(s-4)\zeta^2(s-2)L_{-4}(s)}{\zeta(2s-4)}.$$

For $\mathcal{R}_4(s)$ and $\mathcal{R}_8(s)$, we need the following companion lemma:

Lemma 3.1. *Suppose $f(n)$ is a multiplicative function. Let p be a prime and let the Dirichlet series*

$$\sum_{n=1}^{\infty}\frac{A(n)}{n^s} := \sum_{m=0}^{\infty}\frac{a_m}{p^{ms}}\sum_{n=1}^{\infty}\frac{f(n)}{n^s}$$

be the product of $L_f(s)$ and a power series in p^{-s}. Then

$$\sum_{n=1}^{\infty}\frac{A^2(n)}{n^s} = L_{f^2}(s)\sum_{m=0}^{\infty}\frac{a_m^2}{p^{ms}} + 2L_{f^2}(s)\left(\sum_{l=0}^{\infty}\frac{f^2(p^l)}{p^{ls}}\right)^{-1} \times \sum_{k=1}^{\infty}\left\{\sum_{m=0}^{\infty}\frac{a_{m+k}a_m}{p^{ms}}\right\}\left\{\sum_{l=0}^{\infty}\frac{f(p^l)f(p^{l+k})}{p^{ls}}\right\}p^{-ks}. \tag{3.6}$$

Proof. Since

$$\sum_{n=1}^{\infty}A(n)n^{-s} = \sum_{n=1}^{\infty}\sum_{m=0}^{\infty}a_mf(n)(p^mn)^{-s} = \sum_{n=1}^{\infty}\left\{\sum_{\substack{m=0\\ p^m|n}}^{\infty}a_mf\left(\frac{n}{p^m}\right)\right\}n^{-s},$$

we deduce

$$\begin{aligned}\sum_{n=1}^{\infty}A^2(n)n^{-s} &= \sum_{n=1}^{\infty}\left\{\sum_{\substack{m=0\\ p^m|n}}^{\infty}a_mf\left(\frac{n}{p^m}\right)\right\}^2 n^{-s}\\ &= \sum_{m_1,m_2=0}^{\infty}a_{m_1}a_{m_2}\sum_{\substack{n=1\\ p^{m_1},p^{m_2}|n}}^{\infty}f\left(\frac{n}{p^{m_1}}\right)f\left(\frac{n}{p^{m_2}}\right)n^{-s}. \qquad (3.7)\end{aligned}$$

For any $m_1, m_2 \geq 1$ we let $M := \max(m_1, m_2)$ and $m := \min(m_1, m_2)$. Then the last summation (over n) in (3.7) is

$$\begin{aligned}
&= \sum_{\substack{n=1 \\ p^M | n}}^{\infty} f\left(\frac{n}{p^M}\right) f\left(\frac{n}{p^m}\right) n^{-s} \\
&= \frac{1}{p^{Ms}} \sum_{n=1}^{\infty} f(n) f(np^{M-m}) n^{-s} \\
&= \frac{1}{p^{Ms}} \sum_{l=0}^{\infty} \sum_{\substack{n=1 \\ (p,n)=1}}^{\infty} f(np^l) f(np^{M-m+l}) p^{-ls} n^{-s} \\
&= \frac{1}{p^{Ms}} \sum_{l=0}^{\infty} f(p^l) f(p^{M-m+l}) p^{-ls} \sum_{\substack{n=1 \\ (p,n)=1}}^{\infty} \frac{f^2(n)}{n^s} \qquad (3.8)
\end{aligned}$$

since $f(n)$ is multiplicative. By writing

$$\sum_{n=1}^{\infty} \frac{f^2(n)}{n^s} = \sum_{l=0}^{\infty} \sum_{\substack{n=1 \\ (p,n)=1}}^{\infty} \frac{f^2(np^l)}{(np^l)^s} = \sum_{l=0}^{\infty} \frac{f^2(p^l)}{p^{ls}} \sum_{\substack{n=1 \\ (p,n)=1}}^{\infty} \frac{f^2(n)}{n^s},$$

we deduce that

$$\sum_{\substack{n=1 \\ (p,n)=1}}^{\infty} \frac{f^2(n)}{n^s} = L_{f^2}(s) \left(\sum_{l=0}^{\infty} f^2(p^l) p^{-ls} \right)^{-1}. \qquad (3.9)$$

Using (3.7), (3.8) and (3.9), we have

$$\begin{aligned}
\sum_{n=1}^{\infty} A^2(n) n^{-s} = L_{f^2}(s) \left(\sum_{l=0}^{\infty} f^2(p^l) p^{-ls} \right)^{-1} \times \\
\times \sum_{m_1, m_2 = 0}^{\infty} \frac{a_{m_1} a_{m_2}}{p^{\max(m_1, m_2)s}} \sum_{l=0}^{\infty} \frac{f(p^l) f(p^{l+|m_1 - m_2|})}{p^{ls}}. \qquad (3.10)
\end{aligned}$$

The contribution corresponding to $m_1 = m_2$ in the above double summation is

$$\sum_{m=0}^{\infty} \frac{{a_m}^2}{p^{ms}} \sum_{l=0}^{\infty} \frac{f^2(p^l)}{p^{ls}} \qquad (3.11)$$

and the contribution corresponding to $m_1 \neq m_2$ is

$$\begin{aligned}
&= 2 \sum_{m_2 < m_1}^{\infty} \frac{a_{m_1} a_{m_2}}{p^{m_1 s}} \sum_{l=0}^{\infty} \frac{f(p^l) f(p^{l+m_1-m_2})}{p^{ls}} \\
&= 2 \sum_{m=0}^{\infty} \sum_{k=1}^{\infty} \frac{a_{m+k} a_m}{p^{(m+k)s}} \sum_{l=0}^{\infty} \frac{f(p^l) f(p^{l+k})}{p^{ls}} \\
&= 2 \sum_{k=1}^{\infty} \left\{ \sum_{m=0}^{\infty} \frac{a_{m+k} a_m}{p^{ms}} \right\} \left\{ \sum_{l=0}^{\infty} \frac{f(p^l) f(p^{l+k})}{p^{ls}} \right\} \frac{1}{p^{ks}}. \qquad (3.12)
\end{aligned}$$

Now (3.6) follows from (3.10), (3.11) and (3.12). □

On applying Lemma 3.1 to (3.2) and (3.4) and using (2.2), we have

$$\mathcal{R}_4(s) = 64\frac{(8\cdot 2^{3-3s} - 10\cdot 2^{2-2s} + 2^{1-s} + 1)\zeta(s-2)\zeta^2(s-1)\zeta(s)}{(1+2^{1-s})\zeta(2s-2)};$$

and

$$\mathcal{R}_8(s) = 256\frac{(32\cdot 2^{6-2s} - 3\cdot 2^{3-s} + 1)\zeta(s-6)\zeta^2(s-3)\zeta(s)}{(1+2^{3-s})\zeta(2s-6)}.$$

Therefore, we have completed the proof of the following Theorem.

Theorem 3.2. *We may write*

$$\mathcal{R}_2(s) = \frac{(4\zeta(s)L_{-4}(s))^2}{(1+2^{-s})\zeta(2s)}, \qquad \Re(s) > 1;$$

$$\mathcal{R}_4(s) = 64\frac{(8\cdot 2^{3-3s} - 10\cdot 2^{2-2s} + 2^{1-s} + 1)\zeta(s-2)\zeta^2(s-1)\zeta(s)}{(1+2^{1-s})\zeta(2s-2)}, \qquad \Re(s) > 3;$$

$$\mathcal{R}_6(s) = 16\frac{(17-32\cdot 2^{-s})}{(1-16\cdot 2^{-2s})}\frac{\zeta(s-4)L_{-4}^2(s-2)\zeta(s)}{\zeta(2s-4)} - \frac{128}{(1+4\cdot 2^{-s})}\frac{L_{-4}(s-4)\zeta^2(s-2)L_{-4}(s)}{\zeta(2s-4)}, \qquad \Re(s) > 5;$$

and

$$\mathcal{R}_8(s) = 256\frac{(32\cdot 2^{6-2s} - 3\cdot 2^{3-s} + 1)\zeta(s-6)\zeta^2(s-3)\zeta(s)}{(1+2^{3-s})\zeta(2s-6)} \qquad \Re(s) > 7.$$

Since $\epsilon\zeta(1+\epsilon) \to 1$ as $\epsilon \to 0$, the value of the $\lim_{\epsilon\to 0}\epsilon\mathcal{R}_N(N-1+\epsilon)$ at its largest pole is, respectively:

$$\lim_{\epsilon\to 0}\epsilon\mathcal{R}_4(3+\epsilon) = 96\zeta(3) = 3W_4$$

$$\lim_{\epsilon\to 0}\epsilon\mathcal{R}_6(5+\epsilon) = 240\zeta(5) = 5W_6$$

and

$$\lim_{\epsilon\to 0}\epsilon\mathcal{R}_8(7+\epsilon) = \frac{4480}{17}\zeta(7) = 7W_8.$$

The formulae for $\mathcal{R}_N(s)$ in Theorem 3.2 enable us to estimate the average order of $r_N^2(n)$ for $N = 2, 4, 6, 8$. Following from Sierpinski's result on the circle problem (cf. Satz 509 of [17])

$$\sum_{n\le x} r_2(n) = \pi x + O(x^{1/3}), \tag{3.13}$$

we have

$$\sum_{n\le x} r_2^2(n) = 4x\log x + 4\alpha x + O(x^{2/3}) \tag{3.14}$$

where $\alpha := 2\gamma + \frac{8}{\pi}L'_{-4}(1) - \frac{12}{\pi^2}\zeta'(2) + \frac{1}{3}\log 2 - 1 = 2.0166216\cdots$. Indeed, one can prove (3.14) as follows. Let

$$\sum_{n=1}^{\infty} h_n n^{-s} := \{4\zeta(s)L_{-4}(s)\}^2 = \left(\sum_{n=1}^{\infty} r_2(n)n^{-s}\right)^2. \tag{3.15}$$

By the hyperbola method and (3.13), one has

$$\begin{aligned} H(x) &:= \sum_{n\le x} h_n = \sum_{\substack{m,d\ge 1\\ md\le x}} r_2(m)r_2(d) \\ &= 2\sum_{m\le\sqrt{x}} r_2(m)\sum_{n\le x/m} r_2(n) - \left(\sum_{n\le\sqrt{x}} r_2(n)\right)^2 \\ &= 2\sum_{m\le\sqrt{x}} r_2(m)\left\{\pi\frac{x}{m} + O\left(\frac{x^{1/3}}{m^{1/3}}\right)\right\} - \{\pi x^{1/2} + O(x^{1/6})\}^2 \\ &= \pi^2 x\log x + C_1 x + O(x^{2/3}), \end{aligned}$$

for some constant C_1. Now by (2.5) we have

$$\mathcal{R}_2(s) = \sum_{n=1}^{\infty} r_2^2(n)n^{-s} = \sum_{m=1}^{\infty} h_m m^{-s}\sum_{n=1}^{\infty} l_n n^{-s}$$

where h_n is given (3.15) and

$$\sum_{n=1}^{\infty} l_n n^{-s} = (1+2^{-s})^{-1}\zeta^{-1}(2s) = \sum_{j=0}^{\infty}(-1)^j 2^{-js}\sum_{k=1}^{\infty}\mu(k)k^{-2s}$$

has abscissa of absolute convergence $1/2$ and

$$\sum_{n\le x}|l_n| = O(x^{1/2}\log x).$$

Here $\mu(n)$ is the Möbius function. Now by an elementary convolution argument

$$\begin{aligned} \sum_{n\le x} r_2^2(n) &= \sum_{n\le x} l_n H(x/n) \\ &= \sum_{n\le x} l_n\{\pi^2\frac{x}{n}\log\frac{x}{n} + C_1\frac{x}{n} + O(\frac{x^{2/3}}{n^{2/3}})\} \\ &= 4x\log x + C_2 x + O(x^{2/3}) \end{aligned} \tag{3.16}$$

for some constant C_2. To evaluate the value of C_2, we first note that for any $\sigma > 1$, we have

$$\sum_{n\le x}\frac{r_2^2(n)}{n^\sigma} = \int_{1^-}^{x} u^{-\sigma}d\sum_{n\le u} r_2^2(n)$$

and hence from (3.16) and letting $x\longrightarrow+\infty$, we get

$$\mathcal{R}_2(\sigma) = \sigma\int_1^\infty\left(\frac{\sum_{n\le u} r_2^2(n) - 4u\log u - C_2 u}{u^{\sigma+1}}\right)du + \frac{4}{(\sigma-1)^2} + \frac{4+\sigma C_2}{\sigma-1}.$$

The above integral converges when $\sigma\longrightarrow 1^+$ and hence

$$\lim_{\sigma\longrightarrow 1^+}\left\{\mathcal{R}_2(\sigma) - \frac{4}{(\sigma-1)^2}\right\}(\sigma-1) = 4 + C_2. \tag{3.17}$$

Now in view of (2.5), $\mathcal{R}_2(s)$ has a pole at $s=1$ of order 2. So the limit in (3.17) in fact is the residue of $\mathcal{R}_2(s)$ at $s=1$ which can be evaluated by the method in §5 below and it is equal to

$$4\left(2\gamma+\frac{8}{\pi}L'_{-4}(1)-\frac{12}{\pi^2}\zeta'(2)+\frac{1}{3}\log 2\right).$$

This completes the proof of (3.14)

It is also worth to note that Sierpinski's result has been slightly improved and so the error term in (3.14) could be improved accordingly. For example, the term $O(x^{2/3})$ can be replaced by $O(x^{284/429})$ if we employ Nowak's result in [19] which replaces the term $O(x^{1/3})$ in (3.13) by $O(x^{139/429})$.

We now consider the case $N=4$. In view of Theorem 3.2, $\mathcal{R}_4(s)/\zeta(s-2)$ is equal to the product of a finite Dirichlet series and the five Dirichlet series $\zeta(s-1),\zeta(s-1),\zeta(s),\zeta^{-1}(2s-2)$ and $(1+2^{1-s})^{-1}$, each of which has the property that the coefficient of n^{-s} is $O(n)$. Hence from the formula for $\mathcal{R}_4(s)$ in Theorem 3.2,

$$\mathcal{R}_4(s)=\zeta(s-2)\sum_{n=1}^{\infty}g_n n^{-s},$$

where $|g_n|=O(nd_5(n))$ and $d_k(n)$ is the number of ways of expressing n in the form $n=n_1n_2\cdots n_k$ with $n_1,n_2,\cdots,n_k$ positive integers. It follows that

$$\begin{aligned}
\sum_{n\le x}r_4^2(n) &= \sum_{n\le x}g_n\sum_{m\le x/n}m^2\\
&= \sum_{n\le x}g_n\left(\frac{1}{3}\left(\frac{x}{n}\right)^3+O\left(\frac{x^2}{n^2}\right)\right)\\
&= \frac{x^3}{3}\sum_{n=1}^{\infty}\frac{g_n}{n^3}+O\left(x^3\left|\sum_{n>x}\frac{g_n}{n^3}\right|\right)+O\left(x^2\sum_{n\le x}\frac{|g_n|}{n^2}\right)\\
&= \frac{x^3}{3}\sum_{n=1}^{\infty}\frac{g_n}{n^3}+O\left(x^3\sum_{n>x}\frac{d_5(n)}{n^2}\right)+O\left(x^2\sum_{n\le x}\frac{d_5(n)}{n}\right)\\
&= \frac{x^3}{3}\sum_{n=1}^{\infty}\frac{g_n}{n^3}+O(x^2\log^5 x)
\end{aligned}$$

because $\sum_{n\le x}d_k(n)\sim xP_k(\log x)$ for some polynomial $P_k(X)$ of degree $k-1$ (see Chapter XII in [26]). Now since

$$\sum_{n=1}^{\infty}\frac{g_n}{n^3}=\lim_{s\longrightarrow 3^+}\mathcal{R}_4(s)/\zeta(s-2)=\lim_{\epsilon\longrightarrow 0}\epsilon\mathcal{R}_4(3+\epsilon)=3W_4$$

so we have

$$\sum_{n\le x}r_4^2(n)=W_4x^3+O(x^2\log^5 x).$$

The cases for $N=6$ and $N=8$ can be treated in the same manner as

$$\mathcal{R}_6(s)=\zeta(s-4)\sum_{n=1}^{\infty}\frac{b_n}{n^s}+L_{-4}(s-4)\sum_{n=1}^{\infty}\frac{c_n}{n^s}$$

and

$$\mathcal{R}_8(s) = \zeta(s-6)\sum_{n=1}^{\infty}\frac{d_n}{n^s}$$

where b_n and c_n are $\ll n^2 d_5(n)$ and d_n is $\ll n^3 d_5(n)$. Therefore, we have

$$\sum_{n\le x} r_N^2(n) = W_N x^{N-1} + O(x^{N-2}) \tag{3.18}$$

for $N = 6, 8$ with W_N given by (1.3).

For $N \neq 2, 4, 6, 8$, lacking the closed forms for $\mathcal{R}_N(s)$, we can't follow the argument above to estimate the average order for $r_N^2(n)$. However, as suggested by the referee, the asymptotic value for $\sum_{n\le x} r_N^2(n)$, at least for $N \ge 5$, can be obtained from the singular series formula for $r_N(n)$ given by Hardy (see p.342 of [12] or p.155 of [11]), which may be written as

$$r_N(n)\frac{\Gamma(N/2)}{\pi^{N/2}}n^{1-N/2} = \sum_{k=1}^{\infty}\sum_{\substack{1\le h\le k\\(h,k)=1}}\left(\frac{G(h,k)}{k}\right)^N e^{-2\pi i h n/k} + O(n^{1-N/4}) \tag{3.19}$$

where $G(h,k) = \sum_{j=1}^{k} e^{2\pi i h j^2/k}$ is the standard quadratic Gauss sum. In fact, using a well-known result on quadratic Gauss sum (e.g. p.138 of [11])

$$|G(h,k)| = \begin{cases} \sqrt{k} & \text{if } k \equiv 1 \pmod 2;\\ 0 & \text{if } k \equiv 2 \pmod 4;\\ \sqrt{2k} & \text{if } k \equiv 0 \pmod 4; \end{cases} \tag{3.20}$$

for $(h,k) = 1$, we have

$$\begin{aligned} r_N(n)\frac{\Gamma(N/2)}{\pi^{N/2}}n^{1-N/2} &= \sum_{k\le x^{1/2}}\sum_{\substack{1\le h\le k\\(h,k)=1}}\left(\frac{G(h,k)}{k}\right)^N e^{-2\pi i h n/k} + O(x^{1-N/4})\\ &:= P(n) + O(x^{1-N/4}) \end{aligned}$$

for $N \ge 5$ and $n \le x$. By (3.20), we have $|P(n)| \ll 1$ and hence

$$r_N(n)^2 = \frac{\pi^N}{\Gamma(N/2)^2}n^{N-2}|P(n)|^2 + O(x^{3N/4-1}).$$

It follows that

$$\sum_{n\le x} r_N(n)^2 = \frac{\pi^N}{\Gamma(N/2)^2}\sum_{n\le x} n^{N-2}|P(n)|^2 + O(x^{3N/4}). \tag{3.21}$$

It remains to estimate the sum $\sum_{n\le x} n^{N-2}|P(n)|^2$ which is equal to

$$\sum_{1\le k_1,k_2\le x^{1/2}}\sum_{\substack{1\le h_i\le k_i\\(h_i,k_i)=1,i=1,2}}\left(\frac{G(h_1,k_1)}{k_1}\right)^N\left(\frac{\overline{G(h_2,k_2)}}{k_2}\right)^N\sum_{n\le x} n^{N-2}e^{-2\pi i n(\frac{h_1}{k_1}-\frac{h_2}{k_2})}. \tag{3.22}$$

We now note that when $\frac{h_1}{k_1} \neq \frac{h_2}{k_2}$, we have

$$\left| \sum_{n \leq x} e^{-2\pi i n(\frac{h_1}{k_1} - \frac{h_2}{k_2})} \right| \leq k_1 k_2$$

and hence the contribution for those terms $\frac{h_1}{k_1} \neq \frac{h_2}{k_2}$ to (3.22) is

$$\ll x^{N-2} \left(\sum_{k \leq x^{1/2}} k^{2-N/2} \right)^2.$$

Using this, (3.22) and (3.21), we have

$$\begin{aligned} \sum_{n \leq x} r_N(n)^2 &= \frac{\pi^N}{(N-1)\Gamma(N/2)^2} \left(\sum_{k \leq x^{1/2}} B(k) \right) x^{N-1} + O(x^{N-2} + x^{3N/4}) \\ &= \frac{\pi^N}{(N-1)\Gamma(N/2)^2} \left(\sum_{k=1}^{\infty} B(k) \right) x^{N-1} + O(x^{N-2} + x^{3N/4}) \end{aligned}$$

where

$$B(k) := \sum_{\substack{1 \leq h \leq k \\ (h,k)=1}} \left| \frac{G(h,k)}{k} \right|^{2N}.$$

Note that when $N = 6$, we have a better error term in (3.18). The function $k \to B(k)$ is multiplicative in k (see p.156 of [11]) and from (3.20), $B(1) = 1, B(2) = 0, B(2^l) = 2^{-N(l-1)}\phi(2^l)$ for any $l \geq 2$ and $B(p^j) = p^{-Nj}\phi(p^j)$ for any $j \geq 1$ and odd prime p. It then follows from the Euler product formula that

$$\sum_{k=1}^{\infty} B(k) = (1 - 2^{-(N-1)})^{-1} \prod_{p>3} \frac{1 - p^{-N}}{1 - p^{-(N-1)}} = \frac{1}{(1 - 2^{-N})} \frac{\zeta(N-1)}{\zeta(N)}.$$

We finally conclude that

Theorem 3.3. *We have*

$$\sum_{n \leq x} r_2^2(n) = 4x \log x + 4\alpha x + O(x^{2/3})$$

$$\sum_{n \leq x} r_4^2(n) = W_4 x^3 + O(x^2 \log^5 x)$$

and

$$\sum_{n \leq x} r_6^2(n) = W_6 x^5 + O(x^4)$$

For $N \geq 5, N \neq 6$ and $x \geq 1$, we have

$$\sum_{n \leq x} r_N^2(n) = W_N x^{N-1} + O(x^{N-2} + x^{3N/4}).$$

Here $\alpha = 2\gamma + \frac{8}{\pi} L'_{-4}(1) - \frac{12}{\pi^2}\zeta'(2) + \frac{1}{3}\log 2 - 1 = 2.0166216\cdots$.

This proves Wagon's conjecture for $N \geq 4$. Theorem 3.3 can also be found in [8] and it contains the same basic arguments for getting the error bounds on $r_N^2(n)$ summatory for $N \geq 5$. The estimate $O(x^{N-2})$ in fact is the best possible as will be discussed elsewhere.

4. Closed Forms for Dirichlet Series of Quadratic Forms

There is a rich parallel theory of L-functions over imaginary quadratic fields. In this vein, let $r_{2,P}(n)$ be the number of solutions to $x^2 + Py^2 = n$ (again counting sign and order). Denote

$$\mathcal{L}_{2,P}(s) := \sum_{n=1}^{\infty} r_{2,P}(n) n^{-s}, \qquad \mathcal{R}_{2,P}(s) := \sum_{n=1}^{\infty} r_{2,P}(n)^2 n^{-s}.$$

It is known that when the quadratic form $x^2 + Py^2$ has disjoint discriminants (that is, it has exactly one form per genus), then one has the following formula (see (9.2.8) in [3])

$$\begin{aligned} \mathcal{L}_{2,P} &= 2^{1-t} \sum_{\mu | P} L_{\epsilon_\mu \mu}(s) L_{-4P\epsilon_\mu/\mu}(s) \\ &= \sum_{n=1}^{\infty} \left\{ 2^{1-t} \sum_{\mu | P} \left(\frac{\epsilon_\mu \mu}{n}\right) * \left(\frac{-4P\epsilon_\mu/\mu}{n}\right) \right\} n^{-s} \end{aligned} \tag{4.1}$$

where P is an odd square-free number, t is the number of distinct factors of P and $\epsilon_\mu := \left(\frac{-1}{\mu}\right)$.

Explicitly, (4.1) holds for all *type one* numbers. These include and may comprise:

$$P = 5, 13, 21, 33, 37, 57, 85, 93, 105, 133, 165, 177, 253, 273, 345, 357, 385, 1365.$$

It is known that there are only finitely many such disjoint discriminants. We call such P **solvable**. Using (4.1), we have

$$\begin{aligned} \mathcal{R}_{2,P}(s) &= \sum_{n=1}^{\infty} 2^{2-2t} \sum_{\mu_1 \mu_2 | P} \left[\left(\frac{\epsilon_{\mu_1}\mu_1}{n}\right) * \left(\frac{-4P\epsilon_{\mu_1}/\mu_1}{n}\right) \right] \cdot \left[\left(\frac{\epsilon_{\mu_2}\mu_2}{n}\right) * \left(\frac{-4P\epsilon_{\mu_2}/\mu_2}{n}\right) \right] n^{-s} \\ &= 2^{2-2t} \sum_{\mu_1 \mu_2 | P} \sum_{n=1}^{\infty} \left[\left(\frac{\epsilon_{\mu_1}\mu_1}{n}\right) * \left(\frac{-4P\epsilon_{\mu_1}/\mu_1}{n}\right) \right] \cdot \left[\left(\frac{\epsilon_{\mu_2}\mu_2}{n}\right) * \left(\frac{-4P\epsilon_{\mu_2}/\mu_2}{n}\right) \right] n^{-s}. \end{aligned}$$

We now notice that $\mathcal{R}_{2,P}(s)$ is a sum of Dirichlet series in the form of Theorem 2.1. We may apply Theorem 2.1 on letting

$$f_i(n) := \left(\frac{\epsilon_{\mu_i}\mu_i}{n}\right), \qquad g_i(n) := \left(\frac{-4P\epsilon_{\mu_i}/\mu_i}{n}\right),$$

for $i = 1, 2$. Then

$$\begin{aligned} L_{f_1 f_2}(s) &= \sum_{n=1}^{\infty} \left(\frac{\epsilon_{\mu_1}\mu_1}{n}\right)\left(\frac{\epsilon_{\mu_2}\mu_2}{n}\right) n^{-s} \\ &= \sum_{\substack{n=1 \\ (n,(\mu_1,\mu_2))=1}}^{\infty} \left(\frac{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}{n}\right) n^{-s} \\ &= L_{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}(s) \prod_{p|(\mu_1,\mu_2)} \left(1 - \left(\frac{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}{p}\right) p^{-s}\right) \end{aligned}$$

where $\mu_i^* := \mu_i/(\mu_1, \mu_2)$ and $\prod_{p|n}$ denotes the product over all prime factors of n. Similarly, we have

$$L_{g_1 g_2}(s) = L_{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}(s) \prod_{p|\frac{2P}{(\mu_1,\mu_2)}} \left(1 - \left(\frac{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}{p}\right) p^{-s}\right);$$

$$L_{f_1 g_2}(s) = L_{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}(s) \prod_{p|\mu_1^*} \left(1 - \left(\frac{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}{p}\right) p^{-s}\right);$$

$$L_{f_2 g_1}(s) = L_{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}(s) \prod_{p|\mu_2^*} \left(1 - \left(\frac{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}{p}\right) p^{-s}\right)$$

and

$$L_{f_1 f_2 g_1 g_2}(s) = \zeta(s) \prod_{p|2P} (1 - p^{-s}).$$

Our basic Theorem 2.1 gives

$$\begin{aligned} \mathcal{R}_{2,P}(s) = 2^{2(1-t)} &\sum_{\mu_1,\mu_2|P} L^2_{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}(s) L^2_{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}(s) \zeta(2s)^{-1} \\ &\times \prod_{p|2P} \left\{1 + \left[\left(\frac{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}{p}\right) + \left(\frac{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}{p}\right)\right] p^{-s}\right\}^{-1}. \end{aligned}$$

We have similar closed forms of L-functions for the quadratic form $x^2 + 2Py^2$ with discriminant $-8P$ (see (9.2.9) in [3]):

$$\mathcal{L}_{2,2P} = 2^{1-t} \sum_{\mu|P} L_{\epsilon_\mu \mu}(s) L_{-8P\epsilon_\mu/\mu}(s).$$

We deduce from Theorem 2.1, in the same way, that

$$\begin{aligned} \mathcal{R}_{2,2P}(s) = 2^{2(1-t)} &\sum_{\mu_1,\mu_2|P} L^2_{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}(s) L^2_{-8P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}(s) \zeta(2s)^{-1} \\ &\times \prod_{p|2P} \left\{1 + \left[\left(\frac{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}{p}\right) + \left(\frac{-8P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}{p}\right)\right] p^{-s}\right\}^{-1} \end{aligned}$$

for the *type two* integers

$$P = 1, 3, 5, 11, 15, 21, 29, 35, 39, 51, 65, 95, 105, 165, 231.$$

We note that $210 = 2 \times 105$ yields the invariant which Ramanujan sent to Hardy in his famous letter.

We may reprise with the following theorem:

Theorem 4.1. *Let P be a solvable square-free integer and let t be the number of distinct factors of P. We have for P respectively of type one and type two:*

$$\mathcal{R}_{2,P}(s) = 2^{2(1-t)} \sum_{\mu_1,\mu_2|P} L^2_{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}(s) L^2_{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}(s)\zeta(2s)^{-1} \times \prod_{p|2P}\left\{1+\left[\left(\frac{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}{p}\right)+\left(\frac{-4P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}{p}\right)\right]p^{-s}\right\}^{-1}, \tag{4.2}$$

and

$$\mathcal{R}_{2,2P}(s) = 2^{2(1-t)} \sum_{\mu_1,\mu_2|P} L^2_{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}(s) L^2_{-8P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}(s)\zeta(2s)^{-1} \times \prod_{p|2P}\left\{1+\left[\left(\frac{\epsilon_{\mu_1^*\mu_2^*}\mu_1^*\mu_2^*}{p}\right)+\left(\frac{-8P\epsilon_{\mu_1^*\mu_2^*}/\mu_1^*\mu_2^*}{p}\right)\right]p^{-s}\right\}^{-1}$$

where $\epsilon_\mu = \left(\frac{-1}{\mu}\right)$ and $\mu_i^* = \mu_i/(\mu_1,\mu_2)$.

In particular, the prime cases provide:

Corollary 4.2. *We have*

$$\mathcal{R}_{2,p}(s) = \frac{2\zeta^2(s)L^2_{-4p}(s)}{(1+2^{-s})(1+p^{-s})\zeta(2s)} + \frac{2L^2_p(s)L^2_{-4}(s)}{(1-2^{-s})(1+p^{-s})\zeta(2s)}$$

for $p = 5, 13, 37$, while

$$\mathcal{R}_{2,2}(s) = \frac{4\zeta^2(s)L^2_{-8}(s)}{(1+2^{-s})\zeta(2s)}.$$

Similarly,

$$\mathcal{R}_{2,2p}(s) = \frac{2\zeta^2(s)L^2_{-8p}(s)}{(1+2^{-s})(1+p^{-s})\zeta(2s)} + \frac{2L^2_{-p}(s)L^2_8(s)}{(1-2^{-s})(1-p^{-s})\zeta(2s)},$$

for $p = 3, 11$ while

$$\mathcal{R}_{2,2p}(s) = \frac{2\zeta^2(s)L^2_{-8p}(s)}{(1+2^{-s})(1+p^{-s})\zeta(2s)} + \frac{2L^2_p(s)L^2_{-8}(s)}{(1-2^{-s})(1-p^{-s})\zeta(2s)}$$

for $p = 5, 29$.

Closed forms for $\mathcal{L}_{2,P}(s)$ are also accessible for some P other than those of *type one* or *type two*. For example, (see Table VI of [10]) one has

$$\mathcal{L}_{2,3}(s) = (2 + 4^{1-s})\,\zeta(s)L_{-3}(s). \tag{4.3}$$

and hence by Theorem 2.1 and Lemma 3.1, we obtain

$$\mathcal{R}_{2,3}(s) = 4\frac{1+2^{3-2s}}{1+3^{-s}}\frac{(\zeta(s)\,L_{-3}(s))^2}{\zeta(2s)}. \tag{4.4}$$

We may also derive many formulae for non-square free integers via modular transformations [3]. We contain ourselves with the simplest example which is

$$\mathcal{R}_{2,4}(s) = \frac{4 - 2^{2-s} + 2^{4-2s}}{1 + 2^{-s}} \frac{(\zeta(s)\, L_{-4}(s))^2}{\zeta(2s)}$$

as a consequence of a quadratic transformation leading to

$$\mathcal{L}_{2,4}(s) = (2^{-1} - 2^{-1-s} + 4^{-s})\mathcal{L}_2(s).$$

There are some simple closed forms of the generating functions for more general binary quadratic forms found in [10]. Let

$$\mathcal{L}_{(a,b,c)}(s) := \sum_{(n,m)\neq(0,0)} \frac{1}{(am^2 + bmn + cn^2)^s} = \sum_{n=1}^{\infty} \frac{r_{(a,b,c)}(n)}{n^s}$$

and $\mathcal{R}_{(a,b,c)}(s) := \sum_{n=1}^{\infty} \frac{r_{(a,b,c)}(n)^2}{n^s}$ where $r_{(a,b,c)}(n)$ is the number of representations of n by the quadratic form $ax^2 + bxy + cy^2$. Then, we have (e.g. (26) of [25])

$$\sum_{h(D)} \mathcal{L}_{(a,b,c)}(s) = \omega(D)\zeta(s)L_D(s)$$

where the sum is taken over the $h(D)$ inequivalent reduced quadratic forms of discriminant $D := b^2 - 4ac$ and $\omega(-3) = 6, \omega(-4) = 4$ and $\omega(D) = 2$ for $D < -4$. In particular, for $c = 2, 3, 5, 11, 17, 41$, $h(D) = 1$ and the result is especially simple:

$$\mathcal{L}_{(1,1,c)}(s) = 2\zeta(s)L_D(s).$$

Hence from Theorem 2.1, we have

$$\mathcal{R}_{(1,1,c)}(s) = \frac{4(\zeta(s)L_D(s))^2}{(1 + |D|^{-s})\zeta(2s)},$$

with similar formulae for $(a, b, c) = (1, 1, 1)$ and $(1, 0, 1)$.

Thanks to the *On-Line Encyclopedia of Integer Sequences*

`http://www.research.att.com/~njas/sequences/`

we discover that the sequence $2, 3, 5, 11, 17, 41$ is exactly the so-called Euler "lucky" numbers which are the numbers n such that $m \to m^2 - m + n$ has prime values for $m = 0, \cdots, n - 1$.

5. The Average Order of $r_{2,P}(n)$

We start with the average order of $r_{2,P}$. The results in this section, in fact, can be obtained by a convolution argument such as we used to prove (3.18) in §3. This, however, does not seem to yield better error estimates, especially in the power of N, in Theorem 5.1 and 5.3 below. So we instead apply Perron's formula. Both methods would seem to add an unnecessary if unobtrusive 'ε'.

Theorem 5.1. *Let P be a solvable square-free integer, $x > 1$ and $\epsilon > 0$. We have for either $N = P$ of type one or $N = 2P$ of type two:*

$$\sum_{n\leq x} r_{2,N}(n) = \frac{\pi}{\sqrt{N}}x + O((xN)^{\frac{1}{2}+\epsilon}).$$

where the implicit constants are independent of x and P.

Proof. In view of (4.1), we have for $n \geq 1$

$$r_{2,P}(n) = 2^{1-t} \sum_{\mu|P} \left(\frac{\epsilon_\mu \mu}{n}\right) * \left(\frac{-4P\epsilon_\mu/\mu}{n}\right) \leq 2^{1-t} \sum_{\mu|P} \sigma_0(n) \leq 2\sigma_0(n). \tag{5.1}$$

It follows from (1.1) that

$$\mathcal{L}_{2,P}(\sigma) \ll \sum_{n=1}^{\infty} \frac{\sigma_0(n)}{n^\sigma} = \zeta(\sigma)^2 \ll \frac{1}{(\sigma-1)^2}$$

as $\sigma \to 1^+$. Now in view of Perron's formula (see Theorem 1 in §1 of Chapter V in [16]), for any $c > 1$, $\epsilon > 0$ and $x, T \geq 1$ we have

$$\sum_{n \leq x} r_{2,P}(n) = \frac{1}{2\pi i} \int_{c-iT}^{c+iT} \mathcal{L}_{2,P}(s) \frac{x^s}{s} ds + O(x^c T^{-1}(c-1)^{-2} + x^{1+\epsilon} T^{-1}). \tag{5.2}$$

In order to evaluate the above integral, we need the following well-known estimates for $\zeta(s)$ and L-functions.

Lemma 5.2. *We have*

$$\zeta(\sigma + i\xi) \ll \begin{cases} \frac{1}{\sigma-1} & \text{if } 1 < \sigma \leq 2 \text{ and } \xi = 0 \\ \log|\xi| & \text{if } 1 \leq \sigma \text{ and } |\xi| \geq e \\ |\xi|^{\frac{1-\sigma}{2}} \log|\xi| & \text{if } 0 \leq \sigma \leq 1 \text{ and } |\xi| \geq e \end{cases}$$

and

$$\frac{1}{\zeta(\sigma + i\xi)} \ll \log^7 |\xi|$$

if $\sigma \geq 1$ and $|\xi| \geq e$. If χ is a non-principal character modulo q, we have

$$L(\sigma + i\xi, \chi) \ll \log q(|\xi| + 2)$$

for $\sigma \geq 1$ while if χ is a primitive character modulo $q \geq 3$ and $0 \leq \sigma \leq 1$, then

$$L(\sigma + i\xi, \chi) \ll (q(|\xi| + 2))^{\frac{1-\sigma}{2}} \log q(|\xi| + 2).$$

As usual, we estimate the integral in (5.2) by replacing the integral over the rectangle R with vertices $b \pm iT$ and $c \pm iT$ with $b = \frac{1}{\log x}$ and then calculate the residues of the poles of the integrand inside R. In view of (4.2), the only pole of $\mathcal{R}_{2,P}(s)\frac{x^s}{s}$ inside R is $s = 1$, which comes from $\zeta(s)$, and its residue at $s = 1$ is $2^{1-t} L_{-4P}(1)x$ because $\lim_{s\to 1}(s-1)\zeta(s) = 1$.

For solvable P, i.e, $x^2 + Py^2$ having one form per genus, the class number equals the number of genera — which is 2^t (see p. 198 of [24]). Hence $L_{-4P}(1) = \frac{2^{t-1}\pi}{\sqrt{P}}$ for type one P and $L_{-8P}(1) = \frac{2^{t-1}\pi}{\sqrt{2P}}$ for type two P by (4.11) in [11]. Thus, the residue of $\mathcal{R}_{2,P}(s)\frac{x^s}{s}$ at $s = 1$ is $\frac{\pi}{\sqrt{P}}x$.

Next, using the estimates in Lemma 5.2 and (4.2), we may prove that for $|\xi| \leq T$,

$$\mathcal{L}_{2,P}(\sigma + i\xi) \ll \begin{cases} (P(|\xi| + 2))^{(1-\sigma)} \log^2(PT) & \text{if } b \leq \sigma \leq 1, \\ \log^2(PT) & \text{if } 1 \leq \sigma \leq c. \end{cases}$$

It then follows that

$$\begin{aligned}\frac{1}{2\pi i}\int_{b-iT}^{b+iT}\mathcal{L}_{2,P}(s)\frac{x^s}{s}ds &\ll \int_{-T}^{T}|\mathcal{L}_{2,P}(b+i\xi)|\frac{x^b}{|b+i\xi|}d\xi \\ &\ll PT\log^2(PT) \qquad (5.3)\end{aligned}$$

and

$$\begin{aligned}&\frac{1}{2\pi i}\int_{b\pm iT}^{c\pm iT}\mathcal{L}_{2,P}(s)\frac{x^s}{s}ds \\ &\ll \left\{\int_b^1+\int_1^c\right\}|\mathcal{L}_{2,P}(\sigma\pm iT)|\frac{x^\sigma}{T}d\sigma \\ &\ll P(\log PT)^2\int_b^1\left(\frac{x}{PT}\right)^\sigma d\sigma + T^{-1}(\log PT)^2\int_1^c x^\sigma d\sigma \\ &\ll x^cT^{-1}\log^2(PT)\log x. \qquad (5.4)\end{aligned}$$

Now by choosing $c=1+\frac{1}{\log x}$ and $T=(x/P)^{\frac{1}{2}}$, we get from (5.2)–(5.4) that

$$\sum_{n\le x}r_{2,P}(n)=\frac{\pi}{\sqrt{P}}x+O((xP)^{\frac{1}{2}+\epsilon}).$$

The case for type two P can be proved in the same way. This completes the proof of Theorem 5.1. □

For any square-free integer N, we define a constant α by:

$$\alpha(N):=2\gamma+\sum_{p|2N}\frac{\log p}{p+1}+2\frac{L'_{-4N}(1)}{L_{-4N}(1)}-\frac{12}{\pi^2}\zeta'(2)-1 \qquad (5.5)$$

where γ is Euler's constant and $\sum_{p|n}$ is the summation over all prime factors of n.

Theorem 5.3. *Let P be a solvable square-free integer. Let $x>1$ and $\epsilon>0$. We have for either $N=P$ of type one or $N=2P$ of type two:*

$$\sum_{n\le x}r_{2,N}(n)^2=\frac{3}{N}\left(\prod_{p|2N}\frac{2p}{p+1}\right)(x\log x+\alpha(N)x)+O(N^{\frac{1}{4}+\epsilon}x^{\frac{3}{4}+\epsilon})$$

where the implicit constants are independent of both x and P.

Proof. It follows from (1.2) and (5.1) that

$$\mathcal{R}_{2,P}(\sigma)\ll\sum_{n=1}^{\infty}\frac{\sigma_0(n)^2}{n^\sigma}=\frac{\zeta^4(\sigma)}{\zeta(2\sigma)}\ll\frac{1}{(\sigma-1)^4}$$

as $\sigma\to1^+$. Similar to (5.2), for any $c>1$, $\epsilon>0$ and $x,T\ge1$, we have

$$\sum_{n\le x}r_{2,P}(n)^2=\frac{1}{2\pi i}\int_{c-iT}^{c+iT}\mathcal{R}_{2,P}(s)\frac{x^s}{s}ds+O(x^cT^{-1}(c-1)^{-4}+x^{1+\epsilon}T^{-1}). \qquad (5.6)$$

We estimate the integral in (5.3) by replacing the integral over the rectangle R with vertices $\frac{1}{2}\pm iT$ and $c\pm iT$ and then calculate the residues of the poles of the integrand inside R. In view of (4.2), the only pole of $\mathcal{R}_{2,P}(s)\frac{x^s}{s}$ inside R is $s=1$

of order 2 which comes from $\zeta(s)^2$ and corresponds to the terms when $\mu_1 = \mu_2$ in the double summation of (4.2):

$$2^{2(1-t)}\sigma_0(P)\zeta(s)^2 L_{-4P}(s)^2\zeta(2s)^{-1}\prod_{p|2P}(1+p^{-s})^{-1}\frac{x^s}{s} := F(s) \tag{5.7}$$

and its residue at $s=1$ is

$$\begin{aligned} &= \lim_{s\to 1}\frac{d}{ds}\left\{(s-1)^2F(s)\right\} \\ &= \lim_{s\to 1}(s-1)^2F(s)\lim_{s\to 1}\frac{d}{ds}\log\left\{(s-1)^2F(s)\right\}. \end{aligned}$$

Since P is solvable, so

$$\begin{aligned} \lim_{s\to 1}(s-1)^2F(s) &= 2^{2(1-t)}\sigma_0(P)L_{-4P}^2(1)\zeta(2)^{-1}\prod_{p|2P}(1+p^{-1})^{-1}x \\ &= \frac{3}{P}\left(\prod_{p|2P}\frac{2p}{p+1}\right)x. \end{aligned}$$

In view of (5.5) and (5.7), we have

$$\begin{aligned} &\lim_{s\to 1}\frac{d}{ds}\log\left\{(s-1)^2F(s)\right\} \\ &= 2\gamma + \sum_{p|2P}\frac{\log p}{p+1} + 2\frac{L'_{-4P}(1)}{L_{-4P}(1)} - \frac{12}{\pi^2}\zeta'(2) - 1 + \log x \\ &= \alpha(P) + \log x \end{aligned}$$

because $\lim_{s\to 1}\left(\frac{1}{s-1}+\frac{\zeta'(s)}{\zeta(s)}\right) = \gamma$. Therefore the residue of $\mathcal{R}_{2,P}(s)\frac{x^s}{s}$ at $s=1$ is

$$\frac{3}{P}\left(\prod_{p|2P}\frac{2p}{p+1}\right)\left(x\log x + \alpha(P)x\right). \tag{5.8}$$

Next using the estimates in Lemma 5.2 and (4.2), one can prove that for $|\xi| \le T$,

$$\mathcal{R}_{2,P}(\sigma+i\xi) \ll \begin{cases} P^{(1-\sigma)+\epsilon}(|\xi|+2)^{2(1-\sigma)}\log^A T & \text{if } \frac{1}{2} \le \sigma \le 1, \\ P^{\epsilon}\log^A T & \text{if } 1 \le \sigma \le c. \end{cases}$$

It then follows that

$$\begin{aligned} \frac{1}{2\pi i}\int_{\frac{1}{2}-iT}^{\frac{1}{2}+iT}\mathcal{R}_{2,P}(s)\frac{x^s}{s}ds &\ll \int_{-T}^{T}|\mathcal{R}_{2,P}(\frac{1}{2}+i\xi)|\frac{x^{\frac{1}{2}}}{|\frac{1}{2}+i\xi|}d\xi \\ &\ll P^{\frac{1}{2}+\epsilon}x^{\frac{1}{2}}T\log^A T \end{aligned} \tag{5.9}$$

and

$$\begin{aligned}
&\frac{1}{2\pi i}\int_{\frac{1}{2}\pm iT}^{c\pm iT}\mathcal{R}_{2,P}(s)\frac{x^s}{s}ds\\
\ll\ &\left\{\int_{\frac{1}{2}}^{1}+\int_{1}^{c}\right\}|\mathcal{R}_{2,P}(\sigma\pm iT)|\frac{x^\sigma}{T}d\sigma\\
\ll\ &P^{1+\epsilon}T(\log T)^A\int_{\frac{1}{2}}^{1}\left(\frac{x}{PT^2}\right)^\sigma d\sigma+P^\epsilon T^{-1}(\log T)^A\int_1^c x^\sigma\,d\sigma\\
\ll\ &P^\epsilon x^c T^{-1}\log^A T. \qquad (5.10)
\end{aligned}$$

Now by choosing $c=1+\frac{1}{\log x}$ and $T=(x/P)^{\frac{1}{4}}$, we get from (5.6) and (5.8)-(5.10) that

$$\sum_{n\le x}r_{2,P}(n)^2=\frac{3}{P}\left(\prod_{p|2P}\frac{2p}{p+1}\right)(x\log x+\alpha(P)x)+O(P^{\frac{1}{4}+\epsilon}x^{\frac{3}{4}+\epsilon}).$$

The case for type two P can be proved in the same way. This completes the proof of Theorem 5.3. □

In particular, we have established:

Theorem 5.4. *For any $x\ge 1$, we have*

$$\sum_{n\le x}r_{2,p}(n)^2=\frac{8}{p+1}(x\log x+\alpha(p)x)+O(x^{\frac{3}{4}+\epsilon})$$

for $p=5,13,37$ and

$$\sum_{n\le x}r_{2,2p}(n)^2=\frac{4}{p+1}(x\log x+\alpha(2p)x)+O(x^{\frac{3}{4}+\epsilon})$$

for $p=1,3,5,11,29$. Here the implicit constants are again independent of x.

Similarly, in view of (4.3) and (4.4), we have for $x>1$,

$$\sum_{n\le x}r_{2,3}(n)=\frac{\pi}{\sqrt{3}}x+O(x^{\frac{1}{2}+\epsilon})$$

and

$$\sum_{n\le x}r_{2,3}(n)^2=2(x\log x+\alpha_3 x)+O(x^{\frac{3}{4}+\epsilon}) \qquad (5.11)$$

where $\alpha_3:=2\gamma-\frac{4}{3}\log 2+\frac{1}{4}\log 3+\frac{6\sqrt{3}}{\pi}L'_{-3}(1)-\frac{12}{\pi^2}\zeta'(2)-1$.

Also

$$\sum_{n\le x}r_{2,4}(n)=\frac{\pi}{2}x+O(x^{\frac{1}{2}+\epsilon})$$

and

$$\sum_{n\le x}r_{2,4}(n)^2=\frac{3}{2}(x\log x+\alpha_4 x)+O(x^{\frac{3}{4}+\epsilon})$$

where $\alpha_4:=2\gamma-\frac{2}{3}\log 2+\frac{8}{\pi}L'_{-4}(1)-\frac{12}{\pi^2}\zeta'(2)-1$.

Akin to Wagon's conjecture, we make the following conjecture.

Quadratic Conjecture. *For any square-free P,*

$$\sum_{n\le x} r_{2,P}(n) \sim \frac{\pi}{\sqrt{P}}x$$

and

$$\sum_{n\le x} r_{2,P}(n)^2 \sim \frac{3}{P}\left(\prod_{p|2P}\frac{2p}{p+1}\right)x\log x$$

as $x\to\infty$.

In view of Theorem 5.3, (3.14) and (5.11), our conjecture is true for solvable P and for $P=1,3$. We have also confirmed it for $P=7$ and 15 from the representations of

$$\mathcal{L}_{2,7}(s) = 2\,(1-2^{1-s}+2^{1-2s})\zeta(s)L_{-7}(s)$$

and

$$\mathcal{L}_{2,15}(s) = (1-2^{1-s}+2^{1-2s})\zeta(s)L_{15}(s) + (1+2^{1-s}+2^{1-2s})L_{-3}(s)L_5(s)$$

again given in [10], which leads to

$$\mathcal{R}_{2,7}(s) = 4\frac{(1-3\cdot 2^{-s}+2^{2-2s})}{(1+2^{-s})(1+7^{-s})}\frac{(\zeta(s)L_{-7}(s))^2}{\zeta(2s)}$$

and

$$\mathcal{R}_{2,15}(s) = \frac{2(1-3\cdot 2^{-s}+2^{2-2s})}{(1+2^{-s})(1+3^{-s})(1+5^{-s})}\frac{(\zeta(s)L_{-15}(s))^2}{\zeta(2s)} + \frac{2(1+3\cdot 2^{-s}+2^{2-2s})}{(1-2^{-s})(1-3^{-s})(1-5^{-s})}\frac{(L_{-3}(s)L_5(s))^2}{\zeta(2s)},$$

and may be analyzed by the methods above.

6. Sums of Three Squares and Other Powers

6.1. **Three Squares.** Odd squares are notoriously less amenable to closed forms. In this subsection, we primarily record some results for $r_3(n)$, the number of representations of n as a sum of three squares. Following Hardy, Bateman in [2] gives the following formula for $r_3(n)$. Let

$$\chi_2(n) := \begin{cases} 0 & \text{if } 4^{-a}n \equiv 7 \pmod 8;\\ 2^{-a} & \text{if } 4^{-a}n \equiv 3 \pmod 8;\\ 3\cdot 2^{-1-a} & \text{if } 4^{-a}n \equiv 1,2,5,6 \pmod 8\end{cases}$$

where a is the highest power of 4 dividing n.

Then

$$r_3(n) = \frac{16\sqrt{n}}{\pi}L_{-4n}(1)\,\chi_2(n) \times \prod_{p^2|n}\left(\frac{p^{-\tau}-1}{p^{-1}-1}+p^{-\tau}\left(1-\frac{1}{p}\left(\frac{-p^{-2\tau}n}{p}\right)\right)^{-1}\right) \tag{6.1}$$

where $\tau=\tau_p$ is the highest power of p^2 dividing n.

The Dirichlet series for $r_3(n)$ deriving from (6.1) is not as malleable as those of (3.1)-(3.4), but we are able to derive a nice expression in terms of Bessel functions.

Let K_s be the *modified Bessel function of the second kind.* Then we have (see [27], p. 183)

$$K_s(x) = \frac{1}{2}\left(\frac{x}{2}\right)^s \int_0^\infty e^{-t-\frac{x^2}{4t}} \frac{dt}{t^{s+1}}. \tag{6.2}$$

By the substitution $t = \frac{1}{u}$ in (6.2), we get

$$K_s(x) = \frac{1}{2}\left(\frac{x}{2}\right)^s \int_0^\infty e^{-\frac{x^2 u}{4}-\frac{1}{u}} u^{s-1} du. \tag{6.3}$$

Let

$$\theta_3(q) := \sum_{n=-\infty}^{\infty} q^{n^2}$$

be the classical Jacobean theta function. In view of the Poisson summation formula, we have, for $t > 0$

$$\theta_3(e^{-\pi t}) = t^{-\frac{1}{2}}\theta_3(e^{-\pi/t}).$$

Since the Mellin transform of $e^{-\alpha t}$ for $\alpha \neq 0$ is $M_s(e^{-\alpha t}) = \Gamma(s)\alpha^{-s}$, so we have (letting $q = e^{-\pi t}$)

$$\begin{aligned}
\mathcal{L}_3(s) &= 3\sum_{n,m,p\in\mathbb{Z}} \frac{n^2}{(n^2+m^2+p^2)^{s+1}} \\
&= \frac{3\pi^{s+1}}{\Gamma(s+1)} \sum_{n,m,p\in\mathbb{Z}} n^2 M_{s+1}(q^{n^2+m^2+p^2}) \\
&= \frac{3\pi^{s+1}}{\Gamma(s+1)} M_{s+1}\left(\sum_{n\in\mathbb{Z}} n^2 q^{n^2}\theta_3^2(q)\right) \\
&= \frac{3\pi^{s+1}}{\Gamma(s+1)} \sum_{n\in\mathbb{Z}} n^2 \int_0^\infty e^{-n^2\pi t}\theta_3^2(e^{-\pi/t}) t^{s-1} dt \\
&= \frac{3\pi^{s+1}}{\Gamma(s+1)} \sum_{n\in\mathbb{Z}} n^2 \sum_{m=1}^{\infty} r_2(m) \int_0^\infty e^{-n^2\pi t-\frac{\pi m}{t}} t^{s-1} dt \\
&\quad + \frac{3\pi^{s+1}}{\Gamma(s+1)} \sum_{n\in\mathbb{Z}} n^2 \int_0^\infty e^{-n^2\pi t} t^{s-1} dt. \qquad (6.4)
\end{aligned}$$

The first term of (6.4) is

$$\begin{aligned}
&= \frac{6\pi^{s+1}}{\Gamma(s+1)} \sum_{n=1}^{\infty} n^2 \sum_{m=1}^{\infty} r_2(m) \int_0^\infty e^{-n^2\pi t-\frac{\pi m}{t}} t^{s-1} dt \\
&= \frac{6\pi^{s+1}}{\Gamma(s+1)} \sum_{m=1}^{\infty} r_2(m)(\pi m)^s \sum_{n=1}^{\infty} n^2 \int_0^\infty e^{-n^2\pi^2 m x-1/x} x^{x-1} dx, \qquad (x = \frac{t}{\pi m}) \\
&= \frac{12\pi^{s+1}}{\Gamma(s+1)} \sum_{m=1}^{\infty} r_2(m) m^{s/2} \sum_{n=1}^{\infty} \frac{1}{n^{s-2}} K_s(2\pi n\sqrt{m})
\end{aligned}$$

by (6.3) and the second term is

$$\begin{aligned} &= \frac{6\pi^{s+1}}{\Gamma(s+1)} \sum_{n=1}^{\infty} \frac{1}{n^{2s-2}\pi^s} \int_0^{\infty} e^{-x} x^{s-1} ds \\ &= \frac{6\pi}{s} \zeta(2s-2). \end{aligned}$$

This proves the following result:

$$\mathcal{L}_3(s) = \frac{6\pi}{s}\zeta(2s-2) + \frac{12\pi^{s+1}}{\Gamma(s+1)} \sum_{m=1}^{\infty} r_2(m) m^{s/2} \sum_{n=1}^{\infty} \frac{1}{n^{s-2}} K_s(2\pi n\sqrt{m}). \tag{6.5}$$

There is a corresponding formula for $\sum(-1)^n r_3(n)/n^s$ which corresponds to Madelung's constant (see p. 301 in [3]). The second term of (6.5) can be rewritten as

$$\frac{12\pi^{s+1}}{\Gamma(s+1)} \sum_{k>0} k^{\frac{s}{2}} K_s(2\pi\sqrt{k}) \sum_{n^2|k} \frac{r_2(k/n^2)}{n^{2s-2}}.$$

Moreover, these Bessel functions are elementary when s is a half-integer. Most nicely, for 'jellium', which is the Wigner sum analogue of Madelung's constant, we have

$$\mathcal{L}_3(1/2) = -\pi + 3\pi \sum_{m>0} \frac{r_2(m)}{\sinh^2(\pi\sqrt{m})},$$

and the exponential convergence is entirely apparent.

For a survey of other rapidly convergent lattice sums of this type see [3] and [6].

There is a corresponding formula for $\mathcal{L}_N(s)$, for all $N \geq 2$, in which we obtain a Bessel-series in $r_{N-1}(m)$:

$$\begin{aligned} \mathcal{L}_N(s) = \sum_{n>0} \frac{r_N(n)}{n^s} &= \frac{2N\,\Gamma(s-\frac{N-3}{2})}{\Gamma(s+1)} \pi^{\frac{N-1}{2}} \zeta(2s-N+1) \\ &+ \frac{4N\,\pi^{s+1}}{\Gamma(s+1)} \sum_{m>0} \frac{m^{\frac{1}{2}s}\, r_{N-1}(m)}{m^{\frac{N-3}{4}}} \sum_{n>0} \frac{n^{\frac{N+1}{2}}}{n^s} K_{s-\frac{N-3}{2}}(2n\pi\sqrt{m}). \end{aligned} \tag{6.6}$$

There is an equally attractive integral representation (see [27] p. 172) for:

$$K_s(x) = \left(\frac{2}{x}\right)^s \frac{\Gamma(s+1/2)}{\Gamma(1/2)} \int_0^{\infty} \frac{\cos(xt)}{(1+t^2)^{s+1/2}}\, dt$$

at least when $x > 1/2$. This leads to

$$\sum_{n>0} \frac{r_3(n)}{n^s} = 2L_{-4}(s+\frac{1}{2},\frac{1}{2}) \sum_{m>0} r_2(m) \int_0^{\infty} \frac{C_{s-2}(\sqrt{m}t)}{(1+t^2)^{s+1/2}}\, dt$$

where

$$C_s(x) = \sum_{n>0} \frac{\cos(2\pi\, nx)}{n^s}$$

is a *Clausen-type* function. For $s = 2k$, even integer, this evaluates to

$$C_{2k}(x) = \frac{(2\pi)^{2k}}{(-1)^{k-1} 2(2k)!} B_{2k}(x)$$

where B_k is a Bernoulli polynomial.

Obviously this also extends to reworkings of (6.6). For example, the $N = 2$ case yields

$$4\,L_{-4}(s+\frac{1}{2},\frac{1}{2})\zeta(2\,s-1)+\frac{16\,\pi^{1+s}}{\Gamma(s+1)}\sum_{n=1}^{\infty}\frac{\sigma_{2\,s-1}(n)}{n^{s-\frac{3}{2}}}K_{s+\frac{1}{2}}(2\,n\pi)=4\,\zeta(s)L_{-4}(s).$$

This in turn, with $s = 2$, becomes

$$4\,\pi^3\sum_{n=1}^{\infty}\sigma_3(n)e^{-2\,n\pi}\left(1+\frac{3}{2}\,\frac{1}{n\pi}+\frac{3}{4}\,\frac{1}{n^2\pi^2}\right)\frac{1}{n}=\frac{2}{3}\pi^2\mathrm{G}-\frac{3}{2}\zeta(3),$$

where $\mathrm{G} := \sum_{\mathrm{n}\geq 0}(-1)^{\mathrm{n}}\,(2\mathrm{n}+1)^{-2}$ is *Catalan's constant*.

There is a puissant formula for θ_2^3 due to Andrews [1] (given with a typographical error in [3] p. 286). It is

$$\theta_2^3(q)=8\sum_{n=0}^{\infty}\sum_{j=0}^{2\,n}\left(\frac{1+q^{4\,n+2}}{1-q^{4\,n+2}}\right)q^{(2\,n+1)^2-(j+1/2)^2}. \tag{6.7}$$

Lamentably we have not been able to use it to study $\mathcal{R}_3$, or even $\mathcal{L}_3$ any further than was achieved in [6].

6.2. **Twelve and Twenty-four Squares.** Explicit 'divisor' formulae for $r_{12}(n)$ and $r_{24}(n)$ are also known (e.g. p. 200 of [20] and §9 of Chapter 9 in [15]): they are

$$r_{12}(n)=8(-1)^{n-1}\sum_{d|n}(-1)^{d+n/d}d^5+16\omega(n)$$

and

$$r_{24}(n)=\frac{16}{691}\sigma_{11}^*(n)+\frac{128}{691}\left((-1)^{n-1}259\tau(n)-512\tau(\frac{1}{2}n)\right)$$

where $\sigma_{11}^*(n)=\sum_{d|n}d^{11}$ if n is odd and $\sigma_{11}^*(n)=\sum_{d|n}(-1)^d d^{11}$ if n is even,

$$q((1-q^2)(1-q^4)(1-q^6)\cdots)^{12}=\sum_{n=1}^{\infty}\omega(n)q^n$$

and

$$q((1-q)(1-q^2)(1-q^3)\cdots)^{24}=\sum_{n=1}^{\infty}\tau(n)q^n.$$

Here $\tau(n)$ is the famous Ramanujan's τ-function.

We have recorded these representations because, while $N = 12$ and $N = 24$ (due to Ramanujan, see Chapter IX of [13]) are the next most accessible even cases, neither directly lead to an appropriate closed form for $\mathcal{L}_N$ let alone for $\mathcal{R}_N$. This is thanks to the impediment offered by ω and τ respectively: which encode knowledge, via the Jacobi triple-product, of all the representations of n as a sum of 4 or 8 squares. The divisor functions do produce appropriate L-function representations. Thus, using Ramanujan's ζ-function

$$g_{24}(s):=\sum_{n=1}^{\infty}\frac{\tau(n)}{n^s}=\prod_p\left(1-\tau(p)p^{-s}+p^{11-2s}\right)^{-1},$$

which is discussed in detail in Chapter X of [13], it transpires that τ is multiplicative, with the preceding lovely Euler product. Additionally,

$$\begin{aligned}\mathcal{L}_{24}(s) = \sum_{n=1}^{\infty} \frac{r_{24}(n)}{n^s} &= \frac{16}{691}(2^{12-2s} - 2^{1-s} + 1)\zeta(s)\zeta(s-11) \\ &+ \frac{128}{691}(259 + 745 \cdot 2^{4-s} + 259 \cdot 2^{12-2s})g_{24}(s).\end{aligned}$$

Similarly with $g_{12}(s) := \sum_{n=1}^{\infty} \frac{\omega(n)}{n^s}$ one has

$$\mathcal{L}_{12}(s) = \sum_{n=1}^{\infty} \frac{r_{12}(n)}{n^s} = 8(1 - 2^{6-2s})\zeta(s)\zeta(s-5) + 16g_{12}(s).$$

We also note that the analysis in [13], due to Rankin (see [22]), provides an 'almost closed form' for

$$f(s) := \sum_{n=1}^{\infty} \frac{\tau^2(n)}{n^s} = \prod_p \left(1 + \tau^2(p)p^{-s} - p^{22-2s} - \frac{2\tau^2(p)p^{-s}}{1+p^{11-s}}\right)^{-1}.$$

Rankin studied the above function $f(s)$ in [22] and showed that $f(s)$ has an analytic continuation to a meromorphic function on $\mathbb{C}$ with the only poles at $s = 12$ and at the complex zeros of $\zeta(2s-22)$, all lying to the left of $\Re(s) = 12$. In [22], Rankin proved his famous result that $\tau(n) = O(n^{29/5})$. His proof depends on a functional equation of $f(s)$, namely,

$$\begin{aligned}&(2\pi)^{-2s}\Gamma(s)\Gamma(s-11)\zeta(2s-22)f(s) = \\ &(2\pi)^{2s-46}\Gamma(23-s)\Gamma(12-s)\zeta(24-2s)f(23-s).\end{aligned}$$

is invariant as $s \to 23 - s$. Finally, we note that a recent paper by Ewell [9] has a new divisor like recursion for τ.

Acknowledgments

The second author wishes to thank Professor P. Borwein for his support concerning this paper. The authors also wish to thank Greg Fee for some useful computational assistance and Stan Wagon and Richard Crandall for many stimulating exchanges. Finally, the authors wish to express their gratitude to Professor Paul Bateman for his thoughtful and gracious comments and suggestions, especially for improving and simplifying the error estimates to the average order of $r_N^2(n)$ in §3.

References

[1] G. E. Andrews, "The Fifth and Seventh Order Mock Theta Functions," *Transactions of the AMS*, **293** (1986), 113-134.

[2] P. Bateman, "On the Representation of a Number as the Sum of Three Squares", *Transactions of the AMS*, **71** (1951), 70-101.

[3] J.M. Borwein and P.B. Borwein, *Pi and the AGM. A study in analytic number theory and computational complexity*, CMS, Monographs and Advanced Texts, 4. John Wiley & Sons, New York, 1987. Paperback, 1998.

[4] L. Carlitz, "A note on the multiplication formulas for the Bernoulli and Euler polynomials," *Proceedings of the AMS*, **4** (1953), 184-188.

[5] R.D. Connors and J.P. Keating, "Degeneracy moments for the square billiard," *J. Phys. G: Nucl. Part. Phys.* **25** (1999), 555-562.

[6] R. E. Crandall, "New representations for the Madelung constant," *Experimental Mathematics*, **8:4** (1999), 367-379.

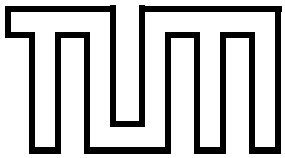
[7] R. E. Crandall, "Signal processing applications in additive number theory," (2001) preprint.
[8] R. Crandall and S. Wagon, "Sums of squares: Computational aspects," (2001) preprint.
[9] J.A. Ewell, "New representations of Ramanujan's tau function," *Proc. Amer. Math. Soc.* **128** (1999), 723-726.
[10] M. Glasser and I. Zucker. "Lattice Sums," in *Theoretical Chemistry : Advances and Perspectives,* **5** (1980), 67-139.
[11] E. Grosswald, *Representations of Integers as Sums of Squares*, Springer-Verlag, 1985.
[12] G.H. Hardy, *Collected Papers*, Vol I, Oxford University Press, 1969.
[13] G.H. Hardy, *Ramanujan*, Cambridge University Press, 1940. Revised Amer. Math. Soc. , 1999.
[14] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, 5th Ed., Oxford, 1979.
[15] L.K. Hua, *Introduction to Number Theory*, Springer-Verlag, 1982.
[16] A.A. Karatsuba, *Basic Analytic Number Theory*, Springer-Verlag, 1991.
[17] E. Landau, *Vorlesungen über Zahlentheorie*, Leipzig, Hirzel, 1927.
[18] E. Landau, *Collected works*, Vol. 4. (German) Edited and with a preface in English by P. T. Bateman, L. Mirsky, H. L. Montgomery, W. Schaal, I. J. Schoenberg, W. Schwarz and H. Wefelscheid. Thales-Verlag, Essen, 1986.
[19] W. Nowak, "Zum Kreisproblem", österreich. Akad. Wiss. Math.-Natur. Kl. Sitzungsber. II **194** (1985), no. 4-10, 265-271.
[20] H. Rademacher, *Topics in Analytic Number Theory*, Springer-Verlag, 1973.
[21] S. Ramanujan, "Some formulae in the analytic theory of numbers" , *Messenger of Math.*, **45** (1916), 81-84.
[22] R. Rankin, "Contributions to the theory of Ramanujan's function $\tau(n)$ and similar arithmetical functions (I), (II), (III)", *Proc. Cambridge Philos. Soc.*, **35, 36** (1939), (1940), 351-356, 357-372, 150-151.
[23] M.M. Robertson and I.J. Zucker, "Exact Values for Some Two-dimensional Lattice Sums," *J. Phys. A: Math. Gen.* **8** (1975), 874-881.
[24] H.E. Rose, *A Course in Number Theory*, Oxford Science Publications, 2nd Ed, 1994.
[25] D. Shanks, "Calculation and Applications of Epstein Zeta Functions", *Math. Comp.*, **29** (1975), 271-287.
[26] E.C. Titchmarsh, *The Theory of the Riemann Zeta-Function*, Oxford Science Publications, 2nd Ed, 1986.
[27] G.N. Watson, *A Treatise on the Theory of Bessel Functions*, Cambridge University Press, 1966.

CECM, Department of Mathematics, Simon Fraser University, Burnaby B.C., Canada, V5A 1S6. Email: jborwein@cecm.sfu.ca, choi@cecm.sfu.ca

Scheduling Independent and Identically Distributed Tasks with In-Tree Constraints on Three Machines in Parallel

Moritz G. Maaß

München, 2001



Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen

Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Diplomarbeit

Thema:	Scheduling Independent and Identically Distributed Tasks with In-Tree Constraints on Three Machines in Parallel
Bearbeiter:	Moritz G. Maaß
Aufgabensteller:	Prof. Dr. Ernst W. Mayr
Betreuer:	Prof. Dr. Ernst W. Mayr
Abgabedatum:	15. Oktober 2001

Ich versichere, dass ich diese Diplomarbeit
selbständig verfasst und nur die angegebenen
Quellen und Hilfsmittel verwendet habe.

27. August 2001

Contents

List of Figures

Chapter 1

Introduction

Scheduling – the problem of allocating resources to different tasks – is a classical problem of computer science. As this is one of the fundamental problems of organizing work in a world based on the division of labor, scheduling problems have their roots in the domain of operations research, an economical, mathematical and military discipline. With the availability of computers, a lot more problems have become tractable in reasonable time (compared to the time and resources saved by an optimized work schedule).

A scheduling problem generally consists of a set of resources (such as processors, workers, machines), a set of tasks, usually some constraints (due dates, machine constraints, precedence relationships, etc.), and an optimization objective (usually a time based term). The information is mostly given as the number and the size of the tasks (the amount of processing needed per task), the number of processors (machines) and their capabilities (the processing speed), some external constraints in the form of arrival and due dates or precedence constraints (an order on the processing sequence), and the objective to optimize. The first scheduling problems studied by computer scientists were problems where most information was deterministic and known a priori. Although it turned out soon that a lot of these problems were computationally intractable (NP-hard problems), optimal algorithms are known for quite a lot of relevant problems. Usually, even if a problem is NP-hard, some versions of it can be solved. A lot of approximation algorithms with good time bounds are known (see [Pin95, Bru95] for general overviews).

Precedence constraints in their most general form can be modeled by a directed acyclic graph. Unfortunately, even the simple problem of scheduling identical tasks on multiple processors in parallel with general precedence constraints to minimize the makespan (the total processing time) is NP-hard [Ull75, Ull76]. When the problem is limited to precedence constraints that form an in-tree or an out-tree, then the problem becomes solvable by the highest level first (HLF, sometimes also CP – critical path) rule [Hu61]. If the limited problem is generalized by allowing tasks of lengths 1 and 2, then the problem is again NP-hard [Ull75].

The assumption that all information for a scheduling problem is known a priori is often unrealistic. Things may happen coincidentally or there may be no chance to predict the behavior of a problem component. To take this into account, a scheduling problem is modeled with random variables. An early approach was made by Chandy and Reynolds [CR75], who modeled the task processing time as independent identically, exponentially distributed random variables and proved that the HLF strategy minimizes the expected makespan for two machines and precedence constraints that form an in-tree. They also gave a counter example for the three machines case. Under the same problem with two machines Bruno [Bru85] showed that the HLF strategy maximizes the probability that all tasks are finished by a given time (stochastically minimizes the total expected processing time) and that the HLF strategy also maximizes the probability that the sum of the finishing times is below a given value (only the preemptive case is considered). Pinedo and Weiss [PW85] were able to show that the HLF strategy minimizes the expected makespan if the processing times of all tasks at level l are independent identically, exponentially distributed (tasks at different levels may have different distributions). They consider the preemptive and the non-preemptive case. A modified HLF strategy even minimizes the expected makespan for more general (but discrete) distributions. Frostig

[Fro88] considers increasing hazard rate (IHR) distributions and proves that a modified HLF strategy (ties are broken by selecting tasks with lowest hazard rate, the strategy hence called HLF:LHR) stochastically minimizes the total expected processing time. Going back to the original problem definition by Chandy and Reynolds with a fixed number of machines, in-tree constraints and independent identically, exponentially distributed task running times, an optimal strategy for three or more machines is unknown. Papadimitriou and Tsitsiklis [PT87] have shown that the HLF strategy is asymptotically optimal as the number of tasks tends to infinity.

We take Chandy and Reynolds definition as a basis here and scrutinize the three machines version. Therefore, we have developed an algorithm with exponential running time to optimally solve the three machines version of the problem. With this algorithm we were able to generate a large number of small in-trees (with less than 19 nodes) and compare the optimal strategy with other strategies. We prove that not only the HLF strategy, but all static list scheduling strategies (a static list scheduling orders all tasks at the beginning and always schedules the first available task from the list) cannot be optimal. Furthermore, if one considers intermediate states in the scheduling processes, even all semi-static list scheduling strategies (the task list is generated per tree) cannot be optimal. As a result, an optimal schedule must take already scheduled leaves from a prior step into account.

We will show that the structure of the problem depends on subtrees of the precedence constraints in-tree with only two leaves. Their influence on the problem lies in the fact that in the end-phase of the scheduling process such a subtree with only two leaves is reached and one machine must be left idle. The expected makespan depends on how early and which such subtree is reached. Based on this structure we have developed some algorithms and compared their performance with each other and with the HLF strategy. With only two machines, the problem structure can be broken down analogously resulting in subtrees with one leaf only, which gives us another hint on the optimality of the HLF strategy for two machines. An optimal algorithm for the three machines version of the problem could not be found.

This thesis is structured as follows. Chapter 2 establishes the notation used throughout the work. The experienced reader might as well skim through the chapter and jump back later if needed. Chapter 3 gives some background on scheduling and scheduling strategies. The widely used three parts classification scheme is introduced and the problem at hand is classified. In chapter 4 the use of the exponential distribution in this particular scheduling problem is introduced and the main properties are extracted. A sidelong glance at the geometric distribution is taken. The in-tree precedence constraints are looked at more closely in chapter 5. We give an introductory example, establish a frame for scheduling tasks with independent identically, exponentially distributed running times and in-tree constraints and formalize the optimization objective. In chapter 6 we develop a dynamic programming algorithm with exponential running time to calculate the optimal solutions. We define some evaluation criteria for different strategies and apply these to the HLF and other strategies in chapter 7. The structure of the problem is broken down into two-leaves-subtrees in chapter 8. The results are applied to develop algorithms based on combinatorial approximations in chapter 9 and based on a node-wise reflection in chapter 10. We try to draw a conclusion and give an outlook in chapter 11.

I want to thank Prof. Dr. Ernst W. Mayr for suggesting the problem to me and for helpful remarks, hints, corrections, and encouragement. I also want to express gratitude to Alexander Offtermatt-Souza for some fruitful discussions. I am in debt to Hanjo Täubig and my mother Maren Stiller-Maaß for reading and correcting my final work. Last but not least I want to thank Ina Jahnel for infinite patience and support.

Chapter 2

Basic Definitions

This work will deal a lot with graphs, in particular with trees. We therefore start with some general notation for later use. The experienced reader can just skim through this section and jump back when needed.

Let $G = (V, E)$ be a graph, where $V \neq \emptyset$ is a non-empty set of vertices and $E \subseteq \{\{u, v\} | u \in V \wedge v \in V\}$ is a set of unordered pairs of nodes, the edges. If $E \subseteq V \times V$, the graph is said to be directed. Each edge in a directed graph is an ordered pair of vertices. If $v \in V$ is a vertex of G we will also use the shorthand $v \in G$. In the following we only look at directed graphs.

Let $e = (u, v) \in E$ be an edge. Then $start(e) = u$ and $end(e) = v$. For a node $v \in V$, we will use $in(v) = \{(u, v) | (u, v) \in E\}$ and $out(v) = \{(v, u) | (v, u) \in E\}$. The in-degree of node v is $d_{in}(v) = |in(v)|$, the out-degree is $d_{out}(v) = |out(v)|$.

A path p from node u to node v is an n-tuple of edges $p = path(u, v) = (e_1, e_2, \ldots, e_n)$ where $e_i \in E, i = 1, \ldots, n$ and $start(e_1) = u \wedge end(e_n) = v \wedge \forall i = 2..n : end(e_{i-1}) = start(e_i)$. A path can also be denoted by its nodes. In this case a path is an n-tuple of nodes $path(u, v) = (u, w_1, w_2, \ldots, w_{n-2}, v)$. The length of a path is the number of nodes it contains.

Let $paths_G(u, v)$ be the set of all paths from u to v in G.

A cycle is a path $(u, w_1, w_2, \ldots, w_{n-2}, u)$, where $\forall i = 1..n-2 : w_i \neq u$ and the length of the path is at least 2.

A graph that contains no cycles is called a **directed acyclic graph (DAG)**.

A **tree** T is a graph with n nodes, $n - 1$ edges and with no cycles.

A **rooted tree** B is a pair (r, T) where T is a tree and $r \in T$ is a node of T. A rooted tree $B = (r, T)$ is called an out-tree, if and only if $\forall v \in T : v = r \vee \exists path(r, v)$. A rooted tree $B = (r, T)$ is called an in-tree, if and only if $\forall v \in T : v = r \vee \exists path(v, r)$.

Rooted trees with all maximal paths either starting or ending at the root are also called **distinct oriented trees with labeled vertices** (see [Knu97], section 2.3.4.4) or **rooted and labeled trees**.

In the following all trees will be in-trees and we will therefore not need to define the root of the tree explicitly. For an in-tree B the root is the only node r with out-degree $d_{out}(r) = 0$. Let $root(B) = r$.

A node $p \in B$ is an ancestor of $v \in B$ in in-tree B, if there exists a path $path(v, p)$ (p is "closer to root" than v).

In in-trees there is no need to specify a path by a tuple since there can only be one path between two nodes. We will therefore let $path(u, v)$ be the set of nodes rather than a tuple. If there exists a path from a to b, then let $dist(a, b) = |path(a, b)| - 1$ (which is the number of edges between a and b).

For the following let B be an in-tree and r be its root.

If a is a node in B, then let $height(a) = dist(a, r)$ (the number of node levels).

Let $height(B) = \max_{a \in B}\{height(a)\} + 1$.

Let a and b be nodes in B. Let p be a node such that p is a common ancestor of a and b ($\exists path(a,p) \wedge \exists path(b,p)$). p is said to be the lowest common ancestor (lca), if for all v that are common ancestors of a and b, $height(p) = max_v\{height(v)\}$. (Note: The max is introduced here because we let trees grow in an unusual direction - upwards. The lowest common ancestor is hence really a highest common ancestor, but lca is the name commonly used.). We define $lca(a,b) := p$. There is always a lowest common ancestor for any nodes a, b, because there is a path to the root from any node in the tree.

Let $leaves(B)$ be the set of leaves of tree B.

Let a be a node in B, then let $B|_a$ be the subtree of B, whose root is a.

Two trees $B_1 = (V_1, E_1)$ and $B_2 = (V_2, E_2)$ are isomorphic, if there exists a bijective function $f : V_1 \to V_2$, s.t. $\forall(v,u) \in E_1 : (f(v), f(u)) \in E_2$ and $\forall(v,u) \in E_2 : (f^{-1}(v), f^{-1}(u)) \in E_1$.

If we build equivalence classes of trees with respect to isomorphism, we can represent each class by a rooted, unordered tree that only describes the tree structure.

The number of different trees with n nodes differs immensely based on what trees are considered the same:

Lemma 2.1 (Number of Rooted and Labeled Trees). *The number of rooted and labeled trees is* n^{n-1}.

Proof. From Cayley's Formula we know that the number of labeled trees is n^{n-2}. There are n nodes, choosing a distinct root yields n^{n-1}. More detailed proof can be found in [Knu97], section 2.3.4.4. □

The number of rooted, unordered trees is smaller (we drop the labeling and a lot of trees are now isomorphic). The sequence a_n of the number of rooted, unordered trees with n nodes starts with $1, 1, 2, 4, 9, 40, 48, \ldots$ and it can also be found as sequence number $A000081$ in [Onl]. There is no closed formula for this number, but a_n grows asymptotically:

Lemma 2.2 (Asymptotic Growth of the Number of Rooted, Unordered Trees with n Nodes). *The number of rooted, unordered trees with n nodes a_n is asymptotically*

$$a_n = \frac{1}{\alpha^{n-1}n}\sqrt{\beta/2\pi n} + \mathcal{O}\left(\frac{1}{\sqrt{n^5}\alpha^n}\right)$$

Where $1/\alpha \approx 2.955765285652$ *and* $\sqrt{\beta/2\pi} \approx 0.439924012571$

Proof. See [Knu97], section 2.3.4.4. □

As a rule of thumb, the above formula tells us that a_n grows nearly as fast as 3^n.

Chapter 3

Scheduling

Michael Pinedo defines scheduling in [Pin95] as

> Scheduling concerns the allocation of limited resources to tasks over time. It is a decision-making process that has as a goal the optimization of one or more objectives.

Following [Pin95] we will give a brief introduction into the scheduling field.

3.1 A Classification Scheme for Scheduling Problems

A scheduling problem can be described by a triple $(\alpha|\beta|\gamma)$. The first part (α) describes the machines (also resources), the second part (β) describes the tasks and the third part (γ) describes the optimization objective. This notation was introduced by Lawler et al. [LLR82], the presented version is a slightly adapted one from Pinedo [Pin95].

The following notations for the description of the resources are known:

1	A single machine is available for processing the given task(s).
Pm	There are m identical machines available for processing the given tasks in parallel.
Qm	There are m machines with different processing speeds v_j available for working on the given tasks in parallel. (Machine j can process task i in p_i/v_j time).
Rm	There are m machines with different processing capabilities available for working on the given tasks in parallel. The machine speed depends also on the task, so that machine j can process task i at speed v_{ij}.
Fm	In the Flow Shop environment with m machines in series each task has to be processed on each machine, starting with the first and ending with the last machine.
FFs	In the Flexible Flow Shop environment there are s stages and a number of machines in parallel at each stage. Each task has to be processed in every stage, starting with the first and ending with the last one. No task may overtake another one.
Om	The Open Shop environment relaxes constraints from Fm further such that the task may be routed differently (determined by the scheduler). Some tasks may have processing time zero on some machines.
Jm	In the Job Shop Environment each task has to follow a specific route through the m machines.

The description of the tasks usually includes further constraints, some of which may be:

r_j Tasks have release times r_j.

s_{jk} Tasks have setup times depending on the sequence they are assigned to a machine (s_{jk} is the time needed to set up the machines between jobs j and k).

$prmp$ A processor processing a task may be preempted and be reassigned to another task by the scheduler. Usually one assumes that a processor once assigned to a task will only be available after that task has finished.

$prec$ Precedence constraints define precedence between different tasks, requiring that some tasks must be finished before others can be started. The precedence constraints can be described by a DAG $G = (V, E)$, where V are the tasks and $(u, v) \in E$ is an edge from u to v, if u must be finished before v can be started.

$chains$ This is a special case of precedence constraints ($prec$), where the maximal in-degree and the maximal out-degree of any node in the precedence graph is smaller or equals to one ($\forall v \in V : d_{in}(v) \leq 1 \wedge d_{out}(v) \leq 1$).

$outtree$ This is a special case of precedence constraints ($prec$), where the maximal in-degree of any node in the precedence graph is smaller or equals to one ($\forall v \in V : d_{in}(v) \leq 1$). The task-nodes form an out-tree or an out-forest.

$intree$ This is a special case of precedence constraints ($prec$), where the maximal out-degree of any node in the precedence graph is smaller or equals to one ($\forall v \in V : d_{out}(v) \leq 1$). The task-nodes form an in-tree or an in-forest.

$brkdwn$ Machines can break down and not be available during some time.

M_j In the environment Pm this means that not all machines can process all tasks, only machines in set M_j can process task j

$no\text{-}delay$ No machine may be idle if there is a free task that can be processed. Usually all problems are considered no-delay (that is only greedy strategies are considered).

We will usually assume that there are n tasks to be processed. The optimization criterion makes use of the following variables:

p_{ij} is the time that task j has been processed on machine i. The index i can be omitted if the machines do not differ or if only one machine is available. p_j then denotes the total time that task j was processed on any machine.

d_j is the due date. That is the time by which a task should be finished.

w_j is a weight that defines the priority given to a task in relation to other tasks.

C_j is the completion time of task j (the time that task j has received all the processing that it needed).

U_j denotes whether a task finished on time. It is 1 if $d_j >= C_j$ and 0 otherwise.

L_j is the lateness of a task. It is defined as $L_j = C_j - d_j$.

X_{ij} is the random processing time of task j on machine i. This is a pendant to p_{ij} for a stochastic scheduling environment.

$\frac{1}{\lambda_{ij}}$ is the expected value of X_{ij} ($\mathbb{E}(X_{ij})$ – often the exponential distribution is used, hence the fraction here).

R_j is the random release time of task j (similar to r_j).

D_j is the random due date of task j (similar to d_j).

The following are common optimization criteria (most of which are only included for completeness):

C_{max} The **makespan** is the time that the last job has finished:

$$C_{max} = max\{C_1, \ldots, C_n\}$$

$\mathbb{E}(C_{max})$ The **expected makespan** is the expected time that the last job has finished:

$$\mathbb{E}(C_{max}) = \mathbb{E}(max\{C_1, \ldots, C_n\})$$

L_{max} The **maximum lateness** is defined as

$$L_{max} = max\{L_1, \ldots, L_n\}$$

$\sum w_j C_j$ The **total weighted completion time** is defined as

$$\sum_{j=0}^{n} w_j C_j$$

$\sum w_j(1 - e^{-rC_j})$ The **discounted total weighted completion time** is defined as

$$\sum_{j=0}^{n} w_j C(1 - e^{-rC_j})$$

$\sum w_j T_j$ The **total weighted tardiness** is defined as

$$\sum_{j=0}^{n} w_j T_j$$

$\sum w_j U_j$ The **weighted number of tardy jobs** is defined as

$$\sum_{j=0}^{n} w_j U_j$$

With this notation it is possible to classify the problem we are investigating in this work. We have three identical machines available and n tasks that are **independent identically, exponentially distributed** and have **in-tree constraints**:

$$(P3|intree|\mathbb{E}(C_{max}))$$

We will also take a short look at the preemptive version:

$$(P3|intree, prmp|\mathbb{E}(C_{max}))$$

3.2 Scheduling Strategies

The key to most problems is to find a scheduling strategy that optimizes the chosen criterion. A scheduling strategy (or policy) determines which tasks are processed at what times. The strategy is feasible, if no constraints are violated. In the following we will only deal with feasible strategies.

Let T be the set of tasks (in our case an in-tree), let $n = |T|$ be the number of task and let m be the number of machines. At time t let α_S^t be the tasks that are scheduled by the strategy S. Obviously $\forall t : \alpha_S^t \subseteq T \wedge |\alpha_S^t| \leq m$. The value of the chosen optimization criterion should depend on nothing else but S.

A classification that is not shown in the above scheme is the difference between stochastic and deterministic scheduling. The classical problems in scheduling were all deterministic. In this work we are dealing with a stochastic scheduling problem. Different classes of scheduling strategies are known for the case of stochastic scheduling problems. The concrete parameters (such as processing time) of a stochastic scheduling problem are random variables. Hence there exist multiple concrete instances of the same stochastic scheduling problem that can differ and reach states that are not reached by another instance (e.g. with multiple machines and random processing times jobs may finish in different order). Therefore certain information (e.g. task i finishes before task j) will only become available after some time. Scheduling strategies can thus be classified by whether they (can) take this information into account or not.

Definition 3.1 (Static List Policy). *A **static list scheduling policy** is a policy where all tasks are ordered at the beginning. At any point in time the highest available tasks are scheduled. The decisions made at the beginning are binding throughout the total execution.*

Definition 3.2 (Dynamic Policy). *A **dynamic policy** allows the scheduler to take all prior information available into account when a new task is to be scheduled. The decisions made are only binding for a single step.*

Chapter 4

Exponentially Distributed Task Processing Times

4.1 Motivation

In deterministic scheduling models all variables needed for optimization are known beforehand. For a real world problem this is usually not the case. Stochastic scheduling models are closer to reality because they do not assume the input values to be known. A common issue in a lot of real-world planning problems is that the durations of tasks are not known a priori.

Take the development of a software as an example: A software consists of a number of modules that can be developed concurrently. The development tasks and other tasks, such as testing, depend on one another. To generate an optimal project plan one would need to know the time that the development of the components needs. Although a time can be guessed from prior experience the real time is unknown.

To be able to achieve any results on such problems, we usually assume that the expected duration is known (e.g. it can be guessed from prior projects). The expected values are then used to generate an optimal plan with respect to the expected total completion time.

Often it does not suffice to know the expected values, but one also needs to know something about the character of the underlying distribution. For continuous random distributions the character can be described by a distribution and a density function. The knowledge of the density functions allows to deduce more complex results.

One well-liked distribution is the exponential distribution because of its special properties. The exponential distribution can be easily combined with the like and – even more important – the exponential distribution is memory-less (see Lemma 4.3).

4.2 Continuous Distributions

A random variable X is continuously distributed, if it can take any value in $\mathbb{R}$ (or a continuous interval of $\mathbb{R}$). A continuous distribution can best be described by a density function f. Integrating over f yields the probability that the value of X falls in a certain range (note, that the probability of a continuous function taking a single value is zero). The probability of X taking a value below t is denoted by the distribution function F:

$$P[X \leq t] = F(t) = \int_{-\infty}^{t} f(x)dx$$

There are many well-known distributions, having different beneficial properties (such as well describing natural processes or being easy to handle).

For distributions describing random time periods the probability that the time interval is negative is zero ($\forall t \leq 0 : f(t) = 0, F(t) = \int_0^t f(x)dx$). A time distribution can be used to describe the time until an event occurs.

A time distribution is **memory-less**, if it behaves the same after some time has passed, i.e. under such distribution the probability that an event occurs in the next two minutes is the same as the probability that an event occurs in the next two minutes after two minutes have elapsed. Formally this is true for continues distributions, if

$$f(t) = \frac{f(t+t_0)}{1-F(t_0)} = \frac{f(t+t_0)}{\int_{t_0}^{\infty} f(x)dx}.$$

Another definition of the memory-less property often used is

$$P[X > t] = P[X > t + t_0 | X > t_0].$$

Lemma 4.1 (Expected Value of a Memory-Less Distribution). *If X is a randomly time distributed variable with density function f and X is memory-less, then the expected value $\mathbb{E}(X|_{t_0})$ of X after time zero is $\mathbb{E}(X) + t_0$.*

Proof. The expected value of X is

$$\mathbb{E}(X) = \int_0^{\infty} t \cdot f(t)dt$$

After some time t_0 has elapsed the expected value of $X|_{t_0}$ is

$$\mathbb{E}(X|_{t_0}) = \frac{\int_{t_0}^{\infty} t \cdot f(t)dt}{\int_{t_0}^{\infty} f(t)dt} =$$

$$\int_0^{\infty} (t+t_0) \cdot \frac{f(t+t_0)}{\int_{t_0}^{\infty} f(t)dt}dt =$$

$$\int_0^{\infty} (t+t_0) \cdot f(t)dt =$$

$$\int_0^{\infty} t \cdot f(t)dt + t_0 \int_0^{\infty} f(t)dt =$$

$$\mathbb{E}(X) + t_0$$

□

Memory-Less time distributions are very helpful in scheduling problems because they reduce the search space to discrete time points:

Lemma 4.2 (Discrete Preemption Points). *If decisions are based only on random variables with memory-less time distributions (and possible non-random variables), then there are only a discrete number of points where a preemption may occur.*

Proof. Let t_i and t_j be two times with $t_i < t_j$ and let no event occur between t_i and t_j. All information available at point t_i is the history of events and the behavior of the distributions described by their density functions. Because no events occur, the history stays the same for point t_j. Since all distributions are memory-less, the behavior of the density functions does not change. Therefore the identical information is available at time t_i and t_j and thus the scheduler would schedule the same task. A reevaluation at time t_j would not lead to a change. As a result no preemption will occur. □

Another nice property of memory-less time distributions is that a lot of times a problem reduction can be made. It makes no difference what happened before a certain set of tasks with some scheduled tasks is reached. The work and time already invested in a task does not change the expected remaining time needed.

While the above properties often allow the achievement of results, they also hint at the problem of applying the results in the real world. There surely exist a lot of tasks that are not memory-less (e.g., the building of a wall or the digging of a hole) for which the results cannot be applied. On the other hand one can argue that there exist tasks for which the memory-less property makes sense. Examples for such tasks are (under certain assumptions) the catching of a fly, reaching a certain person over the phone, and others. Some tasks also look memory-less to an extend that might make it reasonable to model them as such. The solving of a riddle, or other tasks that include searching for a solution to a problem without knowing an algorithm are examples of such tasks.

The main reason for postulating the memory-less property is certainly the nice mathematical properties of such distributions.

4.3 The Exponential Distribution

The exponential distribution is one of the most commonly used distributions. The exponential distribution is the only continues memory-less time distribution (see Theorem 2.25 in [SS01]).

Let X be an exponentially distributed random variable with parameter λ.

If the finishing time of a task is exponentially distributed with variable X, then $\frac{1}{\lambda}$ is the expected finishing time.

Let f be the density of the exponential distribution:

$$f = x \rightarrow \begin{cases} \lambda e^{-\lambda x} & \text{if } x >= 0, \\ 0 & \text{otherwise.} \end{cases}$$

Let F be the exponential distribution function:

$$F = x \rightarrow \begin{cases} 1 - e^{-\lambda x} & \text{if } x >= 0, \\ 0 & \text{otherwise.} \end{cases}$$

The relation between F and f is

$$\begin{aligned} F(x) &= \int_{-\infty}^{x} f(t)dt \\ &= \begin{cases} \int_{-\infty}^{0} 0 \cdot dt + \int_{0}^{x} \lambda \cdot e^{-\lambda t} dt = & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \\ &= \begin{cases} -e^{-\lambda t}\big|_{0}^{x} & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \\ &= \begin{cases} -e^{-\lambda x} - (-1) & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \\ &= \begin{cases} 1 - e^{-\lambda x} & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The expected processing time is $\frac{1}{\lambda}$:

$$\begin{aligned}
\mathbb{E}(X) &= \int_{-\infty}^{\infty} t \cdot f(t)dt \\
&= \int_{0}^{\infty} t \cdot \lambda \cdot e^{-\lambda t}dt \\
&= (-1)\int_{0}^{\infty} t \cdot (-\lambda) \cdot e^{-\lambda t}dt \\
&= -t \cdot e^{-\lambda t}\Big|_{0}^{\infty} - \int_{0}^{\infty}(-1) \cdot e^{-\lambda t}dt \\
&= -t \cdot e^{-\lambda t}\Big|_{0}^{\infty} - \tfrac{1}{\lambda} \cdot e^{-\lambda t}\Big|_{0}^{\infty} \\
&= 0 - 0 - \tfrac{1}{\lambda} \cdot (0 - 1) \\
&= \tfrac{1}{\lambda}
\end{aligned}$$

Among others the memory-less property of the exponential distribution makes the distribution well-liked by scheduling researchers.

Lemma 4.3 (Memory-less Property of the Exponential Distribution). *The exponential distribution is memory-less.*

Proof. We will show

$$f(t) = \frac{f(t + t_0)}{\int_{t_0}^{\infty} f(x)dx}$$

for the exponential distribution.

We first calculate the probability of an exponentially distributed X taking a value larger than t_0:

$$\int_{t_0}^{\infty} f(x)dx = \int_{t_0}^{\infty} \lambda e^{-\lambda x}dx = (-e^{-\lambda x})\Big|_{t_0}^{\infty} = (-e^{-\lambda\infty}) - (-e^{-\lambda t_0}) = e^{-\lambda t_0}$$

With that we can proceed to show that

$$\frac{f(t + t_0)}{\int_{t_0}^{\infty} f(x)dx} = \frac{\lambda e^{-\lambda(t+t_0)}}{e^{-\lambda t_0}} = \lambda e^{-\lambda(t+t_0-t_0)} = \lambda e^{-\lambda t} = f(t)$$

□

A proof that every memory-less time distribution must be exponentially distributed is given in [SS01] (Theorem 2.25).

When a number of tasks with independent identically, exponentially distributed processing times is scheduled on a number of machines the outcome is random.

Lemma 4.4 (Minimum of two exponentially distributed Variables). *Let X_1 be an exponentially distributed random variable with expected value $\frac{1}{\lambda_1}$ and let X_2 be an exponentially distributed random variable with expected value $\frac{1}{\lambda_2}$. Let $Y = min(X_1, X_2)$. Y is exponentially distributed with expected value $\frac{1}{\lambda_1+\lambda_2}$.*

Proof. Let $f_1 = \lambda_1 \cdot e^{-\lambda_1 \cdot t}$ be the density function for X_1. Let $f_2 = \lambda_2 \cdot e^{-\lambda_2 \cdot t}$ be the density function for X_2.

The density function f_{min} of Y is either the density of X_1, if X_1 is smaller than or equals X_2 or the density of X_2 otherwise:

$$f_{min}(x) = f_1(x) \cdot P[X_1 \leq X_2] + f_2(x) \cdot P[X_2 \leq X_1] = f_1(x) \cdot P[x \leq X_2] + f_2(x) \cdot P[x \leq X_1] \quad (4.1)$$

The value of $P[x \leq X_2]$ (and $P[x \leq X_1]$ analogous) can be calculated using f_2:

$$P[x \leq X_2] = \int_x^\infty f_2(t)dt = \int_x^\infty \lambda_2 \cdot e^{-\lambda_2 \cdot t} dt = (-1) \cdot e^{-\lambda_2 t} \Big|_x^\infty = 0 - (-1) \cdot e^{-\lambda_2 x} = e^{-\lambda_2 x} \quad (4.2)$$

Using 4.2 in 4.1 yields:

$$f_{min}(x) = \lambda_1 \cdot e^{-\lambda_1 \cdot x} \cdot e^{-\lambda_2 x} + \lambda_2 \cdot e^{-\lambda_2 \cdot x} \cdot e^{-\lambda_1 x} = (\lambda_1 + \lambda_2) \cdot e^{-(\lambda_1 + \lambda_2) \cdot x} \quad (4.3)$$

Hence Y is exponentially distributed with parameter $\lambda_1 + \lambda_2$. □

Lemma 4.5 (Minimum of n independent identically, exponentially distributed Variables). *When n tasks with independent identically, exponentially distributed processing times are scheduled on n machines, where the expected time of a task to finish is $\frac{1}{\lambda}$, then the expected time of the first task to finish is $\frac{1}{n \cdot \lambda}$.*

Proof. The proof follows directly from Lemma 4.4. Combining the first two of the n tasks yields an exponential distributed task with expected finishing time $\frac{1}{\lambda + \lambda}$. Continuing iteratively leads to the stated result. □

In the following we will look at scheduling with three machines. If three machines are working concurrently at a given in-tree after expected time $\frac{1}{3} \cdot \frac{1}{\lambda}$ the first machine should finish. If the constraints are such that only two of the machines can work at tasks, the first machine is expected to finish after time $\frac{1}{2} \cdot \frac{1}{\lambda}$. If only a single machine can work at tasks, the machine is expected to finish after time $\frac{1}{\lambda}$. Since all these times share a common factor $\frac{1}{\lambda}$, we will assume that $\frac{1}{\lambda} = 1$ for subsequent work. All results can be easily extended to an arbitrary value for λ by multiplying with $\frac{1}{\lambda}$.

The expected value of the maximum of a number of independent identically, exponentially distributed variables follows directly from Lemma 4.5:

Lemma 4.6 (Maximum of n independent identically, exponentially distributed Variables). *When n tasks with independent identically, exponentially distributed processing times are scheduled on n machines, where the expected time of a task to finish is $\frac{1}{\lambda}$, then the expected time of the last task to finish is $\frac{1}{\lambda} \cdot H_n$, where H_n denotes the n-th harmonic number.*

Proof. The proof follows directly from Lemma 4.5 and the memory-less property of the exponential distribution. The expected value of the finishing time of the first tasks is (by Lemma 4.5): $\frac{1}{n \cdot \lambda}$. After time $\frac{1}{n \cdot \lambda}$ the expected value of the finishing time of the first remaining task is $\frac{1}{n \cdot \lambda} + \frac{1}{(n-1) \cdot \lambda}$. Continuing these steps leads to the expected finishing time of the last process to finish, which is

$$\frac{1}{n \cdot \lambda} + \frac{1}{(n-1) \cdot \lambda} + \ldots + \frac{1}{2 \cdot \lambda} + \frac{1}{\lambda} = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{1}{i} = \frac{1}{\lambda} H_{n \cdot}$$

□

4.4 Other Distributions Besides the Exponential

Before we restrict ourselves for the rest of this work to the exponential distribution, we will take a short look at another distribution which shares an important property of the exponential distribution. The main properties of the exponential distribution used throughout this work are the memory-less property and the expected minimum of n variables. Hence any distribution that has these properties and that is a time distribution can be used. There is no other continuous memory-less distribution, but there is a discrete distribution that is memory-less.

The **geometric distribution** is the only discrete distribution that has the memory-less property. For discrete random variable X with range $\mathbb{N}^+$, the geometric distribution with parameter $0 \leq p \leq 1$ is defined through

$$P[X = t] = p \cdot (1-p)^{t-1},$$

which is the equivalent to the density function. The distribution function's equivalent is

$$P[X \leq t] = \sum_{i=1}^{t} P[X = i] = \sum_{i=0}^{t-1} p \cdot (1-p)^i = p\frac{1-(1-p)^t}{1-(1-p)} = 1-(1-p)^t.$$

The following lemma helps us to determine the expected value of the geometric distribution.

Lemma 4.7.

$$\sum_{i=0}^{\infty} i \cdot p^i = \frac{p}{(1-p)^2}$$

Proof. We know the geometric sum $\sum_{i=0}^{n} p^i = \frac{1-p^{n+1}}{1-p}$, hence $\sum_{i=0}^{\infty} p^i = \frac{1}{1-p}$.
We go the reverse way for the proof and start with the closed formula:

$$\frac{p}{(1-p)^2} = p \cdot \frac{1}{1-p} \cdot \frac{1}{1-p} = p \cdot \Big(\sum_{i=0}^{\infty} p^i\Big) \cdot \Big(\sum_{i=0}^{\infty} p^i\Big) =$$

$$p \cdot \Big(\sum_{i=0}^{\infty} \sum_{j=0}^{i} p^j \cdot p^{i-j}\Big) = p \cdot \Big(\sum_{i=0}^{\infty} (i+1)p^i\Big) = \sum_{i=0}^{\infty} (i+1)p^{i+1} = \sum_{i=0}^{\infty} i \cdot p^i$$

□

The expected value of the geometric distribution is therefore

$$\sum_{t=1}^{\infty} t \cdot p \cdot (1-p)^{t-1} = \frac{p}{1-p} \sum_{t=0}^{\infty} t \cdot (1-p)^t = \frac{p}{1-p} \cdot \frac{1-p}{(1-(1-p))^2} = \frac{1}{p}.$$

The geometric distribution is memory-less because

$$P[X = t + t_0 | X > t_0] = \frac{P[X = t + t_0]}{P[X > t_0]} = \frac{p \cdot (1-p)^{t+t_0-1}}{1 - P[X \leq t_0]}$$

$$= \frac{p \cdot (1-p)^{t+t_0-1}}{1-(1-(1-p)^{t_0})} = \frac{p \cdot (1-p)^{t+t_0-1}}{(1-p)^{t_0}} = p \cdot (1-p)^{t-1} = P[X = t].$$

Any memory-less discrete distribution with range $\mathbb{N}^+$ must be a geometric distribution:

Lemma 4.8 (The Memory-Less Property of Discrete Distributions). *A discrete distribution X with range $\mathbb{N}^+$ and the memory-less property $P[X > n+m | X > m] = P[X > n]$ is geometrically distributed.*

Proof. Let X be any memory-less discrete distribution with range $\mathbb{N}^+$. It follows that

$$P[X > n + m] = P[X > n + m | X > m] \cdot P[X > m] = P[X > n] \cdot P[X > m].$$

Induction yields

$$\forall n \in \mathbb{N}^+ : P[X > n] = P[X > 1 + (n-1)] = P[X > 1] \cdot P[X > n-1]$$

$$= P[X > 1] \cdot (P[X > 1])^{n-1} = (P[X > 1])^n.$$

Let $P[X > 1] = 1 - P[X \leq 1] = 1 - P[X = 1] = 1 - p$ with $0 \leq p \leq 1$. It follows that $P[X > n] = (1 - p)^n$. Therefore,

$$P[X = n] = 1 - P[X > n] - P[X \leq n - 1] = 1 - P[X > n] - (1 - P[X > n - 1])$$
$$= (1 - p)^{n-1} - (1 - p)^n = (1 - p)^{n-1} \cdot (1 - (1 - p)) = p \cdot (1 - p)^{n-1}.$$

As a result, X must be geometrically distributed with parameter p. □

Lemma 4.1 can be easily extended to discrete memory-less distributions.

The minimum $Y = min(X_1, X_2)$ of two independent identically, geometrically distributed random variables X_1 and X_2 with parameter p_1 and p_2 is a random variable that is geometrically distributed with parameter $p_1 + p_2 - p_1p_2$:

$$P[Y = t] =$$
$$P[X_1 = t] \cdot P[X_2 = t] + P[X_1 = t] \cdot P[X_2 > t] + P[X_2 = t] \cdot P[X_1 > t] =$$
$$p_1(1 - p_1)^{t-1} \cdot p_2(1 - p_2)^{t-1} + p_1(1 - p_1)^{t-1} \cdot (1 - p_2)^t + p_2(1 - p_2)^{t-1} \cdot (1 - p_1)^t =$$
$$\big((1 - p_1)(1 - p_2)\big)^{t-1} \cdot \big(p_1p_2 + p_1(1 - p_2) + p_2(1 - p_1)\big) =$$
$$\big(1 - (p_1 + p_2 - p_1p_2)\big)^{t-1} \cdot \big(p_1p_2 + p_1 - p_1p_2 + p_2 - p_2p_1)\big) =$$
$$\big(1 - (p_1 + p_2 - p_1p_2)\big)^{t-1} \cdot \big(p_1 + p_2 - p_1p_2)\big)$$

The expected value of the minimum of two geometrically distributed variables is hence $\mathbb{E}(Y) = \frac{1}{p_1+p_2-p_1p_2}$. If $p = p_1 = p_2$, then $\mathbb{E}(Y) = \frac{1}{2p-p^2} = \frac{1}{p} \cdot \frac{1}{2-p}$. Therefore the new expected value is not linearly dependent on the original one. As a result, although the geometric distribution is the discrete counterpart to the exponential distribution, it cannot be used in its place for the results of this thesis.

Applying a distribution that does not have the memory-less property seems to result in an even larger search-space. Whether the geometric distribution allows different results might be an interesting problem.

For the rest of this work we only consider independent identically, exponentially distributed task processing times.

Chapter 5

In-Tree Constraints

When scheduling with in-tree constraints the dependencies between the tasks form an in-tree. This is the case, if each task constraints at most one other task and each task can be constrained by an arbitrary number of tasks. The problem is hierarchically structured with a single ending task that is the last one to be processed.

For simplicity we assume that the constraints form a single tree. If not, we simply add a new root that has all trees of the forest as children (the expected time of the forest is the expected time of the constructed tree minus the expected time of processing the new root task, since the new root task does not need to be finished).

5.1 An Introductory Example

Amount	Measure	Ingredient – Preparation Method
1	cup	Sugar
2/3	cup	All-purpose flour (divided 1/3 and 1/3 cups)
2	large	Eggs – lightly beaten
1 1/3	cups	Sour cream
1	teaspoon	Vanilla extract
3	cups	Fresh or frozen blackberries – thaw if frozen
1		Unbaked 9-inch pastry shell
1/3	cup	Firmly packed brown sugar
1/4	cup	Chopped toasted pecans
3	tablespoons	Softened butter
		Whipped cream, Fresh whole berries (opt.)

1. Preheat oven to 400 degrees.
2. Mix together sugar, 1/3 cup flour, eggs, sour cream and vanilla. Blend until smooth.
3. Gently fold in blackberries.
4. Spoon mixture into pastry shell.
5. Bake 30-35 minutes or until center is set.
6. Combine remaining 1/3 cup flour, brown sugar, pecans and softened butter. Mix together well.
7. Sprinkle over hot pie.
8. Return pie to oven for 10 minutes or until golden brown.
9. Remove from oven and cool on wire rack.
10. Garnish with whipped cream and whole berries if desired.

Figure 5.1: Blackberry Cream Pie Recipe (see http://members.aol.com/Jimg005/pie6.html)

Take a recipe for a Blackberry Cream Pie (see Figure 5.1) for an example of a hierarchical problem. Assume

that every step listed is expected to take the same time (“long” steps can be divided, s.t. all steps take about the same expected time). If the pie is prepared by more than one person, tasks can be executed in parallel. Figure 5.2 shows two trees, the first being the sole structure displaying what needs to be done in sequence and what can be done in parallel (the node ’P’ represents the preparing of the whole berries), the second tree has been rearranged, such that each task takes about the same amount of time (the whole thing is abused a bit as to result in a ’nicer’ tree – baking for 30–35 minutes can hardly be modeled with an exponential distribution).

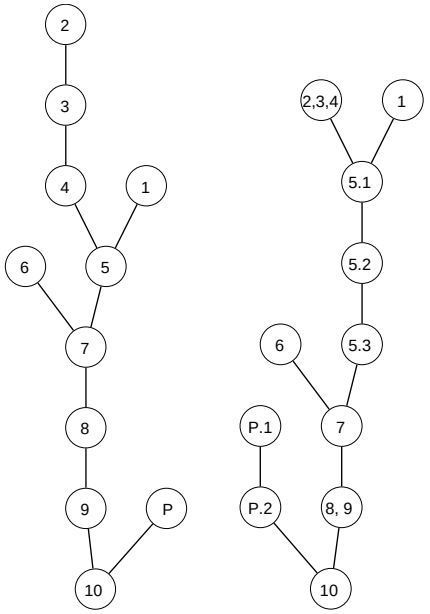


Figure 5.2: Hierarchical Structure of Preparing a Blackberry Cream Pie

Under the assumption that the duration of each step of the tree depicted in Figure 5.2 (b) is an independent identically, exponentially distributed random value, the tree is an instance of $(P3|intree|\mathbb{E}(C_{max}))$. Assume that the expected value for a single task is $\frac{1}{\lambda} = 10\,min$, then there are four possibilities of an initial schedule (where two are equivalent):

Task of Person A	Task of Person B	Task of Person C	Expected Time
2,3,4	6	P.1	1h, 17min 53s
1	6	P.1	1h, 17min 53s
1	2,3,4	6	1h, 16min, 5s
1	2,3,4	P.1	1h, 16min, 4s

As the calculated results show, it is advantageous to start preparing the whole berries before starting the second layer by about one second (although this does not seem obvious because task 6 is represented by a higher node).

5.2 Calculating an Optimal Solution for a Smaller Example

To show how the expected times for a schedule are calculated we will take a smaller example tree that is shown in Figure 5.3.

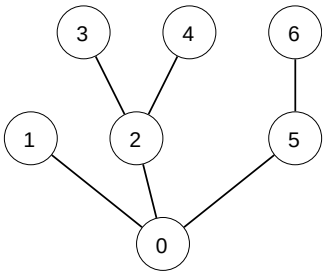


Figure 5.3: Small Tree Example

The expected time can be calculated by a recursive algorithm (see section 5.3). The scheduling process is divided into time intervals, at the end of each a machine finishes a task and the scheduler assigns the machine to a new task (if available). This points are often called decision points. For a tree B there are $|B|$ decision points. From each decision point there are one or more possibilities to choose a new schedule. Once the schedule is chosen, the machines start (or continue) working on their tasks until one machine finishes. The decisions are assumed to take no time (there are no setup times). Which machine finishes first is a random event. Figure 5.4 shows a DAG of subtrees with scheduled nodes that describes this process. The dark-framed DAG-nodes are those where the scheduler decides on the next step, from the light-framed DAG-nodes a random decision is taken. The expected length of the interval between two decision points depends on the number of machines working (see Lemma 4.5 in the previous chapter). The goal is to choose the schedules in such a way that the expected sum of the interval lengths is minimized.

5.3 Calculating the Solution Value of a Strategy

In the following we will be concerned with the scheduling problem $(P3|intree|\mathbb{E}(C_{max}))$. The number of machines is fixed to 3. The optimization criterion will always be the expected total processing time. The individual task processing times will all be independent identically, exponentially distributed with parameter λ. From chapter 4 we know that we can assume $\lambda = 1$ and multiply the result with $\frac{1}{\lambda}$. Therefore a description of a concrete instance of the problem will only need to include the precedence tree with all the tasks. The expected total processing depends only upon the strategy S chosen and the in-tree B of the constraints. From 4.2 we know that there are only $n = |B|$ points in time where scheduling decisions can be made by the strategy. In the following let $\alpha_S^B(t)$ be the set of leaves chosen for processing at time t. If the problem requires non-preemptive scheduling, if there is more than one machine available, and if the problem is a stochastic one, then α does not only depend on the time t, but rather on the history of the schedule (which tasks have finished, or have become available) and on which machines are still processing other tasks. For $(P3|intree|\mathbb{E}(C_{max}))$ the history can simply be described by the remaining tree B' and by the set of currently scheduled tasks β (e.g. as marked leaves). α_S can then be seen as a function of B' and β, where at the beginning $B' = B$ and $\beta = \emptyset$: $\alpha_S(B', \beta)$ are the tasks which are to be scheduled. The time does not play a role any more because of the memory-less property of the exponential distribution.

We will also use the function α to identify a strategy (leaving out the subscript S if it is clear from context). From the previous chapter we know the expected time of the first of m machines to finish. Let there be n tasks. An instance of a schedule will follow this scheme:

1. Schedule at least three tasks that are leaves of the tree.

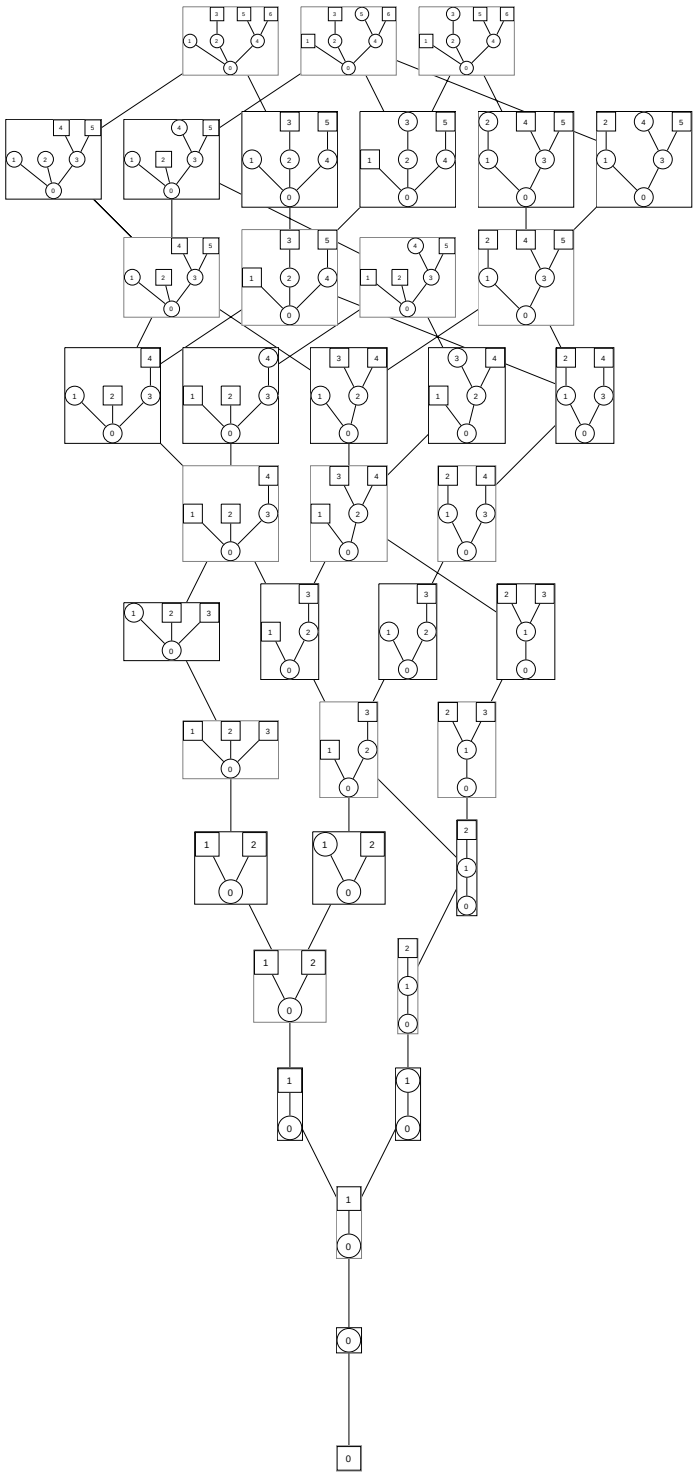


Figure 5.4: The Calculation DAG for the Example Tree in Figure 5.3. (The subtrees are ordered left to right in ascending expected total processing time. Each DAG node is represented with the subtree it stands for. In these subtrees, square nodes denote scheduled leaves.)

2. After some time t_i one task (the i-th) is finished and the machine can be assigned to another leaf, if there is still one left (some task that has been an inner node might have become available now because all its children have been finished).

3. The previous step is iterated until the last task finishes.

The total time is

$$T = \sum_{1}^{n} t_i$$

Let the n tasks be assigned numbers 1 through n. In a concrete instance of a schedule task i will be finished as $p(i)$-th where p is a permutation over $\{1, \ldots, n\}$. For a given in-tree B let $CP(B)$ (Concrete instance Permutations) be the set of all permutations that correspond to a concrete instance of a schedule (they are feasible, if for any nodes i and j: if j is an ancestor of i, then j will never occur before i in any permutation in $CP(B)$: $\forall i, j \in B : \exists path(i, j) \Rightarrow \forall p \in CP(B) : p(i) < p(j)$). We will call a feasible permutation an **execution path**.

The expected time of the above process is determined by the expected finishing times in step 2. Since all tasks are independent identically distributed, the expected time $\mathbb{E}(t_i)$ depends only on the number of machines that were working at the time task i was finished.

A recursive algorithm for calculation of the total expected processing time, given a strategy function $\alpha_S(B, \beta)$ is given as Algorithm 1.

Algorithm 1 $\mathbf{Frame(B, \beta, \alpha_S)}$

```
1: proc  Frame(B, β, α_S) :
2: if |B| = 1 then
3:    return(1)
4: else
5:    σ ⟵ α_S(B, β)
6:    s ⟵ 0
7:    for all i ∈ σ do
8:       β' ⟵ σ \ {i}
9:       B' ⟵ B \ {i}
10:      s ⟵ s + Frame(B', β', α_S)
11:   end for
12:   s ⟵ (s + 1)/(|σ|)
13:   return(s)
14: end if
```

Given an execution path $p \in CP(B)$, we can determine the tasks i_1 and i_2 (permuted numbers) after which there are only one or two leaves left. Let $s_2(p) = i_2$ and $s_1(p) = i_1$. s_1 and s_2 are dependent on the tree B and the execution path p.

Let $P_\alpha^B[p]$ be the probability that the execution path described by p occurs under the strategy α and tree B.

Theorem 5.1 (Expected Processing Time for Strategy α). *The expected processing time $\mathbb{E}(T_\alpha)$ for a scheduling problem with n independent identically, exponentially distributed tasks with individual expected processing time $\frac{1}{\lambda}$ and constraints defined by the in-tree B is*

$$\mathbb{E}(T_\alpha) = \frac{1}{\lambda} \cdot \sum_{p \in CP} P_\alpha^B[p] \cdot \left(s_2(p) \cdot \frac{1}{3} + (s_1(p) - s_2(p)) \cdot \frac{1}{2} + (n - s_1(p)) \right) \tag{5.1}$$

Proof. The proof follows from the above definitions and by summing over all possible concrete schedule instances:

$$\mathbb{E}(T_\alpha) = \sum_{p \in CP} P_\alpha^B[p] \cdot \Big(\text{Expected time of concrete schedule instance corresponding to } p\Big)$$

$$= \sum_{p \in CP} P_\alpha^B[p] \cdot \left(\sum_{i=1}^{s_2(p)} \frac{1}{3} \cdot \frac{1}{\lambda} + \sum_{i=s_2(p)+1}^{s_1(p)} \frac{1}{2} \cdot \frac{1}{\lambda} + \sum_{i=s_1(p)+1}^{n} \frac{1}{1} \cdot \frac{1}{\lambda} \right)$$

$$= \frac{1}{\lambda} \cdot \sum_{p \in CP} P_\alpha^B[p] \cdot \left(\sum_{i=1}^{s_2(p)} \frac{1}{3} + \sum_{i=s_2(p)+1}^{s_1(p)} \frac{1}{2} + \sum_{i=s_1(p)+1}^{n} 1 \right)$$

$$= \frac{1}{\lambda} \cdot \sum_{p \in CP} P_\alpha^B[p] \cdot \left((s_2(p) - 1 + 1) \cdot \frac{1}{3} + (s_1(p) - s_2(p) - 1 + 1) \cdot \frac{1}{2} + (n - s_1(p) - 1 + 1) \right)$$

$$= \frac{1}{\lambda} \cdot \sum_{p \in CP} P_\alpha^B[p] \cdot \left(s_2(p) \cdot \frac{1}{3} + (s_1(p) - s_2(p)) \cdot \frac{1}{2} + (n - s_1(p)) \right)$$

$$= \frac{1}{\lambda} \cdot \sum_{p \in CP} P_\alpha^B[p] \cdot \left(n - \frac{1}{6} \cdot s_2(p) - \frac{1}{2} \cdot s_1(p) \right)$$

□

Equation 5.1 can also be rewritten such that the emphasis lies more on the number of times three, two, or only one machine can work at the tree.

Let TWO be the number of the task after which there are only two leaves left (the concrete value of $s_2(p)$ for a concrete execution path p), and let ONE be the number of the task after which there is only one leaf left (the concrete value of $s_1(p)$ for a concrete execution path p). Clearly for a given tree B and a strategy α TWO and ONE are random variables.

Corollary 5.1. *Let $\mathbb{E}(TWO)$ ($\mathbb{E}(ONE)$) be the expected value of TWO (ONE) under strategy α and in-tree B. The expected processing time $\mathbb{E}(T_\alpha)$ for a scheduling problem with n exponentially distributed tasks with individual expected finishing time $\frac{1}{\lambda}$ and constraints defined by the in-tree B is*

$$\mathbb{E}(T_\alpha) = \frac{1}{\lambda} \cdot \left(n - \frac{1}{6} \cdot \mathbb{E}(TWO) - \frac{1}{2} \cdot \mathbb{E}(ONE) \right) \tag{5.2}$$

or

$$\mathbb{E}(T_\alpha) = \frac{1}{\lambda} \cdot \left(\frac{1}{3} \cdot \mathbb{E}(TWO) + \frac{1}{2} \cdot (\mathbb{E}(ONE) - \mathbb{E}(TWO)) + 1 \cdot (n - \mathbb{E}(ONE)) \right) \tag{5.3}$$

where $\mathbb{E}(TWO)$ is the expected number of steps with three machines, $(\mathbb{E}(ONE) - \mathbb{E}(TWO))$ is the expected number of steps with two machines, and $(n - \mathbb{E}(ONE))$ is the expected number of steps with one machine.

Proof. The expected values of ONE and TWO are:

$$\mathbb{E}(TWO) = \sum_{p \in CP} P_\alpha^B[p] \cdot s_2(p)$$

$$\mathbb{E}(ONE) = \sum_{p \in CP} P_\alpha^B[p] \cdot s_1(p)$$

Therefore from Theorem 5.1:

$$\mathbb{E}(T_\alpha) = \frac{1}{\lambda} \cdot \left(\sum_{p \in CP} P_\alpha^B[p] \cdot \left(\frac{1}{3} \cdot s_2(p) + \frac{1}{2} \cdot (s_1(p) - s_2(p)) + (n - s_1(p)) \right) \right) =$$

$$\frac{1}{\lambda} \cdot \left(\sum_{p \in CP} P_\alpha^B[p] \cdot \left(-\frac{1}{6} \cdot s_2(p) - \frac{1}{2} \cdot s_1(p) + n \right) \right) =$$

$$\frac{1}{\lambda} \cdot \left(n \cdot \sum_{p \in CP} P_\alpha^B[p] - \frac{1}{6} \cdot \sum_{p \in CP} P_\alpha^B[p] \cdot s_2(p) - \frac{1}{2} \cdot \sum_{p \in CP} P_\alpha^B[p] \cdot s_1(p) \right) =$$

$$\frac{1}{\lambda} \cdot \left(n - \frac{1}{6} \cdot \mathbb{E}(TWO) - \frac{1}{2} \cdot \mathbb{E}(ONE) \right)$$

The second equation is a simple transformation of the first. □

As can be seen from Theorem 5.1 and Corollary 5.1 the value $\frac{1}{\lambda}$ is always an outer constant factor. We will therefore often use 1 as the expected value of the exponential distributions of the task processing times. The corresponding results only need to be multiplied by the real expected value (e.g. $\frac{1}{\lambda}$ ="one hour", $\frac{1}{\lambda}$ ="3 days", ...).

Chapter 6

Calculating the Optimal Schedule for $(P3|intree|\mathbb{E}(C_{max}))$

6.1 A Simple Recursive Algorithm

A trivial algorithm to calculate the total expected processing time would calculate the probabilities $P_\alpha^B[p]$ of every execution path $p \in CP(B)$ and multiply by the expected processing time $s_2(p) \cdot \frac{1}{3} + (s_1(p) - s_2(p)) \cdot \frac{1}{2} + (n - s_1(p))$.

Because all processing times are independently distributed, given a set of scheduled tasks every task has the same probability of finishing first. An algorithm can simply calculate the optimal expected processing time in the following steps:

1. For each possible set of leaves to schedule,

   for each leaf in that set,

   remove the leaf from the tree and recursively calculate the optimal value for the case that this leaf is finished first.

2. Add all cases, divide by the number of cases and add the minimal expected finishing time of the first node (1, $\frac{1}{2}$, or $\frac{1}{3}$, see chapter 4).

3. Select the best of all schedules.

The central procedure is given as Algorithm 2.

The algorithm that starts the calculation and receives the result in s_{min} looks like

$s_{min} \longleftarrow \infty$
for all $\beta \subseteq leaves(B)$ with $|\beta| = 2 \vee (|\beta| < 2 \wedge \beta = leaves(B)$ **do**
  $s \longleftarrow Opt_{recursive}(B, \beta)$
  if $s < s_{min}$ **then**
    $s_{min} \longleftarrow s$
  end if
end for

We will not give the time complexity of this algorithm here, but first proceed with a small modification. An upper bound on the space complexity is easy to prove.

Lemma 6.1. *$(P3|intree|\mathbb{E}(C_{max}))$ belongs to $PSPACE$.*

Proof. The above algorithm solves $(P3|intree|\mathbb{E}(C_{max}))$ optimally. In each call to $Opt_{recursive}$ the size of the tree is reduced by one node. Hence the call stack is at most $|B|$ deep. For each call the parameters

Algorithm 2 $\mathbf{Opt_{recursive}(B, \beta)}$

Require: $|\beta| = 2 \vee (|leaves(B)| < 3 \wedge |leaves(B)| - 1 \leq |\beta| \leq |leaves(B)|)$

```
1: proc  Opt_recursive(B, β) :
2: if |leaves(B)| = 1 then
3:    return(|B|)
4: else
5:    if |leaves(B)| = 2 then
6:       σ ⟵ leaves(B)
7:       s ⟵ 0
8:       for all i ∈ σ do
9:          β' ⟵ σ \ {i}
10:         B' ⟵ B \ {i}
11:         s ⟵ s + Opt_recursive(B', β')
12:      end for
13:      s ⟵ (s + 1)/(|σ|)
14:      return(s)
15:   else          /* |leaves(B)| ≥ 3 */
16:      s_min ⟵ ∞
17:      for all l ∈ leaves(B) \ β do
18:         σ ⟵ β ∪ {l}
19:         s ⟵ 0
20:         for all i ∈ σ do
21:            β' ⟵ σ \ {i}
22:            B' ⟵ B \ {i}
23:            s ⟵ s + Opt_recursive(B', β')
24:         end for
25:         s ⟵ (s + 1)/(|σ|)
26:         if s < s_min then
27:            s_min ⟵ s
28:         end if
29:      end for
30:      return(s_min)
31:   end if
32: end if
```

and some sets of at most constant size must be remembered. As a result, the maximal space needed is $\mathcal{O}(|B|^2)$. □

6.2 A Dynamic Programming Algorithm

It is easy to see that Algorithm 2 visits a lot of trees multiple times. We can use a hash table to remember optimal schedules for the pair $\langle B, \beta \rangle$. The algorithm could thus be altered to look in a central hash table H. The procedure $Opt_{DP}(B, \beta)$ for this algorithm is given as Algorithm 3.

Algorithm 3 $\mathbf{Opt_{DP}(B, \beta)}$

Require: $|\beta| = 2 \vee (|leaves(B)| < 3 \wedge |leaves(B)| - 1 \leq |\beta| \leq |leaves(B)|)$

```
1: proc  Opt_DP(B, β) :
2: if H contains ⟨B, β⟩ then
3:    return(H(⟨B, β⟩))
4: end if
5: if |leaves(B)| = 1 then
6:    return(|B|)
7: else
8:    if |leaves(B)| = 2 then
9:       σ ⟵ β
10:      s ⟵ 0
11:      for all i ∈ σ do
12:         β' ⟵ σ \ {i}
13:         B' ⟵ B \ {i}
14:         s ⟵ s + Opt_DP(B', β')
15:      end for
16:      s ⟵ (s + 1)/(|σ|)
17:      H ⟵ H ∪ (⟨B, β⟩, s)
18:      return(s)
19:   else          /* |leaves(B)| ≥ 3 */
20:      s_min ⟵ ∞
21:      for all l ∈ leaves(B) \ β do
22:         σ ⟵ β ∪ {l}
23:         s ⟵ 0
24:         for all i ∈ σ do
25:            β' ⟵ σ \ {i}
26:            B' ⟵ B \ {i}
27:            s ⟵ s + Opt_DP(B', β')
28:         end for
29:         s ⟵ (s + 1)/(|σ|)
30:         if s < s_min then
31:            s_min ⟵ s
32:         end if
33:      end for
34:      H ⟵ H ∪ (⟨B, β⟩, s_min)
35:      return(s_min)
36:   end if
37: end if
```

The number of steps performed depends on the number of subtrees visited. With the usage of the hash table no subtree is visited twice with the same combination of leaves (we could equally use a two step hashing, first from the tree to a hash table that hashes from a set of leaves to the minimal expected processing time).

Each subtree B' of B is visited at most $\mathcal{O}(leaves(B')^2) = \mathcal{O}(|B'|^2) = \mathcal{O}(|B|^2)$ times. The number of subtrees of a tree B depends on the tree structure, because the structure determines what subsets of the set of all permutation over $\{1, \ldots, |B|\}$ are feasible (no non-leaves are removed at any point). An upper bound for the number of different subtrees is the number of permutations over $\{1, \ldots, |B|\}$, which is $|B|!$. These considerations lead to the following lemma:

Lemma 6.2 (Complexity of Algorithm 3). *The running time of Algorithm 3 is in* $\mathcal{O}(|B'|^2 \cdot |B|!)$.

Proof. See considerations above. □

6.3 Excluding Isomorphic Subtrees

Algorithm 3 can be further optimized by removing isomorphic subtrees from the list of considered subtrees and by further ignoring sets of scheduled leaves of any subtree that are isomorphic to already seen sets of scheduled leaves. To do this we need a unique representation of a subtree with some scheduled leaves.

Observation 6.1 (Unique Representation of Rooted, Unordered Trees). *Rooted, unordered trees can be represented uniquely by recursively sorting the children by their in-degrees. If two children have the same in-degree, the degrees of the children's children are compared recursively. Scheduled leaves can be included by defining their in-degree as* -1 *(while other leaves' in-degree is* 0*).*

The algorithm for sorting a tree is given as Algorithm 5. We assume in-trees and that the children of a node n are ordered and can be retrieved by $child_i^n$.

Algorithm 4 $\mathbf{sortTree_{compare}(a, b)}$

```
1: if d_in(a) < d_in(b) then
2:    return(−1)
3: else if d_in(a) > d_in(b) then
4:    return(1)
5: else
6:    for i from 1 to d_in(a) do
7:       if sortTree_compare(child_i^a, child_i^b) < 0 then
8:          return(−1)
9:       else
10:         if sortTree_compare(child_i^a, child_i^b) > 0 then
11:            return(1)
12:         end if
13:      end if
14:   end for
15:   return(0)
16: end if
```

Algorithm 5 $\mathbf{sortTree(B)}$

```
1: r ⟵ root(B)
2: for all c ∈ in(r) do          /* all children of r */
3:    sortTree(B|_c)             /* sort subtree with root c */
4: end for
5: Sort the children of r with sortTree_compare().
```

Figure 6.1 shows an example of a tree and the same tree recursively sorted. The trees unique representation is $(2, 1, 3, 0, -1, -1, 2, 0, 2, 0, -1)$.

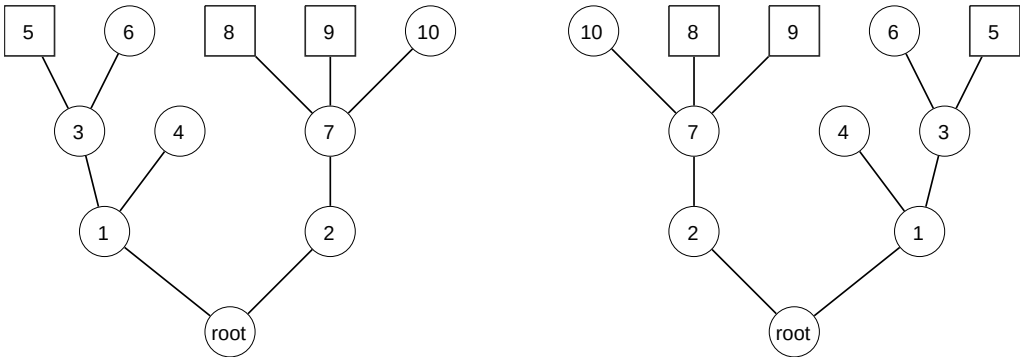


Figure 6.1: Example of a Tree and Same Tree Reordered

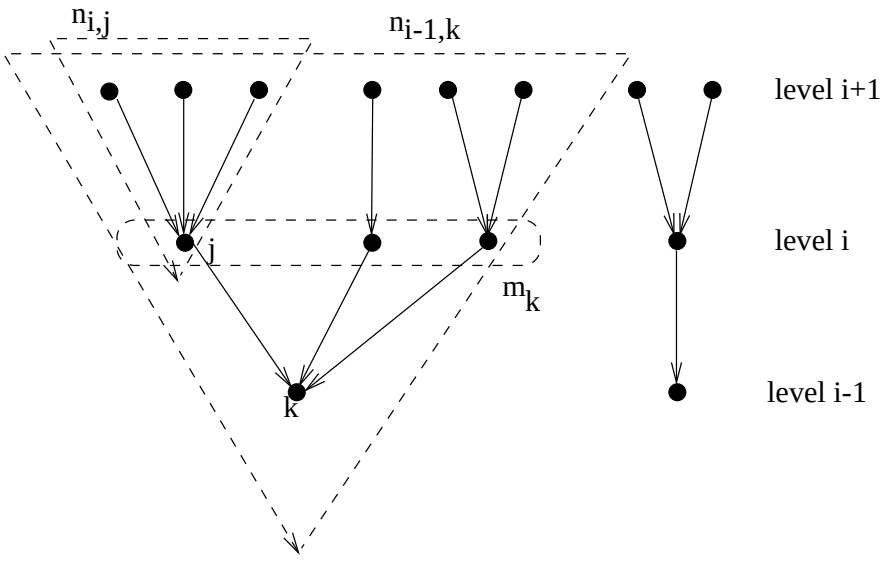


Figure 6.2: Sorting a Tree for a Unique Representation

Lemma 6.3 (Sorting a Tree for a Unique Representation). *Sorting a tree to extract the unique representation for comparison with other trees takes time $\mathcal{O}(n^2 \cdot \log(n))$.*

Proof. Let the tree have n nodes and height h. Let l_i be the number of nodes at level i (where $l_0 = 1$ is the root level and l_h is the highest level).

Let there be $n_{i,j}$ nodes in the subtree of node j at level i, then $\sum_{j=1}^{l_i} n_{i,j} = \sum_{j=i}^{h} l_j$.

Comparing a child j with another child j' of a node k at level $i-1$ takes at most $\min\{n_{i,j}, n_{i,j'}\} \leq n_{i,j}$. Suppose k has m_k children. With a sorting algorithm (such as mergesort) that makes $\mathcal{O}(nlog(n))$ comparisons ($log(n)$ per item), the sorting of the children of node k takes (see Figure 6.2)

$$\sum_{j=1}^{m_k} \log(m_k) \cdot n_{i,j} = \log(m_k) \cdot \sum_{j=1}^{m_k} n_{i,j} = \log(m_k) n_{i-1,k}$$

The number of operations t_{i-1} needed for sorting the children of the nodes at level $i-1$ is:

$$t_{i-1} = \sum_{k=1}^{l_{i-1}} \log(m_k) n_{i-1,k} \leq \log(l_{i-1}) \sum_{k=1}^{l_{i-1}} n_{i-1,k} \leq \log(l_{i-1}) \sum_{k=i-1}^{k} l_k \leq n \log(n)$$

Since there are at most n levels the sum over all levels is $\mathcal{O}(n^2 \log(n))$. □

This bound is not very tight and we conjecture that the real bound for sorting a tree is $\mathcal{O}(n \log(n))$.

We can prove a slightly better bound of $\mathcal{O}(n^2)$:

Lemma 6.4 (Sorting a Tree for a Unique Representation 2). *Sorting a tree to extract the unique representation for comparison with other trees takes time $\mathcal{O}(n^2)$.*

Proof. Let B be a tree that is sorted to the unique representation, let $n = |B|$. Let $sortOps(v)$ be the number of times the in-degree of node v was compared to the in-degree of another node. Let v_{max} be a node with $sortOps(v_{max}) = \max_v\{sortOps(v)\}$ and let $s_{max} = sortOps(v_{max})$.

Let v_{max} have k ancestors, ancestor 1 being the direct parent and ancestor k being the root. Further, let m_i be the number of comparisons of v_{max} that occurred during the sorting of the children of the i-th ancestor, and let n_i be the number of children of ancestor i. Since we use mergesort, the number of comparisons of the subtree including v_{max} can be at most $\log_2(n_i)$, therefore $m_i \leq \log_2(n_i)$. The maximal number of comparisons per node is then:

$$s_{max} = \sum_{i=1}^{k} m_i \leq \sum_{i=1}^{k} \log_2(n_i) \leq \sum_{i=1}^{k} n_i \leq n$$

As a result the total number of comparisons per node is $\mathcal{O}(n)$, and the total running time of the sorting algorithm is $\mathcal{O}(n^2)$ □

We will represent all isomorphic trees with a tree that has been sorted with Algorithm 5 and then labeled with a simple DFS. In the DAG of all (non-isomorphic) trees an edge will lead from a tree B to a tree B', if $B' = B \setminus \{a\}$ for some node a. For each edge from B to a smaller tree we need to construct $B' = B \setminus \{a\}$ and sort it. Running a simple DFS we can determine the mapping of node numbers from B to B'.

An outline of the algorithm that excludes isomorphic subtrees is given as Algorithm 6, the details are given as Algorithms 7, 8, and 9.

For simplicity we have left out the special cases, where there are only two or less leaves left in a subtree. Also note that there might be more than one optimal set of leaves to schedule, hence we might want to return a set of optimal leaf-triples in Algorithm 9.

Obviously the third part of the algorithm takes time $\mathcal{O}(leaves(B)^2) = \mathcal{O}(n^2)$.

Algorithm 6 $\mathbf{Opt_{DP,ISO}(B)}$

1: Build the DAG D of all isomorphic subtrees. Label the edges between each subtree with the leaf that was removed and a mapping of the node numbers.
2: Calculate schedule for minimal expected processing time for all nodes in D
3: Calculate minimal expected processing time for B and corresponding schedule and return

Algorithm 7 Part 1 of $\mathbf{Opt_{DP,ISO}(B}, \beta)$

1: $\sigma \longleftarrow \{B\}$
2: $D \longleftarrow$ empty DAG
3: $p \longleftarrow$ new node for B
4: insert p in D
5: **while** $|\sigma| > 0$ **do**
6: $\quad\sigma' \longleftarrow \emptyset$
7: $\quad$**for all** B' in σ **do**
8: $\quad\quad p \longleftarrow$ node in D for B'
9: $\quad\quad$**for all** $a \in leaves(B')$ **do**
10: $\quad\quad\quad B'_a \longleftarrow B' \setminus a$
11: $\quad\quad\quad S_a \longleftarrow sortTree(B'_a)$
12: $\quad\quad\quad m \longleftarrow$ mapping from $DFS(B'_a)$ to $DFS(S_a)$
13: $\quad\quad\quad$**if** $S_a \in \sigma'$ **then**
14: $\quad\quad\quad\quad n \longleftarrow$ node for S_a in D
15: $\quad\quad\quad$**else**
16: $\quad\quad\quad\quad \sigma' \longleftarrow \sigma' \cup \{S_a\}$
17: $\quad\quad\quad\quad n \longleftarrow$ new node for S_a
18: $\quad\quad\quad\quad$insert n in D
19: $\quad\quad\quad\quad tree(n) \longleftarrow S_a$
20: $\quad\quad\quad$**end if**
21: $\quad\quad\quad$add edge from p to n in D labeled with a, m
22: $\quad\quad$**end for**
23: $\quad$**end for**
24: $\quad\sigma \longleftarrow \sigma'$
25: **end while**

Algorithm 8 Part 2 of $\mathbf{Opt_{DP,ISO}}(\mathbf{B}, \beta)$

for i from 1 to $|B|$ **do**
  for all n with $n \in D$ and $|tree(n)| = i$ **do**
    for all $a, b \in leaves(tree(n))$ **do**
      $v_{min} \longleftarrow \infty$
      $l_{min} \longleftarrow \perp$
      $(n_a, m_a) \longleftarrow$ node pointed to by edge labeled with a, mapping stored as edge label
      $(n_b, m_b) \longleftarrow$ node pointed to by edge labeled with b, mapping stored as edge label
      for all $c \in leaves(tree(n)) \setminus \{a, b\}$ **do**
        $(n_c, m_c) \longleftarrow$ node pointed to by edge labeled with c, mapping stored as edge label
        $v \longleftarrow \Big(opt_2(n_a, m_a(DFS(b)), m_a(DFS(c))) + opt_2(n_b, m_b(DFS(a)), m_b(DFS(c))) + opt_2(n_c, m_c(DFS(a)), m_c(DFS(b))) + 1 \Big)/3$
        if $v < v_{min}$ **then**
          $v_{min} \longleftarrow v$
          $l_{min} \longleftarrow c$
        end if
      end for
      $opt(n, DFS(a), DFS(b)) \longleftarrow (l_{min}, v_{min})$
    end for
  end for
end for

Algorithm 9 Part 3 of $\mathbf{Opt_{DP,ISO}}(\mathbf{B}, \beta)$

$v_{min} \longleftarrow \infty$
$s_{min} \longleftarrow \perp$
$n \longleftarrow$ node for B in D
$S \longleftarrow tree(n)$
for all $a, b \in leaves(S)$ **do**
  if $opt_2(n, DFS(a), DFS(b)) < v_{min}$ **then**
    $v_{min} \longleftarrow opt_2(n, DFS(a), DFS(b))$
    $s_{min} \longleftarrow \langle a, b, opt_1(n, DFS(a), DFS(b)) \rangle$
  end if
end for
$\mathbf{return}(v_{min}, s_{min})$

The second part basically iterates over all nodes in the DAG of subtrees once in lines 1 and 2. The inner part of the loop in line 8 is iterated $\mathcal{O}(leaves(B')^3) = \mathcal{O}(leaves(B)^3) = \mathcal{O}(n^3)$ times. All operations are otherwise constant. Hence the second part takes time $\mathcal{O}(n^3 \cdot |D|)$, where D is the DAG of subtrees of B.

The first part of the algorithm constructs D, the DAG of subtrees of B. The loops in lines 5 and 7 are executed $|D|$ times. The inner part of the loop at line 9 is therefore executed $|D| \cdot leaves(B') = \mathcal{O}(n|D|)$ times. The set look-ups can be implemented in average time $\mathcal{O}(1)$ (one comparison) and the sorting of the tree takes times $\mathcal{O}(n^2)$ (see Lemma 6.4). Hence the overall running time of part three is $\mathcal{O}(n^3|D|)$.

Summing up we get:

Lemma 6.5 (Running Time of Algorithm 6). *Algorithm 6 takes time $\mathcal{O}(|B|^3|D|)$, where D is the DAG of subtrees of B.*

Proof. See considerations above. □

The question that is left is the size of the DAG of subtrees of B. This DAG excludes isomorphic subtrees, otherwise the worst case size would be $n!$ (following the same consideration as for Algorithm 3 in section 6.2).

6.4 A Bound for the Worst Case Size of the DAG of Subtrees

We will only give a lower bound for the worst case size of D. This lower bound is reached by analyzing the number of non-isomorphic subtrees of binary trees. The n-th binary tree B_n is constructed by adding two complete binary trees B_{n-1} to a new root.

Lemma 6.6 (Asymptotic Size of Subtree DAG of Complete Binary Trees). *The asymptotic size of the subtree DAG of a complete binary tree B_n with height n is*

$$2^{c \cdot 2^n}$$

where $0.6346563536974 \leq c \leq 0.634775057128 75$.

Proof. Because of the recursive composition of binary trees we will use induction to calculate the number of distinct subtrees.

The size of the DAG of B_0 is $b_0 = 1$ and of B_1 is $b_1 = 3$.

Let b_n be the number of distinct subtrees of B_n. When combining the binary tree B_n of the next order, we can estimate the number of non-isomorphic subtrees as the combination of two non-isomorphic subtrees of B_{n-1}. Combining the same kinds of subtree twice would introduce a pair of isomorphic subtrees, hence we can enumerate the subtrees of B_{n-1} and only combine subtrees where the first number is higher or equal to the second. Additionally we can combine all non-isomorphic subtrees of B_{n-1} with no tree (or the empty tree) on the other side and add the subtree for the single root node. This leads to the following recursive formula:

$$b_n = \sum_{i=0}^{b_{n-1}} i + \sum_{i=0}^{b_{n-1}} 1 + 1 = \sum_{i=0}^{b_{n-1}} (i+1) + 1 = \sum_{i=1}^{b_{n-1}+1} i + 1 = \sum_{i=0}^{b_{n-1}+1} i = \frac{1}{2}(b_{n-1}+1)(b_{n-1}+2)$$

To get an asymptotic bound for $b_n = (b_{n-1}+1)(b_{n-1}+2)/2$, we will take the (dual) logarithm:

$$
\begin{aligned}
& & b_n &= \tfrac{1}{2}(b_{n-1}+1)(b_{n-1}+2) \\
\Rightarrow & & \log_2(b_n) &= \log_2(\tfrac{1}{2}(b_{n-1}+1)(b_{n-1}+2)) \\
& & &= \log_2(\tfrac{1}{2}b_{n-1}^2 + \tfrac{3}{2}b_{n-1} + 1) \\
& & &= \log_2(\tfrac{1}{2}b_{n-1}^2 \cdot (1 + \tfrac{3}{b_{n-1}} + \tfrac{2}{b_{n-1}^2})) \\
& & &= \log_2(\tfrac{1}{2}) + \log_2(b_{n-1}^2) + \log_2(1 + \tfrac{3}{b_{n-1}} + \tfrac{2}{b_{n-1}^2}) \\
& & &= -1 + 2\log_2(b_{n-1}) + \log_2(1 + \tfrac{3}{b_{n-1}} + \tfrac{2}{b_{n-1}^2}) \\
& & &= -1 + 2(-1 + 2\log_2(b_{n-2}) + \log_2(1 + \tfrac{3}{b_{n-2}} + \tfrac{2}{b_{n-2}^2})) + \log_2(1 + \tfrac{3}{b_{n-1}} + \tfrac{2}{b_{n-1}^2}) \\
& & &= -1 \cdot \textstyle\sum_{i=0}^{n-2} 2^i + 2^{n-1}\log_2(b_1) + \sum_{i=0}^{n-2} 2^i \cdot \log_2(1 + \tfrac{3}{b_{n-1-i}} + \tfrac{2}{b_{n-1-i}^2}) \\
& & &= -2^{n-1} + 1 + 2^{n-1}\log_2(b_1) + \textstyle\sum_{i=0}^{n-2} 2^i \cdot \log_2(1 + \tfrac{3}{b_{n-1-i}} + \tfrac{2}{b_{n-1-i}^2}) \\
\Rightarrow & \log_2(b_n)/2^n & &= -\tfrac{1}{2} + \tfrac{1}{2^{n-1}} + \tfrac{\log_2(3)}{2} + \textstyle\sum_{i=0}^{n-2} \tfrac{1}{2^{n-i}} \cdot \log_2(1 + \tfrac{3}{b_{n-1-i}} + \tfrac{2}{b_{n-1-i}^2}) \\
& & &= \tfrac{\log_2(3)}{2} - \tfrac{1}{2} + \tfrac{1}{2^{n-1}} + \textstyle\sum_{i=2}^{n} \tfrac{1}{2^i} \cdot \log_2(1 + \tfrac{3}{b_{i-1}} + \tfrac{2}{b_{i-1}^2})
\end{aligned}
$$

Since b_n is an increasing function, the term $\frac{1}{2^i} \cdot \log_2(1 + \frac{3}{b_{i-1}} + \frac{2}{b_{i-1}^2})$ decreases very fast as $i \to \infty$:

$$
\begin{aligned}
\log_2(b_n)/2^n &\underset{n\to\infty}{\longrightarrow} c \\
\log_2(b_n) &\underset{n\to\infty}{\longrightarrow} c \cdot 2^n \\
b_n &\underset{n\to\infty}{\longrightarrow} 2^{c\cdot 2^n}
\end{aligned}
$$

To calculate the bounds on c, we simply evaluate $\sum_{i=2}^{n} \frac{1}{2^i} \cdot \log_2(1 + \frac{3}{b_{i-1}} + \frac{2}{b_{i-1}^2})$ for the first four terms, resulting in the lower bound 0.6346563536974. Since the individual $\log$-terms decrease, we add $\sum_{i>=6} \frac{1}{2^i} \cdot \log_2(1 + \frac{3}{b_4} + \frac{2}{b_4^2})$ to receive the upper bound of 0.63477505712875. □

Since the number of nodes of the binary trees is asymptotic to 2^n, the size of the subtree DAG of a binary tree with n nodes (we could always calculate the size of the next smallest binary tree) is $\Omega(2^{c\cdot n})$. This is also a lower bound for worst case size of the subtree DAG of a tree with n nodes.

The dynamic programming algorithm is therefore at least exponential.

From our empirical results and from their inner structure we conjecture that binomial trees are the trees with the largest number of non-isomorphic subtrees. The number of subtrees b_n (including isomorphic ones) of the n-th binomial tree is given by the recurrence

$$
b_n = \begin{cases} b_{n-1} \cdot (b_{n-1}+1) & \text{if } n > 0, \\ 1 & \text{otherwise.} \end{cases}
$$

The asymptotic behavior of b_n is given by $2^{c\cdot 2^n}$, with $0.67618 \leq c \leq 0.67819$ (see [Urq95], page 434). This is of course also an upper bound on the number of non-isomorphic subtrees.

6.5 Optimizations for $\mathrm{Opt}_{\mathrm{DP,ISO}}(\mathrm{B})$

Although the exponential nature of the algorithm cannot be altered, we will give some further hints on optimizing the algorithm.

The following optimizations can additionally be applied:

- Iterate only over edges of DAG nodes (in line 3 and 8 of Algorithm 8.) This should exclude redundant work for leaves whose removal will lead to the same tree (thus avoid to meet these redundancies twice, because they were already observed in line 13 of Algorithm 8)

- In the preemptive case, there is no need to store a mapping from the nodes of one tree to another, the nodes can simply be reassigned. Therefore the inner loop of Algorithm 8 can easily be replaced by sorting the edges of the corresponding DAG node by the expected processing time of their destination nodes and take the nodes labeling the lowest three edges as schedule.

- For this work there was the need to quickly find counter examples to given scheduling strategies. If all trees with a given number of nodes are to be calculated, part 1 of Algorithm 6 can be replaced by an algorithm similar to Algorithm 7, that starts at the single node tree and level wise calculates the DAG by adding leaves (instead of removing). The complexity is the same as for Algorithm 6. The resulting DAG size is asymptotic to (see 2.2):

$$a_n = \frac{1}{\alpha^{n-1}n}\sqrt{\beta/2\pi n} + \mathcal{O}\left(\frac{1}{\sqrt{n^5}\alpha^n}\right)$$

  where $1/\alpha \approx 2.955765285652$ and $\sqrt{\beta/2\pi} \approx 0.439924012571$.

- When generating the DAG of all rooted, unordered tree as described above, the computation of level $n+1$ only depends upon level n.

  The size of the levels grows with 1, 1, 2, 4, 9, 20, 48, 115, 286, 719, 1842, 4766, 12486, 32973, 87811, 235381, 634847, 1721159, ... To be able to reach to level 18 (all trees with 18 nodes) measures must be taken to keep the size of the tree small. Since today it seems virtually impossible to reach anything higher than 26 nodes with a normal Workstation (5759636510 trees, the number exceeding a 32bit integer constant), a tree representation can stick to numbers smaller than 32.

  Also it is nearly impossible to keep this amount of data in main memory. Therefore the trees need to be clustered. A trivial clustering could take the degree of the root node (although this only "gains" a level).

6.6 Using m Machines to Schedule an In-Tree

The only values of m for which an efficient algorithm for $(Pm|intree|\mathbb{E}(C_{max}))$ is known are 1 and 2. When $m > 2$ it seems very hard to find an efficient (polynomial time) algorithm (although HLF is near optimal). The following result shows that the usage of more machines results in a better total expected time and thus gives a reason, why it is of interest to be able to schedule more than two machines.

Lemma 6.7. *For any given in-tree B and m machines, let $\mathbb{E}(T_{\alpha_m}(B))$ be the optimal expected total processing time for $(Pm|intree|\mathbb{E}(C_{max}))$. Then*

$$\forall m_l, m_s : m_l > m_s \Rightarrow \forall B : \mathbb{E}(T_{\alpha_{m_l}}(B)) \leq \mathbb{E}(T_{\alpha_{m_s}}(B))$$

(Using more machines always results in a better optimal expected total processing time).

Proof. Given the optimal strategy α_{m_s} for m_s machines we will show that there is a strategy for m_l machines with an expected value that is no worse than $\mathbb{E}(T_{\alpha_{m_s}}(B))$. This strategy $\hat{\alpha}_{m_l}$ will be a strategy that always schedules the nodes that α_{m_s} would schedule and some additional nodes. More precise, at any point in time t the strategy $\hat{\alpha}_{m_l}$ will always schedule at least those nodes that α_{m_s} would schedule at time t. We will prove that such a strategy exists in the following by induction over the discrete decision points of $\hat{\alpha}_{m_l}$.

The induction hypothesis is that $\hat{\alpha}_{m_l}$ schedules all nodes that α_{m_s} would schedule no later than α_{m_s}. The beginning is easy, since at $t = 0$ we simply schedule the same nodes as α_{m_s} (and some more).

Lets look the point t, at which under schedule $\hat{\alpha}_{m_l}$ a machine has just finished its task. Because we have always scheduled all nodes scheduled by α_{m_s}, these nodes are either under processing or already finished. Since work has begun at any task no later than under α_{m_s}, no such task can finish later.

We can reconstruct the tree that α_{m_s} would see from consulting the history. (α_{m_s} is known, we know the processing time that each finished task had required, and all tasks that would be finished under α_{m_s} are also finished under $\hat{\alpha}_{m_l}$). From that tree and α_{m_s} we can derive what tasks would be scheduled at the time t. (There is no setup time between the finishing of one machine and the reassignment of that machine to a new task. Hence, there is always a complete set of scheduled tasks and a definite tree.)

If all tasks that α_{m_s} would schedule are scheduled (or can be scheduled at that instance if two decision points are exactly the same), there is no problem and we can just schedule any additional node.

If a task that would be scheduled under α_{m_s} at time t has already been finished, we must take care that by scheduling other nodes we do not "miss" a decision point of α_{m_s}. Suppose the next task that would finish under α_{m_s} is one that is not finished yet but scheduled. Then this task will finish under $\hat{\alpha}_{m_l}$ schedule at least as early as under α_{m_s}. Therefore, for this case the lemma holds. If the next task that would finish under α_{m_s} is a task that has already been finished, we can exactly determine which one of the already finished ones that would be, because we know all processing times of all finished tasks. We can simply assume that the task finishes first and look at the resulting tree and the next set of nodes to be scheduled. We can continue speculating on the upcoming decision points under α_{m_s} until we find a point where all tasks are scheduled but not finished or where we can schedule an additional task.

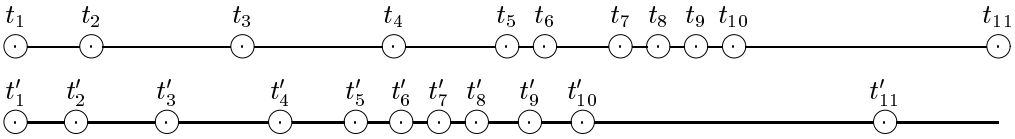


Figure 6.3: Anticipating Decision Points

This way we always stay ahead of α_{m_s}. Since the next decision point (the time until the next task is finished) is not known, we must anticipate all possible decision points that α_{m_s} might reach by finishing nodes that we have already finished (see Figure 6.3, at decision point t'_{10} we must anticipate α_{m_s}'s decisions points t_7, t_8, t_9, and t_{10} because we will not be able to schedule a new machine until t'_{11}).

Because no task is scheduled later with m_l machines than with m_s machines, $\hat{\alpha}_{m_l}$ finishes at least as early as α_{m_s} for any tree and for any instantiation of the random variable for the task processing times. As a result, the total expected processing time cannot be smaller with fewer machines.

□

Hence, for this problem the saying "too many cooks spoil the broth" does not hold.

Chapter 7

Falsification and Evaluation of Scheduling Strategies

7.1 Falsification and Evaluation Criteria

In the following we will take a look at known scheduling strategies and classes of strategies. For a given tree B let $\dot{\alpha}_{OPT}(B)$ be the set of all optimal solutions to $(P3|intree|\mathbb{E}(C_{max}))$ with in-tree constraints B.

A strategy S is **optimal**, if and only if for all trees B $\alpha_S(B) \in \dot{\alpha}_{OPT}(B)$. If a strategy S results in a set $\dot{\alpha}_S(B)$ of solutions, then S is **optimal**, if and only if $\dot{\alpha}_S(B) \subseteq \dot{\alpha}_{OPT}(B)$.

Especially if a strategy will result in a set of solutions, the strategy might be **can-optimal**. This is the case, if $\dot{\alpha}_S(B) \cap \dot{\alpha}_{OPT}(B) \neq \emptyset$ and $\dot{\alpha}_S(B) \setminus \dot{\alpha}_{OPT}(B) \neq \emptyset$ because $\dot{\alpha}_S(B)$ is a set and we could assume that a solution is chosen at random. Hence there is a chance that an optimal solution is drawn (the strategy can be optimal).

If $\dot{\alpha}_S(B) \cap \dot{\alpha}_{OPT}(B) = \emptyset$, the solution is **non-optimal**.

To decide whether a strategy is not optimal, a single counter example suffices. If a counter-example with $\dot{\alpha}_S(B) \not\subseteq \dot{\alpha}_{OPT}(B)$ exists, the strategy S is at most can-optimal.

Figure 7.1 shows an example. For the optimal schedule, the node 3 must be scheduled and two nodes can be picked from the nodes 5, 6, and 7. A can-optimal scheduling strategy is an algorithm that results in multiple possible schedules of which some are optimal and some are not optimal. Figure 7.1 (b) shows such a case where an algorithm's solution can be any three member set from the nodes 3, 5, 6, and 7. If the picked set of scheduled nodes contains node 3, the result is optimal, otherwise it is not optimal. Finally, Figure 7.1 (c) shows the non-optimal case. Any schedule will include node 1, so that no schedule can be optimal. It may often occur that an algorithm only presents a single solution. In this case it is either optimal or non-optimal.

When providing counter examples we will restrict ourselves to the trees with the marked nodes (of the optimal and/or falsified strategy). Displaying the complete DAGs and the points where the falsified strategy loses time would enlarge this work with an unnecessary number of figures. However, we will make the deviations plausible wherever possible (trying to state an intuitive reason, why certain strategies are not optimal). We will also restrict ourselves to showing the smallest existing counter example tree (if possible).

There is no known easier way to choose the optimal strategy that is better than Algorithm 6 yet. When comparing two non-optimal strategies S_1 and S_2 we need to compare the expected processing times reached by applying S_1 and S_2 in comparison to the optimal time. This is not an easy task, because we need to take all trees into account, hence we usually restrict ourself to the falsification of strategies.

Another comparison method is the counting of the number of trees that the strategy fails at. This simply compares all solutions for all trees up to a given size and compares the number of trees that are scheduled

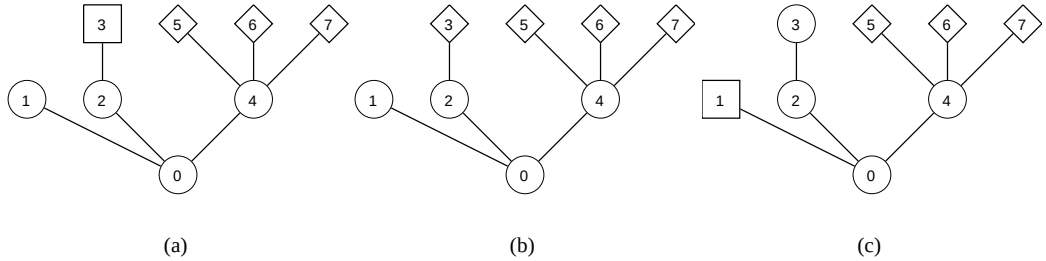


Figure 7.1: Difference Between Optimal (a), Can-Optimal (b), and Non-Optimal (c) (all square nodes must be scheduled, from the diamond nodes enough can be picked at random to get three nodes).

optimally. This quantitative measure also has an impact on the quality of solutions. Any non-optimal schedule for a small tree will be inherited by larger trees of which the smaller is a subtree of.

7.2 Preemptive versus Non-Preemptive Scheduling

We will mainly focus on non-preemptive scheduling because it seems that it is the harder of the two. Unfortunately an optimal preemptive strategy is not necessarily an optimal non-preemptive strategy and vice versa. Figure 7.2 shows the smallest trees, where the optimal preemptive strategy is only can-optimal (a) or non-optimal (b) to the non-preemptive schedule. The intuitive reason in both examples is that it is bad to be left with a high tree with only two nodes left. In (a) removing any leaf leads to the same subtree, but the non-preemptive strategy must already take into account, that the majority of scheduled leaves should then be on the smaller of both sides (that reduces the probability to end up with a tree, with two leaves in two 2-node branches). Basically the same holds for (b), but it is far less easy to see. If one of the leaves $\{3, 4, 5\}$ is finished, both strategies prefer scheduling the remaining two. On the other hand the tree, with one of the leaves $\{8, 10\}$ removed has a slightly lower expected value, so there is a trade-off between being ready for the first case or working towards the second case. For the preemptive schedule this decision needs not be made.

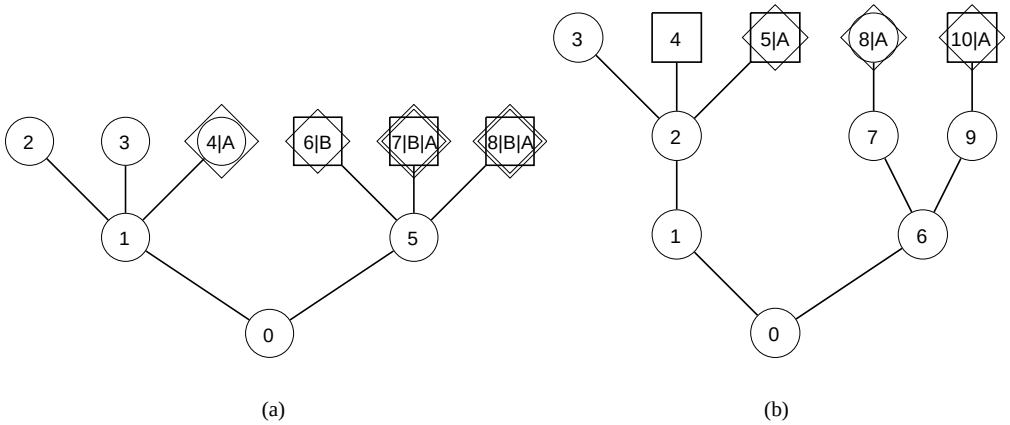


Figure 7.2: Preemptive Schedules in Relation to Non-Preemptive Schedules. (Squares show the non-preemptive optimal solutions. Diamonds and upper case letters show the different optimal preemptive solutions.)

Figure 7.2 (b) is also the smallest tree, where the optimal non-preemptive solution is non-optimal to the optimal preemptive schedule. For all smaller trees the optimal non-preemptive schedule is also optimal as preemptive schedule.

7.3 The Highest Level First Strategy (HLF)

The **highest level first** (HLF) strategy is optimal for the case with two machines in parallel $(P2|intree|\mathbb{E}(C_{max}))$, for the preemptive as well as for the non-preemptive case. This result is due to Chandy and Reynolds [CR75] (the proof is also found in [Pin95]). In [Pin95] various scheduling problems with deterministic input variables are shown to be optimally solved by the HLF rule (which Pinedo calls critical path (CP) rule).

For our problem with three processors [CR75] already contains a counter example tree with 12 nodes where HLF fails. The smallest two counter examples (trees with 11 nodes) showing that HLF is non-optimal are given in Figure 7.3 (a), (b). Figure 7.3 (c) also gives the smallest tree that shows that HLF is at most can-optimal.

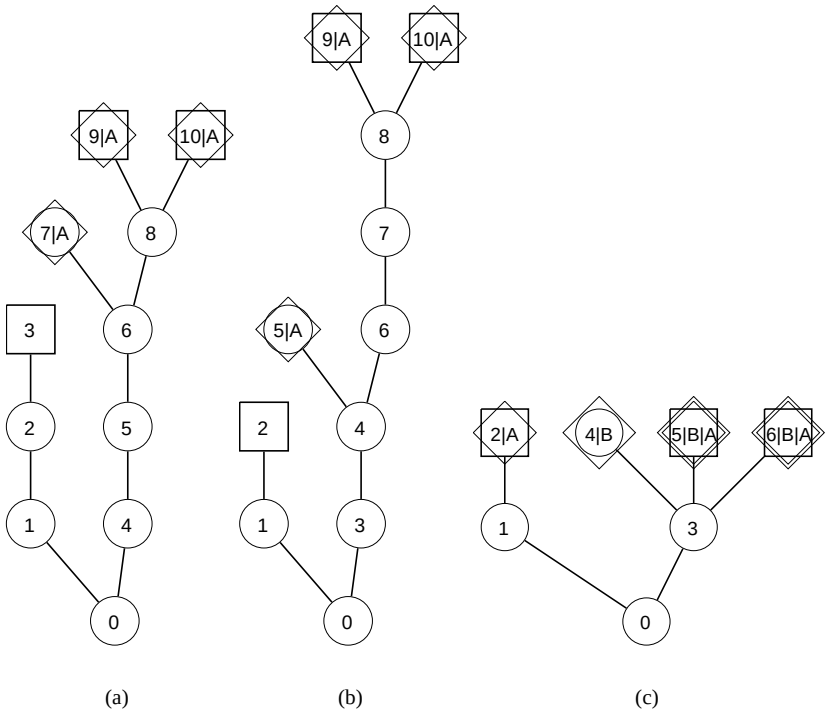


Figure 7.3: HLF is Non-Optimal (Squares show the non-preemptive optimal solutions. Diamonds and upper case letters show the HLF solutions.)

Papadimitriou and Tsitsiklis have shown that HLF is asymptotically near optimal in the sense that the expected processing time of an HLF strategy $\mathbb{E}(T_{HLF})$ divided by the expected processing time of an optimal schedule $\mathbb{E}(T_{OPT})$ approaches 1 very quickly as the tree size grows (see [PT87]), precisely:

Theorem 7.1 (Relative Optimality of HLF). *For any in-tree B of size n and an arbitrary number of processors m, let $\mathbb{E}(T_{HLF}(B))$ be the expected processing time under an HLF scheduling strategy and let $\mathbb{E}(T_\pi(B))$ be the expected processing time under any strategy π. There exists some function*

$\beta : \{1, 2, \ldots\} \to [0, \infty)$ *such that* $\lim_{n\to\infty} \beta(n) = 0$ *and*

$$\mathbb{E}(T_{HLF}(B)) \leq \inf_{\pi} \mathbb{E}(T_{\pi}(B))(1 + \beta(n))$$

Proof. See [PT87]. □

This is a rather strong result. On the one hand we know that HLF is non-optimal and very often only can-optimal, on the other we know that it is close to optimal.

Because often there may be multiple nodes at the highest level, one can distinguish between HLF schedules that provide a tie-breaking method and ones that break ties at random (Chandy and Reynolds call these A-Schedules if a labeling scheme is used and B-Schedules if random decisions are made).

Providing a tie-breaking method still keeps close to the proven optimal property, while being able to improve the algorithm in a lot of schedules. Figure 7.4 shows some variants where HLF was extended to a lexicographical order with HLF in the first component and various other weights in the second component. The next section will show that – while improving HLF – no such approach will ever be able to yield an optimal algorithm. (Note, that for the algorithm in the last line of Figure 7.4, the parent's or an ancestor's in-degree is second in order, if that node is the root of a pod. A pod is a subtree where all leaves are at the same height and the in-degree of the root is equal to the number of leaves. E.g., node 3 of Figure 7.2 is the root of a pod.)

Algorithm (or Question)	Failures on Trees With k Nodes							
	7	8	9	10	11	12	13	14
Q: Number of trees	48	115	286	719	1842	4766	12486	32973
Q: Trees with more than 4 leaves	20	67	207	595	1655	4494	n/a	n/a
HLF without tie-breaking	1	8	33	116	372	1130	3352	9613
Lowest in-degree of leaf-parent		1	6	25	90	288	913	2846
Parent's subtree weight			2	8	36	123	453	1577
Reverse DFS number of leaves			1	6	30	110	n/a	n/a
Lexicographical order of parent's or ancestor's in-degree (if pod), parent's in-degree, and parent's subtree weight					11	58	250	976

Figure 7.4: HLF-Based Algorithms with Different Lexicographical Orders

7.4 Static List Scheduling Strategies

A static list scheduling strategy for $(P3|intree|\mathbb{E}(C_{max}))$ would order all nodes of the tree in a static list. At the beginning the highest available nodes are scheduled (a node is available if it is a leaf or its ancestors have already been processed). Every time a machine is freed, the highest available unscheduled node is assigned to that machine.

Lemma 7.1 (Static List Policies Fail for $(P3|intree|\mathbb{E}(C_{max}))$). *No static list scheduling strategy can be optimal for* $(P3|intree|\mathbb{E}(C_{max}))$.

Proof. Have a look at Figure 7.5. Initially node 4 and two of the nodes $\sigma = \{6, 7, 8\}$ need to be scheduled. Depending on the first node that is finished, different nodes need to be scheduled:

Case 1 If a node from σ is finished first, then the remaining unscheduled node of σ is scheduled, resulting in a scheduling order of $4, \sigma_1, \sigma_2, \sigma_3, \ldots$.

Case 2 If 4 is finished first, one of the nodes $\gamma = \{2, 3\}$ needs to be scheduled. If the newly scheduled node is finished next, the other node from γ needs to be scheduled.

Disregarding the nodes 0, 1, and 5 the following static list schedules can represent the first case:

$$3, 4, \sigma_1, \sigma_2, \sigma_3, 2$$
$$4, 3, \sigma_1, \sigma_2, \sigma_3, 2$$
$$4, \sigma_1, 3, \sigma_2, \sigma_3, 2$$
$$4, \sigma_1, \sigma_2, 3, \sigma_3, 2$$
$$4, \sigma_1, \sigma_2, \sigma_3, 3, 2$$

The following static list schedules can represent the second case:

$$4, \sigma_1, \sigma_2, 2, 3, \sigma_3$$
$$4, \sigma_1, \sigma_2, 3, 2, \sigma_3$$
$$4, \sigma_1, 3, \sigma_2, 2, \sigma_3$$
$$4, 3, \sigma_1, \sigma_2, 2, \sigma_3$$
$$3, 4, \sigma_1, \sigma_2, 2, \sigma_3$$

The lists for these cases contradict each other because in the first case at most one node from γ (the one constraint by 4) can be in the list before the last node from σ, while the second case both nodes from γ must appear in the list before the last node from σ. Therefore, no optimal static list schedule for this instance of $(P3|intree|\mathbb{E}(C_{max}))$ exists. □

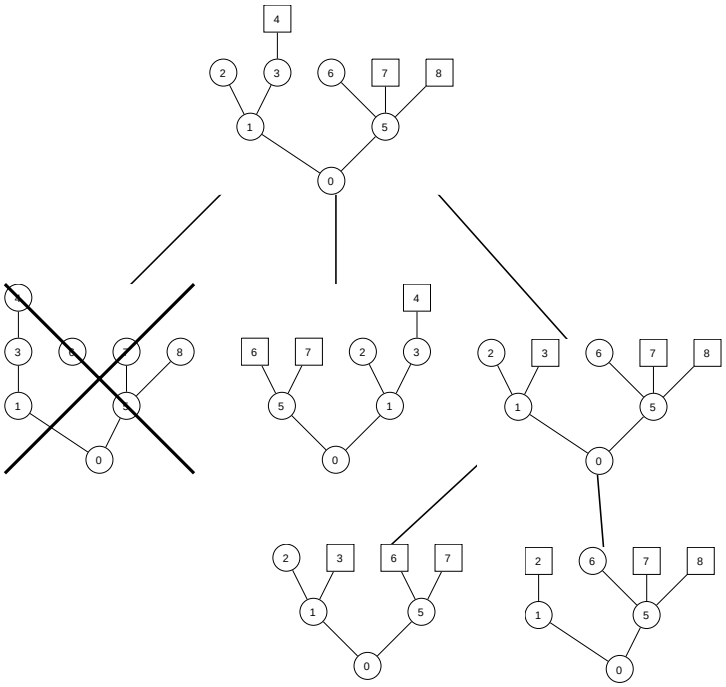


Figure 7.5: An Optimal Non-Preemptive Strategy Cannot be a Static List Schedule. (The complete highest two levels and part of third level of the subtree DAG are shown, squares are scheduled nodes for an optimal schedule and the crossed out tree is never reached under an optimal schedule.)

If we relax the definition of a static list schedule a bit and adapt to the problem at hand we can define **semi-static list schedules**:

Definition 7.1 (Semi-Static List Policy). *A **semi-static list scheduling strategy** is a policy where for any given tree or subtree all tasks are ordered into a list, depending only on the structure of the tree or subtree. Any free machine is assigned to the highest available, unscheduled task.*

(This corresponds to deciding solely by the tree structure).

Obviously there is a semi-static list policy that is optimal for the preemptive problem (just put the three optimal nodes at the top).

This is not the case for the non-preemptive problem:

Lemma 7.2 (Semi-Static List Policies Fail for $(P3|intree|\mathbb{E}(C_{max}))$). *No semi-static list scheduling strategy can be optimal for $(P3|intree|\mathbb{E}(C_{max}))$.*

Proof. Have a look at Figure 7.6. A semi-static list schedule assigns exactly one list to one tree. Let the tree shown as (b) and as the rightmost subtree of the second level of the subtree DAG shown as (a) be denoted as B (the unique representation is $2, 3, 0, 0, 1, 0, 4, 0, 0, 0, 0$).

From (b) the optimal static list for tree B starts with 4 and at least two nodes from $\{7, 8, 9, 10\}$. From (a) the static list for tree B starts with node 4 and nodes 2 and 3. This contradicts the existence of a single list for tree B.

Therefore no semi-static list scheduling strategy can be optimal for $(P3|intree|\mathbb{E}(C_{max}))$. □

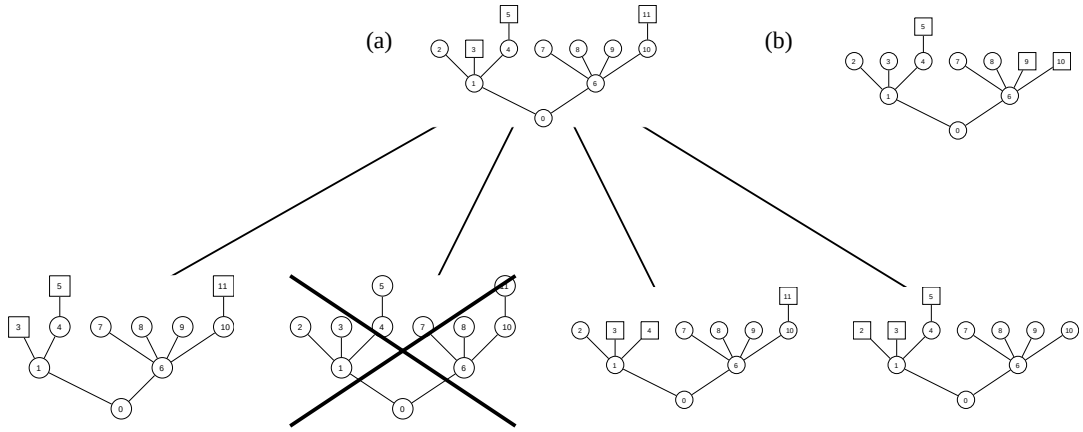


Figure 7.6: An Optimal Non-Preemptive Strategy Cannot be a Semi-Static List Schedule. (The highest two levels of the subtree DAG are shown in (a), squares are scheduled nodes for an optimal schedule, and the crossed out tree is never reached under an optimal schedule. The optimal schedule for the rightmost subtree of second level in (a) when scheduled by itself is shown in (b).)

As a result we can exclude a large number of strategies that depend solely on the tree structure (including HLF and variants). Any optimal strategy must take already scheduled tasks into account. This can be either accomplished by looking at sets of three nodes or by scheduling nodes on a given tree one by one and including information about the previous scheduled nodes. Surely there could be an algorithm that – given a tree – returns three leaves that correspond to the optimal initial assignment, but this algorithm would be of little help in subsequent steps.

7.5 Further Scheduling Strategies

A **Monte Carlo** algorithm is an algorithm that has a bounded probability of giving the correct answer, but it can also give an incorrect answer (Las Vegas algorithms never give a wrong answer, they will answer "'don't know"' instead). Iterating a Monte Carlo algorithm a number of times will result in reducing the probability of giving a wrong answer. A good Monte Carlo Algorithm makes a problem quite tractable (e.g. prime number testing – for which actually a Las Vegas algorithm exists). Since the problem at hand is stochastic in nature it seems interesting to try a Monte Carlo-like method. Such a method would basically work as follows. We choose a schedule and randomly execute it. The execution simulates the scheduling by randomly choosing a leaf which finishes first in each step. In this way, one execution takes time $\mathcal{O}(n)$. For each step either one third, one half, or one is added, corresponding to the expected length of the step with three, two, or one machine working at the same time. If this is iterated a large number of times the average of the sums should reflect the real expected value of the simulated schedule.

Unfortunately, this is not possible because we do not know how to continue after the first leaf of the initial schedule has "'finished"'. It is obviously not possible to check all possibilities again – this would result in the algorithm from section 6.1. Two rules seem plausible at that point: choose a random leaf to schedule or choose a random leaf at a highest level to schedule (because HLF is at least asymptotically optimal). Both variants have been tested a number of times and the results are presented in Figure 7.7. It may seem that 100 iterations per schedule is not a lot, but the number of schedules to test grows cubicly with the number of leaves.

Algorithm (or Question)	Failures on Trees With k Nodes								
	4	5	6	7	8	9	10	11	12
Q: Number of trees	4	9	20	48	115	286	719	1842	4766
Q: Trees with more than 4 leaves		1	5	20	67	207	595	1655	4494
HLF				1	8	33	116	372	1130
Monte Carlo 100 (HLF) 1. run				1	9	32	156	514	n/a
Monte Carlo 100 (HLF) 2. run				1	6	43	155	526	n/a
Monte Carlo 100 (HLF) 3. run					3	31	144	527	n/a
Monte Carlo 100 (random) 1. run				1	4	25	117	406	n/a
Monte Carlo 100 (random) 2. run				1	8	31	144	438	n/a
Monte Carlo 100 (random) 3. run					8	37	139	421	n/a

Figure 7.7: Comparison of Monte Carlo Algorithms to HLF

The method does not seem very stable either. Figure 7.8 shows a tree for which the above described Monte Carlo method results in very unpredictable results. Although the algorithm seldom produces a very bad result (e.g. schedules node 7 of the tree in Figure 7.8), it varies strongly between the other three possibilities (scheduling three leaves from $\{3, 4, 5\}$, two leaves from $\{3, 4, 5\}$ and one from $\{9, 10\}$, or one leaf from $\{3, 4, 5\}$ and two from $\{9, 10\}$ - see Figure 7.9).

The results shown in Figure 7.9 suggest that a larger number of iterations might result in a near optimal algorithm. Does a number exist for which the Monte Carlo algorithm hardly ever fails? We will see later in section 10.1 that the differences between two schedules may become very small, possibly as small as 3^{-n-2} where n is the size of the tree. If this is indeed the case, then the following reasoning suggests that no smallest sufficient number of iterations for the Monte Carlo method exists. For each compared schedule a number of l iterations is made. For each iteration a sum of the terms 1, $1/2$, and $1/3$ is calculated and from the total the arithmetic mean is calculated by summing all sub-sums and dividing by the number of iterations. The sum of all sub-sums is itself a sum of the terms 1, $1/2$, and $1/3$. When comparing two different schedules, the Monte Carlo algorithm compares these sums. The smallest possible difference between the two sums is $1/6$ (if same terms in both sums are coupled, the terms $1/2$ and $1/3$ remain). Both sums should reflect the expected value for their corresponding schedule. If the difference is 3^{-n-2}, then $1/6$ divided by l should be well below that difference. Otherwise the results do not seem to be exact enough

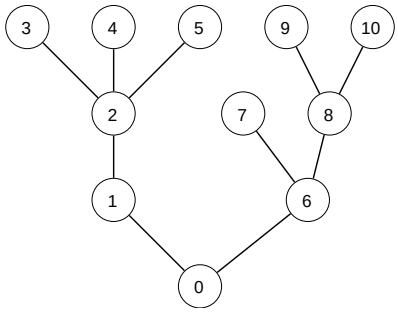


Figure 7.8: Example Tree for Instability of the Monte Carlo Method. (The optimal schedule is $\{3, 4, 5\}$ with an expected processing time of $35261/5832$.)

Schedule	Probability of random HLF	Percentage with given number of iterations 100	1000	10000	100000
Three leaves from $\{3, 4, 5\}$ (optimal)	10%	10%	15%	50%	90%
Two leaves from $\{3, 4, 5\}$ and one from $\{9, 10\}$	60%	70%	50%	40%	10%
One leaf from $\{3, 4, 5\}$ and two from $\{9, 10\}$	30%	20%	35%	10%	0%

Figure 7.9: Stability of Monte Carlo for Tree in Figure 7.8 and 20 Runs

to distinguish both schedules with a high enough probability. Hence, l needs to be $Oof3^n$ which results in the Monte Carlo algorithm being as intractable as the dynamic programming algorithm. The higher the chosen number of iterations, the better the algorithm seems to be able to distinguish schedules with smaller differences. On the other hand, the results suggest that Monte Carlo does not perform better than HLF and is of higher complexity.

Chapter 8

Taking a Closer Look at Two-Leaves Subtrees

8.1 Yet Another Way to Calculate the Expected Processing Time

From 5.1 and Corollary 5.1 we know two ways to calculate the expected processing time for a strategy α. Both of these are based on the probabilities of execution paths (an execution path being the order in which nodes are finished, a permutation over all nodes) under strategy α. This corresponds to using Algorithm 1 for calculating the expected processing time. We will now find another way to calculate the expected processing time.

Let B be a tree and D be the subtree DAG for B and let B_0 be the subtree corresponding to only the root of B. Suppose we could calculate the probability of reaching any given subtree B' under strategy α. If we can find an anti-chain of "independent" subtrees in the subtree DAG, whose expected processing times are easily calculated, then we would be able to give the total expected processing time. An anti-chain C is a set of trees that have the following properties:

1. No tree in the chain is reachable (in D) from any other tree in the chain.
2. Every execution path (a path from B to B_0) includes exactly one element from the chain C.

To be able to calculate the total expected processing time $\mathbb{E}(T_\alpha(B))$, all trees in any path from B to a node in C should have at least three leaves.

Theorem 8.1 (Expected Processing Time by Subtree Weights). *Let B ($n = |B|$) be a tree, D its subtree DAG, and C an anti-chain with the above stated property that $\forall p \in paths_D(B, B_0)$: let $B_p = p \cap C \wedge$ let $p' = path(B, B_p) :\Rightarrow \forall B' \in p' : \big(leaves(B') \geq 3 \vee B' = B_p\big)$.*

The total expected processing time is then

$$\mathbb{E}(T_\alpha(B)) = \sum_{B' \in C} P_\alpha[B'] \cdot \left(\mathbb{E}(T_\alpha(B')) + \frac{1}{3}(n - |B'|)\right)$$

Proof. Let the nodes of B be $V = \{1, \ldots, n\}$. For any subtree B' of B, let the nodes of B' be $V' = \{i_1, \ldots, i_{|B'|}\}$.

The probability of reaching B' is the sum over all execution paths p with $\forall i \in V' : p(i) > n - |B'|$ (that is B' occurs during p).

$$P_\alpha[B'] = \sum_{p \in C P(B) \wedge \forall i \in V' : p(i) > n - |B'|} P_\alpha^B[p]$$

The probability that an execution path p of B ends with an execution path p' of B' is:

$$P[p' \sqsupset_{suff} p] = P_\alpha[B'] \cdot P_\alpha^{B'}[p'] = \sum_{p \in CP(B) \wedge p' \sqsupset_{suff} p} P_\alpha^B[p]$$

The expected total processing time for B' is (by Theorem 5.1):

$$\mathbb{E}(T_\alpha(B')) = \sum_{p' \in CP(B')} P_\alpha^{B'}[p'] \cdot \left(s_2(p') \cdot \frac{1}{3} + (s_1(p') - s_2(p')) \cdot \frac{1}{2} + (|B'| - s_1(p')) \right)$$

If p ends with p' we will call p' a suffix of p denoted by $p' \sqsupset_{suff} p$ or $p = suffix_{|B'|}(p)$ (suffix of the length of $|B|$). Since before B' is reached there are always at least three leaves left, if $p \in CP(B)$ and $p' \in CP(B')$ where p' is a suffix of p, then

$$\forall i \in V' : p(i) = n - |B'| + p'(i) : s_2(p) = n - |B'| + s_2(p') \text{ and } s_1(p) = n - |B'| + s_1(p')$$

Because C is an anti-chain the following holds true:

$$\forall p \in CP(B) : \exists B' \in C : \left(\left(\forall i \in V' : p(i) > n - |B'| \right) \wedge \left(\nexists B'' \in C \setminus \{B'\} : \left(\forall i \in V'' : p(i) > n - |B''| \right) \right) \right)$$

The total execution time is therefore

$$\begin{aligned}
& \mathbb{E}(T_\alpha(B)) \\
= & \sum_{p \in CP(B)} P_\alpha^B[p] \cdot \left(s_2(p) \cdot \frac{1}{3} + (s_1(p) - s_2(p)) \cdot \frac{1}{2} + (n - s_1(p)) \right) \\
= & \sum_{B' \in C} \left(\sum_{p \in CP(B) \wedge \forall i \in V' : p(i) > n - |B'|} P_\alpha^B[p] \cdot \left(s_2(p) \cdot \frac{1}{3} + (s_1(p) - s_2(p)) \cdot \frac{1}{2} + (n - s_1(p)) \right) \right) \\
= & \sum_{B' \in C} \left(\sum_{p \in CP(B) \wedge \forall i \in V' : p(i) > n - |B'|} P_\alpha^B[p] \cdot \left(\left(n - |B'| + s_2(suffix_{|B'|}(p)) \right) \cdot \frac{1}{3} \right. \right. \\
& \quad + \left((n - |B'| + s_1(suffix_{|B'|}(p))) - (n - |B'| + s_2(suffix_{|B'|}(p))) \right) \cdot \frac{1}{2} \\
& \quad \left. \left. + \left(n - (n - |B'| + s_1(suffix_{|B'|}(p))) \right) \right) \right) \\
= & \sum_{B' \in C} \left(\sum_{p \in CP(B) \wedge \forall i \in V' : p(i) > n - |B'|} P_\alpha^B[p] \cdot \left(\frac{n - |B'|}{3} + s_2(suffix_{|B'|}(p)) \cdot \frac{1}{3} \right. \right. \\
& \quad \left. \left. + \left(s_1(suffix_{|B'|}(p)) - s_2(suffix_{|B'|}(p)) \right) \cdot \frac{1}{2} + \left(|B'| - s_1(suffix_{|B'|}(p)) \right) \right) \right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{B' \in C} \Bigg(\frac{n-|B'|}{3} \Big(\sum_{p \in CP(B) \wedge \forall i \in V' : p(i) > n-|B'|} P_\alpha^B[p] \Big) \\
&\quad + \sum_{p' \in CP(B')} \Bigg(\sum_{p \in CP(B) \wedge p' \sqsupset_{suff} p} P_\alpha^B[p] \cdot \Big(s_2(\mathit{suffix}_{|B'|}(p)) \cdot \frac{1}{3} \\
&\quad + \big(s_1(\mathit{suffix}_{|B'|}(p)) - s_2(\mathit{suffix}_{|B'|}(p)) \big) \cdot \frac{1}{2} + \big(|B'| - s_1(\mathit{suffix}_{|B'|}(p)) \big) \Big) \Bigg) \Bigg) \\
&= \sum_{B' \in C} \Bigg(\frac{n-|B'|}{3} P_\alpha[B'] \\
&\quad + \sum_{p' \in CP(B')} \Bigg(\sum_{p \in CP(B) \wedge p' \sqsupset_{suff} p} P_\alpha^B[p] \cdot \Big(s_2(p') \cdot \frac{1}{3} \\
&\quad + \big(s_1(p') - s_2(p') \big) \cdot \frac{1}{2} + \big(|B'| - s_1(p') \big) \Big) \Bigg) \Bigg) \\
&= \sum_{B' \in C} \Bigg(\frac{n-|B'|}{3} P_\alpha[B'] \\
&\quad + \sum_{p' \in CP(B')} \Big(s_2(p') \cdot \frac{1}{3} + \big(s_1(p') - s_2(p') \cdot \frac{1}{2} \big) + \big(|B'| - s_1(p') \big) \Big) \\
&\qquad \cdot \Big(\sum_{p \in CP(B) \wedge p' \sqsupset_{suff} p} P_\alpha^B[p] \Big) \Bigg) \\
&= \sum_{B' \in C} \Bigg(\frac{n-|B'|}{3} P_\alpha[B'] \\
&\quad + P_\alpha[B'] \cdot \sum_{p' \in CP(B')} \Big(s_2(p') \cdot \frac{1}{3} + \big(s_1(p') - s_2(p') \big) \cdot \frac{1}{2} + \big(|B'| - s_1(p') \big) \Big) \cdot P_\alpha^{B'}[p'] \Bigg) \\
&= \sum_{B' \in C} P_\alpha[B'] \cdot \Big(\mathbb{E}(T_\alpha(B') + \frac{1}{3}(n - |B'|) \Big)
\end{aligned}
$$

□

This result could lead to an algorithm, maximizing the probability of reaching cheap subtrees, provided

- an anti-chain C can be found with the stated property,
- the expected processing times of the elements of the anti-chain can be calculated, and
- the chain is small enough to be evaluated efficiently.

Obviously any level of the subtree DAG (all nodes representing subtrees with the same number of nodes) is an anti-chain. Choosing a high enough level will satisfy the stated property. Unfortunately the number of nodes in the level can be large and the problem of calculating the expected processing times of the chain elements is only a little easier than the original problem. Finally a way of optimizing the probabilities that a cheap tree is met needs to be found.

Figure 8.1 shows the DAG for the tree in Figure 5.3 with the probabilities of the subtrees shown by their shade of gray (the DAG is the same as the one in Figure 5.4 on page 19).

In the rest of this chapter we will try to find a cure for some of this points.

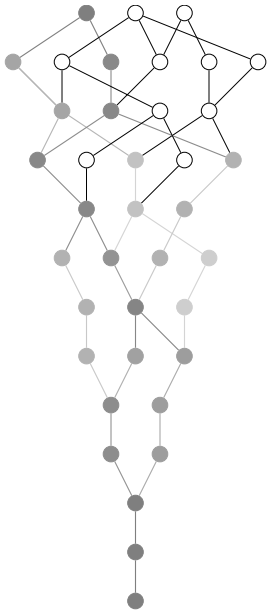


Figure 8.1: Probabilities in Subtree DAG for Example Tree in Figure 5.3 for the Optimal Schedule (subtrees represented by empty nodes are not reached at all, the shade of gray of a node represents the probability from light low to dark high).

8.2 The Optimal Expected Processing Time for Two-Leaves-Trees

In the following we will show how the optimal expected processing time for two-leaves-trees can be found. Note, that a strategy has no choice, once there are only two leaves left, and hence the optimal expected processing time for any strategy and for the optimal strategy is the same for any two-leaves-tree.

We will describe all trees with two leaves by three numbers $(a|k|l)$:

a - The number of nodes from the root to the first node where the tree branches (excluding that node).

k - The number of nodes in the left branch.

l - The number of nodes in the right branch.

Obviously, for any two-leaves-tree B, $a + k + l + 1 = |B|$. See Figure 8.2 for an example of this classification.

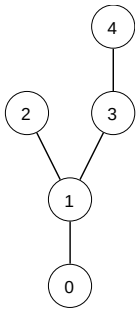


Figure 8.2: Two-Leaves-Tree Example for $(1|1|2)$.

For the calculation of the expected processing time of B we will first have a look at trees with $a = 0$. At every step a machine from one of the branches finishes and the branch is decreased. Either branch is

equally likely, so both possibilities can be weighted with $1/2$. A machine is expected to finish after $1/2$ of the expected processing time for one task because two machines are working in parallel. If there is only one branch left, the remaining branch and the branching node will be processed with a single machine. The expected processing time q can thus be described by the following recursion:

$$q(k,l) = \begin{cases} k+1 & \text{if } l = 0, \\ l+1 & \text{if } k = 0, \\ \big(1 + q(k-1,l) + q(k,l-1)\big)/2 & \text{otherwise.} \end{cases}$$

If $a \geq 0$, the nodes below the branching node are also processed with a single machine, hence the total processing time of a two-leaves-tree $(a|k|l)$ is

$$p(a,k,l) = q(k,l) + a$$

Observation 8.1 (Expected Processing Time of Two-Leaves-Tree). *If B is a two-leaves-tree with two branches of the size k and l and the height of the branching node a, its expected processing time is $p(a,k,l)$*

Algorithm 10 $\mathbf{p}(B)$

Require: $leaves(B) = 2$

```
1: proc p(B) :
2: l_1 ⟵ first leaf of B
3: l_2 ⟵ second leaf of B
4: n ⟵ lca(l_1, l_2)
5: a ⟵ height(n)
6: k ⟵ height(l_1) − a
7: l ⟵ height(l_2) − a
8: for i from 1 to k do
9:    t(0, i) ⟵ i + 1
10: end for
11: for i from 1 to k do
12:    t(i, 0) ⟵ i + 1
13: end for
14: for i from 1 to k do
15:    for j from 1 to l do
16:       t(i, j) ⟵ (t(i − 1, j) + t(i, j − 1) + 1)/2
17:    end for
18: end for
19: return(t(k,l) + a)
```

Lemma 8.1 (Calculating the Total Expected Processing Time of Two-Leaves-Trees). *Algorithm 10 calculates the total expected processing time of a two-leaves-tree B in $\mathcal{O}(|B|^2)$.*

Proof. Algorithm 10 is a simple dynamic programming version of the recursive definition of $q(a,k,l)$. Its running time is dominated by the loops in lines 14 and 15, hence its running time is $\mathcal{O}(k \cdot l) = \mathcal{O}(|B|^2)$. □

A simple consequence is the following corollary.

Corollary 8.1. *For a given tree B the total expected processing times of all two-leaves-tree subtrees B' can be calculated in $\mathcal{O}(|B|^2)$.*

A non-recursive version of the equation for $p(a,k,l)$ can be derived, looking at the dynamic programming tableau of Algorithm 10 and summing the values added in each entry separately (the initialization parts with $l = 0$ or $k = 0$ and the $+1/2$-part). The following theorem proves the correctness of the derived formula.

(Using dynamic programming to calculate all values for the binomial coefficients, the non-recursive form should also be evaluatable in time $\mathcal{O}(|B|^2)$).

Theorem 8.2 (Total Expected Processing Time of Two-Leaves-Trees). *Let*

$$\begin{aligned} s(a,k,l) &= \sum_{i=1}^{k} \left(\frac{1}{2}\right)^{l+i-1} \cdot \binom{l+i-2}{i-1} \cdot (k-i+2) \\ &+ \sum_{j=1}^{l} \left(\frac{1}{2}\right)^{k+j-1} \cdot \binom{k+j-2}{j-1} \cdot (l-j+2) \\ &+ \sum_{i=1}^{k}\sum_{j=1}^{l} \left(\frac{1}{2}\right)^{k-i+l-j+1} \cdot \binom{k-i+l-j}{l-j} \\ &+ a \end{aligned}$$

If $l > 0 \wedge k > 0$*, then* $s(a,k,l) = p(a,k,l)$.

We need the following lemmas to establish the proof:

Lemma 8.2.

$$S_n = \sum_{i=0}^{n} \left(\frac{1}{2}\right)^{i} \cdot (n-i) = \left(\frac{1}{2}\right)^{n-1} + 2n - 2$$

Proof. (By Induction on n)

For $n = 0$, left hand side:

$$lhs = \left(\frac{1}{2}\right)^{0} \cdot 0$$

For $n = 0$, right hand side:

$$rhs = \left(\frac{1}{2}\right)^{0-1} + 2 \cdot 0 - 2 = 0$$

From n to $n+1$:

$$\begin{aligned} S_{n+1} &= \sum_{i=0}^{n+1} \left(\frac{1}{2}\right)^{i} \cdot (n+1-i) \\ &= n+1+\sum_{i=1}^{n+1} \left(\frac{1}{2}\right)^{i} \cdot (n+1-i) \\ &= n+1+\frac{1}{2}\cdot\sum_{i=1}^{n+1} \left(\frac{1}{2}\right)^{i-1} \cdot (n-(i-1)) \\ &= n+1+\frac{1}{2}\cdot\sum_{i=0}^{n} \left(\frac{1}{2}\right)^{i} \cdot (n-i)) \\ &= n+1+\frac{1}{2}\cdot S_n \\ &= n+1+\frac{1}{2}\cdot\left(\left(\frac{1}{2}\right)^{n-1} + 2n - 2\right) \\ &= n+1+\left(\frac{1}{2}\right)^{n} + n - 1 \\ &= \left(\frac{1}{2}\right)^{(n+1)-1} 2\cdot(n+1) - 2 \end{aligned}$$

□

Lemma 8.3. *If $k > 0m$ then*

$$\binom{r-1}{k} + \binom{r-1}{k-1} = \binom{r}{k}$$

Proof. (See [GKP94], p. 159). □

We are now ready to prove the theorem.

Proof of Theorem 8.2. (By Induction)
Let $s'(k,l) = s(a,k,l) - a$, we will then only need to prove $q(k,l) = s'(k,l)$.
For $l = 1$, right hand side:

$$\begin{aligned}
s'(k,1) &= \sum_{i=1}^{k} \left(\frac{1}{2}\right)^{1+i-1} \cdot \binom{1+i-2}{i-1} \cdot (k-i+2) \\
&+ \sum_{j=1}^{1} \left(\frac{1}{2}\right)^{k+j-1} \cdot \binom{k+j-2}{j-1} \cdot (1-j+2) \\
&+ \sum_{i=1}^{k} \sum_{j=1}^{1} \left(\frac{1}{2}\right)^{k-i+1-j+1} \cdot \binom{k-i+1-j}{1-j} \\
&= \sum_{i=1}^{k} \left(\frac{1}{2}\right)^{i} \cdot \binom{i-1}{i-1} \cdot (k-i+2) \\
&+ \left(\frac{1}{2}\right)^{k} \cdot \binom{k+1-2}{1-1} \cdot (1-1+2) \\
&+ \sum_{i=1}^{k} \left(\frac{1}{2}\right)^{k-i+1} \cdot \binom{k-i+1-1}{1-1} \\
&= \left(\sum_{i=1}^{k} \left(\frac{1}{2}\right)^{i} \cdot 1 \cdot (k-i+2)\right) + \left(\frac{1}{2}\right)^{k} \cdot \binom{k-1}{0} \cdot 2 + \sum_{i=1}^{k} \left(\frac{1}{2}\right)^{k-(i-1)} \cdot \binom{k-i}{0} \\
&= \left(\sum_{i=1}^{k} \left(\frac{1}{2}\right)^{i} \cdot (k-i+2)\right) + \left(\frac{1}{2}\right)^{k} \cdot 2 + \sum_{i=1}^{k} \left(\frac{1}{2}\right)^{i} \\
&= \left(\frac{1}{2} \cdot \sum_{i=1}^{k} \left(\frac{1}{2}\right)^{i-1} \cdot \big((k+1) - (i-1)\big)\right) + \left(\frac{1}{2}\right)^{k} \cdot 2 + 1 - \left(\frac{1}{2}\right)^{k} \\
&= \left(\frac{1}{2} \cdot \sum_{i=0}^{k-1} \left(\frac{1}{2}\right)^{i} \cdot \big((k+1) - i\big)\right) + \left(\frac{1}{2}\right)^{k} + 1 \\
&= \left(\frac{1}{2} \cdot \sum_{i=0}^{k+1} \left(\frac{1}{2}\right)^{i} \cdot \big((k+1) - i\big)\right) - \left(\frac{1}{2}\right)^{k+1} \cdot 1 - \left(\frac{1}{2}\right)^{k+2} \cdot 0 + \left(\frac{1}{2}\right)^{k} + 1 \\
&= \frac{1}{2} \cdot \left(\left(\frac{1}{2}\right)^{k} + 2k\right) + \left(\frac{1}{2}\right)^{k+1} + 1 \\
&= \left(\frac{1}{2}\right)^{k} + k + 1
\end{aligned}$$

For $l = 1$, left hand side:

$$\begin{aligned}
q(k,1) &= (q(k-1,1)+k+1+1)/2 \\
&= ((q(k-2,1)+k+1)/2+k+2)/2 \\
&= (((q(k-3,1)+k-1+1)/2+k+1)/2+k+2)/2 \\
&\vdots \\
&= \underbrace{((\ldots(\ldots(}_{k\text{-times}} q(0,1) \underbrace{+3)/2\ldots k+2-j+1)/2\ldots+k+1)/2+k+2)/2}_{k\text{-times}} \\
&= \left(\frac{1}{2}\right)^{k-1} + \sum_{i=1}^{k}(k+3-i)\cdot\left(\frac{1}{2}\right)^{i} \\
&= \left(\frac{1}{2}\right)^{k-1} + \frac{1}{2}\cdot\sum_{i=1}^{k}\big(k+2-(i-1)\big)\cdot\left(\frac{1}{2}\right)^{i-1} \\
&= \left(\frac{1}{2}\right)^{k-1} + \frac{1}{2}\cdot\sum_{i=0}^{k-1}(k+2-i)\cdot\left(\frac{1}{2}\right)^{i} \\
&= \left(\frac{1}{2}\right)^{k-1} + \frac{1}{2}\cdot\left(-\left(\frac{1}{2}\right)^{k}\cdot 2 - \left(\frac{1}{2}\right)^{k+1}\cdot 1 - \left(\frac{1}{2}\right)^{k+2}\cdot 0 + \sum_{i=0}^{k+2}(k+2-i)\cdot\left(\frac{1}{2}\right)^{i}\right) \\
&= \left(\frac{1}{2}\right)^{k-1} - \left(\frac{1}{2}\right)^{k} - \left(\frac{1}{2}\right)^{k+2} + \frac{1}{2}\cdot\sum_{i=0}^{k+2}(k+2-i)\cdot\left(\frac{1}{2}\right)^{i} \\
&= \left(\frac{1}{2}\right)^{k-1} - \left(\frac{1}{2}\right)^{k} - \left(\frac{1}{2}\right)^{k+2} + \frac{1}{2}\cdot\left(\left(\frac{1}{2}\right)^{k+2-1} + 2(k+2) - 2\right) \\
&= \left(\frac{1}{2}\right)^{k-1} - \left(\frac{1}{2}\right)^{k} - \left(\frac{1}{2}\right)^{k+2} + \left(\frac{1}{2}\right)^{k+2} + (k+2) - 1 \\
&= \left(\frac{1}{2}\right)^{k} + k + 1
\end{aligned}$$

Since the case $k = 1$ is equivalent the basis of the induction is established.

We will prove that $s'(k,l) = q(k,l) = \big(q(k-1,l)+q(k,l-1)+1\big)/2 = \big(s'(k-1,l)+s'(k,l-1)+1\big)/2$ (the recursion works for the sum) for each of the three sum parts individually for $l > 1$ and $k > 1$:

Let $s_1(k,l) = \sum_{i=1}^{k}\left(\frac{1}{2}\right)^{l+i-1}\cdot\binom{l+i-2}{i-1}\cdot(k-i+2)$, let $s_2(k,l) = \sum_{j=1}^{l}\left(\frac{1}{2}\right)^{k+j-1}\cdot\binom{k+j-2}{j-1}\cdot(l-j+2)$, and let $s_3(k,l) = \sum_{i=1}^{k}\sum_{j=1}^{l}\left(\frac{1}{2}\right)^{k-i+l-j+1}\cdot\binom{k-i+l-j}{l-j}$

Clearly, $s'(k,l) = s_1(k,l) + s_2(k,l) + s_3(k,l)$.

We start to prove $s_1(k,l) = \frac{1}{2}\cdot s_1(k-1,l) + \frac{1}{2}\cdot s_1(k,l-1)$:

$$\begin{aligned}
&\frac{1}{2}\cdot s_1(k-1,l) + \frac{1}{2}\cdot s_1(k,l-1) \\
= \;& \frac{1}{2}\cdot\sum_{i=1}^{k}\left(\frac{1}{2}\right)^{l-1+i-1}\cdot\binom{l-1+i-2}{i-1}\cdot(k-i+2) \\
&+\frac{1}{2}\cdot\sum_{i=1}^{k-1}\left(\frac{1}{2}\right)^{l+i-1}\cdot\binom{l+i-2}{i-1}\cdot(k-1-i+2)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\cdot\sum_{i=1}^{k}\left(\frac{1}{2}\right)^{l+i-2}\cdot\binom{l+i-3}{i-1}\cdot(k-i+2)\\
&\quad+\frac{1}{2}\cdot\sum_{i=2}^{k}\left(\frac{1}{2}\right)^{l+i-2}\cdot\binom{l+i-3}{i-2}\cdot(k-i+2)\\
&= \sum_{i=1}^{k}\left(\frac{1}{2}\right)^{l+i-1}\cdot\left(\binom{l+i-3}{i-1}+\binom{l+i-3}{i-2}\right)\cdot(k-i+2)\\
&= \sum_{i=1}^{k}\left(\frac{1}{2}\right)^{l+i-1}\cdot\binom{l+i-2}{i-1}\cdot(k-i+2) \qquad \text{(using Lemma 8.3)}\\
&= s_1(k,l)
\end{aligned}
$$

The proof for $s_2(k,l)=\frac{1}{2}\cdot s_2(k-1,l)+\frac{1}{2}\cdot s_2(k,l-1)$ is analogous to the previous one.
That leaves to prove $s_3(k,l)=\frac{1}{2}\cdot s_3(k-1,l)+\frac{1}{2}\cdot s_3(k,l-1)+\frac{1}{2}$:

$$
\begin{aligned}
\frac{1}{2}\cdot s_3(k-1,l)+\frac{1}{2}\cdot s_3(k,l-1)+\frac{1}{2} &= \frac{1}{2}\cdot\sum_{i=1}^{k-1}\sum_{j=1}^{l}\left(\frac{1}{2}\right)^{k-i+l-j}\cdot\binom{k-i+l-j-1}{l-j}\\
&\quad+\frac{1}{2}\cdot\sum_{i=1}^{k}\sum_{j=1}^{l-1}\left(\frac{1}{2}\right)^{k-i+l-j}\cdot\binom{k-i+l-j-1}{l-j-1}+\frac{1}{2}
\end{aligned}
$$

The two sums can be swapped, hence we will add the missing element to each sum, so that both sum over l and k. $\binom{l-j-1}{l-j}$ evaluates to zero for $j<l$ because the factor $l-j-1-(l-j)+1=0$ will appear in the denominator at the last position. The missing element for the first sum is

$$
\frac{1}{2}\cdot\sum_{j=1}^{l}\left(\frac{1}{2}\right)^{l-j}\cdot\binom{l-j-1}{l-j}=\frac{1}{2}\cdot\left(\frac{1}{2}\right)^{0}\cdot\binom{0-1}{0}=\frac{1}{2}.
$$

The missing element for the second sum is ($\binom{x}{-1}=0$ by definition)

$$
\frac{1}{2}\cdot\sum_{i=1}^{k}\left(\frac{1}{2}\right)^{k-i}\cdot\binom{k-i-1}{-1}=0
$$

Hence we need to subtract $\frac{1}{2}$ when adding the missing elements:

$$
\begin{aligned}
&= \frac{1}{2}\cdot\sum_{i=1}^{k}\sum_{j=1}^{l}\left(\frac{1}{2}\right)^{k-i+l-j}\cdot\binom{k-i+l-j-1}{l-j}-\frac{1}{2}\\
&\quad+\frac{1}{2}\cdot\sum_{i=1}^{k}\sum_{j=1}^{l}\left(\frac{1}{2}\right)^{k-i+l-j}\cdot\binom{k-i+l-j-1}{l-j-1}+\frac{1}{2}\\
&= \frac{1}{2}\cdot\sum_{i=1}^{k}\sum_{j=1}^{l}\left(\frac{1}{2}\right)^{k-i+l-j}\cdot\left(\binom{k-i+l-j-1}{l-j}+\binom{k-i+l-j-1}{l-j-1}\right)\\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}\left(\frac{1}{2}\right)^{k-i+l-j+1}\cdot\binom{k-i+l-j}{l-j}\\
&= s_3(k,l)
\end{aligned}
$$

As a result,

$$\begin{aligned}
&\big(s'(k-1,l)+s'(k,l-1)+1\big)/2\\
&\quad=\big(s_1(k-1,l)+s_2(k-1,l)+s_3(k-1,l)+s_1(k,l-1)+s_2(k,l-1)+s_3(k,l-1)+1\big)/2\\
&=\frac{1}{2}\big(s_1(k-1,l)+s_1(k,l-1)\big)+\frac{1}{2}\big(s_2(k-1,l)+s_2(k,l-1)\big)+\frac{1}{2}\big(s_3(k-1,l)+s_3(k,l-1)+1\big)\\
&\qquad=s_1(k,l)+s_2(k,l)+s_3(k,l)\\
&\qquad=s'(k,l)
\end{aligned}$$

□

A closed form for the sums could not be derived.

8.3 Two-Leaves-Subtrees as Pseudo Anti-Chain

Two-leaves-subtrees have the nice property that their expected processing time is relatively easy to calculate. Unfortunately they do not form an anti-chain. If we still want to use the result of Theorem 8.1, we can do the following:

1. Calculate the probabilities that the two-leaves-subtree at hand is the first subtree reached that has less than three leaves, or
2. adapt the weights of the expected processing time for a two-leaves-subtree, such that it takes into account that smaller subtrees are also reached.

Since the probabilities of reaching smaller subtrees from greater ones are fixed, this gives us basically the choice, whether we want to include the discounting of the probability in our calculation or whether we want to discount the weights.

Combining Observation 8.1 with Theorem 8.1, the weight that a two-leaves-subtree $(a|k|l)$ has with respect to the tree B with size n is:

$$w(k,l,a,n)=p(a,k,l)+\frac{n-a-k-l-1}{3}$$

The probability that a subtree $(a|k|l)$ is the first subtree to be reached with two leaves is

$$P_\alpha^f[(a|k|l)]=P_\alpha[(a|k|l)]-\frac{1}{2}P_\alpha[(a|k+1|l)]-\frac{1}{2}P_\alpha[(a|k|l+1)]$$

Let C_2 be the set of all two-leaves-subtrees of B, replacing this in the equation from Theorem 8.1, we get:

Corollary 8.2 (Expected Processing Time by Two-Leaves-Subtree Weights).

$$\mathbb{E}(T_\alpha(B))=\sum_{(a|k|l)\in C_2}P_\alpha^f[(a|k|l)]\cdot w(k,l,a,n)$$

Lets define a discounted weight $w_d(k,l,a,n)$ as:

$$w_d(k,l,a,n)=\begin{cases}\frac{1}{6} & \text{if } k>1 \text{ and } l>l,\\ \frac{2k-l+2a+n+4}{6} & \text{if } k>1 \text{ and } l=l,\\ \frac{2l-k+2a+n+4}{6} & \text{if } k=1 \text{ and } l>l,\\ \frac{4a+2n+9}{6} & \text{if } k=1 \text{ and } l=l.\end{cases}$$

Lemma 8.4 (Discounted Two-Leaves-Subtree Weights).

$$\sum_{(a\,|k\,|l)\in C_2} P_\alpha^f[(a|k|l)]\cdot w(k,l,a,n) = \sum_{(a\,|k\,|l)\in C_2} P_\alpha[(a|k|l)]\cdot w_d(k,l,a,n)$$

Proof.

$$\begin{aligned}
&\sum_{(a\,|k\,|l)\in C_2} P_\alpha^f[(a|k|l)]\cdot w(k,l,a,n)\\
=&\sum_{(a\,|k\,|l)\in C_2} \Big(P_\alpha[(a|k|l)] - \frac{1}{2}P_\alpha[(a|k+1|l)] - \frac{1}{2}P_\alpha[(a|k|l+1)]\Big)\cdot w(k,l,a,n)\\
=&\sum_{(a\,|k\,|l)\in C_2} P_\alpha[(a|k|l)]\cdot A(k,l,a,n)
\end{aligned}$$

where

$$A(k,l,a,n) = \begin{cases} w(k,l,a,n) - w(k-1,l,a,n)/2 - w(k,l-1,a,n)/2 & \text{if } k>1 \text{ and } l>l,\\ w(k,l,a,n) - w(k-1,l,a,n)/2 & \text{if } k>1 \text{ and } l=l,\\ w(k,l,a,n) - w(k,l-1,a,n)/2 & \text{if } k=1 \text{ and } l>l,\\ w(k,l,a,n) & \text{if } k=1 \text{ and } l=l. \end{cases}$$

$$= \begin{cases} p(a,k,l) + \frac{n-a-k-l-1}{3} - p(a,k-1,l)/2 - \frac{n-a-k-l-2}{6} & \\ -p(a,k,l-1)/2 - \frac{n-a-k-l-2}{6} & \text{if } k>1 \text{ and } l>l,\\ p(a,k,l) + \frac{n-a-k-l-1}{3} - p(a,k-1,l)/2 - \frac{n-a-k-l}{6} & \text{if } k>1 \text{ and } l=l,\\ p(a,k,l) + \frac{n-a-k-l-1}{3} - p(a,k,l-1)/2 - \frac{n-a-k-l}{6} & \text{if } k=1 \text{ and } l>l,\\ p(a,k,l) + \frac{n-a-k-l-1}{3} & \text{if } k=1 \text{ and } l=l. \end{cases}$$

$$= \begin{cases} q(k,l) + a - q(k-1,l)/2 - a/2 - q(k,l-1)/2 - a/2 & \\ -\frac{n-a-k-l-1-(n-a-k-l-2)}{3} & \text{if } k>1 \text{ and } l>l,\\ q(k,l) + a - q(k-1,l)/2 - a/2 + \frac{n-a-k-l-2}{6} & \text{if } k>1 \text{ and } l=l,\\ q(k,l) + a - q(k,l-1)/2 - a/2 + \frac{n-a-k-l-2}{6} & \text{if } k=1 \text{ and } l>l,\\ q(k,l) + a + \frac{n-a-k-l-1}{3} & \text{if } k=1 \text{ and } l=l. \end{cases}$$

$$= \begin{cases} q(k,l) - q(k-1,l)/2 - q(k,l-1)/2 - 1/2 + 1/2 - \frac{1}{3} & \text{if } k>1 \text{ and } l>l,\\ q(k,1) + a/2 - q(k-1,1)/2 + \frac{n-a-k-l-2}{6} & \text{if } k>1 \text{ and } l=l,\\ q(1,l) + a/2 - q(1,l-1)/2 + \frac{n-a-k-l-2}{6} & \text{if } k=1 \text{ and } l>l,\\ q(1,1) + \frac{3a+n-a-3}{3} & \text{if } k=1 \text{ and } l=l. \end{cases}$$

$$= \begin{cases} \frac{1}{6} & \text{if } k>1 \text{ and } l>l,\\ q(k-1,1)/2 + q(k,0)/2 + 1/2 - q(k-1,1)/2 & \\ +\frac{3a+n-a-k-l-2}{6} & \text{if } k>1 \text{ and } l=l,\\ q(0,l)/2 + q(1,l-1)/2 + 1/2 - q(1,l-1)/2 & \\ +\frac{3a+n-a-k-l-2}{6} & \text{if } k=1 \text{ and } l>l,\\ \frac{5}{2} + \frac{2a+n-3}{3} & \text{if } k=1 \text{ and } l=l. \end{cases}$$

$$= \begin{cases} \frac{1}{6} & \text{if } k>1 \text{ and } l>l,\\ k/2 + 1/2 + 1/2 + \frac{3a+n-a-k-l-2}{6} & \text{if } k>1 \text{ and } l=l,\\ l/2 + 1/2 + 1/2 + \frac{3a+n-a-k-l-2}{6} & \text{if } k=1 \text{ and } l>l,\\ \frac{9+4a+2n}{6} & \text{if } k=1 \text{ and } l=l. \end{cases}$$

$$
\begin{aligned}
&= \begin{cases} \frac{1}{6} & \text{if } k > 1 \text{ and } l > l, \\ \frac{3k+6+3a+n-a-k-l-2}{6} & \text{if } k > 1 \text{ and } l = l, \\ \frac{3l+6+3a+n-a-k-l-2}{6} & \text{if } k = 1 \text{ and } l > l, \\ \frac{9+4a+2n}{6} & \text{if } k = 1 \text{ and } l = l. \end{cases} \\
&= \begin{cases} \frac{1}{6} & \text{if } k > 1 \text{ and } l > l, \\ \frac{2k-l+2a+n+4}{6} & \text{if } k > 1 \text{ and } l = l, \\ \frac{2l-k+2a+n+4}{6} & \text{if } k = 1 \text{ and } l > l, \\ \frac{9+4a+2n}{6} & \text{if } k = 1 \text{ and } l = l. \end{cases} \\
&= w_d(k, l, a, n)
\end{aligned}
$$

□

With $w_d(k, l, a, n)$ we can use two-leaves-subtrees as pseudo-anti-chain. The only things still unknown are the probabilities of reaching two-leaves-subtrees. We will try to deal with that in the next chapter.

8.4 HLF Revisited

The results of the previous sections give another clue, why HLF works for the two machine problem but not for the three machines one. If there are only two machines, the subtrees that matter in the sense of a weight by probability approach are the one-leaf-subtrees. Each such subtree corresponds exactly to a single leaf, the higher that leaf, the larger the expected processing time for being left with the corresponding tree. Hence in the subtree view it seems intuitively right to schedule the highest level leaves first.

For three machines and two-leaves-subtrees the problem is not as clear. For a tree B with n nodes the two-leaves-subtree that is largest in the number of nodes must not necessarily be the one with the largest weight. Consider subtrees $(0|2|3)$ and $(0|1|3)$. The subtrees' expected processing times are $p(0, 2, 3) = 71/16$ and $p(0, 1, 3) = 33/8$. The weights in the sum for the expected processing time of B are

$$w(2, 3, 0, n) = \frac{71}{16} + \frac{n - 0 - 2 - 3 - 1}{3} = \frac{117 + 16n}{48} = \frac{39}{16} + \frac{n}{3} = 2.4375 + \frac{n}{3}$$

and

$$w(1, 3, 0, n) = \frac{33}{8} + \frac{n - 0 - 1 - 3 - 1}{3} = \frac{59 + 8n}{24} = \frac{59}{24} + \frac{n}{3} = 2.458\bar{3} + \frac{n}{3}$$

Hence, $w(1, 3, 0, n) > w(2, 3, 0, n)$, the largest subtree does not have the largest weight. Therefore only scheduling the highest nodes cannot be optimal.

8.5 A Lower and an Upper Bound on the Total Expected Processing Time

Theorem 8.1 helps us to determine a lower and an upper bound on the total expected processing time for a given tree B:

Lemma 8.5 (A Lower and an Upper Bound on the Total Expected Processing Time). *Let $L_{min} = (a_{min}|k_{min}|l_{min})$ be the two-leaves-subtree, having the smallest weight $w(k_{min}, l_{min}, a_{min}, |B|)$, and let $L_{max} = (a_{max}|k_{max}|l_{max})$ be the two-leaves-subtree, having the largest weight $w(k_{max}, l_{max}, a_{max}, |B|)$. Then $w(k_{min}, l_{min}, a_{min}, |B|)$ is a lower and $w(k_{max}, l_{max}, a_{max}, |B|)$ an upper bound for the total expected processing time of B.*

Proof. From Corollary 8.2:

$$\begin{aligned}\mathbb{E}(T_\alpha(B)) &= \sum_{(a|k|l)\in C_2} P_\alpha^f[(a|k|l)] \cdot w(k,l,a,n) \\ &\geq \sum_{(a|k|l)\in C_2} P_\alpha^f[(a|k|l)] \cdot w(k_{min},l_{min},a_{min},|B|) \\ &= w(k_{min},l_{min},a_{min},|B|) \cdot \sum_{(a|k|l)\in C_2} P_\alpha^f[(a|k|l)]1 \\ &= w(k_{min},l_{min},a_{min},|B|)\end{aligned}$$

Similarly,

$$\begin{aligned}\mathbb{E}(T_\alpha(B)) &= \sum_{(a|k|l)\in C_2} P_\alpha^f[(a|k|l)] \cdot w(k,l,a,n) \\ &\leq \sum_{(a|k|l)\in C_2} P_\alpha^f[(a|k|l)] \cdot w(k_{max},l_{max},a_{max},|B|) \\ &= w(k_{max},l_{max},a_{max},|B|) \cdot \sum_{(a|k|l)\in C_2} P_\alpha^f[(a|k|l)]1 \\ &= w(k_{max},l_{max},a_{max},|B|)\end{aligned}$$

□

Chapter 9

The Probability of Reaching a Two-Leaves-Subtree

9.1 Working Towards a Two-Leaves-Subtree

Given a tree B and a two-leaves-subtree L we want to be able to determine the probability that L is the first subtree reached with only two leaves left. For given tree B and subtree L we classify the nodes as nodes belonging to L, nodes that are above L, called 'descends' (set D), and nodes that are not above L, called 'on-descends'(set N). See Figure 9.1 for a schematic view. The descends are all nodes that are descends of the leaves of L, but not in L. The non-descends are all nodes that "grow out at the side" of L, simply $N = (B \setminus L) \setminus D$.

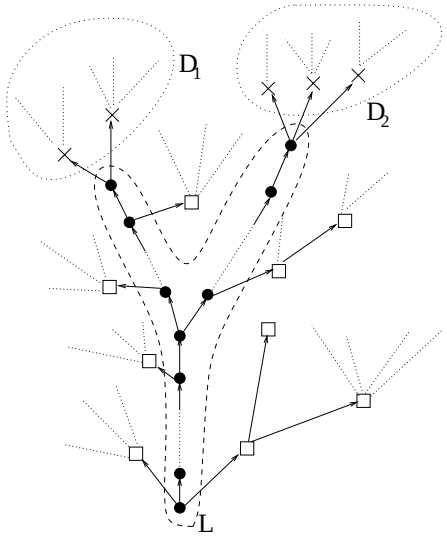


Figure 9.1: A Schematic View for Classifying Tree Nodes Based on a Selected Two-Leaves-Subtree (set L of the two-leaves-subtree's nodes are black dots, the set $D = D_1 \cup D_2$ of the two-leaves-subtree's descend nodes are crosses, and the set N of the two-leaves-subtree's non-descends are boxes)

For any subset of nodes of B that represent a forest, we can calculate a worst case scheduling scenario as the maximal possible number of nodes only schedulable with two or one machine. For a single tree these correspond to the largest one-leaf-subtree and the largest two-leaves-subtree (note, that this is not the expected worst case, but the worst case as it can occur during the actual processing of the tasks).

The highest node corresponds to the largest one-leaf-subtree. The largest two-leaves-subtree can easily be calculated as the largest one-leaf-subtree with the largest additional branch:

Lemma 9.1. *One leaf of the largest two-leaves-subtree L of a tree B must also be a highest leaf.*

Proof. Let r be the root of B. Assume the statement does not hold, then there must be two leaves a,b that define L. Without loss of generality, suppose a is higher than b. Let $c = lca(a, b)$. Then the size of L is $dist(c, r) + dist(a, c) + dist(b, c) + 1$. Let h be the highest leaf ($h \neq a$ and $h \neq b$ by assumption). Clearly,

$$dist(h, r) \geq dist(a, r) \geq dist(b, r) \tag{9.1}$$

Let $d_a = lca(a, h)$, and let $d_b = lca(b, h)$, we will make a case distinction:

a) $d_a \in path(c, a)$: $dist(d_a, h) \geq dist(d_a, a)$ (by 9.1), hence $dist(c, r) + dist(a, c) + dist(b, c) + 1 \leq dist(c, r) + dist(d_a, c) + dist(h, d_a) + dist(b, c) + 1$.

b) $d_b \in path(c, b)$: analogous.

c) $d_a = d_b \wedge d_a \in path(r, c)$, then $dist(h, d_a) \geq dist(a, d_a) \geq dist(b, d_a)$.

In any case the assumption leads to a contradiction. □

Let $|N| = l_3 + l_2 + l_1$, where $l_2 + l_1$ is the size of a maximal forest with two leaves in N and l_1 is a maximal one-leaf-subtree in N. Let $|D| = k_3 + k_2 + k_1$, where $k_2 + k_1$ is the size of a maximal forest with two leaves in D and k_1 is a maximal one-leaf-subtree in D.

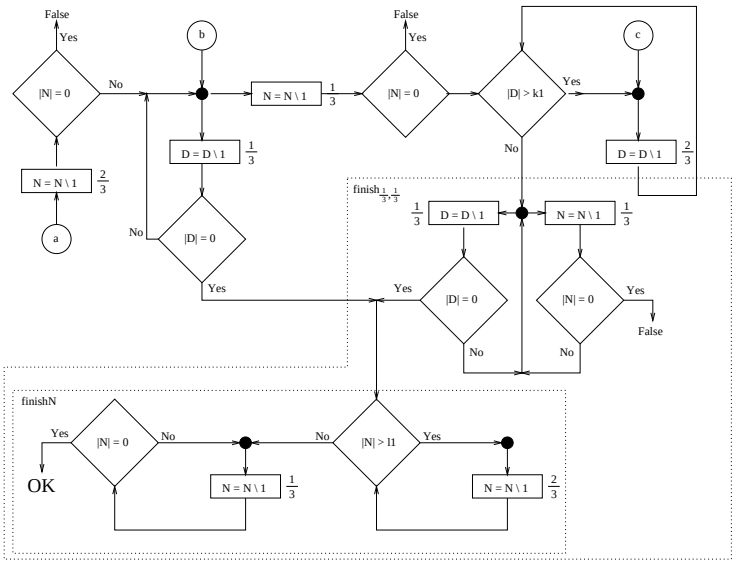


Figure 9.2: Formulae of Case 2 as Diagram

We will now estimate the worst case probability P for reaching a given subtree L of B under the assumption that a set of scheduled nodes α has already been selected and that in subsequent steps the strategy will try to reach L. L can be reached as the first two-leaves-subtree by processing all nodes of D before the last node of N and processing all nodes of $D \cup N$ before any node of L. The process is divided into three major cases of which the complicated ones are shown in Figures 9.2 and 9.3. At the filled black nodes random choices are made, leading to the events in the square boxes. The diamonds represent branches that are decided by the current state (the number of nodes in the sets N, D). These branches depend on the outcome of the previous random choices. The probabilities of the choices are written next to the corresponding occurring

event. To reduce the complexity, parts of the diagrams have been summarized to functions that are marked with dotted lines. These functions form "building blocks" of the formulae, when evaluated by a program.

We will later estimate the time needed to evaluate the formulae below. The complete evaluation will include summing up and multiplying a number of elements which are in turn powers of $\frac{1}{2}$, $\frac{1}{3}$, or $\frac{2}{3}$, or which are binomials. Note that all parameters are $\mathcal{O}(n)$, so that there can be at most $\mathcal{O}(n^2)$ binomials that need to be evaluated and $\mathcal{O}(n)$ powers. Using some sort of dynamic programming algorithm and a look-up scheme, all binomials and powers can be calculated in $\mathcal{O}(n^2)$ and each element in the sum is a simple look-up costing $\mathcal{O}(1)$.

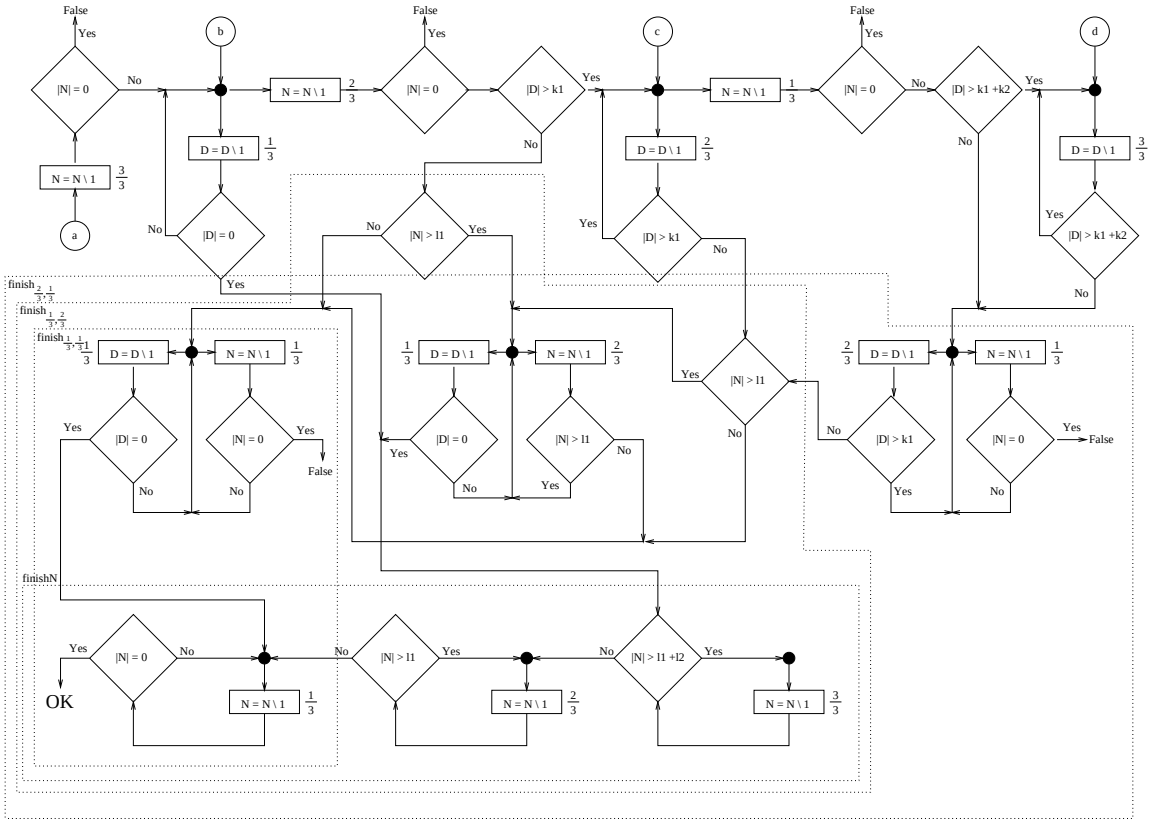


Figure 9.3: Formulae of Case 3 as Diagram

9.1.1 Special Cases

Some probabilities are very easy to calculate. For these special cases we can even give the exact probabilities.

A) $|N| = 0$

There are no non-descend nodes. The last node to be finished, before L can be reached, is a descend. Hence, the tree, one step before L is reached, is also a two-leaves-subtree. The probability of reaching L as the first two-leaves-subtree is therefore zero.

$\Rightarrow P = 0$

B) $|D| = 0$

There are no nodes above the subtree. By our assumption the leaves of L will be scheduled as late as possible. The probability depends upon how many leaves are already scheduled:

(i) $|\alpha \cap L| = 2$
$\Rightarrow P = \left(\frac{1}{3}\right)^{|N|}$

(ii) $|\alpha \cap L| = 1$
$\Rightarrow P = \left(\frac{2}{3}\right)^{(l_2+l_3)} \left(\frac{1}{3}\right)^{l_1}$

(iii) $|\alpha \cap L| = 0$
$\Rightarrow P = \left(\frac{2}{3}\right)^{l_2} \left(\frac{1}{3}\right)^{l_1}$

9.1.2 General Cases

In the general case any leaves might be scheduled. It is assumed that the scheduler tries to schedule leaves of L as late as possible. It is also assumed that nodes in D are scheduled first, afterwards the nodes from N. Since the points where the number of leaves in either D or N drops below one or two are not known exactly, we assume a "Worst Case", that is we assume that a largest subtree (or sub-forest) will remain with one or two leaves left.

Case 1) $|\alpha \cap L| = 2$

It follows, that $D = \emptyset$. This is equivalent to the special case B.

Case 2) $|\alpha \cap L| = 1$

Throughout the process of scheduling towards reaching L, nodes from N and D are finished. Let ΔN be the number of nodes already finished from N. Let ΔD be the number of nodes already finished from D.

If L is reached, the last task finished must be from N. Hence there may be a number (possibly one) of nodes in N that are finished before L is reached. This is expressed in the following function.

$$finishN^2(l_1, l_2, l_3, \Delta N) := \begin{cases} \left(\frac{1}{3}\right)^{l_1} \left(\frac{2}{3}\right)^{l_3+l_2-\Delta N} & \text{if } \Delta N \leq l_2 + l_3, \\ \left(\frac{1}{3}\right)^{l_1+l_2+l_3-\Delta N} & \text{if } \Delta N > l_2 + l_3. \end{cases}$$

Before that there is a phase where there is only one leaf left to schedule in D and one leaf from N is already scheduled. This phase is captured in the following function (the remaining leaves from D are scheduled interleaved with some, but not all, leaves from N in arbitrary order):

$$finish^2_{\frac{1}{3},\frac{1}{3}}(l_1, l_2, l_3, |D|, |N|, \Delta D, \Delta N) := \sum_{i=0}^{|N|-\Delta N-1} \left(\frac{1}{3}\right)^i \left(\frac{1}{3}\right)^{|D|-\Delta D} \binom{|D| - \Delta D + i}{i} finishN^2(l_1, l_2, l_3, \Delta N + i)$$

There are $\mathcal{O}(n)$ elements in the above sum.

With this building blocks the formulae will be easier to describe:

(a) $|\alpha \cap N| = 2$

If L is reached, then one of the machines working at a node in N finishes first. It is assigned to D. Three things can happen: Either all nodes from D are scheduled and finished before the next node from N, or enough nodes from D are finished so that only one leaf is left in D, or

the second node scheduled initially in N is finished while there are still two leaves left to be scheduled in D.

$$\Rightarrow P = \tfrac{2}{3}\Bigg(\quad \left(\tfrac{1}{3}\right)^{|D|} finishN^2(l_1,l_2,l_3,1) + \sum_{i=k_3+k_2}^{|D|-1} \left(\tfrac{1}{3}\right)^{i} \tfrac{1}{3} finish^2_{\frac{1}{3},\frac{1}{3}}(l_1,l_2,l_3,|D|,|N|,i,2) + \sum_{i=0}^{k_3+k_2-1} \left(\tfrac{1}{3}\right)^{i} \tfrac{1}{3} \left(\tfrac{2}{3}\right)^{k_3+k_2-i} finish^2_{\frac{1}{3},\frac{1}{3}}(l_1,l_2,l_3,|D|,|N|,k_3+k_2,2) \Bigg)$$

Because $finish^2_{\frac{1}{3},\frac{1}{3}}$ can sum up to $\mathcal{O}(n)$ elements, the evaluation of the above formula may take $\mathcal{O}(n^2)$ steps.

(b) $|\alpha \cap N| = 1$

This case is essentially the same as the case (a) only that a machine from N must not finish before a leaf in D is scheduled.

$$\Rightarrow P = \left(\tfrac{1}{3}\right)^{|D|} finishN^2(l_1,l_2,l_3,0) + \sum_{i=k_3+k_2}^{|D|-1} \left(\tfrac{1}{3}\right)^{i} \tfrac{1}{3} finish^2_{\frac{1}{3},\frac{1}{3}}(l_1,l_2,l_3,|D|,|N|,i,1) + \sum_{i=0}^{k_3+k_2-1} \left(\tfrac{1}{3}\right)^{i} \tfrac{1}{3} \left(\tfrac{2}{3}\right)^{k_3+k_2-i} finish^2_{\frac{1}{3},\frac{1}{3}}(l_1,l_2,l_3,|D|,|N|,k_3+k_2,1)$$

Because $finish^2_{\frac{1}{3},\frac{1}{3}}$ can sum up to $\mathcal{O}(n)$ elements, the evaluation of the above formula may take $\mathcal{O}(n^2)$ steps.

(c) $|\alpha \cap N| = 0$

There are already two machines working at nodes from D. Nodes from D are finished, until there is only one leaf left to schedule in D.

$$\Rightarrow P = \left(\tfrac{2}{3}\right)^{k_3+k_2} finish^2_{\frac{1}{3},\frac{1}{3}}(l_1,l_2,l_3,|D|,|N|,k_3+k_2,0)$$

The evaluation of the above formula may take $\mathcal{O}(n)$ steps (excluding binomials).

Case 3) $|\alpha \cap L| = 0$

As for case (2), let ΔN be the number of nodes already finished from N and let ΔD be the number of nodes already finished from D.

We again start with defining some building block functions used to describe the complexity.

The last thing that happens is that the remaining nodes from N are finished.

$$finishN^3(l_1,l_2,l_3,\Delta N) := \begin{cases} \left(\tfrac{1}{3}\right)^{l_1}\left(\tfrac{2}{3}\right)^{l_2} & \text{if } \Delta N \le l_3, \\ \left(\tfrac{1}{3}\right)^{l_1}\left(\tfrac{2}{3}\right)^{l_2+l_3-\Delta N} & \text{if } \Delta N > l_3 \wedge \Delta N \le l_2+l_3, \\ \left(\tfrac{1}{3}\right)^{l_1+l_2+l_3-\Delta N} & \text{if } \Delta N > l_2+l_3. \end{cases}$$

One way to finish occurs, when the last sequence of finishing nodes is either N or D with probability $p = \frac{1}{3}$ each (i.e. only one leaf is left in each D and N).

$$finish^3_{\frac{1}{3},\frac{1}{3}}(|D|,|N|,\Delta D,\Delta N) := \sum_{i=0}^{|N|-\Delta N-1} \left(\frac{1}{3}\right)^i \left(\frac{1}{3}\right)^{|D|-\Delta D} \binom{|D|-\Delta D+i}{i} \left(\frac{1}{3}\right)^{|N|-\Delta N-i}$$

The above sum has $\mathcal{O}(n)$ elements.

The next building block will be the point after which D has only one leaf left.

$$finish^3_{\frac{1}{3},\frac{2}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,\Delta D,\Delta N) :=$$
$$\begin{cases} finish^3_{\frac{1}{3},\frac{1}{3}}(|D|,|N|,\Delta D,\Delta N) & \text{if } \Delta N \geq l_2+l_3, \\ \sum_{i=0}^{l2+l3-1-\Delta N} \left(\frac{1}{3}\right)^{|D|-\Delta D} \left(\frac{2}{3}\right)^i \binom{|D|-\Delta D+i}{i} finishN^3(l_1,l_2,l_3,\Delta N+i) & \\ \quad + \sum_{i=0}^{|D|-\Delta D-1} \left(\frac{1}{3}\right)^i \left(\frac{2}{3}\right)^{l2+l3-\Delta N} \binom{l2+l3-\Delta N+i}{i} & \\ \quad\quad \cdot finish^3_{\frac{1}{3},\frac{1}{3}}(|D|,|N|,\Delta D+i,l2+l3) & \text{if } \Delta N < l_2+l_3. \end{cases}$$

The above sum has $\mathcal{O}(n^2)$ elements because it might call $finish^3_{\frac{1}{3},\frac{1}{3}}$ $\mathcal{O}(n)$ times.

The next building block will be the point after which D has less than three leaves left.

$$finish^3_{\frac{2}{3},\frac{1}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,\Delta D,\Delta N) :=$$
$$\sum_{i=0}^{N-\Delta N-1} \left(\frac{2}{3}\right)^{k_3+k_2-\Delta D} \left(\frac{1}{3}\right)^i \binom{k_3+k_2-\Delta D+i}{i}$$
$$\cdot finish^3_{\frac{1}{3},\frac{2}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,k_3+k_2,\Delta N+i)$$

The last building block's sum has $\mathcal{O}(n^3)$ elements because it might call $finish^3_{\frac{2}{3},\frac{1}{3}}$ $\mathcal{O}(n)$ times.

With this building blocks the formulae will be:

(a) $|\alpha \cap N| = 3$

The only thing that can happen at the beginning is that a node from N is finished. After that there are two machines working at nodes in N, and one machine working at nodes in D. This last machine can now finish a number of nodes from D, such that either D is completely finished, only one leaf is left in D, two leaves are left in D, or three leaves are left in D before another node from N is finished.

$$
\begin{aligned}
\Rightarrow P = & \left(\frac{1}{3}\right)^{|D|} finishN^3(l_1, l_2, l_3, 1) \\
& + \sum_{i=k_3+k_2}^{|D|-1} \left(\frac{1}{3}\right)^i \frac{2}{3} finish^3_{\frac{1}{3},\frac{2}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, i, 2) \\
& + \sum_{i=k_3}^{k_3+k_2-1} \left(\frac{1}{3}\right)^i \frac{2}{3} \cdot \Bigg(\\
& \qquad \left(\frac{2}{3}\right)^{k_2+k_3-i} finish^3_{\frac{1}{3},\frac{2}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, k_3+k_2, 2) \\
& \qquad + \sum_{j=0}^{k_2+k_3-i-1} \left(\frac{2}{3}\right)^j \frac{1}{3} finish^3_{\frac{2}{3},\frac{1}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, i+j, 3) \Bigg) \\
& + \sum_{i=0}^{k_3-1} \left(\frac{1}{3}\right)^i \frac{2}{3} \cdot \Bigg(\\
& \qquad \left(\frac{2}{3}\right)^{k_2+k_3-i} finish^3_{\frac{1}{3},\frac{2}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, k_3+k_2, 2) \\
& \qquad + \sum_{j=k_3-i}^{k_2+k_3-i-1} \left(\frac{2}{3}\right)^j \frac{1}{3} finish^3_{\frac{2}{3},\frac{1}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, i+j, 3) \\
& \qquad + \sum_{j=0}^{k_3-i-1} \left(\frac{2}{3}\right)^j \frac{1}{3} finish^3_{\frac{2}{3},\frac{1}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, k_3, 3) \Bigg)
\end{aligned}
$$

Because $finish^3_{\frac{2}{3},\frac{1}{3}}$ is called $\mathcal{O}(n^2)$ times, the evaluation of the above formula may take $\mathcal{O}(n^5)$ steps.

(b) $|\alpha \cap N| = 2$

This case is essentially the same than the case (a) only that a machine from N must not finish before a leaf in D is scheduled.

$$
\begin{aligned}
\Rightarrow P = & \left(\frac{1}{3}\right)^{|D|} finishN^3(l_1, l_2, l_3, 0) \\
& + \sum_{i=k_3+k_2}^{|D|-1} \left(\frac{1}{3}\right)^i \frac{2}{3} finish^3_{\frac{1}{3},\frac{2}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, i, 1) \\
& + \sum_{i=k_3}^{k_3+k_2-1} \left(\frac{1}{3}\right)^i \frac{2}{3} \cdot \Bigg(\\
& \qquad \left(\frac{2}{3}\right)^{k_2+k_3-i} finish^3_{\frac{1}{3},\frac{2}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, k_3+k_2, 1) \\
& \qquad + \sum_{j=0}^{k_2+k_3-i-1} \left(\frac{2}{3}\right)^j \frac{1}{3} finish^3_{\frac{2}{3},\frac{1}{3}}(l_1, l_2, l_3, k_1, k_2, k_3, |D|, |N|, i+j, 2) \Bigg)
\end{aligned}
$$

$$+\sum_{i=0}^{k_3-1}\left(\frac{1}{3}\right)^i\frac{2}{3}\cdot\Bigg(\left(\frac{2}{3}\right)^{k_2+k_3-i} finish^3_{\frac{1}{3},\frac{2}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,k_3+k_2,1)$$
$$+\sum_{j=k_3-i}^{k_2+k_3-i-1}\left(\frac{2}{3}\right)^j\frac{1}{3}finish^3_{\frac{2}{3},\frac{1}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,i+j,2)$$
$$+\sum_{j=0}^{k_3-i-1}\left(\frac{2}{3}\right)^j\frac{1}{3}finish^3_{\frac{2}{3},\frac{1}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,k_3,2)\Bigg)$$

Because $finish^3_{\frac{2}{3},\frac{1}{3}}$ is called $\mathcal{O}(n^2)$ times, the evaluation of the above formula may take $\mathcal{O}(n^5)$ steps.

(c) $|\alpha \cap N| = 1$

There are already two machines working at leaves in D. If the machine working at a node in N finishes before the two machines have finished enough nodes from D so that there are less than three leaves in D, three machines can work for some time at nodes in D.

$$\Rightarrow P = \sum_{i=0}^{k_3-1}\left(\frac{2}{3}\right)^i\frac{1}{3}finish^3_{\frac{2}{3},\frac{1}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,k_3,1)$$
$$+\sum_{i=k_3}^{k_3+k_2-1}\left(\frac{2}{3}\right)^i\frac{1}{3}finish^3_{\frac{2}{3},\frac{1}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,i,1)$$
$$+\left(\frac{2}{3}\right)^{k_3+k_2} finish^3_{\frac{1}{3},\frac{2}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,k_3+k_2,0)$$

Because $finish^3_{\frac{2}{3},\frac{1}{3}}$ is called $\mathcal{O}(n)$ times, the evaluation of the above formula may take $\mathcal{O}(n^4)$ steps.

(d) $|\alpha \cap N| = 0$

Already, three machines are working at nodes in D. They continue to decrease D until there are only two leaves left. This is expressed in one of the building blocks:

$\Rightarrow P = finish^3_{\frac{2}{3},\frac{1}{3}}(l_1,l_2,l_3,k_1,k_2,k_3,|D|,|N|,k_3,0)$

The evaluation $finish^3_{\frac{2}{3},\frac{1}{3}}$ may take $\mathcal{O}(n^3)$ steps.

9.1.3 Usage in an Algorithm

First note, that the evaluation of the above formulae may take $\mathcal{O}(n^5)$ steps (the $\mathcal{O}(n^2)$ for the binomials do not really matter). A tree has at most $\mathcal{O}(n^2)$ two-leaves-subtrees, so comparing all subtrees takes $\mathcal{O}(n^7)$. This makes only sense in relation to the $\mathcal{O}(n^3 2^{cn})$ of the dynamic programming algorithm of chapter 6.2.

An algorithm can use the above formula to select a schedule that is best at reaching the selected subtree in *the worst case* (see section 9.2.3).

9.2 Avoiding a Two-Leaves-Subtree

The idea of the previous section can be applied to the opposite goal, to avoid two-leaf-subtrees. For an algorithm to reach a good total expected processing time it should try to reach subtrees with low expected

processing time and avoid subtrees with high expected processing time. We will take a look at the latter here.

Before we start we take a closer look at the subtrees that result in a high expected processing time. As observed in section 8.4, the largest two-leaves-subtrees are not necessarily the ones having the highest weight. The following lemma will give us a small help in identifying the two-leaves-subtrees with the highest weight.

Lemma 9.2 (Two-Leaves-Subtrees with Highest Weight). *Given a tree B with size n the subtree $(a|k|l)$ having the highest weight $w(n,a,k,l)$ has at least one common leaf with B.*

Proof. Given any two-leaves-subtree $L = (a|k-1|l-1)$, where both leaves of L represent inner nodes in B, then there must exist another two-leaves-subtree $L' = (a|k|l)$ of B. The weight of L is $w(k-1,l-1,a,n)$, the weight of L' is $w(k,l,a,n)$. We will show that $w(k,l,a,n) \geq w(k-1,l-1,a,n)$.

$$\begin{aligned}
& w(k,l,a,n) \geq w(k-1,l-1,a,n) \\
\Leftrightarrow \quad & p(a,k,l) + \frac{n-a-k-l-1}{3} \geq p(a,k-1,l-1) + \frac{n-a-k-l-1+2}{3} \\
\Leftrightarrow \quad & q(k,l) + a \geq q(k-1,l-1) + a + \frac{2}{3} \\
\Leftrightarrow \quad & q(k,l) \geq q(k-1,l-1) + \frac{2}{3}
\end{aligned}$$

If $k > 1$ and $l > 1$, then we can evaluate the recursion of $q(k,l)$ once on each side:

$$\Leftrightarrow \quad \frac{1}{2}q(k-1,l) + \frac{1}{2}q(k,l-1) \geq \frac{1}{2}q(k-2,l-1) + \frac{1}{2}q(k-1,l-2) + \frac{2}{3}$$

The result can be splitted, such that if both equations hold true, the unsplit equation also holds true:

$$\begin{aligned}
\Leftarrow \quad & \frac{1}{2}q(k-1,l) \geq \frac{1}{2}q(k-2,l-1) + \frac{1}{2}\cdot\frac{2}{3} \\
\wedge \quad & \frac{1}{2}q(k,l-1) \geq \frac{1}{2}q(k-1,l-2) + \frac{1}{2}\cdot\frac{2}{3} \\
\Leftrightarrow \quad & q(k-1,l) \geq q(k-2,l-1) + \frac{2}{3} \\
\wedge \quad & q(k,l-1) \geq q(k-1,l-2) + \frac{2}{3}
\end{aligned}$$

Therefore we can reduce the parameters by one, while all parameters stay above zero.

If the equation holds for all pairs (k',l'), where either $k' < k \wedge l' \leq l$ or $k' \leq k \wedge l' < l$, then the equation also holds for k,l.

We only need to show that the equation holds for all pairs $(1,l)$ and $(k,1)$, $k,l \geq 1$.

For $k > 1$ and $l = 1$ we will prove the equation by induction on k:

$k = 1$:

$$q(0,0) + \frac{2}{3} = \frac{5}{3}$$

$$q(1,1) = \frac{5}{2} \geq \frac{5}{3}$$

$k > 1$:

$$\begin{aligned} q(k,1) &= \frac{1}{2} + \frac{1}{2}q(k-1,1) + \frac{1}{2}q(k,0) \\ &= \frac{1}{2} + \frac{1}{2}q(k-1,1) + \frac{1}{2}(k+1) \\ &\geq \frac{1}{2} + \frac{1}{2}\left(q(k-2,0) + \frac{2}{3}\right) + \frac{1}{2}(k+1) \\ &= \frac{1}{2} + \frac{1}{2}(k-1) + \frac{1}{2}(k+1) + \frac{1}{3} \end{aligned}$$

$$\begin{aligned} &= \frac{1}{2}(2k+1) + \frac{1}{3} \\ &= k + \frac{1}{2} + \frac{1}{3} \\ &= k + \frac{5}{6} \\ &\geq k + \frac{2}{3} \\ &= q(k-1,0) + \frac{2}{3} \end{aligned}$$

Since the case $k = 1$ and $l > 1$ is analogous, the equation $w(k,l,a,n) \geq w(k-1,l-1,a,n)$ holds for all $k > 0, l > 0$.

Therefore given any two-leaves-subtree, if we can "add" a leaf to each branch, the resulting subtree has a larger weight. □

Lemma 9.2 will allow us to focus on subtrees, where at most one leaf is an inner node of the super-tree.

We can now more easily find a heaviest two-leaves-subtree and we also know that the descends are all children, grandchildren, or descend from one leaf only. If an algorithm is working towards avoiding the highest weighted two-leaves-subtree, then it will schedule its leaves as early as possible. If there are descends above a leaf, these must be removed first, or all non-descends must be removed before all descends.

Unfortunately there is no known way to decide, whether it is better to remove non-descends and try to reach the higher tree, or whether it is better to remove descends and try to reach the lower tree. For both cases there exists a counter example. An example where $w(k,l+1,a,n) > w(k,l-1,a,n)$ is $w(8,7,a,n) = 58217/12288 + n/3 + a > 28991/6144 + n/3 + a = w(8,5,a,n)$. An example where $w(k,l+1,a,n) < w(k,l-1,a,n)$ is $w(8,6,a,n) = 1197/256 + n/3 + a > 3731/768 + n/3 + a = w(8,4,a,n)$.

We will *assume* that a strategy prefers the lower tree because the other tree leaf might be "accidently" removed, hence leading to a smaller tree after all. Also if the non-descends are removed, there is an earlier point with only two-leaves-left.

We will now calculate the probability P that L is reached under the assumption that it is avoided in favor of a smaller tree and that the worst case occurs (trees are reduced as quickly as possible to few leaves – similar to the assumptions of the preceding section).

9.2.1 Special Cases

The easy cases are treated first. These are the cases, where either no descends or no non-descends exist.

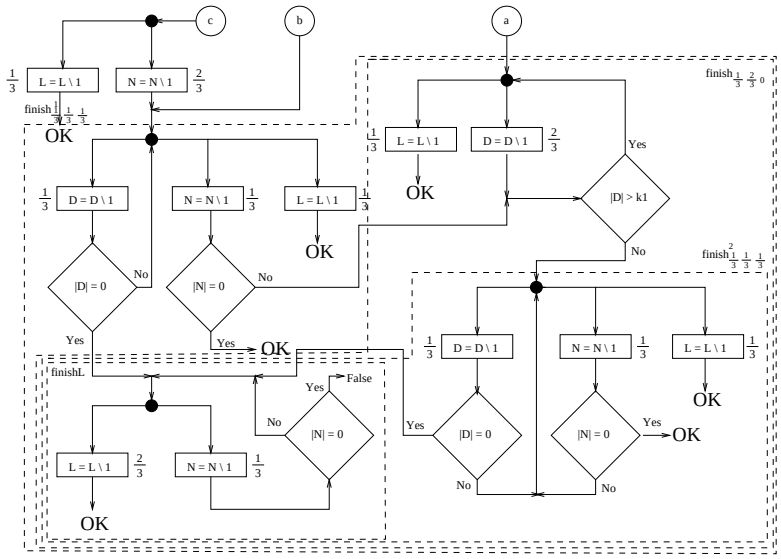


Figure 9.4: Formulae of Case 1 as Diagram

A) If $N = \emptyset$, then

(i) $D = \emptyset$

The tree is already reached, hence

$\Rightarrow P = 0$

(ii) $D \neq \emptyset$

As already mentioned in section 9.1.1, the tree can never be reached as the first two-leaves-subtree.

$\Rightarrow P = 1$

B) If $D = \emptyset$ and $|N| > 0$, then the leaves of L will be scheduled as soon as possible and L is reached only if all remaining nodes of N are finished before a single node from L is finished.

(i) $|\alpha \cap L| = 2$

L is only reached if N can be finished first with the remaining machine.

$\Rightarrow P = 1 - \left(\frac{1}{3}\right)^{|N|}$

(ii) $|\alpha \cap L| = 1$

L is only reached if N can be finished first with the remaining machines. In the first step there are two machines available, otherwise only one.

$\Rightarrow P = 1 - \frac{2}{3} \cdot \left(\frac{1}{3}\right)^{|N|-1}$

(iii) $|\alpha \cap L| = 0$

L is only reached if N can be finished first with the remaining machines. In the first step there are three, in the second step two machines available, otherwise only one.

$\Rightarrow P = 1 - \frac{2}{3} \cdot \left(\frac{1}{3}\right)^{|N|-2}$

9.2.2 General Cases

These are the cases, where neither $D = \emptyset$ nor $N = \emptyset$. Figures 9.4 and 9.5 give an outline of the possible paths similar to the diagrams in section 9.1. For the complexity estimation we will make the same assumptions as in section 9.1.

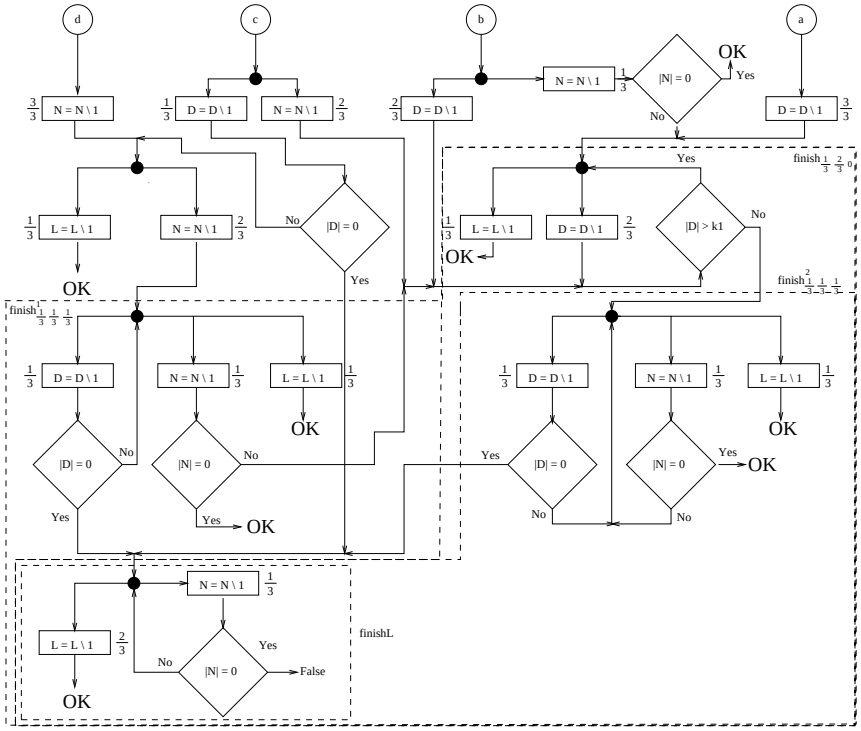


Figure 9.5: Formulae of Case 2 as Diagram

Case 1 If $|\alpha \cap L| = 1$, then all machines are tried to be assigned to D, until $D = \emptyset$. Let ΔN be the number of nodes already finished from N. Let ΔD be the number of nodes already finished from D.

The last chance to avoid L occurs when D is already finished, two leaves of L are scheduled and one leaf from N.

$$finishL^1(N, \Delta N) := 1 - \left(\frac{1}{3}\right)^{N-\Delta N}$$

The evaluation takes $\mathcal{O}(1)$ steps (remember that we calculate binomials and powers beforehand).

Before that a number of times the probability for each set may be equal. We can finish Ns and Ds, until either a node from L is finished, the last node from N is finished, or the last node from D is finished.

We assume that scheduling D is preferred to scheduling N. Then there may be a point, where there is only one leaf left in D. The following function captures this:

$$finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}{}^{2}(N,\Delta N,D,\Delta D) :=$$
$$\frac{1}{3}\cdot\sum_{i=0}^{D-\Delta D-1}\sum_{j=0}^{N-\Delta N-1}\left(\frac{1}{3}\right)^{i+j}\binom{i+j}{j}$$
$$+\sum_{i=0}^{D-\Delta D-1}\left(\frac{1}{3}\right)^{i+N-\Delta N}\binom{i+N-\Delta N}{i}$$
$$+\sum_{i=0}^{N-\Delta N-1}\left(\frac{1}{3}\right)^{i+D-\Delta D}\binom{i+D-\Delta D}{i}\cdot finishL^{1}(N,\Delta N+i)$$

The evaluation of the above sum takes $\mathcal{O}(n^2)$ steps.

If D has more than one leaf, two machines can be assigned to D for some time, which is captured in

$$finish^{1}_{\frac{1}{3},\frac{2}{3},0}(k_1,N,\Delta N,D,\Delta D) :=$$
$$\begin{cases}1-\left(\frac{2}{3}\right)^{D-\Delta D-k_1}+ & \\ \left(\frac{2}{3}\right)^{D-\Delta D-k_1}\cdot finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}{}^{2}(N,\Delta N,D,D-k_1) & \text{if } D-\Delta D>k_1,\\ finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}{}^{2}(N,\Delta N,D,\Delta D) & \text{otherwise.}\end{cases}$$

The evaluation of the above sum takes $\mathcal{O}(n^2)$ steps because of its call to $finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}{}^{2}$.

The case that a node from N is scheduled at the beginning and the machine can potentially be assigned to a node in D is described by

$$finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}{}^{1}(k_1,N,\Delta N,D,\Delta D) :=$$
$$\frac{1}{2}-\frac{1}{2}\left(\frac{1}{3}\right)^{D-\Delta D}+$$
$$\left(\frac{1}{3}\right)^{D-\Delta D}finishL^{2}(N,\Delta N)+$$
$$\begin{cases}\frac{1}{2}-\frac{1}{2}\left(\frac{1}{3}\right)^{D-\Delta D} & \text{if } N-\Delta N=1,\\ \frac{1}{3}\sum_{i=0}^{D-\Delta D-1}\left(\frac{1}{3}\right)^{i}finish^{2}_{\frac{1}{3},\frac{2}{3},0}(k_1,N,\Delta N+1,D,\Delta D+i) & \text{otherwise.}\end{cases}$$

Because of the repeated calls to $finish^{2}_{\frac{1}{3},\frac{2}{3},0}$ the evaluation may take $\mathcal{O}(n^3)$ steps.

Using the above formulae, the probabilities can be estimated in $\mathcal{O}(n^3)$ steps for this case.

(a) $|\alpha\cap N|=0$
$$\Rightarrow P=finish^{2}_{\frac{1}{3},\frac{2}{3},0}(k_1,N,0,D,0)$$

(b) $|\alpha\cap N|=1$
$$\Rightarrow P=finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}{}^{1}(k_1,N,0,D,0)$$

(c) $|\alpha \cap N| = 2$

$$\Rightarrow P = \frac{1}{3} + \frac{2}{3} \cdot finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}(k_1, N, 1, D, 0)$$

Case 2 If $|\alpha \cap L| = 0$, then the first machine to finish is assigned to L. After that all machines are tried to be assigned to D, until $D = \emptyset$.

From the diagram for this case (Figure 9.5), we can observe that this case has the same building block functions as the previous case.

$$finishL^2(N, \Delta N) = finishL^1(N, \Delta N)$$

$$finish^{2}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}(N, \Delta N, D, \Delta D) = finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}(N, \Delta N, D, \Delta D)$$

$$finish^{2}_{\frac{1}{3},\frac{2}{3},0}(k_1, N, \Delta N, D, \Delta D) = finish^{1}_{\frac{1}{3},\frac{2}{3},0}(k_1, N, \Delta N, D, \Delta D)$$

$$finish^{2}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}(k_1, N, \Delta N, D, \Delta D) \quad = \quad finish^{1}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}(k_1, N, \Delta N, D, \Delta D)$$

The estimated probabilities can then be calculated as follows. Each case needs $\mathcal{O}(n^3)$ steps.

(a) $|\alpha \cap N| = 0$

$$\Rightarrow P = finish^{2}_{\frac{1}{3},\frac{2}{3},0}(k_1, N, 0, D, 1)$$

(b) $|\alpha \cap N| = 1$

$$\Rightarrow P = \frac{2}{3} \cdot finish^{2}_{\frac{1}{3},\frac{2}{3},0}(k_1, N, 0, D, 1) + \frac{1}{3} \cdot \begin{cases} 1 & \text{if } |N| = 1, \\ finish^{2}_{\frac{1}{3},\frac{2}{3},0}(k_1, N, 1, D, 0) & \text{otherwise.} \end{cases}$$

(c) $|\alpha \cap N| = 2$

$$\Rightarrow P = \frac{2}{3} \cdot finish^{2}_{\frac{1}{3},\frac{2}{3},0}(k_1, N, 1, D, 0) + \frac{1}{3} \cdot \begin{cases} finishL^2(N, 0) & \text{if } |D| = 1, \\ \frac{1}{3} + \frac{2}{3} \cdot finish^{2}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}(k_1, N, 1, D, 1) & \text{otherwise} \end{cases}$$

(d) $|\alpha \cap N| = 3$

$$\Rightarrow P = \frac{1}{3} + \frac{2}{3} \cdot finish^{2}_{\frac{1}{3},\frac{1}{3},\frac{1}{3}}(k_1, N, 2, D, 1)$$

9.2.3 Usage in an Algorithm

Evaluating the above formulae for a tree B and the heaviest two-leaves-subtree (or any other subtree with a common leaf with B) takes $\mathcal{O}(n^3)$ steps (including all binomials and powers).

The above formulae can be used to select a schedule that minimizes the *worst case* probability of reaching a heavy subtree.

9.3 Performance of Resulting Algorithms

The formulae from the previous sections can be used to approximate probabilities of reaching two-leaves-subtrees under different assumptions (either trying to avoid or trying to reach a subtree).

The formulae need a concrete subtree of B (e.g. defined by two nodes) for evaluation to bound l_1, l_2, l_3, k_1, k_2, and k_3. The results for each concrete subtree can be combined to results for a class of similar subtrees. The results for classes can be combined to a total result which is then minimized or maximized.

Hence there are three questions that need to be answered for a specific algorithm based on the above formulae:

1. Which classes of trees are taken into account (with respect to their weight $w(k, l, a, n)$)?
   a) For avoiding subtrees
   b) For trying to reach subtrees
2. How are the results for a class of subtrees combined?
   a) For avoiding subtrees
   b) For trying to reach subtrees
3. How are the results for all classes of subtrees combined and what value is minimized/maximized?

Intuitively the question 2 should be answered such that we add the probabilities, if we are trying to reach the subtrees of a given class (because we can reach subtree one or subtree two or ... to reach the class), and that we multiply the probabilities, if we are trying to avoid the subtrees of a given class (because we do not want to reach subtree one nor subtree two, nor ...). Empirically this was verified by the fact that if we were trying to avoid (reach) the heaviest (lightest) two-leaves-subtree and minimize (maximize) the probability that it is reached, then taking the product (sum) of all probabilities calculated for the representatives of a class performs best among the operations minimum, maximum, sum, and product.

The answer to question 1 is not as straight forward and is closely related to question 3. If we only choose to look at the class of heaviest subtrees, we need not combine any class results. If we look at all classes of subtrees we can either sum up the results, compare them in lexicographical order or weight them in another way.

Figure 9.6 shows an overview of the performance of selected algorithms. The first column describes an algorithm (or a question), the remaining columns give the number of non-optimal (or can-optimal) solutions found for each set of trees with k nodes . The first two lines give the number of trees and the number of trees with more than three leaves (the only ones where an algorithm is needed). The next two lines give the performance of an algorithm that just chooses the leaves with the highest or lowest DFS number (e.g. an arbitrary algorithm). The fifth line gives the performance of HLF for comparison (note, that there are only eight non-optimal trees, the remaining ones are at least can-optimal).

The algorithms 1 – 3 choose a single subtree-class and take the solution based on the probability of reaching the subtree (either while avoiding it or while trying to reach it). Interestingly, the algorithm avoiding the lightest tree performs better than the one avoiding the heaviest tree.

The algorithms 4 – 7 approximate probabilities for all classes of subtrees and decide on a sum weighted by the subtree weight $w(k, l, a, n)$. All four perform worse than the DFS-choosing algorithms!

The algorithms 8 and 9 are based on the idea to calculate all probabilities and to compare them as a vector in lexicographical order (where they are ordered by subtree weight $w(k, l, a, n)$ either ascending for algorithm 9 or descending for algorithm 8).

Figure 9.6 is by no means complete. There are many other possible ways to combine the approximated probabilities. We only give canonical and interesting combinations, others were tried but had equally disappointing results. Even combining probabilities for trying to avoid and trying to reach a tree (e.g. try to avoid heaviest while reaching lightest tree) did not enhance the results.

Algorithm (or Question)	Failures on Trees With k Nodes								
	4	5	6	7	8	9	10	11	12
Q: Number of trees	4	9	20	48	115	286	719	1842	4766
Q: Trees with more than 4 leaves		1	5	20	67	207	595	1655	4494
Choose leaves by DFS			3	15	56	180	533	n/a	n/a
Choose leaves by DFS (reverse)				1	10	46	175	n/a	n/a
HLF				1	8	33	116	372	1130
1. 1b=lightest, 2b=sum, 3=$\max(p)$				1	4	22	100	416	1568
2. 1a=heaviest, 2a=prod, 3=$\min(p)$			3	15	58	189	566	n/a	n/a
3. 1a=lightest, 2a=prod, 3=$\min(p)$					1	14	102	484	1861
4. 1a=all, 2a=prod, 3=$\min(\sum p \cdot w)$			3	15	58	189	566	n/a	n/a
5. 1a=all, 2a=prod, 3=$\max(\sum p \cdot w)$				1	12	50	188	n/a	n/a
6. 1b=all, 2b=sum, 3=$\max(\sum p \cdot w)$			3	15	58	189	566	n/a	n/a
7. 1b=all, 2b=sum, 3=$\min(\sum p \cdot w)$				1	12	55	199	n/a	n/a
8. 1a=all, 2a=prod, 3=$\min(\bar{p}_>)$			3	15	58	189	566	n/a	n/a
9. 1b=all, 2b=sum, 3=$\max(\bar{p}_<)$					1	8	47	206	785

Figure 9.6: Results for Various Parameterized Algorithms Based on Approximated Subtree Probabilities

One tree that seemed very hard for a lot of algorithms is shown in Figure 9.7. There are basically two scheduling alternatives, either scheduling nodes 2 (or 4), 6, and 7, or scheduling nodes 2, 4, and 6 (or 7). The latter is optimal with an expected value of $1471/324$, the former has the expected value of $1472/324$.

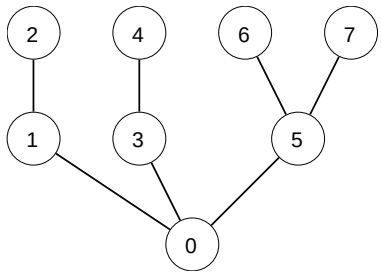


Figure 9.7: Counter Example for "Heavy-Tree-Avoidance" Strategy.

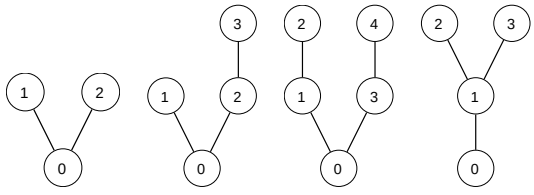


Figure 9.8: Two-Leaves-Subtrees of Tree in Figure 9.7

The two-leaves-subtrees of the tree in Figure 9.7 are shown in Figure 9.8. The table in Figure 9.9 gives the weight of each subtree and the probability of reaching it under the two different schedules. Obviously the probability of reaching the heaviest subtree is higher under the optimal schedule, which is compensated by the lower probability of reaching the second heaviest subtree. To no surprise, the same tree appears again

Subtree	Weight	Schedule 2,7,8 Probability	Schedule 2,4,7 Probability
a	50/12	2/9	2/9
b	55/12	38/81	40/81
c	57/12	8/27	7/27
d	58/12	1/81	2/81

Figure 9.9: Weights and Probabilities for Subtrees in Figure 9.8

in section 10.1 as one of the trees having a smallest difference between two schedules.

Another reason for the failure of this approach might lie in the fact that each class of two-leaves-subtrees (e.g. $(0|2|2)$) is represented multiple times. Furthermore, some nodes take part in more instances than others and some of these nodes are no leaves. The further down a node is in the tree, the harder it seems to predict the success of schedules with respect to that node.

The formulae of sections 9.1 and 9.2 are inexact in multiple ways. Using a reasonable algorithm, the probability of being always left with the largest two-leaves- or one-leaf-subtree is rather small. On the other hand, the leaves cannot be scheduled in arbitrary order as the binomials in the formulae suggest. All this seems to lead to algorithms that perform hardly better than the DFS controlled selection of leaves.

Chapter 10

A Time-Based Approach

10.1 The Computation Tree and the Size of Numbers

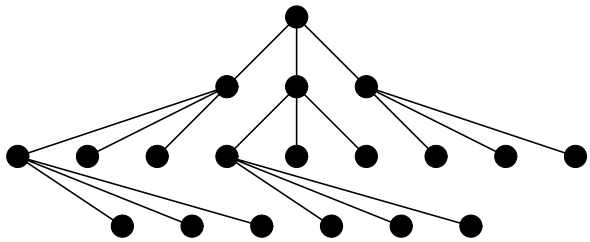
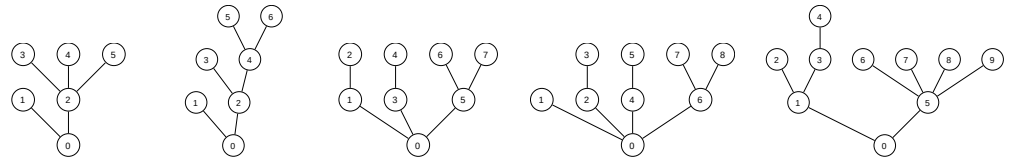


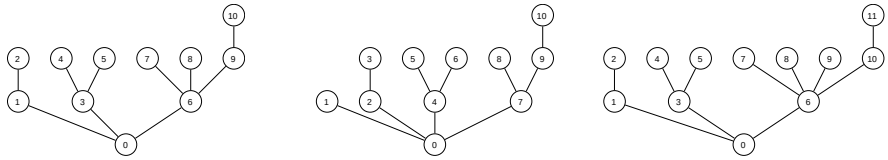
Figure 10.1: Example for a "Computation Tree"

Let the computation tree $CT_\alpha(B)$ of a tree B be the tree of possible states under the schedule α, where a state is defined by a subtree, and where the leaves are the empty subtrees. $CT_\alpha(B)$ has depth $|B|+1$, it is essentially the DAG of subtrees converted to a tree by ignoring isomorphism between subtrees, not merging nodes, and selecting only reached subtrees. We can label each edge with the inverse of the number of nodes scheduled in the tree represented by its parent (the expected length of the corresponding time interval in the execution of the schedule). Let each node be labeled additionally with the sum of the edge labels and the product of the edge labels from the root to that node. The expected processing time of a path represented by a leaf is the sum of the edge labels. The probability of reaching the leaf is the product of the edge labels. The total expected processing time is then the sum of the products of both labels of all leaves. Let $CT^2_\alpha(B)$ be $CT_\alpha(B)$ pruned at the nodes representing two-leaves-subtrees of B. All edges in $CT^{\,2}_\alpha(B)$ are labeled with $1/3$. Let the leaves be labeled with the weight $w(k,l,a,|B|)$ of the represented subtree with respect to B (as defined in section 8.3). The depth of all leaves representing isomorphic subtrees is the same, and since all edges are labeled with $1/3$, the probabilities of reaching these trees are all the same ($(\frac{1}{3})^{\text{depth}}$). Let there be $occ(a|k|l)$ leaves representing the class of two-leaves-subtrees $(a|k|l)$. $CT^2_\alpha(B)$ is clearly a DAG, in which the leaves obviously form a chain. By Theorem 8.1, the total expected processing time can be expressed as

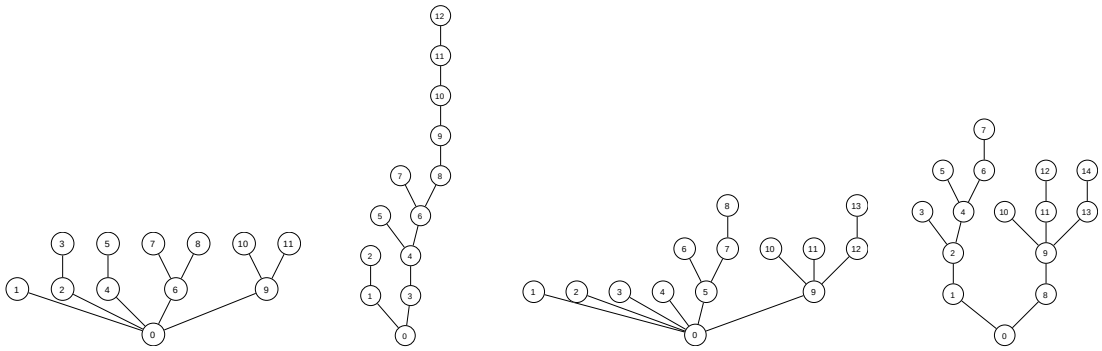
$$\mathbb{E}(T_\alpha(B)) = \sum_{(a\,|k\,|l)\ \text{subtree class in } B} occ(a|k|l) \cdot \left(\frac{1}{3}\right)^{|B|-k-l-a-1} \cdot w(k,l,a,|B|) \qquad (10.1)$$



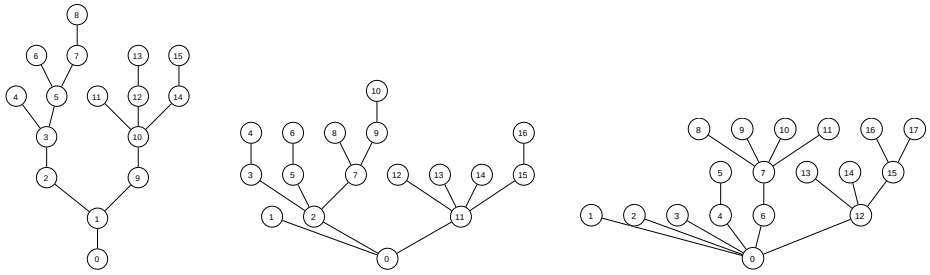
(a) 6 Nodes (b) 7 Nodes (c) 8 Nodes (d) 9 Nodes (e) 10 Nodes



(f) 11 Nodes – relative (g) 11 Nodes – absolute (h) 12 Nodes – relative



(i) 12 Nodes – absolute (j) 13 Nodes (k) 14 Nodes (l) 15 Nodes



(m) 16 Nodes (n) 17 Nodes (o) 18 Nodes

Figure 10.2: Trees with Minimal Expected Processing Time Differences for Different Schedules

Tree	3^{-n-3}	Best Time and Schedule	2nd Best Time and Schedule	Absolute Difference	Relative Difference
Figure 10.2 (a)	$3.7037 \cdot 10^{-2}$	$\{3,4,5\}$ 4.0	$\{1,4,5\}$ 4.0555555555555553582	$5.5555555555555358183 \cdot 10^{-2}$	$1.3888888888888839546 \cdot 10^{-2}$
Figure 10.2 (b)	$1.2345 \cdot 10^{-2}$	$\{3,5,6\}$ 4.7592592592592595224	$\{1,5,6\}$ 4.7870370370370372015	$2.7777777777777679091 \cdot 10^{-2}$	$5.836575875486360587 3 \cdot 10^{-3}$
Figure 10.2 (c)	$4.1152 \cdot 10^{-3}$	$\{2,4,7\}$ 4.5401234567901234129	$\{4,6,7\}$ 4.5432098765432096243	$3.0864197530862114149 \cdot 10^{-3}$	$6.798096532970309162 \cdot 10^{-4}$
Figure 10.2 (d)	$1.3717 \cdot 10^{-3}$	$\{3,5,8\}$ 4.747942386831275563	$\{5,7,8\}$ 4.7489711934156382256	$1.0288065843626625906 \cdot 10^{-3}$	$2.1668472372708734788 \cdot 10^{-4}$
Figure 10.2 (e)	$4.5725 \cdot 10^{-4}$	$\{4,8,9\}$ 5.4478737997256514447	$\{2,4,9\}$ 5.4481310013717427765	$2.5720164609133178146 \cdot 10^{-4}$	$4.7211381090414419133 \cdot 10^{-5}$
Figure 10.2 (f)	$1.5242 \cdot 10^{-4}$	$\{2,5,10\}$ 5.5891632373113848686	$\{4,5,10\}$ 5.5892775491540920285	n/a	$2.0452407248378835569 \cdot 10^{-5}$
Figure 10.2 (g)	$1.5242 \cdot 10^{-4}$	$\{3,6,10\}$ 5.4847584209724127291	$\{5,6,10\}$ 5.4848727328151190008	$1.1431184270627170463 \cdot 10^{-4}$	n/a
Figure 10.2 (h)	$5.0805 \cdot 10^{-5}$	$\{2,5,11\}$ 5.8988721231519578581	$\{4,5,11\}$ 5.8989102270995275035	n/a	$6.4595310381614519832 \cdot 10^{-6}$
Figure 10.2 (i)	$5.0805 \cdot 10^{-5}$	$\{3,5,8\}$ 5.6648757811309247145	$\{5,10,11\}$ 5.6649138850784934718	$3.8103947568757234876 \cdot 10^{-5}$	n/a
Figure 10.2 (j)	$1.6935 \cdot 10^{-5}$	$\{5,7,12\}$ 9.0803743236549312456	$\{2,7,12\}$ 9.0803844450160031698	$1.0121361071924184216 \cdot 10^{-5}$	$1.1146413915511630655 \cdot 10^{-6}$
Figure 10.2 (k)	$5.6450 \cdot 10^{-6}$	$\{8,11,13\}$ 6.3338018707626995152	$\{6,8,13\}$ 6.3338039876486753599	$2.1168859758446956221 \cdot 10^{-6}$	$3.3422042858909098194 \cdot 10^{-7}$
Figure 10.2 (l)	$1.8817 \cdot 10^{-6}$	$\{7,12,14\}$ 7.744037393732323693	$\{5,7,12\}$ 7.744037459885010577	$6.6152686883924616268 \cdot 10^{-8}$	$8.5424028217458808394 \cdot 10^{-9}$
Figure 10.2 (m)	$6.2723 \cdot 10^{-7}$	$\{8,13,15\}$ 8.7440373937323254694	$\{6,8,13\}$ 8.7440374598850123533	$6.6152686883924616268 \cdot 10^{-8}$	$7.5654624866245974647 \cdot 10^{-9}$
Figure 10.2 (n)	$2.0908 \cdot 10^{-7}$	$\{4,6,10\}$ 7.6064558305730187726	$\{6,10,16\}$ 7.6064558763082095183	$4.573519074568821452 \cdot 10^{-8}$	$6.012680775961705 2044 \cdot 10^{-9}$
Figure 10.2 (o)	$6.9692 \cdot 10^{-8}$	$\{9,10,11\}$ 7.6668007361118162279	$\{11,16,17\}$ 7.6668007361118171161	$8.8817841970012523234 \cdot 10^{-16}$	$1.1584733323205696524 \cdot 10^{-16}$

Figure 10.3: Minimal Expected Processing Times for Different Schedules

If we can calculate bounds on $occ(a|k|l)$ or on the $\mathcal{O}(n^2)$ two-leaves-trees individually, we can also bound the optimal value of the expected total processing time for a given tree B. The formulae in sections 9.1 and 9.2 can be interpreted just as such.

From equation 10.1 we can also see that the number representing the total expected processing time for a tree B can grow quite large if expressed as an exact fraction. An upper bound for the total expected processing time is $n = |B|$. The smallest possible occurring two-leaves-subtree is $(0, 1, 1)$. Its last factor in the upper sum is therefore $\left(\frac{1}{3}\right)^{n-3}$. The resulting total expected processing time might hence be a fraction with a maximal denominator of 3^{n-3}, hence the maximal numerator is $n \cdot 3^{n-3}$. Using this upper bound, an upper bound on the size of the occurring numbers is

$$|\log_2(3^{n-3})| + |\log_2(n \cdot 3^{n-3})| = (n-3)\log_2(3) + (n-3)\log_2(3) + \log_2(n) = \mathcal{O}(n).$$

If numbers of this size occur, the unit cost model might no longer be appropriate. If the input size stays below twenty nodes, then 3^n fits into 32 bits and the results are exact to some extend. Above that size each arithmetic operation may take $\mathcal{O}(n)$ (multiplication and division even $\mathcal{O}(n \log_2(n))$). And this additional factor must be considered, unless some smaller bound on the numbers can be obtained. The result of Papadimitriou and Tsitsiklis (in [PT87], see Theorem 7.1 in this paper) can also be read as "the difference between the optimal strategy and another (namely HLF) can become arbitrarily small". If an optimal algorithm considers such other solutions it might well be faced with very small differences (as low as $3^{-(n-3)}$). Figure 10.2 and Figure 10.3 support this hypothesis. It seems that we are already dealing with rounding errors in the trees with 18 nodes. The IEEE 754 double precision values have a fractional part of 52 bits, which suffices for exact values of sizes below 16 decimal digits.

10.2 Motivation

If we look at a concrete two-leaves-subtree S defined by its two leaves i, j, then there are four important decision points during the scheduling of B. At decision point t_1 the first leaf is scheduled, at decision point t_2 the second leaf is scheduled, at decision point t_3 one of the scheduled leaves is finished, and at decision point t_4 the remaining subtree is S (S is reached). See Figure 10.4 for a schematic view of this. Of course not all these points can (or do) occur for every subtree. If the subtree is reached, it will at least go through t_1, t_2, and t_4 ($t_1 = t_2$ only if both are 0). If the tree is not reached (i.e. a leaf of S is finished while there are still three leaves left in B), then the point t_1 might occur, if t_1 occurs, t_2 might occur, and if t_1 occurs, t_3 might occur.

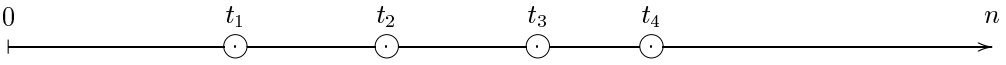


Figure 10.4: Schematic view of the Time Line of a Concrete Schedule.

Suppose, only t_1 and t_3 occur. What is the expected distance between them? The probability that $t_3 - t_1 = k$ is $\frac{1}{3} \cdot \left(\frac{2}{3}\right)^{k-1}$. Hence the expected value of k is

$$\sum_{k\geq 1} k \cdot \frac{1}{3} \cdot \left(\frac{2}{3}\right)^{k-1} = \frac{1}{3} \cdot \sum_{k\geq 0} k \cdot \left(\frac{2}{3}\right)^{k-1} = \frac{1}{3} \cdot \frac{1}{\left(1-\frac{2}{3}\right)^2} = 3.$$

If t_1, t_2, and t_3 occur, the expected distance between t_2 and t_3 is

$$\sum_{k\geq 1} k \cdot \frac{2}{3} \cdot \left(\frac{1}{3}\right)^{k-1} = \frac{2}{3} \cdot \frac{1}{\left(1-\frac{1}{3}\right)^2} = \frac{3}{2}.$$

The expected distance between t_1 and t_3 in this case depends upon the distance between t_1 and t_2, which in turn depends on the scheduling strategy. These values are no surprise: the expected processing time of a task is $\frac{1}{\lambda}$. With three (or two) machines every $\frac{1}{3\lambda}$ ($\frac{1}{2\lambda}$) a decision point occurs. Dividing gives the above fractions.

10.3 Total Expected Processing Time

In section 10.1 we defined the computation tree $CT^2_\alpha(B)$ for tree B and strategy α and derived the equation

$$\mathbb{E}(T_\alpha(B)) = \sum_{(a|k|l) \text{ subtree in } B} occ(a|k|l) \cdot \left(\frac{1}{3}\right)^{|B|-k-l-a-1} \cdot w(k,l,a,|B|)$$

Obviously, there can be multiple concrete subtrees in the class of two-leaves-subtrees $(a|k|l)$. Each concrete two-leaves-subtree can be uniquely identified by its two leaves. Let $occ(l_1, l_2)$ be the number of times that a two-leaves-subtree with leaves l_1 and l_2 occurs in $CT^2(B)$. Let $size(l_1, l_2)$ be the number of nodes in the subtree. For a given tree B there is also an easy mapping M between two nodes n_1, n_2 (that are no ancestors) and the class $(a|k|l)$ of two-leaves-subtrees that the such-defined subtree belongs to. The above equation can be rewritten as

$$\mathbb{E}(T_\alpha(B)) = \sum_{n_1,n_2 \in B, \exists path(n_1,n_2)} occ(n_1,n_2) \cdot \left(\frac{1}{3}\right)^{|B|-size(n_1,n_2)} \cdot w(M(n_1,n_2),|B|)$$

For a given strategy α, a tree B with nodes $\{1,\ldots,n\}$, let t_i be the order number of node i, i.e. node i is scheduled as t_i-th node. t_i does not depend on the absolute processing lengths of the tasks, but rather on the finishing order of the tasks (starting with a fixed schedule, each order defines a unique resulting subtree that determines the next node to be scheduled by α).

For nodes i,j and fixed t_i, t_j we can calculate the probability $P[i,j]$ that the two-leaves-subtree will not occur (without loss of generality $t_i < t_j$):

$$\begin{aligned}
P[i,j] &= P_{i \text{ is finished before } j \text{ is scheduled}} + (1 - P_{i \text{ is finished before } j \text{ is scheduled}}) \cdot P_{j \text{ is finished before there are no other leaves left}} \\
&= \left(\sum_{k=1}^{t_2-t_1} \left(\frac{2}{3}\right)^{k-1} \cdot \frac{1}{3}\right) + \left(1 - \sum_{k=1}^{t_2-t_1} \left(\frac{2}{3}\right)^{k-1} \cdot \frac{1}{3}\right) \cdot \left(\sum_{k=1}^{|B|-size(i,j)-t_2} \left(\frac{1}{3}\right)^{k-1} \frac{2}{3}\right) \\
&= \left(1 - \left(\frac{2}{3}\right)^{t_2-t_1}\right) + \left(\left(\frac{2}{3}\right)^{t_2-t_1}\right) \cdot \left(1 - \left(\frac{1}{3}\right)^{|B|-size(i,j)-t_2}\right) \\
&= 1 - \left(\frac{2}{3}\right)^{t_2-t_1} + \left(\frac{2}{3}\right)^{t_2-t_1} - \left(\frac{2}{3}\right)^{t_2-t_1} \cdot \left(\frac{1}{3}\right)^{|B|-size(i,j)-t_2} \\
&= 1 - \left(\frac{2}{3}\right)^{t_2-t_1} \cdot \left(\frac{1}{3}\right)^{|B|-size(i,j)-t_2} \\
&= 1 - 2^{t_2} \cdot \left(\frac{3}{2}\right)^{t_1} \cdot \left(\frac{1}{3}\right)^{|B|-size(i,j)}
\end{aligned}$$

Hence, the probability that a subtree will occur under a given scheduling order is $2^{t_2} \cdot \left(\frac{3}{2}\right)^{t_1} \cdot \left(\frac{1}{3}\right)^{|B|-size(i,j)}$ – the later the first (or the second) node is scheduled, the higher the probability that the tree will occur. One can also see from this equation that the probability that a two-leaves-subtree occurs grows with its size by the factor three per node. Also, if the node of a subtree is scheduled, then the probability decreases at least by two-thirds or by one-half (in comparison to not scheduling the node in a given turn). When the probability that a two-leaves-subtree for nodes i,j occurs decreases, the corresponding tree occurs less often and $occ(i,j)$ decreases (the opposite is also true).

10.4 Resulting Algorithms

We will use the above formulae in Algorithm 11. The basic idea is to try to decrease the total expected processing time by reducing the probabilities reaching a subset of subtrees that contributes a largest part. A subtree contributes more if it has a large weight and is relatively large at the same time (the smaller the subtree the smaller is the "general" probability of reaching it – the factor $\left(\frac{1}{3}\right)^{|B|-size(i,j)}$). The algorithm iteratively schedules leaves that belong to such strongly contributing subtrees. After selecting a leaf the remaining subtree weights are adjusted.

In detail, we are only looking at feasible subtrees. A subtree is considered feasible, if it has a leaf in common with B and if it can be replaced by a tree with a smaller weight (that is it can be avoided in favor of something). All these trees are weighted (with the weights from section 8.3) and the weight is discounted by their "general" probability (the size induced factor $\left(\frac{1}{3}\right)^{|B|-size(i,j)}$).

The weight adjustment for a scheduled leaf k replaces the subtrees' (that k is a leaf of) weights by two-thirds of its weight (because the probability of reaching decreases by two-thirds for each time interval the first node is scheduled earlier – if a node is not scheduled at the current decision point it is scheduled the earliest one interval later).

Algorithm 11 A Discounted-Weight-Based Algorithm.

1: Let S be the set of all concrete subtrees that can be avoided in favor of a lighter tree.
2: Let $\forall i \in B : w_n(i) = 0$ be an initially zero node weight.
3: **for all** $t \in S$ **do**
4: Let t be of type $M(t) = (a|k|l)$.
5: Let S be the size of t
6: Let $w = w(k, l, a, |B|) \cdot 3^s$
7: Let t's leaves be i, j.
8: **if** i and j are scheduled **then**
9: $w = 0$
10: **else**
11: **if** i is scheduled **then**
12: $w = \frac{2}{3} \cdot w$
13: **end if**
14: **if** j is scheduled **then**
15: $w = \frac{2}{3} \cdot w$
16: **end if**
17: **end if**
18: Set $w_n(i) = w_n(i) + w$
19: Set $w_n(j) = w_n(j) + w$
20: **end for**
21: Select node k with highest weight
22: If less than three leaves are selected goto 1

The first tree that Algorithm 11 fails on is shown in Figure 10.5 (a) (the optimal schedule is $\alpha_{OPT} = \{5, 6, 7\}$, while Algorithm 11 schedules leaf 2).

Although Algorithm 11 performs worse than the best algorithm so far (see Figure 9.6), it has a better running time (Evaluation of algorithm number 9 takes time $\mathcal{O}(n^7)$, while Algorithm 11 can be evaluated in $\mathcal{O}(n^3)$).

We know that HLF performs asymptotically well, so it seems reasonable to try to keep the properties from Theorem 7.1 while improving the performance. The theorem holds for any HLF strategy, so we will simply adjust the strategy so that tie-breaks are solved in a favorable way. An algorithm based on that idea is given as Algorithm 12. The exchange steps are performed in line 10 based on the comparison of two nodes i, j under the assumption that either i is scheduled before j or vice versa. If one of the nodes is scheduled before the other, then the probabilities of the two-leaves-subtree associated with that node are decreased

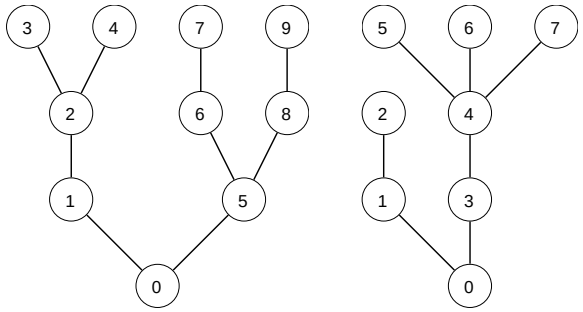


Figure 10.5: First Tree that (a) Algorithm 11 or (b) Algorithm 12 Fails on

by at least a factor $2/3$ (or $1/2$ if another node in that tree is already scheduled from a previous interval). Hence the difference in the contribution is $1/3$ ($1/2$, respectively) of the weight (which we discount by the "general" probability as above). The nodes with the largest contributions are preferred.

Algorithm 12 Improving HLF Through Exchange Steps.

Let $(a_i)_{1 \leq i \leq n}$ be an array with all nodes from B.
Sort $(a_i)_{1 \leq i \leq n}$ by the height of the leaves, s.t. $\forall i < j \in leaves(B) : height(a_i) \geq height(a_j)$ and $\forall a_i \in leaves(B), a_j \in B \setminus leaves(B) : i < j$.
$start \longleftarrow 1$
$end \longleftarrow 1$
while $end \leq 3 \wedge a_{start} \in leaves(B)$ **do**
  while $height(a_{start}) = height(a_{end+1})$ **do**
    $end \longleftarrow end + 1$
  end while
  $end \longleftarrow end + 1$ /* nodes in interval $[start, \ldots, end)$ all have the same height */
  Sort $(a_i)_{start \leq i \leq end}$ by the minimal difference in the contribution of their subtrees to the total expected processing time when preferring the one over the other.
end while
Schedule leaves a_1, a_2, a_3.

The algorithm can collect the two-leaves-subtrees for each node at the beginning and calculate all the contributions of all nodes of an interval before sorting. Algorithm 12 can thus be evaluated in $\mathcal{O}(n^3)$.

As can be seen in Figure 9.6, Algorithm 12 performs better than any algorithm seen so far (while still maintaining the HLF property). The first tree that Algorithm 12 fails on is shown in Figure 10.5 (b) (the optimal schedule is $\alpha_{OPT} = \{3, 4, 9\}$, while Algorithm 12 schedules leaf $\{4, 7, 9\}$).

Algorithm (or Question)	Failures on Trees With k Nodes								
	4	5	6	7	8	9	10	11	12
Q: Number of trees	4	9	20	48	115	286	719	1842	4766
Q: Trees with more than 4 leaves		1	5	20	67	207	595	1655	4494
HLF				1	8	33	116	372	1130
Algorithm 11					1	10	55	232	847
Algorithm 12							1	8	31

Figure 10.6: Comparison of Previous Algorithms to the New Time-Based Ones

Chapter 11

Conclusion and Outlook

In this work we have dealt with the specific problem of scheduling tasks with independent identically, exponentially distributed processing times and in-tree constraints on three processors in parallel. From [CR75] we knew that an HLF strategy is not optimal, although it is optimal in the deterministic case [Hu61]. On the other hand, the HLF strategy can serve as an – asymptotically optimal – approximation. An algorithm would therefore need to fill the gap between the HLF approximation and the optimal solution while still being of reasonable complexity (the amount of time saved by an optimal solution compared to the time needed to calculate the optimal solution).

The only optimal algorithm found is described in chapter 6. The running time of this algorithm could be enhanced over the naive recursive version by a dynamic programming approach and the elimination of redundancies and isomorphic trees. This led to an improvement of the main term in the asymptotic running time from about 3^n to approximately 1.6^n. The algorithm still has exponential running time and its use for larger trees is hence prohibitive. The algorithm never the less served well in generating a large amount of solutions for trees with eighteen nodes or less, which were used to check against other strategies.

Figure 11.1 gives an overview of the tested strategies and their performance. There are four categories of algorithms shown in the figure. The first pair is selected from the HLF strategies, the second from Monte Carlo techniques, the third from the algorithms based on combinatorial estimation of the probabilities of reaching two-leaves-subtrees (chapter 9), and the last pair is based on node weights calculated from a time based view of the relation between scheduled nodes and two-leaves-subtrees (chapter 10). By means of an example we have shown that no static list scheduling strategy can be optimal (see section 7.4). The HLF strategy is not optimal either and we have found a smaller example that confirms this. On the other hand, it was shown by Papadimitriou and Tsitsiklis [PT87] that the HLF strategy is asymptotically optimal. The strategies can therefore be divided into strategies which may be made optimal but have no proven bound (3, 4, 5, 6, 8 in Figure 11.1) and strategies based on HLF that cannot be optimal but have a proven bound (1, 2, 7 in Figure 11.1). An apparent precondition for an optimal strategy is that the strategy should at least be able to generate the optimal initial assignments. The intermediate steps (that lead to the exclusion of the static list scheduling strategies) were not checked since no strategy was found that was optimal even in the first step.

We were able to break down the problem structure yielding new approaches to algorithms. One of the difficulties of having to deal differently with steps whether there are one, two, or three machines working in parallel could be eliminated. For all (sub-)trees with less than three leaves the expected processing time can be calculated in $\mathcal{O}(n^2)$ (see chapter 8). In the calculation of the total processing time of a tree with three or more leaves the processing times of its two-leaves-subtrees can be used in various ways as a basis (see chapters 8, 9, and 10). This same approach might be extended to the k-machines version of the problem. The time to calculate the expected processing times of the $(k-1)$-leaves-subtrees should then be of the order $\mathcal{O}(n^{k-1})$.

For practical purposes the algorithms developed in this thesis can be used as heuristics. Most algorithms presented in this work seem to improve over 'naked' HLF. Figure 11.1 shows the number of trees a strategy

fails on ordered by the size of the trees. Since non-optimality of solutions in smaller trees is inherited by solutions to larger trees (e.g. in the intermediate steps) this quantitative improvement is also a qualitative improvement. The best performing algorithms are lexicographical extensions of HLF. Hence, they enjoy the property of asymptotic optimality proven by Papadimitriou and Tsitsiklis. The algorithms from chapter 9 make the heaviest use of the two-leaves-subtree problem structure. They have better results than the basic HLF strategy, but they are also quite complicated with a running time of $\mathcal{O}(n^7)$. They are on the other hand not based on something definitely non-optimal. The algorithm with the number 5 in Figure 11.1, based on a combinatorial estimation of the probabilities of reaching two-leaves-subtrees, generates correct solutions for all trees shown in Figure 7.3 on page 37 (which are the counter-examples to HLF). For a real-world problem we would select the algorithm in line 7 of Figure 11.1 with a running time of $\mathcal{O}(n^3)$ since it stays on the safe side by preselecting with HLF while successfully using the two-leaves-subtree problem structure. If time requirements are very tight, a simpler HLF-variant still seems the best choice to be made. The following considerations seem to argue against the use of a Monte Carlo technique.

Algorithm (or Question)	Failures on Trees With k Nodes								
	6	7	8	9	10	11	12	13	14
Q: Number of trees	20	48	115	286	719	1842	4766	12486	32973
Q: Trees with more than 4 leaves	5	20	67	207	595	1655	4494	n/a	n/a
1. HLF		1	8	33	116	372	1130	3352	9613
2. Best tie-breaking method for HLF (see Figure 7.4)						11	58	250	976
3. MC100 (HLF) rounded average (see Figure 7.7)		1	6	35	152	522	n/a	n/a	n/a
4. MC100 (random) rounded average (see Figure 7.7)		1	7	31	133	422	n/a	n/a	n/a
5. Two-Leaves-Tree/combinatorial-based best algorithm (see Figure 9.6, 9.)			1	8	47	206	785	n/a	n/a
6. Two-Leaves-Tree/combinatorial-based 2nd best algorithm (see Figure 9.6, 3.)			1	14	102	484	1861	n/a	n/a
7. Two-Leaves-Tree/node-weight-based best algorithm (HLF) (see Figure 10.6, Algorithm 12)					1	8	31	n/a	n/a
8. Two-Leaves-Tree/node-weight-based 2nd best algorithm (see Figure 10.6, Algorithm 11)			1	10	55	232	847	n/a	n/a

Figure 11.1: Overview of Results

The results obtained in this work also indicate that calculating an optimal solution is a difficult problem. The algorithm described in section 5.3 that calculates the solution value of a tree under a given scheduling strategy itself has exponential running time. If the size of numbers and their differences are as small as indicated in section 10.1, then their representation size is linear. Unless another way of calculating the value of a solution is found, it is unclear whether the problem belongs to NP at all. An alternating Turing machine (ATM) could be used to verify that the solution lies below a given value if it can guess a division of the solution to check at each universal node. The ATM's computation tree corresponds to the one in section 10.1. If a number is linear in size, then there are exponentially many divisions of it. Hence this construction does not even lead to a polynomial algorithm for an ATM.

Why does a third machine increase the complexity so much? Pinedo and Weiss remark in their paper [PW85] that the makespan under HLF with two machines is actually distributed independently from the in-tree constraints. This is clearly not the case for the three machines problem version. As shown in chapters

8, 9, and 10 the third machine leads to a problem structure, where the processing priority of a node is not only dependent of its children, but also depends on other nodes for which there are no precedence constraints.

A similar phenomenon occurs with the scheduling of tasks with (deterministic) unit execution time and arbitrary precedence constraints. For the problem $(P2|prec, p_i = 1|C_{max})$ a polynomial algorithm exists (first polynomial algorithm by Fujii et al. [FKN69, FKN71]). The general problem $(Pm|prec, p_i = 1|C_{max})$ with m machines is NP-hard as shown by Ullman [Ull75]. Whether the three machines problem is NP-hard or solvable in polynomial time is still an open question [GJ79]. An often observed phenomenon is that the transition from two to three in a question suffices to turn a tractable into an intractable problem (e.g. NODE-COLORING, KNF-SAT). If we look at graphs for the representation of problem structures, we can observe that in graphs with a maximal degree of two only circles and paths are possible, while any graph can be polynomial reduced one with a maximal degree of three.

The reasons why the algorithms developed in chapter 9 fail are discussed in section 9.3. That Algorithm 12 from chapter 10 fails is no surprise. We have already shown in section 7.4 that no static list scheduling policy can be optimal. This rules out any combination of static orders. All modified HLF algorithms are lexicographical orders and hence combinations of static orders. We could refine such orders infinitely many times and still not get an optimal algorithm.

A problem that arose repeatedly in searching for an optimal algorithm was the recursive dependency between the scheduling decisions. For deciding between two alternatives one needs to be able to evaluate these. This in turn requires the knowledge of the applied strategy which we are trying to identify. As an example for that, the algorithms in chapter 9 are inexact because of the estimation of the probability of reaching a two-leaves-subtree depends on how nodes are scheduled in the process. Similarly, for an optimal algorithm based on the ideas of chapter 10 it seems necessary to include information about inner nodes (e.g., one wishes to know the expected time of the scheduling of an inner node). This information influences the current scheduling decision, but the current scheduling decision influences the information needed.

What could be done next? To be able to classify the problem it seems very helpful to get a proven bound on the size of numbers and particularly on the solution difference of two schedules. This might rule out Monte Carlo techniques (see section 7.5). After all it seems that any practical method to establish the membership of the problem in a specific complexity class must tackle the problem of calculating the value of a solution first. On the other hand, the structure of the problem as uncovered in this work could be used in other heuristic techniques (e.g. genetic algorithms) in order to result in better schedules. The best performing HLF variants here can always be used as a cross check on the quality of a solution.

Other authors have extended the problem to different distributions [PW85, Fro88, PT87]. The exponential distribution allows to focus on trees and their structure. The results are independent of the parameter of the exponential distribution. Applying another distribution might either complicate the problem too much or make it easier. For the geometric distribution there exists a dependency between the distribution parameter and the effect of multiple machines working in parallel. The expected time of the first machine to finish with multiple machines decreases with a larger 'event-probability' p. This might reduce the interdependencies between tasks. Thus the problem may be easier and its examination might lead to further interesting results.

Bibliography

[Bru85] John Bruno. On Scheduling Tasks with Exponential Service Times and In-Tree Precedence Constraints. *Acta Informatica*, 22:139–148, 1985.

[Bru95] Peter Bruckner. *Scheduling Algorithms*. Springer, 1995.

[CR75] K. M. Chandy and P. F. Reynolds. Scheduling Partially Ordered Tasks with Probabilistic Execution Times. In *Proceedings of the Fifth Symposium on Operating System Principles*, pages 169–177. Operating System Reviews, 1975.

[FKN69] M. Fujii, T. Kasami, and K. Ninomiya. Optimal Sequencing of Two Equivalent Processors. *SIAM Journal of Applied Mathematics*, 17(4):784–789, 1969.

[FKN71] M. Fujii, T. Kasami, and K. Ninomiya. Erratum: Optimal Sequencing of Two Equivalent Processors. In *SIAM Journal of Applied Mathematics* [FKN69], page 141.

[Fro88] Esther Frostig. A Stochastic Scheduling Problem with Intree Precendence Constraints. *Operations Research*, 36(6):937–943, 1988.

[GJ79] Micheal R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freemann, San Francisco, 1979.

[GKP94] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison-Wesley, 2nd edition, 1994.

[Hu61] T.C. Hu. Parallel Sequencing and Assembly Line Problems. *Operations Research*, 9:841–848, 1961.

[Knu97] Donald E. Knuth. *The Art of Computer Programming*, volume 1. Addison Wesley, 3rd edition, Sep 1997.

[LLR82] E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Recent Developments in Deterministic Sequencing and Scheduling: A Survey. In M. A. H. Dempster, J.K. Lenstra, and A.H.G. Rinnooy Kan, editors, *Deterministic and Stochastic Scheduling*, pages 35–73. Reidel, 1982.

[Onl] Online Encyclopedia of Integer Sequences. http://www.research.att.com/∼njas/sequences/. AT&T Labs Research.

[Pin95] Michael Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1995.

[PT87] Christos H. Papadimitriou and John N. Tsitsiklis. On Stochastic Scheduling With In-Tree Precedence Constraints. *SIAM Journal of Computing*, 16(1):1–6, Feb 1987.

[PW85] Michael Pinedo and Gideon Weiss. Scheduling Jobs with Exponentially Distributed Processing Times and Intree Precedence Constraints on Two Parallel Machines. *Operations Research*, 33:1381–1388, 1985.

[SS01] Thomas Schickinger and Angelika Steger. *Diskrete Strukturen*, volume 2. Springer, 2001.

[Ull75] J. D. Ullman. NP-Complete Scheduling Problems. *Journal of Computer and System Sciences*, 10:384–393, 1975.

[Ull76] J. D. Ullman. Complexity of sequencing problems. In E.G. Coffman Jr., editor, *Computer and Job-Shop Scheduling Theory*, volume 10, pages 139–164. Wiley, New York, 1976.

[Urq95] Alasdair Urquhart. The Complexity of Propositional Proofs. *The Bulletin of Symbolic Logic*, 1(4):425–467, Dec 1995.

Random Sampling from Boltzmann Principles

Philippe Duchon[1], Philippe Flajolet[2], Guy Louchard[3], and Gilles Schaeffer[4]

[1] Université Bordeaux I, 351 Cours de la Libération, F-33405 Talence, France
[2] Algorithms Project, INRIA-Rocquencourt, F-78153 Le Chesnay, France
[3] Université Libre de Bruxelles, Département d'informatique, Boulevard du Triomphe, B-1050 Bruxelles, Belgium
[4] Adage Group, LORIA, F-54000 Villers-les-Nancy, France

Abstract. This extended abstract proposes a surprisingly simple framework for the random generation of combinatorial configurations based on *Boltzmann models*. Random generation of possibly complex structured objects is performed by placing an appropriate measure spread over the whole of a combinatorial class. The resulting algorithms can be implemented easily within a computer algebra system, be analysed mathematically with great precision, and, when suitably tuned, tend to be efficient in practice, as they often operate in linear time.

1 Introduction

In this text, *Boltzmann models* are proposed as a framework for the random generation of structured combinatorial configurations, like words, trees, permutations, constrained graphs, and so on. A Boltzmann model relative to a combinatorial class $\mathcal{C}$ depends on a control parameter $x > 0$ and places an appropriate measure that is spread over the whole of $\mathcal{C}$. Random objets under a Boltzmann model then have a fluctuating size, but objects with the same size invariably occur with the same probability. In particular, a *Boltzmann sampler* (i.e., a random generator that obeys a Boltzmann model), with the size of its output conditioned to be a fixed value n, draws *uniformly* at random an object of size n.

As we demonstrate in this article, Boltzmann samplers can be derived systematically (and simply) for classes that are specified in terms of a basic collection of general-purpose combinatorial constructions. These constructions are precisely the ones that surface recurrently in modern theories of combinatorial analysis; see, e.g., [2, 7, 9] and references therein. As a consequence, one obtains with surprising ease Boltzmann samplers covering an extremely wide range of combinatorial types.

Fixed-size generation is the standard paradigm in the random generation of combinatorial structures, and a vast literature exists on the subject. There, either specific bijections are exploited or general combinatorial decompositions are put to use in order to generate objects at random based on possibility counts—this has come to be known as the "recursive method" originating with Nijenhuis and Wilf [12] and formalized by Flajolet, Zimmermann, and Van Cutsem in [8]. In contrast, the basic principle of Boltzmann sampling is to *relax* the constraint

of generating objects of a strictly fixed size, and prefer to draw objects with a randomly varying size. As we shall see, normally, one can *tune* the value of the control parameter x in order to favour objects of a size in the vicinity of a target value n. If needed, one can pile up a filter that rejects objects whose size is out of range. In this way, Boltzmann samplers may also serve for approximate-size as well as exact-size random generation.

We propose Boltzmann samplers as an attractive alternative to standard combinatorial generators based on the recursive method and implemented in packages like Combstruct (under the computer algebra system MAPLE) and CS (under MUPAD). Boltzmann algorithms are expected to be competitive when compared to many existing combinatorial methods: they only necessitate a small *fixed* number of multiprecision constants that are fairly easy to compute; when suitably optimized, they operate in low polynomial time—often even in linear time. Accordingly, uniform generation of objects with sizes in the range of millions is becoming a possibility, whenever the approach is applicable.

2 Boltzmann models and generators

We consider a class $\mathcal{C}$ of combinatorial objects of sorts, with $|\cdot|$ the size function from $\mathcal{C}$ to $\mathbb{Z}_{\geq 0}$. By $\mathcal{C}_n$ is meant the subclass of $\mathcal{C}$ comprising all the objects in $\mathcal{C}$ having size n. Each $\mathcal{C}_n$ is assumed to be finite. One may think of binary words (with size defined as length), permutations, graphs and trees of various types (with size defined as number of vertices), and so on. Any set $\mathcal{C}$ endowed with a size function and satisfying the finiteness axiom will henceforth be called a *combinatorial class*.

Definition 1. *The* Boltzmann models *of parameter x exist in two varieties, the ordinary version and the exponential version. They assign to any object $\gamma \in \mathcal{C}$ the following probability:*

$$\textit{Ordinary/Unlabelled case: } \mathbb{P}_x(\gamma) = \frac{1}{C(x)} \cdot x^{|\gamma|} \textit{ with } C(x) = \sum_{\gamma \in \mathcal{C}} x^{|\gamma|},$$

$$\textit{Exponential/Labelled case: } \mathbb{P}_x(\gamma) = \frac{1}{\widehat{C}(x)} \cdot \frac{x^{|\gamma|}}{|\gamma|!} \textit{ with } \widehat{C}(x) = \sum_{\gamma \in \mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!}.$$

A Boltzmann generator (or sampler) *$\Gamma C(x)$ for a class $\mathcal{C}$ is a process that produces objects from $\mathcal{C}$ according to a Boltzmann model.*

The normalization coefficients are nothing but the counting generating functions of ordinary type (OGF) for $C(x) := \sum_n C_n x^n$ and exponential type (EGF) for $\widehat{C}(x) := \sum_n C_n x^n/n!$. Only *coherent* values of x defined to be such that $0 < x < \rho_C$ (or $\rho_{\widehat{C}}$), with ρ_f the radius of convergence of f are to be considered.

The name "Boltzmann model" comes from the great statistical physicist Boltzmann whose works (together with those of Gibbs) led to enounce the following principle: *Statistical mechanical configurations of energy equal to E in a system have a probability of*

occurrence proportional to $e^{-\beta E}$. (There, β is an inverse temperature.) If one identifies size of a combinatorial configuration with energy of a thermodynamical system and sets $x = e^{-\beta}$, then what we term the ordinary Boltzmann models become the true model of statistical mechanics. The counting generating function in the combinatorial world then coincides with the normalization constant in the statistical mechanics world where it is known as the *partition function* and is often denoted by Z. Under this perhaps artificial dictionary, Boltzmann models and random combinatorics become united.

For reasons which will become apparent, we have also defined the exponential Boltzmann model. These are appropriate for handling *labelled* combinatorial structures while the ordinary models are to be used for *unlabelled* combinatorial models. In the unlabelled universe, all elementary components of objects ("atoms") are indistinguishable, while in the labelled universe, they are all distinguished from one another by bearing a distinctive mark, say one of the integers between 1 and n if the object considered has size n. (This terminology is standard in combinatorial enumeration and graph theory [2, 7, 9].)

The size of the resulting object under a Boltzmann model is a random variable, denoted throughout by N, whose law is quantified by the following lemma.

Proposition 1. *The random size of the object produced under the ordinary Boltzmann model of parameter x satisfies*

$$\mathbb{E}_x(N) = x\frac{C'(x)}{C(x)}, \qquad \mathbb{E}_x(N^2) = \frac{x^2C''(x) + xC'(x)}{C(x)}. \tag{1}$$

Proof. By construction the probability of drawing an object of size n is $\mathbb{P}_x(N = n) = C_n x^n/C(x)$. Consequently, the probability generating function of N is $C(xz)/C(x)$ and the result follows.

In the next two sections (Sections 3 and 4), we develop a collection of rules by which one can assemble Boltzmann generators from simpler ones. The combinatorial classes considered are built on a small set of constructions that have a wide expressive power. The language in which classes are specified is in essence the same as the one underlying the recursive method [6, 8]: it consists of the constructions of union, product, sequence, set, and cycle. For each allowable class, a Boltzmann sampler can be built in an entirely systematic manner.

3 Ordinary Boltzmann Generators

A *combinatorial construction* builds a new class $\mathcal{C}$ from structurally simpler classes $\mathcal{A}, \mathcal{B}$, in such a way that $\mathcal{C}_n$ is determined from objects in $\{\mathcal{A}_j\}_{j=0}^n, \{\mathcal{B}_j\}_{j=0}^n$. Constructions considered here are disjoint *union* ($+$), cartesian *product* ($\times$), and *sequence* formation ($\mathfrak{S}$). We define these in turn and concurrently build the corresponding Boltzmann sampler ΓC for the composite class $\mathcal{C}$, given random generators $\Gamma A, \Gamma B$ for the ingredients and assuming the values of intervening generating functions $A(x), B(x)$ at x to be known exactly.

Disjoint union. Write $\mathcal{C} = \mathcal{A} + \mathcal{B}$ if $\mathcal{C}$ is the union of disjoint copies of $\mathcal{A}$ and $\mathcal{B}$, while size on $\mathcal{C}$ is inherited from $\mathcal{A}, \mathcal{B}$. One has $C(x) = A(x) + B(x)$. The Boltzmann model corresponding to $C(x)$ is then a mixture of the models associated to $A(x)$ and $B(x)$, with the probability of selecting a particular γ in $\mathcal{C}$ being $\mathbb{P}(\gamma \in \mathcal{A}) = A(x)/C(x)$, $\mathbb{P}(\gamma \in \mathcal{B}) = A(x)/C(x)$. Let us be given a generator for a Bernoulli variable $\mathrm{Bern}(p)$ defined as follows: $\mathrm{Bern}(p) = 1$ with probability p; $\mathrm{Bern}(p) = 0$ with probability $1 - p$; a sampler ΓC given ΓA and ΓB is simply obtained by

function $\Gamma C(x : \mathsf{real})$; let $p_A := A(x)/(A(x) + B(x))$;
if $\mathrm{Bern}(p_A)$ then return$(\Gamma A(x))$ else return$(\Gamma B(x))$ fi.

Cartesian Product. Write $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ if $\mathcal{C}$ is the set of ordered pairs from $\mathcal{A}$ and $\mathcal{B}$, and size on $\mathcal{C}$ is inherited additively from $\mathcal{A}, \mathcal{B}$. For generating functions, one finds $C(x) = A(x) \cdot B(x)$. A random element of $C(x)$ is then obtained by forming a pair $\langle \alpha, \beta \rangle$ with α, β drawn *independently* from the Boltzmann models $A(x), B(x)$, respectively:

function $\Gamma C(x : \mathsf{real})$; return$(\langle \Gamma A(x), \Gamma B(x) \rangle)$ {independent calls}.

Sequences. Write $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ if $\mathcal{C}$ is composed of all the finite sequences of elements of $\mathcal{A}$. The sequence class $\mathcal{C}$ is also the solution to the symbolic equation $\mathcal{C} = \mathbf{1} + \mathcal{A}\mathcal{C}$ (with $\mathbf{1}$ the empty sequence), which only involves unions and products. Consequently, one has $C(x) = (1 - A(x))^{-1}$. This gives rise to a recursive generator for sequences. Once recursion is unwound, the resulting generator assumes a particularly simple form:

function $\Gamma C(x : \mathsf{real})$; let $A(x)$ be the value of the OGF of $\mathcal{A}$;
draw K according to Geometric$(A(x))$;
return the K-tuple $\langle \Gamma A(x), \Gamma A(x), ..., \Gamma A(x) \rangle$ {independent calls}.

Finite sets. There finally remains to discuss initialization (when and how do we stop?). Clearly if $\mathcal{C}$ is finite (and in practice small), one can generate a random element of $\mathcal{C}$ by selecting it according to the finite probability distribution given explicitly by the definition of the Boltzmann model.

Proposition 2. *Define as* specifiable *an unlabelled class that can be specified (in a possibly recursive way) from finite sets by means of disjoint unions, cartesian products, and the sequence construction. Let C be an unlabelled specifiable class. Let x be a "coherent" parameter in $(0, \rho_C)$, and let ΓC be the generator compiled from the definition of $\mathcal{C}$ by means of the three rules above. Then ΓC correctly draws elements from $\mathcal{C}$ according to the ordinary Boltzmann model. It halts with probability 1 and in finite expected time.*

Example 1. Words without long runs. Consider the collection $\mathcal{R}$ of binary words over the alphabet $\mathcal{A} = \{a, b\}$ such that they never have more than m consecutive occurrences of any letter. The set $\mathcal{W}$ of all words is expressible by a regular expression written in our notation $\mathcal{W} = \mathfrak{S}(b) \times \mathfrak{S}\left(a\mathfrak{S}(a)b\mathfrak{S}(b)\right) \times \mathfrak{S}(a)$. This

expresses the fact that any word has a "core" formed with blocks of a's and blocks of b's in alternation that is bordered by a header of b's and a trailer of a's. The decomposition serves for $\mathcal{R}$: e.g., replace any internal $a\mathfrak{S}(a)$ by $\mathfrak{S}_{1\,..\,m}(a)$ and any $b\mathfrak{S}(b)$ by $\mathfrak{S}_{1\,..\,m}(b)$, where $\mathfrak{S}_{1\,..\,m}$ means a sequence of between 1 and m elements. The composition rules given above give rise to a generator for $\mathcal{R}$ of the following form: two generators produce sequences of a's or b's according to a truncated geometric law; a generator for the product $\mathcal{C} := (\mathfrak{S}_{1\,..\,m}(a)\mathfrak{S}_{1\,..\,m}(b))$ is built according to the product rule; a generator for the "core" sequence $\mathcal{D} := \mathfrak{S}(\mathcal{C})$ is constructed according to the sequence rule. The generator assembled *automatically* from the general rules is then

$$\underset{\leq m}{\mathrm{Geom}}\,(x;b)\left\{\,\mathrm{Geom}\,\left[\tfrac{x^2(1-x^m)^2}{(1-x)^2}\right]\circ\left\langle\underset{1\,..\,m}{\mathrm{Geom}}\,(x;a),\ \underset{1\,..\,m}{\mathrm{Geom}}\,(x;b)\right\rangle\right\}\underset{\leq m}{\mathrm{Geom}}\,(x;a).$$

Example 2. Trees (rooted, plane). Take first the class $\mathcal{B}$ of binary trees defined by the recursive specification $\mathcal{B} = \mathcal{Z} + (\mathcal{Z} \times \mathcal{B} \times \mathcal{B})$, where $\mathcal{Z}$ is the class comprising the generic node. The generator ΓZ is deterministic and consists simply of the instruction "output a node" (since $\mathcal{Z}$ is finite and in fact has only one element). The Boltzmann generator ΓB calls ΓZ (and halts) with probability $x/B(x)$ where $B(x)$ is the OGF of binary trees, $B(x) = (1 - \sqrt{1-4x^2})/(2x)$. With the complementary probability corresponding to the strict binary case, it will make a call to ΓZ and two recursive calls to itself. In other words: *the Boltzmann generator for binary trees as constructed automatically from the composition rules produces a random sample of the (subcritical) branching process with probabilities* $x/B(x)$, $xB(x)^2/B(x)$. Unbalanced 2-3 trees are similarly produced from $\mathcal{U} = \mathcal{Z} + \mathcal{U}^2 + \mathcal{U}^3$, unary-binary trees from $\mathcal{V} = \mathcal{Z}(\mathbf{1} + \mathcal{V} + \mathcal{V}^2)$, etc.

Example 3. Secondary structures. This example is inspired by the works of Waterman *et al.*, themselves motivated by the problem of enumerating secondary RNA structures. To fix ideas, consider rooted binary trees where edges contain 2 or 3 atoms and leaves ("loops") contain 4 or 5 atoms. A specification is $\mathcal{S} = (\mathcal{Z}^4 + \mathcal{Z}^5) + (\mathcal{Z}^2 + \mathcal{Z}^3)^2 \times (\mathcal{S} \times \mathcal{S})$. A Bernoulli switch will decide whether to halt or not, two independent recursive calls being made in case it is decided to continue, with the algorithm being sugared with suitable Bernoulli draws. The method is clearly universal for this entire class of problems.

4 Exponential Boltzmann Generators

We consider here *labelled structures* in the precise technical sense of combinatorial theory; see, e.g., [7]. A labelled object of size n is then composed of n distinguishable atom, each bearing a distinctive label that is an integer in the interval $[1, n]$. Labelled combinatorial classes can be subjected to the *labelled product* defined as follows: if $\mathcal{A}$ and $\mathcal{B}$ are labelled classes, the product $\mathcal{C} = \mathcal{A} \star \mathcal{B}$ is obtained by forming all ordered pairs $\langle \alpha, \beta \rangle$ with $\alpha \in \mathcal{A}$ and $\beta \in \mathcal{B}$ and

relabelling them in all possible order-consistent ways. From the definition, a binomial convolution $C_n = \sum_{k=0}^{n} \binom{n}{k} A_k B_{n-k}$, takes care of relabellings. In terms of exponential generating functions, this becomes $\widehat{C}(z) = \widehat{A}(z) \cdot \widehat{B}(z)$.

Like in the ordinary case, we proceed by assembling Boltzmann generators for structured objects from simpler ones.

Disjoint union. The unlabelled construction carries over verbatim.

Labelled product. The cartesian product construction adapts to this case: in order to produce an element from $\mathcal{C} = \mathcal{A} \star \mathcal{B}$, simply produce an independent pair by the cartesian product rule, but using the EGF values $\widehat{A}(x), \widehat{B}(x)$.

Sequences. In the labelled universe, $\mathcal{C}$ is the sequence class of $\mathcal{A}$, written $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ iff it is composed of all the sequences of elements from A up to order-consistent relabellings. Then, the EGF relation $\widehat{C}(x) = (1 - \widehat{A}(x))^{-1}$ holds, and the sequence construction of the generator ΓC from ΓA given in Section 3 and based on the geometric law is applicable.

Sets. This is a new construction that we did not consider in the unlabelled case. The class $\mathcal{C}$ is the set-class of $\mathcal{A}$, written $\mathcal{C} = \mathfrak{P}(\mathcal{A})$ ($\mathfrak{P}$ is reminiscent of "powerset") if $\mathcal{C}$ is the quotient of $\mathfrak{S}\{\mathcal{A}\}$ by the relation that declares two sequences as equivalent if one derives from the other by an arbitrary permutation of the components. It is then easily seen that the EGFs are related by $\widehat{C}(x) = \sum_{k \geq 0} \widehat{A}(x)^k / k! = e^{\widehat{A}(x)}$, where the factor $1/k!$ "kills" the order present in sequences. A moment of reflection shows that, under the exponential Boltzmann model, the probability for a set in $\mathcal{C}$ to have k components is $e^{-\widehat{A}(x)} \widehat{A}(x)^k / k!$, that is, a Poisson law of rate $\widehat{A}(x)$. This gives rise to a simple algorithm for generating sets (analogous to the geometric algorithm for sequences):

function $\Gamma C(x : \mathsf{real})$; let $\widehat{A}(x)$ be the value of the EGF of $\mathcal{A}$;
draw K according to Poisson($\widehat{A}(x)$);
return the K-tuple $\langle \Gamma A(x), \Gamma A(x), ..., \Gamma A(x) \rangle$ {independent calls}.

Cycles. This construction, written $\mathcal{C} = \mathfrak{C}(\mathcal{A})$, is defined like sets but with two sequences being identified if one is a cyclic shift of the other. The EGFs satisfy $\widehat{C}(x) = \sum_{k \geq 0} \widehat{A}(x)^k / k = \log(1 - \widehat{A}(x))^{-1}$. The log-law (also known as "logarithmic series distribution") of rate $\lambda < 1$, is defined by $\mathbb{P}(X = k) = (-\log(1 - \lambda))^{-1} \lambda^k / k$. Then cycles under the exponential Boltzmann model can be drawn like in the case of sets upon replacing the Poisson law by the log-law.

Proposition 3. *Define as* specifiable *a labelled class that can be specified (in a possibly recursive way) from finite sets by means of disjoint unions, cartesian products, as well as sequence, set and cycle constructions. Let $\mathcal{C}$ be a labelled specifiable class. Let x be a "coherent" parameter in $(0, \rho_{\widehat{C}})$, and let ΓC be the generator compiled from the definition of $\mathcal{C}$ by means of the five rules above. Then ΓC correctly draws elements from $\mathcal{C}$ according to the exponential Boltzmann model. It halts with probability 1 and in finite expected time.*

Example 4. Set partitions. A set partition of size n is a partition of the integer interval $[1, n]$ into a certain number of nonempty classes, also called blocks,

the blocks being by definition unordered between themselves. Let $\mathfrak{P}_{\geq 1}$ represent the powerset construction where the number of components is constrained to be ≥ 1. The labelled class of all set partitions is then definable as $\mathcal{S} = \mathfrak{P}(\mathfrak{P}_{\geq 1}(\mathcal{Z}))$, where $\mathcal{Z}$ consists of a single labelled atom, $\mathcal{Z} = \{1\}$. The EGF of $\mathcal{S}$ is the well-known generating function of the Bell numbers, $\widehat{S}(x) = e^{e^x - 1}$. By the composition rules, a random generator is as follows: *Choose the number K of blocks as Poisson$(e^x - 1)$. Draw K independent copies $X_1, X_2, \ldots, X_K$ from the Poisson law of rate x, each conditioned to be at least 1.*

Example 5. Random surjections (or ordered set partitions). These may be defined as functions from $[1, n]$ to $[1, n]$ such that the image of f is an initial segment of $[1, n]$ (i.e., there are no "gaps"). One has for the class $\mathcal{Q}$ of surjections $\mathcal{Q} = \mathfrak{S}(\mathfrak{P}_{\geq 1}(\mathcal{Z}))$. Thus a random generator for $\mathcal{Q}$ first chooses a number of components $K \in \mathrm{Geom}\,(e^x - 1)$ and then launches K Poisson generators.

Example 6. Cycles in permutations. This corresponds to $\mathcal{P} = \mathfrak{P}(\mathfrak{C}_{\geq 1}(\mathcal{Z}))$ and is obtained by a (Poisson$\circ$Log) process. (This example is loosely related to the Shepp–Lloyd model that generates permutations by ordered cycle lengths.)

Example 7. Assemblies of filaments in a liquid. We may model these as sets of sequences, $\mathcal{F} = \mathfrak{P}(\mathfrak{S}_{\geq 1}(\mathcal{Z}))$. The EGF is $\exp(z/(1-z))$. The random generation algorithm is a compound of the form (Poisson$\circ$Geometric), with appropriate parameters. (See A000262 in Sloane's encyclopedia [14].)

5 The realization of Boltzmann samplers

In this section, we examine the way Boltzmann sampling can be implemented and sketch a discussion of complexity issues involved. In this abstract, only the *real-arithmetic model* ($\mathbb{R}$) is considered. There, what is assumed to be given is a random-access machine with unit cost for (exact) real arithmetic operations and elementary transcendental functions over the real numbers.

By definition, a Boltzmann sampler requires as input the value of the control parameter x that defines the Boltzmann model of use. As seen in previous sections, it also needs the finite collection of values at x of the generating functions that intervene in a specification. We assume these values to be provided by what we call the (generating function) "*oracle*". Such constants, which need only be precomputed *once*, are likely to be provided by a multiprecision package or a computer algebra system used as coroutine.

First one has to specify fully generators for the probabilistic laws $\mathrm{Geom}\,(\lambda)$, $\mathrm{Pois}\,(\lambda)$, $\mathrm{Loga}\,(\lambda)$, as well as the Bernoulli generator $\mathrm{Bern}\,(p)$, where the latter outputs 1 with probability p and 0 otherwise. A random generator 'uniform ()' produces at unit cost a random variable uniformly distributed over the real interval $(0, 1)$.

Bernoulli generator. The Bernoulli generator is simply

$$\mathrm{Bern}\,(p) := \mathsf{if}\ \mathrm{uniform}\,() \leq p\ \mathsf{then}\ \mathrm{return}(1)\ \mathsf{else}\ \mathrm{return}(0)\ \mathsf{fi}.$$

This generator serves in particular to draw from unions of classes.

Geometric, Poisson, and Logarithmic generators. For the remaining laws, we let p_k be the probability that a random variable with the desired distribution has value k, namely,

$$\mathrm{Geom}\,(\lambda):\ (1-\lambda)\lambda^k;\quad \mathrm{Pois}\,(\lambda):\ e^{-\lambda}\frac{\lambda^k}{k!};\quad \mathrm{Loga}\,(\lambda):\ \frac{1}{\log(1-\lambda)^{-1}}\frac{\lambda^k}{k}.$$

The general scheme that goes well with real-arithmetic models is the *sequential algorithm*:

$U :=$ uniform $();\ S := 0;\ k := 0;$
while $U < S$ do $S := S + p_k;\ k := k + 1;$ od; return(k).

This scheme is nothing but a straightforward implementation based on inversion of distribution functions (see [4, Sec. 2.1]). For the three distributions under consideration, the probabilities p_k can themselves be computed recurrently on the fly. In particular, under the model that has unit cost for real arithmetic operations and functions, the sequential generators have a useful property: *a variable with outcome k is drawn with a number of operations that is $O(k+1)$.* This has immediate consequences for all classes that are specifiable in the sense of Propositions 2 and 3.

Theorem 1. *Consider a specifiable class $\mathcal{C}$, either labelled or unlabelled. Assume as given an oracle that provides the finite collection of* exact *values of the intervening generating functions at a coherent value x. Then, the Boltzmann generator $\Gamma C(x)$ has a complexity in the number of real-arithmetic operations that is linear in the size of its output object.*

The linear complexity in the abstract model $\mathbb{R}$, as expressed in Theorem 1, provides an indication of the broad type of complexity behaviour one may aim for in practice, namely linear-time complexity. For instance, one may realize a Boltzmann sampler by truncating real numbers to some fixed precision, say using floating point numbers represented on 64 bits or 128 bits. The resulting samplers operate in time linear in the size of the output, though they may fail (by lack of digits in values of generating functions) in a small number of cases, and accordingly must deviate (slightly) from uniformity. Pragmatically, such samplers are likely to suffice for most medium-size simulations.

A sensitivity analysis of truncated Boltzmann samplers would be feasible, though rather heavy to carry out. One could even correct perfectly the lack of uniformity by appealing to an adaptive precision strategy based on guaranteed multiprecision floating point arithmetic. (The reader may get a feeling of the type of analysis involved by referring to the papers by Denise, Zimmermann, and Dutour, e.g., [3], where a thorough examination of the recursive method under this angle has been conducted.) In the full paper [5], we shall discuss bit-level implementations of Boltzmann samplers (see Knuth and Yao's insightful work [10] for context), as well as implementation issues raised by the oracle.

6 Exact-size and approximate-size sampling

Our primary objective in this article is the fast random generation of objects of some large size. Two types of constraints on size are considered. In *exact-size* random sampling, objects of $\mathcal{C}$ should be drawn uniformly at random from the subclass $\mathcal{C}_n$ of objects of size *exactly* n. In *approximate-size* random sampling, objects should be drawn with a size in an interval of the form $[n(1-\varepsilon), n(1+\varepsilon)]$, for some quantity $\varepsilon \geq 0$ called the (relative) *tolerance*, with two objects of the same size still being equally likely to occur. The conditions of exact and approximate-size sampling are immediately satisfied if one filters the output a Boltzmann generator by *rejecting* the elements that do not obey the desired size constraint. The main question is when and how can this rejection process be made reasonably efficient. The major conclusion from this and the next section is as follows: in many cases, including all the examples seen so far, *approximate-size sampling is achievable in linear time under the real-arithmetic model* of Theorem 1. The constants appear to be not too large if a "reasonable" tolerance on size is allowed.

The outcome of a basic Boltzmann sampler has a random size N whose distribution is exactly described by Proposition 1. First, for the rejection sampler tuned at the "natural" value $x = x_n$ such that $\mathbb{E}_{x_n}(N) = n$, a direct application of Chebyshev's inequalities gives:

Theorem 2. *Let $\mathcal{C}$ be a specifiable class and ε a* fixed nonzero *tolerance on size. Assume the following Mean Value and Variance Conditions,*

$$\lim_{x \to \rho^-} \mathbb{E}_x(N) = +\infty, \qquad \lim_{x \to \rho^-} \frac{\sqrt{\mathbb{E}_x(N^2) - \mathbb{E}_x(N)^2}}{\mathbb{E}_x(N)} = 0. \tag{2}$$

Then, the rejection sampler equipped with the value $x = x_n$ defined by the inversion relation $x_n C'(x_n)/C(x_n) = n$ succeeds in one trial with probability tending to 1 as $n \to \infty$. Its total cost is $O(n)$ on average.

The mean and variance conditions *are* satisfied by the class $\mathcal{S}$ of set partitions (Example 4) and the class $\mathcal{F}$ of assemblies of filaments (Example 7).

It is possible to discuss at a fair level of generality cases where rejection sampling is efficient, even though the strong moment conditions of Theorem 2 may not hold. The discussion is fundamentally based on the types of singularities that the generating functions exhibit. This is an otherwise well-researched topic as it is central to asymptotic enumeration [7, 13].

Theorem 3. *Let $\mathcal{C}$ be a combinatorial class that is specifiable. Assume that the generating function $C(z)$ (for $z \in \mathbb{C}$) has an isolated singularity at ρ, which is the unique dominant singularity. Assume also that the singular expansion of $C(z)$ at ρ is of the form (with P a polynomial)*

$$C(z) \underset{z \to \rho}{\sim} P(z) + c_0(1 - z/\rho)^{-\alpha} + o((1 - z/\rho)^{-\alpha}). \tag{3}$$

When the exponent $-\alpha$ is negative, for any fixed nonzero *tolerance ε, the rejection sampler corresponding to $x = x_n$ succeeds in an expected number of trials asymptotic to the constant*

$$\frac{1}{\xi_\alpha(\varepsilon)}, \quad \textit{where} \quad \xi_\alpha(\varepsilon) = \frac{\alpha^\alpha}{\Gamma(\alpha)} \int_{-\varepsilon}^{\varepsilon} (1+s)^{\alpha-1} e^{\alpha(1+s)}\, ds.$$

Moreover the total cost of this rejection sampler is $O(n)$ on average.

Words without long runs, surjections, and permutations (Examples 1, 5, and 6) have generating functions with a polar singularity, corresponding to the singular exponent -1, and hence satisfy the conditions above.

We note here that a condition $-\alpha < 0$ can often be ensured by successive differentiations of generating functions. Combinatorially, this corresponds to a *"pointing"* construction. Boltzmann sampling combined with pointing and rejection is developed in the full article [5] as a viable optimization technique.

7 Singular Boltzmann samplers.

We now discuss two infinite categories of models, where it is of advantage to place oneself right at the singularity $x = \rho_C$ in order to develop a rejection sampler from a Boltzmann model for $\mathcal{C}$. One category covers several of the sequence constructions, the other one corresponds to a wide set of recursive specifications.

Singular samplers for sequences. Define a sequence construction $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ to be supercritical if $\rho_A > \rho_C$. The generating function of $\mathcal{C}$ and $\mathcal{A}$ satisfy $C(x) = (1 - A(x))^{-1}$, so that the supercriticality condition corresponds to $A(\rho_C) = 1$, with the (dominant) singularity ρ_C of $C(x)$ being necessarily a pole.

Theorem 4. *Consider a sequence construction $\mathcal{C} = \mathfrak{S}(\mathcal{A})$ that is supercritical. Generate objects from $\mathcal{A}$ sequentially according to $\Gamma A(\rho_C)$ until the total size becomes at least n. With probability tending to 1 as $n \to \infty$, this produces a random $\mathcal{C}$ object of size $n + O(1)$ in* one *trial. Exact-size random generation is achievable from this generator by rejection in expected time $O(n)$.*

This theorem applies to "cores" of words without long runs (from Example 1) and surjections (Example 5), for which exact-size generation become possible in linear time. It also provides a global setting for a variety of *ad hoc* algorithms developed by Louchard in the context of efficient generation of certain types (directed, convex) of random planar diagrams known as "animals" and "polyominos".

Example 8. Coin fountains ($\mathcal{O}$). These were enumerated by Odlyzko and Wilf. They correspond to Dyck paths taken according to area (disregarding length). The OGF is the continued fraction $O(z) = 1 \,/(1 - z \,/(1 - z^2 \,/(1 - z^3 \,/(\cdots))))$. At top level, the singular Boltzmann sampler of Theorem 4 applies (write $\mathcal{O} = \mathfrak{S}(\mathcal{Q})$ and $O(z) = (1 - Q(z))^{-1}$). The root ρ of $Q(z) = 1$ is easily found to high precision as $\rho = 0.5761487691\cdots$. The objects of $\mathcal{Q}$ needed are with high probability of

size at most $O(\log n)$, so that they can be generated by whichever subexponential method is convenient. The overall (theoretical and practical) complexity is $O(n)$ with *very* low implementation constants. Random generation well in the range of millions is now easy thanks to the singular Boltzmann generator.

Singular samplers for recursive structures. What we call a recursive class $\mathcal{C}$ is the component $\mathcal{C} = \mathcal{F}_1$ of a system of mutually dependent equations:

$$\{\mathcal{F}_1 = \Psi_1\left(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m\right), \ldots, \mathcal{F}_m = \Psi_m\left(\mathcal{Z}; \mathcal{F}_1, \ldots, \mathcal{F}_m\right)\}$$

where the Ψ's are *any* functional term involving *any* constructor defined previously ('+', '×' or '⋆', and $\mathfrak{S}, \mathfrak{P}, \mathfrak{C}$) The system is said to be irreducible if the dependency graph between the $\mathcal{F}_j$ is strongly connected (everybody depends on everybody else). In such a case, the singular type of the generating functions is a square-root, as follows from a mild generalization of a famous theorem by Drmota, Lalley, and Woods; see [7, Ch. 8] and references therein. A consequence is that coefficients of generating functions are of the universal form $\rho^{-n} n^{-3/2}$. In particular objects of a small size are likely to be produced by the singular generator $\Gamma C(\rho_C)$ whereas the expectation of size $\mathbb{E}_{\rho_C}(N)$ is infinite. (In other words, a very high dispersion of sizes is observed.) The singular sampler considered here simply uses the singular value $\rho = \rho_C$ together with an "early-abort" strategy: it aborts its execution as soon as the size of the partially generated object exceeds the tolerance upper bound. The process is repeated till an object within the tolerance bounds is obtained.

Theorem 5. *Let $\mathcal{C}$ be a combinatorial class given by a recursive specification that is irreducible and aperiodic. For any* fixed nonzero *tolerance ε, the "early-abort" rejection sampler succeeds in a number of trials that is $O(n^{1/2})$ on average. Furthermore, the total cost K_n of this sampler satisfies*

$$\mathbb{E}(K_n) \sim \frac{n}{\varepsilon}\left((1-\varepsilon)^{1/2} + (1+\varepsilon)^{1/2}\right). \tag{4}$$

For exact-size generation, the "early-abort" rejection sampler has complexity $O(n^2)$.

The early-abort sampler thus gives linear-time approximate-size random generation for all the simple varieties of trees of Example 2 (including binary trees, unary-binary trees, 2–3 trees, and so on) and for secondary structures (Example 3). For all these cases, exact-size is also achievable in quadratic time. The method is roughly comparable to drawing from a suitably dimensioned critical branching process in combination with abortion and rejection.

The rejection algorithm above is akin to the "Florentine algorithm" invented by Barcucci–Pinzani–Sprugnoli [1] to generate prefixes of Motzkin words and certain directed plane animals. The cost analysis is related to Louchard's work [11].

8 Conclusions

As shown here, combinatorial decompositions allow for random generation in low polynomial time. In particular, approximate-size random generation is often

of a linear time complexity. Given the large number of combinatorial decompositions that have been gathered over the past two decades (see, e.g., [2, 7, 9], we estimate to perhaps a hundred the number of classical combinatorial structures that are amenable to efficient Boltzmann sampling. In contrast with the recursive method [3, 8, 12], memory requirements are kept to a minimum since only a table of constants of size $O(1)$ is required.

In forthcoming works starting with [5], we propose to demonstrate the versatility of Boltzmann sampling including: the generation of unlabelled multisets and powersets, the encapsulation of constructions like substitution and pointing, and the realization of Boltzmann samplers at bit-level. (Linear boolean complexity seems to be achievable in many cases of practical interest.)

Acknowledgements: The authors are grateful to Brigitte Vallée for several architectural comments on an early version of this manuscript. Thanks also to Bernard Ycart, Jim Fill, Marni Mishna, and Paul Zimmermann for encouragements and constructive observations. This work was supported in part by the ALCOM-FT Project IST-1999-14186 of the European Union.

References

1. Barcucci, E., Pinzani, R., and Sprugnoli, R. The random generation of directed animals. *Theoretical Computer Science 127*, 2 (1994), 333–350.
2. Bergeron, F., Labelle, G., and Leroux, P. *Combinatorial species and tree-like structures.* Cambridge University Press, Cambridge, 1998. Translated from the 1994 French original by Margaret Readdy, With a foreword by Gian-Carlo Rota.
3. Denise, A., and Zimmermann, P. Uniform random generation of decomposable structures using floating-point arithmetic. *Theoretical Computer Science 218*, 2 (1999), 233–248.
4. Devroye, L. *Non-Uniform Random Variate Generation.* Springer Verlag, 1986.
5. Duchon, P., Flajolet, P., Louchard, G., and Schaeffer, G. Boltzmann samplers for random combinatorial generation. In preparation, 2002.
6. Flajolet, P., Salvy, B., and Zimmermann, P. Automatic average–case analysis of algorithms. *Theoretical Computer Science 79*, 1 (Feb. 1991), 37–109.
7. Flajolet, P., and Sedgewick, R. *Analytic Combinatorics.* 2001. Book in preparation: Individual chapters are available as INRIA Research Reports 1888, 2026, 2376, 2956, 3162, 4103 and electronically under http://algo.inria.fr/flajolet/Publications/books.html.
8. Flajolet, P., Zimmermann, P., and Van Cutsem, B. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science 132*, 1-2 (1994), 1–35.
9. Goulden, I. P., and Jackson, D. M. *Combinatorial Enumeration.* John Wiley, New York, 1983.
10. Knuth, D. E., and Yao, A. C. The complexity of nonuniform random number generation. In *Algorithms and complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1976).* Academic Press, New York, 1976, pp. 357–428.
11. Louchard, G. Asymptotic properties of some underdiagonal walks generation algorithms. *Theoretical Computer Science 218*, 2 (1999), 249–262.
12. Nijenhuis, A., and Wilf, H. S. *Combinatorial Algorithms*, second ed. Academic Press, 1978.
13. Odlyzko, A. M. Asymptotic enumeration methods. In *Handbook of Combinatorics*, R. Graham, M. Grötschel, and L. Lovász, Eds., vol. II. Elsevier, Amsterdam, 1995, pp. 1063–1229.
14. Sloane, N. J. A. *The On-Line Encyclopedia of Integer Sequences.* 2000. Published electronically at `http://www.research.att.com/~njas/sequences/`.

GRAPHS, DETERMINANTS OF KNOTS AND HYPERBOLIC VOLUME

This is a preprint. I would be grateful for any comments and corrections!

A. Stoimenow*

Department of Mathematics,
University of Toronto,
Canada M5S 3G3
e-mail: stoimeno@math.toronto.edu
WWW: http://www.math.toronto.edu/stoimeno/

Current version: January 21, 2003 First version: January 20, 2000

Abstract. The Kauffman bracket approach is used to give estimates on the size of the determinant (and this way also on the coefficients of the Jones polynomial) of a link of given crossing number, or equivalently on the number of spanning trees of planar graphs with given number of edges. We prove inequalities for the determinant of alternating links in terms of their hyperbolic volume, conjectured non-rigorously by Dunfield. Properties of the knots and links with maximal determinant for given crossing number are investigated. Several number theoretic statements on the determinants of special classes of links are given, leading in particular to knot-theoretic proofs of squareness properties in certain linear recurrent sequences.
Keywords: alternating knots, strongly achiral knots, determinant, Jones polynomial, Lucas numbers, braids, spanning tree, planar graph, hyperbolic volume.
AMS subject classification: 57M25 (primary), 05A20, 05C30, 11B39, 57M12, 57M50 (secondary).

Contents

*Supported by a DFG postdoc grant.

1. Introduction

If Δ_L denotes the (1-variable) Alexander polynomial of a link $L \hookrightarrow S^3$ [Al], then $\det(L) = |\Delta_L(-1)|$ is the order of the homology group $H_1(D_L)$ (over $\mathbb{Z}$) of the double branched D_L cover of S^3 over L (or 0 if this group is infinite) and carries the name "determinant" because of its expression (up to sign) as the determinant of a Seifert [Ro, p. 213] or Goeritz [GL] matrix. This group carries much interesting information on the link (in particular on sliceness [Ro], chirality [HK, St], and unknotting number estimates [We]).

In [Df], Dunfield observed striking relations between the determinant of alternating links and their hyperbolic volume, although not stating this dependence very exactly. Basically, he found that the determinant should have upper and lower bounds, which are exponential in terms of the volume. The main aim of this paper is to prove rigorous versions of Dunfield's inequalities. To state them, here and below $c(L)$ is the crossing number of a link L, and $\mathrm{vol}\,(L)$ denotes the hyperbolic volume of (the complement of) L, or 0 if L is not hyperbolic.

Theorem 1.1 If L is a non-trivial non-split alternating link, then

$$\det(L) \geq 2 \cdot 1.028^{\mathrm{vol}\,(L)}\,.$$

Theorem 1.2 There are numbers $C_1, C_2 > 0$, such that for any hyperbolic alternating link L,

$$\det(L) < \left[\frac{C_1 \cdot c(L)}{\mathrm{vol}\,(L)}\right]^{C_2 \mathrm{vol}\,(L)}\,.$$

Before we come to the proof of these results, we will develop the necessary, and previously initiated, framework to incorporate them into. In [St4] we began the investigation of the question how much the coefficients of the various link polynomials can grow on knots and links of given number of crossings, and showed how via the Kauffman state models [Ka3, Ka2] the problem for the Jones [J] and Alexander [Al] polynomial to be equivalent to this for the determinant. We also found that the maximum will be realized by alternating knots/links. The quest for better estimates of this maximum and studying the properties of the links attaining it will make a substantial part of this paper. In this regard, several questions (suggested by computational results) will be formulated, and partially solved.

For the main theorems we apply the work of Lackenby-Agol-Thurston [La]. However, our proofs will build most decisively on a useful relationship between knot and graph theory, namely that the determinant of an alternating diagram is the number of spanning trees of a certain planar (checkerboard) graph associated to this diagram. This subject was discussed in detail (in the terminology we use also here) in [MS].

Thus most of our results admit direct graph theoretic translations. They will imply certain properties of planar graphs with maximal number of spanning trees for a given number of edges. Graphs with the maximal number of spanning trees have been independently studied by Kelmans and Chelnokov [K, K2, KC]. Their results (and methods) are quite different, since they consider graphs with a fixed number of vertices and relatively high number of edges, most of which are therefore not planar.

In the course of our investigations we were led to consider a certain family of achiral 3-braid links, which leads in turn to a different application. Using work of Hartley and Kawauchi [HK], one can construct many linear recurrent sequences, all of whose odd members are perfect squares. By the correspondence between links and graphs, our particular sequence is found enumerating spanning trees in certain "wheel" graphs.

Our main results improve closely related, but unpublished, previous work of Chris Leininger and Ilya Kofman. They seem qualitatively close to the optimum, modulo the determination and improvement of constants.

The growth of the order of the homology groups of the higher order cyclic branched covers of S^3 over a fixed link L has been studied by Gordon [Go], and later by González-Acuña and Short [GS] and Riley [Ri]. These numbers can still be determined from the Alexander polynomial, but lack a nice combinatorial description, and can be more efficiently approached number-theoretically. (Riley's results indeed use some of the deepest tools from number theory.) Another, although unrelated, occurrence of wheels in knot theory is explored in [BGRT].

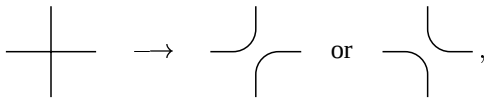
2. The determinant of alternating diagrams

2.1. Estimates for the determinant

Via the relation $\Delta(-1) = V(-1)$ to the Jones polynomial (see [J2, §12]) the determinant provides a bridge between the classical Alexander polynomial and its modern successors [BLM, H, Ka, J], whose nature is rather combinatorial, and it is one of the little topologically understandable information encoded in these invariants. On the other hand, this opens combinatorial approaches for calculating the determinant.

One such approach, which is particularly nice for alternating diagrams, was given by Krebes [Kr] using the Kauffman bracket/state model for the Jones polynomial. (Alternating diagrams are those, in which any strand passes crossings alternatingly over-under.)

If D is an alternating link diagram, then consider $\hat{D} \subset \mathbb{R}^2$, the (image of) the associated immersed plane curve(s). Then $\det(D)$ is equal to the number of ways to splice the crossings (self-intersections) of $\hat{D}$



(1)

so that the resulting collection of disjoint circles has only one component (a single circle; such choices of splicings are called in [Kr] monocyclic states).

To any alternating link diagram one can associate its *checkerboard graph* (see [Ka, DH, Kr, St, Th]). In general the checkerboard graph is a planar graph (with possibly multiple edges) defined up to duality (corresponding to the switch between black and white regions in the checkerboard coloring), and any such graph is the checkerboard graph of some alternating link diagram. (The operations in (1) correspond to contraction and deletion of an edge in the checkerboard graph.)

In [St4, theorem 3.2], we showed via the skein relation for the Jones polynomial that for a diagram D of $c(D)$ crossings, $n(D)$ components, and maximal bridge length $d(D)$ (see [Ki] for latter's definition), we have

$$\left|V(D)\right|_1 := \sum_{2k\in\mathbb{Z}} \left|[V(D)]_{t^k}\right| \leq 5^{c(D)-d(D)}2^{n(D)-1},$$

where $[V]_{t^k}$ is the coefficient of t^k in V. Simple experiments reveal that this bound is not particularly sharp.

The first observation towards an improvement was that using Krebes's approach, we have

Lemma 2.1 With the above notation,

$$\left|V(D)\right|_1 \leq 2^{c(D)-1}.$$

This inequality is of more practical use, since D may have several components, and $d(D)$ is in general small compared to $c(D)$.

Proof. Let D' be the alternating diagram obtained from D by changing crossings. Then by Kauffman's bracket, we have

$$\left|V(D)\right| \leq \det(D')$$

If we resolve any $c(D')-1$ crossings in D' in some arbitrary way, then for the last one there is at most one splitting so as the circle picture to have only one component, so that the result follows. □

Although this lemma already gives (at least in practice) a better estimate, we can push it even a little further. In the theorem below an arborescent diagram is one with Conway basic polyhedron 1^* [Co], or alternatively, a diagram whose checkerboard graph is series-parallel.

Theorem 2.1

1) There exists a constant $C > 0$ such that for any link diagram D of $c(D)$ crossings

$$\det(D) \leq C \cdot \delta^{c(D)}, \tag{2}$$

where $\delta \approx 1.83929$ is the inverse of

$$\delta^{-1} = -\frac{1}{3} - \frac{2}{3\sqrt[3]{17+3\sqrt{33}}} + \frac{\sqrt[3]{17+3\sqrt{33}}}{3} \approx 0.543689,$$

the real positive zero of $f(x) = x^3 + x^2 + x - 1$.

2) If D is an arborescent diagram, then

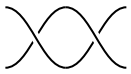
$$\det(D) \leq F_{c(D)+1}, \tag{3}$$

with F_i denoting the *Fibonacci numbers* (defined by $F_1 = 1$, $F_2 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for $n > 2$), and the inequality is sharp (that is, there are relevant diagrams for which equality holds).

Proof. We start with the second part. Let

$$d_n^a := \max\{\det(D) : D \text{ arborescent of } n \text{ crossings}\}$$

An arborescent diagram always has a clasp, a fragment of this type



whose resolution (switching one of the crossings, and eliminating the two crossings by a Reidemeister II move) preserves arborescency. When splicing one of the crossings in the clasp, one of the two resulting diagrams has a kink, so that only one of the splicing of the second crossing can give a circle picture with only one component.

Thus

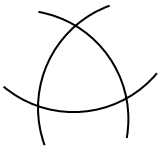
$$d_n^a \leq d_{n-1}^a + d_{n-2}^a,$$

which, together with the trivial correctness for $c(D) = 1, 2$ by induction establishes the inequality (3). The other inequality follows from considering the rational links $L_n = C(\underbrace{1,1,\ldots,1}_{n \text{ times}})$ (here we use Conway's notation [Co]). To see that $\det(L_n) = F_{n+1}$ is an easy calculation with iterated fractions.

The argument for the first part is analogous. Let[1]

$$d_n^\infty := \max\{\det(D) : D \text{ link diagram of } n \text{ crossings}\}$$

Then either D has a clasp, or a triangle



(4)

Then the above argument modifies to show that

$$d_n^\infty \leq d_{n-1}^\infty + d_{n-2}^\infty + d_{n-3}^\infty \qquad (n > 2), \tag{5}$$

and thus d_n^∞ can be estimated by (properly scaled) *Tri*bonacci numbers. □

Remark 2.1

[1] The strange superscript is used for conformity with notation which will be introduced later.

1) We have the explicit expression

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right], \tag{6}$$

so that for arborescent diagrams (2) holds with the smaller base $\dfrac{\sqrt{5}+1}{2} \approx 1.61803$ instead of $\delta \approx 1.83929$.

2) The constant C in (2), the way that it comes from the Tribonacci number estimate, can be certainly effectively calculated, but it does not appear appropriate to do so. The standard way is to apply partial fraction decomposition to the generating (rational) function, obtaining a rather nasty expression involving the real and imaginary parts of the zeros of the denominator polynomial, which in the case of a cubic are already complex enough. Moreover, C can be successively improved by noting that (5) will hardly be sharp in general. Writing down the first values of d_n^∞ we get

n	0	1	2	3	4	5	6
d_n^∞	1	1	2	3	5	8	16

We see that (5) is sharp for $n = 6$, but not for $n < 6$ (because a diagram of $n < 6$ crossings has a clasp, so that we have the simplified recursion $d_n^\infty \le d_{n-1}^\infty + d_{n-2}^\infty$), and it will certainly not be for high n. Thus one can start the iteration on the right of (5) with higher and higher values of n and smaller initial data, obtaining a sequence of constants C with decreasing numerical value but increasing arithmetical complexity ... However, it is worth remarking that, because of connected sums, in every case $C = 1$ must do the job.

2.2. Links with maximal determinant

Again it appears appropriate to make an experiment how good the bound is compared to the actual values of d_n. In [St4, §4] we replaced $c(D)$ by $\operatorname{span} V(D) - 1$ giving an inelucidative picture dominated by non-alternating knots. Thus here we consider only alternating knots and links of given crossing number.

For what follows it will be helpful to make some definitions.

Definition 2.1 Let $S \subset \mathbb{N}$. Then define

$$d_n^S = \max\{\det(D) : n(D) \in S, c(D) = n\},$$

where $n(D)$ is the number of components of D, and let K_n^S be a link attaining the maximum. Set $K_n^i := K_n^{\{i\}}$ and $d_n^i := d_n^{\{i\}}$, $K_n^\infty := K_n^{\mathbb{N}}$, $d_n^\infty := d_n^{\mathbb{N}}$, $K_n := K_n^1$, $d_n := d_n^1$.

This definition already contains a question.

Question 2.1 Is K_n^S unique for all S and n?

In all special cases I checked it was so. However, it is not clear in general. For what follows let us avoid any possible ambiguity by choosing one fixed maximizing link for each n and S. The properties of K_n^S we will state below will be valid *whatever* choice of K_n^S is made, that is, they hold for *all* knots/links that could be chosen as K_n^S.

With this understanding, we point out the following important fact remarked in [St4].

Theorem 2.2 K_n^S is alternating for each n and S.

As tabulation (up to crossing numbers sufficing to give some more concrete picture) are available only for knots, we made a more serious calculation only for $S = \{1\}$. The knots K_n for $n \le 16$ reveal many similarities and are listed in table 1, together with the indication of (lack of) some specific properties and, beside their determinants d_n, some other classical invariants (the genera are not included because their behaviour will later be clarified). The last 6 knots, which are not given in Rolfsen's tables [Ro, appendix], are drawn on figure 1. They are numbered according to the tables in [HT].

The meaning of the properties "flype-free" and "clasp-free" is as follows (for the definition of flypes, see [MT, MT2]).

ncr	kid	det	fibered	claspf	flypef	achir	invert	σ	al braid	bind
3	1	3	✔							
4	1	5	✔							
5	2	7			✔					
6	3	13	✔							
7	7	21	✔							
8	18	45	✔							
9	40	75	✔							
10	123	121	✔							
11	266	209	✔							
12	868	377	✔							
13	3478	663	✔							
14	17895	1145	✔							
15	82477	2037	✔							
16	361172	3581	✔							

(7)

Table 1: The knots K_n for $n \leq 16$ and some of their data (from left to right): crossing number, knot identifier, determinant, fiberedness, clasp-freeness, flype-freeness, achirality, invertibility, signature, existence of alternating braid representation, braid index.

Definition 2.2 A knot or link is called *flype-free*, if there is no essential flype applicable on its alternating diagram, that is, by [MT, MT2], it has only one alternating diagram (modulo moves in S^2).

Definition 2.3 A knot or link is called *clasp-free*, if there is no (possibly trivial) sequence of flypes making any of its alternating diagrams to have a clasp.

The table reveals some striking coincidences and leads to some (more or less justifiable) conjectures (we defer the discussion of the braid index to the end of the paper, because braids will be considered in more detail subsequently).

Conjecture 2.1

1) K_n is fibered for $n \neq 5$.

2) K_n is clasp-free for $n \geq 8$

3) K_n is flype-free for $n \neq 7$.

4) K_n is invertible for odd n and $-$achiral for even n.

5) $\sigma(K_n) \in \{-2, 0, 2\}$.

6) K_n is (the closure of) an alternating braid except for $n = 5, 12$.

7) K_n is prime.

8) K_n is unique.

Although sufficient experimental data is not available for links, it appears that similar phenomena occur there as well. In the following we start the investigation of such phenomena – flype-freeness, clasp-freeness and primality, and give some relations between properties of d_n and such of K_n.

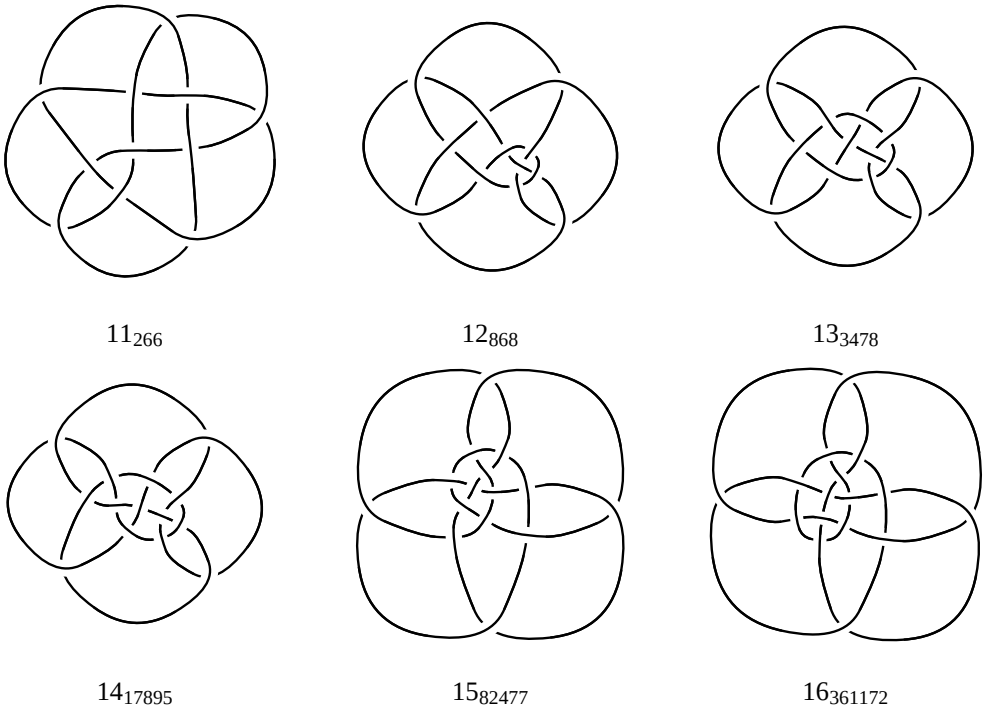


Figure 1: The knots of 11 to 16 crossings with maximal determinant.

2.3. Properties of maximal determinant links

We start with a statement on clasp-freeness.

Theorem 2.3 Let $S = \{1\}$ or $S = \infty$. Then K_n^S is clasp-free for infinitely many values of n. For $S = \infty$, more specifically, every subset of $\mathbb{N}$ of the form $\{x,\ x+2,\ x+4, \ldots,\ x+160\}$ contains at least two such n.

Unfortunately, at this stage we have no tools to deal with some of the points in conjecture 2.1. There is some causality between the various properties. For, example alternating braids are fibered. On the other hand, evidence for other such relations from the common knot tables can be misleading. It may appear at first glance, for example, that clasp-free alternating knots are fibered, too. However, this is not always true. The simplest example of a clasp-free alternating non-fibered knot is 13_{4695}.

The proof of theorem 2.3 initiates from some more conditional, but still self-contained properties of K_n related to such of d_n.

Proposition 2.1

a) If $d_n > \max(3d_{n-2}, d_{n-1} + 2d_{n-3})$, then K_n is clasp-free.

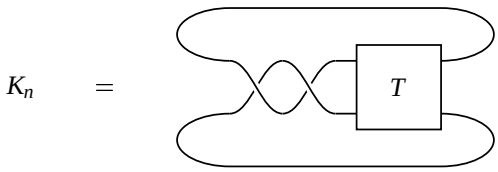
b) If for $S = \{1\}$ or $S = \infty$ we have $d_n^S > d_l^S d_{n-l}^S$ for any $1 < l < n-1$, then K_n^S is prime.

c) If for $S = \infty$ we have $d_n^S > 3d_l^S d_{n-l-1}^S$ for any $1 < l < n-2$, then K_n^S is flype-free.

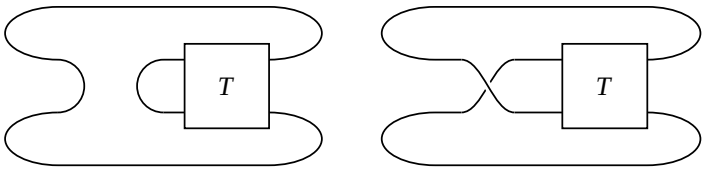
d) If for $S = \infty$, $d_n^S > \min(3d_{n-2}^S, d_{n-1}^S + 2d_{n-3}^S)$, then K_n^S is clasp-free.

Proof.

a) Assume K_n has a clasp, i.e.



Then splicing of the one crossings in the clasp gives a knot and a 2 component link.



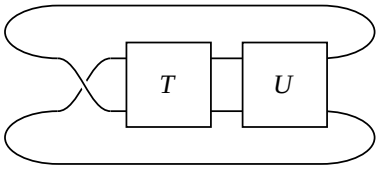
(8)

Case 1. (a) is the knot and (b) is the (2 component) link. Then (a) contributes at most d_{n-2} to d_n and (b) has a mixed crossing (unless it is split in which case it has zero determinant), whose two splicings give again knots, so the contribution is at most $2d_{n-2}$.

Case 2. (b) is the knot and (a) is the link. Then (b) contributes $\leq d_{n-1}$ and (a) contributes after splicing a mixed crossing $\leq 2d_{n-3}$.

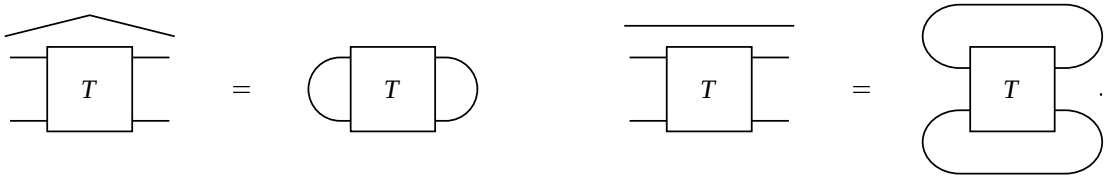
b) This is straightforward from the multiplicativity of the determinant under connected sum and the result of Menasco [Me].

c) Assume that K_n is not flype-free, in particular a diagram of $K = K_n$ is of the form



(9)

with $c(T), c(U) > 1$. Let the two possible closures of a tangle be denoted as follows:



With this notation (9) can be written as

$$K = \overline{1, T, U},$$

where '1' is the 1-tangle and the comma operator denotes tangle sum in the Conway [Co] sense. Then by Krebes' calculus [Kr] for his invariant Kr we have

$$\frac{\det(K_n)}{*} = \mathrm{Kr}(1,T,U) = \frac{\pm 1}{1} \oplus \frac{\pm \det(\overline{T})}{\det(\widehat{T})} \oplus \frac{\pm \det(\overline{U})}{\det(\widehat{U})}$$

so that comparing the numerators we obtain

$$d_n = \det(K_n) = \pm\big(\det(\overline{T}) + \det(\widehat{T})\big) \cdot \det(\widehat{U}) \pm \det(\widehat{T}) \det(\overline{U}) \leq 3 d_{n-l-1} d_l,$$

with $l = c(T)$. Here $\oplus$ is the "fraction" addition in $\mathbb{Z} \times \mathbb{Z}/(p,q) \sim (-p,-q)$.

d) Use the inequality $d_n^S \leq d_{n-1}^S + d_{n-2}^S$ following from the clasp and

$$d_{n-1}^S \leq 2d_{n-2}^S,$$

following from splicing any arbitrary crossing in an $n-1$ crossing link diagram. □

We now prove several, mostly unconditional, inequalities between the d_n^S.

Lemma 2.2

a) $d_{n+1}^1 \geq d_n^1$.

b) $d_n^k \leq 2d_{n-1}^{k-1}$.

c) $d_n^1 \leq d_{n-1}^1 + d_{n-1}^2$.

d) $d_{n+3}^\infty \leq 7d_n^\infty$.

e) $d_{n+2}^1 \leq 5d_n^1$.

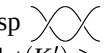
f) $d_{n+3}^1 \leq 11d_n^1$.

g) $d_{n+1}^\infty \leq 2d_n^\infty$.

h) If K_{n+2}^∞ is not clasp-free, then $d_{n+2}^\infty \leq 3d_n^\infty$.

i) If K_{n+3}^∞ is not clasp-free, then $d_{n+3}^\infty \leq 6d_n^\infty$.

Proof.

a) Replace a crossing  in K_n^1 by a clasp  such that the resulting diagram is again a diagram of a knot K'. Then K' has $n+1$ crossings and that $\det(K') \geq \det(K)$ follows by splicing one of the 2 crossings in the clasp.

b) This follows directly from splicing any mixed crossing in K_n^k (if such does not exist, then K_n^k is split, and has zero determinant, which is impossible).

c) This follows directly from splicing any crossing in K_n^1.

d) This follows by splicing the 3 crossings on the edges of a triangle in K_{n+3}^∞. One of the resulting 8 diagrams has a split loop.

e) We have

$$d_{n+2}^1 \overset{b)}{\leq} d_{n+1}^1 + d_{n+1}^2 \overset{b),\,c)}{\leq} d_n^1 + d_n^2 + 2d_n^1 \overset{a)}{\leq} 5d_n^1.$$

f) Follows as e), but applying b) and c) once more, before applying a).

g) This is trivial.

h) This follows from proposition 2.1, d).

i) Use g) and h). □

We now come to the proof of theorem 2.3.

Proof of theorem 2.3. Let K be a knot or link of $n = c(K)$ crossings. Then $d_n^\infty \geq \det(K)$, and $d_{n+k}^\infty \geq d_k^\infty \cdot \det(K)$ for any k because of connected sums with K. Therefore, $d_{k+2n}^\infty \geq \det(K)^2 d_k^\infty$. Assume now l of the links $K_{k+2}^\infty, K_{k+4}^\infty, \ldots, K_{k+2n}^\infty$ have a clasp. Then by g) and h) of lemma 2.2 we have

$$d_{k+2n}^\infty \leq 3^l 4^{n-l} d_k^\infty .$$

On the other hand

$$d_{k+2n}^\infty \geq \det(K)^2 d_k^\infty .$$

Thus $\det(K)^2 \leq 3^l 4^{n-l}$, or

$$\sqrt[n]{\det(K)} \leq \sqrt[2n]{3^l 4^{n-l}} = 2\left(\frac{\sqrt{3}}{2}\right)^{l/n} .$$

Thus

$$l \leq n \cdot \frac{\ln \sqrt[n]{\det(K)} - \ln 2}{\ln \sqrt{3} - \ln 2} . \tag{10}$$

It is clear that this to give a non-trivial estimate, one must have $\sqrt[n]{\det(K)} > \sqrt{3}$. This, unfortunately, is not the case for knots of ≤ 16 crossings, and we need to look at more complicated examples. Luckily, however, the determinant can be computed via the Seifert matrix in polynomial time. A package for this using braid representations was written by S. Orevkov for MATHEMATICA™ [Wo]. Using it I found the closed 81 crossing alternating 10-string braid

$$K = \widehat{\ }\Big(\left(\sigma_1\sigma_3\sigma_5\sigma_7\sigma_9\sigma_2^{-1}\sigma_4^{-1}\sigma_6^{-1}\sigma_8^{-1}\right)^9 \Big)$$

(with σ_i being, as usual, the Artin generators), where $\det(K) = 24743382596536452489$, and hence $\mu_K := \det(K) \cdot 3^{-c(K)/2} \approx 1.17503$.

Putting this into (10) gives a r.h.s. of integer part 79, so the result follows for $S = \infty$.

Now let $S = \{1\}$. If for almost all n the knot K_n had a clasp, then

$$d_n \leq \max\left(3d_{n-2}, d_{n-1} + 2d_{n-3}\right)$$

coming from proposition 2.1.a) shows $d_n \leq C \cdot \sqrt{3}^n$ (the zero of $2x^3 + x - 1$ on $[0,\infty)$ close to $1/2$ is higher than $1/\sqrt{3}$, so that the higher rate of growth comes from the first alternative in the maximum), contradicting the existence of the above quoted example. □

We remark that the inequality $d_{a+b} \geq d_a d_b$ implies that $\tilde{d} := \lim_{n\to\infty} \sqrt[n]{d_n}$ exists and that

$$\lim_{n\to\infty} \sqrt[n]{d_n} = \sup_K \sqrt[c(K)]{\det(K)} ,$$

where the supremum is taken over all (alternating) knots K. Thus we have

Corollary 2.1 $\sqrt{3} < \sqrt[81]{24743382596536452489} \approx 1.7355032 \leq \lim_{n\to\infty} \sqrt[n]{d_n} \leq \delta$. □

We should also point out that the lower bound $\sqrt{3}$ is of no special importance – in can be successively improved by finding knots K with higher value of $\sqrt[c(K)]{\det(K)}$, and for this calculating the determinant of appropriate more and more complicated knots. This will be done in §4, thus improving theorem 2.3 for $S = \infty$, and showing that at least $2/9$ of all K_n^∞ are clasp-free.

Question 2.2 Is $\tilde{d} = \delta$, or $\tilde{d_\infty} := \lim_{n\to\infty} \sqrt[n]{d_n^\infty} = \delta$?

Remark 2.2 If we have a knot K of k crossings with $\mu_K > 1$ and know d_n for $n < k$, then we can obtain an explicit (upper) estimate depending on $\varepsilon > 0$ of the smallest number n_0 with $\sqrt[n]{d_n} > \sqrt{3} + \varepsilon$ for any $n > n_0$, which – if sufficiently small – can be used to prove the clasp-freeness of K_n for almost all n. There is little hope to be able to proceed this way, though. Indeed $\mu_K > 1$ occurs only for rather complicated knots, and it does not seem feasible to calculate d_n for n larger than about 20. For example, for

$$K = \widehat{\ }\Big(\left(\sigma_1\sigma_3\sigma_5\sigma_7\sigma_9\sigma_2^{-1}\sigma_4^{-1}\sigma_6^{-1}\sigma_8^{-1}\right)^7 \Big),$$

we have $\mu_K = 0.98\ldots$, although it already has crossing number 63.

Remark 2.3 Similarly to (10), parts i) and d) of lemma 2.2 show that

$$l \leq n \cdot \frac{\ln \sqrt[n]{\det(K)} - \ln \sqrt[3]{7}}{\ln \sqrt[3]{6} - \ln \sqrt[3]{7}} \tag{11}$$

for the number l of elements x in sets of the form $\{a, a+3, \ldots, a+3(n-1)\}$ with K_x^∞ clasp-free (and $n = c(K)$). However, the problem to find a knot K with $\sqrt[c(K)]{\det(K)} > \sqrt[3]{6} \approx 1.81$ is still computationally inaccessible, even if $\sqrt[3]{6} < \delta$, so (11) is of little practical use as of now. In the knot case ($S = \{1\}$), I was unable to prove an analogon of lemma 2.2 h) and i), such that '$\sqrt{3}$' in (10) could be replaced by something still $< \delta$. Thus, a direct estimate is so far not possible.

Another property of the K_i follows from the work we have done in [St5], which rewards us with an easy proof of a growth statement for the *genera* $g(K_n)$ of the K_n (see [Ga]).

Theorem 2.4 $g(K_n) \to \infty$. More exactly, for any $\varepsilon > 0$ we have $g(K_n) \geq \log_{8+\varepsilon} n$ for n large enough.

Proof. That $g(K_n) \to \infty$ follows by [St5, theorem 3.1], because $\det(K)$ grows only polynomially in $c(K)$ for alternating knots K of fixed genus. The specific growth statement comes from an estimate of this polynomial from [St6]. We derived such an estimate in [St6, theorem 3.1]:

$$\det(K) \leq \max_{0 \leq d' \leq d_{g(K)}} \left[\frac{Cc(K)}{d'} \right]^{d'}, \tag{12}$$

for K alternating and some constant $C > 1$, where $d_{g(K)}$ can be defined by

$$d_g := \min \left\{ i \in \mathbb{N} : \limsup_{n \to \infty} \frac{|A_{n,g}|}{n^i} = 0 \right\}, \tag{13}$$

with

$$A_{n,g} := \{\, K \text{ alternating}, g(K) = g, c(K) = n \,\}. \tag{14}$$

We also use the fact proved in [St5] that

$$d_g = O(8^g). \tag{15}$$

Assume now that there is a sequence $\{n_i\}$ and an $\varepsilon' > 0$ with $g(K_{n_i}) / \log_8(n_i) \leq 1 - \varepsilon'$. Then

$$n_i^{\varepsilon''} \gg 8^{g(K_{n_i})} \tag{16}$$

for any $\varepsilon'' \in (1 - \varepsilon', 1)$.

We have that the maximal value of

$$f_n(d') = \left(\frac{n}{d'}\right)^{d'}$$

for $d' \in (0,d]$ is attained for $d' = \min\{d, \frac{n}{e}\}$, and for $d \le n/e$ the function f_n is monotonously growing ($e = 2.71828\ldots$). Because of (16) for i large enough the former alternative in the maximum applies, and (12) and (15) give

$$\det(K_{n_i}) \le \left[\frac{C n_i}{8^{g(K_{n_i})}}\right]^{C' 8^{g(K_{n_i})}}$$

for some constants C and C'. Using (16) we get

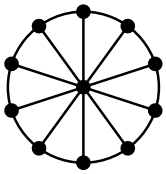
$$\det(K_{n_i}) \le (C n_i)^{C' n_i^{\varepsilon''}} \tag{17}$$

But because of $\varepsilon'' < 1$ we have $C' n_i^{\varepsilon''} (\ln n + \ln C) \ll C'' n_i$ for any $C'' > 0$, which exponentiated shows from (17) that $\{\det(K_{n_i})\}$ grows subexponentially, a contradiction. □

3. Recursive sequences and alternating braids

3.1. Squares in linear recurrences

Consider the wheel graph W_n of $n+1$ vertices.



$= \quad W_{10}$

The number c_n of spanning trees in W_n can be computed by distinguishing the number of edges of the spanning tree incident to the central vertex of the wheel, and counting the spanning forests of the necklace graph remaining from the spanning tree in W_n after removing the central vertex. This was carried out in [My, p. 469–470]. The resulting sequence is 1, 5, 16, 45, 121, ... and can be expressed by the *Lucas numbers* L_n given by $L_1 = 2$, $L_2 = 1$ and $L_n = L_{n-1} + L_{n-2}$ for $n > 2$; the relation is $c_n = L_{2n} - 2$. Another occurrence of this sequence is in [Re] as the number of certain unimodular matrices. See also [My2] and [Sl, sequence 004146].

An alternative expression of c_n using Fibonacci numbers is

$$c_n = F_{2n} + 2\sum_{i=1}^{n-1} F_{2i}\,. \tag{18}$$

Its equivalence to the above one can be shown by elementary generating series arguments, for example.

A closer look on the numbers c_n reveals that for odd n, c_n is a (perfect) square. Although there have been, in particular recently, many related results, e. g. [Ch, DF, Du, Du2, Es, MD, Mr], I did not find an explicit statement of this observation. Nonetheless, it is suggestive that this phenomenon should not be the result of an accidental coincidence, and indeed a combinatorial explanation of it is possible by writing down the explicit formula for L_n

$$L_n = \left(\frac{1+\sqrt{5}}{2}\right)^{n-1} + \left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\,. \tag{19}$$

However, the same phenomenon occurs also with (the odd index members of) some closely related sequences like

$$c'_n = c_n + F_n^2 + 2F_{2n} \quad \text{and} \quad c''_n = c_n + 4F_n^2 + 4F_{2n}\,. \tag{20}$$

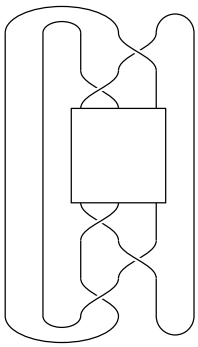
We will give an explanation of such a phenomenon in terms of knot theory (showing how to find further such sequences and prove their squareness in a much easier and more elegant way than via the naive arithmetical approach). It turns out, that the numbers c_n occur as determinants of some (alternating 3-braid) knots and links.

3.2. Determinants of alternating braids

Originally the examples $K_8 = 8_{18}$ and $K_{10} = 10_{123}$ suggested to consider for the proof of theorem 2.3 for $S = 1$ more closely the sequence of alternating 3-braids $\widehat{(\sigma_1\sigma_2^{-1})^k}$. Although these braids closely fail in giving the desired examples, they can be used to give an estimate for arbitrary alternating 3-braids and establish the connection to the (modified) Lucas numbers mentioned in the introduction of this section.

Lemma 3.1 $\det\big(\widehat{(\sigma_1\sigma_2^{-1})^k}\big) = c_k$.

Proof. Consider the 2 uppermost crossings of $(\sigma_1\sigma_2^{-1})^k$, the ones from the last factor in the power.



Splicing the uppermost one as $\smile$ gives the rational knot $C(\underbrace{1,1,\ldots,1}_{2k-1})$, whose determinant as we mentioned is F_{2n}. Splicing the uppermost crossing as $)$ $($ and the second uppermost one as $\smile$ gives, after deleting the kink from the lowermost crossing, a rational link $C(\underbrace{1,1,\ldots,1}_{2k-3})$ with determinant F_{2k-2}. Finally, splicing both crossings as $)$ $($ gives $(\sigma_1\sigma_2^{-1})^{k-1}$, and then the result follows by induction from (18). □

Corollary 3.1 If β is an alternating 3-braid, then $\det(\hat{\beta}) \leq \left(\frac{\sqrt{5}+1}{2}\right)^{c(\hat{\beta})}$, with the inequality in general sharp up to an additive constant.

Proof. Use that any $\beta \in B_3$, except for the ones in the lemma, have a clasp. Splicing one of the crossings in the clasp, we obtain a 3-braid with one crossing less and a rational knot. The contribution of the rational knot of $c(\hat{\beta}) - 2$ crossings to $\det(\hat{\beta})$ is estimated by theorem 2.1.2) to

$$\frac{1}{\sqrt{5}}\left(\frac{2}{\sqrt{5}+1}\right)\cdot\left(\frac{\sqrt{5}+1}{2}\right)^{c(\hat{\beta})} + C \tag{21}$$

for some fixed constant C, and this of the braid by induction on $c(\hat{\beta})$ by

$$\left(\frac{2}{\sqrt{5}+1}\right)\cdot\left(\frac{\sqrt{5}+1}{2}\right)^{c(\hat{\beta})}.$$

But

$$\frac{2}{\sqrt{5}+1} + \frac{1}{\sqrt{5}}\left(\frac{2}{\sqrt{5}+1}\right) = \frac{2}{\sqrt{5}} < 1, \tag{22}$$

so starting the induction for $c(\hat{\beta})$ large enough to gobble the C in (21) by the strict inequality in (22) and checking the initial cases directly, one is done. □

The links of the form $\widehat{(\sigma_1\sigma_2^{-1})^k}$ are not new. They have been considered for a while, notably in [JP] (at least in the knot case $3 \nmid k$). There it was observed that for odd k (for which the knots are also called "turks head knots"), the braid $\widehat{(\sigma_1\sigma_2^{-1})^k}$ is of the form $\beta\overline{\beta}$, where $\overline{\beta}$ is obtained from $\beta \in B_n$ by the map $\sigma_i^{\pm 1} \mapsto \sigma_{n-i}^{\mp 1}$, and hence $\widehat{(\sigma_1\sigma_2^{-1})^k}$ is strongly $+$achiral, i. e., admits an embedding into $\mathbb{R}^3$ fixed by the (orientation-reversing) involution $(x,y,z) \mapsto (-x,-y,-z)$, such that this involution additionally preserves the orientation of the knot/link. By the result of [HK] (stated and proved only for knots but true by the same argument also for links [1]), such knots/links have as Alexander module a double $A \oplus A$, so that in particular the Alexander polynomial, and hence the determinant is a square. (Long [Lo] has stronger shown that such knots are algebraically slice.) This, together with lemma 3.1, shows the statement alluded to in the introduction.

Theorem 3.1 c_k is a square number for k odd (hence so is the number of spanning trees in wheel graphs with an odd number of spokes or the by 2 decreased Lucas number L_n with $n \equiv 2 \bmod 4$). □

The fact that the odd index number knots are still at least achiral (in the usual, weak, sense), shows that by [St] c_n for n even is at least the sum of two squares. Unfortunately, contrarily to the result obtained for the odd index parity, there seems no tool available to examine effectively the even index number case. However, the test of the prime decomposition of c_n leads to conjecture even more, namely that these numbers are of the form $c_n = 5a_n^2$ for n even, and this can be indeed confirmed from the explicit formula for L_n (19). (This observation seems to fit into a more general pattern conjecturally described at the end of this note.)

On the other hand, for odd k it is clear that now a similar procedure can be applied to more general braids. For example applying the argument to $\beta\overline{\beta}$ with $\beta = \sigma_1(\sigma_1\sigma_2^{-1})^k$ and $\beta = \sigma_1^2(\sigma_1\sigma_2^{-1})^k$ gives the property for c'_n and c''_n in (20). Considering $\beta\overline{\beta} = (\sigma_1^l\sigma_2^{-l})^k$ gives a more general version of theorem 3.1.

Theorem 3.2 Let $b_0 = 0$, $b_1 = 1$ and $b_n = b_{n-2} + lb_{n-1}$. Then

$$l\left(2\sum_{i=1}^{k-1} b_{2i} + b_{2k}\right)$$

is a square for k odd. □

Considering 5-braids may give similar, however, less pleasant statements of this kind.

On the other hand, arithmetic results can have some knot theoretic consequences.

Corollary 3.2 Any rational knot $C(1,1,\ldots,1)$ ("twist plat knot" [Ju]) is not algebraically slice.

Proof. Use the result of [Ch] that no odd Fibonacci number > 1 is a square. □

Remark 3.1

1) Of course the same argument shows that $C(1,1,\ldots,1)$ is not strongly $+$achiral, but this follows more generally for any rational knot from the result of Hartley-Kawauchi as the 2-branched cover homology group $H_1(D_K)$ is cyclic (and non-trivial), and hence not a double.

2) A similar property could be shown for the rational knots $C(3,1,\ldots,1)$ from the result on the Lucas numbers.

3) The knot-theoretic counterpart of the non-squareness of c_k for even k is true also by different arguments. It was remarked in [St3] how the work of Murasugi [Mu] on the Alexander polynomial of periodic knots implies that the Alexander polynomial of any non-trivial knot (and analogously, link), which is the closure of the square of some braid (here $(\sigma_1\sigma_2^{-1})^{k/2}$), is not a square, so that the knot is not strongly $+$achiral (although it is weakly $+$achiral).

[1] except in the case, when the Alexander module is not completely torsion, which is, however, trivial, as then the Alexander polynomial vanishes

For general strand number, the results on 3-braids, and more specifically on the powers of $\sigma_1\sigma_2^{-1}$, generalize followingly.

Theorem 3.3 If $\beta_i \in B_n$ are alternating braids of fixed strand number.

1) Then $\lambda_{\{\beta_i\}} := \limsup_{n\to\infty} \sqrt[c(\beta_i)]{\det(\widehat{\beta_i})} \leq \delta$.
2) Moreover, if $\beta_i = \beta^i$ are powers of some fixed braid β, then $\lambda_\beta := \lambda_{\{\beta^i\}}$ is the norm of an algebraic (possibly complex) number of degree $\leq C_n$, where $C_n = \frac{1}{n+1}\binom{2n}{n}$ is the n-th Catalan number.

For the proof of theorem 3.3 we need a technical lemma.

Lemma 3.2 Let $\lambda_1,\ldots,\lambda_l$ $(l>1)$ be distinct unit norm complex numbers and $\{a_{j,n}\}_{n=1}^{\infty}$ for $j=1,\ldots,l$ be sequences with $|a_{j,n}| \geq \varepsilon$ for some $\varepsilon > 0$ and all j,n, and

$$\frac{a_{j,n+1}}{a_{j,n}} \xrightarrow[n\to\infty]{} 1.$$

Then the sequence $s_n := \sum_{j=1}^{l} a_{j,n}\lambda_j^n$ does not converge (in particular, not to 0).

Proof. Assume $s_n \to s$ for some $s \in \mathbb{C}$. If

$$M_n := \begin{pmatrix} 1 & \cdots & 1 \\ \frac{a_{1,n+1}}{a_{1,n}}\lambda_1 & \cdots & \frac{a_{l,n+1}}{a_{l,n}}\lambda_l \\ \frac{a_{1,n+2}}{a_{1,n}}\lambda_1^2 & \cdots & \frac{a_{l,n+2}}{a_{l,n}}\lambda_l^2 \\ \vdots & \ddots & \vdots \\ \frac{a_{1,n+l-1}}{a_{1,n}}\lambda_1^{l-1} & \cdots & \frac{a_{l,n+l-1}}{a_{l,n}}\lambda_l^{l-1} \end{pmatrix},$$

then

$$M_n \begin{pmatrix} a_{1,n}\lambda_1^n \\ \vdots \\ a_{l,n}\lambda_l^n \end{pmatrix} \longrightarrow \begin{pmatrix} s \\ \vdots \\ s \end{pmatrix},$$

and M_n converge to a Vandermonde matrix, which is not singular, so that $||M_n^{-1}||$ is bounded.

Therefore, in particular $\left\{ (a_{1,n}\lambda_1^n,\ldots,a_{l,n}\lambda_l^n) : n > n_0 \right\}$ must lie in some ε'-ball for n_0 large enough. But $|a_{j,n}| \geq \varepsilon$ shows that these components stay outside of some neighborhood of the origin, and

$$\frac{a_{i,n+1}\lambda_i^{n+1}}{a_{i,n}\lambda_i^n} \longrightarrow \lambda_i \neq 1$$

for some i gives a contradiction for ε' small enough. □

Proof of theorem 3.3.

1) This is clearly a consequence of corollary 2.1.
2) Let β_i be n-strand braids and SD_n be the Kauffman algebra of [Ka, definition 3.5] with the special parameter $A = i = \sqrt{-1}$ (so that a separate loop trivializes). It can be shown (see [Ka, theorem 4.3]) that SD_n is generated

by the C_n loop-free diagrams connecting a pair of $n+n$ points on bottom and on top by n lines. The dimension of SD_n is therefore (at most) C_n. For example for $n=3$ we have the following 5 elements:

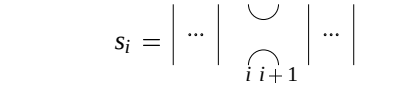
(23)

The multiplication is given by stacking up and eventual killing of the resulting diagram if it has a loop. For example

Let ϕ_β be the linear operator

$$SD_n \ni x \overset{\phi_\beta}{\longmapsto} x\prod_{j=1}^{k}(1+s_{i_j}) \in SD_n$$

with

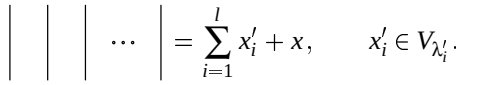


associated to $\beta=\prod_{j=1}^{k}\sigma_{i_j}^{(-1)^{i_j}} \in B_n$.

It can be decomposed (at least over $\mathbb{C}$) into eigen values λ_i and Jordan box spaces V_i. Fix a Jordan basis of ϕ_β (as it has integer coefficients in some rational basis of DS_n, the Jordan basis can be chosen to lie in some degree $\le C_n$ extension of $\mathbb{Q}$) and let $\lambda'_1,\dots,\lambda'_l$ be the eigen values λ_i of ϕ_β of maximal norm, whose V_i are not completely killed by the $\mathbb{C}$-linear extension of the map

$$\chi : DS_n \ni \boxed{T} \longmapsto \det\left(\boxed{\boxed{T}}\right) \in \mathbb{N}\subset\mathbb{C}.$$

Consider the Jordan decomposition of $Id=1\in DS_n$



Since

$$0\neq \det(\hat\beta)=\det(\widehat{1\cdot\beta})=\chi(1\cdot\beta)=\chi(\phi_\beta(1))\,,$$

and ϕ_β preserves all $V_{\lambda'_i}$, there exists a $1\le i_0\le l$ with $x'_{i_0}\neq 0$.

Let $d'_i:=\dim V_{\lambda'_i}$ for $1\le i\le l$. Then each x'_i has a contribution to $\det\left(\widehat{\beta^n}\right)$ of the form

$$\sum_{j=1}^{d'_i} a_j P_j(n)\,(\lambda'_i)^{n-d'_i+j} = O(P_{d'_i}(n)\lambda_i'^n)$$

for some $a_j\in\mathbb{C}$ (the coefficients of x'_i in the Jordan basis of $V_{\lambda'_i}$) and $P_j(n)\in\mathbb{Q}[n]$ with $\deg P_j\le d'_i$. Thus

$$\det(\widehat{\beta^n})=\sum_{j=1}^{l'}\tilde P_j(n)\,\lambda_j'^n \quad + \quad \text{lower magnitude terms} \tag{24}$$

for some $1\le l'\le l$ and $0\neq\tilde P_j(x)\in\mathbb{C}[n]$ (discard possible $\tilde P_j=0$) with $\deg\tilde P_j\le\max_{i=1}^{l} d'_i$. If we show now

$$\limsup_{n\to\infty}\sqrt[n]{\det(\widehat{\beta^n})}=\limsup_{n\to\infty}\sqrt[n]{\sum_j\tilde P_j(n)\,\lambda_j'^n}=|\lambda'_i|\,,$$

we are through, as λ'_i is the root of a polynomial with rational coefficients of degree C_n.

If $l'=1$ the claim is straightforward from (24) and for $l'>1$ this follows from lemma 3.2 by rescaling, setting $a_{j,n}:=\tilde P_j(n)$. □

	0	0	0	1	0
	0	0	1	0	1
	0	1	0	0	1
	1	0	0	0	0
	0	1	1	0	0

Table 2: The table for the pairing $<,>_3$.

The combination of both statements in theorem 3.3 also suggests that if λ is an eigen value of ϕ_β for some β, then $|\lambda| \leq \delta^{c(\beta)}$. Although a dominating eigen value of ϕ_β may have a Jordan space killed by taking the determinant of the usual braid closure, there will often be a (linear combination of) other closure(s) under which not the whole Jordan space is killed (and then for these exotic closures the same argument will apply). It is known from the study of meanders in theoretical physics that indeed one can always find such a closure. Thus we have

Corollary 3.3 Any eigen value λ of ϕ_β for any $\beta \in B_n$ has $|\lambda| \leq \delta^{c(\beta)}$.

Proof. Define a pairing (or binary quadratic form) on DS_n by

$$\left\langle \boxed{T_1}, \boxed{T_2} \right\rangle_n = \begin{cases} 1 & \text{if the } \textit{meander} \text{ has one loop}, \\ 0 & \text{else}. \end{cases}$$

For example, $<,>_3$ is given by the table 2. By the above remark and lemma 3.2 it suffices to know that $<,>_n$ is non-degenerate. This follows from an explicit expression for its determinant, see formula (5.18) in [DGG]. □

Remark 3.2 Table 2 shows easily $<,>_3$ to be non-degenerate. Prior to getting referred by M. Khovanov to the above paper [DGG], by computer I checked it also for $n = 4 \ldots 10$. It is also an easy exercise to see that $< T_1, . >\not\equiv 0$ if T_1 is a single diagram, as one can always find a diagram T_2 with $\widehat{T_1 T_2} = \bigcirc$. However, the argument does not extend in an easy way to arbitrary linear combinations of diagrams, and the proof of non-degeneracy is certainly quite non-trivial.

4. Spanning trees in planar graphs

4.1. Determinant and spanning trees

It is a consequence of Kauffman's bracket [Ka3], that the determinant of an alternating diagram is equal to the number of spanning trees of its checkerboard graph. Lemma 3.1 is just the explicit derivation of a special case of this correspondence, which was discussed extensively in [MS]. In this paper, we use the language of [MS], but do not repeat details to save space.

It is known that for a link K, $\det(K)$ is odd if and only if K is a knot. Thus now the question on the growth of d_n and d_n^∞ could be reformulated entirely in terms of graph theory:

Proposition 4.1 d_n^∞ is the maximal number of spanning trees in a planar graph with n edges (multiple edges allowed and counted by multiplicity). d_n is the maximal *odd* number of spanning trees in a planar graph with n edges. □

Theorem 2.3 can be interpreted like:

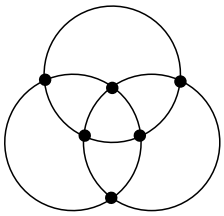
Proposition 4.2 For infinitely may values of n the planar graphs with n edges and maximal number of spanning trees (or maximal odd number of spanning trees) have no valence-two vertices and no multiple edges. □

Further properties of the knots K_n are also related to properties of their checkerboard graph. For example, the uniqueness and flype-freeness of K_n imply the uniqueness of the graph G_n with n edges and d_n spanning trees. The achirality and flype-freeness of K_n imply that G_n is self-dual, and the achirality of K_n for itself by the result of [DH] that G_n has a (possibly different) self-dual planar embedding.

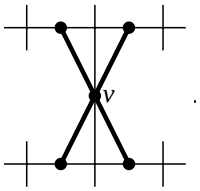
4.2. Planar graphs with many spanning trees

We will use now the graph description of the determinant to improve the estimate in theorem 2.3 for $S = \infty$.

Since any link diagram can itself be considered as a planar graph (each crossing being a vertex of valence 4), we can build a new link diagram of which the previous one (regarded as 4-valent graph) is the checkerboard graph. It turns out that this procedure, when iterated, is very good at generating diagrams with high determinant (or graphs with high number of spanning trees), in particular if we start with a clasp-free diagram (4-valent graph with no multiple edges). The simplest such diagram is this of the Borromean rings

$D_0 \quad =$  .

Call this graph D_0. Then one obtains D_{n+1} from D_n by putting a vertex of D_{n+1} to correspond to an edge of D_n and connecting a vertex v of D_{n+1} as follows:



(Here the thick lines correspond to edges in D_{n+1} and the thin ones to edges in D_n.) Sometimes D_{n+1} is called the *line graph* of D_n. This procedure doubles the number of edges and vertices. However, the determinant can be effectively computed.

Lemma 4.1 Let D be an alternating diagram of n crossings. Then in D there is a $1-1$ correspondence between crossings and bridges (all of length 1). For $i, j = 1, \ldots, n$ define a matrix $M = (m_{i,j})_{i,j=1,\ldots,n-1}$ by setting for $i, j = 1, \ldots, n-1$

$$m_{i,j} := \begin{cases} 2 & i = j, \\ -1 & i \neq j \text{ and bridges } i \text{ and } j \text{ meet at crossing } i \text{ or } j, \\ 0 & \text{otherwise}. \end{cases}$$

Then $\det(D) = \det(M)$.

Proof. This is a classical fact from knot theory. Basically M is a presentation matrix of the Alexander module $\Lambda(t)$ of D specialized at $t = -1$, hence its determinant is the order of this group, which is $\det(D)$. Graph theoretically, this is a variant of the matrix-tree theorem (see [MS]). □

Since calculating determinants has cubic complexity, the complexity of $\det(D_n)$ is exponential in n with basis roughly 8. However, in practice the base is about 16, since the number of digits in the integers gets doubled. Practical computations were possible for $n \leq 9$, using MATHEMATICA™, and their result can be briefly summarized in the

following table, giving the number of digits of $\det(D_i)$, the CPU time for its calculation, and the number of crossings and components nc_i of D_i.

i	# digs $\det(D_i)$	# crsgs D_i	comp. CPU time	nc_i
0	2	6	0	3
1	3	12	0	4
2	6	24	0	6
3	12	48	$0.02''$	8
4	24	96	$0.28''$	12
5	48	192	$4.2''$	16
6	97	384	$1'12''$	24
7	194	768	$11'45''$	32
8	388	1536	$5^h15'40''$	48
9	777	3072	$83^h10'37''$	64

Here D_i is identified with its 4-valent (and not checkerboard) graph, that is,

$$\det(D_i) \;=\; \text{number of spanning trees of } D_{i-1}\,.$$

(The determinants themselves are clearly too messy to print directly, but we will come back to their numerical values in the next section.)

We obtain the following estimates:

Lemma 4.2 $d^{\infty}_{6\cdot 2^i} \geq \det(D_i), \qquad d_{6\cdot 2^i - nc_i + 1} \geq \dfrac{\det(D_i)}{2^{nc_i-1}}.$

Proof. The first claim is trivial. The second one follows by applying $nc_i - 1$ times lemma 2.2 b). □

Corollary 4.1 $\tilde{d}_\infty = \sup_k \sqrt[k]{d^\infty_k} \geq \sqrt[6\cdot 2^i]{\det(D_i)}, \qquad \tilde{d} = \sup_k \sqrt[k]{d_k} \geq \sqrt[6\cdot 2^i - nc_i + 1]{\dfrac{\det(D_i)}{2^{nc_i-1}}}.$ □

Thus we can with every new value for $\det(D_i)$ continuously improve (see table below) the estimate on these suprema (columns 2 and 3), and using (10) also the estimates on the number of clasp-free K^∞_n's in intervals of step 2 of length $6\cdot 2^i$ (column 4), finally showing that at least $701/3072$ (or about $2/9$) of all K^∞_n's are clasp-free (in the sense of Banach density).

i	$\tilde{d}_\infty \geq$	$\tilde{d} \geq$	l/n in (10)
0	1.5874	1.41421	
1	1.64195	1.53746	
2	1.69838	1.62687	
3	1.73436	1.69267	$47/48$
4	1.75794	1.72884	$86/96$
5	1.77219	1.75412	$161/192$
6	1.78064	1.76751	$310/384$
7	1.78549	1.77699	$605/768$
8	1.78824	1.78194	$1195/1536$
9	1.78977	1.78562	$2371/3072$

The reason for these good estimates is that the step from D_n to D_{n+1} only creates new 4-gons in the graph complement, but no triangles, and the 8 triangles of D_0 become more and more distant as n increases. This heuristic is explained also in the next section.

5. Determinant-volume-inequalities

5.1. Motivation and preliminaries

We will now use the spanning tree description of the determinant and the volume inequalities of Lackenby-Agol-Thurston [La] to prove our main results theorem 1.1 and 1.2. We make a few more comments on their origin. As already explained, they were motivated by Dunfield's experimental observations [Df] on the relation between determinant and volume. Evaluation of the knots with few crossings from [HT] led to evidence that for alternating links L, we have something like[1]

$$\det(L) \approx e^{a\mathrm{vol}(L)+b}, \text{ or} \tag{25}$$

$$\det(L) \approx c(L)^{a\mathrm{vol}(L)+b} \tag{26}$$

for some numbers $a > 0$ and b. (M. Khovanov suggested that such a correspondence may extend to non-alternating links if instead of the determinant we take the total degree of his generalization of the Jones polynomial [Ko].) As well-known, experimental evidence can sometimes be misleading. Thus if one wants to make a rigorous statement out of (25) and (26), which is additionally unconditional (that is, valid for all alternating links, and not just for some possibly "generic" subclass), one must take into account degenerate cases.

For example, it is easy to construct knots with bounded volume but $\det \to \infty$ (see [Br]), so that an inequality '$\leq$' in (25) instead of '$\approx$' is impossible. Similarly easy it is to see that an inequality '$\geq$' in (26) is impossible. Take for any a, b an alternating knot of sufficiently large volume (which exists e.g. from the argument of [La]). Then apply t_2^7 twists of [St5] successively at one and the same crossing. Under these moves, the determinant grows linearly, while the volume remains bounded. (In fact it converges to the volume of a certain link by Thurston's hyperbolic surgery theorem, as explained in [Br] and [La].) This in fact shows also that the inequality '$\geq$' in (25), stated in theorem 1.1, is qualitatively close to the best possible, up to at most a linear factor in $c(L)$. That is, for any $a > 1$ and any continuous function $f : \mathbb{R}_+ \to \mathbb{R}_+$, the inequality

$$\det(L) > c(L)^a \cdot f(\mathrm{vol}(L))$$

is false in general.

Contrarily, the r.h.s. of (26) as an upper bound on the determinant is valid. As I was informed subsequently, such a bound was obtained in unpublished work of Chris Leininger and Ilya Kofman, who proved

$$\det(L) \leq c(L)^{\frac{\mathrm{vol}(L)}{V_0}+2}.$$

In theorem 1.2 we claimed that we can qualitatively improve such an inequality, by putting $\mathrm{vol}(L)$ into a denominator of the base. We can make the bound more explicit, thus obtaining an improvement of the inequality of Kofman and Leininger. (They have apparently also obtained a weaker inequality in a special case of theorem 1.1.)

Theorem 5.1 For any alternating hyperbolic link L, we have

$$\det(L) \leq \left[1 + \frac{c(L)}{\max\left(2,\ 1+\frac{\mathrm{vol}(L)}{10V_0}\right)}\right]^{\frac{\mathrm{vol}(L)}{V_0}+2}.$$

The inequalities of Lackenby-Agol-Thurston are essential in establishing a link between graph theory and the volume. To explain these inequalities, we must introduce the notion of twist equivalence of crossings. The version of this relation we present here is slightly different from that of [La], and follows its independently discovered variants in [St5].

[1] Note, that Dunfield uses the expression '$\deg J$', but this presumably means what was shown to be equal to $c(L)$ in [Ka3, Mu2, Th2].

Definition 5.1 As in [St5], we call two crossings p and q of an alternating diagram D *$\sim$-equivalent*, resp. *$\tilde{*}$-equivalent*, if up to flypes they form a reverse resp. parallel clasp. We remarked that $\sim$ and $\tilde{*}$ are equivalence relations, and that if $p \sim q$ and $p \,\tilde{*}\, r$, then $p = q$ or $p = r$, so that the relation $(p \sim q \vee p \,\tilde{*}\, q)$ is also an equivalence relation. (There is the exception of D being the 2-crossing Hopf link diagram, or D having such a diagram occurring as a connected sum factor. Such D is, however, usually easy to deal with separately.) We call this relation *twist equivalence*.

Thus two crossings are twist equivalent if up to flypes they form a clasp. Using [MT, MT2], we can make the following

Definition 5.2 Let $t(D)$ be the *twist number* of an alternating diagram D, which is the number of its twist equivalence classes. For an alternating link L, let $t(L) = t(D)$ be the twist number of L, where D is some alternating diagram of L.

Our notion of twist equivalence is slightly more relaxed than what was called the same in [La], the difference being that there flypes were not allowed. (In [SV] we called this stronger equivalence for reverse clasps neighbored equivalence.) We call Lackenby's equivalence here *strong twist equivalence*. It was largely observed that by flypes all twist equivalent crossing can be made strongly twist equivalent, which Lackenby formulated as the existence of *twist reduced* diagrams. Thus we can work with twist equivalence in our sense as with twist equivalence in Lackenby's sense (or strong twist equivalence in our sense), assuming that the alternating diagram is twist reduced. With this remark, we can state the Lackenby-Agol-Thurston[1] inequalities as follows:

Theorem 5.2 (Lackenby-Agol-Thurston) For an non-trivial prime alternating link L, we have

$$10V_0\big(t(L)-1\big) \geq \mathrm{vol}\,(L) \geq V_0\big(t(L)-2\big),$$

where $V_0 = \mathrm{vol}\,(4_1)/2 \approx 1.01494$ is the volume of the ideal tetrahedron.

Since, in applying this theorem, we want to pass from D to its checkerboard graph G, it is useful to remark how twist equivalence translates from D to G. For this we use some graph terminology, most of which (like *cut vertices*, *deleting* or removing edges, etc.) is standard, or at least (like the *join* of graphs) is explained in [MS]. To save space, we do not repeat all these definitions. A few more definitions seem necessary, though.

Definition 5.3 If G is a planar graph and $e_1, \ldots, e_n$ edges in G, whose deletion disconnects G, then we call $\{e_1, \ldots, e_n\}$ an *n-cut* of G. G is *n-connected* if it has no k-cut for some $k < n$. When a planar embedding of G is fixed, then for each n-cut $\{e_1, \ldots, e_n\}$ of G one can draw a closed curve in the plane, intersecting G transversely only in (single interior points of) the edges $e_1, \ldots, e_n$. We call this loop a *cut curve* (or *loop* or *circuit*) in G. Clearly the curve of a cut determines the cut uniquely.

Note that the definition of G from D is unique only up to duality. Most of the graph properties we will deal with therefore will be symmetric w.r.t. taking the dual graph, and we can use this symmetry to simplify our arguments at some point.

The justification of the below definition follows from its direct translation from the previously considered knot diagram form, and we do not discuss it here in detail. However, note that the property G to be 2-connected and (dually) without loop edges is equivalent to D having no nugatory crossings.

Definition 5.4 Let G be a planar 2-connected graph with no loop edges. We call two edges e and f of G *twist equivalent* if (i) e and f connect the same pair of vertices, or (ii) $\{e, f\}$ is a 2-cut of G. The *twist number* $t(G)$ of G is the number of twist equivalence classes of its edges.

Remark 5.1 Again only one of the two alternatives (i) and (ii) is possible if $e \neq f$, except if e and f are the two edges of  , occurring as a join factor of G.

[1] This (Dylan) Thurston is the son of Bill Thurston, quoted before with the surgery theorem.

5.2. Proof of main results

Our main contribution to the proof of the first main result is

Theorem 5.3 For every planar 2-connected graph G with $t(G) > 0$ and no loop edges,

$$s(G) \geq 2 \cdot \gamma^{t(G)-1},$$

where $s(G)$ is the number of spanning trees of G, and $\gamma \approx 1.324718$ is the inverse of the (unique) real positive root

$$\gamma^{-1} = \sqrt[3]{\frac{25+\sqrt{621}}{54}} + \sqrt[3]{\frac{25-\sqrt{621}}{54}} - \frac{1}{3}$$

of $x^3 + x^2 - 1 = 0$.

This easily implies theorem 1.1.

Proof of theorem 1.1. If G is the checkerboard graph of an alternating diagram D of L, then $\det(L) = s(G)$ and $t(L) = t(D) = t(G)$. By the left (Agol-Thurston) inequality in theorem 5.2, we have

$$\mathrm{vol}\,(L) \leq 10V_0\big(t(D)-1\big) = 10V_0\big(t(G)-1\big) \leq 10V_0 \log_\gamma \frac{s(G)}{2},$$

hence

$$\det(L) = s(G) \geq 2 \cdot \gamma^{\frac{\mathrm{vol}\,(L)}{10V_0}},$$

and $\gamma^{1/10V_0} \approx 1.028093$. □

Remark 5.2 As will be observed, the optimal base in theorem 5.3 is $\leq \frac{1+\sqrt{5}}{2} < \gamma^2$, so that the loss of quality in the constant in theorem 1.1 is more due to the application of the Agol-Thurston inequality, rather than theorem 5.3. Even if Agol-Thurston show that their inequality is (asymptotically) sharp in general, it often fails considerably. From the 6729 prime alternating knots K of ≤ 13 crossings, 5708 (or $\approx 84.8\%$) satisfy the inequality

$$\frac{\mathrm{vol}\,(K)}{10V_0\,(t(K)-1)} < \frac{(\ln\gamma)\cdot(t(K)-1)}{\ln(\det(K)/2)},$$

whose hand-sides measure the unsharpness of both estimates. Up to 15 crossings such K are 99,910 out of 111,528 ($\approx 89.5\%$). This, however, seems natural, since the volume is more complex to understand than the graphs.

For theorem 1.2, we can use the arguments in the proof of theorem 2.4. (Since they have been repeated many times, we will not get into details; see [St, St6] for more explanation.)

Proof of theorem 1.2. Let D be a prime alternating diagram of L. W.l.o.g., assume all twist equivalent crossings are strongly twist equivalent. Let $t = t(D)$ and $n_1, \ldots, n_t$ with $\sum_{i=1}^{t} n_i = c(D)$ be the cardinalities of the twist equivalence classes of D. Each twist equivalence class forms a tangle T_i with Krebes fraction $1/n_i$ or $n_i/1$. Let the alternating diagram D' be obtained from D by replacing each T_i by a single crossing. Then $c(D') = t(D)$. By standard bracket skein module arguments, $\det(D)$ can be calculated as $\det(D')$ by counting monocyclic states, only with weights involving the n_i. Thus $\det(D)$ is a polynomial in the n_i with $\det(D')$ (different) monomials, each of which contains each n_i at most linearly. Clearly $\det(D') < 2^{t(D)}$, and the highest possible contribution of a monomial is this of $\prod_{i=1}^{t} n_i$. If we regard n_i as continuous, then this contribution is maximal if all $n_i = c(D)/t(D)$ are equal. If $t(D) > 2$, then by the Lackenby-Agol-Thurston theorem, $\mathrm{vol}\,(L)$ and $t(D) = t(L)$ are proportional, and we are done. If $t(D) = 2$, then L is one of the rational links $C(p,q)$ with $p, q \geq 2$, which are easily handled directly. From the boundedness statement in Thurston's hyperbolic surgery theorem, $\mathrm{vol}\,(C(p,q))$ are bounded above by the volume of the Borromean rings, and from the convergence statement also from below (and away from 0). Since $\det(C(p,q)) = pq + 1$, proper $C_{1,2}$ can surely be found. □

Proof of theorem 5.1. With the terminology and the argument in the previous proof, we have

$$\det(D) \leq \prod_{i=1}^{t(D)} (1+n_i)\,,$$

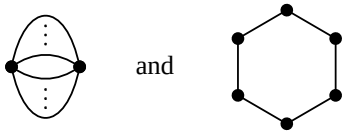
whose right hand-side is maximal again if all $n_i = c(D)/t(D)$. Then use $t(D) \geq 2$ and the Lackenby-Agol-Thurston theorem. □

Remark 5.3 It is easy to construct diagrams D where the state with weight $\prod_{i=1}^{t} n_i$ in the proof of theorem 1.2 is monocyclic, and so an improvement of theorem 5.1 beyond the removing of the additive '1' in the base of the r.h.s. is possible (with this type of argument), only if the Lackenby-Agol-Thurston inequalities are improved.

5.3. Proof of the spanning tree-twist number inequality

The proof of theorem 5.3 underlying theorem 1.1 is more substantial, and will occupy a separate subsection.

Proof of theorem 5.3. We proceed by induction on the number $e(G)$ of edges of G. The cases of $e(G) \leq 4$ are easy to verify. We additionally check the graphs G with $t(G) = 1$, namely those of the types

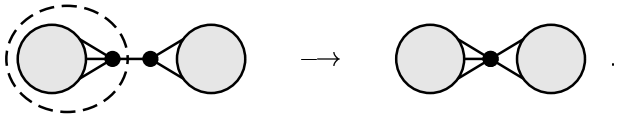


(27)

(two-vertex graph and *chain*).

Now consider the induction step. Assume G has > 4 edges and is not of the forms in (27).

In the following, for a fixed edge a in G, let G_d be G with a deleted and (subsequently) all 1-cut edges contracted:



Similarly, let G_c be G with a contracted and then all loop edges removed. Then

$$s(G) = s(G_c) + s(G_d)\,. \tag{28}$$

Excluding G from being in (27) means that both G_c and G_d are non-empty (i.e. have $t > 0$), so that induction applies on both. The aim will be now to choose a so that both $t(G_c)$ and $t(G_d)$ can be controlled from below. We make now a(n exhaustive, but not necessarily exclusive) case distinction.

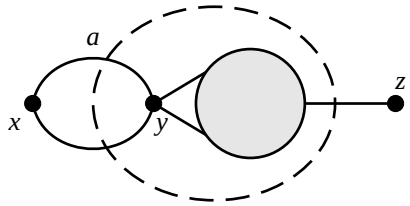
Case A. If G has a cut vertex (i.e., can be written as join $G_1 * G_2$ with $e(G_1), e(G_2) > 0$), then we are done by multiplicativity of $s(G)$ and additivity of $t(G)$ under join.

Case B. Assume G has a multiple edge. Call one of these single copies of the edge a: 

.

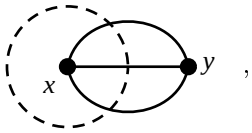
Note that the deletion of an edge never changes the number of equivalence classes under the relation given by alternative (i) in definition 5.4, while the same holds for contraction of an edge and the alternative (ii). Moreover, different twist equivalent edges with property (i) and (ii) are disjoint by remark 5.1.

Note also that, since a has a non-trivial equivalent crossing under property (i), deleting a creates no 1-cut edges (by remark 5.1). Thus $t(G_d) = t(G)$, and induction applies, unless deleting a creates a new 2-cut, i.e. a is in a 3-cut. Then we have (up to the change of the ∞-region)



. (29)

Here x and z may be connected to other vertices outside of the dotted line. Note also that a is not in an edge of multiplicity ≥ 3, because otherwise the only option of having a in a 3-cut is

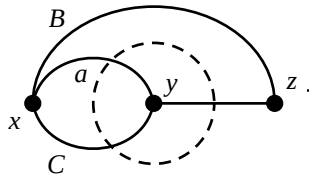


,

and $e(G) = 3$ or x or y are cut vertices, both of which we excluded.

Now consider G_c. Here particular attention is necessary to cycles of length 3. We call such cycles *triangles*, and denote them by $\triangle$. To save space, abbreviate by '$a \in \triangle$' the property 'a is contained in (at least) one triangle'. Similarly we write (for the obvious condition) '$a \in 2\triangle$' etc.

Assume now that in a graph G as in (29), $a \in \triangle$. The only possibility is



. (30)

In this case there exists exactly one $\triangle$ containing a. Therefore, in general there is at most one $\triangle \ni a$. Then $t(G_c) \geq t(G) - 2$, since contracting a we lose its twist equivalence class of edges, and at most two other twist equivalences unify to one under the contraction.

Since G has more than the 4 edges drawn in (30), and x, y and z are no cut vertices, regions B and C are different, and so the dotted line is the circle of the only 3-cut in which a participates. Then $t(G_d) \geq t(G) - 1$, since one identification of twist equivalence classes occurs when deleting a, but with the double edge the twist equivalence class of a remains in G_d (unless G has  as join factor, which we excluded). Then from (28) we have by induction

$$s(G) = s(G_d) + s(G_c) \geq 2\left(\gamma^{t(G_d)-1} + \gamma^{t(G_c)-1}\right) \geq 2\left(\gamma^{t(G)-2} + \gamma^{t(G)-3}\right) \geq 2\gamma^{t(G)-1}.$$

Case C. If G has a 2-cut, then argue using case B and duality.

Case D. Thus we can assume now that G has no 2-cut and no double edges (i.e. is simple and 3-connected).

It is well-known, that in every planar embedding of G there is a ≤ 5-gonal face E. Let a be an edge in its boundary, which we denote as ∂E.

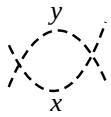
Now, for each 3-cut draw a 3-cut circuit in the plane. (We consider these circuits up to homotopy, which does not change the three edges intersected, and does not change the number of intersections of each edge.) We claim then three properties of this collection of loops.

Claim 1. No 3-cut circuit intersects the same edge twice:  . □

Claim 2. If two 3-cut circuits intersect, then they can be homotoped so that they don't.

Proof. Assume the contrary, and consider two intersecting loops a and b, homotoped so that they have the minimal possible number of intersections. When deforming a into a straight line through ∞ (and hereby temporarily ignoring the effect on G under this homotopy), $a \cup b$ becomes a meander L (see the proof of corollary 3.3; a will be the horizontal dashed line).

Assume first that $L \neq$  . The upper and lower parts of L have at least one minimal arc each (that is, arc not enclosing another one). By the exclusion of and the fact that the meander has (beside the straight line) only one loop, one of its parts must have in fact at least two minimal arcs. Thus $L = a \cup b$ contains at least three 2-gonal regions. The pairs of arcs in their boundaries are also disjoint (except possibly their endpoints). Now let

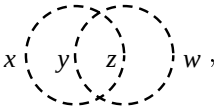


(31)

be such a 2-gon of arcs x and y, and let c_x resp. c_y be the number of intersections of x resp. y with G. By assumption $c_x, c_y \leq 3$. Since $x \cup y$ itself gives a loop and G is 3-connected, we must have $c_x + c_y \geq 3$, unless $x \cup y$ (i) does not intersect G at all, or (ii) only in the fragment of an edge. (In latter case $c_x = c_y = 1$ by claim 1.) Then x and y can be homotoped off each other (i) within a region of G or (ii) along an edge of G. Since this reduces the number of intersections of a and b, we have a contradiction to the assumed minimality of this number.

If, however, we have $c_x + c_y \geq 3$, and this scenario repeats for all (at least 3) 2-gonal regions (31), then a and b must have together at least 9 intersections with G, which contradicts their origin from 3-cuts. This argument excludes the assumption $L \neq$.

If $L =$, then

$a \cup b =$ 

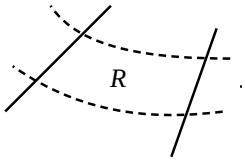
,

and we must have $d_x + d_z = d_y + d_w = 3$. If some of $d_x + d_y$, $d_y + d_z$, $d_z + d_w$, $d_w + d_x = 1$, then G is not 2-connected. If some of these numbers is 0, then a and b can be homotoped off each other within a region of G. Otherwise some of these 4 numbers is 2. Since G has no 2-cut, a pair of the 4 arcs must intersect G in the interior of an edge of G, and then these arcs are homotopable off along this edge. □

It is in fact easy to see from the proof that the types of homotopies of a and b we use to separate them can be chosen so that the total number of intersections of a and b with other loops is not augmented, so that, arguing inductively, we can assume that *there is no pair of intersecting* 3*-cut circuits.*

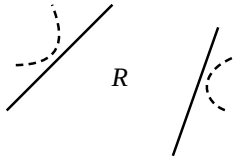
Claim 3. No two 3-cuts have a pair of common edges.

Proof. Assume there were such two 3-cuts, and pair of edges e and f, and consider the cut circuits. By 3-connectivity of G, no edge bounds the same region from both sides, and each pair of edges has at most one common region R. For e and f clearly R must exist, and then both loops must pass (non-intersectingly by claim 2) through R. By the Jordan curve theorem no circuit can leave R and enter it again (more than once). Thus we have



(32)

Modifying (32) to



gives a 2-cut circuit (with two different edges intersected, because the original two 3-cuts were different). This is a contradiction. □

Assume now each edge $a \in \partial E$ has at least 3 3-cut circuits intersecting it. We assumed that ∂E has at most 5 edges. Then it is an easy exercise to check that one cannot install the pieces of the arcs of the 3-cut circuits within E so that the conditions of all 3 claims are satisfied. More strongly, one verifies that

for each pair e and f of neighbored edges in ∂E there exists an edge $a \in \partial E \setminus \{e, f\}$ with at most two 3-cut circuits intersecting it. (33)

(Neighbored edges means edges connected to the same vertex, which is a corner of E.) In particular,

if E is a triangle, then all three edges in ∂E have at most two 3-cut circuits intersecting each. (34)

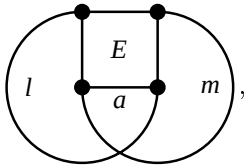
Note also that whether the interior of the triangle is empty (i.e. E is a face) or not is irrelevant for the argument.

If we choose now an edge a with most two 3-cut circuits intersecting it, similarly to case B, we have $t(G_d) \geq t(G) - 3$, since deleting a we lose its twist equivalence class, and at most two pairs of other twist equivalence classes identify. It remains to refine the choice of a so as to control also $t(G_c)$.

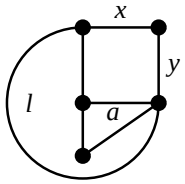
We must count again the triangles containing a. Since we assumed that G has no double edge, two such triangles intersect only in a. Assume now l is a closed loop intersecting a exactly once. Let B and C be the two regions bounded by a, and l be oriented so as to pass from B to C through a. This passage changes the inside/outside position of the curve w.r.t. any of the triangles containing a. Thus, in order to return from C to B, l must intersect at least one edge for each triangle containing a. This observation will be important below and will be referred to as the *in/out-observation*.

Case D.1. Assume each a in the boundary of a ≤ 5-gonal face with at most two 3-cuts on it has (exactly) 2 such 3-cuts. We prove that, by a proper choice, a is in at most one triangle, unless G is one single specific exception.

Namely, assume that every a in the boundary of a ≤ 5-gonal face E with two 3-cuts on it has $a \in 2\triangle$. If L is a 4- or 5-gon,

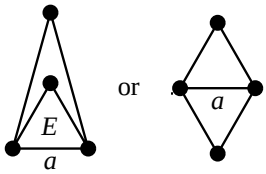


not both L and m are edges. Assume m is not an edge.



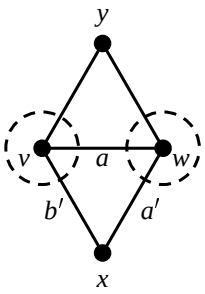
Because of claim 3, one of the two cut circuits intersecting a, call it X, must pass through x or y, and (x,a) and $(y,a) \notin \triangle$. By the in/out-observation above, this cut X must pass an edge in $\partial Y \setminus \{a\}$ for each $\triangle Y \ni a$, and all these edges are distinct. But if we assume that $a \in 2\triangle$, then X intersects at least 4 edges of G, a contradiction.

If $E = \triangle$, and $a \in 2\triangle$, then we have

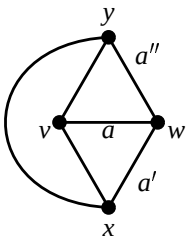


Both cases are equivalent under changing the infinite region, so consider w.l.o.g. only the second one.

Since a is in two 3-cuts, each cut must pass through one side of each of the two triangles by the in/out-observation. Then there are at most two triangles containing a, and both cut loops do not pass through any further edges. Thus the vertices v and w that a connects are trivalent, and the cut loops are going around them (i.e., their interior contains this one single vertex).



Then consider a' instead of a. The only way a' to have a second 3-cut conforming to claims 1-3 is if it goes through a' and b'. Then the same argument as for a shows that $a' \in 2\triangle$ only if x is trivalent and connected to a vertex different from w, which is connected to v. The only such vertex is y. Thus we have



(35)

Now, arguing the same way with a'' as for a, we see that y is also trivalent, and that hence G has no more edges than those 6 drawn in (35). This G has 16 spanning trees and $t(G) = 6$. The inequality in the theorem is directly verified.

Otherwise $t(G_d) \geq t(G) - 3$ and $t(G_c) \geq t(G) - 2$, and we are through by induction.

Case D.2. Assume that there is some a in the boundary of a ≤ 5-gonal face which has not exactly two 3-cuts on it. Then there is an a, which is in at most one 3-cut.

We claim that one can find an a so that (i) a is in at most 2 triangles and is in at most two 3-cuts, but (ii) not in exactly 2 for both, except if G is the graph in (35). Then either

$$t(G_d) \geq t(G) - 2 \quad \text{and} \quad t(G_c) \geq t(G) - 3\,,$$

or

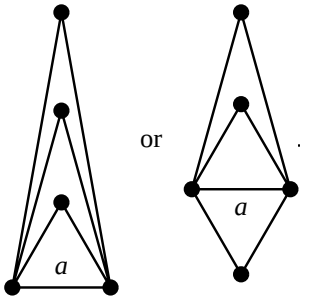
$$t(G_d) \geq t(G) - 3 \quad \text{and} \quad t(G_c) \geq t(G) - 2\,,$$

and again induction applies.

We describe an algorithm how to find some a satisfying (i). (Here the choice of the ∞ region is kept fixed!) The condition (ii) will be dealt with subsequently.

① Start with some a in the boundary of a ≤ 5-gonal fa

② If $a \in 3\triangle$, then there are always $2\triangle$ enclosing e



(36)

(There may be furhter edges in the interior of these triangles.) Note in particular that the valence of the two vertices connected by a is higher than the number of different $\triangle \ni a$.

③ $a' \neq a$ in the boundary of an innermost triangle from the ones drawn in (36). Take $a := a'$ and go to ②.

Since this iteration augments the number of $\triangle$ enclosing a (and there are only finitely many such triangles), at some point the test in ② fails, and a is i
two 3-cuts by the previous argument (34). Then continue with

④ Now both triangles of a may have non-empty int
contains a $\leq$ 5-gon X with $\partial X \not\subseteq \partial\triangle$. Let $a' \in \partial X \setminus \partial\triangle$. We can choose a' so that it has at most two 3-cuts, because $\partial X \cap \partial\triangle$ makes up at most two neighbored edges in ∂X, and we made the observation (33). Then continue with ② and $a := a'$.

⑤ By the same argument as in ②
is found.

This algorithm gives an a which is in at most two triangles, and at most two cuts. We must show now that we can avoid a being in exactly two triangles *and* exactly two cuts.

Assume a were such. Then apply the argument in case D.1. It shows that a connects two trivalent vertices. Let F_1, F_2 be the $2\triangle \ni a$. Since two of the three vertices of F_1 and F_2 are trivalent, they cannot emit edges into their interior, and since G has not cut vertex, F_1 and F_2 must have empty interior, that is, be triangular *faces*. (Actually, we must possibly replace for one of F_1 or F_2 'interior' by 'exterior' in this argument; it depends on the choice of the ∞ region.) Since all the 4 edges in $\partial F_1 \cup \partial F_2 \setminus \{a\}$ connect a trivalent vertex, they cannot belong to more than two triangles, by the argument at the end of ②. If all they belong to tw
one of these 4 edges has at most one cut, and we found the edge we sought.

Now the induction is complete. □

Remark 5.4 If G is series parallel, then case D in the induction never occurs, and (by verifying the graphs G of twist number ≤ 3), one can obtain the better inequality $s(G) \geq F_{t(G)+3}$, for $t(G) \geq 2$, which is then sharp. Compare to part 2) of theorem 2.1.

6. Some heuristics and problems

As the paper attempts the investigation of a relatively new subject, it is not surprising that it opens many more questions than it can answer. Hoping to whet the interest in further investigations, we conclude by mentioning some of these problems.

6.1. Braid index

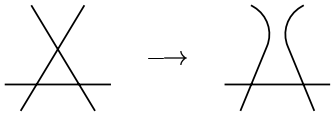
One such problem is that apparently the estimates in theorem 3.3.1) and those of the eigen values of ϕ_β are not sharp. This is related to the following conjecture:

Conjecture 6.1 Only finitely many K_n have the same braid index $b(K_n)$, or alternatively, $\liminf_{n\to\infty} b(K_n) = \infty$.

Unfortunately, solving this conjecture appears to require unimaginable effort at the present state of the art. We should, however, give some rough heuristical motivation for it (although it is far from a rigorous proof).

The reason is that the diagrams $\widehat{\beta_i}$ for braids β_i of fixed strand number have either clasps or triangle regions (4) with bounded (minimal) distance $\leq k = k_l$ between two among them, k_l depending only on the strand number l of the β_i (see [St2]; the distance is here the minimal number of intersections of a path from the one region to the other with the plane curve of the diagram).

This means that, even in the case there is no clasp, the sequence of crossings to splice can be chosen so that we splice the corners of a triangle,



after k steps we obtain a clasp. (This requires to show that there are paths of bounded length between triangles going only through quadrangles.) Therefore, letting

$$\widehat{d_n} := \max\{\det(D) : D \text{ is an } n \text{ crossing diagram obtained by splicing crossings in a } l\text{-braid diagram}\}$$

and applying

$$\widehat{d_n} \leq \widehat{d}_{n-1} + \widehat{d}_{n-2} + \widehat{d}_{n-3}$$

recursively on each summand on the right, in depth k of the recursion we can in fact use the simpler formula $\widehat{d_{n'}} \leq \widehat{d}_{n'-1} + \widehat{d}_{n'-2}$.

Thus if T_n denote the Tribonacci numbers, $\widehat{d_n} \leq \tilde{d_n}$ for a linearly recurrent sequence $\tilde{d_n}$ with

$$\tilde{d_n} = \sum_{i=1}^{k} a_i \tilde{d}_{n-i},$$

and $T_n = \sum_{i=1}^{k} a_i' T_{n-i}$ such that $0 \leq a_i \leq a_i'$ and $a_i < a_i'$ for at least one i. Writing down the generating series of T_n and $\tilde{d_n}$, the denominator polynomials are $f(x) = \sum_{i=1}^{k} a_i x^i - 1$ and $f_1(x) = \sum_{i=1}^{k} a_i' x^i - 1$ resp. On the positive real line, f and f_1 have unique zeros z_f and z_{f_1}, which are the unique zeros of minimal norm for these functions (use $a_i, a_i' \geq 0$ and apply triangle inequality).

Now $z_{f_1}^{-1} = \limsup_{n\to\infty} \sqrt[n]{T_n} = \delta$ and $z_f^{-1} = \limsup_{n\to\infty} \sqrt[n]{\tilde{d_n}}$ show the result because $f_1(x) > f(x)$ for $x > 0$, so that $z_f > z_{f_1}$.

Thus there will be a sequence $\{\delta_l\}$ with $\delta_l < \delta_{l+1} < \delta$ and $\delta_l \to \delta$ such that 'δ' in theorem 3.3.1) can be replaced by 'δ_l' for l-strand braids $\{\beta_i\}$. This would imply the conjecture, under the (again strong and hard to verify) assumption that the answer to question 2.2 is positive.

We could assure the desired switches if we always had two triangles in (one of the two choices of) the checkerboard graph to be connected by a path of bounded length passing only through 4-gons. (Any two regions have bounded distance, since they have bounded distance ot the innermost and outermost region of the braid diagram, and this remains so after any splicings.) This also explains the motivation for considering the examples in §4.

6.2. Large determinant examples

The next question concerns the examples in §4. The determinants $\det(D_i)$ can be given as follows:

i	prime factorization of $\det(D_i)$
0	2^4
1	$2^7\,3^1$
2	$2^{12}\,3^4$
3	$2^{17}\,3^1\,5^6\,7^2$
4	$2^{41}\,3^2\,5^{11}\,7^3$
5	$2^{51}\,3^{16}\,5^2\,11^6\,13^3\,23^2\,37^3\,127^3$
6	$2^{122}\,3^{10}\,5^6\,7^2\,11^3\,13^3\,17^1\,19^2\,31^2\,43^3\,421^3\,4217^3\,9661^3$
7	$2^{141}\,3^4\,7^{10}\,17^9\,769^2\,4241^3\,22391^3\,42767^3\,195863^3\,483557^2\,2072131^2\,6046751^3\,355243279^3$
8	$2^{315}\,3^4\,5^6\,7^2\,11^3\,17^8\,31^3\,47^2\,79^2\,89^3\,97^2\,157^6\,577^1\,6271^2\,20639^3\,291349^2\,1159901^3$ $1579631^3\,43863223^2\,323965910452099^3\,209443904414934601^3\,3786663141306774259^3$
9	??

(The factorization for $i = 9$ was too hard to obtain, and would be too long anyway.)

Note, that these factorizations are strikingly non-generic – the largest prime factors have only about 1/20 of the number of digits of the number, and almost all primes occur in higher powers. (The only power that can be explained so far, and still to much smaller extent than it occurs, is that of 2.)

Question 6.1 Is there a closed formula for $\det(D_i)$? Can it be used to show that $\sqrt[6\cdot 2^i]{\det(D_i)} \to \delta$?

6.3. Strongly +achiral knots and square determinants

We conclude with some remarks around the fact that, by writing out the endomorphisms ϕ_β as matrices, for appropriate β we obtain squareness properties for some linear combinations of entries of such matrices.

Example 6.1 Consider the matrix

$$A = \begin{pmatrix} 1 & 18 & 18 & 24 & 12 \\ 0 & 13 & 0 & 18 & 0 \\ 0 & 0 & 25 & 0 & 18 \\ 0 & 18 & 0 & 25 & 0 \\ 0 & 0 & 18 & 0 & 13 \end{pmatrix}.$$

Then, writing $A^k = \left(a_{i,j}^{(k)}\right)_{i,j=1}^5$, we have that $a_{1,4}^{(2k+1)} + a_{1,5}^{(2k+1)}$ is always a square. This follows again from [HK], as A^T represents the endomorphism ϕ_β for $\beta = \sigma_1\sigma_2^{-2}\sigma_1\sigma_2^{-1}\sigma_1^2\sigma_2^{-1}$ in the basis (23) of DS_3. Interestingly again $a_{1,4}^{(2k)} + a_{1,5}^{(2k)}$ is always of the form $10x^2$, although there is no knot-theoretical explanation of this fact.

The last example, together with some further experiments, leads to the following conjecture.

Conjecture 6.2 If $\beta' \in B_3$ is an alternating braid, and $\beta = \beta'\overline{\beta'}$, then ϕ_β Jordan-decomposes over the quadratic number field $\mathbb{Q}(\sqrt{d})$, or at least over $\mathbb{Q}(\sqrt{d}, i)$, where $d = \det(\widehat{\beta^2})$, and $\sqrt{\det(\widehat{\beta^{2k}})/d} \in \mathbb{Z}$ for all $k > 0$.

As TL_n has an antiautomorphism (turn around by 180°), for $\beta = \beta'\overline{\beta'}$, ϕ_β is conjugate to its inverse, so that the characteristic polynomial $\chi(\phi_\beta)$ of ϕ_β is self-conjugate, i.e., $\chi(\phi_\beta)(x) \doteq \chi(\phi_\beta)(x^{-1})$ (where $\doteq$ denotes equality up to units in $\mathbb{Z}[x, x^{-1}]$). However, $\chi(\phi_\beta)$ turns out to have (at least in all cases calculated in an experiment) some unexpected properties.

For 3-braids the polynomial $\chi(\phi_\beta)$ had the form $(x-1)P(x)^2$ with a quadratic polynomial P, and in fact ϕ_β decomposes into $Id_1 \oplus \phi'_\beta \oplus \phi'_\beta$ (where Id_1 is the 1-dimensional identity map) under a certain, but not plausible, choice of basis.

For 5-braids $\chi(\phi_\beta) = (x-1)^6 P_1(x)^5 P_2(x)^4$ with $P_{1,2}$ being self-conjugate polynomials of degree 4 with alternating coefficients ($[P_i]_{x^j} \cdot [P_i]_{x^{j+1}} < 0$ for $0 \le j < 4$), which additionally seem related, as always $[P_1]_x + [P_2]_{x^2} = +2$. For example, for

$$\beta' = \sigma_3^{-1}\sigma_4\sigma_1^{-1}\sigma_2\sigma_4\sigma_1^{-1}\sigma_3^{-1}\sigma_2\sigma_4\sigma_3^{-1}$$

(and $\beta = \beta'\overline{\beta'}$) we have

$$\chi(\phi_\beta) = (x-1)^6\,(1 - 26\,166x + 2\,297\,755x^2 - 26\,166x^3 + x^4)^5\,(1 - 1533x + 26\,168x^2 - 1533x + x^4)^4\,.$$

It is interesting to see what phenomena occur for more strands, but for 7-braids the dimension of TL_7 is 429, and this renders experiments rather difficult.

These phenomena motivate and merit some possible further investigations in the future.

Acknowledgements. I would like to thank to G. Cornelissen, C. Leininger, K. Rebman and D. Zagier for some helpful remarks and discussions. Also, N. Dunfield and M. Khovanov provided some key references.

References

[Al] J. W. Alexander, *Topological invariants of knots and links*, Trans. Amer. Math. Soc. **30** (1928), 275–306.

[BGRT] D. Bar-Natan, S. Garoufalidis, L. Rozansky and D. P. Thurston, *Wheels, Wheeling, and the Kontsevich Integral of the Unknot*, Israel J. Math. **119** (2000), 217–237, see also `q-alg/9703025`.

[BLM] R. D. Brandt, W. B. R. Lickorish and K. Millett, *A polynomial invariant for unoriented knots and links*, Inv. Math. **74** (1986), 563–573.

[Br] M. Brittenham, *Bounding canonical genus bounds volume*, preprint (1998), available at `http://www.math.unl.edu/~mbritten/personal/pprdescr.html`.

[Ch] J. H. E. Cohn, *On square Fibonacci numbers*, J. London Math. Soc. **39** (1964), 537–540.

[Co] J. H. Conway, *On enumeration of knots and links*, in "Computational Problems in abstract algebra" (J. Leech, ed.), 329-358. Pergamon Press, 1969.

[DH] O. T. Dasbach and S. Hougardy, *A conjecture of Kauffman on amphicheiral alternating knots*, J. Knot Theory Ramifications **5(5)** (1996), 629–635.

[DGG] P. Di Francesco, O. Golinelli and E. Guitter, *Meanders and the Temperley-Lieb algebra*, Comm. Math. Phys. **186(1)** (1997), 1–59.

[DF] A. Di Porto and P. Filipponi, *Some special triangular numbers, and recurring sequences*, Notes Number Theory Discrete Math. **1(1)** (1995), 11–26.

[Du] A. Dujella, *Diophantine quadruples for squares of Fibonacci and Lucas numbers*, Portugal. Math. **52(3)** (1995), 305–318.

[Du2] ——— " ———, *Generalized Fibonacci numbers and the problem of Diophantus*, Fibonacci Quart. **34(2)** (1996), 164–175.

[Df] N. Dunfield, *An interesting relationship between the Jones polynomial and hyperbolic volume*, web document `http://abel.math.harvard.edu/~nathand/dylan`.

[Es] A. Eswarathasan, *On square pseudo-Lucas numbers*, Canad. Math. Bull. **21(3)** (1978), 297–303.

[Ga] D. Gabai, *Foliations and genera of links*, Topology **23** (1984), 381–394.

[GS] F. González-Acuña and H. Short, *Cyclic branched coverings of knots and homology spheres*, Rev. Mat. Univ. Complut. Madrid **4(1)** (1991), 97–120.

[Go] C. McA. Gordon, *Knots whose branched cyclic coverings have periodic homology*, Trans. Amer. Math. Soc. **168** (1972), 357–370.

[GL] ——— " ———, and R. A. Litherland, *On the signature of a link*, Invent. Math. **47(1)** (1978), 53–69.

[HW] G. H. Hardy and E. M. Wright, *Einführung in die Zahlentheorie* (German), R. Oldenbourg, Munich, 1958. (3rd edition)

[HK] R. Hartley and A. Kawauchi, *Polynomials of amphicheiral knots*, Math. Ann. **243(1)** (1979), 63–70.

[H] P. Freyd, J. Hoste, W. B. R. Lickorish, K. Millett, A. Ocneanu and D. Yetter, *A new polynomial invariant of knots and links*, Bull. Amer. Math. Soc. **12** (1985), 239–246.

[HT] J. Hoste and M. Thistlethwaite, *KnotScape*, a knot polynomial calculation and table access program, available at `http://www.math.utk.edu/~morwen`.

[HTW] ——— ” ———, ——— ” ——— and J. Weeks, *The first 1,701,936 knots*, Math. Intell. **20 (4)** (1998), 33–48.

[J] V. F. R. Jones, *A polynomial invariant of knots and links via von Neumann algebras*, Bull. Amer. Math. Soc. **12** (1985), 103–111.

[J2] ——— ” ———, *Hecke algebra representations of of braid groups and link polynomials*, Ann. of Math. **126** (1987), 335–388.

[JP] ——— ” ——— and J. H. Przytycki, *Lissajous knots and billiard knots*, “Knot theory” (Warsaw, 1995), Banach Center Publ. **42**, Polish Acad. Sci., Warsaw, 1998, 145–163.

[Ju] P. Jüde, *85 nützliche & dekorative Knoten*, Weltbild Verlag, Augsburg, 1998.

[Ka] L. H. Kauffman, *An invariant of regular isotopy*, Trans. Amer. Math. Soc. **318** (1990), 417–471.

[Ka2] ——— ” ———, *Quantum invariants of knots, links and three-manifolds*, talks given at the knot theory workshop “Journées Toulousaines autour des tresses et des nœuds” in Toulouse, France, June 2000.

[Ka3] ——— ” ———, *State models and the Jones polynomial*, Topology **26** (1987), 395–407.

[K] A. K. Kelmans, *Graphs with an extremal number of spanning trees*, J. Graph Theory **4(1)** (1980), 119–122.

[K2] ——— ” ——— , *Erratum: ”A certain polynomial of a graph and graphs with an extremal number of trees”, by the author and V. M. Chelnokov*, J. Combinatorial Theory Ser. B **24(3)** (1978), 375.

[KC] ——— ” ——— and V. M. Chelnokov, *A certain polynomial of a graph and graphs with an extremal number of trees*, J. Combinatorial Theory Ser. B **16** (1974), 197–214.

[Ko] M. Khovanov, *A categorification of the Jones polynomial*, Duke Math. J. **101(3)** (2000), 359–426.

[Ki] M. Kidwell, *On the degree of the Brandt-Lickorish-Millett-Ho polynomial of a link*, Proc. Amer. Math. Soc. **100** (1987), 755–761.

[Kr] D. Krebes, *An obstruction to embedding 4-tangles in links*, Jour. of Knot Theory and its Ramifications **8(3)** (1999), 321–352.

[La] M. Lackenby, *The volume of hyperbolic alternating link complements*, with an appendix by I. Agol and D. Thurston, to appear in Proc. London Math. Soc.

[LM] W. B. R. Lickorish and K. C. Millett, *A polynomial invariant for oriented links*, Topology **26(1)** (1987), 107–141.

[Lo] D. D. Long, *Strongly plus-amphicheiral knots are algebraically slice*, Math. Proc. Cambridge Philos. Soc. **95(2)** (1984), 309–312.

[MD] W. L. McDaniel, *Diophantine representation of Lucas sequences*, Fibonacci Quart. **33(1)** (1995), 59–63.

[Me] W. W. Menasco, *Closed incompressible surfaces in alternating knot and link complements*, Topology **23(1)** (1986), 37–44.

[MT] ——— ” ——— and M. B. Thistlethwaite, *The Tait flyping conjecture*, Bull. Amer. Math. Soc. **25 (2)** (1991), 403–412.

[MT2] ——— ” ——— and ——— ” ——— , *The classification of alternating links*, Ann. of Math. **138(1)** (1993), 113–171.

[Mr] J. Morgado, *Note on the Chebyshev polynomials and applications to the Fibonacci numbers*, Portugal. Math. **52(3)** (1995), 363–378.

[Mu] K. Murasugi, *On periodic knots*, Comment. Math. Helv. **46** (1971), 162–174.

[Mu2] ——— ” ———, *Jones polynomial and classical conjectures in knot theory*, Topology **26** (1987), 187–194.

[MS] ——— ” ——— and A. Stoimenow, *The Alexander polynomial of planar even valence graphs*, accepted by Adv. Appl. Math.

[My] B. R. Myers, *On spanning trees, weighted compositions, Fibonacci numbers, and resistor networks*, SIAM Rev. **17** (1975), 465–474.

[My2] ——— ” ———, *Number of spanning trees in a wheel*, IEEE Trans Circuit Theory, **18** (1971), 280–282.

[Re] K. R. Rebman, *The sequence: 1 5 16 45 121 320 . . . in combinatorics*, Fib. Quart. **13** (1975), 51–55.

[Ri] R. Riley, *Growth of order of homology of cyclic branched covers of knots*, Bull. London Math. Soc. **22(3)** (1990), 287–297.

[Ro] D. Rolfsen, *Knots and links*, Publish or Perish, 1976.

[Sl] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, accessible at the e-mail address `sequences@research.att.com`.

[St] A. Stoimenow, *Square numbers, spanning trees and invariants of achiral knots*, preprint `math.GT/0003172`.

[St2] ——— ” ———, *The braid index and the growth of Vassiliev invariants*, J. Of Knot Theory and Its Ram. **8(6)** (1999), 799–813.

[St3] ——— ” ———, *Gauß sum invariants, Vassiliev invariants and braiding sequences*, J. Of Knot Theory and Its Ram. **9(2)** (2000), 221–269.

[St4] ——— ” ———, *On the coefficients of the link polynomials*, to appear in Manuscripta Mathematica.

[St5] ——— ” ———, *Knots of genus one*, Proc. Amer. Math. Soc. **129(7)** (2001), 2141–2156.

[St6] ——— ” ———, *Jones polynomial, genus and weak genus of a knot*, Ann. Fac. Sci. Toulouse **VIII(4)** (1999), 677–693.

[SV] ——— ” ——— and A. Vdovina, *Counting alternating knots by genus*, preprint.

[Th] M. B. Thistlethwaite, *On the Kauffman polynomial of an adequate link*, Invent. Math. **93(2)** (1988), 285–296.

[Th2] ——— ” ———, *A spanning tree expansion for the Jones polynomial*, Topology **26** (1987), 297–309.

[We] H. Wendt, *Die Gordische Auflösung von Knoten*, Math. Z. **42** (1937), 680–696.

[Wo] S. Wolfram, *Mathematica — a system for doing mathematics by computer*, Addison-Wesley, 1989.

WHY DELANNOY'S NUMBERS?

CYRIL BANDERIER AND SYLVIANE SCHWER

Abstract. We present here a survey of most notable Delannoy's works. These works are related to lattice paths enumeration, to the so-called Delannoy numbers, and were the first general way to solve Ballot-like problems. We also give a tentative short biography.

This version corresponds to an update (May 2002) of the abstract submitted (February 2002) by the first author to the 5th lattice path combinatorics and discrete distributions conference (Athens, June 5-7, 2002).

1. Classical lattice paths

A good reference for classical results on lattice paths is [57]; more recent works show that lattice paths are still a subject of an intensive activity in combinatorics [39, 41, 48, 56, 74, 10, 61, 35, 46, 6] and in probability theory [8, 37].

Most of the classical number sequences or lattice paths have a name which is related to some famous mathematician such as the Italian Leonardo Fibonacci ($\sim$1170–$\sim$1250), the French Blaise Pascal (1623–1662), the Swiss Jacob Bernoulli (1654–1705), the Scottish James Stirling (1692–1770), the Swiss Leonhard Euler (1707–1783), the Belgian Eugène Catalan (1814–1894), the German Ernst Schröder (1841–1902), the German Walther von Dyck (1856–1934), the Polish Jan Łukasiewicz (1878–1956), the American Eric Temple Bell (1883–1960), the American Theodore Motzkin (1908-1970), the Indian Tadepalli Venkata Narayana (1930–1987), ... A good indicator of their celebrity is that almost all of them have a biographical entrance in the MacTutor History of Mathematics archive[1]. For some of them, it is quite amusing that they are nowadays more famous in combinatorics for problems which can be explained in terms of lattice paths than in their original field (algebra or logic for Dyck, Schröder, and Łukasiewicz).

Fibonacci numbers appear in his *Liber abbaci* (1202, we believe there is no actual printed edition of it). "Catalan numbers" can be found in various works, including [14, 71]. Catalan called these numbers "Segner numbers"; and the actual terminology is due to Netto who wrote the first classical introduction to combinatorics [60]. The name "Schröder numbers" honors the seminal paper [68] and can be found in Comtet's "Analyse combinatoire" [18] and also into one of his article published in 1970. The name "Motzkin numbers" can be found in [34] and is related to Motzkin's article [58]. The name "Narayana numbers" was coinded by Kreweras by reference to the article [59] (these numbers were also independantly studied by John P. Runyon, a colleague of Riordan. There are called Runyon numbers in Riordan's book [65], p.17). The name "Dyck paths" comes from the more usual "Dyck words/Dyck Language" which are widely used for more than fifty years. We

Date: June 10, 2002.

[1] A web-site from the University of St Andrews: `http://www-groups.dcs.st-and.ac.uk/~history/`

strongly recommend the lecture of R. Stanley, who give some comments about the (surprising old) origin of these names and problems (cf pp. 212–213 of [72]).

2. Delannoy numbers

Delannoy is another "famous" name which is associated to an integer sequence related to lattice paths enumeration. Delannoy's numbers indeed correspond to the sequence $(D_{n,k})_{n,k\in\mathbb{N}}$, the number of walks from $(0,0)$ to (n,k), with jumps $(0,1)$, $(1,1)$, or $(1,0)$.

1	21	221	1561	8361	36365	134245	433905	1256465	3317445	**8097453**
1	19	181	1159	5641	22363	75517	224143	598417	**1462563**	3317445
1	17	145	833	3649	13073	40081	108545	**265729**	598417	1256465
1	15	113	575	2241	7183	19825	**48639**	108545	224143	433905
1	13	85	377	1289	3653	**8989**	19825	40081	75517	134245
1	11	61	231	681	**1683**	3653	7183	13073	22363	36365
1	9	41	129	**321**	681	1289	2241	3649	5641	8361
1	7	25	**63**	129	231	377	575	833	1159	1561
1	5	**13**	25	41	61	85	113	145	181	221
1	**3**	5	7	9	11	13	15	17	19	21
1	1	1	1	1	1	1	1	1	1	1

In this array, the lower left entry is $D_{0,0} = 1$ and the upper right entry is $D_{10,10} = 8097453$. Entry with coordinates (n,k) gives the number of Delannoy walks from $(0,0)$ to (n,k). The three steps $(0,1)$, $(1,1)$, and $(1,0)$ being respectively encoded by x, y and xy, the generating function of Delannoy walks is

$$F(x,y,t) = \sum_{n\geq 0} (x+y+xy)^n t^n = \frac{1}{1-t(x+y+xy)}\,,$$

where t encodes the length (number of jumps) of the walk.

The *central* Delannoy numbers $D_{n,n}$ (`EIS 1850`[2]) are in bold in the above array. They made surface for several problems: properties of lattice and posets, number of domino tilings of the Aztec diamond of order n augmented by an additional row of length $2n$ in the middle, edition distance in pattern matching ... The generating function of the central Delannoy numbers is

$$\begin{aligned} D(z) &:= \sum_{n\geq 0} D_{n,n} z^n = [x^0]\,\frac{1}{1-(zx+z/x+z)} = \frac{1}{\sqrt{1-6z+z^2}} \\ &= 1+3\,z+13\,z^2+63\,z^3+321\,z^4+1683\,z^5+8989\,z^6+48639\,z^7+O(z^8)\,. \end{aligned}$$

The notation $[x^n]F(x)$ stands for the coefficient of x^n in the Taylor expansion of $F(x)$ at $x=0$. The square-root expression is obtained by a resultant or a residue computation (this is classical for diagonal of rational generating functions). This closed form for $D(z)$ gives, by singularity analysis:

$$\begin{aligned} D_{n,n} &= \frac{(3+2\sqrt{2})^n}{\sqrt{\pi}\sqrt{3\sqrt{2}-4}}\left(\frac{n^{-1/2}}{2}+\frac{23\,n^{-3/2}}{32(8+3\sqrt{2})}+\frac{2401\,n^{-5/2}}{2048(113+72\sqrt{2})}+O(n^{-7/2})\right) \\ &\approx 5.82842709^n\left(.57268163\,n^{-1/2}+.06724283\,n^{-3/2}+.00625063\,n^{-5/2}+\dots\right). \end{aligned}$$

[2] This refers to the wonderful On-Line Encyclopedia of Integer Sequences `http://www.research.att.com/~njas/sequences/`

One has also $D_{n,k} = \sum_{i=0}^{n} \binom{n}{i}\binom{k}{i}2^i$. Quite often, people note that there is a link between Legendre polynomials and Delannoy numbers, and indeed $D_{n,n} = P_n(3)$, but this is not a very relevant link as there is no combinatorial correspondence between Legendre polynomials and our lattice paths (this is simply a consequence that the space of "low order recurrences" is quite small for all our combinatorial objects, so there are sometimes some fortuitous coincidences).

It is classical in probability theory [8] and more precisely in the theory of Brownian motion to consider the following constraints for lattice paths:

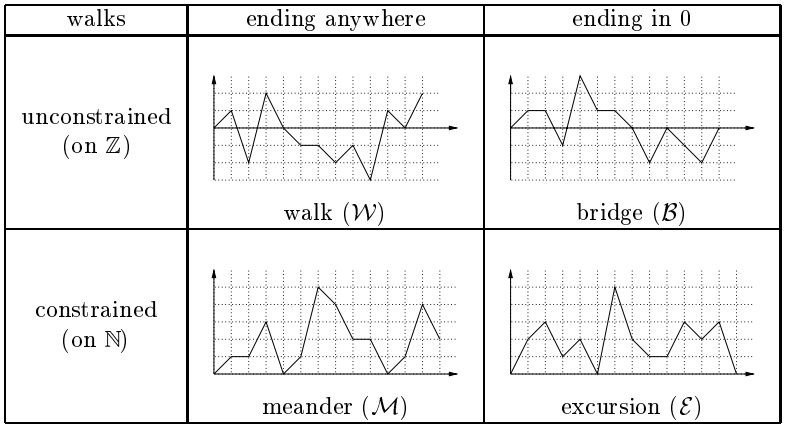


FIGURE 1. The four types of paths: walks, bridges, meanders, and excursions.

For these four kinds of walks and for any finite set of jumps, there exists a nice formula for the corresponding generating function, which appears to be algebraic, and from which one can derive the asymptotics and limit laws for several parameters of the lattice paths (see [6]).

Delannoy numbers $D_{n,n}$ correspond to bridges with a set of jumps $\{+1, -1, 0\}$ (where the 0 jump is in fact of length 2). Consider now the language $\mathcal{D}$ of central Delannoy paths, encoded via the letters a, b, c (for the jumps $+1, -1, +0$ resp.). Excursions with these jumps are called Schröder paths. We note $\mathcal{S}$ the language of Schröder paths (excursions) and $\bar{\mathcal{S}}$ the set of their mirror with respect to the x-axis. Then, the natural combinatorial decomposition $\mathcal{D} = (c^* a \mathcal{S} b + c^* b \bar{\mathcal{S}} a)^* c^*$ (which means that one sees a Delannoy path [bridge] as a sequence of Schröder paths [excursions] above or below the x-axis) leads to

$$D(z^2) = \frac{1}{1 - 2z^2 S(z)\frac{1}{1-z^2}} \frac{1}{1-z^2} = \frac{1}{1 - z^2(1 + 2S(z))}, S(z) = \frac{1 - z^2 - \sqrt{1 - 6z^2 + z^4}}{2z^2}$$

where $S(z)$ is the generating function of Schröder paths. This link between excursions and bridges is always easy to express when the set of jumps is symmetric or with jumps of amplitude at most 1, but there is also a relation between excursions and bridges in a more general case (see [6] for combinatorial and analytical proofs).

Despite all these apparitions of Delannoy numbers, the classical books in combinatorics or computer science which are usually accurate for "redde Caesari quae sunt Caesaris " (e.g., Comtet, Stanley, Knuth) are mute about this mysterious Delannoy.

FIGURE 2. Henri Auguste Delannoy (1833-1915). Portrait provided by the Société des Sciences Naturelles et Archéologiques de la Creuse, where it is exhibited.

3. Henri Auguste Delannoy (1833-1915)

The question "why Delannoy's numbers?" has also been raised in the "Domino mailing list", in 1994 and 2001, by people like J. Propp (who administrates this mailing list) or G. Kuperberg, as Delannoy numbers are also related to domino-tilings of the augmented Aztec diamond. Some people suggested that "Delannoy" was related either to the French mathematician Charles Delaunay (like in Delaunay triangulations) or to the Russian mathematician Boris Nikolaevich Delone, but this is not the case, as we shall see.

The first author's interest to Delannoy numbers comes from a talk, that Marko Petkovšek gave in the Algorithms Seminars, at INRIA in 1999 (a summary of this talk can be found in [5]). As an example, he was dealing with chess king moves (his general result about the nature of different multidimensional recurrences can be found in the article [10]). M. Petkovšek asked the first author what he knew about "Delannoy", as it sounds like a French family name `[dvlanoa]` in approximative phonetic alphabet (`v` like in "duck" and `a` in "have"). There are in fact thousands of Delannoy, mostly in the North of France and in Belgium. This toponym means "de Lannoy", that is to say who originates from the town of Lannoy; "lannoy" meaning a place with a lot of alders. But how to find "our" Delannoy amongst all these homonyms? The terminology "Delannoy numbers" became widely used as it can be found in Comtet's book in the footnote from exercise 20, p.93 [18]: "these numbers are often called Delannoy numbers" without any reference. In the English edition "Advanced Combinatorics" [19], the footnote becomes inserted in the text (p.81) but there is still no reference.

The second author's interest to Delannoy numbers comes from her own works. As a researcher in Temporal Representation and Reasoning, she developed a model based on formal languages theory [69] instead of the logical or relational approaches. This framework allowed her to enumerate easily all possible temporal relations between n independent events-chronologies. Then she tried to know if some of these sequences where already known. In the $n = 2$ case, Sloane's On-Line EIS provided her the name of Delannoy. She asked everybody she met who is Delannoy?

We then found (thanks to Philippe Flajolet for the first author and thanks to her father-in-law for the second author) further informations in several books by Lucas (see [20] for some biographical informations on Édouard Lucas [1842-1891]). Indeed, in the second edition of the first volume of the *Récréations mathématiques* [50], Lucas wrote in the preface "*J'adresse mes plus vifs remerciements à mon ami sincère et dévoué, Henry*[3] *Delannoy, ...* " and at page 13 of this introduction we are told that Henri Delannoy was intendant. In the second volume [55], the fourth recreation was dedicated to *Monsieur Henri Delannoy, ancien élève de l'École Polytechnique, sous-intendant militaire de Première classe.* After the death of his friend Lucas in October 1891, Henri Delannoy contributed with Lemoine and Laisant[4] to the publication of the third and fourth volumes [52, 53] as well as to the book[5] *L'arithmétique amusante* [54].

Like most of the French military intendants, Delannoy was a student from the École Polytechnique (which was the place where military officers received a scientific education). From a database[6] of the former students, one knows that Henri (Auguste) Delannoy is born in 1833 in Bourbonne-les-Bains (Haute-Marne, France). His father was Omère Benjamin Joseph Delannoy (countable officer) and his mother was Françoise Delage; they were living in the city of Bourges. In 1853, he passed the École Polytechnique entrance exam (with rank 62); then he graduated in 1854 with rank 91/106 and finished with rank 67/94 in 1855. It is quite funny that this database also contains (like for any other polytechnician) a physical description of Henri Delannoy: dark brown hair, average brow, average nose, blue eyes, small mouth, round chin, round face, height: 1,68m!

In the archive center of the French Army (in the Château de Vincennes), one can find his record under the number 61241. From this and [1, 4], we know that Delannoy was first in the Artillery corps as sous-lieutenant (with rank 12/37 from the application school), lieutenant (1857), took part to the Italy campaign (27 May- 18 August 1859) and to the Solférino battle (24 June 1859). When he came back, he married his dulcinea Olympe-Marguerite Guillon the 10 November 1859. They had 2 daughters and one boy. Delannoy was promoted captain in 1863. He then became a supplier-administrator: Intendant-Adjoint in 1865, sous-intendant of third classe in 1867, of second classe in 1872, of first class in 1882 (he was yet a widower). He spent three years in Africa (6 Oct. 1866 - 25 Oct. 1869). He was the governor of the military Hospital of Sidi-bel-Abes (Algeria) during the terrible typhus epidemic (he was belonging to the Supply Corps, and they have in charge the sanitary affairs). He translated for himself and perhaps also for his hierarchy

[3] There is no mistake, he was born Henry and asked to change it into Henri. We use in this article the first name Henri, as it was Delannoy's choice and as this was officially approved.

[4] The rôle played by each one will be explained in a forthcoming publication.

[5] Some of these books are available at the web site of the French National Library `http://gallica.bnf.fr/`

[6] Available at `http://bibli.polytechnique.fr:4505/ALEPH0/`

several German books/notes about Supply Corps. He took part in the 1870 war between France and Prussia. It is mentioned without explanation that he was in *Deutschland* on July 26, 1870 (that is, 4 days after the declaration of war ...) and on March 7, 1871 (that is, 3 days before the treaty of London ...). He was decorated with the *médaille d'Italie*, the *décoration sarde de la Valeur militaire*, the *Croix de la Légion d'Honneur* on July 18, 1868, and the *Rosette d'Officier de la Légion d'Honneur* in December 20, 1886. He could have reached the highest military ranks, but he wanted peace and decided to retire (January 9, 1889) in Guéret (the main city of the French department "La Creuse"), beginning a second life dedicated to science and more particularly to mathematics.

Looking for `Delannoy*` in the Zentralblatt volumes of 1860–1920[7], gives the following nine articles [23, 24, 26, 25, 28, 31, 30, 32, 33]. The references, problems, methods, and solutions used in these articles are similar to the ones often mentioned in Lucas' books. Most of Delannoy's articles are signed by Monsieur (H.) Delannoy, military intendant in the city of Orléans (and later, retired military intendant in the city of Guéret). Delannoy was a quite active member of the French Mathematical Society (SMF) in which he was admitted in 1882, introduced by Lucas and Laisant. He disappeared from the SMF's list in 1905, while he still contributed to l'Intermédiaire des Mathématiciens until 1910. This amateur mathematician then sank into oblivion and we found no obituary in the SMF bulletins at the occasion of his death (February 5, 1915).

In his death certificate, he was referred as president of the *Société des Sciences Naturelles et Archéologiques de la Creuse*. This Society is in fact still very dynamic[8]. It eventually appears that this Society, which was presided by Delannoy from 1896 to 1915, has some archives, a part of which was given by Delannoy's family. They include some biographies written when Delannoy was still alive [7, 1], a list of his publications, and also an obituary and a short biography by members of the Society [4, 12]. We shall come back on Delannoy's works in this Society in Section 5 and we know consider Delannoy's contribution to mathematics.

4. Delannoy's mathematical work

Delannoy began his mathematical life reading the mathematical recreations that Lucas began to publish in 1879 in *La Revue Scientifique*. He was in contact with him in 1880 and began immediately to work with him, answering to letters of mathematicians transmitted by Lucas.

The first mention, in a mathematical work, to Delannoy is in an article by Lucas "Figurative arithmetics and permutations (1883)" [49], which deals with enumeration of configurations of 8 queens-like problems (the simplest one being: how to place n tokens on an $n \times n$ array, with no row or column with 2 tokens). Delannoy is there credited for having computed several sequences.

Some years later, in 1886, Delannoy made his first mathematical public intervention in the annual meeting of the "Association Française pour l'Avancements des Sciences". We know give the list of Delannoy's article.

4.1. **Using a chessboard to solve arithmetical problems(1886)** [22]**.** In this article, Delannoy comes back on Lucas' article mentioned above and explains how he

[7] The "Jahrbuch über die Fortschritte der Mathematik" is available at `http://www.emis.de/MATH/JFM/`

[8] `htpp://perso.wanadoo.fr/jp-l/SSC23/`

can use a "chessboard" (in modern words: an array) to get the formula $\frac{n-k+1}{n+1}\binom{n}{n+k}$ for the number of Dyck paths of length n ending at altitude k by using something which is not far from what one calls now the Desiré André reflection principle, which was in fact published one year later [3].

For this, he makes the link $T_{x,y} = \binom{x}{x+y} - \binom{x-1}{x+y} = \frac{y-x+1}{y+1}\binom{x}{x+y}$ between entries from the rectangular array (our walks on $\mathbb{Z}$, here given by the binomial coefficients) and entries from the triangular array (walks constrained to remain in the upper plane, our Dyck paths). The numbers $T_{x,y}$ are called (in English) "ballot numbers", but they are also called Delannoy-Segner numbers in Albert Sade's review (in the Mathematical Reviews) of Touchard's article [75]. Kreweras [47] and Penaud [63] follow this terminology (quoting Riordan or Errera [36] but none of Delannoy's articles which all sank into oblivion). We have for yet not been able to get Errera's memoir. In conclusion, these "Delannoy-(Segner)" numbers $T_{x,y}$ are *not* the "famous" Delannoy numbers $D_{n,k}$ defined in Section 2.

4.2. **The length of the game (1888)** [23]**.** There are several contributions of Rouché and Bertrand in the Comptes de l'Académie des Sciences on the following problem, that they call *the game*: "two players have n francs and play a game, at each round, the winner gets one franc from his opponent. One stops when one of the two players is ruined." When the game is fair, the probability to be ruined at the beginning of the round m is (with $q = \frac{m-n}{2}$):

$$\frac{(-1)^{m-n}}{n}\sum_{k=1}^{n}(-1)^{k-1}\sin\left(\frac{(2k-1)\pi}{2n}\right)\cos^{m-1}\left(\frac{(2k-1)\pi}{2n}\right)$$

$$= \frac{n}{2^{m-1}}\sum_{k=0}^{q/n}(-1)^k\frac{2k+1}{\frac{m+n}{2}+kn}\binom{m-1}{q-kn}.$$

Rouché proves the left hand part with some determinants computations and Delannoy uses lattice paths to get the right hand part (claiming justly that there was a mistake in Rouché's first formula).

One can see this problem as a Dyck walk in the strip $[-n, n]$, that's why the formula is similar to the formula 14 from [21] in their enumeration of planted plane trees of bounded height (Feller [38] gives also some comments on this).

4.3. **How to use a chessboard to solve various probability theory problems (1889)** [24]**.** This is a potpourri of 7 (ballot-like or ruin-like) problems (partially) solved by de Moivre, Laplace, Huyghens, Ampère, Rouché, Bertrand, André, ... for which Delannoy presents his simple solutions, obtained by his lattice paths enumeration method. He calls the lattice "chessboard". The different constraints corresponds to different kind of chessboards: triangular for walks in the upper-plane, rectangular for unconstrained walks, pentagonal for walks bounded from above, hexagonal for walks in a strip (modern authors from statistical physics sometimes talk about walks with a wall or two walls [46]). Delannoy numbers (and the two corresponding binomial formulae) appear at page 51. Delannoy says that it corresponds to the directed walk of a queen (sic), and that this problem was suggested to him by Laisant. This (and the further advertisement by Lucas of Delannoy's works, see e.g. p. 174 of [51] on "Delannoy's arithmetical square", which is exactly the array given in Section 2) answers to the question raised in our title.

4.4. **Various problems about the game (1890)** [26]**.** Using an enumeration argument, simplifying the sum that he obtained and then using the Stirling formula, he gives the asymptotic result $\frac{1}{\sqrt{2\pi}}\sqrt{2n}$ as the difference between the number of won and lost games, after $2n$ games. He also answers to other problems, e.g. what is the probability to have a group of $2, 3, \ldots, 8$ cards of the same color in a packet of 32 cards.

4.5. **Formulae related the binomial coefficients (1890)** [25]**.** He gives several binomial formulae, such as $\sum_{k=0}^{p}(p-2k)^2\binom{p}{k} = p2^p$.

4.6. **On the geometrical trees and their use in the theory of chemicals compounds. (1894)** [27, 28]**.** A chemist asked for some explanations of Cayley's results, mentioned in a German review. Delannoy translated this review and corrected a computational mistake, giving his own method, without knowing [15, 16, 17]. This corresponds to the sequences `EIS 22` (centered hydrocarbons with n atoms) and `EIS 200` (bicentered hydrocarbons with n atoms). Application of combinatorics to enumeration of chemical configurations is a subject which will be later revisited by Pólya [64].

4.7. **How to use a chessboard to solve some probability theory problems (1895)** [31]**.** Delannoy makes a summary of 17 applications of his theory of triangular/square/pentagonal/hexagonal chessboard. The array of Delannoy numbers (see our Section 2) appears on the page 76 from this article.

4.8. **On a question of probabilities studied by d'Alembert (1895)** [30]**.** Delannoy corrects some mistakes in Montfort's solution to a problem raised by d'Alembert.

4.9. **A question of undetermined analysis (1897)** [32]**.** We were not able to get this article. There were in fact two journals whose name was "Journal de Mathématiques élémentaires" (one edited by Vuibert and the other edited by Bourget/Longchamps), none of them seems to contain the quoted article.

4.10. **On the probability of simultaneous events (1898)** [33]**.** A priest wrote an article in which he was bravely contesting the "third Laplace principle" $P(A) \cap P(B) = P(A)P(B)$ for two independent events, arguing with three examples. Delannoy shows that they present a misunderstanding of "independent events", which goes back to the original fuzzy definition by de Moivre.

4.11. **Contributions to "L'Intermédiaire des Mathématiciens" (1894-1908)** [29]**.** This journal was created in 1894 by C.-A. Laisant and Émile Lemoine. It is quite similar to the actual sci.math newsgroups. This journal was indeed only made of problems/questions/solutions/answers.

During the quoted period, numerous famous mathematicians made some contributions to this journal: Appell, Borel, Brocard, Burali-Forti, Cantor, Catalan, Cayley, Cesàro, Chebyshev, Darboux, Dickson, Goursat, Hadamard, Hermite, Jumbert, Hurwitz, Jensen, Jordan, Kempe, Koenigs, Laisant, Landau, Laurent, Lemoine, Lerch, Lévy, Lindelöf, Lipschitz, Moore, Nobel, Picard, Rouché ...

From 1894 until 1908 (date of his last mathematical contribution), Delannoy was an active collaborator: he raised or solved around 70 questions/problems. These are questions number 20, 29, 32, 51, 84, 95, 138, 139, 140, 141, 142, 155, 191, 192, 314, 330, 360, 371, 407, 424, 425, 443, 444, 451, 453, 493, 494, 514, 601, 602, 603,

664, 668, 749, 1090, 1304, 1360, 1459, 1471, 1479, 1551, 1552, 1578, 1659, 1723, 1869, 1875, 1894, 1922, 1925, 1926, 1938, 1939, 2074, 2076, 2077, 2091, 2195, 2212, 2216, 2251, 2305, 2325, 2452, 2455, 2583, 2638, 2648, 2868, 2873, 3326.

These contributions can be classified in three sets: problems and solutions related to combinatorics (enumeration and applications to probabilistic problems), problems and solutions related to elementary number theory (representations of integers as sum of some powers, Fermat-like problems), and questions/answers related to Lucas' books (so mainly recreative mathematics, but not so trivial problems as it includes, *e.g.*, the four color problem).

To these articles, perhaps one should add some récréations of [53] (confer the warning in its preface), and also some problems written by Lucas, but with Delannoy's solutions.

Finally, there are some books [13, 19, 40, 50, 51, 53, 54, 55] (Lucas, Frolow, and Catalan intensively corresponded with Delannoy for their books) or articles [2, 9, 11, 42, 43, 44, 45, 47, 62, 63, 66, 67, 70, 73, 75, 76, 77] which mention either Delannoy numbers or some of Delannoy's results/methods.

5. Other Delannoy's works

Delannoy made some watercolours but his best non mathematical works are in history. Indeed, from 1897 to 1914, he published 29 accurate archaeological/historitical articles in the Mémoires de la Société des Sciences Naturelles et archéologiques de la Creuse.

Let us give a taste of Delannoy's writer talent: here some titles of his articles "On the signification of word *ieuru*", "One more word about *ieuru*", "A riot in Guéret in 1705", "Aubusson's tapestries", "A bigamist in Guéret", "Grapevines in the Creuse", a lot of studies "Criminal trials in the Marche. The case ... ", several studies on abbeys and some "Critical list of the abbots from ... ", and last but not least, "An impotence trial in the 18th century". When he died, at the age of 81, about a dozen of other articles were still in progress.

Delannoy is surely one of the last "self-made" mathematicians who succeeded in getting a name in this field, rivaling professional mathematicians. What he discovered is nowadays well understood and can be classified as "basic enumerative combinatorics". However, despite the simplicity of his tools, it seems to us that Delannoy's work (and more generally, the underlying combinatorics) is a nice example of what could, but is actually not taught to young students (or even in high-schools), as an introduction to research in mathematics, also allowing the use of computers and computer algebra softwares. This kind of mathematics is only present at the mathematical Olympiads. This attractive bridge between enumeration, geometry, probability theory, analysis, ... deserves a better place.

It appears very clearly, thanks to the archives of the Society of Natural Sciences and Archaeology, that besides his own publications, he played a great rôle in checking proofs for numerous mathematicians and historians who wrote to have his contribution. The archive from the Society and from Delannoy's family in Guéret reveals a true *honnête homme*, as defined in the seventeenth century. We plan to have a conference about Delannoy numbers, lattice paths enumeration and related problems in 2003 in Guéret.

Acknowledgements. This work was partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT), by the INRIA postdoctoral programme and by the Max-Planck-Institut.

We want to thank for their technical and friendly help: Jean-Pierre Larduinat (webmaster), Régis Saint-James (secretary) and Étienne Taillemitte (president) from the Société des Sciences Naturelles et Archéologiques de la Creuse (Guéret)[9], Muriel Colombier from the Archives Départementales de la Creuse, Serge Paumier and family Desbaux, from Delannoy's family, Guy Avizou, first maire-adjoint and vice-président of the Conseil Régional de la Creuse, Silke Goebel from the Jahrbuch Project (Electronic Research Archive for Mathematics, Karlsruhe)[10], Anja Becker from the library of the Max-Planck-Institut für Informatik (Saarbrücken)[11], Geneviève Deblock from the Conservatoire numérique des Arts et Métiers (Paris)[12], Brigitte Briot from the library of INRIA (Rocquencourt)[13], Céline Menil from the library of the Maine university (Le Mans)[14], Solange Garnier, Francine Casas and Françoise Thierry from the library of the University of Paris-Nord[15], Nathalie Granottier from the Centre International de Rencontres Mathématiques (Marseilles)[16], and the people from the library of Jussieu (Paris)[17].

We also thank Jean-Michel Autebert, Philippe Flajolet and Peter John for their comments.

[9] http://perso.wanadoo.fr/jp-l/SSC23/
[10] http://www.emis.de/projects/JFM/
[11] http://www.mpi-sb.mpg.de/services/library/
[12] http://cnum.cnam.fr/
[13] http://www-rocq.inria.fr/doc/
[14] http://bu.univ-lemans.fr/
[15] http://www.univ-paris13.fr/
[16] http://www.cirm.univ-mrs.fr/SitBib/Bibli/debut.html
[17] http://bleuet.bius.jussieu.fr/

References

[1] *Dictionnaire Universel du Génie Contemporain*, volume 13. 1893.

[2] A. Aeppli. A propos de l'interprétation géométrique du problème du scrutin. *L'Enseignement Mathématique*, 23:328–329, 1923.

[3] D. André. Note: Calcul des probabilités. Solution directe du problème résolu par M. Bertrand. *Comptes Rendus Mathématique. Académie des Sciences. Paris*, 105:436–437, 1887.

[4] Fernand Autorde and Louis Lacrocq. Nécrologie de M. Henri-Auguste Delannoy. In *Mémoires de la S.S.N.A.C.*, volume 19-2, pages 552–577. Société des Sciences Naturelles et Archéologiques de la Creuse, Guéret, 1915.

[5] Cyril Banderier. Solving discrete initial- and boundary-value problems. In *Proceedings of the Algorithms Seminar Seminars, INRIA Research Report #4056*, 2000. Talk by Marko Petkovšek.

[6] Cyril Banderier and Philippe Flajolet. Basic analytic combinatorics of directed lattice paths. *Theoretical Computer Science*, 281:37–80, 2002.

[7] Berthelot and co. *La grande Encyclopédie, inventaire raisonné des sciences, des lettres et des arts*, volume 13. H. Lamirault et Cie, 1893.

[8] Jean Bertoin and Jim Pitman. Path transformations connecting Brownian bridge, excursion and meander. *Bulletin des Sciences Mathématiques*, 118(2):147–166, 1994.

[9] Joseph Bonin, Louis Shapiro, and Rodica Simion. Some q-analogues of the Schröder numbers arising from combinatorial statistics on lattice paths. *Journal of Statistical Planning and Inference*, 34(1):35–55, 1993.

[10] Mireille Bousquet-Mélou and Marko Petkovšek. Linear recurrences with constant coefficients: the multivariate case. *Discrete Mathematics*, 225(1-3):51–75, 2000. Formal power series and algebraic combinatorics (Toronto, 1998).

[11] Francesco Brenti. Combinatorics and total positivity. *Journal of Combinatorial Theory. Series A*, 71(2):175–218, 1995.

[12] Amédée Carriat. *Dictionnaire bio-bibliographique des auteurs creusois*, volume fascicule 2: B-D. Société des sciences naturelles et archéologiques de la Creuse, 1965.

[13] E. Catalan. Nouvelles notes d'algèbre et d'analyse. *Belg. Mém. XLVIII. 98 S*, 1892.

[14] Eugène Catalan. Note sur une équation aux différences finies. *J. M. Pures Appl.*, 3:508–516, 1838.

[15] Arthur Cayley. On the theory of the analytical forms called tree. *Phil. Mag. XIII. p. 172-176*, 1857.

[16] Arthur Cayley. On the analytical forms called trees, with applications to the theory of chemical combinations. *Rep. Brit. Ass. , 257-305*, 1875.

[17] Arthur Cayley. On the analytical forms called trees. *Sylv., Am. J. IV. 266-268.*, 1881.

[18] Louis Comtet. *Analyse combinatoire. Tomes I, II.* Presses Universitaires de France, Paris, 1970.

[19] Louis Comtet. *Advanced combinatorics.* D. Reidel Publishing Co., Dordrecht, enlarged edition, 1974. The art of finite and infinite expansions.

[20] Anne-marie Décaillot. L'arithméticien édouard Lucas (1842-1891) : théorie et instrumentation. *Revue d'histoire des mathématiques*, 4(2):191–236, 1998.

[21] N. G. de Bruijn, D. E. Knuth, and S. O. Rice. The average height of planted plane trees. In *Graph theory and computing*, pages 15–22. Academic Press, New York, 1972.

[22] Henri Delannoy. Emploi de l'échiquier pour la résolution de problèmes arithmétiques. *Assoc. Franç. Nancy XV*, 1886.

[23] Henri Delannoy. Sur la durée du jeu. *S. M. F. Bull. XVI. 124-128*, 1888.

[24] Henri Delannoy. Emploi de l'échiquier pour la résolution de divers problèmes de probabilité. *Assoc. Franç. Paris XVIII. 43-52*, 1889.

[25] Henri Delannoy. Formules relatives aux coefficients du binôme. *Assoc. Franç. Limoges XIX. 35-37*, 1890.

[26] Henri Delannoy. Problèmes divers concernant le jeu. *Assoc. Franç. Limoges XIX. 29-35*, 1890.

[27] Henri Delannoy. Sur le nombre d'isomères possibles dans une molécule carbonée. *Bulletin de la Société chimique de Paris, 3ème série, T.XI*, pages 239–248, 1892.

[28] Henri Delannoy. Sur les arbres géométriques et leur emploi dans la théorie des combinaisons chimiques. *Assoc. Franç. Caen XXIII. 102-116.*, 1894.

[29] Henri Delannoy. Contributions to "l'intermédiaire des mathèmaticiens". *Intermédiaire des mathèmaticiens*, 1894–1908.

[30] Henri Delannoy. Emploi de l'échiquier pour la résolution de certains problèmes de probabilités. *Assoc. Franç. Bordeaux XXIV, 70-90*, 1895.
[31] Henri Delannoy. Sur une question de probabilités traitée par d'Alembert. *S. M. F. Bull. XXIII. 262-265*, 1895.
[32] Henri Delannoy. Une question d'analyse indéterminée. *Journal de Matématiques élémentaires*, 1897.
[33] Henri Delannoy. Sur la probabilité des événements composés. *S. M. F. Bull. 26, 64-70*, 1898.
[34] Robert Donaghey and Louis W. Shapiro. Motzkin numbers. *J. Combinatorial Theory Ser. A*, 23(3):291–301, 1977.
[35] Philippe Duchon. On the enumeration and generation of generalized Dyck words. *Discrete Mathematics*, 225(1-3):121–135, 2000. Formal power series and algebraic combinatorics (Toronto, 1998).
[36] Alfred Errera. Analysis situs. un probleme d'enumeration. In *Mémoires, Tome XI, Numero 1421, 3e serie*. Académie Royale de Belgique, Brussels, 1931.
[37] Guy Fayolle, Roudolf Iasnogorodski, and Vadim Malyshev. *Random walks in the quarter-plane*. Springer-Verlag, Berlin, 1999. Algebraic methods, boundary value problems and applications.
[38] William Feller. *An introduction to probability theory and its applications. Vol. I.* John Wiley & Sons Inc., New York, 1968. 3rd edition.
[39] Philippe Flajolet. Combinatorial aspects of continued fractions. *Discrete Mathematics*, 32(2):125–161, 1980.
[40] M. Frolow. *Les carrés magiques. Nouvelle étude*. Paris. Gauthier-Villars. VI u. 46 S. gr. 8°. VII. Taf, 1886.
[41] Ira Gessel and Gérard Viennot. Binomial determinants, paths, and hook length formulae. *Advances in Mathematics*, 58(3):300–321, 1985.
[42] Sylviane R. Schwer Jean-Michel Autebert, Matthieu Latapy. Le treillis des chemins de delannoy. Discrete Mathematics, page to appear, 2002.
[43] Peter E. John. Note on a modified pascal triangle connected with the dimer problem. *J.Mol.Struct.(Theochem)*, 277:329–332, 1992.
[44] Shashidhar Kaparthi and H. Raghav Rao. Higher-dimensional restricted lattice paths with diagonal steps. *Discrete Applied Mathematics*, 31(3):279–289, 1991.
[45] Clark Kimberling. Enumeration of paths, compositions of integers, and Fibonacci numbers. *The Fibonacci Quarterly*, 39(5):430–435, 2001.
[46] Christian Krattenthaler, Anthony J. Guttmann, and Xavier G. Viennot. Vicious walkers, friendly walkers and Young tableaux. II. With a wall. *Journal of Physics. A. Mathematical and General*, 33(48):8835–8866, 2000.
[47] G. Kreweras. About Catalan-like lattice paths. *Bulletin of the Institute of Combinatorics and its Applications*, 4:63–64, 1992.
[48] Jacques Labelle and Yeong Nan Yeh. Generalized Dyck paths. *Discrete Mathematics*, 82(1):1–6, 1990.
[49] Édouard Lucas. Sur l'arithmétique figurative. les permutations. *Assoc. Franç. Rouen XII*, 1883.
[50] Édouard Lucas. *Récréations mathématiques. Tome I.* Paris. Gauthier-Villars et Fils., 1891. 2^{e} éd. (first edition in 1881).
[51] Édouard Lucas. *Théorie des nombres. Tome I. Le calcul des nombres entiers. Le calcul des nombres rationnels. La divisibilité arithmétique*. Paris. Gauthier-Villars et Fils., Edition Blanchard 1961, 1891.
[52] Édouard Lucas. *Récréations mathématiques. Tome III*. Paris. Gauthier-Villars et Fils, 1893.
[53] Édouard Lucas. *Récréations mathématiques. Tome IV*. Paris. Gauthier-Villars et Fils, 1894.
[54] Édouard Lucas. *L'arithmétique amusante. Introduction aux Récréations mathématiques. Amusements scientifiques pour l'enseignement et la pratique du calcul*. Paris. Gauthier-Villars et Fils., 1895.
[55] Édouard Lucas. *Récréations mathématiques. Tome II*. Paris. Gauthier-Villars et Fils., 1896. 2^{e} éd. (first edition in 1882).
[56] Donatella Merlini, D. G. Rogers, Renzo Sprugnoli, and M. Cecilia Verri. Underdiagonal lattice paths with unrestricted steps. *Discrete Applied Mathematics*, 91(1-3):197–213, 1999.
[57] Sri Gopal Mohanty. *Lattice path counting and applications*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1979. Probability and Mathematical Statistics.

[58] Theodore Motzkin. Relations between hypersurface cross ratios, and a combinatorial formula for partitions of a polygon, for permanent preponderance, and for non-associative products. *Bull. Amer. Math. Soc.*, 54:352–360, 1948.

[59] Venkata Tadepalli Narayana. Sur les treillis formés par les partitions d'un entier et leurs applications à la théorie des probabilités. *C. R. Acad. Sci. Paris*, 240:1188–1189, 1955.

[60] Eugen Netto. *Lehrbuch der Combinatorik*. Chelsea Publishing Company, New York, 1958. Reprint of the second edition. (first edition in 1901).

[61] Heinrich Niederhausen. Lattice paths between diagonal boundaries. *Electronic Journal of Combinatorics*, 5(1):Research Paper 30, 1998.

[62] Paul Peart. Hankel determinants via Stieltjes matrices. In *Proceedings of the Thirty-first Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 2000)*, volume 144, pages 153–159, 2000.

[63] Jean-Guy Penaud. Une preuve bijective d'une formule de Touchard-Riordan. *Discrete Mathematics*, 139(1-3):347–360, 1995. Formal power series and algebraic combinatorics (Montreal, PQ, 1992).

[64] George Pólya. Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen. *Acta Math.*, 68:145–254, 1937.

[65] John Riordan. *Combinatorial identities*. Robert E. Krieger Publishing Co., Huntington, N.Y., 1979. Reprint of the 1968 original.

[66] E. Rouché. Observations en réponse à une Note de M. Delannoy. *S. M. F. Bull. XVI. 149-150*, 1888.

[67] Horst Sachs and Holger Zernitz. Remark on the dimer problem. *Discrete Applied Mathematics*, 51(1-2):171–179, 1994. 2nd Twente Workshop on Graphs and Combinatorial Optimization (Enschede, 1991).

[68] Ernst Schröder. Vier combinatorische Probleme. *Schlömilch Z. XV. 361-376.*, 1870.

[69] Sylviane R. Schwer. dépendances temporelles : les mots pour le dire. several talk in seminaries, 1997–2000.

[70] Sylviane R. Schwer. S-arrangements avec répétitions. *Comptes Rendus Mathématique. Académie des Sciences. Paris*, 334(4):261–266, 2002.

[71] Andr. de Segner. Enumeratio modorum quibus figurae planae rectilineae per diagonales dividuntur in triangula. *Novi commentarii academiae scientiarum imperialis petropolitanae*, 7:203–210, 1759.

[72] Richard P. Stanley. *Enumerative combinatorics. Vol. 2.* Cambridge University Press, Cambridge, 1999.

[73] Robert A. Sulanke. Counting lattice paths by Narayana polynomials. *Electronic Journal of Combinatorics*, 7(1):Research Paper 40, 2000.

[74] Robert A. Sulanke. Moments of generalized Motzkin paths. *Journal of Integer Sequences*, 3(1):Article 00.1.1, 2000.

[75] Jacques Touchard. Sur un problème de configurations et sur les fractions continues. *Canadian J. Math.*, 4:2–25, 1952.

[76] N. Traverso. Su alcune tavole di addizione per diagonali di passo 1, dedotto dal quadrato aritmetico di Fermat, ed in particolare su quella dell' esagono aritmetico di Delannoy. *Periodico di Mat. 32 [(3) 14], 1-11* , 1917.

[77] Mladen Vassilev and Krassimir Atanassov. On Delanoy [Delaunay] numbers. *Annuaire de l'Université de Sofia "St. Kliment Ohridski".*, 81(1):153–162 (1994), 1987.

E-mail address: `Cyril.Banderier@inria.fr`, `http://algo.inria.fr/banderier`

INRIA-Rocquencourt, 78150 Le Chesnay (France) & Max-Planck-Institut, 66123 Saarbrücken (Germany)

E-mail address: `schwer@lipn.univ-paris13.fr`, `http://www-lipn.univ-paris13.fr/~schwer/`

LIPN- UMR 7030 Université Paris Nord. 99, avenue J.-B. Clément. 93430 Villetaneuse (France)

ENUMERATION OF SOLID 2-TREES

MICHEL BOUSQUET AND CEDRIC LAMATHE

Abstract. The main goal of this paper is to enumerate solid 2-trees according to the number of edges (or triangles) and also according to the edge degree distribution. We first enumerate oriented solid 2-trees using the general methods of the theory of species. In order to obtain non oriented enumeration formulas we use quotient species which consists in a specialization of Pólya theory.

Résumé. Le but de cet article est d'obtenir l'énumération des 2-arbres solides selon le nombre d'arêtes (ou de triangles) ainsi que selon la distribution des degrés des arêtes. Nous obtenons d'abord le dénombrement des 2-arbres solides orientés en utilisant les méthodes de la théorie des espèces. Pour obtenir le dénombrement des 2-arbres solides non orientés, nous utilisons la notion d'espèce quotient qui provient d'une spécialisation de la théorie de Pólya.

1. Introduction

Definition 1. Let $\mathcal{E}$ be a non-empty finite set of n elements called *edges*. A *2-tree* is either a single edge (if $n = 1$) or a non-empty subset $\mathcal{T} \subseteq \wp_3(\mathcal{E})$ whose elements are called *triangles*, satisfying the following conditions:

1. For every pair $\{a, b\} = \{\{a_1, a_2, a_3\}, \{b_1, b_2, b_3\}\}$ of distinct elements of $\mathcal{T}$, we have $|a \cap b| \leq 1$, which means that two distinct triangles share at most one edge.
2. For every ordered pair $(a, b) = (\{a_1, a_2, a_3\}, \{b_1, b_2, b_3\})$ of distinct elements of $\mathcal{T}$, there is a unique sequence $(t_0 = a, t_1, t_2, \ldots, t_k = b)$ such that for $i = 0, 1, \ldots, k-1$, we have $|t_i \cap t_{i+1}| = 1$, which means that each pair of consecutive triangles in this sequence share exactly one edge.

An edge e and a triangle t are *incident* to each other if $e \in t$. The *degree* of an edge is the number of triangles which are incident to that edge. The *edge degree distribution* of a 2-tree is described by a vector $\vec{n} = (n_1, n_2, \ldots)$, where n_i is the number of edges of degree i. We denote by $\mathrm{Supp}(\vec{n})$, the *support* of $\vec{n}$ which is the set of indices i such that $n_i \neq 0$. Figure 1 shows a 2-tree having 11 edges, 5 triangles and edge degree distribution given by $\vec{n} = (8, 2, 1)$.

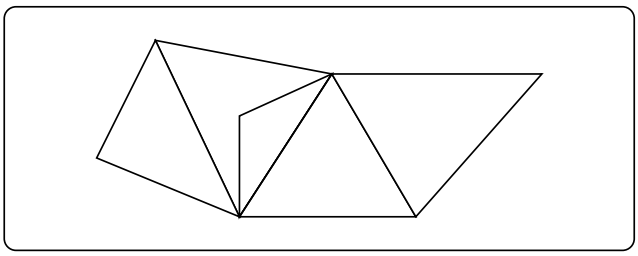


Figure 1. A 2-tree.

Several classes of 2-trees have been studied before. Beineke and Pippert enumerate some k-dimensional trees in [1] labelled at vertices. In [7], Harary and Palmer count unlabelled 2-trees. For the enumeration of plane 2-trees see [10], and for a classification of plane and planar 2-trees see [8]. More recently, in [5, 6], Fowler and al. work on general 2-trees and give asymptotical results. Here, we consider a new class of 2-trees, that is, *solid* 2-trees, *i.e.* 2-trees in which there is a cycle structure on the triangles around each edge.

Lemma 1. Let m, n be two nonnegative integers, and $\vec{n} = (n_1, n_2, \ldots)$, an infinite vector of nonnegative integers. Then

1. There exists a 2-tree having m triangles and n edges if and only if $n = 2m + 1$.
2. There exists a 2-tree having $\vec{n}$ as edge degree distribution if and only if

$$\sum_i n_i = n \quad \text{and} \quad \sum_i i n_i = 3m. \tag{1}$$

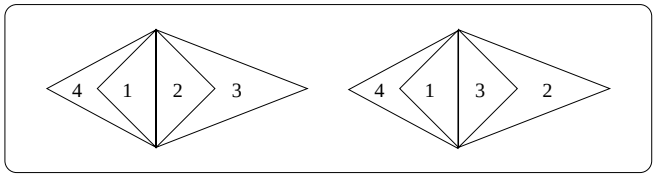


FIGURE 2. Two distinct solid 2-trees but the same 2-tree.

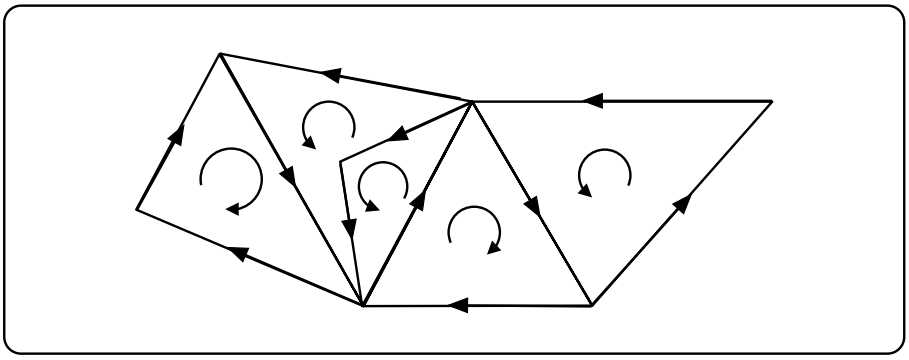


FIGURE 3. A well oriented 2-tree.

A *solid* 2-tree is a 2-tree in which there is a cyclic configuration of triangles around each edge. Figure 2 shows an example of two different solid 2-trees which are in fact the same 2-tree. As we can see, in the case of a solid 2-tree, one has to take into account the cyclic order of the triangles around each edge. A *well oriented* solid 2-tree is obtained from a solid 2-tree in the following way: first, pick any triangle and give a cyclic orientation on its edges. Then each triangle adjacent to the first triangle inherits a circular orientation (see Figure 3). This process is repeated until all edges receive an orientation. By the arborescent nature of the structure, there will be no conflict (the orientation of each edge will always be well defined). Figure 3 shows an example of a well oriented 2-tree. The species of non-oriented and well oriented solid 2-trees will be denoted respectively by $\mathcal{A}$ and $\mathcal{A}_o$. In order to analyze these two species, the following auxiliary species will be used:

- The species of *triangles* X: a single triangle will be denoted by X.
- The species of *edges* Y: a single edge will be denoted by Y.
- The species L of *lists* or *linear orders*.
- The species C and C_3 respectively denoting the species of oriented cycles and of oriented cycles of length 3.
- The species $\mathcal{A}^-$ and $\mathcal{A}_o^{\rightarrow}$ respectively denoting the species of non oriented and well oriented solid 2-trees *rooted at an edge*.
- The species $\mathcal{A}^{\Delta}$ and $\mathcal{A}_o^{\Delta}$ respectively denoting the species of non oriented and well oriented solid 2-trees *rooted at a triangle*.
- The species $\mathcal{A}^{\dot{\Delta}}$ and $\mathcal{A}_o^{\dot{\Delta}}$ respectively denoting the species of non oriented and well oriented solid 2-trees *rooted at a triangle having itself one of its edge distinguished*.

- Finally, the species $\mathcal{B}$ which consists of an oriented root edge Y incident to a linear order (L-structure) of triangles X each of which having its two remaining sides being themselves $\mathcal{B}$-structures. Therefore, the species $\mathcal{B}$ satisfies the following combinatorial equation

$$\mathcal{B}(X,Y) = YL(X\mathcal{B}^2(X,Y)), \tag{2}$$

as illustrated by Figure 4,

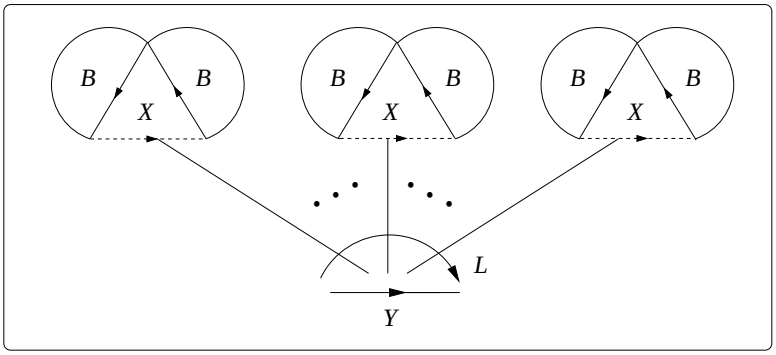


FIGURE 4. A $\mathcal{B}$-structure.

Note that $\mathcal{B}$ has been defined as a *two-sort* species where the sorts are X and Y. Since the numbers of edges n and of triangles m are linked by the relation $n = 2m + 1$, equation (2) above can either be expressed as a one sort species in X alone by setting $Y := 1$, or in Y alone, by setting $X := 1$ respectively, giving the two following equations:

$$\mathcal{B}(X) = L(X\mathcal{B}^2(X)), \tag{3}$$

$$\mathcal{B}(Y) = YL(B^2(Y)). \tag{4}$$

Recall that setting $X := 1$ in a two sort species $F(X,Y)$ essentially means unlabelling the elements of sort X. The second form in equation (4) is more suitable for the use of Lagrange inversion formula. Therefore the species Y of edges will be used as the base singleton species to make our computations. However, the results will be shorter and more elegant when expressed as a function of the number m of triangles.

• Lagrange Inversion Formula

In this paper we make an extensive use of Lagrange inversion formula (see [2]): Let A and R be species satisfying $A(Y) = YR(A)$. If F is another species, then

$$[y^n]F(A(y)) = \frac{1}{n}[y^{n-1}]F'(t)R^n(t), \tag{5}$$

where $[y^n]F(A(y))$ denotes the coefficient of y^n in $F(A(y))$. Another main tool used in this paper is the following dissymmetry theorem which has been proved in [5]. Note that in their paper, the authors made a proof for non solid 2-trees but obviously, the proof is also valid for both well oriented and non oriented solid 2-trees.

Theorem 1. The species $\mathcal{A}_o$ and $\mathcal{A}$ respectively of well oriented and (non oriented) solid 2-trees satisfy the following relations:

$$\mathcal{A}_o^{\rightarrow} + \mathcal{A}_o^{\Delta} = \mathcal{A}_o + \mathcal{A}_o^{\underline{\Delta}}, \tag{6}$$

and

$$\mathcal{A}^{-} + \mathcal{A}^{\Delta} = \mathcal{A} + \mathcal{A}^{\underline{\Delta}}. \tag{7}$$

2. Well oriented solid 2-trees

We begin this section by expressing the species appearing in the dissymmetry theorem (oriented case) in terms of the species $\mathcal{B}$.

Theorem 2. The species $\mathcal{A}_o^{\rightarrow}$, $\mathcal{A}_o^{\Delta}$ and $\mathcal{A}_o^{\underline{\Delta}}$ satisfy the following isomorphisms of species :

$$\mathcal{A}_o^{\rightarrow}(Y) = YC(\mathcal{B}^2(Y)), \tag{8}$$

$$\mathcal{A}_o^{\Delta}(Y) = C_3(\mathcal{B}(Y)), \tag{9}$$

$$\mathcal{A}_o^{\underline{\Delta}}(Y) = \mathcal{B}(Y)^3, \tag{10}$$

where C and C_3 are the species of oriented cycles and of oriented cycles of length 3.

2.1. Enumeration according to the number of edges.

• Labelled case

Let $\mathcal{A}_o[n]$ be the number of edge labelled solid 2-trees over n edges. We similarly define $\mathcal{A}_o^{\rightarrow}[n]$, $\mathcal{A}_o^{\Delta}[n]$ and $\mathcal{A}_o^{\underline{\Delta}}[n]$. Our first task is to determine $\mathcal{A}_o^{\rightarrow}[n]$. By applying Lagrange inversion with $F(t) = C(t^2) = -\log(1-t^2)$ and $R(t) = L(t^2) = (1-t^2)^{-1}$, we find

$$\begin{aligned}[y^n]\mathcal{A}_o^{\rightarrow}(y) &= [y^{n-1}]C(\mathcal{B}^2(y)),\\ &= \frac{2}{3(n-1)}\binom{3(n-1)/2}{n-1}.\end{aligned}$$

Hence, the number $\mathcal{A}_o^{\rightarrow}[n]$ of edge labelled solid 2-trees pointed at an edge over n edges is given by

$$\mathcal{A}_o^{\rightarrow}[n] = n![y^n]\mathcal{A}_o^{\rightarrow}(y) = \frac{2}{3}n(n-2)!\binom{\frac{3(n-1)}{2}}{n-1}. \tag{11}$$

Now, using equation (9) and Lagrange inversion with $F(t) = C_3(t) = t^3/3$ and $R(t) = (1-t^2)^{-1}$, we obtain

$$\mathcal{A}_o^{\Delta}[n] = \frac{1}{3}(n-1)!\binom{\frac{3(n-1)}{2}}{n-1}. \tag{12}$$

To compute $\mathcal{A}_o^{\underline{\Delta}}[n]$, we use equation (10) and Lagrange inversion with $F(t) = t^3$ and $R(t) = (1-t^2)^{-1}$ and we get

$$\mathcal{A}_o^{\underline{\Delta}}[n] = (n-1)!\binom{\frac{3(n-1)}{2}}{n-1}. \tag{13}$$

Using equations (11), (12) and (13) and the dissymmetry theorem, we have:

Proposition 1. The number $\mathcal{A}_o[n]$ of well oriented edge-labelled solid 2-trees over n edges is given by

$$\mathcal{A}_o[n] = \frac{2}{3}(n-2)!\binom{\frac{3(n-1)}{2}}{n-1}, \quad n > 1. \tag{14}$$

Note that if we express equation (14) as a function of m, the number of triangles, we obtain

$$\mathcal{A}_o[m] = \frac{m!}{3}\frac{1}{2m+3}\binom{3m+3}{m+1}, \quad m \geq 1. \tag{15}$$

• Unlabelled case

We first need to compute the generating series $\widetilde{\mathcal{A}_o}^{\rightarrow}(y)$. In order to accomplish this, we use the following property: let F and G be two species, then we have

$$\widetilde{F(G)}(x) = Z_F(\tilde{G}(x), \tilde{G}(x^2), \tilde{G}(x^3), \ldots), \tag{16}$$

where the *cycle index series* Z_F of a species is defined by

$$Z_F(x_1, x_2, \ldots) = \sum_{k\geq 0}\frac{1}{k!}\sum_{\sigma\in\mathcal{S}_k}\mathrm{fix}F[\sigma]x_1^{\sigma_1}x_2^{\sigma_2}x_3^{\sigma_3}\cdots, \tag{17}$$

where $\mathcal{S}_k$ is the symmetric group of order k and σ_i, the number of cycles of length i in σ and $\mathrm{fix}F[\sigma]$ is the number of F-structures left fixed under the relabelling induced by σ. For example, if $F = C$, the species of oriented cycles, we have

$$Z_C(x_1, x_2, \ldots) = \sum_{k\geq 1} \frac{\phi(k)}{k} \log\left(\frac{1}{1-x_k}\right). \tag{18}$$

Now, applying this to the species $\mathcal{A}_o^{\rightarrow} = YC(\mathcal{B}^2)$, we get

$$\begin{aligned} \widetilde{\mathcal{A}_o}^{\rightarrow}(y) &= yZ_C\left(\tilde{\mathcal{B}}^2(y), \tilde{\mathcal{B}}^2(y^2), \tilde{\mathcal{B}}^2(y^3), \ldots\right), \\ &= y\sum_{k\geq 1} \frac{\phi(k)}{k} \log\left(\frac{1}{1-\tilde{\mathcal{B}}^2(y^k)}\right). \end{aligned}$$

We note that since $\mathcal{B}$ is asymmetric (there are exactly $n!$ labelled structures for each unlabelled structures), we have $\tilde{\mathcal{B}}(y) = \mathcal{B}(y)$, hence

$$\begin{aligned} \widetilde{\mathcal{A}_o}^{\rightarrow}[n] &= [y^n]\widetilde{\mathcal{A}_o}^{\rightarrow}(y), \\ &= [y^{n-1}]\sum_{k\geq 1} \frac{\phi(k)}{k} \log\left(\frac{1}{1-\mathcal{B}^2(y^k)}\right). \end{aligned}$$

But

$$\begin{aligned} [y^{n-1}]\log\left(\frac{1}{1-\mathcal{B}^2(y^k)}\right) &= \frac{2k}{n-1}[t^{\frac{n-1}{k}-2}](1-t^2)^{-\frac{n-1}{k}-1}, \\ &= \frac{2k}{3(n-1)}\binom{3(n-1)/2k}{(n-1)/k}. \end{aligned}$$

Obviously, k must divide $n-1$ and $(n-1)/k$ must be even. Letting $d = (n-1)/k$, we finally get

$$\widetilde{\mathcal{A}_o}^{\rightarrow}[n] = \frac{2}{3(n-1)}\sum_d \phi((n-1)/d)\binom{3d/2}{d}, \tag{19}$$

the sum being taken over all even divisors d of $n-1$. To compute $\widetilde{\mathcal{A}_o}^{\triangle}[n]$, we use equation (9) and the fact that

$$Z_{C_3}(y_1, y_2, \ldots) = \frac{1}{3}(y_1^3 + 2y_3).$$

We have

$$[y^n]\mathcal{B}^3(y) = \frac{1}{n}\binom{3(n-1)/2}{n-1},$$

and

$$[y^n]\mathcal{B}(y^3) = [y^{n/3}]\mathcal{B}(y) = \frac{3}{n}\binom{(n-3)/2}{n/3-1},$$

so that

$$\widetilde{\mathcal{A}_o}^{\triangle}[n] = \frac{1}{3n}\binom{\frac{3(n-1)}{2}}{n-1} + \frac{2}{n}\chi(3|n)\binom{\frac{(n-3)}{2}}{\frac{n}{3}-1}, \tag{20}$$

where $\chi(3|n) = 1$ if 3 divides n and 0 otherwise. It can be easily shown, by a very similar way that

$$\widetilde{\mathcal{A}}_o^{\underline{\Delta}}[n] = \frac{1}{n}\binom{\frac{3(n-1)}{2}}{n-1}. \tag{21}$$

And we get the following result:

Proposition 2. The number of unlabelled well oriented solid 2-trees over n edges is given by

$$\widetilde{\mathcal{A}_o}[n] = \frac{2}{3(n-1)}\sum_d \phi\left(\frac{n-1}{d}\right)\binom{3d/2}{d} + \chi(3|n)\frac{2}{n}\binom{\frac{n-3}{2}}{\frac{n}{3}-1} - \frac{2}{3n}\binom{\frac{3(n-1)}{2}}{n-1}, \tag{22}$$

the first sum being taken over all even divisors d of $n-1$.

We can also write $\widetilde{\mathcal{A}_o}[m]$, in function of the number m of triangles, as follows

$$\widetilde{\mathcal{A}_o}[m] = \frac{1}{3m}\sum_{d|m}\phi\left(\frac{m}{d}\right)\binom{3d}{d} + \chi(3|2m+1)\frac{2}{2m+1}\binom{m-1}{\frac{2m-2}{3}} - \frac{2}{3(2m+1)}\binom{3m}{m}.$$

Note that this expression is also the number of unlabelled 3-gonal cacti on m 3-gones (see [3]). The sequence of these numbers is known as sequence A054423 in the on-line encyclopedia of integers sequences ([11]).

2.2. **Enumeration according to edge degree distribution.**

Let $r = (r_0, r_1, r_2, \ldots)$ be an infinite set of formal variables. In order to keep track of the edge degree distribution, we introduce, for a given number n and F, any species, the following weight function:

$$\begin{array}{rccc} w: & F[n] & \longrightarrow & Q[r_1, r_2, \ldots] \\ & s & \mapsto & w(s) \end{array} \tag{23}$$

where $Q[r_1, r_2, \ldots]$ is the ring of polynomials over Q in the variables $r_1, r_2, \ldots$ and where the weight of a given structure s is defined by $w(s) = r_1^{n_1} r_2^{n_2} \cdots$, where n_i is the number of edges of degree i in s. Equations (2), (8), (9) and (10) have the following weighted versions:

$$\mathcal{B}_r = Y L_{r'}(B_r^2), \tag{24}$$

and

$$\mathcal{A}^{\rightarrow}_{o,w}(Y) = Y C_r(\mathcal{B}_r^2), \tag{25}$$

$$\mathcal{A}^{\Delta}_{o,w}(Y) = C_3(B_r), \tag{26}$$

$$\mathcal{A}^{\underline{\Delta}}_{o,w}(Y) = \mathcal{B}_r^3, \tag{27}$$

where C_r is the weighted species of cycles such that a cycle of length i has the weight r_i, and its derivative $L_{r'}$ which is the species of lists where a list of length i has the weight r_{i+1}. These species have the following generating series:

$$C_r(y) = r_1 y + \frac{r_2}{2}y^2 + \frac{r_3}{3}y^3 + \cdots,$$

and

$$L_{r'}(y) = r_1 + r_2 y + r_3 y^2 + \cdots.$$

Let $\vec{n} = (n_1, n_2, n_3, \ldots)$ be a vector of nonnegative integers. Recall that there exists a 2-tree having a total of n edges and n_i edges of degree i if and only if the following relation is satisfied:

$$\sum_i n_i = n \quad \text{and} \quad \sum_i i n_i = 3\left(\frac{n-1}{2}\right). \tag{28}$$

• Labelled case

Let $\vec{n}$ be a vector satisfying (28). Then the number $\mathcal{A}^{\rightarrow}_o[\vec{n}]$ of well oriented edge labelled solid 2-trees pointed at an edge, and having $\vec{n}$ as edge degree distribution, is given by

$$\mathcal{A}^{\rightarrow}_o[\vec{n}] = n![y^n][r_1^{n_1} r_2^{n_2} \cdots]\mathcal{A}^{\rightarrow}_{o,w}(y). \tag{29}$$

We have

$$\begin{aligned} [y^n]\mathcal{A}^{\rightarrow}_{o,w}(y) &= \frac{1}{n-1}[t^{n-2}]\frac{d}{dt}(C_r(t^2))\cdot L_{r'}^{n-1}(t^2), \\ &= \frac{2}{n-1}[t^{n-3}](r_1 + r_2 t^2 + r_3 t^4 + \cdots)^n, \\ &= \frac{2}{n-1}[t^{n-3}]\sum_{\ell_1+\ell_2+\cdots=n}\binom{n}{\ell_1, \ell_2, \ldots} r_1^{\ell_1} r_2^{\ell_2} \cdots t^{2\ell_2+4\ell_3+6\ell_4+\cdots}. \end{aligned}$$

Finally, we obtain

$$[y^n]\mathcal{A}_o^{\rightarrow}(r,y) = \sum_{\ell_1,\ell_2,\ldots} \binom{n}{\ell_1,\ell_2,\ldots} r_1^{\ell_1} r_2^{\ell_2} \cdots,$$

the sum being taken over all vectors $(\ell_1,\ell_2,\ldots)$ satisfying

$$\sum_i \ell_i = n \quad \text{and} \quad \sum_i 2(i-1)\ell_i = n-3.$$

We note that this condition is the same as in (28). Hence using (29) we have

$$\mathcal{A}_o^{\rightarrow}[\vec{n}] = 2n(n-2)!\binom{n}{n_1,n_2,\ldots}. \tag{30}$$

For $\mathcal{A}_o^{\Delta}[\vec{n}]$, we have

$$\mathcal{A}_o^{\Delta}[\vec{n}] = n![y^n][r_1^{n_1} r_2^{n_2}\cdots]\mathcal{A}_{o,w}^{\Delta}(y).$$

But,

$$[y^n]\mathcal{A}_{o,w}^{\Delta}(y) \quad = \quad \frac{1}{n}\sum_{\ell_1,\ell_2,\ldots} \binom{n}{\ell_1,\ell_2,\ldots} r_1^{\ell_1} r_2^{\ell_2} \cdots,$$

the sum being taken on all vectors $(\ell_1,\ell_2,\ldots)$ satisfying $\sum_i \ell_i = n$ and $\sum_i 2(i-1)\ell_i = n-3$, and we obtain

$$\mathcal{A}_o^{\Delta}[\vec{n}] = (n-1)!\binom{n}{n_1,n_2,\ldots}. \tag{31}$$

It can be easily shown that $\mathcal{A}_o^{\Delta}[\vec{n}] = 3\mathcal{A}_o^{\Delta}[\vec{n}]$, hence we have

$$\mathcal{A}_o^{\Delta}[\vec{n}] = 3(n-1)!\binom{n}{n_1,n_2,\ldots}. \tag{32}$$

Now using (30), (31), (32) and the dissymmetry theorem we find

$$\mathcal{A}_o[\vec{n}] = 2(n-2)!\binom{n}{n_1,n_2,\ldots}. \tag{33}$$

• **Unlabelled case**

Let $\vec{n} = (n_1,n_2,\ldots)$ be a coherent edge degree distribution. In order to compute the number $\widetilde{\mathcal{A}_o^{\rightarrow}}[\vec{n}]$ of unlabelled $\mathcal{A}_o^{\rightarrow}$-structures having $\vec{n}$ as edge degree distribution, we use the fact that given two weighted species F_w and G_v, the generating series $\tilde{H}(y)$ of unlabelled H-structures, where $H = F_w(G_v)$, is given by

$$\tilde{H}(y) = Z_{F_w}(\tilde{G}_v(y), \tilde{G}_{v^2}(y^2), \tilde{G}_{v^3}(y^3), \ldots). \tag{34}$$

In the present case, we have $\mathcal{A}_{o,w}^{\rightarrow} = YC_r(\mathcal{B}_r^2)$, and since the species $\mathcal{B}$ is asymmetric, $\tilde{\mathcal{B}}_r(y) = \mathcal{B}_r(y)$, hence

$$\widetilde{\mathcal{A}_o^{\rightarrow}}[\vec{n}] = [y^{n-1}][r_1^{n_1} r_2^{n_2}\cdots] Z_{C_r}(\mathcal{B}_r^2(y), \mathcal{B}_{r^2}^2(y^2), \mathcal{B}_{r^3}^2(y^3), \ldots). \tag{35}$$

But $Z_{C_r}(y_1,y_2,\ldots)$ can be expressed as the following sum:

$$Z_{C_r}(y_1,y_2,\ldots) = \sum_{k\geq 1} \frac{r_k}{k} \sum_{d|k} \phi(d) y_d^{k/d}. \tag{36}$$

Combinatorially speaking, the integer k represents the degree of the root edge. Hence, k may only belong to $\mathrm{Supp}(\vec{n})$, the *support* of $\vec{n}$ which is the set of integers i such that $n_i \neq 0$. Hence, we have

$$\widetilde{\mathcal{A}_o^{\rightarrow}}[\vec{n}] = [y^{n-1}][r_1^{n_1} r_2^{n_2}\cdots] \sum_{k\in \mathrm{Supp}(\vec{n})} \frac{r_k}{k} \sum_{d|k} \phi(d) \mathcal{B}_{r^d}^{2k/d}(y^d). \tag{37}$$

First, we compute

$$[y^{n-1}]\mathcal{B}_{r^d}^{2k/d}(y^d) = [y^{(n-1)/d}]\mathcal{B}_{r^d}^{2k/d}(y).$$

From Lagrange inversion, we have

$$\begin{aligned}[y^m]\mathcal{B}^{\ell}_{r^d}(y) &= \frac{1}{m}[t^{m-1}]\frac{d}{dt}\left(t^{\ell}\right) L^m_{r^{\prime d}}(t^2),\\ &= \frac{\ell}{m}\sum_{\ell_1,\ell_2,\ldots}\binom{m}{\ell_1,\ell_2,\ldots} r_1^{d\ell_1} r_2^{d\ell_2}\cdots, \end{aligned} \tag{38}$$

where the ℓ_i's satisfy $\sum_i \ell_i = m$ and $\sum_i 2(i-1)\ell_i = m-\ell$. Now, letting $m=(n-1)/d$ and $\ell = 2k/d$, we find

$$\widetilde{\mathcal{A}_o}^{\rightarrow}[\vec{n}] = [r_1^{n_1} r_2^{n_2}\cdots]\frac{2}{n-1}\sum_{k\in \mathrm{Supp}(\vec{n})}\sum_{d|k}\phi(d)\sum_{\ell_1,\ell_2,\ldots}\binom{(n-1)/d}{\ell_1,\ell_2,\ldots} r_1^{d\ell_1} r_2^{d\ell_2}\cdots r_k^{d\ell_k+1}\cdots. \tag{39}$$

Finally, we have

Proposition 3. Let $\vec{n}$ be a coherent edge degree distribution, then the number $\widetilde{\mathcal{A}_o}^{\rightarrow}[\vec{n}]$ of unlabelled oriented solid 2-trees pointed at an edge and having $\vec{n}$ as edge degree distribution is given by

$$\widetilde{\mathcal{A}_o}^{\rightarrow}[\vec{n}] = \frac{2}{n-1}\sum_{k\in \mathrm{Supp}(\vec{n})}\sum_{d|\{k,\vec{n}-\delta_k\}}\phi(d)\binom{\frac{n-1}{d}}{\frac{\vec{n}-\delta_k}{d}}, \tag{40}$$

where $\frac{\vec{n}-\delta_k}{d} = (\frac{n_1}{d}, \frac{n_2}{d}, \ldots, \frac{n_k-1}{d}, \ldots)$, for $d\geq 1$ and

$$\binom{\frac{n-1}{d}}{\frac{\vec{n}-\delta_k}{d}} = \binom{\frac{n-1}{d}}{n_1/d, n_2/d, \ldots, (n_k-1)/d, \ldots}.$$

Let $\widetilde{\mathcal{A}}^{\Delta}_o[\vec{n}]$ and $\widetilde{\mathcal{A}}^{\Delta}_o[n]$ be the numbers of unlabelled oriented solid 2-trees pointed respectively at a triangle and at a triangle pointed itself at one of its edge and having $\vec{n}$ as edge degree distribution. We have

Proposition 4. Let $\vec{n}$ be a coherent edge degree distribution, then the numbers $\widetilde{\mathcal{A}_o}^{\Delta}[\vec{n}]$ and $\widetilde{\mathcal{A}_o}^{\Delta}[\vec{n}]$ are given by

$$\widetilde{\mathcal{A}_o}^{\Delta}[\vec{n}] = \frac{1}{n}\binom{n}{n_1,n_2,\ldots} + \frac{\chi(3|\vec{n})}{n}\binom{n/3}{n_1/3, n_2/3,\ldots}, \tag{41}$$

$$\widetilde{\mathcal{A}_o}^{\Delta}[\vec{n}] = \frac{3}{n}\binom{n}{n_1,n_2,\ldots}, \tag{42}$$

where

$$\chi(3|\vec{n}) = \begin{cases} 1, & \text{if all components of } \vec{n} \text{ are multiples of 3}\\ 0, & \text{otherwise.}\end{cases}$$

Proof. Let us start with $\widetilde{\mathcal{A}_o}^{\Delta}[\vec{n}]$. We have

$$\begin{aligned}\widetilde{\mathcal{A}_o}^{\Delta}[\vec{n}] &= [y^n][r_1^{n_1} r_2^{n_2}\cdots]\widetilde{\mathcal{A}}^{\Delta}_{o,w}(y),\\ &= [y^n][r_1^{n_1} r_2^{n_2}\cdots]Z_{C_3}(\tilde{\mathcal{B}}_r(y), \tilde{\mathcal{B}}_{r^2}(y^2),\ldots),\\ &= [y^n][r_1^{n_1} r_2^{n_2}\cdots]Z_{C_3}(\mathcal{B}_r(y), \mathcal{B}_{r^2}(y^2),\ldots).\end{aligned}$$

Since $Z_{C_3}(y_1,y_2,\ldots) = (y_1^3+2y_3)/3$,

$$\widetilde{\mathcal{A}_o}^{\Delta}[\vec{n}] = \frac{1}{3}[y^n][r_1^{n_1} r_2^{n_2}\cdots]\left(\mathcal{B}^3_r(y) + 2\mathcal{B}_{r^3}(y^3)\right) \tag{43}$$

From equation (38) letting $m=n$, $\ell=3$ and $d=1$, we get

$$[y^n]\mathcal{B}^3_r(y) = \frac{3}{n}\sum_{\ell_1,\ell_2,\ldots}\binom{n}{\ell_1,\ell_2,\ldots} r_1^{\ell_1} r_2^{\ell_2}\cdots, \tag{44}$$

where the ℓ_i's satisfy $\sum_i \ell_i = n$ and $\sum_i 2(i-1)\ell_i = n-3$. Now letting $m = n/3$, $\ell = 1$ and $d = 3$, we get

$$[y^n]\mathcal{B}_{r^3}(y^3) = [y^{n/3}]\mathcal{B}_{r^3}(y) = \frac{3}{n}\sum_{\ell_1,\ell_2,\ldots}\binom{n/3}{\ell_1,\ell_2,\ldots}r_1^{3\ell_1}r_2^{3\ell_2}\cdots, \tag{45}$$

where the ℓ_i's satisfy $\sum_i \ell_i = n$ and $\sum_i 2(i-1)\ell_i = n-1$. Now letting $\ell_i = n_i$ in (44) and $\ell_i = n_i/3$ in (45), we get equation (41). We obtain (42) in a very similar way. □

Finally, using the dissymmetry theorem, we obtain the final result of this section:

Proposition 5. Let $\vec{n}$ be a coherent edge degree distribution, then the number $\widetilde{\mathcal{A}_o}[\vec{n}]$ of unlabelled oriented solid 2-trees having $\vec{n}$ as edge degree distribution is given by

$$\widetilde{\mathcal{A}_o}[\vec{n}] = \frac{2}{n-1}\sum_{k\in \mathrm{Supp}(\vec{n})}\sum_{d|\{k,\vec{n}-\delta_k\}}\phi(d)\binom{\frac{n-1}{d}}{\frac{\vec{n}-\delta_k}{d}} + \frac{\chi(3|\vec{n})}{n}\binom{\frac{n}{3}}{\frac{n_1}{3},\frac{n_2}{3},\ldots} - \frac{2}{3n}\binom{n}{n_1,n_2,\ldots}, \tag{46}$$

where

$$\chi(3|\vec{n}) = \begin{cases} 1, & \text{if all components of } \vec{n} \text{ are multiples of 3,}\\ 0, & \text{otherwise,}\end{cases}$$

$$\frac{\vec{n}-\delta_k}{d} = \left(\frac{n_1}{d},\frac{n_2}{d},\ldots,\frac{n_k-1}{d},\ldots\right) \text{ for } d\geq 1,$$

and

$$\binom{\frac{n-1}{d}}{\frac{\vec{n}-\delta_k}{d}} = \binom{\frac{n-1}{d}}{n_1/d, n_2/d,\ldots,(n_k-1)/d,\ldots}.$$

3. Non-oriented solid 2-trees

In order to compute the numbers of labelled and unlabelled solid 2-trees, we use Burnside's Lemma with $\mathbb{Z}_2 = \{Id, \tau\}$, where the action of τ is to reverse the orientation of the structures.

3.1. Enumeration according to the number of edges.

• Labelled case

The labelled case is particularly simple since the only labelled oriented 2-tree which is left fixed under the action of τ is the structure consisting of a single oriented edge. Hence, we have

Proposition 6. The number $\mathcal{A}[n]$ of edge labelled solid 2-trees over n edges is given by

$$\mathcal{A}[n] = \begin{cases} \frac{1}{2}\mathcal{A}_0[n] & \text{if } n>1;\\ 1 & \text{if } n=1.\end{cases} \tag{47}$$

Of course, the same argument will remain valid for all other pointed structures discussed in the previous section.

• Unlabelled case

In the unlabelled case, the action of τ is not so trivial. Figure 5 shows a structure which is left fixed under the action of τ. Let $\mathcal{A}^-$ be the species of unoriented solid 2-trees rooted at an edge. This species can be expressed as the following quotient species (see [4]):

$$\mathcal{A}^- = \frac{\mathcal{A}_o^{\rightarrow}}{\mathbb{Z}_2} = \frac{YC(\mathcal{B}^2(Y))}{\mathbb{Z}_2}, \tag{48}$$

where $\mathbb{Z}_2 = \{\mathrm{Id}, \tau\}$ is the two element group consisting of the identity and τ, whose action is to reverse the orientation of the edges. Hence, an unlabelled $\mathcal{A}^-$-structure is an orbit $\{a, \tau\cdot a\}$ under the action of $\mathbb{Z}_2$ where a is any (oriented) unlabelled $\mathcal{A}_o^{\rightarrow}$-structure.

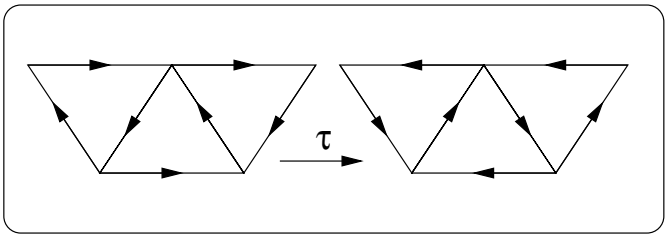


FIGURE 5. An unlabelled 2-tree invariant under the action of τ.

Let us introduce the auxiliary species $\mathcal{B}_{\mathrm{Sym}}$ of τ-symmetric $\mathcal{B}$-structures, *i.e* the species of $\mathcal{B}$-structures left fixed under the edge orientation inversion. Denote by $\mathcal{B}_{\mathrm{Sym}}(y)$ its ordinary generating series. Recall the functional equation verified by the species $\mathcal{B}$:

$$\mathcal{B} = YL(B^2).$$

In order to compute $\mathcal{B}_{\mathrm{Sym}}(y)$, we have to distinguish two cases according to the parity of k, the length of the list of $\mathcal{B}^2$-structures attached to the rooted edge. First consider the case where k is odd (Figure 6 shows an example where $k = 5$). A τ-symmetric $\mathcal{B}$-structure must have a reflective symmetry plane. This plane contains the middle triangle of the list. When an inversion of the orientation of the rooted edge is applied, the two $\mathcal{B}$-structures glued on the two (non root) sides of the middle triangle (structures $\mathcal{B}_5$ and $\mathcal{B}_{5'}$ in Figure 6) are isomorphically exchange. The $k-1$ remaining triangles are exchanged pairwise carrying with them each of their attached $\mathcal{B}$-structures as shown in Figure 6. This gives a factor of $\mathcal{B}^k(y^2)$. We then have to sum the previous expression over all odd values of k. The case where k is even, is very similar except that the symmetry plane must passes between two triangles as shown in Figure 7 and we get the same expression summed over all even values of k. Therefore, we have

$$\mathcal{B}_{\mathrm{Sym}}(y) = y \sum_{k \geq 0} \mathcal{B}^k(y^2) = \frac{y}{1 - \mathcal{B}(y^2)}. \tag{49}$$

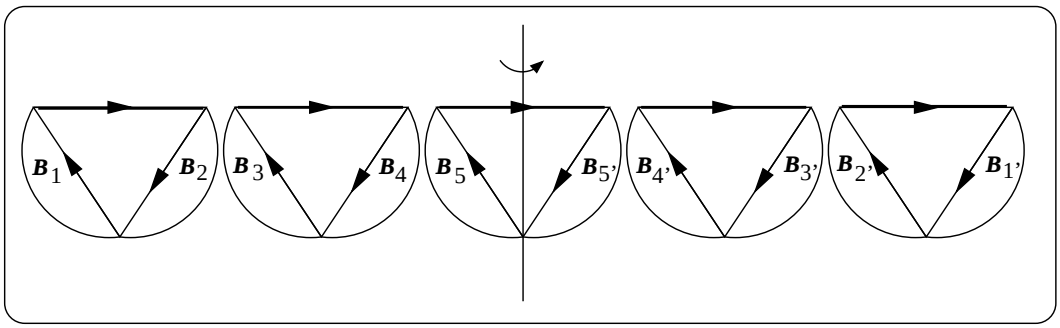


FIGURE 6. A $\mathcal{B}_{\mathrm{Sym}}$-structure, k odd.

From expression (49) and another use of Lagrange inversion, we easily obtain the following result.

Proposition 7. The number $\mathcal{B}_{\mathrm{sym}}[m]$ of τ-symmetric unlabelled oriented $\mathcal{B}$-structures is given by

$$\mathcal{B}_{\mathrm{sym}}[m] = \begin{cases} \dfrac{1}{m+1}\dbinom{3m/2}{m} & \text{if } m \text{ is even,} \\ \dfrac{1}{m}\dbinom{(3m-1)/2}{m+1} + \dfrac{1}{3m}\dbinom{3(m+1)/2}{m+1} & \text{if } m \text{ is odd,} \end{cases} \tag{50}$$

where $m = (n-1)/2$ is the number of triangles and n, the number of edges.

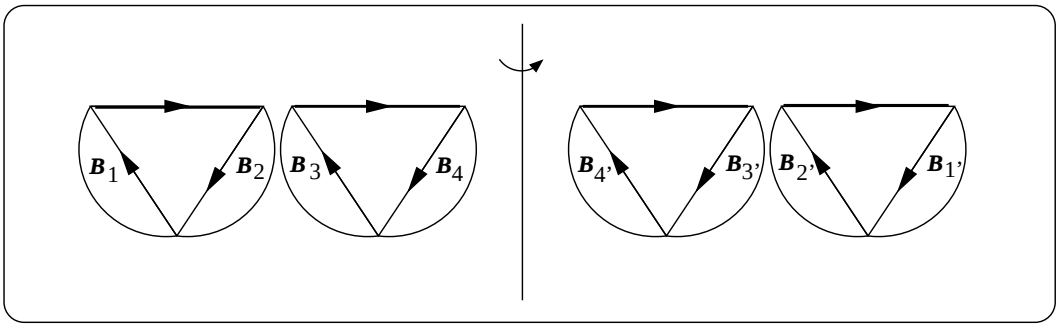


FIGURE 7. A $\mathcal{B}_{\mathrm{Sym}}$-structure, k even.

We now give an expression for the generating function of unlabelled quotient structures, which will allow us to enumerate various kind of unlabelled solid 2-trees (see [4], proposition 2.2.4).

Proposition 8. Let F be any (weighted) species and G, a group acting on F. Then the ordinary generating series of the quotient species F/G is given by

$$(F/G)^{\sim}(y) = \frac{1}{|G|}\sum_{g\in G}\sum_{n\geq 0}|\mathrm{Fix}_{\tilde{F}_n}(g)|_w y^n, \tag{51}$$

where $\mathrm{Fix}_{\tilde{F}_n}(g)$ denotes the set of unlabelled F-structures left fixed under the action of $g\in G$ and $|\mathrm{Fix}_{\tilde{F}_n}(g)|_w$ represents the total weight of this set.

Using an unweighted version of Proposition 8 with $F=\mathcal{A}_o^{\rightarrow}$ and $G=\mathbb{Z}_2$, we obtain

$$\widetilde{\mathcal{A}}^{-}(y) = \frac{1}{2}\sum_{n\geq 0}|\mathrm{Fix}_{\tilde{\mathcal{A}}^{\rightarrow}_{o,n}}(\mathrm{Id})|y^n + \frac{1}{2}\sum_{n\geq 0}|\mathrm{Fix}_{\tilde{\mathcal{A}}^{\rightarrow}_{o,n}}(\tau)|y^n, \tag{52}$$

$$= \frac{1}{2}\widetilde{\mathcal{A}}_o^{\rightarrow}(y) + \frac{1}{2}\mathcal{B}_{\mathrm{sym}}(y), \tag{53}$$

since an oriented $\mathcal{A}^{-}$-structure left fixed under the action of τ is in fact a $\mathcal{B}_{\mathrm{Sym}}$-structure. Then, it becomes easy to extract the coefficient of y^n in relation (53), and we get the number $\mathcal{A}^{-}[n]$ of edge pointed solid 2-trees over n edges

$$\mathcal{A}^{-}[n] = \frac{1}{2}\widetilde{\mathcal{A}}_o^{\rightarrow}[n] + \frac{1}{2}\mathcal{B}_{\mathrm{Sym}}[n]. \tag{54}$$

We now consider the species $\mathcal{A}^{\Delta}$ of triangle rooted solid 2-trees. Since $\mathcal{A}^{\Delta}=\mathcal{A}_o^{\Delta}/\mathbb{Z}_2$, by virtue of Proposition 8, we have

$$\widetilde{\mathcal{A}}^{\Delta}(y) = \frac{1}{2}\sum_{n\geq 0}|\mathrm{Fix}_{\tilde{\mathcal{A}}^{\Delta}_{o,n}}(\mathrm{Id})|y^n + \frac{1}{2}\sum_{n\geq 0}|\mathrm{Fix}_{\tilde{\mathcal{A}}^{\Delta}_{o,n}}(\tau)|y^n, \tag{55}$$

where $|\mathrm{Fix}_{\tilde{\mathcal{A}}^{\Delta}_{o,n}}(\tau)|$, the number of τ-symmetric $\mathcal{A}^{\Delta}$-structures over n edges has to be determined. As shown in Figure 8, such a structure must have an axis of symmetry which coincides with one of the root triangle's medians. Since the structure is already considered up to rotation around the root triangle, the choice among the three possible axes is arbitrary. The base side of the triangle must be a $\mathcal{B}_{\mathrm{Sym}}$-structure while the two other sides must be isomorphic copies of the same $\mathcal{B}$-structure. Therefore,

$$\widetilde{\mathcal{A}}^{\Delta}(y) = \frac{1}{2}\widetilde{\mathcal{A}}_o^{\Delta}(y) + \frac{1}{2}\mathcal{B}_{\mathrm{Sym}}(y)\mathcal{B}(y^2). \tag{56}$$

In a very similar way, since $\mathcal{A}^{\underline{\Delta}}=\mathcal{A}_o^{\underline{\Delta}}/\mathbb{Z}_2$, we obtain

$$\widetilde{\mathcal{A}}^{\underline{\Delta}}(y) = \frac{1}{2}\widetilde{\mathcal{A}}_o^{\underline{\Delta}}(y) + \frac{1}{2}\mathcal{B}_{\mathrm{Sym}}(y)\mathcal{B}(y^2). \tag{57}$$

Finally, using (53), (56) and (57) and using the dissymetry theorem, we get

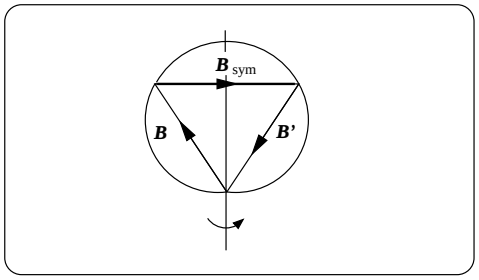


FIGURE 8. A τ-symmetric $\mathcal{A}_o^{\Delta}$-structure.

Proposition 9. The ordinary generating function of solid 2-trees is given by

$$\mathcal{A}(y) = \frac{1}{2}(\mathcal{A}_o(y) + \mathcal{B}_{\mathrm{Sym}}(y)), \tag{58}$$

where $\mathcal{B}_{\mathrm{Sym}}(y)$ is the ordinary generating series of τ-symmetric oriented $\mathcal{B}$-structures. Consequently, the number $\widetilde{\mathcal{A}}[m]$ of unoriented solid 2-trees over m triangles is given by

$$\widetilde{\mathcal{A}}[m] = \frac{1}{2}(\widetilde{\mathcal{A}}_o[m] + \mathcal{B}_{\mathrm{sym}}[m]), \tag{59}$$

where

$$\widetilde{\mathcal{A}_o}[m] = \frac{1}{3m}\sum_{d|m}\phi\left(\frac{m}{d}\right)\binom{3d}{d} + \chi(3|2m+1)\frac{2}{2m+1}\binom{m-1}{\frac{2m-2}{3}} - \frac{2}{3(2m+1)}\binom{3m}{m}.$$

and

$$\mathcal{B}_{\mathrm{sym}}[m] = \begin{cases} \dfrac{1}{m+1}\binom{3m/2}{m} & \text{if } m \text{ is even,} \\ \dfrac{1}{m}\binom{(3m-1)/2}{m+1} + \dfrac{1}{3m}\binom{3(m+1)/2}{m+1} & \text{if } m \text{ is odd.} \end{cases} \tag{60}$$

To express $\widetilde{\mathcal{A}}[m]$ in term of n the number of edges, we only have to set $m := \frac{n-1}{2}$.

3.2. Enumeration of non oriented solid 2-trees according to the edge degree distribution.

We consider again the weight function defined by

$$\begin{array}{rccc} w: & F[n] & \longrightarrow & Q[r_1, r_2, \ldots] \\ & s & \mapsto & w(s), \end{array} \tag{61}$$

where $r = (r_0, r_1, r_2, \ldots)$ is an infinite set of formal variables, F is any species and n is any positive integer.

• Labelled case

As mentioned in the previous section, the only labelled solid 2-tree left fixed under the action of τ consists in a single edge. Hence, given a valid edge degree distribution $\vec{n}$ we have

$$\mathcal{A}[\vec{n}] = \begin{cases} \frac{1}{2}\mathcal{A}_0[\vec{n}] & \text{if } n > 1; \\ 1 & \text{if } n = 1, \end{cases} \tag{62}$$

where n is the number of edges and $\mathcal{A}[\vec{n}] = [y^n][r_1^{n_1} r_2^{n_2} \cdots]\mathcal{A}_w^-(y)$.

• Unlabelled case

Using the weighted versions of equations (53), (56) and (57), we get

$$\widetilde{\mathcal{A}}^{-}_{w}(y) = \frac{1}{2}\widetilde{\mathcal{A}}^{\rightarrow}_{o,w}(y) + \frac{1}{2}\mathcal{B}_{\mathrm{sym},w}(y), \tag{63}$$

$$\widetilde{\mathcal{A}}^{\Delta}_{w}(y) = \frac{1}{2}\widetilde{\mathcal{A}}^{\Delta}_{o,w}(y) + \frac{1}{2}\mathcal{B}_{\mathrm{sym},w}(y)\mathcal{B}_{w}(y^2), \tag{64}$$

$$\widetilde{\mathcal{A}}^{\underline{\Delta}}_{w}(y) = \frac{1}{2}\widetilde{\mathcal{A}}^{\underline{\Delta}}_{o,w}(y) + \frac{1}{2}\mathcal{B}_{\mathrm{sym},w}(y)\mathcal{B}_{w}(y^2). \tag{65}$$

Now applying the dissymmetry theorem leads to

$$\widetilde{\mathcal{A}}(y) = \frac{1}{2}\widetilde{\mathcal{A}}_{o,w}(y) + \frac{1}{2}\mathcal{B}_{\mathrm{sym},w}(y). \tag{66}$$

The only unknown term in the above equation is $\mathcal{B}_{\mathrm{sym},w}(y)$. We first establish an additional condition on the vertex degree distribution for an edge rooted oriented solid 2-tree to be τ-symmetric. Since the root edge must remain fixed and all other edges are exchanged pairwise, the edge degree distribution vector $\vec{n}$ must have all its components even except one odd corresponding to the rooted edge.

For an edge degree distribution $\vec{n} = (n_1, n_2, \ldots)$ satisfying the previous condition, and using the fact that $\mathcal{B}_{\mathrm{sym,w}}(y) = yr_k\mathcal{B}^k(y^2)$, we have

$$\mathcal{B}_{\mathrm{sym,w}}[\vec{n}] = \frac{2k}{n-1}\binom{\frac{n-1}{2}}{\frac{\vec{n}-\delta_k}{2}}, \tag{67}$$

where k is the root edge degree. We now present the final result of this paper.

Proposition 10. Let $\vec{n}$ be a vector satisfying

$$\sum_i n_i = n \quad \text{and} \quad \sum_i i n_i = 3m.$$

Then, the number $\widetilde{\mathcal{A}}[\vec{n}]$ of (non oriented) unlabelled solid 2-trees having $\vec{n}$ as edge degree distribution is given by

$$\widetilde{\mathcal{A}}[\vec{n}] = \frac{1}{2}\widetilde{\mathcal{A}}_o[\vec{n}] + \frac{1}{2}\widetilde{\mathcal{B}}_{\mathrm{sym}}[\vec{n}], \tag{68}$$

where

$$\widetilde{\mathcal{B}}_{\mathrm{sym}}[\vec{n}] = \begin{cases} \dfrac{2k}{n-1}\binom{\frac{n-1}{2}}{\frac{\vec{n}-\delta_k}{2}}, & \text{if } \vec{n} \text{ has a unique odd component,} \\ 0, & \text{otherwise,} \end{cases}$$

δ_k being the vector having 1 at the k^{th} component and 0 everywhere else, and

$$\widetilde{\mathcal{A}_o}[\vec{n}] = \frac{2}{n-1}\sum_{k\in \mathrm{Supp}(\vec{n})}\sum_{d\mid\{k,\vec{n}-\delta_k\}}\phi(d)\binom{\frac{n-1}{d}}{\frac{\vec{n}-\delta_k}{d}} + \frac{\chi(3|\vec{n})}{n}\binom{n/3}{n_1/3, n_2/3, \ldots} - \frac{2}{3n}\binom{n}{n_1, n_2, \ldots}.$$

Appendix.

To conclude this paper, we give here two tables giving the numbers of unlabelled solid 2-trees oriented and unoriented as well as the number of unlabelled τ-symmetric $\mathcal{B}$-structures. The first table gives these numbers according to the number n of edges, and the second, according to edge degree distribution. We use the notation $1^{n_1}2^{n_2}\cdots$, where i^{n_i} means n_i edges of degree i.

n	$\widetilde{\mathcal{A}}_o[n]$	$\mathcal{B}_{\mathrm{sym}}[n]$	$\widetilde{\mathcal{A}}[n]$
1	1	1	1
3	1	1	1
5	1	1	1
7	2	2	2
9	7	3	5
11	19	7	13
13	86	12	49
15	372	30	201
17	1825	55	940
19	9143	143	4643
21	47801	273	24037

$\vec{n}$	$\widetilde{\mathcal{A}}_o[\vec{n}]$	$\mathcal{B}_{\mathrm{sym}}[\vec{n}]$	$\widetilde{\mathcal{A}}[\vec{n}]$
$1^7 2^1 3^1$	2	0	1
$1^8 2^2 3^1$	9	3	6
$1^{12} 2^1 3^1 4^1$	46	0	23
$1^{10} 5^1$	3	1	2
$1^{15} 4^1 5^1$	2	0	1
$1^{16} 3^2 5^1$	17	5	11
$1^{15} 2^2 7^1$	34	0	17

References

[1] L. Beineke and R. Pippert, *The number of labeled k-dimensional trees*, Journal of Combinatorial Theory **6**, 200–205, (1969).

[2] F. Bergeron, G. Labelle, and P. Leroux, *Combinatorial Species and tree-like structures*, Encyclopedia of Mathematics and its Applications, vol. 67, Cambridge University Press, (1998).

[3] M. Bòna, M. Bousquet, G. Labelle and P. Leroux, *Enumeration of m-ary cacti*, Adv. in Appl. Math, **24**, 22–56, (2000).

[4] M. Bousquet, *Espèces de structures et applications au dénombrement de cartes et de cactus planaires*, Thèse de doctorat, UQÀM (1998). Publications du LaCIM, Vol. 24 (1999).

[5] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *Specifying 2-trees*, Proceedings FPSAC'00, Moscow, 26-30 juin 2000, 202-213.

[6] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *The Specification of 2-trees*, Advances in Applied Mathematics, to appear.

[7] F. Harary and E. Palmer, *Graphical Enumeration*, Academic Press, New York, (1973).

[8] G. Labelle, C. Lamathe and P. Leroux, *Développement moléculaire de l'espèce Qdes 2-arbres planaires*, Proceedings GASCom 01, 41–46, (2001).

[9] G. Labelle and P. Leroux, *Enumeration of (uni- or bicolored) plane trees according to their degree distribution*, Discrete Math. **157**, 227–240, (1996).

[10] E. Palmer and R. Read, *On the Number of Plane 2-trees*, J. London Mathematical Society **6**, 583-592, (1973).

[11] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, (1995). `http://www.research.att.com/~njas/sequences`

E-mail address: `[bousq2,lamathe]@math.uqam.ca`

Lacim, Département de Mathématiques,, Université du Québec à Montréal.

Prüfziffersysteme über Quasigruppen

H. Michael Damm

März 1998

Diplomarbeit
am Fachbereich Mathematik und Informatik der
Philipps-Universität Marburg

Betreuer: Prof. Dr. H. Peter Gumm
Zweitgutachter: Prof. Dr. A. Dressler

Prüfziffersysteme über Quasigruppen

Zusammenfassung

Der Begriff *Prüfziffersystem* wurde von H.P. Gumm 1985 eingeführt. Wir untersuchen Prüfziffersysteme über Gruppen und Quasigruppen. Zu jeder Ordnung größer als zwei existiert ein Prüfziffersystem, das alle Einzelfehler und alle Nachbarvertauschungen erkennt. Für den wichtigen Spezialfall der Prüfziffersysteme über Gruppen der Ordnung 10 zeigen wir allerdings, daß diese nicht alle Zwillings-, Sprungzwillingsfehler oder Sprungtranspositionen erkennen.

Bei den Prüfziffersystemen über Quasigruppen werden wir sehen, daß verschiedene Ansätze das Problem ebenfalls nicht lösen können. Dennoch werden wir ein Prüfziffersystem zur Basis 10 angeben, das eine Fehlererkennung von 99,89% aller nicht zufälligen Fehler aufweist.

Inhaltsverzeichnis

Einleitung

Prüfziffern sind unscheinbar und allgegenwärtig. Sie werden von Banken benutzt, um falsch erfaßte Kontonummern oder Bankleitzahlen zu erkennen, der Buchhandel spürt mit ihrer Hilfe falsche ISBN-Nummern auf und schließlich bemerkt der Laserscanner an den Kassen im Supermarkt anhand einer falschen Prüfziffer, daß er den Strichcode der Artikelnummer falsch eingelesen hat.

Die grundlegende Idee besteht darin, daß man aus der vorgegebenen Zahl eine weitere Ziffer berechnet, welche in die Zahl eingebaut wird. Diese Prüfziffer wird so bestimmt, daß Eingabe und Übertragungsfehler erkannt werden können. Dabei wird die errechnete Ziffer mit der Prüfziffer verglichen. Stimmen diese nicht überein, dann wurde die ursprüngliche Zahl verfälscht. Die Prüfziffer wird in der Praxis fast immer an die zu sichernde Zahl angehängt, grundsätzlich spricht aber nichts dagegen, sie in der Mitte einzufügen oder sie an den Anfang zu stellen.

Fehlerstatistik

Um die Qualität eines Prüfzifferverfahrens beurteilen zu können, muß man natürlich zuerst die Art der möglichen Eingabefehler sowie deren Häufigkeit feststellen. Verhoeff [27] hat Ende der sechziger Jahre eine entsprechende Untersuchung mit 6-stelligen Zahlen durchgeführt. Dabei wurde deutlich, daß die sogenannten Einzelfehler, d.h. eine falsch eingegebene Ziffer, am häufigsten vorkommt (siehe Tabelle). Bereits mit großem Abstand folgt die zweithäufigste Fehlerart, nämlich die Vertauschung zweier benachbarter Ziffern (Zahlendreher), vor den anderen möglichen Fehlern.

Fehlerart		Symbol	Häufigkeit
1.	eine falsche Ziffer (Einzelfehler)	$x \to y$	79,0 %
2.	Nachbarvertauschung (Vertauschung einer Ziffer mit der nächsten)	$xy \to yx$	10,2 %
3.	Sprungtransposition (Vertauschung einer Ziffer mit der übernächsten)	$xzy \to yzx$	0,8 %
4.	Zwillingsfehler	$xx \to yy$	0,6 %
5.	phonetische Fehler ($a = 2, ..., 9$)	$a0 \leftrightarrow 1a$	0,5 %
6.	Sprung-Zwillingsfehler	$xzx \to yzy$	0,3 %
7.	sonstige/zufällige Fehler	-	8,6 %

Phonetische Fehler entstehen durch die Verwechslung ähnlich klingender Zahlen, zum Beispiel von „fünfzig" und „fünfzehn". Die Anzahl der Fehler dieses Fehlertyps hängt natürlich von der Sprache ab, d.h. VERHOEFFS Untersuchung gilt genaugenommen nur für die holländische und ähnliche Sprachen wie Deutsch und Englisch.

Im Deutschen existiert eigentlich noch eine weitere Klasse phonetischer Fehler, denn die Zahl 35 (fünf-und-dreißig) wird häufiger mit der 53 verwechselt als z.B. im Englischen (thirty-five). Diese Fehler werden allerdings schon durch die Nachbarvertauschungen abgedeckt und müssen daher nicht gesondert betrachtet werden.

In der Klasse der sonstigen und zufälligen Fehler befinden sich alle sehr seltenen Fehlerarten, wie z.B. $xyx \to yxy$, $wxyz \to xwzy$ oder $xxxx \to yyyy$, sowie Fehler, bei denen kein offensichtlicher Zusammenhang zwischen der korrekten und der fehlerhaften Zahl besteht. Auch wenn kein offensichtlicher Zusammenhang zwischen den Zahlen besteht, kann es trotzdem eine versteckte Verbindung geben, z.B. könnten beide Zahlen zur selben Person gehören, eine ist seine Telefonnummer und die andere seine Kontonummer. Eine andere Möglichkeit besteht darin, daß die korrekte Kundennummer eines anderen Kunden eingegeben wird. Es ist unmöglich jeden dieser Fehler zu erkennen, man kann allerdings erwarten, daß durch die Redundanz des Codes (Zahl + Prüfziffer) eine große Anzahl der zufälligen Fehler erkannt wird.

Eine Fehlerklasse, die nicht weiter betrachtet wird, bildet das Einfügen oder Weglassen einzelner Ziffern. Die Häufigkeit dieser Fehler liegt zwischen zehn und zwanzig Prozent, wobei die letzte Ziffer und die 0 am häufigsten betroffen sind. Es ist also nicht sinnvoll, fehlende Nullen nach der Eingabe automatisch zu ergänzen. Ansonsten fallen fehlende Ziffern anhand der abweichenden Stellenzahl auf.

Mit zunehmender Stellenzahl nimmt sowohl die relative als auch die absolute Häufigkeit von (Mehrfach-)Fehlern zu. Da VERHOEFF die Fehlerstatistik mit sechsstelligen Zahlen ermittelt hat, können die ermittelten Zahlen im allgemeinen nur als Anhaltswerte angesehen werden. So kann, gemäß VERHOEFF, die

Fehlerhäufigkeit von Doppelfehlern (d.h. die Summe der Fehler 2-6) durchaus zwischen zehn und zwanzig Prozent schwanken. Die Fehler verteilen sich auch nicht gleichmäßig auf die einzelnen Stellen, vielmehr sind die letzten beiden Stellen im Vergleich mit den anderen etwa doppelt so häufig betroffen.

Ein weiterer Faktor, der sowohl die absolute als auch die relative Fehlerhäufigkeit beeinflußt, ist die Art der Datenübertragung. Bei der Übermittlung per Telefon ist sicherlich eine andere Fehlerverteilung zu erwarten, als beim Übertragen von hand- oder maschinengeschriebenen Texten.

Kapitel 1

Prüfziffersysteme über Gruppen

In diesem Kapitel werden Prüfziffersysteme untersucht, die auf einer vorgegebenen Gruppe basieren. Die einfachsten Verfahren sind dabei solche, die auf den Restklassenringen $\mathbb{Z}_{10}$,$\mathbb{Z}_{11}$ usw. beruhen, die sogenannten Modulo-Verfahren. Wir werden sehen, daß diese einige Nachteile besitzen und daher in den meisten Fällen in der Praxis nicht benutzt werden sollten. Aus diesem Grund werden wir Prüfziffersysteme basierend auf anderen Gruppen untersuchen. Da es nur zwei Gruppen der Ordnung 10 gibt, nämlich $\mathbb{Z}_{10}$ und die Diedergruppe D_5, sind Prüfziffersysteme basierend auf einer Diedergruppe von besonderem Interesse.

1.1 Modulo-Verfahren

Das einfachste Prüfverfahren für Zahlen mit den Ziffern 0 bis 9 besteht darin, alle Ziffern zu addieren (also die Quersumme zu bilden) und dann den 10er Rest als Prüfziffer anzuhängen. Für die Zahl $x_m x_{m-1} \dots x_1$ mit den Ziffern $x_i \in \mathbb{Z}_{10}$ wird also die Prüfziffer x_0 berechnet durch

$$x_0 \equiv x_m + x_{m-1} + \ldots + x_1 \pmod{10}$$

oder, wenn man + als Gruppenoperation von $\mathbb{Z}_{10}$ ansieht,

$$x_0 = x_m + x_{m-1} + \ldots + x_1.$$

Dieses Verfahren erkennt alle Einzelfehler, da sich beim Ändern einer Ziffer auch die Prüfziffer ändert. Für die Zahl 72201 erhält man z.B. die Prüfziffer 2, denn 7+2+2+0+1=12. Mit angehängter Prüfziffer würde also 722012 abgespeichert. Wenn nun später die Zahl 722212 eingegeben wird, kann man diese Zahl als fehlerhaft erkennen, denn 7+2+2+2+1=14 ergibt die Prüfziffer 4 ungleich 2. Da diese einfache Prüfsumme aber nicht von der Reihenfolge der Ziffern abhängt (Die Gruppe $\mathbb{Z}_{10}$ ist abelsch), wird leider keine einzige Vertauschung erkannt. Eine Eingabe von 272012 statt 722012 kann daher nicht als falsch erkannt werden.

Eine Erweiterung dieses Verfahrens besteht darin, die Prüfziffer nicht aus einer einfachen, sondern aus einer gewichteten Summe der einzelnen Ziffern zu berechnen, d.h.

$$x_0 = a_m x_m + a_{m-1} x_{m-1} + \ldots + a_1 x_1$$

mit $a_i \in \mathbb{Z}_{10}$.

Die Deutsche Post AG benutzt z.B. die Gewichte $a_i = 6$ falls i ungerade und $a_i = 1$ falls i gerade ist, um den Ident- und den Leitcode der Pakete zu sichern. Mit diesen Gewichten können zwar fast alle Nachbarvertauschungen erkannt werden, aber jetzt werden nicht mehr alle Einzelfehler erkannt. Da 6 nicht teilerfremd zu 10 ist, gilt $6 \cdot 5 = 6 \cdot 0$, d.h. es werden an allen ungeraden Positionen Verwechslungen von 5 mit 0, 1 mit 6 und so weiter nicht erkannt.

Auch die Wahl anderer Gewichte führt nicht dazu, daß sowohl alle Einzelfehler als auch alle Nachbarvertauschungen erkannt werden. Um die Einzelfehler erkennen zu können, müssen die Gewichte teilerfremd zu 10 sein. Dies führt aber dazu, daß $(a_i - a_{i-1})$ gerade ist, also ist $(a_i - a_{i-1})$ ein Nullteiler im Ring $\mathbb{Z}_{10}$ und alle Vertauschungen der Form $x_m \ldots x_i x_{i-1} \ldots x_1 \to x_m \ldots x_{i-1} x_i \ldots x_1$ bleiben unerkannt, wenn $(x_i - x_{i-1}) \equiv 5$ (modulo 10).

Auch mit einem noch allgemeineren Ansatz, bei dem statt der Multiplikation mit einem Element a_i eine Permutation auf die einzelnen Ziffern angewendet wird, kann das Problem nicht gelöst werden, denn im Abschnitt „Prüfziffersysteme über abelschen Gruppen" werden wir zeigen, daß über der Gruppe $\mathbb{Z}_{10}$ kein Prüfzifferverfahren existiert.

Die Notwendigkeit, daß sowohl die Gewichte, als auch die Differenzen benachbarter Gewichte Einheiten sein müssen, führt auf den Gedanken, eine Primzahl als Modulus zu benutzen. Die zur 10 nächste Primzahl ist die 11, so daß beim Rechnen in der Gruppe $\mathbb{Z}_{11}$ die Schwierigkeiten bei der Suche nach geeigneten Gewichten zur Fehlererkennung nicht auftreten. Es reicht vielmehr aus, daß benachbarte Gewichte verschieden sind und im Bereich von 1 bis 10 liegen, um alle Einzelfehler und Nachbarvertauschungen zu erkennen.

Ein bekanntes Beispiel einer Modulo-11-Prüfung stellen die Internationalen Standard Buchnummern (ISBN) dar. Eine ISBN hat zehn Ziffern $x_{10} \ldots x_1$ und setzt sich aus vier Abschnitten zusammen, von denen der erste das Land, der zweite den Verlag und der dritte das Buch kennzeichnet. Zuletzt folgt eine Prüfziffer, x_1, die durch die Gleichung

$$10x_{10} + 9x_9 + 8x_8 + \ldots + 2x_2 + x_1 = 0$$

bestimmt wird. Eine gültige ISBN ist z.B. 3-411-04011-4, beim Nachrechnen erhält man: $3 \cdot 10 + 4 \cdot 9 + 1 \cdot 8 + \ldots + 1 \cdot 1 + 4 = 110 \equiv 0 \pmod{11}$.

Das Modulo-11-Verfahren mit den Gewichten 2^i erkennt sogar alle nicht zufälligen Fehler, da die 2 eine primitive zehnte Einheitswurzel ist (vgl. Verhoeff [27]).

Ein gravierender Nachteil bei den Modulo-11-Verfahren ist, daß beim Rechnen der Rest (die Prüfziffer) 10 heraus kommen kann. Es gibt verschiedene Möglichkeiten, mit diesem Problem umzugehen. Man kann etwa bei einem Rest von 10 ein nicht-numerisches Zeichen als Ersatz nehmen. So wird z.B. bei den ISBN-Prüfziffern ein 'X' als elfte Ziffer benutzt. Eine weitere Möglichkeit besteht darin, alle Zahlen, bei denen als Prüfziffer die 10 entsteht, nicht zu verwenden. Laut ECKER und POCH [9] verfährt die Dresdner Bank auf diese Weise.

Im Normalfall sollen die Prüfziffern allerdings aus den gleichen Ziffern bestehen, wie die zu sichernde Zahl. Häufig möchte man auch nicht auf eine fortlaufende Vergabe der Zahlen verzichten, abgesehen davon, daß die Redundanz durch das Weglassen einiger Zahlen deutlich erhöht wird. In den meisten Fällen ist daher das Modulo-11-Verfahren unbrauchbar. Als Alternative bietet sich die zweite Gruppe mit 10 Elementen an, nämlich die Diedergruppe. Prüfzifferverfahren basierend auf Diedergruppen bieten ebenfalls eine sehr gute Fehlererkennung, z.B. werden die Seriennummern deutscher Banknoten mit diesen gesichert. Wir behandeln diese im Kapitel „Gruppen mit Vorzeichen".

1.2 Verallgemeinerung auf beliebige Gruppen

Bei den Modulo-Verfahren wird von den Restklassenringen $\mathbb{Z}_{10}$, $\mathbb{Z}_{11}$ usw. im wesentlichen nur die additive Gruppe benötigt. Die Multiplikation dient nur dazu, eine Permutation der Ziffern zu erzeugen. Für beliebige Gruppen ist es daher sinnvoll, folgende Definition zu treffen (vergleiche SCHULZ [21]):

Definition 1 *Sei $(G, \cdot)$ eine endliche Gruppe der Ordnung n und $m \geq 2$ eine fest gewählte ganze Zahl. Dann ist ein Prüfziffersystem über der Gruppe G definiert durch ein Element $c \in G$ und $m+1$ Permutationen $\tau_m, \ldots, \tau_0$ der Grundmenge G, mit der Eigenschaft $\tau_i \circ \tau_{i-1}^{-1}(x) \cdot y = \tau_i \circ \tau_{i-1}^{-1}(y) \cdot x \Rightarrow x = y$, für $i = 1, \ldots, m$ und alle $x, y \in G$. Zu jeder Zahl $x_m x_{m-1} \ldots x_1$ wird eine Prüfziffer x_0 hinzugefügt, welche die Kontrollgleichung*

$$\tau_m(x_m) \cdot \tau_{m-1}(x_{m-1}) \cdot \ldots \cdot \tau_1(x_1) \cdot \tau_0(x_0) = c$$

erfüllt.

Lemma 1 *1. Für gegebene $x_m, x_{m-1}, \ldots, x_1$ ist die Prüfziffer x_0 eindeutig durch die Kontrollgleichung bestimmt.*

2. Jedes Prüfziffersystem über einer Gruppe erkennt alle Einzelfehler und alle Nachbarvertauschungen.

Beweis zu 1: Da τ_0 eine Permutation ist, ist die Kontrollgleichung eindeutig nach x_0 auflösbar:

$$x_0 = \tau_0^{-1}(\tau_1(x_1)^{-1} \cdot \ldots \cdot \tau_{m-1}(x_{m-1})^{-1} \cdot \tau_m(x_m)^{-1} \cdot c).$$

zu 2: Wenn wir annehmen, daß sowohl $x_m \ldots x_i \ldots x_0$ als auch $x_m \ldots x_i' \ldots x_0$ die Kontrollgleichung erfüllt, dann folgt $\tau_m(x_m) \cdot \ldots \cdot \tau_i(x_i) \cdot \ldots \cdot \tau_0(x_0) = c = \tau_m(x_m) \cdot \ldots \cdot \tau_i(x_i') \cdot \ldots \cdot \tau_0(x_0)$. Nun können sowohl links als auch rechts gleiche Elemente gekürzt werden und es folgt $\tau_i(x_i) = \tau_i(x_i')$ und damit $x_i = x_i'$. Für $x_i \neq x_i'$ können also nicht beide Zahlen $x_m \ldots x_i \ldots x_0$ und $x_m \ldots x_i' \ldots x_0$ die Kontrollgleichung erfüllen, es werden somit alle Einzelfehler erkannt.

Ebenso zeigen wir, daß alle Vertauschungen benachbarter Elemente erkannt werden. Gilt nämlich $\tau_m(x_m) \cdot \ldots \cdot \tau_i(x_i) \cdot \tau_{i-1}(x_{i-1}) \cdot \ldots \cdot \tau_0(x_0) = c = \tau_m(x_m) \cdot \ldots \cdot \tau_i(x_{i-1}) \cdot \tau_{i-1}(x_i) \cdot \ldots \cdot \tau_0(x_0)$, so folgt, nach kürzen der gleichen Elemente auf beiden Seiten, $\tau_i(x_i) \cdot \tau_{i-1}(x_{i-1}) = \tau_i(x_{i-1}) \cdot \tau_{i-1}(x_i)$. Wir setzen $y_{i-1} := \tau_{i-1}(x_{i-1})$ und $y_i := \tau_{i-1}(x_i)$, womit $\tau_i(\tau_{i-1}^{-1}(y_i)) \cdot y_{i-1} = \tau_i(\tau_{i-1}^{-1}(y_{i-1})) \cdot y_i$ folgt. Nach Voraussetzung ist damit $y_i = y_{i-1}$, also $\tau_{i-1}(x_{i-1}) = \tau_{i-1}(x_i)$ und $x_{i-1} = x_i$. Folglich werden alle Nachbarvertauschungen erkannt. □

Bemerkung Es ist für die Erkennung aller Einzelfehler erforderlich, daß die τ_i Permutationen sind. Ebenso ist die Forderung $\tau_i \circ \tau_{i-1}^{-1}(x) \cdot y = \tau_i \circ \tau_{i-1}^{-1}(y) \cdot x \Rightarrow x = y$ nicht nur hinreichend, sondern auch notwendig für die Erkennung aller Nachbarvertauschungen. Gibt es nämlich ein i und $x \neq y$ mit $\tau_i \circ \tau_{i-1}^{-1}(x) \cdot y = \tau_i \circ \tau_{i-1}^{-1}(y) \cdot x$, dann gilt für $x_i := \tau_{i-1}^{-1}(x)$ und $x_{i-1} := \tau_{i-1}^{-1}(y)$ die Gleichung $\tau_i(x_i) \cdot \tau_{i-1}(x_{i-1}) = \tau_i(x_{i-1}) \cdot \tau_{i-1}^{-1}(x_i)$ und $x_i \neq x_{i-1}$. Damit erfüllen aber die Zahlen $x_m \ldots x_i x_{i-1} \ldots x_0$ und $x_m \ldots x_{i-1} x_i \ldots x_0$ die Kontrollgleichung, d.h. es werden nicht alle Nachbarvertauschungen erkannt.

Für weitere Fehlertypen findet man folgende Bedingungen, die für alle $x, y, z \in G$ und alle i erfüllt sein müssen:

Fehlertyp	Bedingungen für die Fehlererkennung
Sprungtranspositionen	$\tau_{i+1} \circ \tau_{i-1}^{-1}(x) \cdot z \cdot y = \tau_{i+1} \circ \tau_{i-1}^{-1}(y) \cdot z \cdot x$ impliziert $x = y$
Zwillingsfehler	$\tau_i \circ \tau_{i-1}^{-1}(x) \cdot x = \tau_i \circ \tau_{i-1}^{-1}(y) \cdot y$ impliziert $x = y$
Sprungzwillingsfehler	$\tau_{i+1} \circ \tau_{i-1}^{-1}(x) \cdot z \cdot x = \tau_{i+1} \circ \tau_{i-1}^{-1}(y) \cdot z \cdot y$ impliziert $x = y$
phonetische Fehler	Für $a = 2, \ldots, n-1$ gilt $\tau_{i+1}(a)\tau_i(0) \neq \tau_{i+1}(1)\tau_i(a)$

Die Bedingungen werden ähnlich wie im obigen Lemma gezeigt. Wir verzichten daher auf einen Beweis.
Da diese Fehlertypen nur sehr selten auftauchen, werden wir uns im folgenden vorrangig mit dem Erkennen der Einzelfehler und der Nachbarvertauschungen beschäftigen. Wie wir sehen, spielen die Permutationen φ, bei denen aus $\varphi(x)\cdot y = \varphi(y)\cdot x$ die Gleichheit von x und y folgt, eine wichtige Rolle. Diese werden anti-symmetrisch genannt (vgl. Kapitel 2). Sie sind erforderlich für die Existenz eines Prüfziffersystems über einer Gruppe. Andererseits kann man mit ihnen auch ein Prüfziffersystem definieren.

Satz 1 *(vgl.* H.P. GUMM [12]*) Sei φ eine anti-symmetrische Permutation der Gruppe G, dann wird durch $\tau_i := \varphi^i$, ein beliebiges Element $c \in G$ sowie der Kontrollgleichung*

$$\varphi^m(x_m)\cdot\varphi^{m-1}(x_{m-1})\cdot\ldots\cdot\varphi(x_1)\cdot x_0 = c$$

ein Prüfziffersystem definiert.

Beweis Es ist $\tau_i \circ \tau_{i-1}^{-1} = \varphi^i \circ \varphi^{-i+1} = \varphi$ und φ erfüllt nach Voraussetzung die geforderte Bedingung. □

1.3 Prüfziffersysteme über abelschen Gruppen

In abelschen Gruppen stehen die anti-symmetrischen Abbildungen in direkter Beziehung zu den von MANN [15] 1942 eingeführten vollständigen Abbildungen. Eine Permutation φ heißt vollständig, wenn $x\cdot\varphi(x) = y\cdot\varphi(y)$ impliziert, daß $x = y$ ist (also wenn $x\cdot\varphi(x)$ wieder eine Permutation ist). Mit Hilfe der vollständigen Abbildungen ist es möglich, orthogonale lateinische Quadrate zu konstruieren.

Lemma 2 *Eine abelsche Gruppe $(G,+)$ besitzt eine vollständige Abbildung genau dann, wenn sie eine anti-symmetrische Abbildung besitzt.*

Beweis Es gilt für alle $x, y \in G$: $\varphi(x) + y = \varphi(y) + x \Leftrightarrow x - \varphi(x) = y - \varphi(y)$. Damit folgt, wenn inv die Abbildung $x \mapsto -x$ bezeichnet,

$$\varphi \text{ anti-symmetrisch} \Leftrightarrow inv \circ \varphi \text{ vollständig}$$

und

$$\varphi \text{ vollständig} \Leftrightarrow inv \circ (inv \circ \varphi) \text{ vollständig} \Leftrightarrow inv \circ \varphi \text{ anti-symmetrisch.}$$ □

Die Frage, wann eine endliche abelsche Gruppe eine vollständige Abbildung besitzt, wurde von PAIGE 1947 gelöst.

Theorem 1 *(*PAIGE [18]*) Eine endliche abelsche Gruppe der Ordnung n besitzt eine vollständige und damit eine anti-symmetrische Abbildung genau dann, wenn n ungerade ist oder wenn G mindestens zwei verschiedene Involutionen enthält (also die 2-Sylowgruppe von G nicht zyklisch ist).*

Beweis 1) Falls n ungerade ist, dann ist $\varphi = (x \mapsto 2x)$ eine anti-symmetrische Permutation, denn aus $2x = 2y$ oder aus $x + x + y = y + y + x$ folgt direkt $x = y$. 2) Der Fall n gerade wird konstruktiv bewiesen. Um den Beweis des Theorems zu vereinfachen, zeigen wir allerdings zunächst einige Lemmata.

Im folgenden sei $n = n(G)$ die Ordnung der Gruppe G und die Summe aller Elemente der Gruppe werde mit $p = p(G)$ bezeichnet, d.h.

$$p(G) = \sum_{x \in G} x.$$

Weiterhin sei δ eine Permutation von G und $\eta = (x \mapsto x + \delta(x))$ eine abgeleitete Abbildung. Die Ordnung von η, bezeichnet mit $O(\eta)$, sei die Anzahl der verschiedenen Elemente $\eta(x)$, für $x \in G$.

Lemma 3 *Wenn G nicht genau ein Element der Ordnung 2 besitzt, dann ist $p(G) = 0$, ansonsten ist $p(G)$ das einzige Element der Ordnung 2.*

Beweis Sei S die eindeutig bestimmte Untergruppe, die aus dem neutralen Element und allen Elementen der Ordnung 2 der Gruppe G besteht. Wenn die Ordnung von $a \in G$ größer als 2 ist, dann ist $a \neq -a$ und deshalb kommen a und $-a$ in der Summe $p(G)$ vor, folglich gilt $p(G) = p(S)$.

Hat S die Ordnung 1, dann ist $p(S) = 0$. Hat S die Ordnung 2, d.h. $S = \{0, g\}$, dann ist $p(S) = 0 + g = g$ und $p(S)$ ist das einzige Element von S (und damit auch von G) der Ordnung 2.

Es bleibt der Fall, daß die Ordnung von S größer als 2 ist. Dann hat S die Ordnung 2^k, $k > 1$ und die k Erzeugenden $g_1, \ldots, g_k$. Jedes Element von S hat eine eindeutige Darstellung der Form $n_1 g_1 + n_2 g_2 + \ldots + n_k g_k$ mit $n_i \in \{0, 1\}$. Folglich ist $p(S) = \sum(n_1 g_1 + n_2 g_2 + \ldots + n_k g_k)$, wobei über die verschiedenen Tupel $(n_1, \ldots, n_k)$ mit $n_i \in \{0, 1\}$ summiert wird. Es gibt 2^k solche Tupel, wobei an jeder Position der Wert 0 genau $2^k/2 = 2^{k-1}$-mal vorkommt. Also ist $p(S) = 2^{k-1} \cdot (g_1 + \ldots + g_k)$ und weil $k > 1$ ist, erhalten wir $p(S) = 0$. □

Lemma 4 *Eine notwendige Bedingung für $O(\eta) = n(G)$ ist, daß $p(G) = 0$.*

Korollar 1 *Wenn $p(G) \neq 0$ ist, dann ist $O(\eta) < n(G)$ für alle Permutationen δ.*

Beweis Angenommen, es existiert eine Permutation δ mit $O(\eta) = n(G)$, d.h. η ist ebenfalls eine Permutation. Die Elemente von G werden mit x_i bezeichnet $(i = 1, 2, \ldots, n)$. Es ist

$$\sum_{i=1}^{n} \eta(x_i) = \sum_{i=1}^{n} (x_i + \delta(x_i)) = \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} \delta(x_i)$$

und es folgt, da η und δ bijektiv sind, $p = p + p$ bzw. $p = 0$. □

Lemma 5 *Wenn für ein δ $O(\eta) \leq n - 2$, wobei $n = n(G)$, dann existiert ein δ' mit $O(\eta') > O(\eta)$.*

Korollar 2 *Es existiert ein δ mit $O(\eta) \geq n(G) - 1$.*

Beweis Sei δ eine Permutation für die $O(\eta) = r \leq n - 2$ gilt. Die Elemente von G werden mit x_i, $i = 1, \ldots, n$, bezeichnet, dabei seien $\eta(x_i)$, $i = 1, \ldots, r$ die r verschiedenen Elemente von $\eta(x)$ mit $x \in G$. Existieren $h, k > r$ mit $x_h + \delta(x_k) \neq \eta(x_i)$ für alle $i \leq r$, dann wird das Problem gelöst durch $\delta'(x_h) := \delta(x_k)$, $\delta'(x_k) := \delta(x_h)$ und $\delta'(x) := \delta(x)$ sonst. Also nehmen wir an, daß dies nicht der Fall sei.

Da $\eta(x_{r+1}) = \eta(x_i)$ für ein $i \leq r$, können wir ohne Beschränkung der Allgemeinheit annehmen, daß $\eta(x_{r+1}) = \eta(x_1)$ ist. Ist $x_1 + \delta(x_{r+2}) \neq \eta(x_i)$, für alle $i \leq r$, dann können wir $\delta'(x_1) := \delta(x_{r+2})$, $\delta'(x_{r+2}) := \delta(x_1)$ und $\delta'(x) := \delta(x)$ sonst setzen, um ein δ' mit $O(\eta') > r$ zu konstruieren (wenigstens sind dann die Elemente $\eta'(x_1), \ldots, \eta'(x_{r+1})$ paarweise verschieden). Aber wenn $x_1 + \delta(x_{r+2}) = \eta(x_i)$ für ein $i \leq r$ gilt, dann können wir o.B.d.A. annehmen, daß $x_1 + \delta(x_{r+2}) = \eta(x_2)$ $(i \neq 1$, denn $x_1 + \delta(x_{r+2}) \neq x_1 + \delta(x_1) = \eta(x_1))$.

Es gilt $x_2 + \delta(x_1) \neq \eta(x_1), \eta(x_2)$. Wenn $x_2 + \delta(x_1) \neq \eta(x_i)$, für alle $i \leq r$, können wir δ ändern durch $\delta'(x_1) := \delta(x_{r+2})$, $\delta'(x_2) := \delta(x_1)$, $\delta'(x_{r+2}) := \delta(x_2)$ und wir erhalten ein δ' mit $O(\eta') > r$ (auch hier sind wenigstens die Elemente $\eta'(x_1), \ldots, \eta'(x_{r+1})$ paarweise verschieden). Andernfalls können wir ohne Einschränkung der Allgemeinheit annehmen, daß $x_2 + \delta(x_1) = \eta(x_3)$ ist.

In dieser Weise fahren wir fort: Nehmen wir an, wir hätten die Stelle erreicht, wo

$$x_1 + \delta(x_{r+2}) = \eta(x_2), \qquad x_{i+1} + \delta(x_i) = \eta(x_{i+2}), \qquad i = 1, 2, \ldots, k \tag{1.1}$$

gilt. Hieraus erhalten wir die Gleichungen

$$\eta(x_1) + \delta(x_{r+2}) = \eta(x_{i+1}) + \delta(x_i), \qquad i = 1, 2, \ldots, k+1. \tag{1.2}$$

Dies zeigen wir durch Induktion: Es ist $\eta(x_1) + \delta(x_{r+2}) = x_1 + \delta(x_1) + \delta(x_{r+2}) = x_1 + \delta(x_{r+2}) + \delta(x_1) = \eta(x_2) + \delta(x_1)$ und für $1 \leq j \leq k$ gilt $\eta(x_{j+1}) + \delta(x_j) = x_{j+1} + \delta(x_{j+1}) + \delta(x_j) = x_{j+1} + \delta(x_j) + \delta(x_{j+1}) = \eta(x_{j+2}) + \delta(x_{j+1})$.

Nun gilt $x_{k+2}+\delta(x_{k+1}) \neq \eta(x_i)$ für alle $i \leq k+2$, denn andernfalls folgt mit 1.2 $\eta(x_i)+\delta(x_{k+2}) = x_{k+2}+\delta(x_{k+1})+\delta(x_{k+2}) = \eta(x_{k+2})+\delta(x_{k+1}) = \eta(x_i)+\delta(x_{i-1})$, bzw. $\delta(x_{k+2}) = \delta(x_{i-1})$, was unmöglich ist, da $i \leq k+2$.

Ist $x_{k+2}+\delta(x_{k+1}) \neq \eta(x_i)$ für alle $i \leq r$, dann setzen wir $\delta'(x_1) := \delta(x_{r+2})$, $\delta'(x_{i+1}) := \delta(x_i)$, $i = 1, 2, \ldots, k+1$, $\delta'(x_{r+2}) := \delta(x_{k+2})$ und erhalten eine Permutation δ' mit $O(\eta') > r$.

Gilt dagegen $x_{k+2}+\delta(x_{k+1}) = \eta(x_i)$ für ein $i \leq r$, dann können wir o.B.d.A. annehmen, daß $i = k+3$ gilt und wir können die Gleichung $x_{k+2}+\delta(x_{k+1}) = \eta(x_{k+3})$ zu den Gleichungen 1.1 dazunehmen. In jedem Fall erreichen wir, da $O(\eta)$ endlich ist, eine Summe $x_j+\delta(x_{j-1}) \neq \eta(x_i)$ für alle $i \leq r$. Damit ist der Beweis des Lemmas abgeschlossen. Das Korollar ist offensichtlich. □

Wir zeigen nun den verbleibenden Fall des Theorems.
Sei die Ordnung von G gerade (d.h. G hat wenigstens ein Element der Ordnung 2). Besitzt G eine vollständige Abbildung, dann folgt mit Lemma 4, daß $n(G) = 0$ ist und mit Lemma 3, daß G mindestens zwei Elemente der Ordnung 2 besitzt.

Hat G wenigstens zwei Involutionen, dann ist $p = p(G) = 0$. Durch das Korollar können wir annehmen, daß eine Permutation δ existiert mit $O(\eta) \geq n-1$. Mit $\eta(x_i)$, $i = 1, \ldots, n-1$ bezeichnen wir $n-1$ Elemente, die paarweise verschieden sind und mit z das verbleibende Element der Gruppe. Dann gilt

$$\sum_{i=1}^{n-1}(x_i+\delta(x_i)) = \sum_{i=1}^{n-1}\eta(x_i).$$

Wir erhalten $p-x_n+p-\delta(x_n) = p-z$ und damit $x_n+\delta(x_n) = z$, also ist $O(\eta) = n$ und G besitzt die vollständige Abbildung δ. □

Im folgenden geben wir einige Ergebnisse von SIEMON [23] wieder:

Satz 2 *(SIEMON)*

1. *Die identische Abbildung $x \mapsto x$ einer endlichen Gruppe der Ordnung n ist genau dann vollständig, wenn n ungerade ist.*

2. *Ist $\mathbb{Z}_n$ eine zyklische Gruppe der Ordnung n, dann ist die durch $f(x) := x^k$ definierte Abbildung genau dann vollständig, wenn $ggT(k,n) = 1$ und $ggT(k+1,n) = 1$.*

Beweis zu 1) Sei n ungerade, $n = 2k+1$, dann gilt $x^{2k+2} = x = (x^{k+1})^2$ also ist x^2 surjektiv und, da G endlich ist, damit auch injektiv. Folglich ist $x \mapsto x$ eine vollständige Abbildung.

Ist dagegen n gerade, dann besitzt G ein Element a der Ordnung 2, somit ist $a^2 = e = e^2$ und x^2 ist nicht injektiv, also auch keine Permutation.

zu 2) Die Eigenschaft $ggT(k,n) = 1$ bzw. $ggT(k+1,n) = 1$ ist äquivalent zu x^k bzw. x^{k+1} injektiv. Daraus folgt die Behauptung. □

Bemerkung

1. Ist $ggT(k,n) = 1$, dann ist die Abbildung $f(x) := x^k$ ein Automorphismus von $\mathbb{Z}_n$.

2. Für n gerade gibt es kein k das die Bedingung $ggT(k,n) = ggT(k+1,n) = 1$ erfüllt, denn entweder ist k oder $k+1$ gerade.

Theorem 2 *(SIEMON) Eine Gruppe G der Ordnung $n = 4k+2$, $k \geq 1$, besitzt keine vollständige Abbildung und, falls G abelsch ist, auch keine anti-symmetrische Abbildung.*

Mit etwas mehr Theorie können wir den Beweis von SIEMON deutlich verkürzen, wir verschieben ihn daher auf den Abschnitt „Gruppen mit Vorzeichen".

Korollar 3 *Eine zyklische Gruppe $\mathbb{Z}_n$ der Ordnung n besitzt eine vollständige bzw. anti-symmetrische Abbildung genau dann, wenn n ungerade ist.*

Beweis Der Fall n ungerade wurde bereits gezeigt. Ist n gerade, dann ist $n/2$ das einzige Element der Ordnung 2 in $\mathbb{Z}_n$, also besitzt $\mathbb{Z}_n$ keine vollständige Abbildung.

Korollar 4 *Über den Gruppen $\mathbb{Z}_{2k}$, $k \geq 1$, insbesondere über $\mathbb{Z}_{10}$, existiert kein Prüfziffersystem.*
Über Gruppen der Ordnung $n = 4k+2$, $k \geq 1$ existiert kein Prüfziffersystem, das alle Zwillings- oder Sprungzwillingsfehler erkennt.

Die Gruppe $\mathbb{Z}_{10}$ eignet sich also grundsätzlich nicht dazu, ein Prüfziffersystem zu definieren. Für die Erkennung der Zwillings- und Sprungzwillingsfehler benötigen wir eine vollständige Abbildung (siehe Tabelle Seite 14, $\tau_{i-1} \circ \tau_i^{-1}$ bzw. $z \cdot \tau_{i-1} \circ \tau_{i+1}^{-1}$ sind vollständige Abbildungen), daher können diese Fehler in Gruppen der Ordnung $n = 4k+2$, insbesondere $n = 10$, nicht erkannt werden.

Für den nicht abelschen Fall ist bislang noch keine vollständige Lösung bekannt. 1950 bewies BATEMANN [2], daß alle unendlichen Gruppen eine vollständige Abbildung besitzen. Also haben alle unendlichen abelschen Gruppen eine anti-symmetrische Abbildung. HALL und PAIGE [14] haben 1955 gezeigt, daß in S_n $(n > 3)$, A_n $(n > 3)$ und auflösbaren Gruppen mit nicht-zyklischer 2-Sylowgruppe eine vollständige Abbildung existiert. Sie zeigten außerdem, daß eine endliche Gruppe mit zyklischer 2-Sylowgruppe keine vollständige Abbildung besitzt und vermuteten, daß in jeder endlichen Gruppe mit nicht-zyklischer 2-Sylowgruppe eine vollständige Abbildung existiert.

Wir werden später zeigen, daß die Diedergruppe D_5 mit 10 Elementen eine anti-symmetrische Abbildung besitzt. Nach Theorem 2 besitzt sie aber keine vollständige Abbildung, d.h. im nicht abelschen Fall unterscheiden sich die beiden Begriffe voneinander.

Zum Abschluß dieses Abschnittes sei noch angemerkt, daß man auf dem direkten Produkt $G_1 \times G_2$ der Gruppen G_1 und G_2 (nicht notwendig abelsch) mit den vollständigen Abbildungen g_1 und g_2 in natürlicher Weise eine vollständige Abbildung definieren kann.

Lemma 6 *Sind g_1, g_2 vollständige Abbildungen der Gruppen G_1 bzw. G_2, dann ist $f = (x, y) \mapsto (g_1(x), g_2(y))$ eine vollständige Abbildung von $G_1 \times G_2$.*

Beweis Daß f bijektiv ist, ergibt sich aus der Bijektivität von g_1 und g_2. Ebenso folgt aus der Vollständigkeit von g_1 und g_2, daß $(x, y) \cdot f(x, y) = (x \cdot g_1(x), y \cdot g_2(y))$ eine Permutation und damit vollständig ist. □

Kapitel 2

Anti-symmetrische Abbildungen

Eine Gruppe läßt die Definition eines Prüfziffersystems genau dann zu, wenn sie eine anti-symmetrische Abbildung besitzt (siehe Kapitel 1). In diesem Kapitel beschäftigen wir uns daher ausführlich mit der Existenz, den Eigenschaften und der Konstruktion von anti-symmetrischen Abbildungen.

Definition 2 *(*Gallian [10]*) Eine Permutation φ einer Gruppe $(G, \cdot)$ heißt anti-symmetrisch, wenn für alle $x, y \in G$ gilt*

$$\varphi(x) \cdot y = \varphi(y) \cdot x \quad \Rightarrow \quad x = y.$$

Die Menge aller anti-symmetrischen Abbildungen einer Gruppe G werde mit $Ant(G)$ bezeichnet.

Bemerkung $Ant(G)$ ist für eine Gruppe G der Ordnung $n > 1$ keine Untergruppe von S_n, da die Identität nicht anti-symmetrisch ist. Es gilt nämlich für beliebige $x \neq e$

$$Id(x) \cdot e = x \cdot e = e \cdot x = Id(e) \cdot x.$$

Aus der Eigenschaft $\varphi(x) \cdot y = \varphi(y) \cdot x \Rightarrow x = y$ folgt nicht, daß φ injektiv ist, z.B. wird sie von $\varphi(x) := e$ erfüllt. Da solche Funktionen nicht alle Einzelfehler erkennen, meinen wir mit dem Begriff „anti-symmetrische Abbildung" daher immer eine anti-symmetrische Permutation.

In der Literatur (z.B. Verhoeff [27]) wird der Begriff „anti-symmetrisch" auch für Permutationen mit der Eigenschaft $y \cdot \varphi(x) = x \cdot \varphi(y) \Rightarrow x = y$ benutzt. Diese können aber bijektiv auf die anti-symmetrischen Permutationen gemäß Definition 2 abgebildet werden:

Lemma 7 *Ist φ eine Permutation mit $y \cdot \varphi(x) = x \cdot \varphi(y) \Rightarrow x = y$, dann ist φ^{-1} anti-symmetrisch.*

Beweis Es gelte $\varphi^{-1}(x) \cdot y = \varphi^{-1}(y) \cdot x$. Wir setzen $\tilde{x} := \varphi^{-1}(x), \tilde{y} := \varphi^{-1}(y)$ und es folgt $\tilde{x} \cdot \varphi(\tilde{y}) = \tilde{y} \cdot \varphi(\tilde{x})$. Nach Voraussetzung ist damit $x = y$. □

2.1 Gruppen mit anti-symmetrischen Abbildungen

In diesem Abschnitt präsentieren wir eine Arbeit von GALLIAN und MULLIN [10], welche sich mit der Existenz anti-symmetrischer Abbildungen beschäftigt. In Kapitel 1 haben wir den direkten Zusammenhang zwischen vollständigen und anti-symmetrischen Abbildungen bei abelschen Gruppen gezeigt. Diese Arbeit überträgt die wesentlichen Ergebnisse von HALL und PAIGE [14] bzgl. vollständiger Abbildungen bei nicht-abelschen Gruppen auf anti-symmetrische Abbildungen.

2.1.1 Beispiele

Definition 3 *Unter der* Diedergruppe der Ordnung $2n$ *versteht man die Gruppe* $D_n = \{e, a, a^2, \ldots, a^{n-1}, b, ba, ba^2, \ldots, ba^{n-1}\}$, *wobei* a, b *zwei erzeugende Elemente sind, die den Relationen* $a^n = e$ *(und* $a^m \neq e$ *für* $1 \leq m < n$*),* $b^2 = e \neq b$ *und* $ab = ba^{-1}$ *genügen. (Abkürzende Schreibweise:* $D_n =< a, b | a^n = b^2 = e, ab = ba^{-1} >$*).*

Theorem 3 *Die folgenden Gruppen besitzen anti-symmetrische Abbildungen:*

1. $D_n =< a, b | a^n = b^2 = e, ab = ba^{-1} >$, $n \geq 3$ *(die Diedergruppe der Ordnung* $2n$*)*
2. $Q_n =< a, b | a^{2n} = b^4 = e, b^2 = a^n, ab = ba^{-1} >$, $n \geq 2$
   (die verallg. Quaternionengruppe der Ordnung $4n$*)*
3. $SD_n^+ =< a, b | a^{4n} = b^2 = e, ab = ba^{2n+1} >$, n *gerade (Semi-Diedergruppe der Ordnung* $8n$*)*
4. $SD_n^- =< a, b | a^{4n} = b^2 = e, ab = ba^{2n-1} >$, n *gerade (Semi-Diedergruppe der Ordnung* $8n$*)*

Beweis 1) Die folgende Abbildung ist anti-symmetrisch: Wenn n ungerade ist, sei $\varphi(a^i) = a^{2-i}$ und $\varphi(ba^i) = ba^i$. Wenn n gerade ist, d.h. $n = 2k$, wird φ definiert durch:

$$\begin{array}{llll}
\varphi(e) = b & \varphi(a) = e & & \\
\varphi(a^i) = a^{1-i} & \text{für } 2 \leq i \leq k & \varphi(a^i) = ba^{1-i} & \text{für } k+1 \leq i \leq n-1 \\
\varphi(ba^i) = a^{i+1} & \text{für } 0 \leq i \leq k-1 & \varphi(ba^i) = ba^{i+1} & \text{für } k \leq i \leq n-2 \\
\varphi(ba^{n-1}) = ba & & &
\end{array}$$

2) Wir definieren φ durch

$$\begin{aligned}
\varphi(e) &= e \qquad \varphi(a^i) = ba^{-i} \quad \text{für } 1 \le i \le n \\
\varphi(a^i) &= a^{-i} \quad \text{für } n+1 \le i \le 2n-1 \\
\varphi(ba^i) &= ba^{i+1} \quad \text{für } 0 \le i \le n-2 \\
\varphi(ba^i) &= a^{i+1} \quad \text{für } n-1 \le i \le 2n-2 \qquad \varphi(ba^{2n-1}) = b
\end{aligned}$$

3) Wir definieren φ durch

$$\begin{aligned}
\varphi(a^i) &= a^{4n-1-i} \quad \text{für } 0 \le i \le 2n-1 \\
\varphi(a^i) &= ba^{4n-1-i} \quad \text{für } 2n \le i \le 4n-1 \\
\varphi(ba^i) &= ba^{4n-i} \quad \text{für } 1 \le i \le 2n \qquad \varphi(b) = e \\
\varphi(ba^i) &= a^{4n-i} \quad \text{für } 2n+1 \le i \le 4n-1
\end{aligned}$$

4) Wir definieren φ durch

$$\begin{aligned}
\varphi(a^i) &= a^{4n-1-i} \quad \text{für } 0 \le i \le 2n-1 \\
\varphi(a^i) &= ba^{4n-1-i} \quad \text{für } 2n \le i \le 4n-1 \\
\varphi(ba^i) &= a^i \quad \text{für } 0 \le i \le 2n-1 \\
\varphi(ba^i) &= ba^i \quad \text{für } 2n \le i \le 4n-1
\end{aligned}$$

zu 1) Daß die genannte Permutation für ungerades n anti-symmetrisch ist, zeigen wir im Abschnitt „Beispiele", Seite 58. Im Fall n gerade müssen wir eine Vielzahl verschiedener Fälle unterscheiden. Dazu sei $0 \le i \le k-1$ und $k \le j \le n-2$. Wir zeigen exemplarisch $\varphi(x)y \neq \varphi(y)x$ für einige $x \neq y$.

	x	y	$\varphi(x)y$		$\varphi(y)x$
a.	e	a^{i+1}	ba^{i+1}	$\neq$	a^{-i}
b.	e	a^{j+1}	ba^{j+1}	$\neq$	ba^{-j}
c.	e	ba^i	a^i	$\neq$	a^{i+1}
d.	a^{i+1}	ba^j	$a^{-i}ba^j = ba^{j+i}$	$\neq$	$ba^{j+1}a^{i+1}$
e.	ba^j	ba^{n-1}	$ba^{j+1}ba^{n-1} = a^{n-j-2}$	$\neq$	$baba^j = a^{j-1}$

Bei b. folgt aus $j+1 = n-j$ die Gleichung $2j+1 = 2k$, Widerspruch (da n gerade). Bei d. können die beiden Seiten nur gleich sein, falls $n = 2$ ist. Dann kommt dieser Fall allerdings nicht vor, da kein j existiert mit $1 \le j \le 0$. Und schließlich folgt bei e. aus $n-j-2 = j-1$ die Gleichung $2k-1 = 2j$, Widerspruch.

Die anderen Fälle und Behauptungen können analog gezeigt werden. □

2.1.2 Existenztheoreme

Theorem 4 *Sei G eine Gruppe und a ein Element von G. Die Abbildung $\varphi(x) = x^{-1}a$ ist genau dann anti-symmetrisch, wenn a mit keinem Element der Ordnung 2 kommutiert.*

Beweis Sei $\varphi(x) = x^{-1}a$ anti-symmetrisch. Offensichtlich ist φ für jedes a eine Permutation. Es ist also ausreichend zu untersuchen für welche a gilt: $x \neq y \Rightarrow \varphi(x)y \neq \varphi(y)x$. Angenommen es existieren verschiedene x und y s.d. $\varphi(x)y = \varphi(y)x$ oder äquivalent dazu $x^{-1}ay = y^{-1}ax$ gilt. Multiplikation von links mit y und von rechts mit x^{-1} ergibt

$$(yx^{-1})a(yx^{-1}) = a. \tag{2.1}$$

Wir setzen $z := yx^{-1}$. Es folgt $a^2z = zazaz = za^2$ und mit Induktion

$$a^{2n}z = za^{2n} \qquad \text{für alle } n \tag{2.2}$$

Wenn die Ordnung von a gerade ist, z.B. $2m$, dann hat a^m die Ordnung 2 und kommutiert mit a. Wenn andererseits die Ordnung von a ungerade, d.h. $2k+1$ ist, dann folgt mit 2.2:

$$za = za^{2(k+1)} = a^{2(k+1)}z = az$$

Also kommutiert z mit a. Benutzt man diese Eigenschaft zusammen mit 2.1, dann erhält man

$$zaz = z^2a = a.$$

Folglich hat $z = yx^{-1}$ die Ordnung 2 und kommutiert mit a. Dies zeigt, daß $\varphi(x) := x^{-1}a$ eine anti-symmetrische Abbildung ist, wenn a mit keinem Element der Ordnung 2 kommutiert.

Um den Beweis abzuschließen, nehmen wir an, daß a mit einem Element z der Ordnung 2 kommutiert. Es folgt, daß $\varphi(x) := x^{-1}a$ nicht anti-symmetrisch ist, denn es gilt:

$$\varphi(z)e = z^{-1}a = za = az = e^{-1}az = \varphi(e)z$$

und $z \neq e$. □

Korollar 5 *Alle Gruppen mit ungerader Ordnung besitzen eine anti-symmetrische Abbildung.*

Beweis Da eine Gruppe mit ungerader Ordnung kein Element der Ordnung 2 besitzt, ist $\varphi(x) := x^{-1}a$ eine anti-symmetrische Abbildung für alle a. □

Korollar 6 *Für alle $n > 2$ besitzen die symmetrischen Gruppen S_n und die alternierenden Gruppen A_n anti-symmetrische Abbildungen.*

Beweis Wenn n ungerade ist, ist jeder n-Zykel in A_n und kommutiert mit keinem Element der Ordnung 2. Ist n gerade so gilt dies für jeden $(n-1)$-Zykel. □

Korollar 7 *Wenn eine endliche Gruppe ein Element a besitzt, dessen Zentralisator $Z(a)$ ungerade Ordnung hat, dann besitzt die Gruppe eine anti-symmetrische Abbildung.*

Beweis Wenn die Ordnung von $Z(a) = \{x \in G | xa = ax\}$ ungerade ist, dann kommutiert a mit keinem Element der Ordnung 2. □

2.1.3 Erweiterungstheoreme

Theorem 5 *Sei G eine Gruppe mit Normalteiler H und es existieren anti-symmetrische Abbildungen φ auf H und ψ auf G/H, dann existiert eine anti-symmetrische Abbildung γ auf G. Kurz gesagt: Die Klasse der Gruppen mit anti-symmetrischen Abbildungen ist gegen Erweiterung abgeschlossen.*

Beweis Seien $u_1H, \ldots, u_rH$ die Elemente von G/H. Wir definieren die Abbildung $\psi^* : \{u_1, \ldots, u_r\} \to \{u_1, \ldots, u_r\}$ durch die Bedingung

$$\psi^*(u_i)H = \psi(u_iH). \tag{2.3}$$

Da jedes Element von G eindeutig als Produkt $g = hu$ geschrieben werden kann, wobei $h \in H$ und $u = u_i$ für ein i, ist die Abbildung $\gamma : G \to G$, $\gamma(g) = \gamma(hu) := \psi^*(u)\varphi(h)$ wohldefiniert. Wir zeigen nun, daß γ anti-symmetrisch ist. Seien $g = hu$ und $g' = h'u'$ Elemente von G, dann folgt aus $\gamma(g)g' = \gamma(g')g$:

$$\psi^*(u)\varphi(h)h'u' = \psi^*(u')\varphi(h')hu. \tag{2.4}$$

Durch Multiplikation mit H erhält man $\psi^*(u)Hu'H = \psi^*(u')HuH$. Mit 2.3 folgt $\psi(uH)u'H = \psi(u'H)uH$ und damit, weil ψ anti-symmetrisch ist, $uH = u'H$ bzw. $u = u'$, da die Repräsentanten fest gewählt sind. Nun wird aus 2.4 die Gleichung

$$\psi^*(u)\varphi(h)h'u = \psi^*(u)\varphi(h')hu.$$

Nach kürzen von $\psi^*(u)$ und u bleibt die Gleichung $\varphi(h)h' = \varphi(h')h$, woraus, wegen der Anti-Symmetrie von φ, $h = h'$ folgt und insgesamt $g = g'$. Also ist γ eine anti-symmetrische Abbildung. □

Definition 4 *([3]) Seien $(G, \cdot, ^{-1}, e)$ und $(X, +, -, 0)$ Gruppen und $\pi : G \to Aut(X)$ ein Gruppenhomomorphismus. Das semi-direkte Produkt von X und G relativ zu π wird definiert durch:*

$$X \times_\pi G = \{(x, a) | x \in X, a \in G\}$$

mit der Operation $(x_1, a_1)(x_2, a_2) = (x_1 + \pi(a_1)[x_2], a_1a_2)$, *für* $x_1, x_2 \in X$ *und* $a_1, a_2 \in G$.

Proposition 1 *1. Das semi-direkte Produkt* $X \times_\pi G$ *ist eine Gruppe.*

2. *Die Menge* $\{(x, a) \in X \times_\pi G | x = 0\}$ *ist eine Untergruppe von* $X \times_\pi G$, *welche isomorph zu* G *ist.*

3. *Die Menge* $N = \{(x, a) \in X \times_\pi G | a = e\}$ *ist ein Normalteiler von* $X \times_\pi G$, *die isomorph zu* X *ist und* $(X \times_\pi G)/N$ *ist isomorph zu* G.

Beweis zu 3.: Es ist klar, daß X isomorph zu N ist. Definiere $\varphi : X \times_\pi G \to G$ durch $\varphi(x, a) = a$, dann ist φ ein Homomorphismus mit $\varphi(X \times_\pi G) = G$ und $Kern(\varphi) = N$. Der Homomorphiesatz zeigt nun $(X \times_\pi G)/N \cong G$. □

Korollar 8 *Wenn* A *und* B *Gruppen mit anti-symmetrischen Abbildungen sind und* $\pi : G \to Aut(X)$ *ein Gruppenhomomorphismus ist, dann besitzt das semi-direkte Produkt* $A \times_\pi B$ *eine anti-symmetrische Abbildung.*

Beweis $A \times_\pi B$ hat eine Untergruppe isomorph zu A und die Faktorgruppe $(A \times_\pi B)/A$ ist isomorph zu B. Mit dem Erweiterungstheorem folgt nun die Behauptung.

Korollar 9 *Sind* A *und* B *Gruppen mit anti-symmetrischen Abbildungen, dann besitzt das direkte Produkt* $A \times B$ *eine anti-symmetrische Abbildung.*

Beweis Spezialfall vom vorherigen Korollar, wobei π alle Elemente auf die Identität abbildet.

2.1.4 Einfache Gruppen

Die einfachen Gruppen spielen angesichts Theorem 5 eine entscheidende Rolle bei der Bestimmung der endlichen Gruppen mit anti-symmetrischen Abbildungen. Die Klassifikation der einfachen Gruppen (GORENSTEIN [11]) zeigt, daß jede endliche einfache Gruppe von einem der folgenden Typen ist: eine zyklische Gruppe mit Primzahlordnung, eine alternierende Gruppe, ein Mitglied einer von sechzehn Familien vom Lie-Typ oder eine von 26 sporadischen Gruppen.

Theorem 6 *Jede endliche einfache Gruppe, außer* $\mathbb{Z}_2$, *besitzt eine anti-symmetrische Abbildung.*

Beweis Die zyklischen und alternierenden Gruppen werden von Korollar 5 und Korollar 6 abgedeckt. Mit dem Atlas der endlichen Gruppen [7] kann man verifizieren, daß in allen 26 sporadischen einfachen Gruppen der Zentralisator der Elemente mit maximaler Primzahlordnung ungerade Ordnung hat und diese Gruppen daher eine anti-symmetrische Abbildung besitzen (Korollar 7). Korollar 7 kann auch

auf die Lie-Gruppen angewendet werden, da LYONS, SOLOMON und SEITZ [10, Gallian] gezeigt haben, daß jede einfache Lie-Gruppe ein Element besitzt, dessen Zentralisator ungerade Ordnung hat. □

2.1.5 Verallgemeinerte Diedergruppen

Definition 5 *Sei G eine abelsche Gruppe. Die verallgemeinerte Diedergruppe dih(G) wird definiert durch das semi-direkte Produkt $G \times_\pi \mathbb{Z}_2$, wobei $\pi(0) = id$ und $\pi(1) = inv$.*

Beispiel Die Diedergruppe D_n ist das semi-direkte Produkt von $\mathbb{Z}_n$ und $\mathbb{Z}_2$: $D_n = \mathbb{Z}_n \times_\pi \mathbb{Z}_2$.

Lemma 8 *Für zwei abelsche Gruppen A und B gilt: dih($A \times B$) $\cong A \times_\gamma$ dih(B), wobei $\gamma(b, 0) = id$ und $\gamma(b, 1) = inv$.*

Beweis Die Abbildung $\psi : \mathrm{dih}(A \times B) \to A \times_\gamma \mathrm{dih}(B)$, $\psi((a, b), z) = (a, (b, z))$, mit $a \in A$, $b \in B$ und $z \in \mathbb{Z}_2$ ist ein Isomorphismus. □

Theorem 7 *Sei G eine nicht-triviale abelsche Gruppe, dann besitzt dih(G) eine anti-symmetrische Abbildung.*

Beweis Fallunterscheidung:
1. Fall: Ist G eine zyklische Gruppe der Ordnung n, dann gilt $\mathrm{dih}(G) \cong D_n$ und G hat demnach eine anti-symmetrische Abbildung.
2. Fall: Hat G ungerade Ordnung und ist nicht zyklisch, dann kann man G faktorisieren in eine zyklische Gruppe Z_m und eine Gruppe H mit ungerader Ordnung. Es folgt mit Hilfe von Lemma 8:

$$\mathrm{dih}(G) \cong \mathrm{dih}(H \times Z_m) \cong H \times_\gamma \mathrm{dih}(Z_m).$$

H besitzt eine anti-symmetrische Abbildung (H hat ungerade Ordnung) und es gilt $\mathrm{dih}(Z_m) \cong D_m$. Mit Korollar 8 folgt, daß $\mathrm{dih}(G)$ eine anti-symmetrische Abbildung besitzt.
3. Fall: Ist G eine nicht-zyklische 2-Gruppe, dann gilt $G \cong Z_{2^{i_1}} \times Z_{2^{i_2}} \times \ldots \times Z_{2^{i_s}}$ mit $s \geq 2$. Folglich hat das Zentrum $Z(\mathrm{dih}(G))$ die Ordnung 2^s und ist isomorph zu $H = Z_2 \times Z_2 \times \ldots \times Z_2$. Da H abelsch ist und wenigstens zwei Involutionen enthält, besitzt H eine vollständige und damit eine anti-symmetrische Abbildung. Der Quotient $\mathrm{dih}(G)/Z(\mathrm{dih}(G))$ ist isomorph zu $\mathrm{dih}(Z_{2^{(i_1-1)}} \times Z_{2^{(i_2-1)}} \times \ldots \times Z_{2^{(i_s-1)}})$ und besitzt, durch Induktion nach dem Maximum von i_j nachweisbar, eine anti-symmetrische Abbildung und wir können mit Korollar 8 folgern, daß G eine anti-symmetrische Abbildung besitzt.
4. Fall: Nun habe G gerade Ordnung, sei aber keine 2-Gruppe, dann kann man G in

zwei nicht-triviale Gruppen faktorisieren: eine Gruppe H mit ungerader Ordnung und eine 2-Gruppe N. Es folgt, daß

$$\mathrm{dih}(G) \cong \mathrm{dih}(H \times N) \cong H \times_\gamma \mathrm{dih}(N).$$

H und $\mathrm{dih}(N)$ haben anti-symmetrische Abbildungen also auch $\mathrm{dih}(G)$. □

Wir adaptieren nun die Argumente von HALL und PAIGE [14] zur Charakterisierung der endlichen p-Gruppen, die eine vollständige Abbildung besitzen, um zu zeigen, daß die selbe Charakterisierung auch für anti-symmetrische Abbildungen gilt.

Theorem 8 *Eine nicht-triviale endliche p-Gruppe hat eine anti-symmetrische Abbildung genau dann, wenn sie keine zyklische 2-Gruppe ist.*

Beweis Sei G eine nicht-triviale endliche p-Gruppe. Wenn p ungerade ist, dann hat G eine anti-symmetrische Abbildung. Wenn G eine zyklische 2-Gruppe ist, dann wissen wir durch das Resultat von PAIGE, Theorem 1 (Seite 16), daß G keine vollständige und damit auch keine anti-symmetrische Abbildung besitzt. Der Fall, daß G eine nicht-zyklische abelsche 2-Gruppe ist wurde ebenfalls von PAIGE gezeigt. Deshalb können wir uns auf den Fall beschränken, daß G eine nicht-abelsche 2-Gruppe ist mit der Ordnung 2^n.

Besitzt G eine zyklische Untergruppe der Ordnung 2^{n-1} dann ist bekanntlich G entweder eine Diedergruppe, eine verallgemeinerte Quaternionen-Gruppe oder eine Semi-Diedergruppe (PAIGE [14]). Nach Theorem 3 haben diese Gruppen eine anti-symmetrische Abbildung.

Also nehmen wir an, daß G keine zyklische Untergruppe der Ordnung 2^{n-1} hat. Wenn G genau ein Element der Ordnung 2 enthält, dann wäre sie eine verallgemeinerte Quaternionen-Gruppe und hätte eine zyklische Untergruppe der Ordnung 2^{n-1} (siehe PAIGE [14]), im Widerspruch zu unserer Annahme. Also hat G wenigstens zwei Elemente der Ordnung 2 und wenigstens eins davon im Zentrum. Diese beiden Elemente erzeugen eine 4-Gruppe V. Wenn V in zwei verschiedenen maximalen Untergruppen M_1 und M_2 enthalten ist, dann ist $M_1 \cap M_2 = K \supset V$ ein Normalteiler von G und sowohl K als auch G/K sind nicht zyklisch. Also haben, mit Induktion, K und G/K anti-symmetrische Abbildungen und, mit dem Erweiterungstheorem, damit auch G.

Wir nehmen daher an, daß V in genau einer maximalen Untergruppe M_1 enthalten ist. Weil G nicht zyklisch ist, enthält sie eine weitere maximale Untergruppe M_2 (PAIGE [14]). Wenn $M_1 \cap M_2$ nicht zyklisch ist, dann stellt sie eine normale nicht-zyklische Untergruppe K mit nicht-zyklischer Quotientengruppe dar und wir können mit Induktion schließen, daß G eine anti-symmetrische Abbildung besitzt.

Nun sei $M_1 \cap M_2$ zyklisch. Es muß M_1 eine Gruppe der Ordnung 2^{n-1} sein, die eine zyklische Untergruppe der Ordnung 2^{n-2} und die 4-Gruppe V enthält.

Demnach muß M_1 entweder eine Diedergruppe, eine Semi-Diedergruppe oder eine abelsche Gruppe sein. In jedem Fall können wir

$$M_1 =< a, b | a^{2^{n-2}} = b^2 = e, ba = a^k b >$$

schreiben, wobei k gleich $-1, 2^{n-3} \pm 1$ oder 1 ist, abhängig davon, ob M_1 einer Dieder-, Semi-Dieder- oder einer abelschen Gruppe entspricht, und es ist $M_1 \cap M_2 =< a >$. Sei c ein Element von M_2 aber nicht von M_1. Da $< a >$ normal in M_2 ist, muß $c^2 = a^r$ gelten, wobei r gerade ist, da sonst c die Ordnung 2^{n-1} hat, was wir ausgeschlossen haben. Weil $< a >$ normal in G ist, erhalten wir $cb = bca^s$ für ein s.

Sei H die Untergruppe $< a^2, b >$. Eine einfache Rechnung zeigt, daß H mit den Elementen a, c und ac kommutiert, wenn s gerade ist. Also ist H ein Normalteiler in G. H ist nicht zyklisch, also wissen wir durch Induktion, daß H eine anti-symmetrische Abbildung hat und der Quotient $G/H \cong Z_2 \times Z_2$ hat ebenfalls eine anti-symmetrische Abbildung. Mit dem Erweiterungstheorem folgt nun, daß G eine anti-symmetrische Abbildung besitzt.

Es bleibt der Fall, daß s ungerade ist. Wir untersuchen die Untergruppe, die durch cb erzeugt wird. Wir haben

$$(cb)^2 = cbcb = cb^2ca^s = c^2a^s = a^{s+r}, \qquad \text{wobei } s + r \text{ ungerade ist.}$$

Also hat $(cb)^2$ die Ordnung 2^{n-2}, was $ord(cb) = 2^{n-1}$ impliziert. Aber dies widerspricht unserer Annahme, daß G keine zyklische Untergruppe der Ordnung 2^{n-1} besitzt. Damit haben wir die Behauptung bewiesen. □

Korollar 10 *Eine endliche nilpotente Gruppe mit trivialer oder nicht-zyklischer 2-Sylow-Untergruppe hat eine anti-symmetrische Abbildung.*

Beweis Eine endliche nilpotente Gruppe ist das direkte Produkt ihrer Sylow-Untergruppen. Die p-Sylow-Untergruppen mit ungeradem p haben gemäß Theorem 8 eine anti-symmetrische Abbildung. Da die 2-Sylow-Untergruppe trivial oder nicht-zyklisch ist, hat sie ebenfalls eine anti-symmetrische Abbildung. Demnach hat auch das direkte Produkt, also G, eine anti-symmetrische Abbildung. □

Die obengenannten Theoreme reichen aus, um zu zeigen, daß alle nicht-abelschen Gruppen der Ordnung kleiner als 36 eine anti-symmetrische Abbildung besitzen, mit Ausnahme der Gruppe $< a, b | a^3 = b^8 = e, ab = ba^2 >$ der Ordnung 24, wobei bei dieser Gruppe ebenfalls gezeigt werden kann, daß sie eine besitzt.

Da es keinen Anhaltspunkt gibt, daß eine nicht-abelsche Gruppe keine anti-symmetrische Abbildung besitzt, vermuten GALLIAN und MULLIN, daß alle nicht-abelschen Gruppen eine anti-symmetrische Abbildung besitzen.

2.2 Invarianten von $Ant(G)$

Wir untersuchen nun welche Transformationen die Menge der anti-symmetrischen Abbildungen einer Gruppe invariant lassen.

Satz 3 *Sei $\varphi(x)$ eine anti-symmetrische Abbildung einer Gruppe $(G,\cdot)$, dann ist auch die Abbildung $a \cdot \varphi(x \cdot b)$, $a, b \in G$, anti-symmetrisch.*

Beweis Aus $(a \cdot \varphi(x \cdot b)) \cdot y = (a \cdot \varphi(y \cdot b)) \cdot x$ folgt mit dem Assoziativgesetz und der Kürzungsregel $\varphi(x \cdot b) \cdot y = \varphi(y \cdot b) \cdot x$. Die Gleichung wird nun von rechts mit b durchmultipliziert, also gilt $\varphi(x \cdot b) \cdot y \cdot b = \varphi(y \cdot b) \cdot x \cdot b$ und es folgt, da φ anti-symmetrisch ist, $x \cdot b = y \cdot b$ und damit $x = y$. Also ist $a \cdot \varphi(x \cdot b)$ anti-symmetrisch. □

Satz 4 *Wenn $\varphi(x)$ eine anti-symmetrische Abbildung der Gruppe $(G,\cdot)$ ist, dann ist für jedes $c \in G$ auch $\varphi(c \cdot x) \cdot c$ anti-symmetrisch.*

Beweis Es gilt, da φ anti-symmetrisch ist: $\varphi(c \cdot x) \cdot c \cdot y = \varphi(c \cdot y) \cdot c \cdot x \Rightarrow c \cdot x = c \cdot y \Rightarrow x = y$. □

Mit $l_a = (x \mapsto a \cdot x)$ werde die Linksmultiplikation und mit $r_b = (x \mapsto x \cdot b)$ die Rechtsmultiplikation bezeichnet. Die Transformationen $L_a = (\varphi \mapsto l_a \circ \varphi)$, $R_b = (\varphi \mapsto \varphi \circ r_b)$ und $M_c = (\varphi \mapsto r_c \circ \varphi \circ l_c)$ bilden jeweils eine Gruppe mit der Verknüpfung $\circ$, die isomorph zu G ist.

Satz 5 *(Verhoeff [27]) Sei φ eine anti-symmetrische Abbildung und ψ ein Automorphismus, dann ist auch $\psi \circ \varphi \circ \psi^{-1}$ anti-symmetrisch.*

Beweis Aus $\psi \circ \varphi \circ \psi^{-1}(x) \cdot y = \psi \circ \varphi \circ \psi^{-1}(y) \cdot x$ folgt mit $y = \psi(\psi^{-1}(y))$ und $x = \psi(\psi^{-1}(x))$, daß $\psi(\varphi(\psi^{-1}(x)) \cdot \psi^{-1}(y)) = \psi(\varphi(\psi^{-1}(y)) \cdot \psi^{-1}(x))$ gilt. Nachdem wir ψ auf beiden Seiten gekürzt haben, folgt aus der Anti-Symmetrie von φ, daß $\psi^{-1}(x) = \psi^{-1}(y)$ ist und damit $x = y$. □

Auch hier bilden die Transformationen $T_\psi = (\varphi \mapsto \psi \circ \varphi \circ \psi^{-1})$ eine Gruppe. Diese ist isomorph zur Automorphismen-Gruppe $Aut(G)$.

Bemerkung Satz 4 läßt sich auch mit Satz 5 und 3 beweisen. Dazu setzt man $\psi(x) := c^{-1}xc$ und $a = b = c$.

Andere Möglichkeiten, wie aus einer vorgegebenen anti-symmetrischen Abbildung weitere konstruiert werden können, findet man im Abschnitt „Automorphismen und Anti-Automorphismen" sowie im Kapitel „Gruppen mit Vorzeichen".

2.3 Äquivalenzklassen

Auf der Menge der anti-symmetrischen Abbildungen einer Gruppe definieren wir in diesem Abschnitt eine Äquivalenzrelation. Diese ermöglicht es uns, eine Übersicht über alle anti-symmetrischen Abbildungen einer Gruppe zu gewinnen.

Mit 0 bezeichnen wir im folgenden das neutrale Element der Gruppe. Weiterhin sei φ ein Automorphismus der Gruppe G und l_a bzw. r_a die Links- bzw. Rechtsmultiplikation mit dem Element $a \in G$.
Die genannten Abbildungen haben folgende Eigenschaften:

1. $l_a \circ l_b = l_{a \cdot b}, \quad r_a \circ r_b = r_{b \cdot a}, \quad l_a^{-1} = l_{a^{-1}}, \quad r_a^{-1} = r_{a^{-1}}$.
2. $l_a \circ r_b = r_b \circ l_a$.
3. $\varphi \circ l_b = l_{\varphi(b)} \circ \varphi, \quad \varphi \circ r_a = r_{\varphi(a)} \circ \varphi$.

Die ersten beiden Eigenschaften folgen aus dem Assoziativgesetz, für die letzte gilt: $\varphi \circ l_b(x) = \varphi(b \cdot x) = \varphi(b) \cdot \varphi(x) = l_{\varphi(b)} \circ \varphi(x)$ und $\varphi \circ r_a(x) = \varphi(x \cdot a) = \varphi(x) \cdot \varphi(a) = r_{\varphi(a)} \circ \varphi(x)$.

Die Äquivalenzklassen werden nun wie folgt definiert:

Definition 6 *Seien f, g Permutationen. f und g heißen äquivalent, $f \sim g$, wenn Elemente $a, b \in G$ und ein Automorphismus φ existieren, so daß gilt:*

$$f = l_b \circ \varphi^{-1} \circ g \circ \varphi \circ r_a.$$

Die Relation $\sim$ bildet eine Äquivalenzrelation auf der Menge der Permutationen:

1. $\sim$ ist reflexiv: $f = l_0 \circ Id \circ f \circ Id \circ r_0$
2. $\sim$ ist symmetrisch: $f = l_b \circ \varphi^{-1} \circ g \circ \varphi \circ r_a$ genau dann, wenn

$$\begin{aligned} g &= \varphi \circ l_b^{-1} \circ f \circ r_a^{-1} \circ \varphi^{-1} \\ &\underset{1.,2.}{=} \varphi \circ l_{b^{-1}} \circ f \circ \circ r_{a^{-1}} \circ \varphi^{-1} \\ &\underset{3.}{=} l_{\varphi(b^{-1})} \circ \varphi \circ f \circ \varphi^{-1} \circ r_{\varphi(a^{-1})} \end{aligned}$$

   Also: $f \sim g$ impliziert $g \sim f$.

3. $\sim$ ist transitiv: Sei $f = l_{b_1} \circ \varphi_1^{-1} \circ g \circ \varphi_1 \circ \circ r_{a_1}$ und $g = l_{b_2} \circ \varphi_2^{-1} \circ h \circ \varphi_2 \circ r_{a_2}$, dann folgt

$$\begin{aligned} f &= l_{b_1} \circ \varphi_1^{-1} \circ l_{b_2} \circ \varphi_2^{-1} \circ h \circ \varphi_2 \circ r_{a_2} \circ \varphi_1 \circ r_{a_1} \\ &\underset{3.}{=} l_{b_1} \circ l_{\varphi_1^{-1}(b_2)} \circ (\varphi_2 \circ \varphi_1)^{-1} \circ h \circ (\varphi_2 \circ \varphi_1) \circ r_{\varphi_1(a_2)} \circ r_{a_1} \\ &\underset{1.}{=} l_{b_1 \varphi_1^{-1}(b_2)} \circ (\varphi_2 \circ \varphi_1)^{-1} \circ h \circ (\varphi_2 \circ \varphi_1) \circ r_{a_1 \varphi_1(a_2)} \end{aligned}$$

Damit ist $f \sim g, g \sim h \Rightarrow f \sim h$ gezeigt.

Aus dem vorherigen Abschnitt können wir das folgende Korollar ableiten.

Korollar 11 *Seien $f, g \in S_n$ mit $f \sim g$, dann gilt*

$$f \text{ anti-symmetrisch} \Leftrightarrow g \text{ anti-symmetrisch.}$$

Beispiel Die Diedergruppe D_5 besitzt 34040 anti-symmetrische Abbildungen. Es können 3040 von $x^{-1} \cdot a$ abgeleitet werden (siehe Satz 21, Seite 53). Für die restlichen 31000 anti-symmetrischen Abbildungen erhalten wir folgende Repräsentanten der Äquivalenzklassen:

1. Diese 15 Permutationen erzeugen 30000 anti-symmetrische Abbildungen (rechts: Zyklenschreibweise):

$$\begin{array}{lcl}
[0215637894] & = & (21)(53)(67894) \\
[0215638974] & = & (21)(53)(68794) \\
[0215647938] & = & (21)(5467983) \\
[0215694378] & = & (21)(59873)(64) \\
[0215748396] & = & (21)(5473)(896) \\
[0215748936] & = & (21)(5479683) \\
[0215794638] & = & (21)(5983)(764) \\
[0215867394] & = & (21)(5673)(894) \\
[0215874936] & = & (21)(5796483) \\
[0215897436] & = & (21)(5967483) \\
[0245678931] & = & (246835791) \\
[0245718936] & = & (247968351) \\
[0256714893] & = & (251)(647893) \\
[0256743918] & = & (2547981)(63) \\
[0257918436] & = & (251)(749683)
\end{array}$$

2. Die Permutation $[0215643978] = (21)(5463)(987)$ erzeugt die restlichen 1000 anti-symmetrischen Abbildungen.

Bemerkung Mit $[a_0 a_1 \ldots a_9]$ meinen wir die Permutation $x \mapsto a_x$, d.h.

$$[a_0 a_1 \ldots a_9] = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{pmatrix}.$$

Dieses Beispiel zeigt auch, daß die Klassen nicht unbedingt gleich groß sein müssen.

2.4 Automorphismen und Anti-Automorphismen

In diesem Abschnitt untersuchen wir einen Spezialfall, nämlich anti-symmetrische Automorphismen bzw. Anti-Automorphismen. Automorphismen sind bekanntlich bijektive Permutationen einer Gruppe $\varphi : G \to G$ mit $\varphi(xy) = \varphi(x)\varphi(y)$. Der Begriff „Anti-Automorphismus" wird dagegen nicht so häufig benutzt, er wird aber ganz ähnlich definiert:

Definition 7 *Eine bijektive Abbildung $\psi : G \to G$ einer Gruppe G heißt* Anti-Automorphismus*, wenn für alle $x, y \in G$ gilt: $\psi(xy) = \psi(y)\psi(x)$. Ein Anti-Automorphismus oder ein Automorphismus heißt* fixpunktfrei*, wenn für alle $x \neq 0$ gilt: $\psi(x) \neq x$.*

Die folgenden Eigenschaften eines Anti-Automorphismus werden genauso gezeigt wie bei einem Automorphismus:

1. $\psi(0) = 0$
2. $\psi(x)^{-1} = g(x^{-1})$
3. $ord(\psi(x)) = ord(x)$

Bemerkung Da für einen (Anti-)Automorphismus ψ immer $\psi(0) = 0$ gilt, ist es sinnvoll, bei der Definition von „fixpunktfrei" die Stelle $x = 0$ auszuschließen.

Die Menge der Automorphismen einer Gruppe können wir bijektiv auf die Menge der Anti-Automorphismen abbilden. Es gilt nämlich

$$\varphi \text{ ist ein Automorphismus} \quad \Leftrightarrow \quad \varphi \circ inv \text{ ist ein Anti-Automorphismus}$$

Ist φ ein Automorphismus, dann gilt

$$\varphi \circ inv(xy) = \varphi((xy)^{-1}) = \varphi(y^{-1}x^{-1}) = \varphi(y^{-1})\varphi(x^{-1}) = \varphi \circ inv(y)\varphi \circ inv(x^{-1})$$

und $\varphi \circ inv$ ist ein Anti-Automorphismus. Ist andererseits $\varphi \circ inv$ ein Anti-Automorphismus, dann haben wir

$$\varphi(xy) = \varphi((y^{-1}x^{-1})^{-1}) = \varphi \circ inv(x^{-1})\varphi \circ inv(y^{-1}) = \varphi(x)\varphi(y)$$

und φ ist ein Automorphismus.

Wenn wir also die Automorphismen einer Gruppe kennen, dann können wir auch ohne weiteres alle Anti-Automorphismen dieser Gruppe bestimmen.

Ist ψ ein Anti-Automorphismus, so ist für zwei beliebige Automorphismen φ_1, φ_2 auch $\varphi_1 \circ \psi \circ \varphi_2$ ein Anti-Automorphismus:

$$\begin{aligned} \varphi_1 \circ \psi \circ \varphi_2(xy) &= \varphi_1 \circ \psi(\varphi_2(x)\varphi_2(y)) = \varphi_1(\psi(\varphi_2(y))g(\varphi_2(x))) \\ &= \varphi_1(\psi(\varphi_2(y)))\varphi_1(\psi(\varphi_2(x))) \\ &= \varphi_1 \circ \psi \circ \varphi_2(y)\varphi_1 \circ \psi \circ \varphi_2(x) \end{aligned}$$

Das Gleiche gilt auch, wenn φ_1 und φ_2 zwei Anti-Automorphismen sind.

Wie man leicht sieht, gilt außerdem: Sind ψ_1, ψ_2 Anti-Automorphismen, dann ist $\psi_1 \circ \psi_2$ ein Automorphismus.
Ist ψ ein (fixpunktfreier) Anti-Automorphismus, dann ist auch ψ^{-1} ein (fixpunktfreier) Anti-Automorphismus:

$$\psi^{-1}(xy) = \psi^{-1}(\psi(\psi^{-1}(x))\psi(\psi^{-1}(y))) = \psi^{-1}(\psi(\psi^{-1}(y)\psi^{-1}(x))) = \psi^{-1}(y)\psi^{-1}(x)$$

und

$$\psi(x) = x \qquad \Leftrightarrow \qquad x = \psi^{-1}(x).$$

Damit haben wir gezeigt, daß die Menge der Anti-Automorphismen zusammen mit den Automorphismen eine Gruppe bildet. Gibt es einen Anti-Automorphismus der auch ein Automorphismus ist, so folgt

$$xy = \psi(\psi^{-1}(x))\psi(\psi^{-1}(y)) = \psi(\psi^{-1}(y)\psi^{-1}(x)) = yx$$

und die Gruppe ist abelsch. In einer abelschen Gruppe sind Anti-Automorphismen auch Automorphismen, während in einer nicht-abelschen Gruppe kein Anti-Automorphismus ein Automorphismus ist.

Mit Hilfe der Anti-Automorphismen finden wir eine zusätzliche Möglichkeit, aus einer vorgegebenen anti-symmetrischen Abbildung eine weitere zu konstruieren.

Satz 6 *Sei φ eine anti-symmetrische Abbildung und ψ ein Anti-Automorphismus, dann ist auch $\psi \circ \varphi^{-1} \circ \psi^{-1}$ anti-symmetrisch.*

Beweis Es gelte $\psi\circ\varphi^{-1}\circ\psi^{-1}(x)y = \psi\circ\varphi^{-1}\circ\psi^{-1}(y)x$. Wir setzen $\tilde{x} := \varphi^{-1}\circ\psi^{-1}(x)$ und $\tilde{y} := \varphi^{-1} \circ \psi^{-1}(y)$, womit $\psi(\tilde{x})\psi(\varphi(\tilde{y})) = \psi(\tilde{y})\psi(\varphi(\tilde{x}))$ folgt. Wir nutzen die Eigenschaft aus, daß ψ ein Anti-Automorphismus ist, um $\psi(\varphi(\tilde{y})\tilde{x}) = \psi(\varphi(\tilde{x})\tilde{y})$ zu erhalten. In dieser Gleichung kürzen wir ψ. Aus der resultierenden Gleichung folgt $\tilde{x} = \tilde{y}$, denn $\varphi \in Ant(G)$. Da $\varphi^{-1} \circ \psi^{-1}$ bijektiv ist, haben wir damit $x = y$, und der Satz ist bewiesen. □

Wir untersuchen nun, wann ein (Anti-)Automorphismus anti-symmetrisch ist.

Satz 7 *Ein Anti-Automorphismus ist genau dann anti-symmetrisch, wenn er fixpunktfrei ist.*

Beweis Sei ψ ein fixpunktfreier Anti-Automorphismus und es gelte $\psi(x)y = \psi(y)x$. Die Gleichung wird von links mit $\psi(y)^{-1}$ und von rechts mit y^{-1} durchmultipliziert, es folgt $\psi(y^{-1})\psi(x) = \psi(xy^{-1}) = xy^{-1}$. Da ψ fixpunktfrei ist, muß

$xy^{-1} = 0$, bzw. $x = y$ gelten. Also ist ψ anti-symmetrisch. Wenn andererseits ψ einen Fixpunkt $x \neq 0$ besitzt, dann gilt $\psi(0)x = x = \psi(x) = \psi(x)0$ und ψ ist nicht anti-symmetrisch. □

Satz 8 *Sei φ ein Automorphismus. φ ist genau dann anti-symmetrisch, wenn für alle $y \in G$ gilt: $y^{-1}\varphi(z)y$ ist fixpunktfrei.*

Beweis Die Gleichung $\varphi(x)y = \varphi(y)x$ ist äquivalent zu $\varphi(y^{-1})\varphi(x) = xy^{-1}$ bzw. $\varphi(y^{-1}x) = x(y^{-1}x)x^{-1}$. Mit $z := y^{-1}x$ ist dies äquivalent zu $\varphi(z) = yzy^{-1}$ bzw. $y^{-1}\varphi(z)y = z$. □

Lemma 9 *Sei ψ ein Anti-Automorphismus und φ ein (Anti-)Automorphismus, dann gilt:*

$$\psi \textit{ fixpunktfrei} \quad \Leftrightarrow \quad \varphi^{-1} \circ \psi \circ \varphi \textit{ fixpunktfrei}$$

Beweis Es ist $\psi(x) = x$ äquivalent zu $\varphi^{-1}(\psi(\varphi(\varphi^{-1}(x)))) = \varphi^{-1}(x)$. □

Wir können nun Satz 4 von GALLIAN und MULLIN etwas anders formulieren:

Satz 9 *Der Anti-Automorphismus $\psi(x) := a^{-1}x^{-1}a$ ist genau dann fixpunktfrei, wenn x^{-1} und $a^{-1}xa$ keinen gemeinsamen Fixpunkt $x \neq 0$ besitzen.*

Beweis $\psi(x)$ fixpunktfrei $\Leftrightarrow$ $\psi(x)$ anti-symmetrisch $\Leftrightarrow$ $a\psi(x) = x^{-1}a$ anti-symmetrisch $\Leftrightarrow$ a kommutiert mit keinem Element der Ordnung 2 $\Leftrightarrow$ für alle $x \neq 0$ gilt: $x^{-1} = x \Rightarrow a^{-1}xa \neq x$. □

Satz 10 *Sei $h \circ g$ ein Anti-Automorphismus der Gruppe G der Ordnung n, $h, g \in S_n$ mit den Eigenschaften: $ord(g) = 2$, $ord(h)$ ungerade und $g \circ h = h \circ g$. Genau dann ist $h \circ g$ fixpunktfrei, wenn g und h keinen gemeinsamen Fixpunkt $x \neq 0$ besitzen.*

Beweis Sei $h \circ g$ nicht fixpunktfrei, d.h. es existiert ein $x \neq 0$ mit $h(g(x)) = x$. Es folgt $h(g(h(g(x)))) = h(h(g(g(x)))) = h(h(x)) = x$. Da die Ordnung von h ungerade, sprich $2k + 1$ ist, haben wir $x = h^{2k+1}(x) = h(h^{2k}(x)) = h(x)$. Damit gilt $h(g(x)) = g(h(x)) = g(x) = x$ und $x \neq 0$ ist ein gemeinsamer Fixpunkt von g und h. Haben andererseits g und h den gemeinsamen Fixpunkt $x \neq 0$, dann ist $h(g(x)) = h(x) = x$ und $h \circ g$ ist nicht fixpunktfrei. □

Bemerkung Jede Permutation p kann in zwei Permutationen h und g zerlegt werden, so daß die Bedingungen des Satzes erfüllt sind. Dazu schreibt man p als

Produkt disjunkter Zykel. Die Transpositionen werden dann zu g zusammengefaßt, der Rest zu h.

Wir geben nun notwendige und hinreichende Konditionen für die Erkennung der anderen Fehlertypen an, wenn wir ein Prüfziffersystem $\tau^n(x_n)\cdot\ldots\cdot\tau(x_1)\cdot x_0 = c$ mit einem (Anti-)Automorphismus τ benutzen. Dazu sei φ ein Automorphismus und ψ ein Anti-Automorphismus.

Nachbarvertauschungen

a.) Für alle $x \neq 0$: $\psi(x) \neq x$ (d.h. ψ ist fixpunktfrei)

b.) Für alle $x \neq 0$ und alle $y \in G$ gilt: $y^{-1}\varphi(x)y \neq x$ (d.h. $y^{-1}\varphi(x)y$ ist ein fixpunktfreier Automorphismus)

Sprungtranspositionen

a.) Für alle $x \neq 0$ und alle $z \in G$ gilt: $z^{-1}\psi^2(x)z \neq x$ (d.h. ψ^2 ist ein antisymmetrischer Automorphismus)

b.) wie a. für φ

Zwillingsfehler

a.) Für alle $x \neq 0$: $\psi(x^{-1}) \neq x$ (d.h. $\psi \circ inv$ ist ein fixpunktfreier Automorphismus)

b.) Für alle $x \neq 0$ und alle $z \in G$ gilt: $z^{-1}\varphi(x^{-1})z \neq x$ (d.h. $z^{-1}\varphi(x^{-1})z$ ist ein fixpunktfreier Anti-Automorphismus)

Sprungzwillingsfehler

a.) Für alle $x \neq 0$ und alle $z \in G$ gilt: $z^{-1}\psi^2(x^{-1})z \neq x$ (d.h. $z^{-1}\psi^2(x^{-1})z$ ist ein fixpunktfreier Anti-Automorphismus)

b.) wie a. für φ

phonetische Fehler

a.) Für $a = 2, \ldots, n-1$ gilt: $\psi(a)a^{-1} \neq \psi(1) \neq a^{-1}\psi(a)$

b.) Für $a = 2, \ldots, n-1$ gilt: $\varphi(a)a^{-1} \neq \varphi(1)$

Beweis Die Aussagen werden analog zu Satz 7 und 8 gezeigt. Die phonetischen Fehler werden erkannt, falls für $a = 2, \ldots, n-1$ gilt $\psi^{i+1}(a)\psi^i(0) \neq \psi^{i+1}(1)\psi^i(a)$. Da $\psi(0) = 0$ ist haben wir $\psi^{i+1}(a)\psi^i(a^{-1}) \neq \psi^{i+1}(1)$. Je nachdem ob i gerade oder ungerade ist, ist dies äquivalent zu $\psi^i(\psi(a)a^{-1}) \neq \psi^{i+1}(1)$ oder $\psi^i(a^{-1}\psi(a)) \neq \psi^{i+1}(1)$. Wir können ψ^i kürzen und erhalten damit die angegebene Bedingung.

2.5 Eine Abschätzung von $|Ant(G)|$

Die folgenden Sätze zeigen, daß eine große Anzahl Permutationen nicht anti-symmetrisch sein kann.

Lemma 10 *Sei $(G, \cdot)$ eine Gruppe, $G \neq \{e\}$. Die Permutationen $g(x) = a \cdot x \cdot b$ mit $a, b \in G$ sind nicht anti-symmetrisch.*

Beweis Es gilt für beliebige $y \in G$ $g(b^{-1}) \cdot y = a \cdot b^{-1} \cdot b \cdot y = a \cdot y = a \cdot y \cdot b \cdot b^{-1} = g(y) \cdot b^{-1}$. Da in G ein Element $y \neq b^{-1}$ existiert, ist g nicht anti-symmetrisch. □

Lemma 11 *Wenn $g(x) = g(0) \cdot x$ für ein $x \neq 0$ ist, dann ist g nicht anti-symmetrisch.*

Beweis $g(x) \cdot 0 = g(x) = g(0) \cdot x$. □

Mit diesem Lemma findet man eine untere Grenze für die Anzahl der Permutationen, die nicht anti-symmetrisch sein können. Man kann nämlich die Anzahl der Permutationen mit $g(x) = g(0) \cdot x$, für ein $x \neq 0$, berechnen.

Wir bestimmen die Anzahl $a(n)$ der Permutationen die an der ersten ($y = 0$) und einer weiteren Stelle $y \in \{1, \ldots, n\}$ mit einer vorgegebenen Permutation $g \in S_{n+1}$ übereinstimmen. $a(n)$ ist nicht von g abhängig, daher wählen wir o.B.d.A. $g(x) = x$. Da die erste Stelle einer weiteren Permutation p gleich 0 sein muß, reicht es, die letzten n Stellen zu betrachten. Die letzten n Stellen sind aber Permutationen aus S_n, d.h.

$$\begin{aligned} a(n) &= |\{p \in S_{n+1} : p(0) = 0 \text{ und } |\{1 \leq y \leq n : p(y) = y\}| \geq 1\}| \\ &= |\{p \in S_n : |\{0 \leq y \leq n-1 : p(y) = y\}| \geq 1\}|. \end{aligned}$$

(Mit $|M|$ bezeichnen wir die Anzahl der Elemente in der Menge M.)

Dieses Problem ist in der Literatur bekannt als „Mausefallenspiel mit n Karten":

> *Gegeben seien n Ziffern und n numerierte Umschläge. Wieviele Möglichkeiten gibt es, die Ziffern in die Umschläge einzulegen, so daß mindestens eine Zahl in den richtigen Umschlag kommt.*

Die gesuchte Lösung $a(n)$ kann wie folgt berechnet werden (siehe [25], Zahlenreihe: A002467):

1. $a(0) = 0$, $a(1) = 1$ und $a(n) = (n-1)(a(n-1) + a(n-2))$, oder
2. $a(0) = 0$, $a(1) = 1$, $a(n) = n \cdot a(n-1) - (-1)^n$, oder

3. $a(0) = 0$, $a(n) = [n!\frac{e-1}{e}]$, wobei $e = 2,71828...$ die Eulersche Zahl ist und $[\ ]$ die (verschobene) Gaußklammer (auf die nächste ganze Zahl runden).

Nun betrachten wir alle Permutationen aus S_n, die mit $a \cdot x$ an der Stelle 0 und einer weiteren Stelle übereinstimmen. Die Anzahl dieser Permutationen ist gleich $a(n-1)$. Da für $a \neq b$ auch $a \cdot 0 \neq b \cdot 0$ ist, sind die Permutationen, die mit $a \cdot x$ an der ersten Stelle übereinstimmen und die, die mit $b \cdot x$ an der ersten Stelle übereinstimmen, alle verschieden. Es gibt daher mindestens $n \cdot a(n-1)$ Permutationen in S_n, die nicht anti-symmetrisch sind.

Satz 11 *Die Anzahl der nicht anti-symmetrischen Permutationen ist größer oder gleich* $n \cdot a(n-1) = n \cdot [(n-1)!\frac{e-1}{e}]$, *also*

$$n! - |Ant(G)| \geq n \cdot [(n-1)!\frac{e-1}{e}].$$

Durch einfaches Umformen erhalten wir nun ein Abschätzung für $|Ant(G)|$.

Theorem 9 *Sei G eine Gruppe der Ordnung n, dann gilt*

$$|Ant(G)| \leq n! - n \cdot [(n-1)!\frac{e-1}{e}] \leq \frac{n!}{e} + \frac{n}{2}.$$

Beweis Wir können die verschobene Gaußklammer wie folgt abschätzen:

$$[(n-1)!\frac{e-1}{e}] \geq (n-1)!\frac{e-1}{e} - \frac{1}{2}.$$

Damit folgt

$$\begin{aligned} |Ant(G)| &\leq n! - n \cdot [(n-1)!\frac{e-1}{e}] \leq n! - n!\frac{e-1}{e} + \frac{n}{2} \\ &\leq \frac{n!}{e} + \frac{n}{2}. \end{aligned}$$

□

Für Gruppen der Ordnungen 2 bis 12 geben wir eine Abschätzung von $|Ant(G)|$ an:

n	$a(n-1)$	$n!$	$n! - n \cdot a(n-1)$
2	1	2	0
3	1	6	3
4	4	24	8
5	15	120	45
6	76	720	264
7	455	5.040	1.855
8	3.186	40.320	14.832
9	25.487	362.880	133.497
10	229.384	3.628.800	1.334.960
11	2.293.839	39.916.800	14.684.571
12	25.232.230	479.001.600	176.214.840

Für $n = 2, 3, 4$ sind die Abschätzungen sogar bestmöglich. Die Gruppe $\mathbb{Z}_2$ besitzt keine, die Gruppe $\mathbb{Z}_3$ genau 3 und die Kleinsche-Vierergruppe genau 8 anti-symmetrische Abbildungen.

Für größere n gilt $\frac{|Ant(G)|}{n!} \approx \frac{1}{e}$. Demnach können höchstens 36,8% der Permutationen einer beliebigen Gruppe anti-symmetrisch sein.

2.6 Konstruktion anti-symmetrischer Abbildungen

Die anti-symmetrischen Abbildungen einer Gruppe $(G, *, ^{-1}, 0)$ können, ähnlich wie die Primzahlen, durch eine Siebmethode bestimmt werden. Man beginnt dabei mit einer Matrix $M = (m_{ij})$ mit $n \times n$ Elementen ($n = |G|$), wobei in jeder Zeile die Zahlen $0, 1, \ldots, n-1$ stehen, d.h. $m_{ij} = j$ für $i, j = 0, \ldots, n-1$.

Der folgende Algorithmus erzeugt anti-symmetrische Abbildungen oder er zeigt, daß bestimmte Abbildungen nicht anti-symmetrisch sein können.

1) Für i von 0 bis $n-1$ tue 2-6

    2) Sind alle Elemente der i-ten Zeile von M gestrichen, dann Abbruch

    3) Wähle ein Element m_{ij} der Zeile i aus, das noch nicht gestrichen ist, und setze $\varphi(i) := j$

    4) Falls $i = n-1$, dann Ende

    5) Für k von $i+1$ bis $n-1$ tue

        6) Streiche die Elemente $m_{k,j}$ und $m_{k,j*k*i^{-1}}$ aus der k-ten Zeile

Beispiel Im folgenden wird mit diesem Algorithmus eine anti-symmetrische Abbildung der Gruppe $\mathbb{Z}_5$ bestimmt. Dazu sei $M := \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$.

Das Element m_{00} ist nicht gestrichen, wir können also $\varphi(0) := 0$ setzen. Die Elemente $m_{1,0}$, $m_{1,0+1-0}$, $m_{2,0}$, $m_{2,0+2-0}$, $m_{3,0}$, $m_{3,0+3-0}$, $m_{4,0}$, $m_{4,0+4-0}$ werden gestrichen.

$$\begin{bmatrix} \mathbf{0} & 1 & 2 & 3 & 4 \\ \not{0} & \not{1} & 2 & 3 & 4 \\ \not{0} & 1 & \not{2} & 3 & 4 \\ \not{0} & 1 & 2 & \not{3} & 4 \\ \not{0} & 1 & 2 & 3 & \not{4} \end{bmatrix}$$

Nun können wir $\varphi(1) := 2$ wählen die Elemente $m_{2,2}$, $m_{2,2+2-1}$, $m_{3,2}$, $m_{3,2+3-1}$, $m_{4,2}$, $m_{4,2+4-1}$ werden gestrichen.

$$\begin{bmatrix} \mathbf{0} & 1 & 2 & 3 & 4 \\ \not{0} & \not{1} & \mathbf{2} & 3 & 4 \\ \not{0} & 1 & \not{2} & \not{3} & 4 \\ \not{0} & 1 & \not{2} & \not{3} & \not{4} \\ \not{0} & 1 & \not{2} & 3 & \not{4} \end{bmatrix}$$

Als nächstes muß $\varphi(2) := 4$ gewählt werden, da sonst in der vorletzten Zeile alle Elemente gestrichen wären und der Algorithmus im nächsten Durchlauf abbrechen würde. $m_{3,4}$, $m_{3,4+3-2}$, $m_{4,4}$, $m_{4,4+4-2}$ werden gestrichen.

$$\begin{bmatrix} \mathbf{0} & 1 & 2 & 3 & 4 \\ \not{0} & \not{1} & \mathbf{2} & 3 & 4 \\ \not{0} & 1 & \not{2} & \not{3} & \mathbf{4} \\ \not{0} & 1 & \not{2} & \not{3} & \not{4} \\ \not{0} & \not{1} & \not{2} & 3 & \not{4} \end{bmatrix}$$

Es bleibt $\varphi(3) := 1$, wodurch die Elemente $m_{4,1}$ und $m_{4,1+4-3}$ gestrichen werden. Im letzten Durchlauf ist damit die Wahl $\varphi(4) := 3$ möglich.

$$\begin{bmatrix} \mathbf{0} & 1 & 2 & 3 & 4 \\ \not{0} & \not{1} & \mathbf{2} & 3 & 4 \\ \not{0} & 1 & \not{2} & \not{3} & \mathbf{4} \\ \not{0} & \mathbf{1} & \not{2} & \not{3} & \not{4} \\ \not{0} & \not{1} & \not{2} & \mathbf{3} & \not{4} \end{bmatrix}$$

Der Algorithmus endet mit dem Ergebnis, daß $\varphi = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 2 & 4 & 1 & 3 \end{pmatrix}$ eine anti-symmetrische Abbildung von $\mathbb{Z}_5$ ist. Der folgende Satz zeigt, daß der Algorithmus wie gewünscht arbeitet:

Satz 12 *Es gilt:*

1. *Jede vorgegebene anti-symmetrische Abbildung kann mit dem obengenannten Algorithmus konstruiert werden.*
2. *Endet der Algorithmus im Schritt 4), dann ist die konstruierte Abbildung φ anti-symmetrisch.*
3. *Bricht der Algorithmus im Schritt 2) ab, dann existiert keine anti-symmetrische Abbildung, welche mit dem bis dahin definierten φ übereinstimmt.*

Beweis zu 1: Sei $\psi \in Ant(G)$. Es ist zu zeigen, daß für $k = 0, \ldots, n-1$ das Element $m_{k,\psi(k)}$ nicht gestrichen ist und damit die Wahl $\varphi(k) = \psi(k)$ möglich ist. Dies wird durch Induktion nach k gezeigt.
1) In der Zeile $k = 0$ ist kein Element gestrichen, man kann also $\varphi(0) := \psi(0)$ setzen.
2) Es gelte $\varphi(j) = \psi(j)$ für $j = 0, \ldots, k-1 < n-1$. Nimmt man an, daß das Element $m_{k,\psi(k)}$ in Schritt 6 gestrichen wurde, dann gibt es ein $i < k$ mit $\varphi(i) = \psi(k)$

oder mit $\varphi(i) * k * i^{-1} = \psi(k)$. Im ersten Fall folgt $\psi(i) = \psi(k)$ und ψ wäre keine Permutation. Im zweiten Fall wäre $\psi(i) * k = \psi(k) * i$. Beides steht im Widerspruch zur Voraussetzung $\psi \in Ant(G)$. Also ist das Element $m_{k,\psi(k)}$ nicht gestrichen und die Wahl $\varphi(k) := \psi(k)$ in Schritt 3 ist möglich.
Der Algorithmus bricht auch nicht ab, da das Elemente $m_{k,\psi(k)}$ nicht gestrichen ist.
Dies zeigt, daß jede anti-symmetrische Abbildung mit dem Algorithmus konstruiert werden kann.
zu 2: Es gelte $\varphi(k)*i = \varphi(i)*k$ für $i \leq k$ (o.B.d.A.) und damit $\varphi(k) = \varphi(i)*k*i^{-1}$. Da für $k > i$ das Element $\varphi(i) * k * i^{-1}$ gestrichen wird und die Wahl $\varphi(k) = \varphi(i) * k * i^{-1}$ nicht möglich ist, muß $k \leq i$ gelten, also ist $k = i$ und φ ist eine anti-symmetrische Abbildung.
zu 3: Dies ist eine direkte Folgerung aus 1. □

Um alle anti-symmetrischen Abbildungen einer Gruppe zu bestimmen, muß man offensichtlich in Schritt 3 nacheinander alle nicht gestrichenen Elemente auswählen. Der folgende rekursive Algorithmus verwirklicht dieses Vorgehen.

Erzeuge alle anti-symmetrischen Abbildungen, die mit der Permutation φ bis zur Stelle $i-1$ übereinstimmen und benutze dabei die Matrix $M = (m_{ij})$:

```
Prozedur AntiSymm(i);

1) Ist i = n, dann gib die anti-symmetrische Abbildung φ aus,
   sonst:

   2) Für alle Elemente m_ij der Zeile i, die nicht
      gestrichen sind, tue 3-7

      3) Setze φ(i) := j

      4) Für k von i+1 bis n-1 tue (falls i < n-1)

         5) Streiche die Elemente m_{k,j} und m_{k,j*k*i^-1} aus der
            k-ten Zeile von M

      6) AntiSymm(i+1),
         d.h.  erzeuge alle anti-symmetrischen Abbildungen,
         die mit der Permutation φ bis zur Stelle i überein-
         stimmen.

      7) Widerrufe die Änderungen der Schritte 4+5 (Elemente die
         bereits vorher gestrichen waren, bleiben gestrichen!)
```

Der Algorithmus wird mit dem Aufruf `AntiSymm(0)` gestartet. Wobei $m_{ij} := j$ für $i, j = 0, \ldots, n-1$.

Das folgende Pascal-Programm implementiert diesen Algorithmus. In der Matrix M wird dabei aber nicht das Element $m_{ij} = j$ gespeichert, sondern vielmehr wie oft dieses Element gestrichen wurde. Das Streichen eines Elements entspricht dann dem Erhöhen dieses Elements um 1. Die Schritte 4+5 werden dann durch Erniedrigen der veränderten Elemente um 1 rückgängig gemacht.

```
program antsymm;
const Basis = 10;
type TDigit  = 0..Basis-1;
     TAbb    = array[TDigit] of TDigit;
     TMatrix = array[TDigit,TDigit] of TDigit;
var  phi : TAbb;
     M   : TMatrix;

procedure AntiSymm(i : TDigit);
var j,k : Integer;
begin
  if i=Basis then {phi ausgeben}
  else begin
    for j:=0 to Basis-1 do          {Für jedes Element der i-ten Zeile}
      if M[i,j]=0 then begin        {Wenn M[i,j] nicht gestrichen ist,}
        phi[i]:=j;                  {dann setze phi(i):=j}
        for k:=i+1 to Basis-1 do begin
          Inc(M[k,j]);              {Streiche die Elemente M[k,j]}
          Inc(M[k,j*k*i^(-1)]);     {und M[k,j*k*i^(-1)]}
        end;
        AntiSymm(i+1);              {Rekursiver Aufruf}
        for k:=i+1 to Basis-1 do begin
          Dec(M[k,j]);              {Streichungen rückgängig machen}
          Dec(M[k,j*k*i^(-1)]);
        end;
      end;
  end;
end;

var j,k : TDigit;
begin
  for j:=0 to Basis-1 do
    for k:=0 to Basis-1 do M[k,j]:=0;
  AntiSymm(0);
end.
```

Damit das Programm lauffähig ist, muß noch die Verknüpfung $*$ und das Inverse i^{-1} eines Elements definiert werden. Für die Diedergruppe lautet eine entsprechende Implementierung (vgl. H.P. GUMM [13]):

```
const Basis_2 = Basis div 2;  {Basis muß gerade sein!}

function inv(x : TDigit) : TDigit;
begin
  if x<Basis_2 then inv:=(Basis_2-x) mod Basis_2
               else inv:=x;
end;

function add(x,y : TDigit) : TDigit;
begin
  if x<Basis_2 then begin
    if y<Basis_2 then add:= (x+y) mod Basis_2
                 else add:=((x+y) mod Basis_2)+Basis_2
  end
  else begin
    if y<Basis_2 then add:=((x-y) mod Basis_2)+Basis_2
                 else add:=(x-y+Basis_2) mod Basis_2
  end;
end;
```

Damit ist `j*k*i^(-1)=add(add(j,k),inv(i))`. Für die Gruppe $\mathbb{Z}_n$ kann man die Operationen $+$,$-$ und **mod** benutzen, d.h.

```
j*k*i^(-1)=(Basis+j+k-i) mod Basis.
```

Bei der Konstruktion aller anti-symmetrischen Abbildungen einer Gruppe können wir noch folgende Eigenschaft ausnutzen:

Lemma 12 *Sei $g \in Ant(G)$, dann existiert eine eindeutig bestimmte anti-symmetrische Abbildung $g_0 \in Ant(G)$, mit $g_0(0) = 0$, und ein eindeutig bestimmtes $b \in G$ mit $g = l_b \circ g_0$.*

Beweis Sei $g_0 = l_{g(0)^{-1}} \circ g$, dann ist $g_0 \in Ant(G)$ und es gilt $g_0(0) = g(0)^{-1} \cdot g(0) = 0$. Wenn $l_{b_1} \circ g_0 = l_{b_2} \circ h_0$ gilt, mit $g_0(0) = 0 = h_0(0)$, dann folgt $b_1 = b_1 \cdot g_0(0) = b_2 \cdot h_0(0) = b_2$ und damit $g_0 = h_0$. $\square$

Wir müssen also lediglich die anti-symmetrischen Abbildungen g_0 konstruieren, für die $g_0(0) = 0$ ist. Alle anderen erhalten wir dann durch die Links-Multiplikation

mit einem Element aus der Gruppe. Für den Algorithmus bedeutet dies, daß $\varphi(0) = 0$ gesetzt wird und in der Matrix M die Elemente $m_{k,0}$ und $m_{k,k}$, $k = 1, \ldots, n-1$, gestrichen werden. Der Aufruf erfolgt dann mit `AntiSymm(1)`.

Aus diesem Lemma folgt auch, daß die Anzahl der anti-symmetrischen Abbildungen durch die Anzahl der Elemente der Gruppe teilbar ist.

Mit diesem Programm haben wir die Anzahl der anti-symmetrischen Abbildungen der Diedergruppen D_3 bis D_8 und der zyklischen Gruppen $\mathbb{Z}_3$ bis $\mathbb{Z}_{15}$ bestimmt:

Ordnung $2 \cdot s$	$\lvert Ant(D_s)\rvert$	Rechenzeit	$\lvert Ant(D_s)\rvert/(2s)!$
$2 \cdot 3$	120	nicht meßbar	16,667%
$2 \cdot 4$	1.472	nicht meßbar	3,651%
$2 \cdot 5$	34.040	ca. 50ms	0,938%
$2 \cdot 6$	1.412.928	2,5s	0,295%
$2 \cdot 7$	100.229.976	1m 36s	0,114%
$2 \cdot 8$	6.744.202.240	1h 48m 40s	0,032%

Ordnung n	$\lvert Ant(\mathbb{Z}_n)\rvert$	Rechenzeit	$\lvert Ant(\mathbb{Z}_n)\rvert/n!$
3	3	nicht meßbar	50,000%
5	15	nicht meßbar	12,500%
7	133	nicht meßbar	2,639%
9	2.025	nicht meßbar	0,558%
11	37.851	ca. 110ms	0,095%
13	1.030.367	4s	0,017%
15	36.362.925	2m 3s	0,003%

Bemerkung Für n gerade haben wir bereits gezeigt, daß $|Ant(\mathbb{Z}_n)| = 0$ gilt.

Kapitel 3

Gruppen mit Vorzeichen und ihre anti-symmetrischen Abbildungen

Bei der Untersuchung der Diedergruppen stießen wir unabhängig von J. Širáň, M. Škoviera [24] auf den Begriff des Vorzeichens eines Gruppenelements. Im ersten Abschnitt beweisen wir zunächst einige, von J. Širáň, M. Škoviera erwähnte, grundlegende Eigenschaften. Danach zeigen wir, daß Gruppen der Ordnung $4k+2$ eine nicht-triviale Vorzeichenfunktion besitzen. Damit können wir einen sehr kurzen Beweis von Theorem 2 (Siemon, Seite 19) ableiten. Einen wichtigen Spezialfall untersuchen wir im Abschnitt „Anti-symmetrische Abbildungen der Diedergruppe". Wie bereits gezeigt, ist die Diedergruppe D_5 die einzige Gruppe der Ordnung 10, über der ein Prüfziffersystem existiert.

3.1 Gruppen mit Vorzeichen

Definition 8 *Eine Gruppe $(G,\cdot)$ besitzt das Vorzeichen sgn_G, falls $sgn_G : G \to \{-1,+1\}$ ein Homomorhpismus ist, d.h. $sgn_G(x\cdot y) = sgn_G(x)\cdot sgn_G(y)$. Ein Vorzeichen sgn_G heißt nicht-trivial, wenn sgn_G surjektiv ist. Das Vorzeichen eines Gruppenelements $x \in G$ ist $sgn_G(x)$. Die Elemente mit Vorzeichen $+1$ heißen positiv und die mit Vorzeichen -1 negativ. Die Menge aller positiven Elemente von G werde mit G^+, die der negativen Elemente mit G^- bezeichnet.*

Jede Gruppe G der Ordnung n besitzt das triviale Vorzeichen $x \mapsto 1$. Eine weitere Möglichkeit ein Vorzeichen auf G zu definieren erhalten wir, wenn wir G in die symmetrische Gruppe einbetten: Mit $l_a = (x \mapsto a\cdot x)$, der Linksmultiplikation mit a, ist $\alpha = (a \mapsto l_a)$ ein Isomorphismus von G auf $\bar{G} = \{l_a | a \in G\} \subseteq S_n$. Auf S_n können wir die Signatur eines Elements als Vorzeichen benutzen (vgl. Meyberg [17]):

$$sgn_G(a) := sgn_{\bar{G}}(l_a) \tag{3.1}$$

wobei $sgn_{\bar{G}}(l_a) = 1$, falls l_a als Produkt von einer geraden Anzahl Transpositionen dargestellt werden kann und $sgn_{\bar{G}}(l_a) = -1$ sonst. Ist $\bar{G}$ keine Untergruppe von A_n, so ist dieses Vorzeichen nicht-trivial.

Für das Vorzeichen gilt:

1. $sgn_G(e) = 1$.
2. $sgn_G(x^{-1}) = sgn_G(x)^{-1} = sgn_G(x)$.
3. $G = G^+ \cup G^-$.

Lemma 13 *G^+ ist ein Normalteiler mit Index ≤ 2.*

Beweis Als Kern des Homomorphismus sgn_G ist G^+ ein Normalteiler in G und alle Nebenklassen eines Normalteilers sind gleichmächtig. □

Korollar 12 *Gruppen mit ungerader Ordnung und einfache Gruppen (außer $\mathbb{Z}_2$) besitzen nur die triviale Vorzeichenfunktion $sgn_G : x \mapsto 1$.*

Elemente mit ungerader Ordnung haben das Vorzeichen 1, denn aus $ord(x) = 2k + 1$ folgt

$$sgn(x) = sgn(x^{2k+2}) = sgn(x)^{2k+2} = sgn(x)^{k+1} sgn(x)^{k+1} = 1.$$

Einfache Gruppen (außer $\mathbb{Z}_2$) haben keine Normalteiler mit Index 2.

Lemma 14 *Ist U eine Untergruppe mit Index 2 in G, dann wird durch*

$$sgn(x) = \begin{cases} 1 & \text{falls } x \in U \\ -1 & \text{sonst} \end{cases}$$

ein Vorzeichen auf G definiert.

Beweis U ist ein Normalteiler in G und es gilt

$$x \cdot y \in U \qquad \Leftrightarrow \qquad x, y \in U \text{ oder } x, y \notin U$$

und

$$x \cdot y \notin U \qquad \Leftrightarrow \qquad x \in U, y \notin U \text{ oder } x \notin U, y \in U.$$

Folglich ist $sgn : G \to \{-1, 1\}$ ein Homomorphismus. □

Die letzten beiden Lemmata fassen wir wie folgt zusammen

Satz 13 *Eine Gruppe besitzt eine nicht-triviale Vorzeichenfunktion genau dann, wenn sie eine Untergruppe mit Index 2 besitzt.*

Bemerkung Falls G ein nicht-triviales Vorzeichen besitzt, so gilt $G/G^+ \cong \mathbb{Z}_2$, d.h. man kann die Gruppe G als Erweiterung von G^+ durch $\mathbb{Z}_2$ ansehen.
Beispiel Die zyklische Gruppe $\mathbb{Z}_n$ besitzt ein nicht-triviales Vorzeichen genau dann, wenn n gerade ist. Ist n ungerade, dann hat $\mathbb{Z}_n$ kein Vorzeichen. Ist n gerade dann wird durch

$$sgn(x) = \begin{cases} 1 & \text{falls } x \text{ gerade} \\ -1 & \text{falls } x \text{ ungerade} \end{cases}$$

ein Vorzeichen auf $\mathbb{Z}_n$ definiert.

Beispiel Auf der Gruppe der Anti-Automorphismen vereinigt mit den Automorphismen einer Gruppe G wird durch $sgn(\varphi) = 1$, falls φ ein Automorphismus ist und $sgn(\varphi) = -1$, falls φ ein Anti-Automorphismus ist, ein Vorzeichen definiert. Dieses Vorzeichen ist genau dann nicht trivial, wenn G nicht abelsch ist (vgl. Abschnitt „Automorphismen und Anti-Automorphismen").

Satz 14 *Eine Gruppe G der Ordnung $n = 2(2k+1)$, $k \geq 1$, besitzt ein nicht-triviales Vorzeichen.*

Beweis Wir zeigen, daß das in 3.1 definierte Vorzeichen nicht-trivial ist, d.h. es existiert ein $a \in G$ mit $sgn_G(a) = sgn_{\bar{G}}(l_a) = -1$. Da die Ordnung von G gerade ist, existiert ein $a \in G$ der Ordnung 2. Daher ist $l_a \circ l_a = Id$ und l_a besteht aus $2k+1$ Transpositionen ($x_i \in G$ geeignet)

$$l_a = (e \;\; l_a(e))(x_2 \;\; l_a(x_2)) \ldots (x_{2k+1} \;\; l_a(x_{2k+1})).$$

Da $2k+1$ ungerade ist, folgt $sgn_G(a) = sgn_{\bar{G}}(l_a) = -1$. □

Korollar 13 *Jede Gruppe der Ordnung $n = 2(2k+1)$, $k \geq 1$, besitzt einen Normalteiler der Ordnung $2k+1$.*

Beweis Da eine solche Gruppe ein nicht-triviales Vorzeichen besitzt, hat die zugehörige Menge der positiven Elemente G^+ die Ordnung $2k+1$ und ist daher ein Normalteiler mit der gesuchten Eigenschaft. □

Wir geben nun den noch fehlenden Beweis von Theorem 2 an. Wir müssen zeigen, daß eine Gruppe G der Ordnung $2(2k+1)$, $k \geq 1$, keine vollständige Abbildung besitzt. G besitzt ein nicht-triviales Vorzeichen sgn (Satz 14) und G^+

ist ein Normalteiler der Ordnung $2k+1$ (Korollar 13).
Wenn G eine vollständige Abbildung f besitzt, dann folgt:

$$\begin{aligned}\prod_{x\in G} sgn(x) &= \prod_{x\in G^+} sgn(x) \prod_{x\in G^-} sgn(x) \\ &= 1\cdot(-1)^{|G^-|} = (-1)^{|G^+|} = (-1)^{2k+1} = -1\end{aligned}$$

und

$$\begin{aligned}\prod_{x\in G} sgn(x) &= \prod_{x\in G} sgn(xf(x)) = \prod_{x\in G} sgn(x) \prod_{x\in G} sgn(f(x)) \\ &= \prod_{x\in G} sgn(x) \prod_{x\in G} sgn(x) = (-1)\cdot(-1) = 1,\end{aligned}$$

also ein Widerspruch. Demnach kann G keine vollständige Abbildung besitzen. □

3.2 Anti-symmetrische Abbildungen

In diesem Abschnitt zeigen wir weitere Möglichkeiten, wie wir aus einer anti-symmetrischen Abbildung neue konstruieren können.

Wir nennen zwei Permutationen p, q elementfremd, wenn für alle x gilt:

$$p(x) = x \qquad \text{oder} \qquad q(x) = x.$$

Satz 15 *Sei $(G,\cdot)$ eine Gruppe mit Vorzeichen sgn, $g, g\circ r \in Ant(G)$, und es gelte für alle $x \in G$: $sgn(g(x)) = c\cdot sgn(x)$ und $sgn(r(x)) \neq sgn(x)$ mit $c \in \{-1,1\}$, dann ist für jede Zerlegung von r in elementfremde Faktoren, $r = p\circ q$, auch $g\circ p \in Ant(G)$.*

Beweis Annahme: $g\circ p$ ist nicht anti-symmetrisch, d.h. es existieren $a, b \in G$, $a \neq b$ mit

$$g\circ p(a)\cdot b = g\circ p(b)\cdot a.$$

Offensichtlich gilt dann $p(a) \neq a$ oder $p(b) \neq b$, sonst wäre g nicht anti-symmetrisch:

$$g\circ p(a)\cdot b = g(a)\cdot b = g(b)\cdot a = g\circ p(b)\cdot a.$$

Wenn dagegen $p(a) \neq a$ und $p(b) \neq b$ gilt, dann kommen a und b nicht in q vor

und es folgt $q(a) = a$, $q(b) = b$ und damit

$$\begin{aligned} g \circ r(a) \cdot b &= g \circ p \circ q(a) \cdot b \\ &= g \circ p(a) \cdot b \\ &= g \circ p(b) \cdot a \\ &= g \circ p \circ q(b) \cdot a \\ &= g \circ r(b) \cdot a \end{aligned}$$

im Widerspruch zur Voraussetzung $q \circ r \in Ant(G)$. Es bleibt daher nur die Möglichkeit $p(a) \neq a, p(b) = b$. (Die Annahme ist in a und b symmetrisch, es ist also auch $p(a) = a, p(b) \neq b$ abgedeckt.) Auch in diesem Fall kommt a in p vor und damit nicht in q, d.h. $q(a) = a$. Aus der Annahme folgt somit die Gleichung

$$g \circ r(a) \cdot b = g \circ p \circ q(a) \cdot b = g \circ p(a) \cdot b = g \circ p(b) \cdot a = g(b) \cdot a.$$

Ein Vergleich der Vorzeichen zeigt aber

$$sgn(g \circ r(a) \cdot b) = c \cdot sgn(r(a))sgn(b) \neq c \cdot sgn(a)sgn(b) = sgn(g(b) \cdot a).$$

Damit ist die Annahme widerlegt, d.h. es gilt für alle $a, b \in G$, $a \neq b$, $g \circ p(a) \cdot b \neq g \circ p(b) \cdot a$, folglich ist $g \circ p$ eine anti-symmetrische Abbildung. □

Ein ähnlicher Satz gilt für eine nachgeschaltete Permutation.

Satz 16 *Sei $(G, \cdot)$ eine Gruppe mit Vorzeichen sgn, $g, r \circ g \in Ant(G)$ und es gelte für alle $x \in G$: $sgn(g(x)) = c \cdot sgn(x)$ und $sgn(r(x)) \neq sgn(x)$, mit $c \in \{-1, 1\}$, dann ist für jede Zerlegung von r in elementfremde Faktoren, $r = p \circ q$, auch $p \circ g \in Ant(G)$.*

Beweis Seien $\tilde{r} := g^{-1} \circ r \circ g$, $\tilde{p} := g^{-1} \circ p \circ g$ und $\tilde{q} := g^{-1} \circ q \circ g$, dann ist $g \circ \tilde{r} = r \circ g$ anti-symmetrisch und $\tilde{r} = \tilde{p} \circ \tilde{q}$. Die Permutationen $\tilde{p}$ und $\tilde{q}$ sind elementfremd, denn wenn $\tilde{p}(a) \neq a$ gilt, dann folgt $p(g(a)) \neq g(a)$ und, da p und q elementfremd sind, $q(g(a)) = g(a)$. Also gilt $\tilde{q}(a) = g^{-1}(q(g(a))) = g^{-1}(g(a)) = a$ und a kommt in q nicht vor. Außerdem gilt

$$\begin{aligned} sgn(\tilde{r}(x)) &= sgn(g^{-1} \circ r \circ g(x)) = c \cdot sgn(g \circ g^{-1} \circ r \circ g(x)) \\ &= c \cdot sgn(r \circ g(x)) \\ &\neq c \cdot sgn(g(x)) = sgn(x). \end{aligned}$$

Nun sind die Voraussetzungen des vorherigen Satzes für $\tilde{r}$,$\tilde{p}$ und $\tilde{q}$ erfüllt und es folgt $g \circ \tilde{p} = g \circ g^{-1} \circ p \circ g = p \circ g \in Ant(G)$. □

Wie der Beweis zeigt, besitzt jede anti-symmetrische Abbildung $\tilde{p} \circ g$ ($\tilde{p}, g$ gemäß Satz 16) ein weitere Darstellung $g \circ p$ (p gemäß Satz 15).

Am Beweis von Satz 15 sehen wir außerdem, daß wir das nicht-trivale Vorzeichen der Gruppe nur an einer Stelle benutzen. Für beliebige Gruppen gilt daher:

Satz 17 *Sei $(G, \cdot)$ eine Gruppe, $g, g \circ r \in Ant(G)$ und es gelte für alle $x, y \in G$*

$$g \circ r(x) \cdot y = g(y) \cdot x \Rightarrow x = y,$$

dann ist für jede Zerlegung von r in elementfremde Faktoren, $r = p \circ q$, auch $g \circ p \in Ant(G)$.

Bemerkung Aus $g, g \circ r \in Ant(G)$ folgt nicht $g \circ r(x) \cdot y = g(y) \cdot x \Rightarrow x = y$ wie folgendes Gegenbeispiel der Diedergruppe D_5 (Gruppentafel auf Seite 52) zeigt: Sei $g(x) = x^{-1} \cdot 2$ und $r(x) = x \cdot 1$, dann gilt $g, g \circ r \in Ant(D_5)$ (Satz 20, Seite 52), aber

$$g \circ r(0) \cdot 3 = g(1) \cdot 3 = 1 \cdot 3 = 4 = 2 \cdot 2 = g(3) \cdot 0.$$

Für die speziellen anti-symmetrischen Abbildungen $g(x) = b \cdot x^{-1}a$ können wir konkret angeben, für welche p die Voraussetzungen von Satz 15 erfüllt sind.

Satz 18 *Sei $(G, \cdot)$ eine Gruppe mit Vorzeichen sgn, und $g(x) = b \cdot x^{-1}a$ sei eine anti-symmetrische Abbildung. Zudem erfülle die Permutation*

$$p = (k_1 \ \ l_1)(k_2 \ \ l_2) \ldots (k_n \ \ l_n),$$

wobei $(k_i \ \ l_i)$ eine Transposition ist ($k_i, l_i \in G$), die Bedingungen:

1. *$l_1^{-1} \cdot k_1 = l_j^{-1} \cdot k_j$, für $j = 2, \ldots, n$*
2. *$ord(l_1^{-1} \cdot k_1) = 2$, $sgn(l_1^{-1} \cdot k_1) = -1$*

dann ist auch $g \circ p(x)$ anti-symmetrisch.

Beweis Es sei $p = (k_1 \ \ l_1)(k_2 \ \ l_2) \ldots (k_n \ \ l_n)$ eine Permutation mit den genannten Eigenschaften. Aus den Voraussetzungen folgt, daß die Transpositionen $(k_i \ \ l_i)$ und $(k_j \ \ l_j)$ entweder gleich oder elementfremd sind. Wir nehmen daher o.B.d.A. an, daß sie paarweise elementfremd sind. Sei $c := l_1^{-1} \cdot k_1$ und $r(x) := x \cdot c$, dann ist $g \circ r \in Ant(G)$ und man sieht leicht, daß für die Zerlegung von r in elementfremde Faktoren

$$r = (k_1 \ \ l_1)(k_2 \ \ l_2) \ldots (k_n \ \ l_n) \circ q = p \circ q$$

gilt, denn $r(l_i) = l_i \cdot l_1^{-1} \cdot k_1 = l_i \cdot l_i^{-1} \cdot k_i = k_i$ und $r \circ r = id$. Außerdem ist $sgn(g(x)) = sgn(b \cdot x^{-1} \cdot a) = sgn(a \cdot b)sgn(x)$ und $sgn(r(x)) = sgn(x \cdot l_1^{-1} \cdot k_1) = sgn(x)sgn(l_1^{-1} \cdot k_1) = -sgn(x) \neq sgn(x)$. Damit sind die Voraussetzungen von Satz 15 für g und $r = p \circ q$ erfüllt und es folgt $g \circ p \in Ant(G)$. □

Ganz analog zeigt man den folgenden Satz, der ein Spezialfall von Satz 16 ist.

Satz 19 *Sei $(G, \cdot)$ eine Gruppe mit Vorzeichen sgn, und $g(x) = b \cdot x^{-1} \cdot a \in Ant(G)$. Zudem erfülle die Permutation*

$$p = (k_1 \ \ l_1)(k_2 \ \ l_2) \ldots (k_n \ \ l_n)$$

die folgenden Bedingungen:

1. $l_1 \cdot k_1^{-1} = l_j \cdot k_j^{-1}$, *für* $j = 2, \ldots, n$
2. $ord(l_1 \cdot k_1^{-1}) = 2$, $sgn(l_1 \cdot k_1^{-1}) = -1$

dann ist auch $p \circ g(x)$ anti-symmetrisch.

3.3 Anti-symmetrische Abbildungen der Diedergruppe

Die Diedergruppe D_5 spielt eine wichtige Rolle bei der Suche nach einem Prüfziffersystem, denn sie ist die einzige Gruppe der Ordnung 10 die eine anti-symmetrische Abbildung besitzt. In diesem Abschnitt zeigen wir daher einige Eigenschaften der anti-symmetrischen Abbildungen der Diedergruppe. Dazu benutzen wir die Matrixschreibweise der Diedergruppe (H.P. Gumm [12]) ($s > 2$)

$$D_s = \{(e, x) | e \in \{-1, 1\} \text{ und } x \in \mathbb{Z}_s\}$$

mit der Verknüpfung

$$(e, x) \cdot (f, y) := (e \cdot f, e \cdot y + x).$$

In der ersten Komponente wird in der multiplikativ geschriebenen Gruppe $\mathbb{Z}_2$ gerechnet, in der zweiten im Ring $\mathbb{Z}_s$. Den Paaren (e, x) ordnen wir die Ziffern $\{0, \ldots, 2s-1\}$ auf folgende Weise zu:

$$(1, x) \mapsto x \qquad (-1, x) \mapsto s + x$$

Für D_5 haben wir damit die Gruppentafel:

$\cdot$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Durch die Matrixschreibweise können wir eine anti-symmetrische Abbildung $g \in Ant(D_s)$ in der Form

$$g(e,x) = (g_1(e,x), g_2(e,x))$$

schreiben, wobei $g_1 : D_s \to \{-1,1\}$ und $g_2 : D_s \to \{0,\ldots,s-1\}$ surjektive Abbildungen sind.

Als Folgerung von Theorem 4 (Seite 24) und Satz 3 (Seite 30) erhalten wir

Satz 20 *Die Abbildungen $b \cdot x^{-1} \cdot a$ mit $b \in D_s$, ($s > 2$ ungerade) und $a \in \{1,...,s-1\}$ sind anti-symmetrisch.*

Beweis Die Abbildung $x^{-1} \cdot a$, $a \in \{1,...,s-1\}$ ist anti-symmetrisch, da a mit keinem Element der Ordnung 2 kommutiert: Weil s ungerade ist, haben nur die Elemente $(-1,c)$ die Ordnung 2, denn aus $(1,x) \cdot (1,x) = (1,2x) = (1,0)$ folgt $x = 0$. Angenommen $a = (1,a)$ kommutiert mit dem Element $(-1,c)$, d.h.

$$(1,a) \cdot (-1,c) = (-1,c+a) = (-1,c-a) = (-1,c) \cdot (1,a).$$

Es folgt $c + a = c - a$ bzw. $2a = 0$ und damit $a = 0$, im Widerspruch zu $a \in \{1,...,s-1\}$. Also ist $x^{-1}a$ und somit auch $bx^{-1}a$ anti-symmetrisch. □

Bei einem Vergleich dieses Satzes mit dem vorherigen Abschnitt, stellt sich die Frage, ob Satz 18 anwendbar ist. Dies ist möglich, wie der folgende Satz zeigt. Zunächst führen wir allerdings das Vorzeichen eines Elements der Diedergruppe ein. Auf der Diedergruppe wird durch die Funktion $sgn : D_s \to \{-1,1\}$,

$$sgn(e,x) := e$$

ein nicht-triviales Vorzeichen definiert, denn es gilt

$$sgn((e,x) \cdot (f,y)) = sgn(ef, ey + x) = ef = sgn(e,x)sgn(f,y).$$

Wenn wir die Darstellung der Diedergruppe mit den Ziffern $\{0, \ldots, 2s-1\}$ benutzen, dann ist $sgn(x) = sgn(1,x) = 1$, $0 \leq x \leq s-1$ und $sgn(s+x) = sgn(-1,x) = -1$, $s \leq s+x \leq 2s-1$.

Satz 21 *Sei* $g(x) := b \cdot x^{-1} \cdot a \in Ant(D_s)$ *und* $p := (k_1\ l_1)(k_2\ l_2)\ldots(k_n\ l_n)$ *mit den Eigenschaften* $s \leq k_i \leq 2s-1$, $0 \leq l_i \leq s-1$, $k_i \neq k_j$ *und* $l_1^{-1} \cdot k_1 = l_j^{-1} \cdot k_j$, *bzw.* $l_1 \cdot k_1^{-1} = l_j \cdot k_j^{-1}$, $j = 2, \ldots, n$, *dann ist auch* $g \circ p$ *bzw.* $p \circ g \in Ant(D_s)$.

Beweis Es ist $sgn(k_1) = -1 \neq 1 = sgn(l_1)$ und $(l_1^{-1} \cdot k_1) \cdot (l_1^{-1} \cdot k_1) = (1,y) \cdot (-1,x) \cdot (1,y) \cdot (-1,x) = (-1, x+y) \cdot (-1, x+y) = (1, -x-y+x+y) = (1,0)$ also $ord(l_1^{-1} \cdot k_1) = ord(l_1 \cdot k_1^{-1}) = 2$. Mit Satz 18 bzw. Satz 19 folgt die Behauptung. □

Mit diesem Satz können wir 3040 anti-symmetrische Abbildungen der Diedergruppe D_5 aus den Permutationen $b \cdot x^{-1} \cdot a$ konstruieren (bislang waren nur 40 bekannt).

Beispiel Wir wählen $b = 0, a = 1$, dann ist $g(x) = x^{-1} \cdot 1 = (10)(42)(98765) \in Ant(D_5)$. Aus Satz 21 folgt, daß z.B. auch $g \circ (50) \circ (61) \in Ant(D_5)$ oder $g \circ (61) \cdot (83) \cdot (94) = (50) \circ (73) \circ (82) \circ g \in Ant(D_5)$.

3.3.1 Fehlererkennung

Für die Komponentenfunktionen g_1, g_2 der Diedergruppe können wir weitere Eigenschaften zeigen. Außerdem beweisen wir am Ende dieses Abschnitts, daß über den Diedergruppen D_s, $s > 2$ ungerade, kein Prüfziffersystem existiert, welches alle Sprungtranspositionen erkennt.

Satz 22 *Sei* $g \in Ant(D_s)$ $(s > 2)$, $g(e,x) = (g_1(e,x), g_2(e,x))$, *dann hängt* g_2 *von* e *und von* x *ab.*

Beweis Fall 1: g_2 hänge nur von e ab, d.h. $g_2(e,x) = g_2(e)$. In diesem Fall besitzt $g_2(D_s)$ höchstens zwei Elemente (nämlich $g_2(1,0)$ und $g_2(-1,0)$) und $g(D_s)$ besitzt höchstens vier Elemente $(\pm 1, g_2(\pm 1, 0))$. Damit kann g keine Permutation sein, da D_s für $s > 2$ mindestens sechs Elemente hat. Also ist $g \notin Ant(D_s)$.
Fall 2: g_2 hänge nur von x ab, d.h. $g_2(e,x) = g_2(x)$. Auch hier folgt, daß g nicht anti-symmetrisch ist, denn es gilt entweder $g_1(1,0) = g_1(-1,0)$ und g wäre nicht injektiv, $g(1,0) = (g_1(1,0), g_2(0)) = (g_1(-1,0), g_2(0)) = g(-1,0)$, oder $g_1(1,0) = -g_1(-1,0)$ und damit

$$\begin{aligned} g(1,0) \cdot (-1,0) &= (g_1(1,0), g_2(0)) \cdot (-1,0) = (-g_1(1,0), g_2(0)) \\ &= (g_1(-1,0), g_2(0)) \\ &= g(-1,0) \cdot (1,0). \end{aligned}$$

Folglich ist g nur anti-symmetrisch wenn g_2 von e und von x abhängt. $\square$

Satz 23 *Sei $g \in Ant(D_s)$ ($s > 2$ ungerade), $g(e,x) = (g_1(e,x), g_2(e,x))$, dann hängt g_1 entweder nur von e oder von e und x ab.*

Beweis Wenn g_1 nur von x abhängt, dann gilt für alle $x \in \{0, \ldots, s-1\}$ $g_1(1,x) = g_1(-1,x)$. Die Anzahl $a_1 := |\{(e,x) \in D_s | g_1(e,x) = 1\}|$ und $a_{-1} := |\{(e,x) \in D_s | g_1(e,x) = -1\}|$ der Elemente, für die $g_1(e,x)$ gleich 1 bzw. -1 ist, muß daher gerade sein, $a_i = 2k_i$. Weiterhin gilt $a_1 + a_{-1} = 2k_1 + 2k_{-1} = |D_s| = 2s$, woraus $k_1 + k_{-1} = s$ folgt. Da s ungerade ist, muß $k_1 \neq k_{-1}$ und deshalb $a_1 \neq a_{-1}$ gelten. Damit kann g nicht injektiv sein, denn es gilt $|\{(e,x) \in D_s | e = 1\}| = s = |\{(e,x) \in D_s | e = -1\}|$. Also ist g im Fall, daß g_1 nur von x abhängt, nicht anti-symmetrisch. $\square$

Satz 24 *Sei $g \in Ant(D_s)$, $s > 2$, $g(e,x) = (e, g_2(e,x))$, dann sind $g_2(1,x)$ und $g_2(-1,-x)$ anti-symmetrische Abbildungen von $\mathbb{Z}_s$.*

Beweis Sei $c \in \{-1,1\}$, wir zeigen $g_2(c,cx) \in Ant(\mathbb{Z}_s)$. Dazu sei $g_2(c,cx) + y = g_2(c,cy) + x$. Es folgt $g(c,cx) \cdot (c,cy) = (1, y + g_2(c,cx)) = (1, x + g_2(c,cy)) = g(c,cy) \cdot (c,cx)$ und damit $(c,cx) = (c,cy)$ bzw. $x = y$. $\square$

Satz 25 *Sei $g \in Ant(D_s)$ ($s > 2$ gerade), $g(e,x) = (g_1(e,x), g_2(e,x))$, dann hängt g_1 entweder nur von x oder von e und x ab.*

Beweis Wenn g_1 nur von e abhängt, dann wäre $g_2(1,x)$ eine anti-symmetrische Abbildung von $\mathbb{Z}_s$. Wir haben aber bereits gezeigt, daß $\mathbb{Z}_s$ für gerades s keine anti-symmetrische Abbildung besitzt. $\square$

Abschließend zeigen wir eine wichtige Eigenschaft der anti-symmetrischen Abbildungen der Diedergruppe.

Satz 26 *Sei $g \in Ant(D_s)$, $s > 2$ ungerade, dann existiert ein $c \in D_s$, so daß $g(x) \cdot c \notin Ant(D_s)$ ist.*

Korollar 14 *Über der Diedergruppe D_s, $s > 2$ ungerade, existiert kein Prüfziffersystem das alle Sprungtranspositionen erkennt.*

Beweis Es existiert ein Element $(-1,x) \in D_s$, so daß $-g_1(1,0) = g_1(-1,x)$ gilt, denn sonst hätten wir mindestens $s+1$ Elemente mit dem gleichen Vorzeichen und g wäre keine Permutation. In $\mathbb{Z}_s$ existiert ein multiplikativ Inverses von 2, nämlich $1/2 = k+1$, wenn $s = 2k+1$ ist. Wir setzen

$$c := \big(1, 1/2 \cdot g_1(-1,x)\big(g_1(1,0)x + g_2(1,0) - g_2(-1,x)\big)\big)$$

und es folgt

$$\begin{aligned}
g(1,0)\cdot c\cdot(-1,x) &= \big(g_1(1,0),g_2(1,0)\big)\cdot\big(1,1/2\cdot g_1(-1,x)\cdot\\
&\quad \big(g_1(1,0)x+g_2(1,0)-g_2(-1,x)\big)\big)\cdot\big(-1,x\big)\\
&= \big(g_1(1,0),g_1(1,0)\cdot 1/2\cdot g_1(-1,x)\cdot\\
&\quad \big(g_1(1,0)x+g_2(1,0)-g_2(-1,x)\big)+g_2(1,0)\big)\cdot\big(-1,x\big)\\
&= \big(-g_1(1,0),g_1(1,0)x+g_1(1,0)\cdot 1/2\cdot g_1(-1,x)\cdot\\
&\quad \big(g_1(1,0)x+g_2(1,0)-g_2(-1,x)\big)+g_2(1,0)\big)\\
&= \big(-g_1(1,0),1/2\big(g_1(1,0)x+g_2(1,0)+g_2(-1,x)\big)\big)\\
g(-1,x)\cdot c\cdot(1,0) &= g(-1,x)\cdot c\\
&= \big(g_1(-1,x),g_2(-1,x)\big)\cdot\big(1,1/2\cdot g_1(-1,x)\cdot\\
&\quad \big(g_1(1,0)x+g_2(1,0)-g_2(-1,x)\big)\big)\\
&= \big(g_1(-1,x),g_1(-1,x)\cdot 1/2\cdot g_1(-1,x)\cdot\\
&\quad \big(g_1(1,0)x+g_2(1,0)-g_2(-1,x)\big)+g_2(-1,x)\big)\\
&= \big(-g_1(1,0),1/2\big(g_1(1,0)x+g_2(1,0)+g_2(-1,x)\big)\big)
\end{aligned}$$

also $g(1,0)\cdot c\cdot(-1,x) = g(-1,x)\cdot c\cdot(1,0)$ und $g(x)\cdot c$ ist nicht anti-symmetrisch. Außerdem erkennt g nicht alle Sprungtranspositionen (vgl. Seite 14). □

Zusammen mit Korollar 4 (Seite 19) haben wir damit gezeigt:

Theorem 10 *Sei $s > 2$ ungerade. Über der Gruppe D_s existiert kein Prüfziffersystem, das alle Sprungtranspositionen oder alle Zwillings- oder alle Sprungzwillingsfehler erkennt.*

3.3.2 Automorphismen und Anti-Automorphismen der Diedergruppe

Die Automorphismen und die Anti-Automorphismen sind bei der Bestimmung der Äquivalenzklassen bzw. bei der Suche nach anti-symmetrischen Abbildungen sehr wichtig. Für die Diedergruppe D_s, $s > 2$, kann man diese recht einfach bestimmen. Dazu nutzt man aus, daß die Diedergruppe von den Elementen $(1,1)$ und $(-1,0)$ erzeugt wird. Das Element $(1,1)$ hat in D_s die Ordnung s und $(-1,x)$ hat für alle x die Ordnung 2. Da für jeden Automorphismus oder Anti-Automorphismus φ die Elemente $\varphi(x)$ und x die gleiche Ordnung haben, muß $\varphi(1,1) = (1,d)$ für ein $d \in \mathbb{Z}_s$ gelten. d muß dabei eine Einheit des Ringes $\mathbb{Z}_s$ sein, d.h. $ggT(d,s) = 1$, sonst hätte $(1,d)$ nicht die Ordnung s. Ebenso folgt, daß $\varphi(-1,0) = (-1,c)$ ist mit $c \in \mathbb{Z}_s$, denn andernfalls würde φ alle Elemente auf die Untergruppe

$\{(1,x)|x \in \mathbb{Z}_s\}$ abbilden und φ wäre demnach nicht surjektiv. Wie wir bereits gezeigt haben, reicht es, die Automorphismen einer Gruppe zu bestimmen, denn die Anti-Automorphismen lassen sich durch $inv \circ \varphi$ darstellen, wobei φ ein Automorphismus ist. Mit der Verknüpfung $(e,x)\cdot(f,y) = (ef, ey+x)$ der Diedergruppe erhalten wir die folgenden Eigenschaften des Automorphismus φ:

1. $\varphi(1,x) = \varphi(1,1)^x = (1,d)^x = (1,dx)$, für alle $x \in \mathbb{Z}_s$.
2. $\varphi(-1,x) = \varphi((1,x)\cdot(-1,0)) = \varphi(1,x)\cdot\varphi(-1,0) = (1,dx)\cdot(-1,c) = (-1,c+dx)$, für alle $x \in \mathbb{Z}_s$.

Die Eigenschaften 1 und 2 sind also notwendig dafür, daß φ ein Automorphismus ist. Sie sind aber auch hinreichend. Dazu seien d eine Einheit von $\mathbb{Z}_s$, c ein Element von $\mathbb{Z}_s$ und $\varphi : D_s \to D_s$ eine Abbildung mit $\varphi(1,x) = (1,dx)$, $\varphi(-1,x) = (-1,c+dx)$. Mit dieser Definition ist φ bijektiv, denn aus $\varphi(e,x) = \varphi(f,y)$ folgt $e = f$ und $dx = dy$ bzw. $c+dx = c+dy$ und damit, weil d eine Einheit ist, $x = y$. Also ist φ injektiv und damit auch surjektiv (D_s ist endlich). φ ist außerdem ein Homomorphismus, denn es gilt für alle $x, y \in \mathbb{Z}_s$:

- $\varphi((1,x)\cdot(1,y)) = \varphi(1,x+y) = (1,dx+dy) = (1,dx)\cdot(1,dy) = \varphi(1,x)\cdot\varphi(1,y)$
- $\varphi((1,x)\cdot(-1,y)) = \varphi(-1,y+x) = (-1,c+dy+dx) = (1,dx)\cdot(-1,c+dy) = \varphi(1,x)\cdot\varphi(-1,y)$
- $\varphi((-1,x)\cdot(1,y)) = \varphi(-1,-y+x) = (-1,c-dy+dx) = (-1,c+dx)\cdot(1,dy) = \varphi(-1,x)\cdot\varphi(1,y)$
- $\varphi((-1,x)\cdot(-1,y)) = \varphi(1,-y+x) = (1,-dy+dx) = (-1,c+dx)\cdot(-1,c+dy) = \varphi(-1,x)\cdot\varphi(-1,y)$

Also ist φ ein Automorphismus.

Damit haben wir für die Anti-Automorphismen der Diedergruppe die Darstellung

$$\begin{aligned} inv \circ \varphi(1,x) &= (1,dx)^{-1} = (1,-dx) \\ inv \circ \varphi(-1,x) &= (-1,c+dx)^{-1} = (-1,c+dx). \end{aligned}$$

Der folgende Satz faßt dieses Ergebnis zusammen:

Satz 27 *Seien $\varphi, \psi : D_s \to D_s$ Abbildungen der Diedergruppe D_s,$s > 2$, dann gilt:*

1. *Genau dann ist φ ein Automorphismus, wenn eine Einheit d und ein Element c von $\mathbb{Z}_s$ existieren mit $\varphi(1,x) = (1,dx)$ und $\varphi(-1,x) = (-1,c+dx)$.*

2. *Genau dann ist ψ ein Anti-Automorphismus, wenn eine Einheit d und ein Element c von $\mathbb{Z}_s$ existieren mit $\psi(1,x) = (1,-dx)$ und $\psi(-1,x) = (-1, c + dx)$.*

Wenn s ungerade ist, d.h. $s = 2k+1$, dann besitzt 2 ein multiplikativ Inverses, nämlich $1/2 = k+1$. Die Funktion $(1-e)/2$ ist dann gleich 0, wenn $e = 1$ ist und gleich 1, wenn $e = -1$ ist. Die Automorphismen der Diedergruppe D_s haben daher für $s > 2$ ungerade alle die Form $\varphi(e,x) = (e, (1-e)/2 \cdot c + dx) = (e, (1-e)\tilde{c} + dx) = (e, \tilde{c} - e\tilde{c} + dx)$ und die Anti-Automorphismen haben die Form $\psi(e,x) = (e, \tilde{c} - e\tilde{c} - edx)$ mit $d \in \mathbb{Z}_s^*$ und $\tilde{c} \in \mathbb{Z}_s$.

Damit können wir die folgenden Sätze zeigen:

Satz 28 *Die Diedergruppe D_s $s > 2$ besitzt keinen anti-symmetrischen Automorphismus und sie besitzt einen fixpunktfreien, d.h. anti-symmetrischen, Anti-Automorphismus genau dann, wenn s ungerade ist.*

Beweis Ist s gerade, dann ist $s/2$ das einzige Element der Ordnung 2 in $\mathbb{Z}_s$. Da jeder (Anti-)Automorphismus ψ die Ordnung und das Vorzeichen $e = sgn(e,x)$ erhält, gilt $\psi(1, s/2) = (1, s/2)$ und ψ ist nicht anti-symmetrisch.

Ist s ungerade, dann kommutiert $(1,1)$ mit keinem Element der Ordnung 2, $(1,1) \cdot (-1,x) = (-1, x+1) \neq (-1, x-1) = (-1,x) \cdot (1,1)$, und damit ist $\psi(x) = (1,-1) \cdot x^{-1} \cdot (1,1)$ ein fixpunktfreier Anti-Automorphismus. Ist φ ein Automorphismus mit $\varphi(-1,0) = (-1,c)$, dann definieren wir $z := (-1, 1/2 \cdot c)$ und es folgt

$$\begin{aligned} \varphi(-1,0) &= (-1,c) = (-1, 1/2 \cdot c + 1/2 \cdot c) \\ &= (-1, 1/2 \cdot c)(-1,0)(-1, 1/2 \cdot c) \\ &= z^{-1}(-1,0)z. \end{aligned}$$

Also ist φ nicht anti-symmetrisch (Satz 8, Seite 35). □

Satz 29 *Sei ψ ein Anti-Automorphismus der Diedergruppe D_s, d.h. $\psi(1,x) = (1,dx)$ und $\psi(-1,x) = (-1, c-dx)$ mit $c \in \mathbb{Z}_s$, $d \in \mathbb{Z}_s^*$, dann ist ψ genau dann fixpunktfrei, wenn $d-1$ eine Einheit und $d+1$ kein Teiler von c (in $\mathbb{Z}_s$) ist.*

Beweis Wenn ψ einen Fixpunkt $(1,x) \neq (1,0)$ oder $(-1,x)$ besitzt, dann gilt $\psi(1,x) = (1,dx) = (1,x)$ oder $\psi(-1,x) = (-1, c-dx) = (-1,x)$. Im ersten Fall folgt $dx = x$ bzw. $(d-1)x = 0$ und $d-1$ ist keine Einheit. Im zweiten Fall ist $c - dx = x$ also $c = (d+1)x$ und $d+1$ teilt c. Da nur Äquivalenzumformungen benutzt wurden, folgt damit auch die Rückrichtung. □

Beispiel Für die Diedergruppe D_5 können wir $c = 1,2,3,4$ und $d = 4$ wählen.

3.3.3 Beispiele

In diesem Abschnitt geben wir verschiedene Literatur-Beispiele von anti-symmetrischen Abbildungen der Diedergruppe an und zeigen, daß diese Spezialfälle der bisher erarbeiteten Sätze darstellen. Mit diesen Sätzen gelingt uns jeweils ein deutlich kürzerer Beweis der Behauptungen.

Beispiel (VERHOEFF [27]) Definiere φ durch $\varphi(a^k) = a^{-k}$ und $\varphi(a^j b) = a^{j-d}b$, $d \neq 0$, dann ist $\varphi \in Ant(D_s)$, s ungerade.
Beweis Wir schreiben φ zunächst in der Matrixschreibweise: $\varphi(1,k) = (1,-k)$ und $\varphi(-1,j) = (-1, j-d)$. Es folgt, daß $\varphi(e,x) = (1,-1/2\cdot d)\cdot(e,x)^{-1}\cdot(1,1/2\cdot d)$ ist, denn

$$\begin{aligned}(1,-1/2\cdot d)\cdot(e,x)^{-1}\cdot(1,1/2\cdot d) &= (1,-1/2\cdot d)\cdot(e,-ex)\cdot(1,1/2\cdot d)\\ &= (e,-ex-1/2\cdot d+1/2\cdot ed)\\ &= \begin{cases}(1,-x) & \text{falls } e=1\\ (-1,x-d) & \text{falls } e=-1.\end{cases}\end{aligned}$$

Nach Satz 20 (Seite 52) ist damit φ eine anti-symmetrische Abbildung von D_5.

Beispiel (H.P. GUMM [12]) Für $a,b \in \mathbb{Z}_s$, s ungerade, $a \neq 0$ ist $\varphi(e,x) := (e, e(a-x)+b) \in Ant(D_s)$.
Beweis Es ist $\varphi(e,x) = (1,b)(e,x)^{-1}(1,a) = (1,b)(e,-ex)(1,a) = (e,-ex+b+ea) = (e,e(a-x)+b)$ und wir können wieder Satz 20 anwenden.

Beispiel (GALLIAN/MULLIN [10]) Die im Beweis von Theorem 3 (Seite 22) definierte Abbildung der Diedergruppe D_n, n ungerade, in Matrixschreibweise lautet $\varphi(1,x) = (1,2-x)$, $\varphi(-1,x) = (-1,x)$. Mit $a=b=1$ ist dies ein Spezialfall des vorherigen Beispiels.

Beispiel (STEVEN J. WINTERS [28]) Für jede ungerade Zahl $s > 2$ definiere die Permutation (Zyklenschreibweise)

$$\varphi = (0)(1,s-1)(2,s-2)\ldots(\frac{s-1}{2},\frac{s+1}{2})(s,s+1,\ldots,2s-1).$$

Dann ist φ eine anti-symmetrische Abbildung von D_s.

Beweis Die Matrixschreibweise dieser Permutation lautet

$$\varphi(e,x) = \begin{cases}(1,-x) & \text{falls } e=1\\ (-1,x+1) & \text{falls } e=-1.\end{cases}$$

Für $d = -1$ stimmt diese Permutation mit der von VERHOEFF gefundenen überein.

Beispiel Die elfstelligen Seriennummern der deutschen Banknoten werden mit der Permutation $\varphi = [1576283094] = [7046913258]^{-1}$ (vgl. Seite 100) gesichert [21]. Die Nummern enthalten an den Positionen 1,2 und 10 statt Ziffern Buchstaben. Diese werden vor der Prüfung gemäß folgender Tabelle umgesetzt:

A	D	G	K	L	N	S	U	Y	Z
0	1	2	3	4	5	6	7	8	9

Die Prüfgleichung lautet

$$\varphi(x_{10}) \cdot \varphi^2(x_9) \cdot \ldots \cdot \varphi^{10}(x_1) \cdot x_0 = 0.$$

Wäre die vorletzte Stelle der Seriennummer kein Buchstabe, sondern eine Ziffer, dann würde das Verfahren nicht alle Vertauschungen von x_0 mit x_1 erkennen. Aber durch den Buchstaben besteht keine Verwechslungsgefahr. Bei der benutzten Prüfgleichung ist nicht φ, sondern φ^{-1} eine anti-symmetrische Abbildung, da die Potenzen von φ aufsteigend gewählt wurden.

Für die Seriennummer DG2661778N1 eines 10-DM Scheines ergibt sich beispielsweise

Seriennummer	DG2661778N1
codierte Zahl	1226617785 1
$\varphi(x_{10}), \ldots, \varphi^{10}(x_1), x_0$	5016372799 1

Die Diedermultiplikation liefert $5 \cdot 0 \cdot 1 \cdot 6 \cdot 3 \cdot 7 \cdot 2 \cdot 7 \cdot 9 \cdot 9 \cdot 1 = 0$, d.h. die Seriennummer ist gültig.

Kapitel 4

Prüfziffersysteme über Quasigruppen

In diesem Kapitel verallgemeinern wir den Begriff des Prüfziffersystems auf Quasigruppen. Wir untersuchen verschiedene Ansätze und geben am Ende Prüfziffersysteme zu den Basen 6, 8 und 10 an, die eine bessere oder zumindest gleich gute Fehlererkennung bieten, wie Prüfziffersysteme basierend auf Gruppen. Außerdem zeigen wir, daß eine Reihe von Quasigruppen keine bessere Fehlererkennung bieten können als Prüfziffersysteme über Gruppen.

4.1 Allgemeine Ergebnisse

Wir stellen zuerst zwei Möglichkeiten vor, wie der Begriff „Prüfziffersystem" verallgemeinert werden kann.

Definition 9 *Sei* $D = \{0, \ldots, m-1\}$ *eine Menge von Ziffern,* $c \in D$ *und* $g : D^{n+1} \to D$ *eine Abbildung. Die Menge* $P_{g,c} := \{(d_n, \ldots, d_0) \in D^{n+1} | g(d_n, \ldots, d_0) = c\}$ *heißt* implizites Prüfziffersystem zur Basis m, *wenn gilt:*

1. $g(d_n, \ldots, d_i, \ldots, d_0) = g(d_n, \ldots, d_i', \ldots, d_0) = c$ *impliziert* $d_i = d_i'$
2. $g(d_n, \ldots, d_i, d_{i-1}, \ldots, d_0) = g(d_n, \ldots, d_{i-1}, d_i, \ldots, d_0) = c$ *impliziert* $d_i = d_{i-1}$
3. *für alle* $d_n, \ldots, d_1 \in D$ *existiert ein* $d_0 \in D$ *s.d.* $g(d_n, \ldots, d_1, d_0) = c$

oder, vgl. H.P. Gumm [12]

Definition 10 *Sei* $D = \{0, \ldots, m-1\}$ *eine Menge von Ziffern und* $f : D^n \to D$ *eine Abbildung. Die Menge* $P_f' := \{(d_n, \ldots, d_0) \in D^{n+1} | f(d_n, \ldots, d_1) = d_0\}$ *heißt* explizites Prüfziffersystem zur Basis m, *wenn gilt:*

1. $f(d_n, \ldots, d_i, \ldots, d_1) = f(d_n, \ldots, d_i', \ldots, d_1)$ *impliziert* $d_i = d_i'$

2. $f(d_n, \ldots, d_i, d_{i-1}, \ldots, d_1) = f(d_n, \ldots, d_{i-1}, d_i, \ldots, d_1)$ *impliziert* $d_i = d_{i-1}$

3. $f(d_n, \ldots, d_2, d_0) = d_1$, *wobei* $f(d_n, \ldots, d_1) = d_0$, *impliziert* $d_0 = d_1$

Bei den impliziten Prüfziffersystemen ist die Prüfziffer d_0 einer vorgegebenen Zahl $d_n d_{n-1} \ldots d_1$ die eindeutig bestimmte Lösung der Gleichung $g(d_n, \ldots, d_1, d_0) = c$. Dabei garantiert uns die dritte Eigenschaft, daß überhaupt eine Lösung existiert, mit der ersten Eigenschaft folgt deren Eindeutigkeit. Die zweite Eigenschaft sorgt für die Erkennung aller Nachbarvertauschungen.
Die expliziten Prüfziffersysteme haben den Vorteil, daß sich die Prüfziffer nicht als Lösung einer Gleichung ergibt, sondern daß diese direkt durch $f(d_n, \ldots, d_1)$ ausgerechnet werden kann. Die dritte Eigenschaft dient hier dazu, die Vertauschung der letzten Ziffer mit der Prüfziffer zu erkennen.
Beide Definitionen lassen es auch zu, eine Prüfziffer zu bestimmen, die in die ursprüngliche Zahl an einer Position i eingebaut wird. Dazu bestimmt man die eindeutige Lösung p der Gleichung

$$g(d_n, \ldots, d_{i+1}, p, d_i, \ldots, d_1) = c$$

bzw.

$$f(d_n, \ldots, d_{i+1}, p, d_i, \ldots, d_2) = d_1.$$

Die gesicherte Zahl lautet in beiden Fällen $d_n d_{n-1} \ldots d_{i+1} p d_i \ldots d_2 d_1$.

Die Definitionen sind im folgenden Sinne äquivalent: Zu jedem expliziten Prüfziffersystem P'_f erhält man ein implizites Prüfziffersystem $P_{g,c}$ mit der Eigenschaft $f(d_n, \ldots, d_1) = d_0 \Leftrightarrow g(d_n, \ldots, d_0) = c$ (und damit $P'_f = P_{g,c}$) durch die Definitionen $c := 0$ und

$$g(d_n, \ldots, d_0) := \begin{cases} 0 & \text{falls } f(d_n, \ldots, d_1) = d_0 \\ 1 & \text{sonst} \end{cases}$$

Und umgekehrt erhält man zu jedem impliziten Prüfziffersystem $P_{g,c}$ ein explizites Prüfziffersystem P'_f indem man $f(d_n, \ldots, d_1)$ durch die eindeutig bestimmte Lösung x der Gleichung $g(d_n, \ldots, d_1, x) = c$ definiert, also

$$f(d_n, \ldots, d_1) := x \Leftrightarrow g(d_n, \ldots, d_1, x) = c.$$

Schwieriger ist die Lösung des folgenden Problems: Wenn f eine bestimmte Darstellung (z.B. mit Quasigruppen) besitzt, gibt es dann auch ein g, das eine analoge Darstellung mit der Eigenschaft

$$f(d_n, \ldots, d_1) = d_0 \Leftrightarrow g(d_n, \ldots, d_0) = c \tag{4.1}$$

besitzt? Für das genannte Beispiel kann man die Fragestellung positiv beantworten, wie der folgende Satz zeigt:

Satz 30 *1. Zu jedem expliziten Prüfziffersystem P'_f, wobei f eine Darstellung mit $n-1$ Quasigruppen $*_i$ besitzt, $f(d_n,\ldots,d_1) = (\ldots((d_n *_n d_{n-1}) *_{n-1} d_{n-2}) *_{n-1} \ldots) *_2 d_1$, existiert eine Quasigruppe $*_1$ und ein $c \in D$, so daß 4.1 für $g(d_n,\ldots,d_0) := f(d_n,\ldots,d_1) *_1 d_0 = (\ldots((d_n *_n d_{n-1}) *_{n-1} d_{n-2}) *_{n-1} \ldots) *_1 d_0$ gilt.*

2. Zu jedem impliziten Prüfziffersystem $P_{g,c}$, wobei g eine Darstellung mit n Quasigruppen $*_i$ besitzt, $g(d_n,\ldots,d_0) = (\ldots((d_n *_n d_{n-1}) *_{n-1} d_{n-2}) *_{n-1} \ldots) *_1 d_0$, existiert eine Quasigruppe $*'_2$, so daß 4.1 für $f(d_n,\ldots,d_1) := ((\ldots((d_n *_n d_{n-1}) *_{n-1} d_{n-2}) *_{n-1} \ldots) *_3 d_2) *'_2 d_1$ gilt.

Beweis Mit $x *_1 y := x - y$ (Rechnung in der Gruppe $\mathbb{Z}_m$) und $c := 0$ folgt Behauptung 1. $*'_2$ wird durch die Bedingung $x *'_2 y = z \Leftrightarrow (x *_2 y) *_1 z = c$ definiert. Damit folgt Behauptung 2. (Das $*'_2$ eine Quasigruppe ist, folgt aus den Kürzungsregeln der Quasigruppen $*_2$ und $*_1$, siehe unten)

Sollen allerdings alle benutzten Quasigruppen gleich sein, so führen die unterschiedlichen Definitionen i.allg. auch zu unterschiedlichen Codewörtern:

Beispiel Seien $D := \{0,\ldots,6\}$, $f(x,y) := x - 2y$, $g(x,y,z) := (x-2y) - 2z$ und $c := 0$. Es gilt $f(5,2) = 1$, aber $g(5,2,1) = -1 \neq 0 = g(5,2,4)$, d.h. im ersten Fall ist 52-1, im zweiten Fall 52-4 die gesicherte Zahl.

4.2 n-Quasigruppen

Zunächst definieren wir einige Grundbegriffe.

Definition 11 *Eine Quasigruppe ist eine Algebra $(Q,*)$ mit der Eigenschaft, daß die Gleichungen $a * x = b$ und $y * a = b$ für jedes Paar a,b eine eindeutige Lösung x, bzw. y besitzen.*

Eine n-Quasigruppe ist eine Algebra (Q,f), $f : Q^n \to Q$, so daß für $i = 1,\ldots,n$ und alle $x_n,\ldots,x_{i+1},x_{i-1},\ldots,x_1,x_0 \in Q$ die Gleichung

$$f(x_n,\ldots,x_{i+1},x,x_{i-1},\ldots,x_1) = x_0$$

eine eindeutig bestimmte Lösung $x \in Q$ besitzt.

Bemerkung Die Quasigruppen sind ein Spezialfall der n-Quasigruppen. Sie werden daher im Zusammenhang mit n-Quasigruppen als binäre Quasigruppen bezeichnet.

Bekanntlich ist ein endlicher Gruppoid genau dann eine Quasigruppe, wenn in ihm die Kürzungsregeln $a * x = a * y \Rightarrow x = y$ und $x * a = y * a \Rightarrow x = y$ gelten.

Definition 12 *Zwei n-Quasigruppen f, g sind isotop, falls Permutationen α, $\beta_n, \ldots, \beta_1$ existieren mit*

$$\alpha(f(x_n, \ldots, x_1)) = g(\beta_n(x_n), \ldots, \beta_1(x_1)),$$

sie heißen isomorph, falls $\alpha = \beta_n = \ldots = \beta_1$ gilt.

Bemerkung Isotopie und Isomorphie definieren eine Äquivalenzrelation auf der Menge der n-Quasigruppen.

Definition 13 *Die* Parastrophie f_α *einer n-Quasigruppe f und der Permutation $\alpha \in S_{n+1}$ wird definiert durch*

$$f(x_n, \ldots, x_1) = x_0 \quad \Leftrightarrow \quad f_\alpha(x_{\alpha(n)}, \ldots, x_{\alpha(n)}) = x_{\alpha(0)}.$$

Sie heißt hauptsächlich *wenn $\alpha(0) = 0$ ist.*

Offensichtlich sind die Parastrophien einer n-Quasigruppe wieder eine n-Quasigruppe. Für eine Quasigruppe $(Q, *)$ definieren wir speziell:

$$\begin{aligned} x *_t y = z \quad &\Leftrightarrow \quad y * x = z \\ x/y = z \quad &\Leftrightarrow \quad y = x * z \\ x\backslash y = z \quad &\Leftrightarrow \quad x = z * y \\ x/_t y = z \quad &\Leftrightarrow \quad x = y * z \\ x\backslash_t y = z \quad &\Leftrightarrow \quad y = z * x \end{aligned}$$

Es gilt $x/(x * y) = y$, $x * (x/y) = y$ und $(x * y)\backslash y = x$, $(x\backslash y) * y = x$.

Definition 14 *Die Quasigruppen $(Q, *)$ und $(Q, \cdot)$ heißen orthogonal, wenn die Paare $(x * y, x \cdot y)$ für alle $x, y \in Q$ paarweise verschieden sind. Eine Quasigruppe $(Q, *)$ heißt selbstorthogonal, wenn sie orthogonal zu $(Q, *_t)$ ist.*

Lemma 15 *Zwei endliche Quasigruppen $(Q, *)$ und $(Q, \cdot)$ der Ordnung m sind genau dann orthogonal, wenn für alle $a, b \in Q$ die Gleichungen $x * y = a$, $x \cdot y = b$ eine eindeutig bestimmte Lösung $x, y \in Q$ besitzen.*

Beweis Seien $(Q, *)$ und $(Q, \cdot)$ orthogonal, und es gelten die Gleichungen $x' * y' = x * y = a$, $x' \cdot y' = x \cdot y = b$. Dies ist äquivalent zur Gleichheit der Paare $(x' * y', x' \cdot y')$ und $(x * y, x \cdot y)$. Die Paare sind aber nach Voraussetzung genau dann gleich, wenn $x = x'$ und $y = y'$ gilt. Also ist die Lösung der Gleichungen eindeutig.

Wir müssen noch zeigen, daß die Gleichungen überhaupt eine Lösung besitzen. Da die Paare $(x * y, x \cdot y)$ alle verschieden sind, haben wir m^2 verschiedene Paare. Folglich muß es für jedes Paar (a, b) Elemente $x, y \in Q$ mit $(x * y, x \cdot y) = (a, b)$ geben. Die Rückrichtung folgt analog. □

Für die Verknüpfungstafel einer Quasigruppe ist der Begriff *lateinisches Quadrat* üblich. Lateinische Quadrate heißen orthogonal, wenn ihre zugehörigen Quasigruppen orthogonal sind. Die orthogonalen lateinischen Quadrate spielen im Zusammenhang mit den endlichen affinen Ebenen eine wichtige Rolle.

Beispiel 1. Die folgenden beiden Quasigruppen sind orthogonal.

$*$	0	1	2
0	0	1	2
1	2	0	1
2	1	2	0

$\cdot$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\longrightarrow$

$(*,\cdot)$	0	1	2
0	(0,0)	(1,1)	(2,2)
1	(2,1)	(0,2)	(1,0)
2	(1,2)	(2,0)	(0,1)

2. Die folgende Quasigruppe ist selbstorthogonal.

$*$	0	1	2	3
0	0	1	2	3
1	2	3	0	1
2	3	2	1	0
3	1	0	3	2

$*_t$	0	1	2	3
0	0	2	3	1
1	1	3	2	0
2	2	0	1	3
3	3	1	0	2

Orthogonale lateinische Quadrate wurden zuerst von EULER Ende des 18. Jahrhunderts untersucht. Für das Kreuzprodukt zweier orthogonaler lateinischer Quadrate benutzte er den Begriff „Griechisch-Lateinisches-Quadrat", da er für das eine Quadrat griechische und für das andere lateinische Buchstaben verwendete. Griechisch-Lateinische-Quadrate werden aus diesem Grund auch Euler-Quadrate genannt.

Definition 15 *Eine Quasigruppe* $(Q,*)$ *ist* anti-symmetrisch, *wenn* $x*y = y*x \Rightarrow x = y$ *gilt. Analog heißt eine* n*-Quasigruppe* anti-symmetrisch, *wenn*

$$f(x_n,\ldots,x_i,x_{i-1},\ldots,x_1) = f(x_n,\ldots,x_{i-1},x_i,\ldots,x_1) \Rightarrow x_i = x_{i-1}.$$

Definition 16 *Eine Permutation* φ *einer Quasigruppe* $(Q,*)$ *heißt anti-symmetrische bzw. vollständige Abbildung, falls gilt:*

$$\varphi(x) * y = \varphi(y) * x \qquad \Rightarrow \qquad x = y$$

bzw.

$$\varphi^{-1}(x) * x = \varphi^{-1}(y) * y \qquad \Rightarrow \qquad x = y.$$

Die Menge aller anti-symmetrischen bzw. vollständigen Abbildungen einer Quasigruppe werde mit $Ant(Q,*)$ *bzw.* $Com(Q,*)$ *bezeichnet.*

Bemerkung Die zweite Eigenschaft ist äquivalent zu

$$x * \varphi(x) = y * \varphi(y) \qquad \Rightarrow \qquad x = y.$$

Die Definition der vollständigen Abbildungen stimmt also mit der für Gruppen getroffenen Definition überein.

Definition 17 *Sei f eine n-Quasigruppe dann wird die n-Quasigruppe $\hat{f}$ definiert durch*

$$\hat{f}(x_n, \ldots, x_1) = x_0 \Leftrightarrow f(x_0, x_1, \ldots, x_{n-1}) = x_n$$

Wir zeigen nun den Zusammenhang zwischen n-Quasigruppen und Prüfziffersystemen.

Satz 31 *1. Jede n-Quasigruppe erkennt alle Einzelfehler. Wenn g eine anti-symmetrische n-Quasigruppe ist, so definiert $P_{g,c}$ ein implizites Prüfziffersystem für alle $c \in D$.*

2. P'_f ist ein explizites Prüfziffersystem genau dann, wenn $P'_{\hat{f}}$ ein explizites Prüfziffersystem ist.

3. P'_f ist genau dann ein explizites Prüfziffersystem, wenn f und $\hat{f}$ anti-symmetrische n-Quasigruppen sind.

Beweis 1) Folgt direkt aus den Definitionen.
2) Da $(\hat{\hat{f}}) = f$ gilt, reicht es, eine Richtung zu zeigen. Sei P'_f ein Prüfziffersystem. $\hat{f}$ ist eine n-Quasigruppe und erkennt daher alle Einzelfehler. Es gelte nun

$$\hat{f}(d_n, \ldots, d_i, d_{i-1}, \ldots, d_1) = d_0 = \hat{f}(d_n, \ldots, d_{i-1}, d_i, \ldots, d_1)$$

oder

$$\hat{f}(d_n, \ldots, d_1) = d_0, \hat{f}(d_n, \ldots, d_2, d_0) = d_1.$$

Es folgt

$$f(d_0, \ldots, d_{i-1}, d_i, \ldots, d_{n-1}) = d_n = f(d_0, \ldots, d_i, d_{i-1}, \ldots, d_{n-1}), \text{ falls } i < n$$

und

$$f(d_0, \ldots, d_{n-1}) = d_n,\ f(d_0, \ldots, d_{n-2}, d_n) = d_{n-1}, \text{ falls } i = n.$$

Mit den Eigenschaften von f folgt nun, daß $d_i = d_{i-1}$ ist. Also ist $P'_{\hat{f}}$ ein Prüfziffersystem.
3) Sei P'_f ein Prüfziffersystem. Aus 2) folgt, daß auch $P'_{\hat{f}}$ ein Prüfziffersystem ist

und damit f und $\hat{f}$ anti-symmetrische n-Quasigruppen sind. Sind umgekehrt f und $\hat{f}$ anti-symmetrische n-Quasigruppen so sind die Bedingungen 1 und 2 der Definition 10 (Seite 61) für f erfüllt. Bedingung 3 folgt aus der Anti-Symmetrie von $\hat{f}$:

$$\begin{aligned} & f(d_n,\ldots,d_2,d_1)=d_0, f(d_n,\ldots,d_2,d_0)=d_1 \\ \Leftrightarrow\ & \hat{f}(d_0,d_1,\ldots,d_{n-1})=\hat{f}(d_1,d_0,d_2,\ldots,d_{n-1}) \\ \Leftrightarrow\ & d_0=d_1. \end{aligned}$$

□

Lemma 16 *Sei (Q,f) eine anti-symmetrische n-Quasigruppe und φ,ψ Permutationen von Q, dann ist auch $(Q,\bar{f})$ mit*

$$\bar{f}(x_n,\ldots,x_1) := \psi^{-1}(f(\varphi(x_n),\ldots,\varphi(x_1)))$$

anti-symmetrisch.

Beweis Aus $\bar{f}(x_n,\ldots,x_{i-1},x_i,\ldots,x_1)=\bar{f}(x_n,\ldots,x_i,x_{i-1},\ldots,x_1)$ folgt

$$f(\varphi(x_n),\ldots,\varphi(x_i),\varphi(x_{i-1}),\ldots,\varphi(x_1))=f(\varphi(x_n),\ldots,\varphi(x_{i-1}),\varphi(x_i),\ldots,\varphi(x_1)).$$

Da (Q,f) anti-symmetrisch ist, folgt $\varphi(x_i)=\varphi(x_{i-1})$ und damit $x_i=x_{i-1}$. □

Der folgende Satz stellt einen Zusammenhang zwischen anti-symmetrischen Abbildungen und anti-symmetrischen Quasigruppen her:

Satz 32 *Eine Quasigruppe, und insbesondere eine Gruppe, besitzt eine anti-symmetrische Abbildung genau dann, wenn sie isotop zu einer anti-symmetrischen Quasigruppe ist.*

Beweis Die Quasigruppe $(Q,\cdot)$ besitze die anti-symmetrische Abbildung φ, dann ist $(Q,*)$ mit $x*y:=\varphi(x)\cdot y$ eine anti-symmetrische Quasigruppe. Sei umgekehrt die anti-symmetrische Quasigruppe $(Q,*)$ isotop zur Quasigruppe $(Q,\cdot)$, also $\gamma(x*y)=\alpha(x)\cdot\beta(y)$ mit den Permutationen α,β,γ. Die Quasigruppe $x\bullet y:=\gamma(\beta^{-1}(x)*\beta^{-1}(y))$ ist wieder anti-symmetrisch (Lemma 16 für $n=2$). Es folgt, daß $\alpha\circ\beta^{-1}$ eine anti-symmetrische Abbildung von $(Q,\cdot)$ ist, denn

$$\begin{aligned} x\bullet y &= \gamma(\beta^{-1}(x)*\beta^{-1}(y)) \\ &= \gamma(\gamma^{-1}(\alpha(\beta^{-1}(x))\cdot\beta(\beta^{-1}(y)))) \\ &= \alpha\circ\beta^{-1}(x)\cdot y \\ &= \alpha\circ\beta^{-1}(y)\cdot x \\ &= y\bullet x \end{aligned}$$

ist nur erfüllt wenn $x = y$ ist. □

Satz 33 *In einer Isotopieklasse besitzt entweder jede oder keine Quasigruppe eine anti-symmetrische bzw. vollständige Abbildung.*

Beweis Aus dem vorherigen Satz folgt, daß jede Quasigruppe, die isotop ist zu einer Quasigruppe mit anti-symmetrischer Abbildung, ebenfalls eine anti-symmetrische Abbildung besitzt.

Sei $(Q, *)$ eine Quasigruppe mit vollständiger Abbildung φ^{-1} und $(Q, \cdot)$ mit $x \cdot y = \gamma^{-1}(\alpha(x) * \beta(y))$ sei isotop zu $(Q, *)$, dann ist $\tilde{\varphi}^{-1} := \alpha^{-1} \circ \varphi^{-1} \circ \beta$ eine vollständige Abbildung von $(Q, \cdot)$: Aus

$$\tilde{\varphi}(x) \cdot x = \gamma^{-1}(\alpha(\tilde{\varphi}(x)) * \beta(x)) = \gamma^{-1}(\alpha(\tilde{\varphi}(y)) * \beta(y)) = \tilde{\varphi}(y) \cdot y$$

folgt

$$\varphi^{-1}(\beta(x)) * \beta(x) = \varphi^{-1}(\beta(y)) * \beta(y).$$

φ^{-1} ist eine vollständige Abbildung von $(Q, *)$, also folgt $\beta(x) = \beta(y)$ und damit $x = y$. □

Definition 18 *Eine Quasigruppe besitzt die* Transversale (φ_1, φ_2)*, wenn* φ_1, φ_2 *und* $\varphi_1(x) * \varphi_2(x)$ *Permutationen sind.*

Lemma 17 *Eine Quasigruppe besitzt eine vollständige Abbildung genau dann, wenn sie eine Transversale besitzt.*

Beweis Ist (φ_1, φ_2) eine Transversale, so ist $\varphi_2 \circ \varphi_1^{-1}$ eine vollständige Abbildung. Andererseits erhalten wir durch die vollständige Abbildung φ die Transversale (Id, φ). □

Wenn die Quasigruppe $(Q, *)$ die anti-symmetrische bzw. vollständige Abbildung φ besitzt, dann ist φ^{-1} eine anti-symmetrische bzw. vollständige Abbildung der Parastrophie $(Q, *_t)$. Für vollständige Abbildungen gilt außerdem:

Lemma 18 *(vgl.* Belousov [4]*) Besitzt die Quasigruppe* $(Q, *)$ *eine vollständige Abbildung, dann gilt dies auch für jede Parastrophie von* $(Q, *)$*.*

Beweis Wir zeigen die Behauptung mit Lemma 17. $(Q, *)$ besitze die Transversale (φ_1, φ_2). Wir definieren die Permutation φ_3 durch $\varphi_3(x) := \varphi_1(x) * \varphi_2(x)$. Damit ist (φ_1, φ_3) eine Transversale der Parastrophie $(Q, /)$, denn $\varphi_1(x)/\varphi_3(x) = \varphi_1(x)/(\varphi_1(x) * \varphi_2(x)) = \varphi_2(x)$ ist eine Permutation. Die Behauptung für die anderen Parastrophien folgt analog. □

4.3 Reduzible n-Quasigruppen

In diesem Abschnitt geben wir ein Kriterium an, mit dem wir entscheiden können, ob eine n-Quasigruppe eine Darstellung mit binären Quasigruppen hat.

Definition 19 *(vgl. [6]) Eine n-Quasigruppe (Q, f), $n > 2$ heißt reduzibel wenn eine Permutation $\alpha \in S_n$ und Quasigruppen $g : Q^{n-k+1} \to Q$, $h : Q^k \to Q$ $(1 \leq k < n-1)$ existieren mit*

$$x_0 = f(x_n, \ldots, x_1) = g(x_{\alpha_n}, \ldots, x_{\alpha_{k+1}}, h(x_{\alpha_k}, \ldots, x_{\alpha_1})).$$

Sie heißt total reduzibel wenn $n-1$ binäre Quasigruppen (Q, g_i), $i = 1, \ldots, n-1$ existieren, so daß f als Komposition der g_i dargestellt werden kann. Eine n-Quasigruppe heißt irreduzibel wenn sie nicht reduzibel ist.

Theorem 11 *(*Verhoeff [27]*) Es existieren irreduzible n-Quasigruppen.*

Man könnte vermuten, daß die Anti-Symmetrie-Eigenschaft verhindert, daß eine n-Quasigruppe irreduzibel ist. Dies ist allerdings nicht der Fall, wie folgendes Theorem zeigt:

Theorem 12 *Es existieren irreduzible anti-symmetrische n-Quasigruppen.*

Beweis Die folgende 3-Quasigruppe f ist irreduzibel und anti-symmetrisch:

$k =$	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
j=0	0 1 2 3 4 5	1 2 0 5 3 4	2 0 1 4 5 3	3 4 5 1 2 0	4 5 3 0 1 2	5 3 4 2 0 1
1	2 0 1 4 5 3	0 1 2 3 4 5	1 2 0 5 3 4	5 3 4 2 0 1	3 4 5 1 2 0	4 5 3 0 1 2
2	1 2 0 5 3 4	2 0 1 4 5 3	0 1 2 3 4 5	4 5 3 0 1 2	5 3 4 2 0 1	3 4 5 1 2 0
3	4 5 3 0 1 2	3 4 5 1 2 0	5 3 4 2 0 1	0 1 2 4 5 3	2 0 1 5 3 4	1 2 0 3 4 5
4	5 3 4 2 0 1	4 5 3 0 1 2	3 4 5 1 2 0	1 2 0 3 4 5	0 1 2 4 5 3	2 0 1 5 3 4
5	3 4 5 1 2 0	5 3 4 2 0 1	4 5 3 0 1 2	2 0 1 5 3 4	1 2 0 3 4 5	0 1 2 4 5 3
	i=0	1	2	3	4	5

Wenn $f(i,j,k)$ zerlegbar wäre in $g(i,j)$ und $h(i,j)$, dann würde gelten

1. $f(i,j,k) = g(i, h(j,k))$ oder
2. $f(i,j,k) = g(j, h(i,k))$ oder
3. $f(i,j,k) = g(k, h(i,j))$.

Im ersten Fall folgt aus $f(i,j,k) = f(i,j',k')$, daß $h(j,k) = h(j',k')$ und folglich, daß $f(i',j,k) = f(i',j',k')$. Es gilt aber $f(0,0,0) = f(0,3,3) = 0$ und $f(3,0,0) = 3 \neq 4 = f(3,3,3)$. Analog folgt im zweiten bzw. dritten Fall, daß $f(i,j,k) =$

$f(i',j,k') \Rightarrow f(i,j',k) = f(i',j',k')$ bzw. $f(i,j,k) = f(i',j',k) \Rightarrow f(i,j,k') = f(i',j',k')$. Es gilt aber $f(0,0,0) = f(1,0,2) = 0$, $f(0,3,0) = 4 \neq 5 = f(1,3,2)$ und $f(0,0,0) = f(3,3,0) = 0$, $f(0,0,3) = 3 \neq 4 = f(3,3,3)$.

Daß diese 3-Quasigruppe die Vertauschungen $j \leftrightarrow k$ erkennt, läßt sich leicht an den anti-symmetrischen Quasigruppen $f(0,j,k), \ldots, f(5,j,k)$ ablesen. Wenn wir eine andere Projektionsebene wählen, dann sieht man ebenso, daß auch die Quasigruppen $f(i,j,0), \ldots, f(i,j,5)$ anti-symmetrisch sind.

$i=$	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
j=0	0 1 2 3 4 5	1 2 0 4 5 3	2 0 1 5 3 4	3 5 4 1 0 2	4 3 5 2 1 0	5 4 3 0 2 1
1	2 0 1 5 3 4	0 1 2 3 4 5	1 2 0 4 5 3	4 3 5 2 1 0	5 4 3 0 2 1	3 5 4 1 0 2
2	1 2 0 4 5 3	2 0 1 5 3 4	0 1 2 3 4 5	5 4 3 0 2 1	3 5 4 1 0 2	4 3 5 2 1 0
3	4 3 5 0 2 1	5 4 3 1 0 2	3 5 4 2 1 0	0 1 2 4 5 3	1 2 0 5 3 4	2 0 1 3 4 5
4	5 4 3 1 0 2	3 5 4 2 1 0	4 3 5 0 2 1	2 0 1 3 4 5	0 1 2 4 5 3	1 2 0 5 3 4
5	3 5 4 2 1 0	4 3 5 0 2 1	5 4 3 1 0 2	1 2 0 5 3 4	2 0 1 3 4 5	0 1 2 4 5 3
	k=0	1	2	3	4	5

Damit ist f eine irreduzible anti-symmetrische 3-Quasigruppe. □

Verhoeff [27] gab ein Beispiel für eine irreduzible 3-Quasigruppen bereits für $m = 4$ an. Diese definiert aber kein Prüfziffersystem, da sie nicht anti-symmetrisch ist.

Eine interessante Anwendungsmöglichkeit der irreduziblen n-Quasigruppen ergibt sich aus der Tatsache, daß man mit einer kleinen Anzahl gesicherter Zahlen nicht auf das verwendete Prüfziffersystem schließen kann. Damit kann man verhindern, daß absichtlich falsche Zahlen (z.B. Kreditkartennummern) mit gültiger Prüfziffer eingegeben werden.

Im folgenden beschäftigen wir uns mit reduziblen n-Quasigruppen.

Satz 34 *Wenn das explizite Prüfziffersystem P'_f auf der (total) reduziblen n-Quasigruppe f beruht, dann existiert ein äquivalentes implizites Prüfziffersystem $P_{g,c}$, bei dem g eine (total) reduzible $n+1$-Quasigruppe ist. Die Umkehrung gilt im allgemeinen nicht.*

Beweis Wir definieren

$$g(d_n, \ldots, d_1, d_0) := f(d_n, \ldots, d_1) - d_0$$

und $c = 0$. Damit folgt der erste Teil der Behauptung. Sei nun P'_f ein explizites Prüfziffersystem und f eine irreduzible n-Quasigruppe. Dann ist g eine, offensichtlich reduzible, $n+1$-Quasigruppe. Wenn eine n-Quasigruppe $\tilde{f}$ mit

$$\tilde{f}(d_n, \ldots, d_1) = d_0 \Leftrightarrow g(d_n, \ldots, d_0) = 0$$

existiert, dann folgt, daß $\tilde{f}(d_n, \ldots, d_1) = d_0 = f(d_n, \ldots, d_1)$ und damit $f = \tilde{f}$ gilt. Also ist $\tilde{f}$ irreduzibel und es existiert kein reduzibles explizites Prüfziffersystem, das äquivalent zu $P_{g,c}$ ist. □

Theorem 13 *Sei f eine n-Quasigruppe über der Menge Q und $c_{n-2}, \ldots, c_1 \in Q$ beliebige, aber fest gewählte Konstanten. Es gibt Quasigruppen $*_i$, $i = 2, \ldots, n$ mit*

$$f(x_n, \ldots, x_1) = (\ldots((x_n *_n x_{n-1}) *_{n-1} x_{n-2}) *_{n-1} \ldots) *_2 x_1,$$

genau dann, wenn für $i = 1, \ldots, n-2$ gilt

$$\begin{aligned} f(x_n, \ldots, x_{i+1}, c_i, c_{i-1}, \ldots, c_1) &= f(x'_n, \ldots, x'_{i+1}, c_i, c_{i-1}, \ldots, c_1) \\ \Rightarrow f(x_n, \ldots, x_{i+1}, x, c_{i-1}, \ldots, c_1) &= f(x'_n, \ldots, x'_{i+1}, x, c_{i-1}, \ldots, c_1) \end{aligned}$$

Beweis Es gelte $f(x_n, \ldots, x_1) = (\ldots((x_n *_n x_{n-1}) *_{n-1} x_{n-2}) *_{n-1} \ldots) *_2 x_1$ für die Quasigruppen $*_i$. Aus

$$f(x_n, \ldots, x_{i+1}, c_i, c_{i-1}, \ldots, c_1) = f(x'_n, \ldots, x'_{i+1}, c_i, c_{i-1}, \ldots, c_1)$$

folgt mit Hilfe der Kürzungsregel für die Quasigruppen $*_j$, $j = 2, \ldots, i+1$

$$(\ldots(x_n *_n x_{n-1}) *_{n-1} \ldots) *_{i+2} x_{i+1} = (\ldots(x'_n *_n x'_{n-1}) *_{n-1} \ldots) *_{i+2} x'_{i+1}$$

und damit

$$\begin{aligned} f(x_n, .., x_{i+1}, x, c_{i-1}, .., c_1) &= (..((..(x_n *_n x_{n-1}) *_{n-1} ..) *_{i+1} x) *_i c_{i-1}..) *_2 c_1 \\ &= (..((..(x'_n *_n x'_{n-1}) *_{n-1} ..) *_{i+1} x) *_i c_{i-1}..) *_2 c_1 \\ &= f(x'_n, .., x'_{i+1}, x, c_{i-1}, .., c_1). \end{aligned}$$

Für die Rückrichtung definieren wir die Quasigruppen $*_i$, $i = 2, \ldots, n-1$ folgendermaßen:

$$x *_i y := f(x_n, \ldots, x_i, y, c_{i-2}, \ldots, c_1), \text{ falls } f(x_n, \ldots, x_i, c_{i-1}, c_{i-2}, \ldots, c_1) = x,$$

und

$$x *_n y := f(x, y, c_{n-2}, \ldots, c_1).$$

Die $*_i$ sind wohldefiniert, weil die Gleichung $f(0, \ldots, 0, y, c_{i-1}, c_{i-2}, \ldots, c_1) = x$ eine Lösung y besitzt und weil aus

$$f(x_n, \ldots, x_i, c_{i-1}, c_{i-2}, \ldots, c_1) = x = f(x'_n, \ldots, x'_i, c_{i-1}, c_{i-2}, \ldots, c_1)$$

folgt, daß $f(x_n,\ldots,x_i,y,c_{i-2},\ldots,c_1) = f(x'_n,\ldots,x'_i,y,c_{i-2},\ldots,c_1)$ ist. Außerdem gilt:

$$\begin{aligned} f(x_n,\ldots,x_1) &= f(x_n,\ldots,x_2,c_1) *_2 x_1 \\ &= (f(x_n,\ldots,x_3,c_2,c_1) *_3 x_2) *_2 x_1 \\ &\vdots \\ &= (\ldots(f(x_n,x_{n-1},c_{n-2},\ldots,c_1) *_{n-1} x_{n-2}) *_{n-1} \ldots) *_2 x_1 \\ &= (\ldots((x_n *_n x_{n-1}) *_{n-1} x_{n-2}) *_{n-1} \ldots) *_2 x_1 \end{aligned}$$

□

Ist f auf diese Weise zerlegbar, dann gilt sogar für beliebige x_j, x'_j

$$\begin{aligned} f(x_n,\ldots,x_{i+1},x_i,x_{i-1},\ldots,x_1) &= f(x'_n,\ldots,x'_{i+1},x_i,x_{i-1},\ldots,x_1) \\ \Rightarrow f(x_n,\ldots,x_{i+1},x'_i,x_{i-1},\ldots,x_1) &= f(x'_n,\ldots,x'_{i+1},x'_i,x_{i-1},\ldots,x_1) \end{aligned}$$

und die Voraussetzungen des Theorems sind für verschiedene Konstanten c_j erfüllt. Für verschiedene $c_1 \neq c'_1$ sind aber die Quasigruppen $x *_n y := f(x,y,c_{n-2},\ldots,c_1)$ und $x *'_n y := f(x,y,c_{n-2},\ldots,c'_1)$ verschieden, denn aus $x *_n y = x *'_n y$ folgt, da f eine n-Quasigruppe ist, $c_1 = c'_1$. Wir sehen also, daß für f unterschiedliche Darstellungen existieren.

Theorem 14 *(*Belousov [6]*) Eine n-Quasigruppe f ist reduzibel genau dann, wenn folgende Abschlußbedingung für eine hauptsächliche Parastrophie f_α erfüllt ist ($1 < k < n$):*

$$\begin{aligned} f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},x_{\alpha_k},\ldots,x_{\alpha_1}) &= f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},y_{\alpha_k},\ldots,y_{\alpha_1}) \\ \Rightarrow f_\alpha(x'_{\alpha_n},\ldots,x'_{\alpha_{k+1}},x_{\alpha_k},\ldots,x_{\alpha_1}) &= f_\alpha(x'_{\alpha_n},\ldots,x'_{\alpha_{k+1}},y_{\alpha_k},\ldots,y_{\alpha_1}) \end{aligned}$$

Den folgenden Beweis dieser Aussage haben wir unabhängig von Belousov gefunden.

Beweis Sei f reduzibel, also

$$f(x_n,\ldots,x_1) = g(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},h(x_{\alpha_k},\ldots,x_{\alpha_1})).$$

Aus

$$\begin{aligned} f(x_n,\ldots,x_1) &= f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},x_{\alpha_k},\ldots,x_{\alpha_1}) \\ &= g(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},h(x_{\alpha_k},\ldots,x_{\alpha_1})) \\ &= g(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},h(y_{\alpha_k},\ldots,y_{\alpha_1})) \\ &= f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},y_{\alpha_k},\ldots,y_{\alpha_1}) \end{aligned}$$

folgt $h(x_{\alpha_k},\ldots,x_{\alpha_1}) = h(y_{\alpha_k},\ldots,y_{\alpha_1})$ und damit

$$\begin{aligned} f_\alpha(x'_{\alpha_n},\ldots,x'_{\alpha_{k+1}},x_{\alpha_k},\ldots,x_{\alpha_1}) &= g(x'_{\alpha_n},\ldots,x'_{\alpha_{k+1}},h(x_{\alpha_k},\ldots,x_{\alpha_1})) \\ &= g(x'_{\alpha_n},\ldots,x'_{\alpha_{k+1}},h(y_{\alpha_k},\ldots,y_{\alpha_1})) \\ &= f_\alpha(x'_{\alpha_n},\ldots,x'_{\alpha_{k+1}},y_{\alpha_k},\ldots,y_{\alpha_1}) \end{aligned}$$

Es gelte nun die Abschlußbedingung für die n-Quasigruppe f und die Permutation α, $1 \le k < n$. Wir definieren die beiden Abbildungen g und h durch:

$$\begin{aligned} h(x_{\alpha_k},\ldots,x_{\alpha_1}) &:= f_\alpha(0,\ldots,0,x_{\alpha_k},\ldots,x_{\alpha_1}) \\ \text{und}\quad g(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},y) &:= f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},x_{\alpha_k},\ldots,x_{\alpha_1}), \\ &\quad \text{falls } h(x_{\alpha_k},\ldots,x_{\alpha_1}) = y. \end{aligned}$$

Die $(n-k+1)$-Quasigruppe g ist wohldefiniert, weil die Gleichung $h(0,\ldots,0,x) = y$ für jedes y eine eindeutig bestimmte Lösung x besitzt (h ist eine k-Quasigruppe), und außerdem folgt, falls $h(x_{\alpha_k},\ldots,x_{\alpha_1}) = h(x'_{\alpha_k},\ldots,x'_{\alpha_1}) = y$, d.h.

$$f_\alpha(0,\ldots,0,x_{\alpha_k},\ldots,x_{\alpha_1}) = f_\alpha(0,\ldots,0,x'_{\alpha_k},\ldots,x'_{\alpha_1}),$$

daß

$$\begin{aligned} f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},x_{\alpha_k},\ldots,x_{\alpha_1}) &= g(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},y) \\ &= f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},x'_{\alpha_k},\ldots,x'_{\alpha_1}) \end{aligned}$$

gilt. Aus der Definition von g und h folgt nun

$$\begin{aligned} f(x_n,\ldots,x_1) &= f_\alpha(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},x_{\alpha_k},\ldots,x_{\alpha_1}) \\ &= g(x_{\alpha_n},\ldots,x_{\alpha_{k+1}},h(x_{\alpha_k},\ldots,x_{\alpha_1})). \end{aligned}$$

Also ist f reduzibel. □

Bemerkung Anstatt der $0 \in Q$ können wir, wie im vorhergehenden Theorem, verschiedene Konstanten $c_n,\ldots,c_{k+1} \in Q$ benutzen, um die Abbildungen g und h zu definieren.

4.4 Existenz von Prüfziffersystemen

Die Existenz von Prüfziffersystemen für beliebige Basen größer 2 wurde von H.P. Gumm [12] 1985 bewiesen.

Theorem 15 *(H.P. Gumm) Für jede Basis $m > 2$ und alle $n \ge 2$ existiert eine Abbildung $f : D^n \to D$ bzw. $g : D^{n+1} \to D$, so daß P'_f bzw. $P_{g,0}$ ein Prüfziffersystem definiert.*

Beweis Wir können den Beweis durch die in Kapitel 2 aufgebaute Theorie etwas verkürzen. Ist m ungerade, dann besitzt $\mathbb{Z}_m$ die anti-symmetrische Abbildung $\tau(x) := -x$. Ist $m = 2k$ gerade, dann wissen wir, daß die Diedergruppe D_k (neutrales Element '0') mit m Elementen eine anti-symmetrische Abbildung τ besitzt. In beiden Fällen definiert daher die Gleichung

$$f(x_n, x_{n-1}, \ldots, x_2, x_1) := [\tau^n(x_n)\tau^{n-1}(x_{n-1}) \ldots \tau^2(x_2)\tau(x_1)]^{-1} = x_0$$

bzw.

$$g(x_n, x_{n-1}, \ldots, x_1, x_0) := \tau^n(x_n)\tau^{n-1}(x_{n-1}) \ldots \tau^2(x_2)\tau(x_1)x_0 = 0$$

ein Prüfziffersystem zur Basis m. □

Korollar 15 *Für alle $n \geq 2$ und für alle $m > 2$ existiert eine anti-symmetrische n-Quasigruppe zur Basis m.*

Zur Basis 2 existiert kein Prüfziffersystem (und damit auch keine anti-symmetrische n-Quasigruppe), denn den Zahlen 00, 01 und 10 müßten verschiedene Prüfziffern aus der Menge $\{0, 1\}$ zugeordnet werden, was unmöglich ist.

Eine weitere Möglichkeit ein Prüfziffersystem zur Basis p^m, wobei p eine Primzahl ist, zu definieren, bietet der Galois-Körper mit p^m Elementen. Mit der gewichteten Summe $\sum_{i=0}^{n} a_i x_i = 0$ erhalten wir ein Prüfziffersystem, falls $a_i \neq 0$ und benachbarte Gewichte verschieden sind. Auch hier sehen wir, daß der Fall $p = 2$ ausgeschlossen ist, da wir nur $a_i = 1$ wählen können und damit benachbarte Gewichte gleich sind.

Des weiteren ist erwähnenswert, daß wir aus zwei Prüfziffersystemen zu den Basen m_1 und m_2 auf natürliche Weise ein Prüfziffersystem zur Basis $m_1 \cdot m_2$ erhalten, indem wir jede Zahl der Basis $m_1 \cdot m_2$ als eindeutiges Paar (d_1, d_2) darstellen, wobei d_1 eine Ziffer der Basis m_1 und d_2 eine Ziffer der Basis m_2 ist. Danach berechnen wir die Prüfziffern p_1 und p_2 getrennt für jede Komponente und wandeln das Paar (p_1, p_2) zurück in die zugehörige Zahl der Basis $m_1 \cdot m_2$. Es ist leicht einzusehen, daß die Eigenschaften 1–3 der Definitionen 9 und 10 (Seite 61) sowohl bei den impliziten als auch bei expliziten Prüfziffersystemen erhalten bleiben.

4.5 Prüfziffersysteme über Quasigruppen

In diesem Abschnitt untersuchen wir die total reduziblen Prüfziffersysteme der Form

$$g(x_n, x_{n-1}, \ldots, x_0) = (\ldots(x_n *_n x_{n-1}) *_{n-1} \ldots) *_1 x_0 = d$$

mit den (endlichen) Quasigruppen $(Q, *_i)$, $i = 1, \ldots, n$, und $d \in Q$. Wir benutzen die implizite Form, weil dadurch die Bedingungen für die Fehlererkennung einfacher formuliert werden können. Für die einzelnen Fehlerarten stellen wir die Anforderungen an die benutzten Quasigruppen zusammen. Zunächst zeigen wir, daß durch eine solche Prüfgleichung alle Einzelfehler erkannt werden können. Es gelte

$$(\ldots((\ldots(x_n *_n x_{n-1}) *_{n-1} \ldots) *_{i+1} x_i)\ldots) *_1 x_0 = d$$

und

$$(\ldots((\ldots(x_n *_n x_{n-1}) *_{n-1} \ldots) *_{i+1} x_i')\ldots) *_1 x_0 = d.$$

Wir setzen $c := (\ldots(x_n *_n x_{n-1}) *_{n-1} \ldots) *_{i+2} x_{i+1}$ und kürzen die Elemente $x_{i-1}, \ldots, x_0$ auf der rechten Seite. Es folgt

$$c *_i x_i = c *_i x_i'$$

und durch Kürzen von c erhalten wir $x_i = x_i'$.

Da $d \in Q$ beliebig gewählt werden kann, haben wir damit auch die Injektivität, und weil Q endlich ist, auch die Surjektiviät der Translationen $x \mapsto g(x_n, \ldots, x_{i+1}, x, x_{i-1}, \ldots, x_0)$ für alle i gezeigt. Folglich definiert g eine $(n+1)$-Quasigruppe über der Menge Q.

Die Transposition benachbarter Elemente wird genau dann erkannt, wenn die folgenden Implikationen für alle i und alle $c, x, y \in Q$ gelten:

$$\begin{aligned} x *_n y = y *_n x \quad &\Rightarrow \quad x = y \\ (c *_{i+1} x) *_i y = (c *_{i+1} y) *_i x \quad &\Rightarrow \quad x = y. \end{aligned} \tag{4.2}$$

Diese Aussage wird genauso gezeigt, wie die Aussage zur Erkennung der Einzelfehler. Zunächst werden die gleichen Elemente auf der rechten Seite gekürzt, dann werden die gleichen Elemente auf der linken Seite zu c zusammengefaßt.

Wir haben damit den folgenden Satz bewiesen:

Satz 35 *Mit den Quasigruppen $(Q, *_i)$ wird durch*

$$g(x_n, x_{n-1}, \ldots, x_0) := (\ldots(x_n *_n x_{n-1}) *_{n-1} \ldots) *_1 x_0$$

genau dann eine anti-symmetrische $(n+1)$-Quasigruppe definiert, wenn $*_n$ anti-symmetrisch ist und jede Zeile der Quasigruppe $*_{i+1}$ eine anti-symmetrisch Abbildung der Quasigruppe $*_i$ ist.

Die anderen Fehlerarten benötigen weitere Voraussetzungen, die für alle i und für alle $x, y, z, c \in Q$ erfüllt sein müssen.

Sprungtranspositionen:

$$\begin{aligned}(x *_n z) *_{n-1} y &= (y *_n z) *_{n-1} x \quad \Rightarrow \quad x = y \\ ((c *_{i+1} x) *_i z) *_{i-1} y &= ((c *_{i+1} y) *_i z) *_{i-1} x \quad \Rightarrow \quad x = y.\end{aligned}$$

Zwillingsfehler:

$$\begin{aligned}x *_n x &= y *_n y \quad \Rightarrow \quad x = y \\ (c *_{i+1} x) *_i x &= (c *_{i+1} y) *_i y \quad \Rightarrow \quad x = y.\end{aligned}$$

Sprungzwillingsfehler:

$$\begin{aligned}(x *_n z) *_{n-1} x &= (y *_n z) *_{n-1} y \quad \Rightarrow \quad x = y \\ ((c *_{i+1} x) *_i z) *_{i-1} x &= ((c *_{i+1} y) *_i z) *_{i-1} y \quad \Rightarrow \quad x = y.\end{aligned}$$

Im Vergleich zu den Prüfziffersystemen über Gruppen müssen wir also eine Vielzahl verschiedener Bedingungen überprüfen. Eine Verbesserung dieser Situation wäre erreichbar, wenn wir jeweils bei der zweiten Voraussetzung umklammern könnten. Dann wäre es möglich, das Element c ebenfalls zu kürzen. Diesen Gedanken werden wir im Abschnitt „Verallgemeinerte Assoziativität" ausführen. Zunächst zeigen wir jedoch einige wichtige Eigenschaften einer Quasigruppe in einem Prüfziffersystem.

Satz 36 *Jede Quasigruppe in einem Prüfziffersystem besitzt eine anti-symmetrische Abbildung. Erkennt das Prüfziffersystem alle Zwillingsfehler, dann besitzt jede Quasigruppe eine vollständige Abbildung, erkennt es alle Sprungzwillingsfehler, dann besitzt jede Quasigruppe außer ggf. $*_n$ eine vollständige Abbildung.*

Beweis Sei $*_i$ eine Quasigruppe eines Prüfziffersystems über Quasigruppen, d.h. sie erfüllt die Bedingung 4.2. Ist $i = n$, dann ist $*_n$ anti-symmetrisch und die Identität ist eine anti-symmetrische Abbildung. Für $*_i$, $i < n$, definieren wir $\varphi(x) := c *_{i+1} x$ für eine beliebige Konstante c. Damit ist $\varphi(x) *_i y = \varphi(y) *_i x$ äquivalent zur zweiten Bedingung von 4.2 und φ ist eine anti-symmetrische Abbildung von $*_i$.

Erkennt das Prüfziffersystem alle Zwillingsfehler, dann ist die Identität eine vollständige Abbildung von $*_n$ und φ mit $\varphi^{-1}(x) := c *_{i+1} x$ eine vollständige Abbildung von $*_i$, $i < n$. Falls das Prüfziffersystem alle Sprungzwillingsfehler erkennt, dann ist für fest gewählte $c, z \in Q$ die Permutation φ mit $\varphi^{-1}(x) := x *_n *z$ Element von $Com(Q, *_{n-1})$ und die Permutation φ mit $\varphi^{-1}(x) := (c *_{i+1} x) *_i z$ ist Element von $Com(Q, *_{i-1})$, $i < n$. □

Zusammen mit Satz 33 (Seite 68) sehen wir, daß viele Quasigruppen ungeeignet sind, ein Prüfziffersystem zu definieren, das alle (Sprung-)Zwillingsfehler erkennt.

Insbesondere eignen sich die Isotopien einer Gruppe ohne anti-symmetrische oder vollständige Abbildung nicht. Speziell für den Fall $m = 10$ bedeutet dies, daß wir kein Prüfziffersystem finden werden, das alle (Sprung-)Zwillingsfehler erkennt und in dem Quasigruppen vorkommen, die zu einer Gruppe isotop sind.

Da wir bereits gezeigt haben, daß $\mathbb{Z}_{2k}$ keine anti-symmetrische und jede Gruppe der Ordnung $2k$ für ungerades k keine vollständige Abbildung besitzt, erhalten wir das Korollar:

Korollar 16 *In einem Prüfziffersystem über Quasigruppen der Ordnung $2k$ ist keine Quasigruppe zur Gruppe $\mathbb{Z}_{2k}$ isotop. Erkennt das Prüfziffersystem alle Zwillings- oder alle Sprungzwillingsfehler und ist k ungerade, dann ist keine Quasigruppe isotop zu einer Gruppe der Ordnung $2k$.*

Der im Abschnitt „Quasigruppen isotop zu einer Gruppe" beschriebene Ansatz ist daher hauptsächlich für andere Ordnungen von Interesse.

Wir zeigen nun einen interessanten Zusammenhang zwischen Prüfziffersystemen über Quasigruppen und orthogonalen lateinischen Quadraten.

Satz 37 *Seien $(Q, *_i)$ die Quasigruppen eines Prüfziffersystems, das alle Zwillingsfehler erkennt, dann ist die Quasigruppe $(Q, *_i)$, $i = 1, \ldots, n-1$, orthogonal zu der durch*

$$x *'_i y = z \quad :\Leftrightarrow \quad z *_{i+1} y = x$$

definierten.

Beweis Wir müssen zeigen, daß die Gleichungen $x *_i y = a$ und $x *'_i y = b$ für alle $a, b \in Q$ eine Lösung besitzen. Die zweite Gleichung ist äquivalent zu $b *_{i+1} y = x$. Wir setzen diese in die erste Gleichung ein und erhalten die Bedingung, daß $(b *_{i+1} y) *_i y = a$ für alle $a, b \in Q$ eine Lösung besitzt. Weil das Prüfziffersystem alle Zwillingsfehler erkennt, ist $\varphi_b(y) = (b *_{i+1} y) *_i y$ eine Permutation und die Gleichung besitzt eine Lösung $y = \varphi_b^{-1}(a)$. □

Ganz analog zeigt man den folgenden Satz:

Satz 38 *Seien $(Q, *_i)$ die Quasigruppen eines Prüfziffersystems, das alle Sprungzwillingsfehler erkennt, dann ist die Quasigruppe $(Q, *_i)$, $i = 1, \ldots, n-2$, orthogonal zu den durch*

$$x *'_{c,i} y = z \quad :\Leftrightarrow \quad (c *_{i+2} y) *_{i+1} z = x$$

definierten Quasigruppen mit $c \in Q$, und $*_{n-1}$ ist orthogonal zu

$$x *''_{n-1} y = z \quad :\Leftrightarrow \quad y *_n z = x.$$

Wenn ein Prüfziffersystem über Quasigruppen der Ordnung m alle Zwillings- oder alle Sprungzwillingsfehler erkennt, dann können wir also eine ganze Reihe verschiedener orthogonaler lateinischer Quadrate konstruieren. Hat außerdem die Gleichung $a *_n x = x *_n b$ oder für feste $c_1, c_2 \in Q$ die Gleichung $(c_1 *_{i+2} y) *_{i+1} a = (c_2 *_{i+2} y) *_{i+1} b$ für alle $a, b \in Q$ eine Lösung, dann ist, wie man leicht durch die Definition der Quasigruppen sieht, $*'_{n-1}$ orthogonal zu $*''_{n-1}$ bzw. $*'_{c_1,i}$ orthogonal zu $*'_{c_2,i}$. In diesem Fall hätten wir drei paarweise orthogonale lateinische Quadrate der Ordnung m.

Die Frage, für welche Ordnungen ein Paar orthogonaler lateinischer Quadrate, bzw. ein griechisch-lateinisches Quadrat existiert, blieb lange ungeklärt. EULER wußte (1780), daß es kein griechisch-lateinisches Quadrat der Ordnung 2 gibt und er kannte Konstruktionen für ungerade oder durch 4 teilbare Ordnungen. Basierend auf vielfältigen Untersuchungen vermutete er, daß griechisch-lateinische Quadrate der Ordnung $4k+2$ nicht existieren. G. TARRY bewies 1900 durch Ausschluß aller Möglichkeiten, daß es kein griechisch-lateinisches Quadrat der Ordnung 6 gibt [8], womit er die Vermutung von EULER stützte. Trotzdem gelang es PARKER, BOSE und SHRIKHANDE 1960, also 180 Jahre nach EULERS Vermutung, ein griechisch-lateinisches Quadrat der Ordnung 10 zu konstruieren [8]. Außerdem lieferten sie eine Konstruktion für die fehlenden geraden Ordnungen, die nicht durch vier teilbar sind (außer für 2 und 6).

0A	7E	8B	6H	9C	3J	5I	4D	1G	2F
6I	1B	7F	8C	0H	9D	4J	5E	2A	3G
5J	0I	2C	7G	8D	1H	9E	6F	3B	4A
9F	6J	1I	3D	7A	8E	2H	0G	4C	5B
3H	9G	0J	2I	4E	7B	8F	1A	5D	6C
8G	4H	9A	1J	3I	5F	7C	2B	6E	0D
7D	8A	5H	9B	2J	4I	6G	3C	0F	1E
4B	5C	6D	0E	1F	2G	3A	7H	8I	9J
1C	2D	3E	4F	5G	6A	0B	9I	7J	8H
2E	3F	4G	5A	6B	0C	1D	8J	9H	7I

Grichisch-lateinisches Quadrat der Ordnung 10.

Ob dagegen drei paarweise orthogonale lateinische Quadrate der Ordnung 10 existieren, ist bis heute unbekannt. Wir können allerdings zeigen:

Satz 39 *([8]) Die Anzahl der paarweise orthogonalen lateinischen Quadrate der Ordnung n ist nicht größer als $n-1$.*

Beweis Die Elemente der lateinischen Quadrate können umbenannt werden, ohne die Eigenschaft der Orthogonalität zu zerstören. Daher permutieren wir die

Elemente so, daß die erste Zeile jedes lateinischen Quadrates gleich $0, 1, 2, \ldots$ ist. Nun folgt, da die 1 bei jedem Paar lateinischer Quadrate mit sich selbst in der ersten Zeile zusammenfällt, daß die 1 in der ersten Spalte nicht zweimal an der gleichen Position steht. Weil sie auch nicht an der Position $(0, 0)$ steht, gibt es folglich nur $n - 1$ mögliche Positionen für 1. □

4.6 Verallgemeinerte Assoziativität

Definition 20 *(*R. SCHAUFFLER [20]*) Die vier Quasigruppen $(Q, *_1)$, $(Q, *_2)$, $(Q, *_3)$ und $(Q, *_4)$, definiert auf der gleichen Grundmenge Q, erfüllen das verallgemeinerte Assoziativgesetz, wenn für alle $x, y, z \in Q$ die folgende Gleichung gilt:*

$$(x *_1 y) *_2 z = x *_3 (y *_4 z).$$

Eine Menge Ω von Quasigruppen heißt im Ganzen assoziativ (oder Assoziativsystem) wenn zu je zwei Quasigruppen $*_1, *_2 \in \Omega$ zwei weitere Quasigruppen $*_3, *_4 \in \Omega$ existieren, so daß diese das verallgemeinerte Assoziativgesetz erfüllen. $*_3, *_4$ heißen dann rechts assoziiert zu $*_1, *_2$ und $*_1, *_2$ links assoziiert zu $*_3, *_4$.

Die Menge der Quasigruppen mit den Elementen $0, 1, \ldots, n$ werde mit Ω_n bezeichnet. Es wäre sehr nützlich, wenn Ω_n im Ganzen assoziativ wäre, dann könnten wir immer umklammern und die notwendigen Bedingungen zur Fehlererkennung wären deutlich einfacher nachzuprüfen. Leider ist dies i.allg. nicht der Fall, wie das folgende Theorem zeigt.

Theorem 16 *(*R. SCHAUFFLER [20]*) Ω_n ist nur dann im Ganzen assoziativ, wenn $n \leq 3$ ist.*

Beweis Für $n = 1$ ist die Aussage trivial.

Für $n = 2$ haben wir die beiden Quasigruppen

$+$	0	1
0	0	1
1	1	0

und

$*$	0	1
0	1	0
1	0	1

Die erste Quasigruppe ist die zyklische Gruppe $(\mathbb{Z}_2, +)$, die zweite läßt sich darstellen durch $x * y = x + y + 1$ und ist isotop zu $(\mathbb{Z}_2, +)$. Wie man nun leicht sieht, gilt damit das verallgemeinerte Assoziativgesetz für die vier möglichen Paare $++$, $*+$, $+*$ und $**$.

Auch für $n = 3$ sind alle Quasigruppen isotop zu $(\mathbb{Z}_3, +)$, denn es existieren für alle Quasigruppen $(Q, *)$ vier Permutationen p_1, p_2, p_3, p_4, so daß

$$x * y = p_1(x) + p_2(y) = p_3(x + p_4(y))$$

gilt. Für die 12 Quasigruppen der Ordnung 3 geben wir in der folgenden Tabelle jeweils die zugehörigen Permutationen mit der genannten Eigenschaft an:

Q	p_1	p_2	p_3	p_4	Q	p_1	p_2	p_3	p_4
0 1 2 1 2 0 2 0 1	[012]	[012]	[012]	[012]	0 2 1 1 0 2 2 1 0	[012]	[021]	[012]	[021]
0 1 2 2 0 1 1 2 0	[021]	[012]	[021]	[021]	0 2 1 2 1 0 1 0 2	[021]	[021]	[021]	[012]
1 0 2 2 1 0 0 2 1	[012]	[102]	[012]	[102]	1 2 0 2 0 1 0 1 2	[012]	[120]	[012]	[120]
1 0 2 0 2 1 2 1 0	[021]	[102]	[021]	[201]	1 2 0 0 1 2 2 0 1	[021]	[120]	[021]	[210]
2 0 1 0 1 2 1 2 0	[012]	[201]	[012]	[201]	2 1 0 0 2 1 1 0 2	[012]	[210]	[012]	[210]
2 0 1 1 2 0 0 1 2	[021]	[201]	[021]	[102]	2 1 0 1 0 2 0 2 1	[021]	[210]	[021]	[120]

Für zwei beliebige Quasigruppen $*_1$, $*_2$ gilt also $x *_1 y = p_1(x) + p_2(y)$ und $x *_2 y = p_3(x + p_4(y))$. Es folgt

$$\begin{aligned}(x *_1 y) *_2 z &= p_3((p_1(x) + p_2(y)) + p_4(z)) \\ &= p_3(p_1(x) + (p_2(y) + p_4(z))) \\ &= x *_3 (y *_4 z),\end{aligned}$$

wobei wir die Quasigruppen $*_3, *_4$ durch $x *_3 y = p_3(p_1(x) + y)$ und $y *_4 z = p_2(y) + p_4(z)$ definieren.

Wir zeigen nun, daß es für $n \geq 4$ stets Quasigruppen gibt, die keine rechtsassozierten Quasigruppen besitzen. Wenn die Quasigruppen $*_1, *_2, *_3, *_4$ das verallgemeinerte Assoziativgesetz erfüllen, d.h. es gilt für alle $x, y, z \in Q$

$$(x *_1 y) *_2 z = x *_3 (y *_4 z),$$

dann gibt es nur n verschiedene Permutationen

$$\varphi_{y,z}(x) := (x *_1 y) *_2 z = x *_3 (y *_4 z),$$

denn $y *_4 z$ nimmt nur n unterschiedliche Werte an. Im folgenden Beispiel erhalten wir aber mehr als n verschiedene Permutationen. Diese Quasigruppen befinden sich daher nicht in einem Assoziativsystem.

Sei $p = (0\ 1)$ die Transposition, welche die Elemente 0 und 1 vertauscht. Wir definieren die Quasigruppen durch

$$x *_1 y := x + p(y) \pmod n \qquad x *_2 y := p(x) + y \pmod n.$$

Beide Quasigruppen entstehen aus der Gruppe $(\mathbb{Z}_n, +)$, indem die ersten beiden Spalten bzw. Zeilen vertauscht werden. Die Permutationen $\varphi_{0,i}$, $i = 0, \ldots, n-1$, sind paarweise verschieden, denn es gilt

$$\varphi_{0,i}(0) = (0 *_1 0) *_2 i = p(0+1) + i = 0 + i = i.$$

Für die Permutation $\varphi_{1,0}$ erhalten wir $\varphi_{1,0}(0) = p(0 + p(1)) + 0 = 1$ und $\varphi_{1,0}(1) = p(1 + p(1)) + 0 = 0$. Weil $\varphi_{0,1}(1) = p(1 + p(0)) + 1 = p(2) + 1 = 3$ gilt, unterscheidet sich $\varphi_{1,0}$ von den Permutationen $\varphi_{0,i}$ in wenigstens einer Stelle. Damit haben wir aber $n+1$ paarweise verschiedene Permutationen und die Quasigruppen $*_1, *_2$ genügen nicht dem verallgemeinerten Assoziativgesetz. □

Theorem 17 *(Aczél, Belousov, Hosszú [1]) Erfüllen die vier Quasigruppen $(Q, *_1)$, $(Q, *_2)$, $(Q, *_3)$ und $(Q, *_4)$ das verallgemeinerte Assoziativgesetz,*

$$(x *_1 y) *_2 z = x *_3 (y *_4 z), \tag{4.3}$$

dann existiert eine Verknüpfung $\circ$, so daß $(Q, \circ)$ eine Gruppe bildet, zu der die $*_i$ isotop sind. Im Detail: Es existieren 5 Permutation $\alpha, \beta, \gamma, \delta, \epsilon$ von Q, so daß

$$\begin{aligned} x *_1 y &= \delta^{-1}(\alpha(x) \circ \beta(y)), \\ x *_2 y &= \delta(x) \circ \gamma(y), \\ x *_3 y &= \alpha(x) \circ \epsilon(y), \\ x *_4 y &= \epsilon^{-1}(\beta(x) \circ \gamma(y)). \end{aligned} \tag{4.4}$$

Die Gruppe, zu der die $*_i$ isotop sind, ist bis auf Isomorphie eindeutig bestimmt. Andererseits erfüllen alle Isotopien einer beliebigen Gruppe mit den genannten Eigenschaften das verallgemeinerte Assoziativgesetz.

Beweis Die letzte Behauptung wird einfach durch Einsetzen von 4.4 in 4.3 gezeigt, wobei wir die Assoziativität der Gruppe $(Q, \circ)$ benutzen.

Um die erste Aussage beweisen zu können, definieren wir zunächst die Permutationen

$$\rho_i(x) := x *_i a, \qquad \lambda_i(x) := a *_i x, \qquad (i = 1, 2, 3, 4),$$

wobei a ein beliebiges, fest gewähltes Element aus Q ist. Wir setzen $x = z = a$ in 4.3 und erhalten

$$\rho_2(\lambda_1(y)) = \lambda_3(\rho_4(y)). \tag{4.5}$$

Wir erhalten nun durch Substitution von $x = a, y = \lambda_1^{-1}(\rho_2^{-1}(u)), z = \lambda_4^{-1}(\lambda_3^{-1}(v))$ und $x = \rho_1^{-1}(\rho_2^{-1}(u)), y = a, z = \lambda_4^{-1}(\lambda_3^{-1}(v))$ und $x = \rho_1^{-1}(\rho_2^{-1}(u)), y = \rho_4^{-1}(\lambda_3^{-1}(v))$, $z = a$ in 4.3 die Gleichungen

$$\rho_2^{-1}(u) *_2 \lambda_4^{-1}(\lambda_3^{-1}(v)) = \lambda_3(\lambda_1^{-1}(\rho_2^{-1}(u)) *_4 \lambda_4^{-1}(\lambda_3^{-1}(v)))$$

und

$$\rho_2^{-1}(u) *_2 \lambda_4^{-1}(\lambda_3^{-1}(v)) = \rho_1^{-1}(\rho_2^{-1}(u)) *_3 \lambda_3^{-1}(v)$$

und

$$\rho_2(\rho_1^{-1}(\rho_2^{-1}(u)) *_1 \rho_4^{-1}(\lambda_3^{-1}(v))) = \rho_1^{-1}(\rho_2^{-1}(u)) *_3 \lambda_3^{-1}(v).$$

Die letzten drei Gleichungen zeigen, daß alle 4 Ausdrücke in ihnen gleich sind. Wir benennen diesen gemeinsamen Wert mit

$$u \circ v.$$

Damit erhalten wir 4.4 wenn wir

$$\alpha := \rho_2\rho_1, \quad \delta := \rho_2, \quad \gamma := \lambda_3\lambda_4, \quad \epsilon := \lambda_3$$

und (vergleiche 4.5)

$$\beta := \lambda_3\rho_4 = \rho_2\lambda_1$$

setzen.

Setzen wir 4.4 in 4.3 ein, dann sehen wir, daß die Operation $x \circ y$ assoziativ ist und, als Isotopie einer Quasigruppe, ebenfalls eine Quasigruppe ist. Bekanntlich sind die Gruppen genau die assoziativen Quasigruppen und so bildet Q eine Gruppe mit der Operation $\circ$. Die Eindeutigkeit, bis auf Isomorphie, der Gruppe $(G, \circ)$ folgt aus dem folgenden Theorem.

Theorem 18 *([1]) Isotope Gruppen sind isomorph, d.h. wenn die Gruppen $(Q, \circ)$ und $(R, \cdot)$ isotop sind*

$$\varphi(x \circ y) = \psi(x) \cdot \chi(y), \tag{4.6}$$

dann sind sie isomorph

$$\kappa(x \circ y) = \kappa(x) \cdot \kappa(y). \tag{4.7}$$

Beweis Sei e das neutrale Element von $(Q, \circ)$. Wir setzen $y = e$ bzw. $x = e$ in 4.6 und erhalten

$$\psi(x) = \varphi(x) \cdot b^{-1}$$

und

$$\chi(y) = a^{-1} \cdot \varphi(y),$$

wobei $a = \psi(e)$, $b = \chi(e)$ und a^{-1}, b^{-1} die Inversen in $(R, \cdot)$ sind. Setzen wir diese Gleichungen wieder in 4.6 ein, dann erhalten wir

$$\varphi(x \circ y) = \varphi(x) \cdot b^{-1} \cdot a^{-1} \cdot \varphi(y)$$

und wenn wir diese Gleichung von links mit a^{-1} und von rechts mit b^{-1} durchmultiplizieren und

$$\kappa(x) := a^{-1} \cdot \varphi(x) \cdot b^{-1}$$

definieren, so erhalten wir 4.7. □

Korollar 17 *In einem Assoziativsystem sind alle Quasigruppen zur selben Gruppe isotop.*

Wie bereits erwähnt, können wir bei Quasigruppen, die das verallgemeinerte Assoziativgesetz erfüllen, die Voraussetzungen vereinfachen, die für das Erkennen der einzelnen Fehlerarten notwendig sind. Seien $*'_{i+1}, *'_i$ rechtsassozierte Quasigruppen von $*_{i+1}$ und $*_i$. Wir erhalten für die einzelnen Fehlertypen folgende Bedingungen:

Satz 40 *Sei Ω ein Assoziativsystem. Durch die Quasigruppen $*_i \in \Omega$ und die Gleichung*

$$g(x_n, x_{n-1}, \ldots, x_0) = (\ldots(x_n *_n x_{n-1}) *_{n-1} \ldots) *_1 x_0 = d$$

wird genau dann ein Prüfziffersystem definiert, wenn $*_n$ anti-symmetrisch ist und für jedes Paar $*_{i+1}, *_i$ rechtsassoziierte Quasigruppen $*'_{i+1}, *'_i \in \Omega$ existieren, so daß $*'_i$ anti-symmetrisch ist.

Sind die Quasigruppen $*_n, *'_i$ selbstorthogonal, dann erkennt dieses Prüfziffersystem zusätzlich noch alle Zwillingsfehler.

Beweis Die genannte Gleichung erkennt alle Einzehlfehler und sie erkennt alle Nachbarvertauschungen, falls

$$\begin{aligned} x *_n y &= y *_n x \quad \Rightarrow \quad x = y \\ (c *_{i+1} x) *_i y &= (c *_{i+1} y) *_i x \quad \Rightarrow \quad x = y. \end{aligned}$$

gilt. Die erste Bedingung ist nach Voraussetzung erfüllt. Um die zweite Bedingung nachzuweisen, nehmen wir an, daß $(c *_{i+1} x) *_i y = (c *_{i+1} y) *_i x$ gilt. Nach Voraussetzung existieren rechtsassozierte Quasigruppen $*'_{i+1}, *'_i$, wobei $*'_i$ anti-symmetrisch ist, so daß $(s *_{i+1} t) *_i u = s *'_{i+1} (t *'_i u)$ für alle $s, t, u \in Q$ gilt. Damit folgt $c *'_{i+1} (x *'_i y) = c *'_{i+1} (y *'_i x)$. Wir kürzen c auf beiden Seiten der Gleichung und erhalten $x *'_i y = y *'_i x$. Da $*'_i$ anti-symmetrisch ist, folgt $x = y$ und die zweite Bedingung ist erfüllt. Bei der Rückrichtung sieht man leicht, daß falls entweder $*'_i$ oder $*_n$ nicht anti-symmetrisch ist, die Gleichung nicht alle Nachbarvertauschungen erkennen kann.

Für den zweiten Teil der Behauptung müssen wir zeigen, daß aus $x *_n x = y *_n y$ bzw. $x *'_i x = y *'_i y$ die Gleichheit von x und y folgt. Dazu benutzen wir das folgende Lemma:

Lemma 19 *Eine selbstorthogonale Quasigruppe $(Q, *)$ ist anti-symmetrisch und es gilt:*

$$x * x = y * y \qquad \Rightarrow \qquad x = y.$$

Beweis Weil Q selbstorthogonal ist, sind die Paare $(x * y, y * x)$ für alle $x, y \in Q$ paarweise verschieden. Gäbe es $x, y \in Q$ mit $x \neq y$ und $x * y = y * x$, dann wären die Paare $(x * y, y * x)$ und $(y * x, x * y)$ gleich und Q wäre nicht selbstorthogonal. Ebenso folgt aus $x * x = y * y$, daß die Paare $(x * x, x * x)$ und $(y * y, y * y)$ gleich sind, also folgt entweder $x = y$ oder Q ist nicht selbstorthogonal. □

Für $m = 10$ existiert allerdings keine selbstorthogonale Quasigruppe in einem Assoziativsystem. Denn solch eine Quasigruppe wäre isotop zu der Diedergruppe, die dann eine vollständige Abbildung hätte. Aber D_5 besitzt keine vollständige Abbildung, Theorem 2 (Seite 19).

Bemerkung Selbstorthogonale Quasigruppen werden auch *anti-abelsch* genannt (vgl. Dénes, Keedwell [8]). Eine anti-symmetrische Quasigruppe muß aber nicht anti-abelsch sein, wie das folgende Gegenbeispiel zeigt:

$*$	0	1	2
0	0	1	2
1	2	0	1
2	1	2	0

Diese Quasigruppe ist offensichtlich anti-symmetrisch, aber es gilt $0 * 0 = 1 * 1$, also ist sie nicht anti-abelsch.

4.7 Quasigruppen isotop zu einer Gruppe

In diesem Abschnitt untersuchen wir die speziellen Eigenschaften von Prüfziffersystemen über Quasigruppen, die isotop zu einer Gruppe sind. Im vorherigen

Abschnitt haben wir gesehen, daß der dortige Ansatz mit Quasigruppen in einem Assoziativsystem ebenfalls zu diesen Quasigruppen führt.

Im folgenden seien die Quasigruppen $(Q, *_i)$ isotop zu der Gruppe $(Q, \cdot)$, d.h. es existieren Permutationen $\varphi_{i,1}, \varphi_{i,2}, \varphi_{i,3}$, so daß

$$x *_i y = \varphi_{i,3}(\varphi_{i,1}(x) \cdot \varphi_{i,2}(y))$$

gilt.

Die entsprechenden Voraussetzungen an die $\varphi_{i,j}$ für das Erkennen der einzelnen Fehlertypen erhält man nun einfach durch Einsetzen dieser Gleichungen in die Bedingungen 4.2, Seite 75f, für Quasigruppen. Wir zeigen allerdings eine etwas einfacher zu erfüllende Voraussetzung:

Satz 41 *Die Quasigruppen $(G, *_i)$ seien isotop zu $(G, \cdot)$. Sie definieren ein Prüfziffersystem, falls folgende Bedingungen erfüllt sind, $i = 1, \ldots, n-1$,*

$$\varphi_{n,1} \circ \varphi_{n,2}^{-1} \in Ant(G) \tag{4.8}$$

$$\varphi_{i,1} \circ \varphi_{i+1,3} \circ \varphi_{i+1,2} \circ \varphi_{i,2}^{-1} \in Ant(G) \tag{4.9}$$

$$\varphi_{i,1} \circ \varphi_{i+1,3} \in Aut(G). \tag{4.10}$$

Beweis Es ist $x *_n y = y *_n x$ äquivalent zu

$$\varphi_{n,3}(\varphi_{n,1}(x) \cdot \varphi_{n,2}(y)) = \varphi_{n,3}(\varphi_{n,1}(y) \cdot \varphi_{n,2}(x)).$$

Wir kürzen $\varphi_{n,3}$, setzen $\tilde{x} := \varphi_{n,2}(x)$ und $\tilde{y} := \varphi_{n,2}(y)$ womit $\varphi_{n,1}(\varphi_{n,2}^{-1}(\tilde{x})) \cdot \tilde{y} = \varphi_{n,1}(\varphi_{n,2}^{-1}(\tilde{y})) \cdot \tilde{x}$ folgt. Nach Voraussetzung ist $\varphi_{n,1} \circ \varphi_{n,2}^{-1}$ anti-symmetrisch und daher impliziert diese Gleichung $x = y$.

Nun nehmen wir an, daß $(c *_{i+1} x) *_i y = (c *_{i+1} y) *_i x$ bzw.

$$\begin{aligned} &\varphi_{i,3}\big[\varphi_{i,1}\big(\varphi_{i+1,3}(\varphi_{i+1,1}(c) \cdot \varphi_{i+1,2}(x))\big) \cdot \varphi_{i,2}(y)\big] \\ = &\varphi_{i,3}\big[\varphi_{i,1}\big(\varphi_{i+1,3}(\varphi_{i+1,1}(c) \cdot \varphi_{i+1,2}(y))\big) \cdot \varphi_{i,2}(x)\big] \end{aligned}$$

gilt. Wir definieren $\tilde{x} := \varphi_{i,2}(x)$, $\tilde{y} := \varphi_{i,2}(y)$, $\tilde{c} := \varphi_{i+1,1}(c)$ und $\alpha := \varphi_{i,1} \circ \varphi_{i+1,3}$, $\beta := \varphi_{i+1,2} \circ \varphi_{i,2}^{-1}$. Es folgt

$$\alpha(\tilde{c} \cdot \beta(\tilde{x})) \cdot \tilde{y} = \alpha(\tilde{c} \cdot \beta(\tilde{y})) \cdot \tilde{x}$$

und, weil α ist ein Automorphismus ist,

$$\alpha(\tilde{c}) \cdot \alpha \circ \beta(\tilde{x})) \cdot \tilde{y} = \alpha(\tilde{c}) \cdot \alpha \circ \beta(\tilde{y})) \cdot \tilde{x}.$$

Wir können $\alpha(\tilde{c})$ auf beiden Seiten der Gleichung kürzen. Da wir vorausgesetzt haben, daß $\alpha \circ \beta = \varphi_{i,1} \circ \varphi_{i+1,3} \circ \varphi_{i+1,2} \circ \varphi_{i,2}^{-1}$ eine anti-symmetrische Abbildung

ist, folgt aus der resultierenden Gleichung $\tilde{x} = \tilde{y}$ bzw. $x = y$. Damit haben wir gezeigt, daß die Quasigruppen $(G, *_i)$ ein Prüfziffersystem definieren. □

Bemerkung Die ersten beiden Eigenschaften sind auch notwendig für das Erkennen aller Nachbarvertauschungen.

Beispiel Sei φ eine anti-symmetrische Abbildung der Gruppe $(G, \cdot)$. Wir wählen $\varphi_{i,2} := \varphi^{i-1}$, $\varphi_{n,1} := \varphi^n$, $\varphi_{i,1} \circ \varphi_{i+1,3} = Id$ und $\varphi_{1,3}$ beliebig. Damit sind die Voraussetzungen des Satzes für die Quasigruppen $x *_i y := \varphi_{i,3}\big(\varphi_{i,1}(x) \cdot \varphi_{i,2}(y)\big)$ erfüllt. Es folgt, daß

$$(\dots(x_n *_n x_{n-1}) *_{n-1} \dots) *_1 x_0 = c$$

ein Prüfziffersystem definiert.

Konkret wählen wir die anti-symmetrische Abbildung $\varphi := [02413]$ der Gruppe $(\mathbb{Z}_5, +)$ (vgl. Seite 40) und $\varphi_{i,1} := \varphi_{i,3} := [10324]$. Damit erhalten wir für $n = 3$ die folgenden Quasigruppen:

$*_3$	0	1	2	3	4
0	1	4	2	3	0
1	2	3	0	1	4
2	0	1	4	2	3
3	4	2	3	0	1
4	3	0	1	4	2

$*_2$	0	1	2	3	4
0	0	2	1	3	4
1	1	3	4	0	2
2	2	1	3	4	0
3	3	4	0	2	1
4	4	0	2	1	3

$*_1$	0	1	2	3	4
0	0	3	2	4	1
1	1	0	3	2	4
2	2	4	1	0	3
3	3	2	4	1	0
4	4	1	0	3	2

4.7.1 Lineare Quasigruppen

Um die Anforderungen an die Quasigruppen zu verringern, ist es sinnvoll, sogenannte lineare Quasigruppen zu betrachten, da diese eine sehr einfache Darstellung besitzen.

Definition 21 *([5]) Sei $(Q, \cdot)$ eine Gruppe mit den Automorphismen ψ_1, ψ_2 und einem fest gewählten $c \in Q$. Die Quasigruppe $(Q, *)$ heißt* lineare Quasigruppe (der Gruppe $(Q, \cdot)$), *falls die Gleichung $x * y = \psi_1(x) \cdot c \cdot \psi_2(y)$ für alle $x, y \in Q$ erfüllt ist.*

Satz 42 *Die Menge der linearen Quasigruppen einer Gruppe $(Q, \cdot)$ bildet ein Assoziativsystem.*

Beweis Seien $(Q, *_1)$ und $(Q, *_2)$ lineare Quasigruppen der Gruppe $(Q, \cdot)$, d.h. $x *_1 y = \psi_1(x) \cdot c \cdot \psi_2(y)$ und $x *_2 y = \varphi_1(x) \cdot d \cdot \varphi_2(y)$ mit $\psi_1, \psi_2, \varphi_1, \varphi_2 \in Aut(Q, \cdot)$,

$c, d \in Q$. Es gilt:

$$\begin{aligned}(x *_1 y) *_2 z &= \varphi_1(\psi_1(x) \cdot c \cdot \psi_2(y)) \cdot d \cdot \varphi_2(z) \\ &= \varphi_1 \circ \psi_1(x) \cdot \varphi_1(c) \cdot \big(\varphi_1 \circ \psi_2(y) \cdot d \cdot \varphi_2(z)\big) \\ &= x *_3 (y *_4 z)\end{aligned}$$

mit $x *_3 y := \varphi_1 \circ \psi_1(x) \cdot \varphi_1(c) \cdot y$ und $y *_4 z := \varphi_1 \circ \psi_2(y) \cdot d \cdot \varphi_2(z)$ und $*_3$, $*_4$ sind lineare Quasigruppen der Gruppe $(Q, \cdot)$. □

Der folgende Spezialfall der linearen Quasigruppen wurde von ECKER und POCH untersucht. Dabei setzen wir $*_i := *$ für alle i:

Satz 43 *(*ECKER, POCH [9]*) Sei $\mathbb{Z}_n = \{0, \ldots, n-1\}$, $n \geq 2$ und $h, k, l \in \mathbb{Z}_n$ mit h und k teilerfremd zu n. Dann ist $Q_n = (\mathbb{Z}_n, *)$ mit $x * y = (h \cdot x + k \cdot y + l) \bmod n$ eine lineare Quasigruppe. Nachbarvertauschungen werden erkannt, falls $h-1$ und $h-k$ teilerfremd zu n sind, und Sprungtranspositionen werden erkannt, falls $h-1$, $h+1$ und $h^2 - k$ teilerfremd zu n sind.*

Beweis Die Abbildungen $x \mapsto h \cdot x$ und $y \mapsto k \cdot y$ sind Automorphismen der Gruppe $\mathbb{Z}_n$, da h und k teilerfremd zu n sind. Wir zeigen die erste Eigenschaft von 4.2 (Seite 75). Dazu sei $x * y = y * x$, also $hx + ky + l = hy + kx + l$. Es folgt $(h-k)(x-y) = 0$ und mit der Eigenschaft, daß $h-k$ eine Einheit in $\mathbb{Z}_n$ ist, $x - y = 0$ bzw. $x = y$. Nun nehmen wir an, daß $(c * x) * y = (c * y) * x$ gilt, d.h. $h(hc + kx + l) + ky + l = h(hc + ky + l) + kx + l$. Wir kürzen auf beiden Seiten die gleichen Terme und erhalten $hkx + ky = hky + kx$. k ist eine Einheit, daher können wir auch k kürzen. Es folgt $(h-1)(x-y) = 0$ und damit, weil wir $h-1$ als Einheit vorausgesetzt haben, $x = y$. Die entsprechenden Bedingungen für Sprungtranspositionen werden ganz analog gezeigt. □

Korollar 18 *(*ECKER, POCH [9]*) Sei q teilerfremd zu n, $1 \leq q \leq n-2$ $(n \geq 3)$. Wenn wir $h = -q$, $k = 1$ und $l = 0$ setzen, dann werden alle Nachbarvertauschungen erkannt, vorausgesetzt, daß $q+1$ teilerfremd zu n ist. Zusätzlich werden alle Sprungtranspositionen erkannt, falls $q-1$ teilerfremd zu n ist.*

Ist n gerade, dann ist entweder h oder $h-1$ gerade und daher nicht teilerfremd zu n. Also gibt es für gerades n keine lineare Quasigruppe, die alle Nachbarvertauschungen erkennt. Dies liegt u.a. auch am folgenden Zusammenhang:

Satz 44 *Ein Prüfziffersystem über linearen Quasigruppen der Gruppe $(G, \cdot)$ ist ein Prüfziffersystem über dieser Gruppe.*

Beweis Seien $(G, *_i)$, $i = 1, \ldots, n$, lineare Quasigruppen der Gruppe $(G, \cdot)$ mit $x *_i y = \psi_i(x) \cdot c_i \cdot \varphi_i(y)$. Wir zeigen die Behauptung durch vollständige Induktion

nach der Anzahl der beteiligten Quasigruppen. Für eine Quasigruppe $*_n$ haben wir

$$x_n *_n x_{n-1} = \psi_n(x_n) \cdot c_n \cdot \varphi_n(x_{n-1}).$$

Wir setzen $\tau_n(x) := \psi_n(x)$ und $\tau_{n-1}(x) := c_n \cdot \varphi_n(x)$ und erhalten den Induktionsanfang

$$x_n *_n x_{n-1} = \tau_n(x_n) \cdot \tau_{n-1}(x_{n-1}).$$

Nun gelte $(\dots(x_n *_n x_{n-1}) *_{n-1} \dots) *_i x_{i-1} = \tau_n(x_n) \cdot \ldots \cdot \tau_{i-1}(x_{i-1})$. Es folgt

$$(\tau_n(x_n) \cdot \ldots \cdot \tau_{i-1}(x_{i-1})) *_{i-1} x_{i-2} = \psi_{i-1}(\tau_n(x_n) \cdot \ldots \cdot \tau_{i-1}(x_{i-1})) \cdot c_{i-1} \cdot \varphi_{i-1}(x_{i-2}).$$

Mit $\tilde{\tau}_j(x) := \psi_{i-1} \circ \tau_j(x)$, $j = n, \dots, i-1$, $\tilde{\tau}_{i-2}(x) := c_{i-1} \cdot \varphi_{i-1}(x)$ und der Eigenschaft, daß ψ_{i-1} ein Automorphismus ist, haben wir $(\dots(x_n *_n x_{n-1}) *_{n-1} \dots) *_i x_{i-2} = \tilde{\tau}_n(x_n) \cdot \ldots \cdot \tilde{\tau}_{i-2}(x_{i-2})$ gezeigt. Damit folgt die Behauptung. □

Bemerkung Wir haben die Eigenschaft, daß ψ_n und die φ_i Automorphismen sind nicht benutzt. Die gleiche Aussage gilt daher auch für Quasigruppen, bei denen ψ_n und die φ_i nur Permutationen sind, aber keine Automorphismen.

Der Ansatz mit linearen Quasigruppen führt also auf die bereits in Kapitel 1 behandelten Prüfziffersysteme über Gruppen. Da wir schon gezeigt haben, daß über der Gruppe $\mathbb{Z}_{2n}$, kein Prüfziffersystem existiert, kann es auch kein Prüfziffersystem über linearen Quasigruppen der Gruppe $\mathbb{Z}_{2n}$ geben.

4.8 Total anti-symmetrische Quasigruppen

Ein naheliegender Ansatz zur Reduktion der Anzahl der Paare orthogonaler lateinischer Quadrate, ist es, anstatt verschiedener Quasigruppen, nur eine einzelne Quasigruppe zu betrachten. Dies hat auch praktische Vorteile, da wir in diesem Fall die Stellenzahl der zu sichernden Zahlen einfach erhöhen können, während bei verschiedenen Quasigruppen nicht klar ist, wie wir eine weitere Quasigruppe hinzu nehmen können, ohne dabei die Fehlererkennung zu zerstören.

Wir betrachten daher die Prüfziffersysteme der Form

$$(\dots((x_n * x_{n-1}) * x_{n-2}) * \dots) * x_0 = c. \tag{4.11}$$

Eine Quasigruppe heißt *total anti-symmetrisch*, falls sie anti-symmetrisch ist und außerdem gilt:

$$(c * x) * y = (c * y) * x \qquad \Rightarrow \qquad x = y. \tag{4.12}$$

Damit definiert die Gleichung 4.11 genau dann ein Prüfziffersystem, wenn $*$ eine total anti-symmetrische Quasigruppe ist.

4.8.1 Konstruktion

In diesem Abschnitt geben wir einen Algorithmus an, mit dessen Hilfe total anti-symmetrische Quasigruppen konstruiert werden können. Die Eigenschaft 4.12 können wir durch die Parastrophie $(Q, /)$ etwas anders formulieren. Wir setzen $\tilde{x} := c * x$, also $c/\tilde{x} = x$ und erhalten die neue Bedingung

$$\tilde{x} * y = (c * y) * (c/\tilde{x}) \quad \Rightarrow \quad c/\tilde{x} = y \quad (\text{für alle } c, \tilde{x}, y \in Q)$$

Nun sei $M = m(i, j, k)$ ein Würfel mit $m(i, j, k) := k$, $(i, j, k = 0, 1, \ldots, n-1)$. Wir konstruieren die Quasigruppe $*$ durch:

1) Für i von 0 bis $n-1$ tue 2-10

   2) Für j von 0 bis $n-1$ tue 3-7

      3) Sind alle Elemente $m(i, j, 0), \ldots, m(i, j, n-1)$ gestrichen, dann Abbruch

      4) Wähle ein Element $m(i, j, k)$ aus, das noch nicht gestrichen ist und setze $i * j := k$

      5) Streiche die Elemente $m(i+1, j, k), \ldots, m(n-1, j, k)$

      6) Streiche die Elemente $m(i, j+1, k), \ldots, m(i, n-1, k)$

      7) Streiche das Element $m(j, i, k)$, falls $j > i$

   8) Streiche die Elemente $m(c * y, c/x, x * y)$, $c = 0, \ldots, i-1,\ x = i,\ y = 0, \ldots, n-1$

   9) Streiche die Elemente $m(c * y, c/x, x * y)$, $c = i,\ x = 0, \ldots, i,\ y = 0, \ldots, n-1$

   10) Gibt es in den Schritten 8 oder 9 $x \neq c * y$ mit $x * y = (c * y) * (c/x)$, $c * y \leq i$, dann Abbruch

Erläuterungen: Die Schritte 5+6 sorgen dafür, daß eine Quasigruppe konstruiert wird. Durch Schritt 7 werden nur anti-symmetrische Quasigruppen erzeugt. Die Schritte 8-10 beschleunigen den Algorithmus erheblich, denn es werden schon während der Konstruktion diejenigen Quasigruppen ausgeschlossen, die nicht total anti-symmetrisch sind. Wir zeigen dies in dem folgenden Satz:

Satz 45 *Eine Quasigruppe $(Q, \cdot)$ kann genau dann mit dem Algorithmus konstruiert werden, wenn sie total anti-symmetrisch ist.*

Beweis Wir zeigen zunächst mit vollständiger Induktion, daß eine total anti-symmetrische Quasigruppe $(Q, \cdot)$ mit dem Algorithmus konstruiert werden kann. Es sei $(i', j') < (i, j)$ falls $i' < i$ oder falls $i' = i$ und $j' < j$ ist. Wir beginnen die

Induktion mit $(i,j) = (0,0)$. Da für $i = j = 0$ noch kein Element gestrichen ist, können wir $0 * 0 := 0 \cdot 0$ setzen. Nun gelte $i' * j' = i' \cdot j'$ für alle $(i',j') < (i,j)$.
Annahme: In Durchlauf (i,j) ist das Element $m(i,j,i \cdot j)$ gestrichen, d.h. es existiert ein $(i',j') < (i,j)$, so daß in Durchlauf (i',j') $m(i,j,i \cdot j)$ gestrichen wurde. Wir unterscheiden die folgenden Fälle:
Fall 1: $m(i,j,i \cdot j)$ wurde in Schritt 5 oder 6 gestrichen. Entweder gilt dann $j = j'$ und $i' \cdot j = i' * j = i \cdot j$ oder $i = i'$ und $i \cdot j' = i * j' = i \cdot j$. In beiden Fällen folgt $(i',j') = (i,j)$ im Widerspruch zu $(i',j') < (i,j)$.
Fall 2: $m(i,j,i \cdot j)$ wurde in Schritt 7 gestrichen, also $i' = j$ und $j' = i$. Es folgt $j \cdot i = j * i = i \cdot j$. Nach Voraussetzung an $(Q,\cdot)$ impliziert dies $i = j$ und damit $(i',j') = (i,j)$, Widerspruch.
Fall 3: $m(i,j,i \cdot j)$ wurde in Schritt 8 gestrichen, d.h. $i' < i$ und es existiert ein $c < i'$, $y \in \{0,\ldots,n-1\}$ mit $c \cdot y = c * y = i$, $c/i' = j$, $i' \cdot y = i' * y = i \cdot j$. Die zweite Gleichung ist äquivalent zu $i' = c * j = c \cdot j$. Wir setzen diese und die erste Gleichung in die dritte ein und erhalten $(c \cdot j) \cdot y = (c \cdot y) \cdot j$. Dies impliziert nach Voraussetzung $j = y$ und wir erhalten aus der dritten Gleichung $i' = i$ im Widerspruch zu $i' < i$.
Fall 4: $m(i,j,i \cdot j)$ wurde in Schritt 9 gestrichen, d.h. $i' < i$ und es existiert ein $x \leq i' < i$, $y \in \{0,\ldots,n-1\}$ mit $i' \cdot y = i' * y = i$, $i'/x = j$ und $x \cdot y = x * y = i \cdot j$. Auch hier folgt durch Einsetzen in die letzte Gleichung $(i' \cdot j) \cdot y = (i' \cdot y) \cdot j$. Demnach ist $j = y$ und $i = x$ im Widerspruch zu $x < i$.

Damit ist die Annahme widerlegt, d.h. das Element $m(i,j,i \cdot j)$ ist nicht gestrichen. Wir setzen daher $i * j := i \cdot j$.

Der Algorithmus bricht nicht in Schritt 3 ab, da wir gezeigt haben, daß mindestens ein Element, nämlich $m(i,j,i \cdot j)$ nicht gestrichen ist. Wenn der Algorithmus in Schritt 10 abbrechen würde, dann gäbe es $c,x \in \{0,\ldots,i\}$, $y \in \{0,\ldots,n-1\}$, $c * y \leq i$, $x \neq c * y = c \cdot y$ mit $x * y = (c * y) * (c/x)$. Wir setzen $z := c/x$ bzw. $x = c * z = c \cdot z$ und es folgt $x \cdot y = (c \cdot y) \cdot z = (c \cdot z) \cdot y$. Wir erhalten $z = y$, woraus $x = c \cdot y$ folgt im Widerspruch zu $x \neq c \cdot y$. Damit haben wir gezeigt, daß die Quasigruppe $(Q,\cdot)$ mit dem Algorithmus konstruiert werden kann.

Sei nun andererseits $(Q,*)$ eine Quasigruppe, die mit dem Algorithmus konstruiert wurde. Wir nehmen an, es gäbe $c,x,y \in \{0,\ldots,n-1\}$, $x \neq c * y$ mit $x * y = (c * y) * (c/x)$. Sei $i := max(c,x)$, also $c,x \leq i$. Wir betrachten nun den Algorithmus in Durchlauf i bei den Schritten 8-10.
Fall 1: $c = i$, $x \leq i$, $c * y \leq i$. In Schritt 9 gibt es demnach $x \neq c * y$ mit $x * y = (c * y) * (c/x)$, $c * y \leq i$ und der Algorithmus bricht ab.
Fall 2: $c < i$, $x = i$, $c * y \leq i$. In Schritt 8 sind damit die Bedingungen von Schritt

10 erfüllt und der Algorithmus bricht ab.
Fall 3: $c = i$, $x \leq i$, $c * y > i$. Es ist $(x, y) < (c * y, c/x)$. In Schritt 9 wurde das Element $m(c * y, c/x, x * y)$ gestrichen, daher ist die Wahl $(c * y) * (c/x) = x * y$ im Durchlauf $(i', j') = (c * y, c/x)$ nicht möglich, im Widerspruch dazu, daß wir $(Q, *)$ mit dem Algorithmus konstruiert haben.
Fall 4: $c < i$, $x = i$, $c * y > i$. Auch hier ist $(x, y) < (c * y, c/x)$ und es folgt analog zu Fall 3, daß in Schritt 8 $m(c * y, c/x, x * y)$ gestrichen wurde und deshalb ist $(c * y) * (c/x) \neq x * y$, Widerspruch.
Nun nehmen wir an, es gäbe $x \neq y$ mit $x * y = y * x$, o.B.d.A. $x < y$. Im Durchlauf (x, y) wird das Element $m(y, x, x * y)$ gestrichen und kann daher im Durchlauf (y, x) nicht ausgewählt werden, also $x * y \neq y * x$, Widerspruch.
Damit ist der Beweis des Satzes abgeschlossen. □

Wie bei den anti-symmetrischen Abbildungen kann man alle total anti-symmetrischen Quasigruppen konstruieren, indem man in Schritt 4) nacheinander alle nicht gestrichenen Elemente auswählt und den Algorithmus rekursiv aufruft.

Mit diesem rekursiven Algorithmus haben wir die Anzahl der total anti-symmetrischen Quasigruppen mit einer Linkseins und derer, die zusätzlich noch alle Sprungtranspositionen erkennen, bestimmt.

Ordnung	Anzahl (Gesamt)	total anti-symmetrisch	Sprungtrans-positionen
3	2	1	0
4	24	2	2
5	1.344	18	12
6	1.128.960	0	0
7	12.198.297.600	2.400	480
8	2.697.818.265.354.240	31.680	1.440

Die Werte für $n = 3, 4, 5, 6$ bestätigen die von ECKER und POCH [9] bestimmte Anzahl. Die Rechenzeit für $n = 7$ betrug ca. 6 Minuten, die für $n = 8$ ca. 12,5 Stunden. Für die Konstruktion der total anti-symmetrischen Quasigruppen haben wir die folgenden Sätze ausgenutzt.

Satz 46 *Ist $(Q, *)$ eine total anti-symmetrische Quasigruppe, φ und ψ Permutationen, dann wird durch $x \cdot y := \psi^{-1}(\psi(x) * \varphi(y))$ ebenfalls eine total anti-symmetrische Quasigruppe definiert, falls $\psi \circ \varphi^{-1} \in Ant(Q, *)$ oder $(Q, \cdot)$ eine Linkseins besitzt.*

Beweis Wir nehmen zunächst an, daß $(c \cdot x) \cdot y = (c \cdot y) \cdot x$ gilt, dann folgt mit der Definition von $(Q, \cdot)$, daß $(\psi(c) * \varphi(x)) * \varphi(y) = (\psi(c) * \varphi(y)) * \varphi(x)$ ist. Da $(Q, *)$ total anti-symmetrisch ist, folgt $\varphi(x) = \varphi(y)$ und demnach $x = y$.

Falls $(Q,\cdot)$ eine Linkseins 0 besitzt, dann folgt aus $x\cdot y=(0\cdot x)\cdot y=(0\cdot y)\cdot x=y\cdot x$ die Gleichung $x=y$. Also ist $(Q,\cdot)$ total anti-symmetrisch.
Gilt $\psi\circ\varphi^{-1}\in Ant(Q,*)$, dann impliziert $x\cdot y=\psi^{-1}(\psi(x)*\varphi(y))=\psi^{-1}(\psi(y)*\varphi(x))=y\cdot x$ die Gleichung $\psi(x)*\varphi(y)=\psi(y)*\varphi(x)$ und damit $\psi(\varphi^{-1}(\varphi(x)))*\varphi(y)=\psi(\varphi^{-1}(\varphi(y)))*\varphi(x)$. Nach Voraussetzung ist $\psi\circ\varphi^{-1}$ eine anti-symmetrische Abbildung der Quasigruppe $(Q,*)$, deshalb folgt $\varphi(x)=\varphi(y)$ und $x=y$. Folglich ist $(Q,\cdot)$ total anti-symmetrisch. □

Korollar 19 *Ist $(Q,*)$ eine total anti-symmetrische Quasigruppe, so ist auch $(Q,\cdot)$ mit $x\cdot y:=\varphi^{-1}(\varphi(x)*\varphi(y))$ total anti-symmetrisch.*

Beweis Setze $\psi:=\varphi$, dann ist $\psi\circ\varphi^{-1}=Id\in Ant(Q,*)$, weil $(Q,*)$ anti-symmetrisch ist.

Satz 47 *Sei $(Q,*)$ eine total anti-symmetrische Quasigruppe, dann existiert eine total anti-symmetrische Quasigruppe mit Linkseins, $(Q,\cdot)$ und eine anti-symmetrische Abbildung $\varphi^{-1}\in Ant(Q,\cdot)$ mit $x*y=x\cdot\varphi(y)$.*

Beweis Sei $\varphi(x):=0*x$ und $x\cdot y:=x*\varphi^{-1}(y)$, dann gilt $y=0*\varphi^{-1}(y)$ und demnach $0\cdot y=0*\varphi^{-1}(y)=y$ für alle $y\in Q$, also besitzt $(Q,\cdot)$ eine Linkseins und ist nach Satz 46 total anti-symmetrisch. Außerdem gilt $\varphi^{-1}\in Ant(Q,\cdot)$, denn aus $\varphi^{-1}(x)\cdot y=\varphi^{-1}(y)\cdot x$ folgt $\varphi^{-1}(x)*\varphi^{-1}(y)=\varphi^{-1}(y)*\varphi^{-1}(x)$. $(Q,*)$ ist anti-symmetrisch daher erhalten wir $\varphi^{-1}(x)=\varphi^{-1}(y)$ bzw. $x=y$. Damit haben wir $x\cdot\varphi(y)=x*\varphi^{-1}(\varphi(y))=x*y$ und die Behauptung ist bewiesen. □

Wir können also alle total anti-symmetrischen Quasigruppen bestimmen, indem wir die total anti-symmetrischen Quasigruppen mit Linkseins und deren anti-symmetrische Abbildungen konstruieren.

Satz 48 *Eine total anti-symmetrische Quasigruppe mit Linkseins ist isomorph zu einer total anti-symmetrischen Quasigruppe mit Linkseins $(Q,\cdot)$, $Q=\{0,\ldots,n-1\}$, für die*

$$1\cdot x\leq x+2,\qquad x=0,\ldots,n-1$$

gilt.

Beweis Sei $(Q,*)$ eine total anti-symmetrische Quasigruppe mit Linkseins 0. Wir beweisen den Satz mit vollständiger Induktion.

Falls $1*0\leq 2$ ist (Bem.: in diesem Fall gilt $1*0=2$) , dann ist nichts zu zeigen. Gilt $1*0>0+2=2$, dann definieren wir $\varphi(1*0):=2$, $\varphi(2):=1*0$ und

$\varphi(x) := x$ sonst. Die Quasigruppe $(Q, \cdot)$ mit $x \cdot y := \varphi(\varphi(x) * \varphi(y))$ ist isomorph zu $(Q, *)$, da $\varphi^{-1} = \varphi$ gilt und es ist $1 \cdot 0 = \varphi(\varphi(1) * \varphi(0)) = \varphi(1 * 0) = 2 \leq 0 + 2$.

Nun sei $(Q, *)$ isomorph zu $(Q, *')$, und es gelte $1 *' x \leq x + 2$ für $0 \leq x \leq k < n - 3$. Ist $1 *' (k+1) > k + 3$, dann setzen wir $\varphi(1 *' (k+1)) := k + 3$, $\varphi(k + 3) := 1 *' (k + 1)$ und $\varphi(x) := x$ sonst. Damit erfüllt die Quasigruppe $(Q, \cdot)$ mit $x \cdot y := \varphi(\varphi(x) * \varphi(y))$ die gesuchte Bedingung, denn es gilt

$$1 \cdot x = \varphi(\varphi(1) *' \varphi(x)) = \varphi(1 *' x) = 1 *' x \leq x + 2, \qquad \text{für } 0 \leq x \leq k$$

und

$$1 \cdot (k+1) = \varphi(\varphi(1) *' \varphi(k+1)) = \varphi(1 *' (k+1)) = k + 3 = (k+1) + 2$$

und $(Q, \cdot)$ ist isomorph zu $(Q, *)$.

Da für $x \geq n - 3$ die Aussage $1 * x \leq x + 2$ trivial ist (denn schließlich ist $1 * x \leq n - 1$ für alle $x \in Q$), haben wir damit den Satz bewiesen. □

Wir brauchen also nur solche Quasigruppen zu konstruieren, welche eine Linkseins besitzen, und für die $1 * x \leq x + 2$ gilt. Die Gesamtanzahl erhalten wir durch das Auszählen der verschiedenen isomorphen Quasigruppen. Damit verkürzt sich die Rechenzeit erheblich, z.B. für $n = 8$ von schätzungsweise einer Woche auf etwa einen halben Tag.

Außerdem haben wir den Algorithmus noch dadurch beschleunigt, daß wir beim Streichen eines Elements gleich überprüfen, ob bereits alle Elemente $m(i, j, k)$, $k = 0, \ldots, n - 1$ gestrichen wurden. Dies erreichen wir, indem wir mitzählen, wieviele Elemente noch nicht gestrichen sind. Sind alle Elemente gestrichen, so können wir den aktuellen Durchlauf abbrechen und in der Rekursion eine Ebene höher gehen.

Für $n = 10$ haben wir auch nach längerer Suche keine total anti-symmetrische Quasigruppe gefunden. Da die Zahl der Quasigruppen mit n stark anwächst (siehe McKay, Rogoyski [16]), konnten wir allerdings nur einen sehr geringen Prozentsatz überprüfen. Ecker und Poch haben sogar die Vermutung ausgesprochen, daß Quasigruppen der Ordnung $4k + 2$ nicht total anti-symmetrisch sein können. Wir stützen diese Vermutung durch den folgenden Satz:

Satz 49 *Es existiert keine zu D_s, $s > 2$ ungerade, isotope Quasigruppe, die total anti-symmetrisch ist. Die Quasigruppen, die zu $\mathbb{Z}_{2k}$ isotop sind, können nicht anti-symmetrisch sein.*

Beweis Nehmen wir an, wir hätten eine total anti-symmetrische Quasigruppe $(Q, *)$, die isotop zu D_s ist, d.h. $x * y = \varphi_3(\varphi_1(x)\varphi_2(y))$. Damit ist $(c * x) * y = \varphi_3(\varphi_1(c * x)\varphi_2(y)) = \varphi_3(\varphi_1(\varphi_3(\varphi_1(c)\varphi_2(x)))\varphi_2(y))$. Mit $\tilde{\varphi} = \varphi_1 \circ \varphi_3$, $\tilde{c} =$

$\varphi_1(c)$, $\tilde{x} = \varphi_2(x)$ und $\tilde{y} = \varphi_2(y)$ folgt $(c * x) * y = \varphi_3(\tilde{\varphi}(\tilde{c}\tilde{x})\tilde{y})$. Da $(Q, *)$ total anti-symmetrisch ist, folgt, daß für alle $\tilde{c} \in D_s$ die Permutation $\tilde{\varphi}(\tilde{c}\tilde{x})$ eine anti-symmtrische Abbildung von D_s ist. Nach Satz 4 (Seite 30) ist damit auch $\tilde{\varphi}(\tilde{c}^{-1}\tilde{c}\tilde{x})\tilde{c}^{-1} = \tilde{\varphi}(\tilde{x})\tilde{c}^{-1} \in Ant(D_s)$ im Widerspruch zu Satz 26 (Seite 54).

Da $\mathbb{Z}_n$ keine anti-symmetrische Abbildung besitzt, gilt dies auch für alle Isotopien. □

Wenn die Vermutung stimmt, dann existiert kein Prüfziffersystem basierend auf einer einzelnen Quasigruppe der Ordnung 10. Falls es allerdings eine total anti-symmetrische Quasigruppe der Ordnung 10 geben sollte, dann bedeutet dies allerdings noch nicht, daß diese eine bessere Fehlererkennung der anderen Fehler (Zwillingsfehler, Sprungzwillingsfehler) bietet als ein Prüfziffersystem basierend auf der Diedergruppe D_5. Im nächsten Abschnitt werden wir zeigen, daß bestimmte Quasigruppen nicht alle Fehler erkennen können.

Für ungerade n gilt allerdings:

Satz 50 *Es existieren total anti-symmetrische Quasigruppen für alle ungeraden n.*

Beweis In $(\mathbb{Z}_n, +)$ sind die Abbildungen $\varphi(x) = c - x$ anti-symmetrisch für alle $c \in \mathbb{Z}_n$. Wir definieren nun $x * y := -x + y$ und haben damit $(c * x) * y = -(-c + x) + y = c - x + y$ und $*$ ist total anti-symmetrisch. □

4.9 Quasigruppen mit Vorzeichen

Dieser Abschnitt verallgemeinert den Begriff des Vorzeichens auf Quasigruppen.

Definition 22 *Eine (endliche) Quasigruppe $(Q, *)$ mit einem Homomorphismus $sgn : Q \to \{-1, +1\}$ heißt Quasigruppe mit Vorzeichen sgn. Ist sgn surjektiv, dann heißt das Vorzeichen* nicht-trivial. *Die Menge der positiven bzw. negativen Elemente wird mit Q^+ bzw. Q^- bezeichnet.*

Eigenschaften:

1. Besitzt $(Q, *)$ eine Links- oder Rechtseins e, dann ist $sgn(e) = 1$.
2. $Q = Q^+ \cup Q^-$ und $Q^+ \cap Q^- = \emptyset$.
3. Es gilt $|Q^+| = |Q^-|$, falls das Vorzeichen auf Q nicht trivial ist, sonst $Q^+ = Q$ und $Q^- = \emptyset$.

zu 1: $sgn(e) = sgn(e * e) = sgn(e)sgn(e) = 1$.
zu 2: klar.
zu 3: Sei $x \in Q^- \neq \emptyset$, dann ist $x * x \in Q^+$, denn $sgn(x * x) = sgn(x)sgn(x) = 1$, also ist das Vorzeichen sgn nicht trivial. Es ist $x * Q^+ \subseteq Q^-$, und da $x * y_1 \neq x * y_2$ für $y_1 \neq y_2$ gilt, haben wir $|Q^+| \leq |Q^-|$. Ebenso gilt $x * Q^- \subseteq Q^+$ und damit $|Q^-| \leq |Q^+|$. Folglich haben wir $|Q^+| = |Q^-|$. Ist das Vorzeichen trivial, dann gilt $Q^- = \emptyset$ und $Q = Q^+$.

Satz 51 *Quasigruppen mit ungerader Ordnung besitzen nur die triviale Vorzeichenfunktion $sgn(x) = 1$.*
Eine Quasigruppe der Ordnung $2k$ besitzt ein nicht-triviales Vorzeichen genau dann, wenn sie eine Unterquasigruppe der Ordnung k besitzt.

Beweis Falls eine Quasigruppe ein nicht-triviales Vorzeichen besitzt, dann gilt $|Q^+| = |Q^-|$ und die Anzahl der Elemente der Quasigruppe ist gerade, denn $|Q| = |Q^+| + |Q^-| = k + k = 2k$. Q^+ ist in diesem Fall eine Unterquasigruppe der Ordnung k, denn wenn $x, y \in Q^+$ sind, dann gilt $sgn(x * y) = sgn(x)sgn(y) = 1 \cdot 1 = 1$ und es folgt $x * y \in Q^+$.
Andererseits können wir mit einer Unterquasigruppe U der Ordnung k ein Vorzeichen definieren durch

$$sgn(x) = \begin{cases} 1 & \text{falls } x \in U \\ -1 & \text{sonst.} \end{cases}$$

Wir müssen zeigen, daß $sgn(x * y) = sgn(x)sgn(y)$ gilt. Dazu sei $x \in U$. Für $y \in U$ sind auch die Elemente $x * y \in U$. Da $x * y_1 \neq x * y_2$ für $y_1 \neq y_2$ gilt, ist $x * U = U$. Aus diesem Grund ist für $z \notin U$ auch $x * z \notin U$, denn wenn $x * z \in U$ wäre, dann gäbe es ein $y \in U$ mit $x * z = x * y$ und damit ist $y = z$, Widerspruch. Genauso folgt, daß für $z \notin U$ auch $z * x \notin U$ ist. Nun sei $z \in \bar{U} := Q \setminus U$. Wir haben gezeigt, daß für $x \in U$, $z * x$ und $x * z$ Elemente von $\bar{U}$ sind, d.h. $z * U = U * z = \bar{U}$, denn die Elemente $z * x$ bzw. $x * z$ sind für verschiedene x ebenfalls verschieden und $|U| = |\bar{U}|$. Ist $y \in \bar{U}$, dann muß $z * y, y * z \in U$ gelten, denn sonst gäbe es ein $x \in U$ mit $z * y = z * x$ bzw. $y * z = x * z$ und es würde $x = y$ und daher ein Widerspruch folgen. Damit haben wir gezeigt, daß sgn ein Homomorphismus ist. □

Theorem 19 *Sei Q eine Quasigruppe der Ordnung $4k + 2$ mit nicht trivialem Vorzeichen, dann besitzt Q keine vollständige Abbildung.*

Beweis Siehe Beweis auf der Seite 48.

Wir haben bereits gezeigt, daß in einer Isotopieklasse entweder alle oder keine Quasigruppe eine vollständige Abbildung besitzt. Daher wissen wir, daß jede Quasigruppe der Ordnung $4k+2$, die zu einer Quasigruppe mit Vorzeichen isotop ist, keine vollständige Abbildung besitzt. Also gilt:

Korollar 20 *Prüfziffersysteme zur Basis $4k+2$ über Quasigruppen, von denen wenigstens eine isotop zu einer Quasigruppe mit Vorzeichen ist, erkennen nicht alle Zwillings- und auch nicht alle Sprungzwillingsfehler.*

Das Gleiche gilt auch für alle Parastrophien dieser Quasigruppen.

4.9.1 Beispiele

Die Quasigruppe $(\mathbb{Z}_n, *)$, n gerade, mit

$$x * y = \begin{cases} (x+y) \bmod n & \text{falls } x \text{ gerade} \\ (x-y-k) \bmod n & \text{falls } x \text{ ungerade,} \end{cases}$$

$k \in \mathbb{Z}_n$ gerade, besitzt das nicht-triviale Vorzeichen

$$sgn(x) = \begin{cases} 1 & \text{falls } x \text{ gerade} \\ -1 & \text{falls } x \text{ ungerade,} \end{cases}$$

denn die Menge der geraden Zahlen bildet eine Unterquasigruppe von $(\mathbb{Z}_n, *)$. Dies wird besonders deutlich, wenn wir die Zeilen und Spalten so permutieren, daß zuerst die geraden und dann die ungeraden Zahlen kommen. Für $n = 10, k = 0$ haben wir z.B. die Quasigruppe

*	0	2	4	6	8	1	3	5	7	9
0	0	2	4	6	8	1	3	5	7	9
2	2	4	6	8	0	3	5	7	9	1
4	4	6	8	0	2	5	7	9	1	3
6	6	8	0	2	4	7	9	1	3	5
8	8	0	2	4	6	9	1	3	5	7
1	1	9	7	5	3	0	8	6	4	2
3	3	1	9	7	5	2	0	8	6	4
5	5	3	1	9	7	4	2	0	8	6
7	7	5	3	1	9	6	4	2	0	8
9	9	7	5	3	1	8	6	4	2	0

Wir zeigen, daß $(\mathbb{Z}_n, *)$ für alle geraden n, k eine Quasigruppe definiert. Wir setzen $y := a - x$, falls x gerade und $y := x - a - k$, falls x ungerade ist, und haben

so eine eindeutige Lösung der Gleichung $x * y = a$ mit vorgegebenen $x, a \in \mathbb{Z}_n$. Sind $y, b \in \mathbb{Z}_n$ vorgegeben und y und b entweder beide gerade oder beide ungerade, dann ist $x := b - y$ gerade und Lösung der Gleichung $x * y = b$. Falls y und b unterschiedliche Vorzeichen haben, dann ist $x := b + y + k$ ungerade und löst die Gleichung $x * y = b$. Um zu zeigen, daß diese Lösung eindeutig ist, nehmen wir an, daß $x_1 * y = x_2 * y = b$ gilt. Haben x_1 und x_2 das gleiche Vorzeichen, dann können wir y und ggf. k auf beiden Seiten der Gleichung kürzen und erhalten $x_1 = x_2$. Gilt dagegen x_1 ungerade und x_2 gerade, so haben wir die Gleichung $x_1 - y - k = x_2 + y$ bzw. $x_1 = x_2 + 2y + k$. Auf der rechten Seite der Gleichung steht eine gerade, auf der linken eine ungerade Zahl, da n gerade ist haben wir daher einen Widerspruch. Damit folgt, daß $(\mathbb{Z}_n, *)$ eine Quasigruppe ist. □

Weitere Beispiele können wir aus der Arbeit von ECKER und POCH entnehmen. Sie definieren über den folgenden Quasigruppen der Ordnung $2n = 4k + 2$ ein Prüfziffersystem (siehe letzten Abschnitt). Für $x, y \in \mathbb{Z}_n$ sei $x *_1 y$ bzw. $x *_2 y$ definiert durch:

$$\begin{array}{rcl}
0 \le x, y \le n-1 & : & x *_1 y = (y - x) \bmod n \\
0 \le x \le n-1, n \le y \le 2n-1 & : & x *_1 y = n + ((y - x) \bmod n) \\
n \le x \le 2n-1, 0 \le y \le n-1 & : & x *_1 y = n + ((-x - y) \bmod n) \\
n \le x, y \le 2n-1 & : & x *_1 y = (y - x + 1) \bmod n
\end{array}$$

bzw.

$$\begin{array}{rcl}
0 \le x, y \le n-1 & : & x *_2 y = (y - x) \bmod n \\
0 \le x \le n-1, n \le y \le 2n-1 & : & x *_2 y = n + ((y + x) \bmod n) \\
n \le x \le 2n-1, 0 \le y \le n-1 & : & x *_2 y = n + ((y - x + 1) \bmod n) \\
n \le x, y \le 2n-1 & : & x *_2 y = (-y + x + 1) \bmod n
\end{array}$$

Auch diese Quasigruppen besitzen ein nicht triviales Vorzeichen, denn für $0 \le x, y \le n-1$ gilt $0 \le x *_{1,2} y \le n-1$ und damit habe wir eine Unterquasigruppe der Ordnung $2k+1$. Also können wir Korollar 20 (Seite 96) anwenden und es folgt, daß über dieser Quasigruppe kein Prüfziffersystem existiert, welches alle (Sprung-)Zwillingsfehler erkennt.

Im Gegensatz zu den Gruppen, muß eine Quasigruppe der Ordnung $4k + 2$ nicht unbedingt ein nicht-triviales Vorzeichen besitzen, wie das folgende Beispiel zeigt.

$*$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	6	2	9	4	3	7	5	1	0	8
2	5	6	4	8	7	3	1	9	2	0
3	9	5	6	7	0	1	3	8	4	2
4	3	8	5	6	1	2	9	0	7	4
5	8	3	0	5	6	9	4	2	1	7
6	7	0	3	2	5	6	8	4	9	1
7	2	4	7	1	9	8	0	3	6	5
8	4	7	1	9	8	0	2	6	5	3
9	1	9	8	0	2	4	7	5	3	6

Die Identität ist eine vollständige Abbildung dieser Quasigruppe, denn die Elemente $x * x$ auf der Diagonalen sind paarweise verschieden. Nach Theorem 19 kann sie daher nicht isotop zu einer Quasigruppe mit nicht-trivialem Vorzeichen sein, insbesondere besitzt sie selbst nur das triviale Vorzeichen.

4.10 Total anti-symmetrische Abbildungen

Wenn wir den Ansatz im Abschnitt „Total anti-symmetrische Quasigruppen" mit Kapitel 1 vergleichen (insbesondere Satz 1, Seite 15), dann ist der deutlichste Unterschied, daß wir die Prüfziffer nur mit einer Quasigruppe berechnet haben, ohne eine Permutation auf die einzelnen Elemente anzuwenden.

Wir untersuchen nun diese Möglichkeit mit dem Ansatz

$$((\dots((\varphi^n(x_n) * \varphi^{n-1}(x_{n-1})) * \varphi^{n-2}(x_{n-2})) * \dots) * \varphi(x_1)) * x_0 = c \qquad (4.13)$$

wobei $(Q, *)$ eine Quasigruppe und φ eine Permutation ist.

Zunächst zeigen wir, daß dies im wesentlichen dem vom Ecker und Poch vorgeschlagenen Ansatz entspricht. Sie definieren die Prüfziffer durch

$$x_0 := (\dots((x_n * \varphi(x_{n-1})) * \varphi^2(x_{n-2})) * \dots) * \varphi^{n-1}(x_1) \qquad (4.14)$$

und verzichten auf das Erkennen der Vertauschung $x_0 \leftrightarrow x_1$. Dies erscheint aufgrund der Tatsache, daß die Häufigkeit der Fehler mit wachsender Stellenzahl zunimmt, als wenig sinnvoll. Außerdem können wir auf die einzelnen Stellen x_i die Permutation φ^{-n} anwenden, ohne daß die Anti-Symmetrie-Eigenschaft der durch die Gleichung definierten n-Quasigruppe verlorengeht (Satz 16, Seite 67). Wir erhalten somit die Form 4.13, wobei es besser ist, die Prüfziffer implizit durch die genannte Gleichung zu bestimmen.

Für eine Quasigruppe reicht es nicht aus, daß φ eine anti-symmetrische Abbildung ist. Wir benötigen zusätzlich noch die Bedingung

$$(c * \varphi(x)) * y = (c * \varphi(y)) * x \quad \Rightarrow \quad x = y,$$

damit alle Nachbarvertauschungen erkannt werden. Wir nennen eine Permutation die anti-symmetrisch ist und zusätzlich diese Bedingung erfüllt *total anti-symmetrisch*. In einer Gruppe sind anti-symmetrische Abbildungen auch total anti-symmetrisch, für Quasigruppen gilt dies aber i.allg. nicht.

4.10.1 Konstruktion

Total anti-symmetrische Abbildungen einer Quasigruppen $(Q, *)$ können ganz ähnlich wie die anti-symmetrischen Abbildungen einer Gruppe konstruiert werden. In einer Quasigruppe haben wir allerdings im allgemeinen kein inverses Element. Dieses Problem können wir aber leicht lösen, indem wir die Parastrophien $(Q, /)$ und $(Q, \backslash)$ betrachten. Dann sind die Implikationen

$$\begin{aligned} \varphi(x) * y = \varphi(y) * x \quad &\Rightarrow \quad x = y \\ (c * \varphi(x)) * y = (c * \varphi(y)) * x \quad &\Rightarrow \quad x = y \end{aligned}$$

äquivalent zu

$$\begin{aligned} \varphi(x) = (\varphi(y) * x)\backslash y \quad &\Rightarrow \quad x = y \\ \varphi(x) = c/(((c * \varphi(y)) * x)\backslash y) \quad &\Rightarrow \quad x = y. \end{aligned}$$

Damit erhalten wir einen Algorithmus, der die total anti-symmetrischen Abbildungen einer Quasigruppe konstruiert, indem wir statt des Elements $m_{k,j*k*i^{-1}}$ bei Gruppen, das Element $m_{k,(j*k)\backslash i}$ und für alle $c \in Q$ die Elemente $m_{k,c/(((c*j)*k)\backslash i)}$ streichen (vgl. Seite 39). Mit diesem Algorithmus können wir sehr effektiv die total anti-symmetrischen Abbildungen einer vorgegebenen Quasigruppe konstruieren. Wir haben nun für verschiedene Quasigruppen die total anti-symmetrischen Abbildungen bestimmt und deren Erkennungsquote der anderen Fehlerarten untersucht. Dabei fanden wir eine Quasigruppe, die eine bessere Fehlererkennung bietet als die Diedergruppe.

Zum Vergleich geben wir zunächst die Erkennungsquote des von Ecker und Poch definierten „Shift-Code" an. Sie benutzen die im Abschnitt 4.9.1 (Seite 97) definierten Quasigruppen mit der Prüfgleichung 4.14 und der Permutation $\varphi(x) := x + 1$. Damit erzielten sie die folgenden Fehlererkennungsraten für die Quasigruppe $*_1$ der Ordnung 10: Sprungtranspositionen (Spr.): 84,89%, Zwillingsfehler (Zw.): 71,11%, Sprungzwillingsfehler (SprZw.): 87,67% und phonetische Fehler (Ph.): 76,19%.

Bei der zweiten angegebenen Quasigruppe stellten wir fest, daß diese zusammen mit φ nicht alle Nachbarvertauschungen erkennt. Es gilt für $2n = 4k + 2$, $k > 1$:

$$0 *_2 \varphi(3k+2) = 0 *_2 (3k+3) = 2k + 1 + ((3k+3) \bmod 2k+1) = 3k+3$$

und

$$(3k+2) *_2 \varphi(0) = (3k+2) *_2 1 = 2k + 1 + ((1 - 3k - 2 + 1) \bmod 2k+1) = 3k+3.$$

Folglich ist $(3k+2) *_2 \varphi(0) = 0 *_2 \varphi(3k+2)$.

Das Ergebnis von ECKER und POCH [9, Seite 299] ist für diese Quasigruppe daher falsch.

Bei der Diedergruppe fanden wir total anti-symmetrische Permutationen, die eine deutlich höhere Fehlererkennung bieten als der Shift-Code von ECKER und POCH.

Ordnung	Permutation	Spr.	Zw.	SprZw.	Ph.
6	[034152]	82,22%	86,67%	82,22%	80,00%
	[305214]	82,22%	86,67%	82,22%	100,00%
8	[07526431]	89,29%	100,00%	89,29%	91,43%
	[43571602]	92,86%	92,86%	92,86%	97,14%
10	[0458613297]	92,00%	95,56%	92,00%	90,48%
	[0542978136]	92,00%	91,11%	92,00%	100,00%
	[7046913258]	94,22%	95,56%	94,22%	96,83%

Bei der Suche nach anderen Quasigruppen, die eine bessere Fehlererkennung haben als die Diedergruppe, fanden wir die Quasigruppe $(\mathbb{Z}_n, *)$, n gerade, definiert durch

$$x * y = \begin{cases} (x+y) \bmod n & \text{falls } x \text{ gerade} \\ (x-y-2) \bmod n & \text{falls } x \text{ ungerade} \end{cases}$$

(vgl. Abschnitt 4.9.1) und die folgenden total anti-symmetrischen Abbildungen:

Ordnung	Permutation	Spr.	Zw.	SprZw.	Ph.
6	[013425]	82,22%	86,67%	82,22%	100,00%
	[014352]	82,22%	86,67%	82,22%	100,00%
8	[12053467]	92,86%	100,00%	92,86%	94,29%
	[01526374]	92,86%	100,00%	92,86%	100,00%
10	[0137268459]	92,00%	95,56%	92,00%	100,00%
	[0147389625]	92,00%	95,56%	92,00%	100,00%
	[2096813574]	94,22%	95,56%	94,22%	96,83%

Mit der Permutation [2096813574] erreichen wir eine Fehlererkennung von 99,89% aller nicht zufälligen Fehler (einschließlich der Einzelfehler und der Nachbarvertauschungen, die zu 100% erkannt werden). Die Permutation [0147389625] bietet eine Fehlererkennung von 99,87%. Sie hat den Vorteil, daß die 0 fixiert wird, womit führende Nullen die Prüfziffer nicht verändern und Formatfehler erkannt werden können. Im Vergleich zur Permutation [0542978136] der Diedergruppe erkennt dieses Prüfziffersystem mehr Fehler und ist diesem daher vorzuziehen.

Schlußbemerkung

Wir haben gesehen, daß das Problem, ein Prüfziffersystem zur Basis 10 zu bestimmen, welches alle Sprung-/Zwillingsfehler, alle Sprungtranspositionen und alle phonetischen Fehler erkennt, keine naheliegende Lösung besitzt. Die Frage, ob es überhaupt ein solches Prüfziffersystem gibt, bleibt offen. Die Existenz zu widerlegen, erscheint allerdings sehr schwer, da ein entsprechender Beweis auf den speziellen Eigenschaften der Zahl 10 beruhen muß, denn zur Basis 11 existiert ein entsprechendes Prüfziffersystem. Trotzdem können wir mit dem im letzten Abschnitt angegebenen Prüfziffersystem 99,89% aller nicht zufälligen Fehler erkennen und somit eine sehr hohe Fehlererkennung gewährleisten.

Literaturverzeichnis

[1] J. Aczél, V.D. Belousov, M. Hosszú. *Generalized associativity and bisymmetry on quasigroups*. Acta Math. Acad. Sci. Hungar 11 (1960), 127-136.

[2] P. Batemann. *Complete mappings of infinite groups*. Amer. Math. Monthly 57 (1950), 621-622.

[3] J. A. Beachy, W. D. Blair. *Abstract Algebra, Second Edition*. Waveland Press, Illinois 1996.

[4] V.D. Belousov. *Extensions of quasigroups*. Bull. Akad. Stiince RSS Moldoven No. 8 (1967), 3-24. (Russisch)

[5] G.B. Belyavskaya, A.Kh. Tabarov. *Characteristic of linear and alinear quasigroups*. Diskretn. Mat. 4, No.2 (1992), 142-147. (Russisch)

[6] O. Chein, H.O. Pflugfelder, J.D.H. Smith. *Quasigroups and Loops, Theory and Applications*. Sigma Series in Pure Mathematics, Volume 8 (1990), Heldermann Verlag Berlin.

[7] J. Conway, R. Curtis, S. Norton, R. Parker, R. Wilson. *Atlas of Finit Groups*. Oxford 1985.

[8] J. Dénes, A.D. Keedwell. *Latin Squares and theire Applications*. New York: Academic Press (1974).

[9] A. Ecker, G. Poch. *Check Character Systems*. Computing 37 (1986), 277-301.

[10] J. A. Gallian, M. Mullin. *Groups with Anti-symmetric Mappings*. Archive der Math. 65 (1995), 273-280.

[11] D. Gorenstein. *Classifying the finit simple groups*. Bull. Amer. Math. Soc. 14 (1986), 1-98.

[12] H.P. Gumm. *A New Class of Check-Digit Methods for Arbitrary Number Systems*. IEEE Tran. Inf. Th. 31 (1985), 102-105.

[13] H.P. Gumm. *Encoding of Numbers to Detect Typing Errors.* Inter. J. Applied Eng. Ed. 2 (1986), 61-65.

[14] M. Hall, L.J. Paige. *Complete mappings of finite groups.* Pacific J. Math. 5 (1955), 541-549.

[15] H.B. Mann. *The construction of orthogonal latin squares.* Ann. Math. Statistics 13 (1942), 418-423.

[16] B. D. McKay, E. Rogoyski. *Latin Squares of Order 10.* The Electronic Journal of Combinatorics 2 (1995) #N3.

[17] K. Meyberg. *Algebra, Teil 1.* Carl Hanser Verlag München Wien 1980.

[18] L.J. Paige. *A note on finite abelian groups.* Bull. Amer. Math. Soc. 53 (1947), 590-593.

[19] L.J. Paige. *Complete mappings of finite groups.* Pacific J. Math. 1 (1951), 111-116.

[20] R. Schauffler. *Die Assoziativität im Ganzen, besonders bei Quasigruppen.* Math. Z. 67 (1957), 428-435.

[21] R.-H. Schulz. *Codierungstheorie. Eine Einführung.* Vieweg V. Braunschweig/Wiesbaden 1991.

[22] R.-H. Schulz. *A note on Check character Systems using Latin squares.* Discr. Math. 97 (1991) 371-375.

[23] H. Siemon. *Anwendungen der elementaren Gruppentheorie in Zahlentheorie und Kombinatorik.* Stuttgart: Klett-Verlag 1981.

[24] J. Širáň, M. Škoviera. *Groups with sign structure and theire antiautomorphisms.* Discr. Math. 108 (1992), 189-202.

[25] N. J. A. Sloane, S. Plouffe. *The Encyclopedia of Integer Sequences.* Academic Press, San Diego, 1995.

[26] S. K. Stein. *On the foundations of quasigroups.* Trans. Amer. Math. Soc. 85 (1957), 228-256.

[27] J. Verhoeff. *Error detecting decimal codes.* Math. Centre Tracts 29, Amsterdam 1969.

[28] Steven J. Winters. *Error Detecting Schemes Using Dihedral Groups.* UMAP Journal 11 (1990), 299-308.

Erklärung

Hiermit versichere ich, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Marburg, den 6. März 1998

A Combinatorial Miscellany

Anders Björner[1]
Matematiska Institutionen
Kungl. Tekniska Högskolan
S-100 44 Stockholm
SWEDEN

Richard P. Stanley[2]
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139
U.S.A.

May 11, 1998

1 Introduction.

A recent newcomer to the the center stage of modern mathematics is the area called *combinatorics*. Although combinatorial mathematics has been pursued since time immemorial, and at a reasonable scientific level at least since Leonhard Euler (1707–1783), the subject has come into its own only in

[1]Partially supported by the Mathematical Sciences Research Institute (Berkeley, CA) and by the Göran Gustafsson Foundation for Research in Natural Sciences and Medicine.
[2]Partially supported by NSF grant #DMS-9500714.

the last few decades. The reasons for the spectacular growth of combinatorics come both from within mathematics itself and from the outside.

Beginning with the outside influences, it can be said that the recent development of combinatorics is somewhat of a cinderella story. It used to be looked down on by "mainstream" mathematicians as being somehow less respectable than other areas, in spite of many services rendered to both pure and applied mathematics. Then along came the prince of computer science with its many mathematical problems and needs — and it was combinatorics that best fitted the glass slipper held out.

The developments within mathematics that have contributed to the current strong standing of combinatorics are more difficult to pinpoint. One is that, after an era where the fashion in mathematics was to seek generality and abstraction, there is now much appreciation of and emphasis on the concrete and "hard" problems. Another is that it has been gradually more and more realized that combinatorics has all sorts of deep connections with the mainstream areas of mathematics, such as (to name the most important ones) algebra, geometry, probability and topology.

Our aim with this article is to give the reader some answers to the questions "What is combinatorics, and what is it good for?" We will do that not by attempting any kind of general survey, but by describing a few selected problems and results in some detail. We want to bring you both some examples of problems from "pure" combinatorics, some examples illustrating its interactions with other parts of mathematics, and a few glimpses of its use for computer science. Fortunately, the problems and results of combinatorics are usually quite easy to state and explain, even to the layman. Its accessibility is one of its many appealing aspects. For instance, most popular mathematical puzzles and games, such as Rubik's cube and jigsaw puzzles, are essentially problems in combinatorics.

To achieve our stated purpose it has been necessary to concentrate on a few topics, leaving many of the specialities within combinatorics without mention. Naturally, the choice will reflect our own interests. The suggestions for further reading point to some more general accounts that can help remedy this shortcoming.

With some simplification, combinatorics can be said to be the mathematics of the finite. One of the most basic properties of a finite collection of objects is its number of elements. For instance, take words formed from the letters a, b, and c, using each letter exactly once. There are six such words:

$$abc, \quad acb, \quad bac, \quad bca, \quad cab, \quad cba.$$

Now, say that we have n distinct letters. How many words can be formed? The answer is $n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1$, because the first letter can be chosen in n ways, then the second one in $n-1$ ways (since the letter already chosen as the first letter is no longer available), the third one in $n-2$ ways, and so on. Furthermore, the total number must be the product of the number of individual choices.

The number of words that can be formed with n letters is an example of an *enumerative* problem. Enumeration is one of the most basic and important aspects of combinatorics. In many branches of mathematics and its applications you need to know the number of ways of doing something. One of the classical problems of enumerative combinatorics is to count partitions of various kinds, meaning the number of ways to break an object into smaller objects of the same kind. The study of partition enumeration was begun by Euler and is very active to this day. We will exposit some parts of this theory. All along the way there are interesting connections with algebra, but unfortunately these are too sophisticated to be given a detailed treatment here. We also illustrate (in Section 11) the relevance of partitions to applied problems.

Another, more recent, topic within enumeration is to count the number of *tilings*. These are partitions of a geometric region into smaller regions of some specified kinds. We will give some glimpses of recent progress in this area. The mathematical roots are in this case mainly from statistical mechanics.

Combinatorics is used in many ways in computer science, for instance for the construction and analysis of various algorithms. (Remark: *algorithms* are the logically structured systems of commands that instruct computers how to perform prescribed tasks.) Of this young but already huge and rapidly growing area we will give here but the smallest glimpse, namely a couple of examples from complexity theory. This is the part of theoretical computer

science that concerns itself with questions about computer calculations of the type "How hard is it?", "How much time will it take?" Proving that you cannot do better than what presently known methods allow is often the hardest part, and the part where the most mathematics is needed. Our examples are of this kind.

To illustrate the surprising connections that exist between combinatorics and seemingly unrelated parts of mathematics we have chosen the links with topology. This is an area which on first acquaintance seems far removed from combinatorics, having to do with very general infinite spaces. Nevertheless, the tools of algebraic topology have proven to be of use for solving some problems from combinatorics and theoretical computer science. Again, the theme of enumeration in its various forms pervades some of this border territory.

Our final topic is a glimpse of progress made in the combinatorial study of convex polytopes. In three dimensions these are the decorative solid bodies with flat polygon sides (such as pyramids, cubes and geodesic domes) that have charmed and intrigued mathematicians and laymen alike since antiquity. In higher dimensions they can be perceived only via mathematical tools, but they are just as beautiful and fascinating. Of this huge subject we discuss the question of laws governing the numbers of faces of various dimensions on the boundary of a polytope.

Understanding this article should for the most part require hardly any knowledge of mathematics beyond high-school algebra. Only some details in the boxes and in the last few sections (having to do with topology) are a bit more demanding.

2 Partitions.

A fundamental concept in combinatorics is that of a partition. In general, a partition of an object is a way of breaking it up into smaller objects. We will be concerned here with partitions of *positive integers* (positive whole numbers). Later on we will encounter also other kinds of partitions. The

subject of partitions has a long history going back to Gottfried Wilhelm von Leibniz (1646–1716) and Euler, and has been found to have unexpected connections with a number of other subjects.

A *partition* of a positive integer n is a way of writing n as a sum of positive integers, ignoring the order of the summands. For instance, $3+4+2+1+1+4$ represents a partition of 15, and $4+4+3+2+1+1$ represents the same partition. We allow a partition to have only one part (summand), so that 5 is a partition of 5. There are in fact seven partitions of 5, given by

$$\begin{aligned}
&5\\
&4+1\\
&3+2\\
&3+1+1\\
&2+2+1\\
&2+1+1+1\\
&1+1+1+1+1.
\end{aligned}$$

We denote the number of partitions of n by $p(n)$, so for instance $p(5) = 7$. By convention we set $p(0) = 1$, and similarly for related partition functions discussed below. The problem of evaluating $p(n)$ has a long history. There is no simple formula in general for $p(n)$, but there are remarkable and quite sophisticated methods to compute $p(n)$ for "reasonable" values of n. For instance, as long ago as 1938 Derrick Henry Lehmer (1905–1991) computed $p(14,031)$ (a number with 127 digits!), and nowadays a computer would have no trouble computing $p(10^{12})$, a number with 1,113,996 digits. It is also possible to codify all the numbers $p(n)$ into a single object known as a *generating function*. A generating function (in the variable x) is an expression of the form

$$F(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots,$$

where the coefficients $a_0, a_1, \ldots$ are numbers. We call a_n the *coefficient* of x^n, and call a_0 the *constant term*. (The notation x^0 next to a_0 is suppressed.) The generating function $F(x)$ differs from a polynomial in x in that it can have infinitely many terms. We regard x as a formal symbol, and do not think of it as standing for some unknown quantity. Thus the generating function $F(x)$ is just a way to represent the sequence $a_0, a_1, \ldots$.

It is natural to ask what advantage is gained in representing a sequence in such a way. The answer is that generating functions can be manipulated in

various ways that are often useful for combinatorial problems. For instance, letting $G(x) = b_0 + b_1x + b_2x^2 + \cdots$, we can add $F(x)$ and $G(x)$ by the rule

$$F(x) + G(x) = (a_0 + b_0) + (a_1 + b_1)x + (a_2 + b_2)x^2 + \cdots.$$

In other words, we simply add the coefficients, just as we would expect from the ordinary rules of algebra. Similarly we can form the product $F(x)G(x)$ using the ordinary rules of algebra, in particular the law of exponents $x^i x^j = x^{i+j}$. To perform this multiplication, we pick a term $a_i x^i$ from $F(x)$ and a term $b_j x^j$ from $G(x)$ and multiply them to get $a_i b_j x^{i+j}$. We then add together all such terms. For instance, the term in the product involving x^4 will be

$$a_0 \cdot b_4x^4 + a_1x \cdot b_3x^3 + a_2x^2 \cdot b_2x^2 + a_3x^3 \cdot b_1x + a_4x^4 \cdot b_0$$

$$= (a_0b_4 + a_1b_3 + a_2b_2 + a_3b_1 + a_4b_0)x^4.$$

In general, the coefficient of x^n in $F(x)G(x)$ will be

$$a_0b_n + a_1b_{n-1} + a_2b_{n-2} + \cdots + a_{n-1}b_1 + a_nb_0.$$

Consider for instance the product of $F(x) = 1 + x + x^2 + x^3 + \cdots$ with $G(x) = 1 - x$. The constant term is just $a_0b_0 = 1 \cdot 1 = 1$. If $n > 1$ then the coefficient of x^n is $a_nb_0 + a_{n-1}b_1 = 1 - 1 = 0$ (since $b_i = 0$ for $i > 1$, so we have only two nonzero terms). Hence

$$(1 + x + x^2 + x^3 + \cdots)(1 - x) = 1.$$

For this reason we write

$$\frac{1}{1 - x} = 1 + x + x^2 + x^3 + \cdots.$$

Some readers will recognize this formula as the sum of an infinite geometric series, though here the formula is "formal," that is, x is regarded as just a symbol and there is no question of convergence. Similarly, for any $k \geq 1$ we get

$$\frac{1}{1 - x^k} = 1 + x^k + x^{2k} + x^{3k} + \cdots. \tag{1}$$

Now let $P(x)$ denote the (infinite) product

$$P(x) = \frac{1}{1 - x} \cdot \frac{1}{1 - x^2} \cdot \frac{1}{1 - x^3} \cdots.$$

We may also write this product as

$$P(x) = \frac{1}{(1-x)(1-x^2)(1-x^3)\cdots}. \tag{2}$$

Can any sense be made of this product? According to our previous discussion, we can rewrite the right-hand side of equation (2) as

$$P(x) = (1 + x + x^2 + \cdots)(1 + x^2 + x^4 + \cdots)(1 + x^3 + x^6 + \cdots)\cdots.$$

To expand this product as a sum of individual terms, we must pick a term x^{m_1} from the first factor, a term x^{2m_2} from the second, a term x^{3m_3} from the third, etc., multiply together all these terms, and then add all such products together. In order not to obtain an infinite (and therefore meaningless) exponent of x, it is necessary to stipulate that when we pick the terms $x^{m_1}, x^{2m_2}, x^{3m_3}, \ldots$, only finitely many of these term are not equal to 1. (Equivalently, only finitely many of the m_i are not equal to 0.) We then obtain a single term $x^{m_1+2m_2+3m_3+\cdots}$, where the exponent $m_1+2m_2+3m_3+\cdots$ is finite. The coefficient of x^n in $P(x)$ will then be the number of ways to write n in the form $m_1 + 2m_2 + 3m_3 + \cdots$ for nonnegative integers $m_1, m_2, m_3, \ldots$. But writing n in this form is the same as writing n as a sum of m_1 1's, m_2 2's, m_3 3's, etc. Such a way of writing n is just a partition of n. For instance, the partition $5 + 5 + 5 + 4 + 2 + 2 + 2 + 2 + 1 + 1 + 1$ of 30 corresponds to choosing $m_1 = 3, m_2 = 4, m_4 = 1, m_5 = 3$, and all other $m_i = 0$. It follows that the coefficient of x^n in $P(x)$ is just $p(n)$, the number of partitions of n, so we obtain the famous formula of Euler

$$p(0) + p(1)x + p(2)x^2 + \cdots = \frac{1}{(1-x)(1-x^2)(1-x^3)\cdots}. \tag{3}$$

Although equation (3) is very elegant, one may ask whether it is of any use. Can it be used to obtain interesting information about the numbers $p(n)$? To answer that, let us show how simple manipulation of generating functions (due to Euler) gives a surprising connection between two types of partitions. Let $r(n)$ be the number of partitions of n into *odd* parts. For instance, $r(7) = 5$, the relevant partitions being

$$7 = 5 + 1 + 1 = 3 + 3 + 1 = 3 + 1 + 1 + 1 + 1 = 1 + 1 + 1 + 1 + 1 + 1 + 1.$$

Let

$$R(x) = r(0) + r(1)x + r(2)x^2 + r(3)x^3 + \cdots.$$

Exactly as equation (3) was obtained we get

$$R(x) = \frac{1}{(1-x)(1-x^3)(1-x^5)(1-x^7)\cdots}. \tag{4}$$

Similarly, let $q(n)$ be the number of partitions of n into *distinct* parts, that is, no integer can occur more than once as a part. For instance, $q(7) = 5$, the relevant partitions being

$$7 = 6+1 = 5+2 = 4+3 = 4+2+1.$$

Note that $r(7) = q(7)$. In order to explain this "coincidence," let

$$Q(x) = q(0) + q(1)x + q(2)x^2 + q(3)x^3 + \cdots.$$

The reader who understands the derivation of equation (3) will have no trouble seeing that

$$Q(x) = (1+x)(1+x^2)(1+x^3)\cdots. \tag{5}$$

Now we come to the ingenious trick of Euler. Note that by ordinary "high school algebra," we have

$$1 + x^n = \frac{1-x^{2n}}{1-x^n}.$$

Thus from equation (5) we obtain

$$\begin{aligned} Q(x) &= \frac{1-x^2}{1-x} \cdot \frac{1-x^4}{1-x^2} \cdot \frac{1-x^6}{1-x^3} \cdots \\ &= \frac{(1-x^2)(1-x^4)(1-x^6)(1-x^8)\cdots}{(1-x)(1-x^2)(1-x^3)(1-x^4)\cdots}. \end{aligned} \tag{6}$$

When we cancel the factors $1-x^{2i}$ from both the numerator and denominator, we are left with

$$Q(x) = \frac{1}{(1-x)(1-x^3)(1-x^5)\cdots},$$

which is just the product formula (4) for $R(x)$. This means that $Q(x) = R(x)$. Thus the coefficients of $Q(x)$ and $R(x)$ are the same, so we have proved that $q(n) = r(n)$ for all n. In other words, *for every n the number of partitions of n into distinct parts equals the number of partitions of n into odd parts.*

The above argument shows the usefulness of working with generating functions. Many similar generating function techniques have been developed that make generating functions a fundamental tool of enumerative combinatorics.

Once we obtain a formula such as $q(n) = r(n)$ by an indirect means like generating functions, it is natural to ask whether there might be a simpler proof. For the problem at hand, we would like to correspond to each partition of n into distinct parts a partition of n into odd parts, such that every partition of n into odd parts is associated with exactly one partition of n into distinct parts, and conversely every partition of n into distinct parts is associated with exactly one partition of n into odd parts. In other words, we want a *one-to-one correspondence* or *bijection* between the partitions of n into odd parts and the partitions of n into distinct parts. Such a bijection would yield a *bijective proof* of the formula $q(n) = r(n)$. Exhibiting a bijection between two different (finite) sets is considered the most elegant and natural way to show that they have the same number of elements. Such bijective proofs can involve considerable ingenuity, while the method of generating functions often yields a more mechanical proof technique.

We now would like to give a bijective proof of Euler's formula $q(n) = r(n)$. Several such proofs are known; we give the perhaps simplest of these, due to James Joseph Sylvester (1814–1897). It is based on the fact that every positive integer n can be uniquely written as a sum of distinct powers of two — this is simply the binary expansion of n. For instance, $10000 = 2^{13} + 2^{10} + 2^9 + 2^8 + 2^4$. Suppose we are given a partition into odd parts, such as

$$202 = 19 + 19 + 19 + 11 + 11 + 11 + 11 + 9 + 7 + 7 + 7 + 5$$
$$+5 + 5 + 5 + 5 + 5 + 5 + 5 + 5 + 5 + 5 + 5 + 5 + 1 + 1 + 1 + 1 + 1 + 1.$$

We can rewrite this partition as

$$3 \cdot 19 + 4 \cdot 11 + 1 \cdot 9 + 3 \cdot 7 + 13 \cdot 5 + 6 \cdot 1,$$

where each part is multiplied by the number of times it appears. This is just the expression $m_1 + 2m_2 + 3m_3 + \cdots$ for a partition discussed above. Now write each of the numbers m_i as a sum of distinct powers of 2. For the above example, we get

$$202 = (2 + 1) \cdot 19 + 4 \cdot 11 + 1 \cdot 9 + (2 + 1) \cdot 7 + (8 + 4 + 1) \cdot 5 + (4 + 2) \cdot 1.$$

Expand each product into a sum:

$$202 = (38 + 19) + 44 + 9 + (14 + 7) + (40 + 20 + 5) + (4 + 2). \qquad (7)$$

We have produced a partition of the same number n with distinct parts. That the parts are distinct is a consequence of the fact that every integer n can be uniquely written as the product of an odd number and a power of 2 (keep on dividing n by 2 until an odd number remains). Moreover, the whole procedure can be reversed. That is, given a partition into distinct parts such as

$$202 = 44 + 40 + 38 + 20 + 19 + 14 + 9 + 7 + 5 + 4 + 2,$$

group the terms together according to their largest odd divisor. For instance, 40, 20, and 5 have the largest odd divisor 5, so we group them together. We thus recover the grouping (7). We can now factor the largest odd divisor d out of each group, and what remains is the number of times d appears as a part. Thus we have recovered the original partition. This reasoning shows that we have indeed produced a bijection between partitions of n into odd parts and partitions of n into distinct parts. It provides a "natural" explanation of the fact that $q(n) = r(n)$, unlike the generating function proof which depended on a miraculous trick.

The subject of partitions is replete with results similar to Euler's, in which two sets of partitions turn out to have the same number of elements. The most famous of these results is called the *Rogers-Ramanujan identities*, after Leonard James Rogers (1862–1933) and Srinivasa Aiyangar Ramanujan (1887–1920), who proved these identities in the form of an identity between generating functions. It was Percy Alexander MacMahon (1854–1929) who interpreted them combinatorially as follows.

First Rogers-Ramanujan Identity. *Let $f(n)$ be the number of partitions of n whose parts differ by at least 2. For instance, $f(13) = 10$, the relevant partitions being*

$$13 = 12 + 1 = 11 + 2 = 10 + 3 = 9 + 4 = 8 + 5 = 9 + 3 + 1$$

$$= 8 + 4 + 1 = 7 + 5 + 1 = 7 + 4 + 2.$$

Similarly, let $g(n)$ be the number of partitions of n whose parts are of the form $5k + 1$ or $5k + 4$ (i.e., leave a remainder of 1 or 4 upon division by 5).

For instance, $g(13) = 10$*:*

$$11+1+1 = 9+4 = 9+1+1+1+1 = 6+6+1 = 6+4+1+1+1$$
$$= 6+1+1+1+1+1+1+1 = 4+4+4+1 = 4+4+1+1+1+1+1$$
$$= 4+1+1+1+1+1+1+1+1 = 1+1+1+1+1+1+1+1+1+1+1+1+1.$$

Then $f(n) = g(n)$ *for every* n*.*

Second Rogers-Ramanujan Identity. *Let* $u(n)$ *be the number of partitions of* n *whose parts differ by at least 2 and such that 1 is not a part. For instance,* $u(13) = 6$*, the relevant partitions being*

$$13 = 11+2 = 10+3 = 9+4 = 8+5 = 7+4+2.$$

Similarly, let $v(n)$ *be the number of partitions of* n *whose parts are of the form* $5k+2$ *or* $5k+3$ *(i.e., leave a remainder of 2 or 3 upon division by 5). For instance,* $v(13) = 6$*:*

$$13 = 8+3+2 = 7+3+3 = 7+2+2+2 = 3+3+3+2+2 = 3+2+2+2+2+2.$$

Then $u(n) = v(n)$ *for every* n*.*

The Rogers-Ramanujan identities have been given many proofs, but none of them is really easy. The important role played by the number 5 seems particularly mysterious. For a long time it was an open problem to find a bijective proof of the Rogers-Ramanujan identities, but such a proof was finally given in 1980 by Adriano Mario Garsia (b. 1928) and Stephen Carl Milne (b. 1949). However, their proof is very complicated, and it would still be of great interest to find a simple, conceptual bijective proof.

The Rogers-Ramanujan identities and related identities are not just number-theoretic curiosities. They have arisen completely independently in several seemingly unrelated areas. To give just one example, a famous open problem in statistical mechanics, known as the *hard hexagon model*, was solved in 1980 by Rodney James Baxter (b. 1940) using the Rogers-Ramanujan identities.

The subject of partition identities has received so much attention since Euler that one would not expect a whole new class of relatively simple identities to have remain undiscovered until recently. However, just such a class

of identities was found by Mireille Bousquet-Mélou (b. 1967) and Kimmo Eriksson (b. 1967) beginning in 1996. We will state one of the simplest of their identities to give the reader the striking flavor of their results.

The *Lucas numbers* L_n are defined by the conditions $L_1 = 1$, $L_2 = 3$, and $L_{n+1} = L_n + L_{n-1}$ for $n \geq 2$. Thus $L_3 = 4$, $L_4 = 7$, $L_5 = 11$, $L_6 = 18$, $L_7 = 29$, etc. Those familiar with Fibonacci numbers will see that the Lucas numbers satisfy the same recurrence as Fibonacci numbers, but with the initial conditions $L_1 = 1$ and $L_2 = 3$, rather that $F_1 = F_2 = 1$ for Fibonacci numbers. Let $f(n)$ be the number of partitions of n all of whose parts are Lucas numbers L_{2m+1} of odd index. For instance, we have $f(12) = 5$, corresponding to the partitions

$$\begin{array}{l} 1+1+1+1+1+1+1+1+1+1+1+1 \\ 4+1+1+1+1+1+1+1 \\ 4+4+1+1+1+1 \\ 4+4+4 \\ 11+1 \end{array} \quad .$$

Let $g(n)$ be the number of partitions of n into parts $a_1 \leq a_2 \leq \cdots \leq a_k$ such that $a_i/a_{i-1} > \frac{1}{2}(3+\sqrt{5}) = 2.618\cdots$ for all i. For instance, $g(12) = 5$, corresponding to the partitions

$$12, \quad 11+1, \quad 10+2, \quad 9+3, \quad \text{and} \quad 8+3+1.$$

Note that the number $\frac{1}{2}(3+\sqrt{5})$ used to define $g(n)$ is the square of the "golden ratio" $\frac{1}{2}(1+\sqrt{5})$.

The surprising result of Bousquet-Mélou and Eriksson is that $f(n) = g(n)$ for all n.

3 Plane partitions.

A partition such as $8+6+6+5+2+2+2+2+1+1$ may be regarded simply as a linear array of positive integers,

$$8\ 6\ 6\ 5\ 2\ 2\ 2\ 2\ 1\ 1$$

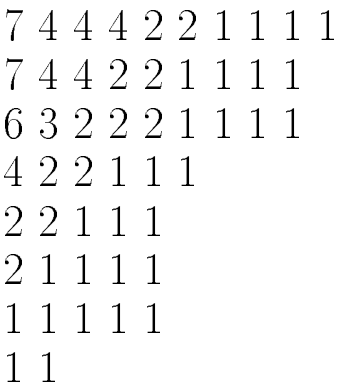
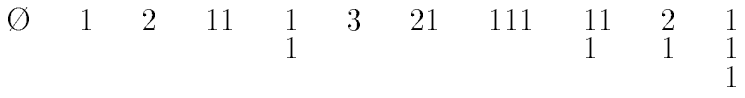


Figure 1: A plane partition

whose entries are *weakly decreasing*, i.e., each entry is greater than or equal to the one on its right. Viewed in this way, one can ask if there are interesting "multidimensional" generalizations of partitions, in which the parts do not lie on just a line, but rather on some higher dimensional object. The simplest generalization occurs when the parts lie in a plane. Rather than having the parts weakly decreasing in a single line, we now want the parts to be weakly decreasing in every row and column. More precisely, let λ be a partition with its parts $\lambda_1, \lambda_2, \ldots, \lambda_\ell$ written in weakly decreasing order, so $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\ell > 0$. We define a *plane partition* π of *shape* λ to be a left-justified array of positive integers (called the *parts* of π) such that (1) there are λ_i parts in the ith row, and (2) every row (read left-to-right) and column (read top-to-bottom) is weakly decreasing. An example of a plane partition is given in Figure 1.

We say that π is a plane partition *of* n if n is the sum of the parts of π. Thus the plane partition of Figure 1 is a plane partition of 100, of shape $(10, 9, 9, 6, 5, 5, 5, 2)$. It is clear what is meant by the *number of rows* and *number of columns* of π. For the example in Figure 1, the number of rows is 8 and the number of columns is 10. The plane partitions of integers up to 3 (including the empty set $\emptyset$, which is regarded as a plane partition of 0) are given by



.

Thus, for instance, there are six plane partitions of 3.

In 1912 MacMahon began a study of the theory of plane partitions. MacMahon was a mathematician well ahead of his time. He worked in virtual isolation on a variety of topics within enumerative combinatorics that did not become fashionable until many years later. A highlight of MacMahon's work was a simple generating function for the number of plane partitions of n. More precisely, let $pp(n)$ denote the number of plane partitions of n, so that $pp(0) = 1,\ pp(1) = 1,\ pp(2) = 3,\ pp(3) = 6,\ pp(4) = 13$, etc.

MacMahon's Theorem.

$$pp(0) + pp(1)x + pp(2)x^2 + pp(3)x^3 + \cdots$$
$$= \frac{1}{(1-x)(1-x^2)^2(1-x^3)^3(1-x^4)^4\cdots}. \tag{8}$$

Unlike Euler's formula (3) for the generating function for the number $p(n)$ of ordinary partitions of n, MacMahon's remarkable formula is by no means easy to prove.

MacMahon's proof was an intricate induction argument involving manipulations of determinants. Only much later a bijective proof was found by Edward Anton Bender (b. 1942) and Donald Ervin Knuth (b. 1938). Their proof was based on the *Schensted correspondence*, a central result in enumerative combinatorics and its connections with the branch of mathematics known as *representation theory*. This correspondence was first stated by Gilbert de Beauregard Robinson (1906–1992) in a rather vague form in 1938 (with some assistance from Dudley Ernest Littlewood (1903–1979)), and later more explicitly by Craige Eugene Schensted (b. 1927 or 1928) in 1961. Schensted's motivation for looking at this correspondence is discussed in Section 5. The version of Schensted's correspondence used here is due to Knuth.

We now give a brief account of the proof of Bender and Knuth. Using equation (1), the product on the right-hand side of (8) may be written

$$\frac{1}{(1-x)(1-x^2)^2(1-x^3)^3(1-x^4)^4\cdots} = (1 + x + x^2 + \cdots)(1 + x^2 + x^4 + \cdots)$$

$$(1+x^2+x^4+\cdots)(1+x^3+x^6+\cdots)(1+x^3+x^6+\cdots)(1+x^3+x^6+\cdots)\cdots. \quad (9)$$

In general, there will be k factors of the form $1 + x^k + x^{2k} + x^{3k} + \cdots$. We must pick a term out of each factor (with only finitely many terms not equal to 1) and multiply them together to get a term x^n of the product. A bijective proof of (8) therefore consists of associating a plane partition of n with each choice of terms from the factors $1 + x^k + x^{2k} + \cdots$, such that the product of these terms is x^n.

Our first step is to encode a choice of terms from each factor by an array of numbers called a *two-line array*. A typical two-line array A looks like

$$A = \begin{array}{c} 3\ 3\ 3\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 1 \\ 3\ 1\ 1\ 2\ 2\ 2\ 1\ 1\ 4\ 4\ 3\ 3\ 3 \end{array}. \quad (10)$$

The first line is a (finite) weakly decreasing sequence of positive integers. The second line consists of a positive integer below each entry in the first line, such that the integers in the second line appearing below equal integers in the first line are in weakly decreasing order. For instance, for the two-line array A above, the integers appearing below the 2's of the first line are 2 2 2 1 1 (in that order). Such a two-line array encodes a choice of terms from the factors of the product (9) as follows. Let a_{ij} be the number of columns $\begin{smallmatrix} i \\ j \end{smallmatrix}$ of A. For instance (always referring to the two-line array (10)), $a_{33} = 1,\ a_{31} = 2,\ a_{13} = 3,\ a_{23} = 0$. Given a_{ij}, let $k = i + j - 1$. Then choose the term $x^{a_{ij} \cdot k}$ from the ith factor of (9) of the form $1 + x^k + x^{2k} + \cdots$. For instance, since $a_{33} = 1$ we have $k = 5$ and choose the term $x^{1 \cdot 5} = x^5$ from the third factor of the form $1 + x^5 + x^{10} + \cdots$. Since $a_{31} = 2$ we have $k = 3$ and choose the term $x^{2 \cdot 3} = x^6$ from the third factor of the form $1 + x^3 + x^6 + \cdots$, etc. In this way we obtain a one-to-one correspondence between a choice of terms from each factor of the product (9) (with only finitely terms not equal to 1) and two-line arrays A.

We now describe the part of the Bender-Knuth bijection which is the Schensted correspondence. It will be described as an algorithm that we call the *Schensted algorithm*. We will insert the numbers in each line of the two-line array A into a successively evolving plane partition, yielding in fact a pair of plane partitions. These plane partitions will have the special property of being *column-strict*, that is, the (nonzero) entries are *strictly* decreasing in each column. Thus after we have inserted the first i numbers of the first

and second lines of A, we will have a pair P_i and Q_i of column-strict plane partitions. We insert the numbers of the second line of A successively from left-to-right by the following rule. Assuming that we have inserted the first $i-1$ numbers, yielding P_{i-1} and Q_{i-1}, we insert the ith number a of the second row of A into P_{i-1}, by putting it as far to the right as possible in the first row of P_{i-1} so that this row remains weakly decreasing. In doing so, it may displace (or *bump*) another number b already in the first row. Then insert b into the second row according to the same rule, that is, as far to the right as possible so that the second row remains weakly decreasing. Then b may bump a number c into the third row, etc. Continue this "bumping procedure" until finally a number is inserted at the end of the row, thereby not bumping another number. This yields the column-strict plane partition P_i. (It takes a little work, which we omit, to show that P_i is indeed column-strict.) Now insert the ith number of the first row of A (that is, the number just above a in A) into Q_{i-1} to form Q_i, by placing it so that P_i and Q_i have the same *shape*, that is, the same number of elements in each row. If A has m columns, then the process stops after producing P_m and Q_m, which we denote simply as P and Q.

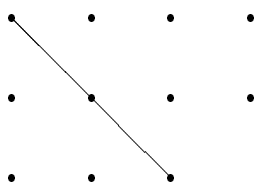
Example. Figure 2 illustrates the bumping procedure with the two-line array A of equation (10). For instance, to obtain P_{10} from P_9 we insert 4 into the first row of P_9. The 4 is inserted into the second column and bumps the 2 into the second row. The 2 is also inserted into the second column and bumps the 1 into the third row. The 1 is placed at the end of the third row. To obtain Q_{10} from Q_9 we must place 1 so that P_{10} and Q_{10} have the same shape. Hence 1 is placed at the end of the third row. From the bottom entry ($i = 13$) of Figure 2 we obtain:

$$P = \begin{array}{l} 4\,4\,3\,3\,3\,1 \\ 3\,2\,2\,2\,1 \\ 1\,1 \end{array}, \qquad Q = \begin{array}{l} 3\,3\,3\,2\,2\,2 \\ 2\,2\,1\,1\,1 \\ 1\,1 \end{array}. \tag{11}$$

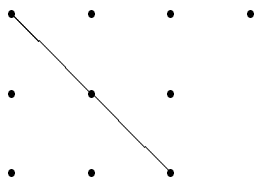
The final step of the Bender-Knuth bijection is to merge the two column-strict plane partitions P and Q into a single plane partition π. We do this by merging column-by-column, that is, the kth columns of P and Q are merged to form the kth column of π. Let us first merge the first columns of P and Q in equation (11). The following diagram illustrates the merging procedure:

i	P_i	Q_i
1	3	3
2	3 1	3 3
3	3 1 1	3 3 3
4	3 2 1 1	3 3 3 2
5	3 2 2 1 1	3 3 3 2 2
6	3 2 2 2 1 1	3 3 3 2 2 2
7	3 2 2 2 1 1 1	3 3 3 2 2 2 2
8	3 2 2 2 1 1 1 1	3 3 3 2 2 2 2 2
9	4 2 2 2 1 1 3 1 1	3 3 3 2 2 2 2 2 1
10	4 4 2 2 1 1 3 2 1 1	3 3 3 2 2 2 2 2 1 1
11	4 4 3 2 1 1 3 2 2 1 1	3 3 3 2 2 2 2 2 1 1 1
12	4 4 3 3 1 1 3 2 2 2 1 1	3 3 3 2 2 2 2 2 1 1 1 1
13	4 4 3 3 3 1 3 2 2 2 1 1 1	3 3 3 2 2 2 2 2 1 1 1 1 1

Figure 2: The Schensted correspondence



The number of dots in each row on or to the right of the main diagonal (which runs southeast from the upper left-hand corner) is equal to 4, 3, 1, the entries of the first column of P. Similarly, the number of dots in each column on or below the main diagonal is equal to 3, 2, 1, the entries of the first column of Q. The total number of dots in each row is 4, 4, 3, and we let these numbers be the entries of the first column of π. In the same way, the second column of π has entries 4, 3, 3, as shown by the following diagram:



When this merging procedure is carried out to all the columns of P and Q, we obtain the plane partition

$$\pi = \begin{array}{l} 4\,4\,3\,3\,3\,1 \\ 4\,3\,3\,3\,2\,1 \\ 3\,3\,1 \end{array} . \tag{12}$$

This gives the desired bijection that proves MacMahon's formula (8). Of course there are many details to be proved in order to verify that this procedure has all the necessary properties. The key point is that every step is *reversible*. A good way to convince yourself of the accuracy of the procedure is to take the plane partition π of equation (12) and try to reconstruct the original choice of terms from the product $1/(1-x)(1-x^2)^2\cdots$.

By analyzing more carefully the above bijective proof, it is possible to extend the formula (8) of MacMahon. Write $[i]$ as short for $1-x^i$. Without going into any of the details, let us simply state that if $pp_{rs}(n)$ denotes the number of plane partitions of n with at most r rows and at most s columns,

where say $r \leq s$, then

$$1 + pp_{rs}(1)x + pp_{rs}(2)x^2 + \cdots =$$

$$\frac{1}{[1][2]^2[3]^3\cdots[r]^r[r+1]^r\cdots[s]^r[s+1]^{r-1}[s+2]^{r-2}\cdots[r+s-1]}. \qquad (13)$$

For instance, when $r = 3$ and $s = 5$ the right-hand side of equation (13) becomes

$$\frac{1}{(1-x)(1-x^2)^2(1-x^3)^3(1-x^4)^3(1-x^5)^3(1-x^6)^2(1-x^7)}$$

$$= 1+x+3x^2+6x^3+12x^4+21x^5+39x^6+64x^7+109x^8+175x^9+280x^{10}+\cdots.$$

For example, the fact that the coefficient of x^4 is 12 means that there are 12 plane partitions of 4 with at most 3 rows and at most 5 columns. These plane partitions are given by

$$4 \quad 3\,1 \quad 2\,2 \quad 2\,1\,1 \quad 1\,1\,1\,1 \quad \begin{matrix}3\\1\end{matrix} \quad \begin{matrix}2\\2\end{matrix} \quad \begin{matrix}2\,1\\1\end{matrix} \quad \begin{matrix}1\,1\\1\,1\end{matrix} \quad \begin{matrix}1\,1\,1\\1\end{matrix} \quad \begin{matrix}2\\1\\1\end{matrix} \quad \begin{matrix}1\,1\\1\\1\end{matrix}.$$

By more sophisticated arguments (not a direct bijective proof) one can extend equation (13) even further, as follows. Let $pp_{rst}(n)$ denote the number of plane partitions of n with at most r rows, at most s columns, and with largest part at most t. Then

$$1 + pp_{rst}(1)x + pp_{rst}(2)x^2 + \cdots =$$

$$\frac{[1+t][2+t]^2[3+t]^3\cdots[r+t]^r[r+1+t]^r\cdots[s+t]^r[s+1+t]^{r-1}[s+2+t]^{r-2}\cdots[s+r-1+t]}{[1][2]^2[3]^3\cdots[r]^r[r+1]^r\cdots[s]^r[s+1]^{r-1}[s+2]^{r-2}\cdots[s+r-1]}. \qquad (14)$$

Note that the right-hand sides of equations (13) and (14) have the same denominator. The numerator of (14) is obtained by replacing each denominator factor $[i]$ with $[i+t]$. Equation (14) was also first proved by MacMahon, and is the culmination of his work on plane partitions. It is closely related to some facts in *representation theory* , a subject that at first sight seems to have no connection with plane partitions. (See the Box "Connections with representation theory".) MacMahon's results have many other variations which give simple product formulas for enumerating various classes of plane

partitions. It seems natural to try to extend these results to even higher dimensions. Thus a three-dimensional analogue of plane partitions would be *solid partitions*. All attempts (beginning in fact with MacMahon) to find nice formulas for general classes of solid partitions have resulted in failure. It seems that plane partitions are fundamentally different in behavior than their higher dimensional analogues.

As a concrete example of equation (14), suppose that $r = 2$, $s = 3$, and $t = 2$. The right-hand side of (14) becomes

$$\frac{(1-x^3)(1-x^4)^2(1-x^5)^2(1-x^6)}{(1-x)(1-x^2)^2(1-x^3)^2(1-x^4)}$$

$$= 1 + x + 3x^2 + 4x^3 + 6x^4 + 6x^5 + 8x^6 + 6x^7 + 6x^8 + 4x^9 + 3x^{10} + x^{11} + x^{12}.$$

The Schensted correspondence has a number of remarkable properties that were not needed for the derivation of MacMahon's formula (8). The most striking of these properties is the following. Consider a two-line array A such as (10) which is the input to the Schensted correspondence. Now interchange the two rows, and sort the columns so that the first row is weakly decreasing, and the part of the second row below a fixed number in the first row is also weakly decreasing. Call this new two-line array the *transposed array* A'. For the two-line array A of equation (10) we have

$$A' = \begin{array}{c} 4\ 4\ 3\ 3\ 3\ 3\ 2\ 2\ 2\ 1\ 1\ 1\ 1 \\ 1\ 1\ 3\ 1\ 1\ 1\ 2\ 2\ 2\ 3\ 3\ 2\ 2 \end{array} \tag{15}$$

Thus the Schensted correspondence can be applied to A'. If (P, Q) is the pair of column-strict plane partitions obtained by applying the Schensted correspondence to A, then applying this correspondence to A' produces the pair (Q, P), that is, the roles of P and Q are reversed! Keeping in mind the totally different combinatorial rules for forming P and Q, it seems almost miraculous when trying a particular example such as (10) and (15) that we obtain such a simple result. We can use this "symmetry property" of the Schensted correspondence to enumerate further classes of plane partitions. In particular, a plane partition is called *symmetric* if it remains the same when reflected about the main diagonal running from the upper left-hand corner in the southeast direction. An example of a symmetric plane partition is given

by

$$\begin{array}{l} 5\ 3\ 3\ 2\ 1\ 1\ 1 \\ 3\ 3\ 3\ 2\ 1 \\ 3\ 3\ 2\ 1\ 1 \\ 2\ 2\ 1 \\ 1\ 1\ 1 \\ 1 \\ 1 \end{array}$$

Let $s(n)$ denote the number of symmetric plane partitions of n. For instance, $s(5) = 4$, as shown by

$$\begin{array}{cccc} 5 & 31 & 21 & 111 \\ & 1 & 11 & 1 \\ & & & 1 \end{array} .$$

Without going into any details, let us just say that the symmetry property of the Schensted correspondence just described yields a bijective proof, similar to the proof we have given of MacMahon's formula (8), of the generating function

$$s(0) + s(1)x + s(2)x^2 + \cdots$$

$$= \frac{1}{(1-x)(1-x^3)(1-x^4)(1-x^5)(1-x^6)(1-x^7)(1-x^8)^2(1-x^9)(1-x^{10})^2\cdots}.$$

The exponent of $1-x^{2k-1}$ in the denominator is 1, and the exponent of $1-x^{2k}$ is $\lfloor k/2 \rfloor$, the greatest integer less than or equal to $k/2$.

4 Standard Young tableaux.

There is a special class of objects closely related to plane partitions that are of considerable interest. Let λ be an ordinary partition of n with parts $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\ell$. A *standard Young tableau* (SYT) of shape λ is a left-justified array of positive integers, with λ_i integers in the ith row, satisfying the following two conditions: (1) The entries consist of the integers $1, 2, ..., n$, each occurring exactly once, and (2) the entries in each row and column are increasing. An example of an SYT of shape $(4, 3, 2)$ is given by

$$\begin{array}{l} 1\ 3\ 4\ 6 \\ 2\ 7\ 8 \\ 5\ 9 \end{array} \qquad (16)$$

There are exactly ten SYT of size four (that is, with four entries), given by

$$\begin{array}{cccccccccc} 1\,2\,3\,4 & \begin{array}{c}1\,2\,3\\4\end{array} & \begin{array}{c}1\,2\,4\\3\end{array} & \begin{array}{c}1\,3\,4\\2\end{array} & \begin{array}{c}1\,2\\3\,4\end{array} & \begin{array}{c}1\,3\\2\,4\end{array} & \begin{array}{c}1\,2\\3\\4\end{array} & \begin{array}{c}1\,3\\2\\4\end{array} & \begin{array}{c}1\,4\\2\\3\end{array} & \begin{array}{c}1\\2\\3\\4\end{array} \end{array}$$

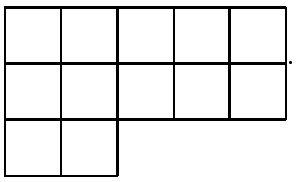
Standard Young tableaux have a number of interpretations which make them of great importance in a variety of algebraic, combinatorial, and probabilistic problems. Here we will only mention a classical problem called the *ballot problem*, which has numerous applications in probability theory. Given a partition $\lambda = (\lambda_1, \ldots, \lambda_\ell)$ as above with $\lambda_1 + \cdots + \lambda_\ell = n$, we suppose that an election is being held among ℓ candidates $A_1, \ldots, A_\ell$. At the end of the election candidate A_i receives λ_i votes. The voters vote in succession one at a time. We record the votes of the voters as a sequence $a_1, a_2, \ldots, a_n$, where $a_j = i$ if the jth voter votes for A_i. The sequence $a_1, a_2, \ldots, a_n$ is called a *ballot sequence* (of shape λ) if at no time during the voting does any candidate A_i trail another candidate A_j with $j > i$. Thus the candidates maintain their relative order (allowing ties) throughout the election. For instance, the sequence $1, 2, 1, 3, 1, 3, 4, 2$ is not a ballot sequence, since at the end A_2 and A_3 receive the same number of votes, but after six votes A_2 trails A_3. On the other hand, the sequence $1, 2, 1, 3, 1, 2, 4, 3$ is a ballot sequence. Despite the difference in their descriptions, a ballot sequence is nothing more than a disguised version of an SYT. Namely, if T is an SYT, then define $a_j = i$ if j appears in the ith row of T. A little thought should convince the reader that the sequence $a_1, a_2, \ldots, a_n$ is then a ballot sequence, and that all ballot sequences come in this way from SYT's. For instance, the SYT of equation (16) corresponds to the ballot sequence $1, 2, 1, 1, 3, 1, 2, 2, 3$.

It is natural (at least for a practitioner of combinatorics) to ask how many SYT there are of a given shape λ. This number is denoted f^λ. For instance, there are nine SYT of shape $(4, 2)$, which we write as $f^{4,2} = 9$. These nine SYT are given by

$$\begin{array}{ccccccccc} 1\,2\,3\,4 & 1\,2\,3\,5 & 1\,2\,3\,6 & 1\,2\,4\,5 & 1\,2\,4\,6 & 1\,2\,5\,6 & 1\,3\,4\,5 & 1\,3\,4\,6 & 1\,3\,5\,6 \\ 5\,6 & 4\,6 & 4\,5 & 3\,6 & 3\,5 & 3\,4 & 2\,6 & 2\,5 & 2\,4 \end{array}$$

A formula for f^λ (stated in terms of ballot sequences) was given by MacMahon in 1900. A simplified version was given by James Sutherland Frame

(1907–1997), Robinson (mentioned earlier in connection with the Schensted correspondence), and Robert McDowell Thrall (b. 1914) in 1954, and is known as the Frame-Robinson-Thrall *hook-length formula*. To state this formula, we define a *Young diagram* of shape λ as a left-justified array of squares with λ_i squares in the ith row. For instance, a Young diagram of shape $(5,5,2)$ looks like



An SYT of shape λ can then be regarded as an insertion of the numbers $1,2,\ldots,n$ (each appearing once) into the squares of a Young diagram of shape λ such that every row and column is increasing. If s is a square of a Young diagram, then define the *hook-length* of s to be the number of squares to the right of s and in the same row, or below s and in the same column, counting s itself once. In the following figure, we have inserted inside each square of the Young diagram of shape $(5,5,2)$ its hook-length.

7	6	4	3	2
6	5	3	2	1
2	1			

The *hook product* H_λ of a partition λ is the product of the hook-lengths of its Young diagram. Thus for instance from the above figure we see that

$$H_{5,5,2} = 7 \cdot 6 \cdot 4 \cdot 3 \cdot 2 \cdot 6 \cdot 5 \cdot 3 \cdot 2 \cdot 1 \cdot 2 \cdot 1 = 362,880.$$

The Frame-Robinson-Thrall formula can now be stated. Here λ is a partition of n and $n!$ (read "n factorial") is short for $1 \cdot 2 \cdots n$.

Hook-length Formula.

$$f^\lambda = \frac{n!}{H_\lambda}. \tag{17}$$

For instance,

$$f^{5,5,2} = \frac{12!}{362,880} = 1320.$$

It is remarkable that such a simple formula for f^λ exists, and no really simple proof is known. The proof of Frame-Robinson-Thrall amounts to simplifying MacMahon's formula for f^λ, which MacMahon obtained by solving difference equations (the discrete analogue of differential equations). Other proofs were subsequently given, including several bijective proofs, but none is as simple as the proof we have sketched of equation (8) using Schensted's correspondence.

In addition to their usefulness in combinatorics, SYT also play a significant role in the theory of symmetry. This important theory (known in mathematics as "the representation theory of the symmetric group") was developed primarily by Alfred Young (1873–1940), who was a clergyman by profession and a fellow of Clare College, Cambridge, a Canon of Chelmsford, and Rector of Birdbrook, Essex (1910–1940). Roughly speaking, this theory describes the possible "symmetry states" of n objects. See the Box entitled "Connections with representation theory" for more details.

A *permutation* of the numbers $1, 2, \ldots, n$ is simply a rearrangement, that is, a way of listing these numbers in some order. For instance, $5, 2, 7, 6, 1, 4, 3$ (also written as just 5276143 when no confusion can arise) is a permutation of $1, 2, 3, 4, 5, 6, 7$. The number of permutations of $1, 2, \ldots, n$ is $n! = n(n-1)\cdots 2 \cdot 1$. This fact was motivated in the Introduction, where we spoke about words with n distinct letters, which are easily seen to be equivalent to permutations.

It is an immediate consequence of the theory of symmetry that the number of ordered pairs of SYT of the same shape and with n squares is equal to $n!$, i.e. the number of permutations of n objects. For instance, when $n = 3$ we get the six pairs

$$\begin{pmatrix} 1\,2\,3 & 1\,2\,3 \end{pmatrix} \quad \begin{pmatrix} 1\,2 & 1\,2 \\ 3 & 3 \end{pmatrix} \quad \begin{pmatrix} 1\,2 & 1\,3 \\ 3 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1\,3 & 1\,2 \\ 2 & 3 \end{pmatrix} \quad \begin{pmatrix} 1\,3 & 1\,3 \\ 2 & 2 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{pmatrix}.$$

The fact that the number of pairs of SYT of the same shape and with n

$$\begin{array}{ll} 1\,3\,4 & 9\,7\,6 \\ 2\,6\,8 & 8\,4\,2 \\ 5\,9 & 5\,1 \\ 7 & 3 \end{array}$$

Figure 3: An SYT and its corresponding reverse SYT

squares is $n!$ can also be expressed by the formula

$$\sum_{\lambda \vdash n} \left(f^\lambda\right)^2 = n!, \tag{18}$$

where $\lambda \vdash n$ denotes that λ is a partition of n. A combinatorialist will immediately ask whether there is a bijective proof of this formula. In other words, given a permutation w of the numbers $1, 2, \ldots, n$, can we associate with w a pair (T_1, T_2) of SYT of the same shape and with n squares, such that every such pair occurs exactly once? In fact we have already seen the solution to this problem — it is just a special case of the Schensted correspondence! There is only one minor technicality that needs to be explained before we apply the Schensted correspondence. Namely, the column-strict plane partitions we were dealing with before have every row and column *decreasing*, while SYT have every row and column *increasing*. However, given a plane partition whose entries are the integers $1, 2, \ldots, n$, each appearing once (so it will automatically be column-strict), we need only replace i by $n + 1 - i$ to obtain an SYT of the same shape. We will call a plane partition whose (nonzero) parts are the integers $1, 2, \ldots, n$, each appearing once, a *reverse SYT*. An example of an SYT and the corresponding reverse SYT obtained by replacing i with $n + 1 - i$ is shown in Figure 3.

So consider now a permutation such as $5, 2, 6, 1, 4, 7, 3$. Write this as the second line of a two-line array whose first line is $n, n-1, \ldots, 1$. Here we get the two-line array

$$A = \begin{array}{c} 7\ 6\ 5\ 4\ 3\ 2\ 1 \\ 5\ 2\ 6\ 1\ 4\ 7\ 3 \end{array}.$$

When we apply the Schensted correspondence to this two-line array, we will obtain a pair of column-strict plane partitions of the same shape whose parts

are $1, 2, \ldots, n$, each appearing once. Namely, we get

$$\begin{array}{ll} 7\,4\,3 & 7\,6\,4 \\ 6\,2\,1 & 5\,3\,1 \\ 5 & 2 \end{array} .$$

If we replace i by $8 - i$, we get the following pair of SYT of the same shape $(3, 3, 1)$:

$$\begin{array}{ll} 1\,4\,5 & 1\,2\,4 \\ 2\,6\,7 & 3\,5\,7 \\ 3 & 6 \end{array} .$$

The process is reversible; that is, beginning with a pair (P, Q) of SYT of the same shape, we can reconstruct the permutation that produced it. (The details of this argument are left as an exercise.) Therefore the number of pairs of SYT of the same shape and with n entries is equal to the number of permutations $a_1, \ldots, a_n$ of $1, 2, \ldots, n$, yielding the formula (18). This remarkable connection between permutations and tableaux is the foundation for an elaborate theory of permutation enumeration. In the next section we give a taste of this theory.

BOX: Connections with representation theory. In this box we assume familiarity with the fundamentals of representation theory. First we consider the group $G = \mathrm{GL}(n, \mathbb{C})$ of all invertible linear transformations on an n-dimensional complex vector space V. We will identify G with the group of $n \times n$ invertible complex matrices. A *polynomial representation* of G of degree N is a homomorphism $\varphi : G \to \mathrm{GL}(N, \mathbb{C})$, such that for $A \in G$, the entries of the matrix $\varphi(A)$ are polynomials (independent of the choice of A) in the entries of A. For instance, one can check directly that the map $\varphi : \mathrm{GL}(2, \mathbb{C}) \to \mathrm{GL}(3, \mathbb{C})$ defined by

$$\varphi : \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \to \begin{bmatrix} a^2 & 2ab & b^2 \\ ac & ad + bc & bd \\ c^2 & 2cd & d^2 \end{bmatrix} \tag{19}$$

preserves multiplication (and the identity element), and hence is a polynomial representation of $\mathrm{GL}(2, \mathbb{C})$ of degree 3. Let $\varphi : \mathrm{GL}(n, \mathbb{C}) \to \mathrm{GL}(N, \mathbb{C})$ be a polynomial representation. If the eigenvalues of A are $x_1, \ldots, x_n$, then the

eigenvalues of $\varphi(A)$ are *monomials* in the x_i's. For instance, in equation (19) one can check that if x_1 and x_2 are the eigenvalues of A, then the eigenvalues of $\varphi(A)$ are x_1^2, x_1x_2, and x_2^2. The *trace* of $\varphi(A)$ (the sum of the eigenvalues) is therefore a polynomial in the x_i's which is a sum of N monomials. This polynomial is called the *character* of φ, denoted $\mathrm{char}(\varphi)$. For φ as in (19), we have

$$\mathrm{char}(\varphi) = x_1^2 + x_1x_2 + x_2^2.$$

Some of the basic facts concerning the characters of $\mathrm{GL}(n,\mathbb{C})$ are the following:

- Every polynomial representation (assumed finite-dimensional) of the group $\mathrm{GL}(n,\mathbb{C})$ is completely reducible, i.e., a direct sum of irreducible polynomial representations. These irreducible constituents are unique up to equivalence.
- The characters of irreducible representations are homogeneous symmetric functions in the variables $x_1,\ldots,x_n$, and only depend on the representation up to equivalence.
- The characters of inequivalent irreducible representations are linearly independent.

The effect of these properties is that once we determine the character of a polynomial representation φ of $\mathrm{GL}(n,\mathbb{C})$, then there is a unique way to write this character as a sum of irreducible characters. The representation φ is determined up to equivalence by the multiplicity of each irreducible character in $\mathrm{char}(\varphi)$. Hence we are left with the basic question of describing the irreducible character of $\mathrm{GL}(n,\mathbb{C})$. The main result is the following.

Fundamental theorem on the polynomial characters of $\mathbf{GL}(n,\mathbb{C})$. *The irreducible characters of* $\mathrm{GL}(n,\mathbb{C})$ *are in one-to-one correspondence with the partitions* $\lambda = (\lambda_1,\ldots,\lambda_n)$ *with at most* n *parts. The irreducible character* $s_\lambda = s_\lambda(x_1,\ldots,x_n)$ *corresponding to* λ *is given by*

$$s_\lambda(x_1,\ldots,x_n) = \sum_T x^T,$$

where T ranges over all column-strict plane partitions of shape λ and largest part at most n, and where x^T denotes the monomial

$$x^T = x_1^{\text{number of 1's in } T} x_2^{\text{number of 2's in } T} \cdots .$$

For instance, let $n = 2$ and let $\lambda = (2,0)$ be the partition with just one part equal to two (and no other parts). The column-strict plane partitions of shape $(2,0)$ with largest part at most 2 are just 11, 21, and 22. Hence (abbreviating $s_{(2,0)}$ as s_2),

$$s_2(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2.$$

This is just the character of the representation defined by equation (19). Hence this representation is one of the irreducible representations of $\mathrm{GL}(2, \mathbb{C})$.

As another example, suppose that $n = 3$ and $\lambda = (2,1,0)$. The corresponding column-strict plane partitions are

$$\begin{array}{cccccccc} 2\,1 & 2\,2 & 3\,1 & 3\,1 & 3\,2 & 3\,2 & 3\,3 & 3\,3 \\ 1 & 1 & 1 & 2 & 1 & 2 & 1 & 2 \end{array} .$$

Hence

$$s_\lambda(x_1, x_2, x_3) = x_1^2 x_2 + x_1 x_2^2 + x_1^2 x_3 + 2 x_1 x_2 x_3 + x_2^2 x_3 + x_1 x_3^2 + x_2 x_3^2.$$

The fact that we have eight column-strict plane partitions in this case is closely related to the famous "Eightfold Way" of particle physics. (The corresponding representation of $\mathrm{GL}(3, \mathbb{C})$, when restricted to $\mathrm{SL}(3, \mathbb{C})$, is just the adjoint representation of $\mathrm{SL}(3, \mathbb{C})$.)

The symmetric functions $s_\lambda(x_1, \ldots, x_n)$ are known as *Schur functions* (in the variables $x_1, \ldots, x_n$) and play an important role in many aspects of representation theory, the theory of symmetric functions, and enumerative combinatorics. In particular, they are closely related to the irreducible representations of a certain *finite* group, namely, the symmetric group $\mathfrak{S}_k$ of all permutations of the set $\{1, 2, \ldots, k\}$. This relationship is best understood by a "duality" between $\mathrm{GL}(n, \mathbb{C})$ and $\mathfrak{S}_k$ discovered by Issai Schur (1875–1941).

Recall that we are regarding $\mathrm{GL}(n, \mathbb{C})$ as acting on an n-dimensional vector space V. Thus $\mathrm{GL}(n, \mathbb{C})$ also acts on the kth tensor power $V^{\otimes k}$ of

V. On the other hand, the group $\mathfrak{S}_k$ acts on $V^{\otimes k}$ by permuting tensor coordinates. Schur's famous "double centralizer" theorem asserts that the actions of $\mathrm{GL}(n,\mathbb{C})$ and $\mathfrak{S}_k$ centralize each other, i.e., every endomorphism of $V^{\otimes k}$ commuting with the action of $\mathrm{GL}(n,\mathbb{C})$ is a linear combination of the actions of the elements of $\mathfrak{S}_k$, and *vice versa*. From this one can show that the action of the group $\mathfrak{S}_k \times \mathrm{GL}(n,\mathbb{C})$ on $V^{\otimes k}$ breaks up into irreducible constituents in the form

$$V^{\otimes k} = \coprod_{\lambda} \left(M^\lambda \otimes F_\lambda \right), \tag{20}$$

where (a) $\coprod$ denotes a direct sum of vector spaces, (b) λ ranges over all partitions of k into at most n parts, (c) F_λ is the irreducible $\mathrm{GL}(n,\mathbb{C})$-module corresponding to λ, and M^λ is an irreducible $\mathfrak{S}_k$-module. Thus when $k \leq n$, λ ranges over *all* partitions of k. The $p(k)$ irreducible $\mathfrak{S}_k$-modules M^λ are pairwise nonisomorphic and account for all the irreducible $\mathfrak{S}_k$-modules. Hence the irreducible $\mathfrak{S}_k$-modules are naturally indexed by partitions of k. Using the Schensted correspondence (or otherwise), it is easy to prove the identity

$$(x_1 + x_2 + \cdots + x_n)^k = \sum_{\lambda} f^\lambda s_\lambda(x_1, \ldots, x_n),$$

where λ ranges over all partitions of k and f^λ denotes as usual the number of SYT of shape λ. Comparing with equation (20) and using the fact that the character of $\mathrm{GL}(n,\mathbb{C})$ acting on $V^{\otimes k}$ is $(x_1 + \cdots + x_n)^k$, we see that $\dim M^\lambda = f^\lambda$. Thus the f^λ's for λ a partition of k are the degrees of the irreducible representations of $\mathfrak{S}_k$. Since the sum of the squares of the degrees of the irreducible representations of a finite group G is equal to the order (number of elements) of G, we obtain equation (18) (with n replaced by k).

We have only given the briefest glimpse of the connections between tableau combinatorics and representation theory, but we hope that it gives the reader with sufficient mathematical background the flavor of this subject.

---

5 Increasing and decreasing subsequences.

In this section we discuss an unexpected connection between the Schensted correspondence and the enumeration of a certain class of permutations. This connection was discovered by Schensted and was his reason for inventing his famous correspondence. If $w = a_1 a_2 \cdots a_n$ is a permutation of $1, 2, \ldots, n$, then a *subsequence* v of length k of w is a sequence of k distinct terms of w appearing in the order in which they appear in w. In symbols, we have $v = a_{i_1} a_{i_2} \cdots a_{i_k}$, where $i_1 < i_2 < \cdots < i_k$. For instance, some subsequences of the permutation 6251743 are 2573, 174, 6, and 6251743. A subsequence $b_1 b_2 \cdots b_k$ of w is said to be *increasing* if $b_1 < b_2 < \cdots < b_k$, and *decreasing* if $b_1 > b_2 > \cdots > b_k$. For instance, some increasing subsequences of 6251743 are 67, 257, and 3, while some decreasing subsequences are 6543, 654, 743, 61, and 3.

We will be interestested in the length of the *longest* increasing and decreasing subsequences of a permutation w. Denote by $i(w)$ the length of the longest increasing subsequence of w, and by $d(w)$ the length of the longest decreasing subsequence. By careful inspection one sees for instance that $i(6251743) = 3$ and $d(6251743) = 4$. It is intuitively plausible that there should be some kind of tradeoff between the values $i(w)$ and $d(w)$. If $i(w)$ is small, say equal to k, then any subsequence of w of length $k + 1$ must contain a pair of decreasing elements, so there are "lots" of pairs of decreasing elements. Hence we would expect $d(w)$ to be large. An extreme case occurs when $i(w) = 1$. Then there is only one choice for w, namely, $n, n-1, \ldots, 1$, and we have $d(w) = n$.

How can we quantify the feeling that that $i(w)$ and $d(w)$ cannot both be small? A famous result of Pal Erdős (1913–1996) and George Szekeres (b. 1911), obtained in 1935, gives an answer to this question and was one of the first results in the currently very active area of *extremal combinatorics*.

Erdős-Szekeres Theorem. *Let w be a permutation of $1, 2, \ldots, n$, and let p and q be positive integers for which $n > pq$. Then either $i(w) > p$ or $d(w) > q$. Moreover, this is best possible in the sense that if $n = pq$ then we can find at least one permutation w such that $i(w) = p$ and $d(w) = q$.*

An equivalent way to formulate the Erdős-Szekeres theorem is by the inequality

$$i(w) \cdot d(w) \geq n,$$

showing clearly that $i(w)$ and $d(w)$ cannot both be small. For instance, both cannot be less than $\sqrt{n}$, the square root of n.

After Erdős and Szekeres proved their theorem, an extremely elegant proof was given in 1959 by Abraham Seidenberg (1916–1988) based on a ubiquitous mathematical tool known as the *pigeonhole principle*. This principle states that if $m + 1$ pigeons fly into m pigeonholes, then at least one pigeonhole contains more than one pigeon. As trivial as the pigeonhole principle may sound, it has numerous nontrivial applications. The hard part in applying the pigeonhole principle is deciding what are the pigeons and what are the pigeonholes.

We can now describe Seidenberg's proof of the Erdős-Szekeres theorem. Given a permutation $w = a_1 a_2 \cdots a_n$ of $1, 2, \ldots, n$, we define numbers $r_1, r_2, \ldots, r_n$ and $s_1, s_2, \ldots, s_n$ as follows. Let r_i be the length of the longest increasing subsequence of w that ends at a_i, and similarly let s_i be the length of the longest decreasing subsequence of w that ends at a_i. For instance, if $w = 6251743$ as above then $s_4 = 3$ since the longest decreasing subsequences ending at $a_4 = 1$ are 621 and 651, of length three. More generally, we have for $w = 6251743$ that $(r_1, \ldots, r_7) = (1, 1, 2, 1, 3, 2, 2)$ and $(s_1, \ldots, s_7) = (1, 2, 2, 3, 1, 3, 4)$.

Key fact. *The n pairs (r_1, s_1), $(r_2, s_2), \ldots, (r_n, s_n)$ are all distinct.*

To see why this fact is true, suppose i and j are numbers such that $i < j$ and $a_i < a_j$. Then we can append a_j to the end of the longest increasing subsequence of w ending at a_i to get an increasing subsequence of greater length that ends at a_j. Hence $r_j > r_i$. Similarly, if $i < j$ and $a_i > a_j$, then we get $s_j > s_i$. Therefore we cannot have both $r_i = r_j$ *and* $s_i = s_j$, which proves the key fact.

Now suppose $n > pq$ as in the statement of the Erdős-Szekeres theorem. We therefore have n distinct pairs (r_1, s_1), $(r_2, s_2), \ldots, (r_n, s_n)$ of positive integers. If every r_i were at most p and every s_i were at most q, then there

are only pq possible pairs (r_i, s_i) (since there are at most p choices for r_i and at most q choices for s_i). Hence two of these pairs would have to be equal. (This is where the pigeonhole principle comes in — we are putting the "pigeon" i into the "pigeonhole" (r_i, s_i) for $1 \leq i \leq n$. Thus there are n pigeons, where $n > pq$, and at most pq pigeonholes.) But if two pairs are equal, then we contradict the key fact above. It follows that for some i either $r_i > p$ or $s_i > q$. If $r_i > p$ then there is an increasing subsequence of w of length at least $p+1$ ending at a_i, so $i(w) > p$. Similarly, if $s_i > q$ then $d(w) > q$, completing the proof of the main part of the Erdős-Szekeres theorem.

It remains to show that the result is best possible, as explained above. In other words, given p and q, we need to exhibit at least one permutation w of $1, 2, \ldots, pq$ such that $i(w) = p$ and $d(w) = q$. It is easy to check that the following choice of w works:

$$w = (q-1)p+1, (q-1)p+2, \ldots, qp, (q-2)p+1, (q-2)p+2, \ldots, (q-1)p, \\ \ldots, 2p+1, 2p+2, \ldots, 3p, p+1, p+2, \ldots, 2p, 1, 2, \ldots, p. \qquad (21)$$

This completes the proof of the Erdős-Szekeres theorem.

Though the Erdős-Szekeres theorem is very elegant, we can ask for even more information about increasing and decreasing subsequences. For instance, rather than exhibiting a single permutation w of $1, 2, \ldots, pq$ satisfying $i(w) = p$ and $d(w) = q$, we can ask how many such permutations there are. This much harder question can be answered by using an unexpected connection between increasing and decreasing subsequences on the one hand, and the Schensted correspondence on the other.

There are two fundamental properties of the Schensted correspondence that are needed for our purposes. Suppose we apply the Schensted correspondence to a permutation $w = a_1 a_2 \cdots a_n$ of $1, 2, \ldots, n$, getting two column-strict plane partitions P and Q whose parts are $1, 2, \ldots, n$. The first property we need of the Schensted correspondence is a simple description of the first row of P.

Property 1. *Suppose that the first row of P is $b_1 b_2 \cdots b_k$. Then b_i is the last (rightmost) term in w such that the longest decreasing subsequence of w ending at that term has length i.*

For instance, suppose $w = 843716925$. Then

$$P = \begin{array}{l} 9\,7\,6\,5 \\ 8\,3\,2 \\ 4\,1 \end{array} .$$

The first row of P is 9765. Consider the third element of this row, which is 6. Then 6 is the rightmost term of w for which the longest decreasing subsequence of w ending at that term has length three. Indeed, 876 is a decreasing subsequence of length three ending at 6, and there is none longer. The terms to the right of 6 are 9, 2, and 5. The longest decreasing subsequences ending at these terms have length 1, 4, and 4, respectively, so 6 is indeed the rightmost term for which the longest decreasing subsequence ending at that term has length three.

See the Box for a proof by induction of Property 1.

BOX. *Proof of Property 1.* Recall that $w = a_1 a_2 \cdots a_n$. We prove by induction on j that after the Schensted algorithm has been applied to $a_1 a_2 \cdots a_j$, yielding a pair (P_j, Q_j) of column-strict plane partitions, then the ith entry in the first row of P_j is the rightmost term of the sequence $a_1 a_2 \cdots a_j$ such that the longest decreasing subsequence ending at that term has length i. Once this is proved, then set $j = n$ to obtain Property 1.

The assertion is clearly true for $j = 1$. Assume true for j. Suppose that the first row of P_j is $c_1 c_2 \cdots c_r$. By the induction hypothesis, c_i is the rightmost term of the sequence $a_1 a_2 \cdots a_j$ such that the longest decreasing subsequence ending at that term has length i. We now insert a_{j+1} into the first row of P_j according to the rules of the Schensted algorithm. It will bump the leftmost element c_i of this row which is less than a_{j+1}. (If there is no element of the first row of P_j which is less than a_{j+1}, then a_{j+1} is inserted at the end of the row. We then set $i = r + 1$, so that a_{j+1} is in all cases the ith element of the first row of P_{j+1}.) We need to show that the longest decreasing subsequence of the sequence $a_1 a_2 \cdots a_{j+1}$ ending at a_{j+1} has length i, since clearly a_{j+1} will be the *rightmost* element of $a_1 a_2 \cdots a_{j+1}$ with this property (since it is the rightmost element of the entire sequence).

If $i = 1$, then a_{j+1} is the largest element of the sequence $a_1 a_2 \cdots a_{j+1}$, so the longest decreasing subsequence ending at a_{j+1} has length one, as desired. If $i > 1$, then there is a decreasing subsequence of $a_1 a_2 \cdots a_j$ of length $i-1$ ending at c_{i-1}. Adjoining a_{j+1} to the end of this subsequence produces a decreasing subsequence of length i ending at a_{j+1}. It remains to show that there cannot be a longer decreasing subsequence ending at a_{j+1}. If there were, then there would be some term a_s in w to the left of a_{j+1} and larger than a_{j+1} such that the longest decreasing subsequence ending at a_s has length i. Thus when a_s is inserted into P_{s-1} during the Schensted algorithm, it becomes the ith element of the first row. It can only be bumped by terms *larger* than a_s. In particular, when a_{j+1} is inserted into the first row, the ith element is larger than a_s, which is larger than a_{j+1}. This contradicts the definition of the bumping procedure and completes the proof.

The second property we need of the Schensted correspondence was first proved by Schensted. To describe this property we require the following definition. If λ is a partition, then the *conjugate* partition λ' of λ is the partition whose Young diagram is obtained by interchanging the rows and columns of the Young diagram of λ. In other words, if $\lambda = (\lambda_1, \lambda_2, \ldots)$, then the *column* lengths of the Young diagram of λ' are $\lambda_1, \lambda_2, \ldots$. For instance, if $\lambda = (5,3,3,2)$ then $\lambda' = (4,4,3,1,1)$, as illustrated in Figure 4.

Property 2. Suppose that when the Schensted correspondence is applied to a permutation $w = a_1 a_2 \cdots a_n$, we obtain the pair (P, Q) of reverse SYT. Let $\bar{w} = a_n a_{n-1} \cdots a_1$, the *reverse* permutation of w. Suppose that when the Schensted correspondence is applied to $\bar{w}$, we obtain the pair $(\bar{P}, \bar{Q})$ of reverse SYT. Then the shape of $\bar{P}$ (or $\bar{Q}$) is conjugate to the shape of P (or Q).

Actually, an even stronger result than Property 2 is true, though we do not need it for our purposes. The reverse SYT $\bar{P}$ is actually the *transpose* of P, obtained by interchanging the rows and columns of P. (The connection between Q and $\bar{Q}$ is more subtle and has led to much interesting work.) The proof of Property 2 is too complicated for inclusion here, though it is entirely elementary.

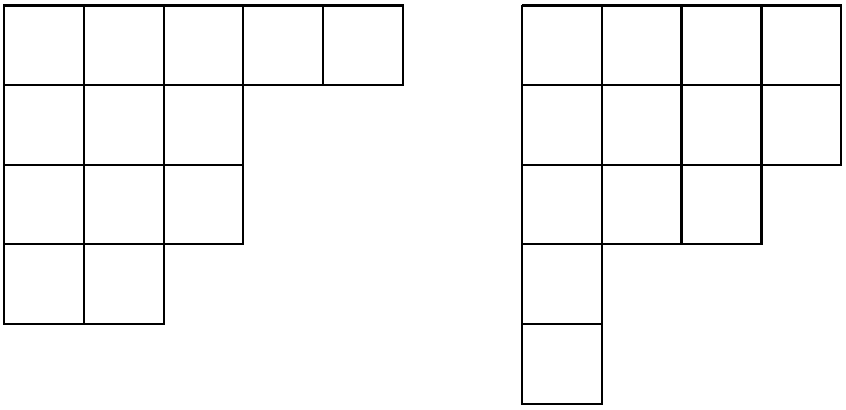


Figure 4: The Young diagram of a partition and its conjugate

We now have all the ingredients to state the main result (due to Schensted) on longest increasing and decreasing subsequences. If we apply the Schensted correspondence to the permutation w and get a pair (P, Q) of reverse SYT of shape $\lambda = (\lambda_1, \lambda_2, \ldots)$, then Property 1 tells us that

$$d(w) = \lambda_1.$$

In words, the length of the longest decreasing subsequence of w is equal to the largest part of λ (the length of the first row of P). Now apply the Schensted correspondence to the reverse permutation $\bar{w}$, obtaining the pair $(\bar{P}, \bar{Q})$ of reverse SYT. When we reverse a permutation, increasing subsequences are changed to decreasing subsequences and *vice versa*. In particular, $d(\bar{w}) = i(w)$. By Property 1, $d(\bar{w})$ is just the length of the first row of $\bar{P}$. By Property 2, the length of the first row of $\bar{P}$ is just the length of the first *column* of P. Thus $i(w) = \ell(\lambda)$, the number of parts of λ.

We have shown that for a permutation w with $i(w) = p$ and $d(w) = q$, the shape λ of the corresponding reverse SYT P (and Q) satisfies $\ell(\lambda) = p$ and $\lambda_1 = q$. Hence the number $A_n(p, q)$ of permutations w of $1, 2, \ldots, n$ with $i(w) = p$ and $d(w) = q$ is equal to the number of pairs (P, Q) of reverse SYT of the same shape λ, where λ is a partition of n with $\ell(\lambda) = p$ and $\lambda_1 = q$. How many such pairs are there? Given the partition λ, the number of choices for P is just f^λ, the number of SYT of shape λ. (Recall that the number of SYT of shape λ and the number of reverse SYT of shape λ is the same, since

we can replace i by $n+1-i$.) Similarly there are f^λ choices for Q, so there are $\left(f^\lambda\right)^2$ choices for (P,Q). Hence we obtain our main result on increasing and decreasing subsequences:

Schensted's Theorem. *The number $A_n(p,q)$ of permutations w of $1,2,\ldots,n$ satisfying $i(w)=p$ and $d(w)=q$ is equal to the sum of all $\left(f^\lambda\right)^2$, where λ is a partition of n satisfying $\ell(\lambda)=p$ and $\lambda_1=q$.*

Let us see how the Erdős-Szekeres theorem follows immediately from Schensted's theorem. If a partition λ of n satisfies $\ell(\lambda)=p$ and $\lambda_1=q$, then

$$\begin{aligned} n &= \lambda_1+\lambda_2+\cdots+\lambda_p \\ &\le q+q+\cdots+q \quad (p \text{ terms in all}) \\ &= pq. \end{aligned}$$

Hence if $n>pq$, then either $\ell(\lambda)\ge p+1$ or $\lambda_1\ge q+1$. If we apply the Schensted correspondence to a permutation w of $1,2,\ldots,n$ then we get a pair of reverse SYT of some shape λ, where λ is a partition of n. We have just shown that $\ell(\lambda)\ge p+1$ or $\lambda_1\ge q+1$, so by Schensted's theorem either $i(w)\ge p+1$ or $d(w)\ge q+1$.

We can evaluate each f^λ appearing in Schensted's theorem by the hook-length formula. Hence the theorem is most interesting when there are few partitions λ satisfying $\ell(\lambda)=p$ and $\lambda_1=q$. The most interesting case occurs when $n=pq$. The fact that there is at least *one* permutation satisfying $i(w)=p$ and $d(w)=q$ (when $n=pq$) shows that the Erdős-Szekeres theorem is best possible (see equation (21)). Now we are asking for a much stronger result — how many such permutations are there? By Schensted's theorem, we first need to find all partitions λ of n such that $\ell(\lambda)=p$ and $\lambda_1=q$. Clearly there is only one such partition, namely, the partition with p parts all equal to q. Hence for this partition λ we have $A_n(p,q)=\left(f^\lambda\right)^2$. We may assume for definiteness that $p\le q$ (since $A_n(p,q)=A_n(q,p)$). In that case the hook-lengths of λ are given by 1 (once), 2 (twice), 3 (three times), ..., p (p times), $p+1$ (p times), ..., q (p times), $q+1$ ($p-1$ times), $q+2$ ($p-2$ times), ..., $p+q-1$ (once). We finally obtain the amazing formula (for $n=pq$)

$$A_n(p,q)=\left[\frac{(pq)!}{1^1 2^2\cdots p^p(p+1)^p\cdots q^p(q+1)^{p-1}(q+2)^{p-2}\cdots(p+q-1)^1}\right]^2.$$

For instance, when $p = 4$ and $q = 6$ we easily compute that

$$\begin{aligned} A_{24}(4,6) &= \left[\frac{24!}{1^1\,2^2\,3^3\,4^4\,5^4\,6^4\,7^3\,8^2\,9^1}\right]^2 \\ &= 19,664,397,929,878,416. \end{aligned}$$

This large number is still only a small fraction .00000003169 of the total number of permutations of $1, 2, \ldots, 24$.

6 Reduced decompositions.

There is a remarkable and unexpected connection between standard Young tableaux and the building up of a permutation by interchanging (transposing) two adjacent entries. We begin with the *identity permutation* $1, 2, \ldots, n$. We wish to construct from it a given permutation as quickly as possible by interchanging adjacent elements. By "as quickly as possible," we mean in as few interchanges (called *adjacent transpositions*) as possible. This will be the case if we always transpose two elements a, b appearing in ascending order. For instance, one way to get the permutation 41352 from 12345 with a minimum number of adjacent transpositions is as follows, where we have marked in boldface the pair of elements to be interchanged:

$$1\mathbf{23}45 \to 13\mathbf{24}5 \to 1\mathbf{34}25 \to \mathbf{14}325 \to 413\mathbf{25} \to 41352. \tag{22}$$

Such sequences of interchanges are used in some of the *sorting algorithms* studied in computer science (see Section 11), although there it is natural to consider the reverse process whereby a list of numbers such as 41352 is step-by-step converted to the "sorted" list 12345. Note that the five steps in the sequence (22) are the minimum possible, since in the final permutation 41352 there are five pairs (i, j) out of order, i.e., i appears to the left of j and $i > j$ (namely, $(4,1), (4,3), (4,2), (3,2), (5,2)$), and each adjacent transposition can make at most one pair which was in order go out of order. It would be inefficient to transpose a pair (a, b) that is in order in the final permutation, since we would only have to change it back later. A pair of elements of a permutation w that is out of order is called an *inversion* of w. The number of inversions of w is denoted $\mathrm{inv}(w)$ and is an important

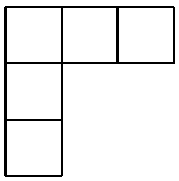
invariant of a permutation, in a sense measuring how "mixed up" the permutation is. For instance, $\mathrm{inv}(41352) = 5$, the inversions being the five pairs $(4,1), (4,3), (4,2), (3,2), (5,2)$.

A sequence of adjacent transpositions that converts the identity permutation to a permutation w in the smallest possible number of steps (namely, $\mathrm{inv}(w)$ steps) is called a *reduced decomposition* of w. Equation (22) shows one reduced decomposition of the permutation $w = 41352$, but there are many others. We can therefore ask for the number of reduced decompositions of w. We denote this number by $r(w)$. The reader can check that every permutation of the numbers $1, 2, 3$ has only one reduced decomposition, except that $r(321) = 2$. The two reduced decompositions of 321 are $123 \to 213 \to 231 \to 321$ and $123 \to 132 \to 312 \to 321$.

The remarkable connection between $r(w)$ and SYT's is the following. For each permutation w, one can associate a *small* collection $Y(w)$ of Young diagrams (with repetitions allowed) whose number of squares is $\mathrm{inv}(w)$, such that $r(w)$ is the sum of the number of SYT whose shapes belong to $Y(w)$. We are unable to explain here the exact rule (based on a variant of the Schensted correspondence) for computing $Y(w)$, but we will discuss the most interesting special case. We also will not explain exactly what is meant by a "small" collection, but in general its number of elements will be much smaller than $r(w)$ itself.

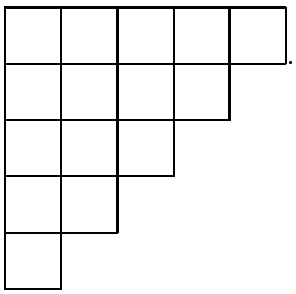
Example. Here are a few examples of the collection $Y(w)$.

(a) If $w = 41352$ (the example considered in equation (22)), then $Y(w)$ consists of the single diagram



of shape $(3,1,1)$. Since there are six SYT of this shape (computed from the hook-length formula (17) or by direct enumeration), it follows that there are six reduced decompositions of 41352.

(b) If $w = 654321$ then again $Y(w)$ is given by a single diagram, this time



Hence

$$\begin{aligned} r(w) &= f^{(5,4,3,2,1)} \\ &= \frac{15!}{1^5 \cdot 3^4 \cdot 5^3 \cdot 7^2 \cdot 9} \\ &= 292,864. \end{aligned}$$

(c) If $w = 321654$, then $Y(w)$ consists of the diagrams whose shapes are (writing for instance 42 as short for $(4,2)$) 42, 411, 33, 321, 321, 3111, 222, 2211. Note that the shape 321 appears twice. We get

$$\begin{aligned} r(w) &= f^{42} + f^{411} + f^{33} + 2f^{321} + f^{3111} + f^{222} + f^{2211} \\ &= 9 + 10 + 5 + 2 \cdot 16 + 10 + 5 + 9 \\ &= 80. \end{aligned}$$

Clearly the formula for $r(w)$ will be the simplest when $Y(w)$ consists of a single partition λ, for then we have $r(w) = f^\lambda$, given explicitly by (17). A simple though surprising characterization of all permutations for which $Y(w)$ consists of a single partition is given by the next result. Such permutations are called *vexillary* after the Latin word *vexillum* for "flag," because of a relationship between vexillary permutations and certain polynomials known as *flag Schur functions*.

Vexillary theorem. *Let $w = w_1 w_2 \cdots w_n$ be a permutation of $1, 2, \ldots, n$. Then $Y(w)$ consists of a single partition λ if and only if there do not exist $a < b < c < d$ such that $w_b < w_a < w_d < w_c$. Moreover, if α_i is the number of j's for which $i < j$ and $w_i > w_j$, then the parts of λ are just the nonzero α_i's.*

As an illustration of the above theorem, let $w = 526314$. One sees by inspection that w satisfies the conditions of the theorem. We have $(\alpha_1, \ldots, \alpha_6) = (4, 1, 3, 1, 0, 0)$. Hence $\lambda = (4, 3, 1, 1)$ and $r(w) = f^{(4,3,1,1)} = 216$.

It is immediate from the above result that all the permutations of $1, \ldots, n$ for $n \leq 3$ are vexillary, and that there is just one nonvexillary permutation of $1, 2, 3, 4$, namely, 2143. It has been computed that if $v(n)$ denotes the number of vexillary permutations of $1, 2, \ldots, n$ then $v(5) = 103$ (out of 120 permutations of $1, 2, \ldots, n$ in all), $v(6) = 513$ (out of 720), $v(7) = 2761$ (out of 5040), and $v(8) = 15767$ (out of 40320). Simple methods for computing and approximating $v(n)$ have been given by Julian West (b. 1964) and Amitai Regev (b. 1940), and an explicit formula for $v(n)$ was found by Ira Gessel (b. 1951).

There is one vexillary permutation of particular interest. This is the permutation $w_0 = n, n-1, \ldots, 1$, for which $\lambda = (n-1, n-2, \ldots, 1)$. There is an elegant bijection between the SYT of shape $(n-1, n-2, \ldots, 1)$ and the reduced decompositions of w_0, due to Paul Henry Edelman (b. 1956) and Curtis Greene (b. 1944). Begin with an SYT of shape $(n-1, n-2, \ldots, 1)$ and write the number i at the end of the ith row, with n written at the bottom of the first column. We will call the numbers outside the diagram *exit numbers*. An example is given by:

1	3	4	6	1
2	8	10	2	
5	9	3		
7	4			
5				

Now take the largest number in the SYT (in this case 10) and let it "exit" the diagram to the southeast (between the 2 and 3). Whenever a number exits the diagram, transpose the two exit numbers that it goes between.

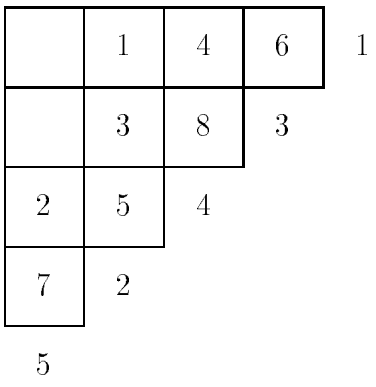
Hence we now have:

1	3	4	6	1
2	8		3	
5	9	2		
7	4			
5				

In the hole left by the 10, move the largest of the numbers directly to the left or above the hole. Here we move the 8 into the hole, creating a new hole. Continue to move the largest number directly to the left or above a hole into the hole, until such moves are no longer possible. Thus after exiting the 10, we move the 8, 3, and 1 successively into holes, yielding:

	1	4	6	1
2	3	8	3	
5	9	2		
7	4			
5				

Now repeat this procedure, first exiting the largest number in the diagram (ignoring the exit numbers), then transposing the two exit numbers between which this largest number exits, and then filling in the holes by the same method as before. Hence for our example 9 exits, 5 fills in the hole left by 9, and 2 fills in the hole left by 5, yielding:



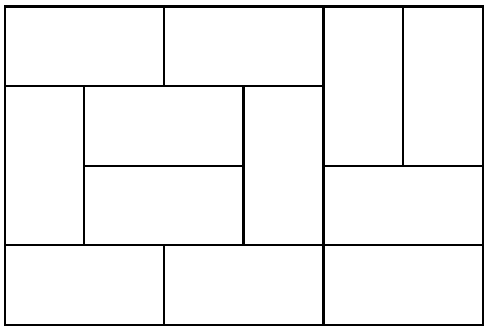
Continue in this manner until all the numbers are removed from the original SYT. The remarkable fact is that the exit numbers, read from top to bottom, will now be $n, n-1, \ldots, 1$. We began with the exit numbers in the order $1, 2, \ldots, n$, and each exit from the diagram transposed two adjacent exit numbers. The size (number of entries) of the original SYT is equal to $n(n-1)/2$, which is the number of inversions of the permutation $n, n-1, \ldots, 1$. Hence we have converted $1, 2, \ldots, n$ to $n, n-1, \ldots, 1$ by $n(n-1)/2$ adjacent transpositions, thereby defining a reduced decomposition of w_0. Edelman and Greene prove that this algorithm yields a bijection between SYT of shape $(n-1, n-2, \ldots, 1)$ and reduced decompositions of w_0. For the above example, the reduced decomposition is given by $12345 \to 13245 \to 13425 \to 14325 \to 14352 \to 41352 \to 41532 \to 45132 \to 45312 \to 45321 \to 54321$.

7 Tilings.

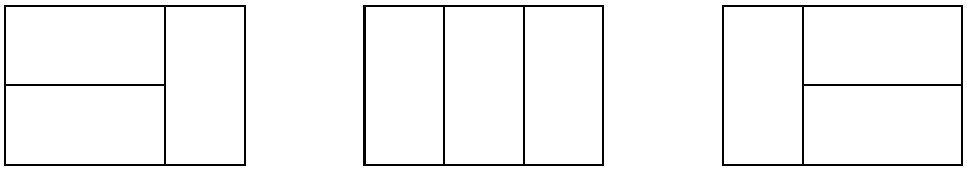
The final enumerative topic we will discuss concerns the partitioning of some planar or solid shape into smaller shapes. Such partitions are called *tilings*. The combinatorial theory of tilings is connected with such subjects as geometry, group theory, and logic, and has applications to statistical mechanics, coding theory, and many other topics. Here we will be concerned with the purely enumerative question of counting the number of tilings.

The first significant result about the enumeration of tilings was due to

the Dutch physicist Pieter Willem Kasteleyn (1924–1996) and independently to the British physicist Harold Neville Vazeille Temperley (b. 1915) and the British-born physicist Michael Ellis Fisher (b. 1931). Motivated by work related to the adsorption of diatomic molecules on a surface and other physical problems, they were led to consider the tiling of a chessboard by dominos (or dimers). More precisely, consider an $m \times n$ chessboard B, where at least one of m and n is even. A *domino* consists of two adjacent squares (where "adjacent" means having an edge in common). The domino can be oriented either horizontally or vertically. Thus a tiling of B by dominos will require exactly $mn/2$ dominos, since there are mn squares in all, and each domino has two squares. The illustration below shows a domino tiling of a 4×6 rectangle.



Let $N(m, n)$ denote the number of domino coverings of an $m \times n$ chessboard. For instance, $N(2, 3) = 3$, as shown by:



We have in fact that

$$N(2, n) = F_{n+1}, \tag{23}$$

where F_{n+1} denotes a Fibonacci number, defined by the recurrence

$$F_1 = 1, \quad F_2 = 1, \quad F_{n+1} = F_n + F_{n-1}.$$

To prove equation (23), we need to show that $N(2,1) = 1$, $N(2,2) = 2$, and $N(2,n+2) = N(2,n+1) + N(2,n)$. Of course it is trivial to check that $N(2,1) = 1$ and $N(2,2) = 2$. In any domino tiling of a $2 \times (n+2)$ rectangle, either the first column consists of a vertical domino, or else the first two columns consist of two horizontal dominos. In the former case we are left with a $2 \times (n+1)$ rectangle to tile by dominos, and in the latter case a $2 \times n$ rectangle. There are $N(2,n+1)$ ways to tile the $2 \times (n+1)$ rectangle and $N(2,n)$ ways to tile the $2 \times n$ rectangle, so the recurrence $N(2,n+2) = N(2,n+1) + N(2,n)$ follows, and hence also (23).

The situation becomes much more complicated when dealing with larger rectangles, and rather sophisticated techniques such as the "transfer-matrix method" or the "Pfaffian method" are needed to produce an answer. The final form of the answer involves trigonometric functions (see Box), and it is not even readily apparent (without sufficient mathematical background) that the formula gives an integer. It follows, however, from the subject known as *Galois theory* that $N(2n,2n)$ is in fact the square or twice the square of an integer, depending on whether n is even or odd. For instance, $N(8,8) = 12,988,816 = 3604^2$, while $N(6,6) = 6728 = 2 \cdot 58^2$. It is natural to ask for a *combinatorial* reason why these numbers are squares or twice squares. In other words, in the case when n is even we would like a combinatorial interpretation of the number $M(2n)$ defined by $N(2n,2n) = M(2n)^2$, and similarly when n is odd. While a formula for $M(2n)$ was known making it obvious that it was an integer (so not involving trigonometric functions), it was only in 1992 that William Carl Jockusch (b. 1967) found a direct combinatorial interpretation of $M(2n)$. In 1996 Mihai Adrian Ciucu (b. 1968) found an even simpler interpretation of $M(2n)$ as the number of domino tilings of a certain region R_n, up to a power of two. The region R_n is defined to be the board consisting of $2n-2$ squares in the first three rows, then $2n-4$ squares in the next two rows, then $2n-6$ squares in the next two rows, etc., down to two squares in the last two rows. All the rows are left-justified. The board R_4 is illustrated in Figure 5.

If $T(n)$ denotes the number of domino tilings of R_n, then Ciucu's formula states that

$$N(2n,2n) = 2^n T(n)^2.$$

If n is even, say $n = 2r$, then $N(2n,2n) = (2^r T(n))^2$, while if n is odd,

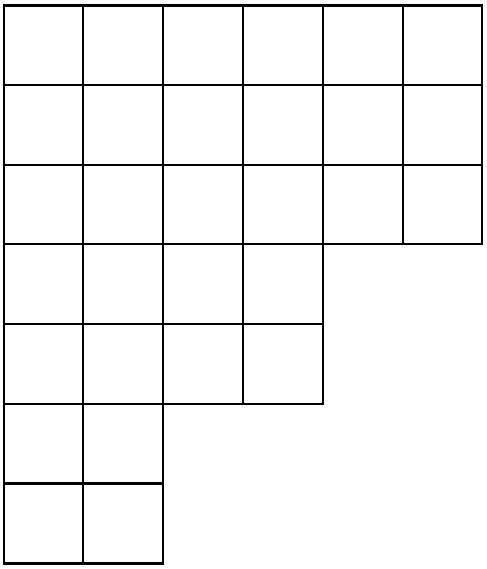


Figure 5: The board R_4.

say $n = 2r + 1$, then $N(2n, 2n) = 2(2^r T(n))^2$, so we recover the result that $N(2n, 2n)$ is a square or twice a square depending on whether n is even or odd.

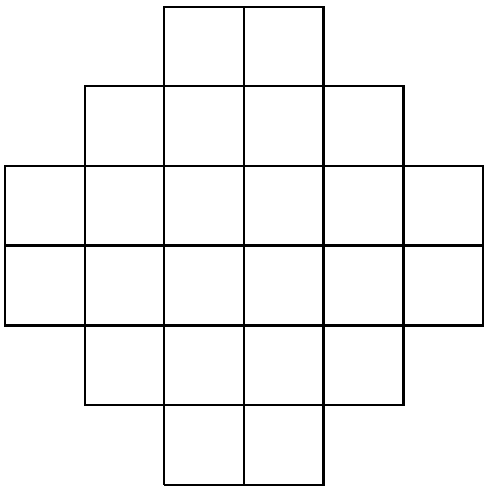
BOX. The formula for the number $N(2m, 2n)$ of domino tilings of a $2m \times 2n$ chessboard:

$$N(2m, 2n) = 4^{mn} \prod_{s=1}^{m} \prod_{t=1}^{n} \left(\cos^2 \frac{s\pi}{2m+1} + \cos^2 \frac{t\pi}{2n+1} \right).$$

Although the formula for the number of domino tilings of a chessboard is rather complicated, there is a variant of the chessboard for which a very simple formula for the number of domino tilings exists. This new board

is called an *Aztec diamond*, and was introduced by Noam David Elkies (b. 1966), Gregory John Kuperberg (b. 1967), Michael Jeffrey Larsen (b. 1962), and James Gary Propp (b. 1960). Their work has stimulated a flurry of activity on exact and approximate enumeration of domino tilings, as well as related questions such as the appearance of a "typical" domino tiling of a given region.

The Aztec diamond AZ_n of order n consists of two squares in the first row, four squares in the second row beginning one square to the left of the first row, six squares in the third row beginning one square to the left of the second row, etc., up to $2n$ squares in the nth row. Then reflect the diagram created so far about the bottom edge and adjoin this reflected diagram to the original. For instance, the Aztec diamond AZ_3 looks as follows:



Let $\mathrm{az}(n)$ be the number of domino tilings of the Aztec diamond AZ_n. For instance, AZ_1 is just a 2×2 square, which has two domino tilings (both dominos horizontal or both vertical). Hence $\mathrm{az}(1) = 2$. It's easy to compute by hand that $\mathrm{az}(2) = 8$, and a computer reveals that $\mathrm{az}(3) = 64 = 2^6$, $\mathrm{az}(4) = 1024 = 2^{10}$, $\mathrm{az}(5) = 32768 = 2^{15}$, etc. The evidence quickly becomes overwhelming for the conjecture that

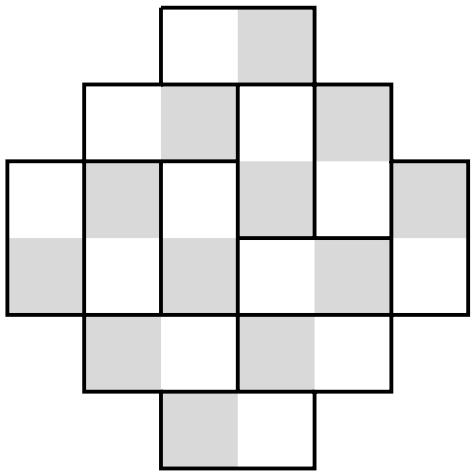
$$\mathrm{az}(n) = 2^{\frac{1}{2}n(n+1)}. \tag{24}$$

It is rather mysterious why Aztec diamonds seem to be so much more nicely

behaved regarding their number of domino tilings than the more natural $m \times n$ chessboards.

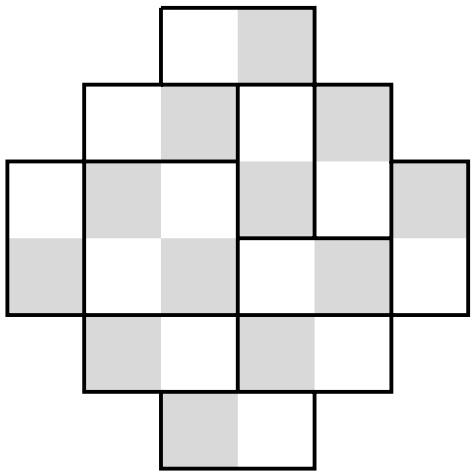
A proof of the conjecture (24) is the main result of Elkies *et al.* mentioned above. They gave four different proofs, showing the surprising connections between Aztec diamonds and various other branches of mathematics. (For instance, it is not a coincidence that $2^{\frac{1}{2}n(n+1)}$ is the degree of an irreducible representation of the group $\mathrm{GL}(n+1, \mathbb{C})$.) Of course a combinatorialist would like to see a purely combinatorial proof, and indeed Elkies *et al.* gave such proofs. Other combinatorial proofs have been since given by Ciucu and Propp. We will sketch the fourth proof of Elkies *et al.*, called a proof by *domino shuffling*. The domino shuffling procedure we describe will seem rather miraculous, and there are many details to verify to see that it actually works as claimed. Nevertheless, we hope that our brief description will take some of the mystery out of equation (24).

We first color the squares of the Aztec diamond AZ_n black and white in the usual chessboard fashion, with the first (leftmost) square in the top row colored *white*. Here is a tiling of AZ_3 with the chessboard coloring shown.



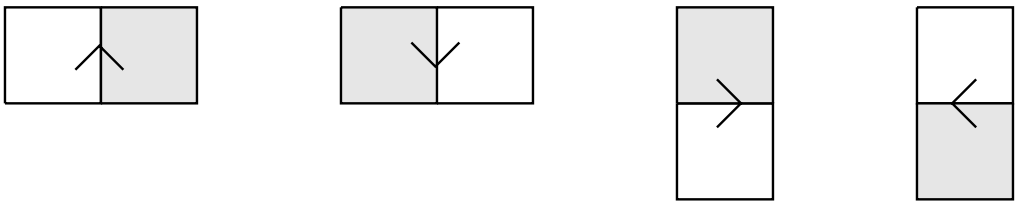
Certain pairs of dominos in the tiling will form a 2×2 square with the top left square colored black. Remove all such pairs of dominos (if any exist).

For the tiling of AZ_3 shown above there is one such pair, and after removing it we get the following tiling:

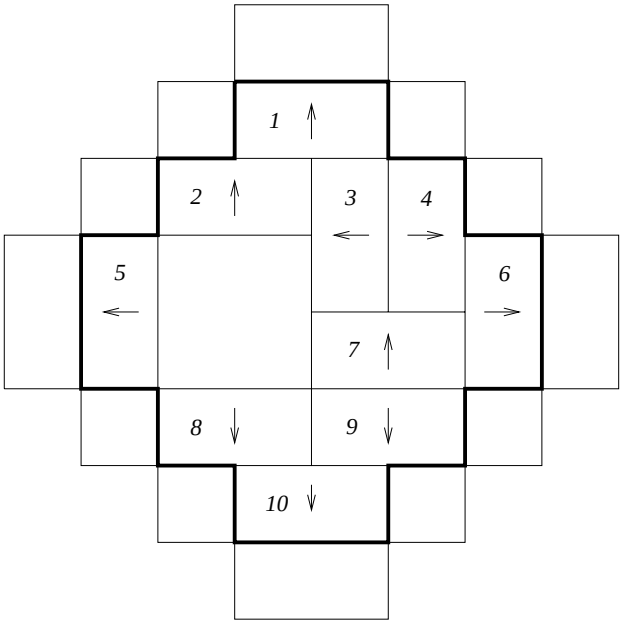


Let us call a tiling T of AZ_n with the 2×2 squares removed as just described a *reduced tiling* of AZ_n, and call T the *reduction* of the original (complete) tiling. Note that if we remove k 2×2 squares from a complete tiling to get a reduced tiling, then there are 2^k ways to tile the 2×2 holes. (Each hole can be tiled either by two horizontal or two vertical dominos.) In other words, given a reduced tiling T of AZ_n with k 2×2 holes, there are 2^k corresponding complete tilings of AZ_n whose reduction is T.

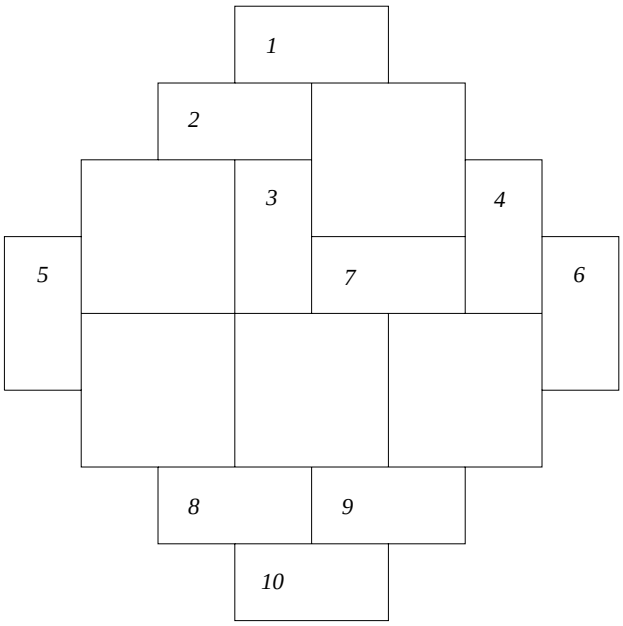
Consider a reduced tiling of AZ_n. Each domino will have one white square and one black square. There are four possible colorings and orientations of a domino, shown in the illustration below. With each of these four possible colored dominos we associate a direction: up, down, right, and left, as indicated below by an arrow.



We can enlarge the Aztec diamond AZ_n to AZ_{n+1} by adding squares around the boundary. Add one square at the beginning and one square at the end of each row, and two squares at the top and bottom. The next illustration shows the earlier reduced tiling of AZ_3, with an arrow placed on each domino according to its coloring and orientation, and the boundary of new squares to give AZ_4. We have also numbered each domino for later purposes.



Now move each domino one unit in the direction of its arrow. This is the *shuffling* operation referred to in the name "domino shuffling." Let k be the number of 2×2 squares removed before shuffling. It can be shown that (a) the dominos do not overlap after shuffling, and (b) the squares of AZ_{n+1} that are not covered by dominos can be uniquely covered with exactly $n + k + 1$ 2×2 squares. The next figure shows the dominos after shuffling (with the same numbers as before), together with the leftover five 2×2 squares (holes).



We now complete the partial tiling of AZ_{n+1} to a complete tiling by putting two dominos in each 2×2 hole. Since there are two ways to tile a 2×2 square, there are 2^{n+k+1} ways to tile all $n+k+1$ of the 2×2 squares. Therefore we have associated 2^{n+k+1} tilings of AZ_{n+1} with each k-hole reduced tiling of AZ_n. The amazing fact is that every tiling of AZ_{n+1} occurs exactly once in this way! In other words, given a tiling of AZ_{n+1}, we can reconstruct which of the dominos were shuffled from a reduced tiling of AZ_n and thus also the $n+k+1$ 2×2 holes that were left over. Since every k-hole reduced tiling T of AZ_n is the reduction of 2^k complete tilings of AZ_n, and since T corresponds to 2^{n+k+1} tilings of AZ_{n+1}, we obtain the recurrence

$$\mathrm{az}(n+1) = 2^{n+1}\mathrm{az}(n).$$

The unique solution to this recurrence satisfying $\mathrm{az}(1) = 2$ is easily seen (for instance by mathematical induction) to be

$$\mathrm{az}(n) = 2^{\frac{1}{2}n(n+1)},$$

proving equation (24).

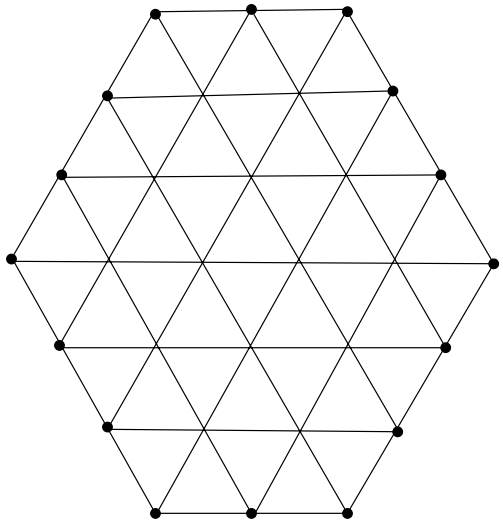


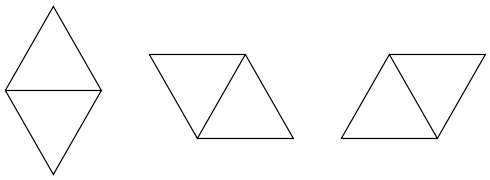
Figure 6: The hexagonal board $H(2,3,3)$

8 Tilings and plane partitions.

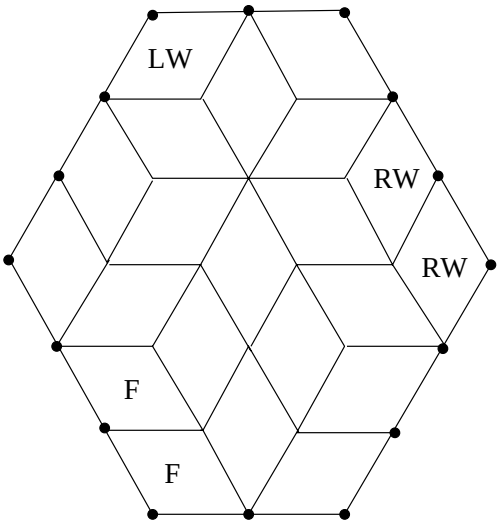
We have encountered several examples of unexpected connections between seemingly unrelated mathematical problems. This is one of the features of mathematics that makes it so appealing to its practitioners. In this section we discuss another such connection, this time between tilings and plane partitions. Other surprising connections will be treated in later sections.

The tiling problem we will be considering is very similar to the problem of tiling an $m \times n$ chessboard with dominos. Instead of a chessboard (whose shape is a rectangle), we will be tiling a hexagon. Replacing the squares of the chessboard will be equilateral triangles of unit length which fill up the hexagon, yielding a "hexagonal board." Let $H(r,s,t)$ denote the hexagonal board whose opposite sides are parallel and whose side lengths (in clockwise order) are r,s,t,r,s,t. Thus opposite sides of the hexagon have equal length just like opposite sides of a rectangle have equal length. Figure 6 shows the hexagonal board $H(2,3,3)$ with its 42 equilateral triangles. In general, the hexagonal board $H(r,s,t)$ has $2(rs+rt+st)$ equilateral triangles.

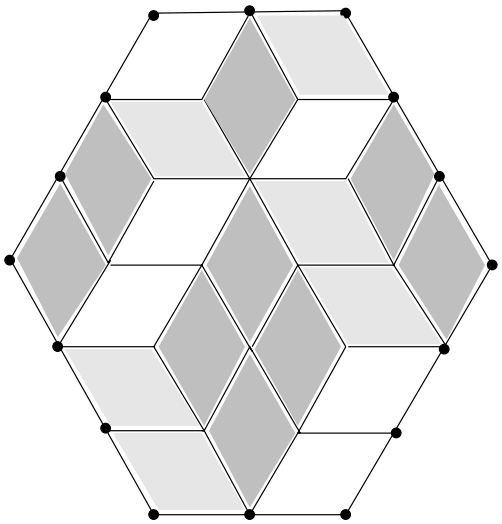
Instead of tiling with dominos (which consist of two adjacent squares), we will be tiling with pieces which consist of two adjacent equilateral triangles. We will call these pieces simply *rhombi*, although they are really only special kinds of rhombi. Thus the number of rhombi in a tiling of $H(r,s,t)$ is $rs + rt + st$. The rhombi can have three possible orientations (compared with the two orientations of a rectangle):



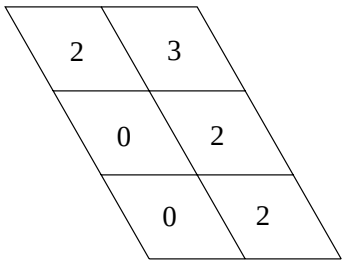
Here is a typical tiling of $H(2,3,3)$



This picture gives the impression of looking into the corner of an $r \times s \times t$ box in which cubes are stacked. The brain will alternate between different interpretations of this cube stacking. To be definite, we have labelled by F the floor, by LW the left wall, and by RW the right wall. Shading the rhombi according to their orientation heightens the impression of a cube stacking, particularly if the page is rotated slightly counterclockwise:



Regarding the floor as a 3×2 parallelogram filled with six rhombi, we can encode the cube stacking by a 3×2 array of numbers which tell the number of cubes stacked above each floor rhombus:



Rotate this diagram 45° counterclockwise, erase the rhombi, and "straighten out," giving the following array of numbers:

$$\begin{array}{ccc} 3 & 2 & 2 \\ 2 & 0 & 0 \end{array}.$$

This array is nothing more than a plane partition whose number of rows is at most r, whose number of columns is at most s, and whose largest part is at most t (where we began with the hexagonal board $H(r, s, t)$)! This

correspondence between rhombic tilings of $H(r,s,t)$ and plane partitions with at most r rows, at most s columns, and with largest part at most t is a bijection. In other words, given the rhombic tiling, there is a unique way to interpret it as a stacking of cubes (once we agree on what is the floor, left wall, and right wall), which we can encode as a plane partition of the desired type. Conversely, given such a plane partition, we can draw it as a stacking of cubes which in turn can be interpreted as a rhombic tiling.

An immediate corollary of the amazing correspondence between rhombic tilings and plane partitions is an explicit formula for the number $N(r,s,t)$ of rhombic tilings of $H(r,s,t)$. For this number is just the number of plane partitions with at most r rows, at most s columns, and with largest part at most t. If we set $x = 1$ in the left-hand side of MacMahon's formula (14) then it follows that we just get $N(r,s,t)$. If we set $x = 1$ in the right-hand side then we get the meaningless expression $0/0$. However, if we write

$$[i] = 1 - x^i = (1-x)(1 + x + \cdots + x^{i-1}),$$

then the factors of $1 - x$ cancel out from the numerator and denominator of the right-hand side of (14). Therefore substituting $x = 1$ is equivalent to replacing $[i]$ by the integer i, so we get the astonishing formula

$$N(r,s,t) =$$

$$\frac{(1+t)(2+t)^2\cdots(r+t)^r(r+1+t)^r\cdots(s+t)^r(s+1+t)^{r-1}(s+2+t)^{r-2}\cdots(s+r-1+t)}{1\cdot 2^2\cdot 3^3\cdots r^r(r+1)^r\cdots s^r(s+1)^{r-1}(s+2)^{r-2}\cdots(s+r-1)}.$$

9 Combinatorics and Topology.

On first acquaintance combinatorics may seem to have a somewhat different "flavor" than the mainstream areas of mathematics, due mainly to what mathematicians call "discreteness." Nevertheless, combinatorics is fortunate to have many beautiful and fruitful links with older and more established areas, such as algebra, geometry, probability and topology. We will now move on to discuss one such connection, perhaps the most surprising one, namely that with topology. First, however, let us say a few words about what mathematicians mean by discreteness.

In mathematics the words "continuous" and "discrete" have technical meanings that are quite opposite. Typical examples of continuous objects are curves and surfaces in 3-space (or, suitably generalized, in higher-dimensional spaces). A characteristic property is that each point on such an object is surrounded by some "neighborhood" of other points, containing points that are in a suitable sense "near" to it. The area within mathematics that deals with the study of continuity is called *topology*. The characteristic property of discrete objects, on the other hand, is that each point is "isolated" — there is no concept of points being "near." Combinatorics is the area that deals with discreteness in its purest form, particularly in the study of finite structures of various kinds.

Several fascinating connections between the continuous and the discrete are known in mathematics — in algebra, geometry and analysis. A quite recent development of this kind, the one we want to talk about here, is that ideas and results from topology can be put to use to solve certain combinatorial problems. We will soon exemplify this with two problems coming from computer science. However, first we will discuss in greater detail the connection between topology and combinatorics that will be used.

Let us take as our example of a topological space the *torus*, a 2-dimensional surface that is well known in ordinary life in the form of an inner-tube, or as the surface of a doughnut (see Figure 7).

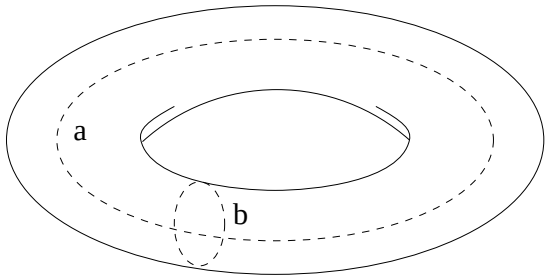


Figure 7: The torus

There is a way to "encode" a space such as the torus into a finite set system, called a *triangulation*. It works as follows. Draw (curvilinear) triangles on the torus so that each edge of a triangle is also the edge of some other

triangle, and the two endpoints of each edge are not the pair of endpoints of any other edge. The triangles should cover the torus so that each point on the torus is in exactly one of the triangles, or possibly in an edge where two triangles meet or at a corner where several triangles meet. We can think of this as cutting the rubber surface of an inner tube into small triangular pieces. Figure 8 shows one way to do this using 14 triangles. In this figure the torus is cut up and flattened out — to get back the original torus one has to roll this flattened version up and glue together the two sides marked 1-2-3-1, and then wrap around the cylinder obtained and glue together the two end-circles marked 1-4-5-1. Note that the two circles 1-2-3-1 and 1-4-5-1 in Figure 8 correspond to the circles marked **a** and **b** that are drawn with dashed lines on the torus in Figure 7.

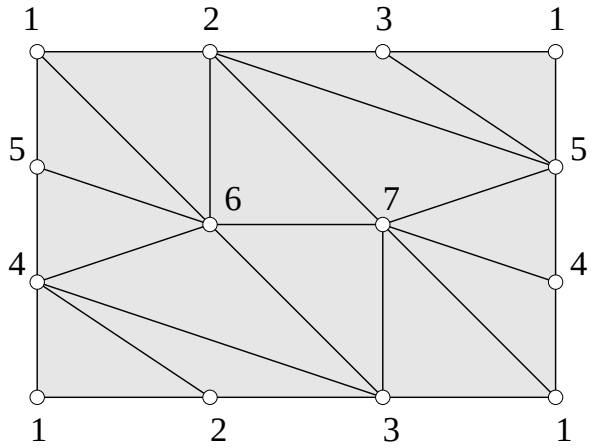


Figure 8: A triangulated torus

Having thus cut the torus apart we now have a collection of 14 triangles. The corners in Figure 8 where triangles come together are called *vertices*, and we can represent each triangle by its 3 vertices. Thus each one of our 14 triangles is replaced by a 3-element subset of {1,2,3,4,5,6,7}. For instance, {1,2,4} and {3,4,6} denote two of the triangles. The full list of all 14 triangles is

$$\begin{array}{lllllll} 124 & 126 & 135 & 137 & 147 & 156 & 234 \\ 235 & 257 & 267 & 346 & 367 & 456 & 457 \end{array} \tag{25}$$

A family of subsets of a finite set which is closed under taking subsets (i.e., if A is a set in the family and B is obtained by removing some elements from A then also B is in the family) is called a *simplicial complex* . Thus, our fourteen 3-element sets and all their subsets form a simplicial complex.

An important fact is that just knowing the simplicial complex — a finite set system — we can fully reconstruct the torus! Namely, knowing the 14 triples we can manufacture 14 triangles with vertices marked in corresponding fashion and then glue these triangles together according to the blueprint of Figure 8 (using the vertex labels) to obtain the torus. To imagine this you should think of the triangles as being flexible (e.g., made of rubber sheet) so that there are no physical obstructions to their being bent and glued together. Also, the torus obtained may be different in size or shape from the original one (smaller, larger, deformed), but these differences are irrelevant from the point of view of topology.

To sum up the discussion: The simplicial complex coming from a triangulation is *a complete encoding* of the torus as a topological object. Every property of the torus that topology can have anything to say about is also a property of this finite set system!

Why would topologists want to use such an encoding? The main reason is that they are interested in computing certain so called *invariants* of topological spaces, such as the "Betti numbers" which we will soon comment on. The spaces they consider (such as the torus) are geometric objects with infinitely many points, on which it is usually hard to perform concrete computations. An associated simplicial complex, on the other hand, is a finite object which is easily adapted to computation (except possibly for size reasons). Topological invariants depend only on the space in question, but their computation may depend on choosing a triangulation or other "combinatorial decomposition". The part of topology that develops this connection is known as *combinatorial topology*. It was initiated by the great French mathematician Jules Henri Poincaré (1854–1912) in the last years of the 1800's and greatly developed in the first half of this century. Eventually the subject took on a more and more algebraic flavor and in the 1940's the area changed name to *algebraic topology*.

The *Betti numbers* of a space are topological invariants that can be said

to measure the number of "independent holes" of various dimensions. It is impossible to give the full technical definition within the framework of this article. Let it suffice to say that the definition depends on certain algebraic constructions and to give some examples. If T is a d-dimensional topological space then there are $d+1$ Betti numbers

$$\beta_0(T), \beta_1(T), ..., \beta_d(T),$$

which are nonnegative integers. Once we have a triangulation of a topological space the computation of Betti numbers is a matter of some very simple (in principle) linear algebra.

For instance, the d-dimensional sphere has Betti numbers $(0, ..., 0, 1)$, reflecting the fact that it has exactly one d-dimensional "hole" (its interior) and no holes of other dimensions. The torus has Betti numbers $(0, 2, 1)$ because there are two essentially different 1-dimensional holes (corresponding to the circles **a** and **b** in Figure 7) and one 2-dimensional hole (the interior). Note that the two circles **a** and **b** are genuine "holes" in the sense that they cannot be continuously deformed to single points within the torus, and that they are "different" holes since one cannot be continuously deformed into the other.

The concept of a 0-dimensional hole is perhaps not so clear on an intuitive level, but having $\beta_0 = 0$ means that the space hangs together in one piece (is connected), and in general $\beta_0(T)+1$ is the number of connected components of the space T. (Note to specialists: Our $\beta_i(T)$'s are really the *reduced* Betti numbers of T, differing from the "ordinary" Betti numbers only in that $\beta_0(T) + 1$ rather than $\beta_0(T)$ is the number of connected components of T.)

We have seen that finite set systems are of use in topology as encodings of topological spaces. But the connection between topological spaces and simplicial complexes opens up a two-way street. What if the mathematics we are doing deals primarily with finite set systems, as is often the case in combinatorics? For instance, say that a combinatorial problem we are dealing with involves the fourteen 3-element sets listed in (25). Could the properties of the associated topological space — the torus — be of any relevance? For instance, could its Betti numbers (measuring the number of "holes" in the space) have something useful to say about the set system as such? We

will show that this may indeed be the case, and this is in fact one of the cornerstones for the "topological method" in combinatorics.

The idea to use topological reasoning in combinatorics is quite old but had a somewhat unfortunate start. It seems to have first occurred in connection with a famous problem of Euler. The following configuration is called a *Graeco-Latin square of order* n: An $n \times n$-matrix of ordered pairs (a, b) of numbers a and b from $1, 2, ..., n$ such that the first entries a are distinct in every row and column, the second entries b are distinct in every row and column, and all n^2 possible pairs occur. For instance, here is a Graeco-Latin square of order 3:

$$\begin{array}{ccc} 1,1 & 2,2 & 3,3 \\ 2,3 & 3,1 & 1,2 \\ 3,2 & 1,3 & 2,1. \end{array}$$

Euler stated without proof in his paper "Recherches sur une espèce de carrés magique" from 1782 that such configurations cannot exist for $n = 6, 10, 14, 18, ...$ His claim was proven correct for $n = 6$ by Gaston Tarry (1843–1913) in 1901. In 1922 Harris F. MacNeish (18??–19??) published a paper in *Annals of Mathematics* supposedly proving Euler's claim for all remaining values of n. Unfortunately his argument, which was based on topology, was incorrect. In fact, subsequent research has shown that Euler's claim itself is false, except for the single case of $n = 6$!

After this unsuccessful start it took a long time before the idea resurfaced — topological proofs for combinatorial results have come to the fore only in the last two decades. Let us now go on to see a couple of concrete examples.

BOX: Borsuk and combinatorics

The Polish mathematician Karol Borsuk (1905–1982) made some fundamental contributions to the early development of topology. In 1933 he

published a paper entitled (in translation) "Three theorems about the n-dimensional euclidean sphere". That paper contains, among other wonderful things, a famous theorem and a famous open problem. Let us state them (within this box we will assume familiarity with the topological terminology used).

Borsuk's Theorem. *If the k-dimensional sphere is covered by $k+1$ closed sets, then one of these sets must contain a pair of antipodal points.*

Borsuk's Problem. *Is it true that every set of diameter one in k-dimensional real space $\mathbb{R}^k$ can be partitioned into at most $k+1$ sets of smaller diameter?*

This work of Borsuk has interacted with combinatorics in a remarkable way. In 1978 László Lovász (b. 1948) solved a difficult combinatorial problem — the "Kneser Conjecture" from 1955 — by using Borsuk's theorem. Then, in 1992 the debt to topology was repaid when Jeffry Ned Kahn (b. 1950) and Gil Kalai (b. 1955) solved Borsuk's problem using some results from pure combinatorics. By stating the relevant results on the combinatorial side we hope to give a small glimpse of these interactions, which are quite unexpected.

The answer to Borsuk's problem is definitely "yes" when $k = 1$, the statement then comes down to dividing a line segment of length 1 into two shorter segments, which is clearly possible. It was also long known that the statement is true for $k = 2$ and $k = 3$, and it was generally believed that the statement is true for all dimensions k — this became known as *Borsuk's conjecture.*

It therefore came as a great surprise that the answer to Borsuk's problem is actually "no", contrary to what "everyone" had believed for nearly 60 years. But one has to go to very high dimensions ($k \approx 1,000$) to find counterexamples with the Kahn-Kalai method. The problem is still open for $k = 4$.

The combinatorial result from which the solution to Borsuk's problem follows is this 1981 theorem of Peter Frankl (b. 1953) and Richard Michael Wilson (b. 1945).

Frankl-Wilson Theorem. *Let k be a power of a prime number, and let F be a family of $2k$-element subsets of $\{1,2,\ldots,4k\}$ such that no two members of F have k elements in common. Then F has at most $2 \cdot \binom{4k-1}{k-1}$ members.*

The Kneser conjecture — now a theorem of Lovász — is the following statement:

Lovász' Theorem. *If the n-element subsets of a $(2n+k)$-element set are partitioned into $k+1$ classes, then some class will contain a pair of disjoint n-element sets.*

The details of how this conclusion is derived from Borsuk's theorem, as well as the argument for solving Borsuk's problem using the Frankl-Wilson theorem, must be left aside. See the suggested reading for further information.

---

10 Complexity of graph properties.

A major theme in theoretical computer science is to estimate the complexity of computational tasks. By "complexity" is here meant the amount of time and of computational resources needed. By constructing algorithms one shows that a task can be done in a certain number of steps. It is often the more difficult part to show that there is no "faster" way, i.e. requiring fewer steps.

Examples of this will be given in this and the following section. We begin by considering algorithms that test whether graphs have a certain given property P. For example, P could be the property of being *connected*, meaning that you can get from any node to any other node by walking along a path of edges. The left graph in Figure 9 is connected whereas the right one is disconnected, since there is no way to get from nodes 1, 2 or 3 to nodes 4 or 5.

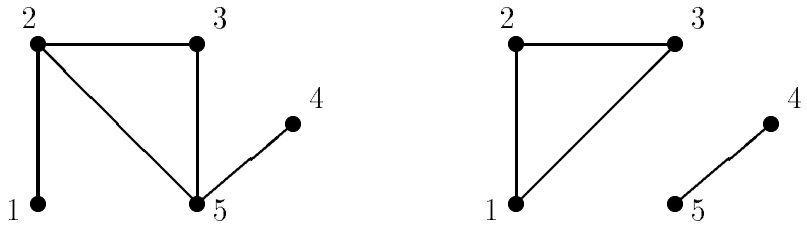


Figure 9: A connected and a disconnected graph

Connectedness is a very basic property of graphs which can be decided at a glance on small examples represented as a drawing. But say you have a graph with 1 million nodes, coming perhaps from a communications network or a chip design, which is presented only as a list of edges (adjacent pairs of nodes) — then it is not quite so clear what to do if one wants to decide whether the graph is connected, making efficient use of computational resources. Among the interesting questions one can ask is whether it is possible to decide connectedness of the graph without checking for all possible pairs of nodes (there are nearly 500 billion of them) whether they are edges of the graph or not? If this were so it could conceivably lead to valuable saving of time and resources.

A basic general question to ask then is this: For a given property P of graphs, is there some algorithm that decides for every graph G whether it has property P without knowing for every pair of nodes whether they span an edge of G or not? If this is not the case, i.e. if every P-testing algorithm must for at least some graph have complete knowledge about all its edges, then P is said to be an *evasive* property.

For instance, connectedness is an evasive property. To see this we can argue as follows. Imagine that we have a computer running a program that tests graphs for connectedness. The graphs to be tested, whose nodes we may assume are labeled $1, 2, ..., n$, are presented to the computer in the form of an $n \times n$ upper-triangular matrix of zeros and ones, with a 1 entry in row i and column j, for $i < j$, if (i, j) is an edge of the graph and a 0 entry otherwise.

For instance, here are the matrices representing the graphs in Figure 9:

$$\begin{array}{ccccc} * & 1 & 0 & 0 & 0 \\ & * & 1 & 0 & 1 \\ & & * & 0 & 1 \\ & & & * & 1 \\ & & & & * \end{array} \qquad\qquad \begin{array}{ccccc} * & 1 & 1 & 0 & 0 \\ & * & 1 & 0 & 0 \\ & & * & 0 & 0 \\ & & & * & 1 \\ & & & & * \end{array}$$

The computer is allowed to inspect only one entry of this matrix at a time, and what we want to show is that for some graph it must in fact inspect all of them. To find such a worst-case graph we can imagine playing the following game with the computer. Say that instead of deciding on the graph in advance, we write the zeros and ones (specifying its nonedges and edges) into the matrix only at the last moment, as the computer demands to inspect them. Say furthermore that we do this according to the following strategy (designed to keep the computer making as many queries as possible): When the computer goes to inspect the (i, j) entry of the matrix (according to whatever algorithm it is using), then

- write 0 into position (i, j) if it is not possible to conclude from the partial information known to the computer at that time — including this last 0 — that the graph is disconnected,
- otherwise, write 1 into position (i, j).

It is an elementary but somewhat tricky argument to show that this strategy will force the computer to inspect all entries of the matrix before it can decide whether the corresponding graph is connected or not. We will outline a proof by induction.

The crucial step will be to prove the following statement:
Suppose that at some stage 1 *is written into position* (i, j)*. Let* A *be the set of nodes that are at that stage connected to* i *by* 1*-marked edges, and let* B *be the set of nodes connected to* j *by* 1*-marked edges. Then all possible edges between nodes in* $A \cup B$ *have been inspected at that stage.*
(Clarification: "at that stage" refers to the configuration existing at the time when 1 is assigned to the position/edge (i, j), namely, at that time some

other edges have already been inspected and are marked with 0 or 1, while the remaining have not yet been inspected.)

Note that $A \cap B = \emptyset$, and that $|A \cup B| \geq 2$ since $i \in A$ and $j \in B$. The statement is clearly true if $|A \cup B| = 2$, and we proceed by induction on $|A \cup B|$, that is, the number of elements of $A \cup B$. Suppose that $|A \cup B| > 2$. Since 1 (and not 0) is written into position (i, j) that means that there is some partition $C \cup D = \{1, 2, \ldots, n\}$ into nonempty disjoint subsets C and D such that $i \in C$, $j \in D$ and all possible edges $\{c, d\}$ with $c \in C$, $d \in D$ and $\{c, d\} \neq \{i, j\}$ are already marked with 0. Clearly, we must have $A \subseteq C$ and $B \subseteq D$, so in particular all edges between a node in A and a node in B have already been inspected. Also, all edges between two nodes both in A have by the induction assumption been inspected, and similarly for B. This covers all possible edges between nodes in $A \cup B$ and the claim follows.

Suppose now that connectedness/disconnectedness can be decided after inspection of k matrix entries, and that k is the minimum such number. According to our strategy for writing 0 or 1, the outcome can never be that the graph is disconnected. Also, if the kth entry is 0 and the graph is connected we have a contradiction, since then the information needed to conclude connectedness would have been available already before the kth entry was inspected. So, the kth entry is 1, and since the conclusion is that the graph is connected the claim above implies that all other entries have already been inspected before the kth one. This proves that connectedness is an evasive graph property.

It has been decided for many graph properties whether they are evasive. It turns out that among the evasive ones are many that are *monotone*, meaning that if the property holds for some graph then it will also hold if more edges are added. For instance, connectedness is an example of a monotone property. Mounting evidence from work in the late 1960's by several researchers led to the following conjecture.

Evasiveness Conjecture. *Every monotone nontrivial graph property is evasive.*

By "nontrivial" is here meant that there is at least one graph that has the

property and one that doesn't. Since monotonicity is usually very easy to verify whereas evasiveness is not, this conjecture — if true — would simplify deciding evasiveness for many graph properties. Tedious case-by-case arguments, such as the ones we carried out for the property of connectedness, would not be needed.

The best general result known to date on this topic is the following theorem of Jeffry Kahn, Michael Ezra Saks (b. 1956) and Dean Grant Sturtevant (b. 1955) from 1984:

Kahn–Saks–Sturtevant Theorem. *The evasiveness conjecture is true for graphs on p^k nodes, for any prime number p and integer $k \geq 1$.*

This verifies the conjecture for infinitely many values of n, the number of nodes, but leaves it open when n is the product of at least two distinct primes. Thus, the smallest values of n left open are $6, 10, 12, 14, 15, ...$; however the case of $n = 6$ was also verified by Kahn *et al.* The general conjecture remains open, beginning with the case $n = 10$.

The proof of Kahn *et al.* makes surprising use of topology. The key idea is to view a monotone graph property for graphs on n vertices as a simplicial complex with a high degree of symmetry, to whose associated space a topological fixed point theorem can be applied. Here is how.

We will keep in mind some particular monotone graph property P and consider graphs on the nodes $1, 2, ..., n$. Such a graph is specified by the pairs (i, j) of nodes that are connected by an edge. Let us take the set of these pairs as the ground set for a set family Δ_n^P, whose members are the edge-sets of graphs *not* having property P. The set family Δ_n^P is closed under taking subsets, since monotonicity implies that removal of edges from a graph that doesn't have property P cannot produce a graph having that property.

Let us illustrate the idea for the case $n = 4$, taking as our monotone property connectedness. There are 6 possible edges in a graph on the nodes $1, 2, 3, 4$; see Figure 10.

The simplicial complex Δ_4^{conn} of disconnected graphs on four vertices is shown in Figure 11.

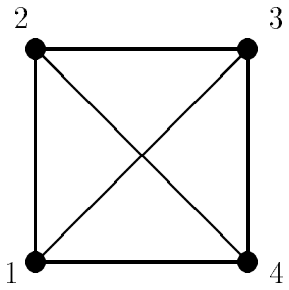


Figure 10: The 6 edges spanned by 4 nodes

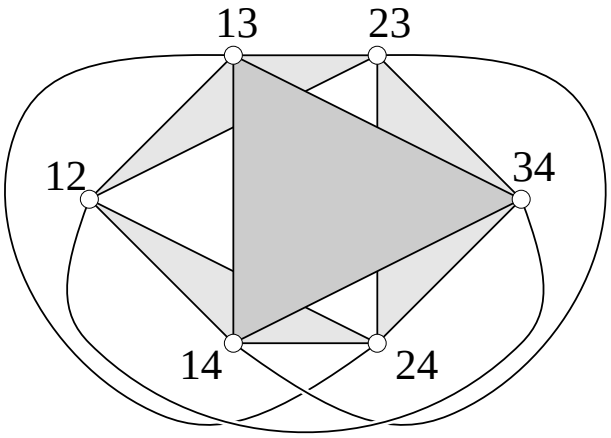
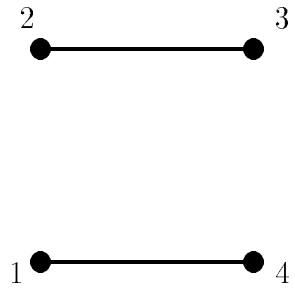
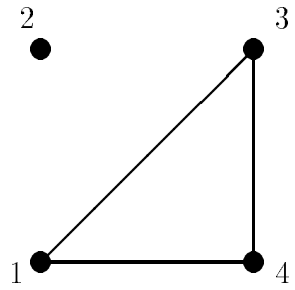


Figure 11: The complex of disconnected graphs on 4 nodes

In the rubber-sheet model depicted it consists of 4 triangles and 3 edges (curved line segments) glued together. To understand this picture the reader should think how to translate the vertices, edges and triangles of Δ_4^{conn} into disconnected graphs. For instance, the edge between 14 and 23 in Figure 11 corresponds to the disconnected graph



and the triangle with vertices 13, 14 and 34 corresponds to the disconnected graph



Observe in Figure 11 that the space represented by the complex Δ_4^{conn} has many "holes" — in the terminology used before this means that Δ_4^{conn} has some nonzero Betti numbers. It turns out to be a general fact, not hard to prove, that if the property P is *not* evasive then Δ_n^P is *acyclic*, meaning that all Betti numbers of Δ_n^P are equal to zero.

There are several theorems in topology to the effect that certain mappings f of an acyclic space to itself must have *fixed points*, i.e. points x such that $f(x) = x$. The best known one — one of the classics of topology — is Luitzen Egbertus Jan Brouwer's (1881–1966) theorem from 1904, which says that every continuous mapping of an n-dimensional ball to itself has a fixed point. The one needed for the present application is a fixed point theorem of Robert Oliver (b. 1949) from 1975, which (stripped of some technical details) says that for certain groups G of symmetry mappings of an acyclic simplicial complex Δ to itself there is a point x in the associated space such that $f(x) = x$ for *all* mappings f in G.

The complex Δ_n^P of a monotone graph property has a natural group of

symmetries, namely the symmetric group S_n of all permutations of the set of nodes $1, 2, ..., n$. Permuting the nodes amounts to a relabeling (node i gets relabeled $f(i)$, etc.), and it is clear that such a relabeling will not affect whether the graph in question has property P. Therefore every permutation of $1, 2, ..., n$ induces a self-symmetry of the complex Δ_n^P of graphs not having property P.

The pieces needed for the proof of Kahn *et al.* are now at hand. Here is how they argued.

Suppose P is a monotone property for graphs on n nodes that is *not* evasive. Then, as was already mentioned, the associated complex Δ_n^P is acyclic. If furthermore $n = p^k$ then due to some special properties of prime-power numbers (the existence of finite fields) one can construct a subgroup G of S_n having the special properties needed for Oliver's fixed point theorem. Hence there is a point x in the space associated to Δ_n^P such that $f(x) = x$ for all permutations f in G. However, this means that there is a nonempty set A in the complex Δ_n^P (that is, a graph with edge-set A not having property P) such that $f(A) = A$ for all f in G. Since G is *transitive* (meaning that if u and v are two vertices of Δ_n^P then $u = f(v)$ for some mapping f in G), A must consist of *all* vertices of Δ_n^P; that is, A is the complete graph. We have obtained that the complete graph on nodes $1, 2, \ldots, n$ does not have property P, and since P is monotone that means that *no* graph on $1, 2, \ldots, n$ can have property P, so P is trivial.

The argument shows that for monotone graph properties P on a prime-power number of nodes *nonevasive* implies *trivial*, or which is logically the same: *nontrivial* implies *evasive*.

Viewing a graph property (such as connectedness) as a simplicial complex and submitting it to topological study may seem strange. One can wonder if this point of view is of any value other than — by remarkable coincidence — for the evasiveness conjecture. It has recently become clear that this is indeed the case. Namely, the complexes Δ_n^{conn} of disconnected graphs on n vertices have arisen and play a role in the work of Victor Anatol'evich Vassiliev (b. 1956) on knot invariants. Also some other monotone graph properties have naturally presented themselves as simplicial complexes in other mathematical contexts.

11 Complexity of sorting and distinctness.

The following is a very basic situation studied in complexity theory. A sequence of real numbers $x_1, x_2, \ldots, x_n$ is given. A computer is asked to decide some property of the sequence or to restructure it using only pairwise comparisons. This means that the computer is allowed to learn about the input sequence only by inspecting pairs x_i and x_j and deciding whether $x_i > x_j$, $x_i < x_j$ or $x_i = x_j$. The question then is: How many such comparisons must the computer make in the worst case when using the best algorithm? This number, as a function of n, is called the *complexity* of the problem.

The following notation is used to state such results. To say that the complexity is "$\sim f(n)$", where $f(n)$ is some function, means that there exist positive constants c_1 and c_2 such that

$$c_1 \cdot f(n) < \text{complexity} < c_2 \cdot f(n).$$

While this notation doesn't give the exact numerical value of the complexity (which is often hard, if not impossible, to determine) it reveals its order of growth, which is what is usually taken as the main indication if a problem is computationally easy or hard. In the following formulas the function "$\log n$" will frequently appear. Readers not familiar with the logarithm function can take this to mean roughly the number of digits needed to write the number n in base 10, so that for instance $\log 1997 \approx 4$.

Here are some basic facts.

1. **Sorting.** *To rearrange the n numbers increasingly $x_{i_1} \leq x_{i_2} \leq \cdots \leq x_{i_n}$ requires $\sim n \log n$ comparisons.*

2. **Median.** *To find j such that x_j is "in the middle", meaning that half of the x_i's are less than or equal to x_j and half of the x_i's are greater than or equal to x_j, requires $\sim n$ comparisons. In fact, it has been shown that $2n$ comparisons are needed and that $3n$ comparisons suffice.*

3. **Distinctness.** *To decide whether all entries x_i are distinct, that is whether $x_i \neq x_j$ when $i \neq j$, requires $\sim n \log n$ comparisons.*

The problem we wish to discuss, which was only recently resolved, is a generalization of the distinctness problem. Namely,

k-equal problem: *for $k \geq 2$, decide whether some k entries are equal, that is, can we find $i_1 < i_2 < \cdots < i_k$ such that $x_{i_1} = x_{i_2} = \cdots = x_{i_k}$?*

For example, are there nine equal entries in the following list of numbers?

24791374685848713955196742346159463314867729559243628541178369725819 32

Answer: Yes, there are nine copies of the number "4". Are there ten equal entries? Answer: No. If pairwise comparisons are the only type of operation allowed, how should one go about settling these questions in an efficient manner, and how many comparisons would be needed?

Here are a few immediate observations. If $k = 2$ the problem reduces to the distinctness problem, so the complexity is $\sim n \log n$. At the other end of the scale, if $k > \frac{n}{2}$ the complexity is $\sim n$, because we can argue as follows. The median x_j can be found using $3n$ comparisons. If there are $k > \frac{n}{2}$ equal entries then the median must be one of them. Thus after comparing x_j with the other $n-1$ entries x_i we gain enough information to conclude whether there are some k entries that are equal. This procedure requires in all $4n-1$ comparisons. On the other hand it is easy to see that at least $n-1$ comparisons are needed in the worst case, so there are both upper and lower bounds of the form "constant times n" to the complexity.

We have seen that the complexity of the k-equal problem decreases from $\sim n \log n$ to $\sim n$ when the parameter k grows from 2 to above $\frac{n}{2}$, so the k-equal problem seems to get easier the larger k gets. The exact form of this relationship is given in the following result from 1992 of Anders Björner (b. 1947), László Lovász and Andrew Chi-Chih Yao (b. 1946).

Theorem. *The complexity of the k-equal problem is $\sim n \log \frac{2n}{k}$.*

The upper bound is obtained via a partial sorting algorithm based on repeated median-finding. It generalizes what was described for the case $k > \frac{n}{2}$

above. We shall leave it aside.

The lower bound — proving that at least $n \log \frac{2n}{k}$ comparisons are needed (up to some constant) by *every* algorithm in the worst case — is the difficult and mathematically more interesting part. The proof uses a combination of topology and combinatorics. A detailed description would take us too far afield, but we will attempt to get some of the main ideas across.

Let us first look at the situation from a geometric point of view. Each equation $x_{i_1} = x_{i_2} = \cdots = x_{i_k}$ determines an $(n - k + 1)$-dimensional linear subspace of $\mathbb{R}^n$, the n-dimensional space consisting of all n-tuples $(x_1, x_2, \ldots, x_n)$ of real numbers x_i. The k-equal problem is from this point of view to determine whether a given point $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ lies in at least one such subspace, or — which is the same — lies in the union of all the subspaces $x_{i_1} = x_{i_2} = \cdots = x_{i_k}$.

Removal of linear subspaces disconnects $\mathbb{R}^n$. For instance, removal of a plane (a 2-dimensional subspace) cuts $\mathbb{R}^3$ into two pieces, whereas removal of a line (a 1-dimensional subspace) leaves another kind of "hole". These are precisely the kinds of holes that are measured by the topological Betti numbers (as was discussed in Section 9). Going back to the general situation, it seems clear that if *all* the subspaces $x_{i_1} = x_{i_2} = \cdots = x_{i_k}$ are removed from $\mathbb{R}^n$ then lots of holes of different dimensions will be created. This must mean that the sum of Betti numbers of $M_{n,k}$, the part of space $\mathbb{R}^n$ that remains after all these subspaces have been removed, is a large number:

$$\beta(M_{n,k}) = \beta_0(M_{n,k}) + \beta_1(M_{n,k}) + \cdots + \beta_n(M_{n,k}).$$

The idea now is that if the space $M_{n,k}$ is complicated topologically, as measured by this sum of Betti numbers, then this ought to imply that it is computationally difficult to determine whether a point $\mathbf{x}$ lies on it. This turns out to be true in the following quantitative form.

Fact 1. *The complexity of the k-equal problem is at least* $\log_3 \beta(M_{n,k})$.

Here $\log_3$ denotes logarithm to the base 3, which differs by a constant factor from the logarithm to the base 10 that was mentioned earlier.

So, now the problem has been converted into a topological one — to compute or estimate the sum of Betti numbers $\beta(M_{n,k})$. This can be done via a formula of Robert Mark Goresky (b. 1950) and Robert Duncan MacPherson (b. 1944), which expresses these Betti numbers in terms of some finite simplicial complexes associated to certain partitions. To get further we need to introduce a few more concepts from combinatorics.

We began this paper by discussing partitions of numbers, and we shall return once more to the ubiquitous concept of partitions. Here we need, however, the notion of *partitions of sets*. A partition of a finite set A is a way of breaking it into smaller pieces, namely a collection of pairwise disjoint subsets whose union is A. (None of these subsets is allowed to be empty — in other words, all the subsets have at least one element.) For instance, here are the 15 partitions of the set $\{1, 2, 3, 4\}$:

1234, 12—34, 13—24, 14—23, 1—234, 2—134, 3—124, 4—123,
12—3—4, 13—2—4, 14—2—3, 23—1—4, 24—1—3, 34—1—2,
1—2—3—4

In the following we will use $\{1, 2, \ldots, n\}$ as the ground set and for fixed k (an integer between 2 and n) consider the collection of all partitions of this set that have no parts of sizes $2, 3, \ldots, k-1$. Denote this collection by $\Pi_{n,k}$. For instance, $\Pi_{4,2}$ is the collection of *all* partitions of $\{1, 2, 3, 4\}$ (there are no forbidden parts), while $\Pi_{4,3}$ is the following subcollection (now parts of size 2 are forbidden):

1234, 1—234, 2—134, 3—124, 4—123, 1—2—3—4

There is a natural way to compare set partitions, saying that partition π is less than partition σ (written $\pi \leq \sigma$) if π is obtained from σ by further partitioning its parts. This way we get an order structure on the set $\Pi_{n,k}$, which can be illustrated in a diagram. Figure 12 shows the order diagram of $\Pi_{4,2}$ and Figure 13 shows that of $\Pi_{4,3}$.

These diagrams are to be understood so that a partition π is less than a

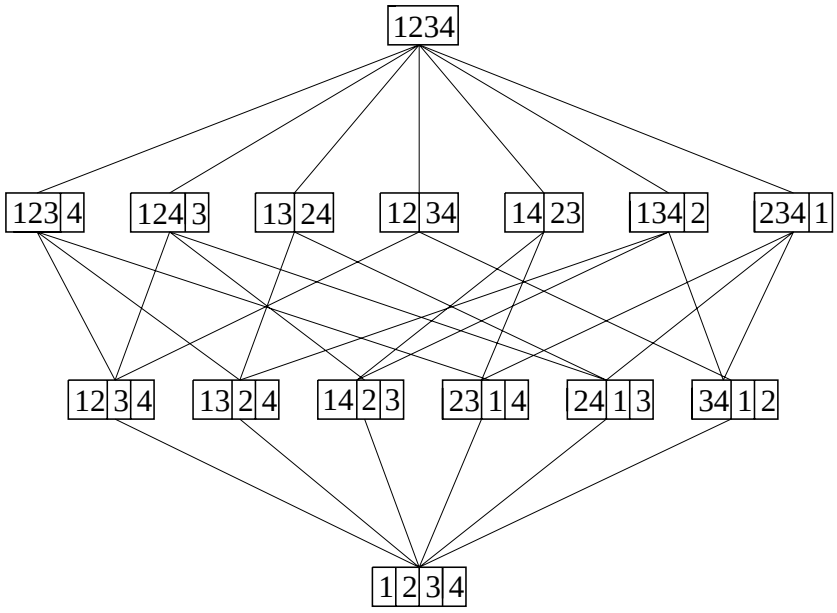


Figure 12: $\Pi_{4,2}$

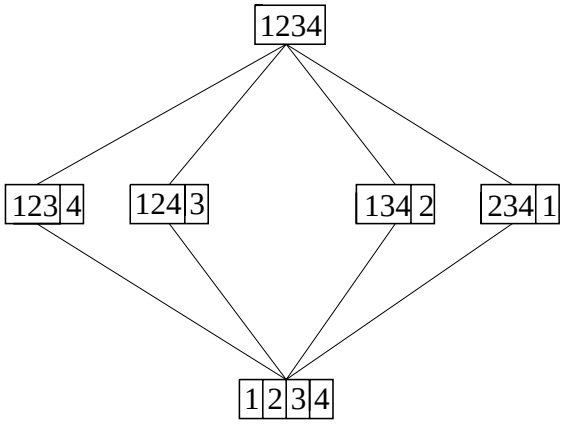


Figure 13: $\Pi_{4,3}$

partition σ if and only if there is a downward path from σ to π in the order diagram, corresponding to further breaking up of σ's parts.

Now, consider the *Möbius function* (see the **BOX**) computed over the poset $\Pi_{n,k}$. Let $\mu_{n,k}$ denote the value that the Möbius function attains at the partition with only one part, which is at the top of the order diagram. For example, computation as demonstrated in the **BOX** over the posets in Figures 12 and 13 shows that $\mu_{4,3} = 3$ and $\mu_{4,2} = -6$.

We can now return to the discussion of the k-equal problem. Where we left off was with the question of how to estimate the sum of Betti numbers $\beta(M_{n,k})$. The formula of Goresky and MacPherson mentioned earlier implies, by an argument involving among other things the topological significance of the Möbius function, the following relation:

Fact 2. $\beta(M_{n,k}) \geq |\mu_{n,k}|$.

Putting Facts 1 and 2 together, the complexity question for the k-equal problem has been reduced to the problem of showing that the combinatorially defined numbers $|\mu_{n,k}|$ grow sufficiently fast. For this we turn to the method of generating functions, already introduced in the early sections on counting number partitions. Certain recurrences for the numbers $\mu_{n,k}$ lead, when interpreted at the level of generating functions, to the following formula:

$$\exp\left(\sum_{n\geq 1} \mu_{n,k} \frac{x^n}{n!}\right) = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^{k-1}}{(k-1)!}. \tag{26}$$

To make sense of this you have to imagine inserting the series $y = \sum_{n\geq 1} \mu_{n,k} \frac{x^n}{n!}$ into the exponential series $\exp(y) = \sum_{n\geq 0} \frac{y^n}{n!}$, and then expanding in powers of x. Also, since $\mu_{n,k}$ has so far been defined only for $k \leq n$ we should mention that we put $\mu_{n,k} = 0$ for $1 < n < k$ and $\mu_{1,k} = 1$.

From this relation between the numbers $\mu_{n,k}$ and the polynomial on the right-hand-side (which is a truncation of the exponential series) we can extract the following explicit information.

Fact 3. *Let $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$ be the complex roots of the polynomial* $1 + x +$

$\frac{x^2}{2!} + \cdots + \frac{x^{k-1}}{(k-1)!}$. *Then*

$$\mu_{n,k} = -(n-1)!\ \left(\alpha_1^{-n} + \alpha_2^{-n} + \cdots + \alpha_{k-1}^{-n}\right).$$

For instance, if $k = 2$ there is only one root $\alpha_1 = -1$, and we get

$$\mu_{n,2} = (-1)^{n-1}(n-1)!.$$

Also, in this case the formula (26) specializes to

$$\exp\left(\sum_{n\geq 1}(-1)^{n-1}\frac{x^n}{n}\right) = 1 + x,$$

which is well-known to all students of the calculus in the equivalent form

$$\log(1+x) = \sum_{n\geq 1}(-1)^{n-1}\frac{x^n}{n}.$$

If $k = 3$ there are 2 roots $\alpha_1 = -1 + i$ and $\alpha_2 = -1 - i$, where $i = \sqrt{-1}$, and using some formulas from elementary complex algebra we get

$$\mu_{n,3} = -(n-1)!\ \left((-1+i)^{-n} + (-1-i)^{-n}\right) = -(n-1)!\,2^{1-\frac{n}{2}}\cos\frac{3\pi n}{4}. \quad (27)$$

We have come to a point where we know on the one hand from Facts 1 and 2 that

$$\textit{the complexity of the } k\textit{-equal problem} \geq \log_3|\mu_{n,k}|,$$

and on the other that the Möbius numbers $\mu_{n,k}$ are given in terms of the roots $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$ as stated in Fact 3. It still remains to show that the numbers $|\mu_{n,k}|$ are large enough so that $\log_3|\mu_{n,k}|$ produces the desired complexity lower bound. For this reason it comes as a chilling surprise to discover that these numbers are not always very large. In fact, formula (27) shows that

$$\mu_{n,3} = 0, \quad \text{for } n = 6, 10, 14, 18, 22, \ldots.$$

It can also be shown that $\mu_{2k,k} = 0$ for all odd numbers k.

So, we are not quite done — but almost! With a little more work it can be shown from the facts presented so far that $|\mu_{n,k}|$ is, so to say, "sufficiently large for sufficiently many n" (for fixed k). With this, and a "monotonicity argument" to handle the cases where $|\mu_{n,k}|$ itself is not large but nearby values are, it is possible to wrap up the whole story and obtain the initially stated lower bound of the form "constant times $n \log \frac{2n}{k}$".

Let us mention in closing that it is possible to work with Betti numbers the whole way, never passing to the Möbius function as described here. This route is a bit more complicated but results in a better constant for the lower bound.

BOX: The Möbius function.

The *Möbius function* is one of the most important tools of algebraic combinatorics. It assigns a very significant integer to every finite "poset". This word is an abbreviation which stands for "partially ordered set"; for simplicity we will assume that all posets considered have a bottom and a top element. Figure 14 shows a poset of eight elements with bottom element "a" and top element "h".

The Möbius function $\mu(x)$ is recursively defined for any finite poset as follows: Put $\mu(x_0) = 1$ for the bottom element x_0 of the poset, then require that

$$\mu(x) = -\sum_{y<x} \mu(y)$$

for all other elements x. This formula means that we are to define $\mu(x)$ so that when we sum $\mu(y)$ for all y less than or equal to x the resulting sum equals zero. This can clearly be done as long as one knows the values $\mu(y)$ for all elements y less than x. The reader can see how this recursive definition works by computing the Möbius function of the poset in Figure 14, starting from the bottom. We get recursively:

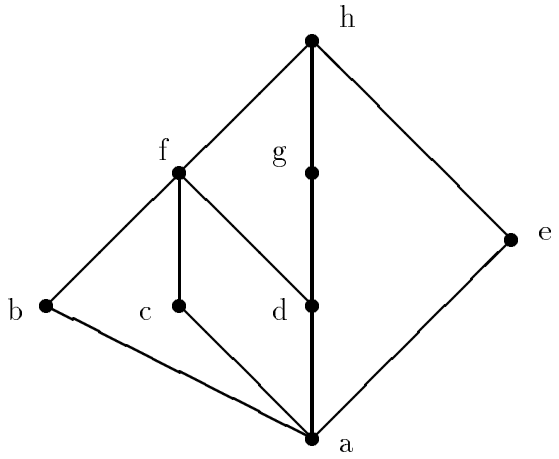


Figure 14: A small poset

$\mu(a) = 1$, by definition,
$\mu(b) = -\mu(a) = -1$,
$\mu(c) = -\mu(a) = -1$,
$\mu(d) = -\mu(a) = -1$,
$\mu(e) = -\mu(a) = -1$,
$\mu(f) = -\mu(a) - \mu(b) - \mu(c) - \mu(d) = -1 + 1 + 1 + 1 = 2$,
$\mu(g) = -\mu(a) - \mu(d) = -1 + 1 = 0$,
$\mu(h) = -\mu(a) - \mu(b) - \mu(c) - \mu(d) - \mu(e) - \mu(f) - \mu(g)$
$= -1 + 1 + 1 + 1 + 1 - 2 - 0 = 1$.

Figure 15 shows the same poset with computed Möbius function values.

The Möbius function has its origin in number theory, where it was introduced by August Ferdinand Möbius (1790–1868). (Möbius is best known to nonmathematicians for his eponymous connection with the "Möbius strip." The Möbius strip itself was well-known long before Möbius, but Möbius was one of the first persons to systematically investigate its mathematical properties.) The posets relevant to number theory are subsets of the positive integers ordered by divisibility. For instance, see the divisor diagram of the number "60" in Figure 16. A calculation based on this diagram, analogous

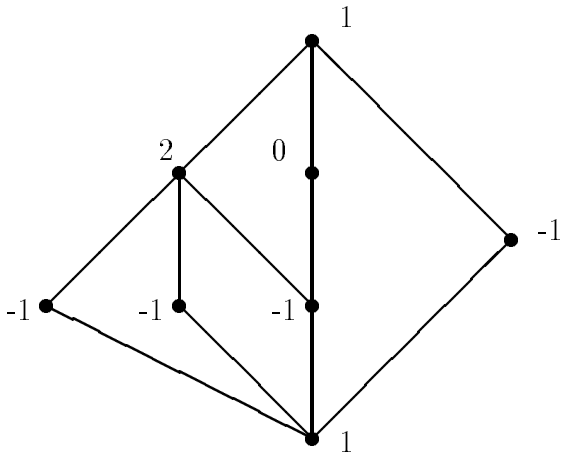


Figure 15: Values of the Möbius function

to the one we just carried out over Figure 14, will show that $\mu(60) = 0$. In the case of the classical Möbius function of number theory there is however a faster way to compute. Namely, for $n > 1$ one has that $\mu(n) = 0$ if the square of some prime number divides n, and that otherwise $\mu(n) = (-1)^k$ where k is the number of prime factors in n. Hence, for example: $\mu(60) = 0$ since $2^2 = 4$ divides 60; and $\mu(30) = -1$ since we have the prime factorization $30 = 2 \cdot 3 \cdot 5$ with an odd number of prime factors.

The Möbius function is very important in number theory. Let it suffice to mention — for those who have the background to know what we are referring to — that both the Prime Number Theorem and the Riemann Hypothesis (considered by many to be the most important unsolved problem in all of mathematics) are equivalent to statements about the Möbius function. Namely, letting $M(n) = \sum_{k=1}^{n} \mu(k)$, it is known that

$$\text{Prime Number Theorem} \iff \lim_{n\to\infty} \frac{M(n)}{n} = 0,$$

$$\text{Riemann Hypothesis} \iff |M(n)| \leq n^{1/2+\epsilon}, \text{ for all } \epsilon > 0 \text{ and all sufficiently large } n.$$

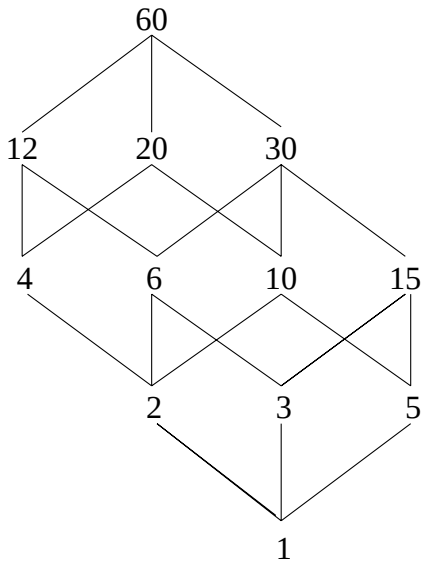


Figure 16: The divisors of "60".

The Möbius function is an indispensable tool in enumerative combinatorics because it can be used to "invert" summations over a partially ordered index set. Here is a statement of the "Möbius inversion formula" in a special case. If a function $f : P \to \mathbb{Z}$ from a poset P to the integers is related to another function $g : P \to \mathbb{Z}$ by the partial summation formula

$$f(x) = \sum_{y \geq x} g(y),$$

then the value $g(x_0)$ at the bottom element x_0 of P can be expressed in terms of f via the formula

$$g(x_0) = \sum_{y \in P} \mu(y) f(y).$$

The Möbius function also has a topological meaning, which is the reason it turns up in "Fact 2" of this section. The connection is as follows. Let P be a poset with bottom element b and top element t. Define the set family $\Delta(P)$ to consist of all chains (meaning: totally ordered subsets) $x_1 < x_2 < \cdots < x_k$ in $\overline{P} = P \setminus \{b, t\}$, meaning P with b and t removed. Then $\Delta(P)$ is a simplicial

complex (since a subset of a chain is also a chain), so as discussed in Section 9 there is an associated topological space.

For instance, let P be the divisor poset of the number "60" shown in Figure 16. Then $\overline{P} = P \setminus \{1, 60\}$ has the following twelve maximal chains

$$
\begin{array}{c}
2 \text{ — } 4 \text{ — } 12 \\
2 \text{ — } 4 \text{ — } 20 \\
2 \text{ — } 6 \text{ — } 12 \\
2 \text{ — } 6 \text{ — } 30 \\
2 \text{ — } 10 \text{ — } 20 \\
2 \text{ — } 10 \text{ — } 30 \\
3 \text{ — } 6 \text{ — } 12 \\
3 \text{ — } 6 \text{ — } 30 \\
3 \text{ — } 15 \text{ — } 30 \\
5 \text{ — } 10 \text{ — } 20 \\
5 \text{ — } 10 \text{ — } 30 \\
5 \text{ — } 15 \text{ — } 30
\end{array}
$$

As was explained in Section 9 these twelve triples of the simplicial complex should be thought of as describing twelve triangles that are to be glued together along common edges. This gives the topological space shown in Figure 17 — a 2-dimensional disc.

So, what does all this have to do with the Möbius function? The relation is this. Let $\beta_i(P)$ be the ith Betti number of the simplicial complex $\Delta(P)$, and let $\mu(P)$ denote the value that the Möbius function attains at the top element of P. Then,

$$\mu(P) = \beta_0(P) - \beta_1(P) + \beta_2(P) - \beta_3(P) + \cdots. \tag{28}$$

For instance, the space depicted in Figure 17 is a disc. The important thing here is that this space has no holes of any kind. Hence, all Betti numbers $\beta_i(P)$ are zero, implying via formula (28) that $\mu(P) = 0$. This "explains" topologically why $\mu(60) = 0$, a fact we already knew from simpler considerations. On the other hand, if P is the divisor diagram of "30" (which

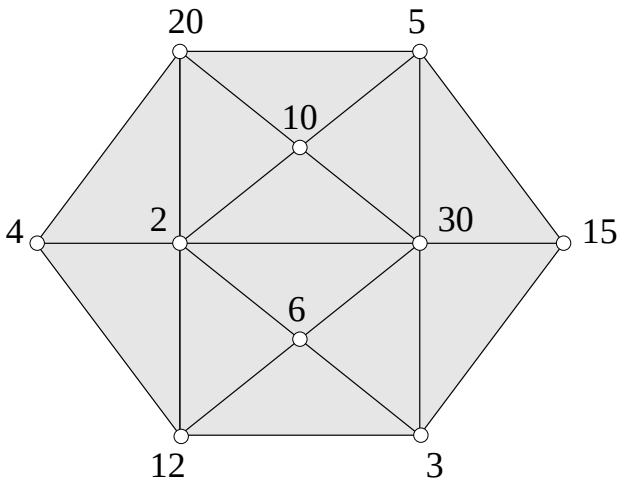


Figure 17: The simplicial complex of proper divisors of "60".

can be seen as a substructure in Figure 16), then $\Delta(P)$ is the circle 2 — 6 — 3 — 15 — 5 — 10 — 2 (a substructure in Figure 17). This circle has a one-dimensional hole, so $\beta_1(P) = 1$. All other Betti numbers are zero, hence formula (28) gives that $\mu(30) = -1$, another fact we have already encountered.

---

12 Face numbers of polytopes.

Among the many results of Euler that have initiated fruitful lines of development in combinatorics, the one that is perhaps most widely known is "Euler's formula" for 3-dimensional polytopes from 1752. It goes as follows.

A 3-*polytope* P (or, 3-dimensional convex polytope, to be more precise) is for a mathematician a bounded region of space obtained as the intersection of finitely many halfspaces (and not contained in any plane). For the layman it can be described as the kind of solid body you can create from a block

of cheese with a finite number of plane cuts with a knife. For instance, take the ordinary cube shown in Figure 18 — it can be cut out with six plane cuts. The cube is one of the five *Platonic solids: tetrahedron, cube, octahedron, dodecahedron and icosahedron*, known and revered by the Greek mathematicians in antiquity.

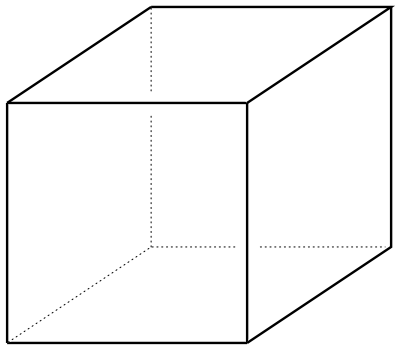


Figure 18: The cube.

A polytope that is dear to all combinatorialists is the "permutohedron", shown in Figure 19. Its 24 corners correspond to the $24 = 4 \cdot 3 \cdot 2 \cdot 1$ permutations of the set $\{1, 2, 3, 4\}$. The precise rule for constructing the permutohedron and for labelling its vertices with permutations is best explained in 4-dimensional space and will be left aside. Note that the pairs of permutations that correspond to edges of the permutohedron are precisely pairs that differ by a switch of two adjacent entries, such as "2143 — 2134" or "3124 — 3214". Thus, edge-paths on the boundary of the permutohedron are precisely paths consisting of such "adjacent transpositions", giving geometric content to the topic of reduced decompositions, that was discussed in Section 6.

The boundary of a 3-polytope is made up of pieces of dimension 0, 1 and 2 called its *faces*. These are the possible areas of contact if the polytope is made to touch a plane surface, such as the top of a table. The 0-faces are the corners, or vertices, of the polytope. The 1-faces are the edges, and the 2-faces are the flat surfaces, such as the six squares bounding the cube. The permutohedron has fourteen 2-faces, six of which are 4-sided and eight are 6-sided.

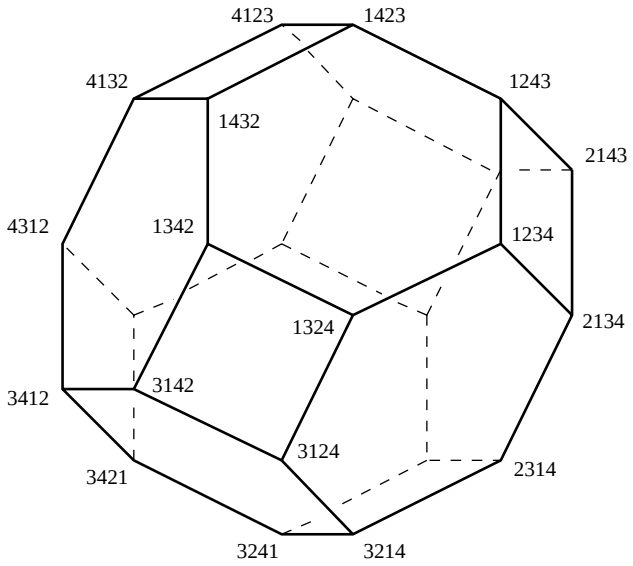


Figure 19: The permutohedron.

Euler's formula has to do with counting the number of faces of dimensions 0, 1 and 2. Namely, let f_i be the number of i-dimensional faces.

Euler's Formula. *For any* 3*-polytope:*

$$f_0 - f_1 + f_2 = 2.$$

Let us verify this relation for the cube and the permutohedron, see Figures 18 and 19.

	f_0	f_1	f_2	$f_0 - f_1 + f_2$
Cube	8	12	6	$8 - 12 + 6 = 2$
Permutohedron	24	36	14	$24 - 36 + 14 = 2$

From a modern mathematical point of view there is no difficulty in defining higher-dimensional polytopes. Thus, a *d-polytope* is a full-dimensional bounded intersection of closed halfspaces in $\mathbb{R}^d$. Such higher-dimensional

polytopes have taken on great practical significance in the last fifty years because of their importance for linear programming. The term "linear programming" refers to techniques for optimizing a linear function subject to a collection of linear constraints. The linear constraints cut out a feasible region of space, which is a d-polytope (possibly unbounded in this case). The combinatorial study of the structure of polytopes has interacted very fruitfully with this applied area.

It can be shown that the same definition of the *faces* of a polytope works also in higher dimensions (namely "the possible areas of contact if the polytope is made to touch a plane surface in $\mathbb{R}^d$"), and that there are only finitely many faces of each dimension $0, 1, \ldots, d-1$. Thus we may define the number f_i of i-dimensional faces for $i = 0, 1, \ldots, d-1$. These numbers for a given polytope P are collected into a string

$$f(P) = (f_0, f_1, \ldots, f_{d-1}),$$

called the *f-vector* of P. For instance, we have seen that $f(\text{cube}) = (8, 12, 6)$ and $f(\text{permutohedron}) = (24, 36, 14)$.

Is there an Euler formula for f-vectors in higher dimensions? This question was asked early on, and by the mid-1800's some mathematicians had discovered the following beautiful fact.

Generalized Euler Formula. *For any d-polytope:*

$$f_0 - f_1 + f_2 - \cdots + (-1)^{d-1} f_{d-1} = 1 + (-1)^{d-1}.$$

However, the early discoverers experienced serious difficulty with proving this formula. It is generally considered that the first complete proof was given around the year 1900 by Henri Poincaré.

Having seen this formula it is natural to ask: *What other relations, if any, do the face numbers f_i satisfy*? This question opens the doors to a huge and very active research area, pursued by combinatorialists and geometers. Many equalities and inequalities are known for various classes of polytopes, such as upper bounds and lower bounds for the numbers f_i in terms of the dimension d and the number f_0 of vertices.

The boldest hope one can have for the study of f-vectors of polytopes is to obtain a complete characterization. By this is meant a reasonably simple set of conditions by which one can recognize if a given string of numbers is the f-vector of a d-polytope or not. For instance, one may ask whether

$$(14, 89, 338, 850, 1484, 1834, 1604, 971, 380, 76) \tag{29}$$

is the f-vector of a 10-polytope? We find that

$$14 - 89 + 338 - 850 + 1484 - 1834 + 1604 - 971 + 380 - 76 = 0,$$

in accordance with the generalized Euler formula. Had this failed we would know for sure that we are not dealing with a true f-vector, but agreeing with the Euler formula is certainly not enough to draw any conclusion. What other "tests" are there, strong enough to tell for sure whether this is the f-vector of a 10-polytope?

An answer is known for dimension 3; namely, (f_0, f_1, f_2) is the f-vector of a 3-polytope if and only if

$$\begin{array}{ll} (i) & f_0 - f_1 + f_2 = 2, \\ (ii) & f_0 \leq 2f_2 - 4, \\ (iii) & f_2 \leq 2f_0 - 4. \end{array}$$

However, already the next case of 4 dimensions presents obstacles that with present methods are unsurmountable. Thus, no characterization of f-vectors of general polytopes is known. But if one narrows the class of polytopes to the so called "simplicial" ones there is a very substantial result that we will now formulate.

A *d-simplex* is a d-polytope which is cut out by exactly $d+1$ plane cuts. In other words, it has $d+1$ maximal faces, which is actually the minimum possible for a d-polytope. A 1-simplex is a line segment, a 2-simplex is a triangle, a 3-simplex is a tetrahedron, and so on; see Figure 20. In general, a d-simplex is the natural d-dimensional analogue of the tetrahedron.

A d-polytope is said to be *simplicial* if all its faces are simplices. It comes to the same to demand that all maximal faces are $(d-1)$-simplices. For instance, a 3-polytope is simplicial if all 2-faces are triangular, as in Figure

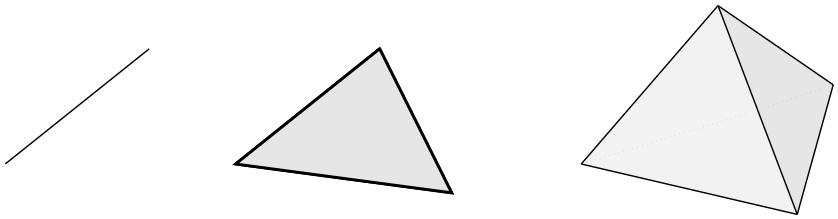


Figure 20: A d-simplex, $d = 1, 2, 3$.

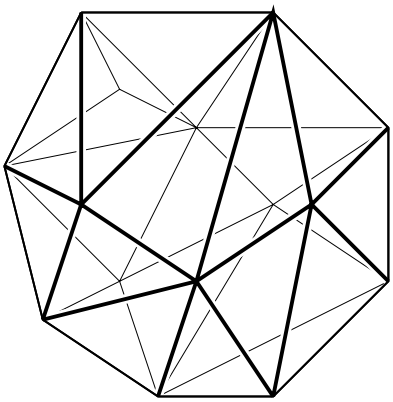


Figure 21: A simplicial 3-polytope.

21; so the octahedron and icosahedron are examples of simplicial polytopes but the cube and permutohedron are not. If a polytope is simplicial then its faces form a simplicial complex in the sense defined in Section 9. The class of simplicial polytopes is special from some points of view, but nevertheless very important in polytope theory. For instance, if one seeks to maximize the number of i-faces of a d-polytope with n vertices, the maximum is obtained simultaneously for all i by certain simplicial polytopes.

In 1971 Peter McMullen (b. 1942) made a bold conjecture for a characterization of the f-vectors of simplicial polytopes. A key role in his proposed conditions was played by certain "g-numbers," so his conjecture became known as the "g-conjecture." In 1980 two papers, one by Louis Joseph Billera (b. 1943) and Carl William Lee (b. 1954) and one by Richard Peter Stanley (b. 1944), provided the two major implications that were needed for a proof of

the conjecture. Their combined efforts thus produced what is now known as the "g-theorem." To state the theorem we need to introduce an auxiliary concept.

By a *multicomplex* we mean a nonempty collection M of monomials in indeterminates $x_1, x_2, \ldots, x_n$ such that if $m \in M$ and m' divides m then $m' \in M$. Figure 22 shows the multicomplex $M = \{1, x, y, z, x^2, xy, yz, z^2, x^2y, z^3\}$ ordered by divisibility.

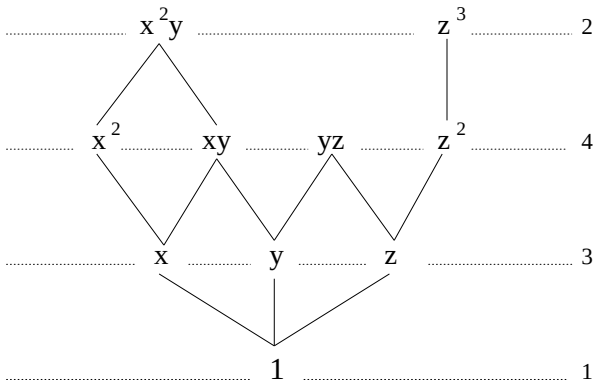


Figure 22: A multicomplex.

An *M-sequence* is a sequence $(1, a_1, a_2, a_3, \ldots)$ such that each a_i is the number of monomials of degree i in some fixed multicomplex. For instance, the M-sequence coming from the multicomplex M in Figure 22 is $(1, 3, 4, 2)$. A multicomplex and an M-sequence can very well be infinite, but only finite ones will concern us here. If some zeros are added or removed at the end of a finite M-sequence it remains an M-sequence.

The "M" in M-sequence is mnemonic both for "multicomplex" and for "Macaulay", in honor of Francis Sowerby Macaulay (1862-1937) who first seems to have studied the concept in a paper from 1927. Macaulay's purpose was entirely algebraic (to characterize so called Hilbert functions of certain graded algebras), but the underlying combinatorics of his investigations has turned out to have far-reaching ramifications.

We are now ready to formulate the g-theorem, characterizing the f-

vectors of simplicial d-polytopes. Let δ be the greatest integer less than or equal to $d/2$, and let $M_d = (m_{i,j})$ be the matrix with $(\delta+1)$ rows and d columns and with entries

$$m_{i,j} = \binom{d+1-i}{d-j} - \binom{i}{d-j}, \quad \text{for } 0 \le i \le \delta,\ 0 \le j \le d-1.$$

Here we are using the *binomial coefficients*, defined by

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}, \text{ for } 0 \le k \le n, \quad \binom{n}{k} = 0 \text{ otherwise,}$$

where $n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1$, and $0! = 1$.

For example, with $d = 10$ we get

$$m_{2,8} = \binom{10+1-2}{10-8} - \binom{2}{10-8} = \frac{9!}{2! \cdot 7!} - \frac{2!}{2! \cdot 0!} = 36 - 1 = 35,$$

and the whole matrix is

$$M_{10} = \begin{pmatrix} 11 & 55 & 165 & 330 & 462 & 462 & 330 & 165 & 55 & 11 \\ 1 & 10 & 45 & 120 & 210 & 252 & 210 & 120 & 45 & 9 \\ 0 & 1 & 9 & 36 & 84 & 126 & 126 & 84 & 35 & 7 \\ 0 & 0 & 1 & 8 & 28 & 56 & 70 & 55 & 25 & 5 \\ 0 & 0 & 0 & 1 & 7 & 21 & 34 & 31 & 15 & 3 \\ 0 & 0 & 0 & 0 & 1 & 5 & 10 & 10 & 5 & 1 \end{pmatrix}$$

These matrices M_d determine a very surprising link between M-sequences and f-vectors.

The g-theorem. *The matrix equation*

$$\mathbf{f} = \mathbf{g} \cdot M_d$$

gives a one-to-one correspondence between f-vectors $\mathbf{f}$ of simplicial d-polytopes and M-sequences $\mathbf{g} = (g_0, g_1, \ldots, g_\delta)$.

The equation $\mathbf{f} = \mathbf{g} \cdot M_d$ is to be understood as follows. Multiply each entry in the first row of M_d by g_0, then multiply each entry in the second row

by g_1, and so on. Finally, after all these multiplications add the numbers in each column. Then the first column sum will equal f_0, the second column sum will equal f_1, and so on.

To exemplify the power of this theorem let us return to a question posed earlier; namely, is the vector $\mathbf{f}$ displayed in equation (29) the f-vector of a 10-polytope? This question can now be answered if sharpened from "10-polytope" to "simplicial 10-polytope". Easy computation shows that

$$\mathbf{f} = (1, 3, 4, 2, 0, 0) \cdot M_{10},$$

and we know from Figure 22 that $(1, 3, 4, 2, 0, 0)$ is an M-sequence. Hence, $\mathbf{f}$ is indeed the f-vector of some simplicial 10-polytope.

Having seen this, one can wonder if we were just lucky with this relatively small example. Perhaps for large d it is as hard to determine if a sequence is an M-sequence as to determine if a sequence is an f-vector coming from a simplicial polytope. This is not the case. There exists a very easy criterion in terms of binomial coefficients that quickly tests an integer sequence for being an M-sequence. We will however not state it here.

The proof of the g-theorem is very involved and calls on a lot of mathematical machinery. The part proved by Billera and Lee — that for every M-sequence $\mathbf{g}$ there exists a simplicial polytope with the corresponding f-vector — requires some very delicate geometrical arguments. The part proved by Stanley — that conversely to every simplicial polytope there corresponds an M-sequence in the stated way — uses tools from algebraic geometry in an essential way. Here is a brief statement for readers with sufficient background. There are certain complex projective varieties, called *toric varieties*, associated to d-polytopes with rational coordinates, and the fact that the sequence $\mathbf{g}$ corresponding to the f-vector of a polytope is an M-sequence ultimately derives from a multicomplex that can be constructed in the cohomology algebra of such a variety.

The g-vector associated to a simplicial polytope via the g-theorem is rich in geometric, algebraic and combinatorial meaning, yet it is still poorly understood and the subject of much current study.

In this paper we have several times commented on the many surprising,

remarkable and mysterious connections that exist between different mathematical objects, different mathematical problems and different mathematical areas. Take for example the Schensted correspondence described in Section 3, connecting permutations and pairs of standard Young tableaux; or the connections between combinatorics and representation theory or combinatorics and topology described in earlier sections. The g-theorem is one more example of this kind, establishing an unsuspected link between the combinatorial structure of multicomplexes of monomials and the facial structure of simplicial polytopes — two seemingly totally unrelated classes of objects.

In closing, let us once more mention that no characterization is known for f-vectors of general polytopes of dimension greater than 3. The success in the case of simplicial polytopes depends on some very special structure, available in that case but lacking or much more complex in general. Thus, the study of f-vectors, initiated by Euler's discovery almost 250 years ago, is likely to remain an important challenge for many years to come.

13 Further reading.

We refer here mainly to general accounts that should be at least partially accessible to the layman and that give lots of further references.

For a broad view of current combinatorics, with a wealth of information and references, see

- *Handbook of Combinatorics* (R. Graham, M. Grötschel and L. Lovász, eds.), North-Holland, Amsterdam/New York, and MIT Press, Cambridge, Massachusetts, 1995.

A good reference for number partitions is

- G. E. Andrews, *The Theory of Partitions*, Encyclopedia of Mathematics and Its Applications, Vol. 2, Addison-Wesley, Reading, Massachusetts, 1976.

For the work of Bousquet-Mélou and Eriksson mentioned at the end of Section 2, see

- M. Bousquet-Mélou and K. Eriksson, Lecture hall partitions, Parts 1 and 2, *The Ramanujan Journal* **1** (1997), 101–111, 165–185.

The basic theory of enumeration is developed in

- R. P. Stanley, *Enumerative Combinatorics*, Volume 1, Wadsworth & Brooks/Cole, Monterey, CA, 1986 (second printing, Cambridge University Press, Cambridge/New York, 1997), and Volume 2, Cambridge University Press, Cambridge/New York, to appear in 1998 or 1999.

The combinatorics of number and set partitions, standard Young tableaux, generating functions and the Möbius function, together with algebraic ramifications, is discussed there. A briefer account of this material is given in

- I. Gessel and R. P. Stanley, Algebraic Enumeration, pp. 1021–1061 in *Handbook of Combinatorics*, see above.

Another introduction to generating functions is given in

- H. S. Wilf, *generatingfunctionology*, Academic Press, San Diego, 1990.

The following book is a nice companion to the study of enumeration:

- N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, 1995. There is also an interactive version on the web at http://www.research.att.com/~njas/sequences/

An introduction to the combinatorics of permutations and Young tableaux can be found in Chapter 7 of the book of Stanley cited above, as well as in Chapter 5.1 of

- D. E. Knuth, *The Art of Computer Programming, Vol. 3*, Addison-Wesley, Reading, Massachusetts, 1973 (updated and reprinted 1997),

and in

- B. E. Sagan, *The symmetric group. Representations, combinatorial algorithms, and symmetric functions*, Wadsworth & Brooks/Cole, Pacific Grove, CA, 1991.

The latter book also gives an accessible introduction to the connections with representation theory.

There is a huge literature on tilings, but most of this is not concerned with enumerative problems. For a wealth of information concerning the non-enumerative aspects see

- B. Grünbaum and G. C. Shephard, *Tilings and Patterns*, Freeman, New York, 1987.

At present there is no good introduction to the enumerative aspects of tilings. The results mentioned in this paper can be found in the following references:

- M. Ciucu, Enumeration of perfect matchings in graphs with reflective symmetry, *Journal of Combinatorial Theory, Series A* **77** (1997), 67–97.
- N. Elkies, G. Kuperberg, M. Larsen, and J. Propp, Alternating-sign matrices and domino tilings, Parts I and II, *Journal of Algebraic Combinatorics* **1** (1992), 111–132, 219–234.
- W. Jockusch, Perfect matchings and perfect squares, *Journal of Combinatorial Theory, Series A* **67** (1994), 100–115.

For connections between combinatorics and topology, including more details about the evasiveness and Kneser conjectures, see either of

- A. Björner, Combinatorics and Topology, *Notices of the American Mathematical Society* **32** (1985), 339–345.

- A. Björner, Topological Methods, pp. 1819–1872 in *Handbook of Combinatorics*, see above.

Connections between combinatorics and computer science is a huge subject. For some glimpses see

- L. Lovász, D. B. Shmoys, and É. Tardos, Combinatorics in Computer Science, pp. 2003–2038 in *Handbook of Combinatorics*, see above.

and for sorting algorithms also the book by Knuth mentioned above.

The disproof of Borsuk's conjecture is reported in

- B. Cipra, Disproving the obvious in higher dimensions, *What's Happening in the Mathematical Sciences* **1** (1993), 21–25.

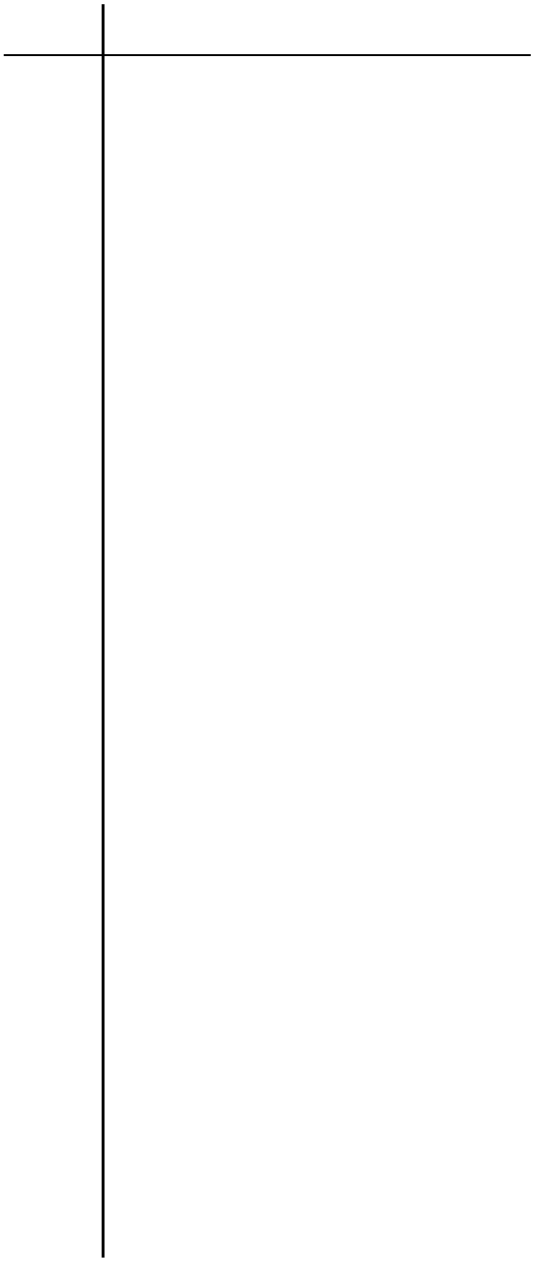
- A. Skopenkov, Borsuk's problem, *Quantum* **7** (1996), 17–21,

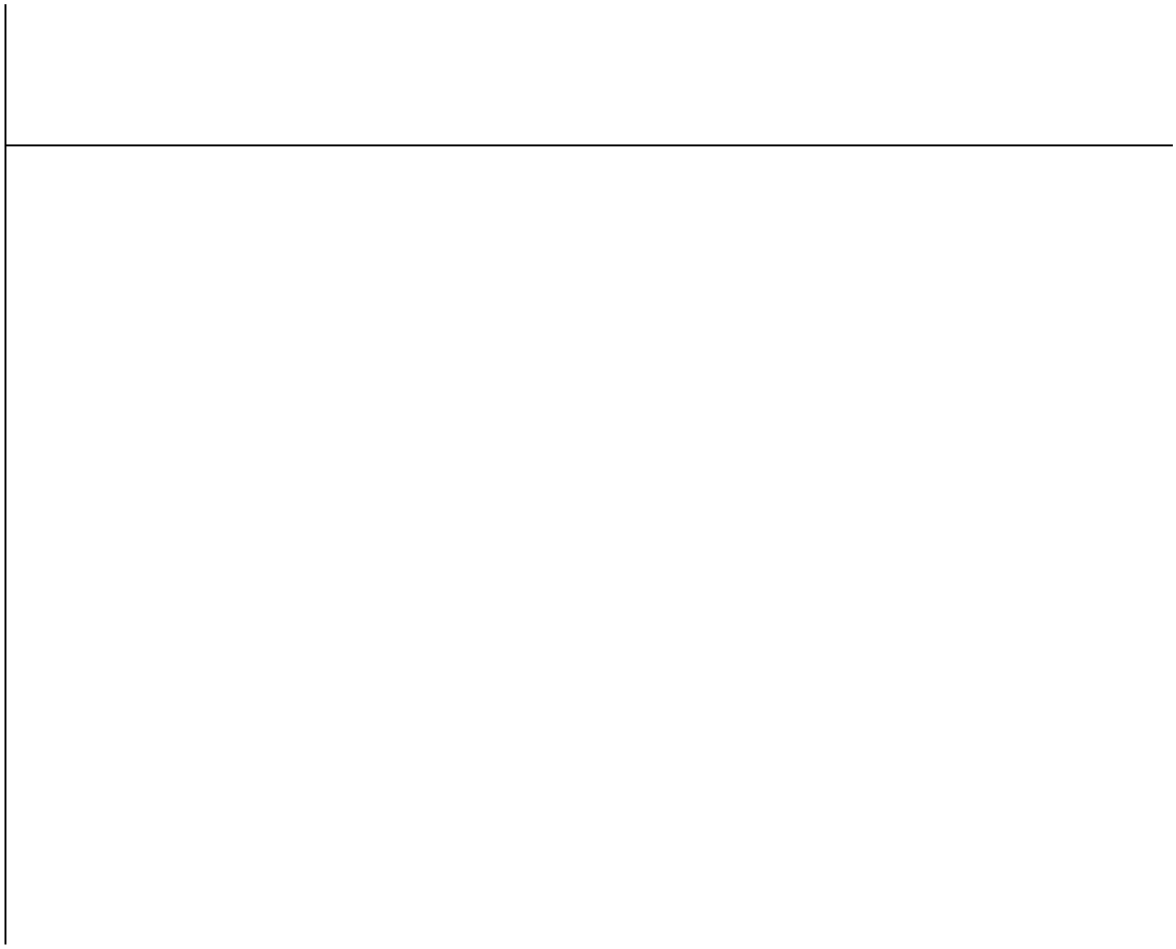
while more about the k-equal problem and its solution can be found in

- A. Björner, Subspace arrangements, in *First European Congress of Mathematics, Paris 1992* (A. Joseph *et al.*, eds.), Progress in Mathematics Series, Volume **119**, Birkhäuser, Boston, 1994, pp. 321–370.

Finally, for convex polytopes and the g-theorem we refer to

- G. M. Ziegler, *Lectures on Polytopes*, GTM Series, Springer-Verlag, Berlin, 1995.





REPORTS
IN
INFORMATICS

ISSN 0333-3590

A Construction for Binary Sequence Sets with Low Peak-to-Average Power Ratio

Matthew G. Parker and Chintha Tellambura

REPORT NO 242 **February 2003**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL
http://www.ii.uib.no/publikasjoner/texrap/ps/2003-242.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

A Construction for Binary Sequence Sets with Low Peak-to-Average Power Ratio

Matthew G. Parker*and Chintha Tellambura†

20th February 2003

Abstract

A recursive construction is provided for sequence sets which possess good Hamming Distance and low Peak-to-Average Power Ratio (PAR) with respect to **any** Local Unitary Unimodular Transform (including all one and multi-dimensional Discrete Fourier Transforms).

1 Introduction

Pairs of Golay Complementary Sequences (CS) have the property that the sidelobes of the Aperiodic Autocorrelation of each sequence in the pair sum to zero [7]. Consequently they have found application in the areas of Telecommunications and Physics for such tasks as channel-sounding, spread-spectrum, and synchronization. It follows that the Peak-to-Average Power Ratio (PAR) with respect to the one-dimensional continuous Discrete Fourier Transform (DFT_1^∞) of each sequence in the pair satisfies PAR ≤ 2. For lengths 2^n one can generate CS pairs using Golay-Rudin-Shapiro (RuS) construction [28, 29]. However it has not yet been proved that all length 2^n CS can be constructed using RuS as $n \to \infty$. Davis and Jedwab have shown that the RuS set comprise a union of certain binary quadratic cosets of Reed-Muller (RM) $(1, n)$ when expressed in Algebraic Normal Form (ANF)[4]. Moreover, as these sequences are a subset of RM$(2, n)$, then the Hamming Distance, D, between sequences in the set satisfies $D \geq 2^{n-2}$. Although the properties of RuS and CS pairs have been known for many years, the description of [4] brought together and formalised much of this work in the context of Reed-Muller codes. This was in response to the pressing demand of Orthogonal Frequency Division Multiplexing (OFDM) communications systems for error-correcting codes where each codeword also has low PAR with respect to (wrt) DFT_1^∞. The low PAR is required to alleviate the linearity requirements of the amplifier at the transmitter. The question of error-correcting codes with low PAR wrt DFT_1^∞ was highlighted by [10], prompting a great deal of research culminating in the fundamental codeset of Davis and Jedwab (DJ set), as outlined in the papers of [4, 23] (equation (6) of this paper), which exploit the properties of RuS. However, a communications engineer will probably point out that the major weakness of the DJ set for OFDM is that its code rate only remains acceptably high for up to about 32 frequency carriers, the

*M.G.Parker is with the Code Theory Group, Inst. for Informatikk, Høyteknologisenteret i Bergen, University of Bergen, Bergen 5020, Norway. E-mail: matthew@ii.uib.no. Web: http://www.ii.uib.no/~matthew/, Author funded by NFR Project Number 119390/431

†C. Tellambura is with the Department of Electrical and Computer Engineering, Electrical and Computer Engineering Research Facility, University of Alberta, Edmonton, Alberta, Canada, T6G 2V4. E-mail: chintha@ee.ualberta.ca. Phone/Fax: +780-492-7228(1811)

rate vanishing as $n \to \infty$, and most current OFDM systems require anywhere from 256 to 8192 frequency carriers. Therefore, in practise, most engineers will implement some form of clipping or Selected-Mapping in order to reduce spectral peaks (PAR) at the OFDM transmitter. In other words, instead of constructing and sending a sequence, the transmitter will generate an arbitrary sequence or sequences, test their PARs, then either clip their peaks before transmission or choose to send the sequence with lowest PAR. Constructive techniques can avoid all this testing, but a major requirement for any constructive coding technique is that its rate remains acceptably high for large numbers of carriers. Higher rates are certainly possible and desirable for PAR $\leq O(n)$ and D large [24]. A generalisation of RuS construction to other starting seeds [16, 17] allows inclusion of more low PAR quadratic cosets of RM$(1, n)$ in the code, thereby improving code rate somewhat. Higher degree cosets can also be added, marginally increasing code rate at price of distance, D, which decreases. However the rate remains unacceptably low for more than about 32 carriers.

This paper provides new answers to this problem by defining constructive techniques for low PAR error-correcting codes of blocklength > 32 with acceptable rate. For instance, we can (almost) construct a rate $\frac{1}{3}$ code of length 64 with distance 16 and PAR ≤ 4.0, a rate $\frac{2}{3}$ code of length 64, distance 4, and PAR ≤ 8.0, and a rate $\frac{1}{2}$ code of length 256, distance 4, and PAR ≤ 16.0. We emphasise 'almost' because, although we most certainly identify and algebraically describe very large codesets with low PAR, our constructions are not strictly implementable yet, due to certain edge symmetries (coding collisions) which compromise invertibility of the encoding. It remains an open challenge to eliminate these coding collisions, and part of the aim of this paper is to present and motivate this challenge in a clear way.

It turns out that our construction also requires the ability to generate all distinct permutation polynomials from $Z_2^t \to Z_2^t$ of algebraic degree $\leq d$ for some d, $1 \leq d < t$. To the best knowledge of the authors, such an algorithm only exists in the literature for the case $d = 1$ (namely "Bruhat Decomposition", or as encountered when generating all possible binary linear error-correcting codes of maximum rank and length) and, for $d = 1$, there are $\prod_{i=0}^{t-1}(2^t - 2^i)$ such polynomials. This paper provides strong motivation to develop further algorithms for the cases $1 < d < t$, along with the enumeration of the size of such sets as t varies.

Another aim of this paper is to advertise the fact that RuS sequences, and their generalisation as described in this paper, have a much stronger property than just a low PAR upper bound wrt the DFT_1^∞. [13, 16, 17, 25] have all pointed out the Bent/Almost Bent properties of the RuS set, and [16, 17] and this paper proves that the RuS set, and their generalisations satisfy PAR $\leq 2^t$ wrt <u>all</u> possible Linear Unitary Unimodular Transforms (LUUTs), including DFT_1^∞ and Walsh-Hadamard Transform (WHT). We will define LUUTs in the sequel. Consequently, the RuS construction and its generalisation have relevance also to Multi-Code CDMA [16, 17, 25], Weight Hierarchy and Quantum Entanglement [18, 19], and Cryptography [27].

To summarise, the main new contributions of this paper are as follows:

- A proposal to measure PAR wrt the infinite set of Linear Unimodular Unitary Transforms (LUUTs), whose rows comprise all possible linear unimodular sequences. This set includes DFT_1^∞, the Walsh-Hadamard Transform (WHT), and many other transforms.
- A construction (Constructions 1 - 3) for large sets of sequences with tight constant upper bound on PAR and good distance properties, where PAR is computed wrt the

infinite set of LUUTs.

Although we acknowledge that our constructions are implicitly covered in the literature by Golay [6, 7], Turyn [34], and others [33, 5], wrt DFT_1^∞, no mention in the literature is given of low PAR constructions wrt to the much larger set of LUUTs and, apart from the special case considered by Davis and Jedwab [4] wrt DFT_1^∞, and the case of low PAR wrt the WHT [3, 25], no attempt has been made to express these constructions in concise Algebraic Normal Form (ANF) or to consider the construction of such sequences, or to consider the Hamming Distance between members of the sequence set.

Our Construction as a Generalisation of Golay-Rudin-Shapiro Construction:
Golay-Rudin-Shapiro (RuS) sequences are a special case of Golay Complementary Pairs as first introduced by Marcel Golay [6, 22]. RuS sequence construction [7, 28, 29] exploits the recursion,

$$\begin{aligned} \mathbf{a}' &= \mathbf{a}|\mathbf{b} \\ \mathbf{b}' &= \mathbf{a}|\overline{\mathbf{b}} \end{aligned} \tag{1}$$

where $\mathbf{a}$ and $\mathbf{b}$ are both bipolar sequences of length N, $\mathbf{a}'$ and $\mathbf{b}'$ are both sequences of length $2N$, '|' means concatenation, and $\overline{\mathbf{b}}$ means the multiplication of elements of $\mathbf{b}$ by -1. The key observation that motivates the constructions of this paper is that we can write (1) as,

$$(\mathbf{a}', \mathbf{b}')^T = \mathbf{E} \odot \begin{pmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{a} & \mathbf{b} \end{pmatrix}$$

where $\mathbf{E} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, and '$\odot$' means point-multiplication of matrix elements. For instance, if $a = (1, 1)$ and $b = (1, -1)$, then $a' = (1, 1, 1, -1)$ and $b' = (1, 1, -1, 1)$.

This paper shows that RuS sequences satisfy PAR ≤ 2.0 wrt all LUUTs precisely because $\mathbf{E}$ is an orthogonal 2×2 matrix. The RuS generalisation of this paper uses sequence recursion where successive $\mathbf{E}$ matrices are arbitrary $R \times R$ Hadamard matrices, such that the generated sequences have PAR $\leq R$. For a given canonical representation of a Hadamard matrix, $\mathbf{E}$, an arbitrary row/column permutation of $\mathbf{E}$ is specified by γ, for row permutation, and θ, for column permutation. In this paper we emphasise the case where $\mathbf{E}$ is the Walsh-Hadamard Transform (WHT) matrix, although the basic construction still works when $\mathbf{E}$ is a more general Hadamard matrix. Given that $\mathbf{E}$ is a WHT, the sequence construction is then primarily specified by the permutations γ_j and θ_j, at each stage of the recursion. As stated earlier, much of the difficulty relating to the construction of this paper arises from an attempt to classify, enumerate, and generate these permutations according to their algebraic degree, as these degrees determine the overall ANF degree of the constructed sequence, which in turn determines the (Reed-Muller) Hamming Distance of the code. This paper therefore gives a strong justification for future research into classification and enumeration of permutation polynomials according to maximum polynomial degree.

Construction 1 provides a way of generating low PAR error-correcting codes of any length, r^n, and over any alphabet. As a special case, Construction 2 generates binary codesets of length 2^n and PAR $\leq 2^t$, comprising ANFs up to degree μ, where $\mu \leq 2t - 2$ for $t > 1$, and $\mu = 2$ for $t = 1$. These codesets have PAR $\leq 2^t$ wrt **all** LUUTs, including one and multi-dimensional continuous DFTs [16, 17]. As LUUTs include WHTs, then our construction gives large codesets of (Near)-Bent functions [15, 3, 26]. These binary sequences are not just (Near)-Bent but are also distant from linear sequences over all (unimodular) alphabets, not just over Z_2 - a particularly strong cryptographic attribute. Construction 2 of this paper can be viewed as a recursive generalisation of a **two-sided** Maiorana-McFarland construction where our sequence set has low PAR wrt **all** LUUTs, not just WHT. We also provide an explicit generation method for the complete quadratic subset of Construction 2

using Bruhat decomposition [2, 1]. In [25], Paterson increases code rate, at the price of increased PAR wrt the WHT, by replacing the inherent one-to-one permutation of Maiorana-McFarland construction with a many-to-one map. Construction 3 of this paper similarly generalises Construction 2 by replacing the constituent permutations with many-to-one and/or one-to-many maps. Throughout this paper, we assume our sequences are of length r^n for some integers, r, n, although we emphasise the case where $r = 2$.

2 Linear Sequences, Linear Unimodular Unitary Transforms (LUUTs) and Peak-to-Average Power Ratio (PAR)

PAR is a spectral measure. We must therefore first define the transforms over which the spectrum is to be computed. We call these transforms *LUUTs* (defined below), and LUUTs have linear rows, so we first define linearity:

Definition 1. *Let* $\mathbf{l} = (l_0, l_1, \ldots, l_{r^n-1})$ *be a length* r^n *complex sequence.* $\mathbf{l}$ *is defined to be* unimodular *if* $|l_i| = |l_j|, \forall i, j$, unitary *if* $\sum_{i=0}^{r^n-1} |l_i|^2 = 1$, *and* r-linear *if,*

$$\begin{array}{ll} \mathbf{l} = & (a_{0,0}, a_{0,1}, \ldots, a_{0,r-1}) \otimes (a_{1,0}, a_{1,1}, \ldots, a_{1,r-1}) \otimes \ldots \otimes (a_{n-1,0}, a_{n-1,1}, \ldots, a_{n-1,r-1}) \\ = & \bigotimes_{i=0}^{n-1} (a_{i,0}, a_{i,1}, \ldots, a_{i,r-1}) \end{array}$$

where $\otimes$ *is the 'left tensor product', such that* $\mathbf{A} \otimes (B_0, B_1, \ldots) = (B_0\mathbf{A}, B_1\mathbf{A}, \ldots)$. *For length* r^n *sequences where* r *is prime,* r*-linear is called* linear.

For example,

$l = \frac{1}{\sqrt{2}}(1, 0, 0, 1)$ is a unitary sequence,
$l = \frac{1}{2}(1, 1, 1, -1)$ is a unimodular unitary sequence,
$l = \frac{1}{2}(1, -1, 1, -1) = \frac{1}{\sqrt{2}}(1, -1) \otimes \frac{1}{\sqrt{2}}(1, 1)$ is a linear, unimodular, unitary sequence

Definition 2. $\mathbf{L_{r,n}}$ *is the infinite set of length* r^n *complex* r*-linear, unitary, unimodular sequences.*

Definition 3. *A* $r^n \times r^n$ *matrix,* $\mathbf{U}$, *is* unitary *if* $\mathbf{U}\mathbf{U}^\dagger = \mathbf{I_{r^n}}$, *where* $\dagger$ *means conjugate transpose, and* $\mathbf{I_{r^n}}$ *is the* $r^n \times r^n$ *identity matrix.*

Definition 4. *A* $r^n \times r^n$ r-Linear Unimodular Unitary Transform *(*r*-LUUT) matrix* $\mathbf{L}$ *has rows taken from* $\mathbf{L_{r,n}}$ *such that* $\mathbf{L}\mathbf{L}^\dagger = \mathbf{I_{r^n}}$. *When* r *is prime,* $r^n \times r^n$ r*-LUUTs are called* LUUTs. *Note that the set of* $r^n \times r^n$ q*-LUUTs is a subset of the set of* $r^n \times r^n$ r*-LUUTs iff* $q|r$.

Example LUUTs for $r = 2$**:** The $2^n \times 2^n$ Walsh-Hadamard (WHT) and Negahadamard (NHT) matrices are LUUTs defined by $\bigotimes_{i=0}^{n-1} \mathbf{H}$, and $\bigotimes_{i=0}^{n-1} \mathbf{N}$, respectively, where $\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $\mathbf{N} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}$, and $i^2 = -1$. For instance, for $n = 2$, the WHT is the LUUT whose rows have the following tensor decomposition:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} (1,1) & \otimes & (1,1) \\ (1,-1) & \otimes & (1,1) \\ (1,1) & \otimes & (1,-1) \\ (1,-1) & \otimes & (1,-1) \end{pmatrix}$$

Similarly, for $n = 2$, the NHT is the LUUT whose rows have the following tensor decomposition:

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & i & i & -1 \\ 1 & -i & i & 1 \\ 1 & i & -i & 1 \\ 1 & -i & -i & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} (1,i) & \otimes & (1,i) \\ (1,-i) & \otimes & (1,i) \\ (1,i) & \otimes & (1,-i) \\ (1,-i) & \otimes & (1,-i) \end{pmatrix}$$

where $i^4 = 1$.

Definition 5. *We define* DFT_1^∞ *for length* 2^n *vectors to be an infinite subset of* $2^n \times 2^n$ *LUUTs, the union of whose rows form a subset of* $\mathbf{L_{2,n}}$ *such that each row factors, as in Definition 1, into a tensor product of length-two vectors* $(a_{i,0}, a_{i,1})$ *which, in turn, must satisfy* $a_{i,0} = \frac{1}{\sqrt{2}}$, $a_{i,1} = \frac{\omega^{ik}}{\sqrt{2}}$ *for some fixed integer* k*, where* ω *is any complex root of unity.*

For instance, for $n = 2$, DFT_1^∞ includes the LUUT which is the 4-point one-dimensional Cyclic DFT whose rows have a tensor decomposition as follows:

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} = \frac{1}{2}\begin{pmatrix} (1,1) & \otimes & (1,1) \\ (1,i) & \otimes & (1,-1) \\ (1,-1) & \otimes & (1,1) \\ (1,-i) & \otimes & (1,-1) \end{pmatrix}$$

where $i^2 = -1$.

DFT_1^∞ also includes the LUUT which is the 4-point one-dimensional NegaCyclic DFT whose rows have a tensor decomposition as follows:

$$\frac{1}{2}\begin{pmatrix} 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^3 & \omega^6 & \omega \\ 1 & \omega^5 & \omega^2 & \omega^7 \\ 1 & \omega^7 & \omega^6 & \omega^5 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} (1,\omega) & \otimes & (1,\omega^2) \\ (1,\omega^3) & \otimes & (1,\omega^6) \\ (1,\omega^5) & \otimes & (1,\omega^2) \\ (1,\omega^7) & \otimes & (1,\omega^6) \end{pmatrix}$$

where $\omega^4 = -1$.

By taking more and more 4×4 LUUTs of this form, we more closely approximate DFT_1^∞ for the case $r = 2, n = 2$. It is also helpful to notice that all rows of DFT_1^∞ occur as a subset of the rows of certain LUUTs formed from tensor products of 2×2 LUUTs. For instance, for $n = 2$, the rows of the 4×4 Cyclic DFT are contained in two rows of each of $\mathbf{H} \otimes \mathbf{H}$ and $\mathbf{N} \otimes \mathbf{H}$. Similarly, the rows of the 4×4 NegaCyclic DFT are contained in two rows of each of $\mathbf{W_1} \otimes \mathbf{N}$ and $\mathbf{W_3} \otimes \mathbf{N}$, where $\mathbf{W_1} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & \omega \\ 1 & -\omega \end{pmatrix}$, $\mathbf{W_3} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & \omega^3 \\ 1 & -\omega^3 \end{pmatrix}$.

Having defined linear unimodular sequences, we are in a position to define PAR with respect to $\mathbf{L_{r,n}}$:

Definition 6. *Let* s_i *be the* i*th element of a length* r^n *vector,* $\mathbf{s}$*. Then* r*-PAR*$(\mathbf{s})$ *is computed by measuring the maximum possible correlation squared of* $\mathbf{s}$ *with* **any** *length* r^n r*-linear unimodular sequence,* $\mathbf{l} \in \mathbf{L_{r,n}}$*:*

$$r\text{-}PAR(\mathbf{s}) = r^n max_{\mathbf{l} \in \mathbf{L_{r,n}}}(|\mathbf{s} \cdot \mathbf{l}|^2) = r^n max_{\mathbf{l} \in \mathbf{L_{r,n}}}(|\sum_{i=0}^{r^n-1} s_i l_i^*|^2)$$

where $\cdot$ *means 'inner product', and* * *means complex conjugate. Similarly,*

$$PA(\mathbf{s}) = r^n max_{\mathbf{l}}(|\mathbf{s} \cdot \mathbf{l}|^2)$$

$\mathbf{l}$ *taken over all rows of a* **fixed, specified** *subset of* $r^n \times r^n$ *unitary transforms,* $\mathbf{U}$

For a length r^n *sequence, the values of* r*-PAR and PA range from* 1.0 *(for an ideal spectrally flat sequence) to* r^n *(for a spectral* δ*-function). When* r *is prime,* r*-PAR is termed PAR.*

We can compute r-PAR of $\mathbf{s}$ by examining the transform spectra of $\mathbf{s}$ wrt **all** r-LUUTs (more practically we can approximate this continuous spectrum by applying a large enough subset of well-chosen r-LUUTs).

Example: Let $\mathbf{s} = 2^{\frac{-3}{2}}(1,1,1,-1,1,-1,1,-1)$. Then PA$(\mathbf{s})$ wrt the LUUT, $\mathbf{H} \otimes \mathbf{H} \otimes \mathbf{N}$, is obtained by first computing the matrix-vector product:

$$\mathbf{S} = (\mathbf{H} \otimes \mathbf{H} \otimes \mathbf{N})\mathbf{s} \quad = 2^{\frac{-3}{2}} \begin{pmatrix} 1 & 1 & 1 & 1 & i & i & i & i \\ 1 & -1 & 1 & -1 & i & -i & i & -i \\ 1 & 1 & -1 & -1 & i & i & -i & -i \\ 1 & -1 & -1 & 1 & i & -i & -i & i \\ 1 & 1 & 1 & 1 & -i & -i & -i & -i \\ 1 & -1 & 1 & -1 & -i & i & -i & i \\ 1 & 1 & -1 & -1 & -i & -i & i & i \\ 1 & -1 & -1 & 1 & -i & i & i & -i \end{pmatrix} 2^{\frac{-3}{2}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

$$= 2^{-3} \begin{pmatrix} 2 \\ 2+4i \\ 2 \\ -2 \\ 2 \\ 2-4i \\ 2 \\ -2 \end{pmatrix}$$

The largest magnitude value in $\mathbf{S}$ is $2^{-3}(2 \pm 4i)$. It follows that PA$(\mathbf{s}) = 2^3(2^{-6}(2^2 + 4^2)) = 2.5$ wrt $\mathbf{H} \otimes \mathbf{H} \otimes \mathbf{N}$. This also means that PAR$(\mathbf{s})$ is lower bounded by 2.5.

2.1 Complementary Sequence Sets (CS Sets)

A Complementary Sequence Set (*CS set*) of R unitary sequences of length R' conventionally has the complementary property that the sum of the one-dimensional Aperiodic Autocorrelations of each sequence in the set results in the δ function of magnitude R (zero sidelobe energy) [7, 33]. Equivalently this means that the sum of the R DFT$_1^\infty$ power spectra of the sequences at each spectral index is $\frac{R}{R'}$, i.e. the DFT$_1^\infty$ power spectral sum of the sequences is completely flat at all spectral indices. This implies that each of the R sequences has PA$\leq R$ wrt the DFT$_1^\infty$. We now modify the CS definition as follows,

Definition 7. *We define the* Complementary Set *(*CS Set*) to mean a set of sequences which is complementary wrt any specified set of unitary transforms, $\{\mathcal{T}\}$, such that the sum of the power spectra of the set of R sequences of length R', wrt any member of the set, $\mathcal{T}$, sum to $\frac{R}{R'}$ at each spectral index [16, 21]. Therefore, for $\mathbf{s}$ a member of the CS set, PAR$(\mathbf{s}) \leq R$.*

We formalise this as follows:

Definition 8. *The rows of an $R \times R'$ matrix, $\mathbf{A}$, form a complementary set of R sequences wrt the set of $R' \times R'$ unitary transform matrices, $\mathcal{T}$, iff, for every $\mathcal{U} \in \mathcal{T}$, $\mathbf{b_i} = \frac{R'}{R}\mathbf{A}\mathbf{u_i^T}$ is unitary, where $\mathbf{u_i}$ is the ith row of $\mathcal{U}$, and the rows of $\mathbf{A}$ are unitary.*

Lemma 1 provides an initial starting CS set for the example of the next section and the subsequent constructions:

Lemma 1. *Let $\mathbf{A}$ be a $R \times R$ unitary matrix. Then the rows of $\mathbf{A}$ form a* CS set *of R sequences wrt all $R \times R$ unitary matrices.*

Proof. Let $\mathbf{B}$ be an $R \times R$ matrix with rows, $\mathbf{b_i}$, where the $\mathbf{b_i}$ are constructed as in Definition 8. Then $\mathbf{B} = \mathbf{A}\mathcal{U}^\mathbf{T}$. Similarly $\mathbf{B}^\dagger = \mathcal{U}^*\mathbf{A}^\dagger$, where $*$ means conjugate. Then $\mathbf{B}\mathbf{B}^\dagger = \mathbf{A}\mathcal{U}^\mathbf{T}\mathcal{U}^*\mathbf{A}^\dagger = \mathbf{I_R}$, where $\mathbf{I_R}$ is the $R \times R$ identity matrix. Therefore $\mathbf{B}$ is unitary which means all $\mathbf{b_i}$ are unitary, and Lemma 1 follows from Definition 8. □

3 Construction Example

Before presenting the formal constructions of this paper, we first provide an example which highlights the main points of the constructions. For clarity of exposition we usually omit the normalisation constant for each matrix or sequence which would ensure the unitarity of the matrix or sequence. For instance, $\mathbf{A}$ below should be multiplied by $\frac{1}{2}$. We also provide and utilise ANFs, $p(x_0, x_1, \ldots, x_{n-1})$, for the binary sequence exponent of the bipolar sequences constructed, where the ith element, p_i of the length 2^n binary sequence, p, is given by $p(x_0 = i_0, x_1 = i_1, \ldots, x_{n-1} = i_{n-1})$, where $(i_0, i_1, \ldots, i_{n-1})$ is the 2-adic expansion of i. For instance, the function $p = x_0 + x_1$ has a truth table

x_0	x_1	p
0	0	0
1	0	1
0	1	1
1	1	0

which can be used to represent the sequence $(-1)^p = (-1)^{0110} = 1, -1, -1, 1$.

The construction strategy is as follows:

3.0.1 Choose Unitary Matrix

Choose, for example, the unitary matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} (-1)^0 \\ (-1)^{x_0} \\ (-1)^{x_1} \\ (-1)^{x_0+x_1} \end{pmatrix}$$

By Lemma 1 the four rows of $\mathbf{A}$ form a CS set wrt any 4×4 unitary matrix, i.e. any 4×4 4-LUUT. We can perform a number of operations on $\mathbf{A}$ to generate a length 16 bipolar sequence with 4-PAR ≤ 4.00 wrt any 4-LUUT (which includes any 2-LUUT).

3.0.2 Concatenate Rows:

Concatenating rows of $\mathbf{A}$ gives the length 16 sequence,

$$\mathbf{s} = 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1 \quad -1 \quad 1 \quad -1 \quad -1 \quad 1 \quad = (-1)^{x_0x_2+x_1x_3}$$

This sequence has 4-PAR$(\mathbf{s}) \leq 4.0$ wrt all 4-LUUTs including all 2-LUUTs. As will be shown, the upper bound is 4.0 because $\mathbf{A}$ is a 4×4 unitary matrix whose four rows form a CS set wrt all 4-LUUTs, which includes all 2-LUUTs. The transform set includes all 2-LUUTs because 2 divides 4. For example, $\mathbf{s}$ has PAs of 3.12, 1.00, and 4.00 wrt DFT$_1^\infty$, WHT, and NHT, respectively. (Note that PA wrt DFT$_1^\infty$ is computed approximately by taking the PA wrt enough 16×16 LUUTs of the form discussed in Definition 5. In other words we 'oversample' the one-dimensional DFT to sufficient precision).

3.0.3 Permute Rows and/or Columns Prior to Concatenation:

Choose any row/column permutation of $\mathbf{A}$ prior to concatenation. For example, choose the concatenation: Row 1 | Row 3 | Row 2 | Row 0, giving,

$$\begin{aligned} \mathbf{s} &= 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad -1 \quad -1 \quad 1 \quad 1 \quad 1 \quad -1 \quad -1 \quad 1 \quad 1 \quad 1 \quad 1 \\ &= (-1)^{x_0x_3+x_1x_2+x_1x_3+x_0} \end{aligned}$$

This sequence also has 4-PAR$(\mathbf{s}) \leq 4.0$ wrt all 4-LUUTs, including all 2-LUUTs. For example, $\mathbf{s}$ has PAs of 1.95, 1.00, and 1.00 under DFT$_1^\infty$, WHT, and NHT, respectively.

As another example, consider the column permutation: Col 3,Col 0,Col 2,Col 1, followed by the row permutation and concatenation: Row 2 | Row 3 | Row 0 | Row 1, giving,

$$\begin{aligned}\mathbf{s} &= -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad 1 \quad -1 \quad -1 \quad 1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1 \\ &= (-1)^{x_0x_2+x_0x_3+x_1x_2+x_0+x_2+x_3+1}\end{aligned}$$

This sequence also has PAR$(\mathbf{s}) \leq 4.0$ wrt all 4-LUUTs, including all 2-LUUTs. For example, $\mathbf{s}$ has PAs of 1.999, 1.00, and 1.00 wrt DFT$_1^\infty$, WHT, and NHT, respectively. (Note that for 4×4 matrices, a combined row and column permutation is equivalent to a row (or column) permutation. This is not generally the case for square matrix dimension > 4).

3.0.4 Generate Cosets

Let $\mathbf{g}$ be any length-4 bipolar vector. Let us express $\mathbf{A}$ as

$$\mathbf{A} = \begin{pmatrix} \mathbf{a_0} \\ \mathbf{a_1} \\ \mathbf{a_2} \\ \mathbf{a_3} \end{pmatrix}$$

where the a_i are length-4 bipolar vectors.

Let $\mathbf{A^g}$ be any matrix of the form,

$$\mathbf{A^g} = \begin{pmatrix} \mathbf{a_0} \odot \mathbf{g} \\ \mathbf{a_1} \odot \mathbf{g} \\ \mathbf{a_2} \odot \mathbf{g} \\ \mathbf{a_3} \odot \mathbf{g} \end{pmatrix}$$

where $\mathbf{a} \odot \mathbf{g} = (a_0g_0, a_1g_1, \ldots, a_3g_3)$, For instance, let $\mathbf{g} = (1, 1, 1, -1)$. Then,

$$\mathbf{A^g} = \begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 \end{pmatrix}$$

Then concatenation of any row/column permutation of $\mathbf{A^g}$ also has 4-PAR ≤ 4.0 wrt all 4-LUUTs, which includes all 2-LUUTs. As an example, consider the column permutation of $\mathbf{A^g}$: Col 0,Col 3,Col 2,Col 1, followed by the row permutation and concatenation: Row 1 | Row 3 | Row 0 | Row 2, giving,

$$\begin{aligned}\mathbf{s} &= 1 \quad 1 \quad 1 \quad -1 \quad 1 \quad -1 \quad -1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 1 \\ &= (-1)^{x_0x_1+x_0x_2+x_0x_3+x_1x_2}\end{aligned}$$

This sequence has 4-PAR$(\mathbf{s}) \leq 4.0$ wrt all 4-LUUTs, including 2-LUUTs. For example, $\mathbf{s}$ has PAs of 2.97, 1.00, and 2.00 wrt DFT$_1^\infty$, WHT, and NHT, respectively.

3.0.5 Symmetric Permutation:

Definition 9. *Let π be any permutation of Z_n. Then π_r is defined to be any* r-symmetric permutation *of Z_{r^n}, where $\pi_r(i) = \sum_{k=0}^{n-1} i_{\pi(k)} r^k$, and i has a radix-r decomposition as $\sum_{k=0}^{n-1} i_k r^k$, $i_k \in Z_r$, $\forall k$. We can then write the r-symmetric permutation of $\mathbf{s}$ as,*

$$\pi_r(\mathbf{s}) = (s_{\pi_r(0)}, s_{\pi_r(1)}, \ldots, s_{\pi_r(r^n-1)})$$

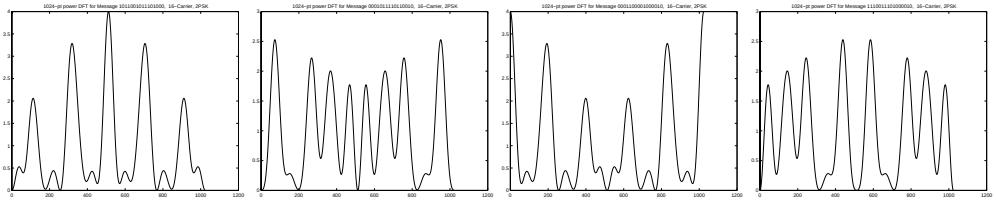


Figure 1: Power Spectrums for Size-4 Complementary Set, $\{\mathbf{s_0}, \mathbf{s_1}, \mathbf{s_2}, \mathbf{s_3}\}$, wrt DFT_1^∞ (x-axis is spectral index, y-axis is power value)

If $\mathbf{s}$ has 4-PAR ≤ 4.0 wrt all 4-LUUTs, then $\pi_2(\mathbf{s})$ has PAR ≤ 4.0 wrt all 2-LUUTs. (Note that because π_2 is a radix-2 permutation, the PAR upper bound no longer covers all 4-LUUTs). For instance, we have just stated that
$\mathbf{s} = 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, 1, 1, 1, -1, 1$ has 4-PAR ≤ 4.0 wrt all 4-LUUTs. Let $\pi = (0)(1, 2, 3)$ be a permutation of Z_4. Then π_2 permutes the indices of $\mathbf{s}$ according to $(0)(1)(2, 4, 8)(3, 5, 9)(6, 12, 10)(7, 13, 11)(14)(15)$ to give,

$$\begin{aligned}\mathbf{s} &= 1 \quad 1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1 \quad -1 \quad -1 \quad 1 \\ &= (-1)^{x_0x_1+x_0x_2+x_0x_3+x_2x_3}\end{aligned}$$

This sequence has $\mathrm{PAR}(\mathbf{s}) \leq 4.0$ wrt all 2-LUUTs. For example, $\mathbf{s}$ has PAs of 2.56, 1.00, and 2.00 wrt DFT_1^∞, WHT, and NHT, respectively.

3.0.6 Form Complementary Sequence (CS) Set:

Let $\mathbf{E}$ be another 4×4 unitary matrix (it could be the same as $\mathbf{A}$). For example,

$$\mathbf{E} = \begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{pmatrix}$$

where the element at row i and column j is $e_{i,j}$. For any row and/or column permutation of $\mathbf{A}$ (or $\mathbf{A^g}$) we can form four length-16 CS. For instance, from subsection 3.0.4, let our constructed sequence be,
$\mathbf{s} = \mathbf{a_0^g}|\mathbf{a_1^g}|\mathbf{a_2^g}|\mathbf{a_3^g} = 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, 1, 1, 1, -1, 1$, where
$\mathbf{a_0^g} = 1, 1, 1, -1$, $\mathbf{a_1^g} = 1, -1, -1, -1$, $\mathbf{a_2^g} = 1, -1, 1, 1$, $\mathbf{a_3^g} = 1, 1, -1, 1$. Then our size-4 CS set is:

$$\begin{array}{lcccccccccccccccc}
\mathbf{s_0} = e_{0,0}\mathbf{a_0^g}|e_{0,1}\mathbf{a_1^g}|e_{0,2}\mathbf{a_2^g}|e_{0,3}\mathbf{a_3^g} = & + & + & + & - & + & - & - & - & + & - & + & + & - & - & + & - \\
\mathbf{s_1} = e_{1,0}\mathbf{a_0^g}|e_{1,1}\mathbf{a_1^g}|e_{1,2}\mathbf{a_2^g}|e_{1,3}\mathbf{a_3^g} = & + & + & + & - & + & - & - & - & - & + & - & - & + & + & - & + \\
\mathbf{s_2} = e_{2,0}\mathbf{a_0^g}|e_{2,1}\mathbf{a_1^g}|e_{2,2}\mathbf{a_2^g}|e_{2,3}\mathbf{a_3^g} = & + & + & + & - & - & + & + & + & + & - & + & + & + & + & - & + \\
\mathbf{s_3} = e_{3,0}\mathbf{a_0^g}|e_{3,1}\mathbf{a_1^g}|e_{3,2}\mathbf{a_2^g}|e_{3,3}\mathbf{a_3^g} = & - & - & - & + & + & - & - & - & + & - & + & + & + & + & - & +
\end{array}$$

where '$+$' is 1 and '$-$' is -1.

Then $|\mathbf{s_0} \cdot \mathbf{l}|^2 + |\mathbf{s_1} \cdot \mathbf{l}|^2 + |\mathbf{s_2} \cdot \mathbf{l}|^2 + |\mathbf{s_3} \cdot \mathbf{l}|^2 = 4.0$ for $\mathbf{l}$ 4-linear. In other words, the four sequences, $\mathbf{s_i}$, form a size-4 CS set wrt any 4-LUUT, which includes any 2-LUUT, as the sum of their power spectrums wrt any 4-LUUT is a constant at every point. Therefore each sequence satisfies 4-$\mathrm{PAR}(\mathbf{s_i}) \leq 4.0$ wrt any 4-LUUT, which includes any 2-LUUT. The power spectrums wrt DFT_1^∞ for each sequence of the above CS set are shown in Fig 1, and the spectrums sum to 4.0 at each spectral index. The power spectrums wrt the 16-point

WHT for each of the four sequences are as follows:

Sequence	Power Spectrum															
$\mathbf{s_0}$	0	4	0	0	4	0	0	0	0	0	4	0	0	0	0	4
$\mathbf{s_1}$	0	0	4	0	0	0	0	4	0	4	0	0	4	0	0	0
$\mathbf{s_2}$	4	0	0	0	0	4	0	0	0	0	0	4	0	0	4	0
$\mathbf{s_3}$	0	0	0	4	0	0	4	0	4	0	0	0	0	4	0	0

The power spectrums wrt the 16-point NHT for each of the four sequences are as follows:

Sequence	Power Spectrum															
$\mathbf{s_0}$	2	0	2	0	0	2	0	2	2	0	2	0	0	2	0	2
$\mathbf{s_1}$	2	0	2	0	0	2	0	2	2	0	2	0	0	2	0	2
$\mathbf{s_2}$	0	2	0	2	2	0	2	0	0	2	0	2	2	0	2	0
$\mathbf{s_3}$	0	2	0	2	2	0	2	0	0	2	0	2	2	0	2	0

In all cases the power spectrums sum to 4.0 at each point. Furthermore, the sequences, $\pi_2(\mathbf{s_i})$ also form a size-4 CS set wrt any 2-LUUT, for any π_2.

3.0.7 Iterate Construction:

Let us now assign

$$\mathbf{A}' = \begin{pmatrix} \mathbf{s_0} \\ \mathbf{s_1} \\ \mathbf{s_2} \\ \mathbf{s_3} \end{pmatrix} = \begin{pmatrix} e_{0,0}\mathbf{a_0^g} & e_{0,1}\mathbf{a_1^g} & e_{0,2}\mathbf{a_2^g} & e_{0,3}\mathbf{a_3^g} \\ e_{1,0}\mathbf{a_0^g} & e_{1,1}\mathbf{a_1^g} & e_{1,2}\mathbf{a_2^g} & e_{1,3}\mathbf{a_3^g} \\ e_{2,0}\mathbf{a_0^g} & e_{2,1}\mathbf{a_1^g} & e_{2,2}\mathbf{a_2^g} & e_{2,3}\mathbf{a_3^g} \\ e_{3,0}\mathbf{a_0^g} & e_{3,1}\mathbf{a_1^g} & e_{3,2}\mathbf{a_2^g} & e_{3,3}\mathbf{a_3^g} \end{pmatrix}$$

for any size-4 CS set of length 16 sequences, $\mathbf{s_i}$, as constructed using the previous subsections. Let $\mathbf{E}'$ be any 4×4 unitary matrix. For instance,

$$\mathbf{E}' = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$\mathbf{A}'$ takes the place of the $\mathbf{A}$ matrix discussed previously. We again perform row permutation or column-segment permutation of $\mathbf{A}'$, with optional coset offset and symmetric permutation to construct sequences of length 64 with 4-PAR ≤ 4.0 wrt any 4-LUUT. (Note that we refer to column-segment permutation because we only permute the 4 segments of each row of $\mathbf{A}'$). $\mathbf{E}'$ now takes the place of the $\mathbf{E}$ matrix discussed previously, allowing us to construct size-4 CS sets of length 64 whose power spectrums sum to a constant wrt any 4-LUUT. For example, we can concatenate the sequences $\mathbf{s_i}$, constructed in subsection 3.0.6, to get the length 64 sequence,

$$\begin{aligned} &\texttt{+++-+---+-++--+-+++-+----+--++-++++--++++-++++-+---++---+-++++-+} \\ &= (-1)^{x_0x_1+x_0x_2+x_0x_3+x_1x_2+x_2x_3+x_2x_5+x_3x_4+x_4x_5} \end{aligned}$$

where '+' and '$-$' are short for 1 and -1, respectively. This sequence has 4-PAR$(\mathbf{s}) \leq 4.0$ wrt all 4-LUUTs, including all 2-LUUTs. For example, $\mathbf{s}$ has PAs of 3.01, 1.00, and 1.00 wrt DFT_1^∞, WHT, and NHT, respectively. Using $\mathbf{E}'$ we can construct the size-4 CS set,

```
+++-+---+-+--+-+++-+----+-++-++++--++++-++++-+---++---+-++++-+
+++-+---+-+++-+-+--+-++++-+--+-++++--++++-++++-+---++---+-++--+
+++-+---+-+--+-+++-+----+-++-++++--++++-++++-+---++---+-++++-+-
+++-+---+-+++-+-+--+-++++-+--+-++++--++++-++++-+---++---+-++--+
```

which is a set of 4 length-64 sequences whose power spectrums sum to a constant wrt any 4-LUUT, which includes any 2-LUUT.

We can iterate the contruction as many times as we like to produce sequences of length 2^{2L} for some positive integer L, where each sequence has 4-PAR ≤ 4.0 wrt any 4-LUUT. (If there is symmetric permutation by π_2 then each sequence generally only has PAR ≤ 4.0 wrt any 2-LUUT, not any 4-LUUT).

3.0.8 Summary of Example Construction

We summarise the construction operations as follows:

- 1. Choose a 4×4 unitary matrix, $\mathbf{A}$.
- 2. Permute rows and/or columns of $\mathbf{A}$.
- 3. Select length-4 sequence, $\mathbf{g}$, to act as coset offset for $\mathbf{A}$.
- 4. Choose 4×4 unitary matrix, $\mathbf{E}$.
- 5. Concatenate the rows of (permuted coset of) $\mathbf{A}$ and multiply each row-segment by the appropriate entry in $\mathbf{E}$, for each row of $\mathbf{E}$, to form a size-4 CS set of length 16 sequences with 4-PAR ≤ 4.0 wrt any 4-LUUT. Define this 4-set as a 4×16 matrix, $\mathbf{A}'$.
- 6. Iterate the construction L times by looping back to step 2, where $\mathbf{A}$, $\mathbf{E}$ and $\mathbf{g}$ are replaced by $\mathbf{A}'$, a new 4×4 unitary matrix, $\mathbf{E}'$, and a new length-4 unitary vector, $\mathbf{g}'$, respectively.
- 7. Finally, symmetrically permute each sequence in the size-4 CS set, using the same permutation, π_2, for each sequence, and define this set as a 4×4^L matrix, each row of which has PAR ≤ 4.0 wrt any 2-LUUT, and such that the four rows form a size-4 CS set.

Our construction can be fully specified by the sequence of 4×4 unitary matrices, $\mathbf{E_j}$, where $\mathbf{A} = \mathbf{A_0} = \mathbf{E_0}$, by the row/column permutations over Z_4 at each iteration, the coset offset at each iteration, the number of iterations of the construction, and the final symmetric permutation over $Z_{2^{2L}}$. Using this construction we can generate a vast number of sequences with low PAR wrt any 2-LUUT. However, the difficulty with the construction arises because the above constructive operations are not disjoint (orthogonal), so it is problematic to count the complete sequence set, and to design hardware/software to implement the construction without gneerating a (small) fraction of the sequences more than once. We tackle the quadratic case in subsection 4.4.

In subsection 4 we formalise the construction and generalise to r-PAR $\leq R$, for any R by using $R \times R$ matrices, $\mathbf{E_j}$, to recursively construct matrices, $\mathbf{A_j}$. Instead of applying the row/column permutations and coset offset to the $\mathbf{A_j}$ matrices, we shall, equivalently, apply these operations to the $\mathbf{E_j}$ matrices.

4 Constructions

4.1 Construction 1

Let $N = r^n$, $R = r^t$. Let $\mathbf{E_j}$ and $\mathbf{A_j}$, $0 \le j < L$, be a sequence of $R \times R$ and $R \times R^{j+1}$ complex matrices, respectively, $\mathbf{E_j}$ a unitary, unimodular matrix with rows $\mathbf{e_{i,j}}$, $\mathbf{A_j}$ with unitary, unimodular rows, $\mathbf{a_{i,j}}$. Let γ_j and θ_j permute Z_R, and $\mathbf{E'_j}$, with rows $\mathbf{e'_{i,j}}$, be the row/column permutation of $\mathbf{E_j}$, specified by γ_j and θ_j, respectively. Let $\mathbf{A_0} = \mathbf{E'_0}$. Then $\mathbf{A_j}$ is formed as,

$$\mathbf{a_{i,j}} = (\mathbf{a_{0,j-1}}|\mathbf{a_{1,j-1}}|\dots|\mathbf{a_{R-1,j-1}}) \odot (\mathbf{1} \otimes \mathbf{e'_{i,j}}) \tag{2}$$

where $\mathbf{x} \odot \mathbf{y} = (x_0 y_0, x_1 y_1, \dots, x_{R^j-1} y_{R^j-1})$, $\mathbf{1}$ is the length R^j all-ones vector, $'|'$ means concatenation, and $\mathbf{e'_{i,j}}$ is the ith row of $\mathbf{E'_j}$.

Theorem 1. *Let $\mathbf{s}$ be a length $N = R^L$ row of $\mathbf{A_{L-1}}$. Then $\pi_r(\mathbf{s})$ satisfies r-PAR$(\pi_r(\mathbf{s})) \le R$ wrt all $N \times N$ r-LUUTs, where π_r is any r-symmetric permutation of $\mathbf{s}$.*

Proof. Assume the rows of $\mathbf{A_{j-1}}$ form a size-R CS set wrt any r-LUUT. Let $\mathbf{l_j}$ and $\mathbf{l}$ be unitary unimodular r-linear rows of length R^{j+1} and R, respectively. Let $\mathbf{b} = R^{j-1}\mathbf{A_{j-1}}\mathbf{l}^{\mathbf{T}}_{\mathbf{j-1}}$. Then, by Definition 8, $\mathbf{b}$ is unitary. By Definitions 2,4,8, the rows of $\mathbf{A_j}$ must form a size-R CS set wrt any r-LUUT if $\mathbf{b}' = R^j \mathbf{A_j}(\mathbf{l_{j-1}} \otimes \mathbf{l})^T$ is unitary $\forall \mathbf{l_{j-1}}, \mathbf{l}$. This follows because $b'_i = \sum_{k=0}^{R-1} (\mathbf{a_{k,j-1}}\mathbf{l}^{\mathbf{T}}_{\mathbf{j-1}})(e'_{i,j,k} l_k) = \sum_{k=0}^{R-1} b_k e'_{i,j,k} l_k$ for $b'_k, b_k, e'_{i,j,k}$ and l_k the kth elements of $\mathbf{b}'$, $\mathbf{b}$,$\mathbf{e'_{i,j}}$ and $\mathbf{l}$, respectively. To make $\mathbf{b}'$ unitary, we require $P = R\sum_{i=0}^{R-1} |b'_i|^2 = R\sum_{i=0}^{R-1} |\sum_{k=0}^{R-1} (b_k e'_{i,k} l_k)|^2 = 1$. Let $\mathbf{z} = \sqrt{R}(b_0 l_0, b_1 l_1, \dots, b_{R-1} l_{R-1})^T$, and $\mathbf{Z} = \mathbf{E'_j z}$. Then $P = 1$ if $\mathbf{Z}$ is unitary, which follows by Parseval's Theorem if $\mathbf{E'_j}$ is a unitary matrix, and if $\mathbf{z}$ is unitary. $\mathbf{E'_j z}$ is a unitary matrix and $\mathbf{z}$ is unitary because $\mathbf{b}$ is unitary and $\mathbf{l}$ is unitary unimodular. It follows that the rows of $\mathbf{A_j}$ form a size-R CS set if the rows of $\mathbf{A_{j-1}}$ form a size-R CS set. The induction is completed by noting that the rows of $\mathbf{A_0} = \mathbf{E'_0}$ form a size-R CS set. Finally, any r-symmetric permutation of $\mathbf{s}$ is allowed because $\mathbf{l}$ and $\mathbf{l_j}$ are both r-linear. □

Note that, if $\mathbf{l_j}$ is not unimodular then Theorem 1 does not, in general, hold.

It is interesting to observe that the Hadamard matrix construction of [14] is related to the constructions of this paper. Using the terminology of [14], their construction is,

$$\mathbf{H} = \begin{pmatrix} c_{11} + \mathbf{B_1} & c_{12} + \mathbf{B_2} & \dots & c_{1m} + \mathbf{B_m} \\ c_{21} + \mathbf{B_1} & c_{22} + \mathbf{B_2} & \dots & c_{2m} + \mathbf{B_m} \\ \dots & \dots & \dots & \dots \\ c_{m1} + \mathbf{B_1} & c_{m2} + \mathbf{B_2} & \dots & c_{mm} + \mathbf{B_m} \end{pmatrix}$$

where $\mathbf{C} = [c_{ij}]$, the $\mathbf{B_i}$ are $T \times T$ Hadamard matrices, and their alphabet comprises $\{0, 1\}$ instead of $\{1, -1\}$, and they use '+', mod 2, instead of $\times$. One can relate this construction to the first iteration of Construction 1 of our paper by equating our $\mathbf{E}$ matrix with their $\mathbf{C}$ matrix, assigning $T = m = R$, and by assigning $\mathbf{B_{i+1}}$ to be derived from $\mathbf{B_i}$ where every column of $\mathbf{B_i}$ is cyclically shifted round by one position. Then we pick out every Rth row of $\mathbf{H}$ to form a CS set of R sequences of length R^2, where every sequence has PAR $\le R$ wrt all LUUTs. There are R such sets. It would be interesting to develop a classification of Hadamard matrices according to the worst-case PAR of the rows of the matrix.

PAR ≤ 8.0

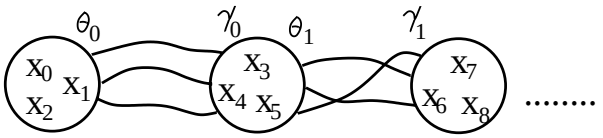


Figure 2: Construction 2 for $t = 3$

4.2 Construction 2 (special case of Construction 1)

Consider Construction 1. Let $r = 2$ and all $\mathbf{E_j}$ be $2^t \times 2^t$ WHTs. Let $\mathbf{x} = \{x_0, x_1, \ldots, x_{n-1}\}$ be n binary variables. Then $\mathbf{s} = 2^{\frac{-n}{2}}(-1)^{\mathbf{p(x)}}$, where,

$$p(\mathbf{x}) = \sum_{j=0}^{L-2} \theta_j(\mathbf{x_j})\gamma_j(\mathbf{x_{j+1}}) + \sum_{j=0}^{L-1} g_j(\mathbf{x_j}) \tag{3}$$

where θ_j and γ_j are any permutations: $Z_2^t \rightarrow Z_2^t$,
$\mathbf{x_j} = \{x_{\pi(tj)}, x_{\pi(tj+1)}, \ldots, x_{\pi(t(j+1)-1)}\}$, $n = Lt$, π permutes Z_n, and g_j is any function from $Z_2^t \rightarrow Z_2$.

To clarify (3) note that, $\forall j$, we can define $\rho(\mathbf{x_j}, \mathbf{x_{j+1}}) = \theta_j(\mathbf{x_j})\gamma_j(\mathbf{x_{j+1}})$ such that ρ can be expanded as the function $\rho : Z_2^{2t} \rightarrow Z_2$, $\rho(\mathbf{x_j}, \mathbf{x_{j+1}}) = \theta_{0,j}(\mathbf{x_j})\gamma_{0,j}(\mathbf{x_{j+1}}) + \theta_{1,j}(\mathbf{x_j})\gamma_{1,j}(\mathbf{x_{j+1}}) + \ldots + \theta_{t-1,j}(\mathbf{x_j})\gamma_{t-1,j}(\mathbf{x_{j+1}})$ where $\theta_j = (\theta_{0,j}, \theta_{1,j}, \ldots, \theta_{t-1,j})$, $\gamma_j = (\gamma_{0,j}, \gamma_{1,j}, \ldots, \gamma_{t-1,j})$ and all $\theta_{i,j}, \gamma_{i,j}$ are balanced functions : $Z_2^t \rightarrow Z_2$, chosen so that θ_j and γ_j are permutations.

Corollary 1. *The length $N = 2^n$ sequences, $\mathbf{s}$, of Construction 2, satisfy PAR$(\mathbf{s}) \leq 2^t$ wrt all $N \times N$ LUUTs.*

Proof. Construction 2 is a special case of Construction 1 where all $\mathbf{E_j}$ are $2^t \times 2^t$ WHTs. The Corollary therefore follows from Theorem 1. □

When $L = 2$ and when θ or γ is the identity permutation, then Construction 2 reduces to the Maiorana McFarland construction over $2t$ variables. [1] It is helpful to illustrate Construction 2 graphically, and Fig 2 illustrates the construction for $t = 3$, where we are also free to permute the indices, i, of x_i using π. An example for Fig 2 could be,

$$\begin{aligned} p(\mathbf{x}) = \quad & (x_0)(x_3 + x_5) + (x_1)(x_5) + (x_1 + x_2)(x_4) + (x_3 + x_4)(x_6 + x_7 + x_8) \\ & + (x_3)(x_6) + (x_5)(x_7) + g_0(x_0, x_1, x_2) + g_1(x_3, x_4, x_5) + g_2(x_6, x_7, x_8) \end{aligned}$$

where g_0, g_1, g_2 are any functions: $Z_2^3 \rightarrow Z_2$. This example has guaranteed 8-PAR ≤ 8.0 wrt all 8-LUUTs, which includes all 2-LUUTs, but with index permutation of the x_i, PAR ≤ 8.0 is only guaranteed wrt all 2-LUUTs.

Theorem 2. *For fixed t, let $\mathbf{P}$ be the subset of $p(\mathbf{x})$ of degree μ or less, generated using Construction 2. Then $D \geq 2^{n-\mu}$, where D is the Hamming Distance between members of*

[1] Thanks to V.Rijmen for pointing out the Maiorana-McFarland connection.

P, *and,*

$$\begin{aligned} |\mathbf{P}| &\leq B = \tfrac{n!}{\Gamma}(\tfrac{2^{t+\binom{t}{2}}\Gamma}{t!})^{\frac{n}{t}} && \mu = 2 \\ &\leq B = \tfrac{n!}{V}(\tfrac{2^{2^t-1}V}{t!})^{\frac{n}{t}} && \mu = 2t-2, t>1 \end{aligned} \tag{4}$$

where $\Gamma = \prod_{i=0}^{t-1}(2^t - 2^i) = |GL(t,2)|$*, (GL is the General Linear Group), and* $V = ((2^t-1)!)^2 - (\Gamma^2 - \Gamma)$*. (For* $t = 1$ *the bound is exact). (Note that this paper does not give upper bounds on the size of* $\mathbf{P}$ *for the intermediate cases where* $2 < \mu < 2t-2$*.)*

Proof. The result on Hamming Distance, D, is a well-known propery of Reed-Muller codes [13]. Let us now prove (4). When $\mu = 2$ then θ and γ are linear permutations. In this case the two-way permutation, $\mathbf{x_j}\gamma(\mathbf{x_{j+1}})$, covers the same set of permutations as $\theta(\mathbf{x_j})\gamma(\mathbf{x_{j+1}})$. So we can set θ to the identity permutation. Each term, $\mathbf{x_j}\gamma_j(\mathbf{x_{j+1}})$, for γ_j linear, is isomorphic to $\mathrm{GL}(t,2)$, where GL is the General Linear Group. Therefore we can represent the linear permutations at each iteration by the set, $\mathrm{GL}(t,2)$ of binary invertible $t \times t$ matrices, where $\Gamma = |\mathrm{GL}(t,2)| = \prod_{i=0}^{t-1}(2^t - 2^i)$. For $L = \frac{n}{t}$ and $L-1$ iterations we have Γ^{L-1} possible combinations of permutations. There are $\frac{1}{2}\prod_{i=1}^{L}\binom{it}{t}$ ways of ordering a linked line of subsets of t disjoint variables out of n variables. At each iteration we can choose g_j from one of $2^{t+\binom{t}{2}}$ quadratic functions of t variables. Over L iterations we therefore have a choice of $(2^{t+\binom{t}{2}})^L$ combinations of functions, g_j. The first part of (4) follows by noting that $\prod_{i=1}^{L}\binom{it}{t} = \frac{n!}{(t!)^L}$.
The case $\mu = 2t-2$ occurs when θ and γ are permutation polynomials each up to degree $t-1$ ($t-1$ is the maximum possible degree of a permutation polynomial from $Z_2^t \to Z_2^t$). Therefore each of θ and γ can be chosen from $\frac{(2^t)!}{2^t}$ different polynomials to make a total of $\left(\frac{(2^t)!}{2^t}\right)^2$ polynomial configurations for one iteration. [2] However remember that the case of $\theta\gamma$ quadratic corresponds to θ and γ both linear in which case we can, without loss of generality, make θ the identity. Therefore instead of contributing Γ^2 configurations, the case of $\theta\gamma$ quadratic contributes only Γ configurations, so the total number of polynomial configurations after one iteration is $V = \frac{(2^t)!}{2^t} - (\Gamma^2 - \Gamma)$. Therefore, after $L-1$ iterations we have V^{L-1} possible combinations of permutations. We therefore replace Γ in the first line of equation (4) with V. At each iteration we can now choose g from one of 2^{2^t-1} functions of t variables of degree $\leq t$ (ignoring constant offset). The second part of (4) follows. □

Definition 10. *A* $[2^n, k, D, W]$ *nonlinear error-correcting code has length* 2^n*, dimension* k *(*$\log_2$ *of the number of codewords), Hamming Distance* D*, and each codeword has PAR* $\leq W$ *wrt all LUUTs.*

Corollary 2. *Application of Construction 2 and reference to Theorem 2 allows us to construct and parameterise* $[2^n, \log_2(|\mathbf{P}|), 2^{n-\mu}, 2^t]$ *nonlinear error-correcting codes.*

4.3 Examples for Construction 2

The WHT, NHT, and DFT_1^∞ are used as 'spot-checks' in the following examples to validate the PAR upper-bound. Furthermore, the PAR is lower-bounded by the maximum PAR resulting from these three spot-checks.

[2] Note that we divide by 2^t so as not to include all offsets of the permutation θ (or γ) by the constant '1', i.e. we ignore permutations which have one or more constituent elements of the form $\theta_{i,j}(\mathbf{x_j}) + 1$ (or $\gamma_{i,j}(\mathbf{x_j}) + 1$). These constant offsets to the permutations are implicitly included by suitable assignments to the g polynomials in (5).

There are, of course, an infinite number of LUUTs, all of which validate the PAR upper-bound for the constructed set.

4.3.1 Example 1, Identity Permutations

Let θ_j and γ_j be identity permutations $\forall j$. Then, $\theta(\mathbf{x_j}) = \gamma(\mathbf{x_j}) = \mathbf{x_j}$ and Construction 2 becomes,

$$p(\mathbf{x}) = \sum_{j=0}^{L-2} \sum_{l=0}^{t-1} x_{\pi(tj+l)} x_{\pi(t(j+1)+l)} + \sum_{j=0}^{L-1} g_j(\mathbf{x_j}) \tag{5}$$

When $\deg(g_j) < 2$, $\forall j$, it is well-known that $\mathbf{s} = 2^{\frac{-n}{2}}(-1)^{p(\mathbf{x})}$ is Bent (PA $= 1$ wrt the WHT) for L even [13] and (perhaps not known) that $\mathbf{s}$ has PA $= 2^{\,t}$ wrt the WHT for L odd. In general, for any g_j, $\mathbf{s}$ has PAR $\leq 2^t$ wrt all LUUTs. For example, if $L = 4$, $t = 3$, and $p(\mathbf{x}) = x_0x_3 + x_1x_4 + x_2x_5 + x_3x_6 + x_4x_7 + x_5x_8 + x_6x_9 + x_7x_{10} + x_8x_{11}$, then $\mathbf{s}$ has PA $= 1.0$ wrt WHT and NHT, and PA $= 7.09$ wrt DFT_1^∞. Similarly, let $g_0(x_0, x_1, x_2) = x_1x_2$, $g_1(x_3, x_4, x_5) = x_3x_4x_5$, and $g_2(x_6, x_7, x_8) = 0$. Then $\mathbf{s}' = 2^{\frac{-n}{2}}(-1)^{p(\mathbf{x})+g_0+g_1+g_2}$ has PAs 4.0, 2.0, and 7.54 wrt WHT, NHT, and DFT_1^∞, respectively. In all cases, PAR $\leq 2^t = 8.0$.

4.3.2 Example 2, PAR ≤ 2.0, ($t = 1$)

Let $t = 1$. We need only consider the identity permutations, $\theta_j(x_{\pi(j)}) = \gamma_j(x_{\pi(j)}) = x_{\pi(j)}$, as $\theta_j(x_{\pi(j)}) = \gamma_j(x_{\pi(j)}) = x_{\pi(j)} + 1$ is implicitly covered by $g_j(\mathbf{x_j})$. From Construction 2,

$$p(\mathbf{x}) \quad = \textstyle\sum_{j=0}^{L-2} x_{\pi(j)} x_{\pi(j+1)} + c_j x_j + k, \qquad c_j, k \in Z_2 \tag{6}$$

This is exactly the DJ set of binary quadratic cosets of RM$(1, n)$, where $n = L$, as described by Davis and Jedwab [4]. This set has PA ≤ 2.0 wrt DFT_1^∞ [4]. Such sequences are Bent for n even [13, 26] and, in [16, 17] it is shown that such a set has PA $= 2.0$ wrt WHT for n odd, and also, wrt NHT, has PA $= 1.0$ for $n \neq 2 \bmod 3$ (NegaBent), and PA $= 2.0$ for $n = 2 \bmod 3$. More generally the DJ set has PAR ≤ 2.0 wrt any LUUT [17], and this agrees with Theorem 1. For example, let $p(\mathbf{x}) = x_0x_4 + x_4x_1 + x_1x_2 + x_2x_3 + x_1 + 1$. Then $\mathbf{s}$ has PAR $= 2.0$ wrt the WHT, NHT, and DFT_1^∞. The DJ set, being cosets of $R(2, n)$, forms a codeset with Hamming Distance, $D \geq 2^{n-2}$. The rate of the DJ codeset is $\frac{(\frac{n!}{2})2^{n+1}}{2^{2^n}}$. Therefore we can construct a $[2^n, \log_2(n!) + n, 2^{n-2}, 2.0]$ error-correcting code. The primary drawback of this code is that its rate vanishes rapidly as n increases.

4.3.3 Example 3, PAR ≤ 4.0, ($t = 2$)

[4, 24, 16, 17, 26] all propose techniques for the inclusion of further quadratic cosets, so as to improve rate at the price of increased PAR. We here propose an improved rate quadratic code (although still vanishing, asymptotically), where PAR ≤ 4.0. To achieve this we set $t = 2$ in Construction 2. For $t = 2$ then the algebraic degree of all sequences is $\mu = 2$. Therefore, as stated in the proof of Theorem 2, we can set θ to the identity permutation. There are $\Gamma = \frac{(2^t)!}{2^t} = 6$ non-trivial linear permutation polynomials, γ_j, (ignoring constant offset). These polynomials map from $Z_2^2 \to Z_2^2$, and comprise the set, $\gamma(x_r, x_s) \in \{(x_r, x_s), (x_r + x_s, x_s), (x_r, x_r + x_s), (x_s, x_r), (x_r + x_s, x_r), (x_s, x_r + x_s)\}$. Substituting for γ_j and g_j in Construction 2 gives a large set of polynomials with PAR≤ 4.0 wrt all LUUTs. We now list, for this construction, the $p(\mathbf{x})$ arising from the 6 invertible polynomials, γ, for one 'iteration' of Construction 2, i.e. for $L = 2$, where $n = Lt = 4$, and where we fix π to the identity.

$$
\begin{aligned}
p(\mathbf{x}) &= x_0x_2 + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)\\
p(\mathbf{x}) &= x_0(x_2 + x_3) + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)\\
p(\mathbf{x}) &= x_0x_2 + x_1(x_2 + x_3) + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)\\
p(\mathbf{x}) &= x_0x_3 + x_1x_2 + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)\\
p(\mathbf{x}) &= x_0(x_2 + x_3) + x_1x_2 + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)\\
p(\mathbf{x}) &= x_0x_3 + x_1(x_2 + x_3) + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)
\end{aligned}
\tag{7}
$$

where $c_0, c_1 \in Z_2$. The permutations, γ_j, above are isomorphic to a distinct invertible boolean $t \times t$ matrix, where $t = 2$ (Section 4.4), as the permutation polynomials form a group isomorphic to the binary General Linear Group, $\text{GL}(t,2)$, where $|\text{GL}(t,2)| = \prod_{i=0}^{t-1}(2^t - 2^i)$ [11]. Explicitly,

$$
\text{GL}(2,2) = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \right\}
$$

Note that, by inspection, any two of the quadratics in (7) are inequivalent under permutation, π, of the indices of the four variables, e.g., $p(\mathbf{x}) = x_0x_2 + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)$ and $p(\mathbf{x}) = x_0(x_2 + x_3) + x_1x_3 + c_0x_0x_1 + c_1x_2x_3 + \text{RM}(1,4)$. An upper bound, B, on $|\mathbf{P}|$ is given by Theorem 2. Substituting $t = 2$ into (4),

$$
|\mathbf{P}| < B = \frac{n!}{6} 24^{\frac{n}{2}} \tag{8}
$$

Therefore we can construct a $[2^n, \log_2(|\mathbf{P}|), 2^{n-2}, 4.0]$ error-correcting code. Exact enumeration and unique generation for this set remains open, due to extra symmetries, induced by π, which occur for $t > 1$. As an example of this π-induced symmetry, consider the two coset leaders, $x_0x_2 + x_1x_3 + g_0(x_0, x_1) + g_1(x_2, x_3)$ and $x_0x_1 + x_2x_3 + g'_0(x_0, x_2) + g'_1(x_1, x_3)$ which both contribute to the count in the above enumeration, but are equal when $g_0(x_0, x_1) = x_0x_1$, $g_1(x_2, x_3) = x_2x_3$, $g'_0(x_0, x_2) = x_0x_2$, $g'_1(x_1, x_3) = x_1x_3$. This equality leads to an overcount and such symmetries render B a strict upper bound for all cases but $t = 1$. We computed the exact number of quadratic coset leaders for $n = 4, 6, 8, 10$, by simply counting the number of distinct coset leaders, and these are compared to the upper bound, B, of (8) in Table 1. They are also compared to the $\frac{n!}{2}$ quadratic coset leaders in the binary DJ set (Example 2). Thus, for instance, Table 1 shows the existence of a $[64, 20.2, 16, 4.0]$ low PAR error-correcting code, i.e. of length 64, dimension $k = 20.2$, distance $D = 16$, and PAR ≤ 4.0, which can be compared with the fundamental DJ binary codeset for $n = 6$, which is a $[64, 15.5, 16, 2.0]$ low PAR error-correcting code. We see that rate has been improved over the DJ codeset at the price of PAR, which also increases. Thus, by assigning $t = 2$ we have a construction for a much larger codeset than

Table 1: The Number of Quadratic Coset Leaders for Construction 2 when $t = 2$

n	4	6	8	10
Theorem 2, (8),(4), $B/2^{n+1}$	72	12960	4354560	2351462400
Exact Computation(3), $\lvert\mathbf{P}\rvert/2^{n+1}$	36	9240	4086096	2317593600
$\lvert$DJ Code $\rvert/2^{n+1}$	12	360	20160	1814400
$\log_2(B/2^{n+1})$	6.2	13.7	22.1	31.1
$\log_2(\lvert\mathbf{P}\rvert/2^{n+1})$	5.2	13.2	22.0	31.1
$\log_2$(Number of homogeneous quadratics)	6	15	28	45

the DJ codeset and with the same Hamming Distance, $D = 2^{n-2}$, but now PAR is upper-bounded by 4.0 instead of 2.0. Table 1 also shows the $\log_2$ of the size of the complete set of homogeneous quadratic functions, and it is evident from Table 1 that $\mathbf{P}$ contains a

significant proportion of these homogeneous quadratic functions for $n \leq 10$. Note that, as n increases, the discrepancy between the upper bound, B, and $|\mathbf{P}|$ becomes negligible as a fraction of $|\mathbf{P}|$. Therefore, in practice, for $n \geq 10$, it may be acceptable, from the viewpoint of an engineer who wishes to use this codeset in an OFDM system, to incorporate the coding collision errors induced by π into the overall error-rate without significant detriment to performance. In which case we can already claim to have constructed an *implementable* low PAR error-correcting code for OFDM systems using 1024 or more carriers which is significantly larger than any previously proposed that uses construction techniques. However Table 1 also indicates that the rate of this code is still unacceptably small for $n \geq 10$. For instance, from Table 1, when $n = 10$, we see that the code rate of $\mathbf{P}$ is $\frac{42.1}{1024}$, which is very small.

As an example of a codeword from this set, let $p(\mathbf{x}) = x_0x_2 + x_1x_2 + x_1x_6 + x_2x_5 + x_6x_3 + x_6x_5 + x_5x_4 + x_3x_7 + x_0x_1 + x_5x_3 + x_7 + x_1$. Then $\mathbf{s}$ has PAs $= 1.0$, 2.0, and 3.43 wrt WHT, NHT, and DFT_1^∞, respectively.

Table 2: The Number of Quadratic Coset Leaders for Construction 2 when $t = 3$

n	6	9	12	15
$\log_2(B/2^{n+1})$	16.7	33.5	51.7	70.9
$\log_2$(Number of homogeneous quadratics)	15	36	66	105

4.3.4 Example 4, PAR ≤ 8.0, ($t = 3$)

There are now $\frac{(2^t)!}{2^t} = 5040$ non-trivial permutation polynomials from $Z_2^3 \to Z_2^3$, and of linear or quadratic degree for each of θ, and γ (ignoring constant-offset). Thus, $\theta\gamma$ can be quadratic, cubic or quartic according to the subset of permutations used. In this paper we only explicitly enumerate upper bounds for the quadratic and quartic cases, leaving the cubic case to future work.

Quadratic Construction ($\mu = 2$):
When $\mu = 2$ we have a quadratic construction, and θ and γ are linear permutations. For this case, as discussed previously, we can, without loss of generalisation, set θ to the identity permutation. There are $\Gamma = (2^3 - 1)(2^3 - 2)(2^3 - 2^2) = 168$ linear permutation polynomials. By inspection, these 168 polynomials can be represented by the following 7 linear permutations which are inequivalent under input and output variable index permutation.

$$\gamma(x_q, x_r, x_s) \in \quad \{(x_q, x_r, x_s), (x_q + x_s, x_r, x_s), (x_q + x_s, x_r + x_s, x_s), (x_q + x_r + x_s, x_r, x_s), (x_q + x_r, x_r + x_s, x_s), (x_q + x_r + x_s, x_r + x_s, x_s), (x_q + x_s, x_r + x_q, x_s + x_q + x_r)\}$$

Substituting for γ and g in Construction 2, with θ fixed as the identity, gives a large set of polynomials with PAR≤ 8.0 wrt all LUUTs. We now list, for this construction, all quadratic $p(\mathbf{x})$ arising from the 7 inequivalent degree-one permutations, γ, for one 'iteration' of Construction 2, i.e. for $L = 2$, where π is fixed as the identity:

$$
\begin{aligned}
p(\mathbf{x}) &= x_0x_3 + x_1x_4 + x_2x_5 + g(\mathbf{x}) \\
p(\mathbf{x}) &= x_0x_3 + x_0x_5 + x_1x_4 + x_2x_5 + g(\mathbf{x}) \\
p(\mathbf{x}) &= x_0x_3 + x_0x_5 + x_1x_4 + x_1x_5 + x_2x_5 + g(\mathbf{x}) \\
p(\mathbf{x}) &= x_0x_3 + x_0x_4 + x_0x_5 + x_1x_4 + x_2x_5 + g(\mathbf{x}) \\
p(\mathbf{x}) &= x_0x_3 + x_0x_4 + x_1x_4 + x_1x_5 + x_2x_5 + g(\mathbf{x}) \\
p(\mathbf{x}) &= x_0x_3 + x_0x_4 + x_0x_5 + x_1x_4 + x_1x_5 + x_2x_5 + g(\mathbf{x}) \\
p(\mathbf{x}) &= x_0x_3 + x_0x_5 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_2x_5 + g(\mathbf{x})
\end{aligned}
$$

where $g(\mathbf{x}) = c_0x_0x_1 + c_1x_0x_2 + c_2x_1x_2 + c_3x_0x_1x_2 + c_4x_3x_4 + c_5x_3x_5 + c_6x_4x_5 + c_7x_3x_4x_5 + \text{RM}(1,6)$, $c_0, c_1, \ldots, c_7 \in Z_2$, with $c_3 = c_7 = 0$. An upper bound, B, to $|\mathbf{P}|$ can be computed from Theorem 2, (4), with $\mu = 2$, and the upper bound is compared to the total number of homogeneous quadratics in n binary variables in Table 2. Once again, a substantial proportion of the possible homogeneous quadratics appear to be contained in $\mathbf{P}$ for $n \leq 15$. As with $t = 2$, exact enumeration and unique generation for this set remains open, due to extra symmetries induced by π. This codeset has Hamming Distance, $D \geq 2^{n-2}$ and PAR ≤ 8.0 wrt all LUUTs. We can therefore construct a $[2^n, \log_2(|\mathbf{P}|), 2^{n-2}, 8.0]$ error-correcting code. For instance, Table 2 shows the existence of a $[64, \simeq 23.7, 16, 8.0]$ low PAR error-correcting code.

Cubic Construction ($\mu = 3$):
For $t = 3$ we can also include cubic forms in Construction 2, where θ and γ are each quadratic or linear. There are 168 linear and $5040 - 168 = 4872$ quadratic permutations for each of θ and μ and, by inspection, this set can be represented by 7 linear and 147 quadratic permutation polynomials which are inequivalent under input and output variable permutation. This makes a total of 154 inequivalent permutation polynomials for $t = 3$ [8, 31]. Substituting for θ, γ and g in Construction 2 gives a large set of polynomials with PAR≤ 8.0 wrt all LUUTs, and Hamming Distance, $D \geq 2^{n-3}$. However, we leave to further work the challenge of upper bounding, enumerating and uniquely generating this set. Here is an example from this codeset, where ijk, uv is short for $x_ix_jx_k + x_ux_v$, π is the identity, θ_j is linear and γ_j is quadratic $\forall j$. Let,

$$
\begin{aligned}
p(\mathbf{x}) = \quad & 034, 035, 045, 135, 145, 234, 235, 245, 367, 368, 378, 567, 568, 69A, 79A, 7AB, \\
& 89A, 345, 9AB, 03, 05, 14, 24, 25, 36, 38, 47, 58, 69, 6A, 6B, 7A, 7B, 89, 8B, 67, 78, AB
\end{aligned}
$$

Then $\mathbf{s}$ has PAs 4.0, 6.625, and 7.66 wrt the WHT, NHT, and DFT_1^∞, respectively. Moreover, PAR ≤ 8.0. Here is another example from this codeset, where π is the identity, θ_0 is linear, γ_0 is quadratic, θ_1 and γ_1 are both linear, and θ_2 is quadratic, γ_2 is linear. Let,

$$
\begin{aligned}
p(\mathbf{x}) = \quad & 034, 035, 045, 134, 135, 145, 234, 235, 245, 789, 67A, 68A, 67B, 68B, \\
& 03, 05, 14, 15, 36, 38, 46, 47, 56, 57, 58, 69, 79, 89, 8A, 7B
\end{aligned}
$$

Then $\mathbf{s}$ has PAs 1.0, 2.5, and 5.44 wrt the WHT, NHT, and DFT_1^∞, respectively. Moreover, PAR ≤ 8.0. Successful enumeration would allow us to construct a $[2^n, k, 2^{n-3}, 8.0]$ error-correcting code.

Quartic Construction ($\mu = 4$):
Finally, for $t = 3$, we can also include quartic forms, $p(\mathbf{x})$, which occur for the subset of cases where both θ and γ are quadratic permutations. This gives a large set of polynomials of degree ≤ 4 with PAR ≤ 8.0 wrt all LUUTs, and Hamming Distance, $D \geq 2^{n-4}$. Table 3 uses (4) to compute an upper bound on the quartic code size for $t = 3$ as n varies. We can therefore construct a $[2^n, \log_2(|\mathbf{P}|), 2^{n-4}, 8.0]$ error-correcting code. For instance, Table 3 shows the existence of a $[64, \simeq 42.9, 4, 8.0]$ error-correcting code.

Table 3: Upper Bound on Size of the Quartic Codeset Using Construction 2 for $t = 3$

n	6	9	12
$\log_2(B)$	42.92	80.91	120.29

We leave the exact enumeration and unique generation of this set to future work. Here is an example from this codeset. Let,

$$p(\mathbf{x}) = \begin{array}{l} 0235, 0245, 023, 025, 1235, 1245, 0234, 0235, 0245, 1234, 1235, 1245, \\ 123, 125, 035, 045, 134, 145, 134, 135, 145, 234, 235, 245, 03, 05, 14, 15 \end{array}$$

Then $\mathbf{s}$ has PAs 6.25, 3.25, and 3.74 wrt the WHT, NHT, and DFT_1^∞, respectively. In all cases, PAR ≤ 8.0.

4.3.5 Example 5, PAR ≤ 16.0, ($t = 4$)

Table 4 uses (4) to compute an upper bound on the sextic ($\mu = 6$) code size for $t = 4$ as n varies. We can therefore construct a $[2^n, \log_2(|\mathbf{P}|), 2^{n-6}, 16.0]$ error-correcting code. For instance, Table 4 shows the existence of a $[256, \simeq 116.6, 4, 16.0]$ error-correcting code.

Table 4: Upper Bound on Size of the Sextic Codeset Using Construction 2 for $t = 4$

n	8	12	16
$\log_2(B)$	116.63	221.08	312.00

We leave the exact enumeration and unique generation of this set to future work.

4.4 A Matrix Construction for all Quadratic Codes from Construction 2

For the case $\mu = 2$ we can, without loss of generality, fix θ to the identity permutation, and then aim to construct all possible linear permutations for γ. Each degree-one permutation, γ: $Z_2^t \to Z_2^t$ can be viewed as a $t \times t$ binary adjacency matrix under the mapping,

$M = \{m_{i,l}\} \Leftrightarrow \gamma_j(\mathbf{x_j}) = (\gamma_{0,j}(\mathbf{x_j}), \gamma_{1,j}(\mathbf{x_j}), \ldots, \gamma_{t-1,j}(\mathbf{x_j})), \qquad \gamma_{l,j} : Z_2^t \to Z_2, \deg(\gamma_{l,j}) = 1, \quad \forall l$
$m_{i,l} = 1$ if $\gamma_{l,j}(\mathbf{x_j})$ contains the linear term, x_i
$m_{i,l} = 0$ otherwise

The above mapping is an isomorphism from degree-one permutations to the General Linear Group, $\mathbf{G} = \mathrm{GL}(t, 2)$, of all binary $t \times t$ invertible matrices, mod 2 [11]. Therefore, to construct all quadratics, $p(\mathbf{x})$, for a given n and t we need to generate all degree one permutations, γ, which can, in turn, be constructed by generating all of $\mathbf{G} = \mathrm{GL}(t, 2)$, as follows [1, 2]:

Definition 11. *A binary $t \times t$* transvection *matrix, X_{ab}, satisfies,*

$$X_{ab} = \{u_{i,j}\}, \quad \begin{array}{ll} \textit{where } u_{i,j} = 1, & i = j, \textit{ and } i = a, j = b \\ u_{i,j} = 0, & \textit{otherwise} \end{array}$$

Definition 12. *The Borel subgroup of* $\mathbf{G}$ *over* Z_2 *is the set of* $t \times t$ *upper-triangular binary matrices,* $\mathbf{B}$.

Definition 13. *The Weyl subgroup of* $\mathbf{G}$ *is the set of* $t \times t$ *permutation matrices,* $\mathbf{W}$.

Arbitrarily assign a fixed ordering, O, to the $\binom{t}{2}$ matrices, X_{ab}, $a < b$. Let $w \in \mathbf{W}$ be a $t \times t$ permutation matrix where w also represents a permutation of Z_t such that $w \begin{pmatrix} a_0 \\ a_1 \\ \cdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} a_{w(0)} \\ a_{w(1)} \\ \cdots \\ a_{w(t-1)} \end{pmatrix}$ For each w, form the matrix product, X_w, comprising all X_{ab} which satisfy $a < b = w(a) > w(b)$, where the X_{ab} in X_w are ordered according to O.

Theorem 3. *[1, 2] ('Bruhat Decomposition')*

$$\mathbf{G} = \mathbf{X'_w W B} \tag{9}$$

where $\mathbf{X'_w}$ *is the set of sub-products of* X_w *that maintain the ordering of the* X_{ab} *matrices in* X_w*, including the identity matrix.*

All linear permutations, γ, can be uniquely constructed using Theorem 3, where $|\mathbf{G}| = \Gamma = \prod_{i=0}^{t-1}(2^t - 2^i)$. This means that we can generate all quadratics, $p(\mathbf{x})$, for Construction 2 for any t and L. However, as indicated previously, the $p(\mathbf{x})$ are not guaranteed to be unique due to the extra symmetries induced by π. We leave to further work the challenge of modifying the Bruhat decomposition to eliminate these residual symmetries.

4.5 Examples of Bruhat Decomposition

$t = 2$**:**
For $t = 2$, $X_{01} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, $\mathbf{B} = \{\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)\}$, $\mathbf{W} = \{\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)\}$. Assign the trivial ordering X_{01} to the one matrix, X_{ab}. Now $w = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ defines the identity permutation $(0)(1)$ and makes $X_w = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. Moreover $w = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ defines the permutation $(0, 1)$ and makes $X_w = X_{01}$. Therefore, when w defines $(0)(1)$ we generate 2 matrices of $\mathbf{G}$, and when w defines $(0, 1)$ we generate 4 matrices of $\mathbf{G}$, bringing the total to 6, which is correct.
$t = 3$**:**
For $t = 3$, $X_{01} = \left(\begin{smallmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$, $X_{02} = \left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$, $X_{12} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{smallmatrix}\right)$, $|\mathbf{B}| = 8$, $|\mathbf{W}| = 6$. We can arbitrarily choose to assign the ordering $X_{01}X_{02}X_{12}$ to the 3 matrices, X_{ab}. The partitioning of matrices in $\mathbf{G}$ is then as follows:

w	X_w	$\lvert$subset of $\mathbf{G}\rvert$
$(0)(1)(2)$	I	8
$(0)(1, 2)$	X_{12}	16
$(0, 1)(2)$	X_{01}	16
$(0, 2)(1)$	$X_{01}X_{02}X_{12}$	64
$(0, 2, 1)$	$X_{01}X_{02}$	32
$(0, 1, 2)$	$X_{02}X_{12}$	32
		Total $= \lvert\mathbf{G}\rvert = 168$

5 A Further Generalisation

Lemma 20 of [25] extends the Maiorana-McFarland construction to a large codeset with near-Bent properties, where a 1-1 map is replaced by a 2^{δ}-1 map. In this section we apply similar ideas to Construction 2 to obtain Construction 3 below, (proofs omitted). Construction 3 is quite complicated and so far we have not found a better way to express the construction. We advise readers to skip this section on first reading. We do, however, provide some examples in the appendix which will help to clarify the construction.

Construction 3 tackles the case when the number of variables in each of the L iterations is allowed to vary. Using the terminology of Construction 1, this implies more than one $\mathbf{E}$ matrix for some iterations, where each $\mathbf{E}$ matrix is unitary and is associated with an independently chosen row/column permutation. Before describing the construction we must first specify some new terminology.

Let permutation $\theta : Z_2^t \rightarrow Z_2^t$ have as domain the t binary variables, $\mathbf{x}$. Let $f : Z_2^u \rightarrow Z_2$ have as domain the set of u binary variables, $\mathbf{z}$. Let us now assume that the form of θ depends on the output of $f(\mathbf{z})$. We write this as $\theta(\mathbf{x})\{f(\mathbf{z})\}$ and this expression can be partly evaluated as,

$$\theta(\mathbf{x})\{f(\mathbf{z})\} = (f(\mathbf{z}) + 1)\theta^0(\mathbf{x}) + f(\mathbf{z})\theta^1(\mathbf{x})$$

where we must define 2 permutations, θ^0 and θ^1, from $Z_2^t \rightarrow Z_2^t$. For brevity we can write this as $\theta\{f\}$. We can generalise this definition to make θ dependent on v associated functions, f_i, from $Z_2^{u_i} \rightarrow Z_2$, $0 \leq i < v$. We write this as $\theta(\mathbf{x})\{f_0(\mathbf{z_0}), f_1(\mathbf{z_1}), \ldots, f_{v-1}(\mathbf{z_{v-1}})\}$, and we must now define 2^v permutations, $\theta^0, \theta^1, \ldots, \theta^{2^v-1}$, from $Z_2^t \rightarrow Z_2^t$, one of which is 'selected' according to the combined outputs of the f_i. For brevity we can write this as $\theta\{f_0, f_1, \ldots, f_{v-1}\}$. We can further abbreviate the notation by labeling $\{F\} = \{f_0, f_1, \ldots, f_{v-1}\}$. We can then *NEST* dependencies $F_0, F_1, F_2, \ldots$. This is written as $\theta = \theta\{F_0\{F_1\{F_2\{\ldots\}\}\}\}$, and means that the form of the functions in F_{i-1} depend on the outputs of the functions F_i. We express the *NEST* operation as,

$$NEST(\theta\{F\}, \{F'\}) \rightarrow \theta\{F\{F'\}\}$$

Let $|F|$ mean the number of functions labeled by F. Let $v = \sum_{i=0}^{Q-1} |F_i|$. Then, if we *NEST* to a depth of Q using the function sets, F_i, $0 \leq i < Q$, then we must define 2^v permutations, $\theta^0, \theta^1, \ldots, \theta^{2^v-1}$, from $Z_2^t \rightarrow Z_2^t$, one of which is 'selected' according to the combined outputs of the F_i. As an example, let $F_0 = \{f_0(\mathbf{z_0}), f_1(\mathbf{z_1}))\}$, and $F_1 = \{f_2(\mathbf{z_2})\}$. Then, with f_0, f_1, f_2 outputting $\rightarrow Z_2$,

$$\theta(\mathbf{x})\{F_0\{F_1\}\} = \theta(\mathbf{x})\{f_0(\mathbf{z_0}), f_1(\mathbf{z_1})\{f_2(\mathbf{z_2})\}\}$$

which, for brevity, can be written as,

$$\theta\{F_0, F_1\} = \theta\{f_0, f_1\{f_2\}\}$$

and can be partially evaluated as,

$$\begin{aligned}&(f_2(\mathbf{z_2}) + 1)((f_1(\mathbf{z_1}) + 1)(f_0(\mathbf{z_0}) + 1)\theta^0(\mathbf{x}) + (f_1(\mathbf{z_1}) + 1)f_0(\mathbf{z_0})\theta^1(\mathbf{x}) + f_1(\mathbf{z_1})(f_0(\mathbf{z_0}) + 1)\theta^2(\mathbf{x})\\ &+f_1(\mathbf{z_1})f_0(\mathbf{z_0})\theta^3(\mathbf{x})) + f_2(\mathbf{z_2})((f_1'(\mathbf{z_1}) + 1)(f_0'(\mathbf{z_0}) + 1)\theta^4(\mathbf{x}) + (f_1'(\mathbf{z_1}) + 1)f_0'(\mathbf{z_0})\theta^5(\mathbf{x})\\ &+f_1'(\mathbf{z_1})(f_0'(\mathbf{z_0}) + 1)\theta^6(\mathbf{x}) + f_1'(\mathbf{z_1})f_0'(\mathbf{z_0})\theta^7(\mathbf{x}))\end{aligned}$$

where f_i' is not necessarily the same as f_i, and where 8 permutations, $\theta^i : Z_2^t \rightarrow Z_2^t$, $0 \leq i < 8$, must be defined with domain $\mathbf{x}$.

We will also decompose the permutation $\theta_j : Z_2^t \rightarrow Z_2^t$ as $\theta_j = (\theta_{0,j}, \theta_{1,j}, \ldots, \theta_{t-1,j})$, where $\theta_{i,j} : Z_2^t \rightarrow Z_2$. Similarly, $\gamma_j : Z_2^t \rightarrow Z_2^t$ is decomposed as $\gamma_j = (\gamma_{0,j}, \gamma_{1,j}, \ldots, \gamma_{t-1,j})$, where $\gamma_{i,j} : Z_2^t \rightarrow Z_2$.

We now define the *EXTEND* operation. Let F be a length $t' - t$ vector of functions of arbitrary domain each of which outputs $\rightarrow Z_2$ (where it is assumed that $t' \geq t$). Then,

$$EXTEND(\theta_j, F) \rightarrow (\theta_{j,0}, \theta_{j,1}, \ldots, \theta_{j,t-1}, F)$$

is a mapping $\rightarrow Z_2^{t'}$. In other words, θ_j has been extended by means of the vector F from a permutation of Z_2^t to a mapping which outputs to $Z_2^{t'}$. Construction 3 uses combinations of *NEST* and *EXTEND* to construct θ'_j and γ'_j, which output (after *NESTING* and *EXTENSION*) to $Z_2^{t\max}$, where $t_{\max}$ is defined below. θ'_j and γ'_j can then be 'multiplied', in the same way as $\theta_j \gamma_j$ in (3), and the resulting expressions added to form the final polynomial, p.

We are now ready to describe Construction 3.

Construction 3: *To construct a function of n boolean variables with PAR $\leq 2^{-t_{max}}$ wrt all LUUTs, we pursue the following strategy (the y_i are auxilliary boolean variables which can be used at the end to select between different sequences):*

- *Choose t_{max} so that $1 \leq t_{max} \leq n$.*
- *Partition the n binary variable indices, $\{0, 1, \ldots, n-1\}$, into L disjoint variable subsets, $\mathbf{S_j}$, such that $t_j = |\mathbf{S_j}| \leq t_{max}$, $\forall j$, $0 \leq j < L$.*
- *For each j, $0 \leq j < L-1$, define θ_j comprising $2^{t_{max} - t_j}$ permutations, $\theta_j^0, \theta_j^1, \ldots, \theta_j^{2^{t_{max}-t_j}-1}$, from $Z_2^{t_j} \rightarrow Z_2^{t_j}$ with domain the set of t_j binary variables $\mathbf{x_j} = \{x_i\}$, $i \in \mathbf{S_j}$. Similarly, for each j, $0 \leq j < L-1$, define γ_j comprising $2^{t_{max} - t_{j+1}}$ permutations, $\gamma_j^0, \gamma_j^1, \ldots, \gamma_j^{2^{t_{max}-t_{j+1}}-1}$, from $Z_2^{t_{j+1}} \rightarrow Z_2^{t_{j+1}}$ with domain the set of t_{j+1} binary variables $\mathbf{x_{j+1}} = \{x_i\}$, $i \in \mathbf{S_{j+1}}$.*
- *For $j = 0$, $j < L-1$, j++ do:*
  {
  . $t = t_j$.
  . *Assign F as the zero vector of length $t_{max} - t_j$.*
  . *For $i = j+1$, $i \leq L-1$, i++ do:*
  . {
  . *if* $t < t_i$
  . {
  . *assign* $\theta_j = NEST(\theta_j, \{\gamma_{i-1,t}, \gamma_{i-1,t+1}, \ldots, \gamma_{i-1,t_i-1}\})$.
  . *set* $t = t_i$.
  . }
  . }
  . *if* $t < t_{max}$
  . *assign* $\theta_j = NEST(\theta_j, \{y_t, y_{t+1}, \ldots, y_{t_{max}-1}\})$.
  . $\theta'_j = EXTEND(\theta_j, F)$.
  . $t = t_{j+1}$.
  . $F = ()$.
  . *For $i = j+1$, $i < L-1$, i++ do:*
  . {
  . *if* $t < t_{i+1}$
  . {
  . *assign* $F = \{\gamma_{i,t}, \gamma_{i,t+1}, \ldots, \gamma_{i,t_{i+1}-1}\}$.
  . *assign* $\gamma_j = NEST(\gamma_j, F)$.
  . *assign* $\gamma_j = EXTEND(\gamma_j, F)$.
  . *set* $t = t_{i+1}$.

PAR ≤ 8.0

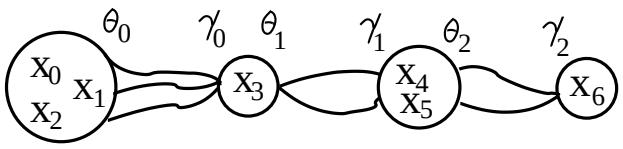


Figure 3: Example of Construction 3 where $t_{\max} = 4$

. }
. }
. *if* $t < t_{max}$
. {
. *assign* $F = \{y_t, y_{t+1}, \ldots, y_{t_{max}-1}\}$.
. *assign* $\gamma_j = NEST(\gamma_j, F)$.
. *assign* $\gamma_j = EXTEND(\gamma_j, F)$.
. }
. $\gamma'_j = \gamma_j$.
}

- *Then* $\mathbf{s} = 2^{\frac{-n}{2}}(-1)^{p(\mathbf{x})}$, *where* p *is given by,*

$$p(\mathbf{x}) = \sum_{j=0}^{L-2} \theta'_j \gamma'_j + \sum_{j=0}^{L-1} g_j(\mathbf{x_j}) \tag{10}$$

where $2^{t_{max}-t_{L-1}}$ *different sequences are generated according to the assignments given to the* $t_{max} - t_{L-1}$ *auxilliary variables,* y_i*, which are present in the* θ'_j *or* γ'_j*, and where the* g_j *are arbitrary functions of* $\mathbf{x_j}$*, outputting* $\rightarrow Z_2$*. (Note that, for this generalisation, the permutation,* π*, of the indices* $\{0, 1, \ldots, n-1\}$ *is implicitly included in the initial index partition operation).*

Corollary 3. *The length* $N = 2^n$ *sequences,* $\mathbf{s}$*, of Construction 3, satisfy PAR*$(\mathbf{s}) \leq 2^{t_{\max}}$ *wrt all* $N \times N$ *LUUTs.*

Fig 3 illustrates Construction 3 for the case of Example 1 in the Appendix, where we are also free to permute indices, i, of x_i.

Corollary 4. *Each of the* $2^{t_{max}-t_{L-1}}$ *sequences,* $\mathbf{s}$*, of Construction 3 is a coset leader for a coset of* $2^{t_{L-1}}$ *sequences formed from any linear offset of* $\mathbf{s}$ *by linear combinations of members of* $\mathbf{x_{L-1}}$*. The union of these* $2^{t_{max}-t_{L-1}}$ *cosets forms a CS set of* $2^{t_{max}}$ *sequences of length* 2^n*.*

The Appendix provides examples for Construction 3.

In Construction 3, if $t_j = t_{\max}, \forall j$, then there is no *NESTING* or *EXTENSION* and the construction simplifies to Construction 2. It remains open to exactly enumerate and uniquely generate the sequences in Construction 3. Note that, just as Construction 2 is a special case of Construction 1, so Construction 3 is a special case of a more general construction where the $\mathbf{E}$ matrices are not necessarily WHT matrices. This further generalisation is conceptually straightforward once Construction 3 is understood. Note also that Construction 3 allows us to add yet more sequences to our low PAR codesets without degrading distance, and these improvements in code rate will be discussed in future papers.

6 Discussion and Open Problems

This paper presented a construction for low PAR error-correcting codes which significantly generalises the fundamental codeset of Davis and Jedwab, and concisely summarises the complementary set constructions of Golay, Turyn, and Tseng and Liu. An important subcase, Construction 2, can be viewed either as recursion or specialisation of a two-sided Maiorana-McFarland construction. The paper highlights the central importance for PAR constructions of generating permutation polynomials of prescribed maximum degree, and provides motivation for further research work in this area, and also motivates the search for solutions to a number of open problems which we will now discuss.

Open Problems:

- The constructions of this paper only provide a unique, implementable encoder if we can provide algorithms to generate all permutations and/or many-to-one/one-to-many mappings of specified maximum algebraic degree. Symmetric permutations are straightforward. Section 4.4 provides a (previously-known) generation scheme for linear permutations (producing 'quadratic' sequences). But the problem of unique generation of permutations of degree greater than one is, as far as the authors know, unsolved. Solutions to this problem would have far-reaching application in cryptography, and this paper shows that such algorithms are central to the development of constructions for low PAR error-correcting codes.

- Given an algorithm to generate all permutation polynomials, then Construction 2 only generates distinct $p(\mathbf{x})$ for $t = 1$. For $t > 1$, π, the permutation of variable indices induces extra symmetries causing a few $p(\mathbf{x})$ to be generated more than once. In other words, for $t > 1$ it is possible that the action of two (or more) distinct permutations, π and π', may result in the same polynomial, $p(\mathbf{x})$. This situation is reflected in (4), which is a strict upper bound for $t > 1$. It remains open to provide an algorithm to generate all distinct $p(\mathbf{x})$. Such an algorithm would replace (4) with an exact expression and provide a 'black-box' encoding solution for OFDM systems. The problem is closest to solution for the case of linear permutations, where Section 4.4 solves the permutation generation part, and it remains to eliminate the coding collisions caused by distinct permutations π. We have not yet tackled the problem of unique generation of codewords for Construction 3, but this is clearly an even harder task.

- It would also be interesting to choose the $\mathbf{E_j}$ other than WHTs for Constructions 1 and 3. In particular, note that the case of $t = 1, 2, 3$ refers to Hadamard matrices of size $2, 4, 8$, respectively (PAR $\leq 2, 4, 8$, respectively). It is known that, for $t \leq 3$, all Hadamard matrices are row/column permutation equivalent to WHT matrices, so Construction 2 covers all cases. However, for $t = 4$, (PAR ≤ 16) we know that there are 5 row/column permutation inequivalent 16×16 Hadamard matrices, one of which is the WHT [32]. Therefore, for $t = 4$, there are essentially 5 different versions of Construction 1, one of which is Construction 2. As t increases we have yet more inequivalent classes of Hadamard matrices. This paper therefore establishes a direct link between the classification of Hadamard matrices, and the classification of PAR classes, and provides a strong motivation to discover manageable ANF descriptions for each of these classes.

- One important way to improve code rate whilst keeping PAR low is to choose rectangular $\mathbf{E_j}$, with more rows than columns, where the rows form a set of near-orthogonal sequences. Application of Construction 1 would then result in a slowly rising PAR bound as L increases, but the rate of the code would also improve compared to the

cases where $\mathbf{E_j}$ is a square matrix. This raises the possibility of even higher rate low PAR error-correcting codes. For instance, in CDMA, the WHT rows can be used as a sequence set, due to their orthogonality. But larger near-orthogonal sequence sets are highly desirable, and the set of Gold sequences is such a set. The set of Kerdock sequences is an even larger set [9]. One could therefore use one of these larger sequence sets to form our $\mathbf{E}$ matrices, one sequence per row. Our row permutation, γ, would then operate over a larger space, resulting in an improved code rate. And the near-orthogonality of the sequence set would ensure the upper-bound on PAR only rose slowly after each iteration of the construction, although computing the precise upper-bound in such cases remains an open challenge.

- In this paper we have proposed the study of PAR wrt all LUUTs. One can completely generalise the set of LUUTs to the set of *Linear Unitary Transforms* (LUTs) by including unitary matrices which are the tensor product of $r \times r$ unitary matrices such that each matrix entry is no longer constrained to have a magnitude of $\frac{1}{\sqrt{r}}$. For instance, linear unitary matrices which have $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\frac{1}{2}\begin{pmatrix} \sqrt{3} & 1 \\ 1 & -\sqrt{3} \end{pmatrix}$ as tensor factors are in the set of LUTs but not the smaller subset represented by LUUTs. It is of interest to study the PAR of sequences wrt all LUTs. This study has been initiated in [18, 19] where it was shown that the length 2^n sequences which represent indicator functions for linear error-correcting codes of blocklength n have PAR wrt all LUTs lower bounded by $2^{\frac{n}{2}}$. Moreover, it is proved in [18] that, for indicator functions which represent linear error-correcting codes (functions outputting to 0 or 1), the worst-case spectral peak wrt all LUTs, (and hence the peak which defines the PAR wrt all LUTs), occurs in one or more of the spectra generated by action of the set of transforms formed from all possible tensor products of the matrices $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. The nice thing about this result is that we don't have to search the complete infinite space of LUTs to find the worst-case spectral peak. However, little more is known about the PAR wrt all LUTs for more general functions. The study has direct relevance to Quantum Entanglement and it has recently been shown that the spectral index of the worst-case spectral peak wrt all LUTs identifies a generalised linear weakness for classical cryptosystems [27], where a large PAR means a large linear bias.

- One celebrated area of study is the unresolved quest to find flat polynomials on the unit circle [12]. This translates, in the terminology of this paper, into the search for a sequence construction of length 2^n (restricted, say, to the alphabet $\{1, -1\}$), such that the sequence has PAR wrt DFT_1^∞ of $1.0 + \epsilon_0$ and a lowest spectral power trough of $1.0 - \epsilon_1$ such that the ϵ terms vanish as length, 2^n, increases. No construction with these properties is known for the bipolar case. We can pose a more general problem. Do flat polynomials exist wrt all LUUTs (not just DFT_1^∞)? And an even more general problem would be: Do flat polynomials exist wrt all LUTs? More realistically, how well can we do for these transform sets?

7 Acknowledgements

We would like to thank Kenneth G. Paterson for giving helpful and encouraging advice regarding this paper.

8 Appendix

We provide some examples for Construction 3.

8.1 Example 1

Let $n = 7$. Consider the partition, $\mathbf{S_0} = \{0, 1, 2\}$, $\mathbf{S_1} = \{3\}$, $\mathbf{S_2} = \{4, 5\}$, $\mathbf{S_3} = \{6\}$, as shown in Fig 3. Then $t_0 = 3$, $t_1 = 1$, $t_2 = 2$, $t_3 = 1$, $t_{\text{max}} = t_0 = 3$, and $L = 4$.

Applying Construction 3, we must initially define the following permutations:

θ_0^0	with domain (x_0, x_1, x_2)
$\gamma_0^0, \gamma_0^1, \gamma_0^2, \gamma_0^3,$ and $\theta_1^0, \theta_1^1, \theta_1^2, \theta_1^3$	with domain (x_3)
$\gamma_1^0, \gamma_1^1,$ and θ_2^0, θ_2^1	with domain (x_4, x_5)
$\gamma_2^0, \gamma_2^1, \gamma_2^2, \gamma_2^3$	with domain (x_6)

It then follows, from Construction 3, that,

$$\begin{array}{ll} \theta_0' \leftarrow \theta_0(x_0, x_1, x_2) & \gamma_0' \leftarrow (\gamma_0(x_3)\{\gamma_{1,1}\{y_2\}\}, \gamma_{1,1}\{y_2\}, y_2) \\ \theta_1' \leftarrow (\theta_1(x_3)\{\gamma_{1,1}\{y_2\}\}, 0, 0) & \gamma_1' \leftarrow (\gamma_1(x_4, x_5)\{y_2\}, y_2) \\ \theta_2' \leftarrow (\theta_2(x_4, x_5)\{y_2\}, 0) & \gamma_2' \leftarrow (\gamma_2(x_6)\{y_1, y_2\}, y_1, y_2) \end{array}$$

Let us now assign, as examples, specific (arbitrary) permutation polynomials to each of the θ_j and γ_j. Let,

$$\begin{array}{ll} \theta_0 = (x_0, x_1, x_2) & \gamma_0^0 = (x_3), \gamma_0^1 = (x_3), \gamma_0^2 = (x_3), \gamma_0^3 = (x_3 + 1) \\ \theta_1^0 = (x_3 + 1), \theta_1^1 = (x_3), \theta_1^2 = (x_3), \theta_1^3 = (x_3) & \gamma_1^0 = (x_4, x_5), \gamma_1^1 = (x_4 + x_5, x_5) \\ \theta_2^0 = (x_4 + x_5, x_5), \theta_2^1 = (x_4, x_5) & \gamma_2^0 = (x_6), \gamma_2^1 = (x_6), \gamma_2^2 = (x_6), \gamma_2^3 = (x_6 + 1) \end{array} \tag{11}$$

Given these permutation assignments we can evaluate:

$$\begin{array}{l} \gamma_0(x_3)\{\gamma_{1,1}\{y_2\}\} = (y_2 + 1)((x_5 + 1)x_3 + x_5x_3) + y_2((x_5 + 1)x_3 + x_5(x_3 + 1)) = x_3 + x_5y_2 \\ \theta_1(x_3)\{\gamma_{1,1}\{y_2\}\} = (y_2 + 1)((x_5 + 1)(x_3 + 1) + x_5x_3) + y_2((x_5 + 1)x_3 + x_5x_3) = x_3 + x_5 + 1 + (x_5 + 1)y_2 \\ \gamma_1(x_4, x_5)\{y_2\} = (y_2 + 1)(x_4, x_5) + y_2(x_4 + x_5, x_5) = (x_4 + x_5y_2, x_5) \\ \theta_2(x_4, x_5)\{y_2\} = (y_2 + 1)(x_4 + x_5, x_5) + y_2(x_4, x_5) = (x_4 + x_5 + x_5y_2, x_5) \\ \gamma_2(x_6)\{y_1, y_2\} = (y_1 + 1)(y_2 + 1)x_6 + y_1(y_2 + 1)x_6 + (y_1 + 1)y_2x_6 + y_1y_2(x_6 + 1) = x_6 + y_1y_2 \end{array}$$

Therefore,

$$\begin{aligned} \theta_0'\gamma_0' &= x_0x_3 + x_1x_5 + y_2(x_0x_5 + x_2) \\ \theta_1'\gamma_1' &= x_3x_4 + x_4x_5 + x_4 + y_2(x_0x_5 + x_2) \\ \theta_2'\gamma_2' &= x_4x_6 + x_5x_6 + y_1x_5 + y_2x_5x_6 + y_1y_2x_4 \end{aligned}$$

Therefore,

$$\sum_{j=0}^{2} \theta_j'\gamma_j' = x_0x_3 + x_1x_5 + x_3x_4 + x_4x_5 + x_4x_6 + x_5x_6 + x_4 + y_1x_5 + y_2(x_0x_5 + x_3x_5 + x_4x_5 + x_5x_6 + x_2 + x_4) + y_1y_2x_4$$

Let us arbitrarily first consider that all g functions in (10) are zero (for ease of exposition). Then, $p = \sum_{j=0}^{2} \theta_j'\gamma_j'$. Moreover we have 4 different choices of sequence, $\mathbf{s}$, depending on the values of y_1 and y_2. Table 5 shows the PARs wrt WHT, NHT, and DFT_1^∞, for each of these 4 sequences.

In all cases the PAR is upper-bounded by $2^{t_{\text{max}}} = 8.0$, as predicted by Corollary 3. Note that, as stated by Corollary 4, the final optional addition of '$+x_6$' onto each of the 4 sequences in Table 5 produces a CS set of 8 sequences (of length 128) wrt all LUUTs.

Table 5: PAs of Example 1 wrt WHT, NHT, and DFT_1^∞

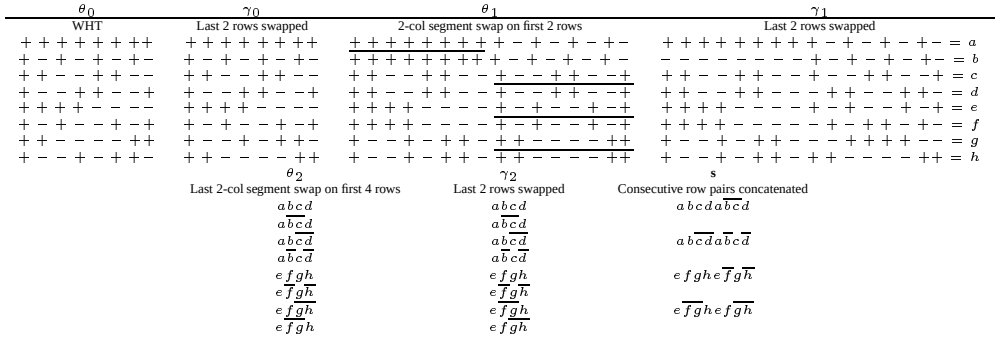
$y_1 y_2$	p	PA: WHT	NHT	DFT_1^∞
00	$x_0x_3 + x_1x_5 + x_3x_4 + x_4x_5 + x_4x_6 + x_5x_6 + x_4$	2.0	1.0	4.18
10	$x_0x_3 + x_1x_5 + x_3x_4 + x_4x_5 + x_4x_6 + x_5x_6 + x_4 + x_5$	2.0	1.0	4.25
01	$x_0x_3 + x_0x_5 + x_1x_5 + x_3x_4 + x_3x_5 + x_4x_6 + x_2$	2.0	1.0	5.79
11	$x_0x_3 + x_0x_5 + x_1x_5 + x_3x_4 + x_3x_5 + x_4x_6 + x_2 + x_4 + x_5$	2.0	1.0	6.02

It is helpful to alternatively construct these sequences visually, by using a generalised version of the strategy outlined in Section 3, which is also the foundation for Construction 1. Although we have not formally proved Construction 3 in this paper, the following construction technique essentially provides the proof for Construction 3. We use unitary WHT matrices, $\mathbf{E_j^k}$, $0 \le k < 2^{t\mathrm{max}-t_j}$. Specifically, for Example 1, we have one 8×8 matrix, $\mathbf{E_0}$, four 2×2 matrices, $\mathbf{E_1^0}, \mathbf{E_1^1}, \mathbf{E_1^2}, \mathbf{E_1^3}$, and two 4×4 matrices, $\mathbf{E_2^0}, \mathbf{E_2^1}$. The rows and columns of $\mathbf{E_j^k}$ are permuted by γ_{j-1}^k and θ_j^k, respectively. Specifically,

θ_0 permutes columns of $\mathbf{E_0}$,
θ_1^k permutes columns of $\mathbf{E_1^k}$, $0 \le k < 4$,
θ_2^k permutes columns of $\mathbf{E_2^k}$, $0 \le k < 2$,

γ_0^r permutes consecutive row pairs of $\mathbf{E_0}$, $0 \le r < 4$
γ_1^r permutes consecutive sets of four rows of column-concatenated $\mathbf{E_1^k}$, $0 \le r < 2$
γ_2^r permutes consecutive row pairs of column-concatenated $\mathbf{E_2^k}$, $0 \le r < 4$

Let us choose the permutations for $\theta$and γ as shown in (11) of Example 1. Then these permutations act in conjunction with the $\mathbf{E}$ matrices as follows (where '$\overline{a}$' means multiply a by -1). Note that, after each γ permutation, the appropriate rows are concatenated before point-multiplying by elements of the appropriate E matrix:



It is straightforward to check that the above 4 sequences, $\mathbf{s}$, correspond exactly to the 4 sequences, $\mathbf{s}$, in Table 5, as represented by p. This example also illustrates that if the $\mathbf{E_j^k}$ are chosen to be row/column inequivalent to WHT matrices, then we can further generalise Construction 3.

Finally, for Example 1, let us now make the g functions non-zero. Arbitrarily, let $g_0(x_0, x_1, x_2) = x_0x_1x_2 + x_2$, $g_1(x_3) = x_3$, $g_2(x_4, x_5) = x_4x_5 + x_5$, and $g_3(x_6) = 0$. Table 6 shows the PAs after addition of $g_0 + g_1 + g_2 + g_3$ onto each of the four sequences of Table 5.

Once again, in all cases the PAR is upper-bounded by $2^{t\mathrm{max}} = 8.0$, as predictedby Corollary 3. Note that, as stated by Corollary 4, the final optional addition of '$+x_6$' onto each of the 4 sequences in Table 6 forms a CS set of 8 sequences wrt all LUUTs.

Table 6: PAs of Example 1 wrt WHT, NHT, and DFT_1^∞ after Addition of $g_0 + g_1 + g_2 + g_3$

$y_1 y_2$	p	PA: WHT	NHT	DFT_1^∞
00	$x_0x_1x_2 + x_0x_3 + x_1x_5 + x_3x_4 + x_4x_6 + x_5x_6 + x_4 + x_2 + x_3 + x_5$	4.5	2.5	4.04
10	$x_0x_1x_2 + x_0x_3 + x_1x_5 + x_3x_4 + x_4x_6 + x_5x_6 + x_4 + x_2 + x_3$	4.5	2.5	4.83
01	$x_0x_1x_2 + x_0x_3 + x_0x_5 + x_1x_5 + x_3x_4 + x_3x_5 + x_4x_5 + x_4x_6 + x_3 + x_5$	4.5	2.0	3.59
11	$x_0x_1x_2 + x_0x_3 + x_0x_5 + x_1x_5 + x_3x_4 + x_3x_5 + x_4x_5 + x_4x_6 + x_4 + x_3$	4.5	2.0	3.51

$$\mathrm{PAR} \leq 8.0$$

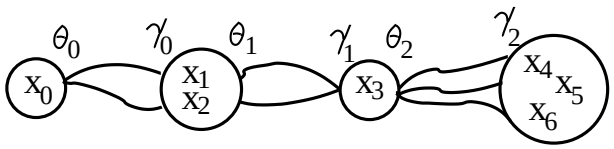


Figure 4: Example of Construction 3 where $t_{\max} = 4$ (Reverse of Figure 3)

8.2 Example 2

Except for the special case of Construction 2, Construction 3 does not give the same set of sequences when starting from the rightmost variable set (as shown), instead of the left-most variable set. Example 2 emphasises this point by describing the construction for the partition of Figure 4, which is clearly the reverse of Figure 3.

The partition is, $\mathbf{S_0} = \{0\}$, $\mathbf{S_1} = \{1, 2\}$, $\mathbf{S_2} = \{3\}$, $\mathbf{S_3} = \{4, 5, 6\}$, as shown in Fig 4. Then $t_0 = 1$, $t_1 = 2$, $t_2 = 1$, $t_3 = 3$, $t_{\max} = t_3 = 3$, and $L = 4$.

Applying Construction 3, we must initially define the following permutations:

$\theta_0^0, \theta_0^1, \theta_0^2, \theta_0^3$	with domain (x_0)
γ_0^0, γ_0^1 and θ_1^0, θ_1^1	with domain (x_1, x_2)
$\gamma_1^0, \gamma_1^1, \gamma_1^2, \gamma_1^3,$ and $\theta_2^0, \theta_2^1, \theta_2^2, \theta_2^3$	with domain (x_3)
γ_2^0	with domain (x_4, x_5, x_6)

It then follows, from Construction 3, that,

$$\begin{array}{ll} \theta_0' \leftarrow (\theta_0(x_0)\{\gamma_{0,1}\{\gamma_{2,2}\}\}, 0, 0) & \gamma_0' \leftarrow (\gamma_0(x_1, x_2)\{\gamma_{2,2}\}, \gamma_{2,2}) \\ \theta_1' \leftarrow (\theta_1(x_1, x_2)\{\gamma_{2,2}\}, 0) & \gamma_1' \leftarrow (\gamma_1(x_3)\{\gamma_{2,1}, \gamma_{2,2}\}, \gamma_{2,1}, \gamma_{2,2}) \\ \theta_2' \leftarrow (\theta_2(x_3)\{\gamma_{2,1}, \gamma_{2,2}\}, 0, 0) & \gamma_2' \leftarrow \gamma_2(x_4, x_5, x_6) \end{array}$$

Let us now assign the same permutations as Example 1, but in reverse, to each of the θ_j and γ_j. Let,

$$\begin{array}{ll} \theta_0^0 = (x_0), \theta_0^1 = (x_0), \theta_0^2 = (x_0), \theta_0^3 = (x_0 + 1) & \gamma_0^0 = (x_1 + x_2, x_2), \gamma_0^1 = (x_1, x_2) \\ \theta_1^0 = (x_1, x_2), \theta_1^1 = (x_1 + x_2, x_2) & \gamma_1^0 = (x_3 + 1), \gamma_1^1 = (x_3), \gamma_1^2 = (x_3), \gamma_1^3 = (x_3) \\ \theta_2^0 = (x_3), \theta_2^1 = (x_3), \theta_2^2 = (x_3), \theta_2^3 = (x_3 + 1) & \gamma_2 = (x_4, x_5, x_6) \end{array}$$

Given these permutation assignments we can evaluate:

$$\begin{array}{l} \theta_0(x_0)\{\gamma_{0,1}\{\gamma_{2,2}\}\} = x_2x_6 + x_0 \\ \gamma_0(x_1, x_2)\{\gamma_{2,2}\} = (x_2x_6 + x_1 + x_2, x_2) \\ \theta_1(x_1, x_2)\{\gamma_{2,2}\}, 0) = (x_2x_6 + x_1, x_2) \\ \gamma_1(x_3)\{\gamma_{2,1}, \gamma_{2,2}\} = x_5x_6 + x_3 + x_5 + x_6 + 1 \\ \theta_2(x_3)\{\gamma_{2,1}, \gamma_{2,2}\} = x_5x_6 + x_3 \end{array}$$

Therefore,

$$\begin{aligned}\theta'_0\gamma'_0 &= x_0x_2x_6 + x_1x_2x_6 + x_0x_1 + x_0x_2\\ \theta'_1\gamma'_1 &= x_1x_5x_6 + x_2x_3x_6 + x_1x_3 + x_1x_5 + x_1x_6 + x_2x_5 + x_1\\ \theta'_2\gamma'_2 &= x_4x_5x_6 + x_3x_4\end{aligned}$$

Therefore,

$$\sum_{j=0}^{2}\theta'_j\gamma'_j = x_0x_2x_6+x_1x_2x_6+x_1x_5x_6+x_2x_3x_6+x_4x_5x_6+x_0x_1+x_0x_2+x_1x_3+x_1x_5+x_1x_6+x_2x_5+x_3x_4+x_1 \quad (12)$$

Let us, arbitrarily, consider that all g functions in (10) are zero (for ease of exposition). Then, $p = \sum_{j=0}^{2}\theta'_j\gamma'_j$. Unlike Example 3, we now only have 1 choice of sequence, $\mathbf{s}$. This sequence has a PA of 8.0, 2.5, and 4.93 wrt the WHT, NHT, and DFT$_1^\infty$, respectively. In all cases the PAR is upper-bounded by $2\,^t\max = 8.0$, as predicted by Corollary 3. Note that, as stated by Corollary 4, a CS set of 8 sequences (of length 128) wrt all LUUTs is formed by $\mathbf{s}$ and all linear offsets of $\mathbf{s}$ over the variables $\{x_4, x_5, x_6\}$.

We can, alternatively, construct this sequence using a generalised version of the strategy outlined in Section 3. We obtain the following construction steps:

θ_0	γ_0	θ_1	γ_1	θ_2	
Cols swapped on last 2 rows	Second pair of rows swapped	Last 2 col segments swapped on last 4 rows	First 2 rows swapped	Col segments swapped on last 2 rows	
++	++	+ + + − + − ++	+ + − + + − − −	+ + − + + − − − + + + − + − ++	= a
+−	+−	+ + + − + − ++	+ + + − + − ++	+ + − + + − − − + + + − + − ++	= b
++	+−	+ + + − + − ++	+ + + − − + − −	+ + + − − + − − + + − + − + ++	= c
+−	++	+ + + − + − ++	+ + − + − + ++	+ + + − − + − − + + − + − + ++	= d
++	++	+ + + − + + − +	+ + + − + + − +	+ + + − + + − + + + − + − − − +	= e
+−	+−	+ + + − ++ − +	+ + − + − − − +	+ + + − + + − + + + − + − − − +	= f
++	++	+ + + − + + − +	+ + + − − − + −	+ + + − − − + − + + − + + + + −	= g
−+	−+	+ + + − + + − +	+ + − + + + + −	+ + + − − − + − + + − + + + + −	= h

Finally, γ_2 generates $\mathbf{s} = abcdefg$

It is straightforward to check that the above sequence, $\mathbf{s}$, corresponds exactly to the $\mathbf{s}$, as represented by p in (12).

References

[1] Alperin, J.L.,Bell, R.B.: **Groups and Representations,** Graduate Texts in Mathematics, Springer, **162**, pp. 39–48, (1995)

[2] Brundan, J.: Web Lecture Notes: Math 607, Polynomial representations of GL$_n$, *http://darkwing.uoregon.edu/∼brundan/teaching.html* pp. 29–31, Spring (1999)

[3] Canteaut, A.,Carlet, C.,Charpin, P.,Fontaine, C.: Propagation Characteristics and Correlation-Immunity of Highly Nonlinear Boolean Functions. EUROCRYPT 2000, Lecture Notes in Comp. Sci., **1807**, pp. 507–522, (2000)

[4] Davis, J.A.,Jedwab, J.: Peak-to-mean Power Control in OFDM, Golay Complementary Sequences and Reed-Muller Codes. IEEE Trans. Inform. Theory **45.** No 7, pp. 2397–2417, Nov. (1999)

[5] Feng, K.,Shiue P.J-S.,Xiang Q., On aperiodic and periodic complementary binary sequences, IEEE Trans. Inf. Theory, **45**, 1, pp. 296–303, Jan. (1999)

[6] Golay, M.J.E.: Multislit spectroscopy. J. Opt. Soc. Amer., **39**, pp. 437–444, (1949)

[7] Golay, M.J.E.: Complementary Series. IRE Trans. Inform. Theory, **IT-7**, pp. 82–87, Apr. (1961)

[8] Harrison, M.A.: The Number of Classes of Invertible Boolean Functions. J. ACM, **10**, pp. 25–28, (1963)

[9] Helleseth, T.,Kumar, P.V.: Sequences with Low Correlation. in *Handbook of Coding Theory*, R.Brualdi,C.Huffman,V.Pless, Eds.

[10] Jones, A.E.,Wilkinson, T.A.,Barton, S.K.: Block Coding Scheme for Reduction of Peak to Mean Envelope Power Ratio of Multicarrier Transmission Schemes. Elec. Lett. **30**, pp. 2098–2099, (1994)

[11] Lidl, L.,Niederreiter, H.: **Introduction to Finite Fields and their Applications** Cambridge Univ Press, pp. 361–362, (1986)

[12] Littlewood, J.E.: On polynomials $\sum \pm z^m$, $\sum \exp(\alpha_m) z^m$,$z = e^{i\theta}$, *J. London Math. Soc.*, **41**, pp. 367–376, (1966)

[13] MacWilliams, F.J.,Sloane, N.J.A.: **The Theory of Error-Correcting Codes** Amsterdam: North-Holland. (1977)

[14] J-S.No, H-Y.Song; "Generalized Sylvester-Type Hadamard Matrices", *Int. Symp. Inf. Theory, Sorrento, Italy*, June 25-30, 2000

[15] Nyberg, K.: Construction of Bent Functions and Difference Sets. *Proc. EuroCrypt90, Lecture Notes in Computer Science (LNCS), Springer, Berlin*, Vol 473, pp. 151–160, (1991)

[16] Parker, M.G.,Tellambura, C.: Generalised Rudin-Shapiro Constructions. *WCC2001, Workshop on Coding and Cryptography, Paris (France)*, Jan 8-12, (2001) *http://www.ii.uib.no/∼matthew/*

[17] Parker, M.G.,Tellambura, C.: Golay-Davis-Jedwab Complementary Sequences and Rudin-Shapiro Constructions. Submitted to IEEE Trans. Inform. Theory, *http://www.ii.uib.no/∼matthew/* March (2001)

[18] Parker, M.G., Rijmen, V.: The Quantum Entanglement of Binary and Bipolar Sequences. Short version in **Sequences and Their Applications**, Discrete Mathematics and Theoretical Computer Science Series, Springer, 2001 Long version at *http://xxx.soton.ac.uk/ps/quant-ph/0107106* or *http://www.ii.uib.no/∼matthew/* Jun (2001)

[19] Parker, M.G.: Spectrally Bounded Sequences, Codes and States: Graph Constructions and Entanglement., *Invited Talk at Eighth IMA International Conference on Cryptography and Coding, Cirencester, UK, 2001, To be published in Lecture Notes in Computer Science, 2001, also http://www.ii.uib.no/∼matthew/*, 17-19 December, 2001

[20] Inequivalent Invertible Boolean Functions for $t = 3$, *http://www.ii.uib.no/∼matthew/mattweb.html*, (2001)

[21] Parker, M.G.,Tellambura, C.: A construction for binary sequence sets with low peak-to-average power ratio. Int. Symp. Inform. Theory, Lausanne, Switzerland, June 30-July 5, (2002)

[22] Parker, M.G.,Paterson, K.G.,Tellambura, C.: Golay Complementary Sequences. Wiley Encyclopedia of Telecommunications, Editor: J.G.Proakis, Wiley Interscience, (2002)

[23] Paterson, K.G.: Generalized Reed-Muller Codes and Power Control in OFDM Modulation. IEEE Trans. Inform. Theory, **46**, No 1, pp. 104-120, Jan. (2000)

[24] Paterson, K.G.,Tarokh V.: On the Existence and Construction of Good Codes with Low Peak-to-Average Power Ratios. IEEE Trans. Inform. Theory **46.** No 6, pp. 1974–1987, Sept (2000)

[25] Paterson, K.G.,: On Codes with Low Peak-to-Average Power Ratio for Multi-Code CDMA. **Sequences and Their Applications**, *Discrete Mathematics and Theoretical Computer Science Series, Springer*, (2001)

[26] Paterson, K.G.: Sequences for OFDM and Multi-Code CDMA: Two Problems in Algebraic Coding Theory. Hewlett-Packard Technical Report, HPL-2001-146, (2001)

[27] Raddum, H.,Parker M.G. Z_4-Linear Cryptanalysis. Technical Report for the New European Schemes for Signatures, Integrity, and Encryption (NESSIE), (2002)

[28] Rudin, W.: Some Theorems on Fourier Coefficients. Proc. Amer. Math. Soc., No 10, pp. 855–859, (1959)

[29] Shapiro, H.S.: Extremal Problems for Polynomials. M.S. Thesis, M.I.T., (1951)

[30] Shepherd, S.J.,Orriss, J.,Barton, S.K.: Asymptotic Limits in Peak Envelope Power Reduction by Redundant Coding in QPSK Multi-Carrier Modulation. IEEE Trans. Comm., **46**, No 1, pp. 5–10, Jan. (1998)

[31] Sloane, N.J.A.: The On-Line Encyclopedia of Integer Sequences. $(1, 2, 154, \ldots)$, *http://www.research.att.com/∼njas/sequences/index.html*

[32] Sloane, N.J.A.: A Library of Hadamard Matrices $(1, 2, 154, \ldots)$, *http://www.research.att.com/ njas/hadamard/index.html*

[33] Tseng, C.-C. Liu, C.L.: Complementary sets of sequences, IEEE Trans. Inform. Theory, **IT-18**, no. 5, pp. 644–651, Sept. (1972)

[34] Turyn, R.: Hadamard matrices, Baumert-Hall units, four-symbol sequences, pulse compression, and surface wave encodings J. Comb. Theory Ser. A, **16**, pp. 313–333, (1974)

Partition Coefficients of Acyclic Graphs *

John L. Pfaltz

University of Virginia

Abstract. We develop the concept of a "closure space" which appears with different names in many aspects of graph theory. We show that acyclic graphs can be almost characterized by the partition coefficients of their associated closure spaces. The resulting nearly total ordering of all acyclic graphs (or partial orders) provides an effective isomorphism filter and the basis for efficient retrieval in secondary storage.

1 Binary Partitions

In this paper we combine two mathematical threads and apply them in a graph-theoretic context. The first thread of binary partitions was studied by Euler as early as 1750. The second thread involving closure spaces is of more recent origin. A binary partition of a positive integer N is its expression as a sum of powers of 2. Mahler [16], and Churchhouse [3] [4] have studied binary partitions from a number theoretic point of view. Because our intention is to connect these partitions with closure spaces, we will confine our attention to the special case where N is also a power of 2.

By a *binary partition* of 2^n we mean a sequence of non-negative integers $< \cdots, a_k \cdots >$, $0 \leq k \leq n$ such that

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 = 2^n \qquad (1)$$

or $\sum_{k=0}^{n} a_k \cdot 2^k = 2^n$. The set of all such partitions we denote by $\mathcal{P}^n$. (From now on we frequently omit the adjective "binary".)

Several characteristics of (1) are readily apparent. First, $a_n \neq 0$ if and only if $a_k = 0$ for all $0 \leq k < n$. Second, since the right hand side is even and all terms $a_k \cdot 2^k$, $k > 0$ must be even, the coefficient a_0 must be even. Third, if $< \cdots, a_k, a_{k-1}, \cdots >$ is a partition, then $< \cdots, a_k - 1, a_{k-1} + 2, \cdots >$ must be as well. And fourth, if $< a_n, \cdots, a_k, \cdots, a_o >$ is a partition of 2^n then $< a_n, \cdots, a_k, \cdots, a_0, 0 >$ is a partition of 2^{n+1}.

With these observations, it is not difficult to write a process which generates all partitions in lexicographic order. Doing so, and displaying each partition, generates the following enumerations of $\mathcal{P}^3$ and $\mathcal{P}^4$. It is quite easy to verify by inspection that each sequence is a partition of 2^n. And because they are in lexicographic order, one can verify that all possible partitions have been generated.

If one were to run the same program with $n = 5$ there would be 202 generated partitions which are are impractical to display in a paper of this length.

* Research supported in part by DOE grant DE-FG05-95ER25254.

n = 3

1	0	0	0
0	2	0	0
0	1	2	0
0	1	1	2
0	1	0	4
0	0	4	0
0	0	3	2
0	0	2	4
0	0	1	6
0	0	0	8

n = 4

1	0	0	0	0
0	2	0	0	0
0	1	2	0	0
0	1	1	2	0
0	1	1	1	2
0	1	1	0	4
0	1	0	4	0
0	1	0	3	2
0	1	0	2	4
0	1	0	1	6
0	1	0	0	8
0	0	4	0	0
0	0	3	2	0
0	0	3	1	2
0	0	3	0	4
0	0	2	4	0
0	0	2	3	2
0	0	2	2	4

0	0	2	1	6
0	0	2	0	8
0	0	1	6	0
0	0	1	5	2
0	0	1	4	4
0	0	1	3	6
0	0	1	2	8
0	0	1	1	10
0	0	1	0	12
0	0	0	8	0
0	0	0	7	2
0	0	0	6	4
0	0	0	5	6
0	0	0	4	8
0	0	0	3	10
0	0	0	2	12
0	0	0	1	14
0	0	0	0	16

Fig. 1. $\mathcal{P}^3$ and $\mathcal{P}^4$

2 Closure Spaces

The preceding discussion of binary partitions will take on additional interest if we introduce the concept of a closure space. We let $\mathbf{U}$ denote some *universe* of elements of interest. Lower case letters $a, b, \cdots, x, y, z$ will denote individual elements of $\mathbf{U}$, and upper case letters will denote subsets. A set, $\mathbf{U}$, and a closure operator, φ, satisfying the following three closure axioms[2]

$$\begin{aligned} X &\subseteq X.\varphi \\ X \subseteq Y \ &\text{ implies } \ X.\varphi \subseteq Y.\varphi \\ X.\varphi.\varphi &= X.\varphi^2 = X.\varphi \end{aligned} \tag{2}$$

are said to be a *closure space* $(\mathbf{U}, \varphi)$, as in [12]. X is said to be *closed*[3] if $X.\varphi = X$. A closure operator, φ, is said to be *uniquely generated* if it also satisfies the following fourth axiom, which serves to distinguish it from a topological closure,

$$X.\varphi = Y.\varphi \ \text{ implies } \ (X \cap Y).\varphi = X.\varphi = Y.\varphi \tag{3}$$

Closure operators satisfying (3) above are uniquely generated in the sense that for any set Z, there exists a unique minimal set $X \subseteq Z$, called its *generator* and denoted $Z.gen$, such that $X.\varphi = Z.\varphi$.[4] The importance of uniquely generated

[2] We will write these expressions using the mixed infix/suffix form more common in algebra. That is, binary set operators will be written using infix and unary transformations will be written using suffix notation, as in $(X \cap Y).f$ to denote the image of $X \cap Y$ under f. This notation greatly simplifies expressions involving transformations of closure spaces; and the redundant dot delimiter is of great value when using computer parsing techniques.

[3] The family $\mathcal{C}$ of closed sets is closed under intersection, and this characterization is equivalent to (2), *c.f.* [9].

[4] Readily, if X_1 and X_2 were distinct minimal generators of $Z.\varphi$, then because $X_1.\varphi = X_2.\varphi = Z.\varphi$, we must have, by (3), $(X_1 \cap X_2).\varphi = Z.\varphi$ contradicting minimality.

closure spaces lies in the fact that in discrete systems they play a role that is in many respects analogous to the vector spaces of classical mathematics. We establish this parallel in the next paragraph.

A closure operator σ, satisfying the three closure axioms of (2), together with the Steinitz-MacLane *exchange* property

$$\text{if } \; y \notin X.\sigma \; \text{ then } \; y \in (X \cup \{x\}).\sigma \; \text{ implies } \; x \in (X \cup \{y\}).\sigma \qquad (4)$$

can be shown to be the closure operator of a matroid [25] [26]. Recall that a *matroid* is a set system which generalizes the independent sets of a linear algebra, and a *vector space* is the closure, usually called the *spanning operator*, of one or more of these independent sets. Now (4) has the familiar interpretation: if y is not in the vector subspace spanned by X, but is in the vector space formed by adjoining x as a basis vector, then x must be in the vector space spanned when y is adjoined to X.

Similarly, a closure φ satisfying the three closure axioms and the *anti-exchange* property

$$\text{if } \; x, y \notin X.\varphi \; \text{ then } \; y \in (X \cup \{x\}).\varphi \; \text{ implies } \; x \notin (X \cup \{y\}).\varphi \qquad (5)$$

is the closure operator of an anti-matroid [7] [15]. In [21] it is shown that

Theorem 1. *A closure operator is uniquely generated if and only if it satisfies the anti-exchange property (5).*

Therefore, uniquely generated closure spaces are precisely the analogs of vector spaces, but with respect to anti-matroids. From now on, we will simply call them *closure spaces*. Because they are uniquely generated, any closure space is completely characterized by enumerating its closed sets and their generators, that is by enumerating $[X.\varphi, X.gen], \forall X \subseteq \mathbf{U}$.

Closure spaces are fairly common in computer science and its applications, although they frequently have other names. Transitive closure, for example of the set of edges in an acyclic graph or of functional dependencies in an acyclic database schema, gives rise to a closure space. The term "convexity" is often applied to closure concepts, and many examples of convexity concepts occurring in graphs can be found in [8] [11] and [14]. Convexity in discrete geometries also yields a number of intuitively satisfying closure spaces. The convex hull operator is the closure operator. See [10] for an excellent treatment of *convex geometries*. Finally, numerous examples of anti-matroids, whose closure will yield a closure space, can be found in the survey of anti-matroids [7] or the text on *greedoids* [15] which generalize an important class of computer algorithms.

We have found that *ideal* and *interval* operators in partially ordered sets, or acyclic graphs provide an abundance of easily accessible examples. It is not hard to show that the path structure of an acyclic graph is uniquely generated [21]. That is, there is a unique, minimal representation[5] of any acyclic graph which

[5] Minimal in the sense that removal of any edge yields a graph with a different path structure, transitive closure, or partial order. We usually illustrate acyclic relationships with basic representations; they are far less cluttered.

we call a *basic graph* [18]. These are commonly used in the implementation of acyclic data structures and processes.

One can organize the *closed sets* of a closure space in many ways. The most natural is to partially order them by inclusion, in which case it can be shown that the partial order will be a lower semi-modular (or meet-distributive) lattice [17] [9]. A more interesting partial order, $\leq_\varphi$, of *all subsets* is given by

$$X \leq_\varphi Y \qquad \text{if and only if} \qquad Y \cap X.\varphi \subseteq X \subseteq Y.\varphi \qquad \forall X, Y \subseteq \mathbf{U}. \tag{6}$$

which is described in [21]. The closure space with this partial order can be shown to be a lattice, $\mathcal{L}_{(U,\varphi)}$, called the *closure lattice* of $(\mathbf{U}, \varphi)$. Figure 2 illustrates the

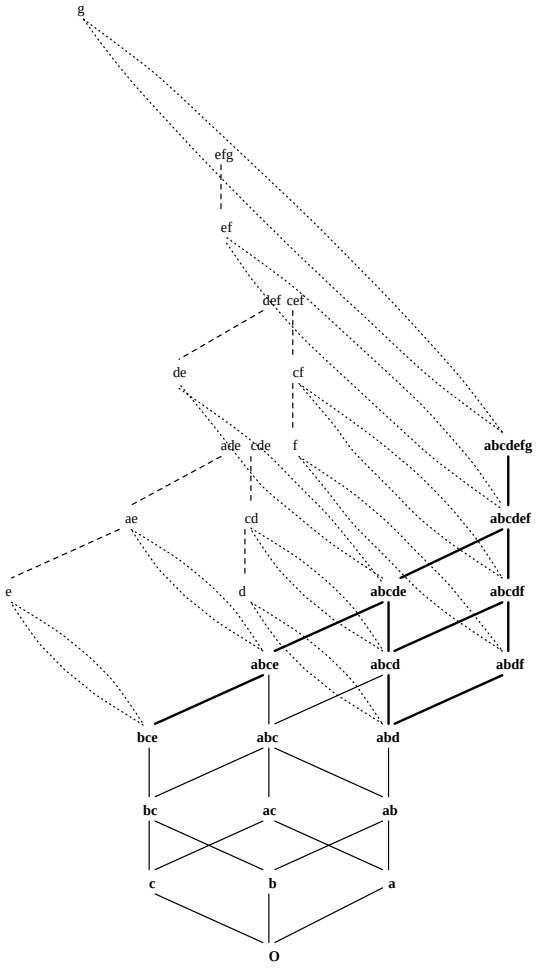


Fig. 2. The closure lattice $\mathcal{L}_{(U,\varphi)}$ of a small 7 point closure space.

structure of a small 7 point closure space. The closed sets of $(\mathbf{U}, \varphi)$ are set in bold face, and connected by solid lines. These closed sets form a sublattice whose partial order is by inclusion. It can be instructive to diagram the points and their set membership of this space. Since $\{g\}$ is the generator of $\mathbf{U} = \{abcdefg\}$, the closure of $\{g\}$, or any set containing the point g, is the whole space. The generator of $\{abce\}$ are the points $\{ae\}$, and so forth. There are 64 subsets whose closure is $\{abcdefg\}$; they constitute the lattice interval $[abcdefg, g]$. To avoid clutter, we simply denote all of them by a single dotted ellipse. Only one of its elements $\{efg\}$ is indicated. (From now on, we also ignore $\{\cdots\}$ delimiting sets of enumerated points.)

Closure lattices such as this have a number of unique properties which are explored in [20]. Central to this development is

Theorem 2. *The poset* $\{Y_i | Y.\varphi \leq_\varphi Y_i \leq_\varphi Y.gen\}$, *is a boolean algebra on* n *elements, where* $n = |Y.\varphi| - |Y.gen|$.

These boolean algebras, $[Y.\varphi, Y.gen]$ are denoted by dotted ellipses in Figure 2. It is not hard to see that each lattice interval, $[Y.\varphi, Y.gen] = \{Y_i | Y.gen \subseteq Y_i \subseteq Y.\varphi\}$, and that $|[Y.\varphi, Y.gen]| = 2^n$. Since every subset $Y \subseteq \mathbf{U}$ is an element of some closure/generator interval, the decomposition of $2^{\mathbf{U}}$ into these intervals is a binary partition of $2^{|U|}$, which we call the *partition coefficients* of $(\mathbf{U}, \varphi)$. For example, the binary partition corresponding to the closure space $(\mathbf{U}, \varphi)$ of Figure 2 is $< 0\ 1\ 0\ 1\ 3\ 4\ 0\ 8 >$ (where $a^n = a^7 = 0$ is the leading coefficient). There is a single interval, $[abcdefg, g]$, of size 2^6; so $a_6 = 1$. The three intervals of size 2^3, $[abcde, de]$, $[abcdf, cf]$, and $[abdf, f]$, imply that $a_3 = 3$. There are eight singleton elements, abc, ab, ac, bc, a, b, c and $\emptyset$, where $X.\varphi = X.gen$. Consequently, $a_0 = 8$. Those partitions for which $a_0 \neq 0$ we call *normal*. Customarily the closure of $\emptyset$ is empty[6], even though it is not required in a the general theory of closure spaces. The closure space of Figure 2 is normal.

These partition coefficients constitute an invariant of the closure space that is independent of representation or isomorphic mappings. It is evident from Theorem 2 and the preceding discussion that for every closure space there is a corresponding binary partition of $2^{|U|}$. It can also be shown that for every binary partition of 2^n there exists a closure space on n elements with that [*closed_set, generator*] structure.

3 Partition Coefficients of Acyclic Graphs

In this section, we apply the concept of closure spaces and their binary partition coefficients to the study of acyclic graphs and partially ordered sets.

With any graph one can postulate a number of invariants. They may be any of a variety of scalar quantities, such as covering or independence numbers [13] or various polynomial expressions, *e.g.* chromatic polynomials [1]. It is desirable

[6] That the convex closure of the empty set should be empty is so reasonable, it is taken to be an axiom in [10] and [11].

if the invariant conveys information about the graph. A fairly popular invariant of G is its *characteristic polynomial* [22]. In fact this terminology is slightly misleading. One is really associating the graph G with a linear transformation τ, for which the adjacency matrix of G is a representation. Now, the characteristic polynomial, eigenvalues, and eigenspaces of τ can be regarded as invariants of G [6].

We now do much the same. Given a poset, or acyclic graph $G = (P, E)$, one can use the path relation ρ to induce a partial order on the point set, P. Now we set $\mathbf{U} = P$, and let

$$\begin{aligned} Y.\varphi_L &= \{x | (x, y) \in \rho, \quad y \in Y\}, \\ Y.\varphi_R &= \{z | (y, z) \in \rho, \quad y \in Y\}, \quad \text{or} \\ Y.\varphi_C &= \{x | (y_1, x) \in \rho, (x, y_2) \in \rho, \quad y_1, y_2 \in Y\}. \end{aligned} \tag{7}$$

The first two closures are *ideal* operators on $\mathbf{U}$, and the last is an *interval* operator.[7]

For any acyclic graph G and uniquely generated closure φ, such φ_L, φ_R or φ_C above, we have an induced closure space. In Figure 3, we illustrate the three different closure spaces obtained by applying φ_L, φ_R, and φ_C to a single 5 point graph. Again, the sub-lattice of closed sets is denoted by solid lines. And, as usual, we will denote the $[closed\_set, generator]$ intervals by dashed ellipses. The partition coefficients of these three closure spaces are $< 0\ 0\ 1\ 3\ 3\ 2 >$, $< 0\ 1\ 0\ 1\ 4\ 4 >$, and $< 0\ 0\ 0\ 1\ 4\ 20 >$ respectively. Readily, different closure operators give rise to different partition coefficients.

We now treat the partition coefficients of this closure space as invariants of G. As observed above, this invariant depends on the closure operator. For the rest of this paper, we use only the ideal closure φ_L of (7). In Figure 4 we show $\mathcal{G}^4$, that is the collection of all basic, acyclic graphs on 4 points, together with the partition coefficients of their closure spaces. Because φ_L is path derived, any graph with additional edges, but the same transitive closure, must have the same associated closure space.

The graphs of $\mathcal{G}^4$ are not uniquely characterized by their coefficients; consider graphs (9) and (10) which both have $< 0\ 0\ 2\ 2\ 4 >$ as partition coefficients. But (9) is connected whereas (10) is disconnected. Unfortunately, this distinction is of little value. The connected, non-isomorphic graphs of Figure 5 both have partition coefficients $< 0\ 1\ 0\ 2\ 2\ 4 >$ with respect to φ_L. We would note that while the partition coefficients of Figures 5(a) and (b) are the same, their corresponding closure spaces, as illustrated by the lattices are distinct. This follows from

[7] In [11], φ_L is called *downset alignment* and φ_C is called *order convexity*, but just plain *convexity* in [17]. There are many conventions for drawing partially ordered sets. In an effort to distinguish between the underlying acyclic graph and its closure space, the author prefers to orient the former horizontally and the latter vertically. Because we illustrate with a left to right horizontal orientation, we use the subscripts, L(eft) and R(ight), to distinguish the ideal operators. The terms *upper/lower ideal* and $\downarrow$ operators are also encountered.

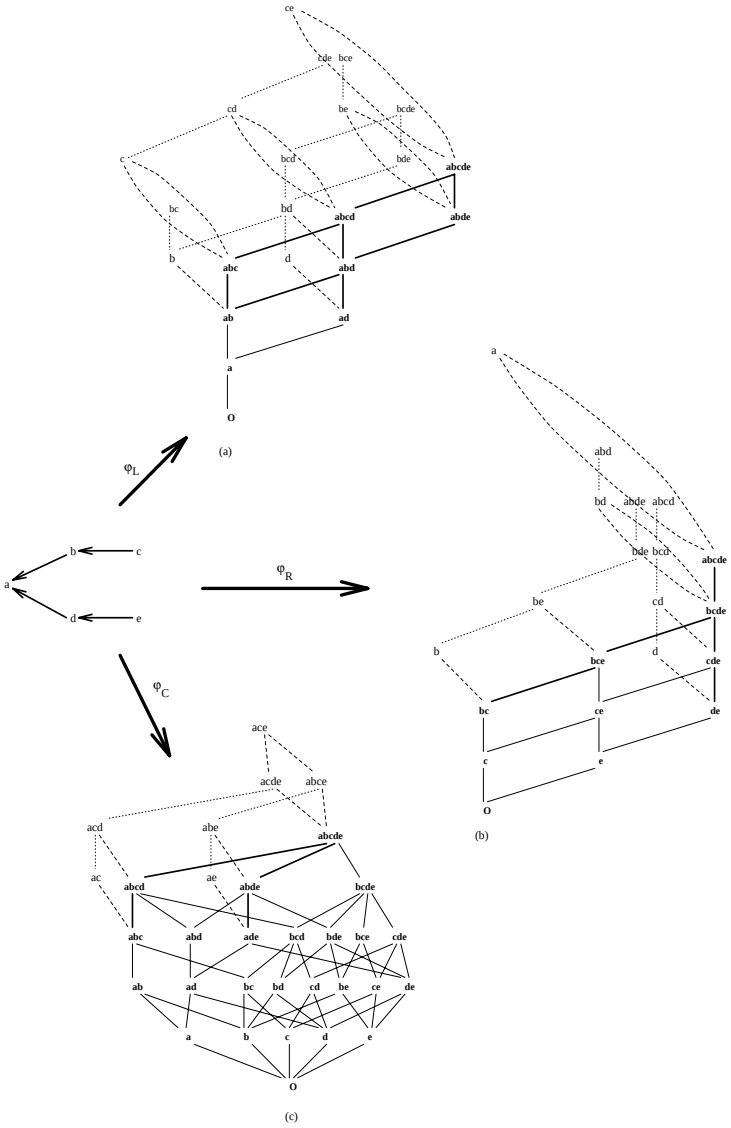


Fig. 3. Different closure spaces arising from the closure operators, φ_L (a), φ_R (b), and φ_C (c).

Theorem 3. Fundamental Theorem of Distributive Lattices *If* $(\mathbf{U}, \varphi)$ *is a finite closure space in which* $\mathbf{U}$ *is partially ordered and* φ *is an ideal operator, then the set of closed sets, partially ordered by inclusion, is a distributed lattice. Moreover, there is a one-to-one correspondence between the set of all distributive lattices and such closure spaces.*

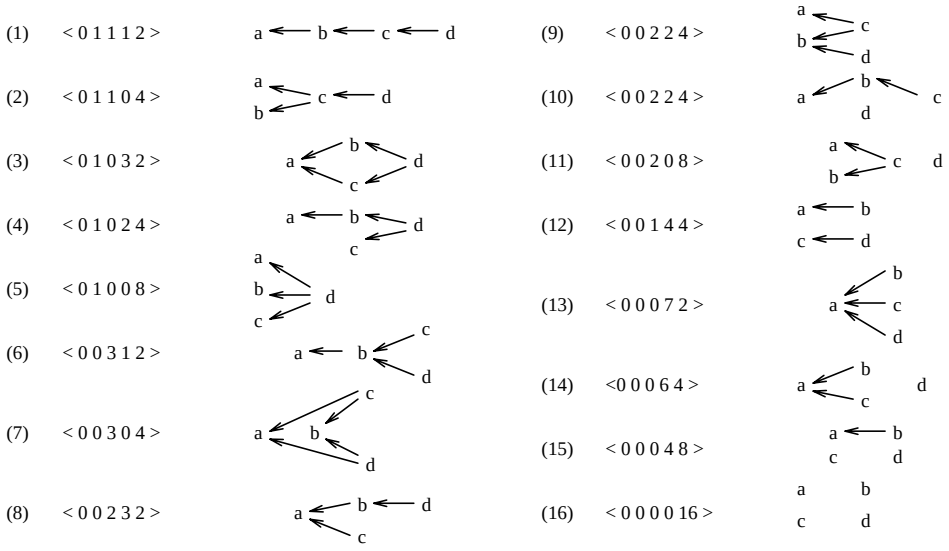


Fig. 4. All basic, acyclic 4 point graphs, $\mathcal{G}^4$ and their partition coefficients (w.r.t. φ_L)

Proof. See theorem 3.4.1 of [24] □

Distinct, non-isomorphic, graphs must have distinct closure spaces, but distinct closure spaces may have the same partition coefficients, just as two distinct linear transformations may have the same characteristic polynomial. Consequently, acyclic graphs cannot be completely characterized by their partition coefficients. Nevertheless, these coefficients convey significant information about the graphs and can be quite useful when manipulating them in computer systems.

The author has created one such computer system, capable of representing arbitrary graphs, whose primary purpose is the study of properties of graph transformations. For many of the studies of interest to us, we must generate all, or a large sample of, non-isomorphic graph on n points. Comparing binary partition coefficients is a useful filter for eliminating obviously non-isomorphic pairs. In Table 1 we display the expected number of acyclic graphs on n points that have the same identical binary partition coefficients, $exp(|G|\ per\ bp)$. For $n = 8$, there exist 16,999 distinct, non-isomorphic, acyclic graphs,[8] having 5,187 distinct partition coefficient sequences; so that an expected 3.277 have the same binary partition coefficients. But two graphs with the same partition coefficients need not have the same number of edges. They frequently do not. As shown on the next line of Table 1, the expected number of graph with identical partition coefficients and the same number of edges, $exp(|G|\ per\ bp\ and\ |E|)$, drops to

[8] The number of distinct n point acyclic graphs, or posets, grows exponentially. It is known that $|\mathcal{G}^n|$ is: 183,231 ($n = 9$), 2,567,284 ($n = 10$) and 46,794,427 ($n = 11$) [5]. No general enumeration formula is known.

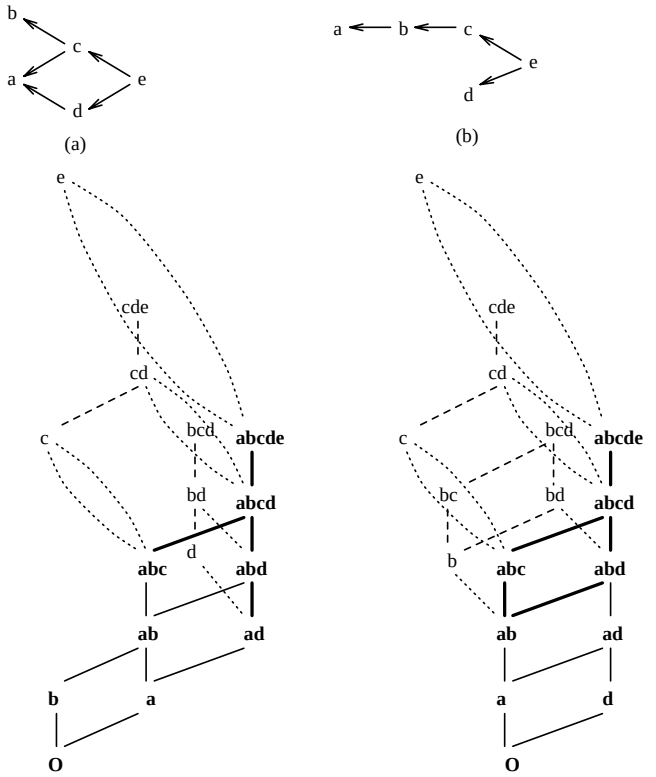


Fig. 5. Two graphs (a) and (b) having the partition coefficients < 0 1 0 2 2 4 > together with their corresponding closure spaces

$n = \|P\|$	3	4	5	6	7	8
$\|\mathcal{G}^n\|$	5	16	63	318	2,405	16,999
$exp(\|G\|\ per\ bp)$	1.00	1.07	1.21	1.53	2.13	3.28
$exp(\|G\|\ per\ bp$ and $\|E\|)$	1.00	1.00	1.03	1.12	1.30	1.66

Table 1. Densities of acyclic graphs on n points when partitioned w.r.t. binary partition (bp) coefficients and w.r.t number of $edges$.

1.656. In practice, these expectations transalate into a effective filter. In a recent application that involved testing 1,034 M random pairs of 8-point graphs for isomorphism (equality), we first applied the edge cardinality filter; 193 M pairs passed this filter. Of these, only 148,762 had identical partition coefficients, and of these 87,710 were actually isomorphic. The probability of being isomorphic, given equal partition coefficients and numbers of edges was 1.69, compared to 1.66 as predicted by the table.

A quick measure of the effectiveness of invariant partition coefficients as an

isomorphism filter can be attained by comparing it with other common filters. In Table 2, we count the number of equivalence classes generated in the family $\mathcal{G}^n$ of all n point acyclic graphs, assuming (a) partition coefficients alone, (b) partition coefficients plus equal edge cardinalities, (c) equal in (left) and out (right) degrees, (d) equal in (left) and out (right) ideals, and (e) equal ideals plus equal edge cardinalities. Readily, the expected number of graphs passing

		nbr of equivalence classes				
		(a)	(b)	(c)	(d)	(e)
n	$\lvert\mathcal{G}^n\rvert$	coeff	$+\lvert E\rvert$	degree	ideal	$+\lvert E\rvert$
4	16	15	16	16	15	16
5	63	52	61	63	52	61
6	318	208	285	125	208	284
7	2,045	962	1,570	432	951	1,551
8	16,999	5,187	10,263	1,588	4,932	9,863

Table 2. Comparison of isomorphism filters on graphs with n points

any filter, as in Table 1, is the expected number of graphs per equivalence class. The similarity of (a) and (b) with (d) and (e) is striking. This should not be too surprising, since φ_L is an ideal operator. But, it is a one-sided ideal operator, whereas (d) and (e) in Table 2 are based on two-sided ideals. Moreover, storage of filter (e) requires $2 \cdot n + 1$ integers whereas filter (b) consists of just $n + 1$ integers. In terms of information content, the partition coefficients are nearly twice as efficient. There may be more effective isomorphism filters, but we know of none with as dense information content.

A lexicographic ordering of the partition coefficients is an invariant, *nearly total* ordering of all acyclic graphs on n points. This can be of considerable value. In particular, we can use binary search to quickly obtain the neighborhood of any desired graph. The 4 point graphs of Figure 4 have been displayed in this order.

Another use of our graph manipulation system has been to gather various counts regarding basic, acyclic graphs on $n = |P|$ points with $e = |E|$ edges. Some of these results are summarized in Table 3. The numbers of trees on n points, connected graphs with $n - 1$ edges, is evident. We would observe that the counts are quite different from the similar table of [5] which has graphs with many more edges. They enumerate the *transitively closed* graphs (or partial orders) with e edges, whereas we enumerate the basic (or minimal) graphs with that order. Using the terminology of this paper, they count the edges in the closure of a partial order, while we count the edges in its generator.

The partition coefficients appear to encode a considerable amount of additional graph specific information. For example, it is not difficult to prove that:

$\|P\| =$	3		4		5		6		7		8	
$\|E\|$	nc	c	nc	c	nc	c	nc	c	nc	c	nc	c
0	1		1		1		1		1		1	
1	1		1		1		1		1		1	
2		3	4		4		4		4		4	
3				8	11		12		12		12	
4				2	2	27	43		46		47	
5						12	14	91	156		170	
6						5	5	87	110	350	670	
7								45	50	532	721	1,376
8								12	12	475	550	3,272
9								3	3	201	216	4,298
10										71	74	3,197
11										14	14	1,565
12										7	7	554
13												186
14												44
15												16
16												4
Totals	2	3	6	10	19	44	80	238	395	1,650	2,487	14,512
$\|\mathcal{G}^n\|$	5		16		63		318		2,045		16,999	

Table 3. Numbers of disconnected (nc) and connected (c) acyclic graphs on $|P|$ points with $|E|$ edges

Theorem 4. *If the closure operator is φ_L, then a_0 must be a power of two, whose exponent denotes the number of minimal (leftmost) elements.*

It also appears that partition coefficients encode a measure of connectivity information. After a tedious sequence of minor lemmas such as

Lemma 5. *Let φ be a path based closure and let $G^{(n)} = (P, E)$ on n points have the closure coefficients $< a_n,\ a_{n-1}, \cdots,\ a_0 >$. Then, $G^{(n+1)} = (P \cup \{x\}, E)$ has the closure coefficients $< 2 \cdot a_n,\ 2 \cdot a_{n-1}, \cdots,\ 2 \cdot a_0 >$.*

Lemma 6. *Let φ be the left (right) ideal closure. $G^{(n)} = (P, E)$ has a greatest (least) point if and only if the partition coefficient $a_{n-1} = 1$.*

one finally derives a curious result,

Theorem 7. *Let φ be an ideal closure. If all the binary partition coefficients of a graph, $< a_n,\ a_{n-1},\ \cdots\ ,\ a_1,\ a_0 >$ (w.r.t an ideal closure) are even, then the graph is disconnected or else there exists a disconnected graph with these binary partition coefficients.*

Suggested by this result, but not stated is the fact that if all the binary partition coefficients associated with a graph are even, then, with very high probability,

the graph is disconnected. On the other hand, if even one coefficient is odd, the graph is probably connected.

Of major concern with the use of closure spaces and their binary partition coefficients as tools for the analysis and filtering of acyclic graphs, is the expected cost of generating them. The straightforward approach of generating all 2^n subsets and calculating their closures is clearly impractical for even moderate sized graphs. Fortunately, this is unnecessary. Given any closed set in a closure space, and its generator, one can easily determine all closed sets that it covers because,

Theorem 8. *If φ is uniquely generated, and if $X \neq \emptyset$ is closed, then $p \in X.gen$ if and only if $Z - \{p\}$ is closed.*

Proof. See Lemma 3.1, [20].

This theorem is treated as the defining property of *extreme points*, which are the generators of convex sets in [10]. It appears in one form or another in many efficient graph algorithms. For example, this property is used in [5] [2] to generate partial orders, where the universe **U** is the edge set; their closure is transitive closure; and a "cover" is a minimal edge set that generates the transitive closure. In [11] and [8] it is exploited to characterize properties of undirected graphs, and efficient algorithms to recognize them.

In our case, we use Theorem 8, to determine the closure space of a graph and its partition coefficients by first putting the entire point set P, which must be closed, in a queue. We then successively remove closed sets Y from the queue, verify that we have not already processed it,[9] then apply $generator(Y)$. We increment a_k where $k = |Y| - |Y.gen|$. For each $y \in Y.gen$, we add $Y - \{y\}$ to the queue. The cost of generating the closure space is approximately $|closedsets| \cdot cost_{generator(Y)}$. Assuming $cost_{generator(Y)}$ is nearly constant[10] given G, the cost of generating partition coefficients will be clearly dominated by the number of closed sets to be processed.

So, the key question becomes "what is the expected number of closed sets in an acyclic graph on n points?" The number of closed sets in any particular G is given by the sum of its partition coefficients, $\sum_{i=0}^{n} a_i$. Table 4 enumerates these expected values for $3 \leq n \leq 8$, first for those graphs with precisely $|E|$ edges, and then for all graphs in $\mathcal{G}^n$. Readily, the worst case behavior is $O(2^n)$; but this occurs only if $|E| = 0$. As $|E|$ increases the number of closed sets decreases towards an asymptote. If G must be connected, so that $|E| \geq n - 1$, the number of number of closed sets is close to the asymptote itself.

4 Counting Binary Partitions

We close by once again considering binary partitions. The space of acyclic graphs grows exponentially. Is it reasonable to expect to characterize them by partition

[9] Because we use a queue, this is a level by level processing of the closed sets. It is possible to reach the same closed set twice. *C.f.* figure 2.

[10] With these small, finite graphs there is a hard upper bound for any n.

	$\|P\|$					
$\|E\|$	3	4	5	6	7	8
0	8.0	16.0	32.0	64.0	128.0	256.0
1	6.0	12.0	24.0	48.0	96.0	192.0
2	4.7	9.2	18.5	37.0	74.0	148.0
3		7.1	14.2	28.2	56.5	113.0
4		6.5	10.9	21.8	43.5	86.9
5			9.5	16.6	33.1	66.0
6			9.6	14.5	25.6	50.7
7				13.7	21.7	39.0
8				13.9	20.2	33.0
9				13.0	19.5	29.7
10					19.2	28.1
11					18.3	27.1
12					18.8	26.5
13						26.1
14						26.0
15						26.8
16						25.2
all graphs	5.6	8.4	12.2	17.1	23.6	32.3

Table 4. Expected numbers of closed sets in graphs with $|P|$ points and $|E|$ edges

coefficients as n becomes large? How many binary partitions are there on n points? It is customary to let $b(n)$, called the binary partition function, denote the *number* of binary partitions of n. As before, our interest is the number of binary partitions of 2^n, that is $b(2^n)$. In [3], it is shown that

$$b(2^n) = c_{n,0} + c_{n,1} \tag{8}$$

where $c_{1,0} = c_{1,1} = 1$, $c_{n+1,0} = c_{n,0} = 1$, and $c_{n+1,i} = \sum_{k=0}^{2i} c_{n,k}$. This particularly simple formulation was executed by Churchhouse on an Atlas computer in 1968 to obtain initial values of the binary partition function. A more complex, but somewhat faster code is given in [19]. With this code one can generate the following Table 5 of partitions of 2^n. The second column counts the number of *normal* partitions in which $a_0 \neq 0$, which in accordance with observation four in Section 1, is always $|\mathcal{P}^n| - |\mathcal{P}^{n-1}|$. Closure spaces associated with acyclic graphs must be normal. The third column counts the number of such non-isomorphic graphs on n points. It is easy to verify all sequences in Sloane's Handbook of Integer Sequences [23]. The point of this table is to illustrate that while the diversity of acyclic graphs on n points has exponential growth, the variety of closure spaces has super exponential growth; specifically $b(2^n) \sim (2^n)^{n/2}$. The concept of uniquely generated closure spaces is clearly rich enough to be embraced as a tool in the study of acyclic graphs and partially ordered spaces.

n	$b(2^n) = \|\mathcal{P}^n\|$	$\|normal\|$	$\|\mathcal{G}^n\|$
1	2	1	1
2	4	2	2
3	10	6	5
4	36	26	16
5	202	166	63
6	1,828	1,626	318
7	27,338	25,510	2,045
8	692,004	664,666	16,999
9	30,251,722	29,559,718	183,231
10	2,320,518,948	2,290,267,226	2,567,284

Table 5. Number of partitions of 2^n, of normal partitions, of acyclic graphs

References

1. G. D. Birkhoff and D. Lewis. Chromatic polynomials. *Trans. Amer. Math. Soc.*, 60:355–451, 1946.
2. Richard A. Brualdi, Hyung Chan Jung, and William T. Trotter, Jr. On the poset of all posets on n elements. *Discrete Applied Mathematics*, 1994. To appear.
3. R.F. Churchhouse. Congruence properties of the binary partition function. *Proc. Cambridge Phil. Soc.*, 66(2):371–376, 1969.
4. R.F. Churchhouse. Binary partitions. In A.O.L. Atkin and B.J. Birch, editors, *Computers in Number Theory*, pages 397–400. Academic Press, 1971.
5. Joseph C. Culberson and Gregory J. E. Rawlins. New results from an algorithm for counting posets. *Order*, 7:361–374, 1991.
6. Dragos Cvetkovic, Peter Rowlinson, and Slobodan Simic. A study of eigenspaces of graphs. *Linear Algebra and Its Applic.*, 182:45–66, Mar. 1993.
7. Brenda L. Dietrich. Matroids and antimatroids — a survey. *Discrete Mathematics*, 78:223–237, 1989.
8. Feodor F. Dragan, Falk Nicolai, and Andreas Brandstadt. Convexisty and hhd-free graphs. Technical Report SM-DU-290, Herhard-Mercator Univ., Duisburg, Germany, May 1995.
9. Paul H. Edelman. Meet-distributive lattices and the anti-exchange closure. *Algebra Universalis*, 10(3):290–299, 1980.
10. Paul H. Edelman and Robert E. Jamison. The theory of convex geometries. *Geometriae Dedicata*, 19(3):247–270, Dec. 1985.
11. Martin Farber and Robert E. Jamison. Convexity in graphs and hypergraphs. *SIAM J. Algebra and Discrete Methods*, 7(3):433–444, July 1986.
12. George Gratzer. *General Lattice Theory*. Academic Press, 1978.
13. Frank Harary. *Graph Theory*. Addison-Wesley, 1969.
14. Robert E. Jamison-Waldner. Partition numbers for trees and ordered sets. *Pacific J. of Math.*, 96(1):115–140, Sept. 1981.
15. Bernhard Korte, Laszlo Lovasz, and Rainer Schrader. *Greedoids*. Springer-Verlag, Berlin, 1991.
16. K. Mahler. On a special functional equation. *J. London Math. Soc.*, 15(58):115–123, Apr. 1940.

17. John L. Pfaltz. Convexity in directed graphs. *J. of Comb. Theory*, 10(2):143–162, Apr. 1971.
18. John L. Pfaltz. *Computer Data Structures*. McGraw-Hill, Feb. 1977.
19. John L. Pfaltz. Partitions of 2^n. Technical Report TR CS-94-22, University of Virginia, June 1994.
20. John L. Pfaltz. Closure lattices. *Discrete Mathematics*, 1995. (to appear), preprint available as Tech. Rpt. CS-94-02 through home page http://uvacs.cs.virginia.edu/.
21. John L. Pfaltz. Partially ordering the subsets of a closure space. *ORDER*, 1995. (submitted).
22. A. J. Schwenk. Computing the characteristic polynomial of a graph. In R. Bari and F. Harary, editors, *Graphs and Combinatorics*, pages 153–172. Springer Verlag, 1974.
23. N. J. A. Sloane. *A Handbook of Integer Sequences*. Academic Press, 1973. On-line version at 'sequences@research.att.com'.
24. Richard P. Stanley. *Enumerative Combinatorics, Vol 1*. Wadsworth & Brooks/Cole, 1986.
25. W. T. Tutte. *Introduction to the Theory of Matroids*. Amer. Elsevier, 1971.
26. D.J.A. Welsh. *Matroid Theory*. Academic Press, 1976.

This article was processed using the LaTeX macro package with LLNCS style

COMPOSITIONS WITH m DISTINCT PARTS

Arnold Knopfmacher
M. E. Mays

University of the Witwatersrand
West Virginia University

February 27, 1996

Abstract. We study $F(n,m)$, the number of compositions of n in which repetition of parts is allowed, but exactly m *distinct* parts are used. We obtain explicit formulas, recurrence relations, and generating functions for $F(n,m)$ and for auxiliary functions related to F. We also consider the analogous functions for partitions.

Introduction

One of the problems considered by Wilf in [7] involves the number of different sizes of parts in a partition of the integer n. This paper investigates the function $F(n,m)$, which gives the number of compositions of n in which repetition of parts is allowed, but m distinct parts are used in all. For example, in the table below we note $F(4,2) = 5$, from the five compositions of 4 with 2 distinct parts: 3+1, 1+3, 2+1+1, 1+2+1, and 1+1+2.

The first observation to make about $F(n,m)$ is that for $m = 1$ there is a composition of n with one distinct part k if and only if k is a divisor of n. Hence $F(n,1) = d(n)$, the divisor counting function. We extend this sum over divisors to the second column in the next section.

On the right hand boundary of the table, we note that the first non-zero entry in column m is at $n = m(m+1)/2$, where $F(m(m+1)/2, m) = m!$ This is because there are $m!$ arrangements of the summands in the first integer with m distinct parts: $1 + 2 + \ldots + m$.

In order to understand $F(n,m)$ we provide explicit formulas, recurrences, and generating functions for functions related to F. Some of the formulas we derive have immediate analogues to formulas for partitions, especially those in which the ordered structure of compositions manifests itself in a general summand involving a binomial coefficient. We develop the partition formulas in the fourth section.

1991 *Mathematics Subject Classification.* Primary 05A15 Secondary 11P81, 11B83, 05A10.
Key words and phrases. composition, partition, Ferrars graph.
The second author thanks the Centre for Applicable Analysis and Number Theory at The University of the Witwatersrand for sponsoring his visit during January and February 1996.

Table 1. *Compositions of n with m distinct parts, $F(n,m)$, $1 \le n \le 16$, $1 \le m \le 5$*

$n\backslash m$	1	2	3	4	5
1	1				
2	2				
3	2	2			
4	3	5			
5	2	14			
6	4	22	6		
7	2	44	18		
8	4	68	56		
9	3	107	146		
10	4	172	312	24	
11	2	261	677	84	
12	6	396	1358	288	
13	2	606	2666	822	
14	4	950	5012	2226	
15	4	1414	9542	5304	120
16	5	2238	17531	12514	480

Basic recurrences

Since $F(n,1) = d(n)$, it is natural to look for an interpretation of later columns involving divisors of n. We begin by offering an explicit formula for the case $m = 2$.

Theorem 1. *For $n \ge 2$,*

$$F(n,2) = \sum_{j=2}^{\lfloor n/3 \rfloor} \sum_{k=1}^{\lfloor n/j \rfloor - 1} \sum_{\substack{d \mid (n-jk) \\ k \ne d < \frac{n-jk}{j-1}}} \binom{j + (n-jk)/d}{j} - \sum_{\substack{j=1 \\ j \mid n}}^{\lfloor n/3 \rfloor} \binom{2j}{j} \left\lfloor \frac{n/j - 1}{2} \right\rfloor + \sum_{k=1}^{n-1} \sum_{\substack{d \mid (n-k) \\ k \ne d}} (1 + (n-k)/d).$$

Proof. Consider a composition of n into two distinct parts. Write one of the parts as k, where $1 \le k \le n-1$ and count the number of compositions according to the number of occurrences of the part k. If k occurs exactly once then the remainder of the composition is just a composition of $n-k$ with one distinct part, say d, with $d \ne k$. For this to be possible we must have $d \mid n-k$ and then the composition of n consists of $(n-k)/d$ parts equal to d and one part k which can be inserted in any of $(n-k)/d + 1$ places. Thus the number of compositions of n with exactly two distinct parts in which one of these parts occurs once only is

$$\sum_{k=1}^{n-1} \sum_{\substack{d \mid (n-k) \\ d \ne k}} (1 + \frac{n-k}{d}) - 2 \left\lfloor \frac{n-1}{2} \right\rfloor .$$

The reason for the subtracted term is that compositions into exactly two parts are counted twice (e.g. for $n = 6$ and $k = 2$ we count the compositions $2+4$ and $4+2$, and again when $k = 4$).

Next suppose the distinct part k occurs twice. That leaves a composition of $n-2k$ consisting of $(n-2k)/d$ copies of part d, where $k \neq d$, $d \mid (n-2k)$, and $d < n-2k$ since the case in which either distinct part occurs only once has already been covered. The number of different orderings of two k's and $(n-2k)/d$ d's is $\binom{2+(n-2k)/d}{2}$. Since the part d appears at least twice we need $n-2k \geq d \geq 2$, whence $1 \leq k \leq \lfloor n/2 \rfloor - 1$. Thus compositions with two distinct parts in which one part occurs twice and the other part occurs two or more times are enumerated by

$$\sum_{k=1}^{\lfloor n/2 \rfloor - 1} \sum_{\substack{d \mid (n-2k) \\ d \neq k \\ d < n-2k}} \binom{2+(n-2k)/d}{2} - \begin{cases} \binom{4}{2} \left\lfloor \frac{(n/2-1)}{2} \right\rfloor & , n \text{ even} \\ 0 & , n \text{ odd.} \end{cases}$$

The subtracted term here deals with compositions of n into two distinct parts which each appear twice (e.g. $6 = 1+1+2+2$), which are counted twice in the left sum. Such compositions are possible only if n is even. In this case we have $\binom{4}{2}$ orderings of the composition.

In general, suppose the part k occurs j times and the other part d occurs j or more times. In the same manner as above we require $d \mid (n-jk)$, $k \neq d$, and $d < (n-jk)/(j-1)$. There are $\binom{j+(n-jk)/d}{j}$ ordered arrangements of j k's and $(n-jk)/d$ d's, and $n - jk \geq jd \geq j$ implies $k \leq \lfloor n/j \rfloor - 1$. In the case that $j \mid n$ there are $\binom{2j}{j}$ compositions with exactly j k's and j d's that are counted twice for given values of k and d. The number of different possibilities for k and d is given by the number of solutions to $n/j = k + d$ which just interchange k and d, this number being $\lfloor (n/j - 1)/2 \rfloor$.

Finally, summing all these cases over j yields the formula. Since $jk + jd \leq n$, $j \leq n/(k+d) \leq n/(1+2) = n/3$ which provides the limit for the outer sum.

In the next theorem and in later theorems we will use the auxiliary function $F(n, m, j)$ to represent the number of compositions of n into m distinct parts, using exactly j parts altogether.

Theorem 2. *For $j \geq 2$ and $n > j$,*

$$F(n,2,j) = \sum_{r=1}^{\lfloor j/2 \rfloor} \sum_{\substack{k=1 \\ (j-r) \mid (n-kr) \\ k \neq (n-kr)/(j-r)}}^{\lfloor n/r \rfloor - 1} \binom{j}{r} - \begin{cases} \binom{j}{\frac{j}{2}} \left\lfloor (\frac{2n}{j} - 1)/2 \right\rfloor, & \text{if } 2 \mid j \text{ and } j \mid 2n \\ 0, & \text{otherwise.} \end{cases}$$

Proof. We proceed as in the proof of the formula for $F(n,2)$ by counting the compositions according to the number of occurrences of one part k. If k occurs exactly once then $n-k$ must be given as a sum of $j-1$ equal numbers, each $(n-k)/(j-1)$, provided this is a positive integer. Thus we require $j-1 \mid n-k$, $k \neq (n-k)/(j-1)$, and there are j possible arrangements of the parts. In the case that $j = 2$ we need to subtract off compositions into two parts which are counted twice. Thus compositions with two distinct parts, j parts in all, and one part occurring once are

enumerated by

$$\sum_{\substack{k=1\\(j-1)\mid(n-k)\\k\neq(n-k)/(j-1)}}^{\lfloor n/j\rfloor-1} j - \begin{cases} 2\left\lfloor\frac{n-1}{2}\right\rfloor & ,j=2\\ 0 & ,j>2.\end{cases}$$

In general suppose part k occurs r times so that $n-rk$ must be a sum of $j-r$ equal numbers. Thus we require $j-r \mid n-kr$ and $k \neq (n-kr)/(j-r)$. There are $\binom{j}{r}$ ways of arranging the sequences of r summands of size k and $j-r$ summands of size $(n-kr)/(j-r)$. As in the earlier proof we require $n-rk \geq r$ so $k \leq \lfloor n/r \rfloor - 1$. To ensure the other distinct part occurs at least r times we need $j-r \geq r$ so $r \leq j/2$ in the outer sum. Compositions are counted twice and must be subtracted off in the event that both distinct parts occur the same number $j/2$ of times. This is possible only if $2 \mid j$ and then $\frac{j}{2}(k+d) = n$ implies that $j \mid 2n$.

We note that from the relation valid for $n \geq 2$,

$$F(n,2) = \sum_{j=1}^{n-1} F(n,2,j),$$

we can recover $F(n,2)$ as a threefold sum using the formula for $F(n,2,j)$ from Theorem 2.

Perhaps these formulas can be extended to later columns, but the number of special cases to consider becomes forbidding. Another family of results enumerates the compositions by first considering possible values of the summands in the composition.

Lemma 3. *Denote by $F^*(n,m,j)$ the number of compositions of n with exactly m distinct parts, j parts in all, and at least one part being a 1. Then*

$$F(n,m,j) = F(n-j,m,j) + F^*(n,m,j). \tag{1}$$

Proof. Divide the compositions counted by $F(n,m,j)$ into two classes: those with at least one part a 1 and those having no 1's. Compositions in the first class are enumerated by $F^*(n,m,j)$. For compositions in the second class, subtract 1 from each of the j parts, to obtain a composition of $n-j$ into m distinct parts, still with j parts in all.

As an application of this lemma, we can fix m and j and consider the sequence of values $\{F(n,m,j)\}$, $n \geq 1$. Even though the sequence may fail to be monotone, each subsequence consisting of every jth term from an arbitrary starting point *will* be monotone.

Lemma 4.

$$F(n,m,j) = F(n-j,m,j) + \sum_k{}^* \binom{j}{k} F(n-j,m-1,j-k), \tag{2}$$

where $\sum^*$ indicates a sum over those k for which a composition of n into m distinct parts, j parts in all, can have exactly k 1's.

Proof. Consider a composition counted by $F^*(n,m,j)$ in (1). Decrease each part by 1. Then n is reduced to $n-j$, m is reduced to $m-1$, and j is reduced by the

number of 1's that were in the original composition. Summing over appropriate k provides

$$F^*(n,m,j) = \sum_k^* \binom{j}{k} F(n-j, m-1, j-k).$$

Surprisingly, the restriction on k imposed by $\sum^*$ is more complicated for $m = 2$ than for larger values of m. For k ones to be summands in a composition of n into j parts for $m = 2$, n must have a representation of the form $n = k \cdot 1 + (j-k) \cdot d$ for $d \geq 2$. Hence we must have $j - k \mid n - k$. We offer a brief table for $m = 2$ that makes the pattern clear in this case.

Table 2. *Summands k for $m = 2$ in $\sum^*$, $3 \leq n \leq 17, 2 \leq j \leq 9$*

$n\backslash j$	2	3	4	5	6	7	8	9
3	1							
4	1	2						
5	1	1, 2	3					
6	1	2	2, 3	4				
7	1	1, 2	1, 3	3, 4	5			
8	1	2	2, 3	2, 4	4, 5	6		
9	1	1, 2	3	1, 3, 4	3, 5	5, 6	7	
10	1	2	1, 2, 3	4	2, 4, 5	4, 6	6, 7	8
11	1	1, 2	3	2, 3, 4	1, 5	3, 5, 6	5, 7	7, 8
12	1	2	2, 3	4	3, 4, 5	2, 6	4, 6, 7	6, 8
13	1	1, 2	1, 3	1, 3, 4	5	1, 4, 5, 6	3, 7	5, 7, 8
14	1	2	2, 3	2, 4	2, 4, 5	6	2, 5, 6, 7	4, 8
15	1	1, 2	3	3, 4	3, 5	3, 5, 6	1, 7	3, 6, 7, 8
16	1	2	1, 2, 3	4	1, 4, 5	4, 6	4, 6, 7	2, 8
17	1	1, 2	3	1, 2, 3, 4	5	2, 5, 6	5, 7	1, 7, 8

For $m \geq 3$, eventually Σ^* is an unrestricted sum.

Proposition 5. *Let $m \geq 3$. Then for $n \geq \frac{m(m+1)}{2} + 2(j-m)$, $\sum_k^* = \sum_{k=1}^{j-(m-1)}$.*

Proof. Let $n = 1^{a_1}2^{a_2}\ldots n^{a_n}$ denote the partition $n = a_1 1 + a_2 2 + \ldots + a_n n$, where a_i is the number of occurrences i in the partition of n. Consider $n_0 = m(m+1)/2 + 2(j-m)$. Then for any value of k, $1 \leq k \leq j - (m-1)$, we have the partition $1^k 2^b 3^1 \ldots (m-1)^1 (m-1+k)^1$, where $b = j - m - k + 2$, which is a partition of $(k-1)1 + (m-1)m/2 + (m+k-1) + 2(j-m-(k-1)) = m(m+1)/2 + 2(j-m) = n_0$ into m distinct parts, namely $1, 2, \ldots, m-1$ and $m+k-1$, with total number of parts $k + m - 2 + b = j$ as required. Note $b = j - m + 2 - k \geq j - m + 2 - (j - m + 1) \geq 1$ so the part 2 does occur at least once. If $n > n_0$, say $n = n_0 + r$ for $r \geq 1$, then the corresponding partition with k 1's, $1 \leq k \leq j - m + 1$, is $n = 1^k 2^b 3^1 \ldots (m-1)^1 \ldots (m-1+k+r)^1$. We note that n_0 is the smallest value of n for which $\sum_k{}^* = \sum_{k=1}^{j-m+1}$, since if $k = 1$ the smallest number with m different parts and j parts in all is $1+2+3+\ldots+m+2(j-m) = n_0$.

Corollary 6.

$$F(n,2)=\sum_{j=1}^{n-1}\sum_{l=1}^{\lfloor\frac{n-1}{j}\rfloor}\sum_{\substack{k\\ j-k\mid n-lj}}\binom{j}{k}=\sum_{\substack{n=a+b\\ 1\le a,b<n}}\sum_{j\mid a}\sum_{\substack{k\mid b\\ k\neq j}}\binom{j}{k}$$

Proof. Observe that $F(n-j,1,j-k)=\begin{cases}1 & ,j-k\mid n-j\\ 0 & ,\text{otherwise}\end{cases}$. Hence (2) gives

$$(3)\quad F(n,2,j)=\sum_{\substack{k\\ j-k\mid n-j}}{}^{*}\binom{j}{k}+F(n-2j,2,j)=\sum_{\substack{k\\ j-k\mid n-j}}\binom{j}{k}+F(n-2j,2,j).$$

The last equality is justified by the remark before Table 2, where the condition on k imposed in Σ^* is $j-k\mid n-k$. Since $n-k=(n-j)+(j-k)$, this condition already holds by the constrain $j-k\mid n-j$. Now (2) can be applied again to the last term, and iterated for all first arguments $n-lk$ as long as $lj<n$. This gives the limits $1\le l\le\left\lfloor\frac{n-1}{j}\right\rfloor$. The first formula in the lemma then follows by writing $F(n,2)=\sum_{j=1}^{n-1}F(n,2,j)$. Now make the change of variables $a=jl$, $b=n-a$, $k=j-k$, and note $\binom{j}{j-k}=\binom{j}{k}$.

Corollary 6 is in the spirit of the sum over divisors of Theorems 1 and 2, but it admits a generalization to later columns.

Theorem 7.

$$(4)\quad F(n,m)=\sum_{\substack{n=a_1+a_2+\ldots+a_m\\ a_i\neq 0}}\sum_{j_1\mid a_1}\sum_{\substack{j_2\mid a_2\\ j_1\neq j_2}}\sum_{\substack{j_3\mid a_3\\ j_2\neq j_3}}\cdots\sum_{\substack{j_m\mid a_m\\ j_{m-1}\neq j_m}}\binom{j_1}{j_2}\binom{j_2}{j_3}\cdots\binom{j_{m-1}}{j_m}.$$

Proof. The outer summation is over compositions of n into m parts. First we build from compositions of n into m parts certain partitions of n into m distinct parts, then permute the parts of the partitions to obtain all possible compositions of n into m distinct parts.

Begin by representing the summands in a composition of n as a rectangular array of dots, one row for each summand. This resembles the Ferrars graph of a partition, except the lengths of the rows do not have to be monotone decreasing. Now attempt to transform this graph into the Ferrars graph of a partition of n with m distinct parts by replacing some of the rows of a_i dots with rectangles of width a_i/j_i dots and height j_i for j_i a divisor of a_i. The rows of the rectangle represent the size of the new parts, a_i/j_i, that replace the old a_i, and the number of parts goes from 1 part of size a_i in the composition to j_i parts of size a_i/j_i in the partition.

For a particular composition $n=a_1+a_2+\ldots a_m$, we obtain a partition of n into m distinct parts exactly when there is a sequence of divisors j_i of a_i with $j_1>j_2>j_3>\ldots>j_m$. Furthermore this process is reversible, with any partition of n into m distinct parts yielding a composition of n into m parts when the parts of

the same size in the partition are combined into one summand in the composition, and with the two partitions yielding the same composition only when rectangular blocks of different dimensions representing successive rows in the two partitions contain the same number of dots.

The inner summations in (4) generate all sequences of divisors $\{j_i\}$ that yield partitions of n into m distinct parts (the conditions that $j_i \neq j_{i+1}$ are sufficient, since if $j_i > j_{i+1}$ one of the binomial coefficients is zero). With j_i parts of size a_i/j_i, $1 \leq i \leq m$, the total number of compositions of n that can be formed by rearranging the summands is given by the multinomial coefficient

$$\binom{j_1}{j_1 - j_2, j_2 - j_3, \ldots, j_{m-1} - j_m, j_m} = \binom{j_1}{j_2}\binom{j_2}{j_3}\cdots\binom{j_{m-1}}{j_m}.$$

Compositions with a fixed part

In [7], Wilf outlines a general technique to obtain mean values for the number of distinct part sizes in a combinatorial structure. The success of his method depends on the multiplicativity of the generating function for the total number of structures of size n. Many common combinatorial structures have such generating functions, so that in addition to partitions his results apply equally well to permutations [7], partitions of sets [4], and polynomials over finite fields [3]. Part of the interest concerning the number of part sizes in a composition stems from the fact that the familiar generating function for compositions is not of the above type. Below we use a different technique based on a simple counting argument to obtain a generating function for the mean value.

Suppose we wish to guarantee that a particular part l (perhaps repeated) occurs in a composition of n. Denote by $C_l(n)$ the number of compositions of n in which at least one l occurs. It will also be necessary to keep track of $C_l(n, j)$, the number of compositions of n into j parts in all, in which at least one part is l. This notation extends the more standard use of $C(n)$ to represent the total number of compositions of n, which is 2^{n-1}, and $C(n, j)$ to represent the number of compositions with exactly j parts, $\binom{n-1}{j-1}$.

For a composition π of n, let $\delta(\pi)$ be the number of distinct parts of π, and let

$$\chi(r, \pi) = \begin{cases} 1 & , r \text{ is a part of } \pi \\ 0 & , \text{ otherwise.} \end{cases}$$

Then

$$\sum_{m=1}^{\infty} mF(n, m) = \sum_{\pi} \delta(\pi) = \sum_{\pi}\sum_{l \geq 1} \chi(l, \pi) = \sum_{l \geq 1}\sum_{\pi} \chi(l, \pi) = \sum_{l \geq 1} C_l(n)$$

Thus the numbers in row Σ in the table below represent $\sum_{l \geq 1} C_l(n)$. They are of special interest because of the above connection with the average number of distinct parts in a composition of n.

Table 3. *Compositions of n into parts in which at least one part is an l, $C_l(n)$, $1 \leq n \leq 12, 1 \leq l \leq 6$*

$l\backslash n$	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	3	6	13	27	56	115	235	478	969	1959
2		1	2	4	9	20	43	91	191	398	824	1697
3			1	2	5	11	25	55	120	258	550	1163
4				1	2	5	12	27	61	135	295	639
5					1	2	5	12	28	63	141	311
6						1	2	5	12	28	64	143
							$\ddots$					$\vdots$
Σ	1	2	6	13	30	66	144	308	655	1380	2891	6024

There is a general recurrence satisfied by the rows of Table 3, which we will recover from the generating functions established in the next theorem.

Theorem 8.

$$\sum_{n=1}^{\infty} C_l(n)t^n = \frac{t}{1-2t} - \frac{t-t^l+t^{l+1}}{1-2t+t^l-t^{l+1}}. \tag{5}$$

Proof. Write $C_l^*(n)$ to be the number of compositions of n with *no* part equal to l, and $C_l^*(n,m)$ to be the number of such compositions with m parts in all. Thus

$$C(n) - C_l^*(n) = C_l(n). \tag{6}$$

Observe that the generating function for $C_l^*(n,m)$ is

$$\sum_{n=0}^{\infty} C_l^*(n,m)t^n = (t+t^2+\ldots+t^{l-1}+t^{l+1}+\ldots)^m = (\frac{t}{1-t}-t^l)^m.$$

Now account for all possible numbers of parts m via

$$\sum_{n=0}^{\infty} C_l^*(n)t^n = \sum_{m=1}^{\infty}\left(\frac{t}{1-t}-t^l\right)^m = \frac{t-t^l+t^{l+1}}{1-2t+t^l-t^{l+1}}.$$

The last step is to recall that the generating function for arbitrary compositions is $\frac{t}{1-2t}$. Thus the generating function for the table entries in row l follows from (6).

Corollary 9. *For $l \geq 1, n \geq l+2$,*

$$C_l(n) = 2C_l(n-1) - C_l(n-l) + C_l(n-l-1) + 2^{n-l-2}.$$

Proof. From the generating function for $C_l^*(n)$ above we obtain

$$\begin{aligned} t-t^l+t^{l+1} &= (1-2t+t^l-t^{l+1})\sum_{n=0}^{\infty} C_l^*(n)t^n \\ &= \sum_{n=0}^{\infty} C_l^*(n)t^n - 2\sum_{n=0}^{\infty} C_l^*(n)t^{n+1} + \sum_{n=0}^{\infty} C_l^*(n)t^{n+l} - \sum_{n=0}^{\infty} C_l^*(n)t^{n+l+1} \\ &= \sum_{n=0}^{\infty} C_l^*(n)t^n - 2\sum_{n=1}^{\infty} C_l^*(n-1)t^n + \sum_{n=l}^{\infty} C_l^*(n-l)t^n - \sum_{n=l+1}^{\infty} C_l^*(n-l-1)t^n. \end{aligned}$$

Equating coefficients of t^n for $n \geq l+2$ gives

$$C_l^*(n) = 2C_l^*(n-1) - C_l^*(n-l) + C_l^*(n-l-1). \tag{7}$$

The last step is to note

$$\begin{aligned} C_l(n) &= 2^{n-1} - C_l^*(n) \\ &= 2^{n-1} - (2C_l^*(n-1) - C_l^*(n-l) + C_l^*(n-l-1)) \\ &= 2(2^{n-2} - C_l^*(n-1)) - (2^{n-l-1} - C_l^*(n-l)) \\ &\quad + (2^{n-l-2} - C_l^*(n-l-1)) + 2^{n-l-2} \\ &= 2C_l(n-1) - C_l(n-l) + C_l(n-l-1) + 2^{n-l-2}. \end{aligned}$$

The first row therefore satisfies the recurrence

$$C_1(n) = C_1(n-1) + C_1(n-2) + 2^{n-3},$$

valid for $n \geq 3$. From this we observe that $C_1(n) = 2^{n-1} - F_{n-1}$, where F_{n-1} is the $n-1$th Fibonacci number.

A combinatorial proof of (7) is of independent interest. Write

$$\begin{aligned} C_l^*(n) &= \#\ \{\text{compositions of } n \text{ with no part } l\} \\ &= \sum_{\substack{k=1 \\ k\neq l}}^{n} \#\{\text{compositions of } n \text{ with no part } l \text{ and first part } k\} \\ &= \sum_{\substack{k=1 \\ k\neq l}}^{n} C_l^*(n-k) \end{aligned}$$

Similarly

$$C_l^*(n-1) = \sum_{\substack{k=1 \\ k\neq l}}^{n-1} C_l^*(n-1-k)$$

Thus

$$\begin{aligned} C_l^*(n) &= C_l^*(n-1) + \sum_{\substack{k=2 \\ k\neq l}}^{n} C_l^*(n-k) \\ &= C_l^*(n-1) + \sum_{\substack{j=1 \\ j+1\neq l}} C_l^*(n-1-j) \\ &= 2C_l^*(n-1) - C_l^*(n-l) + C_l^*(n-l-1) \end{aligned}$$

Corollary 10. *The generating function for the last row of Table 3, labelled Σ, is*

$$\left(1 + \frac{t^2}{1-2t}\right) \sum_{l=1}^{\infty} \frac{t^l}{1-2t+t^l-t^{l+1}}. \tag{8}$$

Proof. Write the summands in

$$\sum_{l=1}^{\infty} \left(\frac{t}{1-2t} - \frac{t-t^l+t^{l+1}}{1-2t+t^l-t^{l+1}} \right)$$

over a common denominator and then remove the common factor.

Hwang and Yeh have derived from the generating function (8) the asymptotic mean value

$$\frac{\sum_{m=1}^{n} mF(n,m)}{2^{n-1}} = \log_2 n - \frac{3}{2} + \frac{\gamma}{\log 2} - \varpi(\log_2 n) + O(n^{-1}\log n),$$

where $\varpi(u)$ is a periodic function of small amplitude. This result and others are contained in [2].

It is interesting to observe that the first l (nonzero) terms of row l in Table 3 are also the first l terms of every subsequent row of the table. Thus there is a sequence beginning $1, 2, 5, 12, 28, \ldots$ which we will call an *envelope* for the rows of the table. We account for this series in the next result.

Corollary 11. *The envelope* $1, 2, 5, 12, 28, 64, 144, \ldots$ *of Table 3 has generating function* $\left(\dfrac{1-t}{1-2t}\right)^2$.

Proof. In the generating function in Corollary 10 the significance of the t^l factor is only that row l is offset. For the envelope, we first consider $1/(1-2t+t^l-t^{l+1})$. Note (by long division) that the series expansion of this function of t matches the series expansion of $1/(1-2t)$ for l terms. Thus the first l terms of

$$\left(1+\frac{t^2}{1-2t}\right)\frac{1}{1-2t+t^l-t^{l+1}}$$

are provided by the simpler function

$$\left(1+\frac{t^2}{1-2t}\right)\frac{1}{1-2t} = \left(\frac{1-t}{1-2t}\right)^2$$

From the generating function we deduce that the nth entry of the envelope sequence (numbered from $n=0$) equals $2^{n-2}(n+3)$ for $n \geq 1$.

By studying the generating function of Corollary 10 we can provide a family of "nested recurrences" for the numbers Σ.

Theorem 12. *Denote by* $S(n)$ *and* $D_i(n)$ *sequences defined for* $n \geq 1$ *by the initial conditions* $S(1)=1, S(2)=2$, *and for any* i $D_i(1)=D_i(2)=1$, *and by the recurrences*

$$S(n) = 2S(n-1) - D_1(n-1) + D_1(n-2) + (2^{n-3}+2^{n-4}+\ldots) + 1$$

$$D_1(n) = 2D_1(n-1) - D_2(n-1) + D_2(n-2) + (2^{n-3}+2^{n-5}+\ldots) + \begin{cases} 1, \text{ if } 2 \mid n-1 \\ 0, \text{ otherwise} \end{cases}$$

$$D_2(n) = 2D_2(n-1) - D_3(n-1) + D_3(n-2) + (2^{n-3}+2^{n-6}+\ldots) + \begin{cases} 1, \text{ if } 3 \mid n-1 \\ 0, \text{ otherwise} \end{cases}$$

$$D_3(n) = 2D_3(n-1) - D_4(n-1) + D_4(n-2) + (2^{n-3}+2^{n-7}+\ldots) + \begin{cases} 1, \text{ if } 4 \mid n-1 \\ 0, \text{ otherwise} \end{cases}$$

$$\vdots$$

Then $S(n)$ is the nth entry of row Σ of Table 3. The sums of powers of 2 include terms as long as the exponents remain non-negative.

Proof. The proof will be by induction on k, the subscript of D. Summing the entries in column n, we apply the recurrence of Corollary 9 summand by summand to obtain

$$S(n) = 2S(n-1) - \sum_{l=1}^{\infty} C_l(n-l) + \sum_{l=1}^{\infty} C_l(n-l-1) + \sum_{l=1}^{n-2} 2^l.$$

We denote $\sum_{l=1}^{\infty} C_l(n-l)$ by $D_1(n)$, where D is chosen mnemonically to represent a sum over a diagonal. The diagonal sum $D_1(n)$ steps through Table 3, repeatedly moving down one entry and to the left one entry from entry $C_1(n)$ in the first row. There is an extra summand of 1 because $C_n(n) = 1$ did not arise from a recurrence but from an initial condition. This explains the first recurrence.

Now consider the recurrence appropriate for D_{k+1}. Recurrence (8) applies to the summands of D_k as well, which arose by stepping down one entry and to the left k entries from entry $C_1(n)$ in the first row. Recurrence (8) applied to the summands of D_k gives a shallower diagonal, stepping down one entry and to the left $k+1$ entries, with an extra summand of 1 arising for $k+1 \mid n-1$ because in every $k+1$st shallow diagonal the initial condition $C_{\lfloor n/(k+1)\rfloor}(n) = 1$ gives a term that does not arise in any recurrence.

The partition $\lambda_1 1 + \lambda_2 2 + \ldots + \lambda_n n = n$ (with λ_i occurrences of i) can be ordered in $(\lambda_1 + \lambda_2 + \ldots + \lambda_n)!/(\lambda_1!\lambda_2! \ldots \lambda_n!)$ ways. If we count compositions of n with j parts by taking ordered arrangements over partitions of n with j parts we get the well known identity

$$\binom{n-1}{j-1} = \sum_{\substack{\lambda_1 1+\lambda_2 2+\ldots+\lambda_n n=n \\ \lambda_1+\lambda_2+\ldots+\lambda_n=j}} \frac{j!}{\lambda_1!\lambda_2! \ldots \lambda_n!}.$$

Similarly we can count compositions of n with j parts of m different sizes by taking ordered arrangements of partitions of n into j parts with m different sizes. The result is expressed in the following proposition.

Proposition 13.

$$F(n,m,j) = \sum_{\substack{\lambda_1 1+\lambda_2 2+\ldots+\lambda_n n=n \\ \lambda_1+\lambda_2+\ldots+\lambda_n=j \\ \#\{i:\lambda_i>0\}=m}} \frac{j!}{\lambda_1!\lambda_2! \ldots \lambda_n!}.$$

As this proposition shows, there is a close connection between partition identities and composition identities. Now we go the other way, and note some results for partitions analogous to the composition results we have derived.

Partitions with m distinct parts

Since partitions may be regarded as compositions with decreasing part size, and compositions may be generated from partitions by permuting the parts, it is not surprising that many of the formulas generated above have analogues for partition counting functions. We begin by recasting Table 1 as a table about partitions.

Table 4. *Partitions of n with m distinct parts, $G(n,m)$, $1 \le n \le 16$, $1 \le m \le 5$*

$n\backslash m$	1	2	3	4	5
1	1				
2	2				
3	2	1			
4	3	2			
5	2	5			
6	4	6	1		
7	2	11	2		
8	4	13	5		
9	3	17	10		
10	4	22	15	1	
11	2	27	25	2	
12	6	29	37	5	
13	2	37	52	10	
14	4	44	67	20	
15	4	44	97	30	1
16	5	55	117	52	2

The first column of Table 4 is again $d(n)$, the divisor counting function. The sums over divisors of the first two theorems have the following versions for partitions, obtained by counting partitions according to occurrences of the distinct part that occurs least often.

Theorem 14. *For $n \ge 2$,*

$$G(n,2) = \sum_{j=2}^{\lfloor n/3 \rfloor} \sum_{k=1}^{\lfloor n/j \rfloor - 1} \sum_{\substack{d \mid (n-jk) \\ k \ne d < \frac{n-jk}{j-1}}} 1 - \sum_{\substack{j=1 \\ j \mid n}}^{\lfloor n/3 \rfloor} \left\lfloor \frac{n/j - 1}{2} \right\rfloor + \sum_{k=1}^{n-1} \sum_{\substack{d \mid (n-k) \\ k \ne d}} 1.$$

Theorem 15. *For $j \ge 2$ and $n > j$,*

$$G(n,2,j) = \sum_{r=1}^{\lfloor j/2 \rfloor} \sum_{\substack{k=1 \\ (j-r) \mid (n-kr) \\ k \ne (n-kr)/(j-r)}}^{\lfloor n/r \rfloor - 1} 1 - \begin{cases} \left\lfloor (\frac{2n}{j} - 1)/2 \right\rfloor, & \text{if } 2 \mid j \text{ and } j \mid 2n \\ 0, & \text{otherwise.} \end{cases}$$

Formula (4) in Theorem 7 has a version for partitions whose proof is immediate: From a composition of n into m parts one seeks the Ferrars graph of a partition into m distinct parts by replacing summands a_i with groups of j_i summands each of size a_i/j_i. All that is different is that, when an acceptable Ferrars graph is found, the partition counts once, instead of having its parts permuted to generate a family of compositions. What results is

Theorem 16.

$$G(n,m) = \sum_{\substack{n = a_1 + a_2 + \ldots + a_m \\ a_i \ne 0}} \sum_{j_1 \mid a_1} \sum_{\substack{j_2 \mid a_2 \\ j_1 > j_2}} \sum_{\substack{j_3 \mid a_3 \\ j_2 > j_3}} \cdots \sum_{\substack{j_m \mid a_m \\ j_{m-1} > j_m}} 1.$$

In structure Table 4 resembles Table 1, but it has an envelope for its columns reminiscent of the envelope for the rows of Table 3.

Theorem 17. *The first $m+1$ non-zero entries in column m of Table 4 are the first $m+1$ non-zero entries of all subsequent columns. The envelope,*

$$1, 2, 5, 10, 20, 36, \ldots,$$

has generating function

$$\prod_{i=1}^{\infty}(1-t^i)^{-2}.$$

Proof. The first non-zero entry in column m occurs at the smallest n for which it is possible to have m distinct summands in a partition of n, which is at T_m, the mth triangular number. The farthest term in the envelope occurring in column m counts partitions of $T_m + m = T_{m+1} - 1$ into m distinct parts.

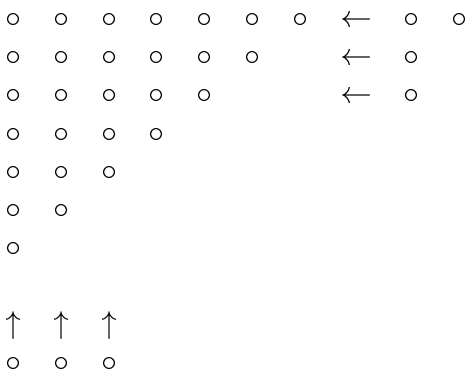
Consider the triangle of dots that is the Ferrars graph of the partition $1 + 2 + \ldots + m$, and let $1 \leq a \leq m$ be given. Any partition of a can be represented as a Ferrars graph on its own, and appended to the triangular Ferrars graph by either of two different methods. One method is to adjoin dots representing the successive summands in the partition row by row to the rows of the triangular Ferrars graph, top to bottom. This results in a partition of $T_m + a$ into m parts in all, all of them distinct. Another method is to adjoin dots representing the successive summands in the partition column by column to the columns of the triangular Ferrars graph, left to right. This results in a partition of $T_m + a$ in which the largest part is m, in which there are more than m parts in all, but in which there are only m distinct parts.

For any representation of $m = a + b$, any of the $p(a)$ partitions of a may be appended to the triangular Ferrars graph of $1 + 2 + \ldots + m$ by the first method, and any of the $p(b)$ partitions of b may be appended by the second method. This results in a Ferrars graph for a partition of $T_m + m$ into exactly m distinct parts. Furthermore the process is reversible. Thus in the Ferrars graph of any partition of $T_m + m$ into m distinct parts, it is possible to strip off the first m dots in the first row, the first $m-1$ dots in the second, ..., the first dot in the mth row. This leaves m dots, in clusters of dots in the upper right and/or lower left, that can be interpreted as partitions of a (upper right) and b (lower left).

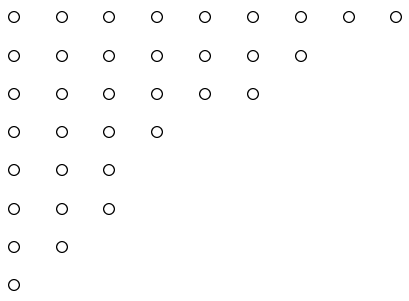
Overall the number of partitions of $T_m + m$ into m distinct parts is thus given by $\sum_{m=a+b} p(a)p(b)$. This sum allows a or b to be 0. This is the coefficient of t^m in the series expansion of $(1+p(1)t+p(2)t^2+\ldots)^2$, and hence the generating function is the square of the generating function for unrestricted partitions:

$$\prod_{i=1}^{\infty}(1-t^i)^{-2}.$$

Figure 1. *The Ferrars graph construction of Theorem 16 for $m = 7$, $a = 4 = 2+1+1$, and $b = 3 = 1+1+1$, yielding the partition of $T_7 + 7 = 35 = 9+7+6+4+3+3+2+1$ with 7 distinct parts.*



After



The combinatorial approach of Theorem 17 also explains the values of the next term beyond the first $m+1$ terms of the envelope, $G(m(m+1)/2+m+1,m)$, because in this case when the partitions of a and b consist of all 1's, the appended partitions span the $t+1$st diagonal to give a Ferrars graph of the partition of T_{m+1} into $m+1$ distinct parts. Excluding these $m+2$ cases gives the correct value of $G(m(m+1)/2+m+1,m)$.

Given the simplicity of the generating function, it is not surprising that the envelope has arisen in many enumeration problems. See, for example, [1, p. 90] in connection with partitions into parts of two kinds.

In analogy with Lemma 3 we have

Lemma 18. *Denote by $G(n,m,j)$ the number of partitions of n with exactly m distinct parts and j parts in all, and by $G^*(n,m,j)$ the number of compositions of n with exactly m distinct parts, j parts in all, and at least one part being a 1. Then*

$$G(n,m,j) = G(n-j,m,j) + G^*(n,m,j).$$

The parallel development continues.

Proposition 19.

$$G(n,m,j) = G(n-j,m,j) + \sum_k{}^* G(n-j,m-1,j-k),$$

where $\sum^*$ indicates a sum over those k for which a partition of n into m distinct parts, j parts in all, can have exactly k 1's.

The same results about the summands of Σ^* earlier, in Proposition 5 and the preceding remarks, apply here as well.

Corollary 20.

$$G(n,2,j) = \sum_{l=1}^{\lfloor \frac{n-1}{j} \rfloor} \sum_{\substack{k \\ (j-k)\mid(n-lj)}} 1.$$

In conclusion we mention that the mean value for the number of distinct parts in a partition of n was obtained by Wilf [7]. He showed that

$$\frac{\sum_{m=1}^{n} mG(n,m)}{p(n)} = \sum_{i=0}^{n-1} p(i)/p(n) \sim \frac{\sqrt{6}}{\pi}\sqrt{n},$$

as $n \to \infty$.

References

1. Hansraj Gupta, *Royal Society Mathematical Tables Volume 4, Tables of Partitions*, Cambridge University Press, 1958.
2. H.-K. Hwang and Y.-N. Yeh, *Measures of distinctness of summands in random partitions and compositions*, (preprint).
3. A. Knopfmacher, *On the number of distinct degree sizes of a polynomial over a finite field*, (preprint).
4. A. M. Odlyzko and L. B. Richmond, *On the number of distinct block sizes in partitions of a set*, Journal of Combinatorial Theory, Series A **38** (1985), 170–181.
5. B. Richmond and A. Knopfmacher, *Compositions with distinct parts*, Aequationes Mathematicae **49** (1995), 86–97.
6. N. J. A. Sloane and S. Plouffe, *Encyclopedia of Integer Sequences*, Academic Press, 1995.
7. Herbert S. Wilf, *Three problems in combinatorial asymptotics*, Journal of Combinatorial Theory, Series A **35** (1983), 199–207.

Department of Computational and Applied Mathematics
Wits 2050, Johannesburg, South Africa

Department of Mathematics
West Virginia University, Morgantown WV 26506-6310
E-mail address: arnoldk@gauss.cam.wits.ac.za, mays@math.wvu.edu

Linköping Electronic Articles in
Computer and Information Science
Vol. 5(2000): nr 38

Automated Theory Formation Applied to Four Learning Tasks

Simon Colton

Linköping University Electronic Press
Linköping, Sweden

http://www.ep.liu.se/ea/cis/2000/038/

Published on December 21, 2000 by
Linköping University Electronic Press
581 83 Linköping, Sweden

Linköping Electronic Articles in
Computer and Information Science
ISSN 1401-9841
Series editor: Erik Sandewall

Typeset by the author using LaTeX
Formatted using étendu *style*

Recommended citation:
<Author>. <Title>. Linköping Electronic Articles in Computer and Information Science, Vol. 5(2000): nr 38. http://www.ep.liu.se/ea/cis/2000/038/*. December 21, 2000.*

This URL will also contain a link to the author's home page.

Abstract

Automated theory formation involves, amongst other things, the production of examples, concepts and statements relating the concepts. The HR program [5] has been developed to form theories in mathematical domains, by calculating examples, inventing concepts, making conjectures, and settling conjectures using the Otter theorem prover [13] and MACE model generator [14].

In addition to providing a plausible model for automated theory formation in pure mathematics, HR has been applied to other problems in Artificial Intelligence. We discuss HR's application to inducing definitions from examples, scientific discovery, problem solving and puzzle generation. For each problem, we look at how a theory formation approach can be applied and mention some initial results from the application of HR. Our aim is not to describe the applications in great detail, but rather to provide an overview of how HR is used for these problems. This will facilitate a comparison of the problems and discussion of the effectiveness of theory formation for these tasks.

Our second aim is to compare HR with the Progol machine learning program [17]. We do this first by looking at the concept formation these programs perform. Also, by suggesting how Progol could be used for the applications mentioned above, we compare the programs in terms of how they can be applied.

Author's address

Division of Informatics
University of Edinburgh
Edinburgh EH1 1HN
United Kingdom

E-mail: `simonco@dai.ed.ac.uk`

1 Introduction

A theory often discusses objects of a particular nature. For example, in pure mathematics, number theory is about integers, whereas graph theory concerns graphs and group theory concerns groups. Similarly, in non-mathematical domains there are objects of interest around which a theory forms, for example acids in chemistry, sub-atomic particles in physics and so on. Theories typically contain (i) examples of the objects of interest, (ii) concepts which discuss the nature of those examples and (iii) statements highlighting relationships between concepts. For example, in finite group theory, there are 14 groups up to isomorphism with 8 or fewer elements. There are also many concepts describing these groups, for example cyclic groups are a particular type of group and the centre of a group is a subset of elements of the group. Group theory also contains many statements relating two or more concepts, for instance if a group is cyclic, then the centre of the group will contain all the elements, i.e. it will be Abelian. Similarly, in chemistry, there are examples of acids, such as hydrochloric and there are specialisations of the concept of acids, for instance organic and inorganic acids. There are also statements about acids, such as: adding an acid to a base will produce a salt and water.

In mathematics, the statements are often *proved* via a sequence of logical inferences. The statements are usually called conjectures until they are proved, when they become theorems. Theories will contain proofs, disproofs and counterexamples as well as open conjectures for which the truth is unknown. In non-mathematical domains it is often possible to formalise the statements and appeal to mathematical proofs. However, sometimes the plausibility of a statement has to be *demonstrated* with experiments and *explained* via more theory formation. For instance, experiments where acids and bases are mixed add plausibility to the above statement, because a salt solution is usually observed. To explain this phenomena, chemists may provide a reaction mechanism to show how the bonds in the chemicals break and re-form during the reaction.

Given this initial synopsis of what theories contain, automated theory formation should be able to at least find examples of the objects of interest, invent new concepts and make plausible statements relating those concepts. In mathematics, theory formation should also involve proving and disproving conjectures. There have been many automatic approaches to these individual tasks. For instance, the Progol program [17] can invent new concepts and the MECHEM program [23] can find reaction pathways in chemistry. Similarly in mathematics, the Mathematica program, [24] can perform calculations and symbolic manipulations, the AGX and Graffiti programs [1], [10] can make conjectures, the Otter program [13] can prove conjectures and the MACE program [14] can find counterexamples.

There have only been a few attempts to automate theory formation as a whole. The AM program [12] was the first to explore mathematical domains using concept formation and conjecture making. The GT program [9] automated more mathematical activities by enabling example generation and theorem proving as well as concept formation and conjecture making. The HR program [5] performs automated theory formation in domains of pure mathematics. Using all of its functionality, HR can start with just the axioms of a finite algebra such as group theory. It will then find examples, invent concepts, make conjectures, prove theorems and find counterexamples to false conjectures. HR can also work in number theory and graph theory and we intend to use HR in more mathematical domains.

As well as providing a plausible model for theory formation, HR has been applied to other problems in Artificial Intelligence. In §3 we discuss four such problems, namely the induction of definitions from examples, scientific discovery, problem solving and puzzle generation. We do not aim to give a complete description of the application of HR to these problems but rather to give an overview of our approach using HR. We also suggest how the Inductive Logic Programming (ILP) program Progol [17] could be used for these tasks and we compare HR and Progol in terms of their application. However, we begin in §2 by comparing HR and Progol in terms of the concepts they form.

2 The HR and Progol Programs

2.1 The HR Program

The HR program [5], named after mathematicians Hardy and Ramanujan, is designed to form theories in domains of mathematics such as group theory, graph theory and number theory. HR starts with background information such as the axioms of a finite algebra, or some concepts in number theory such as the divisors of integers, multiplication and addition. Each concept is supplied with a definition and the user can also supply a finite set of examples, although this is not necessary in algebraic domains, as examples can be generated from the axioms. HR uses one of seven general production rules to base a new concept on either one old concept (in which case we say the production rule is *unary*) or two old concepts (a *binary* production rule). This produces a set of concepts which form the core of the theory.

Each production rule generates a definition and a set of examples for the new concept and table 1 describes the action of each production rule. For example, starting with the concept of divisors of integers in number theory, figure 1 shows how HR constructs the concept of prime numbers. This concept is produced using the size production rule to count the number of subobjects (divisors) followed by the split rule to instantiate this number to 2. This extracts those numbers with exactly two divisors — prime numbers. For a more detailed description of the production rules, see [7].

Rule	Action of Production Rule
Compose	Composes predicates by conjunction
Exists	Introduces existential quantification
Forall	Introduces universal quantification
Match	Equates variables in predicates
Negate	Finds compliments to predicates (negating the property)
Size	Counts the number of subobjects satisfying a predicate
Split	Instantiates variables

Table 1: The action of HR's seven production rules

It is important to note that a concept has (i) a set of examples, (ii) a definition, (iii) a categorisation over the examples HR has available and (iv) a set of conjectures involving the concept. For instance, if HR is working with the integers 1 to 10 in number theory, then the concept of prime numbers will have these examples: $\{2, 3, 5, 7\}$ and the definition given in figure 1. We call this a *specialisation* concept because it produces a binary

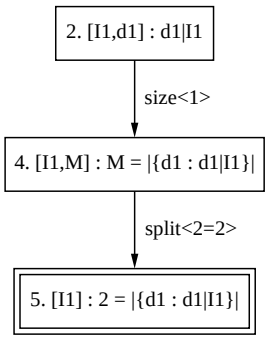


Figure 1: Construction of the concept of prime numbers

categorisation of the integers which specialises the concept of integer into prime and non-prime integers thus:

$$[1, 4, 6, 8, 9, 10], [2, 3, 5, 7]$$

In the theory HR produces, there will also be a set of conjectures about prime numbers, for example that prime numbers are never perfect squares. While producing concepts, HR makes these conjectures using empirical evidence. In particular, if it notices that the examples of a new concept are exactly the same as an old concept (for the data available), it will conjecture that the definitions of the two concepts are logically equivalent — producing an 'if and only if' conjecture. Similarly, if it notices that the examples of one concept are all examples of another concept, it will make an implication conjecture. If it cannot find any examples for a concept, it will make a non-existence conjecture (i.e. that there are no examples whatsoever). In finite algebras, HR invokes the Otter theorem prover [13] to prove the conjectures it makes. Whenever Otter is unsuccessful, HR uses the MACE model generator [14] to find a counterexample to disprove the conjecture. In this way, HR forms a theory which contains concepts, examples, open conjectures, theorems and proofs.

To improve the quality of the theories, HR uses heuristic measures to estimate the worth of concepts and performs a best first search by using the more interesting concepts as the basis for new concepts before the less interesting ones. The user sets weights for a weighted sum of all the measures which is taken as an estimate of the worth of each concept. The measures include intrinsic properties of the concept such as the number of examples it has, as well as relational measures such as the novelty of the categorisation it produces, as discussed in [5]. The quantity and quality of conjectures that a concept appears in is also assessed, with concepts appearing in interesting conjectures assessed as more interesting than those appearing in dull conjectures. The worth of a theorem is assessed by the length of the proof produced by Otter, with longer proofs indicating a more interesting conjecture statement. HR therefore completes a cycle of mathematical activity where concept formation drives conjecture making and theorem proving which in turn improves concept formation. HR improves on previous theory formation programs such as AM [12] and GT [9] by incorporating theorem proving (AM could not prove theorems) and by being able to work in many domains (GT could only work in graph theory).

```
% Mode Declarations
:- modeh(1,square(+intgr))?
:- modeb(1,multiply(+intgr,-intgr,-intgr))?

% Background Knowledge
intgr(1).intgr(2).intgr(3).intgr(4).intgr(5).
intgr(6).intgr(7).intgr(8).intgr(9).intgr(10).
multiply(A,B,C) :- intgr(B), intgr(C), A is B*C.

% Positive Examples
square(1).square(4).square(9).

% Negative Examples
:- square(2). :- square(3). :- square(5).
:- square(6). :- square(7). :- square(8). :- square(10).
```

Figure 2: Input to Progol for learning the concept of square numbers

2.2 The Progol Program

Inductive Logic Programming (ILP) is a general purpose machine learning technique [16]. Concepts are represented as first order logic programs, which has many advantages, including that they can be interpreted by an underlying logic programming language. The Progol program [17] uses ILP with an underlying Prolog interpreter. Progol is usually employed to produce a logic program which defines a set of given positive examples but not the given negative examples. The definitions are based on background predicates supplied by the user.

As an example, Progol can learn the concept of square numbers, given the background knowledge and positive and negative examples in figure 2. Progol produces this answer:

```
square(A) :- multiply(A,B,B).
```

This is a Prolog program which will identify a square number as being the multiplication of some number with itself. The mode declarations at the top of the input in figure 2 determine the format for the logic program to be learned, with `+` indicating the use of a known variable, `-` indicating the introduction of a new variable and `#` indicating possible instantiation. Progol searches for concepts using the U-Learnability framework [19]. In this framework, there is a prior probability distribution over the space of concepts, with the probability being the likelihood that the concept is the required one.

The construction of new concepts is achieved by inverting deductive rules of inference to produce inductive rules. One rule of deduction which is inverted is the resolution rule [20]. In its simplest form, this states that if we know:

$$A \rightarrow B \text{ and } B \rightarrow C$$

then we can deduce that:

$$A \rightarrow C$$

The first two ways to invert resolution involve inverting a single resolution step. In effect, this amounts to asking the question: 'given the observed

clauses [logic programs] in the data, what two clauses could have been resolved together to give this observation?' The absorption and identification inductive rules of inference are obtained in this way:

$$\textbf{Absorption:} \quad \frac{q \leftarrow A \qquad p \leftarrow A, B}{q \leftarrow A \qquad p \leftarrow q, B}$$

$$\textbf{Identification:} \quad \frac{p \leftarrow A, B \qquad p \leftarrow A, q}{q \leftarrow B \qquad p \leftarrow A, q}$$

The absorption rule can be read as: 'Given that I observe $q \leftarrow A$ and $p \leftarrow A, B$, one hypothesis I can make is that this is because $q \leftarrow A$ and $p \leftarrow q, B$ are true and have been resolved to produce the observations. By interpreting this hypothesis as a logic program, the feasibility of it being true can be checked against the data.

Two more induction rules are derived from inverting 2 resolution steps:

$$\textbf{Intra-Construction:} \quad \frac{p \leftarrow A, B \qquad\qquad p \leftarrow A, C}{q \leftarrow B \qquad p \leftarrow A, q \qquad q \leftarrow C}$$

$$\textbf{Inter-Construction:} \quad \frac{p \leftarrow A, B \qquad\qquad q \leftarrow A, C}{p \leftarrow r, B \qquad r \leftarrow A \qquad q \leftarrow r, C}$$

With intra-construction, the hypothesis produced states that clauses $q \leftarrow B$ and $p \leftarrow A, q$ are true and were resolved to give the observed $p \leftarrow A, B$ and clauses $p \leftarrow A, q$ and $q \leftarrow C$ were resolved to give the observed $p \leftarrow A, C$. A new predicate symbol, q, has been introduced and likewise the predicate r is introduced in the inter-construction rule. This phenomena is called **predicate invention** and is often necessary to enable ILP programs to learn the correct definition for a concept. For example, when constructing a logic program for 'insertion sort', intra-construction is required to introduce an 'insert' predicate [18].

2.3 Concept Formation in HR and Progol

There is a striking similarity between the concepts Progol and HR can form. We highlight this using examples from number theory. Firstly, in Progol, concepts are formed which have definitions with conjunctions of predicates and the predicates may have variables repeated within them and between them. This produces concepts that HR can form with its compose, match and exists production rules. For example, given the background concepts of integers and multiplication, HR produces this definition for square numbers:

$$[n] \ : \ \exists \ a \ \ (a \times a = n)$$

and Progol produces this definition:

```
square(A) :- multiply(A,B,B).
```

Secondly, in Progol, the user can set mode declarations describing where background predicates can appear in the invented predicates. Mode declarations also specify whether variables become instantiated and whether negation of predicates is allowed. The ability to instantiate variables corresponds exactly with HR's split production rule, and the ability to negate predicates corresponds with the negate rule. Also, a combination of negated

and existentially quantified predicates corresponds to concepts produced by HR's forall production rule. For example, HR defines even numbers as:

$$[n] \ : \ 2|n$$

Similarly, given the background predicate of divisors and allowed to instantiate variables, Progol produces this definition:

```
even(N) :- divisor(N,2).
```

Finally, we found that if we supply two additional predicates as background knowledge from set theory, namely the standard Prolog predicates of `setof` and `length`, Progol can cover concepts produced by the size production rule. For example, HR defines the τ function (number of divisors of an integer) in this way:

$$[n,t] \ : \ t = |\{a : a|n\}|$$

and Progol produces this equivalent definition:

```
tau(N,T) :- setof(M,divisor(N,M),L),
            length(L,T).
```

Therefore, for each of HR's production rules, Progol can produce concepts of a similar nature. Interestingly, to cover all the production rules requires three different aspects of Progol's functionality, yet only one production rule corresponds to additional background knowledge. Progol has greater coverage of concepts than HR. In particular, Progol can define concepts recursively by specifying a base case and a step case. HR cannot yet produce such concepts, although we plan to implement another production rule to enable this.

3 Applications of Theory Formation

3.1 Inducing Definitions from Examples

The problem of inducing a definition for a concept given some positive examples of the concept and possibly some negative examples is well known in machine learning, and we have explored the possibility of using HR in this fashion. We have used HR to learn definitions for integer sequences, as discussed in [7] and have also applied HR to Michalski-style train problems [15] where the program is asked to find a reason why a certain subset of trains are going east, based on certain characteristics of the train, for example the shape of the carriages.

A naive way to use theory formation for learning tasks is to supply HR with background knowledge and ask it to form a theory, stopping when it has found a concept which matches the data supplied. To focus theory formation, we adapted HR's heuristic search to favour building on concepts which achieved a categorisation closer to the one achieved by the target concept. We found that this approach often failed to learn integer sequences because there was no discernible gradient for the measures HR uses, and so hill climbing was not possible (see [7] for further details).

Instead of the heuristic search, we used a 'unary first' search enhanced with a look ahead mechanism. A *unary first* search is a combination of a depth first and breadth first search: the unary production rules are used

exhaustively for each new concept before returning to the binary production rules with old concepts. In this way, each new concept receives some preliminary development, but is not combined with previous concepts until later. As an example of the look-ahead mechanism, given the sequence $2, 3, 5, 7$ (prime numbers) we have enabled HR to notice that, when it forms the concept of number of divisors, these numbers all have two divisors, a fact which is true of none of the other integers in HR's dataset. Each production rule has an algorithm for noticing a pattern which is true only for the positive examples, and when this happens a suitable theory formation step involving that production rule is added to the top of the agenda. Execution of the step produces a concept which fits the data. The pattern-spotting mechanism is faster than actually performing a theory formation step because there are overheads involved in performing a theory formation step and for the majority of the time, it is possible to quickly tell that there is no patter.

The look ahead mechanism has been successful with both problems about trains and integer sequences, and we supply some results in [7]. It is particularly effective when the concept to be learned is a combination of two old concepts, e.g. the concept of odd prime numbers, which combines the concepts of odd numbers and prime numbers. Depth first, breadth first and unary first searches do not find this concept quickly without the look ahead mechanism. However, with the look ahead mechanism, odd numbers are invented and as soon as prime numbers are introduced, HR notices that the positive examples are both odd and prime (and the negative examples are not). HR then combines these concepts and reaches the solution much faster — the time taken to learn the concept reduces from 384 to just 5 seconds. For more information on HR's application to learning tasks, please see [7].

The Progol program has been specifically designed to perform such induction tasks and has had much success. No further description of how it operates is required.

3.2 Scientific Discovery

In less than an hour, HR can produce more than 2000 concepts in number theory. Hence there is the possibility of HR producing new and interesting concepts, but it is difficult to tell in general whether a concept is either new or interesting. In number theory, however, there is an Encyclopedia of Integer Sequences [22] which contains around 60,000 sequences collected over 35 years by Neil Sloane, with contributions from many mathematicians. If a concept HR produces in number theory can be interpreted as an integer sequence which is missing from the Encyclopedia, this gives some indication — but by no means a guarantee — that the concept may be novel.

We have also used the Encyclopedia to give some indication as to whether the new integer sequences HR produces are interesting. To do this for a chosen sequence S, we have enabled HR to find sequences in the Encyclopedia which are empirically related to S, with the relations interpreted as conjectures about S. As a trivial example, given the sequence of prime numbers, HR makes the conjecture that they are never square numbers. It does this by noticing that none of the prime numbers it has are in the Encyclopedia entry for square numbers. As well as finding disjoint sequences, HR is able to find subsequences and supersequences of the chosen sequences.

Due to the large number of sequences in the Encyclopedia, many sequences related to the chosen one are output and we implemented pruning techniques to discard dull results. For example, it is desirable that a se-

quence conjectured to be disjoint with the chosen sequence has its terms distributed over roughly the same part of the number line as the chosen sequence. If so, the two sequences occupy roughly the same part of the number line yet do not share any terms — which increases the possibility of the conjecture being true and/or interesting. Therefore, HR discards conjectures about disjoint sequences if the overlap of their ranges falls below a minimum specified by the user.

By finding conjectures relating the sequence HR has invented to the sequences already in the Encyclopedia, HR provides some evidence that the sequence is of interest. This 'invent and investigate' approach has successfully led to 20 sequences invented by HR being added to the Encyclopedia, all supplied with interesting conjectures. A good example of this is the sequence of integers where the number of divisors is prime, which HR invented (in as much as it was produced by HR and not present in the Encyclopedia). When asked to find subsequences of this sequence, the first answer produced was the sequence of integers where the *sum* of divisors is prime (submitted to the Encyclopedia by someone else). Interpreted as a conjecture, this result states that, given an integer, n, if the sum of divisors of n is prime, then the number of divisors of n will also be prime. We have subsequently proved this result, and while we do not know for certain whether it is new, it certainly adds interest to the sequence HR invented. For more information on the application of HR to the invention of integer sequences, see [2] or [8].

While HR has produced many new sequences using the invent and investigate approach, it has also produced a new sequence by finding a definition for a given sequence. That is, we determined that the Encyclopedia of Integer Sequences contained a sequence starting a, b, c, d for all a, b, c, d such that $0 < a < b < c < d < 10$ with two exceptions. There was no sequence starting $4, 5, 6, 9$ and no sequence starting $4, 5, 7, 9$. We set HR the task of inventing sequences starting with these terms. In the latter case, within seconds, HR identified that the concept of prime numbers + 2 fitted the examples and this sequence is now in the Encyclopedia. While HR found a solution for the first sequence, the definition was fairly complicated (see [7]), and so we have not submitted it to the Encyclopedia.

In general, Progol has also been used to perform scientific discovery tasks by identifying a definition for a concept for which the categorisation of the examples into positives and negatives was already known. For instance, when applied to data from experiments involving the inhibition of E. Coli Dihydrofolate Reductase [11], the positive examples of the concept were pairs of drugs d_1 and d_2, where d_1 was known to be more effective at the inhibition task than d_2. The task was then to learn a definition for this concept, in effect to find a rule describing why d_2 was less effective. Within the rule derived for the concept, there may be new concepts, but the emphasis is on finding a definition for a known concept. To our knowledge, Progol has not been used in a way similar to HR above, where an entirely new concept and/or statement was identified and shown to be interesting.

3.3 Creative Problem Solving

In his book on mathematical problem solving [25] Paul Zeitz suggests a 'plug-and-chug' method, whereby calculations are performed and the results analysed to see if a pattern emerges which might provide insight into the problem. Zeitz supplies the following problem — taken from a 1930s Hungarian mathematics contest — as an example where this approach leads to the solution:

Show that the product of four consecutive integers is never a square number.

Following the plug and chug method, Zeitz calculates examples of the product of four consecutive integers:

$$1 \times 2 \times 3 \times 4 = 24 \text{ and } 2 \times 3 \times 4 \times 5 = 120$$

The sequence of calculations continues: 24, 120, 360, 840 and a eureka moment occurs with the realisation that these are all one less than a square. Zeitz then makes the conjecture that all such numbers are one less than a square and hence not square numbers. Zeitz states that:

> 'Getting to the conjecture was the crux move. At this point the problem metamorphosed into an exercise!'

To finish the problem, it is necessary to show that the product of four consecutive integers can be written as a square minus 1:

$$n(n+1)(n+2)(n+3) = (n^2+3n+1)^2 - 1.$$

We have applied HR to plug-and-chug problems of this nature, by getting it to make suggestions which might lead to a eureka moment for the user. To do this, HR is given a set of numbers which are related to the problem and asked to suggest properties of the numbers in the hope that one of the suggestions will provide an insight. To do this, for every new concept HR introduces, if all the given numbers have the property prescribed by the concept, then the definition is output. For example, when used for the Hungarian contest problem above, HR is given the numbers 24, 120, 360 and 840. As it forms a theory, it invents types of number and when the numbers 24, 120, 360 and 840 all satisfy the definition of a particular number type, the definition is output. Of course, some suggestions do not provide insight (for example that they are all even numbers). However, HR eventually invents the concept of squares-minus-one and so finds the conjecture which metamorphosed the problem.

The application of HR to problem solving is very recent and we are still experimenting and compiling a corpus of problems where the plug-and-chug approach would help. We hope to attach this functionality to a computer algebra system such as Maple or Mathematica. For more information on the application of HR to problem solving, see [4].

We have not applied Progol to this type of problem, so we can only speculate on how to do this. The problem here is not to learn a definition for a given concept, but rather to learn a *property* of a given concept. In machine learning terminology, the given concept can be thought of as a cluster and this problem is to find a larger cluster containing the given one. With HR, we chose to do this by finding new concepts which were generalisations of the given concept. One way to do this with Progol would be to include some negative examples along with the positive examples and attempt to learn a definition for this concept, which would be a generalisation of the one given. Deciding which negative examples to include would possibly be problematic and systematically choosing them may be too time consuming.

3.4 Puzzle Generation

Theorem proving has attracted much more attention than conjecture making in automated mathematics and similarly, the problem of finding solutions to puzzles [21] has been much more researched than the question of

generating interesting puzzles. We are interested here in one particular type of puzzle, namely odd one out puzzles. Such puzzles ask the problem solver (assumed to be a human from here on) to choose one object out of a set of similar objects and give a reason for the choice. The reason must be in terms of a property which the others share but which is not true of the object they have chosen, hence it is the odd one out.

We formalise the problem of generating odd one out puzzles in the following way: a puzzle is a set of n objects taken from a (possibly infinite) set of examples supplied by the user and a specialisation concept which categorises them into $n-1$ positive examples and 1 negative example. The negative example is the odd one out in the solution and the concept producing the categorisation provides the reason why it is the odd one out. We will concentrate here on the case where $n = 4$. For example, given the integers 1 to 20, then the concept of even numbers and the set of integers $\{2, 10, 17, 20\}$ forms a puzzle because 2, 10 and 20 are even, but 17 is not.

To add to our specification of the problem, we note that the solution to the puzzle must be *satisfying* to human solvers. There are many ways in which a solution could be unsatisfying, but we concentrate on only one here: if there is another solution of similar or lesser complexity than the solution given, this will be unsatisfying. As an example, consider the following puzzle:

Which number is the odd one out?

4 9 16 30

There are at least two simple solutions to this puzzle:

- 9 is the odd one out, because the others are even, yet 9 is odd
- 30 is the odd one out, as the others are square numbers but 30 is not

The first solution is perhaps most likely to be given as the answer because even numbers are more easily recognised than square numbers. However, this does not detract from the fact that the solutions are of similar complexity and if the solver gave one solution but the 'correct' one was the other, the solver would probably be dissatisfied with the puzzle. Hence an additional criteria for puzzles is that they have no other solution of similar or lesser complexity. We can use HR to increase the likelihood that a puzzle satisfies this criteria, but we do not claim to rule out other solutions completely, and any puzzle HR produces may be unsatisfying. However, the same is true of human generated odd one out puzzles.

The application of HR to puzzle generation is still in its early stages. The domain we have used so far has a finite number of examples which we call 'pgrams', a shortening of 'puzzle diagrams'. In figure 3, we give four example pgrams. Each pgram has either a circle, square or triangle in each of the four corners, so there are $3^4 = 81$ pgrams in total. The initial concepts HR starts with in this domain only describe which shapes are in which positions. HR is not yet given more complicated concepts such as diagonals or rotation and reflection of one pgram to produce another.

To produce puzzles, we start HR with just one pgram and use it to perform concept formation with all the production rules other than compose, which enables it to exhaust the search. HR introduces counterexamples to false conjectures, and because there are only 81 pgrams in total, HR searches all of them for a counterexample. The search is exhausted after 25

Figure 3: Four pgrams

Which is the odd one out?

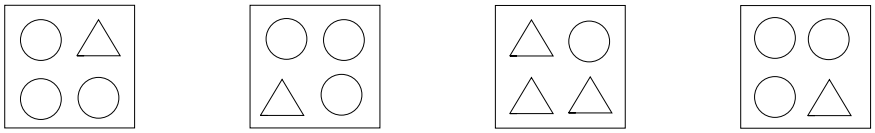


Figure 4: Puzzle generated by HR

seconds by which stage 14 pgrams have been introduced as counterexamples and HR has defined 62 specialisations of pgrams. HR then takes each specialisation concept S in turn and attempts to embed it into a puzzle. To do this, HR searches for 3 positive and one negative example of S. These have to be chosen in such as way that none of the other 61 specialisations provide a rival solution. A rival solution is one for which the odd one out differs to the negative example chosen for S. Choosing examples for S for which there is no rival solution increases the chance that the puzzle will be satisfying, but does not guarantee it.

We are still experimenting with different strategies for producing puzzles and more work needs to be done to increase the yield. Using the above approach, only 5 distinct puzzles were found, including the one in figure 4. This puzzle embeds the concept of having exactly one triangle, hence the odd one out is number 3, and this puzzle is easy to solve. More importantly, however, the rival solutions to the puzzle seem to be more contrived. For instance, number 4 could be considered the odd one out because it has two circles on its bottom-left to top-right diagonal, whereas the others have both a circle and a triangle. HR did not start with the concept of diagonals or invent the concept itself, so it did not notice this rival solution. With the rival solutions being more contrived, it seems likely that, while it is easy to solve, the solution to this puzzle will be satisfying to most people, although we need to confirm this with further experimentation.

Again, we can only speculate on the use of Progol for this application. The learning task we set HR was to produce a set of specialisation concepts which had good coverage of the simple concepts in a domain, so that rival solutions can be checked. By giving Progol many different binary categorisations of the pgrams, it could learn definitions for many concepts, keeping those which are below some pre-defined complexity limit. However, there are far too many ways to categorise the 81 pgrams, so either some selection of the categorisations would be required, or a smaller number of pgrams could be used. For example, there are around 10,000 different ways to categorise 14 pgrams into positive and negative examples, and it may be possible to learn definitions for this set.

Perhaps a more feasible alternative use of Progol for this task would be the following. Firstly, choose 4 pgrams from the set of 81 and choose

one of these to be a negative example, with the other three being positive examples. Then attempt to learn a concept with this categorisation of the examples and record the complexity of any definition produced. If this is achieved, a legal puzzle will have been generated and it will be necessary to check for rival solutions. One way to do this would be to re-categorise the four examples, choosing a new negative example and attempt to find a new definition. If only definitions with much larger complexity than the first one could be found, the puzzle will have no simple rival solution. This approach appears to be as plausible as our approach with HR, although we need to experiment to check this. However, the problem with this approach might the small number of examples: only three positive and four negative examples. With such a small number of examples, Progol may not be able to learn a definition which achieves any compression.

4 A Comparison of the Four Applications

By highlighting some commonalities between the four applications described above, we can draw some conclusions about the application of theory formation in general.

Our first observation is that with all four applications, part of the goal is to learn a concept which has certain properties. This is clear with the application to inducing a definition from examples, where the goal is to find a concept which achieves a given categorisation of the examples supplied. With scientific discovery — in the way that HR performs it — the goal is to find a concept for which even the categorisation is not known. The concept must have the property of being interesting. With the application to creative problem solving, the aim is to find a concept which is a generalisation of the given concept. With the application to puzzle generation, the aim is to find a concept for which examples can be found for a puzzle, for which there is no simple rival solution. To check that there are no rivals, HR also needs to generate a large set of concepts from which a rival might be found.

Hence we can conclude that three main applications of theory formation are (i) to find something about a given concept (i.e. a definition, or a property), (ii) to find an entirely new concept with a particular property and (iii) to find a set of concepts which cover all definitions of a particular form. With the exception of the generation of novel integer sequences, concept formation has been the main aspect of theory formation required for the problem. However, as we discuss in §5, we also hope to apply the conjecture making aspects of theory formation to areas of Artificial Intelligence, in particular constraint satisfaction problems and automated theorem proving.

The role of the user differs between each task. The user takes no part in the puzzle generation or the application to induction of definitions (other than supplying the positive and negative examples and perhaps making some adjustments to the settings). However, in the application to creative problem solving, the user must interpret the property HR suggests and determine whether it provides insight to the problem at hand. Similarly, with the discovery of integer sequences, the user must interpret the relations HR finds as conjectures and attempt to prove or disprove them. In this case, the user also has to choose one of HR's new sequences to investigate.

For each application, HR performed a different search for concepts and was enhanced with an additional module to complete the task after theory formation. For the induction of definitions, a unary-first search was used

and we implemented the lookahead mechanism. For the scientific discovery application, a heuristic search, based on the novelty heuristic (see [5]) was employed and the ability to data mine the Encyclopedia was added. For the problem solving application, a different heuristic search was used and the ability to notice generalisations of the given concept was added. For the puzzle generation, an exhaustive breadth first search was employed and the abilities to choose examples for the puzzle and check for rival solutions were implemented. Hence, while theory formation can provide the initial information for an application and varying the search should improve performance, further processing is required to complete the task.

4.1 A Further Comparison of Progol and HR

While HR and Progol can form similar concepts, they differ in how they can be applied to each problem. When learning definitions for concepts, Progol generates possible answers, then builds new answers from the ones which achieve most compression first. On the other hand, HR does not use the given concept to choose which concepts to build on, until the answer has effectively been found and the lookahead mechanism enables it to take a shortcut to the answer. Without tweaking Progol, it appears that if there are few positive examples of a concept, Progol will not consider complicated definitions for them, as this achieves no compression. This may be a drawback for learning mathematical concepts, where the definitions are often fairly complicated, yet the examples scarce. In contrast, HR will carry on regardless of the complexity of concepts being formed, until an answer is found, or until it runs out of memory, etc. On the other hand, Inductive Logic Programming is a much more powerful technique than HR's lookahead mechanism, because this mechanism does not drive the search until the search is nearly over.

HR's application to scientific discovery was slightly different to Progol. Progol was used to find possibly complicated definitions for scientific concepts, where a categorisation into positive and negative examples was known beforehand. These definitions were, in some cases, interpreted as rules and used to explain the phenomena differentiating the positive and negative examples. Progol has had much success with this approach in many areas, in particular chemistry, biology and medicine. HR's approach to discovery was more exploratory, because we used it to identify concepts new to us. We did not supply HR with any information about the concepts we hoping it would find (such as a categorisation into positive and negatives) other than the fundamental concepts in the domain, e.g. divisors. Because there are so many concepts in a domain, HR had to identify which were interesting during its search so that it could use a heuristic search to reach more interesting concepts. More than this, after HR had found a concept which was missing from the Encyclopedia of Integer Sequences, it mined the Encyclopedia to find interesting conjectures to add further interest to the concept. In contrast, for Progol, there was no need to find reasons why the definitions were interesting, because the fact that one had been found at all to explain the observed phenomena was interesting in itself.

It is more difficult to comment on the problem solving and puzzle generation applications, because we have yet to study how Progol would be best applied to these problems. We have mainly suggested how Progol could be used in terms of applying its definition induction techniques to the problem at hand and we have not looked at any clustering ability Prolog may have, which may be a more suitable approach. Problem solving may be problem-

atic for Progol, because it involves finding a definition not for the concept supplied, but for a generalisation of that concept. We have suggested a macro use of Progol, where negative examples are moved to the positives and a definition sought, which would produce a generalisation. However, this may turn out to be computationally expensive because there are many choices for the positives and negatives. Similarly with puzzle generation, we suggested that Progol try to learn definitions for given sets of examples and then show that no rival solution occurs.

For both the problem solving and puzzle generation applications, we have used HR to find a concept first, with the examples found afterwards. In contrast, our suggested use of Progol has been the opposite — to find the examples first, then find a definition which fits them. Until we perform more experiments with Progol, we cannot determine which approach is better for the two problems. However, in the case of puzzle generation, it is unlikely that a human writing a puzzle would start by writing down four examples, then try to find a concept embedded within them. However, we may find that a constraint based approach is more effective for puzzle generation.

5 Conclusions and Future Applications

Progol is generally used to induce a definition from a set of positive and negative examples, e.g. a definition for a subset of trains which are eastbound which distinguishes them from the westbound trains. This is a reactive process — a concept is immediately sought which defines the examples. It is possible to imagine another scenario, [3] whereby the program is given the same set of 10 trains and predicates describing them, but is allowed, say an hour, to prepare for an east-west question of the above nature. One effective way for a program to spend its time would be to invent many concepts related to trains, in particular, ways of classifying trains into a positive and a negative class. This is a more pro-active machine learning task, where the emphasis is on studying the trains rather than trying to learn a particular feature of them.

We gave the task to study trains as described above to HR, and in one hour it produced 160 specialisation concepts. There are only 638 ways to split 10 objects into two classes,[1] so if the user chose any subset of trains at random, there would be a one in four chance that HR could supply a reason why those trains were eastbound (and the others were not). We have performed similar pro-active learning tasks in number theory, using an agency of theory formation programs [6].

We have shown that theory formation can be applied to different learning tasks and highlighted the task involved, the additional functionality implemented in HR and the role of the user. While we make no claims that theory formation is the best way to approach these problems, we hope to have shown that it can be a useful tool for tasks involving learning. We have also compared HR to Progol both in terms of the concepts they form and their application (or proposed application) to the problems described. We have shown that Progol covers all the concepts that HR can form, but, even though HR was developed specifically in mathematical domains, only one of its production rules corresponds to additional background information in Progol. We have also suggested that for tasks such as puzzle generation, where it is necessary to find a set of concepts rather than just one and problem solving, where it is necessary to find a concept for which the

[1] See sequence A027306 in the Encyclopedia of Integer Sequences [22].

categorisation is not known, theory formation may be more applicable than the definition-inducing functionality that Progol mainly employs. However, we have not tested Progol in these areas and we do not comment in general on whether Progol could be employed to generate puzzles or solve problems of the type discussed above.

In future, we hope to apply HR to constraint satisfaction problems (CSPs), by automatically generating new constraints for a particular CSP. Each conjecture HR makes can, in principal, be turned into a new constraint for the CSP. However, certain conjectures will be less effective than others because they produce less propagation of constraints, and we will enable HR to decide which conjectures to add as constraints. We also hope to apply HR to automated theorem proving, whereby the user supplies a conjecture and requires a proof. We intend to test whether some initial theory formation before a proof attempt can decrease the time taken to prove a theorem. The theory produced would supply lemmas about the concepts in the conjecture statement which could be useful for the proof. As with CSPs, it will be necessary for HR to determine whether or not a lemma would be useful for a particular theorem. By applying HR to different problems, we hope to show that exploratory theory formation of the type HR undertakes embodies an important intelligent activity which has many uses in Artificial Intelligence.

Acknowledgments

I would like to thank Alan Bundy and Toby Walsh for continued detailed input to the HR project. I would also like to thank Stephen Muggleton, Chris Bryant and Richard Greaves for their in-depth discussions about Progol, HR and chemistry. Thanks also to Herbert Simon for enthusiastic discussions about the prospects for automated puzzle generation and problem solving.

References

[1] G Caporossi and P Hansen. Finding relations in polynomial time. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[2] S Colton. Refactorable numbers - a machine invention. *Journal of Integer Sequences*, 2, 1999.

[3] S Colton. Assessing exploratory theory formation programs. In *Proceedings of the AAAI-2000 workshop on new research directions in machine learning*, 2000.

[4] S Colton. Automated plugging and chugging. In M Kerber and M Kohlhase, editors, *Proceedings of the Eighth Symposium on the Integration of Symbolic Computation and Mechanized Reasoning*, 2000.

[5] S Colton, A Bundy, and T Walsh. HR: Automatic concept formation in pure mathematics. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999.

[6] S Colton, A Bundy, and T Walsh. Agent based cooperative theory formation in pure mathematics. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, 2000.

[7] S Colton, A Bundy, and T Walsh. Automatic identification of mathematical concepts. In *Machine Learning: Proceedings of the 17th International Conference*, 2000.

[8] S Colton, A Bundy, and T Walsh. Automatic invention of integer sequences. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 2000.

[9] S Epstein. On the discovery of mathematical theorems. In *Proceedings of the International Joint Conference on Artificial Intellignce*, 1987.

[10] S Fajtlowicz. On conjectures of Graffiti. *Discrete Mathematics 72*, 23:113–118, 1988.

[11] R King, S Muggleton, and M Sternberg. Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. of the National Academy of Sciences*, 89(23):11322–11326.

[12] D Lenat. AM: Discovery in mathematics as heuristic search. In D Lenat and R Davis, editors, *Knowledge-Based Systems in Artificial Intelligence*. McGraw-Hill Advanced Computer Science Series, 1982.

[13] W McCune. The OTTER user's guide. Technical Report ANL/90/9, Argonne National Laboratories, 1990.

[14] W McCune. A Davis-Putnam program and its application to finite first-order model search. Technical Report ANL/MCS-TM-194, Argonne National Laboratories, 1994.

[15] R Michalski and J Larson. Inductive inference of VL decision rules. In *Proceedings of the Workshop in Pattern-Directed Inference Systems (Published in SIGART Newsletter ACM, No. 63)*, 1977.

[16] S Muggleton. Inductive Logic Programming. *New Generation Computing*, 8(4):295–318, 1991.

[17] S Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[18] S Muggleton and L De Raedt. Inductive Logic Programming: Theory and methods. *Logic Programming*, 19-20(2):629–679, 1994.

[19] S Muggleton and D Page. A learnability model for universal representations. *JACM*, submitted 1999.

[20] J Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[21] H Simon and A Newell. Heuristic problem solving: The next advance in operations research. *Operations Research*, 6(1), 1958.

[22] N Sloane. The Online Encyclopedia of Integer Sequences. *http://www.research.att.com/~njas /sequences*, 2000.

[23] R Valdés-Pérez. Machine discovery in chemistry: New results. *Artificial Intelligence*, 74:191–201, 1995.

[24] S Wolfram. *The Mathematica Book, Fourth Edition.* Wolfram Media/Cambridge University Press, 1999.

[25] P Zeitz. *The Art and Craft of Problem Solving.* John Wiley and Sons, 1999.

Integrability, Exact Solvability, and Algebraic Combinatorics: A Three-Way Bridge?

Jim Propp
(propp@math.wisc.edu)

Department of Mathematics,
University of Wisconsin
(visiting Harvard University
and Brandeis University)

presented at the Workshop on
Combinatorics and Integrable Models
Australian National University
July 16, 2002

(last modified November 24, 2002)

I. INTRODUCTION

Some integrable systems

Continuous Painlevé II:

$$y''(x) = 2y^3 + xy + c$$

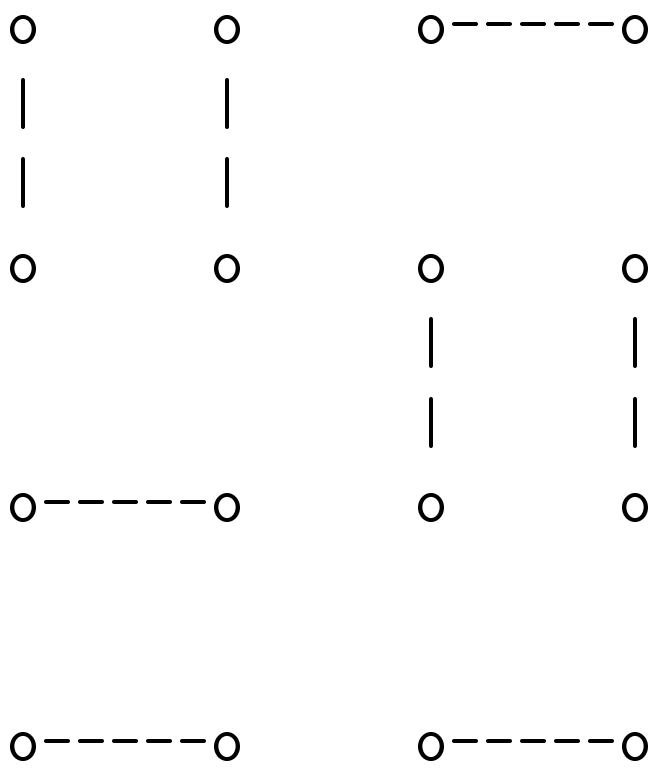
Discrete version:

$$nx_n + t(1 - x_n^2)(x_{n+1} + x_{n-1}) = 0$$

Autonomous discrete version:

$$x_n + t(1 - x_n^2)(x_{n+1} + x_{n-1}) = 0$$

The dimer model



Fisher and Temperley (1961) and Kasteleyn (1961) showed that the number of dimer configurations on the m-by-n rectangle with mn even is asymptotic to C^{mn}, where $C = e^{G/\pi} \approx 1.34$, and G is Catalan's constant $1-1/9+1/25-1/49+1/81-...$

The Discrete Hirota equation (see Zabrodin):

Let u, v, w be fixed vectors with the vector x varying over some fixed coset of $\mathbf{Z}u + \mathbf{Z}v + \mathbf{Z}w$.

Symmetric form:

$$aF(x+u)F(x-u)+bF(x+v)F(x-v)$$
$$+cF(x+w)F(x-w)=0$$

Asymmetric form:

$$F(x+w)F(x-w)=$$
$$aF(x+u)F(x-u)+bF(x+v)F(x-v)$$

Note:

$$\begin{aligned}&(x+u)+(x-u)\\ &=(x+v)+(x-v)\\ &=(x+w)+(x-w).\end{aligned}$$

Rhombus tilings of hexagons

Let $H(a,b,c)$ be the number of rhombus tilings of an a,b,c,a,b,c semiregular hexagon (opposite sides of equal length, all internal angles equal to 120 degrees). Then $H(a,b,c)$ satisfies the discrete Hirota relations

$$H(a,b,c)H(a,b-1,c-1)$$
$$= H(a+1,b-1,c-1)H(a-1,b,c)$$
$$+H(a,b-1,c)H(a,b,c-1)$$

and

$$H(a,b,c)H(a,b,c-2)$$
$$= H(a,b,c-1)H(a,b,c-1)$$
$$-H(a-1,b+1,c-1)H(a+1,b-1,c-1)$$

Plücker relations for Schur functions

Let $s_{m,n}$ be the Schur function whose shape is the rectangular Young diagram with m rows and n columns.
Then we have the formula (observed by Kirillov and later generalized by Kleber):

$$s_{m,n}\, s_{m,n} =$$

$$s_{m-1,n}\, s_{m+1,n} + s_{m,n-1}\, s_{m,n+1}\ .$$

Symmetry check:

$$\begin{aligned}(m,n) + (m,n) \\ = (m-1,n) + (m+1,n) \\ = (m,n-1) + (m,n+1).\end{aligned}$$

Number friezes ("frieze patterns")

(Conway and Coxeter;
Conway and Guy)

Rule: for the pattern $\begin{matrix} & a & \\ b & & c \\ & d & \end{matrix}$, we have $ad=bc-1$.

1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	3	1	2	2	1	3	1	2	2	1	3	1
		2	2	1	3	1	2	2	1	3	1	2	2
			1	1	1	1	1	1	1	1	1	1	1

Number walls

(Conway and Guy; Plouffe and Sloane; Lunnon)

```
                          a
Rule: For the pattern  b c d , we have ae = c^2 - bd
                          e
(with extra provisos we use when a = c^2 - bd = 0)
```

```
1    1    1    1    1    1    1    1    1    1    1

1    1    2    3    5    8   13   21   34   55   89

    -1    1   -1    1   -1    1   -1    1   -1

          0    0    0    0    0    0    0
```

If we start with a row of 1's, then we eventually get a row of 0's if and only if the sequence in the second row satisfies a linear recurrence with constant coefficients.

Somos-4 and Somos-5 sequences

The Somos-4 sequence:

$$1, 1, 1, 1, 2, 3, 7, 23, 59, 314, 1529, 8209, 83313, ...$$

$$S_n S_{n-4} = S_{n-1} S_{n-3} + S_{n-2} S_{n-2}$$

The Somos-5 sequence:

$$1, 1, 1, 1, 1, 2, 3, 5, 11, 37, 83, 274, 1217, 6161, ...$$

$$S_n S_{n-5} = S_{n-1} S_{n-4} + S_{n-2} S_{n-3}$$

II. CONDENSATION OF DETERMINANTS

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,n-1} & m_{1,n} \\ m_{2,1} & m_{2,2} & \dots & m_{2,n-1} & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \dots & m_{n-1,n-1} & m_{n-1,n} \\ m_{n,1} & m_{n,2} & \dots & m_{n,n-1} & m_{n,n} \end{pmatrix}$$

$$M_C = \begin{pmatrix} m_{2,2} & \dots & m_{2,n-1} \\ \vdots & \ddots & \vdots \\ m_{n-1,2} & \dots & m_{n-1,n-1} \end{pmatrix} \quad (\text{“center”})$$

$$M_{TL} = \begin{pmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,n-1} \\ m_{2,1} & m_{2,2} & \dots & m_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \dots & m_{n-1,n-1} \end{pmatrix} \quad (\text{“top left”})$$

with the $n-1$ by $n-1$ minors M_{TR} (“top right”), M_{BL} (“bottom left”), and M_{BR} (“bottom right”) defined similarly.

The Desnanot-Jacobi identity

$$\det(M)\ \det(M_C) =$$

$$\det(M_{TL})\ \det(M_{BR}) - \det(M_{TR})\ \det(M_{BL})$$

Dodgson condensation

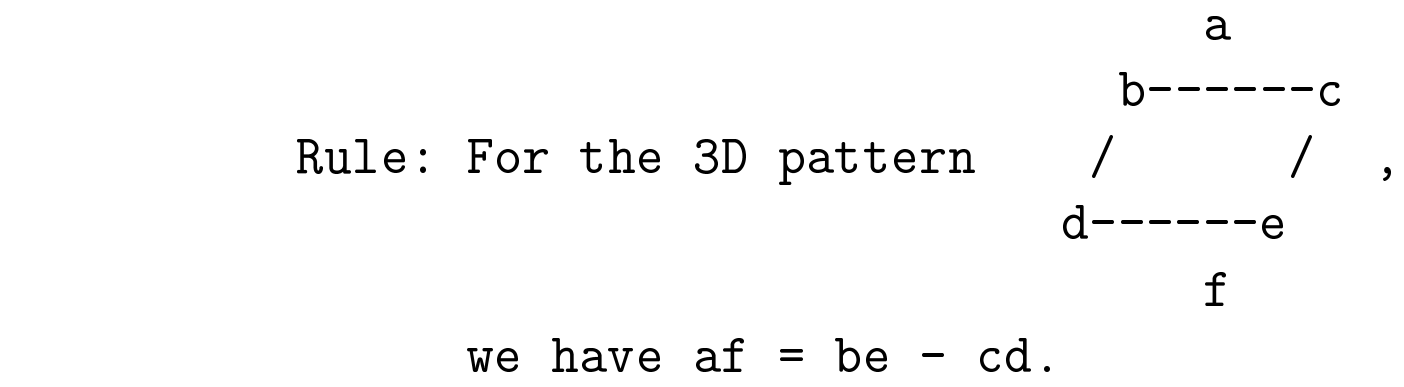
To compute the determinant of an n-by-n matrix, iteratively use this identity to compute the determinants of the connected minors of orders 1, 2, ..., n.
(A "connected" k-by-k minor is formed by taking k consecutive rows and k consecutive columns.)
Note: a minor of order 0 has determinant 1.
Example:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{pmatrix} \to \begin{pmatrix} 1 & 2 & 4 \\ 1 & 6 & 36 \\ 1 & 12 & 144 \end{pmatrix} \to$$

$$\begin{pmatrix} 2 & 12 \\ 2 & 48 \end{pmatrix} \to \begin{pmatrix} 12 \end{pmatrix}$$

Dodgson pyramids

Stacking the matrices in layers, we get a relation of discrete Hirota type, relating the values associated with the vertices of an octahedron:



Condensation applied to tridiagonal matrices (number friezes)

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \to$$

$$\begin{pmatrix} 2 & 1 & \mathbf{0} \\ 1 & 2 & 1 \\ \mathbf{0} & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \to \begin{pmatrix} 0 \end{pmatrix}$$

Condensation applied to Töplitz matrices (number walls)

$$\begin{pmatrix} 13 & 21 & 34 & 55 \\ 8 & 13 & 21 & 34 \\ 5 & 8 & 13 & 21 \\ 3 & 5 & 8 & 13 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \end{pmatrix}$$

$$\begin{pmatrix} 12 & 18 & 27 & 41 \\ 8 & 12 & 18 & 27 \\ 5 & 8 & 12 & 18 \\ 3 & 5 & 8 & 12 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & -9 \\ 4 & 0 & 0 \\ 1 & 4 & 0 \end{pmatrix} \rightarrow$$

$$\begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{-1} \end{pmatrix}$$

III. ALTERNATING-SIGN MATRICES

For $1 \leq m \leq n-1$, the determinant of the n-by-n matrix M can be expressed as a rational function of the determinants of the $(n-m)$-by-$(n-m)$ and $(n-m-1)$-by-$(n-m-1)$ connected minors of M.

(We can arrange these determinants in the form of an $(m+1)$-by-$(m+1)$ and an $(m+2)$-by-$(m+2)$ matrix; we can superimpose these two matrices, obtaining a "bimatrix".)

Theorem (Robbins and Rumsey, 1986): This rational function (the "determinant" of the bimatrix) is formally a Laurent polynomial in the determinants of the connected minors of order $n-m$ and $n-m-1$.

$m = 1$:

$$\ldots \to \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \to \begin{pmatrix} j & k \\ l & m \end{pmatrix} \to \begin{pmatrix} X \end{pmatrix} \ ,$$

$$\begin{aligned} X &= (jm - kl)/e \\ &= j^1 m^1 e^{-1} - k^1 l^1 e^{-1} \\ &= e^{-1} j^1 k^0 l^0 m^1 - e^{-1} j^0 k^1 l^1 m^0 \\ &= \left(\begin{pmatrix} 1 & & 0 \\ & -1 & \\ 0 & & 1 \end{pmatrix} \right) - \left(\begin{pmatrix} 0 & & 1 \\ & -1 & \\ 1 & & 0 \end{pmatrix} \right) \ . \end{aligned}$$

$m = 2$:

$$\ldots \to \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \to \begin{pmatrix} q & r & s \\ t & u & v \\ w & x & y \end{pmatrix} \to$$

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \to \begin{pmatrix} X \end{pmatrix} \ ,$$

$m = 2$ (continued):

X = an alternating sum of eight Laurent monomials

$$= \det\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & \\ 0 & 1 & 0 \\ 0 & -1 & \\ 0 & 0 & 1 \end{pmatrix} - \det\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & \\ 1 & 0 & 0 \\ 0 & -1 & \\ 0 & 0 & 1 \end{pmatrix}$$

$$- \det\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & \\ 0 & 0 & 1 \\ 0 & -1 & \\ 0 & 1 & 0 \end{pmatrix} - \det\begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & \\ 0 & 1 & 0 \\ -1 & 0 & \\ 1 & 0 & 0 \end{pmatrix}$$

$$+ \det\begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & \\ 0 & 0 & 1 \\ -1 & 0 & \\ 1 & 0 & 0 \end{pmatrix} + \det\begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & \\ 1 & 0 & 0 \\ -1 & 0 & \\ 0 & 1 & 0 \end{pmatrix}$$

$$+ \det\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & \\ 1 & -1 & 1 \\ 0 & -1 & \\ 0 & 1 & 0 \end{pmatrix} - \det\begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & \\ 1 & -1 & 1 \\ -1 & 0 & \\ 0 & 1 & 0 \end{pmatrix}.$$

Theorem (Robbins and Rumsey, 1986, continued):
Moreover, "Laurentness" continues to hold if the Dodgson recurrence

$$af = be - cd$$

is replaced by the recurrence

$$af = be + cd$$

or the more general

$$af = be + \lambda cd.$$

This is called the λ-determinant of the bimatrix.

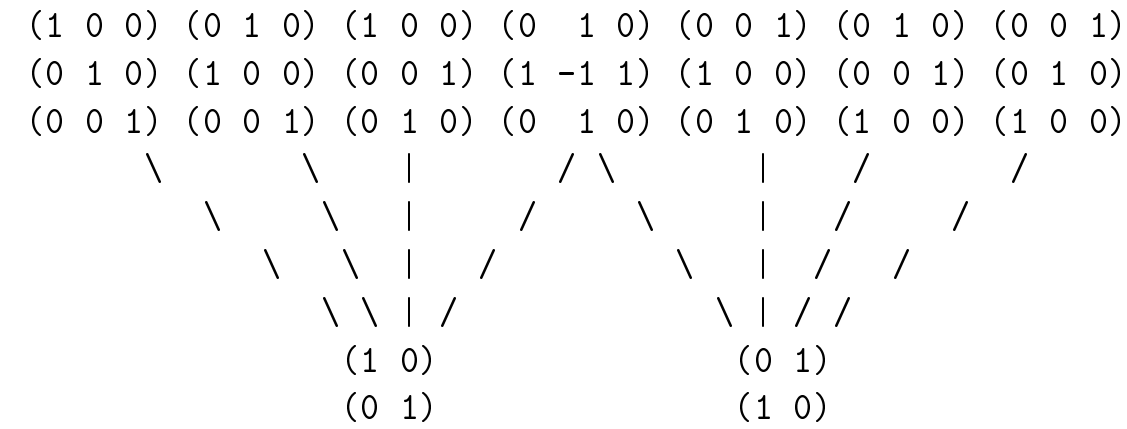
Theorem (Robbins and Rumsey, 1986, continued):
The λ-determinant of a bimatrix is a sum of

$$2^{m(m+1)/2}$$

monomials, in which each coefficient is plus or minus a power of λ, and the exponents of all the variables equal $+1$, 0, or -1. The terms correspond to the compatible pairs of ASMs (Alternating-Sign Matrices) of order m and $m+1$.

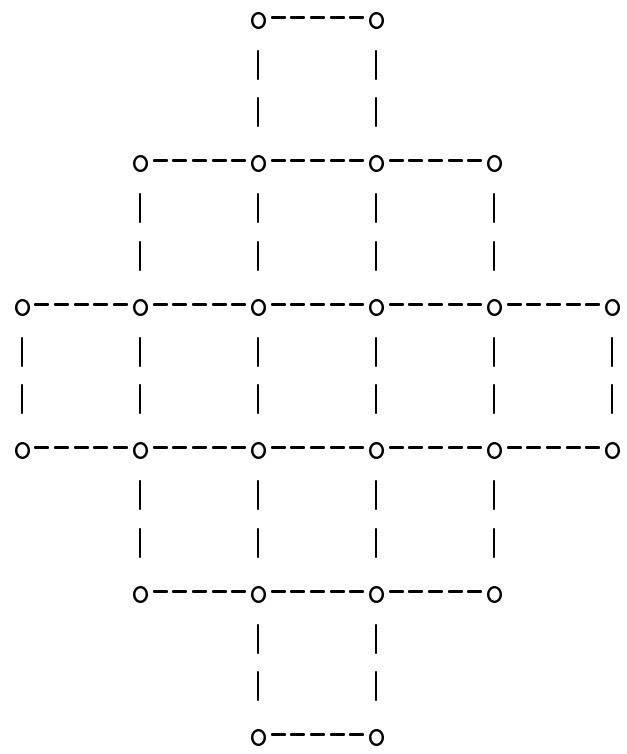
Compatibility

The ASMs of order 3, and the ASMs of order 2 they are respectively compatible with:



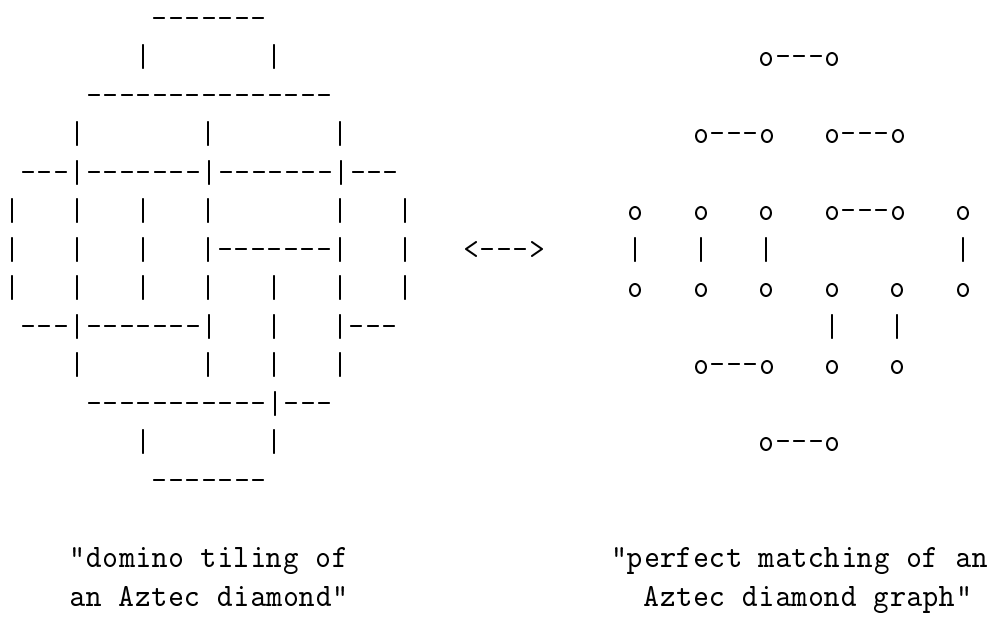
Aztec diamond graphs

Grensing, Carlsen, and Zapp (1980) conjectured that for graphs of the form



the number of perfect matchings (aka dimer configurations) is $2^{n(n+1)/2}$, where the rows and columns have respective lengths $2, 4, 6, \ldots, 2n, 2n, \ldots, 6, 4, 2$.

This was proved by Elkies, Kuperberg, Larsen, and Propp (1992), using the language of tilings instead of dimer covers:

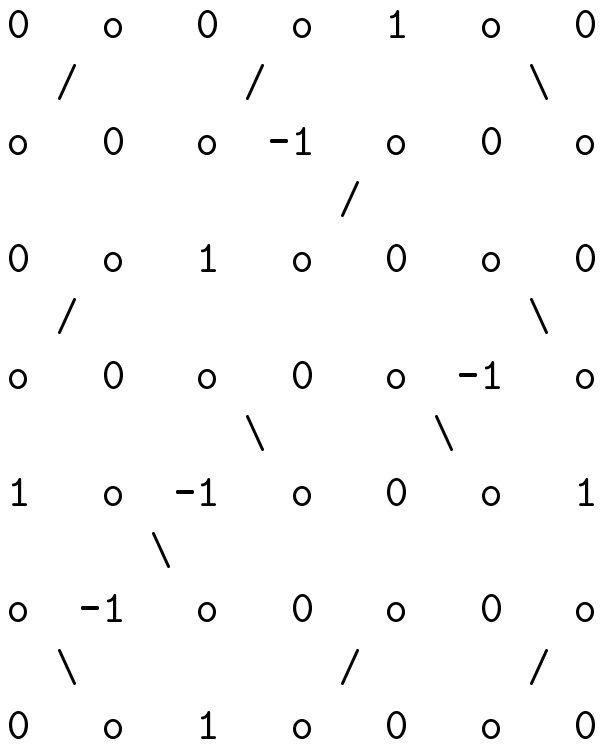


But in a sense the result was proved by Robbins and Rumsey (1986) using the language of compatible pairs of ASMs.

From matchings to pairs of ASMs

(Elkies, Kuperberg, Larsen, Propp)

In each hole, write 1 minus the number of neighboring edges included in the matching.



Decompose the bimatrix into two matrices, and negate the entries in the smaller matrix.

From pairs of ASMs to matchings (Carroll)

Insert 0's into the holes in the bimatrix, and at each location, record the (possibly empty) sum of all the entries above and/or to the left of it, obtaining the "corner-sum matrix" of the bimatrix.

```
0   0   0   0   0   0   0   0
      o       o       o
0   0   0   0   0   1   1   1
  o       o       o       o
0   0   0   0  -1   0   0   0
      o       o       o
0   0   0   1   0   1   1   1
  o       o       o       o
0   0   0   1   0   1   0   0
      o       o       o
0   1   1   1   0   1   0   1
  o       o       o       o
0   1   0   0  -1   0  -1   0
      o       o       o
0   1   0   1   0   1   0   1
```

Then replace each corner-sum by a mark, as prescribed by the following table, based on the dimensions of the sub-array and the value of the sum.

Even-by-even block:
  corner sum is -1: edge (/)
  corner sum is 0: no edge
  corner sum is 1: **incompatible**
Even-by-odd or odd-by-even block:
  corner sum is -1: **incompatible**
  corner sum is 0: no edge
  corner sum is 1: edge (\)
Odd-by-odd block:
  corner sum is -1: **incompatible**
  corner sum is 0: edge (/)
  corner sum is 1: no edge

The generic Dodgson pyramid with formal indeterminates at levels -1 and 0:

Level 1: Laurent polynomials with 2 terms
Level 2: Laurent polynomials with 8 terms
Level 3: Laurent polynomials with 64 terms
... etc.

Specialize!

(i.e., turn the Laurent polynomials into numbers)

The numbers $1, 1, 2, 8, 64, \ldots$ (of the form $2^{n(n+1)/2}$) yield a solution to the discrete Hirota equation, with initial conditions in a slab made of two planes of 1's:

```
1   1   1   1   1
                        1   1   1   1
1   1   1   1   1                       2   2   2
                        1   1   1   1                       8   8
1   1   1   1   1                       2   2   2                       64
                        1   1   1   1                       8   8
1   1   1   1   1                       2   2   2
                        1   1   1   1
1   1   1   1   1
```

If we let $M(n)$ be the number of perfect matchings of the Aztec diamond graph of order n, then we obtain

$$M(n+1)M(n-1) =$$

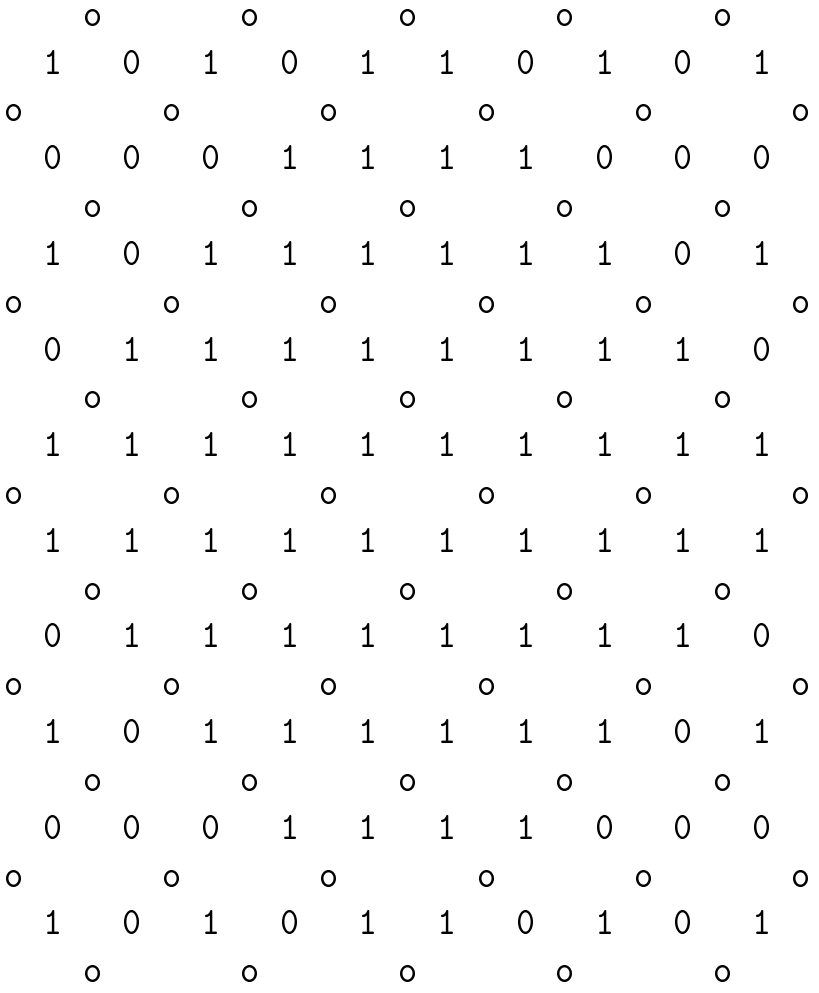
$$M(n)M(n) + M(n)M(n) \ ,$$

a 1-dimensional discrete Hirota equation that gives us an inductive proof of the formula $M(n) = 2^{n(n+1)/2}$.

In 1997 Eric Kuo found a direct combinatorial proof of this formula (via "graphical condensation").

These same methods apply to weighted enumeration of perfect matchings, where the weight of a particular perfect matching is product of the weights of its edges under some fixed weighting scheme.

The number of perfect matchings of a $2n$-by-$2n$ square equals the sum of the weights of the perfect matchings of the Aztec diamond of order $2n-1$, where edges are assigned weight 0 or weight 1 as shown below for the case $n = 3$:



The number of perfect matchings of a $2n$-by-$2n$ square equals the $(\lambda = 1)$-determinant of the $(2n+1)$-by-$(2n+1)$ bimatrix shown below for the case $2n = 6$:

$$\left(\!\!\left(\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right)\!\!\right)$$

(The intermeshed 6-by-6 matrix has all entries equal to 1.)

E.g., for $2n = 4$,

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \to$$

$$\begin{pmatrix} 1 & 4 & 1 \\ 4 & 8 & 4 \\ 1 & 4 & 1 \end{pmatrix} \to \begin{pmatrix} 12 & 12 \\ 12 & 12 \end{pmatrix} \to \begin{pmatrix} 36 \end{pmatrix}$$

IV. SOMOS SEQUENCES

Theorem (Somos; Fomin and Zelevinsky): If we put $S(1) = w$, $S(2) = x$, $S(3) = y$, and $S(4) = z$, and

$$S(n) = \frac{S(n-1)S(n-3) + S(n-2)^2}{S(n-4)}$$

for $n > 4$, then $S(n)$ is a Laurent polynomial in w, x, y, z (i.e., a polynomial in $w, w^{-1}, x, x^{-1}, y, y^{-1}, z, z^{-1}$).

Theorem (Somos; Fomin and Zelevinsky): If we put $S(1) = 1$, $S(2) = 1$, $S(3) = 1$, and $S(4) = 1$, and

$$S(n) = \frac{uS(n-1)S(n-3) + vS(n-2)^2}{S(n-4)}$$

for $n > 4$, then $S(n)$ is a polynomial in u, v.

Theorem (Fomin and Zelevinsky): Fix positive integers a, b, d with $a + b < d$. Put $S(i) = x_i$ for $i = 1, 2, ..., d$ and put

$$S(n) = \frac{uS(n-a)S(n-d+a) + vS(n-b)S(n-d+b)}{S(n-d)}$$

for $n > d$. Then $S(n)$ is polynomial in $x_1, x_1^{-1}, ..., x_d, x_d^{-1}, u, v$.

Special cases:

$d = 4, a = 1, b = 2$: Somos-4

$d = 5, a = 1, b = 2$: Somos-5

Quasi-theorem (Propp, Bousquet-Mélou and West, 2002):

All the coefficients in this polynomial are positive integers.

The values of Somos-4 also yield an intrinsically one-dimensional solution to the discrete Hirota equation, with initial conditions in a tilted slab made of four tilted planes of 1's:

```
1   2   3   7
                3   7   23
1   1   2   3                       23  59
                2   3   7                       314
1   1   1   2                        7  23
                1   2   3
1   1   1   1
```

Un-specialize! (???)

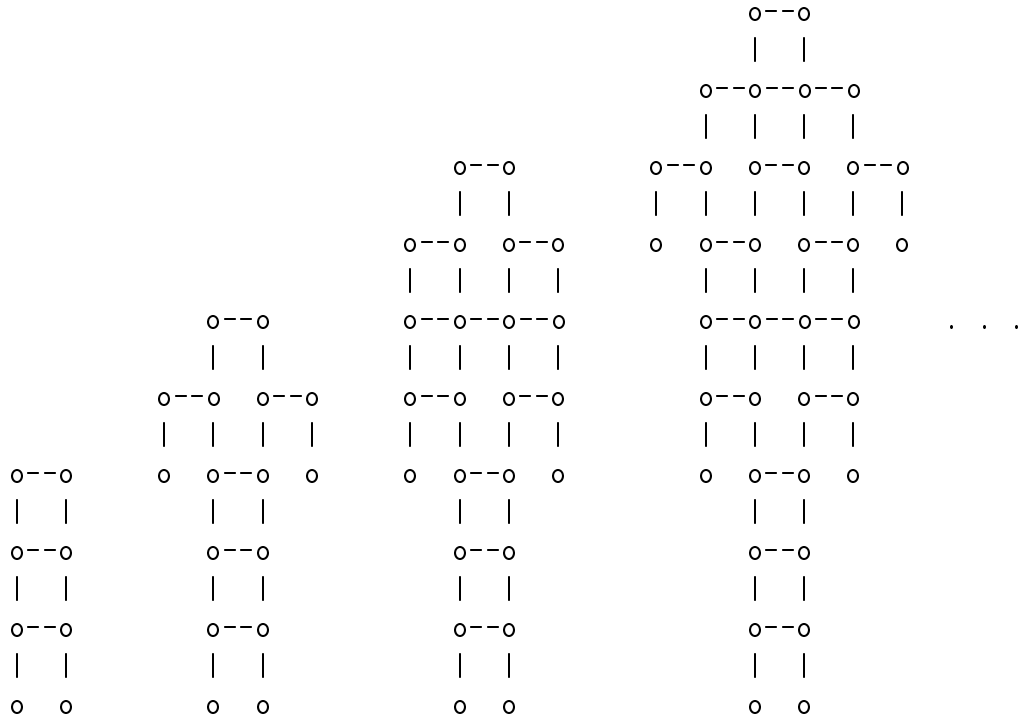
(i.e., turn the numbers back into Laurent polynomials)

Implementing recurrences in MAPLE

```
f := proc(n,i,j) option remember;
    if n<0 then undefined
    elif n<4 then x(n,i,j);
    else simplify( ( f(n-1,i-1,j)*f(n-3,i+1,j)
                   + f(n-2,i,j-1)*f(n-2,i,j+1))
                  /  f(n-4,i,j)); fi; end;
```

Empirically, one finds that `f(n,i,j)` is a Laurent polynomial in the x-variables in which every coefficient equals $+1$, so that the number of terms in `f(n,i,j)` is equal to the result of specializing all the x's to equal 1, which is equal to the nth term of the Somos-4 sequence.

The Somos-4 graphs



V. GROVES

Theorem (Fomin and Zelevinsky): Fix positive integers a, b, c and let $a' = b + c$, $b' = a + c$, and $c' = a + b$. Put $S(i) = x_i$ for $i = 1, 2, ..., a + b + c$ and put

$$S(n) = \frac{uS(n-a)S(n-a') + vS(n-b)S(n-b') + wS(n-c)S(n-c')}{S(n-a-b-c)}$$

for $n > a+b+c$. Then $S(n)$ is polynomial in $x_1, x_1^{-1}, ..., x_{a+b+c}, x_{a+b+c}^{-1}$, u, v, w.

Special cases:

$a = 1, b = 2, c = 3$: Somos-6

$$S(n)S(n-6) = S(n-1)S(n-5)$$
$$+S(n-2)S(n-4) + S(n-3)^2$$

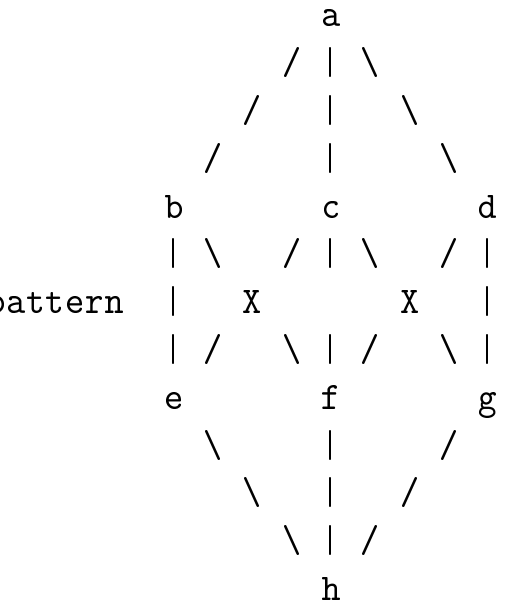
$a = 1, b = 2, c = 4$: Somos-7

$$S(n)S(n-7) = S(n-1)S(n-6)$$
$$+S(n-2)S(n-5) + S(n-3)S(n-4)$$

(Somos-6 and Somos-7 were first proved to give integers by Dean Hickerson.)

The cube recurrence (or Miwa equation)

Rule: For the 3D pattern



,

we have ah = bg + cf + de.

Conjecture (Propp, 1998):
For $i+j+k = -1, 0$, or 1, let $S(i,j,k) = x_{i,j,k}$, and for $i+j+k > 1$ inductively define $S(i,j,k)$ by the cube recurrence

$$\begin{aligned} & S(i,j,k)S(i-1,j-1,k-1) \\ = \; & S(i-1,j,k)S(i,j-1,k-1) \\ & +S(i,j-1,k)S(i-1,j,k-1) \\ & +S(i,j,k-1)S(i-1,j-1,k). \end{aligned}$$

Then for all i,j,k with $i+j+k \geq -1$, $S(i,j,k)$ is a Laurent polynomial in the x-variables, with all coefficients equal to 1 and all exponents bounded between -1 and 4 (inclusive).

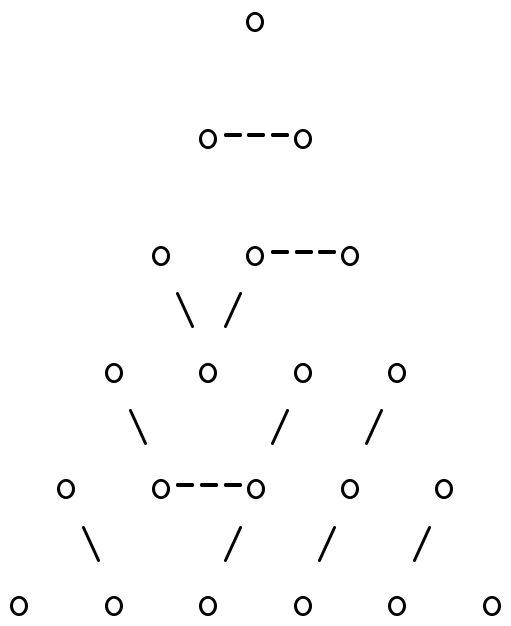
The Laurentness part of the claim was proved by Fomin and Zelevinsky. The rest was proved by my students Gabriel Carroll and David Speyer (with some input from me) in Spring 2002, when they discovered that these Laurent polynomials are encoding a hitherto unstudied kind of combinatorial object, which they call a *grove*.

A grove of order n is a special kind of forest on the triangle graph with $n+1$ vertices on a side. It is required that:

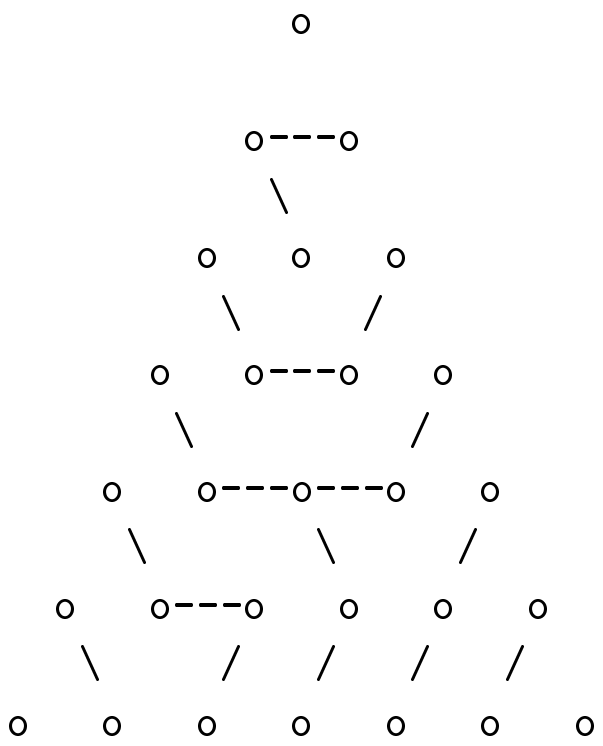
- every vertex is joined to a point on the boundary by a path; and
- two boundary-vertices v, w are joined by a path iff they are related by a reflection through a median that passes through a corner-point P of the triangle with the property that no corner of the triangle is closer to v or w than P.

(It follows that the corner vertices are isolated, and that every vertex other than the three corner vertices belongs to at least one edge.)

A grove of order 5



A grove of order 6



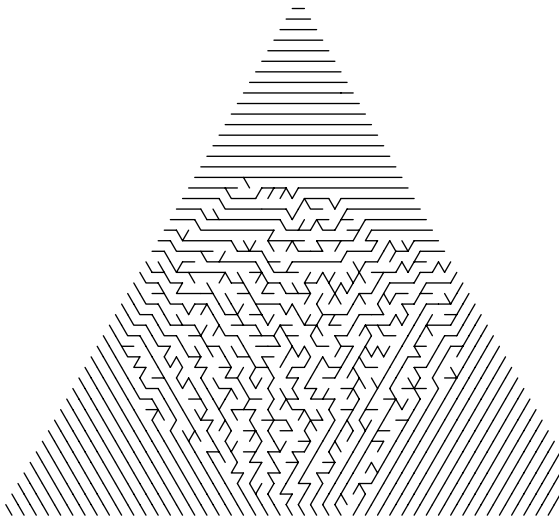
Theorem (Carroll and Speyer, 2002):
The number of groves of order m is

$$3^{\lfloor m^2/4 \rfloor},$$

and these correspond to the monomials in $S(i, j, k)$, where $m = i + j + k$.

We knew the RHS ahead of time; it was the LHS that was mysterious!

```
In[19]:= Grove[50]
```



```
Out[19]= - Graphics -
```

VI. AN INTEGRABLE SYSTEM

Bilinear version of autonomous discrete Painlevé II: Replace x_n by ix_n, so that the recurrence becomes

$$x_n + t(1 + x_n^2)(x_{n+1} + x_{n-1}) = 0.$$

Let $u = x_{-1}$ and $v = x_0$. Then the sequence can also be generated by the recurrence

$$x_{n+1}x_{n-1} = (x_n^2 + d)/(x_n^2 + 1)$$

where

$$d = -u^2 - v^2 - u^2v^2 - uv/t.$$

If we treat d as a formal variable, x_n (with $n \geq 1$) can be written as

$$P_n(u, v, d)/Q_n(u, v, d)$$

where P_n, Q_n are Laurent polynomials whose degrees grow subexponentially (quadratically, in fact).

P_n and Q_n satisfy a pair of joint (1D) discrete Hirota equations:

$$P_{n+1}P_{n-1} = P_nP_n + dQ_nQ_n$$

$$Q_{n+1}Q_{n-1} = P_nP_n + Q_nQ_n$$

with initial conditions $P_{-1} = u$, $P_0 = v$, $Q_{-1} = 1$, $Q_0 = v$.

The one-dimensional family of polynomials P_n, Q_n can be lifted to a three-dimensional family of Laurent polynomials $P(n,i,j)$, $Q(n,i,j)$ satisfying a pair of joint 3D discrete Hirota equations:

$$P(n+1,i,j)P(n-1,i,j) =$$
$$P(n,i-1,j)P(n,i+1,j)+$$
$$dQ(n,i,j-1)Q(n,i,j+1)\ ,$$

$$Q(n+1,i,j)Q(n-1,i,j) =$$
$$P(n,i-1,j)P(n,i+1,j)+$$
$$Q(n,i,j-1)Q(n,i,j+1)$$

The Laurent polynomials $P(n,i,j)$, $Q(n,i,j)$ enumerate weighted perfect matchings of the order n Aztec diamond graph.

VII. CONCLUSION

VIII. BIBLIOGRAPHY

D. Bressoud, "Proofs and Confirmations: The Story of the Alternating Sign Matrix Conjecture", Cambridge University, 1999.

D. Bressoud and J. Propp, How the alternating sign matrix conjecture was solved, *Notices of the AMS* **46** (1999), 637–646; `www.ams.org/notices/199906/fea-bressoud.pdf`.

J.H. Conway and H.S.M. Coxeter, Triangulated polygons and frieze patterns, *Math. Gaz.* **57** (1973) 87–94 (questions), 175–183 (answers).

J.H. Conway and R.K. Guy, "The Book of Numbers", Springer-Verlag, 1996; pp. 85–89.

N. Elkies, G. Kuperberg, M. Larsen, and J. Propp, Alternating-sign matrices and domino tilings, J. Algebraic Combin. **1** (1992), 111–132; `www.math.wisc.edu/~propp/aztec.ps.gz`.

M. Fisher and H.N.V. Temperley, The dimer problem in statistical mechanics — an exact result, *Phil. Mag.*-**6** (1961), 1061–1063.

S. Fomin and A. Zelevinsky, The Laurent phenomenon; `arXiv:math.CO/0104241`.

D. Gale, The Strange and Surprising Saga of the Somos Sequences, Mathematical Intelligencer 13 No. 1 (1991), 40–42. Reprinted in Chapter 1 of D. Gale, "Tracking the Automatic Ant". Springer-Verlag, 1998; pp. 2–5.

D. Gale, Somos Sequence Update, Mathematical Intelligencer 13 No. 4 (1991), 49–50. Reprinted in Chapter 4 of D. Gale, "Tracking the Automatic Ant", Springer-Verlag, 1998; pp. 22–24.

D. Grensing, I. Carlsen, and H.C. Zapp, Some exact results for the dimer problem on plane lattices with non-standard boundaries, *Phil. Mag. A* **41** (1980), 777–781.

P.W. Kasteleyn, The statistics of dimers on a lattice. I. The number of dimer arrangements on a quadratic lattice, *Physica* **27** (1961), 1209–1225.

P.W. Kasteleyn, Graph theory and crystal physics, in: "Graph Theory and Theoretical Physics", Academic Press, 1967; pp. 43–110.

M. Kleber, Plücker relations on Schur functions, `arXiv:math.QA/9907177`.

E. Kuo, Applications of graphical condensation for enumerating matchings and tilings; `http://www.cs.berkeley.edu/~ekuo/condensation.ps`.

F. Lunnon, The number-wall algorithm: an LFSR cookbook, *Journal of Integer Sequences*, **4** (2001), Article 01.1.1; `www.research.att.com/~njas/sequences/JIS/VOL4/LUNNON/numbwall10.html`.

S. Plouffe and N.J.A. Sloane, "The Encyclopedia of Integer Sequences", Academic Press, 1995.

D. Robbins and H. Rumsey, Determinants and alternating-sign matrices, *Advances in Math.* **62** (1986), 169–184.

H. Sakai, Rational surfaces associated with affine root systems and geometry of the Painlevé equations, *Commun. Math. Phys.* **220** (2001), 165–229.

A. Zabrodin: A survey of Hirota's difference equation, `arXiv:solv-int/9704001` (submitted 1997; last revised 2002)

See also links accessible from the following web-pages:

`www.math.wisc.edu/~propp/somos.html`
`www.math.wisc.edu/~propp/bilinear.html`
`www.math.harvard.edu/~propp/reach`

PREDICTING THE NUMBER OF HEXAGONAL SYSTEMS WITH 24 AND 25 HEXAGONS

Frédéric Chyzak[1], Ivan Gutman[2], and Peter Paule[3]

[1]*INRIA-Rocquencourt, B.P. 105, F–78153 Le Chesnay Cedex, France*

[2]*Faculty of Science, University of Kragujevac, P.O. Box 60, YU–34000 Kragujevac, Yugoslavia*

[3]*Research Institute for Symbolic Computation, Johannes Kepler University Linz, A–4040 Linz, Austria*

Abstract

We predict the number of hexagonal systems consisting of 24 and 25 hexagons to be $H_{24} = 122237774262384$ and $H_{25} = 606259305418149$, with 6 and 5 significant digits, respectively. Further estimates for H_n up to $n = 31$ are also given.

Hexagonal Systems

Informally speaking, a *hexagonal system* can be viewed as a connected arrangement of hexagonal cells packed in the same way as the typical honeycomb arrangement in a beehive. More formally, it is a finite connected plane graph with no cut-vertices, in which all interior regions are mutually congruent regular hexagons [1]. Hexagonal systems have from time to time attracted the attention of mathematicians (and were named "*hexagonal animals*", "*honeycomb systems*", "*polyhexes*", etc.), in connection with statistical physics and applications to lattice gas models [2, 3, 4]. But the main interest in them comes from chemistry: hexagonal systems are the natural graph representations of *benzenoid hydrocarbons*, whence the names "*benzenoid graphs*", "*benzenoid systems*", and "*fusenes*" used in the chemical literature. An enormous literature exists on various chemical applications of hexagonal systems. We refer to [5, 6] for details and references.

One of the classical problems in the theory of hexagonal systems is their enumeration. In what follows, the number of non-isomorphic hexagonal systems consisting of n hexagons is denoted by H_n, where "non-isomorphic" means viewed up to translations, rotations, and symmetries. This in turn is equal to the number of n-cyclic benzenoid hydrocarbons. The first few values of H_n are given in Table 1.

The enumeration of hexagonal systems according to area stands as one of the most challenging unsolved problems of combinatorial theory (cf. Section 10.8.5 in [7]). In spite of numerous attempts, no one was successful in applying Pólya's theory [7, 8, 9] or any other technique of combinatorics to find H_n or, at least, in establishing the asymptotic behavior

n	1	2	3	4	5	6
H_n	1	1	3	7	22	81
n	7	8	9	10	11	12
H_n	331	1435	6505	30086	141229	669584
n	13	14	15	16	17	
H_n	3198256	15367577	74207910	359863778	1751594643	

Table 1: Numbers H_n of hexagonal systems with n hexagons $(1 \leq n \leq 17)$

of H_n as n goes to infinity. Consequently, the only way to evaluate H_n is to use a (more or less) brute-force computer-assisted constructive enumeration; details of these methods are outlined in the book [10], in the reviews [11, 12], and elsewhere [13, 14, 15, 16, 17, 18]. Recently, some very efficient algorithms for the construction and counting of hexagonal systems were designed [17, 18], but even with them the calculation of H_n is extremely time- and memory-consuming. For instance, in order to obtain H_{22}, more than 300 days of CPU time were needed; the analogous calculation of H_{23} required 2.4 years of CPU time [18].

The values of H_n for n between 13 and 16 were first reported in 1990 by Knop *et al.* (H_{13} and H_{14} in [13], H_{15} and H_{16} in [14]). Three years later Tošić *et al.* arrived at H_{17} [15, 16]. With this the limit of the performance of the currently available computers had been reached, and further progress had to wait until a completely new algorithm was developed by Caporossi and Hansen [17] and further enhanced by Brinkmann [18]. This enabled the determination of H_{18} to H_{21} [17] as well as H_{22} and H_{23} [18]. It seems to be unlikely that the application of the same technique will be feasible in the case of $n \geq 24$.

It is a natural idea to somehow use the information contained in the sequence H_1, $H_2, \ldots, H_n$ to predict, at least approximately, the value of H_{n+1}. Early attempts in this direction [19, 20] were based on the assumption (without any theoretical justification, but in analogy with other results in graph enumeration) that for n being large enough, H_n can be approximated by some simple elementary function of n. This function was designed so as to depend on a few (usually two) adjustable parameters, the values of which were then determined from H_1, $H_2, \ldots, H_n$. The resulting values of H_{n+1} were eventually shown [13] to be quite accurate, but—of course—far from being exact. The same analysis was later applied to sequences of isomer counts of other homologous series of interest in chemistry [21, 22].

In this paper we report the results of an analogous approach, which, however, is much less arbitrary. Indeed, the class of sequences in which the approximation is searched for is much larger than those classes used so far, and allows for as many parameters as needed. The method is reminiscent of the methods of differential approximants [23] and algebraic approximants [24] used in statistical mechanics, and possesses the sound theoretical and algorithmic foundation of *holonomic functions*. This is the topic of the end of the introduction, which to a certain extend is independent from the rest of the text.

Holonomic Guessing

Being faced with the first five entries 0, 1, 3, 6, and 10 of an infinite sequence of numbers, an obvious guess for the sixth one would be 15. One could even propose the formula $n(n+1)/2$

for the nth entry, but this refined guess cannot be proved unless further information is provided. For instance, such a proof would become an easy task if we knew in addition that we are dealing with the sums of the first n nonnegative integers.

Over the years various computer algebra tools have been developed in order to assist this process of *guessing and proving*. As far as guessing is concerned, this is reflected by the success of Sloane's classical book [25] and its enlarged revision [26]. Each book is basically a table of sequences of integers, collected from all branches of mathematics and sciences. The sequences are arranged in numerical order, and come each with a brief description and references. The mere existence of these "dictionaries" has allowed for a new process of research: after generating the first numbers of a sequence of combinatorial interest, one identifies them with the aid of the tables. The work by Sloane and Plouffe has recently found an electronic and algorithmic supplement [27]: the tables are now electronically available for human search; additionally the on-line system now has a facility where it will algorithmically try to guess a formula or to relate the input sequence to a tabulated one. In particular, the counting sequence of hexagonal systems is now to be found there (known as sequences number A000228, A018190, and A038148). With regard to proving, we only mention Zeilberger's "holonomic systems approach to special function identities" [28] and the developments described in [29].

In this article, the aspect of computer-assisted *holonomic guessing* plays the central role. The first systematic presentation of the underlying theory of univariate holonomic functions has been given by Stanley [30]. The first implementation of these ideas was realized in the form of the Maple package GFUN by Salvy and Zimmermann [31]; it is now used as part of [27]. Another package named GENERATINGFUNCTIONS provides Mathematica users with the same functionality [32].

A detailed description of holonomic theory (e.g., closure properties of holonomic functions, etc.) would go far beyond the scope of this note. Therefore we restrict to introduce only those notions that are relevant to the understanding of the method to be used for predicting the values H_{24} and H_{25}.

For many counting sequences (a_n), the ordinary generating function and its exponential counterpart,

$$\sum_{n=0}^{\infty} a_n x^n \qquad \text{and} \qquad \sum_{n=0}^{\infty} \frac{a_n x^n}{n!},$$

respectively, are *holonomic*, which means that such a function or series satisfies a linear differential equation with polynomial coefficients. Examples of holonomic functions include many familiar power series such as algebraic functions (functions that are solution of a polynomial equation), the exponential function e^x, logarithmic function $\log(1+x)$, and trigonometric functions like $\sin x$. For example, if b_n denotes the number of binary planar trees with $n+1$ leaves (with the convention $b_0 = 1$), then the ordinary generating function of the sequence (b_n) is holonomic since

$$\sum_{n=0}^{\infty} b_n x^n = \frac{1-\sqrt{1-4x}}{2x}$$

is algebraic.

It is not difficult to prove that the series $\sum_{n=0}^{\infty} a_n x^n$ is holonomic if and only if the sequence (a_n) satisfies a linear recurrence with polynomial coefficents, i.e.,

$$p_0(n)a_n + p_1(n)a_{n+1} + \cdots + p_d(n)a_{n+d} = 0,$$

where the p_i's are polynomials in the indeterminate n. This serves as a motivation to call the sequence (a_n) *holonomic* in this case. Algorithmically it is easy to convert each representation—differential equation and recurrence—into the other. Furthermore, both representations serve as the basis for computer-assisted guessing. For example, let us assume that we came up with the first six binary tree numbers $(b_0, b_1, b_2, b_3, b_4, b_5) = (1, 1, 2, 5, 14, 42)$. Then we could use GFUN (or GENERATINGFUNCTIONS) to automatically guess the recurrence

$$(n+2)b_{n+1} - 2(2n+1)b_n = 0.$$

The procedure to produce this guess is essentially based on a simple coefficient comparison method (namely differential Padé-Hermite approximants) for which one has to bound in advance the order of the recurrence and the degree of the polynomial coefficients involved: the product "order times degree" is essentially the number of undetermined coefficients used by the method.

As mentioned above, additional information is needed in order to prove such a guess. For instance, if one knows in advance that the generating function is algebraic, which implies the existence of a holonomic recurrence, then one only needs to know an upper bound for its order. Or, if the holonomic nature is not known in advance, one might observe the convolution recurrence

$$b_n = \sum_{k=0}^{n-1} b_k b_{n-k-1}.$$

In this case transforming the conjectured recurrence of order 1 into the closed form

$$b_n = \frac{1}{n+1}\binom{2n}{n}$$

and substituting it into the convolution formula leads to the verification of a binomial identity. This could again be left to the computer by appying a symbolic summation procedure from [29]. (The numbers b_n above are the well-known Catalan numbers, often denoted by C_n.)

Concerning the problem of enumerating hexagonal systems, we do not know up to now whether the corresponding generating function of (H_n) is holonomic or not. Therefore we would need additional information to actually prove the accuracy of our guess, which can only be considered as a "holonomic approximation". The information we use for our holonomic guessing solely consists in the values of H_n that have been computed so far. In order to provide further evidence, we present a detailed analysis of the stability of the prediction scheme.

Holonomic guessing could also be considered as a kind of computer-assisted "heuristic reasoning", meant in the spirit of Pólya. According to his dictionary of heuristics [33]: *"We are often obliged to use heuristic reasoning. We shall attain complete certainty when we shall have obtained the complete solution, but before obtaining certainty we must often be satisfied with a more or less plausible guess."*

In the present article we use the Maple package GFUN. Analogous procedures are available to Mathematica users [32] and could have been used as well.

1 Warming Up: Predicting the Number of Hexagonal Systems with n Hexagons for n between 18 and 23

When Tošić *et al.* gave the value 1751594643 for H_{17} [15, 16], only the values of H_1, H_2, ... , H_{16} were known. All those results are summarized in Table 1. Using these initial 17 numbers as exclusive information about the sequence (H_n), we proceed to guess a linear recurrence satisfied by a holonomic approximation of the sequence. By means of it we then predict further numbers H_n of hexagonal systems when $18 \leq n \leq 23$, before comparing them with the actual values already known at present.

Prediction Scheme

We use the following prediction scheme:

Step 1. Load the package (as part of the standard distribution of Maple V Release 5), enter the list of numbers known after Tošić *et al.*, and set up a few package parameters.

```
with(share): with(gfun):
L:=[1,1,3,7,22,81,331,1435,6505,30086,141229,669584,
    3198256,15367577,74207910,359863778,1751594643]:
gfun['minordereqn']:=1: gfun['maxordereqn']:=2:
gfun['mindegcoeff']:=0: gfun['maxdegcoeff']:=20:
```

Specifically, we require the package to consider equations of order 1 or 2 with polynomial coefficients of degree between 0 and 10.

Step 2. *Guess* a recurrence satisfied by the sequence which starts with the values above:

```
rec17:=listtorec(L,u(n));
```

which outputs:

$$\texttt{rec17} := [\{p_0(n)u(n) + p_1(n)u(n+1) + p_2(n)u(n+2), u(0) = 1, u(1) = 1\}, \texttt{ogf}]$$

where each p_i above is a polynomial of degree 5 in n with integer coefficients of 52 digits. The explicit values are available in Appendix A.

Step 3. Convert this recurrence into a procedure which computes the nth term of the sequence:

```
pr17:=rectoproc(op(1,rec17),u(n));
```

Remarkably, the output procedure `pr17`, which is too large to be displayed here, has been *automatically generated* by GFUN. Additionally, GFUN *automatically optimized* it, in the sense of minimizing the number of arithmetical operations used in the procedure.

n	18	19	20
H'_n	8553612149	41892180909	205710300568
H_n	8553649747	41892642772	205714411986
$-\delta_n$	$4.4 \cdot 10^{-6}$	$1.1 \cdot 10^{-5}$	$2.0 \cdot 10^{-5}$
n	21	22	23
H'_n	1012535580260	4994621421396	24686078283303
H_n	1012565172403	4994807695197	24687124900540
$-\delta_n$	$2.9 \cdot 10^{-5}$	$3.7 \cdot 10^{-5}$	$4.2 \cdot 10^{-5}$

Table 2: Predicted numbers H'_n of hexagonal systems with n hexagons, actual numbers H_n, and corresponding relative errors $-\delta_n = -(H'_n - H_n)/H_n$ of prediction $(18 \leq n \leq 23)$

Step 4. Compute *predicted values* for hexagonal systems with 18 to 23 hexagons. The predicted values H'_n are in fact rational numbers rounded to the nearest integer. Rather than displaying the Maple output, as obtained by the command

```
seq(i=trunc(pr17(i-1)+1/2),i=18..23);
```

we give the predicted results in Table 2.

Comparison to Recent Results

The numbers obtained in *Step 4* of the previous scheme match *with good accuracy* those obtained by Caporossi and Hansen [17], and by Brinkmann, Caporossi and Hansen [18]. Indeed, the heavy computations described in [18, 17] proved the numbers H_n of hexagonal systems to be those given in Table 2. The table also gives the corresponding relative error

$$\delta_n = \frac{H'_n - H_n}{H_n}$$

of the predicted values H'_n.

In order to perform the calculations of `rec17`, `pr17`, and the estimates, *not more than 3 seconds of CPU time were needed.*

Note that other parameter settings could have been used in Step 1 above. Let us repeat that the number of undetermined coefficients used by the method is essentially the product "order times degree". The algorithm tries to detect equations with a small number of non-zero coefficients in the search space described by the parameters. The other setting

```
gfun['minordereqn']:=0: gfun['maxordereqn']:=20:
gfun['mindegcoeff']:=0: gfun['maxdegcoeff']:=2:
```

yields another equation with low polynomial degree but high order (specifically: order 8 instead of 2, degree 1 instead of 5, 25-digit instead of 52-digit integers). The latter recurrence results in different predicted numbers, which however approximate the actual ones with essentially the same good accuracy. This is why we will not discuss the choice of parameter settings any further.

n	4	5	6	7	8	9	10	11	12	13	14	15	16	17
order	2	2	2	2	2	2	2	2	1	2	2	1	2	2
degree	1	1	2	2	2	3	3	3	5	4	4	7	5	5
digits	1	1	3	4	5	9	13	17	26	28	33	47	48	52
n	18	19	20	21	22	23								
order	1	2	2	1	2	2								
degree	8	6	6	10	7	7								
digits	69	70	78	103	104	116								

Table 3: Parameters for the recurrence obtained by the scheme at nth stage $(4 \leq n \leq 23)$

2 Predicting the Number of Hexagonal Systems with 24 or More Hexagons

In the previous section, we started from a list of known values for the H_n (up to $n = 17$), and derived a *single* recurrence to predict *several* further values (up to $n = 23$). In this section, we follow a more incremental strategy: from a list of known or already predicted values for $H_1, \ldots, H_n$, we derive a recurrence to predict a *single* further value for H_{n+1}. Adjoining it to the initial list, we then iterate the process ℓ times, ending with *several* recurrences, one for each value predicted for $H_{n+1}, \ldots, H_{n+\ell}$.

Prior to this, we provide good numerical evidence for the stability of our incremental prediction scheme, which makes it possible to obtain values for H_{24} and H_{25} of credibly good accuracy.

Stability of the Prediction Scheme

Using all known values $H_1, \ldots, H_n$ for a number $n \leq 23$, one can predict the numbers H_{n+p} for $p \geq 1$ following the same scheme as previously outlined for $n = 17$. This is readily implemented in Maple:

```
L:=[1,1,3,7,22,81,331,1435,6505,30086,141229,669584,
    3198256,15367577,74207910,359863778,1751594643,
    8553649747,41892642772,205714411986,1012565172403,
    4994807695197,24687124900540]:
gfun['minordereqn']:=1: gfun['maxordereqn']:=2:
gfun['mindegcoeff']:=0: gfun['maxdegcoeff']:=20:
for i from 4 to nops(L) do
    rec[i]:=listtorec(L[1..i],u(n));
    pr[i]:=rectoproc(op(1,rec[i]),u(n))
od:
```

Setting the order and degree parameters as indicated in the Maple code above, the recurrences obtained are of small order (1 or 2), but involve polynomials in n of degree linear in n (typically, $\lfloor n/3 \rfloor$) and integers of (experimentally) $O(n \ln n)$ digits. This is summarized in Table 3. Denote by $H_n^{(p)}$ the value for H_{n+p} predicted p steps ahead by the

n	10	11	12	13	14	15
$p=1$	$9.9\cdot10^{-4}$	$4.8\cdot10^{-3}$	$7.8\cdot10^{-4}$	$-1.2\cdot10^{-5}$	$-4.3\cdot10^{-5}$	$1.6\cdot10^{-5}$
$p=2$	$7.2\cdot10^{-3}$	$1.6\cdot10^{-2}$	$3.1\cdot10^{-3}$	$-9.3\cdot10^{-5}$	$-1.8\cdot10^{-4}$	$7.1\cdot10^{-5}$
n	16	17	18	19	20	21
$p=1$	$-2.0\cdot10^{-6}$	$4.4\cdot10^{-6}$	$-5.2\cdot10^{-6}$	$8.6\cdot10^{-6}$	$-1.9\cdot10^{-6}$	$3.7\cdot10^{-6}$
$p=2$	$-3.8\cdot10^{-6}$	$1.1\cdot10^{-5}$	$-1.9\cdot10^{-5}$	$3.4\cdot10^{-5}$	$-6.7\cdot10^{-6}$	$2.0\cdot10^{-5}$
n	22					
$p=1$	$-7.6\cdot10^{-7}$					

Table 4: Relative errors $-\delta_n^{(p)}$ of prediction $(1 \le p \le 2,\ 4 \le n \le 22)$

scheme at the nth stage (i.e., by using the known H_k's for $1 \le k \le n$). This value $H_n^{(p)}$ is obtained as the result of the following Maple command (again, a rational number rounded to the nearest integer):

```
trunc(pr[n](n+p-1)+1/2);
```

Here, n and p are replaced by the corresponding integers.

The comparison of the estimate $H_n^{(p)}$ with the actual value H_{n+p} is achieved via the relative error

$$\delta_n^{(p)} = \frac{H_n^{(p)} - H_{n+p}}{H_{n+p}},$$

which is given in Table 4. Our calculations suggest that for a fixed p, each sequence of the absolute value $|\delta_n^{(p)}|$ of the errors made when predicting p steps ahead decreases with (possibly) some small oscillation.

The errors $\delta_n^{(p)}$ for higher values of p are given in Table 7 (Appendix B). The same remark about their decrease with small oscillation applies to values of p up to 8. Besides, the data in the table also strongly suggests a slow and monotonic variation of $-\delta_n^{(p)}$ with the parameter p (at least when n is greater than 8). More specifically, when $n \ge 8$ the ratio $\mu_n = \delta_n^{(8)}/\delta_n^{(1)}$ never exceeds a few hundreds.

Predictions

Following our calculation scheme and the recurrence computed for $n = 23$, we obtain the predictions for the next values of H_n that are given in Table 5. Note that the predicted values $H_n'' = H_n^{(1)}$ for $n > 23$ have been obtained by defining $H_n^{(p)}$ by the recurrence computed using the known values H_1 to H_{23} together with the *successively predicted ones* $H_{23}^{(1)}, H_{24}^{(1)}, \ldots, H_{n-1}^{(1)}$.

The validity of these predictions for $n = 24$ and $n = 25$ is suggested by the stability of the scheme, as described in the previous section (see Table 4). A similar analysis of Table 7 vindicates the further values and the bounds on the errors to be found in Table 5.

In order to perform the calculations of the recurrences, evaluation procedures, and estimates for each n between 1 and 23, *not more than 60 seconds of CPU time were needed.*

n	24	25	26
H''_n	122237774262384	606259305418149	3011424390300379
error	10^{-6}	10^{-5}	10^{-5}
n	27	28	29
H''_n	14979449994317356	74608167670480920	372053203099446920
error	10^{-5}	10^{-4}	10^{-4}
n	30	31	
H''_n	1857452345893521033	9283108148442320346	
error	10^{-3}	10^{-3}	

Table 5: Predicted numbers H''_n of hexagonal systems with n hexagons and presumable relative error bounds $(24 \leq n \leq 31)$

n	5	6	7	8	9	10	11	12	13
ρ_n	3.682	4.086	4.335	4.533	4.625	4.694	4.741	4.776	4.805
n	14	15	16	17	18	19	20	21	22
ρ_n	4.829	4.849	4.867	4.883	4.898	4.911	4.922	4.933	4.943
n	23	24	25	26	27	28	29	30	
ρ_n	4.951	4.960	4.967	4.974	4.981	4.987	4.992	4.998	

Table 6: Observed ratios $\rho_n = H_{n+1}/H_n$ $(5 \leq n \leq 22)$, as well as predicted ratios $\rho''_n = H''_{n+1}/H''_n$ $(23 \leq n \leq 30)$

Again, the other parameter setting suggested at the end of Section 1 yields a different recurrence (order 11 instead of 2, degree 1 instead of 7, 46-digit instead of 116-digit integers). However, the numbers predicted by this alternative recurrence remain close to the ones in Table 5.

3 Exponential Asymptotic Part

A natural idea is to consider the ratio $\rho_n = H_{n+1}/H_n$ of two successive terms of the sequence of observed numbers of hexagonal systems. Table 6 provides further evidence to corroborate the conjecture of Aboav and Gutman that the limiting value is remarkably close (or exactly equal) to 5 [20].

In the same vein, we observed that each predicted recurrence of the $H_n^{(p)}$ for fixed n asymptotically behaves exponentially, namely $H_n^{(p)} \sim K_n \alpha_n^p$ for a constant K_n and a parameter α_n that is an explicit algebraic number close to, but greater than 5. Furthermore, the greater n is, the closer to 5 the exponential parameter α_n is.

Acknowledgement

The work of F.C. and P.P. has been partially supported by the SFB grant F1305 of the Austrian Science Foundation (FWF). I.G. thanks the Johannes Kepler University in Linz (Austria) for a grant that enabled him to spend one month there in the year 1999.

A Explicit Value for the Recurrence of Section 1

The second-order recurrence in Step 2 of the prediction scheme described in Section 1 involves the following polynomials of degree 5 in n with integer coefficients of 52 digits:

$$\begin{aligned}
p_0 &= -1867772898049832297838775598964134957166764980189512 \\
&\quad - 10884556829407079968697291551132882484933172548220036n \\
&\quad + 12721533878650287528554902964949356722769733250349510n^2 \\
&\quad - 32534753292343260065038199203152144393520351720009 85n^3 \\
&\quad + 31810100631685730641224695385089000001332243568944 2n^4 \\
&\quad - 10942967863460680674924857755657134350847957422779n^5,
\end{aligned}$$

$$\begin{aligned}
p_1 &= 51118124221228019268396136936628708345336584647078 72 \\
&\quad + 35469788015542951395105181875419339475240204323784n \\
&\quad - 33671291121152645147418929533823926195198694878973 36n^2 \\
&\quad + 77028844367015161865182139013917112478567116397031 6n^3 \\
&\quad - 17497962137475978810591830043300350924099002352308n^4 \\
&\quad - 31888353912215559588134817508113297574479341820 08n^5,
\end{aligned}$$

$$\begin{aligned}
p_2 &= -10813465080243232096669460315662452924556311615061 20 \\
&\quad + 18257322986784771879043647738082020010521928020429 0n \\
&\quad + 29667227539257510475538771989575649829391434732023 1n^2 \\
&\quad - 50732258097471360256894519492471993600238207615036n^3 \\
&\quad - 70731069350496206435255977544411922570889741240 79n^4 \\
&\quad + 10276402384143351103899526601205366624396794469 14n^5.
\end{aligned}$$

B More Numerical Results Supporting the Prediction Accuracy

Table 7 is an extended version of Table 4. It suggests that the calculation method proposed in this paper is very stable, far beyond the prediction of the first next two values H_{24} and H_{25} of the sequence.

References

[1] Sachs, H. Perfect matchings in hexagonal systems. *Combinatorica 4*, 1 (1984), 89–99.

[2] Baxter, R. J. Hard hexagons: exact solution. *J. Phys. A 13*, 3 (1980), L61–L70.

[3] Baxter, R. J. *Exactly Solved Models in Statistical Mechanics.* Academic Press, London, 1982.

[4] Andrews, G. E., and Baxter, R. J. A motivated proof of the Rogers-Ramanujan identities. *Amer. Math. Monthly 96*, 5 (1989), 401–409.

[5] Gutman, I., and Cyvin, S. J. *Introduction to the Theory of Benzenoid Hydrocarbons.* Springer-Verlag, Berlin, 1989.

n	4	5	6	7	8	9
$p=1$	0.22	0.24	$3.6\cdot10^{-2}$	$-2.7\cdot10^{-2}$	$4.7\cdot10^{-2}$	$-7.5\cdot10^{-3}$
$p=2$	0.47	0.42	$9.0\cdot10^{-2}$	$1.3\cdot10^{-2}$	0.22	$-2.2\cdot10^{-2}$
$p=3$	0.68	0.60	0.16	3.7	0.65	$-4.2\cdot10^{-2}$
$p=4$	0.81	0.72	0.22	44.	1.6	$-6.7\cdot10^{-2}$
$p=5$	0.90	0.81	0.28	320.	3.3	$-9.6\cdot10^{-2}$
$p=6$	0.94	0.87	0.33	1800.	6.3	0.13
$p=7$	0.97	0.91	0.38	8000.	11.	0.17
$p=8$	0.98	0.94	0.43	31000.	17.	0.20
μ_n	4.4	3.9	12.	$-.11\cdot10^{7}$	360.	26.

n	10	11	12	13	14	15
$p=1$	$9.9\cdot10^{-4}$	$4.8\cdot10^{-3}$	$7.8\cdot10^{-4}$	$-1.2\cdot10^{-5}$	$-4.3\cdot10^{-5}$	$1.6\cdot10^{-5}$
$p=2$	$7.2\cdot10^{-3}$	$1.6\cdot10^{-2}$	$3.1\cdot10^{-3}$	$-9.3\cdot10^{-5}$	$-1.8\cdot10^{-4}$	$7.1\cdot10^{-5}$
$p=3$	$2.1\cdot10^{-2}$	$3.5\cdot10^{-2}$	$7.5\cdot10^{-3}$	$-3.0\cdot10^{-4}$	$-4.4\cdot10^{-4}$	$1.9\cdot10^{-4}$
$p=4$	$4.3\cdot10^{-2}$	$6.2\cdot10^{-2}$	$1.4\cdot10^{-2}$	$-6.9\cdot10^{-4}$	$-8.5\cdot10^{-4}$	$3.9\cdot10^{-4}$
$p=5$	$7.5\cdot10^{-2}$	$9.9\cdot10^{-2}$	$2.4\cdot10^{-2}$	$-1.3\cdot10^{-3}$	$-1.4\cdot10^{-3}$	$6.9\cdot10^{-4}$
$p=6$	0.12	0.14	$3.6\cdot10^{-2}$	$-2.1\cdot10^{-3}$	$-2.2\cdot10^{-3}$	$1.1\cdot10^{-3}$
$p=7$	0.17	0.19	$5.0\cdot10^{-2}$	$-3.2\cdot10^{-3}$	$-3.1\cdot10^{-3}$	$1.6\cdot10^{-3}$
$p=8$	0.22	0.25	$6.7\cdot10^{-2}$	$-4.5\cdot10^{-3}$	$-4.3\cdot10^{-3}$	$2.2\cdot10^{-3}$
μ_n	220.	53.	87.	370.	99.	140.

n	16	17	18	19	20	21
$p=1$	$-2.0\cdot10^{-6}$	$4.4\cdot10^{-6}$	$-5.2\cdot10^{-6}$	$8.6\cdot10^{-6}$	$-1.9\cdot10^{-6}$	$3.7\cdot10^{-6}$
$p=2$	$-3.8\cdot10^{-6}$	$1.1\cdot10^{-5}$	$-1.9\cdot10^{-5}$	$3.4\cdot10^{-5}$	$-6.7\cdot10^{-6}$	$2.0\cdot10^{-5}$
$p=3$	$-8.8\cdot10^{-6}$	$2.0\cdot10^{-5}$	$-4.6\cdot10^{-5}$	$8.3\cdot10^{-5}$	$-1.6\cdot10^{-5}$	
$p=4$	$-1.7\cdot10^{-5}$	$2.9\cdot10^{-5}$	$-9.0\cdot10^{-5}$	$1.6\cdot10^{-4}$		
$p=5$	$-3.0\cdot10^{-5}$	$3.7\cdot10^{-5}$	$-1.6\cdot10^{-4}$			
$p=6$	$-4.9\cdot10^{-5}$	$4.2\cdot10^{-5}$				
$p=7$	$-7.3\cdot10^{-5}$					

n	22
$p=1$	$-7.6\cdot10^{-7}$

Table 7: Error $-\delta_n^{(p)}$ of the prediction and measure $\mu_n = \delta_n^{(8)}/\delta_n^{(1)}$ of its variation with p $(1 \le p \le 8,\ \ 4 \le n \le 23)$

[6] Gutman, I. Topological properties of benzenoid systems. *Topics Curr. Chem. 162* (1992), 1–28.

[7] Harary, F., and Palmer, E. M. *Graphical Enumeration.* Academic Press, New York, 1973.

[8] Pólya, G., and Read, R. C. *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds.* Springer-Verlag, New York, 1987.

[9] Kerber, A. *Algebraic Combinatorics via Finite Group Actions.* BI Wissenschaftsverlag, Mannheim-Wien-Zürich, 1991.

[10] Trinajstić, N., Nikolić, S., Knop, J. V., Müller, W. R., and Szymanski, K. *Computational Chemical Graph Theory: Characterization, Enumeration and Generation of Chemical Structures by Computer Methods.* Horwood, New-York, 1991.

[11] Cyvin, B. N., Brunvoll, J., and Cyvin, S. J. Enumeration of benzenoid systems and other polyhexes. *Topics Curr. Chem. 162* (1992), 65–180.

[12] Brunvoll, J., Cyvin, B. N., and Cyvin, S. J. Benzenoid chemical isomers and their enumeration. *Topics Curr. Chem. 162* (1992), 181–221.

[13] Müller, W. R., Szymanski, K., Knop, J. V., Nikolić, S., and Trinajstić, N. On the enumeration and generation of polyhex hydrocarbons. *J. Comput. Chem. 11*, 2 (1990), 223–235.

[14] Knop, J. V., Müller, W. R., Szymanski, K., and Trinajstić, N. Use of small computers for large computations: enumeration of polyhex hydrocarbons. *J. Chem. Inf. Comput. Sci. 30* (1990), 159–160.

[15] Mašulović, D., Tošić, R., Cyvin, B. N., and Cyvin, S. J. Supplement to the Düsseldorf-Zagreb numbers for polyhexes. *Commun. Math. Chem. 29* (1993), 165–166.

[16] Tošić, R., Mašulović, D., Stojmenović, I., Brunvoll, J., Cyvin, S. J., and Cyvin, B. N. Enumeration of polyhex hydrocarbons to $h = 17$. *J. Chem. Inf. Comput. Sci. 35* (1995), 181–187.

[17] Caporossi, G., and Hansen, P. Enumeration of polyhex hydrocarbons to $h = 21$. *J. Chem. Inf. Comput. Sci. 38* (1998), 610–619.

[18] Brinkmann, G., Caporossi, G., and Hansen, P. Mathematics, chemistry and record hunting. To be published, 1999.

[19] Gutman, I. Number of benzenoid hydrocarbons. *Z. Naturforsch. A 41*, 8 (1986), 1089–1090.

[20] Aboav, D., and Gutman, I. Estimation of the number of benzenoid hydrocarbons. *Chem. Phys. Lett. 148* (1988), 90–92.

[21] Cioslowski, J. Series analysis methods in enumeration of chemical isomers. *Theor. Chim. Acta 76* (1989), 47–51.

[22] Aboav, D., and Gutman, I. Estimation of the number of unbranched catacondensed benzenoid hydrocarbons. *J. Serb. Chem. Soc. 54* (1989), 249–251.

[23] Guttmann, A. J. Asymptotic analysis of power-series expansions. In *Phase Transitions and Critical Phenomena*, vol. 13. Academic Press, London, 1989, pp. 1–234.

[24] Brak, R., and Guttmann, A. J. Algebraic approximants: a new method of series analysis. *J. Phys. A 23*, 24 (1990), L1331–L1337.

[25] Sloane, N. J. A. *A Handbook of Integer Sequences.* Academic Press, New York-London, 1973.

[26] Sloane, N. J. A., and Plouffe, S. *The Encyclopedia of Integer Sequences.* Academic Press, San Diego, CA, 1995.

[27] Sloane, N. J. A. Sloane's on-line encyclopedia of integer sequences. Available on the web from the URL `http://www.research.att.com/~njas/sequences/`.

[28] Zeilberger, D. A holonomic systems approach to special functions identities. *J. Comput. Appl. Math. 32*, 3 (1990), 321–368.

[29] Petkovšek, M., Wilf, H. S., and Zeilberger, D. $A = B$. Peters, Wellesley, MA, 1996.

[30] Stanley, R. P. Differentiably finite power series. *European J. Combin. 1*, 2 (1980), 175–188.

[31] Salvy, B., and Zimmermann, P. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Trans. Math. Softw. 20*, 2 (1994), 163–177.

[32] Mallinger, C. *Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences.* Master thesis, RISC, Johannes Kepler Universität Linz, Austria, Aug. 1996. Available at the URL:

`http://www.risc.uni-linz.ac.at/research/combinat/risc/publications/`.

[33] Pólya, G. *How to Solve It*, second ed. Princeton University Press, Princeton, NJ, 1988.

Appendix

CORRECTIONS AND UPDATES – 1st JANUARY, 2003

The book by K. Ohshika has been translated in english [Ohshi–02]. The main chapters are on Gromov's hyperbolic groups, on automatic groups, and on Kleinian groups.

I.B, and random walks on groups.

There is a nice introduction to random walks and diffusion on groups in [Salof–01], starting with a discussion on shuffling cards. A short exposition of Pólya's recurrence theorem can be found in [DymMc–72].

II.21 and VII.38, subgroup growth, and normal subgroup growth.

For further work concerning numbers of subgroups and normal subgroups of finite index in various groups, see among others [LiSMe–00] and [LarLu].

II.24, and a strong Schottky Lemma.

The classical Table-Tennis Lemma, or Schottky Lemma, is often used to show that a pair of isometries g, h of some hyperbolic space have *powers* g^n, h^n which generate a free group. There is a criterion for g, h to generate a free group in [AlFaN].

On free subgroups of isometry groups, see also [Woess–93] and [Karls].

II.25, II.33, and Möbius groups generated by two parabolics which are not free.

Let $\tilde{\Gamma}_z$ denote the subgroup of $SL(2,\mathbb{C})$ generated by $\begin{pmatrix} 1 & z \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ z & 1 \end{pmatrix}$, so that $\tilde{\Gamma}_z$ is free if $|z| \geq 2$ or if z is transcendental.

Grytczuk and Wójtowicz have shown that $\tilde{\Gamma}_{p/q}$ is *not* free for a set of rational values $z = p/q$ of the parameters which contains infinitely many accumulation points [GryWó–99].

II.28, and arithmeticity of lattices.

In $PSL(2,\mathbb{C})$, all arithmetic lattices which are generated by two elliptic elements and which are not co-compact have been determined [MacMa–01].

II.29$\frac{1}{3}$, more flowers for the herbarium of free groups.

Margulis has discovered a remarkable example of a free subgroup of the affine group of $\mathbb{R}^3$ acting *properly* on $\mathbb{R}^3$ [Margu-83]; an exposition appears in [Drumm–92].

II.29$\frac{2}{3}$, complement on groups with free subgroups.

We reproduce (most of) Problem 12.24 from the Kourovka Notebook.

Given a ring R with identity, the automorphisms of $R[[x]]$ sending x to $x(1 + \sum_{i=1}^{\infty} a_i x^i)$ form a group $N(R)$. We know that $N(\mathbb{Z})$ contains a copy of the free group F_2 of rank 2 (...). Does $N(\mathbb{Z}/p\mathbb{Z})$ contain a copy of F_2 ?

The answer is "yes": see [Camin–97]; it could be a challenge to find a table-tennis proof of this fact. For generalities on these "Nottingham groups" $N(R)$, see [Camin–00].

II.41, a misprint.

There is a misprint in the reference to [Bourb–75], which should be to Chapter VIII, § 2, Exercise 10.

II.41, and dense free subgroups of Lie groups.

The following result [BreGe] answers a question raised by A. Lubotzky and R. Zimmer: *if* Γ *is a dense subgroup of a connected semisimple real Lie group* G*, then* Γ *contains two elements which generate a dense free subgroup of* G. Also: in a connected non-solvable real Lie group of dimension d, any finitely generated dense subgroup contains a dense free subgroup of rank $2d$.

II.42, on Tits' alternative.

Let Γ be a subgroup of the group of homeomorphisms of the circle such that the action of Γ on the circle is minimal. Then, either the action is a conjugate of an isometric action, and therefore Γ contains a commutative subgroup of index at most 2, or Γ contains a so-called quasi-Schottky subgroup, which is in particular a non-abelian free subgroup [Margu–00]. A variation (possibly a simplification ?) of Margulis' original ideas appear in Section 5.2 of [Ghys–01].

For a group Γ of orientation preserving $\mathcal{C}^2$-diffeomorphisms of the circle, it is also known that the existence of an exceptional minimal set implies that Γ has non-abelian free subgroups [Navas].

On $Out(F_n)$, see also [BesFe–00].

Tits' alternative holds for automorphism groups of free soluble groups [Licht–95] and for linear groups over rings of fractions of polycyclic group rings [Licht–93], [Licht–99]. It also holds in a strong sense for subgroups of Coxeter groups [NosVi–02].

If Γ is a Bieberbach group, either both its automorphism group and its outer automorphism group are polycyclic, or both contain non-abelian free subgroups. See [MalSz] for precise criteria to decide which situation holds for a given Bieberbach group, in terms of the associated holonomy representation.

III.4, and examples of non-uniform tree lattices.

For the existence of such non-uniform lattices on uniform trees, see the work of L. Carbone [Carbo–01]. For tree lattices in general, see [BasLu–01].

III.6$\frac{1}{2}$, and further examples of finitely-genrated groups.

Let A be a commutative ring which is a finitely-generated $\mathbb{Z}$-module. Then the group A^* of invertible elements in A is a finitely-generated abelian group.

There is a proof in Section 4.7 of [Samue–67]; its main ingredient is Dirichlet's theorem, according to which the group of units in the ring of integers of a number field $\mathbb{K}$ is a direct product $F \times \mathbb{Z}^{r_1+r_2-1}$, where F is a finite group and r_1 [respectively $2r_2$] is the number of real [respectively complex] embeddings of $\mathbb{K}$ in $\mathbb{C}$.

More generally, if B is a commutative ring which is reduced (this means that 0 is the *only* nilpotent element) and finitely generated over $\mathbb{Z}$, then B^* is finitely generated [Samue–66].

III.18.iv, III.20, and residual finiteness.

On residual finiteness and topological dynamics: see also [Egoro–00].

A proof that finitely-generated linear groups are residually finite appears as Proposition III.7.11 in [LynSc–77].

III.21, on Baumslag-Solitar groups which are Hopfian.

For the equivalence between "$\Gamma_{p,q}$ Hopfian" and "p, q meshed" to hold, the definition should be

> two integers $p, q \geq 1$ are *meshed* if they have precisely the same prime divisors

and *not* the definition as it reads in [BauSo–62], or on page 57. I am grateful to E. Souche who pointed out this correction to me.

III.21, on actions of Baumslag-Solitar groups on the line.

For any p, q with $p > q \geq 1$, there exists a faithful action of the group $BS(p, q)$ on the line by orientation preserving real-analytic diffeomorphisms. In particular, $Diff_+^\omega(\mathbb{R})$ contains Baumslag-Solitar groups which are not residually finite [FarFr].

III.24, on maximal subgroups.

In "familiar" uncountable groups, maximal subgroups cannot be countable. More precisely, Pettis [Petti–52] has shown that, if G is a second category[1] nondiscrete Hausdorff group containing a countable everywhere dense subset, then any proper subgroup H of G lies in an uncountable proper sugroup H_+ of G; if H is countable, H_+ can be taken to be everywhere dense as well.

In their work on maximal subgroups of infinite index in finitely generated linear groups (exluding extensions of solvable groups by finite kernels), Margulis and Soifer have shown that such a group Γ contains a free (*infinitely* generated) subgroup F_∞ which maps *onto* any finite quotient of Γ; they deduce from this that any maximal subgroup of Γ which contains F_∞ is necessarily of infinite index. Soifer and Venkataramana have shown the following result: if Γ is an arithmetic subgroup of a non-compact linear semi-simple group G such that the associated simply connected algebraic group over $\mathbb{Q}$ has the so-called congruence subgroup property, for example if $\Gamma = SL(n, \mathbb{Z})$ with $n \geq 3$, then Γ contains a *finitely generated* free subgroup which maps onto any finite quotient of Γ [SeiVe–00].

III.24 and VIII.39. The Grigorchuk group has the following property: any maximal subgroup in it is of finite index [Pervo–00]. The same property holds for any group commensurable with Γ [GriWi].

[1] Recall that a topological space X is "second category" (= non-meager) if it is *not* the union of countably many subsets whose closures have empty interiors ("ensembles rares"). Baire's theorem shows that locally compact spaces and complete metric spaces are second category, indeed are Baire spaces (= spaces in which countable unions of closed subspaces with empty interiors have empty interiors).

III.B, an additional problem: does $SO(3)$ act non-trivially on $\mathbb{Z}$? (Ulam's problem).

I do not know which uncountable groups can act faithfully on a countable set. Of course, the group $Sym(\mathbb{N})$ of all permutations of $\mathbb{N}$ is itself uncountable, and it has received attention at least since [SchUl–33]. Here is a sketch to show that $\mathbb{R}$, viewed as a discrete group, acts faithfully on $\mathbb{N}$; in other and somehow biased words, this produces "a continuous flow on a discrete space". I am most grateful to Tim Steger for several helpful conversations on this material.

Choose a basis (e_t) of $\mathbb{R}$ as a vector space over $\mathbb{Q}$ which is indexed by the open interval $]0,1[$ of the line. Let C denote the countable set of pairs (a,b) of rational numbers such that $0 < a < b < 1$. For each $(a,b) \in C$, the map

$$\phi_{a,b} \,:\, \mathbb{R} \ni \sum_{t\in(a,b)} x_t e_t \;\longmapsto\; \sum_{t\in]a,b[} x_t \in \mathbb{Q}$$

is well-defined, $\mathbb{Q}$-linear and onto. Observe that, for any $x \neq 0$ in $\mathbb{R}$, there exists $(a,b) \in C$ such that $\phi_{a,b}(x) \neq 0$. Now $\mathbb{N}$ is in bijection with the disjoint union $\bigsqcup_{(a,b)\in C} \mathbb{Q}_{a,b}$ of copies of $\mathbb{Q}$ indexed by C. Define an action ϕ of $\mathbb{R}$ on this union which leaves each $\mathbb{Q}_{a,b}$ invariant and for which $x \in \mathbb{R}$ transforms $q \in \mathbb{Q}_{a,b}$ to $q + \phi_{a,b}(x)$. This ϕ is a faithful action. [Even if it is not important for our argument, observe that the product over $(a,b) \in C$ of the $\phi_{a,b}$ is a $\mathbb{Q}$-linear bijection from $\mathbb{R}$ onto a subspace of the vector space which is a direct product over C of copies of $\mathbb{Q}$.]

The group $\mathbb{R}/\mathbb{Z}$ is a direct sum of the torsion group $\mathbb{Q}/\mathbb{Z}$, which is countable, and a group isomorphic to $\mathbb{R}$ (a $\mathbb{Q}$-vector space of dimension the power of the continuum). It follows from the previous construction that there exists an injective homomorphism from $\mathbb{R}/\mathbb{Z}$ into $Sym(\mathbb{N})$.

In 1960, Ulam asked if the compact group $SO(3)$ of rotations of the usual space, viewed as a discrete group, can act on a countable set (see Section V.2 in [Ulam–60]). As far as I know, this is still open. Previous observations are possibly near what Ulam had in mind when writing his comments in Section II.7 of [Ulam–60].

III.38 and III.D, on finite quotients of the modular group.

For more on which finite simple groups are quotients of $PSL(2,\mathbb{Z})$, see the exposition of [Shale–01].

III.45, uncountably many finitely-generated groups with pairwise non-isomorphic von Neumann algebras.

Let Γ be a torsion-free Gromov-hyperbolic group which is not cyclic. Building up on results of Gromov, Ol'shanskii has shown that Γ has an uncountable family $(\Gamma_\iota)_{\iota\in I}$ of pairwise non-isomorphic quotient groups, all of which are simple and icc [Ol's–93]. N. Ozawa [Ozawa] has shown that, for any given separable factor M of type II_1, the set of those $\iota \in I$ for which the unitary group $\mathcal{U}(M)$ has a subgroup isomorphic to Γ_ι is a countable set. In particular, the set of von Neumann algebras of the groups Γ_ι (which are factors of type II_1) contains uncountably many isomorphism classes.

III.46, on groups with two generators.

It has been shown that two randomly chosen elements of a finite simple group G generate G with probability 1 as $|G| \to \infty$ (work of Dixon, Kantor-Lubotzky, Liebeck-Shalev, see [Shale–01]).

IV.1 and VI.1, on infinite generating sets and related word lengths.

Consider an integer $n \geq 2$, the group $\Gamma = SL(n, \mathbb{Z})$, and the infinite subset S of Γ consisting of those matrices of the form $I + kE_{i,j}$, with $k \in \mathbb{Z}$, $i, j \in \{1, \ldots, n\}$, $i \neq j$, and $E_{i,j}$ the matrix with all entries 0 except one 1 at the intersection of the ith row and the jth column.

As stated in Item III.2, the diameter of Γ with respect to the corresponding S-word length is finite as soon as $n \geq 3$.

IV.3.viii, on stable length: a correction.

The subadditivity

$$\tau(\gamma\gamma') \leq \tau(\gamma) + \tau(\gamma')$$

holds for *commuting* elements $\gamma, \gamma' \in \Gamma$ (as correctly stated by Gersten and Short).

For example, if γ, γ' are the two standard generators of the infinite dihedral group, then $\tau(\gamma\gamma') > 0$ and $\tau(\gamma) = \tau(\gamma') = 0$.

IV.24.i, and values of the indices for subgroups: a question.

Consider the following property of a group Γ: whenever two subgroups Γ_1, Γ_2 of finite indices are abstractly isomorphic, the indices $[\Gamma : \Gamma_1]$ and $[\Gamma : \Gamma_2]$ are equal.

Finitely generated free groups and fundamental groups of closed surfaces have this property, by an easy argument using Euler characteristics.

More generally, it would be interesting to know which groups have this property and which groups don't.

IV.25.vii, a quasi-isometry criterion for existence of lattices.

B. Chaluleau and C. Pittet [ChaPi–01] have answered one of the questions there and have shown:

Let N be a graded simply connected nilpotent real Lie group. If there exists a finitely-generated group which is quasi-isometric to N, then N has lattices.

IV.25, and examples of quasi-isometries.

(x) Say that a metric space X is *quasi-isometrically incompressible* if any quasi-isometric embedding from X into itself is a quasi-isometry. E. Souche [Souch] has shown that finitely generated nilpotent groups and uniform lattices in simple connected real Lie groups are quasi-isometrically incompressible, but that finitely-generated free groups and Baumslag-Solitar groups are not.

(xi) A finitely-generated group cannot be quasi-isometric to an infinite dimensional Hilbert space. Indeed, such a space has the following quasi-isometric-invariant property: for any positive number r, there exists a positive number R such that a ball of radius R contains infinitely many pairwise disjoint balls of radius r; and a finitely-generated group does not have this property.

IV.27, groups which are commensurable up to finite kernels.

Another terminology for commensurable up to finite kernels is *weakly commensurable* subgroups. See § 5.5 in [GorAn–93]; these authors also point out the following fact.

If M is a manifold on which some Lie group act transitively, then $\pi_1(M)$ contains a subgroup of finite index which is isomorphic to a discrete subgroup of a connected Lie group; if M is also compact, then $\pi_1(M)$ contains a subgroup of finite index which is isomorphic to a uniform lattice in some connected Lie group.

IV.29.v and VII.26, and the classification of lattices up to commensurability in some nilpotent Lie groups.

Y. Semenov has classified $\mathbb{Q}$-forms of some real nilpotent Lie algebras, and thus the commensurability classes of lattices in the corresponding nilpotent Lie groups [Semen]. It seems that the following question is open:

does there exist a finite dimensional real nilpotent Lie algebra of which the number k of $\mathbb{Q}$-forms (up to isomorphism) is such that $1 < k < \infty$?

IV.34 & 35, and commensurability. The following exercise is taken from [Gabor–02] and is clearly missing just before IV.34.

Exercise. (i) Show that two groups Γ_1, Γ_2 are commensurable if and only if they have commuting free actions on a set X such that both quotients $\Gamma_1\backslash X, \Gamma_2\backslash X$ are finite.

[Hint for one direction. Let Γ_j' be a subgroup of finite index in Γ_j, $j = 1, 2$, such that there exists an isomorphism $\varphi : \Gamma_1' \longrightarrow \Gamma_2'$. Set $\Delta = \{ (\gamma_1, \gamma_2) \in \Gamma_1 \times \Gamma_2 \mid \gamma_1 \in \Gamma_1', \gamma_2 = \varphi(\gamma_1) \}$. Consider the natural actions of Γ_1 and Γ_2 on $(\Gamma_1 \times \Gamma_2)/\Delta$.

Hint for the other direction. Choose $x_0 \in X$. Consider the natural action of Γ_1 on $\Gamma_2\backslash X$ and the canonical projection $[x_0]_2$ of x_0 in $\Gamma_2\backslash X$. Let Γ_1' be the isotropy subgroup of Γ_1 defined by $[x_0]_2$ and set $\gamma_1 x_0 = \varphi(\gamma_1)x_0$. Check that φ is a well-defined group homomorphism $\Gamma_1' \longrightarrow \Gamma_2$ which is injective and whose image is of finite index in Γ_2.]

(ii) Assume that Γ_1, Γ_2 have commuting free actions on X such that both $\Gamma_1\backslash X, \Gamma_2\backslash X$ are finite, and let Γ_1', Γ_2' be as in the previous hints. Check that

$$\frac{[\Gamma_1 : \Gamma_1']}{[\Gamma_2 : \Gamma_2']} = \frac{|\Gamma_2\backslash X|}{|\Gamma_1\backslash X|}\,.$$

IV.36, on commensurability and torsion.

G. Levitt has observed that a group Γ with infinitely many torsion conjugacy classes can have a subgroup of finite index Γ_0 which is torsion-free.

Indeed, let first Γ_0 be the wreath product $\mathbb{Z} \wr \mathbb{Z} = (\oplus_{i\in\mathbb{Z}}\mathbb{Z}a_i) \rtimes \mathbb{Z}$, where the generator t of $\mathbb{Z}$ acts on the direct sum by a shift; this group is torsion-free. Then let Γ be the semi-direct product of Γ_0 with the automorphism ϕ of Γ_0 of order 2 defined by $\phi(a_i) = -a_i$ for all $i \in \mathbb{Z}$ and $\phi(t) = t$; and let $s \in \Gamma$ denote the element of order 2 which implements ϕ on the subgroup Γ_0. For $v, v' \in \oplus_{i\in\mathbb{Z}}\mathbb{Z}a_i$, the elements sv, sv' are on the one hand of order 2; on the other hand, they are conjugate in Γ if and only if there exist $\epsilon \in \{\pm 1\}$, $k \in \mathbb{Z}$, and $w \in \oplus_{i\in\mathbb{Z}}\mathbb{Z}a_i$ such that $v' = \epsilon t^k v t^{-k} + 2w$; it follows that the conjugacy classes in Γ of $s(a_1 + \cdots + a_n)$ are pairwise distinct $(n \geq 0)$.

A. Erschler has shown that a torsion-free group can be quasi-isometric to a group having torsion of unbounded order [Ersch–b].

The main ingredient of the proof is the construction, for any finitely-generated group A, of another finitely generated group $W^\infty(A)$, using an iterated wreath product construction and an HNN-extension. On the one hand, if A, B are Lipschitz equivalent groups, then $W^\infty(A), W^\infty(B)$ are Lipschitz equivalent; on the other hand, if A is torsion-free and if B has torsion, then $W^\infty(A)$ is torsion-free and $W^\infty(B)$ has torsion of unbounded order. One example is provided by $A = \mathbb{Z}$ and $B = \mathbb{Z} \oplus (\mathbb{Z}/p\mathbb{Z})$.

IV.40, and groups quasi-isometric to abelian groups.

Some of Shalom's ideas are now available in [Shalo].

IV.41, on groups of classes of quasi-isometries.

J. Taback has studied the quasi-isometry groups of $PSL_2(\mathbb{Z}[1/p])$, for p prime. These quasi-isometry groups are all isomorphic to $PSL_2(\mathbb{Q})$, even though the groups are not quasi-isometric for different values of the prime p. For this and other results, see [Tabac–00].

IV.43, and quasi-isometries of Baumslag-Solitar groups.

For the results of K. Whyte quoted from [Whyte–a], see now [Whyte–01].

IV.46, and Lipschitz equivalence.

Here is a question of B. Bowditch. (Private communication, March 2000. See also Item 1.A' in [Gromo–93].) Consider a Penrose tiling of the plane with two prototiles D and K (dart and kite), more precisely a tiling $\mathbb{R}^2 = \bigsqcup_{j\in J} T_j$ with each T_j given together with an isometry onto either D or K. This defines a cell decomposition X of the plane, of which the 0-skeleton $X^{(0)}$ is a discrete subset of the plane.

Is $X^{(0)}$ Lipschitz equivalent to a lattice in $\mathbb{R}^2$?

IV.47.vi, on costs and ℓ^2–Betti numbers.

For a group Γ with cost $\mathcal{C}(\Gamma)$ and ℓ^2–Betti numbers $\beta_j^{(2)}(\Gamma)$, we have always

$$\mathcal{C}(\Gamma) - 1 \geq \beta_1^{(2)}(\Gamma) - \beta_0^{(2)}(\Gamma).$$

Moreover, for a large class of groups (including all groups for which both terms are known), the two terms are indeed equal. See [Gabor], in particular Corollary 3.22.

IV.50, geometric properties and weakly geometric properties.

Following [Ersch–b], it can be useful to be more precise in the terminology concerning a property $(\mathcal{P})$ of finitely generated groups. She suggests the following definitions.

Say $(\mathcal{P})$ is *geometric* if, for a pair (Γ_1, Γ_2) of finitely-generated groups which are quasi-isometric, Γ_1 has Property $(\mathcal{P})$ if and only if Γ_2 is commensurable to a group which has Property $(\mathcal{P})$.

Say $(\mathcal{P})$ is *weakly geometric* if, for a pair (Γ_1, Γ_2) of finitely-generated groups which are quasi-isometric, Γ_1 has Property $(\mathcal{P})$ if and only if Γ_2 is commensurable up to finite kernels to a group which has Property $(\mathcal{P})$.

An example of a property which is weakly geometric and which is not geometric is "being a lattice in $Spin(2,5)$"; see III.18.vi, III.18.x, and IV.42.

V.18, on the group of a remarkable simple closed curve.

It has been shown by Anna Erschler Dyubina that the group of V.18 is not finitely generated. Finding a proof is proposed as Problem 10835 in the American Mathematical Monthly [DyuHa–00].

Problem. *Let Γ be the group defined by the presentation which has an infinite sequence $b_0, b_1, b_2, \ldots$ of generators and an infinite sequence $b_1 b_0 b_1^{-1} = b_2 b_1 b_2^{-1} = b_3 b_2 b_3^{-1} = \ldots$ of relations. Show that Γ is not finitely generated.*

We would like to add a comment and our solution. The nice solution of S.M. Gagola has appeared in the *Monthly*, November 2002.

Comment. In a short paper on wild knots, R.H. Fox discovered *A remarkable simple closed curve* (Annals of Math. **50**, 1949, pages 264–265) which is almost unknotted, a fact that Fox thinks "should be obvious to anyone who has ever dropped a stitch". The fundamental group Γ of the complement of this curve in 3-space has the presentation described above.

For other fundamental groups of complements of wild knots, see [Myers–00].

Our solution. Observe first that there is a homomorphism $\Gamma \longrightarrow \mathbb{Z}$ mapping b_k onto 1 for each $k \geq 0$; hence b_k is of infinite order in Γ for each $k \geq 0$. Observe also that there is a homomorphism σ from Γ onto the symmetric group $\langle x, y \mid x^2 = y^2 = (xy)^3 = 1 \rangle$ such that $\sigma(b_{2j}) = x$ and $\sigma(b_{2j+1}) = y$ for all $j \geq 0$; hence $b_k b_{k+1} \neq b_{k+1} b_k$ for all $k \geq 0$.

For each $n \geq 0$, there is a homomorphism $\phi_n : \Gamma \longrightarrow \Gamma$ such that $\phi_n(b_k) = b_{k+n}$ for all $k \geq 0$. Since the first relation of the presentation defining Γ can be written as $b_0 = b_1^{-1} b_2 b_1 b_2^{-1} b_1$ and since the other relations do not involve b_0, the group Γ has another presentation with generators b_k and relations $b_{k+1} b_k b_{k+1}^{-1} = b_{k+2} b_{k+1} b_{k+2}^{-1}$ for $k \geq 1$. Similarly, for each $n \geq 0$, the group Γ has a presentation with generators b_k and relations $b_{k+1} b_k b_{k+1}^{-1} = b_{k+2} b_{k+1} b_{k+2}^{-1}$ for $k \geq n$, so that ϕ_n is an automorphism of Γ.

Assume now by contradiction that Γ is finitely generated, and therefore generated by $b_0, b_1, \ldots, b_{n+1}$ for some $n \geq 0$. Using again the relations $b_{k+1} b_k b_{k+1}^{-1} = b_{k+2} b_{k+1} b_{k+2}^{-1}$, this time for $0 \leq k \leq n-1$, we see that Γ is generated by $\{b_n, b_{n+1}\}$. Thus Γ is also generated by $\{b_0, b_1\} = \phi_n^{-1}(\{b_n, b_{n+1}\})$, as well as by $\{b_1, b_2\} = \phi_1(\{b_0, b_1\})$.

For each $k \geq 0$, let $\tilde{\Gamma}_{k+1}$ the group abstractly defined by $k+2$ generators $b_0, \ldots, b_{k+1}$ and k relations $b_1 b_0 b_1^{-1} = \ldots = b_{k+1} b_k b_{k+1}^{-1}$. The same argument as above shows that $\tilde{\Gamma}_{k+1}$ has another presentation with 2 generators b_k, b_{k+1} and no relation, hence that $\tilde{\Gamma}_{k+1}$ is free of rank two. As b_0, b_1 do not commute in $\tilde{\Gamma}_{k+1}$, they generate a subgroup of $\tilde{\Gamma}_{k+1}$ which is free of rank two. As this holds for any $k \geq 0$, it follows that the group Γ, generated by b_0 and b_1, is itself free of rank two.

As Γ is free on $\{b_1, b_2\}$, there is a homomorphism $\psi : \Gamma \longrightarrow \mathbb{Z}$ such that $\psi(b_1) = 0$ and $\psi(b_2) = 1$, which is *onto*. On the other hand, as Γ is generated by b_0 and b_1, and as $\psi(b_0) = \psi(b_1^{-1} b_2 b_1 b_2^{-1} b_1) = 0 = \psi(b_1)$, we have $\psi(\Gamma) = \{0\}$. This is a contradiction and ends the proof. □

The group Γ has other straightforward non-finitenes properties. (i) It is not Hopfian, since it is isomorphic to its quotient by the relation $b_0 = 1$. (ii) It maps onto the Baumslag-Solitar group $\langle t, z \mid tzt^{-1} = z^2 \rangle$ by $b_{2n} \longmapsto zt^{-1}$ and $b_{2n+1} \longmapsto t^{-1}$.

V.20, and lattices in Lie groups.

Information on lattices in *complex* Lie groups can be found in [Winke–98].

V.21, and finiteness homological properties of $SL(n, \mathbb{F}_q[T])$.

The finiteness result according to which $SL(n, \mathbb{F}_q[T])$ is of type (F_{n-2}) and not of type (F_{n-1}) for $q \geq 2^{n-2}$ is due independently to H. Abels (as recorded in V.21) and P. Abramenko [Abram–96].

V.22, on commensurability and groups of automorphisms.

G. Levitt has drawn my attention to the fact that, given a group Γ and a subgroup Γ_0 of finite index, there can exist an infinity of automorphisms of Γ which coincide with the identity on Γ_0.

Indeed, let Γ be the infinite dihedral group and let Γ_0 be its infinite cyclic subgroup of index 2. Then the conjugations of Γ by elements of Γ_0 are pairwise distinct.

V.22, on large groups of automorphisms.

The automorphism group of a finitely-generated group is clearly countable. The automorphism group of a countable group need not be; an easy example is provided by an infinite direct sum of copies of any countable group not reduced to one element.

Here is another example, inspired from Ulam and using the notation of the addendum to III.B above. For each $(a, b) \in C$, let $\phi_{a,b} : \mathbb{R} \longrightarrow \mathbb{Q}$ be the

homomorphism defined there and let $\mathbb{Q}^2_{a,b}$ be a copy of $\mathbb{Q}$. The mapping

$$\psi_{a,b} \; : \; \begin{cases} \mathbb{R} & \longmapsto \; \mathrm{Aut}\left(\mathbb{Q}^2_{a,b}\right) \approx GL_2(\mathbb{Q}) \\ x & \longmapsto \qquad \begin{pmatrix} 1 & \phi_{a,b}(x) \\ 0 & 1 \end{pmatrix} \end{cases}$$

is a homomorphism of groups. Define Γ to be the direct sum, over $(a,b) \in C$, of the groups $\mathbb{Q}^2_{a,b}$; then the direct sum of the homomorphisms $\psi_{a,b}$ is an injection of $\mathbb{R}$ into $\mathrm{Aut}(\Gamma)$.

V.26, and some groups of Richard Thompson.

There are three groups, acting respectively on an interval, the circle, and the Cantor set, denoted by F, T, and V in [CanFP–96], and which appear in many different contexts. For T in the context of Teichmüller theory, see several articles by R.C. Penner, including [Penne–97]; for the isomorphism of Penner's group with T, see [Imber–97]. One interesting byproduct of this circle of ideas is that T can be generated by two elements α, β satisfying $\alpha^4 = \beta^3 = 1$, and other relations, such that the subgroup of T generated by α^2, β is the free product $\mathbb{Z}/2\mathbb{Z}) * (\mathbb{Z}/3\mathbb{Z}) \approx PSL_2(\mathbb{Z})$; see [LocSc–97].

V.31, and efficiency.

A. Çevik gives in [Çevik–00] a sufficient condition for the efficiency of wreath products of efficient finite groups.

VI.9, an example of spherical growth series which is not monotonic.

On page 161, the last display, the coefficient of z^2 should be 8 not 6. This was pointed out to me by N.J.A. Sloane. Several growth series which appear in the the book appear also in his database of integer sequences: see

http://www.research.att.com/ njas/sequences/

on the web.

VI.19, on groups with the size of spheres not tending to infinity.

Groups in which the size of spheres does not tend to infinity are virtually cyclic (communicated by Anna Erschler Dyubina). More precisely:

Proposition. *If $\sigma(\Gamma, S; k) \leq C$ for infinitely many values of k, then Γ is virtually cyclic.*

Proof. Consider an arbitrary infinite finitely generated group, and let Φ be its inverse growth function, as in VII.32. First, it follows from the definition and from the obvious inequality $\beta(4k) > 2\beta(k)$ that $\Phi(2\beta(k)) \leq 4k$. Then, it follows from the first result quoted in VII.32 that, for an appropriate constant K, we have

$$\frac{\sigma(n)}{\beta(n-1)} \; \geq \; \frac{1}{8|S|\Phi(2\beta(n-1))} \; \geq \; \frac{1}{8|S|4(n-1)} \; \geq \; \frac{1}{Kn},$$

whence

$$\beta(n-1) \; \leq \; K\sigma(n)n$$

for all $n \geq 1$.

Assume now that $\sigma(n_j) \leq C$ for some constant C and a strictly increasing infinite sequence $(n_j)_{j\geq 1}$. Thus $\beta(n_j - 1) \leq KCn_j$ for any $j \geq 1$. By the strong form of Gromov's theorem (VII.29) on groups of polynomial growth, which is elementary for linear growth and which is due to Van den Dries and Wilkie [VdDW–84b], this implies that Γ is a group of linear growth and therefore a virtually cyclic group. □

VI.20, and the growth of braid groups for Artin generators.

For any integer $n \geq 2$, Artin's *braid group* on n strings has presentation

$$B_n = \left\langle \sigma_1, \ldots, \sigma_{n-1} \;\middle|\; \begin{array}{r} \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \quad (1 \leq i \leq n-2) \\ \sigma_i\sigma_j = \sigma_j\sigma_i \quad (1 \leq i,j \leq n-1,\ |i-j| \geq 2) \end{array} \right\rangle$$

[Magnu–73] and is obviously a quotient of the *locally free group of depth* 1 with $n-1$ generators

$$LF_n = \langle f_1, \ldots, f_{n-1} \mid f_if_j = f_jf_i \quad (1 \leq i,j \leq n-1,\ |i-j| \geq 2) \rangle$$

[Versh–90], [Versh–00], [VeNeB–00]. The value of the exponential growth rate of B_n for the generators σ_i is still unknown; however, Vershik and his co-authors have obtained partial results by comparing B_n with LF_n, more precisely by using the fact that LF_n appears both as a group of which B_n is a quotient and as a subgroup of B_n, the image of the injective homomorphism which maps f_i onto σ_i^2 for $i \in \{1, \ldots, n-1\}$.

For example, if $\omega_n^B, \omega_n^{LF}$ denote respectively the exponential growth rates of B_n, LF_n for the generators discussed here, then

$$\lim_{n\to\infty} \omega_n^{LF} = 7 \qquad \text{and} \qquad \sqrt{7} \leq \omega_n^B \leq 7 \quad \text{for } n \text{ large enough.}$$

VI.B, early papers on growth of groups, and Dye's theorem on orbit equivalence for groups of polynomial growth.

Growth occurs in a paper by Margulis [Margu–67] published one year before those of Milnor ([Miln–68a], [Miln–68b]), where Margulis shows that if a compact three-dimensional manifold admits an Anosov flow, then its fundamental group has exponential growth. For a generalization to higher dimensions, see [PlaTh–72].

Also, between the mid fifties and 1968, some mathematicians in France were aware of the notion of growth of groups. Besides Dixmier (quoted on page 187), Avez had learned this from Arnold in 1965 [Avez–76].

We should also mention the following results of H. Dye. On the one hand, consider the compact abelian group $\prod_{j=0}^{\infty} C_j$, where each C_j is a copy of the group $\{0,1\}$ of order 2, with its normalised Haar measure μ. Let $T : G \longrightarrow G$ be the adding machine, defined by

$$T(x_0, x_1, x_2, \ldots) = (0, 0, \ldots, 1, x_{j+1}, x_{j+2}, \ldots)$$

where j is the smallest index such that $x_j = 0$, and

$$T(1,1,1,1,\ldots) \;=\; (0,0,0,0,\ldots).$$

Then T defines an ergodic action of $\mathbb{Z}$ by measure preserving transformations of the probability space (G,μ). On the other hand, consider any finitely generated group Γ acting by measure preserving transformations on a standard Borel space furnished with a non-atomic probability measure, the action being ergodic.

One of Dye's theorems is that, if Γ is of polynomial growth, then the action of Γ is orbit-equivalent to the odometer action of $\mathbb{Z}$ [Dye–63]; if $\Gamma \approx \mathbb{Z}$, this is already in [Dye–59]. See [Weiss–81] for an exposition, and [OrnWe–80], [CoFeW–81] for related results; in particular, Dye's theorem carries over to *amenable* countable groups.

VI.40, and the functions which are growth functions of semigroups.

Let M be a monoid generated by a finite set S and let $\beta(k) = \beta(M,S;k)$ denote the corresponding growth function (see VI.12). It is obvious that if $\beta(k)$ is unbounded, then $k \prec \beta(k)$; moreover,

$$k \prec \beta(k) \quad \text{and} \quad k \nsim \beta(k) \qquad \text{imply} \qquad k^2 \prec \beta(k)$$

as has been shown[2] by V.V. Beljaev (reported in [Trofi–80]).

Let $f, g : \mathbb{N} \longrightarrow \mathbb{N}$ be two functions such that $k^2 \prec f(k)$ and $g(k) \prec 2^k$. Then there exists a monoid M generated by a finite set S such that the sets

$$\{k \in \mathbb{N} \,|\, \beta(M,S;k) \leq f(k)\} \qquad \text{and} \qquad \{k \in \mathbb{N} \,|\, \beta(M,S;k) \geq g(k)\}$$

are both infinite.

VI.40, and the growth functions of Riemannian manifolds.

For further work after the paper of Grimaldi and Pansu quoted in VI.40, see [GriPa–01] and its bibliography.

VI.42, and growth of groups with respect to weights.

Growth with respect to generating sets and given weights are older than suggested by the references of Chapters VI and VII. In particular, in [PlaTh–76], Plante and Thurston define the growth of a countable group (*not* necessarily of finite type) with respect to a generating set (*not* necessarily finite) and a proper weight on it.

VI.42–43 and VII.35, on relative length functions and relative growth.

In the last line of page 176, read "relative length function" instead of "relative growth function". For relative growth of subgroups of solvable and linear groups, see [Osin–00].

VI.45, on word and Riemannian metrics.

See also [LubMR–00].

[2] This has been shown independently by several other mathematicians.

VI.56, on asymptotics of subadditive functions.

The correct conclusion of (i) should be that the sequence $\left(\frac{\alpha(k)}{k}\right)_{k\geq 1}$ either converges to $\inf_{k\geq 1}\frac{\alpha(k)}{k}$ *or diverges properly to* $-\infty$. (Since sequences appearing in the book are bounded below, the second case does not occur.)

VI.64, and groups of intermediate growth which are not residually finite.

Anna Erschler has shown that there exist uncountably many groups of intermediate growth which are commensurable up to finite kernel with the first Grigorchuk group, but which are not residually finite. She has also shown that there exist groups of intermediate growth which are not commensurable up to finite kernels with any residually finite group. See [Ersch–b].

VII.2, and a version of the Table-Tennis Lemma due to Margulis.

Proposition. *Let Γ be a group acting on a set X and let $a, b \in \Gamma$. Assume that there exists a non-empty subset U of X such that $b(U) \cap U = \emptyset$ and $ab(U) \cup a^2b(U) \subset U$. Then the semi-group generated in Γ by ab and a^2b is free; in particular, it is of exponential growth if Γ is finitely generated.*

Proof. Inside $U_\emptyset \doteqdot U$, the sets $U_1 = ab(U)$ and $U_2 = a^2b(U)$ are disjoint, since

$$ab(U) \cap a^2b(U) \;=\; a\Big(b(U)\cap U_1\Big) \;\subset\; a\Big(b(U)\cap U\Big) \;=\; \emptyset\,.$$

More generally, for each $n \geq 0$, let J_n denote the set of sequences of length n with elements in $\{1,2\}$; for each $\underline{j} = (j_1, \ldots, j_n) \in J_n$, define a subset $U_{\underline{j}} = a^{j_1}ba^{j_2}b\ldots a^{j_n}b(U)$ of U. For any $n \geq 1$ and $\underline{j}' \in J_{n-1}$, observe that the sets $U_{(1,\underline{j}')}$ and $U_{(2,\underline{j}')}$ are disjoint, since

$$U_{(1,\underline{j}')} \cap U_{(2,\underline{j}')} \;=\; a\Big(b(U_{\underline{j}'})\cap U_{(1,\underline{j}')}\Big) \;\subset\; a\Big(b(U)\cap U\Big) \;=\; \emptyset\,,$$

and that both are inside $U_{\underline{j}'}$. Thus, for two sequences $\underline{j}, \underline{j}' \in \bigcup_{n=0}^{\infty} J_n$, either the corresponding subsets $U_{\underline{j}}, U_{\underline{j}'}$ are disjoint, or one is strictly contained in the other; in other words, their inclusion order is that of the infinite rooted 2-ary tree (see Item VIII.1). The proposition follows. □

This version of the Table-Tennis Lemma was communicated by G.A. Margulis to the authors of [EsMoO–02], see VII.19 below.

VII.13, on tight growth of free groups and hyperbolic groups.

It is easy to show that, for any normal subgroup $N \neq 1$ of F_k and the canonical image $\underline{S}_k$ of S_k in Γ/N, the corresponding exponential growth rates satisfy the strict inequality $\omega(F_k/N, \underline{S}_k) < 2k-1$. G. Arzhantseva and I.G. Lysenok have shown the following generalization, which answers a question of [GrHa–97]. Let Γ be a non-elementary hyperbolic group, S a finite generating set and N an infinite normal subgroup of Γ; denote by $\underline{S}$ the canonical image of S in the quotient group Γ/N; then $\omega(\Gamma/N, \underline{S}) < \omega(\Gamma, S)$ [ArjLy].

VII.19, on uniformly exponential growth of solvable groups.

D. Osin has shown that any solvable group of exponential growth has uniformly exponential growth [Osin–a], thus solving Problem VII.19.B (see page 297); this has also been shown independently and shortly afterwards by J. Wilson (unpublished). More generally, Osin has shown that any elementary amenable group of exponential growth has uniformly exponential growth [Osin–b].

Also, D. Osin has shown that the uniform Kazhdan constant of an infinite Gromov hyperbolic groups is zero [Osin–c]

John Wilson has discovered *examples of groups which answer the main problem of Item VII.19* [Wilso]. More precisely, there exist groups which are isomorphic to their permutational wreath product with the alternating group on 31 letters. Let $\Gamma \approx \Gamma \wr A_{31}$ be any group of this kind; on the one hand, there exists a sequence $(S_n = \{x_n, y_n\})_{n \geq 1}$ of generating sets of Γ, with $x_n^2 = y_n^3 = 1$ for all $n \geq 1$, such that the limit of the corresponding exponential growth rates is 1, in formula $\lim_{n \to \infty} \omega(\Gamma, S_n) = 1$; on the other hand, for an appropriate choice of Γ, there exist non-abelian free subgroups in Γ, so that in particular Γ is of exponential growth.

VII.19, on uniformly exponential growth of linear groups.

It is a theorem of A. Eskin, S. Mozes and Hee Oh that, given an integer $N \geq 1$ and a field $\mathbb{K}$ of characteristic 0, a finitely generated subgroup of $GL(N, \mathbb{K})$ is of uniformly exponential growth if and only if it is not virtually nilpotent, namely if and only if it is of exponential growth (result of [EsMoO], announced in [EsMoO–02]).

In particular, this *solves Research Problem VII.19.C* (see page 297).

For other progress on uniformly exponential growth, see [BucHa-00], [GrHa–01a], and [GrHa–01b]. For an exposition on uniformly exponential growth, see [Harpe].

If constants measuring exponential growth often have uniform bounds in terms of the generating sets, other constants exhibit the opposite behaviour. For example, T. Gelander and A. Zuk have shown that, in many cases, Kazhdan constants depend in a crucial way on the chosen generating set [GelZu–02].

VII.29, group growth, and Gromov's theorem.

There is a brief survey on group growth and Gromov's theorem by D.L. Johnson [Johns–00].

Concerning polynomial growth for locally compact groups, V. Losert has published a second part to [Loser–87]: see [Loser–01].

VII.29, and growth of double coset classes.

Consider a *Hecke pair* (G, H), namely a group G and a subgroup H such that all orbits of the natural action of H on G/H are finite, or equivalently such that, for each $g \in G$, the indices of $H \cap gHg^{-1}$ in both H and gHg^{-1} are finite. It is a natural counting problem to estimate for each $g \in G$ the cardinality of the

H-orbit of gH in G/H, or equivalently the number of one-sided classes g_jH in the double class HgH.

The specific case of the pair $(SL(2,\mathbb{Z}[1/p]), SL(2,\mathbb{Z}))$, p a prime, appears in [BeCuH–02].

VII.34, and the growth of Følner sequences.

A question related to our Problem VII.34.A appears as Problem 14.27 in the Kourovka Notebook [Kouro–95], and has been answered in [Barda–01].

VII.38, and the growth of normal subgroups of finite index.

See [LarLu].

VII.39, growth of conjugacy classes, and growth of pseudogroups.

For growth of conjugacy classes in hyperbolic groups, see [CooKn–02] and [CooKn–b].

Growth of pseudogroups appears in connection with foliations in [Plant–75].

VII.40, and growth of infinitely generated groups.

See [PlaTh–76], and the above comment on Item VI.42.

VII.61, on the set of exponential growth rates.

Part of the problem was solved by Anna Erschler Dyubina, who has shown that *the set* Ω_2 *of exponential growth rates of* 2*-generated groups has the power of the continuum* (see [Ersch–02], [Ersch–a]).

VIII.7, and the adding machine.

The adding machine on the infinite 2-ary tree $\mathcal{T}^{(2)}$ can be economically (and recursively, compare VIII.9) defined as the element $\tau \in Aut(\mathcal{T}^{(2)})$ such that

$$\tau \ = \ a(1\ ,\ \tau).$$

Observe that $\tau \neq 1$ since τ exchanges 0 and 1, and that τ is of infinite order since

$$\tau^2 \ = \ a(1,\tau)a(1,\tau) \ = \ (\tau,1)(1,\tau) \ = \ (\tau,\tau).$$

The simple and clever Proposition 20 of [Sidki–00] shows that an element $g \in Aut(\mathcal{T}^{(2)})$ is conjugate to τ if and only if it acts transitively on the set of 2^k vertices of the level $L^{(k)}$ for each $k \geq 0$.

Later, Sidki has shown that a solvable subgroup K of $Aut(\mathcal{T}^{(2)})$ which contains an element such as τ above is an extension of a torsion-free metabelian group by a finite 2-group. If furthermore K is nilpotent then it is torsion-free abelian [Sidki].

VIII.10.ii on automata and finitely generated groups.

This connexion is a very active subject of research; see among others [GriNS–00], [GriZu–a], [GriZu–b], and [Sidki-00].

VIII.31, a result of John Wilson.

At the end of this item, the "recent result" which is quoted was in fact essentially in John Wilson's Ph.D. thesis of 1971, as well as in [Wilso–72]. ("Essentially" in the sense that he did not use the words "branch groups".)

For these, [Grigo] contains comments and a sketchy proof, whereas details can be found in [Wilso].

VIII.32 and VIII.71, and elements of small lengths and large orders in the Grigorchuk group.

Proposition. *For any $n \in \mathbb{N}$, there exists $\gamma \in \Gamma$ such that*

$$\gamma^{2^n} \neq 1 \qquad \text{and} \qquad \ell(\gamma) \leq 2^n.$$

Proof (following a sketch of L. Bartholdi). Let $K = \langle abab \rangle^\Gamma$ be the normal subgroup of Γ of index 16 defined in VIII.30; recall that K is generated by

$$t = (ab)^2 \ , \quad v = (bada)^2 \ , \quad w = (abad)^2$$

and that $\psi^{-1}(K \times K)$ is a subgroup of K (the index is 4 by Exercise VIII.81, but we do not use this here). Let σ be the endomorphism of Γ defined in VIII.57. Since $\sigma(a) = aca$, $\sigma(b) = d$, $\sigma(d) = c$, and $\sigma(c) = b$, we have $\psi\sigma(a) = (d, a)$, $\psi\sigma(b) = (1, b)$, $\psi\sigma(c) = (a, c)$, $\psi\sigma(d) = (a, d)$. It follows that $\psi\sigma(x) = (1, x)$ for $x \in \{t, v, w\}$, and therefore for all $x \in K$.

Define inductively a sequence $(x_i)_{i \geq 0}$ by $x_0 = abab$ and $x_{i+1} = a\sigma(x_i)$. Since

$$\psi\left(a\sigma(x_i)a\sigma(x_i)\right) = (x_i, x_i),$$

the order of x_{i+1} is twice that of x_i. As x_0 is of order 8 by Proposition VIII.16, it follows that the order of x_i is 2^{i+3} for all $i \geq 0$.

On the other hand, denote by w_0 the word $abab$ representing x_0; for each $i \geq 0$, let w_{i+1} the word obtained from w_i by

- substitution of aca, d, b, c in place of a, b, c, d respectively,
- deletion of a if it appears as the first letter and addition of a as a prefix letter otherwise,

so that w_{i+1} represents x_{i+1}. Thus $w_1 = cadacad$ and, for each $j \geq 0$,

- w_{2j+1} is a word of length $2\ell\left(w_{2j}\right) - 1$ which begins with c and ends with a letter from $\{b, c, d\}$,
- w_{2j+2} is a word of length $2\ell\left(w_{2j+1}\right)$ which begins with a and ends with a letter from $\{b, c, d\}$;

in particular, $\ell(x_i) \leq \ell(w_i) < 2^{i+2}$ for all $i > 0$. The proposition follows (with $x = x_{n-2}$ for $n \geq 2$). □

VIII.67, and power series with finitely many different coefficients.

Here is a result of Szegö: a power series with finitely many different coefficients that converges inside the unit disk is either a rational function, or has the unit circle as natural boundary [Szegö–22].

VIII.88, complement on commensurability of finitely-generated subgroups.

It is a remarkable result of Grigorchuk and Wilson that any infinite finitely-generated subgroup of the Grigorchuk group Γ is commensurable to Γ [GriWi]. In other words, Γ has exactly two commensurability classes of finitely-generated subgroups: itself and $\{1\}$.

Here are a few examples of other groups for which all commensurability classes of finitely-generated subgroups are known; in case of torsion-free groups, we do not list the class of $\{1\}$.

(i) Free abelian groups $\mathbb{Z}^n$, with $\mathbb{Z}^j$ for $j \in \{1, \dots, n\}$.

(iii) The Heisenberg group $\begin{pmatrix} 1 & \mathbb{Z} & \mathbb{Z} \\ 0 & 1 & \mathbb{Z} \\ 0 & 0 & 1 \end{pmatrix}$, with $\mathbb{Z}$, $\mathbb{Z}^2$ and the group itself.

(iii) Non-abelian free groups F_n, with $\mathbb{Z}$ and F_2.

(iv) Virtually free groups, for example $PSL(2, \mathbb{Z})$, with finite subgroups, $\mathbb{Z}$ and F_2.

(v) The fundamental group Γ_g of a closed surface of genus $g \geq 2$, with $\mathbb{Z}$, F_2 and the group itself.

(vi) Olshanskii's "monsters" (see the reference in III.5, as well as [AdyLy–92]), in which any proper subgroup is cyclic.

VIII.87, on complex linear representations of the Grigorchuk group.

For each $k \geq 0$, let Γ_k denote as in VIII.35 the finite quotient of the Grigorchuk group which acts naturally on the level $L(k)$ of the binary tree. Choose some point in $L(k)$ and denote by P_k the corresponding isotropy subgroup of Γ_k. Then (Γ_k, P_k) is a *Gelfand pair,* and the natural linear representation of Γ_k on the space $\mathbb{C}^{L(k)}$ splits as a direct sum of $k+1$ pairwise inequivalent irreducible representations, of dimensions $1, 1, 2, 4, \dots, 2^{k-1}$ [BeHaG].

References

Abram–96. P. Abramenko, *Twin buildings and applications to S-arithmetic groups*, Lecture Notes in Math. **1641**, Springer, 1996.

AdyLy–92. S.I. Adyan and I.G. Lysënok, *Groups, all of whose proper subgroups are finite cyclic*, Math. USSR Izvestiya **39** (1992), 905–957.

AlFaN. R.C. Alperin, B. Farb, and G.A. Noskov, *A strong Schottky Lemma for non-positively curved singular spaces*, Preprint (January, 2001).

AnoSi–67. D.V. Anosov and Ya.G. Sinai, *Some smooth ergodic systems*, Russian Math. Surveys **22:5** (1967), 103–167.

ArzLy–02. G. Arzhantseva and I.G. Lysenok, *Growth tightness for word hyperbolic groups*, Math. Z. **241** (2002), 597–611.

Barda–01. V.G. Bardakov, *Construction of a regularly exhausting sequence for groups with subexponential growth*, Algebra i Logica **40** (2001), 22–29.

BarGr–00. L. Bartholdi and R. Grigorchuk, *Spectra of non-commutative dynamical systems and graphs related to fractal groups*, C.R. Acad. Sci. Paris, Série I **331** (2000), 429–434.

BarGr–01. L. Bartholdi and R. Grigorchuk, *Sous-groupes paraboliques et représentations de groupes branchés*, C.R. Acad. Sci. Paris, Série I **332** (2001), 789–794.

BasLu–01. H. Bass and A. Lubotzky, with appendices by H. Bass, L. Carbone, A. Lubotzky, G. Rosenberg, and J. Tits, *Tree lattices*, Birkhäuser, 2001.

BeCuH–02. M.B. Bekka, R. Curtis, and P. de la Harpe, *Familles de graphes expanseurs et paires de Hecke*, C.R. Acad. Sci. Paris, Série I **335** (2002), 463–468.

BeHaG. M.B. Bekka, P. de la Harpe, *Irreducibility of unitary group representations and reproducing kernels Hilbert spaces*, Appendix on *Two point homogeneous compact ultrametric spaces* in collaboration with Rostislav Grigorchuk, Expositiones Math. (to appear).

BesFe–00. M. Bestvina and M. Feighn, *The topology at infinity of* $Out(F_n)$, Inventiones Math. **146** (2000), 651–692.

BreGe. E. Breuillard and T. Gelander, *On dense free subgroups of Lie groups*, Preprint (2002).

Camin–97. R. Camina, *Subgroups of the Nottingham group*, J. of Algebra **196** (1997), 101-113.

Camin–00. R. Camina, *The Nottingham group*, in "New horizons in pro-p groups", M. du Sautoy, D. Segal, and A. Shalev Editors, Birkhäuser (2000), 205–221.

Carbo–01. L. Carbone, *Non-uniform lattices on uniform trees*, Memoir Amer. Math. Soc. **724**, 2001.

Çevik–00. A. S. Çevik, *The efficiency of standard wreath product*, Proc. Edinburgh Math. Soc. **43** (2000), 415–423.

ChaPi–01. B. Chaluleau and C. Pittet, *Exemples de variétés riemanniennes homogènes qui ne sont pas quasi isométriques à un groupe de type fini*, C.R. Acad. Sci. Paris, Sér. I **332** (2001), 593–595.

CoFeW–81. A. Connes, J. Feldman, and B. Weiss, *An amenable equivalence relation is generated by a single transformation*, Ergod. Th. & Dynam. Sys. **1** (1981), 431–450.

CooKn–02. M. Coornaert and G. Knieper, *Growth of conjugacy classes in Gromov hyperbolic groups*, Geometric and Functional Analysis **12** (2002), 464–478.

CooKn–b. M. Coornaert and G. Knieper, *An upper bound for the growth of conjugacy classes in torsionfree word hyperbolic groups*, to appear.

Drumm–92. T.A. Drumm, *Fundamental polyhedra for Margulis space-times*, Topology **31** (1992), 677–683.

Dye–59. H. Dye, *On groups of measure preserving transformations I*, Amer. J. Math. **81** (1959), 119–159.

Dye–63. H. Dye, *On groups of measure preserving transformations II*, Amer. J. Math. **85** (1963), 551–576.

DymMc–72. H. Dym and H.P. McKean, *Fourier series and integrals*, Academic Press, 1972.

Dyubi–00. A. Dyubina, *Instability of the virtual solvability and the property of being virtually torsion-free for quasi-isometric groups*, International Math. Res. Notices **21** (2000), 1098–1101.

DyuHa–00. A. Dyubina Erschler and P. de la Harpe, *Problem 108 35*, Amer. Math. Monthly $\mathbf{107}^9$ (2000), 864.

Egoro–00. A.V. Egorov, *Residual finiteness of groups and topological dynamics*, Sbornik Math. $\mathbf{191}^4$ (2000), 529–541.

Ersch–02. A. Erschler, *On growth rates of small cancellation groups*, Funct. Analy and its Appl. **36** (2002), 93–95.

Ersch–a. A. Erschler, *Growth rates of small cancellation groups*, Proceedings of the workshop Random Walks and Geometry, Vienna, ESI (to appear).

Ersch–b. A. Erschler, *Not residually finite groups of intermediate growth, commensurability and non geometricity*, Preprint (2002).

EsMoO–02. A. Eskin, S. Mozes and Hee Oh, *Uniform exponential growth for linear groups*, International Math. Res. Notices **2002:31** (2002), 1675–1683.

EsMoO. A. Eskin, S. Mozes and Hee Oh, *On uniform exponential growth for linear groups*, Preprint (2002).

FarFr. B. Farb and J. Franks, *Groups of homeomorphisms of one-manifolds I: actions of nonlinear groups*, Preprint (2001).

Gabor–02. D. Gaboriau, *Arbres, groupes, quotients*, Thèse d'habilitation, ENS-Lyon, 8 avril 2002.

Gabor. D. Gaboriau, *Invariants ℓ^2 de relations d'équivalence et de groupes*, Publ. Math. I.H.E.S. (to appear).

GelZu–02. T. Gelander and A. Zuk, *Dependence of Kazhdan constants on generating subsets*, Israel J. Math. **129** (2002), 93–98.

Ghys–01. E. Ghys, *Groups acting on the circle*, l'Enseignement math. (2) **47** (2001), 329–407.

GorOn–93. V.V. Gorbatsevich and A.L. Onishchik, *Lie transformation groups*, in "Lie groups and Lie algebras I", Encycl. Math. Sciences **20**, Springer (1993), 95–229.

GrHa–97. R.I. Grigorchuk and P. de la Harpe, *On problems related to growth, entropy and spectrum in group theory*, J. of Dynamical and Control Systems **3:1** (1997), 51–89.

GrHa–01a. R.I. Grigorchuk and P. de la Harpe, *One-relator groups of exponential growth have uniformly exponential growth*, Math. Notes **69** (2001), 575–577.

GrHa–01b. R.I. Grigorchuk and P. de la Harpe, *Limit behaviour of exponential growth rates for finitely generated groups*, l'Enseignement math., monographie **38**^{2} (2001), 351–370.

GriNS–00. R.I. Grigorchuk, V.V. Nekrashevich, and V.I. Sushchanskii, *Automata, dynamical systems, and groups*, Proc. Steklov Inst. Math. **231** (2000), 128–203.

GriPa–01. R. Grimaldi and P. Pansu, *Nombre de singularités de la fonction croissance en dimension* 2, Bull. Belgian Math. Soc. **8** (2001), 395–404.

GriWi. R.I. Grigorchuk and J. Wilson, *A rigidity property concerning abstract commensurability of subgroups*, Preprint, 2001.

GriZu–a. R. Grigorchuk and A. Zuk, *The lamplighter group as a group generated by a 2-state automaton and its spectrum*, Geometriae Dedicata, to appear.

GriZu–b. R. Grigorchuk and A. Zuk, *A free group generated by a three state automaton*, Internat. J. Algebra Comput., to appear.

GryWó–99. A. Grytczuk and M. Wójtowicz, *Beardon's diophantine equations and non-free Möbius groups*, Bull. London Math. Soc. **32** (1999), 305–310.

Harpe. P. de la Harpe, *Uniform growth in groups of exponential growth*, Geometriae Dedicata, to appear.

Imber–97. M. Imbert, *Sur l'isomorphisme du groupe de Richard Thompson avec le groupe de Ptolémée*, in "Geometric Galois Actions, 2", L. Schneps and P. Lochak Editors, London Math. Soc. Lecture Notes Series **243** (Cambridge Univ. Press 1997), 313–324.

Johns–00. D.L. Johnson, *Growth of groups*, The Arabian Journal for Science and Engineering **25–2C** (2000), 53–68.

Karls. A. Karlsson, *Free subgroups of groups with non-trivial Floyd boundary*, Preprint (January 2002).

LarLu. M. Larsen and A. Lubotzky, *Normal subgroup growth of linear groups: the* (G_2, F_4, E_8)*-theorem*, Prepublication (2001).

Licht–93. A.I. Lichtman, *The soluble subgroups and the Tits alternative in linear groups over rings of fractions of polycyclic group, I*, J. of Pure and Appl. Algebra **86** (1993), 231–287.

Licht–95. A.I. Lichtman, *Automorphism groups of free soluble groups*, J. Algebra **174** (1995), 132–149.

Licht–99. A.I. Lichtman, *The soluble subgroups and the Tits alternative in linear groups over rings of fractions of polycyclic group, II*, J. Group Theory **2** (1999), 173–189.

LisMe–00. V. Liskovets and A. Mednykh, *Enumeration of subgroups in the fundamental groups of orientable circle bundles over surfaces*, Comm. in Algebra **28**^{4} (2000), 1717–1738.

LocSc–97. P. Lochak et L. Schneps, *The universal Ptolemy-Teichmüller groupoid*, in "Geometric Galois Actions, 2", L. Schneps and P. Lochak Editors, London Math. Soc. Lecture Notes Series **243** (Cambridge Univ. Press 1997), 325–347.

Loser–01. V. Losert, *On the structure of groups with polynomial growth II*, Journal London Math. Soc. **63** (2001), 640–654.

LubMR–00. A. Lubotzky, S. Mozes, and M.S. Raghunathan, *The word and Riemannian metrics on lattices of semisimple groups*, Publ. Math. I.H.E.S. **91** (2000), 5–53.

MacMa–01. C. Maclachlan and G.J. Martin, *The non-compact arithmetic generalized triangle groups*, Topology **40** (2001), 927-944.

Magnu–74. W. Magnus, *Braid groups: a survey*, in "Proceedings of the Second International Conference on the Theory of Groups (Australian Nat. Univ., Canberra, 1973)", Lecture Notes in Math. **372** (Springer, 1974), 463–487.

MalSz. W. Malfait and A. Szczepański, *The structure of the (outer) automorphism group of a Bieberbach group*, Preprint (2002).

Margu–67. G.A. Margulis, *Y-flows and three-dimensional manifolds (Appendix to* [AnoSi–67]*)*, Russian Math. Surveys **22:5** (1967), 164–166.

Margu–83. G.A. Margulis, *Free completely discontinuous groups of affine transformations*, Dokl. Akad. Nauk SSSR **272** (1983), 785–788.

Margu–00. G.A. Margulis, *Free subgroups of the homeomorphism group of the circle*, C.R. Acad. Sci. Paris, Série I **331** (2000), 669–674.

Myers–00. R. Myers, *Uncountably many arcs in* $\mathbb{S}^3$ *whose complements have non-isomorphic, indecomposable fundamental groups*, J. Knot Theory Ramifications **9** (2000), 505–521.

Navas. A. Navas, *Sur les groupes de difféomorphismes du cercle engendrés par des éléments proches des rotations*, Preprint (2002).

NosVi–02. G.A. Noskov and E.B. Vinberg, *Strong Tits alternative for subgroups of Coxeter groups*, J. Lie Theory **12** (2002), 259–264.

Ohshi–02. K. Ohshika, *Discrete groups*, Translations of mathematical monographs **207**, Amer. Math. Soc., 2002.

OrnWe–80. D.S. Ornstein and B. Weiss, *Ergodic theory of amenable group actions. I: the Rohlin lemma*, Bull. Amer. Math. Soc. **2** (1980), 161–164.

Osin–00. D. Osin, *Problem of intermediate relative growth of subgroups in solvable and linear groups*, Proc. Steklov Inst. Math. **231** (2000), 316–338.

Osin–01. D. Osin, *subgroup distortions in nilpotent groups*, Comm. in Alg. **29:12** (2001), 5439–5464.

Osin–a. D. Osin, *The entropy of solvable groups*, Ergod. Th. & Dynam. Sys., to appear.

Osin–b. D. Osin, *Algebraic entropy and amenability of groups*, Preprint (June, 2001).

Osin–c. D. Osin, *Kazhdan constants of hyperbolic groups*, Preprint (November, 2001).

Ozawa. N. Ozawa, *There is no separable universal* II_1*-factor*, Preprint (November 2002).

Penne–97. R. Penner, *The universal Ptolemy group and its completions*, in "Geometric Galois Actions, 2", L. Schneps and P. Lochak Editors, London Math. Soc. Lecture Notes Series **243** (Cambridge Univ. Press 1997), 293–312.

Pervo–00. E.L. Pervova, *Everywhere dense subgroups of one group of tree automorphisms*, Proc. Steklov Inst. Math. **231** (2000), 339–350.

Petti–52. B.J. Pettis, *A note on everywhere dense subgroups*, Proc. Amer. Math. Soc. **3** (1952), 322–326.

Plant–75. J.P. Plante, *Foliations with measure preserving holonomy*, Annals of Math. (2) **102** (1975), 327–361.

PlaTh–72. J.P. Plante and W.P. Thurston, *Anosov flows and the fundamental group*, Topology **11** (1972), 147–150.

PlaTh–76. J.P. Plante and W.P. Thurston, *Polynomial growth in holonomy groups of foliations*, Comment. Math. Helv. **51** (1976), 567–584.

Salof–01. L. Saloff-Coste, *Probability on groups: random walks and invariant diffusion*, Notices of the AMS **48:9** (October 2001), 968–977.

Samue–66. P. Samuel, *A propos du théorème des unités*, Bull. Sci. math. **90** (1966), 89–96.

Samue–67. P. Samuel, *Théorie algébrique des nombres*, Hermann, 1967.

SchUl–33. J. Schreier and S. Ulam, *Über die Permutationsgruppe der natürlichen Zahlenfolge*, Studia Math. **4** (1933), 134–141.

Semen. Y. Semenov, *On the rational forms of nilpotent Lie algebras and lattices in nilpotent Lie groups*, l'Enseignement math. (to appear).

Shale–01. A. Shalev, *Asymptotic group theory*, Notices of the Amer. Math. Soc. **48**[4] (April 2001), 383–389.

Shalo. Y. Shalom, *Harmonic analysis, cohomology, and the large scale geometry of amenable groups*, Preprint (2002).

Sidki–00. S. Sidki, *Automorphisms of one-rooted trees: growth, circuit structure, and acyclicity*, J. Math. Sci. (New York) **100** (2000), 1925–1943.

Sidki. S. Sidki, *The binary adding machine and solvable groups*, Preprint (2001).

SoiVe-00. G.A. Soifer and T.N. Venkataramana, *Finitely generated profinitely dense free groups in higher rank semi-simple groups*, Transf. Groups **5** (2000), 93–100.

Souch. E. Souche, *Quasi-isométries et quasi-plans dans l'étude des groupes discrets*, Ph.D. Thesis, Marseille (2001).

Szegö–22. G. Szegö, *Über Potenzreihen mit endlich vielen verschiedenen Koeffizienten*, Sitzungberichte der Preussischen Akademie der Wissenschaften, Phys.-Math. Klasse (1922), 88–91 [Collected Papers, Vol. 1, pages 667–561].

Tabac–00. J. Taback, *Quasi-isometric rigidity for* $PSL_2(\mathbb{Z}[1/p])$, Duke Math. J. **101** (2001), 335–357.

Trofi–80. V.I. Trofimov, *The growth functions of finitely generated semigroups*, Semigroup Forum **21** (1980), 351–360.

Ulam–60. S.M. Ulam, *A Collection of mathematical problems*, Interscience, 1960 [See also S. Ulam, *Sets, numbers, and universes – Selected works*, W.A. Beyer, J. Mycielski and G.-C. Rota Editors, MIT Pres, 1974, pages 503–670].

VdDW–84b. L. van den Dries and A.J. Wilkie, *An effective bound for groups of linear growth*, Arch. Math. **42** (1984), 391–396.

VeNeB–00. A.M. Vershik, S. Nechaev, and R. Bikbov, *Statistical properties of locally free groups with applications to braid groups and growth of random heaps*, Commun. Math. Phys. **212** (2000), 469–501.

Versh–90. A.M. Vershik, *Local algebras and a new version of Young's orthogonal form*, in "Topics in algebra", Banach Center Publications **26, 2** (PWN, 1990), 467–473.

Versh–00. A.M. Vershik, *Dynamic theory of growth in groups: entropy, boundaries, examples*, Russian Math. Surveys **55:4** (2000), 667–753.

Weiss–81. B. Weiss, *Orbit equivalence of nonsingular actions*, Monographie de l'Enseignement mathématique **29** (1981), 77–107.

Whyte–01. K. Whyte, *The large scale geometry of the higher Baumslag-Solitar groups*, GAFA Geom. Funct. Anal. **11** (2001), 1327–1343.

Wilso–72. J. Wilson, *Groups with every proper quotient finite*, Math. Proc. Camb. Phil. Soc. **69** (1972), 373–391.

Wilso. J.S. Wilson, *On exponential growth and unformly exponential growth of groups*, Preprint (2002).

Winke–98. J. Winkelmann, *Complex analytic geometry of complex parallelizable manifolds*, Mémoire **72–73** , Soc. Math. France, 1998.

Woess–93. W. Woess, *Fixed sets and free subgroups of groups acting on metric spaces*, Math. Zeit. **214** (1993), 425-440.

The following references, firstly quoted as preprints, have now appeared.

BacVd. R. Bacher and A. Vdovina, *Counting* 1*-vertex triangulations of oriented surfaces*, Discrete Math. **246** (2002), 13–27.

Bambe. J. Bamberg, *Non-free points for groups generated by a pair of* 2×2 *matrices*, J. London Math. Soc. (2) **62** (2000), 795–801.

BarCe–b. L. Bartholdi and T.G. Ceccherini-Silberstein, *Salem numbers and growth series of some hyperbolic graphs*, Geometriae Dedicata **90** (2002), 107–114.

BarGr–a. L. Bartholdi and R. Grigorchuk, *Lie methods in growth of groups and groups of finite width*, in "Computational and geometric aspects of modern algebra (Edinburgh, 1998)", N. Gilbert, Editor, London Math. Soc. Lecture Note Ser. **275**, Cambridge Univ. Press (2000), 1–27.

BarGr–c. L. Bartholdi and R. Grigorchuk, *On the spectrum of Hecke type operators related to some fractal groups*, Proc. Steklov Inst. Math. **231** (2000), 1–41.

Barth. L. Bartholdi, *Lower bounds on the growth of a group acting on the binary rooted tree*, Internat. J. Algebra Comput. **11** (2001), 73–88.

Bavar. C. Bavard, *Classes minimales de réseaux et rétractions géométriques équivariantes dans les espaces symétriques*, J. London Math. Soc. **64** (2001), 275–286.

BekMa. B. Bekka and M. Mayer, *Ergodic theory and topological dynamics of group actions on homogeneous spaces*, London Math. Soc. Lecture Note Ser. **269**, Cambridge University Press, 2000.

BesFH–a. M. Bestvina, M. Feighn and M. Handel, *The Tits alternative for $Out(F_n)$ I: Dynamics of exponentially growing automorphisms*, Annals of Math. (2) **151** (2000), 517–623.

Bigel. S. Bigelow, *Braid groups are linear*, J. Amer. Math. Soc. **14** (2001), 471–486.

BonSc. M. Bonk and O. Schramm, *Embeddings of Gromov hyperbolic spaces*, GAFA Geom. Funct. Anal. **10** (2000), 266-306.

BruSi. A.M. Brunner and S. Sidki, *The generation of $GL(n,\mathbb{Z})$ by finite state automata*, Internat. J. Algebra Comput. **8** (1998), 127–139.

BucHa. M. Bucher and P. de la Harpe, *Free products with amalgamation and HNN-extensions of uniformly exponential growth*, Mathematical Notes **67** (2000), 686–689.

BuxGo. K-U. Bux and C. Gonzalez, *The Bestvina-Brady construction revisited — geometric computation of Σ-invariants for right angled Artin groups*, Journal London Math. Soc. **60** (1999), 793–801.

CanCo. J.W. Cannon and G.R. Conner, *The combinatorial structure of the Hawaiian earring group*, Topology and its appl. **106** (2000), 225–271.

CecMS. T. Ceccherini-Silberstein, A. Machì and F. Scarabotti, *Il gruppo di Grigorchuk di crescita intermedia*, Rend. Circ. Mat. Palermo (2) **50** (2001), 67–102.

Champ. C. Champetier, *L'espace des groupes de type fini*, Topology **39** (2000), 657–680.

FarMo. B. Farb and L. Mosher, *On the asymptotic geometry of abelian-by-cyclic groups*, Acta Math. **184** (2000), 145–202.

Grigo. R.I. Grigorchuk, *Just infinite branch groups*, in "New horizons in pro-p groups", M. du Sautoy, D. Segal, and A. Shalev Editors, Birkhäuser (2000), 121–179.

Jones. V.F.R. Jones, *Ten problems*, in "Mathematics: frontiers and perspectives", V. Arnold, M. Atiyah, P. Lax, and B. Mazur Editors, Amer. Math. Soc. (2000), 79–91.

Kramm–a. D. Krammer, *The braid group B_4 is linear*, Inventiones Math. **142** (2000), 451–486.

Lamy. S. Lamy, *L'alternative de Tits pour $Aut\left[\mathbb{C}^2\right]$*, J. of Algebra **239** (2001), 413–437.

Ledra. F. Ledrappier, *Some asymptotic properties of random walks on free groups*, in "Topics in probability and Lie groups: boundary theory", J.C. Taylor Editor, CRM Proceedings and Lecture Notes **28** (Amer. Math. Soc. 2001), 117–152.

Leono–01. Yu.G. Leonov, *A lower bound for the growth of a 3-generator 2-group*, Sbornik Math. **192:11** (2001), 1661–1676.

LucTW. A. Lucchini, M.C. Tamburini, and J.S. Wilson, *Hurwitz groups of large rank*, J. London Math. Soc. **61** (2000), 81–92.

MarVi. G.A. Margulis and E.B. Vinberg, *Some linear groups virtually having a free quotient*, J. Lie Theory **10** (2000), 171–180.

Nekra. V. Nekrashevych, *On equivalence of nets in hyperbolic spaces*, Dopov. Nats. Akad. Nauk Ukr. Mat. Prirodozn. Tekh. Nauki **11** (1997), 18–21.

Osin. D. Osin, *Subgroup distortions in nilpotent groups*, Comm. in Algebra **29**^{12} (2001), 5439–5463.

PapWh. P. Papasoglu and K. Whyte, *Quasi-isometries between groups with infinitely many ends*, Comment. Math. Helvetici **77** (2002), 1343–144.

Pauli. F. Paulin, *Un groupe hyperbolique est déterminé par son bord*, J. London Math. Soc., to appear.

Pitte. C. Pittet, *The isoperimetric profile of homogeneous Riemannian manifolds*, J. Differential Geom. **54** (2000), 255–302.

Shalo–a. Y. Shalom, *Explicit Kazhdan constants for representations of semisimple and arithmetic groups*, Ann. Inst. Fourier **50** (2000), 833–863.

Shalo–b. Y. Shalom, *Bounded generation and Kazhdan's property (T)*, Publ. Math. I.H.E.S. **90** (1999), 145–168.

Wilso. J.S. Wilson, *On just infinite abstract and profinite groups*, in "New horizons in pro-p groups", M. du Sautoy, D. Segal, and A. Shalev Editors, Birkhäuser (2000), 181–203.

Woess. W. Woess, *Random walks on infinite graphs and groups*, Cambridge Tracts in Mathematics **138**, Cambridge University Press, 2000.

Chain Lengths in the Dominance Lattice

Edward Early*

November 20, 2002

Abstract

We find the size of the largest union of two chains in the lattice of partitions of n under dominance order. We also present some partial results and conjectures on chains and antichains in this lattice.

Nous trouvons la taille de la plus grande union de deux chaînes dans le treillis des partitions de n sous l'ordre partiel dominant. Nous présentons aussi des résultats partiels et des conjectures sur les chaînes et antichaînes de ce treillis.

1 Introduction

Let P_n denote the poset of partitions of the positive integer n, ordered by dominance (aka majorization), i.e. $\lambda \leq \mu$ if $\lambda_1+\lambda_2\cdots+\lambda_k \leq \mu_1+\mu_2+\cdots+\mu_k$ for all k. This poset is a lattice, and is self-dual under conjugation. P_n is not graded for $n \geq 7$, since there exist saturated chains from $\{n\}$ to $\{1^n\}$ of all lengths from $2n-3$ to $cn^{3/2}$ [2, 5].

Given any poset P, there exists a partition $\lambda(P)$ such that the sum of the first k parts of λ is the maximal number of elements in a union of k chains in P. In fact, the conjugate of λ has the same property with chains replaced by antichains [1, 3, 4]. Let $\lambda_k(P)$ denote the kth part of this partition.

The length $h(P_n)$ of the longest chain in P_n has been known for some time [5]. If $n = \binom{m+1}{2} + r$, $0 \leq r \leq m$, then $h(P_n) = \frac{m^3-m}{3} + rm$. In other words, $\lambda_1(P_n) = \frac{m^3-m}{3} + rm + 1$. Our main result is the following theorem.

Theorem 1. *For $n > 16$, $\lambda_2(P_n) = \lambda_1(P_n) - 6$.*

*This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

Consider the subposet Q_n of P_n consisting of the partitions that appear in chains of length $h(P_n)$. Clearly Q_n is self-dual under conjugation, since conjugation takes a decreasing chain to an increasing chain of the same length. It seems likely that Q_n is a graded lattice, but for our purposes it will suffice to use a weaker statement, namely: for $\lambda \in Q_n$, define $r(\lambda)$ to be the length of the longest chain from $\{n\}$ to λ; then $\lambda \neq \{1^n\}, \{n\}$ is covered by an element μ such that $r(\mu) = r(\lambda) - 1$ and covers an element ν such that $r(\nu) = r(\lambda) + 1$. In other words, every element of Q_n is on a fixed level. Figure 1 shows an example of a poset Q with this property that is not graded. Note that the top element is level 0, and the levels increase as we move down.

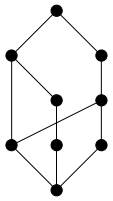


Figure 1: A non-graded poset with well-defined levels.

The covering relation in P_n comes in two flavors. Following the methods of [5], we represent Ferrers diagrams with vertical parts, as illustrated in Figure 2. We say λ covers μ by an H-step if there exists i such that $\mu_i = \lambda_i - 1$, $\mu_{i+1} = \lambda_{i+1} + 1$, and $\mu_k = \lambda_k$ for $k \neq i, i+1$. In terms of Ferrers diagrams, this corresponds to moving a box horizontally one space to the right (and down some distance). The other flavor is a V-step, which is an H-step on the conjugate, and corresponds to moving a box vertically one space down (and right some distance). Chains from $\{n\}$ to $\{1^n\}$ consisting of H-steps followed by V-steps are maximal.

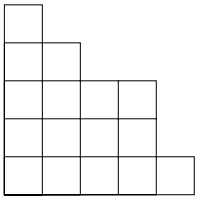


Figure 2: The partition $\{5, 4, 3, 3, 1\}$.

2 Down to work

The cases where $n \leq 16$ will be handled separately, so for now assume $n > 16$.

We will prove Theorem 1 by showing that there exist two disjoint chains in Q_n of lengths $h(P_n)$ and $h(P_n)-6$. Since Q_n is a subposet of P_n, these are also chains in P_n. Since there are six elements of P_n in saturated antichains of size 1, this is clearly the maximum possible number of elements in two chains, thus giving $\lambda_2(P_n)$ exactly.

To that end, we seek two disjoint chains in Q_n from $\{n-2,1,1\}$ and $\{n-3,3\}$ to $\{2,2,2,1^{n-6}\}$ and $\{3,1^{n-3}\}$. Let Q_n^* denote Q_n without the top three and bottom three elements.

Lemma 1. *If Q_n^* has at least two elements on every level, then it has two disjoint chains of maximal length.*

Proof: Clearly we can start two chains with the two elements in the top level, so proceed by induction. The only potential problem is if we reach two elements on level k that both cover only one and the same element on level $k+1$. In that case, take a second element on level $k+1$ and a maximal chain ending at it. This chain has a lowest point of intersection with one of the two old chains, so just replace that old chain with the new one from that point on. See Figure 3. □

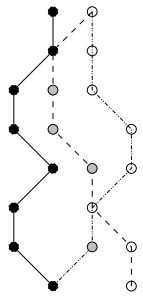


Figure 3: Salvaging a dead end.

Since Q_n^* is self-dual, it will suffice to show that the first half of its levels have at least two elements. We do this by explicitly constructing two disjoint chains to the halfway point. As a first approximation of these chains, take the following construction.

The left chain starts at $\{n-2,1,1\}$. At every step, we take the right-most possible H-step, e.g. the next partition is $\{n-3,2,1\}$. The right chain starts at $\{n-3,3\}$. At every step, we take the left-most possible H-step, e.g. the next partition is $\{n-4,4\}$. The names come from the relative positions of the chains when plotted, as in Figure 4. Both chains will eventually reach $\{m,m-1,\ldots,r+1,r,r,r-1,\ldots,2,1\}$, which is at least the halfway point [5], so the idea is to modify the left chain as little as possible to make it reach the halfway point without intersecting the right chain.

Once we've done that, we can apply Lemma 1 to get two disjoint chains of length $h(P_n) - 6$, then append the top and bottom three elements to one of them two get the desired chains. The following proposition will be used to prove several lemmas concerning the right chain.

Proposition 1. *If $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ is in the right chain, then $\lambda_i - \lambda_{i+1} \leq 2$ for $i = 1, 2, \ldots, k-2$. In other words, only the last difference can be greater than 2. Moreover, excluding the last difference, λ cannot have more than one difference equal to 2.*

Proof: By construction, we are always doing the left-most possible H-step. At first there is nothing to prove, since $k = 2$ through $\{\frac{n}{2}, \frac{n}{2}\}$ or $\{\frac{n+1}{2}, \frac{n-1}{2}\}$. Think in terms of partition diagrams as in the definition of H-steps. If there are no differences greater than 1 (excluding the last one), then push one box from λ_{k-1} to increase the last part (or from λ_k increase the number of parts). Now move to the left, pushing one box at a time until $\lambda_i - \lambda_{i+1} < 2$ for $i = 1, 2, \ldots, k-2$ again. Clearly we never get a difference greater than 2 or more than one difference of 2 unless we had one before, so the result follows by induction. □

3 Proof of Theorem 1

The proof comes in six cases, depending on r. We begin with general calculations that will be used in multiple cases. If $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \ldots\}$ is reachable from $\{n\}$ by only H-steps, such as the elements of the left and right chains, then $r(\lambda) = \lambda_2 + 2\lambda_3 + 3\lambda_4 + \cdots$, since each box in λ_i had to be moved horizontally $i - 1$ times.

Note that any λ in the left chain with $\lambda_1 - \lambda_2 > 2$ is not in the right chain by Proposition 1. This means that the left chain makes it safely to the partition $\{m + r, m - 1, m - 2, \ldots, 2, 1\}$ at level $\frac{m^3-m}{6}$ for $r \geq 2$. For $r > 2$, we can continue safely to $\{m+2, m-1, m-2, \ldots, r-2, r-2, \ldots, 2, 1\}$ (using both assertions in Proposition 1) for an additional $m(r-2) - \binom{r-2}{2}$ levels. So we're done if $2(m(r-2) - \binom{r-2}{2}) \geq rm$. For $r > 4$, this comes down to $m \geq \frac{r^2-5r+6}{r-4} = r - 1 + \frac{2}{r-4}$. For $r = 5$, this means $m \geq 6$. In fact $m = 5$ also works, since we really just needed $m(r-2) - \binom{r-2}{2} \geq \left\lfloor \frac{rm}{2} \right\rfloor$. For $r > 5$, we just need $m \geq r$ (since m must be an integer), but that's as general as possible since $r \leq m$. Thus we've established Theorem 1 when $r \geq 5$.

If $r = 4$, then the above construction gets us to one level shy of where we need to be, since we only reach $\{m + 2, m - 1, m - 2, \ldots, 3, 2, 2, 1\}$ safely.

Since $h(P_n)$ is always even when r is even, the middle level consists of self-conjugate partitions. Note that not all self-conjugate partitions are in Q_n, but one will be if it is covered by an element of Q_n since by duality it covers the conjugate of that element. Now we simply observe that $\{m+2, m-1, m-2, \ldots, 3, 2, 2, 1\}$ covers the self-conjugate partition $\{m+2, m-1, m-2, \ldots, 3, 2, 1, 1, 1\}$. This partition cannot be in the right chain by Proposition 1 (it is also not H-reachable from $\{n\}$ [5]), so this establishes Theorem 1 when $r = 4$.

If $r = 0$, then we safely reach $\{m+1, m-2, m-2, \ldots, 2, 1\}$, one level shy again. Once again, we simply observe that this covers the self-conjugate partition $\{m+1, m-2, m-2, \ldots, 3, 1, 1, 1\}$, which is not in the right chain by Proposition 1, so this establishes Theorem 1 when $r = 0$.

The remaining cases each require a lemma to get past the shortfall in the above argument.

If $r = 1$, then we safely reach $\{m+2, m-2, m-2, m-3, \ldots, 2, 1\}$, but in fact we can go further along the left chain.

Lemma 2. *The partitions* $\{m+1, m-1, m-2, \ldots, 2, 1\}$ *and* $\{m, m-1, \ldots, k+1, k, k, k-2, \ldots, 2, 1\}$, $5 \leq k \leq m$, *do not occur in the right chain.*

Proof: If $\{m+1, m-1, m-2, \ldots, 2, 1\}$ occurred in the right chain, then it would have to be preceded by $\{m+2, m-2, m-2, \ldots, 2, 1\}$ or $\{m+1, m, m-3, \ldots, 2, 1\}$ (otherwise we couldn't have done the left-most H-step), both of which violate Proposition 1.

If $\{m, m-1, \ldots, k+1, k, k, k-2, \ldots, 2, 1\}$ occurred in the right chain, then it would have to be preceded by $\{m, m-1, \ldots, k+1, k, k, k-1, k-4, k-4, \ldots, 2, 1\}$ (note this works even for $k = m$) which violates Proposition 1 unless $k - 4 = 0$, hence the need for $k \geq 5$, or by $\{m, m-1, \ldots, k+1, k+1, k-1, k-2, \ldots, 2, 1\}$. In this case, we can recursively work our way back to $\{m+1, m-1, m-2, \ldots, 2, 1\}$, which is not in the right chain since it would have to be preceded by $\{m+2, m-2, m-2, \ldots, 2, 1\}$ or $\{m+1, m, m-3, \ldots, 2, 1\}$, both of which violate Proposition 1. □

Now apply Lemma 2 to extend the left chain safely to $\{m, m-1, \ldots, 5, 5, 3, 2, 1\}$, which occurs at level $\frac{m^3+5m-24}{6}$. Since $h(P_n) = \frac{m^3+2m}{3}$, it suffices if $m^3+5m-24 \geq m^3+2m$, or $m \geq 8$. $m = 7$ also works since $h(P_{29}) = 119$ and we reach level 59. The case $m = 6$, $n = 22$ can be dealt with individually. The left chain gets to $\{6, 5, 5, 3, 2, 1\}$ at level 37, but intersects the right chain at level 38 with $\{6, 5, 4, 4, 2, 1\}$. However, the right chain reaches $\{6, 5, 4, 4, 3\}$ at level 37, which also covers the self-conjugate partition $\{5, 5, 5, 4, 3\}$, so this establishes Theorem 1 when $r = 1$.

If $r = 2$, then we safely reach $\{m+2, m-1, m-2, m-3, \ldots, 2, 1\}$, but in fact we can go further along the left chain.

Lemma 3. *The partitions* $\{m+1, m, m-2, m-3, \ldots, 2, 1\}$ *and* $\{m+1, m-1, m-2, \ldots, k+1, k, k, k-2, \ldots, 2, 1\}$, $1 \le k \le m-1$, *do not occur in the right chain.*

Proof: If $\{m+1, m, m-2, m-3, \ldots, 2, 1\}$ occurred in the right chain, then it would have to be preceded by $\{m+2, m-1, m-2, m-3, \ldots, 2, 1\}$, $\{m+1, m+1, m-3, m-3, \ldots, 2, 1\}$, or $\{m+1, m, m-1, m-4, m-4, \ldots, 2, 1\}$, all of which violate Proposition 1. Note that we are tacitly assuming that $m > 4$, but that's fine since $n > 16$, so $m \ge 5$.

Since $\{m+1, m-1, m-2, \ldots, k+1, k, k, k-2, \ldots, 2, 1\}$ has two differences of size 2 for $k > 2$, Proposition 1 takes care of those cases (note $k = m-1$ means the partition is $\{m+1, m-1, m-1, m-3, \ldots, 2, 1\}$). If $\{m+1, m-1, m-2, \ldots, 3, 2, 2\}$ occurred in the right chain, then it would have to be preceded by $\{m+2, m-2, m-2, \ldots, 3, 2, 2\}$ or $\{m+1, m, m-3, \ldots, 3, 2, 2\}$, both of which violate Proposition 1. $k = 1$ is similar. □

Now apply Lemma 3 to extend the left chain safely to $\{m+1, m-1, m-2, \ldots, 2, 1, 1\}$, which occurs at level $\frac{m^3+5m}{6}$. Since $h(P_n) = \frac{m^3+5m}{3}$, this establishes Theorem 1 when $r = 2$.

Finally, if $r = 3$, we safely reach $\{m+2, m-1, m-2, \ldots, 2, 1, 1\}$. Now we just modify Lemma 3. Note we could also show that the right chain has no elements ending in 1,1 until it's too late, but this method is cleaner.

Lemma 4. *The partitions* $\{m+1, m, m-2, \ldots, 2, 1, 1\}$ *and* $\{m+1, m-1, m-2, \ldots, k+1, k, k, k-2, \ldots, 2, 1, 1\}$, $4 \le k \le m-1$ *do not occur in the right chain.*

Proof: Exactly the same as Lemma 3, since the second 1 at the end never comes into play. □

Now apply Lemma 4 to extend the left chain safely to $\{m+1, m-1, m-2, \ldots, 5, 4, 4, 2, 1, 1\}$, which occurs at level $\frac{m^3+11m-18}{6}$. Since $h(P_n) = \frac{m^3+8m}{3}$, it suffices if $m^3 + 11m - 18 \ge m^3 + 8m$, or $m \ge 6$. When $m = 5$, we get to level 27, and $h(P_5) = 55$, so this case is fine as well. This establishes Theorem 1 when $r = 3$, and thus completes the proof. □

4 Smaller cases and related questions

The smaller n for which $\lambda_2(P_n) = \lambda_1(P_n) - 6$ are 10, 13, 14, and 15. Figure 4 shows Q_{16}. Since there are levels of size 1 in the middle, P_{16} cannot possibly have two chains of the desired lengths.

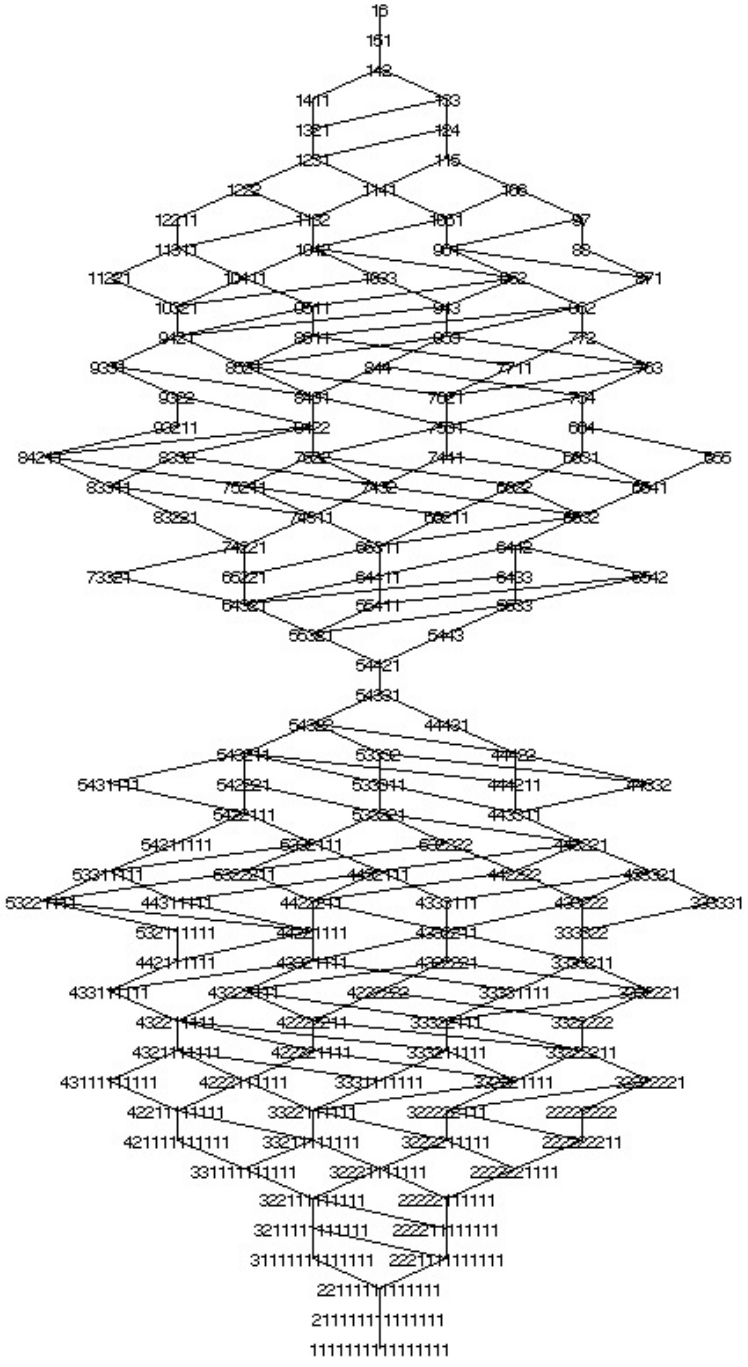


Figure 4: Elements of P_{16} on maximal chains.

More generally, Table 1 shows the partitions of chain lengths for P_n, $1 \leq n \leq 14$. It is interesting to note that in all of these cases, the elements added between $\lambda_{k-1}(P_n)$ and $\lambda_k(P_n)$ form a chain that is added to the previous $k-1$ chains (and similarly for antichains). This is not the case for arbitrary posets, such as Figure 5. The proof of Theorem 1 shows this is the case for every P_n when $k = 2$; it would be interesting to know if it holds for all k.

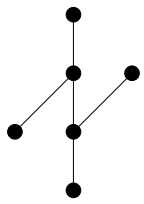


Figure 5: A poset P such that the largest chain is not one of the largest two chains. $\lambda(P) = \{4, 2\}$.

n	$\lambda(P_n)$
1	$\{1\}$
2	$\{2\}$
3	$\{3\}$
4	$\{5\}$
5	$\{7\}$
6	$\{9, 2\}$
7	$\{12, 3\}$
8	$\{15, 7\}$
9	$\{18, 9, 3\}$
10	$\{21, 15, 4, 2\}$
11	$\{25, 18, 10, 3\}$
12	$\{29, 21, 13, 10, 4\}$
13	$\{33, 27, 18, 14, 6, 3\}$
14	$\{37, 31, 24, 19, 15, 6, 3\}$

Table 1: Known values of $\lambda(P_n)$.

While the proof of Theorem 1 is constructive in the cases where $h(P_n)$ is even, so that the middle level consists of self-conjugate partitions, it is not constructive when $h(P_n)$ is odd, since in those cases the proof relies on Lemma 1. It would be interesting to give an explicit construction of two long chains in those cases. Note also that Lemma 1 does not generalize in the most obvious way for finding three chains, due to posets such as the one shown in Figure 6.

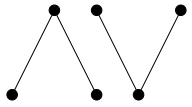


Figure 6: A graded poset with three elements on every level but no three disjoint chains of maximal length.

Conjecture 1. *For large n, $\lambda_i(P_n) - \lambda_{i+1}(P_n)$ depends only on i.*

Note that Proposition 1 holds for the left chain if we exclude the first difference instead of the last, so a partition with a difference of 3 or two differences of 2 in the middle will not be on either chain. We can try to exploit this to construct a third disjoint chain to the middle level, starting at $\{n-5, 4, 1\}$, by keeping the second difference greater than or equal to 3, and similarly for k chains by keeping a different difference large in each one. If r is even, so that we can extend the chains by conjugation, then this will give us k disjoint chains that end on the conjugates of their starting points. By an analogous calculation to the proof of Theorem 1 for $r \geq 5$, this works for getting three chains when $m \geq r - 1 + \frac{12}{r-8}$, proving that $\lambda_2(P_n) - \lambda_3(P_n) = 6$ when $r > 8$ is even and m is suffiently large. The smallest example is $n = 117$, with $m = 14$ and $r = 12$. Note that $\lambda_i(P_n) - \lambda_{i+1}(P_n)$ need not always be 6. It appears that the fourth chain starts just one level further down, so we conjecture that $\lambda_3(P_n) - \lambda_4(P_n) = 2$ for large n.

Let M be the transition matrix from the bases $\{e_\lambda\}$ to $\{m_{\lambda'}\}$ of homogeneous symmetric functions of degree n. Since $M_{\lambda\mu} > 0$ iff $\mu \leq \lambda'$, it is a theorem of Gansner and Saks that a generic matrix with the same 0 entries will have jordan blocks whose sizes are exactly the parts of $\lambda(P_n)$ (see [1]). Using Table 1 and Maple, one can verify that M is sufficiently generic at least for $n \leq 13$.

Another open problem is to find the size $a(n)$ of the largest antichain in P_n. Let $p(n)$ be the number of partitions of n. There is the obvious upper bound $a(n) \leq p(n)$. By Dilworth's theorem, $a(n) \geq p(n)/(h(P_n) + 1)$, so we have $\Omega(n^{-5/2}e^{\pi\sqrt{2n/3}}) \leq a(n) \leq O(n^{-1}e^{\pi\sqrt{2n/3}})$. It would be interesting to find a constructive proof that $a(n)$ is at least as large as the lower bound. In addition to the values of $a(n)$ implied by Table 1, we can see that $a(15) = 9$. Moreover, $\lambda_9(P_{15}) = 2$, with the long antichains being $71^8, 6221^5, 541^6, 53221^3, 52^5, 4431^4, 442221, 433311, 3^5$ and their conjugates. One can also verify that $a(16) = 10$, with $\lambda_{10}(P_{16}) = 5$. The sequence of $a(n)$'s is number A076269 in [6].

One construction that shows $a(n)$ has a lower bound of the form $e^{c\sqrt{n}}$ is as follows. Begin with the antichain 7321^4, 722221, 651^5, 642211, 63322,

553111, 55222, 54421, 4444 in P_{16}. Let $\nu + 7n$ denote a partition ν from the list with $7n$ added to each part. Consider ν to have 7 parts, so some of them might be 0. Then $\{\nu + 7n, \nu + 7(n-1), \ldots, \nu + 7, \nu\}$ is a partition of $N = 16(n+1) + 49\frac{n^2+n}{2} = \frac{49}{2}n^2 + O(n)$. There are 9^{n+1} choices for the ν's, yielding an antichain of size 9^{n+1} in P_N. This yields a lower bound for $a(n)$ of $e^{c\sqrt{n}}$ where $c = \ln 9\sqrt{2/49} = 0.4439\ldots$. By starting with a 28-element antichain in P_{27} where each ν has at most 9 parts, and largest part at most 8, one can similarly get $c = \frac{\ln 28}{6} = 0.555\ldots$. This is still a long way from $\pi\sqrt{2/3} = 2.565\ldots$, but at least it's constructive.

5 Acknowledgements

I thank Richard Stanley for suggesting this problem, and for his many helpful suggestions and comments.

References

[1] T. Britz and S. Fomin, Finite Posets and Ferrers Shapes, *Adv. Math* **158** (2001), 86–127.

[2] T. Brylawski, The lattice of integer partitions, *Discrete Math.* **6** (1973), 201–219.

[3] C. Greene, Some partitions associated with a partially ordered set, *J. Comb. Theory, Ser. A* **20** (1976), 69–79.

[4] C. Greene and D. J. Kleitman, The structure of Sperner k-families, *J. Comb. Theory, Ser. A* **20** (1976), 41–68.

[5] C. Greene and D. J. Kleitman, Longest Chains in the Lattice of Integer Partitions ordered by Majorization, *Europ. J. Combinatorics* **7** (1986), 1–10.

[6] N. J. A. Sloane, editor (2002), The On-Line Encyclopedia of Integer Sequences, http://www.research.att.com/~njas/sequences/.

Combinatorial Properties of One-Dimensional Arrangements

Frédéric Cazals

CONTENTS

Motivated by problems from computer graphics and robotics—namely, ray tracing and assembly planning—we investigate the combinatorial structure of arrangements of segments on a line and of arcs on a circle. We show that there are, respectively, $1\times3\times5\times\cdots\times(2n-1)$ and $(2n)!/n!$ such arrangements; that the probability for the i-th endpoint of a random arrangement to be an initial endpoint is $(2n-i)/(2n-1)$ or $\frac{1}{2}$, respectively; and that the average number of segments or arcs the i-th endpoint is contained in are $(i-1)(2n-i)/(2n-1)$ or $(n-1)/2$, respectively. The constructions used to prove these results provide sampling schemes for generating random inputs that can be used to test programs manipulating arrangements.

We also point out how arrangements are classically related to Catalan numbers and the ballot problem.

1. INTRODUCTION AND MOTIVATION

Consider a set of n intervals in the real line, and assume that all $2n$ endpoints are distinct. We will be interested in the combinatorial properties of such arrangements, that is, the properties that depend solely on the order in which the endpoints occur, rather than their precise position. Specifically, we will count the number of possible arrangements and determine two statistics (averaged over all possible arrangements) for the i-th endpoint in the sequence: the average number of intervals that this point belongs to, and the probability that it is an initial, rather than terminal, endpoint. We also consider the analogous problem for arcs in a circle.

The overview of the paper is the following. Section 1 briefly discusses the applications that led to this investigation. Sections 2 and 3 deal with the linear and circular cases, respectively. Section 4 lists some interesting open problems.

Work on this paper was accomplished while the author was visiting the Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305 USA.

Keywords: combinatorics, computational geometry, algorithms and data structures.

1058-6458/97 $0.50 per page

Assembly Sequencing and Arrangements

Assembly sequencing is a domain of robotics whose purpose is, given a collection of mechanical parts that fit together in a certain way and a class of motions that these parts can be subjected to, to compute a way, if one exists, to get the single parts from the whole assembly. For example, in the simple assembly in Figure 1, if we restrict ourselves to translations in the plane, it is clear that P_1 and P_2 can only be taken apart by a horizontal motion, whereas P_3 and P_4 can be taken apart by motions within an interval of directions.

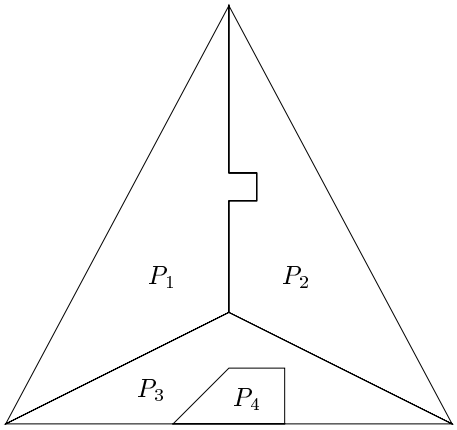


FIGURE 1. A simple assembly.

Analyzing assembly sequences can be of great use in many ways: for example, to check that the product can be disassembled, to ensure that the parts that may be serviced often are easily accessible, or to facilitate recycling by clustering parts made of the same material. Of major practical interest, assembly sequencing is also a difficult algorithmic problem since it is intractable in its general form; see [Natarajan 1988], for example. Restricted, yet interesting, versions of the problem have been shown to have polynomial-time algorithms.

For example, consider the case of planar polygonal assemblies where the only class of motions allowed is infinite translations and where each split results in two subassemblies [Wilson and Latombe 1994; Latombe et al. 1996]. The space of motions is described by the unit circle S^1, since a translation corresponds to a unit vector in the plane. Given two parts, the set of directions along which one can be translated without colliding with the other is described by an arc on S^1, determined by a cone on the Minkowski difference [Latombe 1991]; see Figure 2.

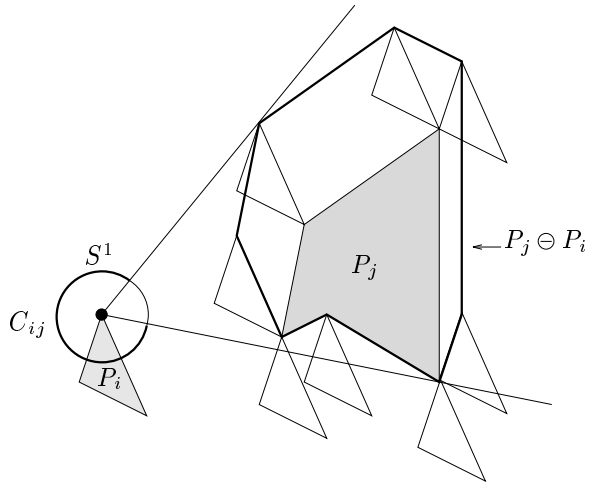


FIGURE 2. The arc of directions of movement of P_i that lead to collision with P_j is given by the cone on the Minkowski difference set $P_j \ominus P_i$.

The blocking relations for all the pairs of parts are thus described by $n(n-1)$ arcs in S^1. Together, they constitute an *arrangement of arcs* that divides S^1 into *endpoints* and *intervals*, as shown on Figure 3. This arrangement is called the *nondirectional blocking graph*, or NDBG, since it gives the blocking relations for any pair of parts and any direction. To each endpoint of the arrangement corresponds a directed graph, called the *directional blocking graph*, having a vertex for each part and an edge between vertices i and j if part i collides with part j when translated along this direction. A topological sorting of the strongly connected components of this directed graph gives the removable subassemblies along this direction. Starting with the full assembly, the disassembly algorithm consists in recursively removing translatable subassemblies with the previous scheme.

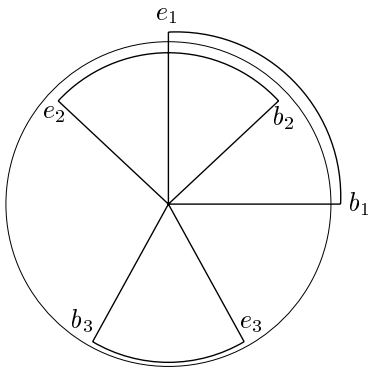


FIGURE 3. Arrangement of arcs on S^1.

FIGURE 4. A ray-traced scene.

Performing a worst-case analysis of this algorithm is pretty easy. Indeed, the NDBG has $O(n^2)$ vertices and each DBG has size $O(n^2)$, which gives a space requirement of $O(n^4)$. The time complexity of the recursive disassembly is $O(n^5)$ since there are at most n levels of recursion, and each level requires examining $O(n^2)$ DBGs for which the reduced graph (graph of the strongly connected components) and a topological sorting have to be computed.

The average-case analysis is much more challenging. Firstly, a precise understanding of the combinatorics of arc arrangements is required. Secondly, some random graph structure is needed for the directional blocking graphs. The latter question is difficult since the number of edges of a DBG depends on the geometric information encoded in the relative position of the pairs of parts, which requires some definition of random assemblies. This goes beyond the scope of this paper. By contrast, the first problem is better defined and raises precise questions such as the generation of a random arrangement (see also [Zimmermann 1994]), the probability of a given endpoint to be an initial or terminal endpoint, the average number of arcs a given endpoint of an arrangement is contained in, and so on. These questions will be addressed in Section 3.

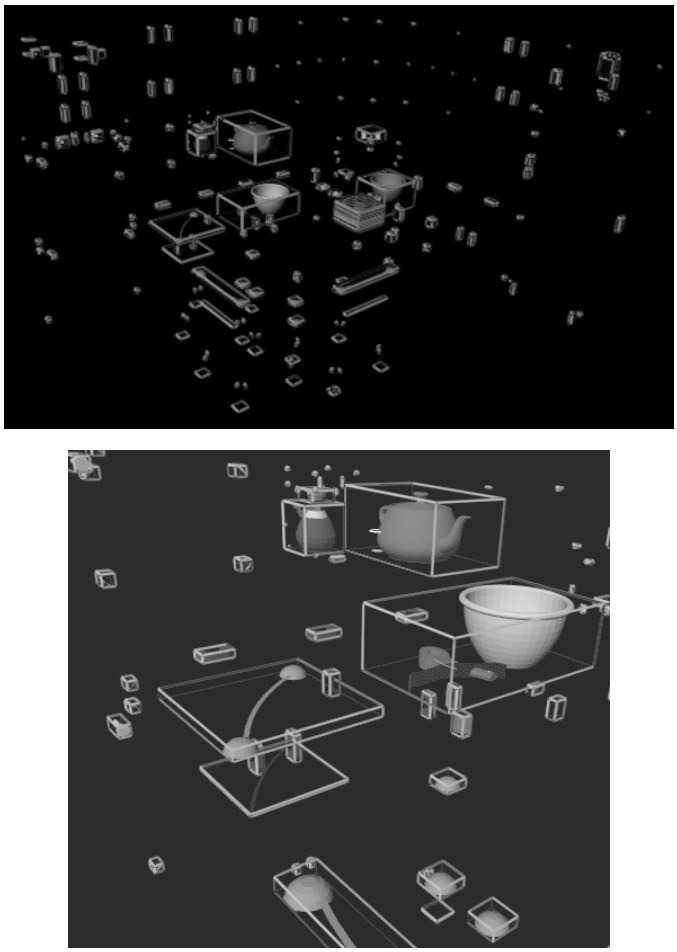


FIGURE 5. Clusters found in the scene. Bottom: detail.

Ray Tracing and Clustering

Ray tracing is a technique from computer graphics that consists in computing views of scenes defined by geometric primitives. Very often these primitives are polygons defined by their geometry and color, a given object of the scene being defined by a set of such polygons. As an example, consider Figure 4, where the kitchen model consists of about 25,000 polygons, and objects such as the bowl on the table or the teapot are made of about 1000 polygons. To sketch the ray-tracing algorithm (see [Foley et al. 1990] for details), let a ray be defined by a point and a direction in three dimensions. Rays are used to simulate the light received by the observer's eye, so that the key operation of the whole algorithm consists in finding, for a given ray, the closest object hit in order to plot the corresponding color on the screen of the computer where the algorithm is run.

Reducing the number of ray-polygon intersection tests has ever been a challenging issue. The main paradigm consists in partitioning the volume containing the scene into small boxes, in order to test for intersection only those polygons stored in the boxes of the partition crossed by the ray of interest. An example of such partitioning, the *uniform grid*, is based on a regular grid aligned with the three coordinate axes. (See [Cazals et al. 1995] for a discussion of grid-like data structures.)

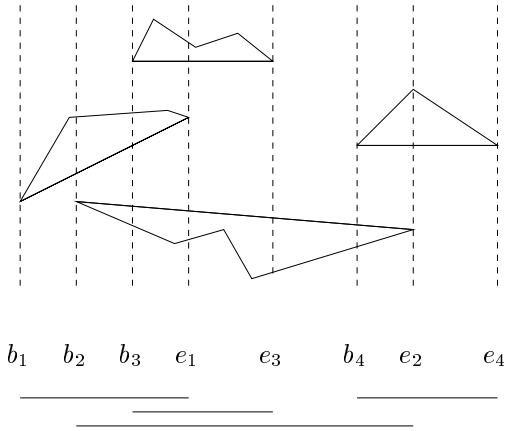


FIGURE 6. Arrangement of line segments.

The problem with this approach is that whenever too many polygons fall into the same box the spatial partitioning does not result in data partitioning, so the number of ray-polygon intersection tests is not reduced significantly. To remedy this problem, it was observed in [Cazals et al. 1995] that using uniform grids for densely populated areas of the scene called clusters could partially solve the problem. Examples of clusters are the neighborhoods of the bowl, teapot, or door knobs, and are depicted on Figure 5. More precisely, a cluster is defined as a subset of objects whose projection along the three axis x, y and z is almost-connected. And, since the projection of a polygon on a line is a line segment, the clustering algorithm analysis turns out to be closely related to the combinatorics of arrangement line segments, as in Figure 6. Thus, the results presented in Section 2 of this paper were recently used in [Cazals and Sbert 1997] in conjunction with integral geometry techniques to define statistics aiming at characterizing standard scenes types such as natural models, architectural scenes, etc.

2. THE LINEAR CASE

Notations and Previous Work

Consider an set of n segments on the line. Let the $2n$ endpoints, which are assumed distinct, be indexed in order by $(1\,..\,2n) = \{1, 2, 3, \ldots, 2n\} \subset \mathbb{Z}$, an orientation having been fixed in advance. From the combinatorial point of view, the arrangement of segments is specified completely by an involution a of $(1..2n)$ without fixed points. More precisely, a segment joining endpoints i and j is denoted $[i, j]$, if $i < j$; the endpoint-pairing involution maps i to j and j to i, and we call i and j the initial and terminal endpoints of the pair. For instance, the three possible arrangements of two segments are shown in Figure 7: they are $\{[1, 2],\ [3, 4]\}$, $\{[1, 3],\ [2, 4]\}$,

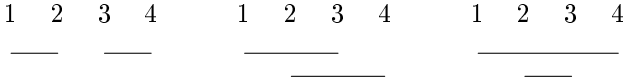


FIGURE 7. The possible arrangements of two segments.

and $\{[1,4],\,[2,3]\}$. The arrangement $\{[1,2],\,[3,4]\}$ is also thought of as the pairing $1 \leftrightarrow 2,\ 3 \leftrightarrow 4$.

Let S_n be the set of all arrangements of n segments, and let $s_n = |S_n|$, where the bars denote cardinality; thus $s_2 = 3$ (compare Figure 7). In general, we have

$$s_n = 1 \times 3 \times \cdots \times (2n-1),$$

as can easily be seen: the pairing can take 1 to any of the $2n-1$ remaining indices; it can take the lowest of the remaining $2n-2$ indices into any of the remaining $2n-3$; and so on.

For a particular arrangement $a \in S_n$ and for $i \in (1\,..\,2n)$, we define $a[i]$ to be B or E according to whether endpoint i begins or ends the respective segment, that is, according to whether $a(i) > i$ or $a(i) < i$. For fixed i, the statistics we are interested in are the probability that $a[i] = B$ (or $a[i] = E$), as a ranges over all of S_n, and the *overlap number* of i, that is, the average number of arcs or line segments in whose interior endpoint i is contained. Formally, we define

$$\begin{aligned}
\beta_i^{(n)} &= |\{a \in S_n : a[i] = B\}|,\\
\varepsilon_i^{(n)} &= |\{a \in S_n : a[i] = E\}|,\\
\tau_i^{(n)} &= \sum_{a \in S_n} |\{(b,e) \in a : b < i < e\}|.
\end{aligned}$$

The corresponding vectors as i ranges over $(1\,..\,2n)$ are denoted $\vec{\varepsilon}^{(n)}$, $\vec{\beta}^{(n)}$, and $\vec{\tau}^{(n)}$. Thus for $n = 2$ we have $\vec{\beta}^{(2)} = [3,2,1,0]$, $\vec{\varepsilon}^{(2)} = [0,1,2,3]$, $\vec{\tau}^{(2)} = [0,2,2,0]$ (Figure 7).

The numbers s_n have appeared in the literature in several forms, in particular in [Touchard 1950; Riordan 1975], which deal with the stamp-folding problems. The value of s_n is given by Touchard. Riordan mentions that the number of pairings of $2n$ points on a circle is also s_n, since such pairings, too, can be seen as involutions of $(1\,..\,2n)$. (More geometrically, one can open up the circle at an arbitrary point; then a pair of points on S^1 corresponds to a segment in the resulting interval, and vice versa.) Finally, a look at the very nice book [Sloane and Plouffe 1995, M3002] shows that the sequence s_n has long been known in connection with the expression of Wallis integrals.

Riordan [1975] also points out the interesting relation between the number of pairings on a circle and the Catalan numbers: pairings where chords are not allowed to intersect give rise to the Catalan numbers $C_n = \binom{2n}{n}/(n+1)$, while pairings that allow crossings between the chords lead to s_n. Riordan cites a correspondence between the Catalan numbers and the ballot problem, also known as the subdiagonal random walks problem [Comtet 1974; Yaglom and Yaglom 1964; Knuth 1973].

Initial and Terminal Endpoints

Theorem 2.1. *For any $i = (1\,..\,2n)$ we have*

$$\beta_i^{(n)} = \frac{2n-i}{2n-1}\,s_n = (2n-i)s_{n-1} = s_n - (i-1)s_{n-1}.$$

Therefore the probability that the i-th endpoint is initial is $(2n-i)/(2n-1)$, and the probability that it is final is $(i-1)/(2n-1)$.

Proof. We use the recursion

$$\beta_i^{(n+1)} = (i-1)\beta_{i-1}^{(n)} + s_n + (2n+1-i)\beta_i^{(n)}, \quad (2.1)$$

for $i = (1\,..\,2n+1)$, with initial condition $\beta_{2n+2}^{(n+1)} = 0$. This recursion can be verified as follows. Given an element of S_{n+1}, let $i \in (1\,..\,2n+1)$ be the initial point of the segment whose terminal endpoint is $2n+2$. If we remove the pair $[i,\,2n+2]$ and renumber, we get a well-defined element of S_n. Conversely, a choice of $a \in S_n$ and $i \in (1\,..\,2n+1)$ yields a unique element $a' \in S_{n+1}$, by the addition of a segment that starts between position $i-1$ and i of a and ends at the far right. (Incidentally, this is another way to derive the value of s_n, since it shows that $|S_{n+1}| = (2n+1)|S_n|$.) Because of the renumbering, we have

$$a'[j] = \begin{cases} a[j] & \text{if } j < i,\\ a[j-1] & \text{if } i < j < 2n+2i; \end{cases}$$

moreover $a'[i] = B$ and $a'[2n+2] = E$. Analyzing the contribution to each $\beta_j^{(n+1)}$ from each value of i, we can write:

$$\begin{aligned}
&[\,\beta_1^{(n+1)}\ \beta_2^{(n+1)}\ \beta_3^{(n+1)}\ \ldots\ \beta_{2n+1}^{(n+1)}\ \beta_{2n+2}^{(n+1)}\,]\\
=\ &[\ \ s_n\ \ \beta_1^{(n)}\ \ \beta_2^{(n)}\ \ldots\ \beta_{2n}^{(n)}\ \ 0\ \ ]\quad (i{=}1)\\
+\ &[\ \beta_1^{(n)}\ \ s_n\ \ \beta_2^{(n)}\ \ldots\ \beta_{2n}^{(n)}\ \ 0\ \ ]\quad (i{=}2)\\
&\qquad\qquad\vdots\\
+\ &[\ \beta_1^{(n)}\ \ \beta_2^{(n)}\ \ \beta_3^{(n)}\ \ldots\ \ s_n\ \ 0\ \ ]\quad (i{=}2n{+}1)
\end{aligned}$$

Summation by columns gives the desired recurrence relation (2.1). (Note that in this relation the undefined quantities $\beta_{i-1}^{(n)}$ when $i = 0$ and $\beta_i^{(n)}$ when $i = 2n+1$ are multiplied by zero, so the equation still makes sense.)

We now prove the closed-form expression for $\beta_i^{(n)}$. We certainly have $\beta_1^{(n)} = 1$; assume by induction that $\beta_i^{(n)} = s_{n-1}(2n-i)$ for $i \in (1\,..\,2n)$. We get, for any $i \in (2\,..\,2n{+}1)$:

$$\begin{aligned}
\beta_i^{(n+1)} &= (i{-}1)s_{n-1}(2n{-}i{+}1) + s_n\\
&\qquad +(2n{-}i{+}1)s_{n-1}(2n{-}i)\\
&= s_n + s_{n-1}(2n{-}1)(2n{-}i{+}1).
\end{aligned}$$

But $s_n = s_{n-1}(2n{-}1)$, which completes the proof for $i \in (1\,..\,2n{+}1)$. The case $i = 2n+2$ is trivial.

The probability that endpoint i is initial in an n-point arrangement is of course $\beta_i^{(n)}/s_n$, and the probability that it is terminal is the complement. This proves the theorem. □

The Overlap Number

Theorem 2.2. *For any $i = (1\,..\,2n)$ we have $\tau_i^{(n)} = (i{-}1)(2n{-}i)s_{n-1}$. Thus, the average overlap number of the i-th endpoint in an n-segment arrangement is $(i{-}1)(2n{-}i)/(2n{-}1)$.*

It is possible to prove this using recursion, much like Theorem 2.1; but here a nicer direct proof:

Proof. Endpoint i is covered by segments of the form $[j,k]$ for $j \in (1\,..\,i{-}1)$ and $k \in (i{+}1\,..\,n)$, and there are $(i-1)(2n-i)$ such segments. Each of them appears exactly s_{n-1} times in the s_n arrangements, since once we have fixed segment $[j,k]$ we are left with an arrangement of $n-1$ segments. □

3. THE CIRCULAR CASE

We now turn to arrangements of arcs in the circle, and answer the same questions that were posed in Section 2 for linear segments. Because all endpoints are equivalent on S^1, the situation is easier.

We start with the number of arrangements:

Theorem 3.1. *The number r_n of arrangements of n arcs on a circle is equal to $(2n)!/n!$.*

Proof. An arrangement of n arcs is specified by a pairing of the $2n$ points, together with n independent binary choice, one for each of pair of endpoints (either arc determined by the pair may appear in the arrangement; see Figure 8). Therefore $r_n = 2^n \cdot s_n = 2^n((2n{-}1)\times(2n{-}3)\times\cdots\times 3\times 1) = (2n)!/n!$. □

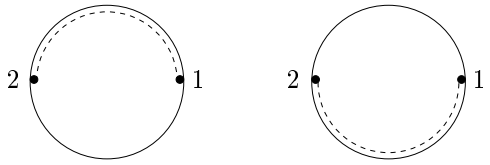


FIGURE 8. Two arcs are determined with equal probability by a choice of two endpoints.

The classification of the arrangements into pairings also yields the probability that a given endpoint is initial. Because, for a given pair of endpoints, each of the two choices of a segment with those endpoints occurs in half the arrangements that include this pair of endpoints, the probability that a fixed endpoint is initial is $\frac{1}{2}$.

The same reasoning shows that the average overlap number of any endpoint in an arrangement of n arcs is $(n-1)/2$: if endpoint i is chosen and we consider the relation of i with any pair (r,s) with $r,s \neq i$, we see that i lies in the interior of the arc with endpoints (r,s) for exactly half the arrangements that include this pair.

4. CONCLUSIONS

The analysis in Section 2 is of interest for computer graphics algorithms dealing with objects' projections along lines. The results in Section 3 may be the first step toward an average case analysis of the NDBG-based algorithm for computing assembly sequences in the simple case of polygons in the plane moved with infinite translations. Although this particular assembly sequencing problem might appear quite restrictive, it is actually one of the few for which it is reasonable to come up with an implementation for, so that any precise analysis would be of interest.

We remark that, from the study of the combinatorial structure of arrangements presented in this paper, it is easy to randomly generate such arrangements in order to test and validate geometric software. An algorithm to do this might go as follows.

Assume we have an array t of integers, of length $2n$, and two functions: swap(t, i, j), which swaps the contents of slots i and j in t, and random(k), which returns an integer in the range $1\,..\,k$. The algorithm returns the endpoint b_i and e_i, for $i \in (1\,..\,k)$, of the arrangement being generated.

```
for i ∈ (1..2n) do
    t[i] ← i;
for i ∈ (1..n) do
    p ← t[random(2n+2−2i)];
    swap(t, 2n+2−2i, p);
    q ← t[random(2n+1−2i)];
    swap(t, 2n+1−2i, q);
    b_i ← inf(p, q);
    e_i ← sup(p, q);
```

Many interesting issues remain open, in particular the calculation of higher moments for the statistics presented here. It would be interesting to find two-dimensional analogs for the results presented here; the work done so far in this direction deals with arrangements of lines in the plane, but not line segments [Edelsbrunner 1986].

ACKNOWLEDGMENTS

This work received a financial support from Matra Datavision for a joint project on assembly planning with the Stanford University Robotics Laboratory.

The author wishes to thank Danny Halperin and Jean-Marc Schlenker for insightful comments, the reviewer of this paper for important pointers to the bibliography, and Stéphane Rivière and Ram Ramkumar for rereading the paper.

The author also thanks Don Greenberg of the Cornell University program of Computer Graphics for the kitchen model in Section 1.

REFERENCES

[Cazals and Sbert 1997] F. Cazals and M. Sbert, "Some integral geometry tools to estimate the complexity of 3D scenes", Technical report, INRIA, 1997.

[Cazals et al. 1995] F. Cazals, G. Drettakis, and C. Puech, "Filtering, clustering and hierarchy construction: a new solution for ray-tracing complex scenes", *Computer Graphics Forum* **14**:3 (1995).

[Comtet 1974] L. Comtet, *Advanced combinatorics*, D. Reidel, Dordrecht and Boston, 1974. Translated from the French by J. W. Nienhuys.

[Edelsbrunner 1986] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer, 1986.

[Foley et al. 1990] J. Foley et al., *Computer Graphics: Principles and Practice*, 2nd ed., Addison Wesley, 1990.

[Knuth 1973] D. E. Knuth, *The art of computer programming*, Addison-Wesley, 1973.

[Latombe 1991] J. C. Latombe, *Robot motion planning*, Kluwer, 1991.

[Latombe et al. 1996] J. Latombe, R. H. Wilson, and F. Cazals, "Assembly sequencing with toleranced parts", *Computer-Aided Design* (1996). To appear.

[Natarajan 1988] B. K. Natarajan, "On planning assemblies", pp. 299–308 in *Proc. Fourth Annual ACM Symp. Comput. Geom.* (Urbana/Champaign, 1988), Assoc. Computing Machinery (ACM), New York, 1988.

[Riordan 1975] J. Riordan, "The distribution of crossings of chords joining pairs of $2n$ points on a circle", *Math. Comp.* **29** (1975), 215–222.

[Sloane and Plouffe 1995] N. J. A. Sloane and S. Plouffe, *The encyclopedia of integer sequences*, Academic Press, 1995.

[Touchard 1950] J. Touchard, "Contribution à l'étude du problème des timbres poste", *Canadian Journal of Mathematics* **2** (1950), 385–398.

[Wilson and Latombe 1994] R. H. Wilson and J.-C. Latombe, "Geometric reasoning about mechanical assembly", *Artificial Intelligence* **71** (1994), 371–396.

[Yaglom and Yaglom 1964] A. M. Yaglom and I. M. Yaglom, *Challenging mathematical problems with elementary solutions*, Holden-Day, 1964.

[Zimmermann 1994] P. Zimmermann, "Gaia: a package for the random generation of combinatorial structures", *Maple Technical Newsletter* **1**:1 (1994), 1–9.

Frédéric Cazals, *i*MAGIS-IMAG, BP 53, 38041 Grenoble cedex 09, France (Frederic.Cazals@imag.fr)

Received July 17, 1995; accepted in revised form September 14, 1996

GENERATING TREES AND THE CATALAN AND SCHRÖDER NUMBERS

JULIAN WEST

ABSTRACT. A permutation $\pi \in S_n$ avoids the subpattern τ iff π has no subsequence having all the same pairwise comparisons as τ, and we write $\pi \in S_n(\tau)$. We present a new bijective proof of the well-known result that $|S_n(123)| = |S_n(132)| = c_n$, the n-th Catalan number. A generalization to forbidden patterns of length 4 gives an asymptotic formula for the vexillary permutations. We settle a conjecture of Shapiro and Getu that $|S_n(3142, 2413)| = s_{n-1}$, the Schröder number, and characterize the deque-sortable permutations of Knuth, also counted by s_{n-1}.

1. Introduction to forbidden subsequences

We regard a permutation $\pi \in S_n$ as a sequence of n elements, $\pi = \{\pi(i)\}_{i=1}^n$. We say that π contains the 3-letter pattern 231 iff there is a triple $1 \le i < j < k \le n$ such that $\pi(k) < \pi(i) < \pi(j)$. Otherwise π *avoids* the pattern. We define τ-avoiding permutations similarly for every $\tau \in S_k$:

Definition 1.1. *For $\tau \in S_k$, a permutation $\pi \in S_n$ is τ-avoiding iff there is no $1 \le i_{\tau(1)} < i_{\tau(2)} < \ldots < i_{\tau(k)} \le n$ such that $\pi(i_1) < \pi(i_2) < \ldots < \pi(i_k)$. The subsequence $\{\pi(i_{\tau(j)})\}_{j=1}^k$ is said to have* type τ.

Two sequences, π, ρ of length n are evidently of the same type iff they have the same pairwise comparisons throughout, namely if $\pi(i) < \pi(j) \leftrightarrow \rho(i) < \rho(j)$. We denote by $S_n(\tau)$ the set of all permutations in S_n which avoid τ. If $R = \{\sigma_1, \sigma_2, \ldots, \sigma_q\}$, we abbreviate $S_n(R) = S_n(\sigma_1, \ldots, \sigma_q) = \bigcap S_n(\sigma_j)$. Fundamental questions are to determine $|S_n(R)|$ viewed as a function of n, and if $|S_n(R)| = |S_n(R')|$ to discover an explicit bijection between $S_n(R)$ and $S_n(R')$.

The most studied case has been to forbid a single pattern of length 3. Because of obvious symmetry arguments described below, there are only two distinct cases to enumerate, $|S_n(123)|$ and $|S_n(132)|$. It happens that these two functions are equal, $|S_n(123)| = |S_n(132)| = c_n = \frac{1}{n+1}\binom{2n}{n}$.

Historically, these two enumerative results were obtained independently [13], [10]. The first satisfactory bijection between the two cases was presented by Rodica Simion and Frank Schmidt [20], and a second was given by Dana Richards [15].

In section two, we present a new bijective proof that $|S_n(123)| = |S_n(132)|$. This proof has the advantage that the enumerative result also follows naturally. In section three, we generalize the result of section two to show that $|S_n(1234)| = |S_n(1243)| = |S_n(2143)|$. Permutations which avoid the pattern 2143 have been studied elsewhere under the name *vexillary permutations*. In section four, we use our techniques to settle a conjecture of Shapiro and Getu, namely that $|S_n(3142, 2413)| = s_{n-1}$, the $n-1$-th Schröder number. The Schröder numbers have many connections with the Catalan numbers.

To conclude this introduction, we detail the symmetry arguments that reduce somewhat the number of problems which can sensibly be posed. A more natural way to think of definition 1.1 is in terms of the familiar *permutation matrices*. If $\pi = \{\pi(i)\}_{i=1}^{n}$, let $M(\pi)$ be the $n \times n$ matrix with entries $m_{i,j} = \delta_{i,\pi(j)}$ in terms of the Kronecker delta. Then a permutation π contains τ as a subsequence if the corresponding matrix $M(\pi)$ contains $M(\tau)$ *as a submatrix*.

In addition to making the definition clearer, this point of view makes trivial the following observation: $M(\pi)$ contains $M(\tau)$ iff the transpose matrix $M(\pi)^T$ contains $M(\tau)^T$. The same may be said for simultaneously reflecting both the matrices $M(\pi)$ and $M(\tau)$ in either a horizontal or a vertical mirror. These operations together generate the dihedral group acting on the permutation matrices in the obvious way. Since $M(\tau)^T = M(\tau^{-1})$, it follows immediately that $|S_n(\tau)| = |S_n(\tau^{-1})|$. The operations corresponding to reflecting the permutation matrix in a mirror carry $\tau = \{\tau(i)\}_{i=1}^{k}$ into $\tau^{|} = \{\tau(k+1-i)\}$ and $\tau^{-} = \{k+1-\tau(i)\}$.

For subsequences of length 3, these elementary considerations provide that $|S_n(123)| = |S_n(321)|$, and that $|S_n(231)| = |S_n(132)| = |S_n(213)| = |S_n(312)|$, reducing the enumerative problem from six to just two cases. For length 4, the number of cases is reduced from 24 to seven, these being represented by 1234, 1243, 1324, 1432, 1423, 2413 and 2143.

2. A Catalan tree

For a given forbidden permutation τ, we define recursively a rooted tree in which the vertices on the n-th level are identified with the permutations of $S_n(\tau)$. Let the root be the permutation $(1) \in S_1(\tau)$, and let each $\pi \in S_n(\tau)$ be a child of the permutation $\pi' \in S_{n-1}(\tau)$ obtained from π by deleting the largest element, n. (Clearly, a deletion cannot introduce a forbidden τ.) Call the resulting tree $T(\tau)$.

Given $\pi \in S_n$, and given $i \in [n+1]$, let

$$\pi^i = (p_1, p_2, \ldots, p_{i-1}, n+1, p_i, p_{i+1}, \ldots, p_n),$$

we will call this *inserting* $n+1$ *into the site* i.

Definition 2.1. *With respect to a particular* τ *we will call site* i *of* $\pi \in S_n(\tau)$ *an* active site *if the insertion of* $n+1$ *into site* i *creates a permutation* $\pi^i \in S_{n+1}(\tau)$.

Clearly the children of π in $T(\tau)$ are just the elements π^i as i ranges over the active sites of π relative to τ. In all proofs involving a structural description of a tree $T(\tau)$, we will rely heavily on the following observations, valid for all π, i.

(1) If π^i does not contain sequences of type τ, neither does π.

(2) If π^i contains sequences of type τ but π does not, then new element $n+1$ participates in all such sequences.

(3) $n+1$ is the largest element of π^i; therefore if it participates in a sequence of type τ, it does so as the largest element of τ.

(4) If the site in π between p_k and p_{k+1} is not active, then neither is the site between p_k and p_{k+1} in π^i.

In the following structural lemmas, we characterize the trees $T(123)$ and $T(132)$. It will here be convenient to label each vertex of $T(\tau)$ with the number of its children (equally, with the number of active sites in the associated permutation). We use the following notation, a *succession rule*, to connect the label of a parent with the label of its t children:

$$(p) \longrightarrow (c_1)(c_2)(c_3)\cdots(c_t).$$

The label (p) will usually include information about the value of t, but in general this will not be sufficient information. It is always our goal to introduce labels leading to a family of succession rules, each globally applicable throughout the tree, and together fully determining its structure. For the trees presently under consideration, one succession rule suffices:

Lemma 2.2. *In $T(123)$,*

$$(t) \longrightarrow (2)(3)(4)\cdots(t+1).$$

Proof. Let π be any node in $T(123)$ having label t. Note that all sites to the left of the first ascent in π are active, but none to the right are. So p_t is the leftmost element which is not a left-to-right minimum. (If $t = n+1$, then π is the descending permutation.)

If $n+1$ is inserted into the leftmost site, the new permutation $\pi^\star = (n+1, p_1, p_2, \ldots, p_n)$ has $t+1$ active sites, namely all those to the left of p_t. On the other hand, if $n+1$ is inserted elsewhere to the left of p_t, say to form π^s, then $n+1$ itself becomes the new leftmost ascent. Hence π^s receives the label s.

The children of π in $T(123)$ are $\pi^\star, \pi^2, \pi^3, \ldots, \pi^t$, and these receive the labels $t+1, 2, 3, \ldots, t$ respectively. □

Example 2.3. *Consider the following typical node of $T(123)$, in which the active sites are numbered from left to right:*

$$\pi = ({}_1 5_2 3_3 1_4 4_\times 2_\times)$$

If we form π^3, we are left with 3 active sites, those to the right vanishing:

$$\pi^3 = ({}_1 5_2 3_3 \mathbf{6}_\times 1_\times 4_\times 2_\times)$$

Lemma 2.4. *In $T(132)$,*

$$(t) \longrightarrow (2)(3)(4)\cdots(t+1).$$

Proof. The active sites are no longer necessarily the first t sites, so suppose they are numbered from the left $a_1, a_2, \ldots, a_t$.

If inserting $n+1$ creates a 132, then $n+1$ plays the part of 3. This cannot happen if $n+1$ becomes the leftmost element, so site 1 is always active ($a_1 = 1$). Furthermore $\pi^\star = \pi^1$ has label $t+1$, because the t active sites of π remain active (and one new one is introduced preceding the new element). For consider inserting $n+2$ in any site of π^1. A subsequence $(n+1, n+2, p_j)$ cannot be of type 132. Hence any 132 created must be of form $(p_i, n+2, p_j)$, but this would have caused the site to be inactive in π.

On the other hand, suppose $n+1$ is inserted into active site a_s for $s \geq 2$. This will render inactive all the sites to the *left* of the insertion, except for the first site. This is because $(p_1, n+2, n+1)$ would be a forbidden sequence. This leaves $t-(s-1)$ to the right of $n+1$, plus the leftmost site, a total of $t-s+2$.

The children of π in $T(132)$ thus receive the labels $t+1, t, \ldots, 3, 2$ respectively as the active sites are considered in order from left to right. □

Example 2.5. *Consider the following typical node of $T(132)$:*

$$\pi = ({}_1 5_2 3_\times 4_3 1_\times 2_4)$$

We insert at the third active site ($a_3 = 4$) to form π^4, we are left with 3 active sites, those to the left vanishing:

$$\pi^4 = ({}_1 5_\times 3_\times 4_\times \mathbf{6}_2 1_\times 2_3)$$

From these two lemmas, we conclude that $T(123)$ and $T(132)$ are isomorphic trees, and it is easy to see that the trees have trivial symmetry groups and so the isomorphism is unique. Since siblings receive distinct labels, a vertex can be uniquely determined in each tree by listing the labels of its ancestors.

Example 2.6. *We list on the left a node of $T(123)$, then the labels of its ancestors from the root down, then the corresponding node of $T(132)$.*

132	$(2,2,2)$	123
312	$(2,2,3)$	312
231	$(2,3,2)$	213
213	$(2,3,3)$	231
321	$(2,3,4)$	321

Example 2.7. *The vertices from the above examples, $(536142) \in T(123)$ and $(534612) \in T(132)$ are carried to each other by the unique bijection induced by the tree isomorphism.*

If a sequence of vertex labels $(f_1, f_2, \ldots, f_n)$, having the property that $f_1 = 2$ and $2 \leq f_i \leq f_{i-1} + 1$ is converted into a sequence $(a_1, a_2, \ldots, a_n)$ according to $a_i = i + 2 - f_i$, then the new sequence will be non-decreasing with $1 \leq a_i \leq i$. Such sequences are a familiar instance of the Catalan numbers, being naturally associated with non-diagonal-crossing lattice paths. We conclude

Theorem 2.8. *For all* $n \geq 1$, $|S_n(123)| = |S_n(132)| = c_n = \frac{1}{n+1}\binom{2n}{n}$.

References to the Catalan number are almost everywhere dense in the combinatorial literature; historically minded readers might be interested in [3] (but references go back at least to Euler and Segner, 1758). The first enumeration of $S_n(123)$ is in [13], for $S_n(132)$ see [10]. The first purely bijective proof that $|S_n(123)| = |S_n(132)|$ was presented in [20]. This bijection has the advantage of fixing the intersection of the two groups. The new bijection presented here does not; see elements 213 and 231 in the above table. On the other hand, we were able to produce the enumerative result with little extra effort.

We strengthen the enumerative result somewhat by counting the number of permutations avoiding 123, with length n and t active sites. First let $N(m, s)$ be the number of nodes on level $m+1$ with $m+2-s$ children. Small values of this function are given in table 1, the first column corresponding to the fact that the tree has one node on level one, labelled 2.

Since there will be exactly one permutation on level $n+1$ having label r for each permutation on level n having a label $\geq r-1$ it follows that (for all $m \geq 1$ and $1 \leq s \leq m$),

$$N(m,s) = \sum_{i=1}^{s} N(m-1,i) \tag{2.9}$$

$$= \sum_{i=1}^{s-1} N(m-1,i) + N(m-1,s) \tag{2.10}$$

$$= N(m,s-1) + N(m-1,s) \tag{2.11}$$

It follows that $N(n, s)$ counts the number of non-diagonal-crossing integer lattice paths from $(0, 0)$ to (m, s), the number of these obeying the same recurrence, and the initial conditions imposed by the first column. In closed form, the number of such paths is well-known to be $\binom{m+s}{s} - \binom{m+s}{s-1}$. Hence

Theorem 2.12. *The number of* $\pi \in S_n(123)$ *having* t *active sites relative to* 123 *is*

$$\binom{2n-t}{n-t+1} - \binom{2n-t}{n-t}$$

The rooted trees $T(\tau)$ introduced here seem to be entirely natural objects, but do not appear widely in the literature. The technique appears to be original to Chung, Graham, Hoggatt and Kleiman, who introduce it to examine the reduced Baxter permutations in [4]. This paper explicitly suggests application to other

classes of permutations, but we have not heard of any such work appearing in the 10 years between that paper and the beginnings of the present work.

The technique is now beginning to be more widely used, and the objects have aquired the name *generating trees*. Recent applications involving permutations include [23], [22], [8], [5]. If the objects generated are restricted permutations, we may wish to speak of *restricted permutation trees*. But there is no reason to stop here. Other classes of combinatorial objects for which generating trees have been produced include directed animals [25], binary trees [8], planar maps [5], and semiorders [below].

The particular object $T(123)$ (without the permutations attached) is an especially natural object; we hereby name it the *Catalan tree*. We imagine that this particular generating tree must appear in other settings; we would be interested to learn of any in addition to the following.

Although she does not use the fact, the mimimal semiorders introduced by Karen Stellpflug in [24], [1] are also generated by a Catalan tree. A partially ordered set is a semiorder iff it can be represented by a set of equal length open intervals in the real line, with the order relation $(a,b) < (c,d)$ iff $b \leq c$. A semiorder has representation number k if it has a representation in which all intervals have integer endpoints and the same length k, but has no such representation with intervals of length $k-1$.

Stellpflug shows how to obtain the minimal k-representable semiorders inductively by the process of duplicating one minimal element. If a minimal k-representable semiorder has r minimal elements, it produces by her construction r minimal $k+1$-representable semiorders, having variously $2, 3, 4, \ldots, r+1$ minimal elements. Noting that this process forms a Catalan tree amounts to an alternate proof of Stellpflug's result that the number of these k-representable semiorders is c_k.

3. Trees for forbidden sequences of length 4

We repeat the arguments of the above section for certain $\tau \in S_4$, retaining the definition of an active site, but augmenting the notion of a label on a node. We begin with the tree $T(1234)$. To each node $\pi \in S_n(1234)$ we associate an ordered pair (x,y) as follows. Let x be the position of the first ascent in π. In the terminology of Schensted [17], x is the index of the first element of the second basic subsequence (or $n+1$ if none exists). Let y be the number of active sites in π. In this instance, y is the index of the first element of the third basic subsequence (or $n+1$).

Lemma 3.1. *In* $T(1234)$,

$$(x,y) \longrightarrow (2,y+1)(3,y+1)\ldots(x,y+1)(x,x+1)(x,x+2)\ldots(x,y)(x+1,y+1)$$

Proof. Let π be a node of $T(1234)$ with label (x,y). The y active sites of π are the first y sites. By considering the new locations of the first elements of the

second and third basic subsequences we verify that π^i is associated in $T(1234)$ with

$$(x+1, y+1) \qquad \text{if } i = 1, \tag{3.2}$$

$$(i, y+1) \qquad \text{if } 2 \le i \le x, \tag{3.3}$$

$$(x, i) \qquad \text{if } x+1 \le i \le y. \tag{3.4}$$

□

Next consider the tree $T(1243)$, in which the nodes are again to be labelled (x, y) according as x is the position of the first ascent, and y is the number of active sites. The y active sites are no longer necessarily the first y sites on the left.

Lemma 3.5. *In $T(1243)$,*

$$(x, y) \longrightarrow (2, y+1)(3, y+1)\dots(x, y+1)(x, x+1)(x, x+2)\dots(x, y)(x+1, y+1)$$

Proof. Let the y active sites be numbered left-to-right as $a_1, a_2, \dots, a_y$. Note that the first x sites are active, as an $n+1$ here cannot find an increasing pair to its left to form a 1243.

The insertion of $n+1$ into site a_i of π splits it into two sites, both potentially active. We may verify that if a site was active in π, it remains active in π^{a_i} unless it falls to the right of x and to the left of position a_i. It is then easy to check that a_i has the associated pair

$$(x+1, y+1) \qquad \text{if } i = 1, \tag{3.6}$$

$$(i, y+1) \qquad \text{if } 2 \le i \le x, \tag{3.7}$$

$$(x, x+y+1-i) \qquad \text{if } x+1 \le i \le y. \tag{3.8}$$

□

We define the tree $T(2143)$ analogously, with x being the position of the first *ascent* and y being as usual the number of active sites. We can prove in an almost identical fashion that

Lemma 3.9. *In $S_n(2143)$,*

$$(x, y) \longrightarrow (2, y+1)(3, y+1)\dots(x, y+1)(x, x+1)(x, x+2)\dots(x, y)(x+1, y+1)$$

Combining the results of lemmas 3.1, 3.5 and 3.9, we hav the following theorem and its immediate corollary.

Theorem 3.10. $T(1234) \cong T(1243) \cong T(2143)$, *and these isomorphisms are unique.*

Proof. In each tree the root is labelled $(2,2)$. Applied recursively, the structural lemmas above ensure the isomorphisms.

From the labels (x,y), we can count the number of children of each node, and the number of children of each child. The sets of these are different for different pairs (x,y), and it is easy to check that no siblings ever have the same label. Therefore the trees have trivial symmetry groups. □

Corollary 3.11. $|S_n(1234)| = |S_n(1243)| = |S_n(2143)|$, *for all positive* n.

Regev has shown [14] that $|S_n(1234)|$ is asymptotic to $c \cdot \frac{9^n}{n^4}$ for a constant c. This result can now be applied to the sets $|S_n(1243)|$ and $|S_n(2143)|$ as well. The permutations of $|S_n(2143)|$ in particular have been extensively studied, as these are precisely the *vexillary permutations*. The vexillary permutations are relevant to the theory of Schubert polynomials, and therefore to the cohomological structure of flag manifolds. They are a superset of the *dominant* and of the *Grassmannian* permutations. For alternative characterizations of the vexillary permutations, see section 2 of Lascoux and Schützenberger [11] and chapter one of MacDonald [12], which also defines the dominant and Grassmannian permutations.

There is considerable work still to be done with restricted permutation trees, even for single forbidden subsequences of length 4. The following questions were posed in [**?**]. The first was answered in the affirmative by Stankova [23], the others are believed to be open.

Question 3.12. *Is* $T(4132) \cong T(3142)$*?*

Question 3.13. *Is* $T(1342)$ *a proper subtree of* $T(1432)$*?*

Question 3.14. *Is* $T(1423)$ *a proper subtree of* $T(1324)$*?*

This is also a convenient place to mention Ira Gessel's conjecture that $S_n(R)$ is P-recursive, for any set R of restrictions [6].

4. A Schröder tree

The Schröder numbers are closely related to the Catalan numbers, but less well known. Like the Catalan numbers, they have many combinatorial interpretations, including one in terms of lattice paths. For some references see [18], [2], [9], [16], [21]. The Catalan numbers, as seen above, count the number of non-diagonal-crossing lattice paths from the origin to (n,n) composed of the vectors $(0,1)$ and $(1,0)$. The Schröder numbers count the number of non-diagonal-crossing lattice paths from the origin to (n,n) composed of the vectors $(0,1)$, $(1,0)$ and $(1,1)$. They are thus the diagonal elements in table 2.

This characterization leads directly to the following formula for the n-th Schröder number, the i-th term being the number of paths using i diagonal steps $(1,1)$.

$$s_n = \sum_{i=0}^{n} \binom{2n-i}{i} c_{n-i}$$

Lou Shapiro and Seyoum Getu conjectured [7] that s_{n-1} is the number of permutations of length n having no subsequence of type 3142 or 2413. We settle this conjecture in the affirmative. The result is of interest, as it is the first non-trivial enumerative result to be obtained for any problem involving forbidden subsequences of length $k \geq 4$. There have since been others obtained by the author, by Stankova, and by Dulucq, Gire and Guibert.

As in the Catalan-tree case, we begin by defining recursively a rooted tree, $T(3142, 2413)$. Let the root be (1), and let each $\pi \in S_n(3142, 2413)$ be a child of the permutation $\pi' \in S_{n-1}(3142, 2413)$ obtained by deleting the largest element of π. Again, we label each vertex with the number of its children. We have the following structural lemma.

Lemma 4.1. *In* $T(3142, 2413)$,

$$(t) \longrightarrow (3)(4)\cdots(t)(t+1)(t+1)$$

Proof. Let π be an arbitrary element on level n of $T(3142, 2413)$, having label (t). We again consider active sites, but instead of clearing sites to left or right, an insertion will clear sites across the middle. By the middle of a permutation, we mean the position of the largest element, n. Note that the two sites immediately adjacent to n are always active: if placing $n+1$ here created a sequence of either type 3142 or 2413, then n would already play the same role in a like sequence, a contradiction.

Now divide the permutation π into *blocks* of contiguous elements, the blocks being separated by the active sites. We note that if a block B is right of n but left of C, then all the elements in B are larger than all those in C. Otherwise, a smaller element b in B and a larger element c in C would form a sequence n, b, c of type 312, rendering inactive the supposedly active site separating the blocks B and C. It follows that the values in the blocks to the right of n decline monotonically to the right of the middle. A symmetric claim can be made on the left of the middle.

In fact, we can say something stronger. Let v and w be two elements in the same block B right of the middle, and let x be an element left of the middle such that $v < x < w$. If v is to the left of w then x, n, w, v is of type $2, 4, 1, 3$, a contradiction. It follows that within B all the elements larger than x are toward the middle, and all those smaller than x are away from the middle. Now consider the site separating these two bunches of elements. If this site is inactive, then it falls within a sequence of type $31|2$ or $2|13$. Suppose such a sequence exists; those elements playing the roles of 1 and 2 in this sequence must be within the block B,

otherwise one of the active sites adjoining the block would also be deactivated. Since the near elements are all larger than the far elements, type 31|2 is excluded. For type 2|13, the element playing the role of 3 cannot be within the block B, since it would then be bigger than the element playing 2, nor can it be in a block to the right, by the remarks of the previous paragraph.

The conclusion is that each block is composed of consecutive elements from $[n]$. Therefore we can order the blocks by taking an arbitrary representative from each one; those $t-1$ elements just toward the middle from the t active sites will do. We call this the *inner subsequence*. This subsequence is unimodal (downwards), since it takes one representative from each block.

Number the active sites $0, 0, 1, 2, ..., t-2$ according as they are associated with the largest, next largest, etc. members of the inner subsequence. We claim that insertion of $n+1$ into the site thus numbered q produces a permutation with $t+1-q$ active sites. For insertion splits one active site into two sites (both automatically active because associated with the largest element), and then q sites are deactivated. The deactivated sites are precisely those which were numbered $< q$, except for the highest-numbered one in this set which is on the far side of the middle.

To see this, let the element associated with the insertion site be x, and assume w.l.o.g. that the insertion is left of the middle. Then $n+1, x, n$ forms a sequence of type 312, deactivating all sites between x and the middle. Likewise, the site right of the middle with the highest number $< q$ is associated with an element y which is greater than x and so the sequence $n+1, x, y$ provides the same service. But the site associated with y is itself not deactivated, nor are those further to the right of y (or left of x).

We check this last claim for sites right of y: if one of these is deactivated, it is because of a sequence involving $n+1$, therefore some $n+1, v, w$ with $v < w$ and v between x and y and w to the right of y. First note that w cannot be greater than y because it is located in a block to the right of y's block. If $w < x$, then x, v, y, w is of type 3142, a contradiction. If $w > x$, then the element of the inner subsequence in its block is likewise greater than x (and less than y). But this contradicts our choice of y as the smallest element of the inner subsequence larger than x and right of the middle. (The claim for sites to the left of x is easy to check from the descending block structure.)

□

As we did with the Catalan trees, we determine a recurrence for the number of permutations on the n-th level of the Schröder tree having t children. Again for simplicity, we let $m = n-2$ and $s = n+1-t$, then seek $f(m, s)$. From the lemma, we can see that

$$f(m, s) = (\sum_{i=0}^{s-1} f(m-1, i) + 2f(m-1, s).$$

Into this we substitute the formula for $f(m, s-1)$ to obtain $f(m,s) = f(m,s-1) - f(m-1,s-1) + 2f(m-1,s)$. We take for our boundary conditions $f(2,s) = 2\delta_{2s}$, since there are 2 permutations on level $n = 2$, each having 3 children. We illustrate this in table 3. In this figure, it is apparent that the diagonal elements are (with one exception) the same as those in table 2, which was governed by the recurrence $g(m,s) = g(m,s-1) + g(m-1,s-1) + g(m-1,s)$. The following very elegant proof that the diagonals are identical was provided by Ian Goulden [personal commmunication].

Lemma 4.2. *$f(i,i) = g(i,i)$ for all $i \geq 1$.*

Proof. In table 2, let $G = \sum_{i=0}^{\infty} g(i,i)z^i$ be the generating function for the diagonal elements. These are the number of non-diagonal-crossing paths from the origin to (i,i). Note that every such path returns to the diagonal for a first time. If this is at (j,j) for some $j < i$, we can decompose the path into a non-diagonal-crossing path of length $j-1$ from $(1,0)$ to $(j,j-1)$ and a non-diagonal-crossing path of length $i-j$ from (j,j) to (i,i). From this observation we derive the equation $G = zG^2 + zG + 1$. (The zG^2 term comes from the paths which begin with a step to the east; the zG term from those which begin with a step to the northeast.)

In the second table, begin by halving all the elements. Let $F = \frac{1}{2}\sum_{i=0}^{\infty} f(i,i)x^i$ be the generating function for the diagonal. F is the sum, taken over all paths beginning at the origin and using k of each of $(0,1)$ and $(1,0)$ and l of $(1,1)$, of $(2)^k(1)^k(-1)^l x^{k+l}$. By the same argument as above, therefore, $F = 2xF^2 + (-1)xF + 1$. Verify that substituting $F = \frac{G+1}{2}$ into this equation produces the familiar generating function equation for the Schröder numbers, as desired. □

Theorem 4.3. *$|S_n(3142,2413)| = g(n-1,n-1) = s_{n-1}$, the $n-1$-th Schröder number.*

Proof. Lemma 4.2 shows that the number of permutations of length $n+1$ avoiding 3142 and 2413 and having 3 active sites is s_{n-1}. But there is exactly one node labelled 3 on level $n+1$ for *every* node on level n (if $n > 2$). So $|S_n(3142,2413)|$ is also equal to s_{n-1}. □

It is interesting that we were able to find a simpler expression for the numbers $|S_n(3142,2413)|$ than for $|S_n(1234)|$. Why? We offer two suggestions.

First, the fact that 3142 is 2413 written in reverse means that $T(3142,2413)$ is invariant under mirror reflection (if embedded in the plane with siblings arranged in lexicographic order). Perhaps this symmetry somehow enables us to obtain a single-parameter labelling, where two parameters were necessary for $T(1234)$.

Second, we see that the permutation matrices corresponding to 3142 and 2413 form a complete symmetry class under the action of the dihedral group D_4. It seems combinatorially more natural to forbid this entire set of objects than to impose a single restriction. Indeed, Shapiro and Getu's attention was drawn to this case by considering a class of permutations characterized by avoiding these

two submatrices (and thus invariant under the action of the dihedral group). We ask, therefore, whether more natural enumerative results may be obtained by forbidding entire symmetry classes of permutations.

We offer a possible method for a second proof that $|S_n(3142, 2413)| = s_{n-1}$. From the generating function equation $G = zG^2 + zG + 1$, it is easy to derive the following recurrence for the Schröder numbers:

$$s_n = s_{n-1} + \sum_{j-1}^{n} s_{j-1} \cdot \ s_{n-j}$$

The corresponding equation for the Catalan numbers is

$$c_{n+1} = \sum_{j=0}^{n} c_j \cdot c_{n-j}$$

and can be used to prove that $|S_n(132)| = c_n$. To count the permutations of length $n+1$ which avoid 132, suppose $n+1$ is fixed in position $j+1$. Then there are j elements to the left, and $n-j$ to the right. Which these elements are is precisely determined by the observation that no element on the left can be smaller than any on the right. How they may be arranged is determined recursively: $|S_{j-1}(132)|$ ways on the left and $|S_{n-j}(132)|$ ways on the right. If either side is empty, we have the base case $S_0(123) = 1 = c_0$. Summing over j, the recurrence follows.

The data we have examined support the following conjecture.

Conjecture 4.4. *Among all the permutations of* $S_n(3142, 2413)$*, take those in which* 1 *appears in position* j*. For each of these, count* 1 *less than the number of active sites (with respect to* 3142 *and* 2413*). Then the total is* $s_{j-1} \cdot \ s_{n-j}$*.*

The Schröder recurrence 4 would follow immediately from this conjecture. By counting active sites, we are tallying the permutations of $S_{n+1}(3142, 2413)$. The terms inside the sum come from letting the position of n range along the permutation of length n. Subtracting 1 for each permutation produces the extra term of s_n outside the sum.

5. Knuth's deque permutations

It has been seen that the framework of forbidden subsequences unifies various problems from the literature which have to do with excluded configurations. For instance, the permutations which can be sorted by passage through a stack are those of $S_n(231)$ [10]. The matrices corresponding to $S_n(3142, 2413)$ are exactly those which do not fill up under 'bootstrap percolation' [19].

We offer a characterization of the permutations which can be sorted by an output restricted double-ended queue, the number of which is also a Schröder number. In [10], Knuth characterizes by $S_n(312)$ those permutations that can be *realized* using a stack. This is equivalent to saying that the permutations which

avoid the inverse permutation, 231, are those which can be *sorted* using a stack. The fashion recently is to adopt the latter viewpoint.

In the same source, Knuth introduces the permutations which can be sorted (realized) using an output-restricted double-ended queue. That is, we are given a queue with three permitted operations, S to insert an element on the left, Q to insert on the right, and X to remove from the left. We ask for which n-permutations can a sequence of $2n$ operations be specified which produces the identity.

Knuth shows that the number of such deque-sortable permutations is the Shröder number: $|\,\mathrm{Deq}_n\,| = s_{n-1}$. We will show that they form precisely the set $S_n(2431, 4231)$. First, note that neither 2431 nor 4231 is deque-sortable. This can be done by trying all sorting sequences of S, Q, and X, and noting that the identity is never produced. Then observe that any permutation containing a subsequence of one of these types cannot be deque-sorted either, because introducing new elements does nothing to undo the essential knot produced by these 4. Thus $\mathrm{Deq}_n \subseteq S_n(2431, 4231)$. We now show that the $|S_n(2431, 4231)| = s_{n-1}$, establishing the equality of the two sets.

In section one we remarked that $|S_n(\tau)| = |S_n(\tau^{-1})|$; therefore $|S_n(2431, 4231)| = |S_n(4132, 4231)|$. But we have

Lemma 5.1. *In* $T(4132, 4231)$,

$$(t) \longrightarrow (3)(4)\cdots(t)(t+1)(t+1)$$

Proof. Let π be an arbitrary node on level n of $T(4132, 4231)$, with label (t). A site is inactive, if and only if there is either a 231 or a 132 to its right. Thus if a site w is inactive, any site v left of w is also inactive, under the influence of the same 231 or 132 which deactivated w. It follows that the t active sites are those furthest to the right in π.

Inserting $n+1$ in the rightmost site creates no 231 or 132, hence deactivates no sites. It therefore gives rise to a child permutation with label $(t+1)$. On the other hand, inserting $n+1$ in any other active site (except the very leftmost) does create some 231s and/or 132s, the rightmost of which begins in the lefthand neighbour of the insertion. Hence all sites left of this point are rendered inactive. If the insertion is into site number $2, 3, \ldots, t$, counting from the right, only the $3, 4, \ldots, t+1$ right of the lefthand neighbour of the insertion remain active. □

Therefore $T(4132, 4231) \cong T(3142, 2413)$, whence $|S_n(2431, 4231)| = s_{n-1}$. We conclude

Theorem 5.2.

$$\mathrm{Deq}_n = S_n(2431, 4231)$$

6. Acknowledgments

Some results in this paper are drawn from my doctoral thesis [?], prepared under the expert supervision of Richard P. Stanley. I am grateful to Bruce Sagan for reviewing an early copy of the manuscript, and to the referees for many helpful suggestions.

References

1. K. P. Bogart and K. Stellpflug Mandych. Discrete representation theory for semiorders. Technical Report PMA-TR93-104, Dartmouth College, March 1993.
2. J. Bonin, L.W. Shapiro, and R. Simion. Some q-analogues of the Schroder numbers arising from combinatorial statistics on lattice paths. *J. of Statistical Planning and Inference*, 34:35–55, 1993.
3. E. Catalan. Note sur une équation aux différences finies. *J. de Mathématiques pures et appliquées*, 3:508–516, 1838.
4. F.R.K. Chung, R.L. Graham, V.E. Hoggatt Jr, and M. Kleiman. The number of baxter permutations. *J. of Combinatorial Theory Series A*, 24:382–394, 1978.
5. S. Dulucq, S. Gire, and J. West. Permutations with forbidden subsequences and non-separable planar maps. *Disc. Math.*, to appear, 1995.
6. I. Gessel. Symmetric functions and p-recursiveness. *J. of Comb. Theory, Ser. A*, 53:257–285, 1990.
7. S. Getu and L. Shapiro. (personal communication).
8. S. Gire. *Arbres, permutations à motifs exclus et cartes planaire: quelques problèmes algorithmiques et cominatoires.* PhD thesis, University of Bordeaux, 1993.
9. D. Gouyou-Beauchamp and B. Vauquelin. Deux propriétés combinatoires des nombres de Schroder. *Theoretical Inf. Appl.*, 22:361–388, 1988.
10. D.E. Knuth. *The Art of Computer Programming, vol. 1, second edition.* Addison-Wesley, Reading MA, 1973.
11. A. Lascoux and M.-P. Schutzenberger. Schubert polynomials and the Littlewood-Richardson rule. *Letters in Math. Phys.*, 10:111–124, 1985.
12. I.G. MacDonald. *Schubert Polynomials.* LaCIM, Montréal, 1991.
13. P.A. MacMahon. *Combinatory Analysis.* Chelsea, New York, 1960 (Originally published by Camb. Univ. Press, London, 1915/16).
14. A. Regev. Asymptotic values for degrees associated with strips of young diagrams. *Adv. in Math.*, 41:115–136, 1981.
15. D. Richards. Ballot sequences and restricted permutations. *Ars Combinatoria*, 25:83–86, 1988.
16. D.G. Rogers and L. Shapiro. Some correspondences involving the Schroder numbers and relations. In D.A. Holton, editor, *Combinatorial Mathematics, Lect. Notes in Math., vol. 686.* Springer-Verlag, Berlin, 1978.
17. C. Schensted. Longest increasing and decreasing subsequences. *Can. J. of Math.*, 13:179–191, 1961.
18. E. Schroder. Vier combinatorische probleme. *Zeitschrift fur Mathematik und Physik*, 15:361–376, 1870.
19. L.W. Shapiro and A.B. Stephens. Bootstrap percolation, the schroder numbers, and the n-kings problem. *SIAM J. Disc. Math.*, 4:275–280, 1991.
20. R. Simion and F.W. Schmidt. Restricted permutations. *Eur. J. of Combinatorics*, 6:383–406, 1985.
21. N.J.A. Sloane. *A Handbook of Integer Sequences.* Academic Press, New York, 1973.

22. Z. Stankova. Classification of forbidden subsequences of length 4. *Eur. J. of Comb.*, (submitted).
23. Z. Stankova. Forbidden subsequences. *Disc. Math.*, 132:291–316, 1994.
24. K. Stellpflug. *Discrete Representations of Semiorders.* PhD thesis, Dartmouth College, 1990.
25. J. West. A catalogue of forbidden-subsequence results. (preprint).

Department of Computer Science, Bordeaux University, Bordeaux, France
E-mail address: west@labri.u-bordeaux.fr

Consequences of Arithmetic for Set Theory

Lorenz HALBEISEN [1]
Department of Mathematics,
ETH Zürich, Switzerland

Saharon SHELAH [2]
Institute of Mathematics,
Hebrew University Jerusalem, Israel

Abstract

In this paper, we consider certain cardinals in ZF (set theory without AC, the Axiom of Choice). In ZFC (set theory with AC), given any cardinals $\mathcal{C}$ and $\mathcal{D}$, either $\mathcal{C} \leq \mathcal{D}$ or $\mathcal{D} \leq \mathcal{C}$. However, in ZF this is no longer so. For a given infinite set A consider $seq^{1\text{-}1}(A)$, the set of all sequences of A without repetition. We compare $\left|seq^{1\text{-}1}(A)\right|$, the cardinality of this set, to $\left|\mathcal{P}(A)\right|$, the cardinality of the power set of A. What is provable about these two cardinals in ZF? The main result of this paper is that $\text{ZF} \vdash \forall A(\left|seq^{1\text{-}1}(A)\right| \neq \left|\mathcal{P}(A)\right|)$ and we show that this is the best possible result. Furthermore, it is provable in ZF that if B is an infinite set, then $\left|fin(B)\right| < \left|\mathcal{P}(B)\right|$, even though the existence for some infinite set B^* of a function f from $fin(B^*)$ onto $\mathcal{P}(B^*)$ is consistent with ZF.

Section 0: *Introduction, Definitions and Basic Theorems*

Introduction: In ZFC the cardinality of ordinal numbers plays an important role, since by AC each set has the cardinality of some ordinal.
We use "alephs" for the cardinalities of ordinals. Thus in ZFC each cardinal number is an aleph. However this need not be the case in ZF.
If we have a model M of ZF in which the axiom of choice fails, then we have more cardinals in M than in a model V of ZFC, even if we have fewer sets in M than in V. (This occurs when the choice-functions are not all in M). This is because the ordinals are in M and hence the alephs as well.

[1] Parts of this work are of the first author's Diplomarbeit at the ETH Zürich. He is grateful to his supervisor, Professor H. Läuchli.

[2] Research partially supported by the Basic Research Fund, Israeli Academy; Publ. No. 488

In this paper we are interested in the relation between three cardinals arising in connection with a set S, namely,

1) the cardinality of the power set of S
2) the cardinality of the finite subsets of S
3) the cardinality of the finite sequences without repetition of S

This section contains definitions and basic theorems provable in ZF.
In the next section we present two relative consistency proofs illustrating possible relations between these cardinals.
The last two sections contain three results provable in ZF. The proofs of these are based on the same idea originally from E. Specker, who used it to prove that the axiom of choice follows from the generalised continuum hypothesis [Sp1]. Assuming the existence of a function we derive a contradiction to Hartogs' Theorem.

Because we do not use AC, our proofs are constructive. But we will see that sometimes arithmetic is powerful enough for our constructions, making it an adequate substitute for AC.

Cardinals: A *cardinal number* $\mathcal{C}$ is the equvalence class of all sets which have the *same size*. (Two sets are said to have the same size *iff* there is a bijection between them.)

Alephs: A cardinal number $\mathcal{C}$ is an *aleph* if it contains a well-ordered set.

We use calligraphic letters to denote cardinals and $\aleph$'s to denote the alephs.
We denote the cardinality of the set s by $\left|s\right|$.

Relations between cardinals: We say that the cardinal number $\mathcal{C}$ is less than or equal to the cardinal number $\mathcal{D}$ *iff* there are sets $c \in \mathcal{C}$, $d \in \mathcal{D}$ and a 1-1 function from c into d.
In this case we write $\mathcal{C} \leq \mathcal{D}$. We write $\mathcal{C} < \mathcal{D}$ for $\mathcal{C} \leq \mathcal{D}$ and $\mathcal{C} \neq \mathcal{D}$.
If $c \in \mathcal{C}$, $d \in \mathcal{D}$ and we have a function from d onto c, then we write $\mathcal{C} \leq^* \mathcal{D}$.

We also need some well-known facts provable in ZF:

Hartogs' Theorem: Given a cardinal $\mathcal{C}$ there is a least aleph, $\aleph(\mathcal{C})$, such that $\aleph(\mathcal{C}) \not\leq \mathcal{C}$.

Proof: See [Je1] p.25 ■

Cantor-Bernstein Theorem: If $\mathcal{C}$ and $\mathcal{D}$ are cardinals with $\mathcal{C} \leq \mathcal{D}$ and $\mathcal{D} \leq \mathcal{C}$, then $\mathcal{C} = \mathcal{D}$.

Proof: See [Je1] p.23 ■

Cantor Normal Form Theorem: Any ordinal α can be written as

$$\alpha = \sum_{i=0}^{j} \omega^{\alpha_i} \cdot k_i$$

with $\alpha \geq \alpha_0 > \alpha_1 > \ldots > \alpha_j \geq 0$, $1 \leq k_i < \omega$, $0 \leq j < \omega$.
Proof: See [Ba] p.57*ff* ■

Corollary 1: The Cantor Normal Form does not depend on AC.
Proof: The proof of the Cantor Normal Form requires no infinite choices. ■

Corollary 2: If $\alpha = \sum_{i=0}^{j} \omega^{\alpha_i} \cdot k_i$ is a Cantor Normal Form, then define $\overleftarrow{\alpha}$ by

$$\overleftarrow{\alpha} := \sum_{i=j}^{0} \omega^{\alpha_i} \cdot k_i = \omega^{\alpha_0} \cdot k_0.$$

Then (in ZF) $|\alpha| = |\overleftarrow{\alpha}|$
Proof: See [Ba] p.60 ■

Corollary 3: For any ordinal α, ZF implies the existence of the following bijections.

$$\begin{array}{lcll} F^{\alpha}_{seq^{1\text{-}1}} & : & \alpha \longrightarrow seq^{1\text{-}1}(\alpha) & (=: \text{finite sequences of } \alpha \text{ without repetition}) \\ F^{\alpha}_{seq} & : & \alpha \longrightarrow seq(\alpha) & (=: \text{finite sequences of } \alpha\,) \\ F^{\alpha}_{fin} & : & \alpha \longrightarrow fin(\alpha) & (=: \text{finite subsets of } \alpha\,) \end{array}$$

Proof: Use the Cantor Normal Form Theorem, Corollary 2, order the finite subsets of α and then use the Cantor-Bernstein Theorem. ■

Section 1: *Consistency results*

In this section we work in the Mostowski permutation model to derive some relative consistency results. The permutation models are models of ZFA, set theory with atoms, (see [Je2] p.44*ff*).
The atoms $x \in A$ may also be considered to be sets which contain only themselves, this means: $x \in A \Rightarrow x = \{x\}$ (see [Sp2] p.197 or [La] p.2).
Thus the permutation models are models for ZF without the axiom of foundation.
However, the Jech-Sochor Embedding Theorem (see [Je] p.208*ff*) implies consistency results for ZF.

In the permutation models we have a set of atoms A and a group $\mathcal{G}$ of permutations of A. Let $\mathcal{F}$ be a normal filter on $\mathcal{G}$ (see [Je] p.199). We say that x is *symmetric* if the group $\mathrm{sym}_{\mathcal{G}}(x) := \{\pi \in \mathcal{G} : \pi(x) = x\}$ belongs to $\mathcal{F}$.

Let us further assume that $\mathrm{sym}_{\mathcal{G}}(a) \in \mathcal{F}$ for every atom a, that is, that all atoms are symmetric (with respect to $\mathcal{G}$ and $\mathcal{F}$) and let $\mathcal{B}$ be the class of all hereditarily symmetric objects.

The class $\mathcal{B}$ is both a permutation model and a transitive class: all atoms are in $\mathcal{B}$ and $A \in \mathcal{B}$. Moreover, $\mathcal{B}$ is a transitive model of ZFA.

Given a finite set $E \subset A$, let $\mathrm{fix}_{\mathcal{G}}(E) := \{\pi \in \mathcal{G} : \ \pi a = a \text{ for all } a \in E\}$ and let $\mathcal{F}$ be the filter on $\mathcal{G}$ generated by $\{\mathrm{fix}_{\mathcal{G}}(E) : \ E \subset A \text{ is finite}\}$.
$\mathcal{F}$ is a normal filter and x is symmetric *iff* there is a finite set of atoms E_x such that $\pi(x) = x$ whenever $\pi \in \mathcal{G}$ and $\pi a = a$ for each $a \in E_x$. Such an E_x is called a support for x.

Now the Mostowski model is constructed as follows: (see also [Je2] p.49*ff*)

1) The set of atoms A is infinite.
2) R is an order-relation on A.
3) With respect to R, A is a dense linear ordered set without endpoints.
4) Let Aut_R be the group of all permutations of A such that for all atoms $x, y \in A$ and each $\pi \in \mathrm{Aut}_R$, if Rxy then $R\pi(x)\pi(y)$.
5) Let $\mathcal{F}$ be generated by $\{\mathrm{fix}(E) : \ E \subset A \text{ is finite }\}$.

We will write $x < y$ instead of Rxy.

The subsets of A (in the Mostowski model) are symmetric sets. Hence each subset of A has a finite support.

If $x \subseteq A$ (in the Mostowski model) and x has non-empty support E_x, then an $a \in E_x$ may or may not belongs to x.

Fact: If $b \notin x \cup E_x$ and there are two elements $a_0, a_1 \in E_x$ with $a_0 < b < a_1$ such that $\forall c(a_0 < c < a_1 \to c \notin E_x)$, then $\forall c(a_0 < c < a_1 \to c \notin x)$.

Otherwise we construct a $\pi \in \mathrm{Aut}_R$ such that $\pi a_i = a_i$ for all $a_i \in E_x$ and $\pi c = b$. Then $\pi(x) \neq x$, which is a contradiction.

We can similarly show that if $a_0 < b < a_1$ and $b \in x \setminus E_x$, then $\forall c(a_0 < c < a_1 \to c \in x)$. The cases when $\neg\exists a_1(a_1 \in E_x \ \wedge \ b < a_1)$ or $\neg\exists a_0(a_0 \in E_x \ \wedge \ b > a_0)$ are similar.

Hence, given a finite set $E \subset A$ ($\left|E\right| =: n$), we can construct $2^n \cdot 2^{n+1} = 2^{2n+1}$ subsets $x \subseteq A$ such that E is a support of x.

Given a finite subset E of A, consider the set $\mathcal{E}$ of subsets of A with support E. We use R to order $\mathcal{E}$ as follows. Given $E_1 = \{a_1, \ldots, a_n\}$ and $E_2 = \{a_1, \ldots, a_n, \ldots, a_{n+k}\}$ with $a_i < a_j$ whenever $i < j$ and given $x \in \mathcal{E}$, if x is the l^{th} subset with support E_1, then x is also the l^{th} subset with support E_2.

Finally, we define the function F: $\mathit{fin}(A) \longrightarrow \mathcal{P}(A)$ by
$E \longmapsto \left|E\right|^{th}$ subset of A constructible with support E.

It is easy to see that F is onto.

If $E \subset A$ is finite, then use R to order the subsets of E and use the corresponding lexicographic order on the set of permutations of subsets of E. The set of permutations of subsets of E is isomorphic to $seq^{1\text{-}1}(E)$. In fact we can order $seq^{1\text{-}1}(E)$ for each finite $E \subset A$.

For each subset $x \subseteq A$ there is exactly one smallest support E_x (=:supp(x)).

If $\big|\text{supp}(x)\big| = n$, then put $\overline{x} := \big|\{y \subseteq A : \text{supp}(y) = \text{supp}(x)\}\big| \leq 2^{2n+1}$ and for $l \leq \overline{x}$ define as above the l^{th} element of $\{y \subseteq A : \text{supp}(y) = \text{supp}(x)\}$.
We say that: "$y \subseteq A$ is the l^{th} subset of A with support supp(x)".

Now choose 24 distinct elements $a_0, \ldots, a_{23} \in A$ and define $A_{24} := \{a_0, \ldots, a_{23}\}$.
A simple calculation shows that

$$\text{if } n \geq 12, \text{ then } 2 \cdot 2^{2n+1} < n! \qquad (*)$$

Take a finite subset E of A and let $y \subseteq A$ be the l^{th} subset of A with $\text{supp}(y) = E$. Put $D := \text{supp}(y) \Delta A_{24}$ (where Δ denotes symmetric difference) and $d := \big|D\big|$.
Define the function $Seq_A : \mathcal{P}(A) \longrightarrow seq^{1\text{-}1}(A)$ by

$$Seq_A(y) := \begin{cases} \text{the } l^{th} \text{ permutation of supp}(y) & \text{if } \big|\text{supp}(y)\big| \geq 12, \\ \text{the } (d! - l - 1)^{th} \text{ permutation of supp}(y) & \text{otherwise.} \end{cases}$$

Seq_A is well defined because of $(*)$ and $d \geq 13$.
It is easy to see that Seq_A is 1-1. If there is a bijection between $\mathcal{P}(A)$ and $seq^{1\text{-}1}(A)$, then we find an ω-sequence$^{1\text{-}1}$ in A using an analogous construction. But this is a contradiction (see section 3).

Even more is true in the Mostowski model, $(\mathcal{A} := \big|\text{Atoms}\big|)$,

$$\mathcal{A} < fin(\mathcal{A}) < \mathcal{P}(\mathcal{A}) < seq^{1\text{-}1}(\mathcal{A}) < fin(fin(\mathcal{A})) < seq(\mathcal{A}) < \mathcal{P}(\mathcal{P}(\mathcal{A})).$$

(We omit the proof).

Our interest here is in the following result.

Theorem 1: The following theories are equiconsistent:
(i) ZF
(ii) ZF + $\exists \mathcal{A}(\mathcal{P}(\mathcal{A}) < seq^{1\text{-}1}(\mathcal{A}))$
(iii) ZF + $\exists \mathcal{A}(\mathcal{P}(\mathcal{A}) \leq^* fin(\mathcal{A}))$

Proof: It was shown above that in the Mostowski model there is a cardinal $\mathcal{A}$, namely the cardinality of the set of atoms, for which both (ii) and (iii) hold.
Unfortunately, the Mostowski model is only a model of ZFA. But it is well-known that Con(ZF)⇒Con(ZFC) and the Jech-Sochor Embedding Theorem provides a model of (ii) and (iii). ■

Theorem 2: The following theories are equiconsistent:
(i) ZF
(ii) ZF + $\exists\mathcal{A}(seq(\mathcal{A}) < fin(\mathcal{A}))$

Proof:
By the Jech-Sochor Embedding Theorem it is enough to construct a permutation model $\mathcal{B}$ in which there is a set A, such that:

(a) there is a 1-1 function from $seq(A)$ into $fin(A)$,
(b) there is no bijection between $seq(A)$ and $fin(A)$.

We construct by induction on $n \in \omega$ the following:

(α) $A_0 := \{\{\emptyset\}\}$; $Sq_0(\{\emptyset\})$:=the empty sequence;
G_0 := the group of all permutations of A_0.

Let k_n be the number of elements of G_n, and $\mathcal{E}_n$ be the set of sequences of A_n in length less or equal than n which are not in range(Sq_n), then

(β) $A_{n+1} := A_n \dot\cup \{(n+1, \zeta, i) : \zeta \in \mathcal{E}_n \text{ and } i < k_n + k_n\}$.

(δ) Sq_{n+1} is a function from A_{n+1} to $seq(A_n)$ defined as follows:

$$Sq_{n+1}(x) = \begin{cases} Sq_n(x) & \text{if } x \in A_n, \\ \zeta & \text{if } x = (n+1, \zeta, i) \in A_{n+1} \setminus A_n. \end{cases}$$

(γ) G_{n+1} is the subgroup of the group of permutations of A_{n+1} containing all permutations h such that for some $g_h \in G_n$ and $j_h < k_n + k_n$ we have

$$h(x) = \begin{cases} g_h(x) & \text{if } x \in A_n, \\ (n+1, g_h(\zeta), i +_n \ j_h) & \text{if } x = (n+1, \zeta, i) \in A_{n+1} \setminus A_n. \end{cases}$$

Where $g_h(\zeta)(m) := g_h(\zeta(m))$ and $+_n$ is the addition modulo $k_n + k_n$.

Let $A := \bigcup\{A_n : n \in \omega\}$ and $Sq := \bigcup\{Sq_n : n \in \omega\}$, then Sq is a function from A onto $seq(A)$.
Further define for each natural number n partial functions f_n from A to $A \cup \{\emptyset\}$ as follows. If $lg(x)$ denotes the length of $Sq(x)$ and $n < lg(x)$, then $f_n(x) := Sq(x)(n)$, otherwise let $f_n(x) = \emptyset$.
Let Aut(A) be the group of all permutations of A.
Then $\mathcal{G} := \{H \in \text{Aut}(A) : \forall n \in \omega(H|_{A_n} \in G_n)\}$ is a group of permutations of A.
Let $\mathcal{F}$ be the normal filter on $\mathcal{G}$ generated by $\{\text{fix}(E) : E \subset A \text{ is finite}\}$ and $\mathcal{B}$ be the class of all hereditarily symmetric objects.
Now $A \in \mathcal{B}$ and for each $n \in \omega$, supp$(f_n) = \emptyset$, hence f_n belongs to $\mathcal{B}$, too.
Now define on A a equivalence relation as follows,

$$x \sim y \quad \textit{iff} \quad \forall n(f_n(x) = f_n(y)).$$

Facts:

1. Every equivalence class of A is finite.
   (Because of each A_n is finite, hence each k_n).

2. $seq(A) = \{\varsigma_x : x \in A\}$ where $\varsigma_x(n) := f_n(x)$, (if $f_n(x) \neq \emptyset$).

3. For every finite subset B of A, there are finite subsets C, Y of A and a natural number $k > 1$ such that $B \subseteq C$, $\forall x \in A \setminus C$ ($\left|\{H(x) : H \in \mathrm{fix}_{\mathcal{G}}(C)\}\right| > k$) and $\left|\{H[Y] : H \in \mathrm{fix}_{\mathcal{G}}(C)\}\right| = k$.
   (Choose A_n ($n \geq 1$) such that $B \subseteq A_n$ and let $C := A_n$. Let $k := k_n + k_n$ and $Y := \{(n+1, \zeta, i) \in A_{n+1} : \ i \text{ is even}\}$. Then Y has exactly two images under $\{h : h \in \mathrm{fix}_{\mathcal{G}}(C)\}$ and $\forall x \in A \setminus C(\ \left|\{h(x) : \ h \in \mathrm{fix}_{\mathcal{G}}(C)\}\right| \geq k_{n+1} + k_{n+1})$.)

Now the function

$$\begin{array}{rccc} \Psi \ : & seq(A) & \longrightarrow & fin(A) \\ & \varsigma & \longmapsto & \{x : \ \varsigma_x = \varsigma\} \end{array}$$

is a 1-1 function in $\mathcal{B}$ from $seq(A)$ into $fin(A)$ (by the facts 1 and 2).
Hence (a) holds in $\mathcal{B}$.

To prove (b), assume there is a 1-1 function $\Phi \in \mathcal{B}$ from $fin(A)$ into $seq(A)$.

Let B be a support of Φ and let C, Y, k be as in fact 3.

If the sequence $\Phi(Y)$ belongs to $seq(C)$, then for some $H \in \mathrm{fix}_{\mathcal{G}}(C)$, $H[Y] \neq Y$, hence $\Phi(H[Y]) \neq \Phi(Y)$. But this contradicts that H maps Φ to itself, (by definition of C, Y and H).

Otherwise there exists an $m \in \omega$ such that $x := \Phi(Y)(m)$ does not belong to the set C.
Hence $\left|\{H(x) : \ H \in \mathrm{fix}_{\mathcal{G}}(C)\}\right| > k$ and $\left|\{H[Y] : \ H \in \mathrm{fix}_{\mathcal{G}}(C)\}\right| = k$, (by fact 3).
Every $H \in \mathrm{fix}_{\mathcal{G}}(C)$ maps Φ to itself, hence $\Phi(Y)$ to $\Phi(H[Y])$. So we have a mapping from a set with k members onto a set with more than k members.
But this is a contradiction. ■

Section 2: ZF $\vdash$ ($\left|fin(S)\right| < \left|\mathcal{P}(S)\right|$) for any infinite set S.

Theorem 3: ZF $\vdash fin(\mathcal{C}) < \mathcal{P}(\mathcal{C})$

Proof: Take $S \in \mathcal{C}$. The natural map from $fin(S)$ into $\mathcal{P}(S)$ is a 1-1 function, hence $\left|fin(S)\right| \leq \left|\mathcal{P}(S)\right|$ is always true.
Assume that there is a bijective function $B : \ fin(S) \longrightarrow \mathcal{P}(S)$. Then, given any ordinal α, we can construct an α-sequence$^{1\text{-}1}$ in $fin(S)$. But this contradicts Hartogs' Theorem.

First we construct an ω-sequence$^{1\text{-}1}$ in $\mathit{fin}(S)$ as follows:

$S \in \mathcal{P}(S)$ and, because S is infinite, $S \notin \mathit{fin}(S)$.
But $B^{-1}(S) \in \mathit{fin}(S)$. So put $s_0 := B^{-1}(S)$ and $s_{n+1} := B^{-1}(s_n)$ $(n \in \omega)$.
Then the set $\{s_i : i < \omega\}$ is an infinite set of finite subsets of S and the sequence $\langle s_0, s_1, \ldots, s_n, \ldots \rangle_\omega$ is an ω-sequence$^{1\text{-}1}$ in $\mathit{fin}(S)$.

If we have already constructed an α-sequence$^{1\text{-}1}$ $\langle s_0, s_1, \ldots, s_\beta, \ldots \rangle_\alpha$ in $\mathit{fin}(S)$ (with $\alpha \geq \omega$), then we define an equivalence relation on S by

$$x \sim y \;\; \textit{iff} \;\; \forall \beta < \alpha (x \in s_\beta \leftrightarrow y \in s_\beta)$$

Take $x \in S$ and suppose that $\mu < \alpha$. Define

$$\begin{aligned} D_{x,\mu} &:= \bigcap_{\iota < \mu} \{s_\iota : x \in s_\iota\} \\ g(x) &:= \{\mu < \alpha : x \in s_\mu \wedge (s_\mu \cap D_{x,\mu} \neq D_{x,\mu})\}. \end{aligned}$$

Fact: Given $x, y \in S$, $g(x) = g(y) \Leftrightarrow x \sim y$.
(In other words $x^\sim = y^\sim$ whenever $g(x) = g(y)$).
Hence there is a bijection between $\{x^\sim : x \in S\}$ and $\{g(x) : x \in S\}$.
Furthermore, $g(x) \in \mathit{fin}(\alpha)$.

Since $\{g(x) : x \in S\} \subseteq \mathit{fin}(\alpha)$, apply F^α_{fin} to obtain $F^\alpha_{\mathit{fin}}[\{g(x) : x \in S\}] \subseteq \alpha$.
Let γ be the order-type of $F^\alpha_{\mathit{fin}}[\{g(x) : x \in S\}]$. Then $\gamma \leq \alpha$ and for each $g(x)$ we obtain an ordinal number $\eta(g(x)) < \gamma$.
Each s_ι $(\iota < \alpha)$ is the union of at most finitely many equivalence classes. Thus there is a 1-1 function

$$\begin{aligned} h : \quad \alpha &\longrightarrow \mathit{fin}(\gamma) \\ \iota &\longmapsto \{\xi : \eta(g(x)) = \xi \wedge x \in s_\iota\}. \end{aligned}$$

Since F^γ_{fin} is a bijection between $\mathit{fin}(\gamma)$ and γ, $F^\gamma_{\mathit{fin}} \circ h$ is a 1-1 function from α into γ and because $\gamma \leq \alpha$ we also have a 1-1 function from γ into α.
The Cantor-Bernstein Theorem yields a bijection between γ and α and hence a bijection G from $\{\eta(g(x)) : x \in S\}$ onto $\{s_\iota : \iota < \alpha\}$.
Now consider the function $\Gamma := B \circ G \circ \eta \circ g$ from S into P(S):

$$\Gamma : \; S \xrightarrow{g} \{g(x) : x \in S\} \xrightarrow{\eta} \{\eta(g(x)) : x \in S\} \xrightarrow{G} \{s_\iota : \iota < \alpha\} \xrightarrow{B} \mathcal{P}(S)$$

Fact: $S_\alpha := \{x \in S : x \notin \Gamma(x)\} \notin \{B(s_\iota) : \iota < \alpha\}$.

Otherwise Take $S_\alpha = B(s_\beta)$ (for some $\beta < \alpha$).
We identify each $x^\sim$ with $g(x)$ using the bijection above.
Then there is a $g(x)$ such that $G \circ \eta((g(x)) = s_\beta$.
Now if $y \in x^\sim$ then $\Gamma(y) = S_\alpha$.
But $y \in S_\alpha \Leftrightarrow y \notin \Gamma(y) \Leftrightarrow y \notin S_\alpha$, which is a contradiction.

But $S_\alpha \subseteq S$ and $B^{-1}(S_\alpha) =: s_\alpha \in \mathit{fin}(S)$ with $s_\alpha \notin \{s_\iota : \iota < \alpha\}$ and we have an $(\alpha+1)$-sequence$^{1\text{-}1}$ in $\mathit{fin}(S)$, namely $\langle s_0, s_1, \ldots, s_\beta, \ldots, s_\alpha \rangle_{\alpha+1}$.

We now see that for an infinite set S there is no bijection between $\mathit{fin}(S)$ and $\mathcal{P}(S)$ and this completes the proof. ■

We note the following facts.

Given a natural number n,	$\mathrm{ZF} \vdash (n \times \mathit{fin}(\mathcal{C}) = \mathcal{P}(\mathcal{C}) \rightarrow n = 2^k$ for a $k \in \omega)$.
Moreover, for each $k \in \omega$	$\mathrm{Con(ZF)} \Rightarrow \mathrm{Con(ZF} + \exists \mathcal{C}(2^k \times \mathit{fin}(\mathcal{C}) = \mathcal{P}(\mathcal{C}))$
	(If $k = 0$, then this is obvious for finite cardinals.)

Sketch of the proof:

For the consistency result, consider the permutation model with an infinite set of atoms A and the empty relation. Then the automorphism group is the complete permutation group. It is not hard to see that any subset of A in this model is either finite or has a finite complement. Take a natural number k and consider (in this model) the set $k \times A$. The cardinality of the set $\mathcal{P}(k \times A)$ is the same as that of the set $2^k \times \mathit{fin}(A)$.

To prove the other fact, assume that n is a natural number which is not a power of 2 and that for some infinite set S there is a bijection B between $n \times \mathit{fin}(S)$ and $\mathcal{P}(S)$. Use the function B to construct an ω-sequence$^{1\text{-}1}$ in $\mathit{fin}(S)$. Then, using Theorem 3, $\omega \leq \mathit{fin}(S) < \mathcal{P}(S)$ and it is easy to see that $n \times \mathit{fin}(S) \leq \mathit{fin}(S) \times \mathit{fin}(S) =: \mathit{fin}(S)^2$. Then $\omega < \mathcal{P}(S) = n \times \mathit{fin}(S) \leq \mathit{fin}(S)^2$ contradicts the fact that if $\aleph_0 \leq \mathcal{P}(\mathcal{C})$, then for any natural number n, $\mathcal{P}(\mathcal{C}) \not\leq \mathit{fin}(\mathcal{C})^n$. (Here $\aleph_0$ denotes the cardinality of ω). The proof of this fact is similar to the proof of Theorem 3. ■

Section 3: $\mathit{seq}^{1\text{-}1}(S)$, $\mathit{seq}(S)$ and $\mathcal{P}(S)$ when S is an arbitrary set.

We show that $\mathrm{ZF} \vdash \mathit{seq}^{1\text{-}1}(\mathcal{C}) \neq \mathcal{P}(\mathcal{C})$ for every cardinal $\mathcal{C} \geq 2$. But we first need the following result.

Lemma: $\mathrm{ZF} \vdash \aleph_0 \leq \mathcal{P}(\mathcal{C}) \rightarrow \mathcal{P}(\mathcal{C}) \not\leq \mathit{seq}^{1\text{-}1}(\mathcal{C})$.

Proof:

Take $S \in \mathcal{C}$. Then, because $\aleph_0 \leq \mathcal{P}(\mathcal{C})$, we have a 1-1 function $f_\omega : \omega \longrightarrow \mathcal{P}(S)$.
Assume that there is a 1-1 function $J : \mathcal{P}(S) \longrightarrow \mathit{seq}^{1\text{-}1}(S)$.
Then $J \circ f_\omega : \omega \longrightarrow \mathit{seq}^{1\text{-}1}(S)$ is also 1-1 and we get an ω-sequence$^{1\text{-}1}$ in $\mathit{seq}^{1\text{-}1}(S)$.
Using this ω-sequence$^{1\text{-}1}$ in $\mathit{seq}^{1\text{-}1}(S)$ we can easily construct an ω-sequence$^{1\text{-}1}$ in S.
If we already have constructed an α-sequence$^{1\text{-}1}$ $\langle s_0, s_1, \ldots, s_\beta, \ldots \rangle_\alpha$ $(\alpha \geq \omega)$ in S, put $T := \{s_\iota : \iota < \alpha\}$. This gives rise to bijective functions,

$$\begin{array}{lrcl} h_0 : & T & \longrightarrow & \alpha \\ h_1 : & \mathit{seq}^{1\text{-}1}(\alpha) & \longrightarrow & \mathit{seq}^{1\text{-}1}(T). \end{array}$$

Let J^{-1} be the inverse of J and denote the inverse of F^{α}_{seq} by $\mathrm{inv}F^{\alpha}_{seq}$.
Further define

$$\Gamma := J^{-1} \circ h_1 \circ \mathrm{inv}F^{\alpha}_{seq} \circ h_0$$

Note: $\mathrm{dom}(\Gamma) \subseteq T$ and $\mathrm{range}(\Gamma) \subseteq \mathcal{P}(S)$ (because J is 1-1).

Fact: $S_\alpha := \{x \in S : \ x \notin \Gamma(x)\} \notin J^{-1}[seq^{1\text{-}1}(T)]$.
Assume not, then $x \in S$ such that $J(S_\alpha) = h_1 \circ \mathrm{inv}F^{\alpha}_{seq} \circ h_0(x)$ yields a contradiction.

Because $J(S_\alpha) \notin seq^{1\text{-}1}(T)$, the sequence $J(S_\alpha)$ has a first element which is not in T, say s_α. Finally, the sequence $\langle s_0, s_1, \ldots, s_\alpha \rangle_{\alpha+1}$ is an $(\alpha+1)$-sequence$^{1\text{-}1}$ in S.
So the existence of a 1-1 function $J : \ \mathcal{P}(S) \longrightarrow seq^{1\text{-}1}(S)$ contradicts Hartogs' Theorem. ■

Theorem 4: If $\mathcal{C} \geq 2$ is any cardinal, then $\mathrm{ZF} \vdash (seq^{1\text{-}1}(\mathcal{C}) \neq \mathcal{P}(\mathcal{C}))$

Proof:
By the Lemma it is enough to prove that if $\mathcal{C} \geq 2$, then $seq^{1\text{-}1}(\mathcal{C}) = \mathcal{P}(\mathcal{C}) \Rightarrow \aleph_0 \leq \mathcal{C}$.
For finite cardinals $\mathcal{C} \geq 2$ the statement is obvious. So let $S \in \mathcal{C}$ be an infinite set and assume that there is a bijective function

$$B : seq^{1\text{-}1}(S) \longrightarrow \mathcal{P}(S).$$

We use this function to construct an ω-sequence$^{1\text{-}1}$ in S.
Let $n^\star$ $(n < \omega)$ be the cardinality of $seq^{1\text{-}1}(n)$.
Then $0^\star = 1;\ 1^\star = 2;\ 2^\star = 5; \ldots\ 16^\star = 56,874,039,553,217; \ldots$ (see [Sl], No. 589), and, in general

$$n^\star = \sum_{i=0}^{n} \frac{n!}{i!}$$

We begin by choosing four distinct elements of S, $S_4 := \{s_0, s_1, s_2, s_3\}$ and use these elements to construct a 4-sequence$^{1\text{-}1}$ $\langle s_0, s_1, s_2, s_3 \rangle_4$ in S. This sequence will give us an order on the set $seq^{1\text{-}1}(S_4)$ (e.g. we order $seq^{1\text{-}1}(S_4)$ by length and lexicographically).
If we have already constructed an n-sequence$^{1\text{-}1}$ $\langle s_0, s_1, \ldots, s_{n-1} \rangle_n$ in S $(n \geq 4)$, put $S_n := \{s_i : \ i < n\}$. Then $B[seq^{1\text{-}1}(S_n)] \subseteq \mathcal{P}(S)$ has cardinality $n^\star$.
We now define an equivalence relation on S by

$$x \sim y \ \ \textit{iff} \ \ \forall q \in seq^{1\text{-}1}(S_n)(x \in B(q) \leftrightarrow y \in B(q)).$$

It is easy to see that for each $q \in seq^{1\text{-}1}(S_n)$

$$B(q) \text{ is the } \textit{disjoint union} \text{ of } \textit{less} \text{ than } n^\star \text{ equivalence classes.} \qquad (1)$$

Take the above order on $seq^{1\text{-}1}(S_n)$. This induces an order on the set of equivalence classes eq:= $\{x^\sim : \ x \in S\}$ and also an order on $\mathcal{P}(\mathrm{eq})$.

If there is a first $r \in \mathcal{P}(\text{eq})$ such that $r \notin B[seq^{1\text{-}1}(S_n)]$, then $q_r := B^{-1}(r)$ is a "new" sequence in S. This is $q_r \notin seq^{1\text{-}1}(S_n)$ and we choose the first element s_n of q_r which is not in S_n.
Hence, the sequence $\langle s_0, s_1, \ldots, s_n \rangle_{n+1}$ is now an $(n+1)$-sequence$^{1\text{-}1}$ in S.
If there is an $s_i \in S_n$ such that $\{s_i\} \notin B[seq^{1\text{-}1}(S_n)]$, then use $B(\{s_i\})$ to construct an $(n+1)$-sequence$^{1\text{-}1}$ in S.
Otherwise our construction stops at S_n and we write $\text{stop}(S_n)$.

Our construction only stops if

$$\begin{array}{ll} \text{for each } s_i \in S_n: & \{s_i\} \in \text{eq and} \\ \text{for each } r \in \mathcal{P}(\text{eq}) & \text{there is a } q_r \in seq^{1\text{-}1}(S_n) \text{ such that } B(q_r) = r. \end{array}$$

If κ ($\kappa < \omega$) is the cardinality of eq, then 2^κ is the cardiniality of $\mathcal{P}(\text{eq})$ and because of (1) we have $\text{stop}(S_n) \Rightarrow 2^\kappa = n^\star$.
It is known that $0^\star = 1 = 2^0$; $1^\star = 2 = 2^1$; $3^\star = 16 = 2^4$ and $n^\star$ is a power of 2 for some $n > 3$, then n has to be bigger than 10^8.
If there are only finitely many $k, n < \omega$ such that $2^k = n^\star$, then there is a least n_0 such that $2^k = {n_0}^\star$ and $\forall n > n_0(\neg\text{stop}(S_n))$.

Refining our construction removes the need for this strong arithmetic condition.

Assume $\text{stop}(S_n)$.
If $x \notin S_n$ then let $S^x_{n+1} := S_n \dot\cup \{x\}$ and $S^x_{n+k} := S^x_{n+1} \dot\cup \{Y\}$ with Y of cardinality $k-1$. Because (n is even)$\Leftrightarrow$ ($n^\star$ is odd) and $\text{stop}(S_n)$, we cannot have $\text{stop}(S^x_{n+1})$ for any $x \notin S_n$.

Now we recommence our construction with the set S^x_{n+1} and construct an $(n+k)$-sequence$^{1\text{-}1}$ $\langle s_0, s_1, \ldots, s_{n+k-1} \rangle_{n+k}$ $(k \geq 2)$ in S.
If the construction also stops at the $(n + \text{stop})^{th}$ stage at the set $S^x_{n+\text{stop}}$ $(\text{stop} \geq 2)$, then we write S^x instead of $S^x_{n+\text{stop}}$.

If there is an $x \in S$ such that S^x is infinite, then our construction does not stop when we recommence with S^x_{n+1} and we can construct an ω-sequence$^{1\text{-}1}$ in S. But this contradicts our Lemma.

So there cannot be such an x and each $x \in S$ is in exactly one *finite* set S^x. If for each $x \in S$, S^x is the union of some elements of eq, then S must be finite, because eq is finite. But this contradicts our assumption that S is infinite.

A subset of S is called *good* if it cannot be written as the union of elements of eq.

Consider the set $T_{\min} := \{x : S^x \text{ is good and of least cardinality}\}$ and let m_{T} be the cardinality of S^x for some x in $T_{\min}$. Further for $x \in T_{\min}$ let $x_= := \{y : S^y = S^x\}$ (this elements of S^x we cannot distinguish) and $m_=$ denote the least cardinality of the sets $x_=$.
If $T_{\min}$ is good, use $B^{-1}(T_{\min})$ to construct an $(n+1)$-sequence$^{1\text{-}1}$ in S.
Otherwise take $x \in T_{\min}$. Because S^x is good

$$B^{-1}(S^x) \notin seq^{1\text{-}1}(S_n).$$

Thus there is a first y in $B^{-1}(S^x)$ which is not in S_n. It is easy to see that $S^y \subseteq S^x$ and if $S^y \neq S^x$ then S^y is not good (because of $x \in T_{\min}$).
But then $B^{-1}(S^x \setminus S^y) \notin seq^{1\text{-}1}(S^y)$ and we may proceed.

So for each $x \in T_{\min}$ construct an m_{T}-sequence$^{1\text{-}1}$ SEQ^x in S such that

$$S^x = S^y \implies \mathrm{SEQ}^x = \mathrm{SEQ}^y.$$

For $i < m_{\mathrm{T}}$ define

$$Q_i := \{s \in S : \ s \text{ is the } i^{th} \text{ element in } \mathrm{SEQ}^x \text{ for some } x \in S\}$$

Assume there is some $j < m_{\mathrm{T}}$ such that Q_j is good. Then $B^{-1}(Q_j) \notin seq^{1\text{-}1}(S_n)$. But $B^{-1}(Q_j) \notin seq^{1\text{-}1}(S)$ and we get an $(n+1)$-sequence$^{1\text{-}1}$ in S.

It remains to justify our assumption.

Note that if for some $i \neq j$, $z \in Q_i \cap Q_j$, then S^z cannot be good. Furthermore for each $x \in T_{\min}$ there is exactly one i_x such that $x \in Q_{i_x}$ and if $z, y \in x_=, \ z \neq y$, then $i_x \neq i_y$. If there are no good Q_i's, $m_=$ cannot exceed κ, (the cardinality of eq). But by the following this is a contradiction:

An easy calculation modulo 2^r $(r \leq 4)$ shows that for each n, if $2^r | n^\star$, then $2^r | (n+2^r)^\star$ and $2^r \not| (n+t)^\star$ if $0 < t < 2^r$.

Assume there is a smallest k $(k \geq 4)$ such that $2^{k+1} | n^\star$ and $2^{k+1} | (n+t)^\star$ for some t with $0 < t < 2^{k+1}$.

Then, because $2^k | 2^{k+1}$, we have $2^k | n^\star$ and $2^k | (n+t)^\star$. Since k is by definition the smallest such number, we know that t must be 2^k.

$$\begin{array}{lrcccccl}
(n+2^k)^\star = \sum\limits_{i=0}^{n+2^k} \frac{(n+2^k)!}{i!} = & 1 \cdot 2 \cdot & \ldots & \cdot 2^k \cdot (2^k+1) \cdot & \ldots & \cdot (2^k+n) & & (1) \\
+ & 2 \cdot & \ldots & \cdot 2^k \cdot & \ldots & \cdot (2^k+n) & & (2) \\
 & & \ddots & & & & & \vdots \\
+ & & & 2^k \cdot & \ldots & \cdot (2^k+n) & & (2^k) \\
 & & & & \ddots & & & \vdots \\
+ & & & & & (2^k+n) & & (2^k+n) \\
+ & & & & & 1 & & (2^k+n+1)
\end{array}$$

It is easy to see that 2^{k+1} divides lines $(1) - (2^k)$ since $k \geq 2$ and $n \geq 2$.

If we calculate the products of lines $(2^k+1) - (2^k+n+1)$, then we only have to consider sums which are not obiviously divisible by 2^{k+1}. So, for a suitable natural number ε we have

$$(n+2^k)^\star = 2^k \cdot (\sum_{j=0}^{n-1} \sum_{i>j}^{n} \frac{n!}{i \cdot j!}) + n^\star + 2^{k+1} \cdot \varepsilon. \tag{2}$$

We know that $2^{k+1} | n^\star$ with $n \geq 3, \ k \geq 4$. And because $n^\star$ is even n has to be odd. If j is $n-1, \ n-2$ or $n-3$, then $\sum\limits_{i>j}^{n} \frac{n!}{i \cdot j!}$ is odd. Moreover, if $0 \leq j \leq (n-4)$, then $\sum\limits_{i>j}^{n} \frac{n!}{i \cdot j!}$ is even. So $\sum\limits_{j=0}^{n-1} \sum\limits_{i>j}^{n} \frac{n!}{i \cdot j!}$ is odd. Hence $2^{k+1} \not| (n+2^k)^\star$, (by (2) and $2^{k+1} | n^\star$).

We return to the proof.

We know that if $2^k = n^\star$ and $(n+t)^\star$ is a power of 2, then 2^k divides t. $(**)$

Take $x \in T_{\min}$ such that $\left|x_=\right| = m_=$. If $y \in S^x$, then

(i) $\left|S^y\right| = n + t_y$ with 2^κ divides t_y,
(ii) either $y \in x_=$ or S^y is not good.

This is because $2^\kappa = n^\star$ and $(**)$.

Hence (for a suitable natural number ε) $m_{\mathrm{T}} = \left|S^x\right| = n + 2^\kappa \cdot \varepsilon + m_=$ (by (ii)), and 2^κ divides $m_=$ (by (i)).

But this implies that $m_=$ must be larger than κ, which justifies our assumption. ■

The statement obtained when $seq^{1\text{-}1}$ is replaced by seq is much easier to prove:

Theorem 5: ZF $\vdash$ $seq(\mathcal{C}) \neq \mathcal{P}(\mathcal{C})$ for all cardinals such that $\emptyset \notin \mathcal{C}$.

Proof: Take $S \in \mathcal{C}$. First note the fact that if $\aleph_0 \leq \mathcal{C}$, then $seq(\mathcal{C}) \not\geq \mathcal{P}(\mathcal{C})$.

(The proof is the same as the proof of the Lemma, except that we can skip the first lines of the proof of the Lemma).

Assume there is a bijection B from $seq(S)$ onto $\mathcal{P}(S)$. Choose an $s_0 \in S$, and define a 1-1 function f_{s_0} from ω into $\mathcal{P}(S)$ by $i \mapsto \xi_i := B(\langle s_0, s_0, \ldots, s_0\rangle)$ (i-times). Use the ξ_i's to construct pairwise disjoint subsets $c_i \subseteq S$ $(i < \omega)$.

Given an n-sequence$^{1\text{-}1}$ $\langle s_0, s_1, \ldots, s_{n-1}\rangle_n$ in S, let $S_n := \{s_i : i < n\}$ and the natural order on S_n induce a well-ordering on the set $seq(S_n)$ with order type ω. Then there is a bijection $h : \omega \longrightarrow seq(S_n)$. Now the function $\Gamma := B \circ h$ is a 1-1 function from ω into $\mathcal{P}(S)$ and $t := \dot{\cup}\{c_i : c_i \subseteq \Gamma(i)\} \notin \{\Gamma(k) : k \in \omega\}$.

Hence $B^{-1}(t)$ is a sequence in S which does not belongs to S_n. Choose $s_n \in S$ to be the first element of $B^{-1}(t)$ not in S_n. Then $\langle s_0, s_1, \ldots, s_n\rangle_{n+1}$ is an $(n+1)$-sequence$^{1\text{-}1}$ in the set S.

We thus construct an ω-sequence$^{1\text{-}1}$ in S, contradicting the previous fact. ■

References

[Ba] H. Bachmann, Transfinite Zahlen, Springer-Verlag, Berlin, 1967

[Je1] Th. Jech, Set Theory, Academic Press, New York, 1978

[Je2] Th. Jech, The Axiom of Choice, North-Holland Publ. Co., Amsterdam, 1973

[La] H. Läuchli, Auswahlaxiom in der Algebra, Comment. Math. Helv., vol.37, 1962, pp.1-18

[Sl] N.J.A. Sloane, A Handbook of Integer Sequences, Academic Press, New York, 1973

[Sp1] E. Specker, Verallgemeinerte Kontinuumshypothese und Auswahlaxiom, Archiv der Mathematik 5, 1954, pp.332-337

[Sp2] E. Specker, Zur Axiomatik der Mengenlehre, Zeitschr. f. math. Logik und Grundl. der Math. 3, 1957, pp.173-210

Chu spaces: Complementarity and Uncertainty in Rational Mechanics

Vaughan Pratt*
Dept. of Computer Science
Stanford University, CA 94305
pratt@cs.stanford.edu

July 3, 1994

1 Introduction to Chu spaces

1.1 Basic notions

A *Boolean Chu space* $\mathcal{A} = (X, \models, A)$ consists of two sets X and A and a binary relation $\models \subseteq X \times A$ from X to A. We call the elements $x, y, \ldots$ of X *states* or *opens*, and the elements $a, b, \ldots$ of A *points*, *propositions*, or *events*. We read $x \models a$ as the Boolean-valued assertion "state x satisfies point (proposition, event) a." Viewed as an event, a is understood as the proposition "event a has happened."

A Chu space can be depicted naturally as a matrix. Figure 1 gives some illustrative examples that we shall refer to in the sequel.

We define $\mathrm{row}_A(x) = \{a \mid x \models a\}$, the set of those column indices a containing a 1 at row x, and dually $\mathrm{col}_A(a) = \{x \mid x \models a\}$. When row_A is an injective function (no repeated rows) we call $\mathcal{A}$ *extensional*, and when col_A is injective (no repeated columns) we call $\mathcal{A}$ T_0 by analogy with topological spaces. When row_A is the identity function on X, X must be a subset of 2^A; we call such a Chu space *normal*, and write it as simply (X, A), $\models$ then being inferrable as the converse $\in^\smile$ of set membership. A normal space is automatically extensional but need not be T_0. The Chu spaces of Figure 1 are all extensional and T_0 but not normal unless v is identified with $\{b, c\}$ etc.

The *dual* $\mathcal{A}^\perp$ of a Chu space $\mathcal{A} = (X, \models, A)$ is simply its transpose $(A, \models^\smile, X)$ as a matrix, with points turned into states and vice versa. We denote the converse of a binary relation $R \subseteq X \times Y$ as $R^\smile$, defined as the subset $\{(y, x) \mid xRy\}$ of $Y \times X$, The dual of a normal space is T_0 but never normal (assuming set membership is well-founded) even if extensional; dualizing a second time however restores it to a normal space since dualizing merely interchanges the two index sets, whatever they are.

*This work was supported by ONR under grant number N00014-92-J-1974.

	a	b	c
s	0	0	0
t	0	0	1
u	0	1	0
v	0	1	1
w	1	0	0
x	1	0	1
y	1	1	0
z	1	1	1

(a)

	a	b	c
s	0	0	0
w	1	0	0
x	1	0	1
y	1	1	0
z	1	1	1

(b)

	a	b	c
s	0	0	0
w	1	0	0
y	1	1	0
z	1	1	1

(c)

	a	b	c
s	0	0	0
x	1	0	1
y	1	1	0
z	1	1	1

(d)

	a	b	c	d
w	0	0	0	0
x	0	0	1	1
y	0	1	0	1
z	0	1	1	0

(e)

	a	b	c	d
x	0	0	1	1
y	0	1	0	1

(f)

Figure 1. Representative Chu spaces presented as matrices.

Boolean Chu spaces are closely related to S. Vickers' notion of a *topological system* [Vic89, p.52], the definition of which Vickers begins thus.

> Let A be a frame; we call its elements *opens*. Let X be a set; we call its elements *points*. Finally, let $\models$ be a subset of $X \times A$. If $(x, a) \in \models$ we write $x \models a$ and say x *satisfies* a.

With Chu spaces, the two differences from a topological system are that A is not a frame but merely a set, and that points and opens are interchanged. The first difference is essential and will be seen to make the category of Chu spaces not only a substantial and very useful extension of that of topological systems but a self-dual one at that. The second difference is a mathematically inessential matter of interpretation: whereas the propositions of a topological system transform contravariantly, those of a Chu space transform covariantly. While there is no formal reason to prefer one orientation over the other, our choice for Chu spaces will be seen to be a natural consequence of thinking of states as *possible* worlds (or paintings of individuals) accumulating disjunctively and propositions as *necessary* constraints (or obstacles, or individuals) accumulating conjunctively.

A *Chu space over* a set K is a triple $(X, \models, A)$ where $\models : X \times A \to K$ is a K-valued function of states and points. Chu spaces over $K = \mathbf{2} = \{0, 1\}$ are thus the Boolean Chu spaces defined above. We generalize row_A by replacing "subset of" by "function to K;" thus $\mathrm{row}_A(x) : A \to K$ is the function whose value at $a \in A$ is $x \models a$, and dually for col_A. For $K = \mathbf{2}$ this amounts to replacing subsets of A by their characteristic functions (functions from A to $\mathbf{2}$). We may understand $\mathrm{row}_A(x)$ as a painting of the points of the space $\mathcal{A}$ with colors from the "palette" K. It is not so natural to view $\mathrm{col}_A(a)$ as a painting of states since in the real world we perceive all the points at once but the states only one at a time. This perspective is relative however; it is appropriate for programs when they are running, but the perspective is reversed when writing the program: its states all exist at the same time in the text of the program. More generally, in game-oriented views

of situations, certain contexts demand seeing things from one player's viewpoint, others require taking the opposing side, and yet anothers call for the neutrality of an umpire.

Chu transforms. For any fixed K, a *Chu transform* $(f,g) : \mathcal{A} \to \mathcal{B}$ between Chu spaces $\mathcal{A} = (X, \models_A, A)$, $\mathcal{B} = (Y, \models_B, B)$ over K consists of two functions $f : A \to B$ and $g : Y \to X$ satisfying the following *adjointness condition*: for all $a \in A$ and $y \in Y$, $g(y) \models_A a \;=\; y \models_B f(a)$. Chu transforms compose as $(f',g')(f,g) = (f'f, gg')$; this composition is evidently associative and has $(1_A, 1_X)$ as its identity at each Chu space $(X, \models, A)$. Chu spaces over K thereby form a category which we denote $\mathbf{Chu}_K$; we abbreviate $\mathbf{Chu}_2$ to $\mathbf{Chu}$.

The adjointness condition may be rephrased in terms of rows, namely $\mathrm{row}_A(g(y)) = \mathrm{row}_B(y) \circ f$, verified by the calculation $\mathrm{row}_A(g(y))(a) = g(y) \models_A a = y \models_B f(a) = (\mathrm{row}_B(y) \circ f)(a)$. That is, every row of B when composed with f must be some row of A, with g a function selecting a suitable row index. It follows that *when the source of a Chu transform (f,g) is extensional, g is determined by f.* We will often restrict attention to extensional Chu spaces, where it suffices to specify only the f component of a Chu transform. In this respect Chu transforms behave like homomorphisms of relational structures, consisting of just a function between their carriers meeting a certain condition. The dual form of this rephrasing is $\mathrm{col}_A \circ g = \mathrm{col}_B(f(a))$, rarely used; it shows that when the target of a Chu transform is T_0, f is determined by g.

Duality. Transposition of the objects of $\mathbf{Chu}_K$ extends in the obvious way to its morphisms, sending (f,g) to (g,f). This makes transposition a functor from $\mathbf{Chu}_K$ to $\mathbf{Chu}_K{}^{\mathrm{op}}$. Transposition is of course an involution, $\mathcal{A}^{\perp\perp} = \mathcal{A}$, and hence an isomorphism of $\mathbf{Chu}_K$ and $\mathbf{Chu}_K{}^{\mathrm{op}}$. This makes transposition a *duality*[1] from $\mathbf{Chu}_K$ to itself, making $\mathbf{Chu}_K$ a *self-dual* category.

Carrier and cocarrier. Whereas an algebra or relational structure has only the one underlying set or *carrier*, a Chu space has both a carrier A and a *cocarrier* X. The cardinality of a Chu space is that of its carrier as usual; we refer to the cardinality of the cocarrier as the cocardinality of $\mathcal{A}$. We define the underlying-set functor $U_K : \mathbf{Chu}_K \to \mathbf{Set}$ as $U_K(X, \models, A) = A$, $U_K(f,g) = f$, and the underlying-antiset functor $V_K : \mathbf{Chu}_K \to \mathbf{Set}^{\mathrm{op}}$ as $V_K(X, \models, A) = X$, $V_K(f,g) = g$. We may understand $\mathbf{Set}^{\mathrm{op}}$ as the category of sets X and *antifunctions*, defined as binary relations $R \subseteq X \times Y$ whose converse $R^\smile \subseteq Y \times X$ is a function $g : Y \to X$. We shall sometimes refer to sets that transform by antifunctions as *menus* to emphasize their disjunctive quality; thus a Kripke structure is a menu of possible worlds.

Seen in this light, a pair $(f,g) : (X, \models_A, A) \to (Y, \models_B, B)$ of functions is a Chu transform just when the following square whose edges are binary relations, and which compose standardly as such, commutes.

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \models_A{}^\smile\downarrow & & \downarrow\models_B{}^\smile \\ X & \xrightarrow{g^\smile} & Y \end{array}$$

Let $\mathbf{Rel}$ denote the category of sets and their binary relations, and let $\mathbf{Rel}^2$ denote its "arrow category," whose objects are the morphisms of $\mathbf{Rel}$ and whose morphisms are the commuting squares of $\mathbf{Rel}$ (e.g. the above diagram). Then $\mathbf{Chu}$ may be understood as the subcategory (not full) of $\mathbf{Rel}^2$ such that each morphism (square) has a function for its upper edge and an antifunction for its lower, as in the diagram.

We regard X as a *disjunctive* set of alternatives ("possible worlds"), and A as a *conjunctive*

[1] A duality is a contravariant equivalence between two categories C and D, meaning an equivalence between C and D^{op}. The duality here is the more intimate relationship of actual isomorphism of categories.

set of entities all simultaneously present. This is the distinction a player of a game draws between her own moves, which form a menu of *opportunities* one of which she chooses, and those of her opponent, forming a set of *risks* all of which she must guard against.

We shall also take this as an abstraction of mind-body duality, with a set understood as a pure body, a menu as a pure mind, and a Chu space as a binary relation from a mind to a body.

Whereas functions may *identify* (when not injective) and *adjoin* (when not surjective), the corresponding operations performed by antifunctions are respectively *copy* and *delete*. Viewed as editing operations between sets of memory locations, the latter come more naturally to computer scientists than the former. In identifying two cells, how does one combine their contents? And what should a newly adjoined cell contain? No such awkward questions arise when copying and deleting cells. Exercise: noting that "contents-of" is a function from cells to values, relate all this to the op in Hom : $\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \to \mathbf{Set}$.

An attractive feature of the category **Set** is that it is both complete and cocomplete (has all limits and colimits). Hence so does $\mathbf{Set}^{\mathrm{op}}$, making it equally attractive. One might think to work in the larger category **Rel**, a self-dual category offering not only both functions and antifunctions in the one category but all binary relations. However although **Rel** has all products and coproducts it does not have other limits or colimits (equalizers, coequalizers, pullbacks, pushouts, etc.).

Chu as a self-dual extension of both **Set** and $\mathbf{Set}^{\mathrm{op}}$ (comonadic in the former, the latter monadic in **Chu**) inherits their bicompleteness. In choosing a self-dual category to do mathematics in, this constitutes a significant advantage for **Chu** over either **Rel** or $\mathbf{Rel}^2$. It is our thesis that Tarski's vision of mathematics founded on binary relations would be more effective if founded on Chu spaces.

We will show later that the category of κ-ary relational structures and their homomorphisms (standardly understood) embeds fully and concretely in $\mathbf{Chu}_{2^\kappa}$. This allows us to think of $\mathbf{Chu}_K$ as a combined logic-and-algebra of relations of arity up to κ. In particular Boolean Chu spaces correspond in this way to the monadic predicate calculus. This makes Chu spaces a rich mathematical playground.

There are only 2^n unary relations on an n-element set, but 2^{2^n} Boolean operations and hence extensional Chu spaces. This indicates that the lower bound of the previous paragraph on the utility of Chu spaces is a very conservative one: most Chu spaces over 2^κ do not correspond to κ-ary relational structures. Yet inspection of T_0 extensional Chu spaces of cardinality up to 3 reveals all of them to be useful for something. (A two-day computer search found 2,4,8,64,3828,37320288 isomorphism classes of extensional T_0 Boolean Chu spaces of cardinality respectively 0 through 5.)

The next two subsections present Boolean Chu spaces from respectively a logical and an algebraic viewpoint.

1.2 Boolean Chu spaces as Boolean operations

The following logical view of *extensional* Chu spaces allows us to identify them with their properties expressed as a theory. An important application is to the realization, or full concrete embedding, of categories of various order-theoretic kinds—posets, topological spaces, semilattices, distributive lattices, etc.—in the single self-dual category **Chu**.

The Chu space $\mathcal{A} = (X, \models, A)$ may be understood as the Boolean operation (abstract formula) φ_A in set A of propositional variables whose satisfying assignments are those determined by X: the operation is true whenever the variables are assigned truth values according to some row of $\mathcal{A}$,

and false for all other truth assignments. The Chu spaces of Figure 1 determine in this way the respective operations *true* (in variable set $A = \{a, b, c\}$), $b \vee c \ \rightarrow \ a$, $(c \rightarrow b) \wedge (b \rightarrow a)$, $a \equiv (b \vee c)$, $\neg a \wedge (d \equiv (b \oplus c))$, and $\neg a \wedge (b \oplus c) \wedge d$ (where $a \oplus b = \neg(a \equiv b)$, i.e. exclusive-or). Operations are abstract in the sense that logical equivalence is identity: $a \wedge b$ and $b \wedge a$ are the same operation.

Conversely every Boolean operation in set A of variables uniquely determines the extensional Chu space $(X, \models, A)$ where X is the set of satisfying assignments and $x \models a$ is the truth of a in assignment x. A concrete formula realizing this abstract operation may be formed as the disjunctive normal form formula having one disjunct per row x; each such disjunct is the conjunction of literals, one per $a \in A$, with a appearing positively or negatively according to whether $x \models a$ or not.

By composing these two translations we see that this operation representation of an arbitrary Chu space $\mathcal{A} = (X, \models, A)$ captures it as $(\mathrm{row}_A(X), \models, A)$. The original column index set is lost, being replaced by its image $\mathrm{row}_A(X) = \{\mathrm{row}_A(x) \mid x \in X\}$ under row_A.

The operation φ_A may be understood as a complete description of the space $\mathcal{A}$. We define a *property* of $\mathcal{A}$ to be any Boolean consequence of φ_A; we think of the properties of $\mathcal{A}$ as the *theory* of $\mathcal{A}$, with φ_A as its single axiom and strongest property. Since φ_A is the (infinite if necessary) conjunction of the properties of $\mathcal{A}$, φ_A and the properties of $\mathcal{A}$ determine each other.

A property of a Chu space $\mathcal{A}$ itself determines a Chu space, obtainable from $\mathcal{A}$ by adjoining additional rows. The set of all properties of a Chu space therefore itself forms a power set, and for normal spaces can be given explicitly as $2^{2^A - X}$. In particular *true* has just one property since $X = 2^A$, while *false* has the set 2^{2^A} of properties, namely all Boolean operations in variables from A.

An *axiomatization* of $\mathcal{A}$ is any set of properties whose conjunction is equivalent to φ_A. While the theory of $\mathcal{A}$ axiomatizes $\mathcal{A}$, we can always do better, starting by omitting the property *true* which holds of all Chu spaces. If only axiom count matters then the single axiom φ_A will always suffice. However φ_A is typically constituted as a conjunction of properties of one or another particular kind, various instances of which give rise to various familiar subcategories of **Chu**.

Given two operations φ_A and φ_B, a function $f : A \rightarrow B$ defines a *renaming* of the variables of $\mathcal{A}$ to those of $\mathcal{B}$. Renaming is the special case of substitution in which the substituted expressions are merely variables. We write $f(\varphi_A)$ for the result of so renaming the variables of φ_A. For example when $f(a) = d$ and $f(b) = f(c) = e$, $f(a \vee (b \wedge c))$ is $d \vee (e \wedge e)$ which is $d \vee e$ (logical equivalence is identity here).

We say that f *preserves properties* when every property of $\mathcal{A}$ (or equivalently just the property φ_A) renames under f to a property of $\mathcal{B}$. Thus if φ_A is $a \vee b$ and φ_B is $c \oplus d$ (exclusive-or) then $f(a) = d, f(b) = c$ preserves properties (since $f(a \vee b) = f(a) \vee f(b) = d \vee c$ is a consequence of $c \oplus d$) but $f(a) = f(b) = c$ does not (since $c \vee c = c$ is not a consequence of $c \oplus d$).

Theorem 1 *Given normal spaces $\mathcal{A} = (X, A)$, $\mathcal{B} = (Y, B)$, a pair of functions $f : A \rightarrow B$, $g : Y \rightarrow X$ is a Chu transform $(f, g) : \mathcal{A} \rightarrow \mathcal{B}$ if and only if f preserves properties.*

Proof: (If) Given $f : A \rightarrow B$, by extensionality of $\mathcal{A}$ there is at most one possible $g : Y \rightarrow X$ making (f, g) is a Chu transform. We have $\varphi_B \rightarrow f(\varphi_A)$, whence every satisfying assignment $row_B(y)$ of φ_B is a satisfying assignment of $f(\varphi_A)$. Hence the assignment $\mathrm{row}_B(y) \circ f$ to variables of A must be a satisfying assignment of φ_A. Hence there must exist $x \in X$ such that $\mathrm{row}_A(x) = \mathrm{row}_B(y) \circ f$; we may therefore satisfy the adjointness condition by setting $g(y) = x$. Doing this for all $y \in Y$ determines $g : Y \rightarrow X$ such that (f, g) is a Chu transform.

(Only if) Let (f,g) be a Chu transform. Each satisfying assignment of φ_B must be $\mathrm{row}_B(y)$ for some $y \in Y$. Now $\mathrm{row}_A(g(y))$ is a satisfying assignment of φ_A. But by adjointness $\mathrm{row}_A(g(y)) = \mathrm{row}_B(y) \circ f$, making $\mathrm{row}_B(y)$ a satisfying assignment of $f(\varphi_A)$. Hence every assignment satisfying φ_B satisfies $f(\varphi_A)$, whence $\varphi_B \to f(\varphi_A)$, whence f preserves properties. ∎

An equivalent condition is that f preserve axioms, since the set of properties preserved by a renaming is closed under arbitrary Boolean operations including arbitrary conjunction.

This view of extensional Boolean Chu spaces as the collection of its properties yields an easy demonstration that extensional Chu spaces can *realize* a great variety of ordered structures: sets, preordered sets, partially ordered sets, Stone spaces, topological spaces, locales, semilattices, complete semilattices, distributive lattices (but not general lattices), algebraic lattices, frames, profinite (Stone) distributive lattices, Boolean algebras, and complete atomic Boolean algebras, to name just the more prominent full, faithful, and concrete subcategories of **Chu**.

Normally these structures stick to their own kind in that each forms its own category, with morphisms staying inside individual categories. Chu spaces bring all these objects into the one self-dual category **Chu**, permitting meaningful morphisms between say semilattices and algebraic lattices, while revealing various Stone dualities such as that between Boolean algebras and Stone spaces, frames and locales, sets and complete atomic Boolean algebras, etc. to be all fragments of the same self-duality.

The notion of realization we intend here is the strong one defined by Pultr and Trnková [PT80]. Informally, one structure *represents* another when they transform in the same way, and *realizes* it when in addition they have the same carrier. Formally, a functor $F : C \to D$ is a *representation* of the objects of C by objects of D when F is a full embedding[2] (a full embedding). A representation F is a *realization* when in addition $U_D(F(A)) = U_C(A)$, where $U_C : C \to \mathbf{Set}$, $U_D : D \to \mathbf{Set}$ are the respective underlying-set functors. Pultr and Trnková give hardly any realizations in their book, concentrating on mere representations. In contrast all the representations of this paper will be realizations.

The category of Boolean operations and their property-preserving renamings is not self-dual since non-T_0 Chu spaces transpose to nonextensional ones. By the same reasoning the full subcategory consisting of T_0 operations, those with no properties $a \equiv b$ for distinct variables a, b, is self-dual. This is a very important fact: it means that to every full subcategory C of this self-dual category we may associate its dual as the image of C under the self-duality. This associates sets to complete atomic Boolean algebras, Boolean algebras to Stone spaces, distributive lattices to Stone-Priestley posets, semilattices to algebraic lattices, complete semilattices to themselves, and so on for many other familiar [Joh82] and not so familiar (self-duality of finite-dimensional vector spaces over $GF(2)$) instances of Stone duality

We now illustrate the general idea with some examples.

Sets. A set is a Chu space axiomatizable with no axioms; equivalently, an extensional T_0 Chu space whose rows form a complete atomic Boolean algebra or CABA, that is, are closed under complement and arbitrary union. That is, sets are dual to CABA's, argued later. The normal Chu space representing the set A is $(2^A, A)$. Every function $f : A \to B$ between sets $(2^A, A)$ and $(2^B, B)$ is a Chu transform because $(2^A, A)$ has all possible rows whence we can always find g making (f, g)

[2] An embedding is a faithful functor $F : C_A \to C_B$, i.e. for distinct morphisms $f \neq g$ of C_A, $F(f) \neq F(g)$, and is *full* when for all pairs a, b of objects of C_A and all morphisms $g : F(a) \to F(b)$ of C_B, there exists $f : a \to b$ in C_A such that $g = F(f)$.

a Chu transform. A better way to see this however is to use the fact that the Chu transforms from $\mathcal{A}$ to $\mathcal{B}$ are those functions $f : A \to B$ that preserve the axioms of $\mathcal{A}$, which must be all functions when $\mathcal{A}$ has the empty set of axioms.

Pointed Sets. A pointed set is a Chu space with the one axiom $a = 0$ (or any other constant from K), this element being the "point." Equivalently it is the result of adjoining a constant column to the Chu realization of a set. (Thus a constant is quite literally a constant column.) For $K = \mathbf{2}$ bipointed sets are also possible, axiomatized as $a = 0, b = 1$; in general up to K points are possible. Chu transforms between pointed sets preserve the point: $f(0) = 0$.

Preorders. A preorder is a Chu space axiomatized by "atomic implications," namely propositions of the form $a \to b$ where a and b are variables. A partial order is a T_0 preorder.

Theorem 2 *A Chu space realizes a preorder if and only if it is extensional and its rows form a complete lattice under arbitrary (including empty and infinite) union and intersection.*

Proof: (Only if) Fix a set Γ of atomic implications defining the given preorder. Suppose that the intersection of some set Z of assignments each satisfying all implications of Γ fails to satisfy some $a \to b$ in Γ. Then it must assign 1 to a and 0 to b. But in that case every assignment in Z must assign 1 to a, whence every such assignment must also assign 1 to b, so the intersection cannot have assigned 0 to b after all. Dually, if the union of Z assigns 1 to a and 0 to b, it must assign 0 to b in every assignment of Z and hence can assign 1 to a in no assignment of Z, whence the union cannot have assigned 1 to a after all. So the satisfying assignments of any set of atomic implications is closed under arbitrary union and disjunction.

(If) Assume the rows of $\mathcal{A}$ under union and intersection form a complete lattice.[3] It suffices to show that the set Γ of atomic implications holding in $\mathcal{A}$ axiomatizes $\mathcal{A}$, i.e. that $\mathcal{A}$ contains all satisfying assignments of Γ. Let $x \subseteq A$ be any such assignment. For each $a \in A$ form the intersection of all rows of $\mathcal{A}$ containing a, itself a row of $\mathcal{A}$ containing a, call it y_a. Now form the union $\bigcup_{a \in x} y_a$ to yield a row z of $\mathcal{A}$, which must be a superset of row x. Now suppose $b \in y - x$. Then there exists $a \in x$ such that $b \in y_a$, whence b is in every row of $\mathcal{A}$ containing a, whence $a \to b$ is in Γ. But x contains a and not b, contradicting the assumption that x satisfies Γ. Hence $b \in y - x$ cannot exist, i.e. $y = x$. However y was constructed from rows of $\mathcal{A}$ by arbitrary union and intersection and therefore is itself a row of $\mathcal{A}$, whence so is x. ∎

This is the essence of the argument showing that posets are dual to profinite distributive lattices [Joh82, p.249], with rows playing the role of ultrafilters or maps to $\perp$, cf. the isomorphism $A \multimap \perp \cong A^{\perp}$ in section 3.3. A normal T_0 Chu space whose columns are closed under arbitrary union and intersection realizes a profinite distributive lattice, which for our purposes suffices for a definition of this notion; consult Johnstone (op.cit.) for an alternative definition.[4] That this is a categorical duality follows immediately from the self-duality of **Chu**.

This result makes it easy to demonstrate the duality of sets and CABA's we promised earlier. One direction is clear: sets contain all possible rows and hence form a CABA by set theory. For the other direction, a CABA is a profinite distributive lattice, whence the theory of its dual is

[3] It is worth mentioning that this is a stronger assumption than that the rows of $\mathcal{A}$ partially ordered by inclusion form a complete lattice, since the meets and joins thereof then need not coincide with intersection and union.

[4] Here is a more conventionally abstract but simple and novel definition for lattices. A distributive lattice is *profinite* when it is complete and its maximal chains are *nowhere dense*, that is, every proper interval (pair $a < b$ and all elements between) includes a *gap*, meaning a proper interval containing only its two endpoints.

axiomatizable by atomic implications. When $a \leq b$ in a poset for distinct a, b there must exist a satisfying assignment making $a = 0$ and $b = 1$; the complementary assignment, which exists in a CABA, then contradicts $a \leq b$, showing that the theory of the dual of a CABA cannot contain any atomic implications $a \to b$ for distinct a, b, and hence is axiomatizable with the empty set of axioms.

To complete the argument that this is a realization we need the Chu transforms between posets realized in this way as Chu spaces to be exactly the monotone functions. Now monotonicity is the condition that if $a \leq b$ holds in (is a property of) the source then $f(a) \leq f(b)$ holds in the target. Since the only axioms are atomic implications, monotonicity is equivalent to being axiom-preserving, equivalently property-preserving, hence a Chu transform, and we are done.

The following fact about posets will prove useful when we come to locales.

Theorem 3 *A Chu space realizing a poset is column-maximal with respect to the requirement that its rows be closed under arbitrary union and intersection.*

Proof: Adjoin a new element c, and let Γ be the set of atomic implications that are properties of the result. Form the intersection y of those rows containing c, itself a row containing c (even if no rows contain c, in which case we get a new row $A \cup \{c\}$ and we are done). This must be the least assignment satisfying Γ such that c holds. Dropping c from that row then yields a new row that still satisfies Γ, so by the previous theorem the row set was not closed under intersection. ∎

Topological spaces. A topological space is an extensional Chu space whose rows are closed under arbitrary union and *finite* (including empty) intersection. The Chu transforms between topological spaces are exactly the continuous functions. This is most easily seen using the form $\mathrm{row}_A(g(y)) = \mathrm{row}_B(y) \circ f$ of the adjointness condition, with composition (of functions $B \to \mathbf{2}$ as the open sets of $\mathcal{B}$) with f (yielding functions $A \to \mathbf{2}$) being exactly the inverse image function $f^{-1} : 2^B \to 2^A$. Lafont and Streicher [LS91] mention in passing this realization along with that of Girard's coherent spaces [Gir87], also in $\mathbf{Chu}_2$, and the realization of vector spaces over the field K in $\mathbf{Chu}_{U(K)}$.

Locales. A *spatial locale* [Isb72, Joh82] is a *column-maximal* T_0 topological space, one to which no point can be added without defeating the requisite closure properties of the rows. A *locale* is the same less the requirement of extensionality that is imposed automatically for topological spaces. That is, a locale is a column-maximal T_0 Chu space whose rows are closed under arbitrary union and finite intersection. In this case we do not attempt to understand the opens of a locale as sets of points: on the contrary, we prefer to understand points of locales as sets of opens, locales being T_0.

A *frame* (op.cit.) is the dual of a locale; unlike the dual of a topological space in general, a frame is always *algebraic* in the sense that the Chu transforms from it are *all* functions preserving its arbitrary joins and finite meets. In contrast the dual of any infinite poset (a kind of topological space) must have all infinite meets, which Chu transforms from it must preserve. Hence no infinite poset (and *a fortiori* no infinite set) can be a locale.

The only difference between a T_0 topological space and a poset is the possibility that the rows (open sets) may not be closed under certain *infinite* intersections. This difference creates a loop-hole whereby T_0 topological spaces unlike posets (see preceding theorem) need not be column-maximal. For finite topological spaces this distinction disappears, so any example of a topological space not a locale *must* be infinite. As remarked above, any infinite poset provides an example of a nonlocale.

A popular example is the chain of natural numbers standardly ordered. Its rows are the order filters or up-sets of $\mathbf{N}$, which are closed under arbitrary intersection; in particular all infinite intersections yield the empty row. This example is not column-maximal: we can add a new point which appears in all rows except the empty one without violating closure under either arbitrary union or *finite* intersection (though the result is no longer closed under intersection of any infinite set of rows that omits the empty row, these intersections being the singleton containing the new point).

Semilattices. The semilattice $(A, \vee)$ is axiomatized by all equivalences $a \vee b \equiv c$ holding in $(A, \vee)$, one for each pair a, b in A. For f to preserve these axioms is to have $f(a) \vee f(b) \equiv f(c)$ hold in the target. But this is just the condition for f to be a semilattice homomorphism, giving us a realization in **Chu** of the category of semilattices.

Equivalently a semilattice is a T_0 extensional Chu space whose columns are closed under binary union and which is row-maximal subject to the other conditions. Row-maximality merely ensures that all rows satisfying the axioms are put in.

The dual of a semilattice is an algebraic lattice [Joh82, p.252].

Complete semilattices. The complete semilattice $(A, \bigvee)$ has all joins, including the empty join and infinite joins. It is axiomatized as for a semilattice, but the left hand sides of its equivalences may be either infinite joins or 0. The dual of a complete semilattice is itself a complete semilattice; thus the subcategory of **Chu** consisting of these complete semilattices is a self-dual subcategory (the same duality).

Distributive Lattices. The idea for the semilattice $(A, \vee)$ is extended to the lattice $(A, \vee, \wedge)$ by adding to the semilattice equations for $\vee$ all equations $a \wedge b = c$ holding in $(A, \vee, \wedge)$ for each a, b in A. Distributivity being a Boolean tautology, it follows that all lattices so represented are distributive. The second (equivalent) formulation of semilattices also extends to distributive lattices along the same lines.

Boolean algebras. A Boolean algebra is a complemented distributive lattice, hence as a Chu space it suffices to add the requirement that the set of rows be closed under complement. Equivalently a Boolean algebra is a T_0 extensional Chu space whose columns form a Boolean algebra under pointwise Boolean combinations (complement and binary union suffice) and which is row-maximal subject to these conditions.

The dual of a Boolean algebra can be obtained as always by transposition. What we get however need not have its set of rows closed under arbitrary union, in which case this dual will not be a topological space. But M. Stone's theorem [Sto36] is that the dual of a Boolean algebra is a totally disconnected compact Hausdorff space. We therefore have to explain how the dual of a Boolean algebra may be taken to be either a topological space or an object which does not obviously behave like a topological space.

There is a straightforward explanation, which at the same time yields a slick proof of Stone's theorem stated as a categorical duality.

The transpose of a Boolean algebra may be made a topological space by closing its rows under arbitrary union. The remarkable fact is that when this adjustment is made to a pair of transposed Boolean algebras, *the set of Chu transforms between them does not change.* (Actually their g components may change, but since these spaces are extensional g is determined by f, which is therefore all that we care about; the f's do not change.)

We prove this fact by first closing the source, then the target, and observing that neither

adjustment changes the set of Chu transforms.

Closing the rows of the source under arbitrary union can only add to the possible Chu transforms, since this makes it easier to find a counterpart for a target row in the source. Let $f : A \to B$ be a function that was not a Chu transform but became one after closing the source. Now the target is still a transposed Boolean algebra so its rows are closed under complement, whence so is the set of their compositions with f. But no new source row has a complement in the new source, whence no new source row can be responsible for making f a Chu transform, so f must have been a Chu transform before closing the source.

Closing the rows of the target under arbitary union can only delete Chu transforms, since we now have new target rows to find counterparts for. But since the new target rows are arbitrary unions of old ones, and all Boolean combinations of rows commute with composition with f (a simple way of seeing that f^{-1} is a CABA homomorphism), the necessary source rows will also be arbitrary unions of old ones, which exist because we previously so closed the source rows. Hence Chu transforms between transposed Boolean algebras are the same thing as Chu transforms, and hence continuous functions, between the topological spaces they generate.

This accounts for the fact that the Chu dual of a Boolean algebra is not a topological space. To complete this to a proof of Stone's theorem it suffices to show that the generated topological spaces are totally disconnected, compact, and Hausdorff, omitted here.

An interesting aspect of this proof of Stone's theorem is that usually a duality is defined as a contravariant *equivalence*. Here, all categorical equivalences appearing in the argument that are not actual isomorphisms are covariant. The one contravariant equivalence derives from the self-duality of **Chu**, which is an *isomorphism* of **Chu** with $\mathbf{Chu}^{\mathrm{op}}$. Those equivalences on either side of this duality that fail to be isomorphisms do so on account of variations in the choice of carrier and cocarrier. We pass through the duality with the aid of two independent sets A and X. But when defining Boolean algebras and Stone spaces, in each case we take X to consist of subsets of A, and it is on account of those conflicting representational details that we must settle for less than isomorphism on at least one side of the duality.

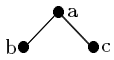
Vector spaces over $GF(2)$. An unexpected entry in this long list of full concrete subcategories of **Chu** is that of vector spaces over $GF(2)$. These are T_0 extensional Chu spaces containing the constantly zero column, with columns closed under binary exclusive-or, and row-maximal subject to these conditions. In the finite case row-maximality is not needed and we obtain symmetric matrices of size a power of two in each dimension. Hence the finite (same as finite-dimensional since the field is finite) vector spaces are self-dual as usual for a finite-dimensional vector space over any field. This follows as the case $k = GF(2)$ of Lafont and Streicher's observation that the category of vector spaces over any field k is realizable in $\mathbf{Chu}_{U(k)}$. There is one finite field of cardinality each power of each prime.

1.3 Chu spaces as partial distributive lattices

In this section we present Chu spaces as two-toned Hasse diagrams or *partial distributive lattices*, abbreviated *pdlat*. This leads to a natural view of a Chu space as a schedule defining a process. The corresponding diagram for its dual gives the corresponding unfolded automaton for the same process.

The columns of Figure 1(b) when ordered pointwise in the obvious way form a partial order

which we may depict with the following *Hasse diagram*.



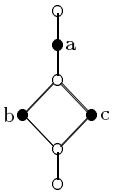
From Fig. 1(b)

But Figure 1(d), which is not isomorphic to Figure 1(b), yields the same partial order. We seek a method of diagramming Chu spaces that distinguishes them up to isomorphism.

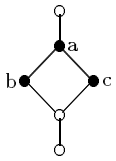
When two columns are equal (contain the same bit pattern) we can at best preorder them. Hasse diagrams are intended for partial orders, but can be adapted to preorders by depicting each maximal clique as a single element labeled with the cardinality of the clique. We therefore address the T_0 case (contrast this with the previous subsection, which assumed extensionality).

Our approach is to close the columns under arbitrary join (disjunction or pointwise OR) and meet (conjunction or pointwise AND), and to depict the resulting poset of columns as a two-toned Hasse diagram whose black elements denote the original columns and whose white ones are those formed by taking the closure. Diagrams we can actually draw will be finite; the "arbitrary" is so that the method will work even for infinite Hasse diagrams, had we only the patience and space to draw them.

When we close up the columns of Figure 1(d) in this fashion, we obtain three new columns: the meet 0001 of b and c, the empty join 0000, and the empty meet 1111. Similarly closing Figure 1(b) yields these columns but in the form 00001, 00000, and 11111, together with a fourth new column, the join of b and c, namely 00111. The corresponding Hasse diagrams are then as follows.



From Fig. 1(b)



From Fig. 1(d)

We can read off from each of these pdlats their common partial ordering of a, b, and c. But we can also read off that the join of b and c is a in 1(d) but not in 1(b). We say that the join of b and c is undefined or does not exist in (b). Their meet exists in neither. We refer to such a structure as a *partial distributive lattice*.

To see that the Chu space can be recovered from the pdlat we identify three sets: the set A of black points, the set 2^X of order filters of X ordered by inclusion (which will be the points of the pdlat without regard for color), and the set X. Ignoring the color scheme for the moment, we start from 2^X as the elements of the given pdlat, a profinite distributive lattice (cf. Theorem 2). We take X to consist of the *complete primes* of 2^X, namely those elements not the join of the set of elements strictly below them. Each element X' of 2^X is then represented as the set of complete primes $Y \subseteq X'$; this representation distinguishes all elements because the lattice is a profinite distributive lattice. The set A of black elements of this lattice then define the columns of the desired Chu space $(X, \leq, A)$ where $\leq$ is the lattice order on 2^X.

1.4 Examples

Up to isomorphism, there are $78 = 2+4+8+64$ T_0 extensional Chu spaces of respective cardinalities 0,1,2,3, which we display in Figure 2 below. (The powers of two are pure coincidence; the next two numbers are 3828 and 37320288.) We show each in all three presentation forms: pdlat, equation, and matrix in that order. There are actually two bit patterns shown, the one on the left being the Chu matrix and the one on the right being a representation of the dual of the pdlat we will explain shortly.

The top row shows the 0, 1, and 2 element spaces in three groups, (a)-(b), (c)-(f), (g)-(h). We show all 6 of those having at most one element, but take advantage of certain symmetries to economize the display of the 72 2- and 3-element spaces.

Note that the four one-element pdlats (c)-(f) differ only in whether their top and/or bottom is extended or retracted. These four cases obtain for all nonempty pdlats, so for the pdlats with more than one element we only bother to display those with both bounds retracted. We express the retraction of 1 equationally as $\bigvee A = 1$, and dually $\bigwedge A = 0$ retracts 0. The corresponding matrices are obtained by deleting the columns of all 0's and all 1's respectively. One final economy: 12 of the 16 3-element top-and-bottom-retracted pdlats are not isomorphic to their order duals, so we have omitted half of them leaving just (j)-(l) and (o)-(q) (the starred ones), along with the four that are isomorphic to their order duals, namely (i), (m), (n), and (r).

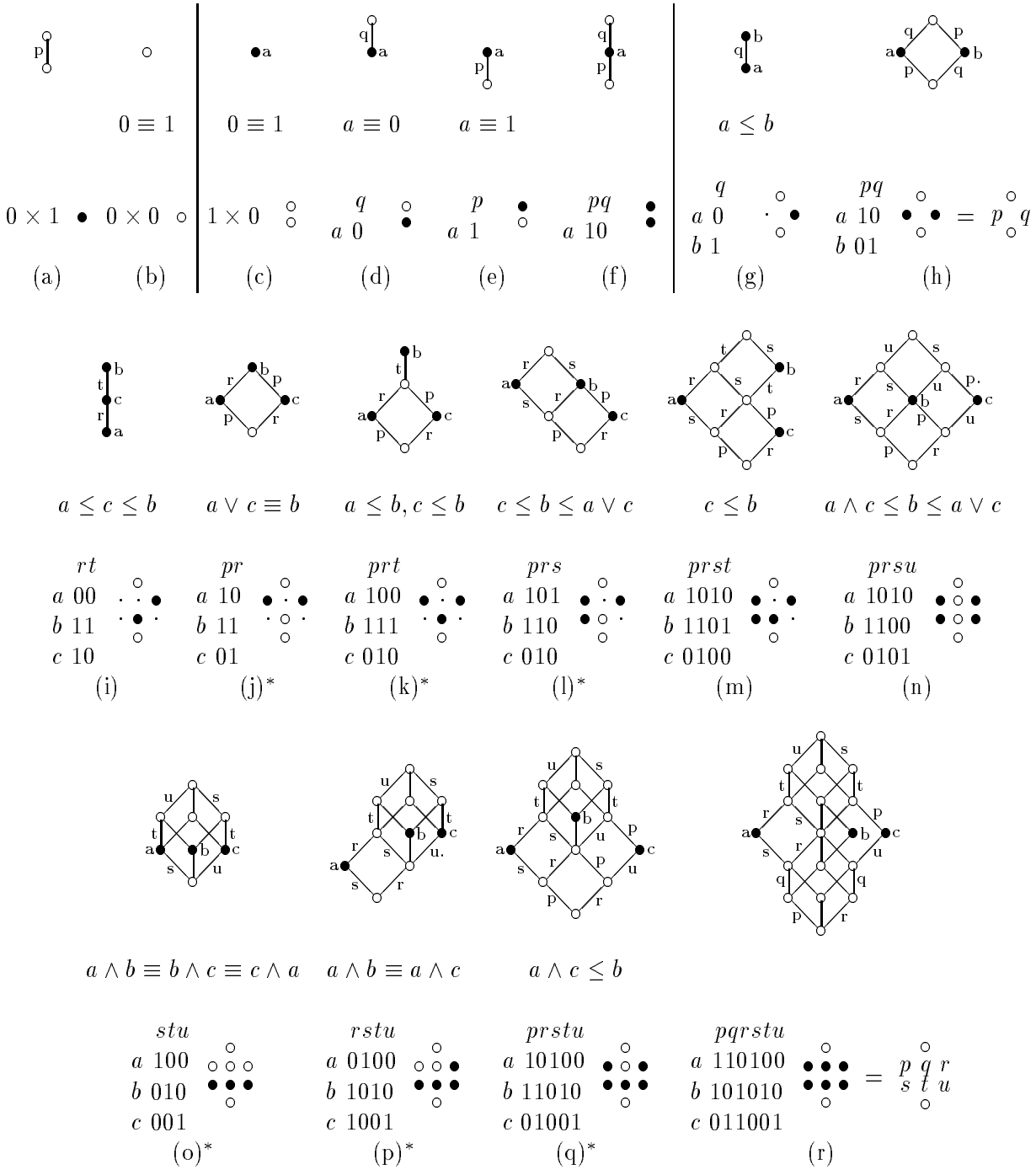
The discrete pdlats have for their lattices the free distributive lattice $\mathcal{F}_i$ on $i = 0$ to 3 generators[5] (The top and bottom of $\mathcal{F}_2$ and $\mathcal{F}_3$ need to be extended to make them the free *bounded* lattice, whence the quotation marks around "discrete" in those two cases.) All other pdlats are obtainable as quotients, viewable as retracts (quotients), of the free ones, with the retractions being specified by the equations. The free lattice on n generators can be seen to have 2^n "dimensions" along which retractions are permitted. Each retraction is expressible as a series of projections each projecting out one dimension. For example (r) retracts to (q) along dimension q, which then disappears as a distinguishable dimension. In any such retraction, if an edge of a square contracts then so does the opposite edge, thereby identifying the square's other two edges.

The diagram to the right of the matrix for A is the pdlat for $A^\perp$, an n-cube in standard position (cf. pdlat (o)) with its edges deleted to reduce clutter (note that (i)-(k) need only 6 of the eight vertices). The solid points denote the real elements of the dual, the circles its lattice-imaginary elements, and the periods its remaining Boolean-imaginary elements.

The real (black) elements of $A^\perp$ correspond to the dimensions of A and to the columns of the matrix representation, e.g. (r) has six dimensions, its matrix six columns, and its dual the six reals p-u. Retracting along a dimension of A corresponds to deleting the corresponding matrix column and the corresponding real of $A^\perp$; the imaginary elements are then whatever can be generated from the remaining reals. (The duals of smaller pdlats are represented similarly: for (g)-(h) the nonbounds are p and q, while for (c)-(f) the bounds themselves are named p and q.)

This is best understood by starting with (r) and regarding the remaining three-element pdlats as quotients of (r), and their corresponding duals as subpdlats of the dual of (r), whose six elements p-u can be tracked as they disappear, e.g. (m) is obtained by retracting along dimension u, the dual of deleting element u. Note that the correspondence between matrix columns and their labels is maintained during this process, the column labelled u for example always being 001.

[5] Sloane [Slo73] gives the size of the free distributive lattices on n generators as sequence 309, "Motonone Boolean Functions," namely 2, 3, 6, 20, 168, 7581, 7828354, 2414682040998, ...



$a \leq c \leq b$ $a \vee c \equiv b$ $a \leq b, c \leq b$ $c \leq b \leq a \vee c$ $c \leq b$ $a \wedge c \leq b \leq a \vee c$

$a \wedge b \equiv b \wedge c \equiv c \wedge a$ $a \wedge b \equiv a \wedge c$ $a \wedge c \leq b$

Figure 2. The T_0 extensional Chu spaces with up to 3 points.

2 Behavior: from event structures to rational mechanics

The classical 1970's conception of an automaton was as a device for accepting a formal language defined as a set of strings, possibly infinite in the case of so-called ω-automata. This conception made two automata equivalent when they accepted the same language. As models of behavior, each string of the accepted language was considered as one of the alternative or *possible* behaviors or *runs* of that automaton, and the symbols in that string all occurred during that run, in the order of occurrence in that string. In this context strings are usually referred to as *traces*.

We shall understand automata formally as follows. A *transition system* (X, Σ, δ) consists of a set X of *states* $x, y, z, \ldots$, a set Σ (or *Act*) of *actions* $a, b, c, \ldots$ (the alphabet), and a set $\delta \subseteq X \times \Sigma \times X$ of *transitions* (x, a, y), the *transition relation*. An *automaton* $(X, \Sigma, \delta, x_0, F)$ is a transition system with a distinguished state x_0, the *initial* state, and a set F of *final* states.

Instead of treating automata and languages as separate notions, it will clarify the sequel if we regard both strings and languages as just special kinds of automata. The string *aba* is taken to be the four-state straight-line automaton $0 \stackrel{a}{\to} 1 \stackrel{b}{\to} 2 \stackrel{a}{\to} 3$ having state set $X = \{0, 1, 2, 3\}$, initial state 0, and set $\{3\}$ of final states. We allow infinite strings, whose state set may then be taken to be the set of natural numbers, with initial state 0 and with no final state. The language L is taken to be the automaton whose state set is formed as the disjoint union of the state sets of the strings of L viewed as automata, with their initial states identified to form the single initial state of L, and with the set of final states of L being those of the individual strings.

Formally, a *language* L is an automaton $(X, \Sigma, \delta, x_0, F)$ with the following properties. (i) Every state in X has zero or one transition leading to it according respectively to whether it is or is not the initial state x_0. (ii) Every state is reachable from the initial state by a finite path of transitions. (iii) Every noninitial state x has zero or one transition leaving it according respectively to whether x is or is not a final state ($x \in F$ or $x \notin F$).

It follows that a language is a rooted tree, with only the root (the initial state) allowed more than one descendant. Further the initial state may or may not belong to the set F of final states, corresponding respectively to whether the empty string is or is not a member of L.

Motivated by such concerns as fair scheduling of concurrent infinite behaviors, a variety of more general acceptance criteria all involving repeated visits to final states have been considered, giving rise to the automata of Büchi, Rabin, and Streett. In these kinds of automata acceptance of an infinite string can only be determined given the whole string. Reflecting the scope to date of our work on applying Chu spaces to the modeling of behavior, the emphasis of this paper will be on finitely observable properties of automata and we shall therefore not consider these more general acceptance criteria here.

The new automata theory raised two objections to this conception, which in due course came to be called respectively branching time and true concurrency.

2.1 Branching Time.

The first objection was raised by Robin Milner in his book on CCS, a Calculus of Communicating Systems [Mil80]. The standard model appears to condense all choices about behavior into a single selection of a string from a language made at the start of the behavior. Real behavior however makes informed decisions on the fly as information comes to hand. Milner proposed a logic that

took deferred branching into account by abandoning the equation $a(b+c) = ab+ac$, and introducing a model, synchronization trees, to serve as counterexamples for this equation. This equation was taken as characteristic of *linear time*, its antonym being *branching time.*

A *synchronization tree* is an automaton satisfying conditions (i) and (ii) for languages. The omission of condition (iii) extends to all states the branching privilege that languages enjoy only at the initial state. (Less significantly it also removes the requirement that a noninitial *leaf* state, one with no transition leaving it, be a final state; one may impose this latter requirement as a weakened form of (iii), but we shall omit this as a complication irrelevant to our purposes.) Whereas the evident synchronization tree $ab + ac$ is a language having five states, the four-state synchronization tree $a(b + c)$ is not a language because the noninitial state to which the a transition leads has two transitions leaving it.

We regard the distinction that synchronization trees draw between $ab+ac$ and $a(b+c)$ as one of *timing.* Both trees entail the decision whether to perform action b or action c. But whereas $ab+ac$ makes this decision at the initial state, before a has been performed, $a(b+c)$ makes it at the state following the a transition. We understand the difference to be the additional information obtained by performing a: $a(b+c)$ makes a *more informed* decision than $ab+ac$ thanks to the information brought to light by a.

Thus a synchronization tree is intermediate in abstractness between automata and languages. While synchronization trees can draw distinctions based on timing of decisions, they cannot draw other distinctions available to general automata related to looping structure and confluence. For example a synchronization tree cannot distinguish a^* (short loop) from $(aa)^*(\epsilon + a)$ (length-two loop), nor $(a + b)c$ (confluence or rejoining) from $ac + bc$ (nonconfluence).

Since cyclic structure entails confluence (a loop must return to either the initial state as a degenerate case of confluence, or to a state reachable by some other path from the initial state), we may regard a synchronization tree as a confluence-free automaton.

Every automaton can be approximated by a synchronization tree in a canonical way, namely the tree whose states are the paths leading from the root and whose transitions are the singleton path extensions (p, a, q) where q is a path extending path p with an a transition. The initial state is the empty path while the final states are those paths terminating in a final state of the given automaton.

In turn, every synchronization tree can be approximated by a language in a canonical way, namely the language whose strings are the root-to-final-state paths and all[6] infinite paths in that tree. The passage from tree to language may be understood as the "teasing apart" of the paths of the tree, moving all branching points up to the root.

2.2 True concurrency

The second objection to the language interpretation of automata, raised sporadically by various people over a long period [Pet62, Gre75, Maz77, Gra81, NPW81, Pra82], was that the standard model assigned a well-defined order to every pair of events (symbol occurrences) in the same string. Besides contradicting relativity, this assumption also contradicts practical engineering issues at all scales, from "data skew" on parallel signal lines within a single chip to detecting when a husband and wife are simultaneously making withdrawals from the same account at remote automatic teller

[6] Or the accepted infinite paths when treating automata endowed with infinitary acceptance criteria.

machines. At the 1988 Königswinter conference on true concurrency, Robin Milner with tongue in cheek referred to global ordered time as "false concurrency;" a more informative term would be *atomic mutual exclusion* which postulates that atomic events are mutually exclusive in the sense of not being permitted to overlap in time.

Our notion of automaton appears to enforce this linear ordering of the events that actually occur during a run of an automaton. A straightforward way to relax this requirement is to generalize the notion of string to that of partial string, called a *partial word* by Grabowski [Gra81] and a *partially ordered multiset* by Pratt [Pra82], later shortened to *pomset* [Pra84, Gis88].

A *pomset* $(A, \leq, \Sigma, \lambda)$ is a set A of *events*[7] partially ordered by a binary relation $\leq$, the *temporal precedence relation*, and a *labeling function* $\lambda : A \to \Sigma$ assigning to each event $a \in A$ its *label* $\lambda(a) \in \Sigma$ denoting the action performed by a. A string is then taken to be the special case where the pomset is linearly ordered.

Event structures [NPW81] generalize pomsets to incorporate a notion of alternative behavior expressed as conflict information. An *unlabeled event structure* $(A, \leq, \#)$ consists of a set A of events partially ordered by $\leq$, together with a symmetric irreflexive binary relation $\#$, denoting conflict, such that if $a\#b$ and $b \leq c$, then $a\#c$. When $a\#b$ holds, this indicates that the events a and b cannot both occur in the same run. A *labeled event structure* $(A, \leq, \#, \Sigma, \lambda)$ adds an action alphabet Σ and a labeling function λ as for pomsets.

Just as a synchronization tree may be understood as a confluence-free automaton, a pomset may be understood as a conflict-free event structure.

2.3 Relating Automata and Event Structures

Automata and labeled event structures may represent each other in the following canonical ways.

Event structures cannot represent any of the information lost in the passage from an automaton to its canonical approximation by a synchronization tree. Hence we translate automata to event structures in two stages via synchronization trees. We already have the first stage, we now give the second.

The synchronization tree $(X, \Sigma, \delta, x_0, F)$ is represented by the labeled event structure $(\delta, \leq, \#, \Sigma, \lambda)$ such that (i) $(x, \alpha, y) \leq (x', \beta, y')$ just when these two transitions occur in that order on some path of the tree, (ii) $(x, \alpha, y)\#(x', \beta, y')$ just when these two transitions do not both appear on the same path in the tree, and (iii) $\lambda(x, \alpha, y) = \alpha$.

In the other direction, the labeled event structure $(A, \leq, \#, \Sigma, \lambda)$ is represented by the *automaton* $(X, \Sigma, \delta, x_0, F)$ where (i) $X \subseteq 2^A$ consists of the conflict-free order ideals of the event structure, namely those subsets $x \subseteq A$ such that $a \in x$ and $b \leq a$ implies $b \in x$, and such that $a\#b$ implies not both $a \in x$ and $b \in x$; (ii) $(x, \alpha, y) \in \delta$ just when $y - x = \{a\}$ where $\lambda(a) = \alpha$; (iii) $x_0 = \{\}$; and (iv) F consists of the maximal elements of X.

For example the discretely ordered conflict-free event structure

$$(\{a, b\}, \{(a, a), (b, b)\}, \{\}, \{a, b\}, \lambda x.x)$$

is approximated by the automaton whose state set X is the power set of $\{a, b\}$; whose transition relation consists of the four transitions $(\{\}, a, \{a\})$, $(\{\}, b, \{b\})$, $(\{a\}, b, \{a, b\})$, and $(\{b\}, a, \{a, b\})$;

[7] We shall henceforth use $A = \{a, b, c, \ldots\}$ to denote events or *action occurrences*, and use $\alpha, \beta, \ldots$ instead of $a, b, \ldots$ for the actions themselves, which we shall understand as labels on events.

whose initial state is empty; and whose final state set is $\{\{a,b\}\}$. Modifying this event structure by ordering it as $a \leq b$ has the effect of deleting the state $\{b\}$ and its two incident transitions. If instead the two events are put in conflict, $a\#b$, this has the effect of deleting the state $\{a,b\}$ from X, and its two incident transitions from δ, and the final state set then becomes $\{\{a\},\{b\}\}$, these now being the two maximal states in X. Making both modifications deletes both those states; the result is equivalent, with respect to this translation, to the event structure having just the one event a.

This translation then yields another, namely from event structure to synchronization tree: just compose the above translation from event structure to automaton with the canonical approximation of an automaton by a synchronization tree treated earlier.

Now when the synchronization tree of the above example is translated to an event structure we find that all pairs a,b of events are related by either $\leq$ or $\#$. On the one hand this event structure is different from the one we began with; on the other, it then translates back to the same synchronization tree. Thus only the first of these translations has lost any information, namely concerning whether or not a and b are mutually exclusive in the sense of not being permitted to happen concurrently.

But note that the critical true-concurrency information, namely the independence of a and b, was lost in the passage from the automaton to the synchronization tree. Since the automaton is acyclic, this passage only "unfolds" confluences, there are no loops to unwind. Those confluences that do arise may be associated with the absence of both conflict and order between the confluent events. In this way the translation to the automaton loses less information; in particular the independence of two events may be recovered from their appearance in the automaton as a confluent pair. Nevertheless some information is lost, as witnessed by the event structure $a \leq b$, $a\#b$ which yields the same automaton as the event structure with just the one event a. The Chu account of automata cleanly resolves this and related issues.

Now event structures and synchronization trees can both exhibit branching structure. However there is an important difference. Whereas any one run of a synchronization tree takes only *one* path out of each branch, a run of a pomset performs *all* events, and a run of an event structure performs all the events of some maximal non-conflicting set of events.

We shall understand this difference by viewing the set X of states of an automaton as a *disjunctive* set, and the set A of events of an event structure as a *conjunctive* set. This distinction is implicit in the transformation of A and X by respectively functions and antifunctions. The following gives some additional insight into this distinction.

Both automata and event structures are graphs. However their edges have the following dual character. An edge (x,a,y) of an automaton *enables* behavior in that it permits an a transition from x to y. This gives automaton edges the character of the modal operator $\diamond$ expressing the possibility of a transition to a state. An edge of either form $a \leq b$ or $a\#b$ of an event structure *constrains* behavior in that it limits the possible states of the corresponding automaton. This associates event structure edges with the modal operator $\Box$ expressing a *necessary constraint.*

2.4 Behavioral Interpretation of Chu Spaces

We interpret the Chu space $(X,\models,A)$ as a process with state set X, event set A, and *occurrence relation* $x \models a$ indicating that in state x event a has already happened. The "has already happened"

in this interpretation is where time enters the Chu space picture of computation.

As an $X \times A$ Boolean matrix, a Chu space is a two-dimensional object. We take the vertical axis to be the information axis and the horizontal as the temporal axis. This corresponds to states being distributed in an information space and events in a temporal space. This is in complete agreement with the dimensions of the logics of *sequential* behavior we have collected under the rubric of "two-dimensional logic" [Pra94, p.156], where we said:

> We shall visualize the dimensions as oriented respectively vertically and horizontally. (This will be recognized as in agreement with 2-category usage, where 1-cell composition is horizontal and 2-cell vertical.) It is natural to associate information or static logical strength with the vertical axis and time or dynamic progress with the horizontal.

In this viewpoint a two-dimensional logic is a set of propositions-cum-actions ordered somewhat independently in these two dimensions, with the vertical ordering imparting the static character of propositions and the horizontal ordering (actually a monoid) conferring the dynamic character of actions. This is an unsorted framework; the program-proposition sort distinction drawn by dynamic logic is viewed from this perspective not as fundamental but merely as a minor syntactic distinction leading to decidability and finite axiomatizability of the propositional case [Pra90]. What Chu spaces do is put the propositions (*including* the programs) of the *language* of dynamic logic, action logic, etc. and the states of their *semantics* on an equal footing as forming the dual sets A and X of respectively events and states.

Recall that a property of a Chu space is a superset of its state set, equivalently, a Boolean consequence of the space viewed as a Boolean proposition. The following succinctly expressed properties correspond naturally to various aspects of concurrency. Any property expressed using states rather than events in the following (e.g. transitions) is to be understood as a property of the dual space.

Transition: A state x can evolve into a state y just when $x \to y$. These are just inclusions between states, all of which we regard as legitimate state transitions.

Temporal order: a occurs before b if $b \to a$. Winskel writes this as $a \vdash b$, called *prime enabling* [Win88]. If $b \to a$, then every state with $b = 1$ must also have $a = 1$, which means that a must have been set to 1 no later than b.

Enabling: General (nonprime) enabling has the form $a, b \vdash e$; $c, d \vdash e$, an enablement of e equivalent to the Boolean formula $e \to (a \wedge b) \vee (c \wedge d)$ (either one of (a with b) or (c with d) suffices to enable e), i.e. any number of pre-events before the $\vdash$, and any number of $\vdash$'s.

Conflict: $a \# b$ is $\neg(a \wedge b)$ (binary or coherent conflict[NPW81]), and means that it is illegal to set both a and b to 1, i.e. they are in conflict. More generally, $\#x$ may be defined as $(\bigwedge_{a \in x} a) \equiv 0$ (no state contains all events in x, i.e. no superset of x is a state of X.)

Internal choice: $x \equiv y \wedge z$ and $\#(y \vee z)$ expresses the choice of conflicting states y or z, made in the state x, so this choice is "internal". This corresponds to the branching construct of programming languages, where a choice is made based on the information accumulated in the current state. We can have a choice between several states, like a `case` statement in C.

Causality: $a \wedge b \equiv c$ asserts that a and b jointly cause c as their *immediate* effect, as it is impossible to have done both a and b without doing c also. On the other hand $c \to a \wedge b$ is mere prime enabling, $a, b \vdash c$, that is it is OK to wait for a while before doing c. This distinction is absent from all other models of concurrency we are aware of.

Nondeterminism: $a \equiv b \vee c$ and $b\#c$ asserts choice of conflicting events b or c. Since the choice is made at the same time as doing a, any information gathered by doing a could not have been used to choose, however, the choice was not available before a. So this choice is made by the environment, and is "external". This contrasts with $b \vee c \to a$, which is mere prime enabling $a \vdash b,\ a \vdash c$.

Synchronization: $a \equiv b$ asserts that a and b must happen simultaneously. A T_0 space has only identity synchronizations $a \equiv a$. Conditional synchronization, $a \to b \equiv c$ however may hold even for T_0 spaces: it holds for causality $(a \wedge b) \equiv c$.

3 Algebra: from linear logic to process algebra

In this section we first treat the linear logic of Chu spaces in its own right. We then give a process algebra interpretation of the connectives of linear logic. In this interpretation linear logic is by no means a complete process algebra; we accordingly round out the language to a more comprehensive process algebra by providing several additional connectives. Whereas the linear logic connectives are naturally defined categorically as functors on **Chu** with suitable universal properties, these additional connectives are defined set theoretically, and some are not even functorial.

3.1 Linear logic

The language of linear logic, LL, closely parallels that of relation algebras, RA. The latter amounts to two copies of the logical connectives *or*, *false*, *and*, *true*, *not*, and *implies*, distinguished as the *logical* and *relative* (relational) forms of those connectives, due to Peirce but anticipated to some extent by by De Morgan [DM60]. To these Schröder [Sch95] added reflexive transitive closure a_0, nowadays a^*, and its De Morgan dual a_1.

Combining the separate involutory logical and relative duals, a^- and $a^\smile$, as a single involutory ($a^{\perp\perp} = a$) dual $a^{\smile -} = a^\perp$ [Pra92a, p.252] weakens the Boolean structure of RA to that of a De Morgan lattice [Dun86, p.184,p.193], since neither $a + a^\perp = 1$ nor $aa^\perp = 0$ hold of binary relations. This seems in practice to leave the utility of RA largely unimpaired, whose operations are then as follows.

Relation Algebra:	*Logical* :	$a+b$	0	ab	1
	Relative :	$a \dagger b$	$0'$	$a;b$	$1'$
	Nonmonotone :	$a^\perp$	$a\backslash b$	b/a	$a \to b$
	Closure :		a^*	a_1	

Interpreted standardly for binary relations over a fixed set, the logical connectives are union, empty, intersection, and the complete relation. The relative connectives are $a \dagger b = (a^-;b^-)^- = (a^\perp;b^\perp)^\perp$ (the De Morgan dual of composition), $\neq$, composition, and $=$. The nonmonotone connectives are complement-of-converse $a^\perp = a^{-\smile}$, right residuation $a\backslash b = a^\perp \dagger b$, left residuation $b/a = b \dagger b^\perp$, and "static implication" $a \to b = a^- + b$. The closure operations are reflexive transitive closure and its De Morgan dual $a_1 = ((a^\perp)^*)^\perp$.

These operations are not independent, and a suitable basis is $a+b$, 0, $a;b$, $1'$, $a^\perp$, and a^*.

The language of linear logic can be closely matched to this as follows. We use Barr and Seely's

notation [Bar91, See89] in preference to Girard's more idiosyncratic notation [Gir87].

	Additives :	$A+B$	0	$A\times B$		1
Linear	*Multiplicatives* :	$A \oplus B$	$\bot$	$A \otimes B$		$\top$
Logic:	*Nonmonotone* :	$A^\bot$		$A \multimap B$		$A \Rightarrow B$
	Exponentials :		$!A$	$?A$		

Additives. Just as the Boolean or static connectives of RA can be inferred solely from the inclusion order among relations, ignoring the monoid structure $a;b$, so can the additives of linear logic be inferred solely from the categorical (morphism) structure on the category of Chu spaces, ignoring the monoidal structure $A \otimes B$. The additives are respectively coproduct, the initial object, product, and the final object.

The coproduct $\mathcal{A}+\mathcal{B}$, where $\mathcal{A} = (X, \models_A, A)$, $\mathcal{B} = (Y, \models_B, B)$, is defined as $(X \times Y, \models, A+B)$ where $X \times Y$ is the cartesian product of the state sets, $A+B$ is the disjoint union $A \times \{0\} \cup B \times \{1\}$ of the event sets, and $\models$ satisfies $(x,y) \models (a,0) \;=\; x \models a$, $(x,y) \models (b,1) \;=\; y \models b$. The associated inclusion $i_A : A \to A+B$ is defined as (i_A, p_X) where $i_A : A \to A+B$ is the inclusion in **Set** and $p_X : X \times Y \to X$ is the projection in **Set**; and analogously for $i_B : B \to A+B$. We verify that (i_A, p_X) is a Chu transform with the calculation $p_X(x,y) \models_A a = x \models_A a = (x,y) \models (a,0) = (x,y) \models i_A(a)$.

To see that $A+B$ with these two inclusions is coproduct, it suffices to exhibit a natural bijection between pairs of Chu transforms $A \xrightarrow{f} C \xleftarrow{g} B$ and those Chu transforms $A+B \xrightarrow{h} C$ such that $hi_A = f$ and $hi_B = g$. The trick is to establish this bijection separately for the covariant and contravariant components and then verify that each pair of maps making up some $A+B \xrightarrow{h} C$ is a Chu transform, and (immediate) that all Chu transforms satisfying $hi_A = f$ and $hi_B = g$ arise in this way.

The initial object 0 is $(1, !, 0)$, meaning the 1-state 0-event Chu space. Initiality is immediate.

We obtain the product $A \times B$ and the final object as the De Morgan duals (with respect to $A^\bot$) of coproduct and the initial object. That is, $A \times B$, in full $(X, \models_A, A) \times (Y, \models_B, B)$, is $(X+Y, \models, A \times B)$ where $\models$ satisfies $(x,0) \models (a,b) \;=\; x \models a$, $(y,1) \models (a,b) \;=\; y \models b$, with associated projections $p_A : A \times B \to A$ defined as (p_A, i_X) where $p_A : A \times B \to A$ is the projection in **Set** and $i_X : X \to X+Y$ is the inclusion in **Set**.

Nonmonotone Connectives. (We treat the monotone connectives first as being easier.) The *dual* $(X, \models, A)^\bot$ of a Chu space is simply its transpose $(A, \models^\smile, X)$. The *linear implication* $(X, \models_A, A) \multimap (Y, \models_B, B)$ is $(A \times Y, \models, A \to B)$ where $A \to B$ denotes the *set* of Chu transforms (f,g) from Chu space A to Chu space B and $\models$ satisfies $(f,g) \models (a,y) = g(y) \models_A a = y \models_B f(a)$ (the second equation is just the adjointness condition, but points up the symmetry that the first equation by itself might otherwise conceal).

The *intuitionistic implication* $(X, \models_A, A) \Rightarrow (Y, \models_B, B)$ is $(A \times Y, \models, A \to B)$ where this time $A \to B$ denotes the set of *functions* $f : A \to B$ from event set A to event set B, and $\models$ satisfies $f \models (a,y) = y \models_B f(a)$.

Multiplicatives. We may define $A \oplus B = A^\bot \multimap B$, $\bot = (\{0\}, \in, \{\{\}, \{0\}\})$, $A \otimes B = (A^\bot \oplus B^\bot)^\bot = (A \multimap B^\bot)^\bot$, and $\top = \bot^\bot = (\{\{\}, \{0\}\}, \in^\smile, \{0\})$. Thus $\bot$ is a 1×2 matrix while $\top$ as its transpose is 2×1; in both cases the two entries are respectively 0 and 1, corresponding respectively to 0 being a nonmember or member of the given set.

Exponentials. We take $!A \;=\; A \Rightarrow \bot$ and $?A = (!A^\bot)^\bot$. We could with isomorphic effect have

defined $!(X, \models, A)$ as the normal Chu space $(2^A, A)$ (realizing the set A) and defined $A \Rightarrow B$ in terms of $!A$, namely via $A \Rightarrow B = !A \otimes B$.

Except when defining these operations, we shall often write their operands as A and B instead of $\mathcal{A}$ and $\mathcal{B}$, provided no confusion is likely.

3.2 Definition by Circuits

The Boolean operation view of Chu spaces leads to a natural way to define certain operations on Chu spaces, namely with Boolean circuits that combine those operations.

The units for sum and tensor product, namely 0 and $\top$, and the exponential $!\mathcal{A}$, are all defined by circuits whose output is constantly 1 and hence whose inputs are ignored. Zero has no inputs, the tensor unit $\top$ has one input, and $!\mathcal{A}$ has for its inputs those of $\mathcal{A}$. (This definition of $!\mathcal{A}$ satisfies the Girard axioms but also satisfies $!!\mathcal{A} \cong !\mathcal{A}$, leaving open the possibility of a less constrained alternative Chu interpretation for $!\mathcal{A}$.)

The sum $\mathcal{A}+\mathcal{B}$ is implemented as a circuit with components $\mathcal{A}$ and $\mathcal{B}$, by forming the conjunction of the outputs of $\mathcal{A}$ and $\mathcal{B}$, with the set of inputs of $\mathcal{A}+\mathcal{B}$ then being the disjoint union $A+B$ of those of $\mathcal{A}$ and $\mathcal{B}$. This construction is illustrated in Figure 3, for which $|A| = 3$ and $|B| = 2$.

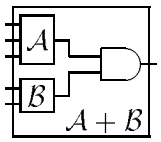


Figure 3. Circuit for Sum

The space thus implemented can be seen to partition its inputs into two disjoint blocks, one judged by $\mathcal{A}$, the other by $\mathcal{B}$, with the space as a whole registering its approval just when all its components approve of their respective blocks.

As it turns out, the categorical product $A \times B = (A^\perp + B^\perp)^\perp$, i.e. the dual of coproduct, is circuit-definable, as follows.

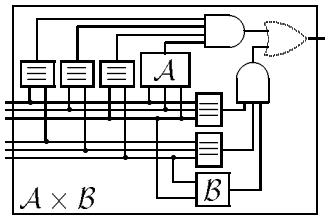


Figure 4. The product of two gates.

Exercise: decipher this.

Tensor product $\mathcal{A} \otimes \mathcal{B}$ is the only operation requiring some work, and is where the circuit approach to defining operations really helps. As one might guess from the fact that the tensor unit has output 1, tensor product is a form of conjunction. But whereas sum is a noninteracting conjunction, tensor product behaves like a generic logical inference, in which the constraints in the components can entail new constraints involving the variables of both components.

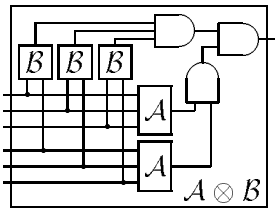


Figure 5. Circuit for Tensor Product

Now as with sum, tensor product assumes no *a priori* relationship between the inputs of its arguments, which are therefore made disjoint. Nevertheless a connection is established between the two sets of inputs, namely by taking the set of inputs of $\mathcal{A} \otimes \mathcal{B}$ to be the cartesian product $A \times B$ of those of $\mathcal{A}$ and $\mathcal{B}$ instead of their sum. Visualizing this product as a rectangular array of inputs, we define *bilinearity* to be the condition that each column of this rectangle (what one obtains by fixing a particular $b \in B$ of $\mathcal{B}$'s inputs) independently satisfies $\mathcal{A}$ while each row satisfies $\mathcal{B}$. This is realized by using $|B|$ distinct copies of $\mathcal{A}$ to monitor each column of $A \times B$, and likewise $|A|$ copies of $\mathcal{B}$ to monitor rows, as illustrated in Figure 5 for the case where the rectangle is 3×2.

3.3 Equational Logic

We have the *equation* $a+b = b+a$ for relation algebras, and it is natural to expect this for linear logic as well. However on closer inspection we notice that each a in the carrier A of $\mathcal{A}$ becomes $(a,0)$ in $A+B$ but $(a,1)$ in $B+A$. We may however claim the *isomorphism* $A+B \cong B+A$, in which $(a,0)$ in $A+B$ is matched up with $(a,1)$ in $B+A$. (A is isomorphic to B when there exist bijections $X_A \cong X_B$, $Y_A \cong Y_B$ of their index sets making their corresponding entries equal.) This applies to the other laws of linear logic as well, with the exception of $A^{\perp\perp} = A$ and definitions (e.g. $A \otimes B = (A \multimap B^{\perp})^{\perp}$. The full list of isomorphisms (and equalities where possible) we know to hold for extensional T_0 Chu spaces is as follows.

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
A+(B+C) & \cong & (A+B)+C & A+0 & \cong & A & A+B & \cong & B+A \\
A \otimes (B \otimes C) & \cong & (A \otimes B) \otimes C & A \otimes \top & \cong & A & A \otimes B & \cong & B \otimes A \\
A \otimes (B+C) & \cong & (A \otimes B)+(A \otimes C) & A \otimes 0 & \cong & 0 & A^{\perp\perp} & = & A
\end{array}$$

From these laws and the definitions of abbreviations we can derive for example $(A \otimes B) \multimap C = (C^{\perp} \otimes (A \otimes B))^{\perp} \cong ((C^{\perp} \otimes A) \otimes B)^{\perp} = B \multimap (C^{\perp} \otimes A)^{\perp} = B \multimap (A \multimap C)$. We also have $A \multimap B = (A \otimes B^{\perp})^{\perp} \cong (B^{\perp} \otimes A^{\perp\perp})^{\perp} = B^{\perp} \multimap A^{\perp}$, and $A \multimap \perp \cong (A \otimes \top)^{\perp} \cong A^{\perp}$. We leave $(A \times B) \Rightarrow C \cong A \Rightarrow (B \Rightarrow C)$ as an exercise. We are not aware of any completeness results for the isomorphism theory of Chu spaces.

Note that $A^{\dagger}$, $A^{\dagger\dagger}$, $A^{\dagger\dagger\dagger}$ ... is $Y_A, K^{Y_A}, K^{K^{Y_A}}, \ldots$, in contrast to $!!A = !A$.

3.4 Process algebra

The linear logic operations $A + B$ and $A \otimes B$ for Chu spaces realize the process algebra operations we have previously called in a series of papers respectively *concurrence* and *orthocurrence* [Pra85, Pra86, CCMP91].

Basic process algebra operations not provided by any linear logic connectives include *choice* $\mathcal{A} \sqcup \mathcal{B}$ and *sequence* $\mathcal{A}; \mathcal{B}$.

We define the *choice* $\mathcal{A}\sqcup\mathcal{B}$ of normal Chu spaces $\mathcal{A}=(X,A)$ and $\mathcal{B}=(Y,B)$, assumed to have disjoint event sets A,B, as $(X\cup Y,A\cup B)$. Disjointness of event sets ensures that the empty state will be the only state the arguments have in common, which this construction therefore identifies. $\mathcal{A}\sqcup\mathcal{B}$ chooses which of $\mathcal{A}$ or $\mathcal{B}$ it is going to do as soon as it performs its first event from one of the arguments, since no state of the other argument contains that event.

The circuit representation of this definition is as follows.

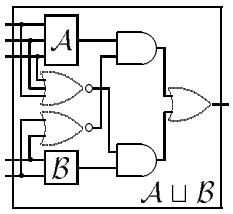


Figure 6. Choice of two Chu spaces.

We define the *sequence* $A;B$ by deleting from the states of $A+B$ those states x which contain an event of B and for which there exists a state $y\supseteq x$ in $A+B$ such that $y-x$ contains an event of A. A simple example of this is when A and B each have one event (respectively a and b) and two states. Then $A+B$ has two events a,b and as states the four subsets of $\{a,b\}$. $A;B$ deletes state $\{b\}$ from this because it contains an event of B and there exists a state $\{a,b\}$ such that $\{a,b\}-\{b\}$ contains an event of A.

We have omitted *iteration* $\mathcal{A}^*$, and more generally recursion, from this treatment.

4 Relational structures

We have seen that **Chu** is a sort of universal category for lattice theory. This section extends the universality of Chu spaces to arbitrary relational structures and their homomorphisms, by passing to $\mathbf{Chu}_K$ for larger K. This is the Chu space version of the passage from propositional logic to first-order logic.

We have already mentioned Lafont and Streicher's observation [LS91, p.45] that the category of vector spaces over a field $\mathbf{K}$ is a full subcategory of $\mathbf{Chu}_K$, and that the category **Top** of topological spaces is a full subcategory of $\mathbf{Chu}_2$. We improve on these observations by showing that *every* κ-ary relational structure is realizable as an object of $\mathbf{Chu}_{2^\kappa}$, giving a strong sense in which Chu spaces form a universal category.

The earliest instance of a universal category is due to Trnková [Trn66]. The universality of the category of semigroups was established by Hedrlín and Lambek [HL69]. These and a number of other such embeddings all took the form of a full and faithful functor that did not preserve underlying sets, for example representing some finite objects as infinite ones. The advantages accruing from the unifying framework of semigroups are then more than offset by the radically different discipline required to do mathematics in the absence of the expected underlying set.

Definition 4 For any ordinal κ, a κ-ary *relational structure* (X,ρ) consists of a set X, the *carrier*, and a κ-ary relation $\rho\subseteq X^\kappa$ on X. A *homomorphism* $f:(X,\rho)\to(Y,\sigma)$ between two such structures is a function $f:X\to Y$ between their underlying sets for which $f\rho\subseteq\sigma$. Here $f\rho$

denotes $\{f\mathbf{a} \mid \mathbf{a} \in \rho\}$, where $\mathbf{a}$ denotes $(a_0, \ldots, a_{\kappa-1})$ and $f\mathbf{a}$ denotes $(fa_0, \ldots, fa_{\kappa-1})$. We denote by $\mathbf{Str}_\kappa$ the category formed by the κ-ary relational structures and their homomorphisms. ∎

It suffices to treat structures with a single carrier and relation, since k carriers can be combined as their disjoint union, kept track of with k unary relations ($\lceil \log_2 k \rceil$ is enough information-theoretically, but not enough to ensure that homomorphisms respect type). Multiple nonempty relations on a set can be joined to form a single relation on the same set, of arity at most the sum of the arities of its constituent relations. For algebras, structures all of whose $(n+1)$-ary relations are n-ary operations, the join may share the input coordinates of the operations, reducing the total arity to the maximum of the input arities plus the number of operations (including constants).

This notion of homomorphism is standard in the strong sense that *any* class of n-ary relational structures and their homomorphisms constitutes a full subcategory of $\mathbf{Str}_\kappa$. Familiar examples of such categories and their arities include those of semigroups (3), monoids (4), groups (3), rings (4), rings with a multiplicative unit (5), fields (4), lattices (3), lattices with top and bottom (5), Boolean algebras (3), vector spaces (4),[8] directed graphs or binary relations (2), multigraphs (4), posets (2), and categories (4).

Many of these numbers benefit from group structure, for which homomorphisms preserve inverses and identities even when these operations are not given explicitly as part of the relation. Units of monoids, including tops and bottoms of lattices, are not so fortunate and each requires its own unary relation in order to be recognized and preserved by homomorphisms.

The universality achieved here is of a different kind from that achieved by say ZF set theory. Externally a model of ZF is a single object of $\mathbf{Str}_2$ of some cardinality, with membership as its only relation, "internally" coding objects larger than any fixed cardinal including its own. Our universality has no separate notion of an internal world; instead we code our objects purely externally.

We now define the promised functor $F : \mathbf{Str}_\kappa \to \mathbf{Chu}_{\mathbf{2}^\kappa}$, namely in definitions 6 and 10, and prove that it is full, faithful, *and concrete*.

The complementarity of constraints and states indicates ρ and $\overline{\rho}$ as the appropriate respective sources of each. We shall define a state to be essentially an element of $\overline{\rho}$, with however a small but essential refinement. The following lemma obtains from the standard constraint-based definition of homomorphism an equivalent state-based characterization.

Lemma 5 $f\rho \subseteq \sigma \;\Leftrightarrow\; f^{-1}\overline{\sigma} \subseteq \overline{\rho}$. Here $\overline{\rho} = A^\kappa - \rho$ and $\overline{\sigma} = B^\kappa - \sigma$.

Proof:

$$\begin{array}{rcll} f\rho \subseteq \sigma & \Leftrightarrow & \rho \subseteq f^{-1}\sigma & \text{(Definition of } f^{-1}\text{)} \\ & \Leftrightarrow & \overline{f^{-1}\sigma} \subseteq \overline{\rho} & \text{(Complement)} \\ & \Leftrightarrow & f^{-1}\overline{\sigma} \subseteq \overline{\rho} & (f^{-1} \text{ preserves Boolean operations}) \end{array}$$

∎

Definition 6 (F on objects). Let $\mathbf{2}^\kappa$ denote the set of κ-bit bit vectors, that is, κ-tuples over $\mathbf{2}$. We define the object part of the functor $F : \mathbf{Str}_\kappa \to \mathbf{Chu}_{\mathbf{2}^\kappa}$ as taking the κ-ary relational structure

[8] Treat as partial rings, with uv defined just when u is on a specified axis. This works equally well for homogeneous vector spaces (all over the one field) and heterogeneous, the only nontrivial field endomorphisms being automorphisms.

(A,ρ) to the Chu space $(X,\models,A)$ defined as follows. Take X to consist of those κ-tuples $x \in (2^A)^\kappa$ of subsets of A for which $\prod_{i<\kappa} x_i \subseteq \overline{\rho}$. Let $\models: X \times A \to 2^\kappa$ satisfy $\models (x,a)_i = 1$ if $a \in x_i$, and 0 otherwise. ∎

Noting the isomorphism of $(2^\kappa)^A$ with $2^{\kappa \times A}$, we may equivalently think of X as follows. A κ-tuple over A is a function from κ to A. Define a κ-ruple (for *relational* tuple) over A to be a binary relation from κ to A. This makes the κ-tuple t the κ-ruple $\{(i,a) \mid t_i = a\}$. Then X consists of those κ-ruples over A extending no κ-ruple of ρ.

It might seem that X could be represented more naturally and conveniently as just the power set of $\overline{\rho}$. But observe that a state x as defined here can be recovered from the set $\prod_i x_i$ of its κ-tuples just when no component x_i is empty. The definition of $f^{-1} : Y \to X$ in Definition 10 below requires each x_i to be available independently even when some are empty.

The crucial test of whether $(X,\models,A)$ faithfully represents (A,ρ) is whether ρ can be recovered from it. We show this constructively as follows.

Lemma 7 *For all* $\mathbf{a} \in A^\kappa$, $\mathbf{a} \in \rho \;\Leftrightarrow\; \forall x{\in}X\ \exists i{<}\kappa :\models (x,a_i)_i = 0$.

Proof:

$$\begin{array}{rcll} \mathbf{a} \in \rho & \Leftrightarrow & \forall x{\in}X : \mathbf{a} \notin \prod_i x_i & \text{(Construction of } X\text{)} \\ & \Leftrightarrow & \forall x{\in}X\ \exists i{<}\kappa : a_i \notin x_i & \text{(Definition of product)} \\ & \Leftrightarrow & \forall x{\in}X\ \exists i{<}\kappa :\models (x,a_i)_i = 0 & \text{(Construction of } \models\text{)} \end{array}$$

∎

Corollary 8 *F is injective on objects.*

Lemma 9 *$(X,\models,A)$ is extensional.*

Proof: If $\mathrm{row}(x) = \mathrm{row}(y)$ then $\forall i < \kappa[a \in x_i \Leftrightarrow a \in y_i]$, so $\forall i : x_i = y_i$, whence $x = y$. ∎

Definition 10 (F on maps). Let $f : (A,\rho) \to (B,\sigma)$ be a homomorphism, with $F(A,\rho) = (X,\models,A)$ and $F(B,\sigma) = (Y,\models',B)$ as per Definition 6. Define $f^{-1} : (2^\kappa)^B \to (2^\kappa)^A$ to take $g : B \to 2^\kappa$ to $gf : A \to 2^\kappa$. Now for all $y \in Y$, $\prod_i y_i \subseteq \overline{\sigma}$ by construction of Y. Hence $\prod_i f^{-1}y_i \subseteq \overline{\rho}$, by Lemma 5. Thus $f^{-1}y \in X$ by construction of X. We may therefore define $F(f)$ as (f,f^{-1}) where $f^{-1} : Y \to X$. ∎

Theorem 11 *The functor F of Definitions 6 and 10 is concrete, faithful, and full.*

Proof: F is concrete by construction, and *a fortiori* faithful.

For fullness consider any Chu transform $(f,g) : F(A,\rho) \to F(B,\sigma)$ where $F(A,\rho) = (A,X)$ and $F(B,\sigma) = (B,Y)$. If $\mathbf{a} \in \rho$, then for every $y \in Y$ there exists $i < \kappa$ such that

$$\begin{array}{rcll} & \models (gy,a_i)_i & = & 0 \quad \text{(Lemma 7 with } x = gy\text{)}, \\ \text{whence} & \models' (y,fa_i)_i & = & 0 \quad \text{(}(f,g)\text{ is a Chu transform)}. \end{array}$$

Hence by Lemma 7, $f\mathbf{a} \in \sigma$, establishing that f is a homomorphism. And since $(X,\models,A)$ is extensional, by Lemma 9, g is determined by f. Hence $F(f) = (f,g)$. ∎

Remarks. (i) Where size matters, X need contain only those states representable as the inverse image of a tuple of singletons. These can be characterized explicitly as those states x with the property that either $x_i = x_j$ or $x_i \cap x_j = \emptyset$ for all $i, j < \kappa$, observing that f^{-1} preserves this property. (ii) Lemma 9 is an inessential bonus. Had Definition 6 produced a nonextensional $(X, \models, A)$, we would simply have enforced extensionality, needed for fullness, by identifying those states having the same extension.

5 Heisenberg uncertainty in Chu spaces

We first discuss ways in which events and states interfere with each other from a lattice theoretic perspective, concentrating on $K = \mathbf{2}$. Passing from lattices to numbers, we treat uncertainty in Chu spaces as a purely quantitative phenomenon devoid of information-theoretic signicance; this holds for arbitrary K. We then relate the phenomenon to a natural model of information flow between Chu spaces.

5.1 Event-State Interference

Events as columns are made of the same bits as states as rows, so it stands to reason the two would interfere with other.

A simple case of interference is given by a pointed set, one with a constant column. The only possible constant row must be for the same constant. In particular for $K = \mathbf{2}$, if A has constant 0, $A^\perp$ cannot have constant 1.

A slightly more complex example is given by proper meets and joins. A meet or join is proper just when the result is not among the arguments.

Theorem 12 *A proper meet in $\mathcal{A}$ precludes some proper join in $\mathcal{A}^\perp$.*

Proof: Let $a \vee b$ be proper. Then there must exist rows x, y such that $x \models a$ and $y \models b$ but not $x \models b$ or $y \models a$. Hence $x \models (a \vee b)$ and $y \models (a \vee b)$. Now suppose $x \wedge y$ exists. Then neither $(x \wedge y) \models a$ or $(x \wedge y) \models b$ hold, and hence neither does $(x \wedge y) \models (a \vee b)$. But this contradicts $x \models (a \vee b)$ and $y \models (a \vee b)$. ∎

Corollary 13 *If $\mathcal{A}$ has all meets then $\mathcal{A}^\perp$ has no proper joins.*

Proof: For if $\mathcal{A}^\perp$ had a proper join it would preclude some proper meet of $\mathcal{A}$. ∎

Exercise: study this interference for infinite joins and meets of various cardinalities.

5.2 Quantitative aspects of uncertainty

Heisenberg's uncertainty principle is $\Delta p.\Delta q \geq \hbar$, the product of the uncertainties in momentum and position exceeds Planck's constant over 2π. This principle is sometimes explained in introductory lectures in terms of the Fourier transform. Here the natural units for a signal and its Fourier transform, namely seconds and Hertz for a signal distributed in time, force the constant $\hbar$ in this inequality to unity, a scaling that is also popular in high-energy particle physics. The inequality

appearing in the principle results from an application of the Schwartz inequality in its proof. Thus the intrinsic uncertainty in momentum of a particle cannot be inferred exactly from uncertainty in its position, which yields only a lower bound on momentum uncertainty.

For Chu spaces this inequality becomes an equality. Here however the reasoning is much simpler. We treat only finite Chu spaces. We define *absolute* uncertainty in state, or *information uncertainty*, to be the precision to which states can be identified, namely $1/|X|$. That is, a Chu space with only four states can specify its "exact" state only to within an uncertainty of $1/4$, i.e. two bits of precision. Likewise the absolute imprecision associated with knowledge of any event, or *temporal uncertainty*, is $1/|A|$. (There are no actual "exact states" distinct from the states in X, this is merely a convenient fiction for making sense of the concept that the elements of finite sets are specified only up to some precision.)

The bits in the matrix of a Chu space of a given size can be chosen independently. For a space with $n = |X| \times |A|$ bits one might guess that $\hbar$ should be 2^{-n}. However the product of information uncertainty $1/|X|$ with temporal uncertainty $1/|A|$ equals $1/n$. Our uncertainty principle must therefore take $\hbar$ to be $1/n$.

Now we customarily think of $\hbar$ as a universal constant, and it is disconcerting to have it depend on the size of the agents in an interaction. The following reasoning plausibly justifies the universal value $\hbar = 1$.

A Chu space having more events naturally has more states, with

$$\log_2(|X|) \leq |A| \leq 2^{|X|}$$

in the case of extensional T_0 spaces (no repeated rows or columns). "Balanced" Chu spaces have $|X| \approx |A|$, this being the situation when there are interesting and fruitful interactions between states and events. With this in mind we normalize to put all Chu spaces on the one comparable scale independent of their size, by defining the *relative* number of states to be $|X|/|A|$, and the relative number of events to be its reciprocal, $|A|/|X|$. This makes the *relative uncertainty* of a state $|A|/|X|$, and of an event $|X|/|A|$. The product of these uncertainties is 1, the uncertainty principle for Chu spaces.

From this point of view uncertainty resides only in the "form factor" of the Chu space and not its absolute size. Long low ones have more precise events, tall skinny ones more precise states.

5.3 Observation

We now justify the word "uncertainty" for these quantities in terms of the following model of information flow between Chu spaces. We briefly sketched this general idea in [Pra92b], but without the benefit of Chu spaces as a concrete model of it yielding the simple calculations of the previous section, a simplicity we had not expected to be possible in 1992. As then, the idea is to define a "message" between spaces A and B. In the present framework such a message becomes a Chu transform $f : A \to B$, as the basic notion of measurement or observation of A in the language of an abstract observer B. In the simplest nontrivial case, the observer is the two-element Boolean algebra $\perp$, which can record only a binary distinction, which it does for each point (event) of A, corresponding to a monochrome (black-and-white) image. This is the "picture" of A that B "gets."

This is a natural enough notion of message on its own, assuming we accept the idea of identifying the observer with the observation language. The following idea of "structure-induced veil,"

that smarter observers reveal less about themselves to any given observer, makes it an even more attractive notion by equipping the abstract calculations of the preceding section with an intuitively plausible meaning in terms of the information content of observations. The supporting principle for this connection is the identity $|A \multimap \bot| = |X_A|$, that is, the number of messages Chu space A can send to the canonical observer $\bot$ equals the number of states of A.

If A is a set (i.e. no structure), say of pixels on a computer screen, the messages it can send to $\bot$ are all possible black-and-white images. If however A has some structure, e.g. a linear ordering imposed on those pixels, the variety of possible messages can drop sharply; we then think of this additional structure as creating a sort of veil that defocuses the screen, making it less distinct. This gives a primitive model of the intuitively plausible idea that while one can see straight through an idiot, deeper thinkers are harder to understand.

As explained in the introduction, we have treated Chu spaces as blank canvases devoid of paint, analogous to the role of vector spaces in computer graphics as blank spaces ready to receive images. That is, Chu spaces as objects give the underlying geometry of behavior. A Chu transform as an observation amounts to a painting of the observed space with "colors" drawn from the points (events) of the observing space viewed as a palette. Using linear transformations to paint vector spaces does not work as well: vector spaces do not make good palettes, whereas Chu spaces provide a wide range of quality of palettes, from much worse than vector spaces (e.g. sets) to much better (e.g. Boolean algebras). Vector spaces fall exactly in the middle of this spectrum; indeed the n-dimensional vector space over $GF(2)$ is representable as a square (whence "in the middle") Chu space [LS91] with 2^n points and 2^n states (the dual points or functionals in the usual sense of vector spaces), with Chu transforms between vector spaces so represented being exactly their linear transformations.

The canonical choice of nontrivial palette B is the two-point one-state Boolean algebra $\bot = (\{0\}, \models, \{0,1\})$ for which $x \models a$ (where x is necessarily 0) is taken to be a itself. This has both a constant 0 and a constant 1, and hence permits *some* painting of every consistent or empty Chu space A (the inconsistent one-point space $(\emptyset, !, \{0\})$ is the only exception: it insists on painting its one point both 0 and 1, only possible with itself as the palette). Dually the canonical choice of canvas A is the one-element set, which permits some painting by every nonempty palette B (the empty palette is the dual of the inconsistent one-point space). (The choice of cardinals here is only to make these choices canonical, and does not affect the other claims.)

It follows from the isomorphism $A \multimap \bot \cong A^\bot$ (section 3.3) that the states of any Chu space are in 1-1 correspondence with its Chu transforms to $\bot$. This is analogous to the corresponding situation for topological spaces, where the open sets of a space are in 1-1 correspondence with its continuous functions to the two-point Sierpinski space defined as having exactly one open set that is a singleton (along with the empty set and whole space), which performs for topological spaces the function performed by $\bot$ for Chu spaces. In general, for any class requiring a given constant row, e.g. the empty set (constantly 0 row) and whole space (constantly 1 row) in the case of sets, posets, and topological spaces, the role of $\bot$ as a dualizing element is not impaired *for that class* when those constant rows are added to it; this is the Chu account of schizophrenia of dualizing objects. Exercise: further generalize what modifications to $\bot$ are possible for the purpose of dualizing, and use the generalization to explain why the two-element set is a dualizer for **Set**.

We can thus read off the opacity of $(X, \models, A)$ directly from X alone, at least when $\bot$ is the observer. This accounts for the relevance of $|X|$ to observation of A: it gives the number of observations of A that the canonical observer $\bot$ can distinguish between.

The relevance of $|A|$ is that this gives the number of observations of the conjugate of A, $A^\perp$, possible by $\perp$. Observing both A and $A\perp$ is analogous to observing the conjugate properties of position and momentum of a particle. Thus the abstract uncertainty principle derived in the preceding section is made real by identifying $|X|$ and $|A|$ with the precision with which canonical observer $\perp$ can observe respectively A and its conjugate $A^\perp$.

For other observers we do not as yet have an interesting story to tell. One problem that can arise is that some observers may be handicapped by an inadequate observation alphabet. We analyze general observers via the identity $A \multimap B \cong (A \otimes B^\perp)^\perp$. Here the number of messages observer B can receive from A is the number of states of the tensor product $A \otimes B^\perp$. For $B = \perp$ we have $\perp^\perp = \top$, the tensor unit in the sense that $A \otimes \top \cong A$, confirming the special case proved above.

But while there is a simple rule for the number of *points* in the tensor product $A \otimes B^\perp$, namely the product of the numbers of points in the arguments, there is no simple general rule for the number of states. One extreme is that when A is a set, i.e. $A = (2^A, A, \in^\smile)$, then $A \multimap B$ has $|B|^{|A|}$ elements. This expresses the fact that any Chu space can be used as the observation language, independently of its structure, when observing anything as naive as a set. (The same (large) extreme is reached by the dual situation in which $B = (Y, \models, B)$ is a Boolean algebra, where $A \multimap B \cong B^\perp \multimap A^\perp$, whence $A \multimap B$ has X^Y elements since $B^\perp = (B, \models^\smile, Y)$ is a set.) A simple example of the other extreme is given by A having a constant 0 (or 1) and B not, in which case $A \multimap B$ is empty since Chu transforms must preserve constants. This is the situation where no measurement is possible because one of the variables (points) being measured always has a value that is missing from the observation alphabet (the empty alphabet is a degenerate case of this).

The general picture of communication between arbitrary Chu spaces then is that the amount of communicable information is highly dependent on the ability of the observer to understand the observed object.

6 Notes and Acknowledgements

Historical notes. Chu spaces are the case $V = \mathbf{Set}$ of the construction described by Po-Hsiang Chu in the appendix of Barr's book on *-autonomous (i.e. self-dual closed) categories [Bar79]. Chu's construction takes a closed monoidal category V with pullbacks and completes it to a self-dual category $\mathbf{Chu}(V, k)$. De Paiva [dP89a, dP89b] and Brown and Gurr [BG90, BGdP91] apply the Chu construction to respectively a version of Gödel's Dialectica and Petri nets. Lafont and Streicher study Chu spaces over K, which they call games [LS91]; the term "Chu construction" had been around previously, and the name "Chu space" for $V = \mathbf{Set}$ was suggested to us by Barr in email. Since Barr gave the basic construction to Chu in the first place it would be fair to be call them Chu-Barr spaces.

Acknowledgements. I must acknowledge first and foremost two years of productive collaboration with my student Vineet Gupta on the development of Chu spaces. Second, nearly a megabyte of email correspondence with Mike Barr on this and related topics has proved immensely insightful. More recently I have been collaborating with Gordon Plotkin, and with more time would have included some of his excellent insights in this paper. A two-month visit by Carolyn Brown, also working on Chu spaces, was very helpful. I also acknowledge useful conversations on the subject with Peter Freyd, Valeria De Paiva, Andreas Blass, Samson Abramsky, Rick Blute, and Robert Seely.

References

[Bar79] M. Barr. $*$-*Autonomous categories, LNM 752*. Springer-Verlag, 1979.

[Bar91] M. Barr. $*$-Autonomous categories and linear logic. *Math Structures in Comp. Sci.*, 1(2), 1991.

[BG90] C. Brown and D. Gurr. A categorical linear framework for Petri nets. In J. Mitchell, editor, *Logic in Computer Science*, pages 208–218. IEEE Computer Society, June 1990.

[BGdP91] C. Brown, D. Gurr, and V. de Paiva. A linear specification language for Petri nets. Technical Report DAIMI PB-363, Computer Science Department, Aarhus University, October 1991.

[CCMP91] R.T Casley, R.F. Crew, J. Meseguer, and V.R. Pratt. Temporal structures. *Math. Structures in Comp. Sci.*, 1(2):179–213, July 1991.

[DM60] A. De Morgan. On the syllogism, no. IV, and on the logic of relations. *Trans. Cambridge Phil. Soc.*, 10:331–358, 1860.

[dP89a] V. de Paiva. The dialectica categories. In *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 47–62, held June 1987, Boulder, Colorado, 1989.

[dP89b] V. de Paiva. A dialectica-like model of linear logic. In *Proc. Conf. on Category Theory and Computer Science, LNCS 389*, pages 341–356, Manchester, September 1989. Springer-Verlag.

[Dun86] J.M. Dunn. Relevant logic and entailment. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume III, pages 117–224. Reidel, Dordrecht, 1986.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gis88] J.L. Gischer. The equational theory of pomsets. *Theoretical Computer Science*, 61:199–224, 1988.

[Gra81] J. Grabowski. On partial languages. *Fundamenta Informaticae*, IV.2:427–498, 1981.

[Gre75] I. Greif. *Semantics of Communicating Parallel Processes*. PhD thesis, Project MAC report TR-154, MIT, 1975.

[HL69] Z. Hedrlín and J. Lambek. How comprehensive is the category of semigroups. *J. Algebra*, 11:195–212, 1969.

[Isb72] J.R. Isbell. Atomless parts of spaces. *Math. Scand.*, 31:5–32, 1972.

[Joh82] P.T. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.

[LS91] Y. Lafont and T. Streicher. Games semantics for linear logic. In *Proc. 6th Annual IEEE Symp. on Logic in Computer Science*, pages 43–49, Amsterdam, July 1991.

[Maz77] A. Mazurkiewicz. Concurrent program schemas and their interpretation. In *Proc. Aarhus Workshop on Verification of Parallel Programs*, 1977.

[Mil80] R. Milner. *A Calculus of Communicating Systems, LNCS 92.* Springer-Verlag, 1980.

[NPW81] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures, and domains, part I. *Theoretical Computer Science*, 13, 1981.

[Pet62] C.A. Petri. Fundamentals of a theory of asynchronous information flow. In *Proc. IFIP Congress 62*, pages 386–390, Munich, 1962. North-Holland, Amsterdam.

[Pra82] V.R. Pratt. On the composition of processes. In *Proceedings of the Ninth Annual ACM Symposium on Principles of Programming Languages*, January 1982.

[Pra84] V.R. Pratt. The pomset model of parallel processes: Unifying the temporal and the spatial. In *Proc. CMU/SERC Workshop on Analysis of Concurrency, LNCS 197*, pages 180–196, Pittsburgh, 1984. Springer-Verlag.

[Pra85] V.R. Pratt. Some constructions for order-theoretic models of concurrency. In *Proc. Conf. on Logics of Programs, LNCS 193*, pages 269–283, Brooklyn, 1985. Springer-Verlag.

[Pra86] V.R. Pratt. Modeling concurrency with partial orders. *Int. J. of Parallel Programming*, 15(1):33–71, February 1986.

[Pra90] V.R. Pratt. Dynamic algebras as a well-behaved fragment of relation algebras. In *Algebraic Logic and Universal Algebra in Computer Science, LNCS 425*, pages 77–110, Ames, Iowa, June 1988, 1990. Springer-Verlag.

[Pra92a] V.R. Pratt. Origins of the calculus of binary relations. In *Proc. 7th Annual IEEE Symp. on Logic in Computer Science*, pages 248–254, Santa Cruz, CA, June 1992.

[Pra92b] V.R. Pratt. Quantum logic, linear logic, and constructivity. In *Computation of Physics workshop: collected abstracts*, Dallas, October 1992. Superseded by published IEEE proceedings version, retitled "Linear Logic for Generalized Quantum Mechanics".

[Pra94] V.R. Pratt. A roadmap of some two-dimensional logics. In J. Van Eijck and A. Visser, editors, *Logic and Information Flow (Amsterdam 1992)*, pages 149–162, Cambridge, MA, 1994. MIT Press.

[PT80] A. Pultr and V. Trnková. *Combinatorial, Algebraic and Topological Representations of Groups, Semigroups, and Categories.* North-Holland, 1980.

[Sch95] E. Schröder. *Vorlesungen über die Algebra der Logik (Exakte Logik). Dritter Band: Algebra und Logik der Relative.* B.G. Teubner, Leipzig, 1895.

[See89] R.A.G Seely. Linear logic, $*$-autonomous categories and cofree algebras. In *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 371–382, held June 1987, Boulder, Colorado, 1989.

[Slo73] N.J.A. Sloane. *A Handbook of Integer Sequences.* Academic Press, 1973.

[Sto36] M. Stone. The theory of representations for Boolean algebras. *Trans. Amer. Math. Soc.*, 40:37–111, 1936.

[Trn66] V. Trnková. Universal categories. *Comment. Math. Univ.Carolinae*, 7:143–206, 1966.

[Vic89] S. Vickers. *Topology via Logic*. Cambridge University Press, 1989.

[Win88] G. Winskel. An introduction to event structures. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, REX'88, LNCS 354*, Noordwijkerhout, June 1988. Springer-Verlag.

Verifying a Border Array in Linear Time

František Franěk Shudi Gao Weilin Lu P. J. Ryan
W. F. Smyth* Yu Sun Lu Yang

Algorithms Research Group
Department of Computing & Software
McMaster University
Hamilton, Ontario
Canada L8S 4L7

May 12, 2000

Abstract

A *border* of a string $\boldsymbol{x}$ is a proper (but possibly empty) prefix of $\boldsymbol{x}$ that is also a suffix of $\boldsymbol{x}$. The *border array* $\boldsymbol{\beta} = \boldsymbol{\beta}[1..n]$ of a string $\boldsymbol{x} = \boldsymbol{x}[1..n]$ is an array of nonnegative integers in which each element $\boldsymbol{\beta}[i]$, $1 \le i \le n$, is the length of the longest border of $\boldsymbol{x}[1..i]$. In this paper we first present a simple linear-time algorithm to determine whether or not a given array $\boldsymbol{y} = \boldsymbol{y}[1..n]$ of integers is a border array of some string on an alphabet of unbounded size and then a slightly more complex linear-time algorithm for an alphabet of any given (bounded) size α. We then consider the problem of generating all possible distinct border arrays of given length n on a bounded or unbounded alphabet, and doing so in time proportional to the number of arrays generated. A previously published algorithm that claims to solve this problem in constant time per array generated is shown to be incorrect, and new algorithms are proposed. We conclude with an equally efficient on-line algorithm for this problem.

1 Introduction

The classical method for computing the border array $\boldsymbol{\beta} = \boldsymbol{\beta}[1..n]$ of a given string $\boldsymbol{x} = \boldsymbol{x}[1..n]$ is the so-called "failure function" algorithm [AHU74], that executes in $O(n)$ time. A recent paper [MSM99] introduces the idea of *b-equivalent* — that is, strings with the same border array — and shows how to construct, on a standard alphabet, *b-canonical* strings that are the unique representatives of each b-equivalent class. The paper then describes an algorithm to generate all possible border arrays of length n together with their corresponding b-canonical strings in time proportional to the number of arrays generated. If b_n denotes the

*communicating author (smyth@mcmaster.ca); also at School of Computing, Curtin University, Perth WA 6845, Australia.

number of distinct border arrays of length n that can exist when the alphabet is unbounded, then the sequence

$$\begin{aligned} B &= \{b_1, b_2, \dots\} \\ &= \{1, 2, 4, 9, 20, 47, 110, 263, 630, 1525, \dots\} \end{aligned}$$

is shown to be a new integer sequence [SP95].

In this paper we extend the results of [MSM99] in two ways:

(1) We describe an $O(n)$-time algorithm that determines whether or not a given array $\boldsymbol{y}[1..n]$ of integers is a border array of some string (on a bounded or unbounded alphabet)

(2) We show that the [MSM99] algorithm to generate all possible border arrays is actually incorrect, in the sense that it requires more than constant time per string generated. We then describe a time- and space-optimal algorithm that generates all border arrays of length at most n (on a bounded or unbounded alphabet) without the need to store the underlying b-canonical strings. These arrays constitute a new infinite class of integer sequences.

Unlike the algorithm described in [MSM99], this algorithm generates a trie in a depth-first fashion and so is not *on-line* — that is, the set of border arrays for $n+1$ cannot be efficiently derived from the set of border arrays for n. We conclude by describing another algorithm that is on-line and achieves constant time per string generated.

2 Identifying Valid Border Arrays For Unbounded Alphabets

This paper deals with arrays $\boldsymbol{y} = \boldsymbol{y}[1..n]$ of nonnegative integers. For these arrays it will be convenient to make use of the notation $\boldsymbol{y}^1[i] = \boldsymbol{y}[i]$ for every $i \in 1..n$, while

$$\boldsymbol{y}^j[i] = \boldsymbol{y}\big[\boldsymbol{y}^{j-1}[i]\big]$$

for every $j > 1$ such that $\boldsymbol{y}^{j-1}[i] \in 1..n$. It follows from the definition of border that for a border array $\boldsymbol{\beta}$, $0 \le \boldsymbol{\beta}[i] < i$ for every i, so that the sequence $i, \boldsymbol{\beta}[i], \boldsymbol{\beta}^2[i], \dots$ is monotone decreasing to zero, hence finite. We state a well-known result [AHU74]:

Lemma 2.1 *For some integer $n \ge 1$, let $\boldsymbol{x} = \boldsymbol{x}[1..n]$ denote a string with border array $\boldsymbol{\beta}$. Let k be the integer such that $\boldsymbol{\beta}^k[n] = 0$. Then*

(a) for every integer $j \in 1..k$, $\boldsymbol{x}\big[1..\boldsymbol{\beta}^j[n]\big]$ is a border of $\boldsymbol{x}[1..n]$;

(b) for any choice of letter λ, every border of $\boldsymbol{x}[1..n{+}1] = \boldsymbol{x}[1..n]\lambda$ has a length that is an element of the following set:

$$\begin{aligned} S^n &= \{S_0^n, S_1^n, \dots, S_k^n\} \\ &= \big\{0, \boldsymbol{\beta}[n]{+}1, \boldsymbol{\beta}^2[n]{+}1, \dots, \boldsymbol{\beta}^k[n]{+}1\big\}. \quad \square \end{aligned}$$

The set S^n defined in Lemma 2.1(b) is called the *admissible set* of the border array $\boldsymbol{\beta}[1..n]$, and each of its elements S^n_j, $j = 0, 1, \ldots, k$ is called an *admissible extension* of $\boldsymbol{\beta}$. Thus the lemma tells us that the only possible border arrays $\boldsymbol{\beta}[1..n+1] = \boldsymbol{\beta}[1..n]m$ are those for which m is an admissible extension. To see that the converse is not true — that is, that not all admissible extensions give rise to border arrays — consider the following example ($n = 11$):

$$\begin{array}{rccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ \boldsymbol{\beta} = & 0 & 0 & 1 & 1 & 2 & 3 & 2 & 3 & 4 & 5 & 6 \end{array}$$

Here the border array $\boldsymbol{\beta}$ corresponds to a string $\boldsymbol{x} = abaababaaba$, for example. In fact, it is easy to see that, up to an isomorphism on the alphabet, this string is the *only* one that corresponds to $\boldsymbol{\beta}$. From Lemma 2.1(b) we see that the admissible extensions m of $\boldsymbol{\beta}$ are

$$m = \left\{ \begin{array}{lllllllll} 0 & & & & & & & & \\ \boldsymbol{\beta}[11]{+}1 & = & 7 & & & & & & \\ \boldsymbol{\beta}^2[11]{+}1 & = & \boldsymbol{\beta}[6]{+}1 & = & 4 & & & & \\ \boldsymbol{\beta}^3[11]{+}1 & = & \boldsymbol{\beta}^2[6]{+}1 & = & \boldsymbol{\beta}[3]{+}1 & = & 2 & & \\ \boldsymbol{\beta}^4[11]{+}1 & = & \boldsymbol{\beta}^3[6]{+}1 & = & \boldsymbol{\beta}^2[3]{+}1 & = & \boldsymbol{\beta}[1]{+}1 & = & 1 \end{array} \right.$$

Of these five admissible extensions, only three ($m = 0, 7, 4$) can actually be used to extend $\boldsymbol{\beta}$ to a border array $00112323456m$; these extensions correspond to appending the letters c, b, a, respectively, to $\boldsymbol{x}$. (Of course, in place of c, any letter other than a or b could be used.) The extensions $m = 1, 2$ do not yield valid border arrays because, even though they give the lengths of borders of $\boldsymbol{x}[1..12] = \boldsymbol{x}[1..11]a$ and $\boldsymbol{x}[1..11]b$, respectively, they do not give the lengths of the *longest* borders.

In order to characterize those values of m that can be used to extend a border array $\boldsymbol{\beta}[1..n]$ to a border array $\boldsymbol{\beta}[1..n]m$, we make use of the following definition:

> A nonzero admissible extension m of a border array $\boldsymbol{\beta}$ is said to be *invalid* if and only if there exists an admissible extension m' of $\boldsymbol{\beta}$ such that $m = \boldsymbol{\beta}[m']$. Any other admissible extension of $\boldsymbol{\beta}$ is said to be a *valid* extension.

It follows from this definition that for $n \geq 1$ the admissible extensions

$$S^n_0 = 0, \quad S^n_1 = \boldsymbol{\beta}[n]{+}1$$

are always valid. We shall see in Theorem 2.2 that every valid extension determines a distinct border array; thus $b_n \geq 2^{n-1}$, as in fact we have seen in the sequence B whose first ten terms were given in the Introduction.

Observe that this definition is useful only for an unbounded alphabet. For example, the border array

$$\boldsymbol{\beta}[1..15] = 001012301234567$$

corresponds to a string

$$\boldsymbol{x} = abacabadabacaba$$

and has five valid extensions $m = 0, 8, 4, 2, 1$ that result from appending the letters e, d, c, b, a, respectively, to $\boldsymbol{x}$. However, if the alphabet size were limited to $\alpha = 4$, we would presumably not wish to regard $m = 0$ as "valid". The following theorem provides a justification for our use of this term.

Theorem 2.2 *For every $n \geq 1$, an integer array $\boldsymbol{y} = \boldsymbol{y}[1..n]$ is a border array if and only if $\boldsymbol{y}[1] = 0$ and each $\boldsymbol{y}[i]$ is a valid extension of $\boldsymbol{y}[1..i{-}1]$, $i = 2, 3, \ldots, n$.*

Proof The result is trivially true for $n = 1$, and so we may suppose $n \geq 2$.

To prove necessity, suppose that for some $i \in 1..n{-}1$,

$$\boldsymbol{y}[1..i] \text{ and } \boldsymbol{y}[1..i{+}1] = \boldsymbol{y}[1..i]m$$

are both border arrays, and let $\boldsymbol{x} = \boldsymbol{x}[1..i{+}1]$ denote a string with border array $\boldsymbol{y}[1..i{+}1]$. By Lemma 2.1(b), m must be an admissible extension of $\boldsymbol{y}[1..i]$. We suppose however that m is invalid and derive a contradiction.

Since m is invalid, there exists an admissible extension $m' > m$ of $\boldsymbol{y}[1..i]$ such that $\boldsymbol{y}[m'] = m$. Then $m' = \boldsymbol{y}^r[i]{+}1$ for some integer $r \geq 1$, and the following statements are true:

(1) $\boldsymbol{x}[1..m] = \boldsymbol{x}[i{-}m{+}2..i{+}1]$ since $\boldsymbol{y}[i{+}1] = m$;

(2) $\boldsymbol{x}[1..m] = \boldsymbol{x}[m'{-}m{+}1..m']$ since $\boldsymbol{y}[m'] = m$;

(3) $\boldsymbol{x}[1..m'{-}1] = \boldsymbol{x}[i{-}m'{+}2..i]$ since $m'{-}1 = \boldsymbol{y}^r[i]$.

From (1) and (2) we conclude that

$$\boldsymbol{x}[m'] = \boldsymbol{x}[m] = \boldsymbol{x}[i{+}1],$$

so that (3) can be extended to

$$\boldsymbol{x}[1..m'] = \boldsymbol{x}[i{-}m'{+}2..i{+}1].$$

Thus $\boldsymbol{x}[1..i{+}1]$ has a border of length $m' > m$, contradicting the assumption that $\boldsymbol{y}[1..i{+}1] = \boldsymbol{y}[1..i]m$ is a border array. We conclude that m must be valid, as required.

To prove sufficiency, let $\boldsymbol{y} = \boldsymbol{y}[1..n]$ be an array such that $\boldsymbol{y}[1] = 0$ and each $\boldsymbol{y}[i]$ is a valid extension of $\boldsymbol{y}[1..i{-}1]$, $i = 2, 3, \ldots, n$. We show by induction that $\boldsymbol{y}$ is a border array of some string.

Since $\boldsymbol{y}[1] = 0$, the result holds for $n = 1$. Suppose then that for $n \geq 2$ and some $i \in 2..n$, $\boldsymbol{y} = \boldsymbol{y}[1..i{-}1]$ is a border array of some string $\boldsymbol{x}[1..i{-}1]$. We show that therefore $\boldsymbol{y}[1..i]$ must be a border array.

Let $m = \boldsymbol{y}[i]$. By hypothesis m is a valid extension of $\boldsymbol{y}[1..i{-}1]$ and so by Lemma 2.1(b) two cases arise:

$\underline{m = 0}$ In this case $\boldsymbol{y}[1..i]$ is a border array of a string $\boldsymbol{x}[1..i-1]\lambda$, where the letter λ is chosen to be distinct from every previous letter in $\boldsymbol{x}[1..i-1]$.

$\underline{m > 0}$ Here $m = \boldsymbol{y}^p[i-1]+1$ for some integer $p \geq 1$, so that by the inductive hypothesis $\boldsymbol{x}[1..m-1]$ is a border of $\boldsymbol{x}[1..i-1]$. Then we can choose $\boldsymbol{x}[i] = \boldsymbol{x}[m]$, so that $\boldsymbol{x}[1..m]$ is a border of $\boldsymbol{x}[1..i]$ — we want to show that it is the longest border.

If $\boldsymbol{x}[1..m]$ is not the longest border of $\boldsymbol{x}[1..i]$, there must exist a longer border $\boldsymbol{x}[1..m']$ such that $m = \boldsymbol{y}[m']$. By Lemma 2.1(b), $m' = \boldsymbol{y}^r[i-1]+1$ for some positive integer $r < p$. But then by definition m is invalid, contrary to the original assumption that each $\boldsymbol{y}[i]$ is a valid extension of $\boldsymbol{y}[1..i-1]$. We conclude that $\boldsymbol{y}[1..i]$ is the border array of the string $\boldsymbol{x}[1..i] = \boldsymbol{x}[1..i-1]\boldsymbol{x}[m]$, as required.

□

This theorem makes clear that an extension $m = \boldsymbol{y}[i]$ of a border array $\boldsymbol{y} = \boldsymbol{y}[1..i-1]$ yields a border array $\boldsymbol{y}[1..i]$ if and only if

(1) m is an admissible extension of $\boldsymbol{y}$;

(2) there exists no admissible extension $m' > m$ of $\boldsymbol{y}$ such that $\boldsymbol{y}[m'] = m$.

The algorithm that determines whether or not a given array is a border array simply evaluates these two conditions in a straightforward manner for every position $i \in 2..n$. Thus the outline of the main algorithm can be expressed as follows:

```
— For y[1..n], n ≥ 1, return either n+1
— or the first position i ∈ 1..n
— such that y[i] is invalid.
if y[1] ≠ 0 then return 1
— repeatedly call the function valid to check each
— value y[i] until the whole array y is processed
i ← 2
while i ≤ n and valid(i, y[1..i]) do
  i ← i+1
return i
```

The Boolean function *valid* returns `TRUE` if and only if conditions (1) and (2) are satisfied by $m = \boldsymbol{y}[i]$, as shown in Figure 1. Since the algorithm processes $\boldsymbol{y}$ position-by-position from left to right, terminating whenever an invalid position is found, we may assume that for every $i \geq 2$, $\boldsymbol{y}[1..i-1]$ is a valid border array. Observe that the algorithm described here makes no reference to any corresponding string $\boldsymbol{x}$, but bases its determination of validity entirely on the properties of the given array $\boldsymbol{y}$.

Thus, based on Theorem 2.2 and this discussion, we may conclude that our algorithm is correct. To see that it executes in $O(n)$ time, we need to show that

```
function valid(i, y[1..i])
  — Given that y[1..i−1] is a border array,
  — return TRUE iff y[i] is valid.

  — First determine whether y[i] is admissible.
if y[i] = 0 then return TRUE
else
   b ← y[i − 1]
   while b > 0 and y[i] ≠ b + 1 do
      b ← y[b]                          (1.1)
   if y[i] ≠ b + 1 then return FALSE
   else
  — Now determine whether y[i] = b+1 satisfies condition
(2).
      b′ ← y[i − 1]
   while b′ > b and b + 1 ≠ y[b′ + 1] do
         b′ ← y[b′]                     (1.2)
      return (b′ ≤ b)
```

Figure 1: The Boolean Function *valid*

the total number of operations performed in the **while** loops of function *valid* is $O(n)$. But this follows as in the failure function algorithm [AHU74] [KMP77]. Specifically, consider the worst case in which *valid* always returns `TRUE`. From one call of *valid* to the next, the value of $b = \boldsymbol{y}[i-1]$ is increased by at most 1. However, each execution of (1.1) decreases b by at least 1. Therefore, (1.1) is executed at most $n-1$ times in total. Also note that for each specific i, (1.2) is executed at most the same number of times as (1.1) because of the condition $b' > b$.

The same analysis applies to the case where `FALSE` is returned except that the algorithm may terminate earlier, thus executing fewer steps. We conclude that our algorithm executes in linear time. This establishes the second main result of this section:

Theorem 2.3 *The algorithm presented in this section correctly determines in time $O(n)$ whether or not a given integer array $\boldsymbol{y}[1..n]$ is a border array.* □

To conclude this section, we remark that a version of Theorem 2.2 appears as Theorem 3.2 in [MSM99]; however, the result as it is given there is much less clear and its proof depends on an elaborate theory of b-canonical strings that we have avoided here with a proof that is elementary.

3 Identifying Valid Border Arrays For Bounded Alphabets

As shown in the previous section, the definition of a *valid extension* of a border array $\boldsymbol{y}[1..n]$ is not appropriate for dealing with strings over an alphabet of a fixed finite size α. The example there shows that the problem is how to determine when 0 is valid in relation to the size of the alphabet. For the rest of this section we assume that $\alpha \geq 2$ is the fixed finite alphabet size. We capture the revised notion of validity in the following definition:

> A nonzero admissible extension m of a border array $\boldsymbol{\beta} = \boldsymbol{\beta}[1..n]$ is said to be an *α-valid extension of $\boldsymbol{\beta}$* if and only if it is a valid extension of $\boldsymbol{\beta}$ (as defined in the previous section). 0 is said to be an *α-valid extension of $\boldsymbol{\beta}$* if and only if for the set
>
> $$M_n = \{m : m > 0, m \text{ an } \alpha - \text{valid extension of } \boldsymbol{\beta}\},$$
>
> $|M_n| < \alpha$.

Lemma 3.1 below shows that $|M_n|$ is precisely the number of letters (of any alphabet of size α) used in all possible extensions of all previous borders in any string of which $\boldsymbol{\beta}$ is a border array.

Let us illustrate using the example of the previous section, where 8,4,2, and 1 are all valid (hence α-valid) extensions of the border array $\boldsymbol{\beta}[1..15] =$ 001012301234567. The string $\boldsymbol{x} = abacabadabacaba$ is a string of which $\boldsymbol{\beta}$ is a border array, and so we see that the alphabet $\{a, b, c, d\}$ must have size at least 4. Hence if $\alpha = 4$, 0 is not α-valid, while if $\alpha \geq 5$, it is α-valid (appending e to $\boldsymbol{x}$ gives 0).

Lemma 3.1 *Let $n \geq 2$ be an integer, and let $\boldsymbol{y}[1..n]$ be a border array. Let m_1 and $m_2 \neq m_1$ be two distinct α-valid extensions of $\boldsymbol{y}[1..n]$. Then for any string $\boldsymbol{x}[1..n]$ such that $\boldsymbol{y}[1..n]$ is its border array, $\boldsymbol{x}[m_1] \neq \boldsymbol{x}[m_2]$.*

Proof By contradiction. We assume that there exists a string $\boldsymbol{x}$ such that $\boldsymbol{x}[m_1] = \boldsymbol{x}[m_2]$, where $m_1 = \boldsymbol{y}^i[n]{+}1$ and $m_2 = \boldsymbol{y}^j[n]{+}1$ for some $1 \leq i < j \leq k$ (k being the smallest integer such that $\boldsymbol{y}^k[n] = 0$). Moreover assume that, for given j, the integer i is the maximum value satisfying this condition.
Since $\boldsymbol{x}\big[\boldsymbol{y}^i[n]{+}1\big] = \boldsymbol{x}\big[\boldsymbol{y}^j[n]{+}1\big]$ and $i < j$, $\boldsymbol{y}\big[\boldsymbol{y}^i[n]{+}1\big] \geq \boldsymbol{y}^j[n]{+}1$. Since $\boldsymbol{y}^j[n]{+}1$ is α-valid, $\boldsymbol{y}\big[\boldsymbol{y}^i[n]{+}1\big] \neq \boldsymbol{y}^j[n]{+}1$, and so $\boldsymbol{y}\big[\boldsymbol{y}^i[n]{+}1\big] > \boldsymbol{y}^j[n]{+}1$. Furthermore, $\boldsymbol{y}\big[\boldsymbol{y}^i[n]{+}1\big] = \boldsymbol{y}^r[n]{+}1$ for some r, $i < r \leq k$. Thus

$$\boldsymbol{x}\big[\boldsymbol{y}^r[n]{+}1\big] = \boldsymbol{x}\big[\boldsymbol{y}^i[n]{+}1\big] = \boldsymbol{x}\big[\boldsymbol{y}^j[n]{+}1\big].$$

Since $\boldsymbol{y}^r[n]{+}1 > \boldsymbol{y}^j[n]{+}1$, $1 \leq i < r < j \leq k$. But this contradicts the maximality of i. □

The following theorem is an obvious modification of Theorem 2.2 for bounded alphabets.

Theorem 3.2 *For every $n \geq 1$, and for any $\alpha \geq 2$, an integer array $\boldsymbol{y} = \boldsymbol{y}[1..n]$ is a border array of a string over an alphabet of size α if and only if $\boldsymbol{y}[1] = 0$ and each $\boldsymbol{y}[i]$ is an α-valid extension of $\boldsymbol{y}[1..i-1]$, $i = 2, 3, \ldots, n$.*

Proof If $\boldsymbol{y}[i] > 0$, the proof follows the same argument as in the proof of Theorem 2.2, so we refer the reader there. Below, we cover the case when $\boldsymbol{y}[i] = 0$.

Necessity: let $\boldsymbol{y}[1..i]$ be a border array and $\boldsymbol{y}[i] = 0$. By contradiction assume that 0 is not an α-valid extension of $\boldsymbol{y}[1..i-1]$; that is, that $|M_{i-1}| = \alpha$. Take any string $\boldsymbol{x}[1..i]$ of which $\boldsymbol{y}[1..i]$ is a border array. Then for some α-valid $m > 0$, $\boldsymbol{x}[m] = \boldsymbol{x}[i]$. It follows that $y[i] \geq m > 0$, a contradiction.

Sufficiency: let $\boldsymbol{y}[1..i-1]$ be a border array and suppose that $|M_{i-1}| < \alpha$. Take any string $\boldsymbol{x}[1..i-1]$ of which $\boldsymbol{y}[1..i-1]$ is a border array. Suppose $\lambda \neq \boldsymbol{x}[m]$ for any $m \in M_{i-1}$ (The fact that such a λ exists follows from Lemma 3.1.) We will show that $\boldsymbol{y}[1..i-1]0$ is a border array of $\boldsymbol{x}[1..i-1]\lambda$. By contradiction assume that the border array $\boldsymbol{y}[1..i]$ of $\boldsymbol{x}[1..i-1]\lambda$ is such that $\boldsymbol{y}[i] \neq 0$. But then by this theorem for nonzero values, $\boldsymbol{y}[i] = m$ for some α-valid m, hence $\lambda = \boldsymbol{x}[i] = \boldsymbol{x}[m]$, a contradiction. □

This result makes it clear that for $\boldsymbol{y}[i] \neq 0$, the algorithm of Section 2 can be used as it stands to determine whether or not $\boldsymbol{y}[i]$ is α-valid. However, whenever $\boldsymbol{y}[i] = 0$ and $i > 1$, the definition of α-valid requires additional processing to determine whether or not the number of valid nonzero extensions of $\boldsymbol{y}[1..i-1]$ equals α. This processing can be viewed as a modification to function *valid*, that affects only the case $\boldsymbol{y}[i] = 0$. The code is presented in Figure 2. A Boolean array $\boldsymbol{V}[1..n]$ indicates whether or not each admissible extension m is valid ($\boldsymbol{V}[m] =$ `TRUE`) or not valid ($\boldsymbol{V}[m] =$ `FALSE`). We initialize $\boldsymbol{V}[1..n]$ to `FALSE` before the first invocation of *valid*.

The new code is straightforward and falls into four parts:

* determine the number k of nonzero admissible extensions m and set each corresponding $\boldsymbol{V}[m]$ to `TRUE`;
* identify invalid admissible extensions and thus compute v, the number of nonzero valid admissible extensions;
* reset to `FALSE` the components of $\boldsymbol{V}$ that were changed in the first part;
* return `TRUE` if and only if $v < \alpha$.

We claim that the revised algorithm correctly determines whether or not a given array $\boldsymbol{y}$ is a border array of some string on an alphabet of size α.

To analyze the complexity of the revised algorithm, we need to show that the extra processing involved when $\boldsymbol{y}[i] = 0$ still requires only $O(n)$ steps. To this end, fix such an i and let s be the number of times (2.1) is executed. Then $s \leq \boldsymbol{y}[i-1] + 1$ since b is set to $\boldsymbol{y}[i-1]$ on the first pass and is reduced by at least 1 on each subsequent pass. Note that s distinct components of $\boldsymbol{V}$ are set to `TRUE`.

```
if y[i] = 0 then
   k ← 0; b ← i − 1
   while b ≠ 0 do
      k ← k+1; b ← y[b]; V[b + 1] ← TRUE                (2.1)
   v ← k; b ← i − 1
   while b ≠ 0 do
      b ← y[b]                                          (2.2)
      b′ ← b + 1
      if V[b′] then
         while y[b′] > 0 do
            v ← v−1; b′ ← y[b′]; V[b′] ← FALSE          (2.3)
    b ← i − 1
   while b ≠ 0 do
      b ← y[b]; V[b + 1] ← FALSE
   return (v < α)

else

   — (The rest of function valid follows.)
```

Figure 2: Determining whether $\boldsymbol{y}[i] = 0$ is α-valid or not

Now the second **while** loop is controlled by the same parameters as the first. In particular, (2,2) is executed s times. Further, each execution of (2.3) sets to FALSE a component of $\boldsymbol{V}$ that was previously TRUE. Thus (2.3) executes at most s times in total – more than once on some passes and not at all on others. The third **while** loop (which resets $\boldsymbol{V}$) is also executed s times.

Now, as in Section 2, $b = \boldsymbol{y}[i-1]$ increases by at most 1 on each call of *valid*. However, b decreases by $\boldsymbol{y}[i-1]$ when $\boldsymbol{y}[i] = 0$. Thus the sum of all these $\boldsymbol{y}[i-1]$ cannot exceed $n-1$ and hence the sum of the corresponding values of s cannot exceed $2(n-2)$. It follows that overall, $O(n)$ time is sufficient to handle the cases in which $\boldsymbol{y}[i] = 0$, in spite of the nested **while** loops. Since the remainder of the processing is unchanged, we conclude:

Theorem 3.3 *The algorithm presented in this section correctly determines in time $O(n)$ whether or not a given integer array $\boldsymbol{y}[1..n]$ is a border array on an alphabet of specified size α.* □

4 Computing All Border Arrays of Length At Most n

In [MSM99] all the border arrays of length at most n are generated by growing a trie T_n of height n in which every simple path of length $k \leq n$ from the root spells out both a unique border array $\boldsymbol{\beta} = \boldsymbol{\beta}[1..k]$ and the b-canonical string $\boldsymbol{x}[1..k]$ corresponding to $\boldsymbol{\beta}$. Thus each node of T_n may be thought of as being

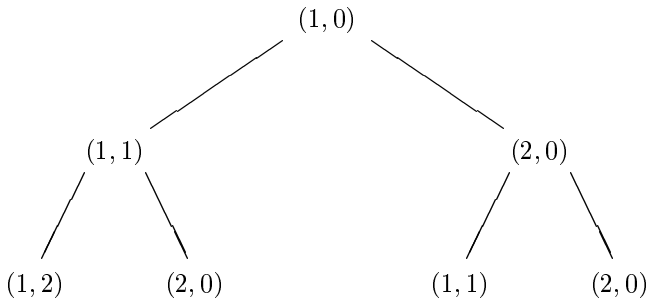


Figure 3: Trie T_3 — All Border Arrays of Length $k \leq 3$

labelled with an integer pair (i, β), where i denotes the i^{th} smallest letter λ_i in an ordered standard alphabet, and β is the value of a corresponding entry in the border array $\boldsymbol{\beta}$. For $n = 3$, T_n appears as shown in Figure 3, representing strings

$$\lambda_1\lambda_1\lambda_1,\ \lambda_1\lambda_1\lambda_2,\ \lambda_1\lambda_2\lambda_1,\ \lambda_1\lambda_2\lambda_2$$

with corresponding border arrays

$$012, 010, 001, 000.$$

The algorithm described in [MSM99] uses the canonical string $\boldsymbol{x}[1..k]$ spelled out by the path from the root to the current node N as a means of determining the children of N. Effectively, standard letters $\lambda_1, \lambda_2, \ldots$ are appended one-by-one to $\boldsymbol{x}[1..k]$ yielding new strings $\boldsymbol{x}[1..k]\lambda_i$, $i = 1, 2, \ldots$; for each new string formed, the corresponding $(k+1)^{\text{st}}$ border array element is computed. The process terminates when a standard letter, say λ_r, is appended for which the corresponding border array value is zero (see Lemma 3.4 in [MSM99]). Thus for each new node of T_n, one step in the failure function calculation is performed, requiring amortized constant time as discussed in Section 2. Hence the claim that T_n is constructed in time proportional to the number of nodes; that is, proportional to the number of border arrays (and corresponding strings) generated.

But the algorithm described in [MSM99] generates T_n in a breadth-first or on-line manner: T_{k+1} is actually computed from the leaf nodes of T_k for every $k \in 1..n-1$. Since for every node N both the corresponding $\boldsymbol{x}[1..k]$ and $\boldsymbol{\beta}[1..k]$ need to be available for the failure function calculation, $\Theta(k)$ time will be required to traverse the path from the root to node N in order to compute them. Thus the time required to compute each child of node N in a breadth-first algorithm is not constant, but rather $\Theta(k/r)$, where r is the number of children of N.

The obvious correction to the [MSM99] algorithm is to build T_n in a depth-first manner that uses two working-storage arrays $\boldsymbol{x} = \boldsymbol{x}[1..n]$ and $\boldsymbol{\beta}[1..n]$ to store the path from the root of T_n to the current node N. Then for each node

N, the current values $\boldsymbol{x}[1..k]$ and $\boldsymbol{\beta}[1..k]$ are known and can be used to compute the children of N: each extension $\boldsymbol{x}[1..k]\lambda_i$, $i = 1, 2, \ldots, r$, can be formed so that corresponding extensions of $\boldsymbol{\beta}[1..k]$ can be computed using a single constant-time step of the failure function algorithm. A depth-first recursion that computes all the children of each N before computing any of N's siblings will then lead to the result claimed in [MSM99]: T_n will be constructed in $\Theta(b_n)$ time using $\Theta(b_n)$ space.

The depth-first approach also enables us to solve efficiently a problem raised in [MSM99]: the computation of a trie T'_n whose node labels consist only of border array values β, omitting the elements of the underlying canonical string $\boldsymbol{x}$. This can easily be accomplished by maintaining the working storage array $\boldsymbol{x}[1..n]$ but not storing the current letter in the current node N: the algorithm will execute recursively in exactly the same way.

Note that it is straightforward to modify each of these depth-first algorithms to generate a trie for a given bounded alphabet A of size α: it is necessary only to replace the number m of children computed at each node by $\min\{r, \alpha\}$. We can now state formally the main result of this section:

Theorem 4.1 *For any given positive integer n, the two algorithms outlined in this section compute all possible border arrays of length at most n on either a bounded or unbounded alphabet in time $\Theta(b_n)$ and space $\Theta(b_n)$, where b_n is the number of arrays generated.* □

We remark that the depth-first algorithms described above have the disadvantage that they provide no means of efficiently computing T_{n+1} (respectively, T'_{n+1}) from T_n (respectively, T'_n). In the next section, we provide an on-line (breadth-first) algorithm that performs the same computation with equal efficiency.

5 An On-line Algorithm for Computing All Border Arrays of Length At Most n

We wish to construct the trie T_n in a breadth-first fashion using only constant time to compute the border array entry of each node. The failure function algorithm satisfies this requirement, but it needs the corresponding arrays $\boldsymbol{x}$ and $\boldsymbol{\beta}$ to be available. The elements of the two arrays are already stored in the ancestor nodes of the tree. We will modify the failure function algorithm to use this information instead of the two arrays.

The failure function algorithm accesses the arrays $\boldsymbol{x}$ and $\boldsymbol{\beta}$ in the following two ways:

(a) $\boldsymbol{\beta}[b]$, which returns the length of the longest border of the position b; and

(b) $\boldsymbol{x}[b+1]$, which returns the letter at the next position of b.

Each node in our trie T_n will represent a border array/b-canonical string pair. Then the above operation (a) is equivalent to finding the node corresponding

to the longest border of the string represented by the current node. Operation (b) is equivalent to finding one of the children of the current node. We refine the tree structure of T_n to simulate these two operations.

First we introduce an extra node E which corresponds to the 0-position or empty string/border. A pointer in E points to the "real" root R of T_n: $(\lambda_1, 0, 1)$, which means that the current letter is λ_1, with empty border, and the length of the string is 1.

Except for E, each node N in T_n represents a border array $\boldsymbol{\beta}[1..j]$ and its b-canonical string $\boldsymbol{x}[1..j]$. We have the following data in each node N:

λ the letter $\boldsymbol{x}[j]$;

β the border array entry $\boldsymbol{\beta}[j]$;

j the length of the corresponding border array/b-canonical string;

pr a pointer to the parent: the node of $\boldsymbol{\beta}[1..j-1]/\boldsymbol{x}[1..j-1]$;

br a pointer to the border: the node of $\boldsymbol{\beta}[1..\boldsymbol{\beta}[j]]/\boldsymbol{x}[1..\boldsymbol{\beta}[j]]$;

ne a pointer to the next node of the same level as N;

cp a list of pointers to the children of N;

ci a pointer to the child whose descendant is currently being processed.

The pointers ci are maintained to form a path from E to the node N to which a child N' is being added. Note that j is just the level in the tree where $E.j = 0$. Also, we define $E.ci = R$ and $R.br = E$. These values do not change.

We now describe the construction of T_n. Beginning with E, each new level is an ordered sequence of nodes. Successive children of a given node are constructed by appending a different letter (using the ordering of the alphabet) to the string represented by the parent until an empty border results. The algorithm for computing the border is given below.

When adding a new node, it is trivial to determine the values of λ, j, pr, ne and cp. Also, β is just the level pointed to by br. We now show how to compute br and ci:

br: When adding a child N' to the current node N, we proceed as follows:

$N_b \leftarrow N.br$; $N_{b+1} \leftarrow N_b.ci$
while $N_b \neq E$ **and** $N'.\lambda \neq N_{b+1}.\lambda$ **do**
  $N_b \leftarrow N_b.br$; $N_{b+1} \leftarrow N_b.ci$
if $N'.\lambda = N_{b+1}.\lambda$ **then**
  $N'.br \leftarrow N_{b+1}$
else
  $N'.br \leftarrow E$

$$N'.\beta \leftarrow N'.br.j$$

First we use $N.br$ to find the border node N_b of N, then use $N_b.ci$ to find the next node N_{b+1} along the path from N_b to N. Now we compare the letters of N' and N_{b+1}. We continue to do so until we find a match or reach the empty border. Finally we set the br and β according to whether the letters match or not.

ci: When a first child of N is added, we can set $N.ci$ to this child. This value does not have to be changed until we start adding grandchildren for N. When finished adding children to a node N, we use $N.ne$ to find the next node. Now we need to update the ci of ancestor nodes of N. For example, suppose that a node N_p has two children N_1 and N_2. When we are adding children to N_1, $N_p.ci = N_1$. Now we want to add children to N_2. We need to update ci so that $N_p = N_2$. Each time we finish adding children to a node N, we call the procedure $update\_ci(N.pr)$. It is possible that N is the last child of its parent N_p. Then we may need to update the ci of the parent nodes recursively. In the following, $next$ denotes the pointer field in the linked list cp, and $cp.first$ returns the first node of cp:

```
update_ci(N.pr)
procedure update_ci(N_p)
  if ci.next ≠ NULL then
    ci ← ci.next
  else
    ci ← cp.first
    if N_p ≠ R then
      update_ci(N_p.pr)
```

We update ci to point to the next child, if there is any. Otherwise, we set ci to point to the first child, and update ci of the parent node.

According to the above description, we have an on-line algorithm that computes all the border arrays/b-canonical strings of length n. Except for cp, all data in each node require constant space. The total number of nodes in cp is equal to the total number of child nodes in the tree, which is $|T_n|-1$. So the algorithm still needs space proportional to the number of arrays generated.

The code for br is basically the same as the failure function algorithm. Thus it takes amortized constant time for each node. When we generate T_{n+1} from T_n, the number of times we update ci is the same as the number of nodes in cp, which is $|T_n|-1$. Thus we get the following conclusion about the complexity of the algorithm:

Theorem 5.1 *For every positive integer n, the algorithm outlined in this section computes all border arrays/b-canonical strings of length n in $\Theta(b_n)$ time and represents them in $\Theta(b_n)$ space.* □

An easy modification can be made to the algorithm for a bounded alphabet of size $\alpha \geq 2$. Theorem 5.1 also holds in this case.

References

[AHU74] Alfred V. Aho, John E. Hopcroft & Alfred D. Ullman, *The Design & Analysis of Computer Algorithms*, Addison-Wesley (1974).

[KMP77] Donald E. Knuth, James H. Morris & Vaughan R. Pratt, **Fast pattern matching in strings**, *SIAM J. Comput. 6-2* (1977) 323-350.

[MSM99] Dennis Moore, W. F. Smyth & Dianne Miller, **Counting distinct strings**, *Algorithmica 23* (1999) 1-13.

[SP95] N. J. A. Sloane & Simon Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press (1995). See also

`http://www.research.att.com/~njas/sequences/`

Acknowledgements

This work was supported in part by grants from the Natural Sciences & Engineering Research Council of Canada.

On rotation distance between binary coupling trees and applications for $3nj$-coefficients

V. Fack, S. Lievens and J. Van der Jeugt[1]
Department of Applied Mathematics and Computer Science,
University of Ghent, Krijgslaan 281-S9, B-9000 Gent, Belgium

Abstract

Generalized recoupling coefficients or $3nj$-coefficients for a Lie algebra (with su(2), the Lie algebra for the quantum theory of angular momentum, as generic example) can always be expressed as multiple sums over products of Racah coefficients (i.e. $6j$-coefficients). In general there exist many such expressions; we say that such an expression is optimal if the number of Racah coefficients in such a product (and, correlated, the number of summation indices) is minimal. The problem of finding an optimal expression for a given $3nj$-coefficient is equivalent to finding a shortest path in a graph G_n. The vertices of this graph G_n consist of binary coupling trees, representing the coupling schemes in the bra/kets of the $3nj$-coefficients. This is the graph of rooted (unordered) binary trees with labelled leaves, and has order $(2n-1)!!$. As the order increases so rapidly, finding a shortest path is computationally achievable only for $n < 11$. We present some mathematical tools to compute or estimate the length of such shortest paths between binary coupling trees. The diameter of G_n is determined explicitly upto $n < 11$, and it is shown to grow like $n \log(n)$. Thus for n large enough, the number of Racah coefficients in the expansion of a $3nj$-coefficient is of order $n \log(n)$. We also show that this problem in Racah-Wigner theory is equivalent to a problem in mathematical biology, where one is concerned with the quantitative comparison of classifications or dendrograms. From this context, some algorithms for approximating the shortest path can be deduced.

PACS : 02.20 (Group Theory); 02.70 (Computational techniques); 03.65F (Algebraic methods in Quantum Theory).

Corresponding author : V. Fack, Department of Applied Mathematics and Computer Science, University of Ghent, Krijgslaan 281-S9, B-9000 Gent, Belgium.
Tel. ++ 32 9 2644808; Fax ++ 32 9 2644995; E-mail Veerle.Fack@rug.ac.be.

[1]Research Associate of the Fund for Scientific Research – Flanders (Belgium)

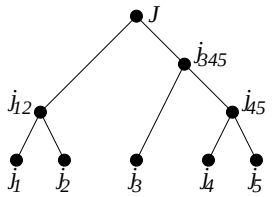
1 Introduction

The subject of the coupling of $n+1$ angular momenta, and the related $3nj$-coefficients, is a difficult one, and the literature is extensive. Classical monographs [1, 2] deal primarily with techniques for implementing graphical methods (known as Yutsis graphs) for carrying out summations over projection quantum numbers in products of Wigner coefficients. In [3, Topic 12], recoupling theory is considered from the point of view of *binary coupling schemes*. A binary coupling scheme is the rooted binary tree representing the order of coupling of a state vector in the tensor product of $n+1$ angular momentum multiplets, labelled respectively by the angular momenta $j_1, j_2, \ldots, j_{n+1}$. The leaves of the binary tree are labelled by these angular momenta $j_1, j_2, \ldots, j_{n+1}$, and the remaining vertices of the tree can be labelled by the intermediate angular momenta. For example, in the tensor product $V_1 \otimes \cdots \otimes V_5$, where each V_i carries a representation of the angular momentum algebra labelled by j_i, the following vector can be considered :

$$\begin{aligned}|((j_1,j_2)j_{12},(j_3,(j_4,j_5)j_{45})j_{345})J,M\rangle = \\ \sum_{\text{all } m_i} C^{j_1,j_2,j_{12}}_{m_1,m_2,m_{12}} C^{j_4,j_5,j_{45}}_{m_4,m_5,m_{45}} C^{j_3,j_{45},j_{345}}_{m_3,m_{45},m_{345}} C^{j_{12},j_{345},J}_{m_{12},m_{345},M} \\ \times |j_1 m_1\rangle \otimes |j_2 m_2\rangle \otimes |j_3 m_3\rangle \otimes |j_4 m_4\rangle \otimes |j_5 m_5\rangle. \end{aligned} \tag{1}$$

Herein, $C^{j,j',j''}_{m,m',m''}$ is a vector-coupling (Wigner or Clebsch-Gordan) coefficient [3, 4]. The binary coupling scheme representing the above vector is given in Figure 1. The projection

Figure 1: Binary coupling scheme representing (1)



quantum number M is not represented in this binary coupling scheme for reasons that will soon become apparent.

There are obviously several ways in which $n+1$ angular momenta can be coupled, and the quantities that typically appear in atomic and nuclear structure computations are the related general recoupling coefficients or $3nj$-coefficients. A general recoupling coefficient (or a generalized $3nj$-coefficient) is defined to be the transformation coefficient between any such two coupling schemes, e.g.

$$\langle(((j_1,j_4)j_{14},(j_2,j_3)j_{23})j_{1423},j_5)J|((j_1,j_2)j_{12},(j_3,(j_4,j_5)j_{45})j_{345})J\rangle. \tag{2}$$

The M-dependence is dropped since such coefficients are independent of M by the Wigner-Eckart theorem [4]. It is a fundamental theorem of recoupling theory [3, p. 455] that each such transformation coefficient (i.e. every generalized $3nj$-coefficient) can be expressed in

terms of sums over products of Racah coefficients ($6j$-coefficients). A famous program of Burke [5], `NJSYM`, is already dealing with this problem. Burke's approach is equivalent to finding a certain path between the two binary coupling schemes (representing the bra- and ket-part of the general recoupling coefficient) by successive elementary transformations on the trees. As we shall explain later, the shorter this path, the better the resulting formula. The path found by `NJSYM` is generally rather long, thus yielding expressions which are far from optimal. In order to improve `NJSYM`, Bar-Shalom and Klapisch [6] developed a new program `NJGRAF` by implementing graphical methods due to Yutsis [1]. Recently, both methods were re-examined and a better implementation was given [7, 8, 9].

In the present paper we consider the method of binary coupling tree transformations as used in `NJSYM` [5] and `NJFORMULA` [7], and relate it to problems in graph theory, mathematical biology, and computer science. For fixed n, we consider the set of all binary coupling trees with $n+1$ leaves (i.e. $n+1$ basic angular momenta). This set is shown to have the natural structure of a graph, G_n, with each vertex of G_n representing a binary coupling tree. Two vertices in G_n are connected by an edge if there exists an elementary transformation (to be defined later) between the corresponding binary coupling trees. In order to find an optimal expression for a general recoupling coefficient (generalized $3nj$-coefficient), it is then sufficient to consider the two vertices in G_n corresponding to the bra- and ket-vector, and to find a shortest path between them in G_n. Although this is a simple reduction of the original problem, the new graph theoretical problem turns out to be as hard as the original problem. One advantage of the equivalent graph theoretical problem is that it has appeared in a number of different contexts, such as computer science and mathematical biology, and thus some properties of the graphs G_n can be found in the literature. In pure graph theory, the problem was first considered by Robinson [10]. In computer science, the equivalent problem is known as finding the rotation distance between unordered rooted binary trees with labelled leaves [11]. In mathematical biology, the problem is known as computing the nearest neighbour interchange metric between dendrograms [12, 13, 14, 15].

Since our main problem is now reduced to finding shortest paths in the graph G_n, we shall study some properties of G_n that are related to distance [16]. In particular we shall be concerned with calculating or estimating the diameter $d(G_n)$ of G_n, since this gives an upper bound for the number of Racah coefficients appearing in the expressions of our generalized $3nj$-coefficients. For $n < 11$, $d(G_n)$ is computed explicitly by means of a computer program. Since the number of vertices of G_n grows rapidly (the order of G_n is $(2n-1)!!$), $d(G_n)$ can no longer be computed for $n \geq 11$. Then, we use a number of techniques from computer science to give upper and lower bounds for $d(G_n)$. We show that $d(G_n)$ grows like $n\log(n)$. Some properties of G_n that are known in the literature are then summarized and converted to our context of $3nj$-coefficients. Finally, we propose some ideas on how to compute approximations for the shortest path problem in G_n.

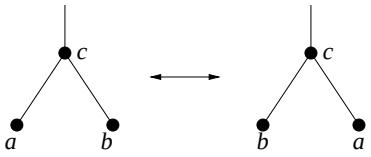
2 Transformations on binary coupling trees

The two parts of a general recoupling coefficient or a $3nj$-coefficient (i.e. the bra- and ket-vector) consist of binary coupling schemes. As shown in the example in Figure 1, the vertices of a binary coupling scheme are labelled by the angular momentum values. The leaves of the binary coupling scheme are labelled by basic angular momentum labels j_i; the other vertices (the coupled vertices) are labelled by the intermediate angular momentum values; and the root or top vertex is labelled by the final angular momentum value J.

As observed by Burke [5] and used in [7], to find an expression of a general recoupling coefficient as a (multiple) sum over products of Racah coefficients, it is sufficient to find a sequence of elementary operations which transform the binary coupling scheme of the bra-vector into the binary coupling scheme of the ket-vector. There are two elementary operations, both corresponding to simple recoupling coefficients and thus to simple contributions in the summation formula for the $3nj$-coefficient.

We refer to [7] for a detailed description of the two elementary operations, and just recall their main properties here. The first elementary operation is called an *exchange* (terminology of [7]) or a *twist* (computer science terminology). It corresponds to the transformation of a state vector of the form $|(a,b)c\rangle$ to $|(b,a)c\rangle$. Its effect on a binary coupling scheme is shown in Figure 2, where a and b can be leaves or coupled vertices. Its

Figure 2: Twist operation



value is determined by the recoupling coefficient

$$\langle (a,b)c \mid (b,a)c\rangle = (-1)^{a+b-c}, \tag{3}$$

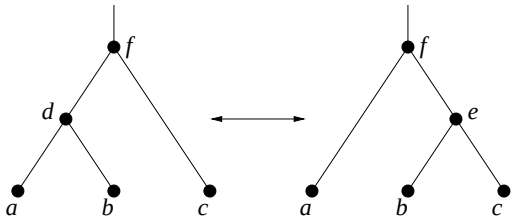
following from the Clebsch-Gordan coefficient property

$$C^{a,b,c}_{\alpha,\beta,\gamma} = (-1)^{a+b-c} C^{b,a,c}_{\beta,\alpha,\gamma}. \tag{4}$$

The second elementary operation is called a *flop* (terminology of [7]) or a *rotation* (computer science terminology in the context of binary search trees). This is a transformation of a state vector of the form $|((a,b)d,c)f\rangle$ to $|(a,(b,c)e)f\rangle$ or vice versa. Its effect on a binary coupling scheme is shown in Figure 3; here again, a, b or c can be leaves or coupled vertices. Its value is determined by the recoupling coefficient

$$\begin{aligned}\langle ((a,b)d,c)f \mid (a,(b,c)e)f\rangle &= \langle (a,(b,c)e)f \mid ((a,b)d,c)f\rangle \\ &= U^{a,b,d}_{c,f,e} = (-1)^{a+b+c+f}\sqrt{(2d+1)(2e+1)}\begin{Bmatrix} a & b & d \\ c & f & e \end{Bmatrix}.\end{aligned} \tag{5}$$

Figure 3: Rotation on binary coupling scheme



Herein, U is a Racah coefficient, and the last symbol is a $6j$-coefficient [4]. The Racah coefficient can be defined as

$$U^{a,b,d}_{c,f,e} = \sum_{m_1,\ldots,m_5} C^{a,b,d}_{m_1,m_2,m_3} C^{d,c,f}_{m_3,m_4,m} C^{b,c,e}_{m_2,m_4,m_5} C^{a,e,f}_{m_1,m_5,m}. \tag{6}$$

In order to obtain an expression in terms of Racah coefficients for a general recoupling coefficient (or $3nj$-coefficient), we start from the binary coupling scheme of the bra-vector and try to transform it into the binary coupling scheme of the ket-vector, by applying a sequence of elementary operations on subtrees of the coupling scheme [7]. Each operation contributes a part in the final expression. For a twist, the only contribution is a sign factor of the form (3). For a rotation, the contribution is a factor $U^{a,b,d}_{c,f,e}$; moreover (when going from left to right in Figure 3, for example) if e does not yet appear as a vertex in the binary coupling scheme of the ket-vector, then this operation also gives rise to a new summation variable $\sum_e$. In this case the rotation is said to create a new vertex e. The final expression for the general recoupling coefficient is then a (multiple) summation formula over the products of all contributions corresponding to the sequence of elementary operations. For an example, see [7, Section 2]. This leads to the following :

Theorem 1 *Consider a general recoupling coefficient or $3nj$-coefficient $\langle I \mid F \rangle$, with I and F two couplings of $n+1$ basic angular momenta. Let i and f be the binary coupling schemes corresponding to I and F respectively. If S is a sequence of elementary operations consisting of s_t twists and s_r rotations transforming i to f, then there exists an expression for the $3nj$-coefficient as a multiple sum with each term consisting of a product of s_r Racah coefficients (and a phase factor). Moreover, the number of summation variables is equal to the number of rotations in S that create a new vertex.*

We shall refer to such an expression as *an expansion of the $3nj$-coefficient in terms of Racah coefficients.*

In order to determine an optimal expression for a $3nj$-coefficient, one should find a sequence of elementary operations consisting of the minimum number of rotations. Indeed, a twist is inexpensive since it contributes only a sign (and never an extra summation variable). A rotation however is expensive since it contributes a Racah coefficient (computationally expensive since this involves the evaluation of a single sum expression),

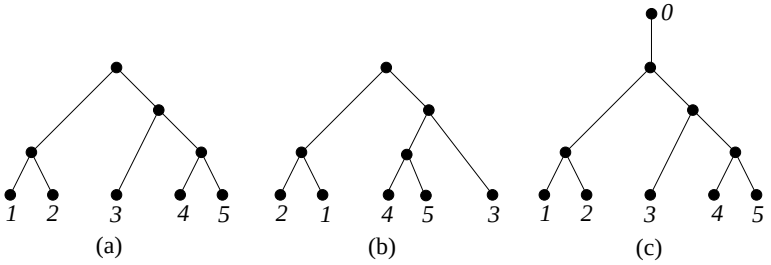
and since it can give rise to an extra summation variable. In the terminology of angular momentum, a twist is irrelevant since the corresponding state vectors are related by a phase factor, whereas a rotation is crucial since the corresponding state vectors are related through different intermediate angular momenta. Thus, we shall say that an expansion of a $3nj$-coefficient is optimal if the number of Racah coefficients appearing in such an expansion is minimal.

With this in mind we can redefine our problem and the basic structure that it is dealing with. A *binary coupling tree* on $n+1$ leaves is a rooted binary tree such that

- the $n+1$ leaves are labelled $1, 2, \ldots, n+1$;
- the internal vertices are not labelled;
- for each internal vertex, one can exchange the left and right children of that vertex.

These are sometimes referred to as *unordered rooted binary trees with labelled leaves*. An example is given in Figure 4. Note that in this figure, (a) and (b) represent the same

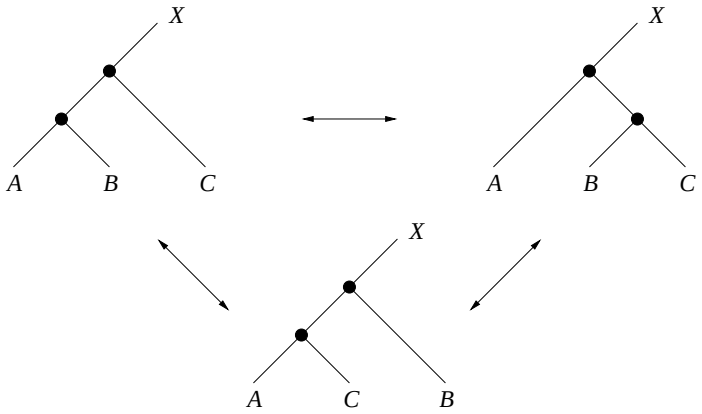
Figure 4: Binary coupling trees



binary coupling tree, since one can freely exchange the left and right children. Sometimes it will be convenient to attach an extra vertex with label 0 to the root of the binary coupling tree, such as in (c). The only elementary operation that now remains is rotation on binary coupling trees (since by definition a twist does not change the binary coupling tree). This is illustrated in Figure 5, where A, B and C represent subtrees and X is a part of the binary coupling tree containing the root (or, equivalently, the label 0).

The relation with binary coupling schemes is obvious. The leaf labels $1, 2, \ldots, n+1$ refer to the angular momentum values $j_1, j_2, \ldots, j_{n+1}$. An internal vertex is no longer explicitly labelled, but it is implicitly labelled by the collection of leaves underneath it. For a given $3nj$-coefficient with binary coupling schemes i and f and for every sequence S consisting of s_t twists and s_r rotations transforming i into f, there exists a sequence of s_r rotations transforming the corresponding binary coupling trees into each other and vice versa. Clearly, from the sequence of rotations between the binary coupling trees, the sequence of twists and rotations between the binary coupling schemes can be reconstructed and hence no information is lost for determining the summation formula for the $3nj$-coefficient. Our basic problem is now reduced to finding a shortest sequence of rotations transforming one binary coupling tree into another.

Figure 5: Binary coupling trees related by a rotation



A binary coupling tree can be given either explicitly as a graph, see Figure 4, or as a bracketing of the leaf labels. For example, the binary coupling tree of Figure 4 could be represented as

$$((1,2),(3,(4,5))) \qquad \text{or} \qquad ((2,1),((4,5),3)). \tag{7}$$

Henceforth we shall use this notation for a binary coupling tree. Sometimes we shall even use it to refer to the underlying binary coupling scheme itself, if the intermediate angular momentum values play no explicit role.

In the following section we shall show that finding a sequence of rotations transforming one binary coupling tree into another one is equivalent to finding a path in a graph G_n. First, there are two important observations.

Remark 2 We wish to draw attention to the fact that this method of binary coupling trees is more generally applicable than the case of the angular momentum algebra considered here. The angular momentum algebra is the Lie algebra $su(2)$, and the multiplets correspond to finite-dimensional irreducible representations of $su(2)$. Also for an arbitrary finite-dimensional semi-simple Lie algebra g, one can consider the $(n+1)$-fold tensor product $V_1 \otimes \cdots \otimes V_{n+1}$. Just as in (1), one can define vectors in this tensor product using the Clebsch-Gordan coefficients of g; then j_i stands for the representation labels of V_i and m_i for the internal labels of the vector. Since the tensor product is in general no longer multiplicity-free, the coupled vectors are labelled by representation labels and an additional label (see, e.g. [17, Section 19.6], or [18] for the example of $su(N)$). But the formal problem of writing a general recoupling coefficient of g [17, Section 19.11] in terms of Racah coefficients of g remains exactly the same as for $su(2)$, and thus the method of binary coupling trees holds here as well. Thus, all the following results in this paper hold for the expansion of a $3nj$-coefficient of an arbitrary semi-simple Lie algebra g in terms of Racah coefficients of g. One can even extend the applicability to non-compact Lie groups, or to infinite-dimensional representations. For example, the method also works for tensor products of positive discrete series representations of $su(1,1)$, since such a tensor product is completely decomposable into a direct sum of positive discrete series representations

(even without multiplicity labels). Even though we continue to use the terminology of angular momentum coupling in the following sections, we wish to emphasize that we have this extended coupling problem in mind.

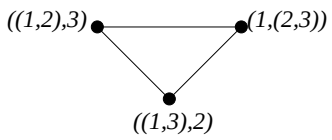
Remark 3 For the case of $su(2)$, the powerful method of Yutsis graphs was developed [1]. This graphical method is extremely useful to find (optimal) expansions for $3nj$-coefficients of $su(2)$ [6, 9], or to classify them. However, intrinsically this method uses various symmetry properties of Clebsch-Gordan or Racah coefficients that are valid for the case of $su(2)$ only. Thus it is no longer valid for the extended case described in the previous Remark and considered in the rest of this paper. For the extended case, only two properties are needed : (4) (or an equivalent one, see [18, (5.17)]) and (6), which is always valid (see eq. (19.49) of [17]).

3 The graph G_n

Let $n > 1$ be fixed, and consider the set of all binary coupling trees with $n + 1$ leaves. Since our only basic operation is rotation, we shall consider the *rotation graph* of binary coupling trees. This graph G_n has as vertex set the set of all binary coupling trees with $n + 1$ leaves, and there is an edge between two vertices if and only if the corresponding binary coupling trees are related through a single rotation. It follows that an optimal expression for a $3nj$-coefficient corresponds to finding a shortest path in G_n between the two binary coupling trees related to the bra- and ket-vector of the $3nj$-coefficient.

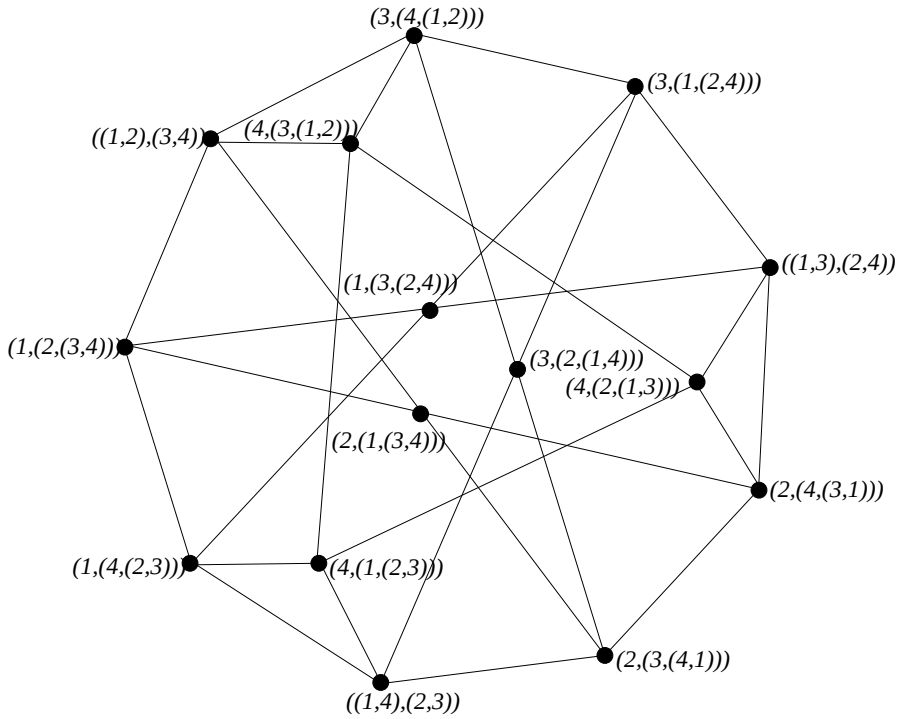
We shall now consider some examples, and deduce some general properties of G_n. For $n = 2$ this graph is simply a triangle. In Figure 6 we give G_2, and use the convention (7) to label the corresponding binary coupling trees. The next graph, G_3, has order 15. This

Figure 6: The graph G_2



graph is shown in Figure 7, using the bracket representation (7) for the binary coupling trees. The equivalence between optimal expressions for $3nj$-coefficients and shortest paths in G_n can be illustrated in Figure 7 for the classical $9j$-coefficient. For this coefficient, the corresponding binary coupling trees are $((1,2),(3,4))$ and $((1,3),(2,4))$. The shortest path in G_3 is of length 3, and thus this implies that this $9j$-coefficient can be written as a single sum expression over the product of three $6j$-coefficients (which is, at least for $su(2)$, a well-known fact). Note that in our terminology, we refer to other coefficients such as the ones corresponding to $\langle(1,(2,(3,4)))|(3,(2,(1,4)))\rangle$, $\langle(1,(2,(3,4)))|(2,(3,(1,4)))\rangle$ or $\langle(1,(2,(3,4)))|((1,2),(3,4))\rangle$ also as $9j$-coefficients, even though the first two reduce to the product of two $6j$-coefficients and the last one to a single $6j$-coefficient (as can be seen in Figure 7 from the corresponding distances in G_3).

Figure 7: The graph G_3



Let us now consider some general properties of the graph G_n. An arbitrary element of G_n, i.e. a binary coupling tree on $n+1$ leaves, has $n-1$ internal edges (i.e. edges containing no leaf). Two rotations can be performed with respect to each internal edge, thus every binary coupling tree is connected by an edge to $2(n-1)$ other binary coupling trees. In other words, G_n is a regular graph of degree $2(n-1)$. For example, G_3 is regular of degree 4.

To determine the number of binary coupling trees on $n+1$ leaves (or the order $|G_n|$ of G_n), consider first a binary coupling tree T on n leaves (with labels $1,\ldots,n$), and extend the root of T with an extra edge ending in the leaf 0 (as in Figure 4). This tree has $2n-1$ edges in total. Therefore, there are $2n-1$ different ways of adding an extra edge ending with leaf label $n+1$ to this tree, namely by attaching it to each consisting edge, see e.g. Figure 8. Thus we have $|G_n| = (2n-1)|G_{n-1}|$, and find (see also [10])

$$|G_n| = (2n-1)!! = (2n-1)(2n-3)\cdots 3\cdot 1. \tag{8}$$

This implies that the order of G_n grows exponentially. Table 1 gives the degree and the order of G_n for $n < 11$.

It is also easy to show that G_n is a connected graph, i.e. for any two binary coupling trees T_1 and T_2, there exists at least one path between T_1 and T_2 [10]. In a sense, this statement is equivalent to the fundamental theorem of recoupling theory [3, p. 455] that

Figure 8: Five ways of attaching an extra leaf label 4 to a given binary coupling tree on labels 1,2,3

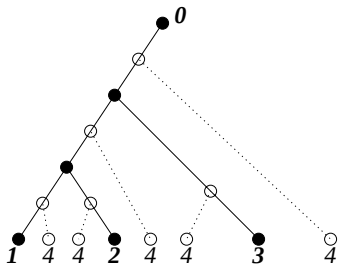


Table 1: Degree and order of G_n

n	2	3	4	5	6	7	8	9	10
$\deg(G_n)$	2	4	6	8	10	12	14	16	18
$\lvert G_n \rvert$	3	15	105	945	10395	135135	2027025	34459425	654729075

each generalized $3nj$-coefficient can be expressed in terms of sums over products of Racah coefficients ($6j$-coefficients).

Clearly, for $n \geq 11$ the order of G_n becomes too large to represent G_n in the RAM of a computer, which is necessary for the computation of shortest paths or of the diameter of G_n. In the following section we shall describe some of the results of our computations for $n < 11$. After that, we shall concentrate on the diameter of G_n, and give some approximations.

4 Distance in G_n

So far, we have reduced our problem to the following : given two binary coupling trees T_1 and T_2 from G_n, find a shortest path between T_1 and T_2, and in particular determine the length of this path (since this determines the number of Racah coefficients in the expansion). The length of a shortest path between T_1 and T_2 is known as the distance $d(T_1, T_2)$ between T_1 and T_2, since this induces a distance function or metric [12, 15].

Let us consider again the example $n = 3$, with G_3 given in Figure 7. Starting from the binary coupling tree $((1, 2), (3, 4))$, one finds that

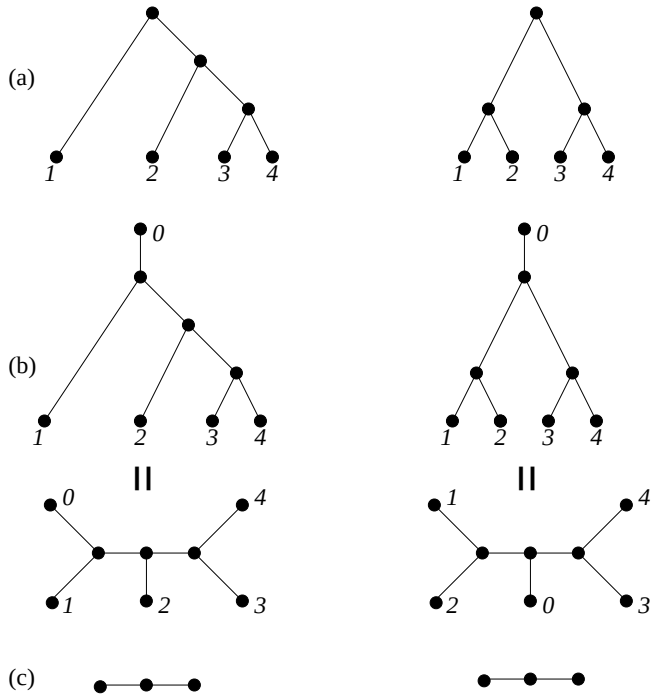
- there are 4 elements of G_3 at distance 1, namely
  $(4, (3, (1, 2)))$, $(3, (4, (1, 2)))$, $(2, (1, (3, 4)))$ and $(1, (2, (3, 4)))$;
- there are 8 vertices of G_3 at distance 2, namely
  $(3, (1, (2, 4)))$, $(3, (2, (1, 4)))$, $(4, (1, (2, 3)))$, $(1, (3, (2, 4)))$, $(1, (4, (2, 3)))$,
  $(4, (2, (1, 3)))$, $(2, (3, (4, 1)))$ and $(2, (4, (3, 1)))$;

- there are 2 vertices at distance 3, namely $((1,3),(2,4))$ and $((1,4),(2,3))$.

The sequence giving the number of elements at distance k ($k = 0, 1, \ldots$) from a given vertex T is called the *distance degree sequence* (DDS) for that vertex T. Thus, in G_3, the distance degree sequence of $((1,2),(3,4))$ is $(1,4,8,2)$. The two elements at maximum distance, in casu $((1,3),(2,4))$ and $((1,4),(2,3))$, give rise to $3nj$-coefficients with the maximum number of Racah coefficients in an optimal expression; in this case $\langle((1,2),(3,4))|((1,3),(2,4))\rangle$ and $\langle((1,2),(3,4))|((1,4),(2,3))\rangle$ give rise to genuine $9j$-coefficients that have as optimal expansion a single sum over products of three $6j$-coefficients.

It is not surprising that the distance degree sequence of $((1,3),(2,4))$ or $((1,4),(2,3))$ is also $(1,4,8,2)$. After all, a permutation of the leaf labels of the binary coupling trees in G_n does not change the structure of G_n. On the other hand, it is at first sight surprising that also the other vertices of G_3 of the form $(a,(b,(c,d)))$ have the same distance degree sequence as $((1,2),(3,4))$. Indeed, the binary coupling trees for $(1,(2,(3,4)))$ or $((1,2),(3,4))$ look different, see Figure 9(a). Thus G_3 has two different *types* of binary

Figure 9: Two different types of binary coupling trees with the same skeleton

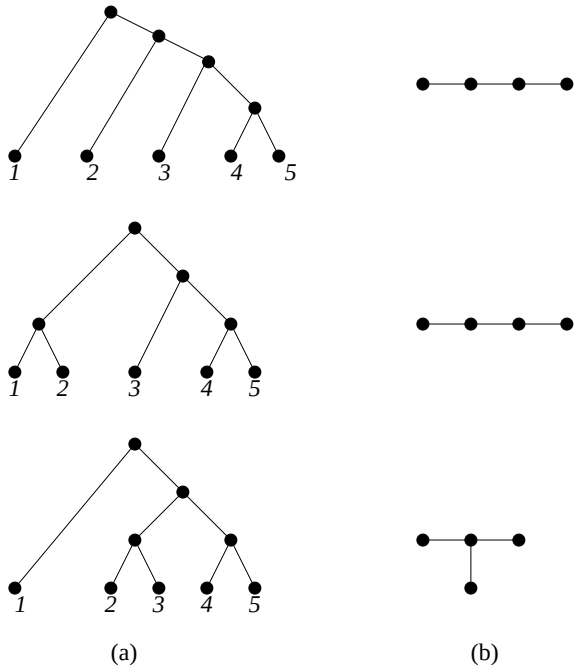


coupling trees, the first of the form $(a,(b,(c,d)))$ and the second of the form $((a,b),(c,d))$. This distinction changes however when one attaches an extra leaf label 0 to the root (Figure 9(b)). The full binary trees with labelled leaves (labels from $0, 1, \ldots, 4$) are now the same, upto a permutation of the labels. Since distance is governed by rotations over internal edges, it follows that the corresponding binary coupling trees will indeed have the same distance degree sequence. Note that there is another way of saying this, by introducing

the skeleton [13]. Generally, the *skeleton* of a binary coupling tree T of G_n with labelled leaves (labels from $0, 1, \ldots, n+1$) is obtained by deleting all leaves and corresponding edges from T. The result is thus an unlabelled tree (of maximum degree 3) with $n-1$ edges, see Figure 9(c) for $n = 3$.

Let us consider the next case $n = 4$ (corresponding to $12j$-coefficients). G_4 has 105 vertices. It is easy to verify that there are now three different types of binary coupling trees, given in Figure 10(a). When one considers their corresponding skeletons, there turn

Figure 10: Binary coupling trees for $n = 4$ and their skeletons



out to be 2 different ones, given in Figure 10(b). Thus, to have a complete picture of the distance in G_4, one should determine the distance degree sequences only for two vertices in G_4, e.g. for $T_1 = (1, (2, (3, (4, 5))))$ and for $T_2 = (1, ((2, 3), (4, 5)))$. We have found that $\mathrm{DDS}(T_1) = (1, 6, 20, 40, 34, 4)$ and $\mathrm{DDS}(T_2) = (1, 6, 24, 30, 44)$. Thus for T_1 there are 4 vertices at distance 5, whereas for T_2 there are no vertices at distance 5, but 44 vertices at distance 4. The vertices at maximum distance of $T_1 = (1, (2, (3, (4, 5))))$ are given by

$$(5, (2, (3, (1, 4)))), \qquad (5, (3, (2, (1, 4)))), \qquad (4, (2, (3, (1, 5)))), \qquad (4, (3, (2, (1, 5)))).$$

As a consequence, for the $12j$-coefficient corresponding to

$$\langle (1, (2, (3, (4, 5)))) | (5, (2, (3, (1, 4)))) \rangle,$$

the optimal expansion is a double sum over products of 5 Racah coefficients. This may seem to contradict the fact that the classical $12j$-coefficients for $su(2)$ have an expression in

terms of a single sum over products of only 4 Racah coefficients, which can be found using the technique of Yutsis. In this context, however, recall the observation in Remark 3. The fact that the optimal expansions for $12j$-coefficients of $su(2)$ can even further be reduced is related to symmetry properties that hold only for $su(2)$ coupling coefficients, and not for the general Lie algebra case considered here.

From the previous examples $n = 3$ and $n = 4$ it is clear that in order to determine distance properties for given G_n it is sufficient (a) to determine the number of skeletons; (b) to determine the distance degree sequence for each skeleton. Of course, this does not yet give the shortest path between any two given elements of G_n. But it does at least yield many other distance concepts (eccentricity, radius, center, periphery, ... [16]), and it also determines one of the most important distance characteristics of G_n, its diameter $d(G_n)$. The *diameter* of G_n is defined as follows :

$$d(G_n) = \max\{d(T_1, T_2) | T_1, T_2 \in G_n\}. \tag{9}$$

Thus it is the maximum value over all possible shortest path lengths of G_n; in other words : it is the length of the longest distance degree sequence.

To determine the number of skeletons t_n is an easy task, since for given n the skeletons are the unlabelled trees of maximum degree 3 (the so-called trivalent trees) with $n - 1$ edges (or, equivalently, with n vertices). This number is known, see e.g. sequence number A000672 of [19], or [20]. The first few values are given in Table 2. In Figure 11 we list the

Table 2: Number of trivalent trees with n vertices

n	2	3	4	5	6	7	8	9	10	11	12
t_n	1	1	2	2	4	6	11	18	37	66	135

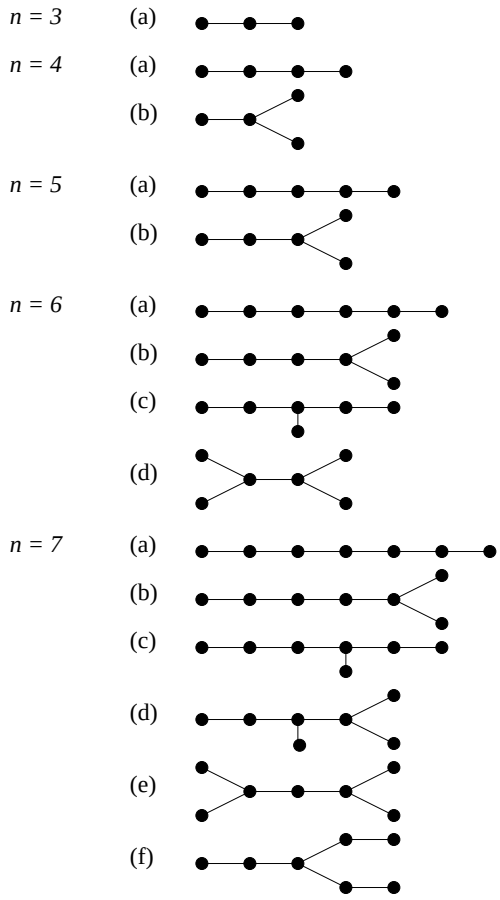
trivalent trees with n vertices (skeletons) upto $n = 7$ (see also [21]).

The purpose is then to calculate the distance degree sequence for a binary coupling tree corresponding to a skeleton. Our method for doing this is described in the following section.

5 The diameter $d(G_n)$

Let T be a given binary coupling tree of G_n. We wish to compute the distance degree sequence DDS(T). Let D_i be the set of elements of G_n at distance i from T. There is one element at distance 0, namely T itself; thus we have $D_0 = \{T\}$. Observe that it is easy to determine the neighbours of T in G_n : these are the elements of G_n at distance 1 from T, i.e. they are the binary coupling trees obtained from T by performing one rotation. Such rotations are easy to perform; as we have already observed in Section 3, every binary coupling tree has $2n - 2$ neighbours. Thus D_1 has $2n - 2$ elements. Next we compute the set of neighbours of the elements of D_1, and delete from this set the ones that were

Figure 11: Skeletons (trivalent trees) with n vertices for $n = 3, \ldots, 7$



already in D_0 or D_1, yielding D_2. Continuing this way, one can determine all D_i, and their orders give DDS(T).

Such a computation requires : (a) a simple data structure for a binary coupling tree, that allows an easy determination of its neighbours (i.e. perform rotations); (b) a proper way of keeping track of the elements of G_n that have already been encountered in D_0, D_1, ..., D_i while D_{i+1} is being determined. We have written a C program to calculate the distance degree sequence in G_n for any given binary coupling tree. Our program is inspired by some techniques of [22] (or, equivalently, [23]), and we shall not go into the details of this program here. Note that by the second requirement it is necessary that the elements of G_n can be stored in the RAM of the computer. Knowing the order of G_n, see (8) or Table 1, it is clear that on any present-day computer the computation cannot be performed beyond $n = 10$. We have calculated all distance degree sequences upto $n = 10$. Table 3 gives the results upto $n = 7$. For the complete results upto $n = 10$ see URL `http://allserv.rug.ac.be/~jvdjeugt/BCT`.

Table 3: Distance degree sequences in G_n. The skeleton types (a), (b), etc. refer to Figure 11.

	0	1	2	3	4	5	6	7	8	9	10	11	12
$n = 3$													
(a)	1	4	8	2									
$n = 4$													
(a)	1	6	20	40	34	4							
(b)	1	6	24	30	44								
$n = 5$													
(a)	1	8	36	110	244	328	198	20					
(b)	1	8	40	120	228	312	220	16					
$n = 6$													
(a)	1	10	56	220	670	1616	2810	3064	1708	236	4		
(b)	1	10	60	238	730	1604	2652	3060	1736	304			
(c)	1	10	60	250	732	1608	2598	2972	1880	276	8		
(d)	1	10	64	268	752	1648	2516	2672	2192	272			
$n = 7$													
(a)	1	12	80	378	1408	4344	11210	23028	34630	35050	20518	4320	156
(b)	1	12	84	404	1520	4688	11546	22420	33584	34748	20832	5104	192
(c)	1	12	84	416	1586	4796	11548	22188	32688	34588	21936	5100	192
(d)	1	12	88	454	1724	5096	11864	21808	30520	33200	24624	5712	32
(e)	1	12	88	430	1688	4912	11844	22352	31616	34224	22368	5248	352
(f)	1	12	84	428	1652	4920	11550	21752	32088	34372	22804	5320	152

From the calculation of the distance degree sequences, one easily deduces the diameter of G_n. Table 4 gives the diameter up to $n = 10$, which is as far as one can compute on a present-day computer, and which goes beyond previously calculated diameters [13, 22].

Table 4: Diameter of G_n

n	2	3	4	5	6	7	8	9	10
$d(G_n)$	1	3	5	7	10	12	15	18	21

The diameter is important as it gives an upper bound for the number of Racah coefficients appearing in an optimal expansion of a $3nj$-coefficient, see Section 2. Since it is so difficult to calculate the diameter explicitly beyond $n = 10$, one may wonder whether a proper approximation of $d(G_n)$ can be determined. This is indeed the case. In this paper we shall give a new lower and upper bound for $d(G_n)$. The details of the proofs are omitted here, and will be given in a separate comprehensive study of diameter properties of G_n [24].

Lemma 4 *The number of elements within distance i from any given binary coupling tree in G_n is less than or equal to $\binom{n+2i}{i} 4^i$.*

This can be shown using so-called short encodings [25]; for a detailed proof see [24].

Theorem 5 *The diameter of G_n satisfies*

$$d(G_n) > \frac{1}{4}\log(n!) > \frac{1}{4}n\log(n/e). \tag{10}$$

Herein (and in what follows), $\log = \log_2$ is the logarithm in basis 2, and e is the basis of the natural logarithm.
To prove the theorem, let $\delta = d(G_n)$, then by the above lemma and (8) we have that

$$\binom{n+2\delta}{\delta} 4^\delta \geq (2n-1)!! = \frac{(2n)!}{2^n n!}. \tag{11}$$

The lhs of (11) can be enlarged by

$$\frac{2^{n+2\delta}}{2n} > \binom{n+2\delta}{\delta},$$

which holds for all integers $n > 0$ and $\delta > 1$, see [24]. The rhs of (11) can be bounded using $\binom{2n}{n} \geq 2^{2n}/(2n)$. Thus (11) yields :

$$2^{4\delta} > n!,$$

from which the theorem follows.

An upper bound follows from

Theorem 6 *The diameter of G_n satisfies ($n > 1$)*

$$d(G_n) < n\lceil\log(n)\rceil - 2^{\lceil\log(n)\rceil} + 2(n - \lceil\log(n+1)\rceil) + 1 < n\lceil\log(n)\rceil + n - 2\lceil\log(n)\rceil + 1. \tag{12}$$

Herein, $\lceil x\rceil$ is the smallest integer larger than or equal to $x > 0$.

The proof of this theorem follows the lines indicated in [22]. Let a *spine* be a binary coupling tree of the form

$$(1, (2, (3, (\ldots (n, n+1)) \cdots); \tag{13}$$

see Figure 12 for its general shape. First, one determines an upper bound for the number of rotations needed to transform such a spine with arbitrary ordered labels into the spine with ordered labels (13). This upper bound is given by $n\lceil\log(n)\rceil - 2^{\lceil\log(n)\rceil} + 1$, see [24]. Next, it is not difficult to see that there are at most $n - \lceil\log(n+1)\rceil$ rotations needed to transform an arbitrary binary coupling tree into a spine. Thus to transform two binary coupling trees T_1 and T_2 into each other, first transform both T_1 and T_2 into a spine, and then transform one spine into the second one. This leads to (12).

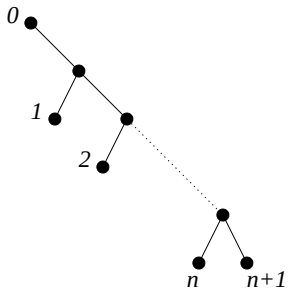
Together, the above two theorems imply that the diameter of G_n is of order $n\log(n)$, i.e.

$$d(G_n) = \Theta(n\log(n)).$$

Note that a weaker upper limit has been given earlier in [11], and weaker upper and lower limits were determined in [22].

The for us important consequence is :

Figure 12: Binary coupling tree which is a spine



Corollary 7 *Consider an optimal expansion of a $3nj$-coefficient in terms of Racah coefficients. Then the number s_r of Racah coefficients appearing in a term of the expansion is of order $n\log(n)$; more explicitly, it is bounded by*

$$\frac{1}{4}n\log(n/e) < s_r < n\lceil\log(n)\rceil + n - 2\lceil\log(n)\rceil + 1.$$

6 Equivalent and related problems

In this paper, we have reduced the problem of finding an optimal expansion of a $3nj$-coefficient to the graph theoretical problem of finding a shortest path between binary coupling trees in G_n. This last problem has been encountered before in different contexts. One of the first papers where this problem is stated, with applications in mind, is [12]. In that context, our binary coupling trees are called *dendrograms*, and the purpose is the computation of a similarity measure or distance (coinciding with our distance $d(T_1, T_2)$) between dendrograms.

Dendrograms can be defined as rooted trees where each of the terminal vertices (leaves) represent an object and where the root vertex represents the entire object-set [26]. According to [26], binary dendrograms can have labelled or unlabelled leaves, and can be ranked or non-ranked (ordered or unordered, in our terminology). An enumeration of four types of binary dendrograms was given in [26]. The ones of interest to us are the non-ranked (unordered) dendrograms with labelled leaves, since they coincide with our binary coupling trees. These are also the dendrograms appearing in the paper of Waterman and Smith [12], and are of importance in mathematical biology. The first area of application is taxonomy. Here, various hierarchical cluster methods are used to construct taxonomic dendrograms. A cluster algorithm can result in dendrograms with differing initial ordering, thus a method of measuring the degree of similarity between dendrograms is of importance [12]. The similarity is computed by means of the distance between dendrograms, coinciding with distance in G_n. A second area of biological research involving dendrograms is in morphogenesis and/or cell differentiation studies, where the development of systems is represented by a tree (decision tree; equivalent to our binary coupling tree). Here again, a tree similarity measure is of importance [12], and is given by distance

in G_n.

Some ways of computing or estimating the similarity of dendrograms have been considered in the literature. Waterman and Smith [12], who introduced similarity measure, also use the term *nearest neighbour interchange metric* to refer to the distance d in G_n. Realizing that d is difficult to compute in general, they introduced another measure c which afterwards turned out to violate the triangle inequality [13, 14, 15]. Brown and Day showed that both d and c are difficult to compute [27]. They designed an approximation to d, and analysed the algorithm for the computation of this approximation. Another approximation was considered in [28], requiring only $O(n)$ time to compute. That in general d is difficult to calculate was indicated by Křivánek [29], who showed that computing d is an NP-complete problem. It should be mentioned that Li *et al* [22] found a mistake in Křivánek's proof, so the question of NP-completeness remains open. Still, at present there is no simple algorithm to compute the distance $d(T_1, T_2)$ between given binary coupling trees in G_n. Therefore, we plan to reconsider the methods of [27] or [28] in order to develop programs that produce close approximations for the distance $d(T_1, T_2)$ (and thus programs that produce expansions of $3nj$-coefficients which are nearly optimal).

The problem of computing the distance d in G_n was also considered by computer scientists [11, 23]. Often, however, computer scientists are more interested in a closely related problem : calculating the rotation distance d in the graph H_n consisting of binary search trees (rooted ordered binary trees, with unlabelled leaves) with n internal vertices. In [11], it was already shown that the diameter $d(H_n) \leq 2n - 2$ (so a linear upper bound, instead of a $n \log(n)$ upper bound for $d(G_n)$). A detailed study [30] revealed that $d(H_n) \leq 2n - 6$, later confirmed by more elementary methods [31, 32]. Although the diameter is easier to estimate in this case, computing the actual distance d in H_n once again turns out to be difficult [33].

One of the properties of the distance function d for G_n, which has received attention in the mathematical biology literature, is that of non-decomposability. This peculiar property is also of interest in our context of $3nj$-coefficients. Let $T, T' \in G_n$ be two binary coupling trees with leaves labelled $1, 2, \ldots, n+1$. Suppose that, in the notation of (7), $T = (t_1, t_2)$ and $T' = (t_1', t_2')$, where $t_1, t_1' \in G_k$ are binary coupling trees with leaves labelled by $1, 2, \ldots, k+1$, and $t_2, t_2' \in G_{n-k-1}$ are binary coupling trees with leaves labelled by $k+2, \ldots, n+1$. The distance d is said to satisfy the decomposition property if for all such T and T' :

$$d(T, T') = d(t_1, t_1') + d(t_2, t_2'),$$

where (by abuse of notation) the first d in the rhs is the distance function in G_k, and the second d in G_{n-k-1}. Otherwise, d is non-decomposable. It was indicated in [14] and shown in [22] that the distance function d for G_n does not satisfy the decomposition property. This implies that for general $3nj$-coefficients, a so-called cut on two lines [1, Figure 14.2][6] in the corresponding Yutsis graph does not necessarily yield the most optimal expansion in Racah coefficients (although it will do so for $n \leq 7$).

7 Conclusion

We have considered the problem of finding an optimal expansion of a general $3nj$-coefficient (for finite-dimensional representations of a semi-simple Lie algebra g) in terms of Racah coefficients of g. This problem was reduced to the shortest path problem in the graph G_n, of which the vertices are given by binary coupling trees on $n+1$ leaves and the edges correspond to rotations. Finding shortest paths in the rotation graph of binary coupling trees turns out to be a difficult problem. Upto $n = 10$, the distance degree sequences of G_n have been calculated explicitly, yielding many distance properties of G_n and in particular implying the diameter $d(G_n)$. This diameter is an upper bound for the number of Racah coefficients appearing in an optimal expression for a $3nj$-coefficient. We have shown that $d(G_n)$ grows like $n\log(n)$ by giving upper and lower bounds for it. Finally, we have shown that our shortest path problem has already appeared in other contexts, such as mathematical biology and computer science, where it has important applications. Methods to find approximations of the shortest path, developed in these areas, can be useful in our context of $3nj$-coefficients and will be studied in the future.

References

[1] A.P. Yutsis, I.B. Levinson and V.V. Vanagas, Mathematical Apparatus of the Theory of Angular Momentum (Israel Program for Scientific Translation, Jerusalem, 1962).

[2] E. El Baz and B. Castel, Graphical Methods of Spin Algebra in Atomic, Nuclear, and Particle Physics (Marcel Dekker, New York, 1972).

[3] L.C. Biedenharn and J.D. Louck, The Racah-Wigner Algebra in Quantum Theory, Encyclopedia of Mathematics and its Applications, Vol. 9, ed. G.-C. Rota (Addison-Wesley, Reading, MA, 1981).

[4] A.R. Edmonds, Angular Momentum in Quantum Physics (Princeton Univ. Press, Princeton, 1957).

[5] P.G. Burke, A program to calculate a general recoupling coefficient, Comput. Phys. Commun. 1 (1970) 241-250.

[6] A. Bar-Shalom and M. Klapisch, NJGRAF – An efficient program for calculation of general recoupling coefficients by graphical analysis, compatible with NJSYM. Comput. Phys. Commun. 50 (1988) 375-393.

[7] V. Fack, S.N. Pitre and J. Van der Jeugt, New efficient programs to calculate general recoupling coefficients. Part I: Generation of a summation formula, Comput. Phys. Commun. 83 (1994) 275-292.

[8] V. Fack, S.N. Pitre and J. Van der Jeugt, New efficient programs to calculate general recoupling coefficients. Part II: Evaluation of a summation formula, Comput. Phys. Commun. 86 (1995) 105-122.

[9] V. Fack, S.N. Pitre and J. Van der Jeugt, Calculation of general recoupling coefficients using graphical methods, Comput. Phys. Commun. 101 (1997) 155-170.

[10] D.F. Robinson, Comparison of labeled trees with valency three, J. Combinatorial Theory Ser. B 11 (1971) 105-119.

[11] K. Culik II and D. Wood, A note on some tree similarity measures, Inform. Process. Lett. 15 (1982) 39-42.

[12] M.S. Waterman and T.F. Smith, On the similarity of dendrograms, J. Theor. Biol. 73 (1978) 789-800.

[13] J.P. Jarvis, J.K. Luedeman and D.R. Shier, Comments on computing the similarity of binary trees, J. Theor. Biol. 100 (1983) 427-433.

[14] W.H.E. Day, Properties of the nearest neighbor interchange metric for trees of small size, J. Theor. Biol. 101 (1983) 275-288.

[15] R.P. Boland, E.K. Brown and W.H.E. Day, Approximating minimum-length-sequence metrics: a cautionary note, Math. Soc. Sci. 4 (1983) 261-270.

[16] F. Buckley and F. Harary, Distance in Graphs (Addison-Wesley, Reading MA, 1990).

[17] B.G. Wybourne, Classical Groups for Physicists (John Wiley & Sons, New York, 1974).

[18] J.-Q. Chen, P.-N. Wang, Z.-M. Lü and X.-B. Wu, Tables of the Clebsch-Gordan, Racah and subduction coefficients of SU(n) groups (World Scientific, Singapore, 1987).

[19] N.J.A. Sloane, On-Line Encyclopedia of Integer Sequences
(URL : `http://www.research.att.com/~njas/sequences/`).

[20] A. Cayley, On the analytical forms called trees, with application to the theory of chemical combinations, Reports British Assoc. Advance. Sci. 45 (1875), 257-305.

[21] F. Harary, Graph Theory (Addison-Wesley, Reading MA, 1969).

[22] M. Li, J. Tromp and L. Zhang, On the nearest neighbour interchange distance between evolutionary trees, J. Theor. Biol. 182 (1996) 463-467.

[23] M. Li, J. Tromp and L. Zhang, Some notes on the nearest neighbour interchange distance, Lect. Notes Computer Science 1090 (1996), 343-351.

[24] V. Fack, S. Lievens and J. Van der Jeugt, On the diameter of the rotation graph of binary coupling trees, University of Ghent preprint (1999).

[25] D.D. Sleator, R.E. Tarjan and W.P. Thurston, Short encodings of evolving structures, SIAM J. Disc. Math. 5 (1992) 428-450.

[26] F. Murtagh, Counting dendrograms : a survey, Discrete Appl. Math. 7 (1984) 191-199.

[27] E.K. Brown and W.H.E. Day, A computationally efficient approximating to the nearest neighbor interchange metric, J. Class. 1 (1984) 93-124.

[28] W.H.E. Day, Optimal algorithms for comparing trees with labeled leaves, J. Class. 2 (1985) 7-28.

[29] M. Křivánek, Computing the nearest neighbor interchange metric for unlabeled binary trees is NP-complete, J. Class. 3 (1986) 55-60.

[30] D.D. Sleator, R.E. Tarjan and W.P. Thurston, Rotation distance, triangulations and hyperbolic geometry, J. Amer. Math. Soc. 1 (1988) 647-681.

[31] E. Mäkinen, On the the rotation distance of binary trees, Inform. Process. Lett. 26 (1987) 271-272.

[32] F. Luccio and L. Pagli, On the upper bound on the rotation distance of binary trees, Inform. Process. Lett. 31 (1989) 57-60.

[33] R.O. Rogers and R.D. Dutton, On distance in the rotation graph of binary trees, Congr. Numer. 120 (1996) 103-113.

Counting Free Binary Trees Admitting a Given Height

Frank Harary
Computer Science Department
New Mexico State University
Las Cruces, NM 88003, USA

Edgar M. Palmer
Mathematics Department
Michigan State University
East Lansing, MI 48823, USA

Robert W. Robinson
Computer Science Department
University of Georgia
Athens, GA 30602, USA

Dedicated to the memory of R. C. Bose, the combinatorial and statistical pioneer.

Suggested running head: Counting Free Binary Trees

Author to whom proofs should be addressed:

Robert W. Robinson
Computer Science Department
415 GSRC
University of Georgia
Athens, GA 30602

Abstract

Recursive equations are derived for the exact number t_h of nonisomorphic free trees which have some rooting as a binary tree of height *h*. Numerical results are calculated using these formulae.

1. Introduction

A **binary tree** *T* can be defined as a rooted tree in which each node has degree at most 3, except that the root has degree at most 2. The **height** of *T* is the maximum distance from the root node to an endnode. Binary trees are much used in theoretical computer science, with height often being a key parameter directly related to the efficiency of associated algorithms. A **free binary tree** *F* is an unrooted tree which has a node *u* (not necessarily unique) such that *F* is a binary tree when rooted at *u*. Our purpose is to derive formulae for the number of unlabeled free binary trees which have a rooting that produces a binary tree of height *h*; we say that such a tree **admits height** *h*. In general our terminology follows [3]. Unlabeled counting does not distinguish between versions of a tree which differ only in the assignment of labels to the nodes.

A **3-tree** has maximum degree at most 3. It is convenient for our purpose of counting free binary trees by admissible height to consider 3-trees first. Obviously every free binary tree is a 3-tree, and conversely since any node of degree 1 or 2 could serve as the root. Figure 1 shows a free binary tree *F* which has four distinct binary rootings. Rooting *F* at node 5 or 6 gives one binary tree of height 5; at 7 gives height 4; at 3 gives height 3; finally, rooting *F* at 8 or 9 gives a second binary tree of height 5. Thus *F* admits height 3, 4, and 5. In the total of free binary trees of order a admitting height 5, for

instance, *F* will be counted just once.

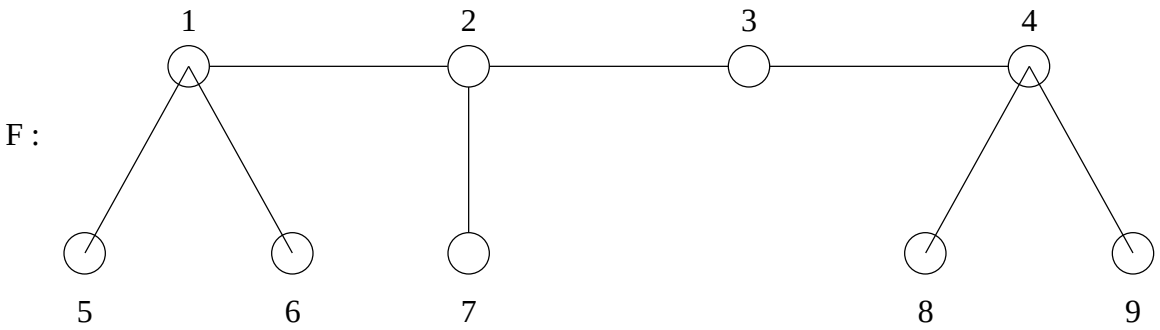


FIGURE 1. A free binary tree which has four binary rootings

Both rooted and unrooted 3-trees have been counted by Cayley and Otter; see [4] for a modern exposition.

2. Planted 3-trees of given height

In a **planted tree**, the root is an endnode. Let p_h be the number of planted 3-trees of height *h*, and let q_h be the number of height less than *h*, including for convenience the empty one with no nodes and no edges.

Then $p_1 = q_1 = 1$, while for all $h \geq 1$,

$$q_{h+1} = q_h + p_h \tag{1}$$

$$p_{h+1} = \binom{1+p_h}{2} + p_h q_h \tag{2}$$

Note that the numbers p_h were known to Etherington [2]; they are sequence number 718 in Sloane's book, [6].

To justify (2), we observe that a planted tree of height $h+1$ has two major subtrees, one of height *h* and the other of height *h* or less. For both to have height *h*, there are $\binom{1+p_h}{2}$ possibilities since we need to select two trees (which may be isomorphic) from among the p_h of height *h*, and their order is immaterial. For the case when one major subtree has height *h* and the other less, the possibilities are enumerated by $p_h q_h$ since the two branches cannot be confused with one another. The empty case admitted by $q_1 = 1$ corresponds to the possibility that the node adjacent to the root has degree 2, so that there is really only one major subtree.

In order to allow for the analysis of free 3-trees, it will be necessary to determine the number $d_{h,i}$ of planted 3-trees of height *h* which have no nodes of degree 1 or 2 at level *i* (distance *i* from the root). Of course all 3-trees of height *h* have one or more nodes of

degree 1 at level *h* and no nodes at any level greater than *h*, so $d_{h,h} = 0$ and $d_{h,i} = p_h$ for all $i > h$. In fact, our interest will be in the number $(p_h - d_{h,i})$ of 3-trees of height *h* which do have a node of degree 1 or 2 at level *i*, for $1 \leq i < h$. However the defining equations are more direct when written in terms of $d_{h,i}$. It will also be convenient to identify the quantity

$$e_{h,i} = 1 + \sum_{1 \leq j < h} d_{j,i} \quad , \tag{3}$$

which bears the same relation to $d_{h,i}$ that q_h bears to p_h. One can then write the recursively defining equations as

$$d_{h+1,i+1} = \binom{1+d_{h,i}}{2} + d_{h,i}\, e_{h,i} \tag{4}$$

$$e_{h+1,i} = e_{h,i} + d_{h,i} \tag{5}$$

for $h > i \geq 1$. These parallel precisely equations (1) and (2). For boundary conditions we have

$$\begin{aligned} d_{h+1,1} &= p_{h+1} - p_h \quad , \\ e_{h+1,1} &= p_h \end{aligned} \tag{6}$$

for all $h \geq 1$. This is because if a planted tree of height $h + 1$ has a node of degree 1 or 2 adjacent to the root, that node must have degree 2 since $h \geq 1$. By suppressing this node, one obtains a tree of height h in a 1-1 fashion, so that

$$p_{n+1} - d_{h+1,1} = p_h \quad .$$

Now

$$e_{h+1,1} = 1 + \sum_{1 \le k \le h} d_{k,1} = 1 + d_{1,1} + \sum_{2 \le k \le h} (p_k - p_{k-1})$$

$$= 1 + d_{1,1} + p_h - p_1$$

$$= p_h$$

since $p_1 = 1$ and $d_{1,1} = 0$.

3. Free 3-trees by admissible height

It does not appear possible to apply the principle of Otter's dissimilarity characteristic [4, p.56] to obtain the number t_h of free 3-trees which have some rooting as a binary tree of height *h*. Instead, we will make use of the fact that every tree has a unique center consisting of a single node or two adjacent nodes. The possibilities for binary rootings of various heights are enumerated separately for these two cases. This approach was used by Cayley [1] when he first counted trees.

Case 1 The center is a single node.

Assuming a nontrivial tree *T*, the diameter is $2h$ for some $h \ge 1$. Then some two branches at the center must have height *h* and the third branch (if there is one) must have height at most *h*. The number of ways to choose these branches is

$$a_h = \binom{2+p_h}{3} + \binom{1+p_h}{2} q_h \quad . \tag{7}$$

The first term counts the number of ways to choose all three branches to have height *h*, and the second gives the number with two branches of height *h* and either no third branch or else a third branch having some height *k*, $1 \le k < h$.

Suppose now that one of the branches at the center of *T* has a node of degree 1 or 2 at level *i*, $i \geq 1$. Then *T* would have height $h + i$ if rooted at such a node, since any path of maximum length must pass through the center. The number of ways that *T* could fail to contain such a node is exactly

$$\binom{2+d_{h,i}}{3} + \binom{1+d_{h,i}}{2} e_{h,i} \quad . \tag{8}$$

This is just as for (7) except that every branch must fail to have a node of degree 1 or 2 at level *i*. Subtracting (8) from (7) will then give the number of 3-trees of diameter $2h$ which have a binary rooting of height $h + i$, $1 \leq i \leq h$.

There remains the possibility of rooting at the central node. The center has degree at most 2 exactly when there are just two branches. In that case the tree has height *h* when rooted at the center, so we have exactly

$$\binom{1+p_h}{2} \tag{9}$$

3-trees of diameter $2h$ which have a binary rooting of height *h*.

Case 2 The center consists of two adjacent nodes.

The diameter is $2h - 1$ for some $h \geq 1$, and we can obtain any such tree in a unique fashion by joining two trees of height *h* at the root, then smoothing out the root node. We refer to these two trees as the branches at the bicenter. Of course their order is unimportant, and they may be isomorphic. Hence there are exactly

$$b_h = \binom{1+p_h}{2} \tag{10}$$

3-trees of diameter $2h - 1$.

In this case a node of level *i* on one of the branches at the bicenter gives a rooting of height $h + i - 1$. The number of 3-trees of diameter $2h - 1$ having no node of level *i* of degree 1 or 2 on either branch at the center is just

$$\binom{1+d_{h,i}}{2} . \tag{11}$$

Subtracting (11) from (10) then gives the number of 3-trees of diameter $2h - 1$ which have a binary rooting of height $h + i - 1$.

The total number t_h of free 3-trees with a binary rooting is just the sum of the numbers obtained in Cases 1 and 2, for the appropriate values of *h* and *i*. More explicitly, for $h \geq 1$ we have

$$t_h = \binom{1+p_h}{2} + \sum_{i=1}^{\lfloor h/2 \rfloor} \left\{ a_{h-i} - \binom{2+d_{h-i,i}}{3} - \binom{1+d_{h-i,i}}{2} e_{h-i,i} \right\}$$
$$+ \sum_{i=1}^{\lfloor (h+1)/2 \rfloor} \left\{ b_{h-i+1} - \binom{1+d_{h-i+1,i}}{2} \right\} . \tag{12}$$

4. Numerical results.

Table I lists p_h for $h \leq 11$. Equations (1) and (2) enable us to calculate the sequence $p_1, p_2, \ldots, p_n$ in *O*(n) time.

Table II gives the values of t_h for $h \leq 10$. Note that $p_{h+1} \geq t_h$. This is because any tree with a binary rooting of height *h* corresponds to a planted 3-tree of height $h + 1$. This correspondence is obtained by adding a new root of degree one adjacent to the original root node. In general there are trees with more than one binary rooting of

height *h*, so that equality does not hold. (An example is provided by the tree *F* of Figure 1, which has two different binary rootings of height 5.) However, it is apparent that $p_{h+1} - t_h$ is small compared to t_h as *h* increases, so that multiple rootings of the same height are relatively rare.

TABLE I *The number of planted 3-trees by height*

h	p_h
1	1
2	2
3	7
4	56
5	2212
6	2595782
7	3374959180831
8	569518350448923906748 4387
9	162175575749223863014205312770713651031687342842 82
10	131504586847961235687181874578063117114329409897598970946516793776 220805297959867258692249572750581
11	864672818102648960261040653715831867092837278673702464113037906939 422113848975628994429633085310830824182159666913797168694932947833 666153033443005805197333617729392377202761080179484074798817 7012

In general, the method employed enables one to compute the values $t_1,t_2,...,t_n$ with $O(n^2)$ integer arithmetic operations and storage of *O(n)* integers. This analysis of complexity takes no account of the rapid increase in the size of the numbers involved. It is clear that log t_n = $O(n^2)$, so this has a significant effect.

First, (1) and (2) are applied to compute p_h and q_h for $h \le n$. Simultaneously (7) and (10) are applied to determine a_h and b_h for $h \le n$, and these values are stored. At the same time, (5) and (6) are used to find $d_{h,1}$ and $e_{h,1}$ for $h \le n$, and these too are stored. The calculation proceeds by induction on *i*, $i = 1, \cdots, \lfloor (n+1)/2 \rfloor$. As the numbers $d_{h,i}$ and $e_{h,i}$ are computed and stored, their contributions to $t_1,...,t_n$ as given in (12) are accummulated. First $d_{h,i+1}$ for $h \le n$ is given by (3), and then $e_{h,i+1}$ for

$h \leq n$ is determined from (4).

By computing the values of $d_{h,i}$ in descending order of *h*, one can overwrite the $d_{h,i}$ array by the $d_{h,i+1}$. Using (4) one calculates the $e_{h,i+1}$ in ascending order, but the $e_{h,i}$ are not needed and so can be overwritten too. In order to avoid separately storing the values $e_{i+1,i}$ needed to start with (4), note that for $i \geq 2$ we have

$$e_{i+1,i} = e_{i,i-1} + p_{i-1},$$

and

$$p_{i-1} = d_{i,i-1}.$$

Now $d_{i,i-1}$ should still be available due to the fact that $d_{h,i}$ only needed computing for $h > i$. This is because $d_{i,i} = 0$ (so can be handled separately) and $d_{h,i}$ for $h < i$ is not called for in (12). For the same reasons $e_{i,i-1}$ should also still be available. Finally, the trees counted by $d_{i,i-1}$ can be obtained in a 1-1 fashion from those of height $i - 1$ by joining two new endnodes to each old endnode. Each new tree then has height *i* but has only nodes of degree 3 at level $i - 1$. Hence $p_{i-1} = d_{i,i-1}$ as claimed above.

TABLE II *The number of free binary trees by height*

h	t_h
1	2
2	7
3	52
4	2133
5	2590407
6	3374951541062
7	569518350447911664037650 9
8	16217557574922386301420514191523784895639577710480
9	13150458684796123568718187457806311711432940989755031827379203302434038821923508109665802351707695 0
10	86467281810264896026104065371583186709283727867370246411303790693942211384897562899442963308531079137280610527854309101413563826111133256812507183116291634662221528525970675542565225209199730909 55

References

1. A. CAYLEY, On the analytical forms called trees, with applications to the theory of chemical combinations, *Rep. Brit. Assoc. Advance. Sci.* **45** (1875), 257-305 = *Math. Papers*, Vol. 9, 427-460.
2. I. M. H. ETHERINGTON, On non-associative combinations, *Proc. Roy. Soc. Edinburgh* **59** (1938/39), 153-162.
3. F. HARARY, "Graph Theory," Addison-Wesley, Reading, 1969.
4. F. HARARY and E. M. PALMER, "Graphical Enumeration," Academic, New York, 1973.
5. R. OTTER, The number of trees, *Ann. of Math.* **49** (1948), 583-599.
6. N. J. A. SLOANE, "A Handbook of Integer Sequences" Academic, New York, 1973.

Admissible partitions and the square of the Vandermonde determinant

Brian G Wybourne
Instytut Fizyki,Uniwersytet Mikołaja Kopernika, 87-100 Toruń, POLAND

Abstract. The expansion of the second power of the Vandermonde determinant as a finite sum of Schur functions is considered.

1. Introduction

Laughlin[1] has described the fractional quantum Hall effect in terms of a wavefunction

$$\Psi^m_{Laughlin}(z_1,\ldots,z_N)=\prod_{i<j}^{N}(z_i-z_j)^{2m+1}\exp\left(-\frac{1}{2}\sum_{i=1}^{N}|z_i|^2\right) \tag{1}$$

The Vandermonde alternating function in N variables is defined as

$$V(z_1,\ldots,z_N)=\prod_{i<j}^{N}(z_i-z_j) \tag{2}$$

$$\frac{\Psi_{Laughlin}}{V}=V^{2m}=\sum_{\lambda\vdash n}c^{\lambda}s_{\lambda} \tag{3}$$

where $n=mN(N-1)$ and the s_λ are Schur functions. The coefficients c_λ are signed integers.

Dunne[2] and Di Francesco *et al*[3] have discussed properties of the expansions while Scharf *et al*[4] have given specific algorithms for computing the expansions for $m=1$ with N from 2 to 9. The author has extended these results to $N=10$ leading to a number of new conjectures.

1.1. Expansion of the Laughlin wavefunction

Henceforth we consider the case where $m=1$. The partitions, (λ), indexing the Schur functions are of weight $N(N-1)$. For a given N the partitions are bounded by a highest partition $(2N-2,2N-4,\ldots,0)$ and a lowest partition $((N-1)^{N-1})$ with the partitions being of length N and $N-1$.

Let

$$n_k=\sum_{i=0}^{k}\lambda_{N-i}-k(k+1)k=0,1,\ldots,N-1 \tag{4}$$

Di Francesco *et al*[3] define *admissible partitions* as satisfying Eq(4) with *all* $n_k\geq 0$. They computed the number of admissible partitions A_N for $N\leq 29$ and conjectured that A_N was the number of distinct partitions arising in the expansion, Eq(3), *provided none of the coefficients vanished*.

The conjecture has been shown[4] to fail for $N \geq 8$. We find the number of admissible partitions associated with vanishing coefficients as

$$(N=8) \quad 8, \ (N=9) \quad 66, \ (N=10) \quad 389$$

The coefficients of s_λ and s_{λ_r} are equal if[2]

$$(\lambda_r) = (2(N-1) - \lambda_N, \ldots, 2(N-1) - \lambda_1) \tag{5}$$

We list the 8 partitions for $N = 8$ as reverse pairs

$$\begin{array}{ll} \{13\ 11\ 985^2 41\} & \{13\ 10\ 9^2 6531\} \quad (Q1) \\ \{13\ 11\ 9854^2 2\} & \{13\ 10\ 987531\} \quad (Q2) \\ \{13\ 11\ 976541\} & \{12\ 10^2 96531\} \quad (Q3) \\ \{12\ 11\ 97^2 4^2 1\} & \{12\ 10^2 7^2 532\} \quad (Q4) \end{array}$$

1.2. The $q-$discriminant

Let $q\mathbf{x} = (qx_1, qx_2, \ldots, qx_N)$ and the $q-$discriminant of $\mathbf{x}$ be

$$D_N(q;\mathbf{x}) = \prod_{1 \leq i \neq j \leq N} (x_i - qx_j) \tag{6}$$

and

$$R_N(q;\mathbf{x}) = \prod_{1 \leq i \neq j \leq N} (x_i - qx_j)(qx_i - x_j) \qquad = \sum_\lambda c^\lambda(q) s_\lambda(\mathbf{x}) \tag{7}$$

So that

$$V_N^2(\mathbf{x}) = \prod_{1 \leq i < j \leq N} (x_i - x_j)^2 = R_N(1;\mathbf{x}) \tag{8}$$

Introduce $q-$polynomials such that

$$R_N(q;\mathbf{x}) = \sum_\lambda c^\lambda(q) s_\lambda(\mathbf{x}) \tag{9}$$

$$\begin{aligned} R_N(q;\mathbf{x}) = & \frac{(-1)^{N(N-1)/2}}{(1-q)^N} \sum_{\nu \subseteq (N-1)^N} ((-q)^{|\nu|)} + (-q)^{N^2 - |\nu|}) \\ & \times s_{(N-1)^N/\nu}(\mathbf{x}) s_{\nu'}(\mathbf{x}) \end{aligned}$$

Such expansions have been evaluated as polynomials in q for all admissible partitions for $N = 2...6$ with many examples for $N = 7, 8, 9$.

N=2	[1]	q	$\{2\}$
	[-3]	$-(q^2+q+1)$	$\{1^2\}$
N=3	[1]	q^3	$\{42\}$
	[-3]	$-q^2(q^2+q+1)$	$\{41^2\} + \{3^2\}$
	[6]	$+q(q^2+q+1)(q^2+1)$	$\{321\}$
	[-15]	$-(q^2+q+1)(q^4+q^2+q+1)$	$\{2^3\}$
N=4	[1]	q^6	$\{642\}$
	[-3]	$-q^5(q^2+q+1)$	$\{641^2\} + \{63^2\} + \{5^2 2\}$
	[6]	$+q^4(q^2+q+1)(q^2+1)$	$\{6321\} + \{543\}$
	..	.....	....

The $q-$polynomials for the four pairs of partitions designated earlier as $Q(1)...Q(4)$ are

$$
\begin{aligned}
&Q(1) - q^{17}(q^2-q+1)^2(q^2+1)^2(q^2+q+1)^5(1-q)^4\\
&Q(2) + q^{16}(q^2-q+1)^2(q^2+1)(q^2+q+1)^6(1-q)^4\\
&Q(3) + q^{16}(q^2-q+1)^2(q^2+1)^3(q^2+q+1)^5(1-q)^4\\
&Q(4) + q^{14}(q^2-q+1)^2(q^2+q+1)^5(1-q)^4\\
&\qquad \times (q^{10}+q^9+3q^8+4q^6+q^5+4q^4+3q^2+q+1)
\end{aligned}
$$

Note the factor $(q-1)^4$ which vanishes for $q=1$.

1.3. A conjecture

The following conjecture has been verified to hold for $N \leq 10$

If a $q-$polynomial is of the form $(-1)^\phi q^p Q(q)$ then under $N \to N+1$

$$\phi \to \phi,\ p \to p+N,\ Q(q) \to Q(q),\ \{\lambda\} \to \{2N-2, \lambda\}$$

Define

$$QS(N) = \sum_\lambda c_\lambda(q)$$

then

$$QS(N) = \prod_{x=0}^{[N/2]} (-3x+1) \prod_{x=0}^{[(N-1)/2]} (6x+1)$$

Di Francesco *etal*[3] establish the remarkable result that the sum of the squares of the coefficients of the second power of the Vandermonde with $q=1$ is

$$\frac{(3N)!}{N!(3!)^N}$$

What is the corresponding result for the $q-$polynomials? For $N=4$ one finds

$$
\begin{aligned}
&q^{24} + 6q^{23} + 22q^{22} + 58q^{21} + 128q^{20} + 242q^{19}\\
&+ 418q^{18} + 646q^{17} + 929q^{16} + 1210q^{15} + 1490q^{14}\\
&+ 1670q^{13} + 1760q^{12} + 1670q^{11} + 1490q^{10} + 1210q^9\\
&+ 646q^8 + 418q^6 + 242q^5 + 128q^4 + 58q^3 + 22q^2 + 6q + 1
\end{aligned}
$$

Note the polynomial is symmetrical and unimodal! Can the general result be found?

Acknowledgments

This work has benefited from interaction with R C King and J-Y Thibon and is supported in part by the Polish KBN Grant 5P03B 5721.

References

[1] Laughlin R B 1983 *Phys. Rev. Lett.* 50 1395
[2] Dunne G V 1993 Int. J. Mod. Phys. B7 4783
[3] Di Francesco P, Gaudin M, Itzykson C and Lesage F 1994 Int. J. Mod. Phys. A9 4257
[4] Scharf T, Thibon J-Y and Wybourne B G 1994 *J. Phys. A: Math. Gen.* 27 4211

Computing the Generating Function of a Series Given Its First Few Terms

François Bergeron and Simon Plouffe

CONTENTS

We outline an approach for the computation of a good candidate for the generating function of a power series for which only the first few coefficients are known. More precisely, if the derivative, the logarithmic derivative, the reversion, or another transformation of a given power series (even with polynomial coefficients) appears to admit a rational generating function, we compute the generating function of the original series by applying the inverse of those transformations to the rational generating function found.

1. INTRODUCTION

We address the problem of finding the generating function $f(x)$ of a power series

$$\alpha(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots,$$

of which we know only a limited number of initial terms. We say that $\alpha(x)$ has *precision* n if all coefficients up to x^n are known. Clearly, in the absence of additional information, the knowledge of $\alpha(x)$ to any finite precision is not sufficient to determine $f(x)$ uniquely.

One instance when the problem can be solved is when $f(x)$ is known a priori to be a rational function

$$\frac{p_0 + p_1 x + \cdots + p_j x^j}{q_0 + q_1 x + \cdots + q_k x^k} \quad \text{with } p_j, q_k \neq 0, \qquad (1.1)$$

and the precision of $\alpha(x)$ is at least $j + k$. Many good algorithms exist for computing $f(x)$ in this case. A naive one is to use the method of indeterminate coefficients in (1.1), with $j + k = n$. Better algorithms make use of (for example) Padé approximants. The function `convert/ratpoly` provided by the computer algebra system Maple [Char et al. 1985] includes the Padé approximants method.

Bergeron had support from NSERC-Canada and FCAR-Québec.

1058-6458/96 $0.50 per page

If we don't know that the generating function is rational, we can still apply a rational function approximation algorithm to $\alpha(x)$, to obtain an expression of the form (1.1) whose Taylor expansion coincides with $\alpha(x)$ throughout the known terms. If we find out that $k+j$ is much less than the precision n, we can consider the rational fraction obtained a good candidate for the generating function $f(x)$. The greater n is with respect to $j+k$, the more confident we can be in our guess.

Our purpose here is to show that one can easily extend the class of series for which a good candidate for a generating function can explicitly be computed from the knowledge of just enough terms of a series. The main idea is to try to transform the series into one that admits a rational generating function. If this transformation is successful, in the sense that the result appears to be rational, one need only apply the inverse transformation to the resulting rational function in order to produce an explicit candidate for the generating function of the original series. Thus, a measure of rationality for series is crucial to our scheme.

Using this idea, we wrote a Maple program that will find generating functions such as

$$\tan x, \qquad \exp(te^x - t), \qquad (1-4x)^{-3/2},$$
$$\exp\left(\frac{1-\sqrt{1-2\,xt}}{x} - t\right) \quad \text{and} \quad \frac{1}{1-xe^{A(x)}},$$

where $A(x)$ is the solution to the functional equation $A(x) = x\exp A(x)$—and even more complex ones. The program is described in Section 2, and examples are given in Section 3 that show it to be surprisingly successful. It typically gives results in a few seconds on a Mips/3000 or on a Macintosh IIfx. Moreover, it works with series whose coefficients are polynomials or rational functions, as well as numbers; the generating function in such cases involves a formal parameter, as in the case of $\exp(te^x - t)$ above, which arises in connection with Stirling polynomials of the second kind (see Example 8 in Section 3).

2. THE PROGRAM

The heart of the program is a test for the existence of a good rational function approximation (1.1) for a given series, where *good* is defined to mean that $k+j$ is less than the precision n of the series. This rationality test is implemented in the function `testrat`, which returns either the rational function that has been found, or the keyword `FAIL`.

The power of the program lies in the association of this rationality test with operations such as differentiation, logarithmic differentiation and reversion. (Recall that a series

$$\alpha(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n + \cdots$$

with $a_0 = 0$ and $a_1 \neq 0$ has a unique *reversion* $\alpha^{\langle -1\rangle}(x)$, that is, a series satisfying $\alpha^{\langle -1\rangle}(\alpha(x)) = x$. The generating function of $\alpha^{\langle -1\rangle}(x)$ is inverse to the generating function of $\alpha(x)$, and the first n terms of $\alpha^{\langle -1\rangle}(x)$ depend only on the first n terms of $\alpha(x)$. The *logarithmic derivative* of a series $\alpha(x)$ is $\alpha'(x)/\alpha(x)$.)

In general, the first step of a computation is to execute some transformation Γ on a given series $\alpha(x)$, then to test the resulting series for rationality. If $\Gamma(\alpha(x))$ admits a good rational generating function $f(x)$, the program computes $\Gamma^{-1}(f(x))$, where Γ^{-1} is the transformation inverse to Γ. Note that some operations Γ, such as differentiation, reduce the precision of the series.

This strategy is implemented by calling `testrat` with the functions `testdrat`, `testdlograt` and `testrevrat`. Each of these three functions takes three arguments: the series, the variable (which we have been calling x), and the type of test that should be performed on the transform. The last argument allows tests to be combined: for example, the call `testrevrat(series,x,testdlograt)` will test the logarithmic derivative of the reversion of the series for rationality. These tests, or compositions of them, are successively called by the main program (named `generating` in the examples that follow), which returns a generating function if possible.

Some renormalization of the series is included in `testdrat`, `testdlograt` and `testrevrat`, so that further operations can always be applied. For instance, a series should preferably be of the form

$$x + a_2x^2 + \cdots + a_nx^n + O(x^{n+1}).$$

for reversion.

3. EXAMPLES

The sidebars on this page and the next show a number of representative examples of use of the program `generating`. In some cases, the output has been simplified, using Maple. We use standard mathematical notation for ease of reading, but the Maple input and output is straightforward. The input for Example 1, for example, would be

```
> generating(x + x^2 + 2 x^3 + 3 x^4 +
>    5 x^5 + 8 x^6 + O(x^7));
```

where > is the Maple prompt. The program outputs either "`The generating function of this series appears to be ...`" or "`I can find no generating function for this series.`"

Some of the examples were selected from the forthcoming second edition of N. J. A. Sloane's *Handbook of Integer Sequences* [Sloane]. We applied the program to a great number of power series, both ordinary and exponential, corresponding to the sequences in that book (that is, the coefficients of the series were the terms of the sequences). We chose our examples either for their intrinsic elegance, or because they appear to be unknown, or to illustrate the power of the method. Some examples illustrate the use of the program on series with polynomial coefficients.

Example 1. This is the series coming from the Fibonacci sequence. Here `generating` uses directly Maple's function `convert/ratpoly`. The smallest precision for which the result comes out right is six, as shown. With a direct use of this `ratpoly` function (and a simple rejection test) we obtained generating functions for about 600 out of the 4568 sequences in [Sloane].

Example 2. Here the program took the derivative.

Example 3. This is a specialization at $t = -1$ of the next example.

Example 4. This is the exponential generating function for Hermite polynomials. Observe how the input series can have polynomial coefficients, and how the number of terms needed to yield a significant result is quite small.

Example 5. Here the program took the logarithmic derivative.

Example 6. Several generating functions with exponents such as $\frac{3}{2}$, $\frac{5}{2}$, $\frac{7}{2}$ and $\frac{11}{2}$ were obtained when we ran our program on the sequences appearing in [Sloane].

Example 7. This is the exponential generating function for Stirling polynomials of the first kind, which count permutations by number of cycles.

	Input	Output
1	$x + x^2 + 2x^3 + 3x^4 + 5x^5 + 8x^6 + O(x^7)$	$\dfrac{-x}{-1 + x + x^2}$
2	$2 + 5x + \frac{11}{2}x^2 + \frac{19}{3}x^3 + \frac{29}{4}x^4 + \frac{41}{5}x^5 + \frac{55}{6}x^6 + \frac{71}{7}x^7 + \frac{89}{8}x^8 + \frac{109}{9}x^9 + O(x^{10})$	$\dfrac{2 - x^2}{(1-x)^2} + \ln\dfrac{1}{1-x}$
3	$1 + x + x^2 + \frac{2}{3}x^3 + \frac{5}{12}x^4 + \frac{13}{60}x^5 + \frac{19}{180}x^6 + \frac{29}{630}x^7 + \frac{191}{10080}x^8 + \frac{131}{18144}x^9 + O(x^{10})$	$\exp\left(x + \frac{1}{2}x^2\right)$
4	$1 - xt + \left(\frac{1}{2} + \frac{1}{2}t^2\right)x^2 - \left(\frac{1}{2}t + \frac{1}{6}t^3\right)x^3 + \left(\frac{1}{8} + \frac{1}{4}t^2 + \frac{1}{24}t^4\right)x^4 - \left(\frac{1}{8}t + \frac{1}{12}t^3 + \frac{1}{120}t^5\right)x^5 + O(x^6)$	$\exp\left(\frac{1}{2}x(-2t + x)\right)$
5	$1 + x + x^2 + \frac{5}{6}x^3 + \frac{17}{24}x^4 + \frac{73}{120}x^5 + \frac{97}{180}x^6 + \frac{2461}{5040}x^7 + \frac{3631}{8064}x^8 + \frac{152531}{362880}x^9 + O(x^{10})$	$\dfrac{\exp\left(\frac{1}{4}x^2 + \frac{1}{2}x\right)}{\sqrt{1-x}}$
6	$1 + 24x + 270x^2 + 2240x^3 + 15750x^4 + 99792x^5 + 588588x^6 + 3294720x^7 + 17721990x^8 + 92378000x^9 + O(x^{10})$	$\dfrac{1 + 10x + 4x^2}{(1-4x)^{7/2}}$
7	$1 + tx + \frac{1}{2}(t^2 + t)x^2 + \frac{1}{6}(t^3 + 3t^2 + 2t)x^3 + \frac{1}{24}(t^4 + 6t^3 + 11t^2 + 6t)x^4 + \frac{1}{120}(t^5 + 10t^4 + 35t^3 + 50t^2 + 24t)x^5 + O(x^6)$	$\left(\dfrac{1}{1-x}\right)^t$

Example 8. This is the exponential generating function for Stirling polynomials of the second kind, which count partitions of a set by number of parts. This result was obtained through a double logarithmic derivative.

Example 9. This illustrates the use of a rationality test on the reversion of a series. The reversion of this generating function is $x/(1+x)^3$; therefore the generating function $f(x)$ is obtained as the real solution of the cubic equation

$$(1+f(x))^3\, x - f(x) = 0.$$

Example 10. This generating function has two parameters, and admits as one specialization the generating function for Laguerre polynomials. One can find a generating function for most of the classical orthogonal polynomials using our program on the first seven or so terms of their series.

Example 11. This generating function counts functions from a set into itself with weight t^k, where k is the number of recurrent points in the function. $\mathrm{Rev}(f(x), x)$ stands for the inverse for composition of $f(x)$. If we denote by $A(x)$ the solution to the functional equation $A(x) = x\exp(A(x))$, the generating function is equal to

$$\frac{1}{1 - txe^{A(x)}}.$$

$A(x)$ is the generating function for rooted trees.

Many other functions such as $\tan x$, $\arctan x$, or $\arcsin x$ also appeared as generating functions in our experiments.

4. CONCLUSIONS

The success of our approach, and also its limitations, depend on the set of transformations tried before a rationality test is made. Many transfor-

	Input	Output
8	$1 + tx + \frac{1}{2}(t^2+t)x^2 + \frac{1}{6}(t+3t^2+t^3)x^3 + \frac{1}{24}(t+7t^2+6t^3+t^4)x^4 + \frac{1}{120}(t+15t^2+25t^3+10t^4+t^5)x^5 + O(x^6)$	$\exp(te^x - t)$
9	$x + 3x^2 + 12x^3 + 55x^4 + 273x^5 + 1428x^6 + 7752x^7 + 43263x^8 + 246675x^9 + O(x^{10})$	$-1 + \frac{(12\sqrt{81x-12} - 108\sqrt{x})^{1/3}}{6\sqrt{x}} - \frac{(12\sqrt{81x-12} + 108\sqrt{x})^{1/3}}{6\sqrt{x}}$
10	$1 + (t+s)x + \frac{1}{2}(t^2+2ts+s^2+t+2s)x^2 + \frac{1}{6}(t^3+3t^2s+3ts^2+s^3+3t^2+9ts+6s^2+2t+6s)x^3 + \frac{1}{24}(t^4+4t^3s+6t^2s^2+4ts^3+s^4+6t^3+24t^2s + 30ts^2+12s^3+11t^2+44ts+36s^2+6t+24s)x^4 + \frac{1}{120}(t^5+5t^4s+10t^3s^2+10t^2s^3+5ts^4+s^5+10t^4 + 50t^3s+90t^2s^2+70ts^3+20s^4+35t^3+175t^2s + 260ts^2+120s^3+50t^2+250ts+240s^2+24t+120s)x^5 + O(x^6)$	$\left(\frac{1}{1-x}\right)^t \exp\left(\frac{sx}{1-x}\right)$
11	$xt + \left(t+t^2\right)x^2 + \left(\frac{3}{2}t+2t^2+t^3\right)x^3 + \left(4t^2+3t^3+\frac{8}{3}t+t^4\right)x^4 + \left(\frac{25}{3}t^2+\frac{15}{2}t^3+\frac{125}{24}t+4t^4+t^5\right)x^5 + \left(18t^2+18t^3+\frac{54}{5}t+12t^4+5t^5+t^6\right)x^6 + \left(\frac{343}{8}t^3+\frac{98}{3}t^4+\frac{2401}{60}t^2+\frac{35}{2}t^5+\frac{16807}{720}t+6t^6+t^7\right)x^7 + \left(\frac{16384}{315}t+7t^7+t^8+24t^6+\frac{160}{3}t^5+\frac{256}{3}t^4+\frac{512}{5}t^3+\frac{4096}{45}t^2\right)x^8 + O(x^9)$	$t\,\mathrm{Rev}\left(\frac{x}{xt+1}\exp\left(-\frac{x}{xt+1}\right), x\right)$

mations beyond differentiation, logarithmic differentiation and reversion may be considered. For instance, one could choose any invertible function $f(x)$ and consider the following transformations on a series $\alpha(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n + O(x^{n+1})$:

$$\theta_f(\alpha(x)) = \text{taylor}(a_0 + a_1f(x) + \cdots + a_nf(x)^n),$$
$$\theta^f(\alpha(x)) = \text{taylor}(f(a_1x + a_2x^2 + \cdots + a_nx^n)).$$

Here taylor(g) stands for the operation of taking the Taylor expansion around 0 of a function g, and θ^f is defined when $a_0 = 0$. If $f^{\langle -1\rangle}$ denotes the reversion of $f(x)$, one easily checks that

$$(\theta_f)^{-1}(g(x)) = g(f^{\langle -1\rangle}(x)),$$
$$(\theta^f)^{-1}(g(x)) = f^{\langle -1\rangle}(g(x)).$$

One nice case is when $f(x) = \ln x$ in θ_f. This transformation allows the computation of generating functions that are rational functions of the exponential. For instance, one could obtain in this manner the generating function

$$\frac{e^x - 1}{2 - e^x}$$

for the series

$$x + \tfrac{3}{2}x^2 + \tfrac{13}{6}x^3 + \tfrac{25}{8}x^4 + \tfrac{541}{120}x^5 + \tfrac{1561}{240}x^6 + \tfrac{47293}{5040}x^7 + \tfrac{36389}{2688}x^8 + \tfrac{7087261}{362880}x^9 + O(x^{10}),$$

which is the exponential series for ordered partitions of a set. As it happens, our program found this generating function by other means, namely by taking the derivative of the reversion of the series, whose generating function is

$$\frac{1}{(1+2x)(1+x)}.$$

To describe other possible extensions of our approach, we recall some definitions. A series $y(x)$, with coefficients in $\mathbf{K}$, is said to be *differentiably finite* or *D-finite* [Stanley 1980] if it satisfies some nontrivial linear differential equation

$$p_0(x)y + p_1(x)y' + \cdots + p_k(x)y^{(k)} = 0 \qquad (4.1)$$

with coefficients $p_j(x) \in \mathbf{K}[x]$. A series $y = y(x)$ is said to be *constructible differentially finite* or *CDF* [Bergeron and Reutenauer 1990] if, for some $k \geq 1$, there exist k series $y_1, \ldots, y_k$, with $y_1 = y$, and polynomials $P_1, \ldots, P_k$ with coefficients in $\mathbf{K}$, satisfying

$$y_i' = P_i(y_1, \ldots, y_k) \quad \text{for } i = 1, \ldots, k. \qquad (4.2)$$

Both of these classes of series contain polynomials, algebraic series, and the Taylor expansion around 0 of usual functions such as e^x, $\log(1+x)$, or the trigonometric functions. They are also closed under addition and multiplication, and under composition with algebraic series. However, the CDF class is not closed under Hadamard (termwise) product, whereas the D-finite class is. On the other hand, CDF is closed under differentiation, integration, inversion $(1/y(x))$, composition and reversion.

Neither class is contained in the other. All CDF series are analytic around 0, so $\sum_n n!\,x^n$ is not CDF, though it is D-finite. On the other hand, the series expansion around 0 of $1/\cos x$ is not D-finite, but is CDF.

Both classes allow for the characterization of a wide range of generating functions. If one knows the form of the liner differential equation (4.1) or the system (4.2)—that is, the number of equations and the degrees of the polynomials—the exact equation or system characterizing a given series or a set of series can then be found from the series' first terms. In the case of D-finite series, this technique has already been proposed and implemented by Guttmann [Brak and Guttmann 1990]. For CDF series, we have an experimental program that has been used to obtain nice new generating functions such as

$$F(u, v, x) = \frac{\alpha^2}{e^x((1+u)\sin(\frac{1}{2}\alpha x) - \cos(\frac{1}{2}\alpha x))^2}, \qquad (4.3)$$

where $\alpha = \sqrt{2v - (1+u)^2}$. This is a generating function (with parameters) for the number of maximal up-going paths in the composition poset (ongoing research in collaboration with S. Dulucq and M. Bousquet-Mélou). Function (4.3) is not D-finite but is CDF. To obtain it, we used the first few terms of the series

$$1 + ux + \tfrac{1}{2}(v+u^2)x^2 + \tfrac{1}{6}(v + 4vu + u^3)x^3 + \tfrac{1}{24}(v + 4v^2 + 6vu + 11vu^2 + u^4)x^4 + \tfrac{1}{120}(v + 14v^2 + 34uv^2 + 8vu + 23u^2v + 26vu^3 + u^5)x^5 + \cdots,$$

obtained by explicit enumeration of the objects considered, in order to find the system

$$F' = F(1+G), \qquad F(u,v,0) = 1,$$
$$G' = v + (1+u)G + G^2/2, \qquad G(u,v,0) = 0.$$

Expression (4.3) is easily computed from this.

Our first implementation of `generating` computed a generating function for either the ordinary or the exponential series of about 1000 out of the 4568 sequences appearing in [Sloane]. Since the first version of this article was written, a Maple package implementing some ideas presented here, as well as others such as the D-finite approach, has been written by Bruno Salvy and Paul Zimmermann of INRIA [Salvy and Zimmermann]. It is now available as a shared package under the name "`gfun`". (To learn more about obtaining shared packages, type `?share` to Maple.) The analogue of our function `generating` in `gfun` is the function `guessgf`. Giving `guessgf` the right set of options results in its using the set of transformations described in Section 2 of this paper.

ACKNOWLEDGEMENTS

We would like to thank G. Labelle, N. J. A. Sloane and an anonymous referee for their constructive comments.

REFERENCES

[Bergeron and Reutenauer 1990] F. Bergeron and C. Reutenauer, "Combinatorial resolution of systems of differential equations, III: A special class of differentially algebraic series", *Europ. J. Combin.* **11** (1990), 501–512.

[Brak and Guttmann 1990] R. Brak and A. J. Guttmann, "Algebraic approximants: a new method of series analysis", *J. Phys.* A**23** (1990), L1331–1337.

[Char et al. 1985] Bruce W. Char et al., *Maple User's Guide*, 4th ed., Watcom Publications, Waterloo, Ont., 1985.

[Salvy and Zimmermann] B. Salvy and P. Zimmermann, "GFUN: A Maple Package for the Manipulation of Generating and Holonomic Functions in one Variable", to appear in *ACM Trans. in Math. Software*.

[Sloane] N. J. A. Sloane, *A Handbook of Integer Sequences*, 2nd ed., Academic Press, to appear (1st ed., 1973).

[Stanley 1980] R. P. Stanley, "Differentiably finite power series", *Europ. J. Combin.* **1** (1980), 175–188.

[Zeilberger 1991] D. Zeilberger, "A Maple program for proving hypergeometric identities", *SIGSAM Bulletin* **25**(3) (1991), 4–13.

François Bergeron, LACIM, Université du Québec à Montréal, Montréal H3C 3P8, Canada
Spring 1993: LaBRI, Université de Bordeaux I, 33405 Talence, France (bergeron@catalan.math.uqam.ca)

Simon Plouffe, LACIM, Université du Québec à Montréal, Montréal H3C 3P8, Canada

Received October 30, 1991; accepted in revised form January 25, 1993

Locally Restricted Compositions

Edward A. Bender
Department of Mathematics
University of California, San Diego
La Jolla, CA 92093-0112
ebender@ucsd.edu

E. Rodney Canfield
Department of Computer Science
University of Georgia
Athens, GA 30602
erc@cs.uga.edu

AMS Subject Classification: 05A15, 05A16

Submitted: April 23, 2003; Accepted: XXXXXX.

Abstract

Compositions $n = a_1 + a_2 + \cdots$, $a_k > 0$, have been studied classically. More recently, compositions with the local restriction $a_k \neq a_{k+1}$ (Carlitz compositions) have been studied by various authors. We consider the compositions with more general local-nonequality restrictions, including multiline compositions. We obtain recursions and bounds on growth rate. Under reasonable assumptions, we show that, in a randomly selected restricted composition, the largest part is almost surely $O(\log n)$ and that the number of parts is asymptotically normally distributed with mean and variance proportional to n.

1 Introduction

Let $R_0, \ldots, R_{m-1}$ be a sequence of finite sets of positive integers. A sequence $a_1, \ldots, a_k$ will be called a k-part, *locally-restricted* composition of n (with restrictions $R_0, \ldots, R_{m-1}$) if

(a) the a_i are strictly positive integers,

(b) $a_1 + a_2 + \cdots + a_k = n$ and

(c) when $t \equiv j \pmod{m}$ we have $a_t \neq a_{t-r}$ for all $r \in R_j$ such that $k \geq t > r$.

Here are some examples:

- $m = 1$ and $R_0 = \emptyset$: unrestricted compositions.
- $m = 1$ and $R_0 = \{1\}$: Carlitz compositions (adjacent parts must differ).
- $m = 1$ and $R = \{1, 2, \ldots, r\}$: parts within distance r must differ. We call these *distance-r compositions*.
- $m = 2$, $R_0 = \{1, 2\}$ and $R_1 = \{2\}$: 2-rowed (restricted) compositions where parts adjacent in row or column must differ. The parts are listed in the order

$$\begin{array}{cccc} a_1 & a_3 & a_5 & \ldots \\ a_2 & a_4 & a_6 & \ldots \end{array}$$

  and we are interested in those compositions having an even number of parts.
- $m = r > 1$, $R_1 = \{m\}$ and $R_i = \{1, m\}$ for $i \neq 1$: r-rowed (restricted) compositions where adjacent parts must differ and we are interested in those compositions where the number of parts is a multiple of r.

We say $C(x, y) = \sum c_{n,k} x^n y^k$ is a composition generating function if $c_{n,k}$ is the number of (suitably locally restricted) compositions of n having exactly k parts. Thus $c_n = [x^n]\, C(x, 1)$ counts (restricted) compositions of n without regard to the number of parts.

In Section 2, we briefly review some results on unrestricted and Carlitz compositions. The remainder of the paper falls into two main parts. The *first part* deals with the problem of obtaining functional equations for generating functions. This requires the introduction into $C(x, y)$ of additional variables keeping track of certain part sizes. Although we cannot solve these recursions, they can be used to compute the number of locally restricted compositions by size and number of parts.

- In Section 3 we discuss the generating function for distance-r compositions; i.e., there is a single restriction set R_0 of the form $R_0 = \{1, \ldots, r\}$. A composition will satisfy these local restrictions if and only if in every interval, or window, of length $r + 1$ one finds no two parts which are equal.

- We discuss an arbitrary single restriction set R_0 in Section 4. To do so, we introduce several interconnected generating functions that depend on which of the last $\max R$ elements of the composition are equal to each other.

- The general case of m restriction sets is discussed in Section 5. This leads to an m-fold increase in the number of generating functions.

The *second part* of the paper deals with approximate and numerical results.

- In Section 6 we prove that $\lim_{n\to\infty} {c_n}^{1/n}$ exists. Of course, the reciprocal of the limit is ρ, the radius of convergence of $C(x,1)$.

- In Section 7, we point out that transfer matrices can be used to obtain upper and lower bounds, and we use them to obtain bounds on ρ for those compositions whose numerical values are computed in Section 9.

- Conectures are discussed in Section 8. We conjecture a stronger result than $\lim c_n^{1/n} = 1/\rho$, namely $c_n \sim B\rho^{-n}$. This implies that the largest part of almost all locally restricted compositions of n is asymptotic to $\log_{1/\rho} n$. We show how some plausible assumptions concerning the $C(x,y)$ lead to the conclusion $c_n \sim B\rho^{-n}$ as well as asymptotic normality for $c_{n,k}$. It is our hope that someone will be able to justify the assumptions or otherwise establish our conjectures.

- Section 9 gives tabulated values of c_n for these cases: Carlitz, distance 2, 2-rowed, and 3-rowed. We also give upper and lower bounds for the radii of convergence computed by the transfer matrix method discussed in Section 7.

Since local restrictions insure that we cannot have arbitrarily long runs of equal parts, we could allow parts to be zero. In other words, replace condition (a) with

(a′) the a_i are non-negative integers.

Our results can be modified to handle this change, the necessary adjustment being to replace xz and xz_i by 1 in the numerators of some generating functions.

2 Unrestricted and Carlitz Compositions

Our review in this section of unrestricted and Carlitz compositions is intended to provide a comparison for the general case. For additional results and references in this area, see the paper by Hitczenko and Louchard [2].

Let $C(x,y)$ be the generating function for compositions with no restrictions. Since a composition consists either of a single part or a composition followed by a single part, we have

$$C(x,y) = \frac{xy}{1-x} + C(x,y)\frac{xy}{1-x} \tag{2.1}$$

and so

$$C(x,y) = \frac{xy}{1-x-xy}. \tag{2.2}$$

The existence of the explicit formula (2.2) allows one to obtain a variety of results easily. For example, since $C(x,1) = \frac{x}{1-2x}$, the number of compositions of n is 2^{n-1}. Since the number with exactly k parts is $\binom{n-1}{k-1}$, the number of parts in a random composition is asymptotic to a normal random variable with mean $n/2$ and variance $n/4$. The distribution of the largest part is known [2], from which it follows that the largest part of a random composition is almost surely asymptotic to $\log_2 n$.

Let $C(x,y)$ be the generating function for Carlitz compositions. To compute it we introduce a new variable as done by Knopfmacher and Prodinger [3]: $C(x,y,z) = \sum a_{n,k,i} x^n y^k z^i$ where $a_{n,k,i}$ is the number of k-part Carlitz compositions of n whose last part is i. We are interested in $C(x,y,1)$. Since a Carlitz composition is either a single part or a Carlitz composition followed by a part different from the last, we have

$$C(x,y,z) = \frac{xyz}{1-xz} + C(x,y,1)\frac{xyz}{1-xz} - yC(x,y,xz), \tag{2.3}$$

because $yC(x,y,xz)$ counts compositions which are Carlitz except that the last two parts are equal. This is not as easily solved as (2.1); however, it can be done. As in [3], note that

$$C(x,y,x^{k-1}z) = \frac{x^k yz}{1-x^k z}C(x,y,1) + \frac{x^k yz}{1-x^k z} - yC(x,y,x^k z).$$

By iteration,

$$C(x,y,x^{k-1}z) = \left(yz\sum_{t=0}^{\infty}\frac{(-y)^t x^{t+k}}{1-x^{t+k}z}\right)C(x,y,1) + \left(yz\sum_{t=0}^{\infty}\frac{(-y)^t x^{t+k}}{1-x^{t+k}z}\right). \tag{2.4}$$

With $k=1$ and $z=1$, we obtain $C(x,y,1) = g(x,y)C(x,y,1) + g(x,y)$ and so

$$C(x,y,1) = \frac{g(x,y)}{1-g(x,y)} \quad \text{where} \quad g(x,y) = -\sum_{t=1}^{\infty}\frac{(-xy)^t}{1-x^t}. \tag{2.5}$$

Note that $g(x,1)$ and $g_y(x,1)$ converge for $|x|<1$. Knopfmacher and Prodinger show that $g(x,1)-1$ has a simple zero at $x=\rho=0.571349\ldots$ and no other zeroes with $|x|\le\rho$. Therefore the number of Carlitz compositions of n is asymptotic to $A\rho^{-n}$. Since

$$C_y(x,1) = \frac{g_y(x,1)}{(1-g_y(x,1))^2},$$

$C_y(x,1)$ has a second order pole at $x=\rho$ and no other singularities for $|x|\le\rho$. Thus $[x^n]\ C_y(x,1) \sim Bn\rho^{-n}$. Thus the average number of parts is asymptotically Bn/A. Furthermore, Louchard and Podinger [4] conclude that the number of parts in a random Carlitz composition is asymptotically normal with variance σ^2 proportional to n. From [3] and [4]

$$A = 0.456387\cdots, \quad B/A = 0.350601\cdots \quad \text{and} \quad \sigma^2 = 0.13391\cdots.$$

It follows from [3] that the largest part of a random Carlitz composition of n is almost surely asymptotic to $\log_{1/\rho} n$.

The number of Carlitz compositions appears as sequence `A003242` in the *On-Line Encyclopedia of Integer Sequences*.

3 Recursions for Distance-r Compositions

Suppose we have a single restriction set R_0 of the particular form $R_0 = \{1, 2, \ldots, r\}$. The argument leading to (2.3) can be extended. Let $C(x, y, z_1, \ldots, z_r)$ be the generating function for these compositions where

- z_i keeps track of the size of the i^{th} part, counting from the right end rather than the left and
- we only consider compositions with at least r parts.

The latter restriction does not alter the asymptotic behavior. Let $y^r f(x, z_1, \ldots, z_r)$ count those compositions with exactly r parts. The generating function for compositions with length greater than r such that R_0 is satisfied except that the last part equals the part $k \le r$ positions earlier is

$$yC(x, y, w_{k,1}z_2, \ldots, w_{k,r-1}z_r, w_{k,r}) \quad \text{where} \quad w_{k,j} = \begin{cases} xz_1 & \text{if } k = j, \\ 1 & \text{otherwise.} \end{cases}$$

Since it is impossible that this part equal a part $j \le r$ positions earlier, we have

$$\begin{aligned} C(x, y, z_1, \ldots, z_r) &= y^r f(x, z_1, \ldots, z_r) + C(x, y, z_2, \ldots, z_r, 1)\frac{xyz_1}{1 - xz_1} \\ &- \sum_{k=1}^{r} yC(x, y, w_{k,1}z_2, \ldots, w_{k,r-1}z_r, w_{k,r}). \end{aligned} \tag{3.1}$$

We can iterate this recursion as was done for Carlitz compositions; however, we were unable to write out simple summations as in (2.4). If we start with the estimate $A = f$ and iterate (3.1), each iteration increases by 1 the smallest power of y for which the estimate is wrong. Thus iteration leads to

$$C(x, y, 1, \ldots, 1) = g(x, y)C(x, y, 1, \ldots, 1) + h(x, y) \tag{3.2}$$

for some formal power series $g(x, y)$ and $h(x, y)$. We believe these series actually converge — more on this in Section 8.

One can use symbolic manipulation to obtain values for $c_{n,k}$. It is less computationally demanding to modify (3.1) by extracting coefficients of y^k: Let $C_k(x, z_1, \ldots, z_r) = [y^k]\, C(x, y, z_1, \ldots, z_r)$. Then,

$$C_{k+1}(x, z_1, \ldots, z_r) = \begin{cases} f(x, z_1, \ldots, z_r), & \text{if } k = r - 1, \\ C_k(x, z_2, \ldots, z_r, 1)\dfrac{xz_1}{1 - xz_1} & \\ \qquad - \displaystyle\sum_{k=1}^{r} C_k(x, w_{k,1}z_2, \ldots, w_{k,r-1}z_r, w_{k,r}), & \text{if } k \ge r. \end{cases} \tag{3.3}$$

We compute f by Möbius inversion over the lattice Π_r of partitions of $\{1,\ldots,r\}$, with the complete refinement at the bottom. For $\pi \in \Pi_r$, let $F_\pi(x, z_1, \ldots, z_r)$ be the generating function for compositions $a_1, \ldots, a_r$ such that $a_i = a_j$ whenever i and j are in the same block of π. Then

$$\begin{aligned} f(x, z_1, \ldots, z_r) &= \sum_{\tau \in \Pi_r} \mu(0,\tau) F_\tau(x, z_1, \ldots, z_r) \\ F_\tau(x, z_1, \ldots, z_r) &= \prod_{B \in \tau} \frac{\prod_{i \in B}(x z_i)}{1 - \prod_{i \in B}(x z_i)}, \end{aligned} \tag{3.4}$$

where $B \in \tau$ means B is a block of τ, and of course μ denotes the Möbius function for the partition lattice.

The previous calculations were for compositions with at least r parts; however, it is more natural to allow any number of parts. To do this, we compute $C(x, y, 1, \ldots, 1)$ and then add the generating functions for compositions with k parts for all $k < r$. These functions can be computed by using (3.4) with r replaced by k.

4 Recursions for a Single Restriction Set

To obtain recursive equations for a general single restriction set $R_0 = \{r_1, \ldots, r_t\}$, we allow both forced equalities and inequalties among the final few parts of a composition. Let $r = \max r_i$. For every partition π of $\{0, \ldots, r\}$, let $C_\pi(x, y, z_1, \ldots, z_r)$ be the generating function for locally restricted compositions having at least r parts, where x, y and z_i are as before. Furthermore, if $a_1, \ldots, a_k$ is such a composition and $0 \le i, j \le r$, then $a_{k-i} = a_{k-j}$ if and only if i and j are in the same block of π. (We take $a_0 = 0$.) Note that, whereas R_0 applies to all parts of the composition, π applies only to the last $r+1$ parts.

Since A is the sum of C_π over all π, it suffices to compute the C_π. It is quite possible that $C_\pi = 0$. For example, if $R_0 = \{1, \ldots, r\}$, then $C_\pi = 0$ unless π is the complete refinement — in which case C_π is the A of the previous section.

Compatibility: We say that π and R_0 are incompatible if there are $i > j$ in the same block of π such that $i - j \in R_0$. We claim that $C_\pi = 0$ if and only if π and R_0 are incompatible. First, suppose they are incompatible and $a_1, \ldots, a_k$ is counted by C_π. From π, we have $a_{k-i} = a_{k-j}$. Looking at $a_1, \ldots, a_{k-j}$ and using R_0, we have $a_{k-j} \neq a_{k-j-(i-j)} = a_{k-i}$, a contradiction. Conversely, suppose that π and R_0 are compatible. We now construct a composition counted by C_π. Let $B_1, \ldots, B_b$ be the blocks of π. Define $a_{r+1-j} = k$ for all $j \in B_k$. The composition $a_1, \ldots, a_{r+1}$ satisfies π. If R_0 requires that $a_i \neq a_j$, then compatibility implies that i and j must be in different blocks of π and so $a_i \neq a_j$ for the composition we constructed.

The remainder of this section is devoted to obtaining recursions for C_π when π and R_0 are compatible.

Define $S(\pi)$ to be a collection of partitions as follows. First, let π' be π with all elements decreased by 1 and with the element -1 in the resulting partition discarded. The set $S(\pi)$ consists of all partitions of $\{0, \ldots, r\}$ such that removal of r gives π'. For

example, if $r = 5$ and $\pi = \{\{0,3\}, \{1,4,5\}, \{2\}\}$, then $\pi' = \{\{2\}, \{0,3,4\}, \{1\}\}$ and $S(\pi)$ contains the four partitions

$$\{\{2,5\}, \{0,3,4\}, \{1\}\} \qquad \{\{2\}, \{0,3,4,5\}, \{1\}\} \qquad \{\{2\}, \{0,3,4\}, \{1,5\}\}$$
$$\{\{2\}, \{0,3,4\}, \{1\}, \{5\}\}.$$

We are now in a position to write down the recursion. First suppose 0 and $i \neq 0$ are in the same block of π. Then

$$C_\pi(x, y, z_1, \ldots, z_r) \;=\; y \sum_{\sigma \in S(\pi)} C_\sigma(x, y, v_1 z_2, \ldots, v_{r-1} z_r, v_r) \;+\; y^r f_\pi(x, z_1, \ldots, z_r), \quad (4.1)$$

where

$$v_k \;=\; \begin{cases} xz_1 & \text{if } k = i, \\ 1 & \text{if } k \neq i. \end{cases}$$

because we must shift indices by one and insure that the new part is the same size as the part i away from it. Now suppose 0 is in a block by itself in π. This is much like the $\{1, \ldots, r\}$ case: We choose the new part arbitrarily and then subtract off the case that it equals one of the parts in another block of π. This corresponds to the part equaling a part in one of the blocks of π'. Thus we have

$$\begin{aligned} C_\pi(x, y, z_1, \ldots, z_r) \;=\; & y \sum_{\sigma \in S(\pi)} \Big(C_\sigma(x, y, z_2, \ldots, z_r, 1) \frac{xz_1}{1 - xz_1} \\ & \qquad - \sum_{B \in \pi'} C_\sigma(x, y, w_{B,1} z_2, \ldots, w_{B,r-1} z_r, w_{B,r}) \Big) \\ & + y^r f_\pi(x, z_1, \ldots, z_r), \end{aligned} \quad (4.2)$$

where the sum on $B \in \pi'$ means that B runs through the blocks of π',

$$w_{B,i} \;=\; \begin{cases} xz_1 & \text{if } i = k_B + 1, \\ 1 & \text{otherwise,} \end{cases}$$

and k_B is a designated element of block B (for example, the smallest). The computation of f_π is similar to that in (3.4):

$$\begin{aligned} f_\pi(x, z_1, \ldots, z_r) \;&=\; \sum_{\substack{\tau \in \Pi_r \\ \tau \geq \pi}} \mu(\pi, \tau) F_\tau(x, z_1, \ldots, z_r), \\ F_\tau(x, z_1, \ldots, z_r) \;&=\; \prod_{B \in \tau} \frac{\prod_{i \in B}(xz_i)}{1 - \prod_{i \in B}(xz_i)}. \end{aligned} \quad (4.3)$$

Let $C_{\pi,k} = [y^k]\, C_\pi$. Just as (3.1) was rewritten as (3.3), one can rewrite (4.1) and (4.2) as

$$C_{\pi,k}(x, z_1, \ldots, z_r) = \begin{cases} f_\pi(x, z_1, \ldots, z_r), & \text{if } k = r, \\ \displaystyle\sum_{\sigma \in S(\pi)} \Big(C_{\sigma,k-1}(x, z_2, \ldots, z_r, 1) \frac{x z_1}{1 - x z_1} \\ \qquad - \displaystyle\sum_{B \in \pi'} C_{\sigma,k-1}(x, w_{B,1} z_2, \ldots, w_{B,r-1} z_r, w_{B,r}) \Big), \\ \qquad\qquad \text{if } k > r \text{ and } \{0\} \in \pi, \\ \displaystyle\sum_{\sigma \in S(\pi)} C_{\sigma,k-1}(x, v_1 z_2, \ldots, v_{r-1} z_r, v_r), & \text{if } k > r \text{ and } \{0\} \notin \pi, \end{cases} \tag{4.4}$$

To eliminate the restriction that there be at least r parts, we proceed as at the end of the previous section: First compute $C_\pi(x, y, 1, \ldots, 1)$. Next, add terms f for each $k < r$ that are computed by using (4.3) with r replaced by k, remembering that $f = \sum f_\pi$, the partition extending over partitions π of Π_k compatible with R_0.

5 Recursions in the General Case

For the general case $R_0, \ldots, R_{m-1}$, one can follow very closely (4.4) above. We describe here a systematic approach, which need not lead to the most economical set of equations. As before, we only consider compositions with at least r parts in our recursions and can later add a correction term that counts compositions with less than r parts.

Define r to be the maximum element of all the R_t. The genrating function $A = C(x, y, z_1, \ldots, z_r)$, C_π and $C_{\pi,k}$ are defined as before: x and y keep track of size and parts and the z_i keep track of the last r parts. In addition, let $C_\pi^{[t]}$ be the sum of $C_{\pi,k} y^k$ over all $k \equiv t$ modulo m and let $C^{[t]}$ be the sum of $C_\pi^{[t]}$ over all π. Note that $C_\pi^{[t]} = C_\pi^{[s]}$ whenever $s \equiv t$ modulo m. The recursion will depend on the number of parts modulo m and so will be expressed in terms of $C_{\pi,k}$ or $C_\pi^{[t]}$.

The notion of compatability must be extended to allow for the fact that there are m sets of restrictions instead of just one. We say that π and the R_k are t-incompatible if there are $i > j$ in the same block of π such that $i - j \in R_{t-j}$, the subscript being understood modulo m. We claim that $C_\pi^{[t]} = 0$ if and only if π and the R_i are t-incompatible. The proof is essentially the same as it was for $m = 1$.

The previous equations now apply with minor modifications. Equation (4.3) for f_π does not depend on compatibility and so is unchanged. For $C_\pi^{[t]}$ and $C_{\pi,k}$ we obtain recursions when π the R_i are t-compatible by minor modifications as follows.

- To compute $C_\pi^{[t]}$, use (4.1) and (4.2) with the superscript $[t]$ on the left, the superscript $[t-1]$ on the right. The f_π term is included in (4.2) only when $t \equiv r$ modulo m.

- To compute $C_{\pi,k}$ where $k \equiv t$ modulo m, use (4.4).

We illustrate these ideas with 2-rowed and 3-rowed restricted compositions. To avoid cumbersome notation with braces, we write a partition such as $\{\{0,3\}\{1,2\}\}$ as $03|12$.

In the 2-rowed case, $m=2$, $R_0=\{1,2\}$ and $R_1=\{2\}$. Thus $r=2$. The 0-compatible partitions are $0|12$ and $0|1|2$ since 0 cannot appear with 1 or 2 because of R_0. The 1-compatible partitions are $01|2$ and $0|1|2$, where the partition $0|12$ was ruled out because 1 and 2 must be in different blocks due to R_0. We have

$$S(0|12)=\{012,\ 01|2\} \quad \text{and} \quad S(0|1|2)=S(01|2)=S(0|1|2)=\{02|1,\ 0|12,\ 0|1|2\}.$$

By (4.3)

$$f_{0|1|2}(x,z_1,z_2) \ = \ \frac{xz_1}{1-xz_1}\,\frac{xz_2}{1-xz_2}-\frac{x^2z_1z_2}{1-x^2z_1z_2}$$

and, by (4.1) and (4.2), with the first two arguments x and y of A omitted to save space,

$$\begin{aligned}
C^{[1]}_{01|2}(z_1,z_2) &= y\Big(C^{[0]}_{0|12}(xz_1z_2,1)+C^{[0]}_{0|1|2}(xz_1z_2,1)\Big)\\
C^{[1]}_{0|1|2}(z_1,z_2) &= y\Big(C^{[0]}_{0|12}(z_2,1)\frac{xz_1}{1-xz_1}-C^{[0]}_{0|12}(xz_1z_2,1)-C^{[0]}_{0|12}(z_2,xz_1)\\
&\qquad +C^{[0]}_{0|1|2}(z_2,1)\frac{xz_1}{1-xz_1}-C^{[0]}_{0|1|2}(xz_1z_2,1)-C^{[0]}_{0|1|2}(z_2,xz_1)\Big)\\
C^{[0]}_{0|12}(z_1,z_2) &= y\left(C^{[1]}_{01|2}(z_2,1)\frac{xz_1}{1-xz_1}-C^{[1]}_{01|2}(xz_1z_2,1)\right)\\
C^{[0]}_{0|1|2}(z_1,z_2) &= y\left(C^{[1]}_{0|1|2}(z_2,1)\frac{xz_1}{1-xz_1}-C^{[1]}_{0|1|2}(xz_1z_2,1)-C^{[1]}_{0|1|2}(z_2,xz_1z_2)\right)
\end{aligned}$$

We are interested in those compositions with even length so that a $2\times s$ rectangular array is filled for some s. Thus we compute $C^{[0]}=C^{[0]}_{0|1|2}+C^{[0]}_{0|12}$.

We now consider 3-rowed restricted compositions. In this case, $m=3$, $R_0=R_3=\{1,3\}$ and $R_1=\{1\}$. For each value of t, the four partitions $0|1|2|3$, $0|13|2$, $02|1|3$ and $02|13$ are t-compatible. In addition, $0|1|23$ is 0-compatible, $01|2|3$ is 1-compatible and $0|12|3$ is 3-compatible. Thus we are led to write down fifteen linked recursions. By studying these equations or reasoning directly, it is possible to reduce the system to six linked recursions. Define

$$D \ = \ C^{[0]}-C^{[0]}_{02|1|3}-C^{[0]}_{02|13}, \qquad F \ = \ C^{[1]}-C^{[1]}_{01|23}, \qquad E \ = \ C^{[2]}-C^{[2]}_{0|12|3}.$$

Then, again omitting the first two arguments x and y,

$$\begin{aligned}
C^{[1]}(z_1,z_2,z_3) &= y\Big(C^{[0]}(z_2,z_3,1)\frac{xz_1}{1-xz_1}-C^{[0]}(z_2,z_3,xz_1)\Big)\\
C^{[2]}(z_1,z_2,z_3) &= y\Big(C^{[1]}(z_2,z_3,1)\frac{xz_1}{1-xz_1}-C^{[1]}(xz_1z_2,z_3,1)-C^{[1]}(z_2,z_3,xz_1)\Big)\\
&\qquad +y^2C^{[0]}(z_3,x^2z_1z_2,1)
\end{aligned}$$

$$
\begin{aligned}
C^{[0]}(z_1,z_2,z_3) &= y\Big(C^{[2]}(z_2,z_3,1)\frac{xz_1}{1-xz_1} - C^{[2]}(xz_1z_2,z_3,1) - C^{[2]}(z_2,z_3,xz_1)\Big) \\
&\quad + y^2F(z_3,x^2z_1z_2,1) + y^3g(x,z_1,z_2,z_3) \\
D(z_1,z_2,z_3) &= C^{[0]}(z_1,z_2,z_3) - yE(z_2,xz_1z_3,1) + y^3h(x,z_1,z_2,z_3) \\
E(z_1,z_2,z_3) &= y\Big(F(z_2,z_3,1)\frac{xz_1}{1-xz_1} - F(xz_1z_2,z_3,1) - F(z_2,z_3,xz_1)\Big) \\
&\quad + y^2C^{[0]}(z_3,x^2z_1z_2,1) \\
F(z_1,z_2,z_3) &= C^{[1]}(z_1,z_2,z_3) - yD(xz_1z_2,z_3,1),
\end{aligned}
$$

where the initial conditions are given by

$$
\begin{aligned}
g(x,z_1,z_2,z_3) &= \frac{xz_1}{1-xz_1}\frac{xz_2}{1-xz_2}\frac{xz_3}{1-xz_3} \\
&\quad - \frac{x^2z_1z_2}{1-x^2z_1z_2}\frac{xz_3}{1-xz_3} - \frac{x^2z_2z_3}{1-x^2z_2z_3}\frac{xz_1}{1-xz_1} \\
&\quad + \frac{x^3z_1z_2z_3}{1-x^3z_1z_2z_3} \\
h(x,z_1,z_2,z_3) &= g(x,z_1,z_2,z_3) - \frac{x^2z_1z_3}{1-x^2z_1z_3}\frac{xz_2}{1-xz_2} + \frac{x^3z_1z_2z_3}{1-x^3z_1z_2z_3}.
\end{aligned}
$$

6 The Limit of $c_n^{1/n}$ Exists

We require the following well-known lemma. Since we have not found a proof in the literature, we include one here.

Lemma 1 *Suppose $b_n \geq 0$ for all sufficiently large n and let ρ be the radius of convergence of $\sum b_n x^n$. If there is a constant $C > 0$ such that for all sufficiently large n and k we have $b_{n+k} \geq Cb_nb_k$, then $\lim_{n\to\infty} b_n^{1/n}$ exists and equals $1/\rho$*

Proof Hadamard's formula for the radius of convergence states

$$\limsup_{n\to\infty} b_n^{1/n} = 1/\rho. \tag{6.1}$$

If $\rho = \infty$, then $b_n^{1/n}$ is a nonnegative sequence whose limsup is 0, and the proof is complete in this case. If $\rho < \infty$, it suffices to show that, for every $\delta > 0$, there is an $A > 0$ such that

$$b_n \geq A(\rho+\delta)^{-n} \tag{6.2}$$

for all sufficiently large n.

Let $\delta > 0$ be given. By hypothesis, there is a K such that $n, k \geq K$ implies that $b_{n+k} \geq C b_n b_k$. Since $C^{-1/t} \to 1$ and since the limsup of $\rho b_t^{1/t}$ is 1, there must be a particular $t > K$ with $\rho b_t^{1/t} > C^{-1/t}(1+\delta/\rho)^{-1}$; that is,

$$b_t > C^{-1}(\rho+\delta)^{-t}.$$

For $n \geq K$, we can find integers q, r such that $n = qt + r$ and $K \leq r < K + t$. Then

$$\begin{aligned} b_n &\geq C b_{qt} b_r \geq C^2 b_{(q-1)t} b_t b_r \cdots \geq C^q (b_t)^q b_r \\ &\geq C^q (C^{-1}(\rho+\delta)^{-t})^q b_r \\ &= (\rho+\delta)^{-n} b_r (\rho+\delta)^r. \end{aligned}$$

By taking A to be the minimum of $b_r(\rho+\delta)^r$ over $K \leq r \leq K+t$ we have achieved our goal (6.2) for all $n \geq K$. The lemma is proved. □

Theorem 1 *If c_n is the number of compositions with restrictions $R_0, \ldots, R_{m-1}$, then the radius of convergence satisfies $(1/2) \leq \rho < \infty$ and $c_n^{1/n} \sim 1/\rho$. This is also true if we let c_n be the number of such compositions where the number of parts is congruent to t modulo m.*

Proof Since there are at most 2^{n-1} locally-restricted compositions, it follows that $(1/2) \leq \rho$. On the other hand $\rho < \infty$. To see this, note that one may form exponentially many valid compositions by concatenating in any order copies of the two compositions $1, 2, \ldots, r+2$ and $2, 1, 3, \ldots, r+2$.

If two nonnegative sequences $c_n^{1/n}$ and $d_n^{1/n}$ both have limits, then so does $(c_n + d_n)^{1/n}$, and likewise for any finite number of sequences. So, we concentrate only on the case when the number of parts is constrained to be congruent to t modulo m.

We will show that for a suitable integer α, the sequence $b_n = c_{n-\alpha}$ satisfies the hypotheses of Lemma 1. Then,

$$c_n^{1/n} = (b_{n+\alpha})^{1/n} = (b_{n+\alpha})^{1/(n+\alpha)} (b_{n+\alpha})^{\alpha/n(n+\alpha)},$$

and since $b_{n+\alpha} \leq 2^{n-1}$ the theorem will be proven.

To show that $b_{n+k} \geq b_n b_k$, we glue together a composition of $n-\alpha$ and a composition of $k-\alpha$, along with a composition of α to serve as a buffer between the two, to form a composition of $n+k-\alpha$. It suffices for the buffer composition to satisfy the following conditions, where r is the maximum integer in any of the restriction sets R_i:

- all parts are distinct,
- no part equals any of the final r parts in the leading composition of n,
- no part equals any of the first r parts in the trailing composition of k,
- there are at least r parts, and

- the number of parts is congruent to $-t$ modulo m.

Let j be an integer which is at least as large as r and is congruent to $-t$ modulo m. Our buffer will have j parts. The proof is complete once we prove the following "obvious" statement.

> For every pair of integers r and j, there exists an integer α such that for every set S of integers satisfying $|S| \leq 2r$ there is a composition of α into j distinct parts which are all different from the elements of S.

The number of compositions of α into j positive parts is $\binom{\alpha-1}{j-1}$. The number of such compositions with a repeated part is bounded above by $j^2\binom{\alpha-2}{j-2}$, and so the fraction of j-part compositions with a repeated part is $O(j^3/\alpha)$. Hence, the number of j-part compositions into distinct parts is asymptotic to $\alpha^{j-1}/(j-1)!$. Suppose $S \subseteq \{1, 2, \ldots\}$ satisfies $|S| \leq 2r$. The number of compositions of α into j parts at least one of which belongs to S is bounded above by $2rj\binom{\alpha-2}{j-2}$. It follows that all α sufficiently large satisfy our requirements. □

7 Transfer Matrix Estimates

In this section, we discuss upper and lower bounds for the counting sequence c_n, whence also for the radius of convergence of the generating function $C(x, 1)$. Let there be given a set of restrictions $R_0, R_1, \ldots R_{m-1}$, and let r be at least as large as m and every member of $\cup R_i$. For integer $p > 0$ define $Lc(n, p)$ to be the number of locally restricted compositions of n in which no part exceeds p; and define $Uc(n, p)$ to be the number with no restriction on part size, but in which *only* those parts less than or equal to p are required to obey the local restrictions. Clearly,

$$Lc(n, 1) \leq Lc(n, 2) \cdots \leq c_n \leq \cdots Uc(n, 2) \leq Uc(n, 1).$$

We now show that for fixed p the sequences can be defined using a transfer matrix.

Let $I = \{1, \ldots, p\}^r$. We say that $\mathbf{i}, \mathbf{j} \in I$ are *t-compatible* if there is a locally restricted composition $a_1, \ldots, a_k$ for some $k > r$ with

- $a_\alpha \leq p$ for $1 \leq \alpha \leq k$,
- $k \equiv t \pmod{m}$,
- $i_\alpha = a_{k+\alpha-r}$ for $1 \leq \alpha < r$, and
- $j_\alpha = a_{k+\alpha-r+1}$ for $1 \leq \alpha < r$.

For each restriction set R_t we define a transfer matrix M_t by

$$M_t(\mathbf{i}, \mathbf{j}) \;=\; \begin{cases} x^{j_{r-1}} & \text{if } \mathbf{i} \text{ and } \mathbf{j} \text{ are } t\text{-compatible,} \\ 0 & \text{otherwise.} \end{cases}$$

For $t \geq m$, let $M_t = M_s$ where $t \equiv s \bmod m$. The generating function for k-part compositions with no part exceeding p is given by

$$\mathbf{u}M_{r+1}M_{r+2}\cdots M_k\mathbf{1},$$

where $\mathbf{1}$ is the all-ones column vector and the row vector $\mathbf{u}$ is given by $u_{\mathbf{i}} = x^{i_1+\cdots+i_r}$, the generating function for the composition $\mathbf{i}$. By summing on k, it is easily seen that there is a row vector $\mathbf{v}$ such that

$$C_p(x) = \sum_{k=0}^{\infty} \mathbf{v}(M_0M_1\cdots M_{m-1})^k\mathbf{1} = \mathbf{v}(I - M_0\cdots M_{m-1})^{-1}\mathbf{1}$$

counts locally restricted compositions in which no part exceeds p. It follows that the radius of convergence of $C_p(x)$ is the smallest $x > 0$ for which $\det(I - M_0\cdots M_{m-1}) = 0$. This is a strictly greater upper bound for the radius of convergence of $C(x)$.

We now count compositions in which parts not exceeding p satisfy the restrictions but parts larger than p are unconstrained. This time let $I = \{1, \ldots, p, \infty\}^r$ and remove the condition "$a_\alpha \leq p$ for $1 \leq \alpha \leq k$" in the definition of t-compatible. Define

$$M_t(\mathbf{i},\mathbf{j}) \;=\; \begin{cases} x^{j_{r-1}} & \text{if } \mathbf{i} \text{ and } \mathbf{j} \text{ are } t\text{-compatible and } j_{r-1} \leq p, \\ \dfrac{x^{p+1}}{1-x^{p+1}} & \text{if } \mathbf{i} \text{ and } \mathbf{j} \text{ are } t\text{-compatible and } j_{r-1} > p, \\ 0 & \text{otherwise,} \end{cases}$$

and proceed as before, this time obtaining a lower bound for the radius of convergence.

8 Consequences of Plausible Assumptions

The results in this section are based on certain plausible assumptions.

Iterating the recursions leads to a system of linear equations for the $C_\pi^{[t]}(x, y, 1, \ldots, 1)$ similar (3.2). Solving produces a formula for $C_\pi^{[t]}(x, y, 1, \ldots, 1)$ of the form $g(x, y)/h(x, y)$. Let ρ be the smallest positive zero of $h(x, 1)$. We believe that $g(x, 1)$ and $h(x, 1)$ have radii of convergence exceeding ρ, that ρ is a simple zero of $h(x, 1)$ and that $h(x, 1)$ has no other zeros of magnitude ρ. This leads us to the following conjecture.

Conjecture 1 *Fix sets $S_0, \ldots, S_{m-1}$. For $0 \leq i < m$, let $R_j = S_{j+i}$, where subscripts are interpreted modulo m. Let $c_n(i, t)$ be the number of locally restricted compositions of n with restrictions $R_0, \ldots, R_{m-1}$ and number of parts congruent to t modulo m. Then $c_n(i, t) \sim B(i, t)\rho^{-n}$ for some $B(i, t)$ and ρ depending on the $S_0, \ldots, S_{m-1}$.*

The part of the conjecture that implies that the radius of convergence does not change with i or t can be proved by prepending (and possibly appending) parts as done with compositions of α in the proof of Theorem 1.

We also believe

Conjecture 2 *The distribution of the number of parts in randomly selected compositions is asymptotically normal with mean and variance proportional to n.*

If we knew enough about the functions $g(x,y)$ and $h(x,y)$ (introduced before Conjecture 1) near $y=1$, we could apply Theorem 1 of [1] with $f(z,w) = C_\pi^{[t]}(z,w,1,\ldots,1)$ in that paper.

We conclude with a conditional theorem on largest part size.

Theorem 2 *If Conjecture 1 is valid, then the largest part of almost all locally restricted compositions of n is asymptotic to $\log_{1/\rho} n$.*

Proof Let $c_n(t)$ be the number of locally-restricted compositions of n with number of parts congruent to t modulo m. From Conjecture 1, there are constants C and D such that $C\rho^{-n} < c_n(t) < D\rho^{-n}$ for all t.

We now prove a lower bound asymptotic to $\log_{1/\rho} n$. Let $b_n(t,k)$ be the number of locally restricted compositions of n with number of parts congruent to t modulo m and all parts less than or equal to k. By adding k to one of the parts in such a composition, we obtain a locally restricted composition of $n+k$ and those we obtain are distinct. Since there are at least n/k parts in the composition of n, we have

$$b_n(t,k)(n/k) \;<\; c_{n+k}(t-k) \;<\; D(1/\rho)^{n+k}$$

and so

$$b_n(t,k) \;<\; D(1/\rho)^{n+k}(k/n) \;=\; o(c_n(t))$$

provided $k\rho^{-k} = o(n)$, which happens for some $k \sim \log_{1/\rho} n$.

We now prove an upper bound for the number of restricted compositions of n with $p > 0$ parts larger than k, and with number of parts equal to t modulo m. Such a compositions can be decomposed into: a composition of n_0, first of the p large parts, a composition of n_1, and so forth, ending with the pth and final large part followed by a composition of n_p. Supposing the p large parts to sum to $s+pk$, $s>0$, we may construct such a composition as follows:

- The p large parts can be chosen in $\binom{s-1}{p-1}$ ways, as can be seen by subtracting k from each of the parts, leaving an unrestricted composition of s having exactly p parts.
- Choose $t_0, t_1, \ldots t_p$ such that $\sum t_i + p$ is congruent to t modulo m, and $0 \le t_i < m$. This can be done in at most m^p ways.
- Select non-negative integers $n_0, \ldots, n_p$ summing to $n-s-pk$. This can be done in less than n^p ways.
- For each $n_i > 0$, choose a locally restricted composition of n_i, whose number of parts is congruent to t_i modulo m; this can be done in less than $D\rho^{-n_i}$ ways. For $n_i = 0$, choose the empty composition. The restrictions used are a cyclic shift of $R_0, \ldots, R_{m-1}$ by i plus the number of parts in the previously chosen compositions.

While not all such compositions satisfy the restrictions, any composition satisfying them has this form. Thus an upper bound is

$$m^p \binom{s-1}{p-1} n^p \prod_{i=0}^{p} D\rho^{-n_i} \;=\; m^p \binom{s-1}{p-1} n^p D^{p+1}(1/\rho)^{n-s-pk} \;=\; \frac{mnD^2}{\rho^{n-s-k}} \binom{s-1}{p-1} (mnD\rho^k)^{p-1}.$$

We sum this first on $p > 0$ and then on $s \geq 1$ obtaining

$$\frac{mnD^2}{\rho^{n-k}} \sum_{s \geq 1} \rho^s (1 + mnD\rho^k)^{s-1} \;\leq\; \frac{mnD^2\rho}{\rho^{n-k}} \frac{1}{1 - \rho(1 + mnD\rho^k)},$$

provided k is large enough so that $\rho(1 + mnD\rho^k) < 1$. Since $c_n > C/\rho^n$, the fraction of locally restricted compositions with at least one part exceeding k is bounded above by

$$\frac{mnD^2\rho^k}{C(1 - \rho(1 + mnD\rho^k)))}.$$

This is $o(1)$ for some $k \sim \log_{1/\rho} n$. Thus we have the desired upper bound. □

Remark. If, instead of Conjecture 1, we assume an asymptotic formula of the form

$$c_n(i,t) \;\sim\; B(i,t) n^\alpha \rho^{-n},$$

then the proof of the lower bound still goes through, but the upper bound must be multiplied by $(1 + \alpha)$ if $\alpha > 0$. If the functions $g(x, 1)$ and $h(x, 1)$ whose quotient equals the generating function $C_\pi^{[t]}(x, 1, 1, \ldots 1)$ are analytic in a circle whose radius is larger than ρ, then we will have the above where integer α is one less than the order of the zero of $h(x, 1)$ at $x = \rho$.

9 Numerical Calculations

It is straightforward to calcuate c_n for small n using exhaustion: run systematically through all compositions and check each individually to see which counts it contributes to. We did this for $n = 15$, and for these restrictions: Carlitz, distance-2, 2-rowed, and 3-rowed. The systems of equations presented in Sections 3 through 5 were programed in `maple`. Computing the coefficients c_n from these equations, and reconciling them with the values computed by exhaustion, provides a check on the equations themselves. It was decided to extend the tables to larger n, $n = 100$ being taken as a goal. This exceeded the capacity of our `maple` program, and so a C-program was developed. The final table, showing the number of 3-rowed compositions of n, required just under three hours of computation. The program uses $O(N^4)$-space and $O(N^5)$ time, where c_n is computed for $n \leq N$. The large integer issue is handled by computing mod p for three large primes p, and assembling the results with the Chinese Remainder Theorem.

Our results are shown in tables at the end of the paper. They include the ratios c_{n-1}/c_n, and lower/upper bounds for the radii of convergence ρ. The ratios c_{n-1}/c_n have been rounded; the values $\rho_{\min}$ and $\rho_{\max}$ have been rounded in the conservative direction — truncation for the min and rounding up for the max.

The bounds on ρ for Carlitz compositions are due to Knopfmacher and Prodinger [3]. The transfer-matrix method described in Section 7. was used to compute the other bounds and to confirm the Knopfmacher and Prodinger bounds. For the first three cases, we used

$p = 20$. One is solving the equation $\det(I - M) = 0$ for x, where the entries of M are functions of x. In the fourth case, 3-rowed compositions, the size of the matrices is $p^3 \times p^3$ and so we used only $p = 8$ in that case. The poorer quality of these bounds is evident.

We used c_{n-1}/c_n to obtain an estimate r for ρ. We found that $c_k \rho^k$ appears to be converging, thus providing some additional support for Conjecture 1. In this way, we obtained

$$\begin{aligned}
c_n &\approx 0.4564(0.57135)^{-n} \quad \text{for Carlitz compositions,} \\
c_n &\approx 0.5273(0.61977)^{-n} \quad \text{for distance-2 compositions,} \\
c_n &\approx 0.2485(0.59024)^{-n} \quad \text{for 2-rowed compositions,} \\
c_n &\approx 0.1932(0.59598)^{-n} \quad \text{for 3-rowed compositions.}
\end{aligned}$$

The estimate for Carlitz compositions agrees with the known asymptotics cited in Section 2.

References

[1] E. A. Bender, Central and local limit theorems applied to asymptotic enumeration, *J. Combin Theory Ser. A* **15** (1973) 91–111.

[2] P. Hitczenko and G. Louchard, Distinctness of compositions of an integer: A probabilistic analysis, *Random Structures and Algorithms* **19** (2001) 407–437.

[3] A. Knopfmacher and H. Prodinger, On Carlitz compositions, *European J. Combin.* **19** (1998) 579–589.

[4] G. Louchard and H. Prodinger, Probabilistic analysis of Carlitz compositions, *Discrete Math. and Theoret. Comput. Sci.* **5** (2002) 71–96.

type	Carlitz		distance-2	
n	c_n	c_{n-1}/c_n	c_n	c_{n-1}/c_n
1	1		0	
2	1	1.00000	0	
3	3	0.33333	2	0.00000
4	4	0.75000	2	1.00000
5	7	0.57143	4	0.50000
6	14	0.50000	10	0.40000
7	23	0.60870	14	0.71429
8	39	0.58974	22	0.63636
9	71	0.54930	36	0.61111
10	124	0.57258	66	0.54545
11	214	0.57944	100	0.66000
12	378	0.56614	164	0.60976
13	661	0.57186	264	0.62121
14	1152	0.57378	418	0.63158
15	2024	0.56917	690	0.60580
20	33202	0.57159	7514	0.61964
30	8958772	0.57135	901398	0.61969
40	2416728950	0.57135	107825100	0.61977
50	651939286323	0.57135	12895364474	0.61977
60	175867831235778	0.57135	1542229841220	0.61977
70	47442292097138542	0.57135	184443985682928	0.61977
80	12798082875707215288	0.57135	22058697837971950	0.61977
90	3452424367654374081818	0.57135	2638124253466256468	0.61977
100	931329647583272532815226	0.57135	315508178730370139526	0.61977
$\rho_{\min}$		0.57134		0.61976
$\rho_{\max}$		0.57135		0.61980

type	2-rowed		3-rowed	
n	c_n	c_{n-1}/c_n	c_n	c_{n-1}/c_n
1	0		0	
2	0		0	
3	2	0.00000	0	
4	2	1.00000	1	0.00000
5	4	0.50000	2	0.50000
6	6	0.66667	7	0.28571
7	10	0.60000	9	0.77778
8	16	0.62500	15	0.60000
9	26	0.61538	23	0.65217
10	54	0.48148	34	0.67647
11	80	0.67500	53	0.64151
12	134	0.59701	84	0.63095
13	240	0.55833	159	0.52830
14	400	0.60000	261	0.60920
15	668	0.59880	466	0.56009
20	9442	0.58801	5953	0.61011
30	1837916	0.59017	1069149	0.59618
40	358124594	0.59023	189188124	0.59592
50	69784269504	0.59024	33462077542	0.59598
60	13598215211918	0.59024	5918452340693	0.59598
70	2649758685280706	0.59024	1046805875625889	0.59598
80	516334018390386674	0.59024	185150129419219640	0.59598
90	100613244517499718346	0.59024	32747774015862230379	0.59598
100	1960557432144409293730 8	0.59024	5792147020120051788530	0.59598
$\rho_{\min}$		0.59023		0.52220
$\rho_{\max}$		0.59024		0.60266

TRANSACTIONS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 00, Number 0, Xxxx XXXX, Pages 000–000
S 0002-9947(XX)0000-0

SPECIAL VALUES OF MULTIPLE POLYLOGARITHMS

JONATHAN M. BORWEIN, DAVID M. BRADLEY, DAVID J. BROADHURST, AND PETR LISONĚK

Abstract. Historically, the polylogarithm has attracted specialists and non-specialists alike with its lovely evaluations. Much the same can be said for Euler sums (or multiple harmonic sums), which, within the past decade, have arisen in combinatorics, knot theory and high-energy physics. More recently, we have been forced to consider multidimensional extensions encompassing the classical polylogarithm, Euler sums, and the Riemann zeta function. Here, we provide a general framework within which previously isolated results can now be properly understood. Applying the theory developed herein, we prove several previously conjectured evaluations, including an intriguing conjecture of Don Zagier.

1. Introduction

We are going to study a class of multiply nested sums of the form

$$\lambda\binom{s_1,\ldots,s_k}{b_1,\ldots,b_k} := \sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} b_j^{-\nu_j} \left(\sum_{i=j}^{k} \nu_i\right)^{-s_j}, \tag{1.1}$$

and which we shall refer to as *multiple polylogarithms*. When $k = 0$, we define $\lambda(\{\}) := 1$, where $\{\}$ denotes the empty string. When $k = 1$, note that

$$\lambda\binom{s}{b} = \sum_{\nu=1}^{\infty} \frac{1}{\nu^s b^\nu} = \mathrm{Li}_s\left(\frac{1}{b}\right) \tag{1.2}$$

is the usual polylogarithm [49, 50] when s is a positive integer and $|b| \geq 1$. Of course, the polylogarithm (1.2) reduces to the Riemann zeta function [26, 43, 65]

$$\zeta(s) = \sum_{\nu=1}^{\infty} \frac{1}{\nu^s}, \quad \Re(s) > 1, \tag{1.3}$$

when $b = 1$. More generally, for any $k > 0$ the substitution $n_j = \sum_{i=j}^{k} \nu_i$ shows that our multiple polylogarithm (1.1) is related to Goncharov's [35] by the equation

$$\mathrm{Li}_{s_k,\ldots,s_1}(x_k,\ldots,x_1) = \lambda\binom{s_1,\ldots,s_k}{y_1,\ldots,y_k}, \quad \text{where} \quad y_j := \prod_{i=1}^{j} x_i^{-1},$$

Received by the editors July 10, 1998. Revised August 9, 1999.
1991 *Mathematics Subject Classification.* Primary: 40B05, 33E20; Secondary: 11M99, 11Y99.
Key words and phrases. Euler sums, Zagier sums, multiple zeta values, polylogarithms, multiple harmonic series, quantum field theory, knot theory, Riemann zeta function.
The research of the first author was supported by NSERC and the Shrum Endowment of Simon Fraser University.

and

$$\mathrm{Li}_{s_k,\ldots,s_1}(x_k,\ldots,x_1) := \sum_{n_1>\cdots>n_k>0} \prod_{j=1}^{k} n_j^{-s_j} {x_j}^{n_j}. \tag{1.4}$$

With each $x_j = 1$, these latter sums (sometimes called *"Euler sums"*), have been studied previously at various levels of generality [2, 6, 7, 9, 13, 14, 15, 16, 31, 38, 39, 42, 51, 59], the case $k = 2$ going back to Euler [27]. Recently, Euler sums have arisen in combinatorics (analysis of quad-trees [30, 46] and of lattice reduction algorithms [23]), knot theory [14, 15, 16, 47], and high-energy particle physics [13] (quantum field theory). There is also quite sophisticated work relating polylogarithms and their generalizations to arithmetic and algebraic geometry, and to algebraic K-theory [4, 17, 18, 33, 34, 35, 66, 67, 68].

In view of these recent applications and the well-known fact that the classical polylogarithm (1.2) often arises in physical problems via the multiple integration of rational forms, one might expect that the more general multiple polylogarithm (1.1) would likewise find application in a wide variety of physical contexts. Nevertheless, lest it be suspected that the authors have embarked on a program of generalization for its own sake, let the reader be assured that it was only with the greatest reluctance that we arrived at the definition (1.1). On the one hand, the polylogarithm (1.2) has traditionally been studied as a function of b with the positive integer s fixed; while on the other hand, the study of Euler sums has almost exclusively focused on specializations of the nested sum (1.4) in which each $x_j = \pm 1$. However, we have found, in the course of our investigations, that a great deal of insight is lost by ignoring the interplay between these related sums when both sequences of parameters are permitted to vary. Indeed, it is our view that it is *impossible* to fully understand the sums (1.2–1.4) without viewing them as members of a broader class of multiple polylogarithms.

That said, one might legitimately ask why we chose to adopt the notation (1.1) in favour of Goncharov's (1.4), inasmuch as the latter is a direct generalization of the Li_n notation for the classical polylogarithm. As a matter of fact, the notation (1.4) (with argument list reversed) was our original choice. However, as we reluctantly discovered, it turns out that the notation (1.1), in which the second row of parameters comprises the reciprocated running product of the argument list in (1.4), is more suitable for our purposes here. In particular, our "running product" notation (1.1), in addition to simplifying the iterated integral representation (4.9) (cf. [33] Theorem 16) and the various duality formulae (Section 6—see eg. equations (6.7) and (6.8)), brings out much more clearly the relationship (Subsection 5.3) between the partition integral (Subection 4.1), in which running products necessarily arise in the integrand; and "stuffles" (Subsections 5.1, 5.2). It seems also that boundary cases of certain formulae for alternating sums must be treated separately unless running product notation is used. Theorem 8.5 with $n = 0$ (Section 8) provides an example of this.

Don Zagier (see eg. [69]) has argued persuasively in favour of studying special values of zeta functions at integer arguments, as these values "often seem to dictate the most important properties of the objects to which the zeta functions are associated." It seems appropriate, therefore, to focus on the values the multiple polylogarithms (1.1) take when the s_j are restricted to the set of positive integers, despite the fact that the sums (1.1) and their special cases have a rich structure as

analytic functions of the complex variables s_j. However, we allow the parameters b_j to take on complex values, with each $|b_j| \geq 1$ and $(b_1, s_1) \neq (1,1)$ to ensure convergence.

Their importance notwithstanding, we feel obliged to confess that our interest in special values extends beyond mere utilitarian concerns. Lewin [49] (p. xi) writes of a "school-boy fascination" with certain numerical results, an attitude which we whole-heartedly share. In the hope that the reader might also be convinced of the intrinsic beauty of the subject, we offer two modest examples. The first [38, 47],

$$\sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} \frac{1}{(\nu_j + \cdots + \nu_k)^2} = \frac{\pi^{2k}}{(2k+1)!}, \qquad 0 \leq k \in \mathbf{Z},$$

generalizes Euler's celebrated result

$$\zeta(2) = \sum_{\nu=1}^{\infty} \frac{1}{\nu^2} = \frac{\pi^2}{6},$$

and is extended to all even positive integer arguments in [7]. The second (see Corollary 1 of Section 8),

$$\begin{aligned} \sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} \frac{(-1)^{\nu_j+1}}{\nu_j + \cdots + \nu_k} &= \sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} \frac{1}{2^{\nu_j}(\nu_j + \cdots + \nu_k)} \\ &= \frac{(\log 2)^k}{k!}, \qquad 0 \leq k \in \mathbf{Z} \end{aligned}$$

can be viewed as a multidimensional extension of the elementary "dual" Maclaurin series evaluations

$$\sum_{\nu=1}^{\infty} \frac{(-1)^{\nu+1}}{\nu} = \sum_{\nu=1}^{\infty} \frac{1}{\nu 2^{\nu}} = \log 2,$$

and leads to deeper questions of duality (Section 6) and computational issues related to series acceleration (Section 7). We state additional results in the next section and outline connections to combinatorics and q-series. In Section 4, we develop several different integral representations, which are then used in subsequent sections to classify various types of identities that multiple polylogarithms satisfy. Sections 8 through 11 conclude the paper with proofs of previously conjectured evaluations, including an intriguing conjecture of Zagier [69] and its generalization.

2. Definitions and Additional Examples

A useful specialization of the general multiple polylogarithm (1.1), which is at the same time an extension of the polylogarithm (1.2), is the case in which each $b_j = b$. Under these circumstances, we write

$$\lambda_b(s_1,\ldots,s_k) := \lambda\binom{s_1,\ldots,s_k}{b,\ldots,b} = \sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} b^{-\nu_j} \left(\sum_{i=j}^{k} \nu_i\right)^{-s_j}, \tag{2.1}$$

and distinguish the cases $b = 1$ and $b = 2$ with special symbols:

$$\zeta := \lambda_1 \quad \text{and} \quad \delta := \lambda_2. \tag{2.2}$$

The latter δ-function represents an iterated sum extension of the polylogarithm (1.2) with argument one-half, and will play a crucial role in computational issues (Section 7) and "duality" identities such as (1). The former coincides with (1.4) when $k > 0$, each $x_j = 1$, and the order of the argument list is reversed, and hence can be viewed as a multidimensional unsigned Euler sum. We will follow Zagier [69] in referring to these as *"multiple zeta values"* or *"MZVs"* for short. By specifying each $b_j = \pm 1$ in (1.1), *alternating Euler sums* [7] are recovered, and in this case, it is convenient to combine the strings of exponents and signs into a single string with s_j in the jth position when $b_j = +1$, and s_j- in the jth position when $b_j = -1$. To avoid confusion, it should be also noted that in [7] the alternating Euler sums were studied using the notation

$$\zeta(s_1, \ldots, s_k) := \sum_{n_1 > \cdots > n_k > 0} \prod_{j=1}^{k} n_j^{-|s_j|} \sigma_j^{-n_j}$$

where $s_1, \ldots, s_k$ are non-zero integers and $\sigma_j := \mathrm{signum}(s_j)$.

Additionally, n repetitions of a substring U will be denoted by U^n. Thus, for example,

$$\lambda(\{2-,1\}^n) := \lambda\begin{pmatrix} 2,1,\ldots,2,1 \\ -1,1,\ldots,-1,1 \end{pmatrix} = \sum_{\nu_1,\ldots,\nu_{2n}=1}^{\infty} \prod_{j=1}^{n} \frac{(-1)^{\nu_{2j-1}}}{\left(\sum_{i=2j-1}^{k} \nu_i\right)^2 \left(\sum_{i=2j}^{k} \nu_i\right)}.$$

Unit Euler sums, that is those sums (1.1) in which each $s_j = 1$, are also important enough to be given a distinctive notation. Accordingly, we define

$$\mu(b_1, \ldots, b_k) := \lambda\begin{pmatrix} 1,\ldots,1 \\ b_1,\ldots,b_k \end{pmatrix} = \sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} b_j^{-\nu_j} \left(\sum_{i=j}^{k} \nu_i\right)^{-1}. \tag{2.3}$$

To entice the reader, we offer a small but representative sample of evaluations below.

Example 2.1. Euler showed that

$$\zeta(2,1) = \sum_{n=1}^{\infty} \frac{1}{n^2} \sum_{k=1}^{n-1} \frac{1}{k} = \sum_{n=1}^{\infty} \frac{1}{n^3} = \zeta(3),$$

and more generally [27, 59], that

$$2\zeta(m,1) = m\zeta(m+1) - \sum_{k=1}^{m-2} \zeta(m-k)\zeta(k+1), \quad 2 \leq m \in \mathbf{Z}.$$

The continued interest in Euler sums is evidenced by the fact that a recent American Mathematical Monthly problem [28] effectively asks for the proof of $\zeta(2,1) = \zeta(3)$.

Two examples of non-alternating, arbitrary depth evaluations for all nonnegative integers n are provided by

Example 2.2.

$$\zeta(\{3,1\}^n) = 4^{-n}\zeta(\{4\}^n) = \frac{2\pi^{4n}}{(4n+2)!},$$

previously conjectured by Don Zagier [69] and proved herein (see Section 11); and

Example 2.3.

$$\zeta(2,\{1,3\}^n) = 4^{-n}\sum_{k=0}^{n}(-1)^k\zeta(\{4\}^{n-k})\Bigg\{(4k+1)\zeta(4k+2) - 4\sum_{j=1}^{k}\zeta(4j-1)\zeta(4k-4j+3)\Bigg\},$$

conjectured in [7] and proved by Bowman and Bradley [11].

Example 2.4. An intriguing two-parameter, arbitrary depth evaluation involving alternations, conjectured in [7] and proved herein (see Section 8), is

$$\begin{aligned}\mu(\{-1\}^m,1,\{-1\}^n) &= (-1)^{m+1}\sum_{k=0}^{m}\binom{n+k}{n}A_{k+n+1}P_{m-k}\\ &\quad+(-1)^{n+1}\sum_{k=0}^{n}\binom{m+k}{m}Z_{k+m+1}P_{n-k},\end{aligned}\tag{2.4}$$

where

$$A_r := \mathrm{Li}_r(\tfrac{1}{2}) = \delta(r) = \sum_{k=1}^{\infty}\frac{1}{2^k k^r},\quad P_r := \frac{(\log 2)^r}{r!},\quad Z_r := (-1)^r\zeta(r).\tag{2.5}$$

The formula (2.4) is valid for all nonnegative integers m and n if the divergent $m=0$ case is interpreted appropriately.

Example 2.5. If the s_j are all nonpositive integers, then

$$\Bigg(\sum_{i=j}^{k}\nu_i\Bigg)^{-s_j} = D_j\,\exp\Bigg(-u_j\sum_{i=j}^{k}\nu_i\Bigg),\quad D_j := \Bigg(-\frac{d}{du_j}\Bigg)^{-s_j}\Bigg|_{u_j=0}.$$

Consequently,

$$\begin{aligned}\lambda\binom{s_1,\ldots,s_k}{b_1,\ldots,b_k} &= \sum_{\nu_1,\ldots,\nu_k=1}^{\infty}\prod_{j=1}^{k}b_j^{-\nu_j}D_j\exp\Bigg(-u_j\sum_{i=j}^{k}\nu_i\Bigg)\\ &= \prod_{j=1}^{k}D_j\sum_{\nu_j=1}^{\infty}b_j^{-\nu_j}\exp\Bigg(-\nu_j\sum_{i=1}^{j}u_i\Bigg)\\ &= \prod_{j=1}^{k}D_j\Bigg\{\frac{1}{b_j\exp\left(\sum_{i=1}^{j}u_i\right)-1}\Bigg\}.\end{aligned}\tag{2.6}$$

In particular, (2.6) implies

$$\lambda\binom{0,\ldots,0}{b_1,\ldots,b_k} = \prod_{j=1}^{k}\frac{1}{b_j-1}.\tag{2.7}$$

Despite its utter simplicity, (2.7) points the way to deeper waters. For example, if we put $b_j = q^{-j}$ for each $j=1,2,\ldots,k$ and note that

$$\lambda\binom{0,0,\ldots,0}{q^{-1},q^{-2},\ldots,q^{-k}} = \sum_{n_1>n_2>\cdots>n_k>0}\prod_{j=1}^{k}q^{n_j},\quad k>0,$$

then (2.7) implies the generating function equality

$$\sum_{k=0}^{\infty} z^k \lambda\binom{0,0,\ldots,0}{q^{-1},q^{-2},\ldots,q^{-k}} = \prod_{n=1}^{\infty}(1+zq^n) = \sum_{k=0}^{\infty} z^k \prod_{j=1}^{k}\frac{q^j}{1-q^j},$$

which experts in the field of basic hypergeometric series will recognize as a q-analogue of the exponential function and a special case of the q-binomial theorem, usually expressed in the more familiar form [32] as

$$(-zq;q)_\infty = \sum_{k=0}^{\infty}\frac{q^{k(k+1)/2}}{(q;q)_k}z^k.$$

The case $k=1$, $b_1=2$, $s_1=-n$ of (2.6) yields the numbers [63] (A000629)

$$\delta(-n) = \lambda_2(-n) = \sum_{k=1}^{\infty}\frac{k^n}{2^k} = \mathrm{Li}_{-n}(\tfrac{1}{2}), \qquad 0 \le n \in \mathbf{Z}, \tag{2.8}$$

which enumerate [45] the combinations of a simplex lock having n buttons, and which satisfy the recurrence

$$\delta(-n) = 1 + \sum_{j=0}^{n-1}\binom{n}{j}\delta(-j), \qquad 1 \le n \in \mathbf{Z}.$$

Also, from the exponential generating function

$$\sum_{n=0}^{\infty}\delta(-n)\frac{x^n}{n!} = \frac{e^x}{2-e^x} = \frac{2}{2-e^x} - 1,$$

we infer [36, 64] that for $n \ge 1$, $\frac{1}{2}\delta(-n)$ also counts

- the number of ways of writing a sum on n indices;
- the number of functions $f : \{1,2,\ldots,n\} \to \{1,2,\ldots,n\}$ such that if j is in the range of f, then so is each value less than or equal to j;
- the number of asymmetric generalized weak orders on $\{1,2,\ldots,n\}$;
- the number of ordered partitions (preferential arrangements) of $\{1,2,\ldots,n\}$.

The numbers $\frac{1}{2}\delta(-n)$ also arise [24] in connection with certain constants related to the Laurent coefficients of the Riemann zeta function. See [63] (A000670) for additional references.

3. Reductions

Given the multiple polylogarithm (1.1), we define the *depth* to be k, and the *weight* to be $s := s_1 + \cdots + s_k$. We would like to know which sums can be expressed in terms of lower depth sums. When a sum can be so expressed, we say it *reduces*. Especially interesting are the sums which completely reduce, i.e. can be expressed in terms of depth-1 sums. We say such sums *evaluate*. The concept of weight is significant, as all our reductions preserve it. More specifically, we'll see that all our reductions take the form of a polynomial expression which is homogeneous with respect to weight.

There are certain sums which evidently cannot be expressed (polynomially) in terms of lower depth sums. Such sums are called "irreducible". Proving irreducibility is currently beyond the reach of number theory. For example, proving the irrationality of expressions like $\zeta(5,3)/\zeta(5)\zeta(3)$ or $\zeta(5)/\zeta(2)\zeta(3)$ seems to be impossible with current techniques.

3.1. **Examples of Reductions at Specific Depths.** The functional equation (an example of a "stuffle" – see Sections 5.1 through 5.3)

$$\zeta(s)\zeta(t) = \zeta(s,t) + \zeta(t,s) + \zeta(s+t)$$

reduces $\zeta(s,s)$.

One of us (Broadhurst), using high-precision arithmetic and integer relations finding algorithms, has found many conjectured reductions. One example is

$$\zeta(4,1,3) = -\zeta(5,3) + \tfrac{71}{36}\zeta(8) - \tfrac{5}{2}\zeta(5)\zeta(3) + \tfrac{1}{2}\zeta(3)^2\zeta(2), \tag{3.1}$$

which expresses a multiple zeta value of depth three and weight eight in terms of lower depth MZVs, and which was subsequently proved. Observe that the combined weight of each term in the reduction (3.1) is preserved. The easiest proof of (3.1) uses Minh and Petitot's basis of order eight [55].

Broadhurst also noted that although $\zeta(4,2,4,2)$ is apparently irreducible in terms of lower depth MZVs, we have the conjectured[1] weight-12 reduction

$$\begin{aligned}\zeta(4,2,4,2) \stackrel{?}{=} &-\tfrac{1024}{27}\lambda(9-,3) - \tfrac{267991}{5528}\zeta(12) - \tfrac{1040}{27}\zeta(9,3) - \tfrac{76}{3}\zeta(9)\zeta(3)\\ &- \tfrac{160}{9}\zeta(7)\zeta(5) + 2\zeta(6)\zeta(3)^2 + 14\zeta(5,3)\zeta(4)\\ &+ 70\zeta(5)\zeta(4)\zeta(3) - \tfrac{1}{6}\zeta(3)^4\end{aligned} \tag{3.2}$$

in terms of lower depth MZVs *and* the alternating Euler sum $\lambda(9-,3)$. Thus, alternating Euler sums enter quite naturally into the analysis. And once the alternating sums are admitted, we shall see that more general polylogarithmic sums are required.

We remark that the depth-two sums in (3.2), namely $\lambda(9-,3)$, $\zeta(9,3)$, and $\zeta(5,3)$, are almost certainly irreducible. For example, if there are integers c_1, c_2, c_3, c_4 (not all equal to 0) such that $c_1\zeta(5,3)+c_2\pi^8+c_3\zeta(3)^2\zeta(2)+c_4\zeta(5)\zeta(3)=0$, then the Euclidean norm of the vector (c_1,c_2,c_3,c_4) is greater than 10^{50}. This result can be proved computationally in a mere 0.2 seconds on a DEC Alpha workstation using D. Bailey's fast implementation of the integer relation algorithm PSLQ [29], once we know the four input values at the precision of 200 decimal digits. Such evaluation poses no obstacle to our fast method of evaluating polylogs using the Hölder convolution (see Section 7).

3.2. **An Arbitrary Depth Reduction.** In contrast to the specific numerical results provided by (3.1) and (3.2), reducibility results for arbitrary sets of arguments can be obtained if one is prepared to consider certain specific combinations of MZVs. The following result is typical in this respect. It states that, depending on the parity of the depth, either the sum or the difference of an MZV with its reversed-string counterpart always reduces. Additional reductions, such as those alluded to in Sections 1 and 2, must await the development of the theory provided in Sections 4–7.

Theorem 3.1. *Let k be a positive integer and let $s_1, s_2, \ldots, s_k$ be positive integers with s_1 and s_k greater than 1. Then the expression*

$$\zeta(s_1, s_2, \ldots, s_k) + (-1)^k \zeta(s_k, \ldots, s_2, s_1)$$

reduces to lower depth MZVs.

[1] Both sides of (3.2) agree to at least 7900 significant figures.

Remark 3.2. The condition on s_1 and s_k is imposed only to ensure convergence of the requisite sums.

Proof. Let $N := (\mathbf{Z}^+)^k$ denote the Cartesian product of k copies of the positive integers. Define an additive weight-function $w : 2^N \to \mathbf{R}$ by

$$w(A) := \sum_{\vec{n} \in A} \prod_{j=1}^{k} n_j^{-s_j},$$

where the sum is over all $\vec{n} = (n_1, n_2, \ldots, n_k) \in A \subseteq N$. For each $1 \leq j \leq k-1$, define the subset P_j of N by

$$P_j := \{\vec{n} \in N : n_j \leq n_{j+1}\}.$$

The Inclusion-Exclusion Principle states that

$$w\Big(\bigcap_{j=1}^{k-1} N \setminus P_j\Big) = \sum_{T \subseteq \{1,2,\ldots,k-1\}} (-1)^{|T|}\, w\Big(\bigcap_{j \in T} P_j\Big). \tag{3.3}$$

We remark that the term on the right-hand side of (3.3) arising from the subset $T = \{\}$ is $\zeta(s_1)\zeta(s_2)\cdots\zeta(s_k)$ by the usual convention for intersection over an empty set. Next, note that the left-hand side of (3.3) is simply $\zeta(s_1, s_2, \ldots, s_k)$. Finally, observe that all terms on the right-hand side of (3.3) have depth strictly less than k—except when $T = \{1, 2, \ldots, k-1\}$, which gives

$$(-1)^{k-1} \sum_{n_1 \leq n_2 \leq \cdots \leq n_k} \prod_{j=1}^{k} n_j^{-s_j} = (-1)^{k-1}\zeta(s_k, \ldots, s_2, s_1) \;+\; \text{lower depth MZVs.}$$

This latter observation completes the proof of Theorem 3.1. □

4. Integral Representations

Writing the definition of the gamma function [59] in the form

$$r^{-s}\Gamma(s) = \int_1^\infty (\log x)^{s-1} x^{-r-1}\, dx, \quad r > 0, \quad s > 0,$$

it follows that if each $s_j > 0$ and each $|b_j| \geq 1$, then

$$\begin{aligned}
\lambda\binom{s_1, \ldots, s_k}{b_1, \ldots, b_k} &= \sum_{\nu_1, \ldots, \nu_k = 1}^{\infty} \prod_{j=1}^{k} b_j^{-\nu_j} \Big(\sum_{i=j}^{k} \nu_i\Big)^{-s_j} \\
&= \sum_{\nu_1=1}^{\infty} \int_1^\infty \frac{(\log x)^{s_1-1}\, dx}{\Gamma(s_1) b_1^{\nu_1} x^{\nu_1+1}} \sum_{\nu_2, \ldots, \nu_k = 1}^{\infty} \prod_{j=2}^{k} (x b_j)^{-\nu_j} \Big(\sum_{i=j}^{k} \nu_i\Big)^{-s_j} \\
&= \frac{1}{\Gamma(s_1)} \int_1^\infty \frac{(\log x)^{s_1-1}}{x b_1 - 1} \lambda\binom{s_2, \ldots, s_k}{x b_2, \ldots, x b_k} \frac{dx}{x},
\end{aligned} \tag{4.1}$$

a representation vaguely remindful of the integral recurrence for the polylogarithm. Repeated application of (4.1) yields the k-dimensional integral representation

$$\lambda\binom{s_1, \ldots, s_k}{b_1, \ldots, b_k} = \int_1^\infty \cdots \int_1^\infty \prod_{j=1}^{k} \frac{(\log x_j)^{s_j-1} dx_j}{\Gamma(s_j)\big(b_j \prod_{i=1}^{j} x_i - 1\big) x_j}, \tag{4.2}$$

which generalizes Crandall's integral [20] for $\zeta(s_1,\ldots,s_k)$. An equivalent formulation of (4.2) is

$$\lambda\binom{s_1,\ldots,s_k}{b_1,\ldots,b_k} = \int_0^\infty \cdots \int_0^\infty \prod_{j=1}^{k} \frac{u_j^{s_j-1}\,du_j}{\Gamma(s_j)(b_j \exp\left(\sum_{i=1}^{j} u_i\right)-1)}, \tag{4.3}$$

the integral transforms in (4.3) replacing the derivatives in (2.6).

Although *depth-dimensional integrals* such as (4.2) and (4.3) are attractive, they are not particularly useful. As mentioned previously, we are interested in reducing the depth whenever this is possible. However, since the weight is an invariant of all known reductions, we seek integral representations which respect weight invariance. As we next show, this can be accomplished by selectively removing logarithms from the integrand of (4.2), at the expense of increasing the number of integrations. At the extreme, the representation (4.2) is replaced by a *weight-dimensional integral* of a rational function.

4.1. **The Partition Integral.** We begin with the parameters in (1.1). Let R_1, $R_2,\ldots,R_n$ be a (disjoint) set partition of $\{1,2,\ldots,k\}$. Put

$$r_m := \sum_{i\in R_m} s_i, \quad 1\le m\le n.$$

If $d_1,d_2,\ldots,d_n$ are real numbers satisfying $|d_m|\ge 1$ for all m and $r_1 d_1\neq 1$, then

$$\begin{aligned}
\lambda\binom{r_1,\ldots,r_n}{d_1,\ldots,d_n} &= \sum_{\nu_1,\ldots,\nu_n=1}^{\infty} \prod_{m=1}^{n} d_m^{-\nu_m}\left(\sum_{j=m}^{n}\nu_j\right)^{-r_m}\\
&= \sum_{\nu_1,\ldots,\nu_n=1}^{\infty} \prod_{m=1}^{n} d_m^{-\nu_m} \prod_{i\in R_m}\left(\sum_{j=m}^{n}\nu_j\right)^{-s_i}\\
&= \sum_{\nu_1,\ldots,\nu_n=1}^{\infty} \prod_{m=1}^{n} d_m^{-\nu_m} \int_1^\infty\cdots\int_1^\infty \prod_{i\in R_m}\frac{(\log x_i)^{s_i-1}\,dx_i}{\Gamma(s_i)\,x_i^{1+\nu_m+\cdots+\nu_n}}.
\end{aligned}$$

Now collect bases with like exponents and note that "$\prod_{m=1}^{n}\prod_{i\in R_m} = \prod_{j=1}^{k}$." It follows that

$$\begin{aligned}
\lambda\binom{r_1,\ldots,r_n}{d_1,\ldots,d_n} &= \int_1^\infty\cdots\int_1^\infty \left\{\sum_{\nu_1,\ldots,\nu_n=1}^{\infty}\prod_{m=1}^{n} d_m^{-\nu_m}\prod_{j=1}^{m}\prod_{i\in R_j} x_i^{-\nu_m}\right\}\\
&\qquad\times\prod_{j=1}^{k}\frac{(\log x_j)^{s_j-1}\,dx_j}{\Gamma(s_j)\,x_j}\\
&= \int_1^\infty\cdots\int_1^\infty \left\{\prod_{m=1}^{n}\left(d_m\prod_{j=1}^{m}\prod_{i\in R_j} x_i - 1\right)^{-1}\right\}\\
&\qquad\times\prod_{j=1}^{k}\frac{(\log x_j)^{s_j-1}\,dx_j}{\Gamma(s_j)\,x_j},
\end{aligned} \tag{4.4}$$

on summing the n geometric series.

Example 4.1. Taking $n=k$, we have $R_m=\{m\}$, and $r_m=s_m$ for all $1\le m\le n$. In this case, (4.4) reduces to the depth-dimensional integral representation (4.2).

Example 4.2. Taking $n = 1$, we have $R_1 = \{1, 2, \ldots, k\}$ and $r_1 = s = \sum_{j=1}^{k} s_i$. If we also put $d := \prod_{j=1}^{k} d_j$, then (4.4) yields the seemingly wasteful k-dimensional integral

$$\lambda\binom{s}{d} = \lambda\binom{\sum_{j=1}^{k} s_j}{\prod_{j=1}^{k} d_j} = \int_1^\infty \cdots \int_1^\infty \left(\prod_{j=1}^{k} d_j x_j - 1 \right)^{-1} \prod_{j=1}^{k} \frac{(\log x_j)^{s_j - 1}\, dx_j}{\Gamma(s_j)\, x_j}$$

for a polylogarithm of depth one.

Example 4.3. Let $s_j = 1$ for each $1 \le j \le k$, $r_0 = 0$ and let $r_1, r_2, \ldots, r_n$ be arbitrary positive integers with $\sum_{m=1}^{n} r_m = k$. For $1 \le m \le n$ define

$$R_m := \bigcup_{j=1}^{r_m} \left\{ j + \sum_{i=1}^{m-1} r_i \right\}.$$

In this case, (4.4) yields a weight-dimensional integral of a rational function in k variables:

$$\lambda\binom{r_1, \ldots, r_n}{d_1, \ldots, d_n} = \int_1^\infty \cdots \int_1^\infty \left\{ \prod_{m=1}^{n} \left(d_m \prod_{i=1}^{u_m} x_i - 1 \right)^{-1} \right\} \prod_{j=1}^{u_n} \frac{dx_j}{x_j}, \tag{4.5}$$

where $u_m = \sum_{i=1}^{m} r_i$. An interesting specialization of (4.5) is

$$\begin{aligned} \zeta(2,1) &= \int_1^\infty \int_1^\infty \int_1^\infty \frac{dx\, dy\, dz}{xyz(xy-1)(xyz-1)} = \int_1^\infty \int_1^\infty \int_1^\infty \frac{dx\, dy\, dz}{xyz(xyz-1)} \\ &= \zeta(3). \end{aligned}$$

Although it may seem wasteful, as in Example 4.1 above, to use more integrations than are required, nevertheless such a technique allows an easy comparison of multiple polylogarithms having a common weight but possessing widely differing depths. For example, from the four equations

$$\begin{aligned} \lambda\binom{s+t}{ab} &= \frac{1}{\Gamma(s)\Gamma(t)} \int_1^\infty \int_1^\infty \frac{(\log x)^{s-1} (\log y)^{t-1}\, dx\, dy}{(abxy-1)xy}, \\ \lambda\binom{s,t}{a,ab} &= \frac{1}{\Gamma(s)\Gamma(t)} \int_1^\infty \int_1^\infty \frac{(\log x)^{s-1} (\log y)^{t-1}\, dx\, dy}{(ax-1)(abxy-1)xy}, \\ \lambda\binom{t,s}{b,ab} &= \frac{1}{\Gamma(s)\Gamma(t)} \int_1^\infty \int_1^\infty \frac{(\log x)^{s-1} (\log y)^{t-1}\, dx\, dy}{(by-1)(abxy-1)xy}, \\ \lambda\binom{s}{a}\lambda\binom{t}{b} &= \frac{1}{\Gamma(s)\Gamma(t)} \int_1^\infty \int_1^\infty \frac{(\log x)^{s-1} (\log y)^{t-1}\, dx\, dy}{(ax-1)(by-1)xy}, \end{aligned} \tag{4.6}$$

and the rational function identity

$$\frac{1}{(ax-1)(by-1)} = \frac{1}{abxy-1} \left(\frac{1}{ax-1} + \frac{1}{by-1} + 1 \right), \tag{4.7}$$

the "stuffle" identity (see Section 5.1)

$$\lambda\binom{s}{a}\lambda\binom{t}{b} = \lambda\binom{s,t}{a,ab} + \lambda\binom{t,s}{b,ab} + \lambda\binom{s+t}{ab} \tag{4.8}$$

follows immediately. The connection between "stuffle" identities and rational functions will be explained and explored more fully in Section 5.3.

4.2. **The Iterated Integral.** A second approach to removing the logarithms from the depth-dimensional integral representation (4.2) yields a weight-dimensional iterated integral. The advantage here is that the rational function comprising the integrand is particularly simple.

We use the notation of Kassel [44] for iterated integrals. For $j = 1, 2, \ldots, n$, let $f_j : [a, c] \to \mathbf{R}$ and $\Omega_j := f_j(y_j)\, dy_j$. Then

$$\begin{aligned} \int_a^c \Omega_1\Omega_2\cdots\Omega_n &:= \prod_{j=1}^{n} \int_a^{y_{j-1}} f_j(y_j)\, dy_j, \quad y_0 := c \\ &= \begin{cases} \int_a^c f_1(y_1) \int_a^{y_1} \Omega_2\cdots\Omega_n\, dy_1 & \text{if } n > 0 \\ 1 & \text{if } n = 0. \end{cases} \end{aligned}$$

For each real number b, define a differential 1-form

$$\omega_b := \omega(b) := \frac{dx}{x - b}.$$

With this definition, the change of variable $y \mapsto 1 - y$ generates an involution $\omega(b) \mapsto \omega(1 - b)$. By repeated application of the self-evident representation

$$b^m m^{-s} = \int_0^b \omega_0^{s-1} y^{m-1}\, dy, \qquad 1 \le m \in \mathbf{Z}$$

one derives from (1.1) that

$$\begin{aligned} \lambda\binom{s_1, \ldots, s_k}{b_1, \ldots, b_k} &= \sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} b_j^{-\nu_j} \int_0^{y_{j-1}} \omega_0^{s_j-1} y_j^{\nu_j-1}\, dy_j, \quad y_0 := 1 \\ &= \prod_{j=1}^{k} \int_0^{y_{j-1}} \omega_0^{s_j-1} \frac{b_j^{-1}\, dy_j}{1 - b_j^{-1} y_j} \\ &= (-1)^k \int_0^1 \prod_{j=1}^{k} \omega_0^{s_j-1} \omega(b_j). \end{aligned} \tag{4.9}$$

Letting $s := s_1 + s_2 + \cdots + s_k$ denote the weight, one observes that the representation (4.9) is an s-dimensional iterated integral over the simplex $1 > y_1 > y_2 > \cdots > y_s > 0$. Scaling by q at each level yields the following version of the linear change of variable formula for iterated integrals:

$$\lambda_q\binom{s_1, \ldots, s_k}{b_1, \ldots, b_k} := \lambda\binom{s_1, \ldots, s_k}{qb_1, \ldots, qb_k} = (-1)^k \int_0^{1/q} \prod_{j=1}^{k} \omega_0^{s_j-1} \omega(b_j) \tag{4.10}$$

for any real number $q \neq 0$.

Having seen that every multiple polylogarithm can be represented (4.9) by a weight-dimensional iterated integral, it is natural to ask whether the converse holds. In fact, any convergent iterated integral of the form

$$\int_0^1 \prod_{r=1}^{s} \omega_{\alpha(r)} \tag{4.11}$$

can always (by collecting adjacent ω_0 factors – note that for convergence, $\alpha(s) \neq 0$) be written in the form

$$\int_0^1 \prod_{j=1}^{k} \omega_0^{s_j-1}\omega(b_j) = (-1)^k \lambda\binom{s_1,\ldots,s_k}{b_1,\ldots,b_k}, \tag{4.12}$$

where

$$0 \neq b_j = \alpha\left(\sum_{i=1}^{j} s_i\right). \tag{4.13}$$

We remark that the iterated integral representation (4.9) and the weight-dimensional non-iterated integral representation (4.5) of Example 4.3 are equivalent under the change of variable $x_j = y_{j-1}/y_j$, $y_0 := 1$, $j = 1, 2, \ldots, s$. In fact, every integral representation of Section 4.1 has a corresponding iterated integral representation under the aforementioned transformation. For example, the depth-dimensional integral (4.2) becomes

$$\lambda\binom{s_1,\ldots,s_k}{b_1,\ldots,b_k} = \prod_{j=1}^{k} \int_0^{y_{j-1}} \frac{(\log(y_{j-1}/y_j))^{s_j-1}\, dy_j}{\Gamma(s_j)(b_j - y_j)}.$$

The explicit observation that MZVs are values of iterated integrals is apparently due to Maxim Kontsevich [69]. Less formally, such representations go as far back as Euler.

5. Shuffles and Stuffles

Although it is natural to study multiple polylogarithmic sums as analytic objects, a good deal can be learned from the combinatorics of how they behave with respect to their argument strings.

5.1. **The Stuffle Algebra.** Given two argument strings $\vec{s} = (s_1, \ldots, s_k)$ and $\vec{t} = (t_1, \ldots, t_r)$, we define the set stuffle$(\vec{s}, \vec{t})$ as the smallest set of strings over the alphabet

$$\{s_1, \ldots, s_k, t_1, \ldots, t_r, \text{“+”}, \text{“,”}, \text{“(”}, \text{“)”}\}$$

satisfying

- $(s_1, \ldots, s_k, t_1, \ldots, t_r) \in \text{stuffle}(\vec{s}, \vec{t})$.
- If a string of the form (U, s_n, t_m, V) is in stuffle$(\vec{s}, \vec{t})$, then so are the strings (U, t_m, s_n, V) and $(U, s_n + t_m, V)$.

Let $\vec{a} = (a_1, \ldots, a_k)$ and $\vec{b} = (b_1, \ldots, b_r)$ be two strings of the same length as $\vec{s}$ and $\vec{t}$, respectively. We now define

$$ST := ST\binom{\vec{s}, \vec{t}}{\vec{a}, \vec{b}} \tag{5.1}$$

to be the set of all pairs $\binom{\vec{u}}{\vec{c}}$ with $\vec{u} \in \text{stuffle}(\vec{s}, \vec{t})$ and $\vec{c} = (c_1, c_2, \ldots, c_h)$ defined as follows:

- h is the number of components of $\vec{u}$,
- $c_0 := a_0 := b_0 := 1$,

- for $1 \le j \le h$, if $c_{j-1} = a_{n-1}b_{m-1}$, then

$$c_j := \begin{cases} a_n b_m, & \text{if } u_j = s_n + t_m, \\ a_n b_{m-1}, & \text{if } u_j = s_n, \\ a_{n-1} b_m, & \text{if } u_j = t_m. \end{cases}$$

5.2. **Stuffle Identities.** A class of identities which we call *"depth-length shuffles"* or *"stuffle identities"* is generated by a formula for the product of two λ-functions. Consider

$$\lambda\binom{\vec{s}}{\vec{a}}\lambda\binom{\vec{t}}{\vec{b}} = \Big\{ \sum_{\nu_1,\dots,\nu_k=1}^{\infty} \prod_{j=1}^{k} a_j^{-\nu_j} \Big(\sum_{i=j}^{k} \nu_i \Big)^{-s_j} \Big\} \Big\{ \sum_{\xi_1,\dots,\xi_r=1}^{\infty} \prod_{j=1}^{r} b_j^{-\xi_j} \Big(\sum_{i=j}^{r} \xi_i \Big)^{-t_j} \Big\}.$$

If we put

$$n_j := \textstyle\sum_{i=j}^{k} \nu_i, \quad m_j := \sum_{i=j}^{r} \xi_i,$$
$$a_j := \textstyle\prod_{i=1}^{j} x_i, \quad b_j := \prod_{i=1}^{j} y_i,$$

then it follows that

$$\lambda\binom{\vec{s}}{\vec{a}}\lambda\binom{\vec{t}}{\vec{b}} = \sum_{\substack{n_1 > \cdots > n_k > 0 \\ m_1 > \cdots > m_r > 0}} \Big(\prod_{j=1}^{k} x_j^{-n_j} n_j^{-s_j} \Big) \Big(\prod_{j=1}^{r} y_j^{-m_j} m_j^{-t_j} \Big).$$

Rewriting the previous expression in terms of λ-functions yields the stuffle formula

$$\lambda\binom{\vec{s}}{\vec{a}}\lambda\binom{\vec{t}}{\vec{b}} = \sum \lambda\binom{\vec{u}}{\vec{c}}, \tag{5.2}$$

where the sum is over all pairs of strings $\binom{\vec{u}}{\vec{c}} \in ST\binom{\vec{s},\vec{t}}{\vec{a},\vec{b}}$.

Example 5.1.

$$\begin{aligned} \lambda\binom{r,s}{a,b}\lambda\binom{t}{c} &= \lambda\binom{r,s,t}{a,b,bc} + \lambda\binom{r,s+t}{a,bc} \\ &\quad + \lambda\binom{r,t,s}{a,ac,bc} + \lambda\binom{r+t,s}{ac,bc} + \lambda\binom{t,r,s}{c,ac,bc}. \end{aligned}$$

When specialized to MZVs, this example produces the identity

$$\zeta(r,s)\zeta(t) = \zeta(r,s,t) + \zeta(r,s+t) + \zeta(r,t,s) + \zeta(r+t,s) + \zeta(t,r,s).$$

The term "stuffle" derives from the manner in which the two (upper) strings are combined. The relative order of the two strings is preserved (shuffles), but elements of the two strings may also be shoved together into a common slot (stuffing), thereby reducing the depth.

5.3. **Stuffles and Partition Integrals.** In Section 4.1, an example was given in which a stuffle identity (4.8) was seen to arise from a corresponding rational function identity (4.7) and certain partition integral representations (4.6). This is by no means an isolated phenomenon. In fact, we shall show that *every* stuffle identity is a consequence of the partition integral (4.4) applied to a corresponding rational function identity.

Theorem 5.2. *Every stuffle identity is equivalent to a rational function identity, via the partition integral.*

Before proving Theorem 5.2, we define a class of rational functions, and prove they satisfy a certain rational function identity. Let $\vec{s} = (s_1, \dots, s_k)$ and $\vec{t} = (t_1, \dots, t_r)$ be vectors of positive integers, and let $\vec{\alpha} = (\alpha_1, \dots, \alpha_k)$ and $\vec{\beta} = (\beta_1, \dots, \beta_r)$ be vectors of real numbers. As in (5.1), put

$$ST = ST\left(\begin{matrix}\vec{s}, \vec{t} \\ \vec{\alpha}, \vec{\beta}\end{matrix}\right),$$

and define

$$T = T(\vec{\alpha}, \vec{\beta}) := \left\{\vec{\gamma} : \begin{pmatrix}\vec{u} \\ \vec{\gamma}\end{pmatrix} \in ST\right\}.$$

Let $f : T \to \mathbf{Q}[\gamma_1, \gamma_2, \dots]$ be defined by

$$f(\gamma_1, \dots, \gamma_h) := \prod_{j=1}^{h} (\gamma_j - 1)^{-1}. \tag{5.3}$$

Then we have the following lemma.

Lemma 5.3. *Let f be defined as in (5.3). Then*

$$f(\vec{\alpha}) f(\vec{\beta}) = \sum_{\vec{\gamma} \in T(\vec{\alpha}, \vec{\beta})} f(\vec{\gamma}).$$

Proof of Lemma 5.3. Apply (5.2) with $\vec{a} = \vec{\alpha}$ and $\vec{b} = \vec{\beta}$. In view of (2.7), the lemma follows on taking $\vec{s}$ and $\vec{t}$ to be zero vectors of the appropriate lengths.

Proof of Theorem 5.2. Let $\vec{s}$, $\vec{t}$, $\vec{a}$, and $\vec{b}$ be as in (5.2). Let $\vec{\alpha}$ and $\vec{\beta}$ be given by

$$\alpha_j := a_j \prod_{i=1}^{j} x_i, \quad \beta_j := b_j \prod_{i=1}^{j} y_i.$$

Applying Lemma 5.3 and the partition integral representation (4.4) to the depth-dimensional integral (4.2) yields

$$\begin{aligned}
\lambda\binom{\vec{s}}{\vec{a}}\lambda\binom{\vec{t}}{\vec{b}} &= \left\{\int_1^\infty\cdots\int_1^\infty f(\vec{\alpha})\prod_{j=1}^{k}\frac{(\log x_j)^{s_j-1}dx_j}{\Gamma(s_j)\,x_j}\right\}\\
&\quad\times\left\{\int_1^\infty\cdots\int_1^\infty f(\vec{\beta})\prod_{j=1}^{r}\frac{(\log y_j)^{t_j-1}dy_j}{\Gamma(t_j)\,y_j}\right\}\\
&=\int_1^\infty\cdots\int_1^\infty\sum_{\vec{\gamma}\in T(\vec{\alpha},\vec{\beta})} f(\vec{\gamma})\left\{\prod_{j=1}^{k}\frac{(\log x_j)^{s_j-1}dx_j}{\Gamma(s_j)\,x_j}\right\}\\
&\quad\times\left\{\prod_{j=1}^{r}\frac{(\log y_j)^{t_j-1}dy_j}{\Gamma(t_j)\,y_j}\right\}\\
&=\sum_{\vec{\gamma}\in T(\vec{\alpha},\vec{\beta})}\int_1^\infty\cdots\int_1^\infty f(\vec{\gamma})\left\{\prod_{j=1}^{k}\frac{(\log x_j)^{s_j-1}dx_j}{\Gamma(s_j)\,x_j}\right\}\\
&\quad\times\left\{\prod_{j=1}^{r}\frac{(\log y_j)^{t_j-1}dy_j}{\Gamma(t_j)\,y_j}\right\}\\
&=\sum_{\binom{\vec{u}}{\vec{c}}\in ST\binom{\vec{s},\vec{t}}{\vec{a},\vec{b}}}\lambda\binom{\vec{u}}{\vec{c}},
\end{aligned}$$

as required.

5.4. **The Shuffle Algebra.** As opposed to depth-length shuffles, or stuffles, which arise from the definition (1.1) in terms of sums, the iterated integral representation (4.9) gives rise to what are called em "weight-length shuffles", or simply "shuffles". Weight-length shuffles take the form

$$\int_0^1\Omega_1\Omega_2\cdots\Omega_n\int_0^1\Omega_{n+1}\Omega_{n+2}\cdots\Omega_{n+m}=\sum\int_0^1\Omega_{\sigma(1)}\Omega_{\sigma(2)}\cdots\Omega_{\sigma(n+m)},\tag{5.4}$$

where the sum is over all $\binom{n+m}{n}$ permutations σ of the set $\{1,2,\ldots,n+m\}$ which satisfy $\sigma^{-1}(i)<\sigma^{-1}(j)$ for all $1\le i<j\le n$ and $n+1\le i<j\le n+m$. In other words, the sum is over all $(n+m)$-dimensional iterated integrals in which the relative orders of the two strings of 1-forms $\Omega_1,\ldots,\Omega_n$ and $\Omega_{n+1},\ldots,\Omega_{n+m}$ are preserved.

Example 5.4.

$$\begin{aligned}
\zeta(2,1)\zeta(2) &= -\int_0^1\omega_0\omega_1^2\int_0^1\omega_0\omega_1\\
&= -6\int_0^1\omega_0^2\omega_1^3-3\int_0^1\omega_0\omega_1\omega_0\omega_1^2-\int_0^1\omega_0\omega_1^2\omega_0\omega_1\\
&= 6\zeta(3,1,1)+3\zeta(2,2,1)+\zeta(2,1,2).
\end{aligned}$$

In contrast, the stuffle formula gives

$$\zeta(2,1)\zeta(2)=2\zeta(2,2,1)+\zeta(4,1)+\zeta(2,3)+\zeta(2,1,2).$$

Note that weight-length shuffles preserve both depth and weight. In other words, the depth (weight) of each term which occurs in the sum over shuffles is equal to

the combined depth (weight) of the two multiple polylogarithms comprising the product.

Though it may appear that the shuffles form a rather trivial class of identities satisfied by iterated integrals, it is worth mentioning that the second proof of Zagier's conjecture (see Corollary 2 of Section 11.2) uses little more than the combinatorial properties of shuffles [8]. In addition, both shuffles and stuffles have featured in the investigations of other authors in related contexts [39, 40, 41, 52, 53, 54, 55, 56, 57, 58, 61].

6. Duality

In [38], Hoffman defines an involution on strings $s_1, \ldots, s_k$. The involution coincides with a notion we refer to as duality. The duality principle states that two MZVs coincide whenever their argument strings are dual to each other, and (as noted by Zagier [69]) follows readily from the iterated integral representation. In [12], Broadhurst generalized the notion of duality to include relations between iterated integrals involving the sixth root of unity; here we allow arbitrary complex values of b_j. Thus, we find that the duality principle easily extends to multiple polylogarithms, and in this more general setting, has far-reaching implications.

6.1. **Duality for Multidimensional Polylogarithms.** We begin with the iterated integral representation (4.9) of Section 4.2. Reversing the order of the omegas and replacing each integration variable y by its complement $1 - y$ yields the dual iterated integral representation

$$\lambda\binom{s_1, \ldots, s_k}{b_1, \ldots, b_k} = (-1)^{s+k} \int_0^1 \prod_{j=k}^{1} \omega(1 - b_j)\omega_1^{s_j - 1}, \tag{6.1}$$

where again $s = s_1 + \cdots + s_k$ is the weight.

Example 6.1. Using (1.1), (4.9), and (6.1), we have

$$\lambda\binom{2, 1}{1, -1} = \int_0^1 \omega(0)\,\omega(1)\,\omega(-1) = -\int_0^1 \omega(2)\,\omega(0)\,\omega(1) = -\lambda\binom{1, 2}{2, 1},$$

which is to say that

$$\sum_{n=1}^{\infty} \frac{1}{n^2} \sum_{k=1}^{n-1} \frac{(-1)^k}{k} = -\sum_{n=1}^{\infty} \frac{1}{n2^n} \sum_{k=1}^{n-1} \frac{2^k}{k^2},$$

a result that would doubtless be difficult to prove by naïve series manipulations alone.

When $b_1 = b_2 = \cdots = b_k = b$, the two dual iterated integral representations (4.9) and (6.1) simplify as follows:

$$\lambda_b(s_1, \ldots, s_k) = (-1)^k \int_0^1 \prod_{j=1}^{k} \omega_0^{s_j - 1}\omega(b) = (-1)^{s+k} \int_0^1 \prod_{j=k}^{1} \omega(1 - b)\omega_1^{s_j - 1}. \tag{6.2}$$

A somewhat more symmetric version of (6.2) is

$$\begin{aligned}(-1)^m\lambda_b(s_1+2,\{1\}^{r_1},\ldots,s_m+2,\{1\}^{r_m}) &= (-1)^r\int_0^1\prod_{j=1}^m\omega_0^{s_j+1}\omega_b^{r_j+1}\\ &= (-1)^s\int_0^1\prod_{j=m}^1\omega_{1-b}^{r_j+1}\omega_1^{s_j+1},\end{aligned}\tag{6.3}$$

where $r:=\sum_j r_j$ and, as usual, $s:=\sum_j s_j$.

6.2. **Duality for Unsigned Euler Sums.** Taking $b=1$ in (6.3), we deduce the *MZV duality formula* (cf. [44] p. 483)

$$\zeta(s_1+2,\{1\}^{r_1},\ldots,s_m+2,\{1\}^{r_m})=\zeta(r_m+2,\{1\}^{s_m},\ldots,r_1+2,\{1\}^{s_1})\tag{6.4}$$

for multidimensional unsigned Euler sums, i.e. multiple zeta values (MZVs) .

Example 6.2. MZV duality (6.4) gives Euler's evaluation $\zeta(2,1)=\zeta(3)$, as well as the generalizations $\zeta(\{2,1\}^n)=\zeta(\{3\}^n)$, and $\zeta(2,\{1\}^n)=\zeta(n+2)$, valid for all nonnegative integers n.

In [60] a beautiful extension of MZV duality (6.4) is given, which also subsumes the so-called sum formula

$$\sum_{\substack{n_j>\delta_{j,1}\\ N=\Sigma_j n_j}}\zeta(n_1,n_2,\ldots,n_k)=\zeta(N),$$

conjectured independently by C. Moen [38] and M. Schmidt [51], and subsequently proved by A. Granville [37]. We refer the reader to Dr. Ohno's article for details.

The duality principle has an enticing converse, namely that *two MZVs with distinct argument strings are equal only if the argument strings are dual to each other.* Unfortunately, although the numerical (and symbolic) evidence in support of this converse statement is overwhelming, it still remains to be proved. In the case of self-dual strings, the conjectured converse of the duality principle implies that such a MZV can equal no other MZV; moreover we find that certain of these completely reduce, i.e. evaluate entirely in terms of (depth-one) Riemann zeta functions.

Example 6.3. The following self-dual evaluation, previously conjectured by Don Zagier [69]

$$\zeta(\{3,1\}^n)=4^{-n}\zeta(\{4\}^n)=\frac{2\pi^{4n}}{(4n+2)!},\qquad 0\le n\in\mathbf{Z},$$

is proved herein (see Section 11).

Example 6.4. The evaluation

$$\zeta(2,\{1,3\}^n)=4^{-n}\sum_{k=0}^n(-1)^k\zeta(\{4\}^{n-k})\Big\{(4k+1)\zeta(4k+2)\\ -4\sum_{j=1}^k\zeta(4j-1)\zeta(4k-4j+3)\Big\},\qquad 0\le n\in\mathbf{Z}$$

conjectured in [7] and recently proved by Bowman and Bradley [11] is also self-dual.

Example 6.5. The self-dual two-parameter generalization of Example 6.3

$$\zeta(\{2\}^m,\{3,\{2\}^m,1,\{2\}^m\}^n) \stackrel{?}{=} \frac{2(m+1)\pi^{4(m+1)n+2m}}{(2(m+1)(2n+1))!}, \qquad 0 \le m,n \in \mathbf{Z},$$

remains to be proved.

We conclude this section with the following result, since the special case $p=1$ has some bearing on the MZV duality formula (6.4).

Theorem 6.6. *Let $|p| \ge 1$. The double generating function equality*

$$1 - \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} x^{m+1}y^{n+1}\lambda_p(m+2,\{1\}^n) = {}_2F_1\left(\begin{array}{c} y,-x \\ 1-x \end{array}\middle|\, \frac{1}{p}\right)$$

holds.

Proof. By definition (2.1) of λ_p,

$$\begin{aligned}
\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} x^{m+1}y^{n+1}\lambda_p(m+2,\{1\}^n) &= y\sum_{m=0}^{\infty}x^{m+1}\sum_{k=1}^{\infty}\frac{1}{k^{m+2}p^k}\prod_{j=1}^{k-1}\left(1+\frac{y}{j}\right)\\
&= \sum_{m=0}^{\infty}x^{m+1}\sum_{k=1}^{\infty}\frac{(y)_k}{k^{m+1}k!p^k}\\
&= \sum_{k=1}^{\infty}\frac{(y)_k}{k!p^k}\left(\frac{x}{k-x}\right)\\
&= -\sum_{k=1}^{\infty}\frac{(y)_k(-x)_k}{k!p^k(1-x)_k}\\
&= 1 - {}_2F_1\left(\begin{array}{c} y,-x \\ 1-x \end{array}\middle|\, \frac{1}{p}\right)
\end{aligned}$$

as claimed. □

Remarks 6.7. In [7] it was noted that the $p=1$ case of Theorem 8.1 is equivalent to the $m=1$ case of MZV duality (6.4) via the invariance of

$$\begin{aligned}
{}_2F_1\left(\begin{array}{c} y,-x \\ 1-x \end{array}\middle|\, 1\right) &= \frac{\Gamma(1-x)\Gamma(1-y)}{\Gamma(1-x-y)}\\
&= \exp\left\{\sum_{k=2}^{\infty}\left(x^k+y^k-(x+y)^k\right)\frac{\zeta(k)}{k}\right\}
\end{aligned} \tag{6.5}$$

with respect to the interchange of x and y. However, it appears that this observation can be traced back to Drinfeld [25]. In connection with his work on series of Lie brackets, Drinfeld encountered a scaled version of the exponential series above, and showed that the coefficients of the double generating function satisfy $c_{mn} = c_{nm}$ and $c_{m0} = c_{0m}$ evaluates to $\zeta(m+2)$, up to a so-called Oppenheimer factor which we omit ([44], p. 468). In our notation, this is essentially the statement that $\zeta(m+2,\{1\}^n) = \zeta(n+2,\{1\}^m)$.

Note that Theorem 8.1 in conjunction with (6.5) shows that $\zeta(m+2,\{1\}^n)$ completely reduces (i.e. is expressible solely in terms of depth-1 Riemann zeta values) for all nonnegative integers m and n. In particular, the coefficient of $x^{m-1}y^2$ gives Euler's formula (Example 2.1); and taking the coefficient of $x^{m-1}y^3$ provides a

much simpler derivation of Markett's formula [51] for $\zeta(m,1,1)$, $m \geq 2$. Thus, the complete reducibility of $\zeta(m+2,\{1\}^n)$ is a simple consequence of the instance (6.5) of Gauss's ${}_2F_1$ hypergeometric summation theorem [1, 3, 62]. Wenchang Chu [19] has elaborated on this idea, applying additional hypergeometric summation theorems to evaluate a wide variety of depth-2 sums, including nonlinear (cf. [31]) sums.

It would be interesting to know if there is a generating function formulation of MZV duality at full strength (6.4). Presumably, it would involve an analogue of Drinfeld's associator in $2m$ non-commuting variables.

6.3. **Duality for Unit Euler Sums.** Recall the δ-function was defined (2.2) as the nested sum extension of the polylogarithm at one-half:

$$\delta(s_1,\ldots,s_k) := \lambda\binom{s_1,\ldots,s_k}{2,\ldots,2} = \sum_{\nu_1,\ldots,\nu_k=1}^{\infty} \prod_{j=1}^{k} 2^{-\nu_j} \left(\sum_{i=j}^{k} \nu_i\right)^{-s_j}. \tag{6.6}$$

Due to its geometric rate of convergence, δ-values can be computed to high precision relatively quickly. On the other hand, the unit Euler μ-sums (2.3) converge extremely slowly when the b_j all lie on the unit circle. In particular, the slow convergence of the unit (± 1) argument μ-sums initially confounded our efforts to create a data-base of numerical evaluations from which to form viable conjectures. Nevertheless, there is a close relationship between the δ-sums and the μ-sums, as we shall presently see.

Taking $b = 2$ in (6.3), we deduce the "delta-to-unit-mu" duality formula

$$\begin{aligned}\delta(s_1+2,\{1\}^{r_1},\ldots,s_m+2,\{1\}^{r_m}) \\ = (-1)^{r+m}\mu(\{-1\}^{r_m+1},\{1\}^{s_m+1},\ldots,\{-1\}^{r_1+1},\{1\}^{s_1+1}).\end{aligned} \tag{6.7}$$

Thus, every convergent unit (± 1) argument μ-sum can be expressed as a (rapidly convergent) δ-sum. The converse follows from the more general, but less symmetric formula, arising from (6.2):

$$\delta(s_1,\ldots,s_k) = (-1)^k \mu(-1,\{1\}^{s_k-1},\ldots,-1,\{1\}^{s_1-1}). \tag{6.8}$$

Example 6.8.

$$\delta(1) = \sum_{\nu=1}^{\infty} \frac{1}{\nu 2^\nu} = -\log(\tfrac{1}{2}) = \sum_{\nu=1}^{\infty} \frac{(-1)^{\nu+1}}{\nu} = -\mu(-1),$$

and more generally, for all nonnegative integers n, we have

$$\delta(n+1) = \sum_{\nu=1}^{\infty} \frac{1}{\nu^{n+1} 2^\nu} = \mathrm{Li}_{n+1}(\tfrac{1}{2}) = -\mu(-1,\{1\}^n). \tag{6.9}$$

Example 6.9. For all nonnegative integers n,

$$\delta(\{1\}^n) = (-1)^n \mu(\{-1\}^n) = (\log 2)^n/n!, \tag{6.10}$$

$$\delta(2,\{1\}^n) = (-1)^{n+1}\mu(\{-1\}^{n+1},1), \tag{6.11}$$

and more generally,

$$\delta(\{1\}^m,2,\{1\}^n) = (-1)^{m+n+1}\mu(\{-1\}^{n+1},1,\{-1\}^m), \qquad 0 \leq m,n \in \mathbf{Z}.$$

Example 6.10.

$$\delta(1,n+1) = \mu(-1,\{1\}^n,-1), \qquad 0 \le n \in \mathbf{Z},$$

and in particular, remembering (2.5,2.8,6.9) that $\delta(r) = \mathrm{Li}_r(\frac{1}{2})$, we have

$$\begin{aligned} \delta(1,0) &= 1 - \log 2 = 1 - \delta(1), \\ \delta(1,2) &= \tfrac{5}{7}\delta(2)\delta(1) - \tfrac{2}{7}\delta(3) + \tfrac{5}{21}\delta^3(1). \end{aligned}$$

Integer relation searches (see [10] or [7] for details) have failed to find a similar formula for $\delta(1,4)$. However,

$$2\delta(1,2n-1) = \sum_{j=1}^{2n-1} (-1)^{j+1}\delta(j)\delta(2n-j), \qquad 1 \le n \in \mathbf{Z}.$$

Also,

$$\delta(1,-n) = \sum_{\nu=0}^{n} \binom{n}{\nu} \frac{B_{n-\nu}\delta(-\nu)}{\nu+1}, \qquad 1 \le n \in \mathbf{Z},$$

where the $\delta(-\nu)$ are the simplex lock numbers (2.8) and the B_ν are the Bernoulli numbers [1]. More generally, if n_1 is a positive integer and $n_2, n_3, \ldots, n_r$ are all nonnegative integers, then

$$\delta(s,-n_r,\ldots,-n_2,-n_1) = \left\{ \prod_{j=1}^{r} \sum_{\nu_j=0}^{\tau_j} A(\nu_j) \right\} \delta(s-\nu_r-1), \quad s \in \mathbf{C},$$

where

$$\tau_j := n_j + \nu_{j-1} + 1, \quad A(\nu_j) := \frac{1}{\nu_j+1}\binom{\tau_j}{\nu_j} B_{\tau_j-\nu_j}, \quad \nu_0 := -1.$$

7. The Hölder Convolution

Richard Crandall [21] (see also [22]) describes a practical method for fast evaluation of MZVs. Here, we develop an entirely different approach which is based on the fact that any multiple polylogarithm can be expressed as a convolution of rapidly convergent multiple polylogarithms. We have used such representations to compute otherwise slowly convergent alternating Euler sums and (unsigned) MZVs to precisions in the thousands of digits. Lest this strike the reader as perhaps an excessive exercise in recreational computation, consider that many of our results were discovered via exhaustive numerical searches [7] for which even hundreds of digits of precision were insufficient, depending on the type of relation sought [10].

A publicly available implementation of our technique is briefly described in Section 7.2. There are also interesting theoretical considerations which we have only begun to explore. See equations (7.3)–(7.5) below for a taste of what is possible.

7.1. **Derivation and Examples.** We have seen how multiple polylogarithms with unit arguments can be expressed in terms of rapidly convergent δ-sums. What if the arguments are not necessarily units? In the iterated integral representation (4.9) the domain $1 > y_j > y_{j+1} > 0$ in $s = \sum_j s_j$ variables splits into $s+1$ parts. Each part is a product of regions $1 > y_j > y_{j+1} > 1/p$ for the first r variables, and $1/p > y_j > y_{j+1} > 0$ for the remaining $s-r$ variables. Next, $y_j \mapsto 1 - y_j$ replaces an integral of the former type by one of the latter type, with $1/p$ replaced by $1/q := 1 - 1/p$.

Motivated by these observations, we consider the string of differential 1-forms which occurs in the integrand of the iterated integral representation (4.9) and define

$$\alpha_r := \begin{cases} b_j, & \text{if } r = \sum_{i=1}^{j} s_i \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$\begin{aligned} \lambda\binom{s_1,\ldots,s_k}{b_1,\ldots,b_k} &= (-1)^k \int_0^1 \prod_{r=1}^{s} \omega(\alpha_r) \\ &= \sum_{r=0}^{s} (-1)^{r+k} \Big\{ \int_0^{1/q} \prod_{j=r}^{1} \omega(1-\alpha_j) \Big\} \Big\{ \int_0^{1/p} \prod_{j=r+1}^{s} \omega(\alpha_j) \Big\}. \end{aligned} \tag{7.1}$$

Thus, by means of (4.11), (4.12), and (4.13), we have expressed the general multiple polylogarithm as a convolution of λ_p with λ_q for any p, q such that the Hölder condition $1/p + 1/q = 1$ is satisfied. For this reason, we refer to (7.1) as the *Hölder convolution*. Note that the Hölder convolution generalizes duality (6.1) for multiple polylogarithms, as can be seen by letting p tend to infinity so that (4.10) $\lambda_p \to 0$, and $q \to 1$.

MZV Example. For any $p > 0$, $q > 0$ with $1/p + 1/q = 1$,

$$\begin{aligned} \zeta(2,1,2,1,1,1) &= \lambda_p(2,1,2,1,1,1) + \lambda_p(1,1,2,1,1,1)\lambda_q(1) \\ &\quad + \lambda_p(1,2,1,1,1)\lambda_q(2) + \lambda_p(2,1,1,1)\lambda_q(3) \\ &\quad + \lambda_p(1,1,1,1)\lambda_q(1,3) + \lambda_p(1,1,1)\lambda_q(2,3) + \lambda_p(1,1)\lambda_q(3,3) \\ &\quad + \lambda_p(1)\lambda_q(4,3) + \lambda_q(5,3) \\ &= \zeta(5,3). \end{aligned}$$

The pattern should be clear. For $1 \le j \le m$, define the concatenation products

$$\begin{aligned} \vec{a}_j &:= \mathop{\mathbf{Cat}}_{i=j}^{m} \{s_i + 2, \{1\}^{r_i}\} = \{s_j + 2, \{1\}^{r_j}, \ldots, s_m + 2, \{1\}^{r_m}\}, \\ \vec{b}_j &:= \mathop{\mathbf{Cat}}_{i=j}^{1} \{r_i + 2, \{1\}^{s_i}\} = \{r_j + 2, \{1\}^{s_j}, \ldots, r_1 + 2, \{1\}^{s_1}\}, \end{aligned}$$

and $\vec{a}_{m+1} := \vec{b}_0 := \{\}$. Then the Hölder convolution for the general MZV case is given by

$$\begin{aligned} \zeta(\vec{a}_m) &= \sum_{j=1}^{m} \Bigg\{ \sum_{t=0}^{s_j+1} \lambda_p(s_j + 2 - t, \{1\}^{r_j}, \vec{a}_{j+1}) \lambda_q(\{1\}^t, \vec{b}_{j-1}) \\ &\qquad + \sum_{\nu=1}^{r_j} \lambda_p(\{1\}^{\nu}, \vec{a}_{j+1}) \lambda_q(r_j + 2 - \nu, \{1\}^{s_j}, \vec{b}_{j-1}) \Bigg\} + \lambda_q(\vec{b}_m) \\ &= \zeta(\vec{b}_m). \end{aligned} \tag{7.2}$$

Of course, $\vec{a}_m$ and $\vec{b}_m$ are the dual strings in the MZV duality formula (6.4). Since the sums λ_p converge geometrically, whereas MZV sums converge only polynomially, (7.2) provides an excellent method of computing general MZVs to high precision with the optimal parameter choice $p = q = 2$. For rapid computation of general multiple polylogarithms, it is simplest to use the Hölder convolution (7.1) directly, translating the iterated integrals into geometrically convergent sums on a case by case basis, using (4.9).

Alternating Example.

$$\begin{aligned}
\lambda(2,1-) &= \int_0^1 \omega(0)\omega(1)\omega(-1) \\
&= \int_0^{1/p} \omega(0)\omega(1)\omega(-1) - \int_0^{1/q} \omega(1) \int_0^{1/p} \omega(1)\omega(-1) \\
&\quad + \int_0^{1/q} \omega(0)\omega(1) \int_0^{1/p} \omega(-1) - \int_0^{1/q} \omega(2)\omega(0)\omega(1) \\
&= \lambda_p(2,1-) + \lambda_p(1,1-)\lambda_q(1) + \lambda_p(1-)\lambda_q(2) - \lambda_q\binom{1,2}{2,1} \\
&= -\lambda\binom{1,2}{2,1}.
\end{aligned}$$

Although we could now work out the explicit form of the analogue to (7.2) in the alternating case, the resulting formula is too complicated in relation to its importance to justify including here.

In addition to the impressive computational implications already outlined, the Hölder convolution (7.1) gives new relationships between multiple polylogarithms, providing a path to understanding certain previously mysterious evaluations. For example, taking $p = q = 2$ shows that every MZV of weight s can be written as a weight-homogeneous convolution sum involving $2s$ δ-functions. Furthermore, employing the weight-length shuffle formula (5.4) to each product shows that every MZV of weight s is a sum of 2^s (not necessarily distinct) δ-values, each of weight s, and each appearing with unit $(+1)$ coefficient. In particular, this shows that the vector space of rational linear combinations of MZVs is spanned by the set of all δ-values. Thus,

$$\begin{aligned}
\zeta(3) &= -\int_0^{1/2} \omega_0\omega_0\omega_1 + \int_0^{1/2} \omega_1 \int_0^{1/2} \omega_0\omega_1 - \int_0^{1/2} \omega_1\omega_1 \int_0^{1/2} \omega_1 \\
&\quad + \int_0^{1/2} \omega_0\omega_1\omega_1 \\
&= \delta(3) + \int_0^{1/2} (\omega_1 \cdot \omega_0\omega_1 + \omega_0 \cdot \omega_1 \cdot \omega_1 + \omega_0\omega_1 \cdot \omega_1) \\
&\quad - \int_0^{1/2} (\omega_1\omega_1 \cdot \omega_1 + \omega_1 \cdot \omega_1 \cdot \omega_1 + \omega_1 \cdot \omega_1\omega_1) + \delta(2,1) \\
&= \delta(3) + \delta(1,2) + \delta(2,1) + \delta(2,1) + \delta(1,1,1) + \delta(1,1,1) + \delta(1,1,1) + \delta(2,1).
\end{aligned}$$

Polylog Example. Applying (7.1) to $\zeta(n+2)$, with $p = q = 2$ provides a lovely closed form for $\delta(2,\{1\}^n)$. Indeed,

$$\zeta(n+2) = \delta(2,\{1\}^n) + \sum_{r=1}^{n+2} \delta(r)\delta(\{1\}^{n+2-r}). \tag{7.3}$$

The desired closed form follows after rearranging the previous equation (7.3) and applying the definition (6.6) and the result (6.10) in the form $\delta(r) = \mathrm{Li}_r(\frac{1}{2})$ and $\delta(\{1\}^r) = (\log 2)^r/r!$, respectively.

Example 7.1. Putting $n = 1$ in (7.3) gives [5]

$$\zeta(3) = \sum_{n=1}^{\infty} \frac{1}{n^3} = \frac{1}{12}\pi^2 \log(2) + \sum_{n=1}^{\infty} \frac{1}{2^n n^2} \sum_{j=1}^{n} \frac{1}{j}. \tag{7.4}$$

In fact, formula (7.3) is non-trivial even when $n = 0$. Putting $n = 0$ in (7.3) gives the classical evaluation of the dilogarithm at one-half:

$$2\mathrm{Li}_2(\tfrac{1}{2}) = \zeta(2) - (\log 2)^2 \quad \text{i.e.} \quad \sum_{n=1}^{\infty} \frac{1}{2^n n^2} = \tfrac{1}{12}\pi^2 - \tfrac{1}{2}(\log 2)^2. \tag{7.5}$$

Differentiation of (7.1) with respect to the parameter p provides another avenue of pursuit which has not yet been fully explored. We have used this approach to derive $\delta(0, \{1\}^n) = \delta(\{1\}^n)$, but in fact, removing the initial zero is trivial from first principles.

7.2. **EZ Face.** A fast program for evaluating MZVs (as well as arithmetic expressions containing them) based on the Hölder convolution formula (7.2) has been developed at the CECM[2], and is available for public use via the World Wide Web interface called "EZ Face" (an abbreviation for Euler Zetas interFace) at the URL

`http://www.cecm.sfu.ca/projects/EZFace/`

This publicly accessible interface currently allows one to evaluate the sums

$$\mathtt{z}(s_1, \ldots, s_k) := \sum_{n_1 > \cdots > n_k} \prod_{j=1}^{k} n_j^{-|s_j|} \sigma_j^{-n_j}$$

for non-zero integers $s_1, \ldots, s_k$ and $\sigma_j := \mathrm{signum}(s_j)$, and

$$\mathtt{zp}(p, s_1, \ldots, s_k) := \sum_{n_1 > \cdots > n_k} p^{-n_1} \prod_{j=1}^{k} n_j^{-s_j}$$

for real $p \geq 1$ and positive integers $s_1, \ldots, s_k$. The code for evaluating these sums was written in C, using routines from GMP, the GNU Multiprecision Library[3]. Our implementation permits the precision of the evaluation to be set anywhere between 10 and 100 digits. Progress is currently underway to extend the scope of sums that can be evaluated. The exact status of the EZ Face is at any moment documented at its "Definitions" and "Using EZ-Face" pages.

In addition to the functions `z` and `zp`, the `lindep` function, based on the LLL algorithm [48] for discovering integer relations [10] satisfied by a vector of real numbers, can be called. An integer relation for a vector of real numbers $(x_1, \ldots, x_n)$ is a non-zero integer vector $(c_1, \ldots, c_n)$ such that $\sum_{i=1}^{n} c_i x_i = 0$. The required syntax is `lindep`$([x_1, \ldots, x_n])$, where $x_1, \ldots, x_n$ is the vector of values for which the relation is sought. One must ensure that the vector of real numbers is evaluated to sufficient precision to avoid bogus relations and other numerical artifacts. The `lindep` code was written by Michael Monagan and Greg Fee, both of the CECM, and is available on request. Send e-mail to either `monagan@cecm.sfu.ca` or `gjfee@cecm.sfu.ca`.

Below, we give some examples showing how EZ Face may be used. The left-aligned lines represent the input to EZ Face, while the centered lines represent the output of EZ Face. All computations are done with the precision of 50 digits.

---

[2]Centre for Experimental and Constructive Mathematics, Simon Fraser University.

[3]`http://www.swox.com/gmp/`

Example 7.2.
```
Pi^6/z(6)
```

```
945.000000000000000000000000000000000000000000000000
```

Example 7.3.
```
lindep([z(4,1,3), z(5,3), z(8), z(5)*z(3), z(3)^2*z(2)])
```

```
36., 36., -71., 90., -18.
```

Example 7.4.
```
lindep([ z(3), Pi^2*log(2), zp(2,2,1), zp(2,3) ])
```

```
12., -1., -12., -12.
```

Example 7.2 is a simple instance of Euler's formula for $\zeta(2n)$. Example 7.3 is the discovery of equation (3.1). Example 7.4 confirms formula (7.4).

8. Evaluations for Unit Euler Sums

As usual, the Hölder conjugates p and q denote real numbers satisfying $1/p + 1/q = 1$, and $p > 1$ or $p \le -1$ for convergence. Our first result is an easy consequence of the binomial theorem.

Theorem 8.1. *The generating function equality*

$$1 + \sum_{n=1}^{\infty} x^n \mu(\{p\}^n) = q^x.$$

holds.

Proof. By definition (2.3) of μ,

$$\begin{aligned} 1 + \sum_{n=1}^{\infty} x^n \mu(\{p\}^n) &= 1 + x \sum_{m=1}^{\infty} \frac{1}{mp^m} \prod_{j=1}^{m-1} \left(1 + \frac{x}{j}\right) \\ &= 1 + \sum_{m=1}^{\infty} \left(\frac{-1}{p}\right)^m \binom{-x}{m} \\ &= (1 - 1/p)^{-x} \\ &= q^x. \end{aligned}$$

□

Corollary 1.

$$\mu(\{p\}^n) = (\log q)^n / n!, \qquad 0 \le n \in \mathbf{Z}.$$

Remarks 8.2. Of course, when $n = 0$, we need to invoke the usual empty product convention to properly interpret $\mu(\{\}) = 1$. Since the mapping $p \mapsto 1 - p$ induces the mapping $q \mapsto 1/q$ under the Hölder correspondence, duality (6.2) takes the particularly appealing form $\mu(\{p\}^n) = (-1)^n \mu(\{1-p\}^n)$ in this context. In particular, $p = -1$ and δ-duality (6.8), (6.10) gives

$$\delta(\{1\}^n) = (-1)^n \mu(\{-1\}^n) = (\log 2)^n / n!, \qquad 0 \le n \in \mathbf{Z},$$

i.e.

$$\sum_{\nu_1,\dots,\nu_n=1}^{\infty}\prod_{j=1}^{n}\frac{1}{2^{\nu_j}(\nu_j+\cdots+\nu_n)}=\sum_{\nu_1,\dots,\nu_n=1}^{\infty}\prod_{j=1}^{n}\frac{(-1)^{\nu_j+1}}{\nu_j+\cdots+\nu_n}$$
$$=\frac{(\log 2)^n}{n!},\qquad 0\le n\in\mathbf{Z},$$

which can be viewed as an iterated sum extension of the well-known result

$$\sum_{\nu=1}^{\infty}\frac{1}{\nu 2^{\nu}}=\sum_{\nu=1}^{\infty}\frac{(-1)^{\nu+1}}{\nu}=\log 2,$$

typically obtained by comparing the Maclaurin series for $\log(1+x)$ when $x=-\frac{1}{2}$ and $x=1$.

We now prove a few results for unit Euler sums that were left as open conjectures in [7]. It will be convenient to employ the following notation:

$$A_r:=\mathrm{Li}_r(\tfrac{1}{2})=\delta(r)=\sum_{k=1}^{\infty}\frac{1}{2^k k^r},\quad P_r:=\frac{(\log 2)^r}{r!},\quad Z_r:=(-1)^r\zeta(r).\tag{8.1}$$

Theorem 8.3. *For all positive integers m,*

$$\mu(\{-1\}^m,1)=(-1)^{m+1}\sum_{k=0}^{m}A_{k+1}P_{m-k}-Z_{m+1}.$$

Proof. From the case (7.3) of the Hölder convolution, we have

$$\delta(2,\{1\}^{m-1})=\zeta(m+1)-\sum_{r=1}^{m+1}\delta(r)\delta(\{1\}^{m+1-r}).$$

Now multiply both sides by $(-1)^m$ and apply the case (6.11) of δ-duality. □

Remarks 8.4. Theorem 8.3 appeared as the conjectured formula (67) in [7], and is valid for all nonnegative integers m if the divergent $m=0$ case is interpreted appropriately. The equivalent generating function identity is

$$\sum_{n=1}^{\infty}x^n\mu(\{-1\}^n,1)=\int_0^{1/2}\frac{(1-t)^x-1}{t}dt$$
$$=\log 2+\sum_{n=1}^{\infty}\left(\frac{1}{x+n}-\frac{1}{n}\right)-\sum_{n=1}^{\infty}\frac{2^{-(x+n)}}{x+n},$$

correcting the misprinted sign in formula (21) of [7].

The asymmetry which marrs Theorem 8.3 is recovered in the generalization (2.4), restated and proved below.

Theorem 8.5. *For all positive integers m and all nonnegative integers n, we have*

$$\begin{aligned}\mu(\{-1\}^m,1,\{-1\}^n) &= (-1)^{m+1}\sum_{k=0}^{m}\binom{n+k}{n}A_{k+n+1}P_{m-k}\\ &+ (-1)^{n+1}\sum_{k=0}^{n}\binom{m+k}{m}Z_{k+m+1}P_{n-k},\end{aligned}\tag{8.2}$$

where A_r, P_r and Z_r are as in (8.1).

Proof. Let m be a positive integer, and let n be a nonnegative integer. We have

$$\begin{aligned}
\mu(\{-1\}^m, 1, \{-1\}^n) &= (-1)^{m+n+1} \int_0^1 \omega_{-1}^m \omega_1 \int_0^y \omega_{-1}^n \\
&= (-1)^{m+n+1} \int_0^1 \omega_{-1}^m \omega_1 \int_1^{1-y} \omega_2^n \\
&= (-1)^{m+n+1} \int_0^1 \omega_{-1}^m \omega_1 \int_{1/2}^{(1-y)/2} \omega_1^n \\
&= (-1)^{m+n+1} \int_0^1 \omega_{-1}^m \omega_1 \left(\log(1+y)\right)^n / n!.
\end{aligned}$$

By duality,

$$\begin{aligned}
m!n!\mu(\{-1\}^m, 1, \{-1\}^n) &= m! \int_0^1 \left(-\log(2-y)\right)^n \omega_0 \omega_2^m \\
&= m! \int_0^1 \left(-\log(2-y)\right)^n \omega_0 \int_0^{y/2} \omega_1^m \\
&= \int_0^1 \left(-\log(2-y)\right)^n \left(\log(1-y/2)\right)^m \, dy/y.
\end{aligned}$$

Letting $t = 1 - y/2$ and forming the generating function, it follows that

$$\begin{aligned}
&\sum_{m=1}^{\infty} \sum_{n=0}^{\infty} x^m y^n \mu(\{-1\}^m, 1, \{-1\}^n) \\
&= \sum_{m=1}^{\infty} \sum_{n=0}^{\infty} \frac{x^m}{m!} \frac{y^n}{n!} \int_{1/2}^1 \left(-\log(2t)\right)^n (\log t)^m \, \frac{dt}{1-t} \\
&= \int_{1/2}^1 \frac{(2t)^{-y}\left(t^x - 1\right)}{1-t} \, dt.
\end{aligned}$$

Expanding $1/(1-t)$ in powers of t and integrating term by term yields

$$\begin{aligned}
&\sum_{m=1}^{\infty} \sum_{n=0}^{\infty} x^m y^n \mu(\{-1\}^m, 1, \{-1\}^n) \\
&= 2^{-y} \sum_{k=1}^{\infty} \left(\frac{1}{k+x-y} - \frac{1}{k-y} \right) - \sum_{k=1}^{\infty} \frac{2^{-(k+x)}}{k+x-y} + \sum_{k=1}^{\infty} \frac{2^{-k}}{k-y}. \qquad (8.3)
\end{aligned}$$

Since $m \geq 1$, we may ignore the terms in (8.3) which are independent of x. Thus formally, but with the divergences coming only from the terms independent of x and hence harmless,

$$\begin{aligned}
&-2^{-x} \sum_{k=1}^{\infty} \frac{2^{-k}}{k+x-y} + 2^{-y} \sum_{k=1}^{\infty} \frac{1}{k+x-y} \\
&= -\sum_{r=0}^{\infty} (-x)^r P_r \sum_{h=1}^{\infty} (y-x)^{h-1} A_h - \sum_{r=0}^{\infty} (-y)^r P_r \sum_{h=1}^{\infty} (x-y)^{h-1} Z_h,
\end{aligned}$$

where we have used the abbreviations in (8.1). It is now a routine matter to extract the coefficient of $x^m y^n$ to complete the proof. □

Remark 8.6. Theorem 8.5 is an extension of conjectured formula (68) of [7], and is valid for all nonnegative integers m and n if the divergent $m = 0$ case is interpreted appropriately.

9. Other Integral Transformations

In Section 6, we proved the duality principle for multiple polylogarithms by using the integral transformation $y \mapsto 1 - x$. Similarly, in this section we prove additional results for multiple polylogarithms by using suitable transformations of variables in their integral representations.

Theorem 9.1. *Let n be a positive integer. Let $b_1, \ldots, b_k$ be arbitrary complex numbers, and let $s_1, \ldots, s_k$ be positive integers. Then*

$$\lambda\begin{pmatrix} s_1, s_2, \ldots, s_k \\ b_1^n, b_2^n, \ldots, b_k^n \end{pmatrix} = n^{s-k} \sum \lambda\begin{pmatrix} s_1, \ldots, s_k \\ \varepsilon_1 b_1, \ldots, \varepsilon_k b_k \end{pmatrix},$$

where the sum is over all n^k cyclotomic sequences

$$\varepsilon_1, \ldots, \varepsilon_k \in \left\{1, e^{2\pi i/n}, e^{4\pi i/n}, \ldots, e^{2\pi i(n-1)/n}\right\},$$

and, as usual, $s := s_1 + s_2 + \cdots + s_k$.

Proof. Write the left-hand side as an iterated integral as in (4.9):

$$L := \lambda\begin{pmatrix} s_1, s_2, \ldots, s_k \\ b_1^n, b_2^n, \ldots, b_k^n \end{pmatrix} = (-1)^k \int_0^1 \prod_{j=1}^{k} \omega_0^{s_j - 1} \omega(b_j^n).$$

Now let $y = x^n$ at each level of integration. This sends ω_0 to $n\omega_0$ and, by partial fractions,

$$\omega(b^n) \mapsto \sum_{r=0}^{n-1} \omega\left(be^{2\pi ir/n}\right).$$

The change of variable gives

$$L = (-1)^k \int_0^1 \prod_{j=1}^{k} (n\omega_0)^{s_j - 1} \sum_{r=0}^{n-1} \omega\left(b_j e^{2\pi ir/n}\right).$$

Now carefully expand the noncommutative product and reinterpret each resulting iterated integral as a λ-function to complete the proof. □

Example 9.2. When $n = 2$ and $k = 1$, Theorem 9.1 asserts that

$$\zeta(s) = 2^{s-1} \sum_{n=1}^{\infty} \frac{1 + (-1)^n}{n^s}.$$

Thus, Theorem 9.1 can be viewed as a cyclotomic extension of the well-known "sum over signs" formula for the alternating zeta function:

$$\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^s} = (1 - 2^{1-s})\zeta(s), \quad \Re(s) > 0.$$

Next we prove two broad generalizations of formulae (24), (26) and (28) of [7]. By a pair of **Cat** operators we mean nested concatenation (similarly as two $\sum$ signs mean nested summation).

Theorem 9.3. *Let $s_1, s_2, \ldots, s_k$ be nonnegative integers. Then*

$$\lambda\begin{pmatrix} 1+s_k, & 1+s_{k-1}, & \ldots, & 1+s_1 \\ -1, & -1, & \ldots, & -1 \end{pmatrix} = \sum \mu\Big(\mathop{\mathbf{Cat}}_{j=1}^{k} \{-1\} \mathop{\mathbf{Cat}}_{i=1}^{s_j} \{\varepsilon_{i,j}\} \Big) \prod_{j=1}^{k} \prod_{i=1}^{s_j} \varepsilon_{i,j}$$

where the sum is over all $2^{s_1+s_2+\cdots+s_k}$ sequences of signs $(\varepsilon_{i,j})$, with each $\varepsilon_{i,j} \in \{1,-1\}$ for all $1 \le i \le s_j$, $1 \le j \le k$, and **Cat** *denotes string concatenation.*

Proof. Let

$$L := \lambda\begin{pmatrix} 1+s_k, & 1+s_{k-1}, & \ldots, & 1+s_1 \\ -1, & -1, & \ldots, & -1 \end{pmatrix} = (-1)^k \int_0^1 \prod_{j=k}^{1} \omega_0^{s_j} \omega_{-1}.$$

Now let us use duality, and then we let $y = 2t/(1+t)$ at each level of integration. We get

$$L = (-1)^k \int_0^1 \prod_{j=1}^{k} \omega_{-1} (\omega_{-1} - \omega_1)^{s_j}.$$

Now let us carefully expand the noncommutative product. We get

$$L = (-1)^k \sum (-1)^{\#\varepsilon_{i,j}=1} \int_0^1 \prod_{j=1}^{k} \omega_{-1} \prod_{i=1}^{s_j} \omega(\varepsilon_{i,j}),$$

where the sum is over all sign choices $\varepsilon_{i,j} \in \{1,-1\}$, $1 \le i \le s_j$, $1 \le j \le k$, and where by $\#\varepsilon_{i,j} = a$ we mean the cardinality of the set $\{(i,j) \mid \varepsilon_{i,j} = a\}$.

Let us now interpret the iterated integrals as λ-functions. In this case, they are all unit Euler μ-sums, as we defined in (2.3). Thus,

$$L = (-1)^k \sum (-1)^{\#\varepsilon_{i,j}=1} (-1)^{k+s} \mu\Big(\mathop{\mathbf{Cat}}_{j=1}^{k} \{-1\} \mathop{\mathbf{Cat}}_{i=1}^{s_j} \{\varepsilon_{i,j}\} \Big),$$

where, as usual, $s := s_1 + s_2 + \cdots + s_k$. Now if r of the $\varepsilon_{i,j}$ equal $+1$, then $s - r$ of them equal -1. Hence,

$$L = \sum (-1)^{\#\varepsilon_{i,j}=-1} \mu\Big(\mathop{\mathbf{Cat}}_{j=1}^{k} \{-1\} \mathop{\mathbf{Cat}}_{i=1}^{s_j} \{\varepsilon_{i,j}\} \Big).$$

Finally, $(-1)^{\#\varepsilon_{i,j}=-1}$ is the same as the product over all the signs $\varepsilon_{i,j}$, and this latter observation completes the proof of Theorem 9.3. □

Theorem 9.3 generalizes several identities conjectured in [7]. For example, we get the conjecture (28) of [7] if we put $s_{n+1} = m$, $s_n = s_{n-1} = \ldots = s_1 = 0$ in Theorem 9.3. Furthermore, (24) of [7] is the case $s_{m+n+1} = s_{m+n} = \ldots = s_{n+2} = 0$, $s_{n+1} = 1$, $s_n = s_{n-1} = \ldots = s_1 = 0$, and (26) of [7] is a special case of Theorem 9.3 as well. Thus *every* multiple polylogarithm with all alternations (or, equivalently, every Euler sum with first position alternating and all the others non-alternating) is a signed sum over unit Euler sums. The representation of the sign coefficients used in Theorem 9.3 is much simpler than the cumbersome form of (28) in [7].

Below we present a dual to Theorem 9.3, which gives *any* unit Euler μ-value in terms of λ-values with all alternations (equivalently, Euler sums with only first position alternating):

Theorem 9.4. *Let $s_1, s_2, ..., s_k$ be nonnegative integers. Then*

$$\mu\Big(\mathop{\mathbf{Cat}}_{j=0}^{k-1}\{-1\}\{1\}^{s_{k-j}}\Big) = \sum \lambda\Big(\mathop{\mathbf{Cat}}_{j=1}^{k}\mathop{\mathbf{Cat}}_{i=1}^{q_j}\{t_{i,j}-\}\Big)$$

where the sum is over all $2^{s_1+s_2+\cdots+s_k}$ positive integer compositions

$$t_{1,j} + t_{2,j} + ... + t_{q_j,j} = s_j + 1, \qquad 1 \le q_j \le s_j + 1, \quad 1 \le j \le k.$$

Proof. Let

$$M := \mu\Big(\mathop{\mathbf{Cat}}_{j=0}^{k-1}\{-1\}\{1\}^{s_{k-j}}\Big) = (-1)^k\delta\Big(\mathop{\mathbf{Cat}}_{j=1}^{k}\{1+s_j\}\Big) = \int_0^1 \prod_{j=1}^{k}\omega_0^{s_j}\omega_2.$$

Again, let us make the change of variable $y = 2t/(1+t)$ at each level. Then

$$M = \int_0^1 \prod_{j=1}^{k}(\omega_0 - \omega_{-1})^{s_j}(-\omega_{-1}).$$

Again, let us carefully expand the noncommutative product. We get

$$M = \sum(-1)^{\#\varepsilon_{i,j}=-1}\int_0^1 \prod_{j=1}^{k}\left[\prod_{i=1}^{s_j}\omega(\varepsilon_{i,j})\right](-\omega_{-1}),$$

where this time, the sum is over all $\varepsilon_{i,j} \in \{0,-1\}$ with $1 \le i \le s_j$, $1 \le j \le k$. Note that each ω_{-1} in the integrand contributes -1 to the sign and $+1$ to the depth. Since

$$(-1)^{\text{depth}}\int_0^1 \text{weight-length string} = \lambda(\text{depth-length string}),$$

it follows that M is a sum of λ-values with all $+1$ coefficients. That is,

$$M = \sum \lambda\binom{\vec{t}_1, \ldots, \vec{t}_k}{-1, \ldots, -1},$$

where the sum is over all vectors

$$\vec{t}_j = (t_{1,j}, \ldots, t_{q_j,j}), \qquad 1 \le q_j \le 1 + s_j,$$

and such that

$$\sum_{i=1}^{q_j} t_{i,j} = 1 + s_j, \qquad 1 \le j \le k.$$

In other words, the sum is over all 2^s independent positive integer compositions (in the technical sense of combinatorics) of the numbers $1 + s_j$, $1 \le j \le k$. □

10. Functional Equations

One fruitful strategy for proving identities involving special values of polylogarithms is to prove more general (functional, differential) identities and instantiate them at appropriate argument values. In the last two sections of this paper we present examples of such proofs.

Lemma 10.1. *Let* $0 \le x \le 1$ *and let*

$$J(x) := \int_0^x \frac{(\log(1-t))^2}{2t}\,dt$$

Then

$$J(-x) = -J(x) + \tfrac{1}{4}J(x^2) + J\left(\frac{2x}{x+1}\right) - \tfrac{1}{8}J\left(\frac{4x}{(x+1)^2}\right). \tag{10.1}$$

Proof. If $L(x)$ and $R(x)$ denote the left-hand and the right-hand sides of (10.1), respectively, then by elementary manipulations (under the assumption $0 < x < 1$) we can show that $dL/dx = dR/dx$. The easy observation $L(0) = R(0) = 0$ then completes the proof. □

Remarks 10.2. The identity (10.1) can be discovered and proved using a computer. Once the "ingredients" (the J-terms) of the identity are chosen, the constant coefficients at them can be determined by evaluating the J-terms at a sufficiently arbitrary value of $x \in]0,1[$ and using an integer relation algorithm [10]. Once the identity is discovered, the main part of the proof (namely showing that $dL/dx = dR/dx$) can be accomplished in a computer algebra system (e.g., using the `simplify()` command of Maple).

Theorem 10.3. *We have*

$$\lambda(2-,1-) = \zeta(2,1)/8. \tag{10.2}$$

Proof. Using notation of Lemma 10.1 let us observe that

$$J(x) = \sum_{n_1 > n_2 > 0} \frac{x^{n_1}}{n_1^2 n_2}.$$

Plugging in $x = 1$ and applying (10.1) now completes the proof. □

Remarks 10.4. Theorem 10.3 is the $n = 1$ case of the conjectured identity (23) of [7], namely

$$\lambda(\underbrace{2-,1-,2,1,\ldots}_{2n}) \stackrel{?}{=} 8^{-n}\zeta(\{2,1\}^n), \tag{10.3}$$

for which we have overwhelming numerical evidence. This evidence also suggests that (10.3) with $n > 1$ seems to be the only case when two Euler sums that do not evaluate (in the sense of the definition in Section 3) have a rational quotient, different from 1. (See also Section 6.2.)

11. Differential Equations and Hypergeometric Series

Here, it is better to work with

$$L(s_1,\ldots,s_k;x) := \lambda_{1/x}(s_1,\ldots,s_k),$$

since then we have

$$\frac{d}{dx}L(s_k,\ldots,s_1;x) = \frac{1}{x}L(-1+s_k,\ldots,s_1;x)$$

if $s_k \ge 2$; while for $s_k = 1$,

$$\frac{d}{dx}L(s_k,\ldots,s_1;x) = \frac{1}{1-x}L(s_{k-1},\ldots,s_1;x).$$

With the initial conditions

$$L(s_k,\dots,s_1;0)=0,\quad k\geq 1,\quad \text{and}\quad L(\{\};x):=1,$$

the differential equations above determine the L-functions uniquely.

11.1. **Periodic Polylogarithms.** If $\vec{s}:=(s_1,s_2,\dots,s_k)$ and $s:=\sum_j s_j$, then every *periodic polylogarithm* $L(\{\vec{s}\}^r)$ has an ordinary generating function

$$L_{\vec{s}}(x,t):=\sum_{r=0}^{\infty} L(\{\vec{s}\}^r;x)t^{rs}$$

which satisfies an algebraic ordinary differential equation in x. In the simplest case, $k=1$, $\vec{s}$ reduces to the scalar s, and the differential equation for the ordinary generating function is $D_s-t^s=0$, where

$$D_s:=\left((1-x)\frac{d}{dx}\right)^1\left(x\frac{d}{dx}\right)^{s-1}.$$

The series solution is a generalized hypergeometric function

$$\begin{aligned} L_s(x,t) &= 1+\sum_{r=1}^{\infty} x^r\frac{t^s}{r^s}\prod_{j=1}^{r-1}\left(1+\frac{t^s}{j^s}\right)\\ &= {}_sF_{s-1}\left(\begin{matrix}-\omega t,-\omega^3 t,\dots,-\omega^{2s-1}t\\ 1,1,\dots,1\end{matrix}\;\middle|\; x\right),\end{aligned}$$

where $\omega=e^{\pi i/s}$, a primitive sth root of -1.

11.2. **Proof of Zagier's Conjecture.** Let ${}_2F_1(a,b;c;x)$ denote the Gaussian hypergeometric function. Then:

Theorem 11.1.

$$\begin{aligned}(11.1)\quad &\sum_{n=0}^{\infty} L(\{3,1\}^n;x)t^{4n}\\ &\qquad={}_2F_1\left(\tfrac{1}{2}t(1+i),-\tfrac{1}{2}t(1+i);1;x\right){}_2F_1\left(\tfrac{1}{2}t(1-i),-\tfrac{1}{2}t(1-i);1;x\right).\end{aligned}$$

Proof. Both sides of the putative identity start

$$1+\frac{t^4}{8}x^2+\frac{t^4}{18}x^3+\frac{t^8+44t^4}{1536}x^4+\cdots$$

and are annihilated by the differential operator

$$D_{31}:=\left((1-x)\frac{d}{dx}\right)^2\left(x\frac{d}{dx}\right)^2-t^4.$$

Once discovered, this can be checked in Mathematica or Maple. □

Corollary 2. (Zagier's Conjecture)[69] *For all nonnegative integers n,*

$$\zeta(\{3,1\}^n)=\frac{2\pi^{4n}}{(4n+2)!}.$$

Proof. Gauss's ${}_2F_1$ summation theorem gives

$$ {}_2F_1\left(a,-a;1;1\right) = \frac{1}{\Gamma(1-a)\Gamma(1+a)} = \frac{\sin(\pi a)}{\pi a}. $$

Hence, setting $x=1$ in the generating function (11.1), we have

$$\begin{aligned}
&\sum_{n=0}^{\infty} \zeta(\{3,1\}^n) t^{4n} \\
&= {}_2F_1\left(\tfrac{1}{2}t(1+i), -\tfrac{1}{2}t(1+i); 1; 1\right) {}_2F_1\left(\tfrac{1}{2}t(1-i), -\tfrac{1}{2}t(1-i); 1; 1\right) \\
&= \frac{2\sin(\frac{1}{2}(1+i)\pi t)\sin(\frac{1}{2}(1-i)\pi t)}{\pi^2 t^2} \\
&= \frac{\cosh(\pi t) - \cos(\pi t)}{\pi^2 t^2} \\
&= \sum_{n=0}^{\infty} \frac{2\pi^{4n} t^{4n}}{(4n+2)!}.
\end{aligned}$$

□

Remark 11.2. The proof is Zagier's modification of Broadhurst's, based on the extensive empirical work begun in [7].

11.3. **Generalizations of Zagier's Conjecture.** In [8] we give an alternative (combinatorial) proof of Zagier's conjecture, based on combinatorial manipulations of the iterated integral representations of MZVs (see Sections 4.2 and 5.4). Using the same technique, we prove in [8] the "Zagier dressed with 2" identity:

$$ \sum_{\vec{s}} \zeta(\vec{s}) = \frac{\pi^{4n+2}}{(4n+3)!} \tag{11.2} $$

where $\vec{s}$ runs over all $2n+1$ possible insertions of the number 2 in the string $\{3,1\}^n$. Still, (11.2) is just the beginning of a large family of conjectured identities that we discuss in [8].

12. Open Conjectures

The reader has probably noticed that many formulae proved in this paper were conjectured in [7]. For the sake of completeness, we now list formulae from [7] that are still open: (18), (23), (25), (27), (29), (44), and (70)–(74). It is possible that some of these conjectures can be proved using techniques of the present paper.

Acknowledgements

Thanks are due to David Borwein, Douglas Bowman and Keith Johnson for their helpful comments. We are especially grateful to the referee for a thorough and detailed report which led to several improvements.

References

[1] Milton Abramowitz and Irene A. Stegun (eds.) *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, Dover, New York, 1972. MR **97b:**00012.

[2] David H. Bailey, Jonathan M. Borwein and Roland Girgensohn, *Experimental Evaluation of Euler Sums*, Experiment. Math., **3** (1994), no. 1, 17–30. MR **96e:**11168.

[3] Wilfrid Norman Bailey, *Generalized Hypergeometric Series*, Cambridge Tracts in Mathematics and Mathematical Physics, No. 32, Stechert-Hafner, Inc., New York, 1964. MR **32**#2625.

[4] A. A. Beilinson, A. B. Goncharov, V. V. Schechtman, and A. N. Varchenko, *Aomoto Dilogarithms, Mixed Hodge Structures and Motivic Cohomology of Pairs of Triangles on the Plane*, the Grothendieck Festschrift, Vol. I, Progr. Math. **86**, Birkhäuser, Boston, (1990), 135–171. MR **92h:**19007.

[5] Bruce C. Berndt, *Ramanujan's Notebooks Part I*, Springer-Verlag, New York-Berlin, 1985, p. 258. MR **86c:**01062.

[6] David Borwein, Jonathan M. Borwein and Roland Girgensohn, *Explicit Evaluation of Euler Sums*, Proc. Edinburgh Math. Soc., **38** (1995), no. 2, 277–294. MR **96f:**11106.

[7] Jonathan M. Borwein, David M. Bradley and David J. Broadhurst, *Evaluations of k-fold Euler/Zagier Sums: A Compendium of Results for Arbitrary k*, Elec. J. Combin., **4** (1997), no. 2, #R5. MR **98b:**11091.

[8] Jonathan M. Borwein, David M. Bradley, David J. Broadhurst and Petr Lisoněk, *Combinatorial Aspects of Multiple Zeta Values*, Elec. J. Combin., **5** (1998), no. 1, #R38. MR **99g:**11100.

[9] Jonathan M. Borwein and Roland Girgensohn, *Evaluation of Triple Euler Sums*, Elec. J. Combin., **3** (1996), no. 1, #R23, with an appendix by David J. Broadhurst. MR **97d:**11137.

[10] Jonathan M. Borwein and Petr Lisoněk, *Applications of Integer Relation Algorithms*, Discrete Math., Proc. FPSAC'97, special issue, to appear.

[11] Douglas Bowman and David M. Bradley, *Resolution of Some Open Problems Concerning Multiple Zeta Evaluations of Arbitrary Depth*, submitted.

[12] David J. Broadhurst, *Massive 3-loop Feynman Diagrams Reducible to SC* Primitives of Algebras of the Sixth Root of Unity*, Eur. Phys. J. C **8** (1999), 311–333.

[13] ———, *On the Enumeration of Irreducible k-fold Euler Sums and Their Roles in Knot Theory and Field Theory*, to appear in J. Math. Phys.

[14] David J. Broadhurst, John A. Gracey and Dirk Kreimer, *Beyond the Triangle and Uniqueness Relations; Non-Zeta Terms at Large N from Positive Knots*, Zeit. Phys., **C75** (1997), 559–574.

[15] David J. Broadhurst and Dirk Kreimer, *Knots and Numbers in ϕ^4 Theory to 7 Loops and Beyond*, Internat. J. Modern Phys. C, **C6** (1995), no. 4, 519–524. MR **97a:**81143.

[16] ———, *Association of Multiple Zeta Values with Positive Knots via Feynman Diagrams up to 9 Loops*, Phys. Lett. B, **393** (1997), no. 3-4, 403–412. MR **98g:**11101.

[17] Jerzy Browkin, *Conjectures on the Dilogarithm*, K-Theory, **3** (1989), no. 1, 29–56. MR **90m:**11185.

[18] ———, *K-Theory, Cyclotomic Equations and Clausen's Function*, in Structural Properties of Polylogarithms, edited by Leonard Lewin, Amer. Math. Soc. Mathematical Surveys and Monographs **37**, Providence, RI, 1991, 233–273. MR 1 148 382.

[19] Wenchang Chu, *Hypergeometric series and the Riemann Zeta function*, Acta Arith., **82** (1997), no. 2, 103-118. MR **98m:**11089.

[20] Richard E. Crandall, *Topics in Advanced Scientific Computation*, Springer-Verlag, New York; TELOS. The Electronic Library of Science, Santa Clara, CA, 1996. MR **97g:**65005.

[21] ———, *Fast Evaluation of Multiple Zeta Sums*, Math. Comp., **67** (1998), no. 223, 1163–1172. MR **98j:**11066.

[22] Richard E. Crandall and Joe P. Buhler, *On the evaluation of Euler Sums*, Experiment. Math., **3** (1995), no. 4, 275-285. MR **96e:**11113.

[23] Hervé Daudé, Philippe Flajolet and Brigitte Vallée, *An Average-Case Analysis of the Gaussian Algorithm for Lattice Reduction*, Combin. Probab. Comput., **6** (1997), no. 4, 397–433. MR **99a:**65196.

[24] Karl Dilcher, *On Generalized Gamma Functions Related to the Laurent Coefficients of the Riemann Zeta Function*, Aequationes Math., **48** (1994), no. 1, 55–85. MR **95h:**11086.

[25] V. G. Drinfeld, *On Quasitriangular Quasi-Hopf Algebras and on a Group that is Closely Connected with* $\mathrm{Gal}(\bar{\mathbf{Q}}/\mathbf{Q})$, (Russian) Algebra i Analiz, **2** (1990), no. 4, 149–181. English translation in Lenningrad Math. J., **2** (1991), no. 4, 829–860. MR **92f:**16047.

[26] Harold M. Edwards, *Riemann's Zeta Function*, Pure and Applied Mathematics, Vol. 58, Academic Press, New York-London, 1974. MR **57**#5922.

[27] Leonhard Euler, *Meditationes Circa Singulare Serierum Genus*, Novi Comm. Acad. Sci. Petropol., **20** (1775), 140–186, Reprinted in "Opera Omnia", ser. I, **15**, B. G. Teubner, Berlin, 1927, pp. 217–267.

[28] Nicholas R. Farnum, *Problem 10635*, Amer. Math. Monthly, **105** (January 1998), p. 68.

[29] Helaman R. P. Ferguson, David H. Bailey and Steve Arno, *Analysis of PSLQ, An Integer Relation Finding Algorithm*, Math. Comp., **68** (1999), no. 225, 351–369. MR **99c:**11157.

[30] Philippe Flajolet, Gilbert Labelle, Louise Laforest and Bruno Salvy, *Hypergeometrics and the Cost Structure of Quadtrees*, Random Structures and Algorithms, **7** (1995), no. 2, 117–144. MR **96m:**68034.

[31] Philippe Flajolet and Bruno Salvy, *Euler Sums and Contour Integral Representations*, Experiment. Math., **7** (1998), no. 1, 15–35. MR **99c:**11110.

[32] George Gasper and Mizan Rahman, *Basic Hypergeometric Series*, with a forward by Richard Askey. Encyclopedia of Mathematics and Its Applications, **35**, Cambridge University Press, Cambridge, 1990. MR **91d:**33034.

[33] Alexander B. Goncharov, *Polylogarithms in Arithmetic and Geometry*, Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Zürich, 1994), 374–387, Birkhäuser, Basel, 1995. MR **97h:**19010.

[34] ———, *The Double Logarithm and Manin's Complex for Modular Curves*, Math. Res. Lett., **4** (1997), no. 5, 617–636. MR **99e:**11086.

[35] ———, *Multiple Polylogarithms, Cyclotomy and Modular Complexes*, Math. Res. Lett., **5** (1998), no. 4, 497–516. MR 1 653 320.

[36] Ian P. Goulden and David M. Jackson, *Combinatorial Enumeration*, with a forward by Gian Carlo-Rota. Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons, New York, 1983, pp. 186–188, pp. 414–418. MR **84m:**05002.

[37] Andrew Granville, *A Decomposition of Riemann's Zeta-Function*, in Analytic Number Theory: London Mathematical Society Lecture Note Series **247**, Y. Motohashi (ed.), Cambridge University Press, 1997, pp. 95–101.

[38] Michael E. Hoffman, *Multiple Harmonic Series*, Pacific J. Math., **152** (1992), no. 2, 275–290. MR **92i:**11089.

[39] ———, *The Algebra of Multiple Harmonic Series*, J. Algebra, **194** (1997), no. 2, 477–495. MR **99e:**11119.

[40] ———, *Quasi-Shuffle Products*, J. Alg. Comb., (to appear).

[41] ———, *Algebraic Structures on the Set of Multiple Zeta Values*, preprint.

[42] Michael E. Hoffman and Courtney Moen, *Sums of Triple Harmonic Series*, J. Number Theory, **60** (1996), no. 2, 329–331. MR **99e:**11113.

[43] Aleksandar Ivić, *The Riemann Zeta-Function*, (The Theory of the Riemann Zeta-function with Applications), John Wiley and Sons, New York, 1985. MR **87d:**11062.

[44] Christian Kassel, *Quantum Groups*, Graduate Texts in Mathematics **155**, Springer-Verlag, New York, 1995. MR **96e:**17041.

[45] Joseph D. E. Konhauser, Dan Velleman and Stan Wagon, *Which Way Did The Bicycle Go?*, Mathematical Association of America, 1996, p. 174.

[46] Gilbert Labelle and Louise Laforest, *Combinatorial Variations on Multidimensional Quadtrees*, J. Combin. Theory Ser. A, **69** (1995), no. 1, 1–16. MR **95m:**05018.

[47] Tu Quoc Thang Le and Jun Murakami, *Kontsevich's Integral for the Homfly Polynomial and Relations Between Values of Multiple Zeta Functions*, Topology Appl., **62** (1995), no. 2, 193–206. MR **96c:**57017.

[48] A. K. Lenstra, H. W. Lenstra Jr. and L. Lovász, *Factoring Polynomials with Rational Coefficients*, Math. Ann., **261** (1982), no. 4, 515–534. MR **84a:**12002.

[49] Leonard Lewin, *Polylogarithms and Associated Functions*, Elsevier North Holland, New York-Amsterdam, 1981. MR **83b:**33019.

[50] Leonard Lewin (ed.), *Structural Properties of Polylogarithms*, Amer. Math. Soc. Mathematical Surveys and Monographs **37** (1991), Providence, RI. MR **93b:**11158.

[51] Clemens Markett, *Triple Sums and the Riemann Zeta Function*. J. Number Theory, **48** (1994), no. 2, 113–132. MR **95f:**11067.

[52] Hoang Ngoc Minh, *Summations of Polylogarithms via Evaluation Transform*, Mathematics and Computers in Simulation, **42** (1996), 707–728.

[53] ———, *Fonctions de Dirichlet d'ordre n et de Paramètre t*, Discrete Math., **180** (1998), 221–241.

[54] Hoang Ngoc Minh and Michel Petitot, *Mots de Lyndon: Générateurs de Relations entre les Polylogarithmes de Nielsen*, presented at FPSAC (Formal Power Series and Algebraic Combinatorics), Vienna, July 1997.

[55] ———, *Lyndon words, Polylogarithms and the Riemann ζ Function*, Discrete Math. (to appear).
[56] Hoang Ngoc Minh, Michel Petitot and Joris van der Hoeven, *Shuffle Algebra and Polylogarithms*, in Proc. FPSAC'98, the 10th International Conference on Formal Power Series and Algebraic Combinatorics, Toronto, June 1998.
[57] ———, *L'algèbre des Polylogarithmes par les Séries Génératrices*, presented at FPSAC (Formal Power Series and Algebraic Combinatorics), Barcellona, June 1999.
[58] ———, *Computation of the Monodromy of Generalized Polylogarithms*, preprint.
[59] Niels Nielsen, *Die Gammafunktion*, Chelsea, New York, 1965, pp. 47–49. MR **32**#2622.
[60] Yasuo Ohno, *A Generalization of the Duality and Sum Formulas on the Multiple Zeta Values*, J. Number Theory, **74** (1999), 39–43.
[61] Chris Reutenauer, *Free Lie Algebras*, London Math. Soc. Monog. **7** (new series), Clarendon Press, Oxford Sciences Publications, Oxford, 1993. MR **94j:**17002.
[62] Lucy Joan Slater, *Generalized Hypergeometric Functions*, Cambridge University Press, Cambridge, 1966. MR **34**#1570.
[63] Neil J. A. Sloane, *Online Encyclopedia of Integer Sequences*, http://www.research.att.com/∼njas/sequences/.
[64] Richard P. Stanley, *Enumerative Combinatorics*, Vol. I, Wadsworth & Brooks/Cole Mathematical Series, Monterey, California, 1986, p. 146. MR **87j:**05003.
[65] Edward Charles Titchmarsh, *The Theory of the Riemann Zeta-function*, (2nd ed.) revised by D. R. Heath-Brown, The Clarendon Press, Oxford University Press, New York, 1986. MR **88c:**11049.
[66] Zdzislaw Wojtkowiak, *The Basic Structure of Polylogarithmic Functional Equations*, in Structural Properties of Polylogarithms, edited by Leonard Lewin, Amer. Math. Soc. Mathematical Surveys and Monographs **37**, Providence, RI, 1991, 205–231. MR 1 148 381.
[67] ———, *Functional Equations of Iterated Integrals with Regular Singularities*, Nagoya Math. J., **142** (1996), 145–159. MR **98b:**14018.
[68] ———, *Mixed Hodge Structures and Iterated Integrals I*, June, 1999. [K-theory preprint #351, http://www.math.uiuc.edu/K-theory]
[69] Don Zagier, *Values of Zeta Functions and their Applications*, First European Congress of Mathematics, Vol. II (Paris, 1992), Prog. Math., **120**, Birkhäuser, Basel-Boston, (1994), 497–512. MR **96k:**11110.

Centre for Experimental and Constructive Mathematics, Simon Fraser University, Burnaby, B.C., V5A 1S6, Canada
E-mail address: jborwein@cecm.sfu.ca

University of Maine, Department of Mathematics and Statistics, 5752 Neville Hall, Orono, Maine 04469–5752, U.S.A.
E-mail address: bradley@gauss.umemat.maine.edu, dbradley@member.ams.org

Physics Department, Open University, Milton Keynes, MK7 6AA, United Kingdom
E-mail address: D.Broadhurst@open.ac.uk

Centre for Experimental and Constructive Mathematics, Simon Fraser University, Burnaby, B.C., V5A 1S6, Canada
E-mail address: lisonek@cecm.sfu.ca

Asymptotic Enumeration Methods

A. M. Odlyzko

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

1. Introduction

Asymptotic enumeration methods provide quantitative information about the rate of growth of functions that count combinatorial objects. Typical questions that these methods answer are: (1) How does the number of partitions of a set of n elements grow with n? (2) How does this number compare to the number of permutations of that set?

There do exist enumeration results that leave nothing to be desired. For example, if a_n denotes the number of subsets of a set with n elements, then we trivially have $a_n = 2^n$. This answer is compact and explicit, and yields information about all aspects of this function. For example, congruence properties of a_n reduce to well-studied number theory questions. (This is not to say that all such questions have been answered, though!) The formula $a_n = 2^n$ also provides complete quantitative information about a_n. It is easy to compute for any value of n, its behavior is about as simple as possible, and it holds uniformly for all n. However, such examples are extremely rare. Usually, even when there is a formula for the function we are interested in, it is a complicated one, involving summations or recurrences. The purpose of asymptotic methods is to provide simple explicit formulas that describe the behavior of a sequence for large values of indices. There is no satisfactory definition of what is meant by "simple" or by "explicit." However, we can illustrate this concept by some examples. The number of permutations of n letters is given by $b_n = n!$. This is a compact notation, but only in the sense that factorials are so widely used that they have a special symbol. The symbol $n!$ stands for $n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 2 \cdot 1$, and it is the latter formula that has to be used to answer questions about the number of permutations. If one is after arithmetic information, such as the highest power of 7, say, that divides $n!$, one can obtain it from the product formula, but even then some work has to be done. For most quantitative purposes, however, $n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$ is inadequate. Since this formula is a product of n terms, most of them large, it is clear that $n!$ grows rapidly, but it is not obvious just how rapidly. Since all but the last term are ≥ 2, we have $n! \geq 2^{n-1}$, and since all but the last two terms are ≥ 3, we have $n! \geq 3^{n-2}$, and so on. On the other hand, each term is $\leq n$, so $n! \leq n^n$. Better bounds can clearly be obtained with

greater care. The question such estimates raise is just how far can one go? Can one obtain an estimate for $n!$ that is easy to understand, compute, and manipulate? One answer provided by asymptotic methods is Stirling's formula: $n!$ is asymptotic to $(2\pi n)^{1/2}(n/e)^n$ as $n \to \infty$, which means that the limit as $n \to \infty$ of $n!(2\pi n)^{-1/2}(n/e)^{-n}$ exists and equals 1. This formula is concise and gives a useful representation of the growth rate of $n!$. It shows, for example, that for n large, the number of permutations on n letters is considerably larger than the number of subsets of a set with $\lfloor \frac{1}{2} n \log n \rfloor$ elements.

Another simple example of an asymptotic estimate occurs in the "problème des rencontres" [81]. The number d_n of *derangements* of n letters, which is the number of ways of handing back hats to n people so that no person receives his or her own hat, is given by

$$d_n = \sum_{k=0}^{n} (-1)^k \frac{n!}{k!} \ . \tag{1.1}$$

This is a nice formula, yet to compute d_n exactly with it requires substantial effort, since the summands are large, and at first glance it is not obvious how large d_n is. However, we can obtain from (1.1) the asymptotic estimate

$$\frac{d_n}{n!} \to e^{-1} \quad \text{as} \quad n \to \infty \ . \tag{1.2}$$

To prove (1.2), we factor out $n!$ from the sum in (1.1). We are then left with a sum of rapidly decreasing terms that make up the initial segment of the series

$$e^{-1} = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \ ,$$

and (1.2) follows easily. It can even be shown that d_n is the nearest integer to $e^{-1}n!$ for all $n \geq 1$, see [81]. The estimate (1.2) does not allow us to compute d_n, but combined with the estimate for $n!$ cited above it shows that d_n grows like $(2\pi n)^{1/2} n^n e^{-n-1}$. Further, (1.2) shows that the fraction of all ways of handing out hats that results in every person receiving somebody else's hat is approximately $1/e$. Results of this type are often exactly what is desired.

Asymptotic estimates usually provide information only about the behavior of a function as the arguments get large. For example, the estimate for $n!$ cited above says only that the ratio of $n!$ to $(2\pi n)^{1/2}(n/e)^n$ tends to 1 as n gets large, and says nothing about the behavior of this ratio for any specific value of n. There are much sharper and more precise bounds for $n!$, and they will be presented in Section 3. However, it is generally true that the simpler the estimate, the weaker and less precise it is. There seems to be an unavoidable tradeoff

between conciseness and precision. Just about the simplest formula that exactly expresses $n!$ is $n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$. (We have to be careful, since there is no generally accepted definition of simplicity, and in many situations it is better to use other exact formulas for $n!$, such as the integral formula $n! = \int_0^\infty t^n e^{-t} dt$ for the Γ-function. There are also methods for evaluating $n!$ that are somewhat more efficient than the straightforward evaluation of the product.) Any other formula is likely to involve some loss of accuracy as a penalty for simplicity.

Sometimes, the tradeoffs are clear. Let $p(n)$ denote the number of partitions of an integer n. The Rademacher convergent series representation [13, 23] for $p(n)$ is valid for any $n \geq 1$:

$$p(n) = \pi^{-1} 2^{-1/2} \sum_{m=1}^{\infty} A_m(n) m^{1/2} \frac{d}{dv} (\lambda_v^{-1} \sinh(C m^{-1} \lambda_v)) \Big|_{v=n} , \tag{1.3}$$

where

$$C = \pi (2/3)^{1/2}, \quad \lambda_v = (v - 1/24)^{1/2} , \tag{1.4}$$

and the $A_m(n)$ satisfy

$$A_1(n) = 1, \quad A_2(n) = (-1)^n \quad \text{for all} \quad n \geq 1 ,$$

$$|A_m(n)| \leq m, \qquad\qquad \text{for all} \quad m, n \geq 1 ,$$

and are easy to compute. Remarkably enough, the series (1.3) does yield the exact integer value of $p(n)$ for every n, and it converges rapidly. (Although this is not directly relevant, we note that using this series to compute $p(n)$ gives an algorithm for calculating $p(n)$ that is close to optimal, since the number of bit operations is not much larger than the number of bits of $p(n)$.) By taking more and more terms, we obtain better and better approximations. The first term in (1.3) shows that

$$p(n) = \pi^{-1} 2^{-1/2} \frac{d}{dv} (\lambda_v^{-1} \sinh(C \lambda_v)) \Big|_{v=n} + O(n^{-1} \exp(C n^{1/2}/2)) , \tag{1.5}$$

and if we don't like working with hyperbolic sines, we can derive from (1.5) the simpler (but less precise) estimate

$$p(n) = \frac{1 + O(n^{-1/2})}{4 \cdot 3^{1/2} n} e^{C n^{1/2}} , \tag{1.6}$$

valid for all $n \geq 1$. Unfortunately, exact and rapidly convergent series such as (1.3) occur infrequently in enumeration, and in general we have to be content with poorer approximations.

The advantage of allowing parameters to grow large is that in surprisingly many cases, even when there do exist explicit expressions for the functions we are interested in, this procedure does yield simple asymptotic approximations, when the influence of less important factors falls

off. The resulting estimates can then be used to compare numbers of different kinds of objects, decide what the most common objects in some category are, and so on. Even in situations where bounds valid for all parameter values are needed, asymptotic estimates can be used to suggest what form those bounds should take. Usually the error terms in asymptotic estimates can be made explicit (although good bounds often require substantial work), and can be used together with computations of small values to obtain universal estimates. It is common that already for n not much larger than 10 (where n is the basic parameter) the asymptotic estimate is accurate to within a few percent, and for $n \geq 100$ it is accurate to within a fraction of a percent, even though known proofs do not guarantee results as good as this. Therefore the value of asymptotic estimates is much greater than if they just provided a picture of what happens at infinity.

Under some conditions, asymptotic results can be used to prove completely uniform results. For example, if there were any planar maps that were not four-colorable, then almost every large planar map would not be four-colorable, as it would contain one of those small pathological maps. Therefore if it could be proved that most large planar maps are four-colorable, we would obtain a new proof of the four-color theorem that would be more satisfactory to many people than the original one of Haken and Appel. Unfortunately, while this is an attractive idea, no proof of the required asymptotic estimate for the normal chromatic number of planar maps has been found so far.

Asymptotic estimates are often useful in deciding whether an identity is true. If the growth rates of the two functions that are supposed to be equal are different, then the coincidence of initial values must be an accident. There are also more ingenious ways, such as that of Example 13.1, for deducing nonexistence of identities in a wide class from asymptotic information. Sometimes asymptotics is used in a positive way, to suggest what identities might hold.

Simplicity is an important advantage of asymptotic estimates. They are even more useful when no explicit formulas for the function being studied are available, and one has to deal with indirect relations. For example, let T_n be the number of rooted unlabeled trees with n vertices, so that $T_0 = 0$, $T_1 = T_2 = 1$, $T_3 = 2$, $T_4 = 4, \ldots$. No explicit formula for the T_n is known. However, if

$$T(z) = \sum_{n=1}^{\infty} T_n z^n \tag{1.7}$$

is the ordinary generating function of T_n, then Cayley and Pólya showed that

$$T(z) = z \exp\left(\sum_{k=1}^{\infty} T(z^k)/k\right) \ . \tag{1.8}$$

This functional equation can be derived using the general Pólya-Redfield enumeration method, an approach that is sketched in Section 15. Example 15.1 shows how analytic methods can be used to prove, starting with Eq. (1.8), that

$$T_n \sim C r^{-n} n^{-3/2} \quad \text{as} \quad n \to \infty \ , \tag{1.9}$$

where

$$C = 0.4399237\ldots \ , \quad r = 0.3383219\ldots \ , \tag{1.10}$$

are constants that can be computed efficiently to high precision. For $n = 20$, $T_n = 12,826,228$, whereas $Cr^{-20}20^{-3/2} = 1.274\ldots \times 10^7$, so asymptotic formula (1.9) is accurate to better than 1%. Thus this approximation is good enough for many applications. It can also be improved easily by adding lower order terms.

Asymptotic enumeration methods are a subfield of the huge area of general asymptotic analysis. The functions that occur in enumeration tend to be of restricted form (often nonnegative and of regular growth, for example) and therefore the repertoire of tools that are commonly used is much smaller than in general asymptotics. This makes it possible to attempt a concise survey of the most important techniques in asymptotic enumeration. The task is not easy, though, as there has been tremendous growth in recent years in combinatorial enumeration and the closely related field of asymptotic analysis of algorithms, and the sophistication of the tools that are commonly used has been increasing rapidly.

In spite of its importance and growth, asymptotic enumeration has seldom been presented in combinatorial literature at a level other than that of a research paper. There are several books that treat it [43, 81, 175, 177, 235, 236, 237, 377], but usually only briefly. The only comprehensive survey that is available is the excellent and widely quoted paper of Bender [33]. Unfortunately it is somewhat dated. Furthermore, the last two decades have also witnessed a flowering of asymptotic analysis of algorithms, which was pioneered and popularized by Knuth. Combinatorial enumeration and analysis of algorithms are closely related, in that both deal with counting of particular structures. The methods used in the two fields are almost the same, and there has been extensive cross-fertilization between them. The literature on theoretical computer science, especially on average case analysis of algorithms, can therefore

be used fruitfully in asymptotic enumeration. One notable survey paper in that area is that of Vitter and Flajolet [371]. There are also presentations of relevant methods in the books [177, 209, 235, 236, 237, 223]. Section 18 is a guide to the literature on these topics.

The aim of this chapter is to survey the most important tools of asymptotic enumeration, point out references for the results and methods that are discussed, and to mention additional relevant papers that have other techniques that might be useful. It is intended for a reader who has already used combinatorial, algebraic, or probabilistic methods to reduce a problem to that of estimating sums, coefficients of a generating function, integrals, or terms in a sequence satisfying some recursion. How such a reduction is to be accomplished will be dealt with sparingly, since it is a large subject that is already covered extensively in other chapters, especially [**?**]. We will usually assume that this task has been done, and will discuss only the derivation of asymptotic estimates.

The emphasis in this chapter is on elementary and analytic approaches to asymptotic problems, relying extensively on explicit generating functions. There are other ways to solve some of the problems we will discuss, and probabilistic methods in particular can often be used instead. We will only make some general remarks and give references to this approach in Section 16.

The only methods that will be discussed in detail are fully rigorous ones. There are also methods, mostly from classical applied mathematics (cf. [31]) that are powerful and often give estimates when other techniques fail. However, we do not treat them extensively (aside from some remarks in Section 16.4) since many of them are not rigorous.

Few proofs are included in this chapter. The stress is on presentation of basic methods, with discussions of their range of applicability, statements of general estimates derivable from them, and examples of their applications. There is some repetitiveness in that several functions, such as $n!$, are estimated several times. The purpose of doing this is to show how different methods compare in their power and ease of use. No attempt is made to present derivations starting from first principles. Some of the examples are given with full details of the asymptotic analysis, to explain the basic methods. Other examples are barely more than statements of results with a brief explanation of the method of proof and a reference to where the proof can be found. The reader might go through this chapter, possibly in a random order, looking for methods that might be applicable to a specific problem, or can look for a category of methods that might fit the problem and start by looking at the corresponding sections.

There are no prerequisites for reading most of this chapter, other than acquaintance with advanced calculus and elementary asymptotic estimates. Many of the results are presented so that they can be used in a cookbook fashion. However, many of the applications require knowledge of complex variables.

Section 2 presents the basic notation used throughout the chapter. It is largely the standard one used in the literature, but it seemed worthwhile summarizing it in one place. Section 3 is devoted to a brief discussion of identities and related topics. While asymptotic methods are useful and powerful, they can often be either augmented or entirely replaced by identities, and this section points out how to use them.

Section 4 summarizes the most important and most useful estimates in combinatorial enumeration, namely those related to factorials and binomial coefficients. Section 5 is the first one to feature an in-depth discussion of methods. It deals with estimates of sums in terms of integrals, summation formulas, and the inclusion-exclusion principle. However, it does not present the most powerful tool for estimation of sums, namely generating functions. These are introduced in Section 6, which presents some of the basic properties of, and tools for dealing with generating functions. While most generating functions that are used in combinatorial enumeration converge at least in some neighborhood of the origin, there are also many nonconvergent ones. Section 7 discusses some estimates that apply to all formal series, but are especially useful for nonconvergent ones.

Section 8 is devoted to estimates for convergent power series that do not use complex variables. While not as powerful as the analytic methods presented later, these techniques are easy to use and suffice in many applications.

Section 9 presents a variety of techniques for determining the asymptotics of recurrence relations. Many of these methods are based on generating functions, and some use analytic methods that are discussed later in the chapter. They are presented at this point because they are basic to combinatorial enumeration, and they also provide an excellent illustration of the power of generating functions.

Section 10 is an introduction to the analytic methods for estimating generating functions. Many of the results mentioned here are common to all introductory complex analysis courses. However, there are also many, especially those in Sections 10.4 and 10.5, are not as well known, and are of special value in asymptotics.

Sections 11 and 12 present the main methods used in estimation of coefficients of analytic

functions in a single variable. The basic principle is that the singularities of the generating function that are closest to the origin determine the growth rate of the coefficients. If the function does not grow too fast as it approaches those singularities, the methods of Section 11 are usually applicable, while if the growth rate is high, methods of Section 12 are more appropriate.

Sections 13–15 discuss extensions of the basic methods of Sections 10–12 to multivariate generating functions, integral transforms, and problems that involve a combination of methods.

Section 16 is a collection of miscellaneous methods and results that did not easily fit into any other section, yet are important in asymptotic enumeration. Section 17 discusses the extent to which computer algebra systems can be used to derive asymptotic information. Finally, Section 18 is a guide to further reading on asymptotics, since this chapter does not provide complete coverage of the topic.

2. Notation

The symbols O, o, and $\sim$ will have the usual meaning throughout this paper:

$$f(z) = O(g(z)) \text{ as } z \to w \text{ means } f(z)/g(z) \text{ is bounded as } z \to w ;$$

$$f(z) = o(g(z)) \text{ as } z \to w \text{ means } f(z)/g(z) \to 0 \text{ as } z \to w ;$$

$$f(z) \sim g(z) \text{ as } z \to w \text{ means } f(z)/g(z) \to 1 \text{ as } z \to w .$$

When an asymptotic relation is stated for an integer variable n instead of z, it will implicitly be taken to apply only for integer values of $n \to w$, and then we will always have $w = \infty$ or $w = -\infty$. An introduction to the use of this notation can be found in [175]. Only a slight acquaintance with it is assumed, enough to see that $(1 + O(n^{-1/3}))^n = \exp(O(n^{2/3}))$ and $\log(n + n^{1/2}) = \log(n) + n^{-1/2} - (2n)^{-1} + O(n^{-3/2})$.

The notation $x \to w^-$ for real w means that x tends to w only through values $x < w$.

Some asymptotic estimates refer to *uniform convergence*. As an example, the statement that $f(z) \sim (1-z)^{-2}$ as $z \to 1$ uniformly in $|\mathrm{Arg}(1-z)| < 2\pi/3$ means that for every $\epsilon > 0$, there is a $\delta < 0$ such that

$$|f(z)(1-z)^2 - 1| \le \epsilon$$

for all z with $0 < |1-z| < \delta$, $|\mathrm{Arg}(1-z)| < 2\pi/3$. This is an important concept, since lack of uniform convergence is responsible for many failures of asymptotic methods to yield useful results.

Generating functions will usually be written in the form

$$f(z) = \sum_{n=0}^{\infty} f_n z^n \ , \tag{2.1}$$

and we will use the notation $[z^n]f(z)$ for the coefficient of z^n in $f(z)$, so that if $f(z)$ is defined by (2.1), $[z^n]f(z) = f_n$. For multivariate generating functions, $[x^m y^n]f(x,y)$ will denote the coefficient of $x^m y^n$, and so on. If a_n denotes a sequence whose asymptotic behavior is to be studied, then in combinatorial enumeration one usually uses either the *ordinary generating function* $f(z)$ defined by (2.1) with $f_n = a_n$, or else the *exponential generating function* $f(z)$ defined by (2.1) with $f_n = a_n/n!$. In this chapter we will not be concerned with the question of which type of generating function is best in a given context, but will assume that a generating function is given, and will concentrate on methods of extracting information about the coefficients from the form we have.

Asymptotic series, as defined by Poincaré, are written as

$$f_n \sim \sum_{k=0}^{\infty} a_k n^{-k} \ , \tag{2.2}$$

and mean that for every $K \geq 0$,

$$f_n = \sum_{k=0}^{K} a_k n^{-k} + O(n^{-K-1}) \quad \text{as} \quad n \to \infty \ . \tag{2.3}$$

The constant implied by the O-notation may depend on K. It is unfortunate that the same symbol is used to denote an asymptotic series as well as an asymptotic relation, defined in the first paragraph of this section. Confusion should be minimal, though, since asymptotic relations will always be written with an explicit statement of the limit of the argument.

The notation $f(z) \approx g(z)$ will be used to indicate that $f(z)$ and $g(z)$ are in some vague sense close together. It is used in this chapter only in cases where a precise statement would be cumbersome and would not help in explaining the essence of the argument.

All logarithms will be natural ones to base e unless specified otherwise, so that $\log 8 = 2.0794\ldots$, $\log_2 8 = 3$. The symbol $\lfloor x \rfloor$ denotes the greatest integer $\leq x$. The notation $x \to 1^-$ means that x tends to 1, but only from the left, and similarly, $x \to 0^+$ means that x tends to 0 only from the right, through positive values.

3. Identities, indefinite summations, and related approaches

Asymptotic estimates are useful, but often they can be avoided by using other methods. For example, the asymptotic methods presented later yield estimates for $\binom{n}{k}2^k$ as k and n vary,

which can be used to estimate accurately the sum of $\binom{n}{k}2^k$ for n fixed and k running over the full range from 0 to n. That is a general and effective process, but somewhat cumbersome. On the other hand, by the binomial theorem,

$$\sum_{k=0}^{n} \binom{n}{k} 2^k = (1+2)^n = 3^n \ . \tag{3.1}$$

This is much more satisfactory and simpler to derive than what could be obtained from applying asymptotic methods to estimate individual terms in the sum. However, such identities are seldom available. There is nothing similar that can be applied to

$$\sum_{k \le n/5} \binom{n}{k} 2^k \ , \tag{3.2}$$

and we are forced to use asymptotic methods to estimate this sum.

Recognizing when some combinatorial identity might apply is not easy. The literature on this subject is huge, and some of the references for it are [172, 174, 186, 216, 336]. Many of the books listed in the references are useful for this purpose. Generating functions (see Section 6) are one of the most common and powerful tools for proving identities. Here we only mention two recent developments that are of significance for both theoretical and practical reasons. One is Gosper's algorithm for indefinite hypergeometric summation [171, 175]. Given a sequence $a_1, a_2, \ldots$, Gosper's algorithm determines whether the sequence of partial sums

$$b_n = \sum_{k=1}^{n} a_k \ , \quad n = 1, 2, \ldots \tag{3.3}$$

has the property that b_n/b_{n-1} is a rational function of n, and if it is, it gives an explicit form for b_n. We note that if b_n/b_{n-1} is a rational function of n, then so is

$$\frac{a_n}{a_{n-1}} = \frac{b_n/b_{n-1} - 1}{1 - b_{n-2}/b_{n-1}} \ . \tag{3.4}$$

Therefore Gosper's algorithm should be applied only when a_n/a_{n-1} is rational.

The other recent development is the Wilf-Zeilberger method for proving combinatorial identities [379, 380]. Given a conjectured identity, it provides an algorithmic procedure for verifying it. This method succeeds in a surprisingly wide range of cases. Typically, to prove an identity of the form

$$\sum_{k} U(n,k) = S(n) \ , \quad n \ge 0 \ , \tag{3.5}$$

where $S(n) \neq 0$, Wilf and Zeilberger define $F(n,k) = U(n,k)/S(n)$ and search for a rational function $R(n,k)$ such that if $G(n,k) = R(n,k)F(n,k-1)$, then

$$F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k) \tag{3.6}$$

holds for all integers n, k with $n \geq 0$, and such that

1) for each integer k, the limit

$$f_k = \lim_{n\to\infty} F(n,k) \tag{3.7}$$

exists and is finite.

2) for each integer $n \geq 0$, $\lim_{k\to\pm\infty} G(n,k) = 0$.

3) $\lim_{k\to-\infty} \sum_{n=0}^{\infty} G(n,k) = 0$.

If all these conditions are satisfied, and Eq. (3.5) holds for $n = 0$, then it holds for all $n \geq 0$.

Example 3.1. *Dixon's binomial sum identity.* This identity states that

$$\sum_k (-1)^k \binom{n+b}{n+k}\binom{b+c}{b+k}\binom{n+c}{c+k} = \frac{(n+b+c)!}{n!\,b!\,c!}\ . \tag{3.8}$$

This can be proved by the Wilf-Zeilberger method by taking

$$R(n,k) = \frac{(b+1-k)(c+1-k)}{2(n+k)(n+b+c+1)} \tag{3.9}$$

and verifying that the conditions above hold. ■

The Wilf-Zeilberger method requires finding a rational function $R(n,k)$ that satisfies the properties listed above. This is often hard to do, especially by hand. Gosper's algorithm leads to a systematic procedure for constructing such $R(n,k)$.

To conclude this section, we mention that a useful resource when investigating sequences arising in combinatorial settings is the book of Sloane [345, 346], which lists several thousand sequences and gives references for them. Section 17 mentions some software systems that are useful in asymptotics.

4. Basic estimates: factorials and binomial coefficients

No functions in combinatorial enumeration are as ubiquitous and important as the factorials and the binomial coefficients. In this section we state some estimates for these quantities, which will be used throughout this chapter and are of widespread applicability. Several different proofs of some of these estimates will be sketched later.

The basic estimate, from which many others follow, is that for the factorial. As was mentioned in the introduction, the basic form of Stirling's formula is

$$n! \sim (2\pi n)^{1/2} n^n e^{-n} \quad \text{as} \quad n \to \infty \, . \tag{4.1}$$

This is sufficient for many enumeration problems. However, when necessary one can draw on much more accurate estimates. For example Eq. 6.1.38 in [297] gives

$$n! = (2\pi n)^{1/2} n^n \exp(-n + \theta/(12n)) \tag{4.2}$$

for all $n \geq 1$, where $\theta = \theta(n)$ satisfies $0 < \theta < 1$. More generally, there is Stirling's asymptotic expansion:

$$\log\{n!(2\pi n)^{-1/2} n^{-n} e^n\} \sim \frac{1}{12n} - \frac{1}{360n^3} + \cdots \, . \tag{4.3}$$

(This is an asymptotic series in the sense of Eq. (2.2), and there is no convergent expansion for $\log\{n!(2\pi n)^{-1/2} n^{-n} e^n\}$ as a power series in n^{-1}.) Further terms in the expansion (4.3) can be obtained, and they involve Bernoulli numbers. In most references, such as Eq. 6.1.37 or 6.1.40 of [297], Stirling's formula is presented for $\Gamma(x)$, where Γ is Euler's gamma function. Expansions for $\Gamma(x)$ translate readily into ones for $n!$ because $n! = \Gamma(n+1)$.

Stirling's approximation yields the expansion

$$\binom{2n}{n} = \frac{4^n}{(\pi n)^{1/2}} \left\{ 1 - \frac{1}{8n} + \frac{1}{128n^2} + \frac{5}{1024n^3} + O(n^{-4}) \right\} \, . \tag{4.4}$$

A less precise but still useful estimate is

$$\binom{n}{\lfloor n/2 \rfloor} \sim \left(\frac{2}{\pi n} \right)^{1/2} 2^n \quad \text{as} \quad n \to \infty \, . \tag{4.5}$$

This estimate is used frequently. The binomial coefficients are *symmetric*, so that $\binom{n}{k} = \binom{n}{n-k}$ and *unimodal*, so that for a fixed n and k varying, the $\binom{n}{k}$ increase monotonically up to a peak at $k = \lfloor n/2 \rfloor$ (which is unique for n even and has two equal high points at $k = (n \pm 1)/2$ for n odd) and then decrease.

More important than Eq. (4.5) are expansions for general binomial coefficients. Eq. (4.2) shows that for $1 \le k \le n-1$,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \left\{\frac{n}{2\pi k(n-k)}\right\}^{1/2} \frac{n^n}{k^k(n-k)^{n-k}} \exp\left(O\left(\frac{1}{k}+\frac{1}{n-k}\right)\right)$$
$$= \left\{\frac{n}{2\pi k(n-k)}\right\}^{1/2} \exp\left(nH\left(\frac{k}{n}\right) + O\left(\frac{1}{k}+\frac{1}{n-k}\right)\right) , \quad (4.6)$$

where

$$H(x) = -x\log x - (1-x)\log(1-x) \quad (4.7)$$

is the entropy function. (We set $H(0) = H(1) = 0$ to make $H(x)$ continuous for $0 \le x \le 1$.) Simplifying further, we obtain

$$\binom{n}{k} = \exp(nH(k/n) + O(\log n)) , \quad (4.8)$$

an estimate that is valid for all $0 \le k \le n$. In many situations it suffices to use the weaker but simpler bound

$$\binom{n}{k} \le \left(\frac{ne}{k}\right)^k , \quad 0 \le k \le n . \quad (4.9)$$

Approximations of this form are used frequently in information theory and other fields.

A general estimate that can be derived by totally elementary methods, without recourse to Stirling's formula, is

$$\binom{n}{k}\binom{n}{\lfloor n/2\rfloor}^{-1} = \exp(-2(k-n/2)^2/n + O(|k-n/2|^3/n^2)) , \quad (4.10)$$

valid for $|k-n/2| \le n/4$, say. It is most useful for $|k-n/2| = o(n^{2/3})$, since the error term is small then. Similarly,

$$\binom{n}{k+r} \sim \binom{n}{k}\left(\frac{n-k}{k}\right)^r \quad \text{as} \quad n\to\infty , \quad (4.11)$$

uniformly in k provided r (which may be negative) satisfies $r^2 = o(k)$ and $r^2 = o(n-k)$. Further, we have

$$(n+k)! \sim n^k \exp(k^2/(2n))n! \quad \text{as} \quad n\to\infty , \quad (4.12)$$

again uniformly in k provided $k = o(n^{2/3})$.

5. Estimates of sums and other basic techniques

When encountering a combinatorial sum, the first reaction should always be to check whether it can be simplified by use of some identity. If no identity for the sum is found, the

next step should be to try to transform the problem to eliminate the sum. Usually we are interested not in single isolated sums, but parametrized families of them, such as

$$b_n = \sum_k a_n(k) \ , \tag{5.1}$$

and it is the asymptotic behavior of the b_n as $n \to \infty$ that is desired. A standard and well-known technique (named the "snake-oil" method by Wilf [377]) for handling such cases is to form a generating function $f(z)$ for the b_n, use the properties of the $a_n(k)$ to obtain a simple form for $f(z)$, and then obtain the asymptotics of the b_n from the properties of $f(z)$. This method will be presented briefly in Section 6. In this section we discuss what to do if those two approaches fail. Sometimes the methods to be discussed can also be used in a preliminary phase to obtain a rough estimate for the sum. This estimate can then be used to decide which identities might be true, or what generating functions to form.

There are general methods for dealing with sums (cf. [234]), many of which are used in asymptotic enumeration. A basic technique of this type is summation by parts. Often sums to be evaluated can be expressed as

$$\sum_{j=1}^{n} a_j b_j \quad \text{or} \quad \sum_{j=1}^{\infty} a_j b_j \ ,$$

where the b_j, say, are known explicitly or behave smoothly, while the a_j by themselves might not be known well, but the asymptotics of

$$A(k) = \sum_{j=1}^{k} a_j \tag{5.2}$$

are known. Summation by parts relies on the identity

$$\sum_{j=1}^{n} a_j b_j = \sum_{k=1}^{n-1} A(k)(b_k - b_{k+1}) + A(n) b_n \ . \tag{5.3}$$

Example 5.1. *Sum of primes.* Let

$$S_n = \sum_{p \le n} p \ , \tag{5.4}$$

where p runs over the primes $\le n$. The Prime Number Theorem [23] states that the function

$$\pi(x) = \sum_{p \le x} 1 \tag{5.5}$$

satisfies

$$\pi(x) \sim \frac{x}{\log x} \quad \text{as} \quad x \to \infty \ . \tag{5.6}$$

(More precise estimates are available, but we will not use them.) We rewrite

$$S_n = \sum_{j=1}^{n} a_j b_j \ , \tag{5.7}$$

where

$$a_j = \begin{cases} 1 & j \text{ is prime} \ , \\ \\ 0 & \text{otherwise} \ , \end{cases} \tag{5.8}$$

and $b_j = j$ for all j. Then $A(k) = \pi(k)$ and summation by parts yields

$$S_n = \sum_{k=1}^{n-1} -\pi(k) + \pi(n)n \ . \tag{5.9}$$

Since

$$\sum_{k=1}^{n-1} \pi(k) \sim \sum_{k=2}^{n-1} \frac{k}{\log k} \sim \frac{n^2}{2 \log n} \quad \text{as} \quad n \to \infty \ , \tag{5.10}$$

we have

$$S_n \sim \frac{n^2}{2 \log n} \quad \text{as} \quad n \to \infty \ . \tag{5.11}$$

■

Summation by parts is used most commonly in situations like those of Example 5.1, to obtain an estimate for one sum from that of another.

Summation by parts is often easiest to carry out, both conceptually and notationally, by using integrals. If we let

$$A(x) = \sum_{k \le x} a_k \ , \tag{5.12}$$

then $A(x) = A(n)$ for $n \le x < n+1$. Suppose that $b_k = b(k)$ for some continuously differentiable function $b(x)$. Then

$$b_k - b_{k+1} = -\int_k^{k+1} b'(x) dx \ , \tag{5.13}$$

and we can rewrite Eq. (5.3) as

$$\sum_{j=1}^{n} a_j b_j = A(n)b(n) - \int_1^n A(x) b'(x) dx \ . \tag{5.14}$$

(One can apply similar formulas even when the b_j are not smooth, but this usually requires Riemann-Stieltjes integrals, cf. [14].) The approximation of sums by integrals that appears in (5.14) is common, and will be treated at length later.

5.1. Sums of positive terms

Sums of positive terms are extremely common. They can usually be handled with only a few basic tools. We devote substantial space to this topic because it is important and because the simplicity of the methods helps in illustrating some of the basic principles of asymptotic estimation, such as approximation by integrals, neglecting unimportant terms, and uniform convergence. For readers not familiar with asymptotic methods, working through the examples of this section is a good exercise that will make it easier to learn other techniques later.

Typical sums are of the form

$$b_n = \sum_k a_n(k) \ , \qquad a_n(k) \geq 0 \ , \tag{5.15}$$

where k runs over some range of summation, often $0 \leq k \leq n$ or $0 \leq k < \infty$, and the $a_n(k)$ may be given either explicitly or only through an asymptotic approximation. What is desired is the asymptotic behavior of b_n as $n \to \infty$. Usually the $a_n(k)$ for n fixed are unimodal, so that either i) $a_n(k) \leq a_n(k+1)$ for all k in the range, or ii) $a_n(k) \geq a_n(k+1)$ for all k, or iii) $a_n(k) \leq a_n(k+1)$ for $k \leq k_0$, and $a_n(k) \geq a_n(k+1)$ for $k > k_0$. The single most important task in estimating b_n is usually to find the maximal $a_n(k)$. This can be done either by combinatorial means (involving knowledge of where the $a_n(k)$ come from), by asymptotic estimation of the $a_n(k)$, or (most common when the $a_n(k)$ are expressed in terms of factorials or binomial coefficients) by finding where the ratio $a_n(k+1)/a_n(k)$ is close to 1. If $a_n(k+1)/a_n(k) < 1$ for all k, then we are in case ii) above, and if $a_n(k+1)/a_n(k) > 1$ for all k, we are in case i). If there is a k_0 in the range of summation such that $a_n(k_0+1)$ is close to $a_n(k_0)$, then we are almost certainly in case iii) and the peak occurs at some k close to k_0. The different cases are illustrated in the examples presented later in this section.

Once $\max a_n(k) = a_n(k_0)$ has been found, the next task is to show that most of the terms in the sum are insignificant. For example, if the sum in Eq. (5.15) is over $0 \leq k \leq n$, and if $a_n(0) = 1$ is the largest term, then

$$\sum_{\substack{k=0 \\ a_n(k) < n^{-2}}}^{n} a_n(k) < n^{-1} \ ,$$

which is negligible if we are only after a rough approximation to b_n, say of the form $b_n \sim c_n$ as $n \to \infty$, or even $b_n = c_n(1 + O(n^{-1}))$ as $n \to \infty$. Once the small terms have been discarded, we are usually left with a short range of summation. It can happen that this range

is extremely short, and the maximal term $a_n(k_0)$ is much larger than any of its neighbors to the extent that $b_n \sim a_n(k_0)$ as $n \to \infty$. More commonly, the number of terms that contribute significantly to b_n does grow as $n \to \infty$, but slowly. Their contribution, relative to that of the maximal term $a_n(k_0)$, can usually be estimated by some simple function of $k - k_0$, and the sum of all of them approximated by an explicit integral. This method is sometimes referred to as Laplace's method for sums (in analogy to Laplace's method for estimating integrals, mentioned in Section 5.5, which proceeds in a similar spirit). There is extensive discussion of this method in [63].

Example 5.2. *Sums of the partition function.* We estimate

$$U_n = \sum_{k=1}^{n} p(k)^k \ , \tag{5.16}$$

where $p(k)$ is the number of partitions of k. Since any partition of $m-1$, say one with c_j parts of size j, can be transformed into a partition of m with $c_1 + 1$ parts of size 1, and c_j of size j for $j \geq 2$, we have $p(m) \geq p(m-1)$ for all $m \geq 2$. Therefore the largest term in the sum in (5.16) is the one with $k = n$. If the only estimate for $p(k)$ that we have is the one given by (1.6), then

$$p(n)^n = \exp(Cn^{3/2} - n\log(4 \cdot 3^{1/2}n) + O(n^{1/2})) \ . \tag{5.17}$$

Since the constant implied by the O-symbol is not specified, this estimate is potentially larger than $p(n)^n$ by a factor of $\exp(cn^{1/2})$, so we can only obtain asymptotics of $\log p(n)^n$, not of $p(n)^n$ itself. This also means that rough estimates of U_n follow easily from (5.17). Since $p(k)^k \leq p(n)^n$ for all $k < n$, and there are n terms in the sum, we have $p(n)^n \leq U_n \leq np(n)^n$, and because of the large error term in (5.17), we obtain

$$U_n = \exp(Cn^{3/2} - n\log(4 \cdot 3^{1/2}n) + O(n^{1/2})) \ . \tag{5.18}$$

Thus the use of the poor estimate (1.6) for $p(n)$ means that we can obtain only a crude estimate for U_n, and there is no need for careful analysis.

Instead of (1.6) we can use the more refined estimate (1.5). Let q_n denote first term on the right side of (1.5). Then we have

$$p(n) = q_n + O(n^{-1}\exp(Cn^{1/2}/2)) = q_n(1 + O(\exp(-Cn^{1/2}/2))) \ , \tag{5.19}$$

so

$$p(n)^n = q_n^n(1 + O(n\exp(-Cn^{1/2}/2))) = q_n^n(1 + O(\exp(-Cn^{1/2}/3))) \ , \tag{5.20}$$

say. Also, for some $\epsilon > 0$ we find from Eq. (1.5) (or Eq. 1.6) that for large n

$$q_{n-1} < q_n - \epsilon n^{-1/2} q_n \ .$$

Thus for large n,

$$\begin{aligned} q_{n-1}^{n-1} &< q_n^{n-1}(1-\epsilon n^{-1/2})^{n-1} \\ &< q_n^n \exp(-\epsilon n^{1/2}/2) \ , \end{aligned}$$

and therefore

$$\sum_{k=1}^{n-1} p(k)^k \le (n-1)p(n-1)^{n-1} < q_n^n \exp(-\epsilon n^{1/2}/3) \ .$$

Thus we obtain

$$U_n = q_n^n(1 + O(\exp(-\delta n^{1/2}))) \tag{5.21}$$

for some $\delta > 0$.

The estimates of U_n presented above relied on the observation that the last term in the sum (5.16) defining U_n is much larger than the sum of all the other terms. This does not happen often. A more typical example is presented by

$$T_n = \sum_{k=1}^{n} p(k) \ . \tag{5.22}$$

As was noted before, $p(n)$ is larger than any of the other terms, but not by enough to dominate the sum. We therefore try the other approaches that were listed at the beginning of this section. We use only the estimate (1.6). Since $(1-x)^{1/2} < 1 - x/2$ for $0 \le x \le 1$, we find that for large n,

$$\begin{aligned} \sum_{k<n-n^{2/3}} p(k) &\le np(n - \lceil n^{2/3} \rceil) \\ &\le \exp(C(n - \lceil n^{2/3} \rceil)^{1/2}) \\ &\le \exp(Cn^{1/2} - Cn^{1/6}/2) \\ &= O(p(n)\exp(-Cn^{1/6}/3)) \ . \end{aligned} \tag{5.23}$$

Thus most of the values of k contribute a negligible amount to the sum. For $k = n - j$, $0 \le j \le n^{2/3}$, we find that

$$p(n-j)/p(n) = (1 + O(n^{-1/3}))\exp(C(n-j)^{1/2} - Cn^{1/2}) \ .$$

Since

$$\begin{aligned} (n-j)^{1/2} &= n^{1/2} - jn^{-1/2}/2 + O(j^2 n^{-3/2}) \ , \\ p(n-j)/p(n) &= \exp(-Cjn^{-1/2}/2 + O(n^{-1/6})) \\ &= (1 + O(n^{-1/6}))\exp(-Cjn^{-1/2}/2) \ . \end{aligned} \tag{5.24}$$

Thus the ratios $p(n-j)/p(n)$ decrease geometrically, and so

$$p(n)^{-1} \sum_{0 \le j \le n^{2/3}} p(n-j) = \frac{(1+O(n^{-1/6}))}{1-\exp(-Cn^{-1/2}/2)} = 2C^{-1}n^{1/2}(1+O(n^{-1/6})) \ . \tag{5.25}$$

Therefore, combining all the estimates,

$$T_n = \sum_{k=1}^{n} p(k) = \frac{1+O(n^{-1/6})}{2 \cdot C \cdot 3^{1/2} \cdot n^{1/2}} \ e^{Cn^{1/2}} \ . \tag{5.26}$$

The $O(n^{-1/6})$ error term above can easily be improved with a little more care to $O(n^{-1/2})$, even if we continue to rely only on (1.6). ■

Before presenting further examples, we discuss some of the problems that can arise even in the simple setting of estimating positive sums. We then introduce the basic technique of approximating sums by integrals.

The lack of uniform convergence is a frequent cause of incorrect estimates. If $a_n(k) \sim c_n(k)$ for each k as $n \to \infty$, it does not necessarily follow that

$$b_n = \sum_k a_n(k) \sim \sum_k c_n(k) \quad \text{as} \quad n \to \infty \ . \tag{5.27}$$

A simple counterexample is given by $a_n(k) = \binom{n}{k}$ and $c_n(k) = \binom{n}{k}(1+k/n)$. To conclude that (5.27) holds, it is usually necessary to know that $a_n(k) \sim c_n(k)$ as $n \to \infty$ uniformly in k. Such uniform convergence does hold if we replace $c_n(k)$ in the counterexample above by $c'_n(k) = \binom{n}{k}(1+k/n^2)$, for example.

There is a general principle that sums of terms that vary smoothly with the index of summation should be replaced by integrals, so that for $\alpha > 0$, say,

$$\sum_{k=1}^{n} k^\alpha \sim \int_1^{n+1} u^\alpha du \quad \text{as} \quad n \to \infty \ . \tag{5.28}$$

The advantage of replacing a sum by an integral is that integrals are usually much easier to handle. Many more closed-form expressions are available for definite and indefinite integrals than for sums. We will discuss extensions of this principle of replacing sums by integrals further in Section 5.3, when we present the Euler-Maclaurin summation formula. Usually, though, we do not need anything sophisticated, and the application of the principle to situations like that of (5.28) is easy to justify. If $a_n = g(n)$ for some function $g(x)$ of a real argument x, then

$$\left| g(n) - \int_n^{n+1} g(u)du \right| \le \max_{n \le u \le n+1} |g(u) - g(n)| \ , \tag{5.29}$$

and so

$$\left|\sum_n g(n) - \int g(u)du\right| \le \sum_n \max_{n\le u\le n+1} |g(u) - g(n)| \ , \tag{5.30}$$

where the integral is over $[a, b+1]$ if the sum is over $a \le n \le b$, $a, b \in Z$. If $g(u)$ is continuously differentiable, then $|g(u) - g(n)| \le \max_{n\le v\le n+1} |g'(v)|$ for $n \le u \le n+1$. This gives the estimate

$$\left|\sum_{n=a}^{b} g(n) - \int_a^{b+1} g(u)du\right| \le \sum_{n=a}^{b} \max_{n\le v\le n+1} |g'(v)| \ . \tag{5.31}$$

Often one can find a simple explicit function $h(w)$ such that $|g'(v)| \le h(w)$ for any v and w with $|v - w| \le 1$, in which case Eq. (5.31) can be replaced by

$$\left|\sum_{n=a}^{b} g(n) - \int_a^{b+1} g(u)du\right| \le \int_a^{b+1} h(v)dv \ . \tag{5.32}$$

For good estimates to be obtained from integral approximations to sums, it is usually necessary for individual terms to be small compared to the sum.

Example 5.3. *Sum of* $\exp(-\alpha k^2)$. In the final stages of an asymptotic approximation one often encounters sums of the form

$$h(\alpha) = \sum_{k=-\infty}^{\infty} \exp(-\alpha k^2) \ , \qquad \alpha > 0 \ . \tag{5.33}$$

There is no closed form for the indefinite integral of $\exp(-\alpha u^2)$ (it is expressible in terms of the Gaussian error function only), but there is the famous evaluation of the definite integral

$$\int_{-\infty}^{\infty} \exp(-\alpha u^2)du = (\pi/\alpha)^{1/2} \ . \tag{5.34}$$

Thus it is natural to approximate $h(\alpha)$ by $(\pi/\alpha)^{1/2}$. If $g(u) = \exp(-\alpha u^2)$, then $g'(u) = -2\alpha u g(u)$, and so for $n \ge 0$,

$$\max_{n\le v\le n+1} |g'(v)| \le 2\alpha(n+1)g(n) \ . \tag{5.35}$$

For the integral in Eq. (5.30) to yield a good approximation to the sum we must show that the error term is smaller than the integral. The largest term in the sum occurs at $n = 0$ and equals 1. The error bound (5.35) that comes from approximating $g(0) = 1$ by the integral of $g(u)$ over $0 \le u \le 1$ is 2α. Therefore we cannot expect to obtain a good estimate unless $\alpha \to 0$. We find that

$$2\alpha(n+1)g(n) \le 4\alpha u g(u/2) \quad \text{for} \quad n \ge 1, \quad n \le u \le n+1 \ ,$$

so (integral approximation again!)

$$\begin{aligned} \sum_{n=1}^{\infty} 2\alpha(n+1)g(n) &\leq 4\alpha \int_1^{\infty} ug(u/2)du \\ &\leq 4\alpha \int_0^{\infty} ug(u/2)du = (8\alpha)^{1/2} \ . \end{aligned} \tag{5.36}$$

Therefore, taking into account the error for $n = 0$ which was not included in the bound (5.36), we have

$$\begin{aligned} h(\alpha) &= \sum_{n=-\infty}^{\infty} \exp(-\alpha n^2) = \int_{-\infty}^{\infty} \exp(-\alpha u^2)du + O(\alpha^{1/2} + \alpha) \\ &= (\pi/\alpha)^{1/2} + O(\alpha^{1/2}) \quad \text{as} \quad \alpha \to 0^+ \ . \end{aligned} \tag{5.37}$$

For this sum much more precise estimates are available, as will be shown in Example 5.9. For many purposes, though, (5.37) is sufficient. ■

Example 5.3 showed how to use the basic tool of approximating a sum by an integral. Moreover, the estimate (5.37) that it provides is ubiquitous in asymptotic enumeration, since many approximations reduce to it. This is illustrated by the following example.

Example 5.4. *Bell numbers* (cf. [63]). The Bell number, $B(n)$, counts the partitions of an n-element set. It is given by [81]

$$B(n) = e^{-1} \sum_{k=1}^{\infty} \frac{k^n}{k!} \ . \tag{5.38}$$

In this sum no single term dominates. The ratio of the $(k+1)$-st to the k-th term is

$$\frac{(k+1)^n}{(k+1)!} \cdot \frac{k!}{k^n} = \frac{1}{k+1} \left(1 + \frac{1}{k}\right)^n \ . \tag{5.39}$$

As k increases, this ratio strictly decreases. We search for the point where it is about 1. For $k \geq 2$,

$$\left(1 + \frac{1}{k}\right)^n = \exp\left(n \log\left(1 + \frac{1}{k}\right)\right) = \exp(n/k + O(n/k^2)) \ , \tag{5.40}$$

so the ratio is close to 1 for n/k close to $\log(k+1)$. We choose k_0 to be the closest integer to w, the solution to

$$n = w \log(w+1) \ . \tag{5.41}$$

For $k = k_0 + j$, $1 \le j \le k_0/2$, we find, since $\log(1 + i/k_0) = i/k_0 - i^2/(2k_0^2) + O(i^3/k_0^3)$,

$$\begin{aligned} \frac{k^n}{k!} &= \frac{k_0^n}{k_0!} \frac{(1 + j/k_0)^n}{k_0^j \Pi_{i=1}^j (1 + i/k_0)} \\ &= \frac{k_0^n}{k_0!} \exp\left(jn/k_0 - j \log k_0 - j^2(n + k_0)/(2k_0^2) + O(nj^3/k_0^3 + j/k_0) \right) . \end{aligned} \tag{5.42}$$

The same estimate applies for $-k_0/2 \le j \le 0$. The term $jn/k_0 - j \log k_0$ is small, since $|k_0 - w| \le 1/2$ and w satisfies (5.41). We find

$$\begin{aligned} n/k_0 - \log k_0 &= n/w - \log(w + 1) + O(n/w^2 + 1/w) \\ &= O(n/w^2 + 1/w) . \end{aligned} \tag{5.43}$$

By (5.41), $w \sim n/\log n$ as $n \to \infty$. We now further restrict j to $|j| \le n^{1/2} \log n$. Then (5.42) and (5.43) yield

$$\frac{k^n}{k!} = \frac{k_0^n}{k_0!} \exp(-j^2(n + k_0)/(2k_0^2) + O((\log n)^6 n^{-1/2})) . \tag{5.44}$$

Approximating the sum by an integral, as in Example 5.3, shows that

$$\sum_{\substack{k \\ |j| \le n^{1/2} \log n}} \frac{k^n}{k!} = \frac{k_0^n}{k_0!} k_0 (2\pi)^{1/2} (n + k_0)^{-1/2} (1 + O((\log n)^6 n^{-1/2})) . \tag{5.45}$$

(An easy way to obtain this is to apply the estimate of Example 5.3 to the sum from $-\infty$ to ∞, and show that the range $|j| > n^{1/2} \log n$ contributes little.) To estimate the contribution of the remaining summands, with $|j| > n^{1/2} \log n$, we observe that the ratio of successive terms is ≤ 1, so the range $1 \le k \le k_0 - \lfloor n^{1/2} \log n \rfloor$ contributes at most k_0 (the number of terms) times the largest term, which arises for $k = k_0 - \lfloor n^{1/2} \log n \rfloor$. By (5.44), this largest term is

$$O(k_0^n (k_0!)^{-1} \exp(-(\log n)^3)) .$$

For $k \ge k_1 \ge k_0 + \lfloor n^{1/2} \log n \rfloor$, we find that the ratio of the $(k+1)$-st to the k-th term is, for large n,

$$\begin{aligned} \le \frac{1}{k_1 + 1} \left(1 + \frac{1}{k_1} \right)^n &= \exp(n/k_1 - \log(k_1 + 1) - n/(2k_1^2) + O(n/k_1^3)) \\ &\le \exp(-(k_1 - k_0)n/k_1^2 + O(n/k_1^3)) \\ &\le \exp(-2n^{-1/2}) \ \le \ 1 - n^{-1/2} , \end{aligned} \tag{5.46}$$

and so the sum of these terms, for $k_1 \le k < \infty$, is bounded above by $n^{1/2}$ times the term for $k = k_1$. Therefore the estimate on the right-hand side of (5.45) applies even when we sum on all k, $1 \le k < \infty$.

To obtain an estimate for $B(n)$, it remains only to estimate $k_0^n/k_0!$. To do this, we apply Stirling's formula and use the property that $|k_0 - w| \le 1/2$ to deduce that

$$B(n) \sim (\log w)^{1/2} w^{n-w} e^w \quad \text{as} \quad n \to \infty \,, \tag{5.47}$$

where w is given by (5.41).

There is no explicit formula for w in terms of n, and substituting various asymptotic approximations to w, such as

$$w = \frac{n}{\log n} + O\left(\frac{n}{(\log n)^2}\right) \tag{5.48}$$

(see Example 5.10) yields large error terms in (5.47), so for accuracy it is usually better to use (5.47) as is. There are other approximations to $B(n)$ in the literature (see, for example, [33, 63]). They differ slightly from (5.47) because they estimate $B(n)$ in terms of roots of equations other than (5.41).

Other methods of estimating $B(n)$ are presented in Examples 12.5 and 12.6. ■

5.2. Alternating sums and the principle of inclusion-exclusion

At the beginning of Section 5, the reader was advised in general to search for identities and transformations when dealing with general sums. This advice is even more important when dealing with sums of terms that have alternating or irregularly changing coefficients. Finding the largest term is of little help when there is substantial cancellation among terms. Several general approaches for dealing with this difficulty will be presented later. Generating function methods for dealing with complicated sums are discussed in Section 6. Contour integration methods for alternating sums are mentioned in Section 10.3. The summation formulas of the next section can sometimes be used to estimate sums with regularly varying coefficients as well. In this section we present some basic elementary techniques that are often sufficient.

Sometimes it is possible to obtain estimates of sums with positive and negative summands by approximating separately the sums of the positive and of the negative summands. Methods of the preceding section or of the next section are useful in such situations. However, this approach is to be avoided as much as possible, because it often requires extremely precise estimates of the two sums to obtain even rough bounds on the desired sums. One method that often works and is much simpler consists of a simple pairing of adjacent positive and negative terms.

Example 5.5. *Alternating sum of square roots.* Let

$$S_n = \sum_{k=1}^{n} (-1)^k k^{1/2} \ . \tag{5.49}$$

We have

$$\begin{aligned} (2m)^{1/2} - (2m-1)^{1/2} &= (2m)^{1/2} \left\{ 1 - \left(1 - \frac{1}{2m} \right)^{1/2} \right\} \\ &= (2m)^{1/2} \left\{ 1 - \left(1 - \frac{1}{4m} + O(m^{-2}) \right) \right\} \\ &= (8m)^{-1/2} + O(m^{-3/2}) \ , \end{aligned} \tag{5.50}$$

so

$$\begin{aligned} \sum_{k=1}^{2\lfloor n/2 \rfloor} (-1)^k k^{1/2} &= \sum_{m=1}^{\lfloor n/2 \rfloor} (8m)^{-1/2} + O(1) \\ &= n^{1/2}/2 + O(1) \ . \end{aligned} \tag{5.51}$$

Hence

$$S_n = \begin{cases} n^{1/2}/2 + O(1) & \text{if } n \text{ is even} \ , \\ -n^{1/2}/2 + O(1) & \text{if } n \text{ is odd} \ . \end{cases} \tag{5.52}$$

■

In Example 5.5, the sums of the positive terms and of the negative terms can easily be estimated accurately (for example, by using the Euler-Maclaurin formula of the next section) to obtain (5.52). In other cases, though, the cancellation is too extensive for such an approach to work. This is especially true for sums arising from the principle of inclusion-exclusion.

Suppose that X is some set of objects and P is a set of properties. For $R \subseteq P$, let $N_=(R)$ be the number of objects in X that have exactly the properties in R and none of the properties in $P \setminus R$. We let $N_\geq(R)$ denote the number of objects in X that have all the properties in R and possibly some of those in $P \setminus R$. The principle of inclusion-exclusion says that

$$N_=(R) = \sum_{R \subseteq Q \subseteq P} (-1)^{|Q \setminus R|} N_\geq(Q) \ . \tag{5.53}$$

(This is a basic version of the principle. For more general results, proofs, and references, see [81, 173, 351].)

Example 5.6. *Derangements of n letters.* Let X be the set of permutations of n letters, and suppose that P_i, $1 \le i \le n$, is the property that the i-th letter is fixed by a permutation, and $P = \{P_1, \dots, P_n\}$. Then d_n, the number of derangements of n letters, equals $N_=(\phi)$, where ϕ is the empty set, and so by (5.53)

$$d_n = \sum_{Q \subseteq P} (-1)^{|Q|} N_\ge(Q) \; . \tag{5.54}$$

However, $N_\ge(Q)$ is just the number of permutations that leave all letters specified by Q fixed, and thus

$$\begin{aligned} d_n &= \sum_{Q \subseteq P} (-1)^{|Q|} (n - |Q|)! \\ &= \sum_{k=0}^{n} (-1)^k (n-k)! \binom{n}{k} = \sum_{k=0}^{n} (-1)^k \frac{n!}{k!} \; , \end{aligned} \tag{5.55}$$

which is Eq. (1.1). ∎

The formula (1.1) for derangements is easy to use because the terms decrease rapidly. Moreover, this formula is exceptionally simple, largely because $N_\ge(Q)$ depends only on $|Q|$. In general, the inclusion-exclusion principle produces complicated sums that are hard to estimate. A frequently helpful tool is provided by the *Bonferroni inequalities* [81, 351]. One form of these inequalities is that for any integer $m \ge 0$,

$$N_=(R) \ge \sum_{\substack{Q \\ R \subseteq Q \subseteq P \\ |Q \setminus R| \le 2m}} (-1)^{|Q \setminus R|} N_\ge(Q) \tag{5.56}$$

and

$$N_=(R) \le \sum_{\substack{Q \\ R \subseteq Q \subseteq P \\ |Q \setminus R| \le 2m+1}} (-1)^{|Q \setminus R|} N_\ge(Q) \; . \tag{5.57}$$

Thus in general

$$\Big| N_=(R) - \sum_{\substack{Q \\ R \subseteq Q \subseteq P \\ |Q \setminus R| \le k}} (-1)^{|Q \setminus R|} N_\ge(Q) \Big| \le \sum_{\substack{Q \\ R \subseteq Q \subseteq P \\ |Q \setminus R| \le k+1}} N_\ge(Q) \; . \tag{5.58}$$

These inequalities are frequently applied for $n = |X|$ increasing. Typically one chooses k that increases much more slowly than n, so that the individual terms $N_\ge(Q)$ in (5.58) can be estimated asymptotically, as the interactions of the different properties counted by $N_\ge(Q)$ is not too complicated to estimate. Bender [33] presents some useful general principles to be used in such estimates (especially the asymptotically Poisson distribution that tends to occur when the method is successful). We present an adaptation of an example from [33].

Example 5.7. *Balls and cells.* Given n labeled cells and m labeled balls, let $a_h(m,n)$ be the number of ways to place the balls into cells so that exactly h of the cells are empty. We consider h fixed. Let X be the ways of placing the balls into the cells (n^m in total), and $P = \{P_1, \dots, P_n\}$, where P_i is the property that the i-th cell is empty. If $R = \{P_1, \dots, P_h\}$, then $a_h(m,n) = \binom{n}{h} N_=(R)$. Now

$$N_\geq(Q) = (n - |Q|)^m \ , \tag{5.59}$$

so

$$\begin{aligned} \sum_{\substack{Q \\ R \subseteq Q \subseteq P \\ |Q \setminus R| = t}} N_\geq(Q) &= \tbinom{n-h}{t}(n-h-t)^m \\ &= n^m e^{-mh/n}(ne^{-m/n})^t (t!)^{-1}(1 + O((t^2+1)mn^{-2} + (t^2+1)n^{-1})) \ , \end{aligned} \tag{5.60}$$

provided $t^2 \leq n$ and $mt^2n^{-2} \leq 1$, say. In the range $0 \leq t \leq \log n$, $n \log n \leq m \leq n^2(\log n)^{-3}$, we find that the right-hand side of (5.60) is

$$n^m e^{-mh/n}(ne^{-m/n})^t (t!)^{-1}(1 + O(mn^{-2}(\log n)^2)) \ .$$

We now apply (5.58) with $k = \lfloor \log n \rfloor$, and obtain

$$\begin{aligned} a_h(m,n) &= \tbinom{n}{h} N_=(R) \sim \tbinom{n}{h} n^m \exp(-mh/n - ne^{-m/n}) \\ &\sim n^m (h!)^{-1} (ne^{-m/n})^h \exp(-ne^{-m/n}) \end{aligned} \tag{5.61}$$

as $m, n \to \infty$, provided $n \log n \leq m \leq n^2(\log n)^{-3}$. Since $a_h(m,n)n^{-m}$ is the probability that there are exactly h empty cells, the relation (5.61) (which we have established only for fixed h) shows that this probability is asymptotically distributed like a Poisson random variable with parameter $n \exp(-m/n)$.

Many additional results on random distributions of balls into cells, and references to the extensive literature on this subject can be found in [241]. ■

Bonferroni inequalities include other methods for estimating $N_=(R)$ by linear combinations of the $N_\geq(Q)$. Recent approaches and references (phrased in probabilistic terms) can be found in [152]. For bivariate Bonferroni inequalities (where one asks for the probability that at least one of two sets of events occurs) see [153, 249].

The Chen-Stein method [75] is a powerful technique that is often used in place of the principle of inclusion-exclusion, especially in probabilistic literature. Recent references are [17, 27].

5.3. Euler-Maclaurin and Poisson summation formulas

Section 5.0 showed that sums can be successfully approximated by integrals if the summands are all small compared to the total sum and vary smoothly as functions of the summation index. The approximation (5.29), though crude, is useful in a wide variety of cases. Sometimes, though, more accurate approximations are needed. An obvious way is to improve the bound (5.29). If $g(x)$ is really smooth, we can expect that the difference

$$a_n - \int_n^{n+1} g(u)du$$

will vary in a regular way with n. This is indeed the case, and it is exploited by the Euler-Maclaurin summation formula. It can be found in many books, such as [63, 175, 297, 298]. There are many formulations, but they do not differ much.

Euler-Maclaurin summation formula. Suppose that $g(x)$ has $2m$ continuous derivatives in $[a,b]$, $a,b \in Z$. Then

$$\begin{aligned}\sum_{k=a}^{b} g(k) &= \int_a^b g(x)dx + \sum_{r=1}^{m} \frac{B_{2r}}{(2r)!}\left\{g^{(2r-1)}(b) - g^{(2r-1)}(a)\right\} \\ &\quad + \frac{1}{2}\{g(a)+g(b)\} + R_m \ ,\end{aligned} \tag{5.62}$$

where

$$R_m = -\int_a^b g^{(2m)}(x)\frac{B_{2m}(x-\lfloor x\rfloor)}{(2m)!}dx \ , \tag{5.63}$$

and so

$$|R_m| \le \int_a^b |g^{(2m)}(x)|\frac{|B_{2m}(x-\lfloor x\rfloor)|}{(2m)!}dx \ . \tag{5.64}$$

In the above formulas, the $B_n(x)$ denote the Bernoulli polynomials, defined by

$$\frac{ze^{xz}}{e^z-1} = \sum_{n=0}^{\infty} B_n(x)\frac{z^n}{n!} \ . \tag{5.65}$$

The B_n are the Bernoulli numbers, defined by

$$\frac{z}{e^z-1} = \sum_{n=0}^{\infty} B_n\frac{z^n}{n!} \ , \tag{5.66}$$

so that $B_n = B_n(0)$, and

$$\begin{aligned} B_0 &= 1 \ , \quad B_1 = -1/2 \ , \quad B_2 = 1/6 \ , \\ B_3 &= B_5 = B_7 = \cdots = 0 \ , \\ B_4 &= -1/30 \ , \quad B_6 = 1/42 \ , \quad B_8 = -1/30, \ \ldots \ . \end{aligned} \tag{5.67}$$

It is known that

$$|B_{2m}(x - \lfloor x \rfloor)| \le |B_{2m}| \ , \tag{5.68}$$

so we can simplify (5.64) to

$$|R_m| \le |B_{2m}|((2m)!)^{-1} \int_a^b |g^{(2m)}(x)| dx \ . \tag{5.69}$$

There are many applications of the Euler-Maclaurin formula. One of the most frequently cited ones is to estimate factorials.

Example 5.8. *Stirling's formula.* We transform the product in the definition of $n!$ into a sum by taking logarithms, and find that for $g(x) = \log x$ and $m = 1$ we have

$$\log n! = \sum_{k=1}^{n} \log k = \int_1^n (\log x) dx + \frac{1}{2} \log n + \frac{1}{2} B_2 \left\{ \frac{1}{n} - 1 \right\} + R_1 \ , \tag{5.70}$$

where

$$R_1 = \int_1^n \frac{B_2(x - \lfloor x \rfloor)}{2x^2} dx = C + O(n^{-1}) \tag{5.71}$$

for

$$C = \int_1^\infty \frac{B_2(x - \lfloor x \rfloor)}{2x^2} dx \ . \tag{5.72}$$

Therefore

$$\log n! = n \log n - n + \frac{1}{2} \log n + C + 13/12 + O(n^{-1}) \ , \tag{5.73}$$

which gives

$$n! \sim C' n^{1/2} n^n e^{-n} \quad \text{as} \quad n \to \infty \ . \tag{5.74}$$

To obtain Stirling's formula (4.1), we need to show that $C' = (2\pi)^{1/2}$. This can be done in several ways (cf. [63]). In Examples 12.1, 12.4, and 12.5 we will see other methods of deriving (4.1). ■

There is no requirement that the function $g(x)$ in the Euler-Maclaurin formula be positive. That was not even needed for the crude approximation of a sum by an integral given in Section 5.0. The function $g(x)$ can even take complex values. (After all, Eq. (5.62) is an identity!) However, in most applications this formula is used to derive an asymptotic estimate with a small error term. For that, some high order derivatives have to be small, which means that $g(x)$ cannot change sign too rapidly. In particular, the Euler-Maclaurin formula usually is not very useful when the $g(k)$ alternate in sign. In those cases one can sometimes use

the differencing trick (cf. Example 5.5) and apply the Euler-Maclaurin formula to $h(k) = g(2k) + g(2k+1)$. There is also Boole's summation formula for alternating sums that can be applied. (See Chapter 2, §3 and Chapter 6, §6 of [298], for example.) Generalizations to other periodic patterns in the coefficients have been derived by Berndt and Schoenfeld [47].

The bounds for the error term R_m in the Euler-Maclaurin formula that were stated above can often be improved by using special properties of the function $g(x)$. For example, when $g(x)$ is analytic in x, there are contour integrals for R_m that sometimes give good estimates (cf. [315]).

The Poisson summation formula states that

$$\sum_{n=-\infty}^{\infty} f(n+a) = \sum_{m=-\infty}^{\infty} \exp(2\pi i m a) \int_{-\infty}^{\infty} f(y) \exp(-2\pi i m y) dy \tag{5.75}$$

for "nice" functions $f(x)$. The functions for which (5.75) holds include all continuous $f(x)$ for which $\int |f(x)| dx < \infty$, which are of bounded variation, and for which $\sum_n f(n+a)$ converges for all a. For weaker conditions that ensure validity of (5.75), we refer to [63, 365]. The Poisson summation formula often converts a slowly convergent sum into a rapidly convergent one. Generally it is not as widely applicable as the Euler-Maclaurin formula as it requires extreme regularity for the Fourier coefficients to decrease rapidly. On the other hand, it can be applied in some situations that are not covered by the Euler-Maclaurin formula, including some where the coefficients vary in sign.

Example 5.9. *Sum of* $\exp(-\alpha k^2)$. We consider again the function $h(\alpha)$ of Example 5.3. We let $f(x) = \exp(-\alpha x^2)$, $a = 0$. Eq. (5.15) then gives

$$h(\alpha) = \sum_{n=-\infty}^{\infty} \exp(-\alpha n^2) = (\pi/\alpha)^{1/2} \sum_{m=-\infty}^{\infty} \exp(-\pi^2 m^2/\alpha) \; . \tag{5.76}$$

This is an identity, and the sum on the right-hand side above converges rapidly for small α. Many applications require the evaluation of the sum on the left in which α tends to 0. Eq. (5.76) offers a method of converting a slowly convergent sum into a tractable one, whose asymptotic behavior is explicit. ■

5.4. Bootstrapping and other basic methods

Bootstrapping is a useful technique that uses asymptotic information to obtain improved estimates. Usually we start with some rough bounds, and by combining them with the relations defining the function or sequence that we are studying, we obtain better bounds.

Example 5.10. *Approximation of Bell numbers.* Example 5.4 obtained the asymptotics of the Bell numbers B_n, but only in terms of w, the solution to Eq. (5.41). We now show how to obtain asymptotic expansions for w. As n increases, so does w. Therefore $\log(w+1)$ also increases, and so $w < n$ for large n. Thus

$$n = w\log(w+1) < w\log(n+1) \ ,$$

and so

$$n(\log(n+1))^{-1} < w < n \ . \tag{5.77}$$

Therefore

$$\log(w+1) = \log n + O(\log\log n) \ , \tag{5.78}$$

and so

$$w = \frac{n}{\log(w+1)} = \frac{n}{\log n} + O\left(\frac{n\log\log n}{(\log n)^2}\right) \ . \tag{5.79}$$

To go further, note that by (5.79),

$$\begin{aligned} \log(w+1) &= \log\left(\frac{n}{\log n}\left(1 + O\left(\frac{\log\log n}{\log n}\right)\right)\right) \\ &= \log n - \log\log n + O((\log\log n)(\log n)^{-1}) \ , \end{aligned} \tag{5.80}$$

and so by applying this estimate in Eq. (5.41), we obtain

$$w = \frac{n}{\log n} + \frac{n\log\log n}{(\log n)^2} + \frac{n(\log\log n)^2}{(\log n)^3} + O\left(\frac{n\log\log n}{(\log n)^3}\right) \ . \tag{5.81}$$

This procedure can be iterated indefinitely to obtain expansions for w with error terms $O(n(\log n)^{-\alpha})$ for as large a value of α as desired. ■

In the above example, w can also be estimated by other methods, such as the Lagrange-Bürmann inversion formula (cf. Example 6.7). However, the bootstrapping method is much more widely applicable and easy to apply. It will be used several times later in this chapter.

5.5. Estimation of integrals

In some of the examples in the preceding sections integrals were used to approximate sums. The integrals themselves were always easy to evaluate. That is true in most asymptotic enumeration problems, but there do occur situations where the integrals are more complicated. Often the hard integrals are of the form

$$f(x) = \int_\alpha^\beta g(t)\exp(xh(t))dt \ , \tag{5.82}$$

and it is necessary to estimate the behavior of $f(x)$ as $x \to \infty$, with the functions $g(t)$, $h(t)$ and the limits of integration α and β held fixed. There is a substantial theory of such integrals, and good references are [54, 63, 100, 315]. The basic technique is usually referred to as Laplace's method, and consists of approximating the integrand by simpler functions near its maxima. This approach is similar to the one that is discussed at length in Section 5.1 for estimating sums. The contributions of the approximations are then evaluated, and it is shown that the remaining ranges of integration, away from the maxima, contribute a negligible amount. By breaking up the interval of integration we can write the integral (5.82) as a sum of several integrals of the same type, with the property that there is a unique maximum of the integrand and that it occurs at one of the endpoints. When $\alpha > 0$, the maximum of the integrand occurs for large x at the maximum of $h(t)$ (except in rare cases where $g(t) = 0$ for that t for which $h(t)$ is maximized). Suppose that the maximum occurs at $t = \alpha > 0$. It often happens that

$$h(t) = h(\alpha) - c(t-\alpha)^2 + O(|t-\alpha|^3) \tag{5.83}$$

for $\alpha \le t \le \beta$ and $c = -h''(\alpha)/2 > 0$, and then one obtains the approximation

$$f(x) \sim g(\alpha) \exp(xh(\alpha))[-\pi/(4xh''(\alpha))]^{1/2} \quad \text{as} \quad x \to \infty \ , \tag{5.84}$$

provided $g(\alpha) \neq 0$. For precise statements of even more general and rigorous results, see for example Chapter 3, §7 of [315]. Those results cover functions $h(t)$ that behave near $t = \alpha$ like $h(\alpha) - c(t-\alpha)^\mu$ for any $\mu > 0$.

When the integral is highly oscillatory, as happens when $h(t) = iu(t)$ for a real-valued function $u(t)$, still other techniques (such as the stationary phase method), are used. We will not present them here, and refer to [54, 63, 100, 315] for descriptions and applications. In Section 12.1 we will discuss the saddle point method, which is related to both Laplace's method and the stationary phase method.

Laplace integrals

$$F(x) = \int_0^\infty f(t) \exp(-xt) dt \tag{5.85}$$

can often be approximated by integration by parts. We have (under suitable conditions on $f(t)$)

$$\begin{aligned} F(x) &= x^{-1} f(0) + x^{-1} \int_0^\infty f'(t) \exp(-xt) dt \\ &= x^{-1} f(0) + x^{-2} f'(0) + x^{-2} \int_0^\infty f''(t) \exp(-xt) dt \ , \end{aligned} \tag{5.86}$$

and so on. There are general results, usually associated with the name of Watson's Lemma, for deriving such expansions. For references, see [100, 315].

6. Generating functions

6.1. A brief overview

Generating functions are a wonderfully powerful and versatile tool, and most asymptotic estimates are derived from them. The most common ones in combinatorial enumeration are the ordinary and exponential generating functions. If $a_0, a_1, \ldots$, is any sequence of real or complex numbers, the *ordinary generating function* is

$$f(z) = \sum_{n=0}^{\infty} a_n z^n \ , \tag{6.1}$$

while the *exponential generating function* is

$$f(z) = \sum_{n=0}^{\infty} \frac{a_n z^n}{n!} \ . \tag{6.2}$$

Doubly-indexed arrays, for example $a_{n,k}$, $0 \le n < \infty$, $0 \le k \le n$, are encoded as two-variable generating functions. Depending on the array, sometimes one uses

$$f(x,y) = \sum_{n=0}^{\infty} \sum_{k=0}^{n} a_{n,k} x^k y^n \ , \tag{6.3}$$

and sometimes other forms that might even mix ordinary and exponential types, as in

$$f(x,y) = \sum_{n=0}^{\infty} \frac{y^n}{n!} \sum_{k=0}^{n} a_{n,k} x^k \ . \tag{6.4}$$

For example, the Stirling numbers of the first kind, $s(n,k)$ ($(-1)^{n+k} s(n,k)$ is the number of permutations on n letters with k cycles) have the generating function (see pp. 50, 212–213, and 234–235 in [81])

$$1 + \sum_{n=1}^{\infty} \frac{y^n}{n!} \sum_{k=1}^{n} s(n,k) x^k = (1+y)^x \ . \tag{6.5}$$

In general, a generating function is just a formal power series, and questions of convergence do not arise in the definition. However, some of the main applications of generating functions in asymptotic enumeration do rely on analyticity or other convergence properties of those functions, and there the domain of convergence is important.

A generating function is just another form for the sequence that defines it. There are many reasons for using it. One is that even for complicated sequences, generating functions are

frequently simple. This might not be obvious for the partition function $p(n)$, which has the ordinary generating function

$$f(z) = \sum_{n=0}^{\infty} p(n) z^n = \prod_{k=1}^{\infty} (1 - z^k)^{-1} \ . \tag{6.6}$$

The sequence $p(n)$, which is complicated, is encoded here as an infinite product. The terms in the product are simple and vary in a regular way with the index, but it is not clear at first what is gained by this representation. In other cases, though, the advantages of generating functions are clearer. For example, the exponential generating function for derangements (Eq. (1.1) and Example 5.6) is

$$\begin{aligned} f(z) &= \sum_{n=0}^{\infty} \frac{d_n}{n!} z^n = \sum_{n=0}^{\infty} \frac{z^n}{n!} \sum_{n=0}^{n} (-1)^k \frac{n!}{k!} \\ &= \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \sum_{n=k}^{\infty} z^n = \frac{e^{-z}}{1-z} \ , \end{aligned} \tag{6.7}$$

which is extremely compact.

Reasons for using generating functions go far beyond simplicity. The one that matters most for this chapter in that generating functions can be used to obtain information about the asymptotic behavior of sequences they encode, information that often cannot be obtained in any other way, or not as easily. Methods such as those of Section 10.2 can be used to obtain immediately from Eq. (6.7) the asymptotic estimate $d_n \sim e^{-1} n!$ as $n \to \infty$. This estimate can also be derived easily by elementary methods from Eq. (1.1), so here the generating function is not essential. In other cases, though, such as that of the partition function $p(n)$, all the sharp estimates, such as that of Hardy and Ramanujan given in (1.5), are derived by exploiting the properties of the generating function. If there is any main theme to this chapter, it is that generating functions are usually the easiest, most versatile, and most powerful way to study asymptotic behavior of sequences. Especially when the generating function is analytic, its behavior at the dominant singularities (a term that will be defined in Section 10) determines the asymptotics of the sequence. When the generating function is simple, and often even when it is not simple, the contribution of the dominant singularity can often be determined easily, although the sequence itself is complicated.

There are many applications of generating functions, some related to asymptotic questions. Averages can often be studied using generating functions. Suppose, for example, that $a_{n,k}$, $0 \le k \le n$, $0 \le n < \infty$, is the number of objects in some class of size n, which have weight k

(for some definition of size and weight), and that we know, either explicitly or implicitly, the generating function $f(x, y)$ of $a_{n,k}$ given by (6.4). Then

$$g(y) = f(1, y) = \sum_{n=0}^{\infty} \frac{y^n}{n!} \sum_{k=0}^{n} a_{n,k} \tag{6.8}$$

is the exponential generating function of the number of objects of size n, while

$$h(y) = \frac{\partial}{\partial x} f(x, y)\Big|_{x=1} = \sum_{n=0}^{\infty} \frac{y^n}{n!} \sum_{k=0}^{n} k a_{n.k} \tag{6.9}$$

is the exponential generating function of the sum of the weights of objects of size n. Therefore the average weight of an object of size n is

$$\frac{[y^n]h(y)}{[y^n]g(y)} \ . \tag{6.10}$$

The wide applicability and power of generating functions come primarily from the structured way in which most enumeration problems arise. Usually the class of objects to be counted is derived from simpler objects through basic composition rules. When the generating functions are chosen to reflect appropriately the classes of objects and composition rules, the final generating function is derivable in a simple way from those of the basic objects. Suppose, for example, that each object of size n in class C can be decomposed uniquely into a pair of objects of sizes k and $n - k$ (for some k) from classes A and B, and each pair corresponds to an object in C. Then c_n, the number of objects of size n in C, is given by the convolution

$$c_n = \sum_{k=0}^{n} a_k b_{n-k} \ , \tag{6.11}$$

(where a_k is the number of objects of size k in A, etc.). Hence if $A(z) = \sum a_n z^n$, $B(z) = \sum b_n z^n$, $C(z) = \sum c_n z^n$ are the ordinary generating functions, then

$$C(z) = A(z)B(z) \ . \tag{6.12}$$

Thus ordered pairing of objects corresponds to multiplication of ordinary generating functions. If $A(z) = \sum a_n z^n$ and

$$b_n = \sum_{k=0}^{n} a_k \ ,$$

then $B(z) = \sum b_n z^n$ is given by

$$B(z) = \frac{A(z)}{1 - z} \ , \tag{6.13}$$

so that the ordinary generating function of cumulative sums of coefficients is obtained by dividing by $1 - z$. There are many more such general correspondences between operations on combinatorial objects and on the corresponding generating functions. They are present, implicitly or explicitly, in most books that cover combinatorial enumeration, such as [81, 173, 351, 377]. The most systematic approach to developing and using general rules of this type has been carried out by Flajolet and his collaborators [139]. They develop ways to see immediately (cf. [134]) that if we consider mappings of a set of n labeled elements to itself, so that all n^n distinct mappings are considered equally likely, then the generating function for the longest path length is given by

$$f(z) = \sum_{k=0}^{\infty} \left(\frac{1}{1 - t(z)} - e^{v_k(z)} \right) , \tag{6.14}$$

where

$$v_k(z) = t_{k-1}(z) + \frac{1}{2} t_{k-2}(z)^2 + \cdots + \frac{1}{k} t_0(k)^k , \tag{6.15}$$

with

$$t_0(z) = z , \quad t_{h+1}(z) = z \exp(t_h(z)) , \tag{6.16}$$

and $t(z) = \lim_{h \to \infty} t_h(z)$ (in the sense of formal power series, so convergence is that of coefficients). Furthermore, as is mentioned in Section 17, many of these rules for composition of objects and generating functions can be implemented algorithmically, automating some of the chores of applying them.

We illustrate some of the basic generating function techniques by deriving the generating function for rooted labeled trees, which will occur later in Examples 6.6 and 10.8. (The rooted unlabeled trees, with generating function given by (1.8), are harder.)

Example 6.1. *Rooted labeled trees.* Let t_n be the number of rooted labeled trees on n vertices, so that $t_1 = 1$, $t_2 = 2$, $t_3 = 9$. (It will be shown in Example 6.6 that $t_n = n^{n-1}$.) Let

$$t(z) = \sum_{n=1}^{\infty} t_n \frac{z^n}{n!} \tag{6.17}$$

be the exponential generating function. If we remove the root of a rooted labeled tree with n vertices, we are left with $k \geq 0$ rooted labeled trees that contain a total of $n - 1$ vertices. The total number of ways of arranging an ordered selection of k rooted trees with a total of $n - 1$ vertices is

$$[z^{n-1}] t(z)^k .$$

Since the order of the trees does not matter, we have

$$\frac{1}{k!}[z^{n-1}]t(z)^k$$

different trees of size n that have exactly k subtrees, and so

$$\begin{aligned} t_n &= \sum_{k=0}^{\infty} \frac{1}{k!}[z^{n-1}]t(z)^k \\ &= [z^{n-1}]\sum_{k=0}^{\infty} t(z)^k/k! \;=\; [z^n]z\exp(t(z)) \;, \end{aligned} \tag{6.18}$$

which gives

$$t(z) = z\exp(t(z)) \;. \tag{6.19}$$

As an aside, the function $t_h(z)$ of Eq (6.16) is the exponential generating function of rooted labeled trees of height $\leq h$. ■

The key to the successful use of generating functions is to use a generating function that is of the appropriate form for the problem at hand. There is no simple rule that describes what generating function to use, and sometimes two are used simultaneously. In combinatorics and analysis of algorithms, the most useful forms are the ordinary and exponential generating functions, which reflects how the classes of objects that are studied are constructed. Sometimes other forms are used, such as the double exponential form

$$f(z) = \sum_{n=0}^{\infty} \frac{a_n z^n}{(n!)^2} \tag{6.20}$$

that occurs in Section 7, or the Newton series

$$f(z) = \sum_{n=0}^{\infty} a_n z(z-1)\cdots(z-n+1) \;. \tag{6.21}$$

Also frequently encountered are various q-analog generating functions, such as the Eulerian

$$f(z) = \sum_{n=1}^{\infty} \frac{a_n z^n}{(1-q)(1-q^2)\cdots(1-q^n)} \;. \tag{6.22}$$

In multiplicative number theory, the most common are Dirichlet series

$$f(z) = \sum_{n=1}^{\infty} a_n n^{-z} \;, \tag{6.23}$$

which reflect the multiplicative structure of the integers. If a_n is a multiplicative function (so that $a_{mn} = a_m a_n$ for all relatively prime positive integers m and n) then the function (6.23)

has an Euler product representation

$$f(z) = \prod_p (1 + a_p p^{-z} + a_{p^2} p^{-2z} + \cdots) \ , \tag{6.24}$$

where p runs over the primes. This allows new tools to be used to study $f(z)$ and through it a_n. Additive problems in combinatories and number theory often are handled using functions such as functions such as

$$f(z) = \sum_{n=1}^{\infty} z^{a_k} \ , \tag{6.25}$$

where $0 \le a_1 < a_2 < \cdots$ is a sequence of integers. Addition of two such sequences then corresponds to a multiplication of the generating functions of the form (6.25).

We next mention the "snake oil method." This is the name given by Wilf [377] to the use of generating functions for proving identities, and comes from the surprising power of this technique. The typical application is to evaluation of sequences given by sums of the type

$$a_n = \sum_k b_{n,k} \ . \tag{6.26}$$

The standard procedure is to form a generating function of the a_n and manipulate it through interchanges of summation and other tricks to obtain the final answer. The generating function can be ordinary, exponential, or (less commonly) of another type, depending on what gives the best results. We show a simple application of this principle that exhibits the main features of the method.

Example 6.2. *A binomial coefficient sum* [377]. Let

$$a_n = \sum_{k=0}^{n} \binom{n+k}{2k} 2^{n-k} \ , \quad n \ge 0 \ . \tag{6.27}$$

We define $A(z)$ to be the ordinary generating function of a_n. We find that

$$\begin{aligned} A(z) &= \sum_{n=0}^{\infty} a_n z^n = \sum_{n=0}^{\infty} z^n \sum_{k=0}^{n} \binom{n+k}{2k} 2^{n-k} \\ &= \sum_{k=0}^{\infty} 2^{-k} \sum_{n=k}^{\infty} 2^n z^n \binom{n+k}{2k} = \sum_{k=0}^{\infty} 2^{-k} (2z)^{-k} \sum_{n=0}^{\infty} \binom{n+k}{2k} (2z)^{n+k} \\ &= \sum_{k=0}^{\infty} 2^{-k} (2z)^{-k} \frac{(2z)^{2k}}{(1-2z)^{2k+1}} = \frac{1}{1-2z} \sum_{k=0}^{\infty} \left(\frac{z}{1-2z} \right)^k \\ &= \frac{1-2z}{(1-4z)(1-z)} = \frac{2}{3(1-4z)} + \frac{1}{3(1-z)} \ . \end{aligned} \tag{6.28}$$

Therefore we immediately find the explicit form

$$a_n = (2^{2n+1} + 1)/3 \quad \text{for} \quad n \geq 0 \ . \tag{6.29}$$

■

We next present some additional examples of how generating functions are derived. We start by considering linear recurrences with constant coefficients.

The first step in solving a linear recurrence is to obtain its generating function. Suppose that a sequence $a_0, a_1, a_2, \ldots$ satisfies the recurrence

$$a_n = \sum_{i=1}^{d} c_i a_{n-i}, \quad n \geq d \ . \tag{6.30}$$

Then

$$\begin{aligned} f(z) &= \sum_{n=0}^{\infty} a_n z^n = \sum_{n=0}^{d-1} a_n z^n + \sum_{n=d}^{\infty} z^n \sum_{i=1}^{d} c_i a_{n-i} \\ &= \sum_{n=0}^{d-1} a_n z^n + \sum_{i=1}^{d} c_i z^i \sum_{n=d}^{\infty} a_{n-i} z^{n-i} \\ &= \sum_{n=0}^{d-1} a_n z^n + \sum_{i=1}^{d} c_i z^i \left(f(z) - \sum_{n=0}^{d-i-1} a_n z^n \right) \ , \end{aligned} \tag{6.31}$$

and so

$$f(z) = \frac{g(z)}{1 - \sum_{i=1}^{d} c_i z^i} \ , \tag{6.32}$$

where

$$g(z) = \sum_{n=0}^{d-1} a_n z^n - \sum_{i=1}^{d} c_i z^i \sum_{n=0}^{d-i-1} a_n z^n \tag{6.33}$$

is a polynomial of degree $\leq d-1$. Eq. (6.32) is the fundamental relation in the study of linear recurrences, and $1 - \sum c_i z^i$ is called the *characteristic polynomial* of the recursion.

Example 6.3. *Fibonacci numbers.* We let $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$, and

$$F(z) = \sum_{n=0}^{\infty} F_n z^n \ .$$

Then by (6.32) and (6.33),

$$F(z) = \frac{z}{1 - z - z^2} \ . \qquad \blacksquare \tag{6.34}$$

Often there is no obvious recurrence for the sequence a_n being studied, but there is one involving some other auxiliary function. Usually if one can obtain at least as many recurrences as there are sequences, one can obtain their generating functions by methods similar to those used for a single sequence. The main additional complexity comes from the need to solve a system of linear equations with polynomial coefficients. We illustrate this with the following example.

Example 6.4. *Sequences with forbidden subwords.* Let $A = a_1 a_2 \cdots a_k$ be a binary string of length k. Define $f_A(n)$ to be the number of binary strings of length n that do not contain A as a subword of k adjacent characters. (Subsequences do not count, so that if $A = 1110$, then A is contained in 1101110010, but not in 101101.) We introduce the correlation polynomial $C_A(z)$ of A:

$$C_A(z) = \sum_{j=0}^{k-1} c_A(j) z^j \ , \tag{6.35}$$

where $c_A(0) = 1$ and for $1 \leq j \leq k-1$,

$$c_A(j) = \begin{cases} 1 & \text{if } \ a_1 a_2 \cdots a_{k-j} = a_{j+1} a_{j+2} \cdots a_k \ , \\ 0 & \text{otherwise .} \end{cases} \tag{6.36}$$

As examples, we note that if $A = 1000$, then $C_A(z) = 1$, whereas $C_A(z) = 1 + z + z^2 + z^3$ if $A = 1111$. The generating function

$$F_A(z) = \sum_{n=0}^{\infty} f_A(n) z^n \tag{6.37}$$

then satisfies

$$F_A(z) = \frac{C_A(z)}{z^k + (1-2z) C_A(z)} \ . \tag{6.38}$$

To prove this, define $g_A(n)$ to be the number of binary sequences $b_1 b_2 \cdots b_n$ of length n such that $b_1 b_2 \cdots b_k = A$, but such that $b_j b_{j+1} \cdots b_{j+k-1} \neq A$ for any j with $2 \leq j \leq n-k+1$; i.e., sequences that start with A but do not contain it any place else. We then have $g_A(n) = 0$ for $n < k$, and $g_A(k) = 1$. We also define

$$G_A(z) = \sum_{n=0}^{\infty} g_A(n) z^n \ . \tag{6.39}$$

We next obtain a relation between $G_A(z)$ and $F_A(z)$ that will enable us to determine both.

If $b_1 b_2 \cdots b_n$ is counted by $f_A(n)$, then for x either 0 or 1, the string $x b_1 b_2 \cdots b_n$ either does not contain A at all, or if it does contain it, then $A = x b_1 b_2 \cdots b_{k-1}$. Therefore for $n \geq 0$,

$$2 f_A(n) = f_A(n+1) + g_A(n+1) \tag{6.40}$$

and multiplying both sides of Eq. (6.40) by z^n and summing on $n \geq 0$ yields

$$2F_A(z) = z^{-1}(F_A(z) - 1) + z^{-1}G_A(z) \ . \tag{6.41}$$

We need one more relation, and to obtain it we consider any string $B = b_1 b_2 \cdots b_n$ that does not contain A any place inside. If we let C be the concatenation of A and B, so that $C = a_1 a_2 \cdots a_k b_1 b_2 \cdots b_n$, then C starts with A, and may contain other occurrences of A, but only at positions that overlap with the initial A. Therefore we obtain,

$$f_A(n) = \sum_{\substack{j=1 \\ c_A(k-j)=1}}^{k} g_A(n+j) \quad \text{for} \quad n \geq 0 \ , \tag{6.42}$$

and this gives the relation

$$F_A(z) = z^{-k} C_A(z) G_A(z) \ . \tag{6.43}$$

Solving the two equations (6.41) and (6.43), we find that $F_A(z)$ satisfies (6.38), while

$$G_A(z) = \frac{z^k}{z^k + (1-2z)C_A(z)} \ . \tag{6.44}$$

The proof above follows that in [182], except that [182] uses generating functions in z^{-1}, so the formulas look different. Applications of the formulas (6.38) and (6.44) will be found later in this chapter, as well as in [182, 130]. Other approaches to string enumeration problems are referenced there as well. Other approaches and applications of string enumerations are given in the references to [182] and in papers such as [18]. ■

The above example can be generalized to provide generating functions that enumerate sequences in which any of a given set of patterns are forbidden [182].

Whenever one has a finite system of linear recurrences with constant coefficients that involve several sequences, say $a_n^{(i)}$, $1 \leq i \leq k$, $n \geq 0$, one can translate these recurrences into linear equations with polynomial coefficients in the generating functions $A^{(i)}(z) = \sum a_n^{(i)} z^n$ for these sequences. To obtain the $A^{(i)}(z)$, one then needs to solve the resulting system. Such solutions will exist if the matrix of polynomial coefficients is nonsingular over the field of rational functions in z. In particular, one needs at least as many equations (i.e., recurrence relations) as k, the number of sequences, and if there are exactly as many equations as sequences, then the determinant of the matrix of the coefficients has to be a nonzero polynomial.

One interesting observation is that when a system of recurrences involving several sequences is solved by the above method, each of the generating functions $A^{(i)}(z)$ is a rational function

in z. What this means is that each of the sequences $a_n^{(i)}$, $1 \le i \le k$, satisfies a linear recurrence with constant coefficients that does not involve any of the other $a_n^{(j)}$ sequences! In principle, therefore, that recurrence could have been found right at the beginning by combinatorial methods. However, usually the degree of the recurrence for an isolated $a_n^{(j)}$ sequence is high, typically about k times as large as the average degree of the k recurrences involving all the $a_n^{(j)}$. Thus the use of several sequences $a_n^{(j)}$ leads to much simpler and combinatorially more appealing relations.

That generating functions can significantly simplify combinatorial problems is shown by the following example. It is taken from [349], and is a modification of a result of Klarner [229] and Pólya [321]. This example also shows a more complicated derivation of explicit generating functions than the simple ones presented so far.

Example 6.5. *Polyomino enumeration* [349]. Let a_n be the number of n-square polyominoes P that are inequivalent under translation, but not necessarily under rotation or reflection, and such that each row of P is an unbroken line of squares. Then $a_1 = 1$, $a_2 = 2$, $a_3 = 6$. We define $a_0 = 0$. It is easily seen that

$$a_n = \sum (m_1 + m_2 - 1)(m_2 + m_3 - 1) \cdots (m_{s-1} + m_s - 1) \ , \tag{6.45}$$

where the sum is over all ordered partitions $m_1 + \cdots + m_s = n$ of n into positive integers m_i. Let $a_{r,n}$ be the sum of terms in (6.45) with $m_1 = r$, where we set $a_{n,n} = 1$, and $a_{r,n} = 0$ if $r > n$ or $n < 0$. Then

$$a_n = \sum_{r=1}^{\infty} a_{r,n} \ , \tag{6.46}$$

$$a_{r,n} = \sum_{i=1}^{\infty} (r + i - 1) a_{i,n-r} \ , \qquad r < n \ . \tag{6.47}$$

Define

$$A(x, y) = \sum_{n=1}^{\infty} \sum_{r=1}^{\infty} a_{r,n} x^r y^n \ , \tag{6.48}$$

so that

$$A(1, y) = \sum_{n=1}^{\infty} a_n y^n \tag{6.49}$$

is the generating function of the a_n, which are what we need to estimate.

By (6.47), we find that

$$A(x, y) = \sum_{n=1}^{\infty} x^n y^n + \sum_{n=1}^{\infty} \sum_{r=1}^{\infty} \sum_{i=1}^{\infty} (r + i - 1) a_i (n - r) x^r y^n$$

$$= \frac{xy}{1-xy} + \frac{x^2y^2}{(1-xy)^2}A(1,y) + \frac{xy}{1-xy}G(x,y) \ , \tag{6.51}$$

where

$$G(y) = \sum_{n=1}^{\infty}\sum_{i=1}^{\infty} i a_{i,n} y^n = \frac{\partial}{\partial x}A(x.y)\Big|_{x=1} \ , \tag{6.52}$$

We now set $x = 1$ in (6.50) and obtain an equation involving $A(1, y)$ and $G(y)$, namely

$$A(1,y) = \frac{y}{1-y} + \frac{y^2}{(1-y)^2}A(1,y) + \frac{y}{1-y}G(y) \ . \tag{6.53}$$

We next differentiate (6.50) with respect to x, and set $x = 1$. This gives us a second equation,

$$G(y) = \frac{y}{(1-y)^2} + \frac{2y^2}{(1-y)^3}A(1,y) + \frac{y}{(1-y)^2}G(y) \ . \tag{6.54}$$

We now eliminate $G(y)$ from (6.53) and (6.54) to obtain

$$A(1,y) = \frac{y(1-y)^3}{1-5y+7y^2-4y^3} \ . \tag{6.55}$$

This formula shows that

$$a_{n+3} = a_{n+2} - 7a_{n+1} + 4a_n \quad \text{for} \quad n \geq 2 \ . \tag{6.56}$$

Using the results of Section 10 we can easily obtain from (6.55) an asymptotic estimate

$$a_n \sim c\alpha^n \quad \text{as} \quad n \to \infty \ , \tag{6.57}$$

where c is a certain constant and $\alpha = 3.205569\ldots$ is the inverse of the smallest zero of $1 - 5y + 7y^2 - 4y^3$. ■

For other methods and results related to polyomino enumeration, see [326, 327].

6.2. Composition and inversion of power series

So far we have only discussed simple operations on generating functions, such as multiplication. What happens when we do something more complicated? There are several frequently occurring operations on generating functions whose results can be described explicitly.

Faà di Bruno's formula [81]. Suppose that

$$A(z) = \sum_{m=0}^{\infty} a_m \frac{z^m}{m!} \ , \quad B(z) = \sum_{n=0}^{\infty} b_n \frac{z^n}{n!} \ , \tag{6.58}$$

are two exponential generating functions with $b_0 = 0$. Then the formal composition $C(z) = A(B(z))$ is well-defined, and

$$C(z) = \sum_{n=0}^{\infty} c_n \frac{z^n}{n!} \tag{6.59}$$

with

$$c_0 = 0, \quad c_n = \sum_{k=1}^{n} a_k B_{n,k}(b_1, b_2, \ldots, b_{n-k+1}) \ , \tag{6.60}$$

where the $B_{n,k}$ are the exponential Bell polynomials defined by

$$\sum_{n,k=0}^{\infty} B_{n,k}(x_1, \ldots, x_{n-k+1}) \frac{t^n u^k}{n!} = \exp\left(u \sum_{m=1}^{\infty} x_m \frac{t^m}{m!} \right) \ , \tag{6.61}$$

with the x_j independent variables.

Faà di Bruno's formula makes it possible to compute successive derivatives of functions such as $\log A(z)$ in terms of the derivatives of $A(z)$. For further examples, see [81, 335, 336]. Faà di Bruno's formula is derivable in a straightforward way from the multinomial theorem.

Composition of generating functions occurs frequently in combinatorics and analysis of algorithms. When it yields the desired generating function as a composition of several known generating functions, the basic problem is solved, and one can work on the asymptotics of the coefficients using Faà di Bruno's formula or other methods. A more frequent event is that the composition yields a functional equation for the generating function, as in Example 6.1, where the exponential generating function $t(z)$ for labeled rooted trees was shown to satisfy $t(z) = z \exp(t(z))$. General functional equations are hard to deal with. (Many examples will be presented later.) However, there is a class of them for which an old technique, the Lagrange-Bürmann inversion formula, works well. We start by noting that if

$$f(z) = \sum_{n=0}^{\infty} f_n z^n \tag{6.62}$$

is a formal power series with $f_0 = 0$, $f_1 \neq 0$, then there is an inverse formal power series $f^{\langle -1 \rangle}(z)$ such that

$$f(f^{\langle -1 \rangle}(z)) = f^{\langle -1 \rangle}(f(z)) = z \ . \tag{6.63}$$

The coefficients of $f^{\langle -1 \rangle}(z)$ can be expressed explicitly in terms of the coefficients of $f(z)$. More generally, we have the following result.

Lagrange-Bürmann inversion formula. Suppose that $f(z)$ is a formal power series with $[z^0]f(z) = 0$, $[z^1]f(z) \neq 0$, and that $g(z)$ is any formal power series. Then for $n \geq 1$,

$$[z^n]\{g(f^{\langle -1 \rangle})(z)\} = n^{-1}[z^{n-1}]\{g'(z)(f(z)/z)^{-n}\} \ . \tag{6.64}$$

In particular, for $g(z) = z$, we have

$$[z^n]f^{\langle -1\rangle}(z) = n^{-1}[z^{n-1}](f(z)/z)^{-n} \ . \tag{6.65}$$

Example 6.6. *Rooted labeled trees.* As was shown in Example 6.1, the exponential generating function of rooted labeled trees satisfies $t(z) = z\exp(t(z))$. If we rewrite it as $z = t(z)\exp(-t(z))$, we see that $t(z) = f^{\langle -1\rangle}(z)$, where $f(z) = z\exp(-z)$. Therefore Eq. (6.65) yields

$$\begin{aligned} [z^n]t(z) &= n^{-1}[z^{n-1}]\exp(-nz) \\ &= n^{-1}n^{n-1}/(n-1)! \ = \ n^{n-1}/n! \ , \end{aligned} \tag{6.66}$$

which shows that t_n, the number of rooted labeled trees on n nodes, is n^{n-1}. ■

Proof of a form of the Lagrange-Bürmann theorem is given in Chapter ?. Extensive discussion, proofs, and references are contained in [81, 173, 205, 375]. Some additional recent references are [159, 208]. There exist generalizations of the Lagrange-Bürmann formula to several variables [173, 169, 208].

The Lagrange-Bürmann formula, as stated above, is valid for general formal power series. If $f(z)$ is analytic in a neighborhood of the origin, then so are $f^{\langle -1\rangle}(z)$ and $g(f^{\langle -1\rangle})(z)$, provided $g(z)$ is also analytic near 0 and $f'(0) \neq 0$, $f(0) = 0$. Most of the presentations of this inversion formula in the literature assume analyticity. However, that is not a real restriction. To prove (6.65), say, in full generality, it suffices to prove it for any n. Given n, if we let

$$F(z) = \sum_{k=0}^{n} f_k z^k \ , \quad G(z) = \sum_{k=0}^{n} g_k z^k \ ,$$

then we see that

$$[z^n]\{g(f^{\langle -1\rangle})(z)\} = [z^n]G(F^{\langle -1\rangle})(z) \ , \tag{6.67}$$

and $F(z)$ and $G(z)$ are analytic, so the formula (6.65) can be applied. Thus combinatorial proofs of the Lagrange-Bürmann formula do not offer greater generality than analytic ones.

While the analytic vs. combinatorial distinction in the proofs of the Lagrange-Bürmann formula does not matter, it is possible to use analyticity of the functions $f(z)$ and $g(z)$ to obtain useful information. Example 6.6 above was atypical in that a simple explicit formula

was derived. Often the quantity on the right-hand side of (6.64) is not explicit enough to make clear its asymptotic behavior. When that happens, and $g(z)$ and $f(z)$ are analytic, one can use the contour integral representation

$$[z^{n-1}]\{g'(z)(f(z)/z)^{-n}\} = \frac{1}{2\pi i}\int_\Gamma g'(z)f(z)^{-n}dz \ , \tag{6.68}$$

where Γ is a positively oriented simple closed contour enclosing the origin that lies inside the region of analyticity of both $g(z)$ and $f(z)$. This representation, which is discussed in Section 10, can often be used to obtain asymptotic information about coefficients $[z^n]g(f^{\langle -1\rangle})(z)$ (cf. [273]).

The Lagrange-Bürmann formula can provide numerical approximations to roots of equations and even convergent infinite series representations for such roots. An important case is the trinomial equation $y = z(1 + y^r)$, and there are many others.

Example 6.7. *Dominant zero for forbidden subword generating functions.* The generating functions $F_A(z)$ and $G_A(z)$ of Example 6.4 both have denominators

$$h(z) = z^k + (1 - 2z)C(z) \ , \tag{6.69}$$

where $C(z)$ is a polynomial of degree $\le k$, with coefficients 0 and 1, and with $C(0) = 1$. It will be shown later that $h(z)$ has only one zero ρ of small absolute value, and that this zero is the dominant influence on the asymptotic behavior of the coefficients of $F_A(z)$ and $G_A(z)$. Right now we obtain accurate estimates for ρ.

For simplicity, we will consider only large k. Since $C(z)$ has nonnegative coefficients and $C(0) = 1$, $h(3/4) \le (3/4)^k - 1/2 < 0$ for $k \ge 3$. On the other hand, $h(1/2) = 2^{-k}$. Therefore $h(z)$ has a real zero ρ with $1/2 < \rho < 3/4$. As $k \to \infty$, $\rho \to 1/2$, since

$$\rho^k = (2\rho - 1)C(\rho) \ , \tag{6.70}$$

and $\rho^k \to 0$ as $k \to \infty$ for $1/2 < \rho < 3/4$, while $2\rho - 1$ and $C(\rho)$ are bounded. We can deduce from (6.69) that

$$2\rho - 1 \sim 2^{-k}C(1/2)^{-1} \quad \text{as} \quad k \to \infty \ , \tag{6.71}$$

uniformly for all polynomials $C(z)$ of the prescribed type. By applying the bootstrapping technique (see Section 5.4) we can find even better approximations. By (6.71),

$$C(\rho) \ = \ C(1/2) + O(|\rho - 1/2|) \ = \ C(1/2) + O(2^{-k}) \ , \tag{6.72}$$

$$\rho^k \ = \ 2^{-k}(1 + O(2^{-k}))^k \ = \ 2^{-k}(1 + O(k2^{-k})) \ , \tag{6.73}$$

so (6.70) now yields

$$\rho = 1/2 + 2^{-k-1}C(1/2)^{-1} + O(k2^{-2k}) \ . \tag{6.74}$$

Even better approximations can be obtained by repeating the process using (6.74). At the next stage we would apply the expansion

$$\begin{aligned} C(\rho) &= C(1/2) + (\rho - 1/2)C'(1/2) + O((\rho - 1/2)^2) \\ &= C(1/2) + 2^{-k-1}C'(1/2) + O(k2^{-2k}) \end{aligned} \tag{6.75}$$

and a similar one for ρ^k.

A more systematic way to obtain a rapidly convergent series for ρ is to use the inversion formula. If we set $u = \rho - 1/2$, then (6.70) can be rewritten as $w(u) = 1$, where

$$w(u) = 2uC(1/2+u)(1/2+u)^{-k} = \sum_{j=1}^{\infty} a_j u^j \ , \tag{6.76}$$

with

$$a_1 = 2^{k+1}C(1/2) \neq 0 \ . \tag{6.77}$$

Hence $u = w^{\langle -1 \rangle}(1)$, and the Lagrange-Bürmann inversion formula (6.65) yields the coefficients of $w^{\langle -1 \rangle}(z)$. In particular, we find that

$$\rho = 1/2+u \approx 1/2+2^{-k-1}C(1/2)^{-1}+k2^{-2k-1}C(1/2)^{-2}-2^{-2k-2}C'(1/2)C(1/2)^{-3}+\cdots \tag{6.78}$$

as a Poincaré asymptotic series. With additional work one can show that the series (6.78) converges, and that

$$\begin{aligned} \rho = 1/2 &+ 2^{-k-1}C(1/2)^{-1} + k2^{-2k-1}C(1/2)^{-2} \\ &- 2^{-2k-2}C'(1/2)C(1/2)^{-3} + O(k^2 2^{-3k}) \ , \end{aligned} \tag{6.79}$$

for example. The same estimate can be obtained by the bootstrapping technique. ■

6.3. Differentiably finite power series

Homogeneous recurrences with constant coefficients are the nicest large set of sequences one can imagine, with rational generating functions, and well-understood asymptotic behavior. The next class in complexity consists of the polynomially-recursive or, *P-recursive sequences*, $a_0, a_1, \ldots$, which satisfy recurrences of the form

$$p_d(n)a_{n+d} + p_{d-1}(n)a_{n+d-1} + \cdots + p_0(n)a_n = 0, \quad n \geq 0 \ , \tag{6.80}$$

where d is fixed and $p_0(n), \ldots, p_d(n)$ are polynomials in n. Such sequences are common in combinatorics, with $a_n = n!$ a simple example. Normally P-recursive sequences do not have explicit forms for their generating functions. In this section we briefly summarize some of their main properties. Asymptotic properties of P-recursive sequences will be discussed in Section 9.2. The main references for the results quoted here are [254, 350].

A formal power series

$$f(z) = \sum_{k=0}^{\infty} a_k z^k \tag{6.81}$$

is called differentiably finite, or *D-finite*, if the derivatives $f^{(n)}(z) = \frac{d^n f(z)}{dz^n}$, $n \geq 0$, span a finite-dimensional vector space over the field of rational functions with complex coefficients. The following three conditions are equivalent for a formal power series $f(z)$:

i) $f(z)$ is D-finite.

ii) There exist finitely many polynomials $q_0(z), \ldots, q_k(z)$ and a polynomial $q(z)$, not all 0, such that

$$q_k(z) f^{(k)}(z) + \cdots + q_0(z) f(z) = q(z) \ . \tag{6.82}$$

iii) There exist finitely many polynomials $p_0(z), \ldots, p_m(z)$, not all 0, such that

$$p_m(z) f^{(m)}(z) + \cdots + p_0(z) f(z) = 0 \ . \tag{6.83}$$

The most important result for combinatorial enumeration is that a sequence $a_0, a_1, \ldots$, is P-recursive if and only if its ordinary generating function $f(z)$, defined by (6.81), is D-finite. This makes it possible to apply results that are more easily proved for D-finite power series.

If $f(z)$ is D-finite, then so is the power series obtained by changing a finite number of the coefficients of $f(z)$. If $f(z)$ is algebraic (i.e., there exist polynomials $q_0(z), \ldots, q_d(z)$, not all 0, such that $q_d(z) f(z)^d + \cdots + q_0(z) f(z) + q_0(z) = 0$), then $f(z)$ is D-finite. The product of two D-finite power series is also D-finite, as is any linear combination with polynomial coefficients. Finally, the Hadamard product of two D-finite series is D-finite. The proofs rely on elementary linear algebra constructions. An important feature of the theory is that identity between D-finite series is decidable.

The concept of a D-finite power series can be extended to several variables [254, 405], and there are generalizations of P-recursiveness [254, 405]. (See also [161].) Zeilberger [405] has used the word *holonomic* to describe corresponding sequences and generating functions.

When we investigate a sequence $\{a_n\}$, sometimes the combinatorial context yields only relations for more complicated object with several indices. While we might like to obtain the generating function $f(z) = \sum a_n z^n$, we might instead find a formula for a generating function

$$F(z_1, z_2, \ldots, z_k) = \sum_{n_1,\ldots,n_k} b_{n_1}, \ldots, n_k z_1^{n_1}, \ldots, z_k^{n_k} \ , \tag{6.84}$$

where $a_n = b_{n,n,\ldots,n}$, say. When this happens, we say that $f(z)$ is a *diagonal* of $F(z_1, \ldots, z_k)$. (There are more general definitions of diagonals in [90, 253, 254, 255], which are recent references for this topic.) Diagonals of D-finite power series in any number of variables are D-finite. Diagonals of two-variable rational functions are algebraic, but there are three-variable rational functions whose diagonals are not algebraic [151].

6.4. Unimodality and log-concavity

A finite sequence $a_0, a_1, \ldots, a_n$ of real numbers is called *unimodal* if for some index k, $a_0 \le a_1 \le \cdots \le a_k$ and $a_k \ge a_{k+1} \ge \cdots \ge a_n$. A sequence $a_0, \ldots, a_n$ of nonnegative elements is called *log-concave* (short for logarithmically concave) if $a_j^2 \ge a_{j-1}a_{j+1}$ holds for $1 \le j \le n-1$. Unimodal and log-concave sequences occur frequently in combinatorics and are objects of intensive study. We present a brief review of some of their properties because asymptotic methods are often used to prove unimodality and log-concavity. Furthermore, knowledge that a sequence is log-concave or unimodal is often helpful in obtaining asymptotic information. For example, some methods provide only asymptotic estimates for summatory functions of sequences, and unimodality helps in obtaining from those estimates bounds on individual coefficients. This approach will be presented in Section 13, in the discussion of central and local limit theorems.

The basic references for unimodality and log-concavity are [222, 352]. For recent results, see also [56] and the references given there. All the results listed below can be found in those sources and the references they list.

In the rest of this subsection we will consider only sequences of nonnegative elements. A sequence $a_0, \ldots, a_n$ will be said to *have no internal zeros* if there is no triple of integers $0 \le i < j < k \le n$ such that $a_j = 0$, $a_i a_k \ne 0$. It is easy to see that a log-concave sequence with no internal zeros is unimodal, but there are sequences of positive elements that are unimodal but not concave. The convolution of two unimodal sequences does not have to be unimodal. However, it is unimodal if each of the two unimodal sequences is also symmetric.

Convolution of two log-concave sequences is log-concave. The convolution of a log-concave and a unimodal sequence is unimodal. A log-concave sequence is even characterized by the property that its convolution with any unimodal sequence is unimodal. This last property is related to the variation-diminishing character of log-concave sequences (see [222]), which we will not discuss at greater length here except to note that there are more restrictive sets of sequences (the Pólya frequency classes, see [56, 222]) which have stronger convolution properties.

The binomial coefficients $\binom{n}{k}$, $0 \le k \le n$, are log-concave, and therefore unimodal. The q-binomial coefficients $\begin{bmatrix} n \\ k \end{bmatrix}_q$ are log-concave for any $q \ge 1$. On the other hand, if we write a single coefficient $\begin{bmatrix} n \\ k \end{bmatrix}_q$ for fixed n and k as a polynomial in q, the sequence of coefficients is unimodal, but does not have to be log-concave.

The most frequently used method for showing that a sequence $a_0, \ldots, a_n$ is log-concave is to show that all the zeros of the polynomial

$$A(z) = \sum_{k=0}^{n} a_k z^k \tag{6.85}$$

are real and ≤ 0. In that case not only are the a_k log-concave, but so are $a_k \binom{n}{k}^{-1}$. Absolute values of the Stirling numbers of both kinds were first shown to be log-concave by this method [195]. There are many unsolved conjectures about log-concavity of combinatorial sequences, such as the Read-Hoggar conjecture that coefficients of chromatic polynomials are log-concave (cf. [57]).

A variety of combinatorial, algebraic, and geometric methods have been used to prove unimodality of sequences, and we refer the reader to [352] for a comprehensive and insightful survey. In Section 12.3 we will discuss briefly some proofs of unimodality and log-concavity that use asymptotic methods. The basic philosophy is that since the Gaussian distribution is log-concave and unimodal (when we extend the definition of these concepts to continuous distributions), these properties should also hold for sequences that by the central limit theorem or its variants are asymptotic to the Gaussian. Therefore one can expect high-order convolutions of sequences to be log-concave at least in their central region, and there are theorems that prove this under certain conditions.

6.5. Moments and distributions

The second moment method is a frequently used technique in probabilistic arguments, as is shown in Chapter ? and [55, 108, 348]. It is based on *Chebyshev's inequality*, which says

that if X is a real-valued random variable with finite second moment $E(X^2)$, then

$$\text{Prob}\left(|X - E(X)| \geq \alpha |E(X)|\right) \leq \frac{E(X^2) - E(X)^2}{\alpha^2 E(X)^2} \ . \tag{6.86}$$

An easy corollary of inequality (6.86) that is often used is

$$\text{Prob}\left(X = 0\right) \leq \frac{E(X^2) - E(X)^2}{E(X)^2} \ . \tag{6.87}$$

(There is a slightly stronger version of the inequality (6.87), in which $E(X)^2$ in the denominator is replaced by $E(X^2)$.) The inequalities (6.86) and (6.87) are usually applied for $X = Y_1 + \cdots + Y_n$, where the Y_j are other random variables. The helpful feature of the inequalities is that they require only knowledge of the pairwise dependencies among the Y_j, which is easier to study than the full joint distribution of the Y_j. For other bounds on distributions that can be obtained from partial information about moments, see [343].

The reason moment bounds are mentioned at all in this chapter is that asymptotic methods are often used to derive them. Generating functions are a common and convenient method for doing this.

Example 6.8. *Waiting times for subwords.* In a continuation and application of Example 6.4, let A be a binary string of length k. How many tosses of a fair coin (with sides labeled 0 and 1) are needed on average before A appears as a block of k consecutive outcomes? By a general observation of probability theory, this is just the sum over $n \geq 0$ of the probability that A does not appear in the first n coin tosses, and thus equals

$$\sum_{n=0}^{\infty} f_A(n) 2^{-n} = F_A(1/2) = 2^k C_A(1/2) \ , \tag{6.88}$$

where the last equality follows from Eq. (6.38). Another, more general, way to derive this is to use $G_A(z)$. Note that $g_A(n)2^{-n}$ is the probability that A appears in the first n coin tosses, but not in the first $n-1$. Hence the r-th moment of the time until A appears is

$$\sum_{n=0}^{\infty} n^r g_A(n) 2^{-n} = \left(z \frac{d}{dz}\right)^r G_A(z) \Big|_{z=1/2} \ . \tag{6.89}$$

If we take $r = 1$, we again obtain the expected waiting time given by (6.88). When we take $r = 2$, we find that the second moment of the time until the appearance of A is

$$\sum_{n=0}^{\infty} n^2 g_A(n) 2^{-n} = 2^{2k+1} C_A(1/2)^2 - (2k-1) 2^k C_A(1/2) + 2^k C_A'(1/2) \ , \tag{6.90}$$

and therefore the variance is

$$\begin{aligned} & 2^{2k}C_A(1/2)^2 - (2k-1)2^k C_A(1/2) + 2^k C'_A(1/2) \\ & \quad = \; 2^{2k}C_A(1/2)^2 + O(k2^k) \; , \end{aligned} \tag{6.91}$$

since $1 \leq C_A(1/2) \leq 2$. Higher moments can be used to obtain more detailed information. A better approach is to use the method of Example 9.2, which gives precise estimates for the tails as well as the mean of the distribution. ■

Information about moments of distribution functions can often be used to obtain the limiting distribution. If $F_n(x)$ is a sequence of distribution functions such that for every integer $k \geq 0$, the k-th moment

$$\mu_n(k) = \int x^k dF_n(x) \tag{6.92}$$

converges to $\mu(k)$ as $n \to \infty$, then there is a limiting measure with distribution function $F(x)$ whose k-th moment is $\mu(k)$. If the moments $\mu(k)$ do not grow too rapidly, then they determine the distribution function $F(x)$ uniquely, and the $F_n(x)$ converge to $F(x)$ (in the weak star sense [50]). A sufficient condition for the $\mu(k)$ to determine $F(x)$ uniquely is that the generating function

$$U(x) = \sum_{k=0}^{\infty} \frac{\mu(2k)x^k}{(2k)!} \tag{6.93}$$

should converge for some $x > 0$. In particular, the standard normal distribution with

$$F(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} \exp(-u^2/2)du \tag{6.94}$$

has $\mu(2k) = 1 \cdot 3 \cdot 5 \cdot 7 \cdot \ldots \cdot (2k-1)$ (and $\mu(2k+1) = 0$), so it is determined uniquely by its moments. On the other hand, there are some frequently encountered distributions, such as the log-normal one, which do not have this property.

7. Formal power series

This section discusses generating functions $f(z)$ that might not converge in any interval around the origin. Sequences that grow rapidly are common in combinatorics, with $a_n = n!$ the most obvious example for which

$$f(z) = \sum_{n=0}^{\infty} a_n z^n \tag{7.1}$$

does not converge for any $z \neq 0$. The usual way to deal with the problem of a rapidly growing sequence a_n is to study the generating function of a_n/b_n, where b_n is some sequence with

known asymptotic behavior. When $b_n = n!$, the ordinary generating function of a_n/b_n is then the exponential generating function of a_n. For derangements (Eqs. (1.1) and (6.7)) this works well, as the exponential generating function of d_n converges in $|z| < 1$ and has a nice form. Unfortunately, while we can always find a sequence b_n that will make the ordinary generating function $f(z)$ of a_n/b_n converge (even for all z), usually we cannot do it in a way that will yield any useful information about $f(z)$. The combinatorial structure of a problem almost always severely restricts what forms of generating function can be used to take advantage of the special properties of the problem. This difficulty is common, for example, in enumeration of labeled graphs. In such cases one often resorts to formal power series that do not converge in any neighborhood of the origin. For example, if $c(n,k)$ is the number of connected labeled graphs on n vertices with k edges, then it is well known (cf. [349]) that

$$\sum_{n=0}^{\infty}\sum_{k=0}^{\infty} c(n,k)\frac{x^k y^n}{n!} = \log\left(\sum_{m=0}^{\infty}\frac{(1+x)^{\binom{m}{2}}y^m}{m!}\right) . \tag{7.2}$$

While the series inside the log in (7.2) does converge for $-2 \le x \le 0$, and any y, it diverges for any $x > 0$ as long as $y \neq 0$, and so this is a relation of formal power series.

There are few methods for dealing with asymptotics of formal power series, at least when compared to the wealth of techniques available for studying analytic generating functions. Fortunately, combinatorial enumeration problems that do require the use of formal power series often involve rapidly growing sequences of positive terms, for which some simple techniques apply. We start with an easy general result that is applicable both to convergent and purely formal power series.

Theorem 7.1. ([33]) *Suppose that* $a(z) = \sum a_n z^n$ *and* $b(z) = \sum b_n z^n$ *are power series with radii of convergence* $\alpha > \beta \ge 0$*, respectively. Suppose that* $b_{n-1}/b_n \to \beta$ *as* $n \to \infty$*. If* $a(\beta) \neq 0$*, and* $\sum c_n z^n = a(z)b(z)$*, then*

$$c_n \sim a(\beta)b_n \quad as \quad n \to \infty . \tag{7.3}$$

The proof of Theorem 7.1, which can be found in [33], is simple. The condition $\alpha > \beta$ is important, and cannot be replaced by $\alpha = \beta$. We can have $\beta = 0$, and that is indeed the only possibility if the series for $b(z)$ does not converge in a neighborhood of $z = 0$.

Example 7.1. *Double set coverings* [33, 80]. Let v_n be the number of choices of subsets $S_1, \ldots, S_r$ of an n-element set T such that each $t \in T$ is in exactly two of the S_i. There is

no restriction on r, the number of subsets, and some of the S_i can be repeated. Let c_n be the corresponding number when the S_i are required to be distinct. We let $C(z) = \sum c_n z^n/n!$, $V(z) = \sum v_n z^n/n!$ be the exponential generating functions. Then it can be shown that

$$C(z) = \exp(-1 - (e^z - 1)/2)A(z) \ , \tag{7.4}$$

$$V(z) = \exp(-1 + (e^z - 1)/2)A(z) \ , \tag{7.5}$$

where

$$A(z) = \sum_{k=0}^{\infty} \exp(k(k-1)z/2)/k! \ . \tag{7.6}$$

We see immediately that $A(z)$ does not converge in any neighborhood of the origin. We have

$$a_n = [z^n]A(z) = 2^{-n} \sum_{k=2}^{\infty} \frac{k^n (k-1)^n}{k!} \ . \tag{7.7}$$

By considering the ratio of consecutive terms in the sum in (7.7), we find that the largest term occurs for $k = k_0$ with $k_0 \log k_0 \sim 2n$, and by the methods of Section 5.1 we find that

$$a_n \sim \frac{\pi^{1/2} k_0^n (k_0 - 1)^n}{n^{1/2} 2^n (k_0 - 1)!} \quad \text{as} \quad n \to \infty \ . \tag{7.8}$$

Therefore $a_{n-1}/a_n \to 0$ as $n \to \infty$, and Theorem 7.1 tells us that

$$c_n \sim v_n \sim e^{-1} n! a_n \quad \text{as} \quad n \to \infty \ . \tag{7.9}$$

■

Usually formal power series occur in more complicated relations than those covered by Theorem 7.1. For example, if f_n is the number of connected graphs on n labeled vertices which have some property, and F_n is the number of graphs on n labeled vertices each of whose connected components has that property, then (cf. [394])

$$1 + \sum_{n=1}^{\infty} F_n \frac{x^n}{n!} = \exp\left(\sum_{n=1}^{\infty} f_n \frac{x^n}{n!} \right) \ . \tag{7.10}$$

Theorem 7.2. ([34]) *Suppose that*

$$\begin{aligned} a(x) &= \sum_{n=1}^{\infty} a_n x^n \ , \quad F(x,y) = \sum_{h,k \geq 0} f_{hk} x^h y^k \ , \\ b(x) &= \sum_{n=0}^{\infty} b_n x^n = F(x, a(x)) \ , \quad D(x) = F_y(x, a(x)) \ , \end{aligned} \tag{7.11}$$

where $F_y(x,y)$ is the partial derivative of $F(x,y)$ with respect to y. Assume that $a_n \neq 0$ and

(i)

$$a_{n-1} = o(a_n) \quad as \quad n \to \infty \ , \tag{7.12}$$

(ii)

$$\sum_{k=r}^{n-r} |a_k a_{n-k}| = O(a_{n-r}) \quad for\ some \quad r > 0 \ , \tag{7.13}$$

(iii) for every $\delta > 0$ there are $M(\delta)$ and $K(\delta)$ such that for $n \geq M(\delta)$ and $h + k > r + 1$,

$$|f_{hk} a_{n-h-k+1}| \leq K(\delta)\delta^{h+k} |a_{n-r}| \ . \tag{7.14}$$

Then

$$b_n = \sum_{k=0}^{r-1} d_k a_{n-k} + O(a_{n-r}) \ . \tag{7.15}$$

Condition (iii) of Theorem 7.2 is often hard to verify. Theorem 2 of [34] shows that this condition holds under certain simpler hypotheses. It follows from that result that (iii) is valid if $F(x, y)$ is analytic in x and y in a neighborhood of $(0, 0)$. Hence, if $F(x, y) = \exp(y)$ or $F(x, y) = 1+y$, then Theorem 7.2 becomes easy to apply. One can also deduce from Theorem 2 of [34] that Theorem 7.2 applies when (i) and (ii) hold, $b_0 = 0$, $b_n \geq 0$, and

$$1 + a(z) = \exp\left(\sum_{k=1}^{\infty} b(z^k)/k \right) \ , \tag{7.16}$$

another relation that is common in graph enumeration (cf. Example 15.1). There are also some results weaker than Theorem 7.2 that are easier to apply [393].

Example 7.2. *Indecomposable permutations* [81]. For every permutation σ of $\{1, \ldots, n)$, let $\{1, \ldots, n\} = \cup I_h$, where the I_h are the smallest intervals such that $\sigma(I_h) = I_h$ for all h. For example, $\sigma = (134)(2)(56)$ corresponds to $I_1 = \{1, 2, 3, 4\}$, $I_2 = \{5, 6\}$, and the identity permutation has n components. A permutation is said to be indecomposable if it has one component. For example, if σ has the 2-cycle $(1n)$, it is indecomposable. Let c_n be the number of indecomposable permutations of $\{1, \ldots, n\}$. Then [81]

$$\sum_{n=1}^{\infty} c_n z^n = 1 - \frac{1}{1 + \sum_{n=1}^{\infty} n! z^n} \ . \tag{7.17}$$

We apply Theorem 7.2 with $a_n = n!$ for $n \geq 1$ and $F(x, y) = 1 - (1 + y)^{-1}$. We easily obtain

$$c_n \sim n! \quad \text{as} \quad n \to \infty \ , \tag{7.18}$$

so that almost all permutations are indecomposable. ■

Some further useful expansions for functional inverses and computations of formal power series have been obtained by Bender and Richmond [40].

8. Elementary estimates for convergent generating functions

The word "elementary" in the title of this section is a technical term that means the proofs do not use complex variables. It does not necessarily imply that the proofs are simple. While some, such as those of Section 8.1, are easy, others are more complicated. The main advantage of elementary methods is that they are much easier to use, and since they impose much weaker requirements on the generating functions, they are more widely applicable. Usually they only impose conditions on the generating function $f(z)$ for $z \in \mathbb{R}^+$.

The main disadvantage of elementary methods is that the estimates they give tend to be much weaker than those derived using analytic function approaches. It is easy to explain why that is so by considering the two generating functions

$$f_1(z) = \sum_{n=0}^{\infty} z^n = (1-z)^{-1} \tag{8.1}$$

and

$$f_2(z) = 3/2 + \sum_{n=1}^{\infty} 2z^{2n} = 3/2 + 2z^2(1-z^2)^{-1} \ . \tag{8.2}$$

Both series converge for $|z| < 1$ and diverge for $|z| > 1$, and both blow up as $z \to 1$. However,

$$f_1(z) - f_2(z) = -\frac{1-z}{2(1+z)} \to 0 \quad \text{as} \quad z \to 1 \ . \tag{8.3}$$

Thus these two functions behave almost identically near $z = 1$. Since $f_1(z)$ and $f_2(z)$ are both $\sim (1-z)^{-1}$ as $z \to 1^-$, $z \in \mathbb{R}^+$, and their difference is $O(|z-1|)$ for $z \in \mathbb{R}^+$, it would require exceptionally delicate methods to detect the differences in the coefficients of the $f_j(z)$ just from their behavior for $z \in \mathbb{R}^+$. There is a substantial difference in the behavior of $f_1(z)$ and $f_2(z)$ for real z if we let $z \to -1$, so our argument does not completely exclude the possibility of obtaining detailed information about the coefficients of these functions using methods of real variables only. However, if we consider the function

$$f_3(z) = 2 + \sum_{n=1}^{\infty} 3z^{3n} = 2 + 3z^3(1-z^3)^{-1} \ , \tag{8.4}$$

then $f_1(z)$ and $f_3(z)$ are both $\sim (1-z)^{-1}$ as $z \to 1^-$, $z \in \mathbb{R}^+$, yet now

$$|f_1(z) - f_3(z)| = O(|z-1|) \quad \text{for all} \quad z \in \mathbb{R} \ .$$

This difference is comparable to what would be obtained by modifying a single coefficient of one generating function. To determine how such slight changes in the behavior of the generating functions affect the behavior of the coefficients we would need to know much more about the functions if we were to use real variable methods. On the other hand, analytic methods, discussed in Section 10 and later, are good at dealing with such problems. They require less precise knowledge of the behavior of a function on the real line. Instead, they impose weak conditions on the function in a wider domain, namely that of the complex numbers.

For reasons discussed above, elementary methods cannot be expected to produce precise estimates of individual coefficients. They often do produce good estimates of summatory functions of the coefficients, though. In the examples above, we note that

$$\sum_{n=1}^{N} [z^n] f_j(z) \sim N \quad \text{as} \quad N \to \infty \tag{8.5}$$

for $1 \le j \le 3$. This holds because the $f_j(z)$ have the same behavior as $z \to 1^-$, and is part of a more general phenomenon. Good knowledge of the behavior of the generating function on the real axis combined with weak restrictions on the coefficients often leads to estimates for the summatory function of the coefficients.

There are cases where elementary methods give precise bounds for individual coefficients. Typically when we wish to estimate f_n, with ordinary generating function $f(z) = \sum f_n z^n$ that converges for $|z| < 1$ but not for $|z| > 1$, we apply the methods of this section to

$$g_n = f_n - f_{n-1} \quad \text{for} \quad n \ge 1, \quad g_0 = f_0 \tag{8.6}$$

with generating function

$$g(z) = \sum_{n=0}^{\infty} g_n z^n = (1-z) f(z) \ . \tag{8.7}$$

Then

$$\sum_{k=0}^{n} g_k = f_n \ , \tag{8.8}$$

and so estimates of the summatory function of the g_k yield estimates for f_n. The difficulty with this approach is that now $g(z)$ and not $f(z)$ has to satisfy the hypotheses of the theorems, which requires more knowledge of the f_n. For example, most of the Tauberian theorems apply only to power series with nonnegative coefficients. Hence to use the differencing trick above to obtain estimates for f_n we need to know that $f_{n-1} \le f_n$ for all n. In some cases (such as that of $f_n = p_n$, the number of ordinary partitions of n) this is easily seen to hold

through combinatorial arguments. In other situations where one might like to apply elementary methods, though, $f_{n-1} \le f_n$ is either false or else is hard to prove. When that happens, other methods are required to estimate f_n.

8.1. Simple upper and lower bounds

A trivial upper bound method turns out to be widely applicable in asymptotic enumeration, and is surprisingly powerful. It relies on nothing more than the nonnegativity of the coefficients of a generating function.

Lemma 8.1. *Suppose that $f(z)$ is analytic in $|z| < R$, and that $[z^n]f(z) \ge 0$ for all $n \ge 0$. Then for any x, $0 < x < R$, and any $n \ge 0$,*

$$[z^n]f(z) \le x^{-n} f(x) \ . \tag{8.9}$$

Example 8.1. *Lower bound for factorials.* Let $f(z) = \exp(z)$. Then Lemma 8.1 yields

$$\frac{1}{n!} = [z^n]e^z \le x^{-n}e^x \tag{8.10}$$

for every $x > 0$. The logarithm of $x^{-n}e^x$ is $x - n\log x$, and differentiating and setting it equal to 0 shows that the minimum value is attained at $x = n$. Therefore

$$\frac{1}{n!} = [z^n]e^z \le n^{-n}e^n \ , \tag{8.11}$$

and so $n! \ge n^n e^{-n}$. This lower bound holds uniformly for all n, and is off only by an asymptotic factor of $(2\pi n)^{1/2}$ from Stirling's formula (4.1). ■

Suppose that $f(z) = \sum f_n z^n$. Lemma 8.1 is proved by noting that for $0 < x < R$, the n-th term, $f_n x^n$, in the power series expansion of $f(x)$, is $\le f(x)$. As we will see in Section 10, it is often possible to derive a similar bound on the coefficients f_n even without assuming that they are nonnegative. However, the proof of Lemma 8.1 shows something more, namely that

$$f_0 x^{-n} + f_1 x^{-n+1} + \cdots + f_{n-1}x^{-1} + f_n \le x^{-n} f(x) \tag{8.12}$$

for $0 < x < R$. When $x \le 1$, this yields an upper bound for the summatory function of the coefficients. Because (8.12) holds, we see that the bound of Lemma 8.1 cannot be sharp in general. What is remarkable is that the estimates obtainable from that lemma are often not far from best possible.

Example 8.2. *Upper bound for the partition function.* Let $p(n)$ denote the partition function. It has the ordinary generating function

$$f(z) = \sum_{n=0}^{\infty} p(n) z^n = \prod_{k=1}^{\infty} (1 - z^k)^{-1} \ . \tag{8.13}$$

Let $g(s) = \log f(e^{-s})$, and consider $s > 0$, $s \to 0$. There are extremely accurate estimates of $g(s)$. It is known [13, 23], for example, that

$$g(s) = \pi^2/(6s) + (\log s)/2 - (\log 2\pi)/2 - s/24 + O(\exp(-4\pi^2/s)) \ . \tag{8.14}$$

If we use (8.14), we find that $x^{-n} f(x)$ is minimized at $x = \exp(-s)$ with

$$s = \pi/(6n)^{1/2} - 1/(4n) + O(n^{-3/2}) \ , \tag{8.15}$$

which yields

$$p(1) + p(2) + \cdots + p(n) \le 2^{-3/4} e^{-1/4} n^{-1/4} (1 + o(1)) \exp(2\pi 6^{-1/2} n^{1/2}) \ . \tag{8.16}$$

Comparing this to the asymptotic formula for the sum that is obtainable from (1.6) (see Example 5.2), we see that the bound of (8.16) is too high by a factor of $n^{1/4}$. If we use (8.16) to bound $p(n)$ alone, we obtain a bound that is too large by a factor of $n^{3/4}$.

The application of Lemma 8.1 outlined above depended on the expansion (8.14), which is complicated to derive, involving modular transformation properties of $p(n)$ that are beyond the scope of this survey. (See [13, 23] for derivations.) Weaker estimates that are still useful are much easier to derive. We obtain one such bound here, since the arguments illustrate some of the methods from the preceding sections.

Consider

$$g(s) = \sum_{k=1}^{\infty} -\log(1 - e^{-ks}) \ . \tag{8.17}$$

If we replace the sum by the integral

$$I(s) = \int_1^{\infty} -\log(1 - e^{-us}) du \ , \tag{8.18}$$

we find on expanding the logarithm that

$$I(s) = \int_1^{\infty} \left(\sum_{m=1}^{\infty} m^{-1} e^{-mus} \right) du = s^{-1} \sum_{m=1}^{\infty} m^{-2} e^{-ms} \ , \tag{8.19}$$

since the interchange of summation and integration is easy to justify, as all the terms are positive. Therefore as $s \to 0^+$,

$$sI(s) \to \sum_{m=1}^{\infty} m^{-2} = \pi^2/6 \ , \tag{8.20}$$

so that $I(s) \sim \pi^2/(6s)$ as $s \to 0^+$. It remains to show that I is indeed a good approximation to $g(s)$. This follows easily from the bound (5.32), since it shows that

$$g(s) = I(s) + O\left(\int_1^{\infty} \frac{se^{-vs}}{1-e^{-vs}} dv\right) \ . \tag{8.21}$$

We could estimate the integral in (8.21) carefully, but we only need rough upper bounds for it, so we write it as

$$\begin{aligned} \int_1^{\infty} \frac{se^{-vs}}{1-e^{-vs}} dv &= \int_s^{\infty} \frac{e^{-u}}{1-e^{-u}} du \\ &= \int_s^1 \frac{e^{-u}}{1-e^{-u}} du + \int_1^{\infty} \frac{e^{-u}}{1-e^{-u}} du \\ &= \int_s^1 \frac{du}{e^u - 1} + c \le \int_s^1 \frac{du}{u} + c = c - \log s \end{aligned} \tag{8.22}$$

for some constant c. Thus we find that

$$g(s) = I(s) + O(\log(s^{-1})) \quad \text{as} \quad s \to 0^+ \ . \tag{8.23}$$

Combining (8.23) with (8.20) we see that

$$g(s) \sim \pi^2/(6s) \quad \text{as} \quad s \to 0^+ \ . \tag{8.24}$$

Therefore, choosing $s = \pi/(6n)^{1/2}$, $x = \exp(-s)$ in Lemma 8.1, we obtain a bound of the form

$$p(n) \le \exp((1+o(1))\pi(2/3)^{1/2} n^{1/2}) \quad \text{as} \quad n \to \infty \ . \quad \blacksquare \tag{8.25}$$

Lemma 8.1 yields a lower bound for $n!$ that is only a factor of about $n^{1/2}$ away from optimal. That is common. Usually, when the function $f(z)$ is reasonably smooth, the best bound obtainable from Lemma 8.1 will only be off from the correct value by a polynomial factor of n, and often only by a factor of $n^{1/2}$.

The estimate of Lemma 8.1 can often be improved with some additional knowledge about the f_n. For example, if $f_{n+1} \ge f_n$ for all $n \ge 0$, then we have

$$x^{-n} f(x) \ge f_n + f_{n+1}x + f_{n+2}x^2 + \cdots \ge f_n(1-x)^{-1} \ . \tag{8.26}$$

For $f_n = p(n)$, the partition function, then yields an upper bound for $p(n)$ that is too large by a factor of $n^{1/4}$.

To optimize the bound of Lemma 8.1, one should choose $x \in (0, R)$ carefully. Usually there is a single best choice. In some pathological cases the optimal choice is obtained by letting $x \to 0^+$ or $x \to R^-$. However, usually we have $\lim_{x\to R^-} f(x) = \infty$, and $[z^m]f(z) > 0$ for some m with $0 \le m < n$ as well as for some $m > n$. Under these conditions it is easy to see that

$$\lim_{x\to 0^+} x^{-n} f(x) = \lim_{x\to R^-} x^{-n} f(x) = \infty \ . \tag{8.27}$$

Thus it does not pay to make x too small or too large. Let us now consider

$$g(x) = \log(x^{-n} f(x)) = \log f(x) - n \log x \ . \tag{8.28}$$

Then

$$g'(x) = \frac{f'}{f}(x) - \frac{n}{x} \ , \tag{8.29}$$

and the optimal choice must be at a point where $g'(x) = 0$. For most commonly encountered functions $f(x)$, there exists a constant $x_0 > 0$ such that

$$\left(\frac{f'}{f}\right)'(x) > 0 \tag{8.30}$$

for $x_0 < x < R$, and so $g''(x) > 0$ for all $x \in (0, R)$ if n is large enough. For such n there is then a unique choice of x that minimizes the bound of Lemma 8.1. However, one major advantage of Lemma 8.1 is that its bound holds for all x. To apply this lemma, one can use any x that is convenient to work with. Usually if this choice is not too far from the optimal one, the resulting bound is fairly good.

We have already remarked above that the bound of Lemma 8.1 is usually close to best possible. It is possible to prove general lower bounds that show this for a wide class of functions. The method, originated in [277] and developed in [305], relies on simple elementary arguments. However, the lower bounds it produces are substantially weaker than the upper bounds of Lemma 8.1. Furthermore, to apply them it is necessary to estimate accurately the minimum of $x^{-n} f(x)$, instead of selecting any convenient values of x. A more general version of the bound below is given in [305].

Theorem 8.1. *Suppose that $f(x) = \sum f_n x^n$ converges for $|x| < 1$, $f_n \ge 0$ for all n, $f_{m_0} > 0$ for some m_0, and $\sum f_n = \infty$. Then for $n \ge m_0$, there is a unique $x_0 = x_0(n) \in (0, 1)$ that*

minimizes $x^{-n}f(x)$. *Let* $s_0 = -\log x_0$, *and*

$$A = \frac{\partial^2}{\partial s^2} \log f(e^{-s})\Big|_{s=s_0} . \tag{8.31}$$

If $A \geq 10^6$ *and for all* t *with*

$$s_0 \leq t \leq s_0 + 20A^{-1/2} \tag{8.32}$$

we have

$$\left| \frac{\partial^3}{\partial s^3} \log f(e^{-s})\Big|_{s=t} \right| \leq 10^{-3} A^{3/2} , \tag{8.33}$$

then

$$\sum_{k=0}^{n} f_k \geq x_0^{-n} f(x_0) \exp(-30 s_0 A^{1/2} - 100) . \tag{8.34}$$

As is usual for Tauberian theorems, Theorem 8.1 only provides bounds on the sum of coefficients of $f(z)$. As we mentioned before, this is unavoidable when one relies only on information about the behavior of $f(z)$ for z a positive real number. The conditions that Theorem 8.1 imposes on the derivatives are usually satisfied in combinatorial enumeration applications and are easy to verify.

Example 8.3. *Lower bound for the partition function.* Let $f(z)$ and $g(s)$ be as in Example 8.2. We showed there that $g(s)$ satisfies (8.24) and similar rough estimates show that $g'(s) \sim -\pi^2/(6s^2)$, $g''(s) \sim \pi^2/(3s^3)$, and $g'''(s) \sim -\pi^2/s^4$ as $s \to 0^+$. Therefore the hypotheses of Theorem 8.1 are satisfied, and we obtain a lower bound for $p(0) + p(1) + \cdots + p(n)$. If we only use the estimate (8.24) for $g(s)$, then we can only conclude that for $x = e^{-s}$,

$$\log(x^{-n} f(x)) = ns + g(s) \sim ns + \pi^2/(6s) \quad \text{as} \quad s \to 0 , \tag{8.35}$$

and so the minimum value occurs at $s \sim \pi/(6n)^{1/2}$ as $n \to \infty$. This only allows us to conclude that for every $\epsilon > 0$ and n large enough,

$$\log(p(0) + \cdots + p(n)) \geq (1 - \epsilon)\pi(2/3)^{1/2} n^{1/2} . \tag{8.36}$$

However, we can also conclude even without further computations that this lower bound will be within a multiplicative factor of $\exp(cn^{1/4})$ of the best upper bound that can be obtained from Lemma 8.1 for some $c > 0$ (and therefore within a multiplicative factor of $\exp(cn^{1/4})$ of the correct value). In particular, if we use the estimate (8.14) for $g(s)$, we find that for some $c' > 0$,

$$p(0) + \cdots + p(n) \geq \exp(\pi(2/3)^{1/2} n^{1/2} - c' n^{1/4}) . \tag{8.37}$$

Since $p(k) \le p(k+1)$, the quantity on the right-hand side of (8.37) is also a lower bound for $p(n)$ if we increase c', since $(n+1)p(n) \ge p(0) + \cdots + p(n)$. ■

The differencing trick described at the introduction to Section 8 could also be used to estimate $p(n)$, since Theorem 8.1 can be applied to the generating function of $p(n+1) - p(n)$. However, since the error term is a multiplicative factor of $\exp(cn^{1/4})$, it is simpler to use the approach above, which bounds $p(n)$ below by $(p(0) + \cdots + p(n))/(n+1)$.

Brigham [58] has proved a general theorem about asymptotics of partition functions that can be derived from Theorem 8.1. (For other results and references for partition asymptotics, see [13, 23, 150].)

Theorem 8.2. *Suppose that*

$$f(z) = \prod_{k=1}^{\infty} (1 - z^k)^{-b(k)} = \sum_{n=0}^{\infty} a(n) z^n \ , \tag{8.38}$$

where the $b(k) \in \mathbf{Z}$, $b(k) \ge 0$ *for all* k, *and that for some* $C > 0$, $u > 0$, *we have*

$$\sum_{k \le x} b(k) \sim C x^u (\log x)^v \quad \textit{as} \quad x \to \infty \ . \tag{8.39}$$

Then

$$\begin{aligned} \log \left(\textstyle\sum_{n \le m} a(n) \right) \sim u^{-1} \{ C u \Gamma(u+2) \zeta(u+1) \}^{1/(u+1)} \\ \cdot (u+1)^{(u-v)/(u+1)} m^{u/(u+1)} (\log m)^{v/(u+1)} \end{aligned} \tag{8.40}$$

as $m \to \infty$.

If $b(k) = 1$ for all k, $a(n)$ is p_n, the ordinary partition function. If $b(k) = k$ for all k, $a(n)$ is the number of plane partitions of n. Thus Brigham's theorem covers a wide class of interesting partition functions. The cost of this generality is that we obtain only the asymptotics of the logarithm of the summatory function of the partitions being enumerated. (For better estimates of the number of plane partitions, for example, see [9, 170, 387]. For ordinary partitions, we have the expansion (1.3).)

Brigham's proof of Theorem 8.2 first shows that

$$f(e^{-w}) \sim C w^{-u} (-\log w)^v \Gamma(u+1) \zeta(u+1) \quad \text{as} \quad w \to 0^+ \tag{8.41}$$

and then invokes the Hardy-Ramanujan Tauberian theorem [328]. Instead, one can obtain a proof from Theorem 8.1. The advantage of using Theorem 8.1 is that it is much easier to generalize. Hardy and Ramanujan proved their Tauberian theorem only for functions whose

growth rates are of the form given by (8.41). Their approach can be extended to other functions, but this is complicated to do. In contrast, Theorem 8.1 is easy to apply. The conditions of Theorem 8.1 on the derivatives are not restrictive. For a function $f(z)$ defined by (8.38) we have $B \to \infty$ if $\sum b(k) = \infty$, and the condition (8.33) can be shown to hold whenever there are constants c_1 and c_2 such that for all $w > 1$, and all sufficiently large m,

$$\sum_{k \le mw} b(k) \le c_1 w^{c_2} \sum_{k \le m} b(k) \ , \tag{8.42}$$

say. The main difficulty in applying Theorem 8.1 to generalizations of Brigham's theorem is in accurately estimating the minimal value in Lemma 8.1.

There are many other applications of Lemma 8.1 and Theorem 8.1. For example, they can be used to prove the results of [158] on volumes of spheres in the Lee metric.

Lemma 8.1 can be generalized in a straightforward way to multivariate generating functions. If

$$f(x,y) = \sum_{m,n \ge 0} a_{m,n} x^m y^n \tag{8.43}$$

and $a_{m,n} \ge 0$ for all m and n, then for any $x, y > 0$ for which the sum in (8.43) converges we have

$$a_{m,n} \le x^{-m} y^{-n} f(x,y) \ . \tag{8.44}$$

Generalizations of the lower bound of Theorem 8.1 to multivariate functions can also be derived, but are again harder than the upper bound [289].

8.2. Tauberian theorems

The Brigham Tauberian theorem for partitions [58], based on the Hardy-Ramanujan Tauberian theorem [328], was quoted already in Section 8.1. It applies to certain generating functions that have (in notation to be introduced in Section 10) a large singularity and gives estimates only for the logarithm of the summatory function of the coefficients. Another theorem that is often more precise, but is again designed to deal with rapidly growing partition functions, is that of Ingham [212], and will be discussed at the end of this section. Most of the Tauberian theorems in the literature apply to functions with small singularities (i.e., ones that do not grow rapidly as the argument approaches the circle of convergence) and give asymptotic relations for the sum of coefficients. References for Tauberian theorems are [117, 154, 190, 212, 325]. Their main advantage is generality and ease of use, as is shown

by the applications made to 0-1 laws in [77, 78, 79]. They can often be applied when the information about generating functions is insufficient to use the methods of Sections 11 and 12. This is especially true when the circle inside which the generating function converges is a natural boundary beyond which the function cannot be continued.

One Tauberian theorem that is often used in combinatorial enumeration is that of Hardy, Littlewood, and Karamata. We say a function $L(t)$ varies slowly at infinity if, for every $u > 0$, $L(ut) \sim L(t)$ as $t \to \infty$.

Theorem 8.3. *Suppose that $a_k \geq 0$ for all k, and that*

$$f(x) = \sum_{k=0}^{\infty} a_k x^k$$

converges for $0 \leq x < r$. If there is a $\rho \geq 0$ and a function $L(t)$ that varies slowly at infinity such that

$$f(x) \sim (r-x)^{-\rho} L\left(\frac{1}{r-x}\right) \quad as \quad x \to r- \quad , \tag{8.45}$$

then

$$\sum_{k=0}^{n} a_k r^k \sim (n/r)^{\rho} L(n)/\Gamma(\rho+1) \quad as \quad n \to \infty \; . \tag{8.46}$$

Example 8.4. *Cycles of permutations* ([33]). If S is a set of positive integers, and f_n the probability that a random permutation on n letters will have all cycle lengths in S (i.e., $f_n = a_n/n!$, where a_n is the number of permutations with cycle length in S), then

$$f(z) = \sum_{n=0}^{\infty} f_n z^n = \prod_{k \in S} \exp(z^k/k) = (1-z)^{-1} \prod_{k \notin S} \exp(-z^k/k) \; . \tag{8.47}$$

If $|\mathbb{Z}^+ \setminus S| < \infty$, then the methods of Sections 10.2 and 11 apply easily, and one finds that

$$f_n \sim \exp\left(-\sum_{k \notin S} 1/k\right) \quad \text{as} \quad n \to \infty \; . \tag{8.48}$$

This estimate can also be proved to apply for $|\mathbb{Z}^+ \setminus S| = \infty$, provided $|\{1, \ldots, m\} \setminus S|$ does not grow too rapidly when $m \to \infty$. If $|S| < \infty$ (or when $|\{1, \ldots, m\} \cap S|$ does not grow rapidly), the methods of Section 12 apply. When $S = \{1, 2\}$, one obtains, for example, the result of Moser and Wyman [292] that the number of permutations of order 2 is

$$\sim (n/e)^{n/2} 2^{-1/2} \exp(n^{1/2} - 1/4) \quad \text{as} \quad n \to \infty \; . \tag{8.49}$$

(For sharper and more general results, see [292, 376].) The methods used in these cases are different from the ones we are considering in this section.

We now consider an intermediate case, with

$$|\{1,\dots,m\}\cap S| \sim \rho m \quad \text{as} \quad m\to\infty\ . \tag{8.50}$$

for some fixed ρ, $0\le\rho\le 1$. This case can be handled by Tauberian techniques. To apply Theorem 8.3, we need to show that $L(t)=f(1-t^{-1})t^{-\rho}$ varies slowly at infinity. This is equivalent to showing that for any $u\in(0,1)$,

$$f(1-t^{-1}) \sim f(1-t^{-1}u)u^{\rho} \quad \text{as} \quad t\to\infty\ . \tag{8.51}$$

Because of (8.47), it suffices to prove that

$$\sum_{k\in S} k^{-1}\{(1-t^{-1})^k-(1-t^{-1}u)^k\} = \rho\log u + o(1) \quad \text{as} \quad t\to\infty\ , \tag{8.52}$$

but this is easy to deduce from (8.50) using summation by parts (Section 5). Therefore we find from Theorem 8.3 that

$$\sum_{n=0}^{m} f_n \sim f(1-1/n)\Gamma(\rho+1)^{-1} \quad \text{as} \quad n\to\infty\ . \tag{8.53}$$

(For additional results and references on this problem see [317].) ■

As the above example shows, Tauberian theorems yield estimates under weak assumptions. These theorems do have some disadvantages. Not only do they usually estimate only the summatory function of the coefficients, but they normally give no bounds for the error term. (See [154] for some Tauberian theorems with remainder terms.) Furthermore, they usually apply only to functions with nonnegative coefficients. Sometimes, as in the following theorem of Hardy and Littlewood, one can relax the nonnegativity condition slightly.

Theorem 8.4. *Suppose that $a_k\ge -c/k$ for some $c>0$,*

$$f(z)=\sum_{k=1}^{\infty} a_k x^k\ , \tag{8.54}$$

and that $f(x)$ converges for $0<x<1$, and that

$$\lim_{x\to 1-} f(x) = A\ . \tag{8.55}$$

Then

$$\lim_{n\to\infty}\sum_{k=1}^{n} a_k = A\ . \tag{8.56}$$

Some condition such as $a_k \geq -c/k$ on the a_k is necessary, or otherwise the theorem would not hold. For example, the function

$$f(x) = \frac{1-x}{1+x} = 1 - 2x + 2x^2 \cdots \tag{8.57}$$

satisfies (8.55) with $A = 0$, but (8.56) fails.

We next present an example that shows an application of the above results in combination with other asymptotic methods that were presented before.

Example 8.5. *Permutations with distinct cycle lengths.* The probability that a random permutation on n letters will have cycles of distinct lengths is $[z^n]f(z)$, where

$$f(z) = \prod_{k=1}^{\infty} \left(1 + \frac{z^k}{k}\right) \ . \tag{8.58}$$

Greene and Knuth [177] note that this is also the limit as $p \to \infty$ of the probability that a polynomial of degree n factors into irreducible polynomials of distinct degrees modulo a prime p. It is shown in [177] that

$$[z^n]f(z) = e^{-\gamma}(1 + n^{-1}) + O(n^{-2}\log n) \quad \text{as} \quad n \to \infty \ , \tag{8.59}$$

where $\gamma = 0.577\ldots$ is Euler's constant. A simplified version of the argument of [177] will be presented that shows that

$$[z^n]f(z) \sim e^{-\gamma} \quad \text{as} \quad n \to \infty \ . \tag{8.60}$$

Methods for obtaining better expansions, even more precise than that of (8.59), are discussed in Section 11.2. For related results obtained by probabilistic methods, see [20].

We have, for $|z| < 1$,

$$\begin{aligned} f(z) &= (1+z)\exp\left(\sum_{k=2}^{\infty} \log(1 + z^k/k)\right) \\ &= (1+z)\exp\left(\sum_{k=2}^{\infty} z^k/k + g(z)\right) \\ &= (1+z)(1-z)^{-1}\exp(g(z)) \ , \end{aligned} \tag{8.61}$$

where

$$g(z) = -z + \sum_{m=2}^{\infty} \frac{(-1)^{m-1}}{m} \sum_{k=2}^{\infty} \frac{z^{mk}}{k^m} \ . \tag{8.62}$$

Since the coefficients of $g(z)$ are small, the double sum in (8.62) converges for $z = 1$, and we have

$$\begin{aligned} g(1) &= \lim_{z\to 1-} g(z) = -1 + \sum_{k=2}^{\infty}\sum_{m=2}^{\infty} \frac{(-1)^{m-1}}{m} k^{-m} \\ &= -1 + \sum_{k=2}^{\infty} \{\log(1+k^{-1}) - k^{-1}\} \\ &= -\log 2 + \lim_{n\to\infty} (\log(n+1) - H_n) = -\log 2 - \gamma \ , \end{aligned} \tag{8.63}$$

where $H_n = 1 + 1/2 + 1/3 + \cdots + 1/n$ is the n-th *harmonic number*. Therefore, by (8.61), we find from Theorem 8.4 that if $f_n = [z^n]f(z)$, then

$$f_0 + f_1 + \cdots + f_n \sim ne^{-\gamma} \quad \text{as} \quad n \to \infty \ . \tag{8.64}$$

To obtain asymptotics of f_n, we note that if $h_n = [z^n]\exp(g(z))$, then by (8.61),

$$f_n = 2h_0 + 2h_1 + \cdots + 2h_{n-1} + h_n \ . \tag{8.65}$$

We next obtain an upper bound for $|h_n|$. There are several ways to proceed. The method used below gives the best possible result $|h_n| = O(n^{-2})$.

Since $g(z)$ has the power series expansion (8.62), and $h_n = [z^n]\exp(g(z))$, comparison of terms in the full expansion of $\exp(g(z))$ and $\exp(v(z))$ shows that $|h_n| \le [z^n]\exp(v(z))$, where $v(z)$ is any power series such that $|[z^n]g(z)| \le [z^n]v(z)$. For $n \ge 2$,

$$[z^n]g(z) = \sum_{\substack{m|n \\ m\ge 2 \\ m<n}} \frac{(-1)^{m-1}}{m} \left(\frac{m}{n}\right)^m \ . \tag{8.66}$$

The term $(m/n)^m$ is monotone decreasing for $1 \le m \le n/e$, since its derivative with respect to m is ≤ 0 in that range. Therefore

$$|[z^n]g(z)| \le \frac{1}{2}\left(\frac{2}{n}\right)^2 + \sum_{3\le m\le n/3} \frac{1}{m}\left(\frac{3}{n}\right)^3 + \frac{2}{n}2^{-n/2} \le 10n^{-2} \ , \tag{8.67}$$

say. Hence we can take

$$v(z) = 10\sum_{n=1}^{\infty} n^{-2}z^n \ , \tag{8.68}$$

and then we need to estimate

$$w_n = [z^n]\exp(v(z)) \ . \tag{8.69}$$

We let $w(z) = \exp(v(z))$, and note that

$$w'(z) = v'(z)w(z) \ , \tag{8.70}$$

so for $n \geq 1$,

$$nw_n = 10 \sum_{k=0}^{n-1} w_k (n-k)^{-1} \ . \tag{8.71}$$

Further, since $v(1) < \infty$, and $w_n \geq 0$ for all n, we have $w_n \leq A = w(1) = \exp(v(1))$ for all n. Let $B = 10^6 A$ and note that $w_n \leq Bn^{-2}$ for $1 \leq n \leq 10^3$. Suppose now that $w_m \leq Bm^{-2}$ for $1 \leq m < n$ for some $n \geq 10^3$. We will prove that $w_n \leq Bn^{-2}$, and then by induction this inequality will hold for all $n \geq 1$. We apply Eq. (8.70). For $0 \leq k \leq 100$, we use $w_k \leq A$, $(n-k)^{-1} \leq 2n^{-1}$. For $100 < k \leq n/2$,

$$w_k (n-k)^{-1} \leq Bk^{-2}(n-k)^{-1} \leq 2Bk^{-2}n^{-1} \ , \tag{8.72}$$

and so

$$\sum_{100 \leq k \leq n/2} w_k (n-k)^{-1} \leq B(40n)^{-1} \ . \tag{8.73}$$

Finally,

$$\sum_{n/2 < k \leq n-1} w_k (n-k)^{-1} \leq 4Bn^{-2} \sum_{n/2 < k \leq n-1} (n-k)^{-1} \leq 4Bn^{-2} H_n \ . \tag{8.74}$$

Therefore, by (8.71),

$$nw_n \leq 2000An^{-1} + B(4n)^{-1} + 4BH_n n^{-2} \leq Bn^{-1} \ , \tag{8.75}$$

which completes the induction step and proves that $w_n \leq Bn^{-2}$ for all $n \geq 1$. ■

There are Tauberian theorems that apply to generating functions with rapidly growing coefficients but are more precise than Brigham's theorem or the estimates obtainable with the methods of Section 8.1. One of the most useful is Ingham's Tauberian theorem for partitions [212]. The following result is a corollary of the more general Theorem 2 of [212].

Theorem 8.5. *Let $1 \leq u_1 < u_2 < \ldots$ be positive integers such that*

$$|\{u_j : u_j \leq x\}| = Bx^\beta + R(x) \ , \tag{8.76}$$

where $B > 0$, $\beta > 0$, and

$$\int_1^y x^{-1} R(x) dx = b \log y + c + o(1) \quad \textit{as} \quad y \to \infty \ . \tag{8.77}$$

Let

$$a(z) = \sum_{n=1}^{\infty} a_n z^n = \prod_{j=1}^{\infty} (1 - z^{u_j})^{-1} , \tag{8.78}$$

$$a^*(z) = \sum_{n=1}^{\infty} a_n^* z^n = \prod_{j=1}^{\infty} (1 + z^{u_j}) . \tag{8.79}$$

Then, as $m \to \infty$,

$$\sum_{n=1}^{m} a_n \sim (2\pi)^{-1/2} (1-\alpha)^{1/2} e^c V^{-\alpha(b+1/2)} m^{(b+1/2)(1-\alpha)-1/2} \exp(\alpha^{-1}(Vm)^{\alpha}) , \tag{8.80}$$

$$\sum_{n=1}^{m} a_n^* \sim (2\pi)^{-1/2} (1-\alpha)^{1/2} 2^b (V^* m)^{-\alpha/2} \exp(\alpha^{-1}(V^* m)^{\alpha}) , \tag{8.81}$$

where

$$\alpha = \beta(\beta+1)^{-1}, \quad V = \{B\beta\Gamma(\beta+1)\zeta(\beta+1)\}^{1/\beta} , \quad V^* = (1-2^{-\beta})^{1/\beta} V . \tag{8.82}$$

If $u_1 = 1$, *then as* $n \to \infty$

$$a_n \sim (2\pi)^{-1/2} (1-\alpha)^{1/2} e^c V^{-\alpha(b-1/2)} n^{(b-1/2)(1-\alpha)-1/2} \exp(\alpha^{-1}(Vn)^{\alpha}) , \tag{8.83}$$

and if $1, 2, 4, 8, \ldots$ *all belong to* $\{u_j\}$, *then*

$$a_n^* \sim (2\pi)^{-1/2} (1-\alpha)^{1/2} 2^b (V^*)^{\alpha/2} n^{\alpha/2-1} \exp(\alpha^{-1}(V^* n)^{\alpha}) . \tag{8.84}$$

Theorem 8.5 provides more precise information than Brigham's Theorem 8.2, but under more restrictive conditions. It is derived from Ingham's main result, Theorem 1 of [212], which can be applied to wider classes of functions. However, that theorem cannot be used to derive Theorem 8.2. The disadvantage of Ingham's main theorem is that it requires knowledge of the behavior of the generating function in the complex plane, not just on the real axis. On the other hand, the region where this behavior has to be known is much smaller than it is for the analytic methods that give more accurate answers, and which are presented in Sections 10–12. Only behavior of the generating functions $\Pi(1 - z^{\lambda_j})^{-1}$ or $\Pi(1 + z^{\lambda_j})$ in an angle $|\mathrm{Arg}|(1 - z)| \le \pi/2 - \delta$ for some $\delta > 0$ needs to be controlled.

Ingham's paper [212] contains an extended discussion of the relations between different Tauberian theorems and of the necessity for various conditions.

9. Recurrences

This section presents some basic methods for handling recurrences. The title is slightly misleading, since almost all of this chapter is devoted to methods that are useful in this area. Almost all asymptotic estimation problems concern quantities that are defined through implicit or explicit recurrences. Furthermore, the most common and most effective method of solving recurrences is often to determine its generating function and then apply the methods presented in the other sections. However, there are many recurrences, and those discussed in Sections 9.4 and 9.5 require special methods that do not fit into other sections. These methods deserve to be included, so it seems preferable to explain them after treating some of the more common types of recurrences, even though those could have been covered elsewhere in this chapter.

Since generating functions are the most powerful tool for handling combinatorial recurrences, all the books listed in Section 18 that help in dealing with combinatorial identities and generating functions are also useful in handling recurrences. Methods for recurrences that are not amenable to generating function methods are presented in [175, 177]. Lueker [264] is an introductory survey to some recurrence methods.

Wimp's book [382] is concerned primarily with numerical stability problems in computing with recurrences. Such problems are important in computing values of orthogonal polynomials, for example, but seldom arise in combinatorial enumeration. However, there are sections of [382] that are relevant to our topic, for example to the discussion of differential equations in Section 9.2.

9.1. Linear recurrences with constant coefficients

The most famous sequence that satisfies a linear recurrence with constant coefficients is that of the Fibonacci numbers, defined by $F_0 = F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$. There are many others that are only slightly less well known. Fortunately, the theory of such sequences is well developed, and from the standpoint of asymptotic enumeration their behavior is well understood. (For a survey of number theoretic results, together with a list of many unsolved problems about such sequences that arise in that area, see [73].) There are even several different approaches to solving linear recurrences with constant coefficients. The one we emphasize here is that of generating functions, since it fits in best with the rest of this chapter. For other approaches, see [287, 298], for example.

Suppose that we have a linear recurrence or a system of recurrences and have found that

the generating function $f(z)$ we are interested in has the form

$$f(z) = \frac{G(z)}{h(z)} \ , \tag{9.1}$$

where $G(z)$ and $h(z)$ are polynomials. The basic tool for obtaining asymptotic information about $[z^n]f(z)$ is the partial fraction expansion of a rational function [205]. Dividing $G(z)$ by $h(z)$ we obtain

$$f(z) = p(z) + \frac{g(z)}{h(z)} \ , \tag{9.2}$$

where $p(z)$, $g(z)$, and $h(z)$ are all polynomials in z and $\deg g(z) < \deg h(z)$. We can assume that $h(0) \neq 0$, since if that were not the case, we would have $g(0) = 0$ (as in the opposite case $f(z)$ would not be a power series in z, but would have terms such as z^{-1} or z^{-2}) and we could cancel a common factor of z from $g(z)$ and $h(z)$. Therefore, if $d = \deg h(z)$, we can write

$$h(z) = h(0) \prod_{j=1}^{d'} \left(1 - \frac{z}{z_j}\right)^{m_j} \ , \tag{9.3}$$

where z_j, $1 \leq j \leq d'$ are the distinct roots of $h(z) = 0$, z_j has multiplicity $m_j \geq 1$, and $\sum m_j = d$. Hence we find [175, 205] that for certain constants $c_{j,k}$,

$$\begin{aligned} f(z) &= p(z) + \sum_{j=1}^{d'} \sum_{k=1}^{m_j} \frac{c_{j,k}}{(1 - z/z_j)^k} \\ &= p(z) + \sum_{j=1}^{d'} \sum_{k=1}^{m_j} c_{j,k} \sum_{h=0}^{\infty} \binom{h+k-1}{k-1} z^h z_j^{-h} \ . \end{aligned} \tag{9.4}$$

Thus

$$a_n = [z^n]p(z) + \sum_{j=1}^{d'} \sum_{k=1}^{m_j} c_{j,k} \binom{h+k-1}{k-1} z_j^{-n} \ . \tag{9.5}$$

When $m_j = 1$,

$$c_{j,1} = \frac{-g(z_j)}{z_j h'(z_j)} \ , \tag{9.6}$$

and explicit formulas for the $c_{j,k}$ when $m_j > 1$ can also be derived [175], but are unwieldy and seldom used.

Example 9.1. *Fibonacci numbers.* As was noted in Example 6.3,

$$F(z) = \sum_{n=0}^{\infty} F_n z^n = \frac{z}{1 - z - z^2} \ .$$

Now

$$h(z) = 1 - z - z^2 = (1 + \phi^{-1} z)(1 - \phi z) \ , \tag{9.7}$$

where $\phi = (1 + 5^{1/2})/2$ is the golden ratio. Therefore

$$F(z) = \frac{1}{\sqrt{5}} \left(\frac{1}{1 - \phi z} - \frac{1}{1 + \phi^{-1} z} \right) \tag{9.8}$$

and for $n \geq 0$,

$$F_n = [z^n] F(z) = 5^{-1/2} (\phi^n - (-\phi)^{-n}) \ . \tag{9.9}$$

■

The partial fraction expansion (9.4) shows that the first-order asymptotics of sequence a_n satisfying a linear recurrence of the form (6.30) are determined by the smallest zeros of the characteristic polynomial $h(z)$. The full asymptotic expansion is given by (9.5), and involves all the zeros. In practice, using (9.5) presents some difficulties, in that multiplicities of zeros are not always easy to determine, and the coefficients $c_{j,k}$ are often even harder to deal with. Eventually, for large n, their influence becomes negligible, but when uniform estimates are required they present a problem. In such cases the following theorem is often useful.

Theorem 9.1. *Suppose that $f(z) = g(z)/h(z)$, where $g(z)$ and $h(z)$ are polynomials, $h(0) \neq 0$, $\deg g(z) < \deg h(z)$, and that the only zeros of $h(z)$ in $|z| < R$ are $\rho_1, \ldots, \rho_k$, each of multiplicity 1. Suppose further that*

$$\max_{|z|=R} |f(z)| \leq W \ , \tag{9.10}$$

and that $R - |\rho_j| \geq \delta$ for some $\delta > 0$ and $1 \leq j \leq k$. Then

$$\left| [z^n] f(z) + \sum_{j=1}^{k} \frac{g(\rho_j)}{h'(\rho_j)} \rho_j^{-n-1} \right| \leq W R^{-n} + \delta^{-1} R^{-n} \sum_{j=1}^{k} |g(\rho_j)/h'(\rho_j)| \ . \tag{9.11}$$

Theorem 9.1 is derived using methods of complex variables, and a proof is sketched in Section 10. That section also discusses how to locate all the zeros $\rho_1, \ldots, \rho_k$ of a polynomial $h(z)$ in a disk $|z| < R$. In general, the zero location problem is not a serious one in enumeration problems. Usually there is a single positive real zero that is closer to the origin than any other, it can be located accurately by simple methods, and R is chosen so that $|z| < R$ encloses only that zero.

Example 9.2. *Sequences with forbidden subblocks.* We continue with the problem presented in Examples 6.4 and 6.8. Both $F_A(z)$ and $G_A(z)$ have as denominators

$$h(z) = z^k + (1 - 2z) C_A(z) \ , \tag{9.12}$$

which is a polynomial of degree exactly k. Later, in Example 10.6, we will show that for $k \geq 9$, $h(z)$ has exactly one zero ρ in $|z| \leq 0.6$, and that for $|z| = 0.55$, $|h(z)| \geq 1/100$. Furthermore, by Example 6.7, $\rho \to 1/2$ as $k \to \infty$. On $|z| = 0.55$,

$$|F_A(z)| \leq 100 \cdot (0.55)^k \ . \tag{9.13}$$

Theorem 9.1 then shows, for example, that for $n > k \geq k_0$,

$$\begin{aligned} \left| [z^n] F_A(z) + \frac{C_A(\rho)\rho^{-n-1}}{h'(\rho)} \right| &\leq 100(0.55)^{k-n} + 40(0.55)^{-n}|h'(\rho)|^{-1} \\ &\leq 50(0.55)^{-n} \ , \end{aligned} \tag{9.14}$$

since by Example 6.7, as $k \to \infty$,

$$h'(\rho) = k\rho^{k-1} - 2C_A(\rho) + (1-2\rho)C_A'(\rho) \sim -2C_A(\rho) \sim -\rho^{-1} \ . \tag{9.15}$$

The estimate (9.14), when combined with the expansions of Example 6.7, gives accurate approximations for p_n, the probability that A does not appear as a block among the first n coin tosses. We have

$$\begin{aligned} p_n &= 2^{-n}[z^n]F_z(z) \\ &= -2^{-n}C_A(\rho)\rho^{-n-1}(h'(\rho))^{-1} + O(\exp(-0.09n)) \ . \end{aligned} \tag{9.16}$$

We now estimate $h'(\rho)$ as before, in (9.15), but more carefully, putting in the approximation for ρ from Example 6.7. We find that

$$h'(\rho) = -\rho^{-1} + O(k2^{-k}) \ , \tag{9.17}$$

and

$$\rho^{-n} = 2^n \exp(-n(2^k C_A(1/2))^{-1} + O(nk2^{-2k})) \ . \tag{9.18}$$

Therefore

$$p_n = \exp(-n(2^k C_A(1/2))^{-1} + O(nk2^{-2k})) + O(\exp(-n/12)) \ . \tag{9.19}$$

This shows that p_n has a sharp transition. It is close to 1 for $n = o(2^k)$, and then, as n increases through 2^k, drops rapidly to 0. (The behavior on the two sides of 2^k is not symmetric, as the drop towards 0 beyond 2^k is much faster than the increases towards 1 on the other side.) For further results and applications of such estimates, see [180, 181]. Estimates such as (9.19) yield results sharper than those of Example 6.8. They also prove (see

Example 14.1) that the expected lengths of the longest run of 0's in a random sequence of length n is $\log_2 n + u(\log_2 n) + o(1)$ as $n \to \infty$, where $u(x)$ is a continuous function that is not constant and satisfies $u(x+1) = u(x)$. (See also the discussion of carry propagation in [236].) For other methods and results in this area, see [18]. ■

Inhomogeneous recurrences with constant coefficients, say,

$$a_n = \sum_{i=1}^{d} c_i a_{n-i} + b_n, \quad n \geq d \ , \tag{9.20}$$

are not covered by the techniques discussed above. One can still use the basic generating function approach to derive the ordinary generating function of a_n, but this time it is in terms of the ordinary generating function of b_n. If b_n does not grow too rapidly, the "subtraction of singularities" method of Section 10.2 can be used to derive the asymptotics of a_n in a form similar to that given by (9.26).

9.2. Linear recurrences with varying coefficients

Linear recurrences with constant coefficients have a nice and complete theory. That is no longer the case when one allows coefficients that vary with the index. This is not a fault of mathematicians in not working hard enough to derive elegant results, but reflects the much more complicated behavior that can occur. The simplest case is when the recurrence has a finite number of terms, and the coefficients are polynomials in n.

Example 9.3. *Two-sided generalized Fibonacci sequences.* Let t_n be the number of integer sequences $(b_j, \ldots, b_2, b_1, 1, 1, a_1, a_2, \ldots, a_k)$ with $j + k + 2 = n$ in which each b_i is the sum of one or more contiguous terms immediately to its right, and each a_i is likewise the sum of one or more contiguous terms immediately to its left. It was shown in [120] that $t_1 = t_2 = 1$ and that

$$t_{n+1} = 2nt_n - (n-1)^2 t_{n-1} \quad \text{for} \quad n \geq 2 \ . \tag{9.21}$$

If we let

$$t(z) = \sum_{n=1}^{\infty} \frac{t_n z^{n-1}}{(n-1)!} \tag{9.22}$$

be a modified exponential generating function, then the recurrence (9.21) shows that

$$t'(z)(1-z)^2 - t(z)(2-z) = 1 \ . \tag{9.23}$$

Standard methods for solving ordinary differential equations, together with the initial conditions $t_1 = t_2 = 1$, then yield the explicit solution

$$t(z) = (1-z)^{-1} \exp((1-z)^{-1}) \left[C + \int_z^1 (1-w)^{-1} \exp(-(1-w)^{-1}) dw \right] , \tag{9.24}$$

where

$$C = e^{-1} - \int_0^1 (1-w)^{-1} \exp(-(1-w)^{-1}) dw = 0.148495\ldots . \tag{9.25}$$

Once the explicit formula (9.24) for $t(z)$ is obtained, the methods of Section 12 give the estimate

$$t_n \sim C(n-1)!(e/\pi)^{1/2} \exp(2n^{1/2})(2n^{1/4})^{-1} \quad \text{as} \quad n \to \infty . \tag{9.26}$$

It is easy to show that the absolute value of

$$(1-z)^{-1} \exp((1-z)^{-1}) \int_z^1 (1-w)^{-1} \exp(-(1-w)^{-1}) dw \tag{9.27}$$

is small for $|z| < 1$. Therefore the asymptotics of the t_n are determined by the behavior of coefficients of

$$C(1-z)^{-1} \exp((1-z)^{-1}) , \tag{9.28}$$

and that can be obtained easily. The estimate (9.26) then follows. ■

To see just how different the behavior of linear recurrences with polynomial coefficients can be from those with constant coefficients, compare the behavior of the sequences in Example 9.3 above and Example 9.4 (given below). The existence of such differences should not be too surprising, since after all even the first order recurrence $a_n = na_{n-1}$ for $n \geq 2$, $a_1 = 1$, has the obvious solution $a_n = n!$, which is not at all like the solutions to constant coefficient recurrences. However, when $a_n = na_{n-1}$, a simple change of variables, namely $a_n = b_n n!$, transforms this recurrence into the trivial one of $b_n = b_{n-1} = \cdots = b_1 = 1$ for all n. Such rescaling is among the most fruitful methods for dealing with nonlinear recurrences, even though it is seldom as simple as for $a_n = n!$.

Example 9.3 is typical in that a sequence satisfying a linear recurrence of the form

$$a_n = \sum_{j=1}^{r} c_j(n) a_{n-j} , \quad n \geq r , \tag{9.29}$$

where r is fixed and the $c_j(n)$ are rational functions (a P-recursive sequence in the notation of Section 6.3) can always be transformed into a differential equation for a generating function. Whether anything can be done with that generating function depends strongly on the

recurrence and the form of the generating function. Example 9.3 is atypical in that there is an explicit solution to the differential equation. Further, this explicit solution is a nice analytic function. This is due to the special choice of the form of the generating function. An exponential generating function seems natural to use in that example, since the recurrence (9.21) shows immediately that $t_n \le (2n-2)(2n-4)\dots 2 = 2^{n-1}(n-1)!$, and a slightly more involved induction proves that t_n grows at least as fast as a factorial. If we tried to use an ordinary generating function

$$u(z) = \sum_{n=1}^{\infty} t_n z^n \ , \tag{9.30}$$

then the recurrence (9.21) would yield the differential equation

$$z^4 u''(z) + z^3 u'(z) + (1 - 2z^2)u(z) = z - z^2 \ , \tag{9.31}$$

which is not as tractable. (This was to be expected, since $u(z)$ is only a formal power series.) Even when a good choice of generating function does yield an analytic function, the differential equation that results may be hard to solve. (One can always find a generating function that is analytic, but the structure of the problem may not be reflected in the resulting differential equation, and there may not be anything nice about it.)

There is an extensive literature on analytic solutions of differential equations (cf. [205, 206, 207, 272, 368, 372]), but it is not easy to apply in general. Singularities of analytic functions that satisfy linear differential equations with analytic coefficients are usually of only a few basic forms, and so the methods of Sections 11 and 12 suffice to determine the asymptotic behavior of the coefficients. The difficulty is in locating the singularities and determining their nature. We refer to [206, 207, 272, 368, 372] for methods for dealing with this difficulty, since they are involved and so far have been seldom used in combinatorial enumeration. There will be some further discussion of differential equations in Section 15.3.

Some aspects of the theory of linear recurrences with constant coefficients do carry over to the case of varying coefficients, even when the coefficients are not rational functions. For example, there will in general be r linearly independent solutions to the recurrence (9.29) (corresponding to the different starting conditions). Also, if a solution a_n has the property that a_{n+1}/a_n tends to a limit α as $n \to \infty$, then $1/\alpha$ is a limit of zeros of

$$1 - \sum_{j=1}^{r} c_j(n) z^j \ , \tag{9.32}$$

and therefore is often a root of

$$1 - \sum_{j=1}^{r} \left(\lim_{n\to\infty} c_j(n) \right) z^j \ . \tag{9.33}$$

Whether there are exactly r linearly independent solutions is a difficult problem. Extensive research was done on this topic 1920's and 1930's [2, 29], culminating in the work of Birkhoff and Trjitzinsky [51, 52, 53, 366, 367]. This work applies to recurrences of the form (9.29) where the $c_j(n)$ have Poincaré asymptotic expansions

$$c_j(n) \sim n^{k_j/k}\{c_{j,0} + c_{j,1}n^{-1/k} + c_{j,2}n^{-2/k} + \cdots\} \quad \text{as} \quad n \to \infty \ , \tag{9.34}$$

where the k_j and k are integers and $c_{j,0} \neq 0$ if $c_j(n)$ is not identically 0 for all n. It follows from this work that solutions to the recurrence are expressible as linear combinations of elements of the form

$$(n!)^{p/q} \exp(P(n^{1/m}))n^{\alpha}(\log n)^h \ , \tag{9.35}$$

where h, m, p, and q are integers, $P(z)$ a polynomial, and α a complex number. An exposition of this theory and how it applies to enumeration has been given by Wimp and Zeilberger [384]. (There is a slight complication in that most of the literature cited above is concerned with recurrences for functions of a real argument, not sequences, but this is not a major difficulty.) There is still a problem in identifying which linear combination provides the derived solution. Wimp and Zeilberger point out that it is usually easy to show that the largest of the terms of the form (9.35) does show up with a nonzero coefficient, and so determines the asymptotics of a_n up to a multiplicative constant. However, the Birkhoff-Trjitzinsky method does not in general provide any techniques for determining that constant.

The major objection to the use of the Birkhoff-Trjitzinsky results is that they may not be rigorous, since gaps are alleged to exist in the complicated proofs [211, 383]. Furthermore, in almost all combinatorial enumeration applications the coefficients are rational, and so one can use the theory of analytic differential equations.

When there is no way to avoid linear recurrences with coefficients that vary but are not rational, one can sometimes use the work of Kooman [243, 244], which develops the theory of second order linear recurrences with almost-constant coefficients.

Example 9.4. *An oscillating sequence.* Let

$$a_n = \sum_{k=0}^{n} \binom{n}{k} \frac{(-1)^k}{k!} \ , \qquad n = 0, 1, \ldots \ . \tag{9.36}$$

Then a_n satisfies the linear recurrence

$$a_{n+2} - \left(2 - \frac{2}{n}\right) a_{n+1} + \left(1 - \frac{1}{n}\right) a_n = 0, \quad n \geq 0 \ . \tag{9.37}$$

The methods of [244] can be used to show that for some constants c and ϕ

$$a_n = cn^{-1/4} \sin(2n^{1/2} + \phi) + o(n^{-1/4}) \quad \text{as} \quad n \to \infty \ , \tag{9.38}$$

which is a much more precise estimate than the crude one mentioned in Example 10.1.

Another, in some ways preferable method for obtaining asymptotic expansions for a_n is mentioned in Example 12.8. It is based on an explicit form for the generating function of a_n, $f(z) = \sum a_n z^n$. An interchange of orders of summation (easily justified for $|z|$ small, say $|z| < 1/2$) shows that

$$\begin{aligned} f(z) &= \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \sum_{n=k}^{\infty} \binom{n}{k} z^n \\ &= \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \frac{z^k}{(1-z)^{k+1}} = \frac{1}{1-z} \exp\left(-\frac{z}{1-z}\right) \ . \end{aligned} \tag{9.39}$$

The saddle point method can then be applied to obtain asymptotic expansions for a_n. ■

9.3. Linear recurrences in several variables

Linear recurrences in several variables that have constant coefficients can be attacked by methods similar to those used in a single variable. If we have

$$a_{m,n} = \sum_{\substack{i=0 \\ i+j>0}}^{d} \sum_{i=0}^{d} c_{i,j} a_{m-i,n-j} \tag{9.40}$$

for $m, n \geq d$, say, then the generating function

$$f(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} a_{m,n} x^m y^n \tag{9.41}$$

satisfies the relation

$$f(x,y)\left(1-\sum_{\substack{i=0\\ i+j>0}}^{d}\sum_{i=0}^{d}c_{i,j}x^iy^j\right) = \sum_{\substack{m=0\\ m>d \text{ or } n>d}}^{\infty}\sum_{n=0}^{\infty}a_{m,n}x^my^n - \sum_{\substack{i=0\\ i+j>0}}^{d}\sum_{i=0}^{d}c_{ij}x^iy^j\sum_{\substack{m,n\\ m\le d-i\\ \text{or } n\le d-i}}a_{m,n}x^my^n\ . \tag{9.42}$$

If $a_{m,n}=0$ for $0\le m<d$ and $n\ge d$ as well as for $0\le n<d$ and $m\ge d$ (so that all the $a_{m,n}$ are fully determined by $a_{m,n}$ for $0\le m<d$, $0\le n<d$), then $f(x,y)$ is a rational function. If this condition does not hold, $f(x,y)$ can be complicated.

The paragraph above shows that under common conditions, constant coefficient recurrences lead to generating functions that are rational even in several variables. However, even when the rational function is determined, there is no equivalent of partial fraction decomposition to yield elegant asymptotics of the coefficients. Coefficients of multivariate generating functions are much harder to handle than those of univariate functions. There are tools (discussed in Section 13), that are usually adequate to handle rational generating functions, but they are not simple.

When the coefficients of the multivariate recurrences vary, available knowledge is extremely limited. Even if the coefficients are polynomials, we obtain a partial differential equation for the generating function. Sometimes there are tricks that lead to a simple solution (cf. Example 15.6), but this is not common.

9.4. Nonlinear recurrences

Nonlinear recurrences come in a great variety of shapes, and the methods that are used to solve them are diverse, depending on the nature of the problem. This section presents a sample of the most useful techniques that have been developed.

Sometimes a nonlinear recurrence has a simple solution because of a nice algebraic factorization. For example, suppose that z_0 is any given complex number, and

$$z_{n+1}=z_n^2-2 \quad \text{for} \quad n\ge 0\ . \tag{9.43}$$

If we set

$$w=(z_0+(z_0^2-4)^{1/2})/2\ , \tag{9.44}$$

we have $z_0 = w + w^{-1}$, and more generally

$$z_n = w^{2^n} + w^{-2^n} \quad \text{for} \quad n \geq 0 \; . \tag{9.45}$$

Eq. (9.45) is easily established through induction. However, this is an exceptional instance, and already recurrences of the type $z_{n+1} = z_n^2 + c$ for c a complex constant lead to deep questions about the Mandelbrot set and chaotic behavior [91].

Since linear recurrences are well understood, the best that one can hope for when confronted with a nonlinear recurrence is that it might be reducible to a linear one. This works in many situations.

Example 9.5. *Planted plane trees.* Let $a_{n,h}$ be the number of planted plane trees with n nodes and height $\leq h$ [64, 177], and let

$$A_h(z) = \sum_{n=0}^{\infty} a_{n,h} z^n \; . \tag{9.46}$$

Since a tree of height $\leq h+1$ has a root and any number of subtrees, each of height $\leq h$,

$$\begin{aligned} A_{h+1}(z) &= z(1 + A_h(z) + A_h(z)^2 + \cdots) \\ &= z(1 - A_h(z))^{-1} \; . \end{aligned} \tag{9.47}$$

Iterating this recurrence, we obtain a finite continued fraction that looks like

$$A_{h+1}(z) = \cfrac{z}{1 - \cfrac{z}{1 - \cfrac{z}{1 - \cdots}}} \; . \tag{9.48}$$

The general theory of continued functions represents a convergent as a quotient of two sequences satisfying recurrences involving the partial quotients. (For references, see [218, 319].) After playing with this idea, one finds that the substitution

$$A_h(z) = \frac{z P_h(z)}{P_{h+1}(z)} \tag{9.49}$$

gives

$$P_{h+1}(z) = P_h(z) - z P_{h-1}(z) \; , \qquad h \geq 2 \; ,$$

where $P_0(z) = 0$, $P_1(z) = 1$. This is a linear recurrence when we regard z as fixed, and so the theory presented before leads to the explicit representation

$$P_h(z) = (1 - 4z)^{-1/2} \left\{ \left(\frac{1 + (1-4z)^{1/2}}{2} \right)^h - \left(\frac{1 - (1-4z)^{1/2}}{2} \right)^h \right\} \; . \tag{9.50}$$

De Bruijn, Knuth, and Rice [64] use this representation to determine the average height of plane trees. ■

Greene and Knuth (p. 30 of [177]) note that the continued fraction method of replacing a convergent by a quotient of elements of two sequences in general leads not to a single sequence of polynomials like the $P_h(z)$ of Example 9.5, but to two sequences. This is only slightly harder to handle, and allows one to linearize more complicated recurrences.

There are many additional ways to linearize a recurrence. (A small list is given on p. 31 of [177].) For example, a purely multiplicative relation $a_n = a_{n-1}^2/a_{n-2}$ is transformed into the linear $\log a_n = 2\log a_{n-1} - \log a_{n-2}$ by taking logarithms. One of the most fruitful tricks of this type is taking inverses. Thus $a_n = a_{n-1}/(1 + a_{n-1})$ is equivalent to

$$\frac{1}{a_n} = \frac{1}{a_{n-1}} + 1 \ , \tag{9.51}$$

which has the obvious solution $a_n^{-1} = a_0^{-1} + n$. (This assumes $a_0 \neq -1/k$ for any $k \in \mathbb{Z}^+$.)

Linearization works well, but is limited in applicability. More widely applicable, but producing answers that are not as clear, is approximate linearization, where a given nonlinear recurrence is close to a linear one. The following example combines approximate linearization with bootstrapping.

Example 9.6. *A quadratic recurrence.* The study of the average height of binary trees in [132] involves the recurrence

$$a_n = a_{n-1}(1 - a_{n-1}) \quad \text{for} \quad n \geq 1 \ , \tag{9.52}$$

with $a_0 = 1/2$. The a_n are monotone decreasing, so we try the inverse trick. We find

$$\frac{1}{a_n} = \frac{1}{a_{n-1}(1 - a_{n-1})} = \frac{1}{a_{n-1}} + 1 + \frac{a_{n-1}}{1 - a_{n-1}} \ . \tag{9.53}$$

Iterating this recurrence (but applying it only to the first term on the right-hand side of Eq. (9.53)) we obtain

$$\begin{aligned} \frac{1}{a_n} &= \frac{1}{a_{n-2}} + 2 + \frac{a_{n-2}}{1 - a_{n-2}} + \frac{a_{n-1}}{1 - a_{n-1}} \\ &= \cdots \\ &= \frac{1}{a_0} + n + \sum_{j=0}^{n-1} \frac{a_j}{1 - a_j} \\ &= n + 2 + \sum_{j=0}^{n-1} \frac{a_j}{1 - a_j} \ . \end{aligned} \tag{9.54}$$

Equation (9.54) shows that $a_n^{-1} > n$, so $a_n < 1/n$. Applying this bound to a_j for $2 \le j \le n-1$ in the sum on the right-hand side of Eq. (9.54), we find that

$$n \le a_n^{-1} \le n + O(\log n) \ . \tag{9.55}$$

When we substitute this into (9.54), we find that $a_n^{-1} = n + \log n + o(\log n)$, and further iterations produce even more accurate estimates. ■

Approximate linearization also works well for some rapidly growing sequences.

Example 9.7. *Doubly exponential sequences.* Many recurrences are of the form

$$a_{n+1} = a_n^2 + b_n \ , \tag{9.56}$$

where b_n is much smaller than a_n^2 (and may even depend on the a_n for $k \le n$, as in $b_n = a_n$ or $b_n = a_{n-1}$). Aho and Sloane [3] found that surprisingly simple solutions to such recurrences can often be found. The basic idea is to reduce to approximate linearization by taking logarithms. We find that if a_0 is the given initial value, and $a_n > 0$ for all n, then the transformation

$$u_n = \log a_n \ , \tag{9.57}$$

$$\delta_n = \log(1 + b_n a_n^{-2}) \ , \tag{9.58}$$

reduces (9.56) to

$$u_{n+1} = 2u_n + \delta_n \ , \quad n \ge 0 \ . \tag{9.59}$$

Therefore

$$\begin{aligned} u_n &= \delta_{n-1} + 2u_{n-1} = \delta_{n-1} + 2\delta_{n-2} + 4u_{n-2} \\ &= \dots \\ &= \sum_{j=1}^{n} 2^{j-1}\delta_{n-j} + 2^n u_0 \\ &= 2^n(u_0 + \delta_0/2 + \delta_1/4 + \cdots + \delta_{n-1}/2^n) \ . \end{aligned} \tag{9.60}$$

If we assume that the δ_k are small, then

$$\alpha = u_0 + \sum_{k=0}^{\infty} \delta_k 2^{-k-1} \tag{9.61}$$

exists, and

$$r_n = u_n - 2^n\alpha = 2^n \sum_{k=n}^{\infty} \delta_k 2^{-k-1} \ . \tag{9.62}$$

If the δ_k are sufficiently small, the difference r_n in (9.62) will be small, and

$$a_n = \exp(u_n) = \exp(2^n\alpha - r_n) \ . \tag{9.63}$$

The expression (9.63) might not seem satisfactory, since both a_n and r_n are expressed in terms of all the a_k, for $k < n$ and for $k \geq n$. The point of (9.63) is that for many recurrences, r_n is negligibly small, while α is given by the rapidly convergent series (9.61), so that only the first few a_n are needed to obtain a good estimate for the asymptotic behavior of a_n. We next discuss a particularly elegant case.

Suppose that $a_n \geq 1$ and $|b_n| < a_n/4$ for all $n \geq 0$. Then $a_{n+1} \geq a_n$ and $|\delta_{n+1}| \leq |\delta_n|$ for $n \geq 0$, and so $|r_n| \leq |\delta_n|$. Hence

$$a_n \exp(-|\delta_n|) \leq \exp(2^n\alpha) \leq a_n \exp(|\delta_n|) \tag{9.64}$$

and since

$$\begin{aligned} \exp(|\delta_n|) &\leq 1 + |b_n|a_n^{-2} < 1 + (4a_n)^{-1} \ , \\ \exp(-|\delta_n|) &\geq (1 + (4a_n)^{-1})^{-1} \geq 1 - (3a_n)^{-1} \ , \end{aligned} \tag{9.65}$$

we find that

$$|a_n - \exp(2^n\alpha)| < (2a_n)^{-1} \leq 1/2 \ . \tag{9.66}$$

If a_n is an integer, then we can assert that it is the closest integer to $\exp(2^n\alpha)$.

The restriction $|b_n| < a_n/4$ is severe. The basic method applies even without it, and the expansion (9.63) is valid, for example, if we only require that $|\delta_{n+1}| \leq |\delta_n|$ for $n \geq n_0$. However, we will not in general obtain results as nice as (9.66) if we only impose these weak conditions.

The method outlined above can be applied to recurrences that appear to be of a slightly different form. Sometimes only a trivial transformation is required. For example, Golomb's nonlinear recurrence,

$$a_{n+1} = a_0 a_1 \cdots a_n + b, \quad a_0 = 1 \ , \tag{9.67}$$

for b a constant, is easily seen to be equivalent to

$$a_{n+1} = (a_n - b)a_n + b, \quad a_0 = 1, \quad a_1 = b + 1 \ . \tag{9.68}$$

The substitution

$$x_n = a_n - b/2 \tag{9.69}$$

transforms (9.68) into

$$x_{n+1} = x_n^2 + (2 - b)b/4 \ , \tag{9.70}$$

which is of the form treated above. (If the x_n are integers, the inequality (9.66) with x_n replacing a_n might not apply to the x_n because the condition $|(2-b)b/4| < |x_k|/4$ might fail for some k. The trick to use here is to start the recurrence with some x_k, say x_{k_0}, so that the condition $|(2-b)b/4| < |x_k|/4$ applies for $k \geq k_0$. The new α for which (9.66) holds will then be defined in terms of $x_{k_0}, x_{k_0+1}, \ldots$.)

In some situations the results presented above cannot be applied, but the basic method can still be extended. That is the case for the recurrence

$$a_{n+1} = a_n a_{n-1} + 1, \quad a_0,\ a_1 \geq 1 \tag{9.71}$$

of [3]. The result is that a_n is the nearest integer to

$$\alpha^{F_n} \beta^{F_{n-1}} \ , \tag{9.72}$$

where α and β are positive constants, and the F_k are the Fibonacci numbers. What matters is that the recurrence leads to doubly exponential (and regular) growth of a_n. Example 15.3 shows how this principle can be applied even when the a_n are not numbers, but polynomials whose coefficients need to be estimated. ■

9.5. Quasi-linear recurrences

This section mentions some methods and results for studying recurrences that have linearity properties, but are not linear. The most important of them are recurrences involving minimization or maximization. They arise frequently in problems that use dynamic programming approaches and in divide and conquer methods. An important example, treated in [147], is that of a sequence f_n, given by $f_0 = 1$ and

$$f_{n+1} = g_{n+1} + \min_{0 \leq k \leq n} (\alpha f_k + \beta f_{n-k}) \quad \text{for} \quad n \geq 0 \ , \tag{9.73}$$

where $\alpha, \beta > 0$, and g_n is some given sequence. Fredman and Knuth showed that if $g_n = 0$ for $n \geq 1$ and $\alpha + \beta < 1$, then

$$f_n \geq c n^{1+1/\gamma} \quad \text{for some} \quad c = c(\alpha, \beta) > 0 \ , \tag{9.74}$$

where γ is the solution to

$$\alpha^{-\gamma} + \beta^{-\gamma} = 1 \ . \tag{9.75}$$

They proved that $\lim_{n\to\infty} f_n n^{-1-1/\gamma}$ exists if and only if $(\log\alpha)/(\log\beta)$ is irrational. They also presented analyses of this recurrence for $\alpha+\beta\geq 1$, as well as of several recurrences that have different g_n.

The value of the Fredman-Knuth paper is less in the precise results they obtain for several recurrences of the type (9.73) than in the methods they develop, which allow one to analyze related problems. A crucial role in their approach is played by the observation that for the g_n they consider, the minimum in (9.73) can be located rather precisely. The conditions for such localization are applicable to many other sequences as well.

Further work on the recurrence (9.73) was done by Kapoor and Reingold [220], who obtained a complete solution under certain conditions. Their solution is complicated, expressed in terms of the weighted external path length of a binary tree. It is sufficiently explicit, though, to give a complete picture of the continuity, convexity, and oscillation properties of f_n. In some cases their solution simplifies dramatically.

Another class of quasi-linear recurrences involves the greatest integer function. Following [104], consider recurrences of the form

$$a(0)=1,\quad a(n)=\sum_{i=1}^{s} r_i a(\lfloor n/m_i\rfloor),\quad n\geq 1, \tag{9.76}$$

where $r_i>0$ for all i, and the m_i are integers, $m_i\geq 2$ for all i. Let $\tau>0$ be the (unique) solution to

$$\sum_{i=1}^{s} r_i m_i^{-\tau}=1\ . \tag{9.77}$$

If there is an integer d and integers u_i such that $m_i=d^{u_i}$ for $1\leq i\leq s$, then $\lim\ a(n)n^{-\tau}$ as $n\to\infty$ does not exist, but the limit of $a(d^k)d^{-k\tau}$ as $k\to\infty$ does exist. If there is no such d, then the limit of $a(n)n^{-\tau}$ as $n\to\infty$ does exist, and can readily be computed. For example, when

$$a(n)=a(\lfloor n/2\rfloor)+a(\lfloor n/3\rfloor)+a(\lfloor n/6\rfloor)\quad\text{for}\quad n\geq 1\ ,$$

this limit is $12(\log 432)^{-1}$. Convergence to the limit is extremely slow, as is shown in [104]. The method of proof used in [104] is based on renewal theory. Several other methods for dealing with recurrences of the type (9.76) are mentioned in [104] and the references listed in that paper. There are connections to other recurrences that are linear in two variables, such as

$$b(m,n)=b(m,n-1)+b(m-1,n)+b(m-1,n-1),\quad m,n\geq 1\ . \tag{9.78}$$

Consider an infinite sequence of integers $2 \le a_1 < a_2 < \dots$ such that

$$\sum_{j=1}^{\infty} a_j^{-1} \log a_j < \infty \ ,$$

and define $c(0) = 0$,

$$c(n) = \sum_{j=1}^{\infty} c(\lfloor n/a_j \rfloor) + 1, \qquad n \ge 1 \ . \tag{9.79}$$

If ρ is the (unique) positive solution to

$$\sum_{j=1}^{\infty} a_j^{-\rho} = 1 \ ,$$

then Erdös [103] showed that

$$c(n) \sim cn^{\rho} \quad \text{as} \quad n \to \infty \tag{9.80}$$

for a positive constant c. Although the recurrence (9.79) is similar to that of Eq. (9.76), the results are different (no oscillations can occur for a recurrence given by Eq. (9.79)) and the methods are dissimilar.

Karp [221] considers recurrences of the type $T(x) = a(x)+T(h(x))$, where x is a nonnegative real variable, $a(x) \ge 0$, and $h(x)$ is a random variable, $0 \le h(x) \le x$, with $m(x)$ being the expectation of $h(x)$. Such recurrences arise frequently in the analysis of algorithms, and Karp proves several theorems that bound the probability that $T(x)$ is large. For example, he obtains the following result.

Theorem 9.2. *Suppose that $a(x)$ is a nondecreasing continuous function that is strictly increasing on $\{x : \ a(x) > 0\}$, and $m(x)$ is a continuous function. Then for all $x \in \mathbb{R}^+$ and $k \in \mathbb{Z}^+$,*

$$\text{Prob}\left(T(x) \ge u(x) + ka(x)\right) \le \left(m(x)/x\right)^k \ ,$$

where $u(x)$ is the unique least nonnegative solution to the equation $u(x) = a(x) + u(m(x))$.

Another result, proved in [176], is the following estimate.

Theorem 9.3. *Suppose that $r, a_1, \dots, a_N \in \mathbb{R}^+$ and that $b \ge 0$. For $n > N$, define*

$$a_n = 1 + \max_{1 \le k \le n-1} \frac{b + a_{n-1} + a_{n-2} + \cdots + a_{n-k}}{k + r} \ . \tag{9.81}$$

Then

$$a_n \sim (n/r)^{1/2} \qquad \text{as} \qquad n \to \infty \ . \tag{9.82}$$

Theorem 9.3 is proved by an involved induction on the behavior of the a_n.

10. Analytic generating functions

Combinatorialists use recurrence, generating functions, and such transformations as the Vandermonde convolution; others, to my horror, use contour integrals, differential equations, and other resources of mathematical analysis.

J. Riordan [336]

The use of analytic methods in combinatorics did horrify Riordan. They are widespread, though, because of their utility, which even Riordan could not deny. About half of this chapter is devoted to such methods, as they are extremely flexible and give very precise estimates.

10.1. Introduction and general estimates

This section serves as an introduction to most of the remaining sections of the paper, which are concerned largely with the use of methods of complex variables in asymptotics. Many of the results to be presented later can be used with little or no knowledge of analytic functions. However, even some slight knowledge of complex analysis is helpful in getting an understanding of the scope and limitations of the methods to be discussed. There are many textbooks on analytic functions, such as [205, 364]. This chapter assumes that the reader has some knowledge of this field, but not a deep one. It reviews the concepts that are most relevant in asymptotic enumeration, and how they affect the estimates that can be obtained. It is not a general introduction to the subject of complex analysis, and the choices of topics, their ordering, and the decision of when to include proofs were all made with the goal of illustrating how to use complex analysis in asymptotics.

There are several definitions of analytic functions, all equivalent. For our purposes, it will be most convenient to call a function $f(z)$ of one complex variable *analytic* in a connected open set $S \subseteq \mathbb{C}$ if in a small neighborhood of every point $w \in S$, $f(z)$ has an expansion as a power series

$$f(z) = \sum_{n=0}^{\infty} a_n (z-w)^n, \quad a_n = a_n(w), \tag{10.1}$$

that converges. Practically all the functions encountered in asymptotic enumeration that are analytic are analytic in a disk about the origin. A necessary and sufficient condition for $f(z)$, defined by a power series (6.1), to be analytic in a neighborhood of the origin is that $|a_n| \le C^n$ for some constant $C > 0$. Therefore there is an effective dichotomy, with common generating functions either not converging near 0 and being only formal power series, or else converging

and being analytic.

A function $f(z)$ is called *meromorphic* in S if it is analytic in S except at a (countable isolated) subset $S' \subseteq S$, and in a small neighborhood of every $w \in S'$, $f(z)$ has an expansion of the form

$$f(z) = \sum_{n=-N(w)}^{\infty} a_n (z-w)^n \ , \qquad a_n = a_n(w) \ . \tag{10.2}$$

Thus meromorphic functions can have poles, but nothing more. Alternatively, a function is meromorphic in S if and only if it is the quotient of two functions analytic in S. In particular, z^{-5} is meromorphic throughout the complex plane, but $\sin(1/z)$ is not. In general, functions given by nice expressions are analytic away from obvious pathological points, since addition, multiplication, division, and composition of analytic functions usually yield analytic or meromorphic functions in the proper domains. Thus $\sin(1/z)$ is analytic throughout $\mathbb{C} \setminus \{0\}$, and so is z^{-5}, while $\exp(1/(1-z))$ is analytic throughout $\mathbb{C} \setminus \{1\}$, but is not meromorphic because of the essential singularity at $z = 1$. Not all functions that might seem smooth are analytic, though, as neither $f(z) = \bar{z}$ ($\bar{z}$ denoting the complex conjugate of z) nor $f(z) = |z|$ is analytic anywhere. The smoothness condition imposed by (10.1) is very stringent.

Analytic continuation is an important concept. A function $f(z)$ may be defined and analytic in S, but there may be another function $g(z)$ that is analytic in $S' \supset S$ and such that $g(z) = f(z)$ for $z \in S$. In that case we say that $g(z)$ provides an analytic continuation of $f(z)$ to S', and it is a theorem that this extension is unique. A simple example is provided by

$$\sum_{n=0}^{\infty} z^n = \frac{1}{1-z} \ . \tag{10.3}$$

The power series on the left side converges only for $|z| < 1$, and defines an analytic function there. On the other hand, $(1-z)^{-1}$ is analytic throughout $\mathbb{C} \setminus \{1\}$, and so provides an analytic continuation for the power series. This is a common phenomenon in asymptotic enumeration. Typically a generating function will converge in a disk $|z| < r$, will have a singularity at r, but will be continuable to a region of the form

$$\{z : |z| < r + \delta, \ |\mathrm{Arg}(z-r)| > \pi/2 - \epsilon\} \tag{10.4}$$

for $\delta, \ \epsilon > 0$. When this happens, it can be exploited to provide better or easier estimates of the coefficients, as is shown in Section 11.1. That section explains the reasons why continuation to a region of the form (10.4) is so useful.

If $f(z)$ is analytic in S, z is on the boundary of S, but $f(z)$ cannot be analytically continued to a neighborhood of z, we say that z is a *singularity* of $f(z)$. Isolated singularities that are not poles are called essential, so that $z = 1$ is an essential singularity of $\exp(1/(1-z))$, but not of $1/(1-z)$. (Note that $z = 1$ is an essential singularity of $f(z) = (1-z)^{1/2}$ even though $f(1) = 0$.) Throughout the rest of this chapter we will often refer to *large singularities* and *small singularities*. These are not precise concepts, and are meant only to indicate how fast the function $f(z)$ grows as $z \to z_0$, where z_0 is a singularity. If $z_0 = 1$, we say that $(1-z)^{1/2}$, $\log(1-z)$, $(1-z)^{-10}$ have small singularities, since $|f(z)|$ either decreases or grows at most like a negative power of $|1-z|$ as $z \to 1$. On the other hand, $\exp(1/(1-z))$ or $\exp((1-z)^{-1/5})$ will be said to have large singularities. Note that for $z = 1+iy$, $y \in \mathbb{R}$, $\exp(1/(1-z))$ is bounded, so the choice of path along which the singularity is approached is important. In determining the size of a singularity z_0, we will usually be concerned with real z_0 and generating functions $f(z)$ with nonnegative coefficients, and then usually will need to look only at z real, $z \to z_0^-$. When the function $f(z)$ is *entire* (that is, analytic throughout $\mathbb{C}$), we will say that ∞ is a singularity of $f(z)$ (unless $f(z)$ is a constant), and will use the large vs. small singularity classification depending on how fast $f(z)$ grows as $|z| \to \infty$. The distinction between small and large singularities is important in asymptotics because different methods are used in the two cases.

A simple closed contour Γ in the complex plane is given by a continuous mapping $\gamma : [0,1] \to \mathbb{C}$ with the properties that $\gamma(0) = \gamma(1)$, and that $\gamma(s) \neq \gamma(t)$ whenever $0 \leq s < t \leq 1$ and either $s \neq 0$ or $t \neq 1$. Intuitively, Γ is a closed path in the complex plane that does not intersect itself. For most applications that will be made in this chapter, simple closed contours Γ will consist of line segments and sections of circles. For such contours it is easy to prove that the complex plane is divided by the contour into two connected components, the inside and the outside of the curve. This result is true for all simple closed curves by the Jordan curve theorem, but this result is surprisingly hard to prove.

In asymptotic enumeration, the basic result about analytic functions is the Cauchy integral formula for their coefficients.

Theorem 10.1. *If $f(z)$ is analytic in an open set S containing 0, and*

$$f(z) = \sum_{n=0}^{\infty} a_n z^n \tag{10.5}$$

in a neighborhood of 0, *then for any* $n \geq 0$,

$$a_n = [z^n]f(z) = (2\pi i)^{-1} \int_\Gamma f(z) z^{-n-1} dz \ , \tag{10.6}$$

where Γ *is any simple closed contour in* S *that contains the origin inside it and is positively oriented (i.e., traversed in counterclockwise direction).*

An obvious question is why should one use the integral formula (10.6) to determine the coefficient a_n of $f(z)$. After all, the series (10.5) shows that

$$n!\, a_n = \frac{d^n}{dz^n} f(z)\Big|_{z=0} \ . \tag{10.7}$$

Unfortunately the differentiation involved in (10.7) is hard to control. Derivatives involve taking limits, and so even small changes in a function can produce huge changes in derivatives, especially high order ones. The special properties of analytic functions are not reflected in the formula (10.7), and for nonanalytic functions there is little that can be done. On the other hand, Cauchy's integral formula (10.6) does use special properties of analytic functions, which allow the determination of the coefficients of $f(z)$ from the values of $f(z)$ along any closed path. This determination involves integration, so that even coarse information about the size of $f(z)$ can be used with it. The analytic methods that will be outlined exploit the freedom of choice of the contour of integration to relate the behavior of the coefficients to the behavior of the function near just one or sometimes a few points.

If the power series (10.5) converges for $|z| < R$, and for the contour Γ we choose a circle $z = r\exp(i\theta)$, $0 \leq \theta \leq 2\pi$, $0 < r < R$, then the validity of (10.6) is easily checked by direct computation, since the power series converges absolutely and uniformly so one can interchange integration and summation. The strength of Cauchy's formula is in the freedom to choose the contour Γ in different ways. This freedom yields most of the powerful results to be discussed in the following sections, and later in this section we will outline how this is achieved. First we discuss some simple applications of Theorem 10.1 obtained by choosing Γ to be a circle centered at the origin.

Theorem 10.2. *If* $f(z)$ *is analytic in* $|z| < R$, *then for any* r *with* $0 < r < R$ *and any* $n \in Z$, $n \geq 0$,

$$|[z^n]f(z)| \leq r^{-n} \max_{|z|=r} |f(z)| \ . \tag{10.8}$$

The choice of Γ in Theorem 10.1 to be the circle of radius r gives Theorem 10.2. If $f(z)$, defined by (10.5), has $a_n \geq 0$ for all n, then

$$|f(z)| \leq \sum_{n=0}^{\infty} a_n |z|^n = f(|z|)$$

and therefore we obtain Lemma 8.1 as an easy corollary to Theorem 10.2. The advantage of Theorem 10.2 over Lemma 8.1 is that there is no requirement that $a_n \geq 0$. The bound of Theorem 10.2 is usually weaker than the correct value by a small multiplicative factor such as $n^{1/2}$.

If $f(z)$ is analytic in $|z| < R$, then for any $\delta > 0$, $f(z)$ is bounded in $|z| < R - \delta$, and so Theorem 10.2 shows that $a_n = [z^n]f(z)$ satisfies $|a_n| = O((R-\delta)^{-n})$. On the other hand, if $|a_n| = O(S^{-n})$, then the power series (10.5) converges for $|z| < S$ and defines an analytic function in that disk. Thus we obtain the easy result that if $f(z)$ is analytic in a disk $|z| < R$ but in no larger disk, then

$$\limsup |a_n|^{1/n} = R^{-1} \ . \tag{10.9}$$

Example 10.1. *Oscillating sequence.* Consider the sequence, discussed already in Example 9.4, given by

$$a_n = \sum_{k=0}^{n} \binom{n}{k} \frac{(-1)^k}{k!} \ , \qquad n = 0, 1, \ldots \ . \tag{10.10}$$

The maximal term in the sum (10.10) is of order roughly $\exp(cn^{1/2})$, so a_n cannot be much larger. However, the sum (10.10) does not show that a_n cannot be extremely small. Could we have $|a_n| \leq \exp(-n)$ for all n, say? That this is impossible is obvious from (9.39), though, by the argument above. The generating function $f(z)$, given by Eq. (9.39), is analytic in $|z| < 1$, but has an essential singularity at $z = 1$, so we immediately see that for any $\epsilon > 0$, $|a_n| < (1+\epsilon)^n$ for all sufficiently large n, and that $|a_n| > (1-\epsilon)^n$ for infinitely many n. (More powerful methods for dealing with analytic generating functions, such as the saddle point method to be discussed in Section 12, can be used to obtain the asymptotic relation for a_n given in Example 9.4.) ■

There is substantial literature dealing with the growth rate of coefficients of analytic functions. The book of Evgrafov [110] is a good reference for these results. However, the estimates presented there are not too useful for us, since they apply to wide classes of often pathological

functions. In combinatorial enumeration we usually encounter much tamer generating functions for which the crude bounds of [110] are obvious or easy to derive. Instead, we need to use the tractable nature of the functions we encounter to obtain much more delicate estimates.

The basic result, derived earlier, is that the power series coefficients a_n of a generating function $f(z)$, defined by (10.5), grow in absolute value roughly like R^{-n}, if $f(z)$ is analytic in $|z| < R$. A basic result about analytic functions says that if the Taylor series (10.5) of $f(z)$ converges for $|z| < R$ but for every $\epsilon > 0$ there is a z with $|z| = R + \epsilon$ such that the series (10.5) diverges at z, then $f(z)$ has a singularity z with $|z| = R$. Thus the exponential growth rate of the a_n is determined by the distance from the origin of the nearest singularity of $f(z)$, with close singularities giving large coefficients. Sometimes it is not obvious what R is. When the coefficients of $f(z)$ are positive, as is common in combinatorial enumeration and analysis of algorithms, there is a useful theorem of Pringsheim [364]:

Theorem 10.3. *Suppose that $f(z)$ is defined by Eq. (10.5) with $a_n \geq 0$ for all $n \geq n_0$, and that the series (10.5) for $f(z)$ converges for $|z| < R$ but not for any $|z| > R$. Then $z = R$ is a singularity of $f(z)$.*

As we remarked above, the exponential growth rate of the a_n is determined by the distance from the origin of the nearest singularity. Theorem 10.3 says that if the coefficients a_n are nonnegative, it suffices to look along the positive real axis to determine the radius of convergence R, which is also the desired distance to the singularity. There can be other singularities at the same distance from the origin (for example, $f(z) = (1 - z^2)^{-1}$ has singularities at $z = \pm 1$), but Theorem 10.3 guarantees that none are closer to 0 than the positive real one.

Since the singularities of smallest absolute value of a generating function exert the dominant influence on the asymptotics of the corresponding sequence, they are called the *dominant singularities*. In the most common case there is just one dominant singularity, and it is almost always real. However, we will sometimes speak of a large set of singularities (such as the k first order poles in Theorem 9.1, which are at different distances from the origin) as dominant ones. This allows some dominant singularities to be more influential than others.

Many techniques, including the elementary methods of Section 8, obtain bounds for summatory functions of coefficients even when they cannot estimate the individual coefficients. These methods succeed largely because they create a dominant singularity. If $f(z) = \sum f_n z^n$ converges for $|z| < 1$, diverges for $|z| > 1$, and has $f_n \geq 0$, then the singularity at $z = 1$ is at

least as large as any other. However, there could be other singularities on $|z| = 1$ that are just as large. (This holds for the functions $f_2(z)$ and $f_3(z)$ defined by (8.2) and (8.4).) When we consider the generating function of $\sum_{k \le n} f_k$, though, we find that

$$h(z) = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} f_k \right) z^n = (1-z)^{-1} f(z) \ , \tag{10.11}$$

so that $h(z)$ has a singularity at $z = 1$ that is much larger than any other one. That often provides enough of an extra boost to push through the necessary technical details of the estimates.

Most generating functions $f(z)$ have their coefficients $a_n = [z^n]f(z)$ real. If $f(z)$ is analytic at 0, and has real coefficients, then $f(z)$ satisfies the reflection principle,

$$f(z) = \overline{f(\overline{z})} \ . \tag{10.12}$$

This implies that zeros and singularities of $f(z)$ come in complex conjugate pairs.

The success of analytic methods in asymptotics comes largely from the use of Cauchy's formula (10.6) to estimate accurately the coefficients a_n. At a more basic level, this success comes because the behavior of an analytic function $f(z)$ reflects precisely the behavior of the coefficients a_n. In the discussion of elementary methods in Section 8, we pointed out that the behavior of a generating function for real arguments does not distinguish between functions with different coefficients. For example, the functions $f_1(z)$ and $f_3(z)$ defined by (8.1) and (8.4) are almost indistinguishable for $z \in \mathbb{R}$. However, they differ substantially in their behavior for complex z. The function $f_1(z)$ has only a first order pole at $z = 1$ and no other singularities, while $f_3(z)$ has poles at $z = 1$, $\exp(2\pi i/3)$, and $\exp(4\pi i/3)$. The three poles at the three cubic roots of unity reflect the modulo 3 periodicity of the coefficients of $f_3(z)$. This is a general phenomenon, and in the next section we sketch the general principle that underlies it. (The degree to which coefficients of an analytic function determine the behavior at the singularities is the subject of Abelian theorems. We will not need to delve into this subject to its full depth. For references, see [190, 364].)

Analytic methods are extremely powerful, and when they apply, they often yield estimates of unparalleled precision. However, there are tricky situations where analytic methods seem as if they ought to apply, but don't (at least not easily), whereas simpler approaches work.

Example 10.2. *Set partitions with distinct block sizes.* Let a_n be the number of partitions of a set of n elements into blocks of distinct sizes. Then $a_n = b_n \cdot n!$, where $b_n = [z^n]f(z)$, with

$$f(z) = \prod_{k=1}^{\infty} \left(1 + \frac{z^k}{k!}\right) . \tag{10.13}$$

The function $f(z)$ is entire and has nonnegative coefficients, so it might appear as an ideal candidate for an application of some of the methods for dealing with large singularities (such as the saddle point technique) that will be presented later. However, on circles $|z| = (n+1/2)/e$, $n \in \mathbb{Z}^+$, $f(z)$ does not vary much, so there are technical problems in applying these analytic methods. On the other hand, combinatorial estimates can be used to show [233] that the b_n behave in a "regularly irregular" way, so that, for example,

$$b_{m(m+1)/2-1} \sim b_{m(m+1)/2} \quad \text{as} \quad m \to \infty , \tag{10.14}$$

$$b_{m(m+1)/2} \sim m b_{m(m+1)/2+1} \quad \text{as} \quad m \to \infty . \tag{10.15}$$

These estimates are obtained by expanding the product in Eq. (10.13) and noting that

$$b_n = \sum_{\substack{1 \le k_1 < \cdots < k_r \\ \sum k_i = n}} \frac{1}{\prod_{i=1}^{r} k_i!} . \tag{10.16}$$

Since factorials grow rapidly, the only terms in the sum in (10.16) that are significant are those with small k_i. The term $b_n z^n$ for $n = m(m+1)/2$ for example, comes almost entirely from the product of $z^k/k!$, $1 \le k \le m$, all other products contributing an asymptotically negligible amount. ■

10.2. Subtraction of singularities

An important basic tool in asymptotics of coefficients of analytic functions is that of subtraction of singularities. If we wish to estimate $[z^n]f(z)$, and we know $[z^n]g(z)$, and the singularities of $f(z) - g(z)$ are smaller than those of $f(z)$, then we can usually conclude that $[z^n]f(z) \sim [z^n]g(z)$ as $n \to \infty$. In practice, given a function $f(z)$, we find the dominant singularities of $f(z)$ (usually poles), and construct a simple function $g(z)$ with those singularities. We illustrate this approach with several examples. The basic theme will recur in other sections.

Example 10.3. *Bernoulli numbers.* The Euler-Maclaurin summation formula, introduced in Section 5.3, involves the Bernoulli numbers B_n with exponential generating function

$$f(z) = \sum_{n=0}^{\infty} B_n \frac{z^n}{n!} = \frac{z}{e^z - 1} . \tag{10.17}$$

The denominator $\exp(z) - 1$ has zeros at 0, $\pm 2\pi i$, $\pm 4\pi i, \ldots$. The zero at 0 is canceled by the zero of z, so $f(z)$ is analytic for $|z| < 2\pi$, but has first order poles at $z = \pm 2\pi i$, $\pm 4\pi i, \ldots$. Consider

$$g(z) = 2\pi i \left(\frac{1}{z - 2\pi i} - \frac{1}{z + 2\pi i} \right) \ . \tag{10.18}$$

Then $f(z) - g(z)$ is analytic for $|z| < 4\pi$, so

$$|[z^n](f(z) - g(z))| = O((4\pi - \epsilon)^{-n}) \quad \text{as} \quad n \to \infty \tag{10.19}$$

for every $\epsilon > 0$. On the other hand,

$$[z^n]g(z) = \begin{cases} 0 & n \text{ odd} \ , \\ 2(2\pi)^{-n} & n \text{ even} \ . \end{cases} \tag{10.20}$$

This gives the leading term asymptotics of B_n. By taking more complicated $g(z)$, we can subtract more of the singularities of $f(z)$ and obtain more accurate expansions for B_n. It is even possible to obtain an exponentially rapidly convergent series for B_n. ■

Example 10.4. *Rational function asymptotics.* As another example of the subtraction of singularities principle, we sketch a proof of Theorem 9.1. Suppose that the hypotheses of that theorem are satisfied. Let

$$u(z) = \sum_{j=1}^{k} \frac{-g(\rho_j)}{\rho_j h'(\rho_j)(1 - z/\rho_j)} \ . \tag{10.21}$$

Then $f(z) - u(z)$ has no singularities in $|z| \le R$, and for $|z| = R$,

$$|f(z) - u(z)| \le |f(z)| + |u(z)| \le W + \delta^{-1} \sum_{j=1}^{k} |g(\rho_j)/h'(\rho_j)| \ . \tag{10.22}$$

Hence, by Theorem 10.2,

$$\Big| [z^n](f(z) - u(z)) \Big| \le W R^{-n} + \delta^{-1} R^{-n} \sum_{j=1}^{k} |g(\rho_j)/h'(\rho_j)| \ . \tag{10.23}$$

On the other hand,

$$[z^n]u(z) = -\sum_{j=1}^{k} \rho_j^{-n-1} g(\rho_j)/h'(\rho_j) \ . \tag{10.24}$$

The last two estimates yield Theorem 9.1. ■

The reader may have noticed that the proof of Theorem 9.1 presented above does not depend on $f(z)$ being rational. We have proved the following more general result.

Theorem 10.4. *Suppose that $f(z)$ is meromorphic in an open set containing $|z| \le R$, that it is analytic at $z = 0$ and on $|z| = R$, and that the only poles of $f(z)$ in $|z| < R$ are at $\rho_1, \ldots, \rho_k$, each of multiplicity 1. Suppose further that*

$$\max_{|z|=R} |f(z)| \le W \tag{10.25}$$

and that $R - |\rho_j| \ge \delta$ for some $\delta > 0$ and $1 \le j \le k$. Then

$$\left| [z^n] f(z) + \sum_{j=1}^{k} r_j \rho_j^{-n-1} \right| \le W R^{-n} + \delta^{-1} R^{-n} \sum_{j=1}^{k} |r_j| \ , \tag{10.26}$$

where r_j is the residue of $f(z)$ at ρ_j.

In the examples above, the dominant singularities were separated from other ones, so their contributions were larger than those of lower order terms by an exponential factor. Sometimes the singularity that remains after subtraction of the dominant one is on the same circle, and only slightly smaller. Section 11 presents methods that deal with some cases of this type, at least when the singularity is not large. What makes those methods work is the subtraction of singularities principle. Next we illustrate another application of this principle where the singularity is large. (The generating function is entire, and so the singularity is at infinity.)

Example 10.5. *Permutations without long increasing subsequences.* Let $u_k(n)$ be the number of permutations of $\{1, 2, \ldots, n\}$ that have no increasing subsequence of length $> k$. Logan and Shepp [257] and Vershik and Kerov [370] established by calculus of variations and combinatorics that the average value of the longest increasing subsequence in a random permutation is asymptotic to $2n^{1/2}$. Frieze [149] has proved recently, using probabilistic methods, a stronger result, namely that almost all permutations have longest increasing subsequences of length close to $2n^{1/2}$. Here we consider asymptotics of $u_k(n)$ for k fixed and $n \to \infty$. The Schensted correspondence and the hook formula express $u_k(n)$ in terms of Young diagrams with $\le k$ columns. For k fixed, there are few diagrams and their influence can be estimated explicitly using Stirling's formula, although Selberg-type integrals are involved and the analysis is complicated. This analysis was done by Regev [329], who proved more general results. Here we sketch another approach to the asymptotics of $u_k(n)$ for k fixed. It is based on a result of Gessel [161]. If

$$U_k(z) = \sum_{n=0}^{\infty} \frac{u_k(n) z^{2n}}{(n!)^2} \ , \tag{10.27}$$

then

$$U_k(z) = \det(I_{|i-j|}(2z))_{1\le i,j\le k} \ , \tag{10.28}$$

where the $I_m(x)$ are Bessel functions (Chapter 9 of [297]). H. Wilf and the author have noted that one can obtain the asymptotics of the $u_k(n)$ by using known asymptotic results about the $I_m(x)$. Eq. (9.7.1) of [297] states that for every $H \in \mathbb{Z}^+$,

$$I_m(z) = (2\pi z)^{-1/2} e^z \left(\sum_{h=0}^{H-1} c(m,h) z^{-h} + O(|z|^{-H}) \right) \ , \tag{10.29}$$

where this expansion is valid for $|z| \to \infty$ with $|\mathrm{Arg}(z)| \le 3\pi/8$, say. The $c(m,h)$ are explicit constants with $c(m,0) = 1$. Let us consider $k = 4$ to be concrete. Then, taking $H = 7$ in (10.29) (higher values of H are needed for larger k) we find from (10.28) that

$$U_4(z) = e^{8z}(3(256\pi^2 z^8)^{-1} + O(|z|^{-9})) \quad \text{for} \quad |z| \ge 1 \ . \tag{10.30}$$

It is also known that $I_m(-z) = (-1)^m I_m(z)$ and $I_m(z)$ is relatively small in the angular region $|\pi/2 - \mathrm{Arg}(z)| < \pi/8$. Therefore $U_4(-z) = U_4(z)$, and one can show that

$$|U_4(z)| = O(|z|^{-1} U_4(|z|)) \tag{10.31}$$

for z away from the real axis.

To apply the subtraction of singularities principle, we need an entire function $f(z)$ that is even, is large only near the real axis, and such that for $x \in \mathbb{R}$, $x \to \infty$,

$$f(x) \sim 3(256\pi^2 x^8)^{-1} \exp(8x) \ . \tag{10.32}$$

The function

$$f^*(z) = 3(128\pi^2 z^8)^{-1} \cosh(8z)$$

is even and has the desired asymptotic growth, but is not entire. We correct this defect by subtracting the contribution of the pole at $z = 0$, and let

$$f(z) = 3(128\pi^2 z^8)^{-1}(\cosh(8z) - 1 - 32z^2 - 512z^4/3 - 16384z^6/45 - 131072z^8/315) \ . \tag{10.33}$$

(It is not necessary to know explicitly the first 8 terms in the Taylor expansion of $\cosh(8z)$ that we wrote down above, as they do not affect the final answer.) With this definition

$$|U_4(z) - f(z)| = O(|z|^{-1} f(|z|)) \tag{10.34}$$

uniformly for all z with $|z| \geq 1$, say, and so if we apply Cauchy's theorem on the circle $|z| = n/4$, say, we find that

$$[z^{2n}](U_4(z) - f(z)) = O(n^{-2n}e^{2n}16^n n^{-9}) \ . \tag{10.35}$$

(The choice of $|z| = n/4$ is made to minimize the resulting estimate.) On the other hand, by Stirling's formula,

$$\begin{aligned} [z^{2n}]f(z) &= 3(128\pi^2)^{-1} \cdot ([z^{2n+8}]\cosh(8z)) \\ &= 3(128\pi^2)^{-1}8^{2n+8}/(2n+8)! \\ &\sim 1536\pi^{-5/2}n^{-2n}16^n e^{2n} n^{-17/2} \quad \text{as} \quad n \to \infty \ . \end{aligned} \tag{10.36}$$

Comparing (10.35) and (10.36), we see that

$$\begin{aligned} u_4(n) = (n!)^2[z^{2n}]U_4(z) &\sim (n!)^2 1536\pi^{-5/2}n^{-2n}16^n e^{2n}n^{-17/2} \\ &\sim 1536\pi^{-3/2}n^{-15/2}16^n \quad \text{as} \quad n \to \infty \ . \end{aligned} \tag{10.37}$$

■

Other methods can be applied to Gessel's generating function to obtain asymptotics of $u_k(n)$ for wider ranges of k ([306]).

The above example obtains a good estimate because the remainder term in (10.30) is smaller than the main term by a factor of $|z|^{-1}$. Had it been smaller only by a factor of $|z|^{-1/2}$, the resulting estimate would have been worthless, and it would have been necessary to obtain a fuller asymptotic expansion of $U_4(z)$ or else use smoothness properties of the remainder term. This is due to the phenomenon, mentioned before, that crude absolute value estimates in either Cauchy's theorem, or the elementary approaches of Section 8, usually lose a factor of $n^{1/2}$ when estimating the n-th coefficient.

The subtraction of singularities principle can be applied even when the generating functions seem to be more complicated than those of Example 10.5. If we consider the problem of that example, but with $k = 5$, then we find that

$$U_5(z) = 3\exp(10z)(5 \cdot 2^{13} \cdot \pi^{5/2}z^{25/2})^{-1}(1 + O(|z|^{-1})) \tag{10.38}$$

as $|z| \to \infty$, with $|\mathrm{Arg}(z)| \leq 3\pi/8$, $U_5(-z) = U_5(z)$, and $U_5(z)$ is entire. We now need an entire function with known coefficients that grows as $\exp(10z)z^{-25/2}$. This is not difficult to obtain, as

$$I_0(10z)z^{-12} - \sum_{j=1}^{12} c_j z^{-j} \tag{10.39}$$

for suitable coefficients c_j has the desired properties.

10.3. The residue theorem and sums as integrals

Sometimes sums that are not easily handled by other methods can be converted to integrals that can be evaluated explicitly or estimated by the residue theorem. If $t(z)$ is a meromorphic function that has first order poles at $z = a,\ a+1, \ldots, b$, with $a \in Z$, each with residue 1, then

$$\sum_{n=a}^{b} f(n) = \frac{1}{2\pi i} \int_\Gamma f(z)t(z)dz \ , \tag{10.40}$$

where Γ is a simple closed contour enclosing $a, a+1, \ldots, b$, provided $f(z)$ is analytic inside Γ and $t(z)$ has no singularities inside Γ aside from the first order poles at $a, a+1, \ldots, b$. If $t(z)$ is chosen to have residue $(-1)^n$ at $z = n$, then we obtain

$$\sum_{n=a}^{b} (-1)^n f(n) = \frac{1}{2\pi i} \int_\Gamma f(z)t(z)dz \ . \tag{10.41}$$

A useful example is given by the formula

$$\sum_{k=0}^{n} \binom{n}{k} (-1)^k f(k) = \frac{(-1)^n n!}{2\pi i} \int_\Gamma \frac{f(z)dz}{z(z-1)\cdots(z-n)} \ . \tag{10.42}$$

The advantage of (10.40) and (10.41) is that the integrals can often be manipulated to give good estimates. This is especially valuable for alternating sums such as (10.41). An analytic function $f(z)$ is extremely regular, so a sum such as that in (10.40) can often be estimated by methods such as the Euler-Maclaurin summation formula (Section 5.3). However, that formula cannot always be applied to alternating sums such as that of (10.41), because the sign change destroys the regularity of $f(n)$. (However, as is noted in Section 5.3, there are generalizations of the Euler-Maclaurin formula that are sometimes useful.) It is hard to write down general rules for applying this method, as most situations require appropriate choice of $t(z)$ and careful handling of the integral. For a detailed discussion of this method, often referred to as Rice's method, see Section 4.9 of [205]. A pair of popular functions to use as $t(z)$ are

$$t_1(z) = \pi/(\sin \pi z), \quad t_2(z) = \pi/(\tan \pi z) \ . \tag{10.43}$$

One can show (Theorem 4.9a of [205]) that if $r(z) = p(z)/q(z)$ with $p(z)$ and $q(z)$ polynomials such that $\deg q(z) \geq \deg p(z) + 2$, and $q(n) \neq 0$ for any $n \in Z$, then

$$\sum_{n=-\infty}^{\infty} r(n) \ = \ -\sum \mathrm{Res}(r(z)t_1(z)) \ , \tag{10.44}$$

$$\sum_{n=-\infty}^{\infty} (-1)^n r(n) \ = \ -\sum \mathrm{Res}(r(z)t_2(z)) \ , \tag{10.45}$$

where the sums on the right-hand sides above are over the zeros of $q(z)$.

Examples of applications of these methods to asymptotics of data structures are given in [141] and [360].

10.4. Location of singularities, Rouché's theorem, and unimodality

A recurrent but only implicit theme throughout the discussion in this section is that of isolation of zeros. For example, to apply Theorem 9.1 we need to know that the polynomial $h(z)$ has only k zeros, each of multiplicity one, in $|z| < R$. Proofs of such results can often be obtained with the help of Rouché's theorem [205, 364].

Theorem 10.5. *Suppose that $f_1(z)$ and $f_2(z)$ are functions that are analytic inside and on the boundary of a simple closed contour Γ. If*

$$|f_2(z)| < |f_1(z)| \quad \textit{for all} \quad z \in \Gamma \ , \tag{10.46}$$

then $f_1(z)$ and $f_1(z) + f_2(z)$ have the same number of zeros (counted with multiplicity) inside Γ.

Example 10.6. *Sequences with forbidden subblocks.* We consider again the topic of Examples 6.4, 6.8, and 9.2, and prove the results that were already used in Example 9.2. We again set

$$h(z) = z^k + (1-2z)C_A(z) \ , \tag{10.47}$$

where the only fact about $C_A(z)$ we will use is that it is a polynomial of degree $< k$ and coefficients 0 and 1, and $C_A(0) = 1$. We wish to show that $h(z)$ has only one zero in $|z| \le 0.6$ if k is large. Write

$$C_A(z) = 1 + \frac{1}{2}\sum_{j=1}^{\infty} z^j + \frac{1}{2}\sum_{j=1}^{\infty} \epsilon_j z^j \ , \tag{10.48}$$

where $\epsilon_j = \pm 1$ for each j. Then

$$C_A(z) = \frac{2-z}{2(1-z)} + u(z) \ , \tag{10.49}$$

where

$$|u(z)| \le \frac{|z|}{2(1-|z|)} \ .$$

For $|z| = r < 1$, we have $|u(z)| \le r/(2(1-r))$. On the other hand, $z \to (2-z)/(1-z)$ maps circles to circles, since it is a fractional linear transformation, so it takes the circle $|z| = r$ to

the circle with center on the real axis that goes through the two points $(2-r)/(1-r)$ and $(2+r)/(1+r)$. Therefore for $|z| = r < 1$,

$$|C_A(z)| \geq \frac{2+r}{2(1+r)} - \frac{r}{2(1-r)} = \frac{1-r-r^2}{1-r^2} \ , \tag{10.50}$$

and so $|C_A(z)| \geq 1/16$ for $|z| = r \leq 0.6$. Hence, if $k \geq 9$, then on $|z| = 0.6$,

$$|(1-2z)C_A(z)| \geq 1/80 > (0.6)^k \ , \tag{10.51}$$

and thus $(1-2z)C_A(z)$ and $h(z)$ have the same number of zeros in $|z| \leq 0.6$. On the other hand, $C_A(z)$ has no zeros in $|z| \leq 0.6$ by (10.50), while $1-2z$ has one, so we obtain the desired result, at least for $k \geq 9$. (A more careful analysis shows that $h(z)$ has only one root inside $|z| = 0.6$ even for $4 \leq k < 9$. For $1 \leq k \leq 3$, there are cases where there is no zero inside $|z| \leq 0.6$.) Example 6.7 shows how to obtain precise estimates of the single zero.

We note that (10.50) shows that for $|z| = 0.55$, $k \geq 9$

$$|h(z)| \geq |1-1.1|0.2 - (0.55)^k \geq 0.02 - 0.01 \geq 1/100 \ , \tag{10.52}$$

a result that was used in Example 9.2. ■

Example 10.7. *Coins in a fountain.* An (n,k) fountain is an arrangement of n coins in rows such that there are k coins in the bottom row, and such that each coin in a higher row touches exactly two coins in the next lower row. Let $a_{n,k}$ be the number of (n,k) fountains, and $a_n = \sum_k a_{n,k}$ the total number of fountains of n coins. The values of a_n for $1 \leq n \leq 6$ are $1, 1, 2, 3, 5, 9$. If we let $a_0 = 1$ then it can be shown [313] that

$$f(z) = \sum_{n=0}^{\infty} a_n z^n = \frac{1}{1 - \frac{z}{1 - \frac{z^2}{1 - \frac{z^3}{1 - \cdots}}}} \ . \tag{10.53}$$

This is a famous continued fraction of Ramanujan. (Other combinatorial interpretations of this continued fraction are also known, see the references in [313]. For related results, see [326, 327].) Although one can derive the asymptotics of the a_n from the expansion (10.53), it is more convenient to work with another expansion, known from previous studies of Ramanujan's continued fraction:

$$f(z) = \frac{p(z)}{q(z)} \ , \tag{10.54}$$

where

$$p(z) = \sum_{r\geq 0}(-1)^r \frac{z^{r(r+1)}}{(1-z)(1-z^2)\dots(1-z^r)} , \tag{10.55}$$

$$q(z) = \sum_{r\geq 0}(-1)^r \frac{z^{r^2}}{(1-z)(1-z^2)\dots(1-z^r)} . \tag{10.56}$$

Clearly both $p(z)$ and $q(z)$ are analytic in $|z|<1$, so $f(z)$ is meromorphic there. We will show that $q(z)$ has a simple real zero x_0, $0.57 < x_0 < 0.58$, and no other zeros in $|z| < 0.62$, while $p(x_0) > 0$. It will then follow from Theorem 10.4 that

$$a_n = cx_0^{-n} + O((5/3)^n) \quad \text{as} \quad n\to\infty , \tag{10.57}$$

where $c = -p(x_0)/(x_0 q'(x_0))$. Numerical computation shows that $c = 0.31236\dots$, $x_0 = 0.576148769\dots$.

To establish the claim about x_0, let $p_n(z)$ and $q_n(z)$ denote the n-th partial sums of the series (10.55) and (10.56), respectively. Write $a(z) = q_3(z)(1-z)(1-z^2)/(1-z^3)$, so that

$$a(z) = 1 - 2z - z^2 + z^3 + 3z^4 + z^5 - 2z^6 - z^7 - z^9 , \tag{10.58}$$

and consider

$$b(z) = \prod_{j=1}^{9}(z - z_j) ,$$

where the z_j are 0.57577, $-0.46997 \pm i0.81792$, $0.74833 \pm i0.07523$, $-1.05926 \pm i0.36718$, $0.49301 \pm i1.58185$, in that order. (The z_j are approximations to the zeros of $a(z)$, obtained from numerical library subroutines. How they were derived is not important for the verification of our proof.) An easy hand or machine computation shows that if $a(z) = \sum_k a_k z^k$, $b(z) = \sum b_k z^k$, then

$$\sum_{k=0}^{9} |a_k - b_k| \leq 1.7\times 10^{-4} ,$$

and so $|a(z) - b(z)| \leq 1.7\times 10^{-4}$ for all $|z| \leq 1$. Another computation shows that $|b(z)| \geq 8\times 10^{-4}$ for all $|z| = 0.62$.

On the other hand, for $0 \leq u \leq 0.62$ and $|z| = u$, we have for $k \geq 5$

$$\left|\frac{z^{(k+1)^2-k^2}}{1-z^{k+1}}\right| \leq \frac{u^{2k+1}}{1-u^{k+1}} \leq \frac{u^9}{1-u^5} . \tag{10.59}$$

Therefore

$$\left|\sum_{k=4}^{\infty}(-1)^k \frac{z^{k^2}}{\Pi_{j=4}^{k}(1-z^j)}\right| \leq \frac{u^{16}}{1-u^4}\sum_{m\geq 0}\left(\frac{u^9}{1-u^5}\right)^m \leq 6\times 10^{-4} , \tag{10.60}$$

and so by Rouché's theorem, $q(z)$ and $b(z)$ have the same number of zeros in $|z| \le 0.62$, namely 1. Since $q(z)$ has real coefficients, its zero is real. This establishes the existence of x_0. An easy computation shows that $q(0.57) > 0$, $q(0.58) < 0$, so $0.57 < x_0 < 0.58$.

To show that $p(x_0) > 0$, note that successive summands in (10.55) decrease in absolute magnitude for each fixed real $z > 0$, and $p(z) > 1 - z^2/(1-z) > 0$ for $0 < z < 0.6$. ■

The method used in the above example is widely applicable to generating functions given by continued fractions. Typically they are meromorphic in a disk centered at the origin, with a single dominant pole of order 1. Usually there is no convenient representation of the form (10.54) with explicit $p(z)$ and $q(z)$, and one has to work harder to establish the necessary properties about location of poles.

It was mentioned in Section 6.4 that unimodality of a sequence is often deduced from information about the zeros of the associated polynomial. If the zeros of the polynomial

$$A(z) = \sum_{k=0}^{n} a_k z^k$$

are real and ≤ 0, then the a_k are unimodal, and even the $a_k \binom{n}{k}^{-1}$ are log-concave. However, weaker properties follow from weaker assumptions on the zeros. If all the zeros of $A(z)$ are in the wedge-shaped region centered on the negative real axis $|\mathrm{Arg}(-z)| \le \pi/4$, and the a_k are real, then the a_k are log-concave, but the $a_k \binom{n}{k}^{-1}$ are not necessarily log-concave. (This follows by factoring $A(z)$ into polynomials with real coefficients that are either linear or quadratic, and noting that all have log-concave coefficients, so their product does too.) One can prove other results that allow zeros to lie in larger regions, but then it is necessary to impose restrictions on ratios of their distances from the origin.

10.5. Implicit functions

Section 6.2 presented functions, such as $f^{\langle -1 \rangle}(z)$, that are defined implicitly. In this section we consider related problems that arise when a generating function $f(z)$ satisfies a functional equation $f(z) = G(z, f(z))$. Such equations arise frequently in graphical enumeration, and there is a standard procedure invented by Pólya and developed by Otter that is almost algorithmic [188, 189] and routinely leads to them. Typically $G(z, w)$ is analytic in z and w in a small neighborhood of $(0, 0)$. Zeros of analytic functions in more than one dimension are not isolated, and by the implicit function theorem $G(z, w) = w$ is solvable for w as a function of

z, except for those points where

$$G_w(z,w) = \frac{\partial}{\partial w} G(z,w) = 1 \ . \tag{10.61}$$

Usually for z in a small neighborhood of 0 the solution w of $G(z,w) = w$ will not satisfy (10.61), and so w will be analytic in that neighborhood. As we enlarge the neighborhood under consideration, though, a simultaneous solution to $G(z,w) = w$ and (10.61) will eventually appear, and will usually be the dominant singularity of $f(z) = w(z)$. The following theorem covers many common enumeration problems.

Theorem 10.6. *Suppose that*

$$f(z) = \sum_{n=1}^{\infty} f_n z^n \tag{10.62}$$

is analytic at $z = 0$, that $f_n \geq 0$ for all n, and that $f(z) = G(z, f(z))$, where

$$G(z,w) = \sum_{m,n \geq 0} g_{m,n} z^m w^n \ . \tag{10.63}$$

Suppose that there exist real numbers $\delta, r, s > 0$ such that

(i) $G(z,w)$ is analytic in $|z| < r + \delta$ and $|w| < s + \delta$,

(ii) $G(r,s) = s$, $G_w(r,s) = 1$,

(iii) $G_z(r,s) \neq 0$ and $G_{ww}(r,s) \neq 0$.

Suppose that $g_{m,n} \in \mathbb{R}^+ \cup \{0\}$ for all m and n, $g_{0,0} = 0$, $g_{0,1} = 1$, and $g_{m,n} > 0$ for some m and some $n \geq 2$. Assume further that there exist $h > j > i \geq 1$ such that $f_h f_i f_j \neq 0$ while the greatest common divisor of $j - i$ and $h - i$ is 1. Then $f(z)$ converges at $z = r$, $f(r) = s$, and

$$f_n = [z^n] f(z) \sim (r G_z(r,s)/(2\pi G_{ww}(r,s)))^{1/2} n^{-3/2} r^{-n} \quad \text{as} \quad n \to \infty \ . \tag{10.64}$$

Example 10.8. *Rooted labeled trees.* As was shown in Example 6.1, the exponential generating function $t(z)$ of rooted labeled trees satisfies $t(z) = z \exp(t(z))$. Thus we have $G(z,w) = z \exp(w)$, and Theorem 10.6 is easily seen to apply with $r = e^{-1}$, $s = 1$. Therefore we obtain the asymptotic estimate

$$t_n/n! = [z^n] t(z) \sim (2\pi)^{-1/2} n^{-3/2} e^n \quad \text{as} \quad n \to \infty \ . \tag{10.65}$$

On the other hand, from Example 6.6 we know that $t_n = n^{n-1}$, a much more satisfactory answer, so that the estimate (10.65) only provides us with another proof of Stirling's formula. ■

The example above involves an extremely simple application of Theorem 10.6. More complicated cases will be presented in Section 15.1.

The statement of Theorem 10.6 is long, and the hypotheses stringent. All that is really needed for the asymptotic relation (10.64) to hold is that $f(z)$ should be analytic on $\{z : |z| \leq r, z \neq r\}$ and that

$$f(z) = c(r-z)^{1/2} + o(|r-z|^{1/2}) \tag{10.66}$$

for $|z-r| \leq \epsilon$, $|\mathrm{Arg}(r-z)| \geq \pi/2 - \epsilon$ for some $\epsilon > 0$. If these conditions are satisfied, then (10.64) follows immediately from either the transfer theorems of Section 11.1 or (with stronger hypotheses) from Darboux's method of Section 11.2. The purpose of Theorem 10.6 is to present a general theorem that guarantees (10.66) holds, is widely applicable, and is stated to the maximum extent possible in terms of conditions on the coefficients of $f(z)$ and $G(z,w)$.

Theorem 10.6 is based on Theorem 5 of [33] and Theorem 1 of [284]. The hypotheses of Theorem 5 of [33] are simpler than those of Theorem 10.6, but, as was pointed out by Canfield [67], the proof is faulty and there are counterexamples to the claims of that theorem. The difficulty is that Theorem 5 of [33] does not distinguish adequately between the different solutions $w = w(z)$ of $w = G(z,w)$, and the singularity of the combinatorially significant solution may not be the smallest among all singularities of all solutions. The result of Meir and Moon [284] provides conditions that assure such pathological behavior does not occur. (The statement of Theorem 10.6 incorporates some corrections to Theorem 1 of [284] provided by the authors of that paper.) It would be desirable to prove results like (10.64) under a simpler set of conditions.

In many problems the function $G(z,w)$ is of the form

$$G(z,w) = g(z)\phi(w) + h(z) \ , \tag{10.67}$$

where $g(z)$, $\phi(w)$, and $h(z)$ are analytic at 0. For this case Meir and Moon have proved a useful result (Theorem 2 of [284]) that implies an asymptotic estimate of the type (10.64). The hypotheses of that result are often easier to verify than those of Theorem 10.6 above. (As was noted by Meir and Moon, the last part of the conditions (4.12a) of [284] has to be replaced by the condition that $y_i > h_i$, $y_j > h_j$, and $y_k > h_k$ for some $k > j > i \geq 1$ with $\gcd(j-i, k-i) = 1$.)

Whenever Theorem 10.6 applies, $f_n = [z^n]f(z)$ equals the quantity on the right-hand side of (10.64) to within a multiplicative factor of $1 + O(n^{-1})$. One can derive fuller expansions for

the ratio when needed.

11. Small singularities of analytic functions

In most combinatorial enumeration applications, the generating function has a single dominant singularity. The methods used to extract asymptotic information about coefficients split naturally into two main classes, depending on whether this singularity is large or small.

In some situations the same generating function can be said to have either a large or a small singularity, depending on the range of coefficients that we are interested in. This is illustrated by the following example.

Example 11.1. *Partitions with bounded part sizes.* Let $p(n,m)$ be the number of (unordered) partitions of an integer n into integers $\le m$. It is easy to see that

$$P_m(z) = \sum_{n=0}^{\infty} p(n,m) z^n = \prod_{k=1}^{m} (1-z^k)^{-1} \ . \tag{11.1}$$

The function $P_m(z)$ is rational, but has to be treated in different ways depending on the relationship of n and m. If n is large compared to m, it turns out to be appropriate to say that $P_m(z)$ has a small singularity, and use methods designed for this type of problems. However, if n is not too large compared to m, then the singularity of $P_m(z)$ can be said to be large. (Since the largest part in a partition of n is almost always $O(n^{1/2}\log n)$ [105], $p(n,m) \sim p(n)$ if m is much larger than $n^{1/2}\log n$.)

Although $P_m(z)$ has singularities at all the k-th roots of unity for all $k \le m$, $z=1$ is clearly the dominant singularity, as $|P_m(r)|$ grows much faster as $r \to 1^-$ than $|P_m(z)|$ for $z = r\exp(i\theta)$ for any $\theta \in (0, 2\pi)$. If m is fixed, then the partial function decomposition can be used to obtain the asymptotics of $p(n,m)$ as $m \to \infty$. We cannot use Theorem 9.1 directly, since the pole of $P_m(z)$ at $z=1$ has multiplicity 1. However, either by using the generalizations of Theorem 9.1 that are mentioned in Section 9.1, or by the subtraction of singularities principle, we can show that for any fixed m,

$$p(n,m) \sim [z^n] \left(\prod_{k=1}^{m} k! \right)^{-1} (1-z)^{-m} \sim \left(\prod_{k=1}^{m} k! \right)^{-1} ((m-1)!)^{-1} \quad \text{as} \quad n \to \infty \ . \tag{11.2}$$

(See [23] for further details and estimates.) This approach can be extended for m growing slowly with n, and it can be shown without much effort that the estimate (11.2) holds for $n \to \infty$, $m \le \log\log n$, say. However, for larger values of m this approach becomes cumbersome, and other methods, such as those of Section 12, are necessary. ■

11.1. Transfer theorems

This section presents some results, drawn from [135], that allow one to translate an asymptotic expansion of a generating function around its dominant singularity into an asymptotic expansion for the coefficients in a direct way. These results are useful in combinatorial enumeration, since the conditions for validity are frequently satisfied. The proofs, which we do not present here, are based on the subtraction of singularities principle, but are more involved than in the cases treated in Section 10.2.

We start out with an application of the results to be presented later in this section.

Example 11.2. *2-regular graphs.* The generating function for 2-regular graphs is known [81] to be

$$f(z) = (1-z)^{-1/2} \exp\left(-\frac{1}{2}z - \frac{1}{4}z^2\right) . \tag{11.3}$$

(A simpler proof can be obtained from the exponential formula, cf. Eq. (3.9.1) of [377].) We see that $f(z)$ is analytic throughout the complex plane except for the slit along the real axis from 1 to ∞, and that near $z = 1$ it has the asymptotic expansion

$$f(z) = e^{-3/4}\left\{(1-z)^{-1/2} + (1-z)^{1/2} + \frac{1}{4}(1-z)^{3/2} + \cdots\right\} . \tag{11.4}$$

Theorem 11.1 below then shows that as $n \to \infty$,

$$\begin{aligned}[z^n]f(z) &\sim e^{-3/4}\left\{\binom{n-1/2}{n} + \binom{n-3/2}{n} + \frac{1}{4}\binom{n-5/2}{n} + \cdots\right\} \\ &\sim \frac{e^{-3/4}}{\sqrt{\pi n}}\left\{1 - \frac{5}{8n} - \frac{15}{128n^2} + \cdots\right\} . \quad \blacksquare \end{aligned} \tag{11.5}$$

The basic transfer results will be presented for generating functions that have a single dominant singularity, but can be extended substantially beyond their circle of convergence. For r, $\eta > 0$, and $0 < \phi < \pi/2$, we define the closed domain $\Delta = \Delta(r, \phi, \eta)$ by

$$\Delta(r, \phi, \eta) = \{z : \ |z| \le r + \eta, \ |\mathrm{Arg}(z - r)| \ge \phi\} . \tag{11.6}$$

In the main result below we will assume that a generating function is analytic throughout $\Delta \setminus \{r\}$. Later in this section we will mention some results that dispense with this requirement. We will also explain why analyticity throughout $\Delta \setminus \{r\}$ is helpful in obtaining results such as those of Theorem 11.1 below.

One advantage to using Cauchy's theorem to recover information about coefficients of generating functions is that it allows one to prove the intuitively obvious result that small smooth

changes in the generating function correspond to small smooth changes in the coefficients. We will use the quantitative notion of a function of slow variation at ∞ to describe those functions for which this notion can be made precise. (With more effort one can prove that the same results hold with a less restrictive definition than that below.)

Definition 11.1. *A function $L(u)$ is of slow variation at ∞ if*

i) There exist real numbers u_0 and ϕ_0 with $u_0 > 0$, $0 < \phi_0 < \pi/2$, such that $L(u)$ is analytic and $\neq 0$ in the domain

$$\{u : \ |\mathrm{Arg}(u - u_0)| \leq \pi - \phi_0\} \ . \tag{11.7}$$

ii) There exists a function $\epsilon(x)$, defined for $x \geq 0$ with $\lim_{x\to\infty} \epsilon(x) = 0$, such that for all $\theta \in [-(\pi - \phi_0), \pi - \phi_0]$ and $u \geq u_0$, we have

$$\left| \frac{L(ue^{i\theta})}{L(u)} - 1 \right| \ < \ \epsilon(u) \tag{11.8}$$

and

$$\left| \frac{L(u \log^2 u)}{L(u)} - 1 \right| \ < \ \epsilon(u) \ . \tag{11.9}$$

Theorem 11.1. *Assume that $f(z)$ is analytic throughout the domain $\Delta \setminus \{r\}$, where $\Delta = \Delta(r, \phi, \eta)$, $r, \eta > 0$, $0 < \phi < \pi/2$, and that $L(u)$ is a function of slow variation at ∞. If α is any real number, then*

A) If

$$f(z) = O\left((z - r)^{\alpha} L\left(\frac{1}{r - z} \right) \right)$$

uniformly for $z \in \Delta \setminus \{r\}$, then

$$[z^n] f(z) = O(r^{-n} n^{-\alpha - 1} L(n)) \quad as \quad n \to \infty \ .$$

B) If

$$f(z) = o\left((z - r)^{\alpha} L\left(\frac{1}{r - z} \right) \right)$$

uniformly as $z \to r$ for $z \in \Delta \setminus \{r\}$, then

$$[z^n] f(z) = o(r^{-n} n^{-\alpha - 1} L(n)) \quad as \quad n \to \infty \ .$$

C) If $\alpha \notin \{0, 1, 2, \ldots\}$ *and*

$$f(z) \sim (r - z)^{\alpha} L\left(\frac{1}{r - z}\right)$$

uniformly as $z \to r$ *for* $z \in \Delta \setminus \{r\}$*, then*

$$[z^n] f(z) \sim \frac{r^{-n} n^{-\alpha - 1}}{\Gamma(-\alpha)} L(n) \ .$$

The restriction that there be only one singularity on the circle of convergence is easy to relax. If there are several (corresponding to oscillatory behavior of the coefficients), their contributions to the coefficients add. The crucial fact is that at each singularity the function $f(z)$ should be continuous except for an angular region similar to that of $\Delta(r, \phi, \eta)$.

The requirement that the generating function $f(z)$ be analytic in the interior of $\Delta(r, \phi, \eta)$ is in general harder to dispense with, at least by the methods of [135]. However, if the singularity at r is sufficiently large, one can obtain the same results with weaker assumptions that only require analyticity inside the disk $|z| < r$. The following result is implicit in [135].

Theorem 11.2. *Assume that* $f(z)$ *is analytic in the domain*$\{z : |z| \leq r, z \neq r\}$ *and that* $L(u)$ *is a function of slow variation at* ∞*. If* α *is any fixed real number with* $\alpha < -1$*, then the implications A), B), and C) of Theorem 11.1 are valid.*

Example 11.3. *Longest cycle in a random permutation.* The average length of the longest cycle in a permutation on n letters is $[z^n] f(z)$, where

$$f(z) = (1 - z)^{-1} \sum_{k \geq 0} \left[1 - \exp\left(- \sum_{j \geq k} j^{-1} z^j \right) \right] \ .$$

It is easy to see that $f(z)$ is analytic in $|z| < 1$, and a double application of the Euler-Maclaurin summation formula shows that $f(z) \sim G(1 - z)^{-2}$ as $z \to 1$, uniformly for $|z| \leq 1$, $z \neq 1$, where

$$G = \int_0^{\infty} \left[1 - \exp\left(- \int_x^{\infty} t^{-1} e^{-t} dt \right) \right] dx = 0.624 \ldots \ . \tag{11.10}$$

Therefore, by Theorem 11.2 with $L(u) = 1$,

$$[z^n] f(z) \sim Gn \quad \text{as} \quad n \to \infty \ , \tag{11.11}$$

a result first proved by Shepp and Lloyd [342] using Poisson approximations and Tauberian theorems. The derivation sketched above follows [134, 135]. The paper [134] contains many

other applications of transfer theorems to random mapping problems. Additional recent papers on the cycle structure of random permutations are [19, 187]. They use probabilistic methods, not transfer theorems, and contain extensive references to other recent works. ■

In applying transfer theorems, it is useful to have explicit expansions and estimates for the coefficients of some frequently occurring functions. We state several asymptotic series:

$$[z^n](1-z)^\alpha \approx \frac{n^{-\alpha-1}}{\Gamma(-\alpha)}\left(1+\sum_{k\geq 1} e_k^{(\alpha)} n^{-k}\right), \quad \alpha \neq 0,1,2,\ldots, \tag{11.12}$$

where

$$e_k^{(\alpha)} = \sum_{j=k}^{2k} (-1)^j \lambda_{k,j} (\alpha+1)(\alpha+2)\cdots(\alpha+j)\ , \tag{11.13}$$

and the $\lambda_{k,j}$ are determined by

$$e^t(1+vt)^{-1-1/v} = \sum_{k,j\geq 0} \lambda_{k,j} v^k t^j\ . \tag{11.14}$$

In particular,

$$\begin{aligned} e_1^{(\alpha)} &= \alpha(\alpha+1)/2, \\ e_2^{(\alpha)} &= \alpha(\alpha+1)(\alpha+2)(3\alpha+1)/24\ . \end{aligned}$$

Also, for $\alpha, \beta \notin \{0,1,2,\ldots\}$,

$$[z^n](1-z)^\alpha(-z^{-1}\log(1-z))^\beta \approx \frac{n^{-\alpha-1}}{\Gamma(-\alpha)}(\log n)^\beta \left(1+\sum_{k\geq 1} e_k^{(\alpha,\beta)}(\log n)^{-k}\right)\ , \tag{11.15}$$

where

$$e_k^{(\alpha,\beta)} = (-1)^k \binom{\beta}{k} \Gamma(-\alpha)\left(\frac{d^k}{ds^k}\Gamma(-s)^{-1}\Big|_{s=\alpha}\right)\ . \tag{11.16}$$

Further examples of asymptotic expansions are presented in [135].

Why is the analyticity of a function $f(z)$ throughout $\Delta(r,\phi,\eta)\setminus\{r\}$ so important? We explain this using as an example a function $f(z)$ that satisfies

$$f(z) = (1+o(1))(1-z)^{1/2} \tag{11.17}$$

as $z\to 1$ with $z\in\Delta=\Delta(1,\pi/8,1)$. We write

$$f(z) = (1-z)^{1/2} + g(z)\ , \tag{11.18}$$

so that

$$|g(z)| = o(|1-z|^{1/2}) \ . \tag{11.19}$$

Since $[z^n](1-z)^{1/2}$ grows like $n^{-3/2}$, we would like to show that

$$|[z^n]g(z)| = o(n^{-3/2}) \quad \text{as} \quad n \to \infty \ . \tag{11.20}$$

If $g(z)$ were analytic in a disk of radius $1+\delta$ for some $\delta > 0$, then we could conclude that $|[z^n]g(z)| < (1+\delta/2)^{-n}$ for large n, a conclusion much stronger than (11.20). However, if all we know is that $g(z)$ satisfies (11.19) in $|z| \le 1$, then we can only conclude from Cauchy's theorem that $[z^n]g(z) = O(1)$, since (11.19) implies that $|g(z)| \le C$ for all $|z| < 1$ and some $C > 0$. Then Theorem 10.2 gives

$$|[z^n]g(z)| \le Cr^{-n} \tag{11.21}$$

uniformly for all $n \ge 0$ and all $r < 1$, and hence $|[z^n]g(z)| \le C$ for all n, a result that is far from what is required. If we know that $g(z)$ can be continued to $\Delta \setminus \{r\}$ and satisfies (11.19) there, we can do a lot better. We choose the contour $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$, pictured in Fig. 1, with

$$\Gamma_1 = \{z : \ |z-1| = 1/n, \ |\mathrm{Arg}(z-1)| \ge \pi/4\} \ , \tag{11.22}$$

$$\Gamma_2 = \{z : \ z = 1 + r\exp(\pi i/4), \ 1/n \le r \le \delta\} \ , \tag{11.23}$$

$$\Gamma_3 = \{z : \ |z| = |1+\delta\exp(\pi i/4)|, \ |\mathrm{Arg}(z-1)| \ge \pi/4\} \ , \tag{11.24}$$

$$\Gamma_4 = \{z : \ z = 1 + r\exp(-\pi i/4), \ 1/n \le r \le \delta\} \ , \tag{11.25}$$

where $0 < \delta < 1/2$. We will show that the integrals

$$g_j = \frac{1}{2\pi i}\int_{\Gamma_j} g(z) z^{-n-1} dz \tag{11.26}$$

on the Γ_j are small. On Γ_3, $g(z)$ is bounded, so we trivially obtain the exponential upper bound

$$|g_3| = O((1+\delta/2)^{-n}) \ . \tag{11.27}$$

On Γ_1, $|g(z)| = o(n^{-1/2})$, $|z^{-n-1}| \le (1-1/n)^{-n-1} = O(1)$, and the length of Γ_1 is $\le 2\pi/n$, so

$$|g_1| = o(n^{-3/2}) \quad \text{as} \quad n \to \infty \ . \tag{11.28}$$

Next, on Γ_2, for $z = 1 + r\exp(\pi i/4)$,

$$\begin{aligned} |z|^{-n} &= |1 + r2^{-1/2} + ir2^{-1/2}|^{-n} = (1 + r2^{1/2} + r^2)^{-n/2} \\ &\le (1+r)^{-n/2} \le \exp(-nr/10) \end{aligned} \tag{11.29}$$

for $0 \le r < 1$. Since $g(z)$ satisfies (11.19), for any $\epsilon > 0$ we have

$$|g(1 + r\exp(\pi i/4))| \le \epsilon r^{1/2} \tag{11.30}$$

if $0 < r \le \eta$ for some $\eta = \eta(\epsilon) \le \delta$. Therefore

$$\begin{aligned} |g_2| &\le \epsilon \int_0^\eta r^{1/2} \exp(-nr/10) dr + O\left(\int_\eta^\infty \exp(-nr/10) dr \right) \\ &\le \epsilon n^{-3/2} \int_0^\infty r^{1/2} \exp(-r/10) dr + O(\exp(-n\eta/10)) \ , \end{aligned} \tag{11.31}$$

and so

$$|g_2| = o(n^{-3/2}) \ . \tag{11.32}$$

Since $|g_4| = |g_2|$, inequalities (11.27), (11.28), and (11.32) show that (11.20) holds.

The critical factor in the derivation of (11.20) was the bound for (11.29) for $|z|^{-n}$ on the segment $z = 1 + r\exp(\pi i/4)$. Integrating on the circle $|z| = 1$ or even on the line $\mathrm{Re}\,(z) = 1$ does not give a bound for $|z|^{-n}$ that is anywhere as small, and the resulting bounds do not approach (11.20) in strength. The use of the circular arc Γ_1 in the integral is only a minor technical device used to avoid the singularity at $z = 1$.

When one cannot continue a function to a region like $\Delta \setminus \{1\}$, it is sometimes possible to obtain good estimates for coefficients by working with the generating function exclusively in $|z| \le 1$, provided some smoothness properties apply. This method is outlined in the next section.

11.2. Darboux's theorem and other methods

A singularity of $f(z)$ at $z = w$ is called algebraic if $f(z)$ can be written as the sum of a function analytic in a neighborhood of w and a finite number of terms of the form

$$(1 - z/w)^\alpha g(z) \ , \tag{11.33}$$

where $g(z)$ is analytic near w, $g(w) \neq 0$, and $\alpha \notin \{0, 1, 2, \ldots\}$. Darboux's theorem [87] gives asymptotic expansions for functions with algebraic singularities on the circle of convergence. We state one form of Darboux's result, derived from Theorem 8.4 of [354].

Theorem 11.3. *Suppose that $f(z)$ is analytic for $|z| < r$, $r > 0$, and has only algebraic singularities on $|z| = r$. Let a be the minimum of $Re\,(\alpha)$ for the terms of the form (11.33) at*

the singularities of $f(z)$ *on* $|z| = r$*, and let* w_j*,* α_j*, and* $g_j(z)$ *be the* w*,* α*, and* $g(z)$ *for those terms of the form (11.33) for which* $Re\,(\alpha) = a$*. Then, as* $n \to \infty$*,*

$$[z^n]f(z) - \sum_j \frac{g_j(w_j)n^{-\alpha_j-1}}{\Gamma(-\alpha_j)w_j^n} + o(r^{-n}n^{-a-1}) \ . \tag{11.34}$$

Jungen [219] has extended Darboux's theorem to functions that have a single dominant singularity which is of a mixed algebraic and logarithmic form. His method can be applied also to functions that have several such singularities on their circle of convergence.

We do not devote much attention to Darboux's and Jungen's theorems because they can be obtained from the transfer theorems of Section 11.1. The only reason for stating Theorem 11.3 is that it occurs frequently in the literature.

Some functions, such as

$$f(z) = \prod_{k=1}^{\infty}(1 + z^k/k^2) \ , \tag{11.35}$$

are analytic in $|z| \le 1$, cannot be continued outside the unit circle, yet are nicely behaved on $|z| = 1$. Therefore there is no dominant singularity that can be studied to determine the asymptotics of $[z^n]f(z)$. To minimize the size of the integrand, it is natural to move the contour of integration in Cauchy's formula to the unit circle. Once that is done, it is possible to exploit smoothness properties of $f(z)$ to bound the coefficients. The Riemann-Lebesgue lemma implies that if $f(z)$ is integrable on the unit circle, then as $n \to \infty$,

$$[z^n]f(z) = (2\pi)^{-1}\int_{-\pi}^{\pi} f(e^{i\theta})\exp(-ni\theta)d\theta = o(1) \ . \tag{11.36}$$

More can be said if the derivative of $f(z)$ exists on the unit circle. When we apply integration by parts to the integral in (11.36), we find

$$[z^n]f(z) = (2\pi n)^{-1}\int_{-\pi}^{\pi} f'(e^{i\theta})\exp(-(n-1)i\theta)d\theta \ , \tag{11.37}$$

and so $|[z^n]f(z)| = o(n^{-1})$ if $f'(z)$ exists and is integrable on the unit circle. Existence of higher derivatives leads to even better estimates. We do not attempt to state a general theorem, but illustrate an application of this method with an example. The same technique can be used in other situations, for example in obtaining better error terms in Darboux's theorem [87].

Example 11.4. *Permutations with distinct cycle lengths.* Example 8.5 showed that for the function $f(z)$ defined by Eq. (8.58), $[z^n]f(z) \sim \exp(-\gamma)$ as $n \to \infty$. This coefficient is the probability that a random permutation on n letters has distinct cycle lengths. The more precise

estimate (8.59) was derived by Greene and Knuth [177] by working with recurrences for the coefficients of $f(z)$ and auxiliary functions. Another approach to deriving fuller asymptotic expansions for $[z^n]f(z)$ is to use the method outlined above. It suffices to show that the function $g(z)$ defined by Eq. (8.62) has a nice expansion in the closed disk $|z| \leq 1$. Since

$$g(z) = -z + \sum_{m=2}^{\infty} \frac{(-1)^{m-1}}{m} \{\mathrm{Li}_m(z^m) - z^m\} \ , \tag{11.38}$$

where the $\mathrm{Li}_m(w)$ are the polylogarithm functions [251], one can use the theory of the $\mathrm{Li}_m(w)$. A simpler way to proceed is to note, for example, that

$$\sum_{k=2}^{\infty} \frac{z^{2k}}{k^2} = \sum_{k=2}^{\infty} \frac{z^{2k}}{k(k-1)} + r(z) \ , \tag{11.39}$$

where

$$r(z) = -\sum_{k=2}^{\infty} \frac{z^{2k}}{k^2(k-1)} \ , \tag{11.40}$$

and so $r'(z)$ is bounded and continuous for $|z| \leq 1$, as are the terms in (8.62) with $m \geq 3$. On the other hand,

$$\sum_{k=2}^{\infty} \frac{z^{2k}}{k(k-1)} = z^2 + (1-z^2)\log(1-z^2) \ , \tag{11.41}$$

so we can write $g(z) = g_1(z) + g_2(z)$, where $g_1(z)$ is an explicit function (given by Eq. (11.41)) such that the coefficients of $\exp(g_1(z))$ can be estimated asymptotically using transfer methods or other techniques, and $g_2(z)$ has the property that $g_2'(z)$ is bounded and continuous in $|z| \leq 1$. Continuing this process, we can find, for every K, an expansion for the coefficients of $f(z)$ that has error term $O(n^{-K})$. To do this, we write $g(z) = G_1(z) + G_2(z)$. In this expansion $G_1(z)$ will be explicitly given and analytic inside $|z| < 1$ and analytically continuable to some region that extends beyond the unit disk with the exception of cuts from a finite number of points on the unit circle out to infinity. Further, $G_2(z)$ will have the property that $G_2^{(K)}(z)$ is bounded and continuous in $|z| \leq 1$. This will then give the desired expansion for the coefficients of $f(z)$. ■

12. Large singularities of analytic functions

This section presents methods for asymptotic estimation of coefficients of generating functions whose dominant singularities are large.

12.1. The saddle point method

The saddle point method, also referred to as the method of steepest descent, is by far the most useful method for obtaining asymptotic information about rapidly growing functions. It is extremely flexible and has been applied to a tremendous variety of problems. It is also complicated, and there is no simple categorization of situations where it can be applied, much less of the results it produces. Given the purpose and limitations on the length of this chapter, we do not present a full discussion of it. For a complete and insightful introduction to this technique, the reader is referred to [63]. Many other books, such as [110, 115, 315, 385] also have extensive presentations. What this section does is to outline the method, show when and how it can be applied and what kinds of estimates it produces. Examples of proper and improper applications of the method are presented. Later subsections are then devoted to general results obtained through applications of the saddle point method. These results give asymptotic expansions for wide classes of functions without forcing the reader to go through the details of the saddle point method.

The saddle point method is based on the freedom to shift contours of integration when estimating integrals of analytic functions. The same principle underlies other techniques, such as the transfer method of Section 11.1, but the way it is applied here is different. When dealing with functions of slow growth near their principal singularity, as happens for transfer methods, one attempts to push the contour of integration up to and in some ways even beyond the singularity. The saddle point method is usually applied when the singularity is large, and it keeps the path of integration close to the singularity.

In the remainder of this section we will assume that $f(z)$ is analytic in $|z| < R \leq \infty$. We will also make the assumption that for some R_0, if $R_0 < r < R$, then

$$\max_{|z|=r} |f(z)| = f(r) \ . \tag{12.1}$$

This assumption is clearly satisfied by all functions with real nonnegative coefficients, which are the most common ones in combinatorial enumeration. Further, we will suppose that $z = r$ is the unique point with $|z| = r$ where the maximum value in (12.1) is assumed. When this assumption is not satisfied, we are almost always dealing with some periodicity in the asymptotics of the coefficients, and we can then usually reduce to the standard case by either changing variables or rewriting the generating function as a sum of several others, as was discussed in Section 10. (Such a reduction cannot be applied to the function of Eq. (9.39),

though.)

The first step in estimating $[z^n]f(z)$ by the saddle point method is to find the saddle point. Under our assumptions, that will be a point $r \in (R_0, R)$ which minimizes $r^{-n}f(r)$. We have encountered this condition before, in Section 8.1. The minimizing $r = r_0$ will usually be unique, at least for large n. (If there are several $r \in (R_0, R)$ for which $r^{-n}f(r)$ achieves its minimum value, then $f(z)$ is pathological, and the standard saddle point method will not be applicable. For functions $f(z)$ with nonnegative coefficients, it is easy to show uniqueness of the minimizing r, as was already discussed in Section 8.1.) Cauchy's formula (10.6) is then applied with the contour $|z| = r_0$. The reason for this choice is that for many functions, on this contour the integrand is large only near $z = r_0$, the contributions from the region near $z = r_0$ do not cancel each other, and remaining regions contribute little. This is in contrast to the behavior of the integrand on other contours. By Cauchy's theorem, any simple closed contour enclosing the origin gives the correct answer. However, on most of them the integrand is large, and there is so much cancellation that it is hard to derive any estimates. The circle going through the saddle point, on the other hand, yields an integral that can be controlled well by techniques related to Laplace's method and the method of stationary phase that were mentioned in Section 5.5. We illustrate with an example, which is a totally self-contained application of the saddle point method to an extremely simple situation.

Example 12.1. *Stirling's formula.* We estimate $(n!)^{-1} = [z^n]\exp(z)$. The saddle point, according to our definition above, is that $r \in \mathbb{R}^+$ that minimizes $r^{-n}\exp(r)$, which is clearly $r = n$. Consider the contour $|z| = n$, and set $z = n\exp(i\theta)$, $-\pi \le \theta \le \pi$. Then

$$\begin{aligned} [z^n]\exp(z) &= \frac{1}{2\pi i}\int_{|z|=n} \frac{\exp(z)}{z^{n+1}}dz \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} n^{-n}\exp(ne^{i\theta} - ni\theta)d\theta \ . \end{aligned} \tag{12.2}$$

Since $|\exp(z)| = \exp(\mathrm{Re}(z))$, the absolute value of the integrand in (12.2) is $n^{-n}\exp(n\cos\theta)$, which is maximized for $\theta = 0$. Now

$$e^{i\theta} = \cos\theta + i\sin\theta = 1 - \theta^2/2 + i\theta + O(|\theta|^3) \ ,$$

so for any $\theta_0 \in (0, \pi)$,

$$\int_{-\theta_0}^{\theta_0} n^{-n}\exp(ne^{i\theta} - ni\theta)d\theta = \int_{-\theta_0}^{\theta_0} n^{-n}\exp(n - n\theta^2/2 + O(n|\theta|^3))d\theta \ . \tag{12.3}$$

(It is the cancellation of the $ni\theta$ term coming from $ne^{i\theta}$ and the $-ni\theta$ term that came from change of variables in z^{-n} that is primarily responsible for the success of the saddle point method.) The $O(n|\theta|^3)$ term in (12.3) could cause problems if it became too large, so we will select $\theta_0 = n^{-2/5}$, so that $n|\theta|^3 \le n^{-1/5}$ for $|\theta| \le \theta_0$, and therefore

$$\exp(n - n\theta^2/2 + O(n|\theta|^3)) = \exp(n - n\theta^2/2)(1 + O(n^{-1/5})) \ . \tag{12.4}$$

Hence

$$\int_{-\theta_0}^{\theta_0} n^{-n} \exp(ne^{i\theta} - ni\theta)d\theta = (1 + O(n^{-1/5}))n^{-n}e^n \int_{-\theta_0}^{\theta} \exp(-n\theta^2/2)d\theta \ .$$

But

$$\begin{aligned} \int_{-\theta_0}^{\theta_0} \exp(-n\theta^2/2)d\theta &= \int_{-\infty}^{\infty} \exp(-n\theta^2/2)d\theta - 2\int_{\theta_0}^{\infty} \exp(-n\theta^2/2)d\theta \\ &= (2\pi/n)^{1/2} - O(\exp(-n^{1/5}/2)) \ , \end{aligned}$$

so

$$\int_{-\theta_0}^{\theta_0} n^{-n} \exp(ne^{i\theta} - ni\theta)d\theta = (1 + O(n^{-1/5}))(2\pi/n)^{1/2}n^{-n}e^n \ . \tag{12.5}$$

On the other hand, for $\theta_0 < |\theta| \le \pi$,

$$\cos\theta \le \cos\theta_0 = 1 - \theta_0^2/2 + O(\theta_0^4) \ ,$$

so

$$n\cos\theta \le n - n^{1/5}/2 + O(n^{-3/5}) \ ,$$

and therefore for large n

$$\left| \int_{\theta_0}^{\pi} n^{-n} \exp(ne^{i\theta} - ni\theta)d\theta \right| \le n^{-n} \exp(n - n^{1/5}/3) \ ,$$

and similarly for the integral from $-\pi$ to $-\theta_0$. Combining all these estimates we therefore find that

$$(n!)^{-1} = [z^n] \exp(z) = (1 + O(n^{-1/5}))(2\pi n)^{-1/2}n^{-n}e^n \ , \tag{12.6}$$

which is a weak form of Stirling's formula (4.3). (The full formula can be derived by using more precise expansions for the integrand.)

Suppose we try to push through a similar argument using the contour $|z| = 2n$. This time, instead of Eq. (12.2), we find

$$[z^n] \exp(z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} 2^{-n}n^{-n} \exp(2ne^{i\theta} - ni\theta)d\theta \ . \tag{12.7}$$

At $\theta = 0$, the integrand is $2^{-n}n^{-n}\exp(2n)$, which is $\exp(n)$ times as large as the value of the integrand in (12.2). Since the two integrals do produce the same answer, and from the analysis above we see that this answer is close to $n^{-n}\exp(n)$ in value, the integral in (12.7) must involve tremendous cancellation. That is indeed what we see in the neighborhood of $\theta = 0$. We find that

$$\exp(2ne^{i\theta} - ni\theta) = \exp(2n - n\theta^2 + ni\theta + O(n|\theta|^3)) \ , \tag{12.8}$$

and the $\exp(ni\theta)$ term produces wild oscillations of the integrand even over small ranges of θ. Trying to work with the integral (12.7) and proving that it equals something exponentially smaller than the maximal value of its integrand is not a promising approach. By contrast, the saddle point contour used to produce Eq. (12.2) gives nice behavior of the integrand, so that it can be evaluated. ■

The estimates for $n!$ obtained in Example 10.1 came from a simple application of the saddle point method. The motivation for the choice of the contour $|z| = n$ is provided by the discussion at the end of the example; other choices lead to oscillating integrands that cannot be approximated by a Gaussian, nor by any other nice function. The example above treated only the exponential function, but it is easy to see that this phenomenon is general; a rapidly oscillating term $\exp(ni\alpha)$ for $\alpha \neq 0$ is present unless the contour passes through the saddle point. When we do use this contour, and the Gaussian approximation is valid, we find that for functions $f(z)$ satisfying our assumptions we have the following estimate.

Saddle point approximation

$$[z^n]f(z) \sim (2\pi b(r_0))^{-1/2} f(r_0) r_0^{-n} \quad \text{as} \quad n \to \infty \ , \tag{12.9}$$

where r_0 is the saddle point (where $r^{-n}f(r)$ is minimized, so that $r_0 f'(r_0)/f(r_0) = n$) and

$$b(r) = r\frac{f'(r)}{f(r)} + r^2\frac{f''(r)}{f(r)} - r^2\left(\frac{f'(r)}{f(r)}\right)^2 = r\left(r\frac{f'(r)}{f(r)}\right)' \ . \tag{12.10}$$

Example 12.2. *Bell numbers.* Example 5.4 showed how to estimate the Bell number B_n by elementary methods, starting with the representation (5.38). The exponential generating function

$$B(z) = \sum_{n=0}^{\infty} B_n \frac{z^n}{n!} \tag{12.11}$$

satisfies

$$B(z) = \exp(\exp(z) - 1) \ ,$$

as can be seen from (5.38) or by other methods (cf. [81]). The saddle point occurs at that $r_0 > 0$ that satisfies

$$r_0 \exp(r_0) = n \ , \tag{12.12}$$

and

$$b(r_0) = r_0(1 + r_0) \exp(r_0) \ , \tag{12.13}$$

so the saddle point approximation says that as $n \to \infty$,

$$B_n \sim n!(2\pi r_0^2 \exp(r_0))^{-1/2} \exp(\exp(r_0) - 1) r_0^{-n} \ . \tag{12.14}$$

The saddle point approximation can be justified even more easily than for the Stirling estimate of $n!$. ■

The above approximation is widely applicable and extremely useful, but care has to be exercised is applying it. This is shown by the next example.

Example 12.3. *Invalid application of the saddle point method.* Consider the trivial example $f(z) = (1 - z)^{-1}$, so that $[z^n]f(z) = 1$ for all $n \geq 0$. Then $f'(r)/f(r) = (1 - r)^{-1}$, and so the saddle point is $r_0 = n/(n + 1)$, and $b(r_0) = r_0/(1 - r_0)^2 = n(n + 1)$. Therefore if the approximation (12.9) were valid, it would give

$$\begin{aligned} [z^n]f(z) &\sim (2\pi n(n + 1))^{-1/2}(n + 1)\left(1 + \frac{1}{n}\right)^n \\ &\sim (2\pi)^{-1/2} e \quad \text{as} \quad n \to \infty \ . \end{aligned} \tag{12.15}$$

Since $(2\pi)^{-1/2} e = 1.0844\ldots \neq 1 = [z^n]f(z)$, something is wrong, and the estimate (12.9) does not apply to this function. ■

The estimate (12.9) gave the wrong result in Example 12.3 because the Gaussian approximation on the saddle point method contour used so effectively in Example 12.1 (and in almost all cases where the saddle point method applies) does not hold over a sufficiently large region for $f(z) = (1 - z)^{-1}$. In Example 12.1 we used without detailed explanation the choice $\theta_0 = n^{-2/5}$, which gave the approximation (12.5) for $|\theta| \leq \theta_0$, and yet led to an estimate for the integral over $\theta_0 < |\theta| \leq \pi$ that was negligible. This was possible because the third order term

(i.e., $n|\theta|^3$) in Eq. (12.5) was small. When we try to imitate this approach for $f(z) = (1-z)^{-1}$, we fail, because the third order term is too large. Instead of $ne^{i\theta} - ni\theta$, we now have

$$-\log(1 - r_0 e^{i\theta}) - ni\theta = -\log(1 - r_0) - \frac{1}{2}n(n+1)\theta^2 - \frac{i}{6}n^2(n+1)\theta^3 + \cdots . \tag{12.16}$$

More fundamentally, the saddle point method fails here because the function $f(z) = (1-z)^{-1}$ does not have a large enough singularity at $z = 1$, so that when one traverses the saddle point contour $|z| = r_0$, the integrand does not drop off rapidly enough for a small region near the real axis to provide the dominant contribution.

When can one apply the saddle point approximation (12.9)? Perhaps the simplest, yet still general, set of sufficient conditions for the validity of (12.9) is provided by requiring that the function $f(z)$ be Hayman-admissible. Hayman admissibility is described in Definition 12.1, in the following subsection. Generally speaking, though, for the saddle point method to apply we need the function $f(z)$ to have a large dominant singularity at R, so that $f(r)$ grows at least as fast as $\exp((\log(R-r))^2)$ as $r \to R^-$ for $R < \infty$, and as fast as $\exp((\log r)^2)$ as $r \to \infty$ for $R = \infty$. The faster the growth rate, the easier it usually is to apply the method, so that $\exp(1/(1-z))$ or $\exp(\exp(1/(1-z)))$ can be treated easily.

In our application of the saddle point method to $\exp(z)$ in Example 12.1 we were content to obtain a poor error term, $1 + O(n^{-1/5})$, in Stirling's formula for $n!$. This was done to simplify the presentation and concentrate only on the main factors that make the saddle point method successful. With more care devoted to the integral one can obtain the full asymptotic expansion of $n!$. (Only the range $|\theta| \le \theta_0$ has to be considered carefully.) This is usually true when the saddle point method is applicable.

This section provided a sketchy introduction to the saddle point method. For a much more thorough presentation, including a discussion of the topographical view of the integrand and the "hill-climbing" interpretation of the contour of integration, see [63].

12.2. Admissible functions

The saddle point method is a powerful and flexible tool, but in its full generality it is often cumbersome to apply. In many situations it is possible to apply general theorems derived using the saddle point method that give asymptotic approximations that are not the sharpest possible, but which allow one to avoid the drudgery of applying the method step by step. The general theorems that we present were proved by Hayman [204] and by Harris and Schoenfeld

[198]. We next describe the classes of functions to which these theorems apply, and then present the estimates one obtains for them. It is not always easy to verify that these definitions hold, but it is almost always easier to do this than to apply the saddle point method from scratch. It is worth mentioning, furthermore, that for many generating functions, there are conditions that guarantee that they satisfy the hypotheses of the Hayman and the Harris-Schoenfeld theorems. These conditions are discussed later in this section.

The definition below is stated somewhat differently than the original one in [204], but can be shown to be equivalent to it.

Definition 12.1. *A function*

$$f(z) \;=\; \sum_{n=0}^{\infty} f_n z^n \tag{12.17}$$

is admissible in the sense of Hayman (or H-admissible) if

i) $f(z)$ *is analytic in* $|z| < R$ *for some* $0 < R \le \infty$,

ii) $f(z)$ *is real for* z *real,* $|z| < R$,

iii) for $R_0 < r < R$,

$$\max_{|z|=r} |f(z)| \;=\; f(r)\ , \tag{12.18}$$

iv) for

$$a(r) \;=\; r\,\frac{f'(r)}{f(r)}\ , \tag{12.19}$$

$$b(r) \;=\; ra'(r) \;=\; r\,\frac{f'(r)}{f(r)} + r^2\frac{f''(r)}{f(r)} - r^2\left(\frac{f'(r)}{f(r)}\right)^2\ , \tag{12.20}$$

and for some function $\delta(r)$, *defined in the range* $R_0 < r < R$ *to satisfy* $0 < \delta(r) < \pi$, *the following three conditions hold:*

$$a)\quad f(re^{i\theta}) \sim f(r)\ \exp(i\theta a(r) - \theta^2 b(r)/2)$$
$$\text{as } r \to R \text{ uniformly for } |\theta| < \delta(r), \tag{12.21}$$

$$b)\qquad f(re^{i\theta}) \;=\; o(f(r)b(r)^{-1/2})$$
$$\text{as } r \to R \text{ uniformly for } |\theta| < \delta(r), \tag{12.22}$$

$$c)\qquad b(r) \to \infty \quad \text{as} \quad r \to R. \tag{12.23}$$

For H-admissible functions, Hayman [204] proved a basic result that gives the asymptotics of the coefficients.

Theorem 12.1. *If $f(z)$, defined by Eq. (12.17), is H-admissible in $|z| < R$, then*

$$f_n = (2\pi b(r))^{-1/2} f(r) r^{-n} \left\{ \exp\left(-\frac{(a(r)-n)^2}{b(r)} \right) + o(1) \right\} \tag{12.24}$$

as $r \to R$, with the $o(1)$ term uniform in n.

If we choose $r = r_n$ to be a solution to $a(r_n) = n$, then we obtain from Theorem 12.1 a simpler result. (The uniqueness of r_n follows from a result of Hayman [204] which shows that $a(r)$ is positive increasing in some range $R_1 < r < R$, $R_1 > R_0$.)

Corollary 12.1. *If $f(z)$, defined by Eq. (12.17), is H-admissible in $|z| < R$, then*

$$f_n \sim (2\pi b(r_n))^{-1/2} f(r_n) r_n^{-n} \quad as \quad n \to \infty \ , \tag{12.25}$$

where r_n is defined uniquely for large n by $a(r_n) = n$, $R_0 < r_n < R$.

Corollary 12.1 is adequate for most situations. The advantage of Theorem 12.1 is that it gives a uniform estimate over the approximate range $|a(r) - n| \lesssim b(r)^{1/2}$. (Note that the estimate (12.24) is vacuous for $|a(r) - n|\ b(r)^{-1/2} \to \infty$.) Theorem 12.1 shows that the $f_n r^n$ are approximately Gaussian in the central region.

There are many direct applications of the above results.

Example 12.4. *Stirling's formula.* Let $f(z) = \exp(z)$. Then $f(z)$ is H-admissible for $R = \infty$; conditions i)–iii) of Definition 12.1 are trivially satisfied, while $a(r) = r$, $b(r) = r$, so iv) also holds for $R_0 = 0$, $\delta(r) = r^{-1/3}$, say. Corollary 12.1 then shows that

$$f_n = \frac{1}{n!} \sim (2\pi n)^{-1/2} e^n n^{-n} \quad \text{as} \quad n \to \infty \ , \tag{12.26}$$

since $r_n = n$, which gives a weak form of Stirling's approximation to $n!$. ■

In many situations the conditions of H-admissibility are much harder to verify than for $f(z) = \exp(z)$, and even in that case there is a little work to be done to verify that condition iv) holds. However, many of the generating functions one encounters are built up from other, simpler generating functions, and Hayman [204] has shown that often the resulting functions are guaranteed to be H-admissible. We summarize some of Hayman's results in the following theorem.

Theorem 12.2. *Let $f(z)$ and $g(z)$ be H-admissible for $|z| < R \leq \infty$. Let $h(z)$ be analytic in $|z| < R$ and real for real z. Let $p(z)$ be a polynomial with real coefficients.*

i) If the coefficients a_n of the Taylor series of $\exp(p(z))$ are positive for all sufficiently large n, then $\exp(p(z)))$ is H-admissible in $|z| < \infty$.

ii) $\exp(f(z))$ and $f(z)g(z)$ are H-admissible in $|z| < R$.

iii) If, for some $\eta > 0$, and $R_1 < r < R$,

$$\max_{|z|=r} |h(z)| = O(f(r)^{1-\eta}) \ , \tag{12.27}$$

then $f(z) + h(z)$ is H-admissible in $|z| < R$. In particular, $f(z) + p(z)$ is H-admissible in $|z| < R$ and, if the leading coefficient of $p(z)$ is positive, $p(f(z))$ is H-admissible in $|z| < R$.

Example 12.5. *H-admissible functions.* a) By i) Theorem 12.2, $\exp(z)$ is H-admissible, so we immediately obtain the estimate (12.26), which yields Stirling's formula. b) Since $\exp(z)$ is H-admissible, part iii) of Theorem 12.2 shows that $\exp(z) - 1$ is H-admissible. c) Applying part ii) of Theorem 12.2, we next find that $\exp(\exp(z) - 1)$ is H-admissible, which yields the asymptotics of the Bell numbers. ■

Hayman's results give only first order approximations for the coefficients of H-admissible functions. In some circumstances it is desirable to obtain full asymptotic expansions. This is possible if we impose additional restrictions on the generating function. We next state some results of Harris and Schoenfeld [198].

Definition 12.2. *A function $f(z)$ defined by Eq. (12.17) is* HS-admissible *provided it is analytic in $|z| < R$, $0 < R \leq \infty$, is real for real x, and satisfies the following conditions:*

A) There is an R_0, $0 < R_0 < R$ and a function $d(r)$ defined for $r \in (R_0, R)$ such that

$$\begin{gathered} 0 < d(r) < 1 \ , \\ r\{1 + d(r)\} < R \ , \end{gathered} \tag{12.28}$$

and such that $f(z) \neq 0$ for $|z - r| < rd(r)$.

B) If we define, for $k \geq 1$,

$$A(z) = \frac{f'(z)}{f(z)}, \quad B_k(z) = \frac{z^k}{k!} A^{(k-1)}(z), \quad B(z) = \frac{z}{2} B_1(z) \ , \tag{12.29}$$

then we have

$$B(r) > 0 \quad \textit{for} \quad R_0 < r < R \quad \textit{and} \quad B_1(r) \to \infty \quad \textit{as} \quad r \to R \ .$$

C) For sufficiently large R_1 and n, there is a unique solution $r = u_n$ to

$$B_1(r) = n + 1, \quad R_1 < r < R \ . \tag{12.30}$$

Let

$$C_j(z,r) = \frac{-1}{B(r)} \left\{ B_{j+2}(z) + \frac{(-1)^j}{j+2} B_1(r) \right\} \ . \tag{12.31}$$

There exist nonnegative D_n, E_n, and n_0 such that for $n \geq n_0$,

$$|C_j(u_n, u_n)| \leq E_n D_n^j, \quad j = 1, 2, \dots \ . \tag{12.32}$$

D) As $n \to \infty$,

$$\begin{aligned} & B(u_n) d(u_n)^2 \to \infty \ , \\ & D_n E_n B(u_n) d(u_n)^3 \to 0 \ , \\ & D_n d(u_n) \to 0 \ . \end{aligned} \tag{12.33}$$

For *HS*-admissible functions, Harris and Schoenfeld obtain complete asymptotic expansions.

Theorem 12.3. *If $f(z)$, defined by (12.17), is* HS*-admissible, then for any $N \geq 0$,*

$$f_n = 2(\pi\beta_n)^{-1/2} f(u_n) u_n^{-n} \left\{ 1 + \sum_{k=1}^{N} F_k(n) \beta_n^{-k} + O(\phi_N(n;d)) \right\} \ \textit{as} \ n \to \infty \ , \tag{12.34}$$

where

$$\beta_n = B(u_n) \ , \tag{12.35}$$

$$F_k(n) = \frac{(-1)^k}{\sqrt{\pi}} \sum_{m=1}^{2k} \frac{\Gamma(m+k+\frac{1}{2})}{m!} \sum_{\substack{j_1+\cdots+j_m=2k \\ j_1,\dots,j_m \geq 1}} \gamma_{j_1}(n) \cdots \gamma_{j_m}(n) \ , \tag{12.36}$$

$$\gamma_j(n) = C_j(u_n, u_n) \ , \tag{12.37}$$

and

$$\phi_N(n;d) = \max\{\mu(u_n, d), E_n'(D_n E_n'' \beta_n^{-1/2})^{2N+2}\} \ ,$$

with

$$E'_n = \min(1, E_n), \quad E''_n = \max(1, E_n) \ , \tag{12.38}$$

$$\mu(r,d) = \max\left\{\lambda(r;d)B(r)^{1/2}, \ \frac{\exp(-B(r)d(r)^2)}{d(r)B(r)^{1/2}}\right\} \ , \tag{12.39}$$

where $\lambda(r;d)$ *is the maximum value of* $|f'(z)/f(z)|$ *for* z *on the oriented path* $Q(r)$ *consisting of the line segment from* $r + ird(r)$ *to* $(1 - d(r)^2)^{1/2} + ird(r)$ *and of the circular arc from the last point to* ir *to* $-r$.

The conditions for *HS*-admissibility are often hard to verify. However, there is a theorem [311] which guarantees that they do hold for a large class of interesting functions.

Theorem 12.4. *If* $g(z)$ *is* H*-admissible, then* $f(z) = \exp(g(z))$ *is* HS*-admissible. Furthermore, the error term* $\phi_N(n;d)$ *of Theorem 12.3 is then* $o(\beta_n^{-N})$ *as* $n \to \infty$ *for every fixed* $N \geq 0$.

Example 12.6. *Bell numbers and* HS*-admissibility.* Since $\exp(x) - 1$ is H-admissible, as we saw in Example 12.5, we find that $\exp(\exp(z) - 1)$ is HS-admissible, and Theorem 12.3 yields a complete asymptotic expansion of the Bell numbers. ■

Theorem 12.4 does not apply when $g(z)$ is a polynomial. As is pointed out by Schmutz [339], for $g(z) = z^4 - z^3 + z^2$ the function $f(z) = \exp(g(z))$ is *HS*-admissible, but Theorem 12.3 does not give an asymptotic expansion because the error term $\phi_N(n;d)$ is too large. Schmutz [339] has obtained necessary and sufficient conditions for Theorem 12.3 to give an asymptotic expansion for the coefficients of $f(z) = \exp(g(z))$ when $g(z)$ is a polynomial.

12.3. Other saddle point applications

Section 12.1 presented the basic saddle point method and discussed its range of applicability. Section 12.2 was devoted to results derived using this method that are general and yet can be applied in a cook-book style, without a deep understanding of the saddle point technique. Such a cook-book approach is satisfactory in many situations. However, often one encounters asymptotic estimation problems that are not covered by any of general results mentioned in Section 12.2, but can be solved using the saddle point method. This section mentions several such results of this type that illustrate the range of problems to which this method is applicable. Additional applications will be presented in Section 15, where other techniques are combined with the saddle point method.

Example 12.7. *Stirling numbers.* The Stirling numbers of the first kind, $s(n,k)$, satisfy (6.5) as well as [81]

$$\sum_{k=0}^{n} s(n,k)z^k = z(z-1)\cdots(z-n+1)\ . \tag{12.40}$$

Since $(-1)^{n+k}s(n,k) > 0$, (which is reflected in the behavior of the generating function (12.40), which grows faster along the negative real axis than along the positive one), we rewrite it as

$$\sum_{k=0}^{n}(-1)^{n+k}s(n,k)z^k = z(z+1)\cdots(z+n-1)\ . \tag{12.41}$$

The function on the right-hand side behaves like a good candidate for an application of the saddle point method. For details, see [295, 296]. ■

The estimates mentioned in Example 12.7 are far from best possible in either the size of the error term or (more important) in the range of validity. References for the best currently known results about Stirling numbers of both the first and second kind are given in [363]. Some of the results in the literature are not rigorous. For example, [363] presents elegant and uniform estimates based on an application of the saddle point method. They are likely to be correct, but the necessary rigorous error analysis has not been performed yet, although it seems that this should be doable. Other results, like those of [232] are obtained by methods that there does not seem to be any hope of making rigorous in the near future. Some of the results, though, such as the original ones of Moser and Wyman [295, 296], and the more recent one of Wilf [378], are fully proved.

The saddle point method can be used to obtain full asymptotic expansions. These expansions are usually in powers of $n^{-1/2}$ when estimating $[z^n]f(z)$, and they hardly ever converge, but are asymptotic expansions as defined by Poincaré (as in Eq. (2.2)). The usual forms of the saddle point method are incapable of providing expansions similar to the Hardy-Ramanujan-Rademacher convergent series for the partition function $p(n)$ (Eq. (3.1)). However, the saddle point method can be applied to estimate $p(n)$. There are technical difficulties, since the generating function

$$f(z) = \sum_{n=0}^{\infty} p(n)z^n = \prod_{k=1}^{\infty}(1-z^k)^{-1} \tag{12.42}$$

has a large singularity at $z = 1$, but in addition has singularities at all other roots of unity. The contribution of the integral for z away from 1 can be crudely estimated to be $O(n^{-1}\exp(Cn^{1/2}/2))$ (the last term in Eq. (1.5)). A simple estimate of the integral near $z = 1$ yields the asymptotic expansion of Eq. (1.6). A more careful treatment of the integral, but

one that follows the conventional saddle point technique, replaces the $1 + O(n^{-1/2})$ term in Eq. (1.6) by an asymptotic (in the sense of Poincare, so nonconvergent) series $\sum c_k n^{-k/2}$. To obtain Eq. (1.5), one needs to choose the contour of integration near $z = 1$ carefully and use precise estimates of $f(z)$ near $z = 1$.

De Bruijn [63] also discusses applications of the saddle point method when the saddle point is not on the real axis, and especially when there are several saddle points that contribute comparable amounts. This usually occurs when there are oscillations in the coefficients. When the oscillations are irregular, the tricks mentioned in Section 10 of changing variables do not work, and the contributions of the multiple saddle points have to be evaluated.

Example 12.8. *Oscillating sequence.* Consider the sequence a_n of Examples 9.4 and 10.1. As is shown in Example 9.4, its ordinary generating function is given by (9.39). It has an essential singularity at $z = 1$, but is analytic every place else. This function is not covered by our earlier discussion. For example, its maximal value is in general not taken on the positive real axis. It can be shown that the Cauchy integral has two saddle points, at approximately $z = 1-(2n)^{-1} \pm in^{-1/2}(1-(4n)^{-1})^{1/2}$. Evaluating $[z^n]f(z)$ by using Cauchy's theorem with the contour chosen to pass through the two points in the correct way yields the estimate (9.38). ■

In applying the saddle point method, a general principle is that multiplying a generating function $f(z)$ with dominant singularity at R by another function $g(z)$ which is analytic in $|z| < R$ and has much lower growth rate near $z = R$ yields a function $f(z)g(z)$ whose saddle point is close to that of $f(z)$. Usually one can obtain a relation of the form

$$[z^n](f(z)g(z)) \sim g(r_0)([z^n]f(z)) \ , \tag{12.43}$$

where r_0 is the saddle point for $f(z)$. This principle (which is related to the one behind Theorem 7.1) is useful, but has to be applied with caution, and proofs have to be provided for each case. For fuller exposition of this principle and general results, see [157]. The advantage of this approach is that often $f(z)$ is easy to manipulate, so the determination of a saddle point for it is easy, whereas multiplying it by $g(z)$ produces a messy function, and the exact saddle point for $f(z)g(z)$ is difficult to determine.

Example 12.9. *Boolean lattice of subsets of* $\{1, \ldots, n\}$. The number a_n of Boolean sublattices of the Boolean lattice of subsets of $\{1, \ldots, n\}$ has the exponential generating function [162]

$$A(z) = \sum_{n=0}^{\infty} a_n \frac{z^n}{n!} = \exp(2z + \exp(z) - 1) \ . \tag{12.44}$$

We can write $A(z) = \exp(2z)B(z)$, where $B(z)$ is the exponential generating function for the Bell numbers (Example 12.2). Since $B(z)$ grows much faster than $\exp(2z)$, it is easy to show that (12.43) applies, and so

$$a_n \sim \exp(2r_0)B_n \quad \text{as} \quad n \to \infty \ , \tag{12.45}$$

where r_0 is the saddle point for $B(z)$. Using the approximation (12.12) of Example 12.2, we find that

$$a_n \sim (n/\log n)^2 B_n \quad \text{as} \quad n \to \infty \ . \tag{12.46}$$

■

The insensitivity of the saddle point approximation to slight perturbations is reflected in slightly different definitions of a saddle point that are used. The saddle point approximation (12.9) for $[z^n]f(z)$ is stated in terms of r_0, the point that minimizes $f(r)r^{-n}$. The discussion of the saddle point emphasized minimization of the peak value of the integrand in Cauchy's formula, which is the same as minimizing $f(r)r^{-n-1}$, since the contour integral (10.6) involves $f(z)z^{-n-1}$. Some sources call the point minimizing $f(r)r^{-n-1}$ the saddle point. It is not important which definition is adopted. The asymptotic series coefficients look slightly differently in the two cases, but the final asymptotic series, when expressed in terms of n, are the same. The reason for slightly preferring the definition that minimizes $f(r)r^{-n}$ is that when the change of variable $z = r\exp(i\theta)$ is made in Cauchy's integral, there is no linear term in θ, and the integrand involves $\exp(-cn\theta^2 + O(|\theta|^3))$. If we minimized $f(r)r^{-n-1}$, we would have to deal with $\exp(-c'i\theta - c''n\theta^2 + O(|\theta|^3))$, which is not much more difficult to handle but is less elegant.

The same principle can be applied when the exact saddle point is hard to determine, and it is awkward to work with an implicit definition of this point. When that happens, there is often a point near the saddle point that is easy to handle, and for which the saddle point approximation holds. We refer to [157] for examples and discussion of this phenomenon.

12.4. The circle method and other techniques

As we mentioned in Section 12.3, the saddle point method is a powerful method that estimates the contribution of the neighborhood of only a single point, or at most a few points. The convergent series of Eq. (1.3) for the partition function $p(n)$ (as well as the earlier non-convergent but asymptotic and very accurate expansion of Hardy and Ramanujan) is obtained

by evaluating the contribution of the other singularities of $f(z)$ to the integral. The m-th term in Eq. (1.3) comes from the primitive m-th roots of unity. To obtain this expansion one needs to use a special contour of integration and detailed knowledge of the behavior of $f(z)$. The details of this technique, called the circle method, can be found in [13, 23].

Convergent series can be obtained from the circle method only when the generating function is of a special form. For results and references, see [8, 10].

Nonconvergent but accurate asymptotic expansions can be derived from the circle method in a much wider variety of applications. It is especially useful when there is no single dominant singularity. For the partition function $p(n)$, all the singularities away from $z = 1$ contribute little, and it is $z = 1$ that creates the dominant term and yields Eq. (1.6). For other functions this is often false. For example, when dealing with additive problems of Waring's type, where one studies $N_{k,m}(n)$, the number of representations of a nonnegative integer n as

$$n = \sum_{j=1}^{m} x_j^k, \quad x_j \in \mathbb{Z}^+ \cup \{0\} \quad \text{for all} \quad j \ , \tag{12.47}$$

the natural generating function to study is

$$\sum_{n=0}^{\infty} N_{k,m}(n) z^n = g(z)^m \ , \tag{12.48}$$

where

$$g(z) = \sum_{h=0}^{\infty} z^{h^k} \ . \tag{12.49}$$

The function $g(z)$ has a natural boundary at $|z| = 1$, but it again grows fastest as z approaches a root of unity from within $|z| < 1$, so it is natural to speak of $g(z)$ having singularities at the roots of unity. The singularity at $z = 1$ is still the largest, but not by much, as other roots of unity contribute comparable amounts, with the contribution of other roots of unity ζ diminishing as the order of ζ increases. All the contributions can be estimated, and one can obtain solutions to Waring's problem (which was to show that for every k, there is an integer m such that $N_{k,m}(n) > 0$ for all n) and other additive problems. For details of this method see [23]. We mention here that for technical reasons, one normally works with generating functions of the form $G_n(z)^m$, where

$$G_n(z) = \sum_{h=0}^{\lfloor n^{1/k} \rfloor} z^{h^k} \ , \tag{12.50}$$

(so that the generating function depends on n), and analyzes them for $|z| = 1$ (since they are now polynomials), but the basic explanation above of why this process works still applies.

13. Multivariate generating functions

A major difficulty in estimating the coefficients of multivariate generating functions is that the geometry of the problem is far more difficult. It is harder to see what are the critical regions where the behavior of the function determines the asymptotics of the coefficients, and those regions are more complicated. Singularities and zeros are no longer isolated, as in the one-dimensional case, but instead form $(k-1)$-dimensional manifolds in k variables. Even rational multivariate functions are not easy to deal with.

One basic tool in one-dimensional complex analysis is the residue theorem, which allows one to move a contour of integration past a pole of the integrand. (We derived a form of the residue theorem in Section 10, in the discussion of poles of generating functions.) There is an impressive generalization by Leray [4, 250] of this theory to several dimensions. Unfortunately, it is complicated, and with few exceptions (such as that of [252], see also [49]) so far it has not been applied successfully to enumeration problems. On the other hand, there are some much simpler tools that can frequently be used to good effect.

An important tool in asymptotics of multivariate generating functions is the multidimensional saddle point method.

Example 13.1. *Alternating sums of powers of binomial coefficients.* Consider

$$S(s,n) = \sum_{k=0}^{2n} (-1)^{k+n} \binom{2n}{k}^s , \tag{13.1}$$

where s and n are positive integers. It has been known for a long time that $S(1,n) = 0$, $S(2,n) = (2n)!(n!)^{-2}$, $S(3,n) = (3n)!(n!)^{-3}$. However, no formula of this type has been known for $s > 3$. De Bruijn (see Chapter 4 of [63]) showed that $S(s,n)$ for integer $s > 3$ cannot be expressed as a ratio of products of factorials. Although his proof is not presented as an application of the multidimensional saddle point method, it is easy to translate it into those terms. $S(s,n)$ is easily seen to equal the constant term in

$$F(z_1, \ldots, z_{s-1}) = (-1)^n (1+z_1)^{2n} \ldots (1+z_{s-1})^{2n} (1 - (z_1 \ldots z_{s-1})^{-1})^{2n} , \tag{13.2}$$

and so

$$S(s,n) = (2\pi i)^{-s+1} \int \cdots \int F(z_1, \ldots, z_{s-1}) z_1^{-1} \ldots z_{s-1}^{-1} dz_1 \ldots dz_{s-1} , \tag{13.3}$$

where the integral is taken with each z_j traversing a circle, say. De Bruijn's proof in effect shows that for s fixed and $n \to \infty$, there are two saddle points at $z_1 = \cdots = z_{s-1} = \exp(2i\alpha)$,

with $\alpha = \pm(2s)^{-1}$, and this leads to the estimate

$$S(s,n) \sim \left\{2\cos\left(\frac{\pi}{2s}\right)\right\}^{2ns+s-1} 2^{2-s}(\pi n)^{(1-s)/2} s^{-1/2} \quad \text{as} \quad n \to \infty \ , \tag{13.4}$$

valid for any fixed integer $s \geq 2$. Since $\cos(\pi(2s)^{-1})$ is algebraic but irrational for $s \geq 4$, the asymptotic estimate (13.4) shows that $S(s,n)$ cannot be expressed as a ratio of finite products of $(a_j n)!$ for any fixed finite set of integers a_j.

In Chapter 6 of [63], de Bruijn derives the asymptotics of $S(s,n)$ as $n \to \infty$ for general real s. The approach sketched above no longer applies, and de Bruijn uses the integral representation

$$S(s,n) = \int_C \left(\frac{\Gamma(2n+1)}{\Gamma(n+z+1)\Gamma(n-z+1)}\right)^s \frac{dz}{2i\sin\pi z} \ ,$$

where C is a simple closed curve that contains the points $-n, -n+1, \ldots, -1, 0, 1, \ldots, n$ in its interior and has no other integer points on the real axis in its closure. A complicated combination of analytic techniques, including the one-dimensional saddle point method, then leads to the final asymptotic estimate of $S(s,n)$. ■

The multidimensional saddle point method works best when applied to large singularities. Just as for the basic one-dimensional method, it does not work when applied to small singularities, such as those of rational functions. Fortunately, there is a trick that often succeeds in converting a small singularity in n dimensions into a large one in $n-1$ dimensions. The main idea is to expand the generating function with respect to one of the variables through partial fraction expansions or other methods. It is hard to write down a general theorem, but the next example illustrates this technique.

Example 13.2. *Alignments of k sequences.* Let $f(k,n)$ denote the number of $k \times m$ matrices of 0's and 1's such that each column sum is ≥ 1 and each row sum is exactly n. (The number of columns, m, can vary, although obviously $k \leq m \leq kn$.) We consider k fixed, $n \to \infty$ [178]. If we let $N(r_1, \ldots, r_k)$ denote the number of $0,1$ matrices with k rows, no columns of all 0's, and row sums $r_1, \ldots, r_k$, then it is easy to see [178] that

$$F(z_1, \ldots, z_k) = \sum_{r_1, \ldots, r_k \geq 0} N(r_1, \ldots, r_k) z_1^{r_1} \cdots z_k^{r_k} = \left(2 - \prod_{j=1}^{k}(1+z_j)\right)^{-1} \ . \tag{13.5}$$

We have $f(k,n) = N(n, \ldots, n)$, and so we need the diagonal terms of $F = F(z_1, \ldots, z_k)$. The function F is rational, so its singularity is small. Moreover, the singularities of F are difficult

to visualize. However, in any single variable F is simple. We take advantage of this feature. Let

$$A(z) = \prod_{j=1}^{k-1}(1+z_j)\ , \tag{13.6}$$

where z stands for $(z_1,\dots,z_{k-1}) \in \mathbb{C}^{k-1}$, and expand

$$\left(2 - \prod_{j=1}^{k}(1+z_j)\right)^{-1} = (2 - A(z)(1+z_k))^{-1} = \sum_{m=0}^{\infty} \frac{A(z)^m z_k^m}{(2-A(z))^{m+1}}\ . \tag{13.7}$$

Therefore

$$N(r_1,\dots,r_{k-1},m) = \frac{1}{(2\pi i)^{k-1}} \int \cdots \int \frac{A(z)^m}{(2-A(z))^{m+1}} \frac{dz_1}{z_1^{r_1+1}} \cdots \frac{dz_{k-1}}{z_{k-1}^{r_{k-1}}}\ . \tag{13.8}$$

The function whose coefficients we are trying to extract is now $A(z)^m/(2-A(z))^{m+1}$, which is still rational. However, the interesting case for us is $m \to \infty$, which transforms the singularity into a large one. We are interested in the case $r_1 = r_2 = \cdots = r_{k-1} = r = n$. Then the integral in (13.8) can be shown to have a saddle point at $z_j = \rho$, $1 \le j \le k-1$, where $\rho = 2^{1/k} - 1$, and one obtains the estimate [178]

$$f(k,n) = r^n n^{-(k-1)/2}\{(\rho\pi^{(k-1)/2}k^{1/2})^{-1}2^{(k^2-1)/(2k)} + O(n^{-1/2})\} \ \text{ as } \ n \to \infty\ . \qquad \blacksquare \tag{13.9}$$

The examples above of applications of the multidimensional saddle point method all dealt with problems in a fixed dimension as various other parameters increase. A much more challenging problem is to apply this method when the dimension varies. A noteworthy case where this has been done successfully is the asymptotic enumeration of graphs with a given degree sequence by McKay and Wormald [279].

Example 13.3. *Simple labeled graphs of high degree.* Let $G(n;d_1,\dots,d_n)$ be the number of labeled simple graphs on n vertices with degree sequence $d_1,d_2,\dots,d_n$. Then $G(n;d_1,\dots,d_n)$ is the coefficient of $z_1^{d_1}z_2^{d_2}\cdots z_n^{d_n}$ in

$$F = \prod_{\substack{j,k=1\\ j<k}}^{n} (1+z_jz_k)\ , \tag{13.10}$$

and so by Cauchy's theorem

$$G(n;d_1,\dots,d_N) = (2\pi i)^{-n} \int \cdots \int F z_1^{-d_1-1}\cdots z_n^{-d_n-1} dz_1\cdots dz_n\ , \tag{13.11}$$

where each integral is on a circle centered at the origin. Let all the radii be equal to some $r > 0$. The integrand takes on its maximum absolute value on the product of these circles at precisely the two points $z_1 = z_2 = \cdots = z_n = r$ and $z_1 = z_2 = \cdots = z_n = -r$. If $d_1 = d_2 = \cdots = d_n$, so that we consider only regular graphs, McKay and Wormald [279] show that for an appropriate choice of the radius r, these two points are saddle points of the integrand, and succeed through careful analysis in proving that if dn is even, and $\min(d, n-d-1) > cn(\log n)^{-1}$ for some $c > 2/3$, then

$$G(n, d, d, \ldots, d) = 2^{1/2}(2\pi n\lambda^{d+1}(1-\lambda)^{n-d})^{-n/2} \exp\left(\frac{-1+10\lambda-10\lambda^2}{12\lambda(1-\lambda)} + O(n^{-\zeta})\right) \quad (13.12)$$

as $n \to \infty$ for any $\zeta < \min(1/4, 1/2 - 1/(3c))$, where $\lambda = d/(n-1)$.

McKay and Wormald [279] also succeed in estimating the number of irregular graphs, provided that all the degrees d_j are close to a fixed d that satisfies conditions similar to those above. The proof is more challenging because different radii are used for different variables and the result is complicated to state. ■.

The McKay-Wormald estimate of Example 13.3 is a true tour de force. The problem is that the number of variables is n and so grows rapidly, whereas the integrand grows only like $\exp(cn^2)$ at its peak. More precisely, after transformations that remove obvious symmetries are applied the integrand near the saddle point drops off like $\exp(-n\sum\theta_j^2)$. This is just barely to allow the saddle point method to work, and the symmetries in the problem are exploited to push the estimates through. This approach can be applied to other problems (cf. [278]), but it is hard to do. On the other hand, when the number of variables grows more slowly, multidimensional saddle point contributions can be estimated without much trouble.

So far this section has been devoted primarily to multivariate functions with large singularities. However, there is also an extensive literature on small singularities. The main thread connecting most of these works is that of central and local limit theorems. Bender [32] initiated this development in the setting of two-variable problems. We present some of his results, since they are simpler than the later and more general ones that will be mentioned at the end of this section.

Consider a double sequence of numbers $a_{n,k} \geq 0$. (Usually the $a_{n,k}$ are $\neq 0$ only for $0 \leq k \leq n$.) We will assume that

$$A_n = \sum_k a_{n,k} < \infty \quad (13.13)$$

for all n, and define the normalized double sequence

$$p_n(k) = a_{n,k}/A_n \ . \tag{13.14}$$

We will say that $a_{n,k}$ satisfies a central limit theorem if there exist functions σ_n and μ_n such that

$$\lim_{n\to\infty} \sup_x \left| \sum_{k \le \sigma_n x + \mu_n} p_n(k) - (2\pi)^{-1/2} \int_{-\infty}^{x} \exp(-t^2/2) dt \right| = 0 \ . \tag{13.15}$$

Equivalently, $p_n(k)$ is asymptotically normal with mean μ_n and variance σ_n^2.

Theorem 13.1. [32]. *Let $a_{n,k} \ge 0$, and set*

$$f(z,w) = \sum_{n,k\ge 0} a_{n,k} z^n w^k \ . \tag{13.16}$$

Suppose that there are (i) a function $g(s)$ that is continuous and $\neq 0$ near $s = 0$, (ii) a function $r(s)$ with bounded third derivative near $s = 0$, (iii) an integer $m \ge 0$, and (iv) $\epsilon, \delta > 0$ such that

$$\left(1 - \frac{z}{r(s)}\right)^m f(z, e^s) - \frac{g(z)}{1 - z/r(s)} \tag{13.17}$$

is analytic and bounded for

$$|z| < \epsilon, \quad |z| < |r(0)| + \delta \ . \tag{13.18}$$

Let

$$\mu = -r'(0)/r(0), \quad \sigma^2 = \mu^2 - r''(0)/r(0) \ . \tag{13.19}$$

If $\sigma \neq 0$, then (13.15) holds with $\mu_n = n\mu$ and $\sigma_n^2 = n\sigma^2$.

A central limit theorem is useful, but it only gives information about the cumulative sums of the $a_{n,k}$. It is much better to have estimates for the individual $a_{n,k}$. We say that $p_n(k)$ (and $a_{n,k}$) satisfy a local limit theorem if

$$\lim_{n\to\infty} \sup_x \left| \sigma_n p_n(\lfloor \sigma_n x + \mu_n \rfloor) - (2\pi)^{-1/2} \exp(-x^2/2) \right| = 0 \ . \tag{13.20}$$

In general, we cannot derive (13.20) from (13.15) without some additional conditions on the $a_{n,k}$, such as unimodality (see [32]). The other approach one can take is to derive (13.20) from conditions on the generating function $f(z,w)$.

Theorem 13.2. [32]. *Suppose that* $a_{n.k} \geq 0$, *and let* $f(z,w)$ *be defined by (13.16). Let* $-\infty < a < b < \infty$. *Define*

$$R(\epsilon) = \{z : a \leq Re(z) \leq b,\ |Im(z)| \leq \epsilon\}\ . \tag{13.21}$$

Suppose there exist $\epsilon > 0$, $\delta > 0$, *an integer* $m \geq 0$, *and function* $g(s)$ *and* $r(s)$ *such that*

(i) $g(s)$ *is continuous and* $\neq 0$ *for* $s \in R(\epsilon)$,

(ii) $r(s) \neq 0$ *and has a bounded third derivative for* $s \in R(\epsilon)$,

(iii) for $s \in R(\epsilon)$ *and* $|z| \leq |r(s)|(1+\delta)$, *the function defined by (13.17) is analytic and bounded,*

(iv)

$$\left(\frac{r'(\alpha)}{r(\alpha)}\right)^2 \neq \frac{r''(\alpha)}{r(\alpha)} \quad for \quad a \leq \alpha \leq b\ , \tag{13.22}$$

(v) $f(z, e^s)$ *is analytic and bounded for*

$$|z| \leq |r(Re(s))|(1+\delta) \quad and \quad s \leq |Im(s)| \leq \pi\ .$$

Then

$$a_{n,k} \sim \frac{n^m e^{-\alpha k} g(\alpha)}{m! r(\alpha)^m \sigma_\alpha (2\pi)^{1/2}} \quad as \quad n \to \infty \tag{13.23}$$

uniformly for $a \leq \alpha \leq b$, *where*

$$\frac{k}{n} = -\frac{r'(\alpha)}{r(\alpha)}\ , \tag{13.24}$$

$$\sigma_\alpha^2 = \left(\frac{k}{n}\right)^2 - \frac{r''(\alpha)}{r(\alpha)}\ . \tag{13.25}$$

There have been many further developments of central and local limit theorems for asymptotic enumeration since Bender's original work [32]. Currently the most powerful and general results are those of Gao and Richmond [155]. They apply to general multivariate problems, not only two-variable ones. Other papers that deal with central and local limit theorems or other multivariate problems with small singularities are [38, 42, 65, 96, 142, 143, 183, 227].

14. Mellin and other integral transforms

When the best generating function that one can obtain is an infinite sum, integral transforms can sometimes help. There is a large variety of integral transforms, such as those of

Fourier and Laplace. The one that is most commonly used in asymptotic enumeration and analysis of algorithms is the Mellin transform, and it is the only one we will discuss extensively below. The other transforms do occur, though. For example, if $f(x) = \sum a_n x^n/n!$ is an exponential generating function of the sequence a_n, then the ordinary generating function of a_n can be derived from it using the Laplace transform

$$\begin{aligned} \int_0^\infty f(xy)\exp(-x)dx &= \sum_n a_n y^n (n!)^{-1} \int_0^\infty x^n \exp(-x)dx \\ &= \sum_n a_n y^n \ . \end{aligned} \tag{14.1}$$

(This assumes that the a_n are small enough to assure the integrals above converge and the interchange of summation and integration is valid.) Related integral transforms can be used to transform generating functions into other forms. For example, to transform an ordinary generating function $F(u) = \sum a_n u^n$ into an exponential one, we can use

$$\frac{1}{2\pi i}\int_{|u|=r} F(u)\exp(w/u)du \ . \tag{14.2}$$

The basic references for asymptotics of integral transforms are [89, 95, 299, 347]. This section will only highlight some of the main properties of Mellin transforms and illustrate how they are used. For a more detailed survey, especially to analysis of algorithms, see [137].

Let $f(t)$ be a measurable function defined for real $t \geq 0$. The *Mellin transform* $f^*(z)$ of $f(t)$ is a function of the complex variable z defined by

$$f^*(z) = \int_0^\infty f(t)t^{z-1}dt \ . \tag{14.3}$$

If $f(t) = O(t^\alpha)$ as $t \to 0^+$ and $f(t) = O(t^\beta)$ as $t \to \infty$, then the integral in (14.3) converges and defines $f^*(z)$ to be an analytic function inside the "fundamental domain" $-\alpha < \mathrm{Re}(z) < -\beta$. As an example, for $f(t) = \exp(-t)$, we have $f^*(z) = \Gamma(z)$ and $\alpha = 0$, $\beta = -\infty$. There is an inversion formula for Mellin transforms which states that

$$f(t) = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} f^*(z)t^{-z}dz \ , \tag{14.4}$$

and the integral is over the vertical line with $\mathrm{Re}(z) = c$. The inversion formula (14.4) is valid for $-\alpha < c < -\beta$, but much of its strength in applications comes from the ability to shift the contour of integration into wider domains to which $f^*(z)$ can be analytically continued.

The advantage of the Mellin transform is due largely to a simple property, namely that if $g(t) = af(bx)$ for b real, $b > 0$, then

$$g^*(z) = ab^{-z}f^*(z) \ . \tag{14.5}$$

This readily extends to show that if

$$F(t) = \sum_k \lambda_k f(\eta_k t) \tag{14.6}$$

(where the λ_k and $\eta_k > 0$ are such that the sum converges and $F(t)$ is well behaved), then

$$F^*(z) = \left(\sum_k \lambda_k \eta_k^{-z} \right) f^*(z) \ . \tag{14.7}$$

In particular, if

$$F(t) = \sum_{k=1}^{\infty} f(kt) \ , \tag{14.8}$$

then

$$F^*(z) = \left(\sum_{k=1}^{\infty} k^{-z} \right) f^*(z) = \zeta(z) f^*(z) \ , \tag{14.9}$$

where $\zeta(z)$ is the Riemann zeta function.

Example 14.1. *Runs of heads in coin tosses.* What is R_n, the expected length of the longest run of heads in n tosses of a fair coin? Let $p(n,k)$ be the probability that there is no run of k heads in a coin tosses. Then

$$R_n = \sum_{k=1}^{n} k(p(n,k+1) - p(n,k)) \ . \tag{14.10}$$

We now apply the estimates of Example 9.2. To determine $p(n,k)$, we take $A = 00\cdots0$, and then $C_A(z) = z^{k-1} + z^{k-2} + \cdots + z + 1$, so $C_A(1/2) = 1 - 2^{-k}$. Hence (9.19) shows easily that in the important ranges where k is of order $\log n$, we have

$$p(n,k) \cong \exp(-n2^{-k}) \ , \tag{14.11}$$

and there R_n is approximated well by

$$r(n) = \sum_{k=0}^{\infty} k(\exp(-n2^{-k-1}) - \exp(-n2^{-k})) \ . \tag{14.12}$$

The function $r(t)$ is of the form (14.6) with

$$\lambda_k = k, \quad \eta_k = 2^{-k}, \quad f(t) = \exp(-t/2) - \exp(-t) \ , \tag{14.13}$$

is easily seen to be well behaved, and so for $-1 < \mathrm{Re}(z) < 0$,

$$r^*(z) = \left(\sum_{k=0}^{\infty} k2^{kz} \right) f^*(z) = 2^z(1-2^z)^{-2} f^*(z) \ . \tag{14.14}$$

Next, to determine $f^*(z)$, we note that for $\mathrm{Re}(z) > 0$ we have

$$\begin{aligned} f^*(z) &= \int_0^\infty f(t)t^{z-1}dt = \int_0^\infty e^{-t/2}t^{z-1}dt - \int_0^\infty e^{-t}t^{z-1}dt \\ &= (2^z - 1)\Gamma(z) \ . \end{aligned} \tag{14.15}$$

By analytic continuation this relation holds for $-1 < \mathrm{Re}(z)$, and we find that for $-1 < \mathrm{Re}(z) < 0$,

$$r^*(z) = 2^z(2^z - 1)^{-1}\Gamma(z) \ . \tag{14.16}$$

We now apply the inversion formula to obtain

$$r(t) = \frac{1}{2\pi i}\int_{-1/2-i\infty}^{-1/2+i\infty} 2^z(2^z - 1)^{-1}\Gamma(z)t^{-z}dz \ . \tag{14.17}$$

The integrand is a meromorphic function in the whole complex plane that drops off rapidly on any vertical line. We move the contour of integration to the line $\mathrm{Re}(z) = 1$. The new integral is $O(t^{-1})$, and the residues at the poles (all on $\mathrm{Re}(z) = 0$) will give the main contribution to $r(t)$. There are first order poles at $z = 2\pi i m \log 2$ for $m \in \mathbb{Z} \setminus \{0\}$ coming from $2^z = 1$, and a single second order pole at $z = 0$, since $\Gamma(z)$ has a first order pole there as well. A short computation of the residues gives

$$r(t) = \log_2 t - \sum_{h=-\infty}^{\infty} (\log 2)^{-1}\Gamma(-2\pi i h(\log 2)^{-1}) \exp(2\pi i h \log_2 t) + O(t^{-1}) \ . \tag{14.18}$$

■

There are other ways to obtain the same expansion (14.18) for $r(t)$ (cf. [181]). The periodic oscillating component in $r(t)$ is common in problems involving recurrences over powers of 2. This happens, for example, in studies of register allocation and digital trees [136, 138, 141]. The periodic function is almost always the same as the one in Eq. (14.18), even when the combinatorics of the problem varies. Technically this is easy to explain, because of the closely related recurrences leading to similar Mellin transforms for the generating functions.

Mellin transforms are useful in dealing with problems that combine combinatorial and arithmetic aspects. For example, if $S(n)$ denotes the total number of 1's in the binary representations of $1, 2, \ldots, n-1$, then it was shown by Delange that

$$S(n) = \frac{1}{2}n\log_2 n + nu(\log_2 n) + o(n) \quad \text{as} \quad n \to \infty \ , \tag{14.19}$$

where $u(x)$ is a continuous, nowhere differentiable function that satisfies $u(x) = u(x+1)$. The Fourier coefficients of $u(x)$ are known explicitly. Perhaps the best way to obtain these results is by using Mellin transforms. See [129, 353] for further information and references.

Mellin transforms are often combined with other techniques. For example, sums of the form $s_n = \sum a_k \binom{n}{k}$ with oscillating a_k lead to generating functions

$$s(z) = \sum_k a_k w(z)^k \ . \tag{14.20}$$

The asymptotic behavior of $s(z)$ near its dominant singularity can sometimes be determined by applying Mellin transforms. For a detailed explanation of the approach, see [137]. Examples of the application of this technique can be found in [13, 280].

15. Functional equations, recurrences, and combinations of methods

Most asymptotic enumeration results are obtained from combinations of techniques presented in the previous sections. However, it is only rarely that the basic asymptotic techniques can be applied directly. This section describes a variety of methods and results that are not easy to categorize. They use combinations of methods that have been presented before, and sometimes develop them further. In most of the examples that will be presented, some relations for generating functions are available, but no simple closed-form formulas, and the problem is to deduce where the singularities lie and how the generating functions behave in their neighborhoods. Once that task is done, previous methods can be applied to obtain asymptotics of the coefficients.

15.1. Implicit functions, graphical enumeration, and related topics

Example 15.1. *Rooted unlabeled trees.* We sketch a proof that T_n, the number of rooted unlabeled trees with n vertices, satisfies the asymptotic relation (1.9). The functional equation (1.8) holds with $T(z)$ regarded as a formal power series. The first step is to show that $T(z)$ is analytic in a neighborhood of 0. This can be done by working exclusively with Eq. (1.8). (There is an argument of this type in Section 9.5 of [188].) Another way to prove analyticity of $T(z)$ is to use combinatorics to obtain crude upper bounds for T_n. We use a combination of these approaches. If a tree with $n \geq 2$ vertices has at least two subtrees at the root, we can decompose it into two trees, the first consisting of one subtree at the root, the other of the root and the remaining subtrees. This shows that

$$T_n \leq T_{n-1} + \sum_{k=1}^{n-1} T_k T_{n-k} \ , \qquad n \geq 2 \ . \tag{15.1}$$

Therefore, if we define $a_1 = 1$, and

$$a_n = a_{n-1} + \sum_{k=1}^{n-1} a_k a_{n-k} \ , \qquad n \geq 2 \ , \tag{15.2}$$

then we have $T_n \leq a_n$. Now if

$$A(z) = \sum_{n=1}^{\infty} a_n z^n \ ,$$

then the defining relation (15.2) yields the functional equation

$$A(z) - z = zA(z) + A(z)^2 \ , \tag{15.3}$$

so that

$$A(z) = (1 - z - (1 - 6z + z^2)^{1/2})/2 \ . \tag{15.4}$$

Since $A(z)$ is analytic in $|z| < 3 - 2\sqrt{2} = 0.17157\ldots$, we have

$$0 \leq T_n \leq a_n = O(6^n) \ . \tag{15.5}$$

It will also be convenient to have an exponential lower bound for T_n. Let b_n be the number of rooted unlabeled trees in which every internal vertex has ≤ 2 subtrees. Then $b_1 = 1$, $b_2 = 1$, and

$$b_n \geq \sum_{k=1}^{\lfloor (n-1)/2 \rfloor} b_k b_{n-k-1} \qquad \text{for} \qquad n \geq 3 \ . \tag{15.6}$$

We use this to show that $b_n \geq (6/5)^n$ for $n \geq 7$. Direct computation establishes this lower bound for $7 \leq n \leq 14$, and for $n \geq 15$ we use induction and $b_n \geq b_k b_{n-k-1}$ with $k = \lfloor (n-1)/2 \rfloor$.

Since $T_n \geq b_n \geq (6/5)^n$, $T(z)$ converges only in $|z| < r$ for some r with $r < 1$. Since $T(0) = 0$, $|T(z)| \leq C_\delta |z|$ in $|z| \leq r - \delta$ for every $\delta > 0$, and therefore

$$u(z) = \sum_{k=2}^{\infty} T(z^k)/k \tag{15.7}$$

is analytic in $|z| < r^{1/2}$, and in particular at $z = r$. Therefore, although we know little about r and $u(z)$, we see that $T(z)$ satisfies $G(z, T(z)) = T(z)$, where

$$G(z, w) = z \exp(w + u(z)) \tag{15.8}$$

is analytic in z and w for all w and for $|z| < r^{1/2}$.

We will apply Theorem 10.6. First, though, we need to establish additional properties of $T(z)$. We have

$$T(z) \exp(-T(z)) = z \exp(u(z)) \to r \exp(u(r)) \quad \text{as} \quad z \to r^- \ , \tag{15.9}$$

and $0 < r\exp(u(r)) < \infty$. Since $T(z)$ is positive and increasing for $0 < z < r$, $T(r)$, the limit of $T(z)$ as $z \to r^-$ must exist and be finite.

We next show that $T(r) = 1$. We have

$$\frac{\partial}{\partial w} G(z,w) = G(z,w) \ . \tag{15.10}$$

We know that $G(z,T(z)) = T(z)$ for $|z| < r$, and in particular for some z arbitrarily close to r. If $T(r) \neq 1$, then by (15.10)

$$\frac{\partial}{\partial w}(G(z,w) - w)\Big|_{w=T(z)} \neq 0 \tag{15.11}$$

in a neighborhood of $z = r$, and therefore $T(z)$ could be continued analytically to a neighborhood of $z = r$. This is impossible, since r is the radius of convergence of $T(z)$, and $T_n \geq 0$ implies by Theorem 10.3 that $T(z)$ has a singularity at $z = r$. Therefore we must have $T(r) = 1$, and $G_w(r,T(r)) = 1$.

We have now shown that conditions (i) and (ii) of Theorem 10.6 hold with the r of that theorem the same as the r we have defined and $s = T(r) = 1$, $\delta = r^{1/2} - r$. Condition (iii) is easy to verify. Finally, the conditions on the coefficients of $T(z)$ and $G(z,w)$ are clearly satisfied.

Since Theorem 10.6 applies, we do obtain an asymptotic expansion for T_n of the form (1.9), with C given by the formula (10.64). It still remains to determine r and C. No closed-form expressions are known for these constants. They are conjectured to be transcendental and algebraically independent of standard constants such as π and e, but no proof is available. Numerically, however, they are simple to compute. Note that

$$\begin{aligned} G_z(r,1) &= \exp(1+u(r))(1+ru'(r)) \\ &= r^{-1} + u'(r) \ , \qquad (15.12) \\ G_{ww}(r,1) &= 1 \ , \qquad (15.13) \end{aligned}$$

so we only need to compute r and $u'(r)$. These quantities can be computed along with $u(r)$ in the same procedure. The basic numerical procedure is to determine r as the positive solution to $T(r) = 1$. To determine $T(x)$ for any positive x, we take any approximation to the $T(x^k)$, $k \geq 1$ (starting initially with x^k as an approximation to $T(x^k)$, say), and combine it with (1.8) (applied with $z = x^m$, $m \geq 1$) to obtain improved approximations. This procedure can be made rigorous. Upper bounds for r, $u(r)$, and $u'(r)$ are especially easy. Since $T_1 = 1$, $T(x) \geq x$

for $0 < x < 1$, and therefore, $T(x^k) \geq x^k$ for $k \geq 1$. Suppose that we start with a fixed value of x and derive some lower bounds of the form $T(x^k) \geq u_k^{(1)} \geq 0$ for $k \geq 1$. Then the functional equation (1.8) implies

$$T(x^m) \geq u_m^{(2)} = x \exp\left(\sum_{k=1}^{\infty} u_{km}/k\right) \qquad m \geq 1 \ . \tag{15.14}$$

This process can be iterated several more times, and to keep the computation manageable, we can always set $u_k^{(j)} = 0$ for $k \geq k_0$. If we ever find a lower bound $T(x) > 1$ by this process, then we know that $r < x$, since $T(r) = 1$. Lower bounds for r are slightly more complicated. ■

We mention here that if U_n denotes the number of unlabeled trees, then the ordinary generating function $U(z) = \sum U_n z^n$ satisfies

$$U(z) = T(z) - T(z)^2/2 + T(z^2)/2 \ . \tag{15.15}$$

Using the results from Example 15.1 about the analytic behavior of $T(z)$, it can be shown that

$$U_n \sim C' r^{-n} n^{-5/2} \ , \tag{15.16}$$

where $r = 0.3383219\ldots$ is the same as before, while $C' = 0.5349485\ldots$.

Example 15.2. *Leftist trees.* Let a_n denote the number of leftist trees of size n (i.e., rooted planar trees with n leaves, such that in any subtree S, the leaf nearest to the root of S is in the right subtree of S [237]). Then $a_1 = a_2 = a_3 = 1$, $a_4 = 2$, $a_5 = 4$. No explicit formula for a_n is known. Even the recurrences for the a_n are complicated, and involve auxiliary sequences. If

$$f(z) = \sum_{n=1}^{\infty} a_n z^n \tag{15.17}$$

denotes the ordinary generating function of a_n, then the combinatorially derived recurrences for the a_n show that [224]

$$f(z) = z + \frac{1}{2} f(z)^2 + \frac{1}{2} \sum_{m=1}^{\infty} g_m(z)^2 \ , \tag{15.18}$$

where the auxiliary generating functions $g_m(z)$ (which enumerate leftist trees with the leftmost leaf at distance $m-1$ from the root) satisfy

$$g_1(z) = z, \quad g_2(z) = z f(z), \quad g_{m+1}(z) = g_m(z) \left[f(z) - \sum_{j=1}^{m-1} g_j(z) \right], \quad m \geq 2 \ , \tag{15.19}$$

and

$$f(z) = \sum_{m=1}^{\infty} g_m(z) \ . \tag{15.20}$$

These generating function relations might not seem promising. If r is the smallest singularity of $f(z)$, then $\sum g_m(z)^2$ is not analytic at r, so we cannot apply Theorem 10.6 in the way it was used in Example 15.1. However, Kemp [224] has sketched a proof that the analytic behavior of $f(z)$ is of the same type as that involved in functions covered by Theorem 10.6, so that it has a dominant square root singularity, and therefore

$$a_n = \alpha c^n n^{-3/2} + O(c^n n^{-5/2}) \ , \tag{15.21}$$

where

$$\alpha = 0.250363429 \ldots , \quad c = 2.749487902 \ldots \ . \tag{15.22}$$

The constants α and c are not known explicitly in terms of other standard numbers such as π or e, but they can be computed efficiently. The $\alpha c^n n^{-3/2}$ term in (15.21) gives an approximation to a_n that is accurate to within 4% for $n = 10$, and within 0.4% for $n = 100$. Thus asymptotic methods yield an approximation to a_n which is satisfactory for many applications. Further results about leftist trees can be found in [225]. ■

15.2. Nonlinear iteration and tree parameters

Example 15.3. *Heights of binary trees.* A binary tree [DEK] is a rooted tree with unlabeled nodes, in which each node has 0 or 2 successors, and left and right successors are distinguished. The size of a binary tree is the number of internal nodes, i.e., the number of nodes with two successors. We let B_n denote the number of binary trees of size n, so that $B_0 = 1$ (by convention), $B_1 = 1$, $B_2 = 2$, $B_3 = 5, \ldots$. Let

$$B(z) = \sum_{n=0}^{\infty} B_n z^n \ . \tag{15.23}$$

Since each nonempty binary tree consists of the root and two binary trees (the left and right subtrees), we obtain the functional equation

$$B(z) = 1 + zB(z)^2 \ . \tag{15.24}$$

This implies that

$$B(z) = \frac{1 - (1 - 4z)^{1/2}}{2z} \ , \tag{15.25}$$

so that

$$B_n = \frac{1}{n+1}\binom{2n}{n} , \tag{15.26}$$

and the B_n are the Catalan numbers. The formula (4.4) (easily derivable from Stirling's formula (4.1)) shows that

$$B_n \sim \pi^{-1/2} n^{-3/2} 4^n \quad \text{as} \quad n \to \infty . \tag{15.27}$$

The height of a binary tree is the number of nodes along the longest path from the root to a leaf. The distribution of heights in binary trees of a given size does not have exact formulas like that of (15.26) for the number of binary trees of a given size. There are several problems on heights that have been answered only asymptotically, and with varying degrees of success. The most versatile approach is through recurrences on generating functions. Let $B_{h,n}$ be the number of binary trees of size n and height $\leq h$, and let

$$b_h(z) = \sum_{n=0}^{\infty} B_{h,n} z^n . \tag{15.28}$$

Then

$$b_0(z) = 0, \quad b_1(z) = 1 , \tag{15.29}$$

and an extension of the argument that led to the relation (15.24) yields

$$b_{h+1}(z) = 1 + z b_h(z)^2 , \quad h \geq 0 . \tag{15.30}$$

The $b_h(z)$ are polynomials in z of degree $2^{h-1} - 1$ for $h \geq 1$. Unfortunately there is no simple formula for them like Eq. (15.25) for $B(z)$, and one has to work with the recurrence (15.30) to obtain many of the results about heights of binary trees. Different problems involve study of the recurrence in different ranges of values of z, and the behavior of the recurrence varies drastically.

For any fixed z with $|z| \leq 1/4$, $b_h(z) \to B(z)$ as $h \to \infty$. For $|z| > 1/4$ the behavior of $b_h(z)$ is more complicated, and is a subject of of nonlinear dynamics [91]. (It is closely related to the study of the Mandelbrot set.) For any real z with $z > 1/4$, $b_h(z) \to \infty$ as $h \to \infty$. To study the distribution of the $B_{h,n}$ as n varies for h fixed, but large, it is necessary to investigate this range of rapid growth. It can be shown [133] that for any λ_1 and λ_2 with $0 < \lambda_1 < \lambda_2 < 1/2$,

$$B_{h,n} = \frac{\exp(2^{h-1}(\beta(r) - r\beta'(r)\log r))}{2^{(h-1)/2}(2\pi(r^2\beta''(r) + r\beta'(r)))^{1/2}}(1 + O(2^{-h/2})) \tag{15.31}$$

uniformly as $h, n \to \infty$ with

$$\lambda_1 < n/2^h < \lambda_2 \ , \tag{15.32}$$

where the function $\beta(x)$ is defined for $1/4 < x < \infty$ by

$$\beta(x) = \log x + \sum_{j=1}^{\infty} 2^{-j} \log\left(1 + \frac{1}{b_j(x) - 1}\right) \ , \tag{15.33}$$

and r is the unique solution in $(1/4, \infty)$ to

$$r\beta'(r) = n2^{-h+1} \ . \tag{15.34}$$

The formula (15.31) might appear circular, in that it describes the behavior of the coefficients $\beta_{h,n}$ of the polynomial $b_h(z)$ in terms of the function $\beta(z)$, which is defined by $b_h(z)$ and all the other $b_j(z)$. However, the series (15.33) for $\beta(z)$ converges rapidly, so that only the first few of the $b_h(z)$ matter in obtaining approximate answers, and computation using (15.33) is efficient. The function $\beta(z)$ is analytic in a region containing the real half-line $x > 1/4$, so the behavior of the $B_{h,n}$ is smooth. It is also known [133] that the behavior of $B_{h,n}$ as a function of n is Gaussian near the peak, which occurs at $n \sim 2^{h-1} \cdot 0.628968\ldots$. The distribution of $B_{h,n}$ is not Gaussian throughout the range (15.32), though.

The proof of the estimate (15.31) is derived from the estimate

$$b_h(z) = \exp(2^{h-1}\beta(z) - \log z)(1 + O(\exp(-\epsilon 2^h))) \ , \tag{15.35}$$

valid in a region along the half-axis $x > 1/4$. The estimates for the coefficients $B_{h,n}$ are obtained by applying the saddle point method. Because of the doubly-exponential rate of growth of $b_h(z)$ for z close to the real axis, it is easy to show that on the circle of integration, the region away from the real axis contributes a negligible amount to $B_{h,n}$. The relation (15.35) is sufficient, together with the smoothness properties of $\beta(z)$, to estimate the contribution of the integral near the real axis. To prove (15.35), one proceeds as in Example 9.7. However, greater care is required because of the complex variables that occur and the need for estimates that are uniform in the variables. The basic recurrence (15.30) shows that

$$\begin{aligned} \log b_{h+1}(z) &= 2\log b_h(z) + \log z + \log\left(1 + \frac{1}{z b_h(z)^2}\right) \\ &= 2\log b_h(z) + \log z + \log\left(1 + \frac{1}{b_{h+1}(z) - 1}\right) \ . \end{aligned} \tag{15.36}$$

Iterating this relation, we find that for $h \geq 1$,

$$\begin{aligned} \log b_{h+1}(z) &= 2^{h+1} \log b_1(z) + (2^h - 1) \log z + \sum_{k=0}^{h-1} 2^k \log \left(1 + \frac{1}{b_{h+1-k}(z) - 1} \right) \\ &= 2^h \left\{ \log z + \sum_{j=1}^{h+1} 2^{-j} \log \left(1 + \frac{1}{b_j(z) - 1} \right) \right\} - \log z \ . \end{aligned} \tag{15.37}$$

The basic equation (15.35) then follows. The technical difficulty is in establishing rigorous bounds for the error terms in the approximations. Details are presented in [133].

Most of the binary trees of a given height h are large, with about $0.3 \cdot 2^h$ internal nodes. This might give the misleading impression that most binary trees are close to the full binary tree of a similar size. However, if we consider all binary trees of a given size n, the average height is on the order of $n^{1/2}$, so that they are far from the full balanced binary trees. The methods that are used to study the average height are different from those used for trees of a fixed height. The basic approach of [133] is to let

$$H_n = \sum_{\substack{T \\ |T|=n}} \mathrm{ht}(T) \ ,$$

where the sum is over the binary trees T of size n, and $\mathrm{ht}(T)$ is the height of T. Then the average height is just H_n / B_n.

The generating function for the H_n is

$$H(z) = \sum_{n=0}^{\infty} H_n z^n = \sum_{h \geq 0} (B(z) - b_h(z)) \ , \tag{15.38}$$

and the analysis of [133] proceeds by investigating the behavior of $H(z)$ in a wedge-shaped region of the type encountered in Section 11.1. If we let

$$\epsilon(z) = (1 - 4z)^{1/2} \ , \tag{15.39}$$

$$e_h(z) = (B(z) - b_h(z))/(2B(z)) \ , \tag{15.40}$$

then the recurrence (15.30) yields

$$e_{h+1}(z) = (1 - \epsilon(z)) e_h(z) (1 - e_h(z)) \ , \quad e_0(z) = 1/2 \ . \tag{15.41}$$

Extensive analysis of this relation yields an approximation to $e_h(z)$ of the form

$$e_h(z) \approx \frac{\epsilon(z)(1 - \epsilon(z))^h}{1 - (1 - \epsilon(z))^h} \ , \tag{15.42}$$

valid for $|\epsilon(z)|$ sufficiently small, $|\mathrm{Arg}\ \epsilon(z)| < \pi/4 + \delta$ for a fixed $\delta > 0$. (The precise error terms in this approximation are complicated, and are given in [133].) This then leads to an expansion for $H(z)$ in a sector $|z - 1/4| < \alpha$, $\pi/2 - \beta < |\mathrm{Arg}(z - 1/4)| < \pi/2 + \beta$ of the form

$$H(z) = -2\log(1 - 4z) + K + O(|1 - 4z|^v) \ , \tag{15.43}$$

where v is any constant, $v < 1/4$, and K is a fixed constant. Transfer theorems of Section 11.1 now yield the asymptotic estimate

$$H_n \sim 2n^{-1}4^n \quad \text{as} \quad n \to \infty \ . \tag{15.44}$$

When we combine (15.44) with (15.27), we obtain the desired result that the average height of a binary tree of size n is $\sim 2(\pi n)^{1/2}$ as $n \to \infty$.

Distribution results about heights of binary trees can be obtained by investigating the generating functions

$$\sum_{h \geq 0} h^r (B(z) - b_h(z)) \ . \tag{15.45}$$

This procedure, carried out in [133] by using modifications of the approach sketched above for the average height, obtains asymptotics of the moments of heights. The method mentioned in Section 6.5 then leads to a determination of the distribution. However, the resulting estimates do not say much about heights far away from the mean. A more careful analysis of the behavior of $e_h(z)$ can be used [126] to show that if $x = h/(2n^{1/2})$, then

$$\frac{B_{h,n} - B_{h-1,n}}{B_n} \sim 2xn^{-1/2} \sum_{m=1}^{\infty} m^2(2m^2x^2 - 3)e^{-m^2x^2} \tag{15.46}$$

as $n, h \to \infty$, uniformly for $x = o((\log n)^{1/2})$, $x^{-1} = o((\log n)^{1/2})$.

For extremely small and large heights, different methods are used. It follows from [126] that

$$\frac{B_{h,n} - B_{h-1,n}}{B_n} \leq \exp(-c(h^2/n + n/h^2)) \tag{15.47}$$

for a constant $c > 0$, which shows that extreme heights are infrequent. (The estimates in [126] are more precise than (15.47).) Bounds of the above form for small heights are obtained in [126] by studying the behavior of the $b_h(z)$ almost on the boundary between convergence and divergence, using the methods of [399]. Let x_h be the unique positive root of $b_h(z) = 2$. Note that $B(1/4) = 2$, and each coefficient of the $b_h(z)$ is nondecreasing as $h \to \infty$. Therefore $x_2 > x_3 > \cdots > 1/4$. More effort shows [126] that x_h is approximately $1/4 + \alpha h^{-2}$ for a certain

$\alpha > 0$. This leads to an upper bound for $B_{h,n}$ by Lemma 8.1. Bounds for trees of large heights are even easier to obtain, since they only involve upper bounds for the $b_h(z) - b_{h-1}(z)$ inside the disk of convergence $|z| < 1/4$. ■

In addition to the methods of [132, 133, 126] that were mentioned above, there are also other techniques for studying heights of trees, such as those of [60, 331]. However, there are problems about obtaining fully rigorous proofs that way. (See the remarks in [126] on this topic.) Most of these methods can be extended to study related problems, such as those of diameters of trees [357].

The results of Example 15.3 can be extended to other families of trees (cf. [132, 133, 126]). What matters in obtaining results such as those of the above example are the form of the recurrences, and especially the positivity of the coefficients.

Example 15.4. *Enumeration of 2,3-trees* [300]. Height-balanced trees satisfy different functional equations than unrestricted trees, which results in different analytic behavior of the generating functions, and different asymptotics. Consider $2,3$-trees; i.e., rooted, oriented trees such that each nonleaf node has either two or three successors, and in which all root-to-leaf paths have the same length. If a_n is the number of $2,3$-trees with exactly n leaves, then $a_1 = a_2 = a_3 = a_4 = 1$, $a_5 = 2, \ldots$, and the generating function

$$f(z) = \sum_{n=1}^{\infty} a_n z^n \tag{15.48}$$

satisfies the functional equation

$$f(z) = z + f(z^2 + z^3) \ . \tag{15.49}$$

Iteration of the recurrence (15.49) leads to

$$f(z) = \sum_{k=0}^{\infty} Q_k(z) \ , \tag{15.50}$$

where $Q_0(z) = z$, $Q_{k+1}(z) = Q_k(z^2 + z^3)$, provided the series (15.50) converges. The Taylor series (15.48) converges only in $|z| < \phi^{-1}$, where $\phi = (1 + 5^{1/2})/2$ is the "golden ratio." Study of the polynomials $Q_k(z)$ shows that the expansion (15.50) converges in a region

$$D = \{z : |z| < \phi^{-1} + \delta, \ |\mathrm{Arg}(z - \phi^{-1})| > \pi/2 - \epsilon\} \tag{15.51}$$

for certain δ, $\epsilon > 0$, and that inside D,

$$f(z) = -c\log(\phi^{-1} - z) + w(\log(\phi^{-1} - z)) + O(|\phi^{-1} - z|) \ , \tag{15.52}$$

where $c = [\phi \log(4 - \phi)]^{-1}$, and $w(t)$ is a nonconstant function, analytic in a strip $|\mathrm{Im}\,(t)| < \eta$ for some $\eta > 0$, such that $w(t + \log(4 - \phi)) = w(t)$. The expression (15.52) only has to be proved in a small vicinity of ϕ^{-1} (intersected with D, of course). Since

$$Q(\phi^{-1} + \nu) = \phi^{-1} + (4 - \phi)\nu + O(|\nu|^2) \tag{15.53}$$

(so that ϕ^{-1} is a repelling fixed point of Q), behavior like that of (15.52) is to be expected, and with additional work can be rigorously shown to hold. Once the expansion (15.52) is established, singularity analysis techniques can then be applied to deduce that

$$a_n \sim \frac{\phi^n}{n} u(\log n) \quad \text{as} \quad n \to \infty \ , \tag{15.54}$$

where $u(t)$ is a positive nonconstant continuous function that satisfies $u(t) = u(t + \log(4 - \phi))$, and has mean value $(\phi \log(4 - \phi))^{-1}$. For details, see [300].

The same methods can be applied to related families of trees, such as those of B-trees. ■

The results of Example 15.3 and the generalizations mentioned above all apply only to the standard counting models, in which all trees with a fixed value of some simple property, such as size or height, are equally likely. Often, especially in computer science applications, it is necessary to study trees produced by some algorithm, and consider all outputs of this algorithm as equally likely. For example, in sorting it is natural to consider all permutations of n elements as equally probable. If random permutations are used to construct binary search trees, then the distribution of heights will be different from that in the standard model, and the two trees of maximal height will have probability of $2/n!$ of occurring. The average height turns out to be $\sim c \log n$ as $n \to \infty$, for $c = 4.311\ldots$ a certain constant given as a solution to a transcendental equation. This was shown by Devroye [92] (see also [93]) by an application of the theory of branching processes. For a detailed exposition of this method and other applications to similar problems, see [270]. The basic generating function approach that we have used in most of this chapter leads to functional iterations which have not been solved so far.

15.3. Differential and integral equations

Section 9.2 showed that differential equations arise naturally in analyzing linear recurrences of finite order with rational coefficients. There are other settings when they arise even more naturally. As is true of nonlinear iterations in the previous section and the functional equations of the next one, differential and integral equations are typically used to extract information about singularities of generating functions. We have already seen in Example 9.3 and other cases that differential equations can yield an explicit formula for the generating function, from which it is easy to deduce what the singularities are and how they affect the asymptotics of the coefficients. Most differential equations do not have a closed-form solution. However, it is often still possible to derive the necessary information about analytic behavior even when there is no explicit formula for the solution. We demonstrate this with a brief sketch of a recent analysis of this type [131]. Other examples can be found in [270].

Example 15.5. *Search costs in quadtrees* [131]. Quadtrees are a well-known data structure for multidimensional data storage [168]. Consider a d-dimensional data space, and let n points be drawn independently from the uniform distribution in the d-dimensional unit cube. We take d fixed and $n \to \infty$. Suppose that the first $n-1$ points have already been inserted into the quadtree, and let D_n be the search cost (defined as the number of internal nodes traversed) in inserting the n-th item. The result of Flajolet and Lafforgue [131] is that D_n converges in distribution to a Gaussian law when $n \to \infty$. If μ_n and σ_n denote the mean and standard deviation of D_n, respectively, then

$$\mu_n \sim 2d^{-1} \log n, \quad \sigma_n \sim d^{-1}(2 \log n)^{1/2} \quad \text{as} \quad n \to \infty \ , \tag{15.55}$$

and for all real $\alpha < \beta$, as $n \to \infty$,

$$Pr(\alpha \sigma_N < D_n - \mu_n < \beta \sigma_n) \sim (2\pi)^{-1/2} \int_\alpha^\beta \exp(-x^2/2) dx \ . \tag{15.56}$$

The results for μ_n and σ_n had been known before, and required much simpler techniques for their solution, see [270]. It was only necessary to study asymptotics of ordinary differential equations in a single variable. To obtain distribution results for search costs, it was necessary to study bivariate generating functions. The basic relation is

$$\sum_k Pr\{D_n = k\} u^k = (2^d u - 1)^{-1} (\phi_n(u) - \phi_{n-1}(u)) \ , \tag{15.57}$$

where the polynomials $\phi_n(u)$ have the bivariate generating function

$$\Phi(u,z) = \sum_{n=0}^{\infty} \phi_n(u) z^n \ . \tag{15.58}$$

which satisfies the integral equation

$$\begin{aligned}\Phi(u,z) = 1 + 2^d u \int_0^z \frac{dx_1}{x_1(1-x_1)} \int_0^{x_1} \frac{dx_2}{x_2(1-x_2)} \int_0^{x_2} \frac{dx_3}{x_3(1-x_3)} \cdots \\ \int_0^{x_{d-2}} \frac{dx_{d-1}}{x_{d-1}(1-x_{d-1})} \int_0^{x_{d-1}} \Phi(u,x_d) \frac{dx_d}{1-x_d} \ .\end{aligned} \tag{15.59}$$

This integral equation can easily be reduced to an equivalent differential equation, which is what is used in the analysis. For $d = 1$ there is an explicit solution

$$\Phi(u,z) = (1-z)^{-2u} \ , \tag{15.60}$$

which shows that D_n can be expressed in terms of Stirling numbers. This is not surprising, since for $d = 1$ the quadtree reduces to the binary search tree, for which these results were known before. For $d = 2$, $\Phi(u,z)$ can be expressed in terms of standard hypergeometric functions. However, for $d \geq 3$ there do not seem to be any explicit representations of $\Phi(u,z)$. Flajolet and Lafforgue use a singularity perturbation method to study the behavior of $\Phi(u,z)$. They start out with the differential system derivable in standard way from the differential equation associated to (15.59) (i.e., a system of d linear differential equations in z with coefficients that are rational in z). Since only values of u close to 1 are important for the distribution results, they regard u as a perturbation parameter of this system. For every fixed u, they determine the dominant singularity of the linear differential system in the variable z, using the indicial equations that are standard in this setting. It turns out that the dominant singularity is a regular one at $z = 1$, and

$$\Phi(u,z) \approx c(u)(1-z)^{-2u^{1/d}} \ , \tag{15.61}$$

at least for z and u close to 1. This behavior of $\Phi(u,z)$ is then used (in its more precise form, with explicit error terms) to deduce, through the transfer theorem methods explained in Section 11, the behavior of $\phi_n(u)$:

$$\phi_n(u) \approx c(u)\Gamma(2u^{1/d})^{-1} n^{2u^{1/d}-1} \ . \tag{15.62}$$

This form, again in a more precise formulation, is then used to deduce that the behavior of D_n is normal near its peak, and that the tails of the distribution are small. ■

15.4. Functional equations

One area that needs and undoubtedly will receive much more attention is that of complicated nonlinear relations for generating functions. Even in a single variable our knowledge is limited. Some of the work of Mahler [267, 268, 269], devoted to functions $f(z)$ satisfying equations of the form $p(f(z), f(z^g)) = 0$, where $p(u, v)$ is a polynomial, shows that it is possible to extract information about the analytic behavior of $f(z)$ near its singularities. This can then be used to study the coefficients.

Sometimes seemingly complicated functional equations do have easy solutions.

Example 15.6. *A pebbling game.* In a certain pebbling game [76], minimal configurations of size n are counted by $T_n(0)$, where $T_n(x)$ is a polynomial that satisfies $T_n(x) = 0$ for $0 \le n \le 2$, $T_3(x) = 4x + 2x^2$, and for $n \ge 3$,

$$T_{n+1}(x) = x^{-1}(1+x)^2 T_n(x) - x^{-1} T_n(0) + x T_n'(0) + n x^n \ . \tag{15.63}$$

The coefficients of $T_n(x)$ are ≥ 0, and

$$T_{n+1}(1) \le 4T_n(1) + T_n(1) + 1 \le 6T_n(1) \ , \tag{15.64}$$

so clearly each coefficient of $T_n(x)$ is $\le 6^n$, say. Let

$$f(x, y) = \sum_{n=0}^{\infty} T_n(x) y^n \ . \tag{15.65}$$

The bound on $T_n(1)$ shows that $f(x, y)$ is analytic in x and y for $|x| < 1$, $|y| < 1/6$, say, with x and y complex. Then the recurrence (15.63) leads to the functional equation

$$\begin{aligned} (x - y(1+x)^2) f(x, y) &= 2x^2(2+x)y^3 + x^2y^2(1 - 2x^2y^2)(1 - xy)^{-2} \\ &\quad - y f(0, y) + x^2 y f_x(0, y) \ , \end{aligned} \tag{15.66}$$

where $f_x(x, y)$ is the partial derivative of $f(x, y)$ with respect to x. We now differentiate the equation (15.66) with respect to x and set $x = 0$. We find that

$$(1 - 2y) f(0, y) = y f_x(0, y) \ , \tag{15.67}$$

and therefore

$$\begin{aligned} (x - y(1+x)^2) f(x, y) &= 2x^2(2+x)y^3 + x^2y^2(1 - 2x^2y^2)(1 - xy)^{-2} \\ &\quad - [y + (2y - 1)x^2] f(0, y) \ . \end{aligned} \tag{15.68}$$

When

$$x = y(1+x)^2 \ , \tag{15.69}$$

the left side of Eq. (15.68) vanishes, and Eq. (15.68) yields the value of $f(0,y)$. Now Eq. (15.69) holds for

$$x = (2y)^{-1}(1-2y \pm (1-4y)^{1/2}) \ .$$

To ensure that (15.69) holds for x and y both in a neighborhood of 0, we set

$$g(y) = (2y)^{-1}(1-2y-(1-4y)^{1/2}) \ . \tag{15.70}$$

Then $g(y) = y(1+g(y))^2$, $g(y)$ is analytic for $|y|$ small, and so substituting $x = g(y)$ in Eq. (15.68) yields

$$\begin{aligned} [y+(2y-1)g(y)^2]f(0,y) &= 2g(y)^2(2+g(y))y^3 \\ &\quad + y^2g(y)^2(1-2y^2g(y)^2)(1-yg(y))^{-2} \ . \end{aligned} \tag{15.71}$$

Thus $f(0,y)$ is an algebraic function of y. Eq. (15.71) was proved only for $|y|$ small, but it can now be used to continue $f(0,y)$ analytically to the entire complex plane with the exception of a slit from $1/4$ to infinity along the positive real axis. There is a first order pole at $y = 1/r$, with $r = 4.1478990357\ldots$ the positive root of

$$r^3 - 7r^2 + 14r - 9 = 0 \ , \tag{15.72}$$

and no other singularities in $|y| < 1/4$. Hence we obtain

$$T_n(0) = [y^n]f(0,y) = cr^n + O((4+\epsilon)^n) \tag{15.73}$$

as $n \to \infty$, for every $\epsilon > 0$, where c is an algebraic number that can be given explicitly in terms of r.

The value of $f(0,y)$ is determined by Eq. (15.71), and together with Eq. (15.68) gives $f(x,y)$ explicitly as an algebraic function of x and y. The resulting expression can then be used to determine other coefficients of the polynomials $T_n(x)$. ■

Example 15.6 was easy to present because of the special structure of the functional equation. The main trick was to work on the variety defined by Eq. (15.69), on which the main term vanishes, so that one can analyze the remaining terms. The same basic approach also works

in more complicated situations. The analysis of certain double queue systems leads to two-variable generating functions for the equilibrium probabilities that satisfy equations such as the following one, obtained by specializing the problem treated in [145]:

$$Q(z,w)f(z,w) = 2z(w-1)f(z,0) + 3w(z-1)f(0,w) \ , \tag{15.74}$$

valid for complex z and w with $|z|, |w| \leq 1$, where

$$Q(z,w) = 6zw - 3w - 2z - z^2w^2 \ . \tag{15.75}$$

The generating function $f(z,w)$ is analytic in z and w. What makes this problem tractable is that on the algebraic curve in two-dimensional complex space defined by $Q(z,w) = 0$, the quantity on the right-hand side of Eq. (15.74) has to vanish, and this imposes stringent conditions on $f(z,0)$ and $f(0,w)$, which leads to their determination. Once $f(z,0)$ and $f(0,w)$ are found, $f(z,w)$ is defined by Eq. (15.74), and one can determine the asymptotics of its coefficients. Treatment of functional equations of the type (15.74) was started by Malyshev [274]. For recent work and references to other papers in this area, see [144, 145]. This approach has so far been successful only for two-variable problems with $Q(z,w)$ of low degree. Moreover, the mathematics of the solution is far deeper than that used in Example 15.6.

16. Other methods

This section mentions a variety of methods that are not covered elsewhere in this chapter but are useful in asymptotic enumeration. Most are discussed briefly, since they belong to large and well developed fields that are beyond the scope of this survey.

16.1. Permanents

Van der Waerden's conjecture, proved by Falikman [113] and Egorychev [98], can be used to obtain lower bounds for certain enumeration problems. It states that if A is an $n \times n$ matrix that is doubly stochastic (entries ≥ 0, all row and column sums equal to 1) then the permanent of A satisfies $\mathrm{per}(A) \geq n^{-n}n!$. (For most asymptotic problems it is sufficient to rely on an earlier result of T. Bang [26] and S. Friedland [148] which gives a lower bound of $\mathrm{per}(A) \geq e^{-n}$ that is worse only by a factor of $n^{1/2}$.) There is also an upper bound for permanents. Minc's conjecture, proved first by Bragman and in a simpler way by Schrijver [340] states that an

$n \times n$ matrix A with $0,1$ entries and row sums $r_1, \ldots, r_n$ has

$$\mathrm{per}(A) \le \prod_{j=1}^{n} (r_j!)^{1/r_j} \ .$$

We now show how these results can be applied.

Example 16.1. *Latin rectangles.* Suppose we are given a $k \times n$ Latin rectangle, $k < n$, so that the symbols are $1, 2, \ldots, n$, and no symbol appears twice in any row or column. In how many ways can we extend this rectangle to a $(k+1) \times n$ Latin rectangle? To get a lower bound, form an $n \times n$ matrix $B = (b_{ij})$, with $b_{ij} = 1$ if i does not appear in column j of the rectangle, and $b_{ij} = 0$ otherwise. Then the row and column sums of B are all equal to $n - k$, so $(n-k)^{-1}B$ is doubly stochastic. Therefore $\mathrm{per}(B)$, which equals the desired number of ways of extending the rectangle, is $\ge (n-k)^n n^{-n} n!$ by van der Waerden's conjecture. By Minc's conjecture, we also have $\mathrm{per}(B) \le ((n-k)!)^{n/(n-k)}$. If we let $L(k,n)$ denote the number of $k \times n$ Latin rectangles, then $L(1,n) = n!$, and the bounds derived above for the number of ways to extend any given rectangle give

$$L(k,n) \ \ge \ \prod_{j=0}^{k-1} \{(n-j)^n n^{-n} n!\} = n^{-kn} (n!)^{2n} ((n-k)!)^{-n} \ , \tag{16.1}$$

$$L(k,n) \ \le \ \prod_{j=0}^{k-1} \{(n-j)!\}^{n/(n-j)} \ . \tag{16.2}$$

Sharper estimates for $L(k,n)$ have been obtained through more powerful and complicated methods by Godsil and McKay [163]. They obtain an asymptotic relation for $L(k,n)$ that is valid for $k = o(n^{6/7})$, and improved estimates for other k. (It is known that for any fixed k, the sequence $L(k,n)$ satisfies a linear recurrence with polynomial coefficients [160].) ■

There are problems in which inequalities for permanents give the correct asymptotic estimates. One such example is presented in [318] which discusses a variation on the "problème des rencontres."

16.2. Probability theory and branching process methods

Many combinatorial enumeration results can be phrased in probabilistic language, and a few probabilistic techniques have appeared in the preceding sections. However, the stress throughout this chapter has been on elementary and generating function approaches to asymptotic enumeration problems. Probabilistic methods provide another way to approach many of

these problems. This has been appreciated more in the former Soviet Union than in the West, as can be seen in the books [240, 241, 338].

The last few years have seen a great increase in the applications of probabilistic methods to combinatorial enumeration and analysis of algorithms. Many powerful tools, such as martingales, branching processes, and Brownian motion asymptotics have been brought to bear on this topic. General introductions and references to these topics can be found in Chapter ? as well as in [5, 11, 20, 21, 27, 92, 93, 108, 258, 260, 262, 270].

16.3. Statistical physics

There is an extensive literature in mathematical physics concerned with asymptotic enumeration, especially in Ising models of statistical mechanics and percolation methods. Many of the methods are related to combinatorial enumeration. For an introduction to them, see Chapter ? or the books [30, 226].

16.4. Classical applied mathematics

There are many techniques, such as the ray method and the WKB method, that have been developed for solving differential and integral equations in what we might call classical applied mathematics. An introduction to them can be found in [31]. They are powerful, but they have the disadvantage that most of them are not rigorous, since they make assumptions about the form or the stability of the solution that are likely to be true, but have not been established. Therefore we have not presented such methods in this survey. For some examples of the nonrigorous applications of these methods to asymptotic enumeration, see the papers of Knessl and Keller [231, 232]. It is likely that with additional work, more of these methods will be rigorized, which will increase their utility.

17. Algorithmic and automated asymptotics

Deriving asymptotic expansions often involves a substantial amount of tedious work. However, much of it can now be done by computer symbolic algebra systems such as Macsyma, Maple, and Mathematica. There are many widely available packages that can compute Taylor series expansions. Several can also compute certain types of limits, and some have implemented Gosper's indefinite hypergeometric summation algorithm [171]. They ease the burden of carrying out the necessary but uninteresting parts of asymptotic analysis. They are especially

useful in the exploratory part of research, when looking for identities, formulating conjectures, or searching for counterexamples.

Much more powerful systems are being developed. Given a sequence, there are algorithms that attempt to guess the generating function of that sequence [46, 162]. It is possible to go much further than that. Many of the asymptotic results in this chapter are stated in explicit forms. As an example, the asymptotics of a linear recurrence is derived easily from the characteristic polynomial and the initial conditions, as was shown in Section 9.1. One needs to compute the roots of the characteristic polynomial, and that is precisely what computer systems do well. It is therefore possible to write programs that will derive the asymptotics behavior from the specification of the recurrence. More generally, one can analyze asymptotics of a much greater variety of generating functions. Flajolet, Salvy, and Zimmermann [124, 139] have written a powerful program for just such computations. Their system uses Maple to carry out most of the basic analytic computations. It contains a remarkable amount of automated expertise in recognizing generating functions, computing their singularities, and extracting asymptotic information about their coefficients. For example, if

$$f(z) = -\log[1 + z\log(1 - z^2)] + (1 - z^3)^{-5} + \exp(ze^z) \ , \tag{17.1}$$

then the Flajolet-Salvy-Zimmermann system can determine that the singularity of $f(z)$ that is closest to the origin is at $z = \rho$, where ρ is the smallest positive root of

$$1 = -\rho\log(1 - \rho^2) \ , \tag{17.2}$$

and then can deduce that

$$[z^n]f(z) = n^{-1}\rho^{-n} + O(n^{-2}\rho^{-n}) \quad \text{as} \quad n \to \infty \ . \tag{17.3}$$

The Flajolet-Salvy-Zimmermann system is even more powerful than indicated above, since it does not always require an explicit presentation of the generating function. Instead, often it can accept a formal description of an algorithm or data structure, derive the generating function from that, and then obtain the desired asymptotic information. For example, it can show that the average path length in a general planar tree with n nodes is

$$\frac{1}{2}\pi^{1/2}n^{3/2} + \frac{1}{2}n + O(n^{1/2}) \quad \text{as} \quad n \to \infty \ . \tag{17.4}$$

What makes systems such as that of [139] possible is the phenomenon, already mentioned in Section 6, that many common combinatorial operations on sets, such as unions and permutations, correspond in natural ways to operations on generating functions.

Further work extending that of [139] is undoubtedly going to be carried out. There are some basic limitations coming from the undecidability of even simple problems of arithmetic, which are already known to impose a limitation on the theories of indefinite integration. If we approximate a sum by an integral

$$\int_a^b x^{-\alpha} dx \ , \tag{17.5}$$

then as a next step we need to decide whether $\alpha = 1$ or not, since if $\alpha = 1$, this integral is $\log(b/a)$ (assuming $0 < a < b < \infty$), whereas if $\alpha \neq 1$, it is $(b^{1-\alpha} - a^{1-\alpha})/(1-\alpha)$. Deciding whether $\alpha = 1$ or not, when α is given implicitly or by complicated expressions, can be arbitrarily complicated. However, such difficulties are infrequent, and so one can expect substantial increase in the applicability of automated systems for asymptotic analysis.

The question of decidability of asymptotic problems and generic properties of combinatorial structures that can be specified in various logical frameworks has been treated by Compton in a series of papers [77, 78, 79]. There is the beautiful recent theory of 0-1 laws for random graphs, which says that certain (so-called first-order) properties are true with probability either 0 or 1 for random graphs. Compton proves that certain classes of asymptotic theories also have 0-1 laws, and describes general properties that have to hold for almost all random structures in certain classes. His analysis uses Tauberian theorems and Hayman admissibility to determine asymptotic behavior. For some further developments in this area, see also [35].

18. Guide to the literature

This section presents additional sources of information on asymptotic methods in enumeration and analysis of algorithms. It is not meant to be exhaustive, but is intended to be used as a guide in searching for methods and results. Many references have been presented already throughout this chapter. Here we describe only books that cover large areas relevant to our subject.

An excellent introduction to the basic asymptotic techniques is given in [175]. That book, intended to be an undergraduate textbook, is much more detailed than this chapter, and assumes no knowledge of asymptotics, but covers fewer methods. A less comprehensive and less elementary book that is oriented towards analysis of algorithms, but provides a good introduction to many asymptotic enumeration methods, is [177].

The best source from which to learn the basics of more advanced methods, including many of those covered in this chapter, is de Bruijn's book [63]. It was not intended particularly

for those interested in asymptotic enumeration, but almost all the methods in it are relevant. De Bruijn's volume is extremely clear, and provides insight into why and how various methods work.

General presentations of asymptotic methods, although usually with emphasis on applications to applied mathematics (differential equations, special functions, and so on) are available in the books [54, 100, 114, 115, 315, 344, 354, 372, 382, 385]. Integral transforms are treated extensively in [89, 95, 116, 299, 365]. Books that deal with asymptotics arising in the analysis of algorithms or probabilistic methods include [11, 55, 108, 209, 223, 240, 241, 270, 338].

Nice general introductions to combinatorial identities, generating functions, and related topics are presented in [81, 351, 377]. Further material can be found in [13, 88, 99, 173, 188, 335, 336].

A very useful book is the compilation [168]. While it does not discuss methods in too much detail, it lists a wide variety of enumerative results on algorithms and data structures, and gives references where the proofs can be found.

Last, but not least in our listing, is Knuth's three-volume work [235, 236, 237]. While it is devoted primarily to analysis of algorithms, it contains an enormous amount of material on combinatorics, especially asymptotic enumeration.

Acknowledgements

The author thanks R. Arratia, E. A. Bender, E. R. Canfield, H. Cohen, P. Flajolet, Z. Gao, D. E. Knuth, V. Privman, L. B. Richmond, E. Schmutz, N. J. A. Sloane, D. Stark, S. Tavaré, H. S. Wilf, D. Zagier and D. Zeilberger for their helpful comments on preliminary drafts of this chapter.

References

[1] J. Aczél, *Lectures on Functional Equations and Their Applications*, Academic Press, 1966.

[2] C. R. Adams, On the irregular cases of linear ordinary difference equations, *Trans. Am. Math. Soc., 30* (1928), pp. 507–541.

[3] A. V. Aho and N. J. A. Sloane, Some doubly exponential sequences, *Fibonacci Quart., 11* (1973), 429–437.

[4] J. A. Aizenberg and A. P. Yuzhakov, *Integral Representations and Residues in Multidimensional Complex Analysis*, Trans. Math. Monographs, No. 58, Amer. Math. Soc., 1983.

[5] D. Aldous, *Probability approximations via the Poisson clumping heuristic*, Springer-Verlag, 1989.

[6] J.-P. Allouche and J. Shallit, The ring of k-regular sequences, *Theoretical Comp. Science, 98* (1992), 163–197.

[7] G. Almkvist, Proof of a conjecture about unimodal polynomials, *J. Number Theory, 32* (1989), 43–57.

[8] G. Almkvist, Exact asymptotic formulas for the coefficients of nonmodular functions, *J. Number Theory, 38* (1991), 145–160.

[9] G. Almkvist, A rather exact formula for the number of plane partitions, *A Tribute to Emil Grosswald*, M. Knapp and M. Sheingorn, eds., Amer. Math. Soc., to appear.

[10] G. Almkvist and G. E. Andrews, A Hardy-Ramanujan-Rademacher formula for restricted partitions, *J. Number Theory, 38* (1991), 135–144.

[11] N. Alon and J. H. Spencer, *The Probabilistic Method*, Wiley, 1992.

[12] G. E. Andrews, Applications of basic hypergeometric functions, *SIAM Review*, *16* (1974), pp. 441–484.

[13] G. E. Andrews, *The Theory of Partitions*, Addison–Wesley, 1976.

[14] T. M. Apostol, *Mathematical Analysis*, Addison Wesley, 1957.

[15] T. M. Apostol, *Introduction to Analytic Number Theory*, Springer, 1976.

[16] J. Arney and E. D. Bender, Random mappings with constraints on coalescence and number of origins, *Pacific J. Mathematics*, *103* (1982), pp. 269–294.

[17] R. Arratia, L. Goldstein, and L. Gordon, Poisson approximation and the Chen-Stein method, *Statistical Science 5* (1990), 402–423.

[18] R. Arratia, L. Gordon, and M. S. Waterman, The Erdös-Rényi law in distribution for coin tossing and sequence matching, *Ann. Statist. 18* (1990), 539–570.

[19] R. Arratia and S. Tavaré, The cycle structure of random permutations, *Ann. Prob., 20* (1992), 1567–1591.

[20] R. Arratia and S. Tavaré, Limit theorems for combinatorial structures via discrete process approximation, *Random Structures Alg., 3* (1992), 321–345.

[21] R. Arratia and S. Tavaré, Independent process approximations for random combinatorial structures, *Adv. Math.* (1993), in press.

[22] F. C. Auluck and C. B. Haselgrove, On Ingham's Tauberian theorem for partitions, *Proc. Cambridge Philos. Soc., 48* (1952), pp. 566–570.

[23] R. Ayoub, *An Introduction to the Analytic Theory of Numbers*, Amer. Math. Soc., 1963.

[24] R. Baeza-Yates, R. Casas, J. Diaz, and C. Martinez, On the average size of the intersection of binary trees, *SIAM J. Comp. 21* (1992), 24–32.

[25] R. A. Baeza-Yates, A trivial algorithm whose analysis is not: a continuation, *BIT, 29* (1989), 378–394.

[26] T. Bang, Om matrixfunktioner som med et numerisk lille deficit viser v. d. Waerdens permanenthypotese, Proc. 1976 Turku Scand. Math. Congress.

[27] A. D. Barbour, L. Holst, and S. Janson, *Poisson Approximation*, Oxford University Press, 1992.

[28] E. W. Barnes, On the homogeneous linear difference equation of the second order with linear coefficients, *Messenger Math., 34* (1904), pp. 52–71.

[29] P. M. Batchelder, *An Introduction to Linear Difference Equations*, Harvard Univ. Press, 1927. Dover reprint, 1967.

[30] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*, Academic Press, 1982.

[31] C. M. Bender and S. A. Orszag, *Applied Mathematical Methods for Scientists and Engineers*, McGraw-Hill, 1978.

[32] E. A. Bender, Central and local limit theorem applied to asymptotic enumeration, *J. Comb. Theory Ser. A., 15* (1973), pp. 91–111.

[33] E. A. Bender, Asymptotic methods in enumeration, *SIAM Review*, *16* (1974), pp. 485–515.

[34] E. A. Bender, An asymptotic expansion for the coefficients of some formal power series, *J. London Math. Soc.*, *9* (1975), pp. 451–458.

[35] E. A. Bender, Z.-C. Gao, and L. B. Richmond, Submaps of maps. I. General 0-1 laws, *J. Comb. Theory, B 55* (1992), 104–117.

[36] E. A. Bender and J. R. Goldman, Enumerative uses of generating functions, *Indiana Univ. Math. J., 20* (1971), pp. 753–765.

[37] E. A. Bender, A. M. Odlyzko, and L. B. Richmond, The asymptotic number of irreducible partitions, *European J. Combinatorics, 6* (1985), pp. 1–6.

[38] E. A. Bender and L. B. Richmond, Central and local limit theorems applied to asymptotic enumeration. II: Multivariate generating functions, *J. Comb. Theory B, 34* (1983), 255–265.

[39] E. A. Bender and L. B. Richmond, An aymptotic expansion for the coefficients of analytic generating functions, *Discrete Math.*, *50* (1984), pp. 135–141.

[40] E. A. Bender and L. B. Richmond, An asymptotic expansion for the coefficients of some power series II: Lagrange inversion, *Discrete Mathematics*, *50* (1984), pp. 135–141.

[41] E. A. Bender and L. B. Richmond, A survey of the asymptotic behaviour of maps, *J. Comb. Theory, Ser. B*, *40* (1986), pp. 297–329.

[42] E. A. Bender, L. B. Richmond, and S. G. Williamson, Central and local limit theorems applied to asymptotic enumeration. III. Matrix recursions, *J. Comb. Theory (A) 35* (1983), 263–278.

[43] E. A. Bender and S. G. Williamson, *Foundations of Applied Combinatorics*, Addison-Wesley, 1991.

[44] F. Bergeron and G. Cartier, Darwin: Computer algebra and enumerative combinatorics, *STACS–88*, R. Cori and M. Wirsing, eds., *LNCS*, *294* (1988), pp. 393–394.

[45] F. Bergeron and G. Labelle and P. Leroux, Functional equations for data structures, *STACS–88*, R. Cori and M. Wirsing, eds., *LNCS*, *294* (1988), pp. 73–80.

[46] F. Bergeron and S. Plouffe, Computing the generating function of a series given its first few terms, *Experimental Math. 1* (1992), 307–312.

[47] B. C. Berndt and L. Schoenfeld, Periodic analogues of the Euler-Maclaurin and Poisson summation formulas with applications to number theory, *Acta Arith. 28* (1975/76), 23–68.

[48] M. V. Berry and C. J. Howls, Hyperasymptotics, *Proc. Royal Soc. London A, 430* (1990), 653–667.

[49] A. Bertozzi and J. McKenna, Multidimensional residues, generating functions, and their application to queueing networks, *SIAM Review 35* (1993), 239–268.

[50] P. Billingsley, *Probability and Measure*, Wiley, 1979.

[51] G. D. Birkhoff, General theory of linear difference equations, *Trans. Amer. Math. Soc.*, *12* (1911), pp. 243–284.

[52] G. D. Birkhoff, Formal theory of irregular linear difference equations, *Acta Math., 54* (1930), pp. 205–246.

[53] G. D. Birkhoff and W. J. Trjitzinsky, Analytic theory of singular difference equations, *Acta Math.*, *60* (1932), pp. 1–89.

[54] N. Bleistein and R. A. Handelsman, *Asymptotic Expansions of Integrals*, 2nd edition, Holt, Rinehart and Winston, New York, 1975.

[55] B. Bollobás, *Random Graphs*, Academic Press, 1985.

[56] F. Brenti, *Unimodal, Log-concave, and Pólya Frequency Sequences in Combinatorics*, Memoirs Amer. Math. Soc., no. 413 (1989).

[57] F. Brenti, G. F. Royle, and D. G. Wagner, Location of zeros of chromatic and related polynomials of graphs, to be published.

[58] N. A. Brigham, A general asymptotic formula for partition functions, *Proc. Amer. Math. Soc., 1* (1950), pp. 182–191.

[59] T. I'a. Bromwich, *An Introduction to the Theory of Infinite Series*, 2nd rev. ed., Macmillan, London, 1955.

[60] G. G. Brown and B. O. Shubert, On random binary trees, *Math. Oper. Res.*, *9* (1984), pp. 43–65.

[61] N. G. de Bruijn, On Mahler's partition problem, *Indagationes Math.*, *10* (1948), pp. 210–220.

[62] N. G. de Bruijn, The difference–differential equation $F'(x) = e^{\alpha x+\beta}F(x-1)$, *Indagationes Math.*, *15* (1953), pp. 449–458.

[63] N. G. de Bruijn, *Asymptotic Methods in Analysis*, North–Holland, Amsterdam, 1958.

[64] N. G. de Bruijn, D. E. Knuth, and S. O. Rice, The average height of planted plane trees, in *Graph Theory and Computing*, R.–C. Read, ed., Academic Press, New York, 1972, pp. 15–22.

[65] E. R. Canfield, Central and local limit theorems for the coefficients of polynomials of binomial type, *J. Comb. Theory*, Series A, *23* (1977), pp. 275–290.

[66] E. R. Canfield, The asymptotic behavior of the Dickman–de Bruijn function, *Congressus Numerantium*, *35* (1982), pp. 139–148.

[67] E. R. Canfield, Remarks on an asymptotic method in combinatorics, *J. Comb. Theory*, Series A, *37* (1984), pp. 348–352.

[68] M. Car, Factorisation dans $\mathbf{F}_q[X]$, *C. R. Acad. Sci. Paris Série I, 294* (1982), pp. 147–150.

[69] M. Car, Ensembles de polynômes irréductibles et thérèmes de densité, *Acta Arith, 44* (1984), pp. 323–342.

[70] L. Carlitz, Permutations, sequences and special functions, *SIAM Review*, *17* (1975), pp. 298–322.

[71] R. Casas, D. Diaz, and C. Martinez, Statistics on random trees, in *Automata, Languages, and Programming* (Proc. 18th ICALP, Madrid, 1991), J. Leach Albert, B. Monien, and M. Rodriguez Artalejo, eds., Springer LNCS #510, 1991, pp. 186–203.

[72] J. W. S. Cassels, On the representation of integers as the sums of distinct summands taken from a fixed set, *Acta Sci. Math. Hungar., 21* (1960), 111–124.

[73] L. Cerlienco, M. Mignotte, and F. Piras, Suites récurrentes linéaires, *L' Enseign. Math., 33* (1987), 67–108.

[74] Ch. A. Charalambides and A. Kyriakoussis, An asymptotic formula for the exponential polynomials and a central limit theorem for their coefficients, *Discrete Math.*, *54* (1985), pp. 259–270.

[75] L. H. Y. Chen, Poisson approximation for dependent trials, *Ann. Prob. 3* (1975) 534–545.

[76] F. R. K. Chung, R. L. Graham, J. A. Morrison, and A. M. Odlyzko, Pebbling a chessboard, *Am. Math. Monthly*, to appear.

[77] K. J. Compton, A logical approach to asymptotic combinatorics. I. First order properties, *Advances in Math.*, *65* (1987), pp. 65–96.

[78] K. J. Compton, 0–1 laws in logic and combinatorics, in *Proceedings NATO Advanced Study Institute on Algorithms and Order*, I. Rival, ed., Reidel, Dordrecht, 1988, pp. 353–383.

[79] K. J. Compton, A logical approach to asymptotic combinatorics. II. Monadic second–order properties, *J. Comb. Theory*, Series A, *50* (1989), pp. 110–131.

[80] L. Comtet, Birecouvrements et birevêtements d'un ensemble fini, *Studia Sci. Math. Hungar. 3* (1968), 137–152.

[81] L. Comtet, *Advanced Combinatorics*, Reidel, Dordrecht, 1974.

[82] C. N. Cooper and R. E. Kennedy, A partial asymptotic formul for the Niven numbers, *Fibonacci Quarterly*, *26* (1988), pp. 163–168.

[83] J. Coquet, A summation formula related to binary digits, *Inventiones math.*, *73* (1983), pp. 107–115.

[84] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience, 1953 (vol. 1) and 1962 (vol. 2).

[85] T. W. Cusick, Recurrences for sums of powers of binomial coefficients, *J. Comb. Theory*, Series A, *52* (1989), pp. 77–83.

[86] H. E. Daniels, Saddlepoint approximations in statistics, *Annals Math. Statistics*, *25* (1954), pp. 631–650.

[87] G. Darboux, Mémoire sur l'approximation des fonctions de très-grands nombres, et sur une classe étendue de développements en série, *J. Math. Pures Appl., 4* (1878), 5–56, 377–416.

[88] F. N. David and D. E. Barton, *Combinatorial Chance*, Griffin, 1962.

[89] B. Davies, *Integral Transforms and Their Applications*, Springer, 1978.

[90] J. Denef and L. Lipshitz, Algebraic power series and diagonals, *J. Number Theory, 26* (1987), pp. 46–67.

[91] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed., Addison-Wesley, 1989.

[92] L. Devroye, A note on the expected height of binary search trees, *J. ACM*, *33* (1986), pp. 489–498.

[93] L. Devroye, Branching processes in the analysis of the heights of trees, *Acta Informatica*, *24* (1987), pp. 277–298.

[94] P. Diaconis and D. Freedman, Finite exchangeable sequences, *Ann. Probab. 8* (1980), 745–764.

[95] G. Doetsch, *Handbuch der Laplace Transformation*, Birkhäuser, Basel, 1955.

[96] M. Drmota, Asymptotic distributions and a multivariate Darboux method in enumeration problems, *J. Comb. Theory A*, to appear.

[97] R. Durrett, *Probability: Theory and Examples*, Wadsworth and Brooks/Cole, 1991.

[98] G. P. Egorychev, The solution of van der Waerden's problem for permanents, *Adv. Math., 42* (1981), 299–305.

[99] G. P. Egorychev, *Integral Representation and the Computation of Combinatorial Sums*, Amer. Math. Soc. 1984.

[100] A. Erdélyi, *Asymptotic Expansions*, Dover reprint, 1956.

[101] A. Erdélyi, General asymptotic expansions of Laplace integrals, *Arch. Rational Mech. Anal.*, *7* (1961), pp. 1–20.

[102] A. Erdélyi and M. Wyman, The asymptotic evaluation of certain integrals, *Arch. Rational Mech. Anal.*, *14* (1963), pp. 217–260.

[103] P. Erdös, On some asymptotic formulas in the theory of 'Factorisatio numerorum', *Annals Math., 42* (1941), 989–993. (Corrections: *44* (1943), 647–651.)

[104] P. Erdös, A. Hildebrand, A. Odlyzko, P. Pudaite, and B. Reznick, The asymptotic behavior of a family of sequences, *Pacific J. Math., 126* (1987), pp. 227–241.

[105] P. Erdös and J. Lehner, The distribution of the number of summands in the partitions of a positive integer, *Duke Math. J., 8* (1941), 335–345.

[106] P. Erdös and J. H. Loxton, Some problems in partitio numerorum, *J. Austral. Math. Soc. (Ser. A) 27* (1979), 319–331.

[107] P. Erdös and B. Richmond, Concerning periodicity in the asymptotic behavior of partition functions, *J. Austral. Math. Soc. A 21* (1976), 447–456.

[108] P. Erdös and J. Spencer, *Probabilistic Methods in Combinatorics*, Academic Press and Akadémiai Kiado, New York, 1974.

[109] P. Erdös and P. Turán, On some problems of a statistical group–theory, I–IV; I: *Z. Wahrscheinlichkeitstheorie u. verw. Gebiete, 4* (1965), pp. 175–186; II–IV: *Acta Math. Acad. Sci. Hungar., 18* (1967), pp. 151–163 and 309–320, *19* (1968), pp. 413–435.

[110] M. A. Evgrafov, *Asymptotic Estimates and Entire Functions*, Gordon and Breach, New York, 1961.

[111] M. A. Evgrafov, *Analytic Functions*, Dover, New York, 1966.

[112] M. A. Evgrafov, Series and integral representations, pp. 1–81 in *Analysis I*, R. V. Gamkrelidze, ed., Springer 1989.

[113] D. I. Falikman, Proof of the van der Waerden conjecture on the permanent of a doubly stochastic matrix, *Mat. Zametki 29* (1981), 931–938. (In Russian.)

[114] M. V. Fedoryuk, *Asymptotics: Integrals and Series*, Nauka, Moscow 1987. (In Russian.)

[115] M. V. Fedoryuk, Asymptotic methods in analysis, pp. 83–191 in *Analysis I*, R. V. Gamkrelidze, ed., Springer 1989.

[116] M. V. Fedoryuk, Integral transforms, pp. 193–232 in *Analysis I*, R. V. Gamkrelidze, ed., Springer 1989.

[117] W. Feller, *An Introduction to Probability Theory*, vol. I, 3rd ed., vol. II, 2nd ed., John Wiley, New York, 1968, 1971.

[118] J. L. Fields, A uniform treatment of Darboux's method, *Arch. Rational Mech. Anal.*, *27* (1968), pp. 289–305.

[119] P. C. Fishburn and A. M. Odlyzko, Unique subjective probability on finite sets, *J. Ramanujan Math. Soc., 4* (1989), pp. 1–23.

[120] P. C. Fishburn, A. M. Odlyzko, and F. S. Roberts, Two–sided generalized Fibonacci sequences, *Fibonacci Quart., 27* (1989), pp. 352–361.

[121] P. Flajolet, Analyse d'algorithmes de manipulation de fichiers, *Institut de Recherche en Informatique et en Automatique*, No. 321, 1978.

[122] P. Flajolet, Combinatorial aspects of continued fractions, *Discrete Math.*, *32* (1980), pp. 125–161.

[123] P. Flajolet, Mathematical methods in the analysis of algorithms and data structures, *Trends in Theoretical Computer Science*, pp. 225–304, Egon Börger, ed., Computer Science Press, 1988.

[124] P. Flajolet, Analytic analysis of algorithms, *Proc. ICALP '92*, Springer Lecture Notes in Computer Science, 1992, to be published.

[125] P. Flajolet and J. Françon, Elliptic functions, continued fractions and doubled permutations, *European J. Combinatorics*, *10* (1989), pp. 235–241.

[126] P. Flajolet, Z. Gao, A. M. Odlyzko, and B. Richmond, The height of binary trees and other simple trees, *Combinatorics, Probability, and Computing* (1993), to appear.

[127] P. Flajolet, G. Gonnet, C. Puech, and J. M. Robson, The analysis of multidimensional searching in quad-trees, pp. 100–109 in *Proc. 2nd ACM-SIAM Symp. Discrete Algorithms*, SIAM, 1991.

[128] P. Flajolet, G. Gonnet, C. Puech, and J. M. Robson, Analytic variations on quadtrees, *Algorithmica*, to appear.

[129] P. Flajolet, P. Grabner, P. Kirschenhofer, H. Prodinger, and R. F. Tichy, Mellin transforms and asymptotics: digital sums, *Theoretical Comp. Sci.*, to appear.

[130] P. Flajolet, P. Kirschenhofer, and R. Tichy, Deviations from normality in random strings, *Probability Theory and Related Fields, 80* (1988), 139–150.

[131] P. Flajolet and T. Lafforgue, Search costs in quadtrees and singularity perturbation asymptotics, to be published.

[132] P. Flajolet and A. M. Odlyzko, The average height of binary trees and other simple trees, *J. Comput. System Sci., 25* (1982), pp. 171–213.

[133] P. Flajolet and A. M. Odlyzko, Limit distributions for coefficients of iterates of polynomials with application to combinatorial enumeration, *Math. Proc. Cambridge Phil. Soc., 96* (1984), pp. 237–253.

[134] P. Flajolet and A. M. Odlyzko, Random mapping statistics, in *Advances in Cryptology: Proceedings of Eurocrypt '89*, J–J. Quisquater, ed., Springer Lecture Notes in Computer Science, *434* (1990), pp. 329–354.

[135] P. Flajolet and A. M. Odlyzko, Singularity analysis of generating function, *SIAM J. Discrete Math., 3* (1990), pp. 216–240.

[136] P. Flajolet, J.–C. Raoult, and J. Vuillemin, The number of registers required to evaluate arithmetic expressions, *Theoretical Computer Science*, *9* (1979), pp. 99–125.

[137] P. Flajolet, M. Régnier, and R. Sedgewick, Some uses of the Mellin integral transform in the analysis of algorithms, in *Combinatorial Algorithms on Words*, A. Apostolico and Z. Galil, eds., Springer, 1985, pp. 241–254.

[138] P. Flajolet and B. Richmond, Generalized digital trees and their difference-differential equations, *Random Structures Algor. 3* (1992), 305–320.

[139] P. Flajolet, B. Salvy, and P. Zimmermann, Automatic average–case analysis of algorithms, *Theoretical Computer Science*, *79* (1991), 37–109.

[140] P. Flajolet and R. Schott, Non–overlapping partitions, continued fractions, Bessel functions and a divergent series, *European J. Combinatorics*, *11* (1990).

[141] P. Flajolet and R. Sedgewick, Digital search trees revisited, *SIAM J. Comput., 15* (1986), 748–767.

[142] P. Flajolet and M. Soria, Gaussian limiting distributions for the number of components in combinatorial structures, *J. Combinatorial Theory,* Series A, *53* (1990), pp. 165–182.

[143] P. Flajolet and M. Soria, General combinatorial schemes with Gaussian limit distributions and exponential tails, *Discrete Math. 114* (1993), 159–180.

[144] L. Flatto, The longer queue model, *Prob. in Eng. Inform. Sci., 3* (1989), 537–559.

[145] L. Flatto and S. Hahn, Two parallel queues created by arrivals with two demands. I. *SIAM J. Appl. Math., 44* (1984), 1041–1053.

[146] G. W. Ford and G. E. Uhlenbeck, Combinatorial problems in the theory of graphs I, II, III, and IV, *Proc. Nat. Acad. Sci. U.S.A., 42* (1956), pp. 122–128, 203–208, 529–535 and *43* (1957), pp. 163–167. (Part II with R. Z. Norman.)

[147] M. L. Fredman and D. E. Knuth, Recurrence relations based on minimization, *J. Math. Anal. Appl., 48* (1974), 534–559.

[148] S. Friedland, A lower bound for the permanent of a doubly stochastic matrix, *Ann. Math. (2) 110* (1979), 167–176.

[149] A. Frieze, On the length of the longest monotone subsequence in a random permutation, *Ann. Appl. Prob., 1* (1991), 301–305.

[150] B. Fristedt, The structure of random partitions of large integers, *Trans. Amer. Math. Soc.*, *337* (1993), 703–735.

[151] H. Furstenberg, Algebraic function fields over finite fields, *J. Algebra, 7* (1967), pp. 271–272.

[152] J. Galambos, Bonferroni inequalities, *Annal. Prob., 5* (1977), 577–581.

[153] J. Galambos and Y. Xu, Some optimal bivariate Bonferroni-type bounds, *Proc. Amer. Math. Soc. 117* (1993), 523–528.

[154] T. H. Ganelius, *Tauberian Remainder Theorems*, Lecture Notes in Math. #232, Springer, 1971.

[155] Z. Gao and L. B. Richmond, Central and local limit theorems applied to asymptotic enumeration. IV: Multivariate generating functions, *J. Appl. Comp. Analysis*, *41* (1992), 177–186.

[156] D. Gardy, Méthodes de col et lois limites en analyse combinatoire, *Theoretical Computer Science, 94* (1992), 261–280.

[157] D. Gardy, Some results on the asymptotic behavior of coefficients of large powers of functions, to be published.

[158] D. Gardy and P. Solé, Saddle point techniques in asymptotic coding theory, pp. 75–81 in *Algebraic Coding*, Proc. 1st French-Soviet Workshop, 1991, G. Cohen, S. Litsyn, A. Lobstein, and G. Zémor, eds., Lecture Notes in Computer Science #573, Springer, 1992.

[159] A. M. Garsia and S. A. Joni, A new expansion for umbral operators and power series inversion, *Proc. Amer. Math. Soc., 64* (1977), 179–185.

[160] I. M. Gessel, Counting Latin rectangles, *Bull. Amer. Math. Soc., 16* (1987), 79-82.

[161] I. M. Gessel, Symmetric functions and P-recursiveness, *J. Comb. Theory (A) 53* (1990), 257–286.

[162] S. Getu, L. W. Shapiro, W.-J. Woan, and L. C. Woodson, How to guess a generating function, *SIAM J. Discrete Math., 5* (1992), 497–499.

[163] C. D. Godsil and B. D. McKay, Asymptotic enumeration of Latin rectangles, *J. Comb. Theory*, Series B, *48* (1990), pp. 19–44.

[164] W. M. Y. Goh and E. Schmutz, The expected order of a random permutation, to be published.

[165] W. M. Y. Goh and E. Schmutz, A central limit theorem on $GL_n(F_q)$, to be published.

[166] W. M. Y. Goh and E. Schmutz, Distribution of the number of distinct parts in a random partition, to be published.

[167] V. L. Goncharov, From the domain of combinatorial analysis, *Izv. Akad. Nauk SSSR Ser. Math., 8*, no. 1 (1944), 3–48. (In Russian. English translation in *Transl. Amer. Math. Soc., 19* (1962), 1–46.)

[168] G. H. Gonnet and R. Baeza-Yates, *Handbook of Algorithms and Data Structures*, 2nd ed., Addison-Wesley, 1991.

[169] I. J. Good, Generalizations to several variables of Lagrange's expansion, with applications to stochastic processes, *Proc. Cambridge Phil. Soc., 56* (1960), 367–380.

[170] B. Gordon and L. Houten, Notes on plane partitions. III, *Duke Math. J., 26* (1969), 801–824.

[171] R. W. Gosper, Jr., Decision procedure for indefinite hypergeometric summation, *Proc. Nat. Acad. Sci. USA 75* (1978), 40–42.

[172] H. W. Gould, *Combinatorial Identities*, 1972 (private printing).

[173] I. Goulden and D. Jackson, *Combinatorial Enumeration*, John Wiley, New York, 1983.

[174] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, 1965.

[175] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison Wesley, 1989.

[176] A. G. Greenberg, B. D. Lubachevsky, and A. M. Odlyzko, Simple, efficient asynchronous parallel algorithms for maximization, *ACM Trans. Programming Languages and Systems* (1988), pp. 313–337.

[177] D. H. Greene and D. E. Knuth, *Mathematics for the Analysis of Algorithms*, 2nd ed., Birkhäuser, Boston, 1982.

[178] J. R. Griggs, P. Hanlon, A. M. Odlyzko, and M. S. Waterman, On the number of alignments of k sequences, *Graphs and Combinatorics, 6* (1990), pp. 133–146.

[179] E. Grosswald, Generalization of a formula of Hayman and its application to the study of Riemann's zeta function, *Illinois J. Math.*, *10* (1966), pp. 9–23. Correction in *13* (1969), pp. 276–280.

[180] L. J. Guibas and A. M. Odlyzko, Maximal prefix–synchronized codes, *SIAM J. Appl. Math., 35* (1978), pp. 401–418.

[181] L. J. Guibas and A. M. Odlyzko, Long repetitive patterns in random sequences, *Z. Wahrscheinlichkeitstheorie u. verwandte Geb., 53* (1980), pp. 241–262.

[182] L. J. Guibas and A. M. Odlyzko, String overlaps, pattern matching, and nontransitive games, *J. Comb. Theory A, 30* (1981), pp. 183–208.

[183] W. J. Gutjahr, The variance of level numbers in certain families of trees, *Random Structures Alg. 3* (1992), 361–374.

[184] J. H. Halton, The properties of random trees, *Information Sciences 47* (1989), 95–133.

[185] R. A. Handelsman and J. S. Lew, Asymptotic expansion of Laplace transforms near the origin, *SIAM J. Math. Analysis*, *1* (1970).

[186] E. R. Hansen, *A Table of Series and Products*, Prentice-Hall, 1975.

[187] J. Hansen, Order statistics for decomposable combinatorial structures, *Rand. Struct. Alg.*, to appear.

[188] F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic Press, 1973.

[189] F. Harary, R. W. Robinson, and A. J. Schwenk, Twenty–step algorithm for determining the asymptotic number of trees of various species, *J. Austral. Math. Soc.* (Series A), *20* (1975), pp. 483–503.

[190] G. H. Hardy, *Divergent Series*, Oxford University Press, London, 1949.

[191] G. H. Hardy and J. E. Littlewood, Tauberian theorems concerning power series and Dirichlet's series whose coefficients are positive, *Proc. London Math. Soc. (2) 13* (1914), 174-191. Reprinted in *Collected Papers of G. H. Hardy*, vol. 6, pp. 510–527.

[192] G. H. Hardy and J. E. Littlewood, Some theorems concerning Dirichlet's series, *Messenger Math., 43* (1914), 134-147. Reprinted in *Collected Papers of G. H. Hardy*, vol. 6, pp. 542–555.

[193] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, 2nd ed., Cambridge Univ. Press, 1952.

[194] G. H. Hardy and S. Ramanujan, Asymptotic formulae for the distribution of integers of various types, *Proc. London Math. Soc. (2) 16* (1917), 112-132. Reprinted in *Collected Papers of G. H. Hardy*, vol. 1, pp. 277–293.

[195] L. H. Harper, Stirling behavior is asymptotically normal, *Ann. Math. Stat., 38* (1967), 410–414.

[196] B. Harris, Probability distributions related to random mappings, *Ann. Math. Statist., 31* (1960), pp. 1042–1062.

[197] B. Harris and C. J. Park, The distribution of linear combinations of the sample occupancy numbers, *Nederl. Akad. Wetensch. Proc. Ser. A, 74 = Indag. Math., 33* (1971), pp. 121–134.

[198] B. Harris and L. Schoenfeld, Asymptotic expansions for the coefficients of analytic functions, *Illinois J. Math., 12* (1968), pp. 264–277.

[199] T. E. Harris, *The Theory of Branching Processes*, Springer, 1963.

[200] W. A. Harris and Y. Sibuya, Asymptotic solutions of systems of nonlinear difference equations, *Arch. Rational Mech. Anal., 15* (1964), 277–395.

[201] W. A. Harris and Y. Sibuya, General solution of nonlinear difference equations, *Trans. Amer. Math. Soc., 115* (1965), 62–75.

[202] C. B. Haselgrove and H. N. V. Temperley, Asymptotic formulae in the theory of partitions, *Proc. Cambridge Phil. Soc.,50* (1954), 225–241.

[203] M. L. J. Hautus and D. A. Klarner, The diagonals of a double power series, *Duke Math. J., 38* (1971), 229–235.

[204] W. K. Hayman, A generalization of Stirling's formula, *J. reine angew. Math., 196* (1956), pp. 67–95.

[205] P. Henrici, *Applied and Computational Complex Analysis*, Wiley: Vol. 1, 1974; Vol. 2, 1977; Vol. 3, 1986.

[206] E. Hille, *Lectures on Ordinary Differential Equations*, Addison-Wesley, 1969.

[207] E. Hille, *Ordinary Differential Equations in the Complex Domain*, Wiley, 1976.

[208] J. J. Hofbauer, A short proof of the Lagrange-Good formula, *Discrete Math., 25* (1979), 135–139.

[209] M. Hofri, *Probabilistic Analysis of Algorithms*, Springer, 1987.

[210] C. Hunter, Asymptotic solutions of certain linear difference equations, with applications to some eigenvalue problems, *J. Math. Anal. Appl., 24* (1968), pp. 279–289.

[211] G. K. Immink, *Asymptotics of Analytic Difference Equations*, Lecture Notes in Math. #1085, Springer, 1984.

[212] A. E. Ingham, A Tauberian theorem for partitions, *Ann. of Math., 42* (1941), pp. 1075–1090.

[213] P. Jacquet and M. Régnier, Trie partitioning process: limiting distributions, pp. 196–210 in *CAAP '86*, P. Franchi-Zannettacci, ed., Lecture Notes in Computer Science #214, Springer, 1986.

[214] P. Jacquet and M. Régnier, Normal limiting distribution of the size of tries, pp. 209–223 in *Performance '87*, P.-J. Courtois and G. Latouche, eds., North-Holland, 1988.

[215] P. Jacquet and M. Régnier, Normal limiting distribution for the size and the external path length of tries, in preparation.

[216] L. B. W. Jolley, *Summation of Series*, 2nd ed., Dover, 1961.

[217] A. T. Jonassen and D. E. Knuth, A trivial algorithm whose analysis is not, *J. Comp. Sys. Sci.*, *16* (1978), pp. 301–322.

[218] W. B. Jones and W. J. Thron, *Continued Fractions: Analytic Theory and Applications*, Addison-Wesley, 1980.

[219] R. Jungen, Sur les séries de Taylor n'ayant que des singularitiés algébrico–logarithmiques sur leur cercle de convergence, *Comment. Math. Helv., 3* (1931), pp. 266–306.

[220] S. Kapoor and E. M. Reingold, Recurrence relations based on minimization and maximization, *J. Math. Anal. Appl., 109* (1985), 591–604.

[221] R. M. Karp, Probabilistic recurrence relations, *Proc. 23rd ACM Symp. Theory of Computing*, 1991, pp. 190–197.

[222] S. Karlin, *Total Positivity, Vol. 1*, Stanford Univ. Press, 1968.

[223] R. Kemp, *Fundamentals of the Average Case Analysis of Particular Algorithms*, Wiley, 1984.

[224] R. Kemp, A note on the number of leftist trees, *Inform. Proc. Letters 25* (1987), 227–232.

[225] R. Kemp, Further results on leftist trees, pp. 103–130 in *Random Graphs '87*, M. Karonski, J. Jaworski, and A. Rucinski, eds., Wiley, 1990.

[226] H. Kesten, *Percolation Theory for Mathematicians*, Birkhäuser, 1982.

[227] P. Kirschenhofer, A tree enumeration problem involving the asymptotics of the 'diagonals' of a power series, *Ann. Discrete Math. 33* (1987), 157–170.

[228] P. Kirschenhofer and H. Prodinger, On some applications of formulae of Ramanujan in the analysis of algorithms, *Mathematika, 38* (1991), 14–33.

[229] D. A. Klarner, A combinatorial formula involving the Fredholm integral equation, *J. Combinatorial Theory, 5* (1968), pp. 59–74.

[230] D. A. Klarner and R. L. Rivest, Asymptotic bounds for the number of convex n-ominoes, *Discrete Math., 8* (1974), 31–40.

[231] C. Knessl and J. B. Keller, Partition asymptotics for recursion equations, *SIAM J. Appl. Math., 50* (1990), 323–338.

[232] C. Knessl and J. B. Keller, Stirling number asymptotics from recursion equations using the ray method, *Studies Appl. Math., 84* (1991), 43–56.

[233] A. Knopfmacher, A. Odlyzko, B. Richmond, G. Szekeres, and N. Wormald, manuscript in preparation.

[234] K. Knopp, *Theory and Application of Infinite Series*, 2nd ed., reprinted by Hafner, 1971.

[235] D. E. Knuth, *The Art of Computer Programming Vol. 1*: *Fundamental Algorithms*, 2nd ed., Addison–Wesley, Reading, 1973.

[236] D. E. Knuth, *The Art of Computer Programming Vol. 2*: *Semi–Numerical Algorithms*, 2nd ed., Addison–Wesley, Reading, 1981.

[237] D. E. Knuth, *The Art of Computer Programming Vol. 3*: *Sorting and Searching*, Addison–Wesley, Reading, 1973.

[238] D. E. Knuth and B. Pittel, A recurrence related to trees, *Proc. Amer. Math. Soc., 105* (1989), 335–349.

[239] D. E. Knuth and A. Schönhage, The expected linearity of a simple equivalence algorithm, *Theoretical Comp. Sci., 6* (1978), 281–315.

[240] V. F. Kolchin, *Random Mappings*, Optimization Software Inc., New York, 1986.

[241] V. F. Kolchin, B. A. Sevast'yanov, and V. P. Chistyakov, *Random Allocations*, Wiley, 1978.

[242] J. Komlos, A. M. Odlyzko, L. H. Ozarow, and L. A. Shepp, On the properties of a tree–structured server process, *Ann. Appl. Prob., 1* (1990), 118–125.

[243] R. J. Kooman, *Convergence Properties of Recurrence Sequences*, Ph.D. Dissertation, Leiden, 1989.

[244] R. J. Kooman and R. Tijdeman, Convergence properties of linear recurrence sequences, *Nieuw Archief Wisk., Ser. 4, 4* (1990), 13–25.

[245] M. D. Kruskal, The expected number of components under a random mapping function, *Amer. Math. Monthly, 61* (1954), pp. 392–397.

[246] M. Kuczma, *Functional Equations in a Single Variable*, Polish Scientific Publishers, Warsaw, 1968.

[247] G. Labelle, Une nouvelle démonstration combinatoire des formules d'inversion de Lagrange, *Adv. Math., 42* (1981), 217–247.

[248] J. C. Lagarias, A. M. Odlyzko, and D. B. Zagier, On the capacity of disjointly shared networks, *Computer Networks and ISDN Systems, 10* (1985), pp. 275–285.

[249] M.-Y. Lee, Bivariate Bonferroni inequalities, *Aequationes Math. 44* (1992), 220–225.

[250] J. Leray, Le calcul différentiel et intégral sur une variété analytique complexe, *Bull. Soc. Math. France 87* (1959), 81–180.

[251] L. Lewin, *Polylogarithms and Associated Functions*, North Holland, 1981.

[252] B. Lichtin, The asymptotics of a lattice point problem associated to a finite number of polynomials. I, *Duke Math. J., 63* (1991), 139–192.

[253] L. Lipshitz, The diagonal of a D–finite power series is D–finite, *J. Algebra*, *113* (1988), pp. 373–378.

[254] L. Lipshitz, D–Finite Power Series, *J. Algebra*, *122* (1989), pp. 353–373.

[255] L. Lipshitz and A. van der Poorten, Rational functions, diagonals, automata and arithmetic, in *Number Theory*, Richard A. Mollin, ed., Walter de Gruyter, Berlin, 1990, pp. 339–358.

[256] B. F. Logan, J. E. Mazo, A. M. Odlyzko, and L. A. Shepp, On the average product of Gauss–Markov variables, *Bell System Tech. J., 62* (1983), pp. 2993–3006.

[257] B. F. Logan and L. A. Shepp, A variational problem for random Young tableaux, *Advances Math., 26* (1977), 206–222.

[258] G. Louchard, The Brownian motion: a neglected tool for the complexity analysis of sorted table manipulation, *RAIRO Theoretical Informatics*, *17* (1983), pp. 365–385.

[259] G. Louchard, The Brownian excursion: a numerical analysis, *Computers and Mathematics with Applications*, *10* (1984), pp. 413–417.

[260] G. Louchard, Brownian motion and algorithm complexity, *BIT 26* (1986), 17–34.

[261] G. Louchard, Exact and asymptotic distributions in digital and binary search trees, *RAIRO informatique théorique et applications*, *21* (1987), pp. 479–495.

[262] G. Louchard, B. Randrianarimanana, and R. Schott, Dynamic algorithms in D. E. Knuth's model; a probabilistic analysis, *Theoretical Comp. Sci., 93* (1992), 201–255.

[263] T. Luczak, The number of trees with a large diameter, to be published.

[264] G. S. Lueker, Some techniques for solving recurrences, *Computing Surveys, 12* (1980), 419–436.

[265] A. J. Macintyre and R. Wilson, Operational methods and the coefficients of certain power series, *Math. Ann., 127* (1954), 243–250.

[266] K. Mahler, On a special functional equation, *J. London Math. Soc. 15* (1940), pp. 115–123.

[267] K. Mahler, On a class of nonlinear functional equations connected with modular functions, *J. Austral. Math. Soc. Ser. A 22* (1976), 65–118.

[268] K. Mahler, On a special nonlinear functional equation, *Proc. Roy. Soc. London Ser. A 378* (1981), 155–178.

[269] K. Mahler, On the analytic relation of certain functional and difference equations, *Proc. Roy. Soc. London Ser. A 389* (1983), 1–13.

[270] H. S. Mahmoud, *Evolution of Random Search Trees*, Wiley, 1992.

[271] H. M. Mahmoud and B. Pittel, Analysis of the space of search trees under the random insertion algorithm, *J. Algorithms*, *10* (1989), pp. 52–75.

[272] B. Malgrange, Sur les points singuliers des équations différentielles, *L'Enseign. Math., 20* (1974), 147–176.

[273] C. L. Mallows, A. M. Odlyzko, and N. J. A. Sloane, Upper bounds for modular form, lattices, and codes, *J. Algebra, 36* (1975), 68–76.

[274] V. A. Malyshev, An analytic method in the theory of two-dimensional positive random walks, *Sibir. Mat. Zh., 13* (1972), 1314–1329 (in Russian).

[275] A. Maté and P. Nevai, Sublinear perturbations of the differential equation $y^{(n)} = 0$ and of the analogous difference equation, *J. Diff. Equations 53* (1984), 234–257.

[276] A. Maté and P. Nevai, Asymptotics for solutions of smooth recurrence relations, *Proc. Amer. Math. Soc., 93* (1985), 423–429.

[277] J. E. Mazo and A. M. Odlyzko, Lattice points in high–dimensional spheres, *Monatsh. Math., 110* (1990), pp. 47–61.

[278] B. D. McKay, The asymptotic numbers of regular tournaments, eulerian digraphs, and eulerian and oriented graphs, *Combinatorica, 10* (1990), 367–377.

[279] B. D. McKay and N. C. Wormald, Asymptotic enumeration by degree sequence of graphs of high degree, *European J. Combinatorics, 11* (1990), 565–580.

[280] G. Meinardus, Asymptotische Aussagen über Partitionen, *Math. Z., 59* (1954), pp. 388–398.

[281] A. Meir and J. W. Moon, On the altitude of nodes in random trees, *Canadian J. Math.*, *30* (1978), pp. 997–1015.

[282] A. Meir and J. W. Moon, On random mapping patterns, *Combinatorica*, *4* (1984), pp. 61–70.

[283] A. Meir and J. W. Moon, Some asymptotic results useful in enumeration problems, *Aequationes Math., 33* (1987), 260–268.

[284] A. Meir and J. W. Moon, On an asymptotic method in enumeration, *J. Comb. Theory*, Series A, *51* (1989), pp. 77–89.

[285] A. Meir and J. W. Moon, The asymptotic behavior of coefficients of powers of certain generating functions, *European J. Comb., 11* (1990), 581–587.

[286] N. S. Mendelsohn, The asymptotic series for a certain class of permutation problems, *Canad. J. Math., 8* (1956), pp. 234–244.

[287] L. M. Milne-Thomson, *The Calculus of Finite Differences*, MacMillan, 1933.

[288] D. S. Mitrinović, *Analytic Inequalities*, Springer, 1970.

[289] D. Moews, Explicit Tauberian bounds for multivariate functions, to be published.

[290] J. W. Moon, Counting labeled trees, *Canad. Math. Monograph No. 1, Canad. Math. Congress*, 1970.

[291] J. W. Moon, Some enumeration results on series-parallel networks, *Annals Discrete Math., 33* (1987), 199–226. (*Random Graphs '85*, M. Karonski and Z. Palka, eds., North-Holland 1987.)

[292] L. Moser and M. Wyman, On the solutions of $x^d = 1$ in symmetric groups, *Canad. J. Math., 7* (1955), pp. 159–168.

[293] L. Moser and M. Wyman, Asymptotic expansions, *Canadian J. Math.*, *8* (1956), pp. 225–233.

[294] L. Moser and M. Wyman, Asymptotic expansions II, *Canadian Journal of Math.*, (1957), pp. 194–209.

[295] L. Moser and M. Wyman, Stirling numbers of the second kind, *Duke Math. J., 25* (1958), 29–43.

[296] L. Moser and M. Wyman, Asymptotic development of the Stirling numbers of the first kind, *J. London Math. Soc., 33* (1958), 133–146.

[297] National Bureau of Standards, *Handbook of Mathematical Functions*, M. Abramowitz and I. A. Stegun, eds., U.S. Gov. Printing Office, 9th printing, 1970.

[298] N. E. Nörlund, *Vorlesungen über Differenzenrechnung*, Springer, 1924. Dover reprint, 1954.

[299] F. Oberhettinger, *Tables of Mellin Transforms*, Springer, 1974.

[300] A. M. Odlyzko, Periodic oscillations of coefficients of power series that satisfy functional equations, *Adv. Math., 44* (1982), pp. 180–205.

[301] A. M. Odlyzko, Some new methods and results in tree enumeration, *Congressus Numerantium, 42* (1984), pp. 27–52.

[302] A. M. Odlyzko, On heights of monotonically labelled binary trees, *Congressus Numerantium, 44* (1985), pp. 305–314.

[303] A. M. Odlyzko, Enumeration of strings, in *Combinatorial Algorithms on Words*, A. Apostolico and Z. Galil, eds., Springer, 1985, pp. 205–228.

[304] A. M. Odlyzko, Applications of symbolic mathematics to mathematics, pp. 95–111 in *Applications of Computer Algebra*, R. Pavalle, ed., Kluwer, 1985.

[305] A. M. Odlyzko, Explicit Tauberian estimates for functions with positive coefficients, *J. Comput. Appl. Math., 41* (1992), 187–197.

[306] A. M. Odlyzko, B. Poonen, H. Widom, and H. S. Wilf, manuscript in preparation.

[307] A. M. Odlyzko and L. B. Richmond, *On the Compositions of an Integer, Combinatorial Mathematics VII*, R.–W. Robinson, G. W. Southern, and W. D. Wallis, eds., Springer–Verlag Lecture Notes in Mathematics #829, 1980, pp. 119–210.

[308] A. M. Odlyzko and L. B. Richmond, On the unimodality of some partition polynomials, *European J. Combinatorics, 3* (1982), pp. 69–84.

[309] A. M. Odlyzko and L. B. Richmond, On the unimodality of high convolutions of discrete distributions *Ann. Prob., 13* (1985), pp. 299–306.

[310] A. M. Odlyzko and L. B. Richmond, On the number of distinct block sizes in partitions of a set, *J. Combinatorial Theory A, 38* (1985) pp. 170–181.

[311] A. M. Odlyzko and L. B. Richmond, Asymptotic expansions for the coefficients of analytic generating functions, *Aequationes Math., 28* (1985), pp. 50–63.

[312] A. M. Odlyzko and H. S. Wilf, Bandwidths and profiles of trees, *J. Combinatorial Theory B, 42* (1987), pp. 348–370. (Condensed summary of results in *Graph Theory and its Applications to Algorithms and Computer Science*, Y. Alavi et al., eds., Wiley, 1985, pp. 605–622.)

[313] A. M. Odlyzko and H. S. Wilf, The editor's corner: n coins in a fountain, *Amer. Math. Monthly, 95* (1988), pp. 840–843.

[314] A. M. Odlyzko and H. S. Wilf, Functional iteration and the Josephus problem, *Glasgow Math. J., 33* (1991), pp. 235–240.

[315] F. W. J. Olver, *Asymptotics and Special Functions*, Academic Press, New York, 1974.

[316] R. Otter, The number of trees, *Ann. of Math., 49* (1948), pp. 583–599.

[317] A. I. Pavlov, On the number of substitutions with cycle lengths from a given set, *Discrete Appl. Math. 2* (1992), 445–459.

[318] S. G. Penrice, Derangements, permanents, and Christmas presents, *Amer. Math. Monthly, 98* (1991), 617–620.

[319] O. Perron, *Die Lehre von den Kettenbrüchen*, Chelsea reprint.

[320] G. Pólya, Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen, *Acta Math., 68* (1937), pp. 145–254.

[321] G. Pólya, On the number of certain lattice polygons, *J. Combinatorial Theory 6* (1969), 102–105.

[322] G. Pólya and R. C. Read, *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*, Springer, 1987.

[323] G. Pólya and G. Szegö, *Problems and Theorems in Analysis*, 2 volumes, English translation, Springer, 1972 and 1976.

[324] A. Popken, Asymptotic expansions from an algebraic standpoint, *Indagationes Math.*, *15* (1953), pp. 131–143.

[325] A. G. Postnikov, Tauberian theory and its applications, *Proc. Steklov Inst. Math., 144*; English translation, *Amer. Math. Soc. Transl.* (1980).

[326] V. Privman and N. M. Svrakic, Difference equations in statistical mechanics: I. Cluster statistics models and II. Solid-on-solid models in two dimensions, *J. Stat. Phys. 51* (1988), 1091–1110 and 1111–1126.

[327] V. Privman and N. M. Svrakic, *Directed Models of Polymers, Interfaces, and Clusters: Scaling and Finite-Size Properties*, Lecture Notes in Physics #338, Springer, 1989.

[328] H. Rademacher, On the partition function, *Proc. London Math. Soc., 43* (1937), pp. 241–254.

[329] A. Regev, Asymptotic values for degrees associated with strips of Young diagrams, *Advances Math., 41* (1981), 115–136.

[330] A. Rényi, Three more proofs and a generalization of a theorem of Irving Weiss, *Magyar Tud. Akad. Mat. Kutató Int. Közl., 7* (1962), 203–214.

[331] A. Rényi and G. Szekeres, On the height of trees, *J. Austral. Math. Soc., 7* (1967), pp. 497–507.

[332] L. B. Richmond, Asymptotic relations for partitions, *J. Number Theory, 4* (1975), 389–405.

[333] L. B. Richmond, The moments of partitions. II, *Acta Arith., 28* (1975), 229–243.

[334] L. B. Richmond, Asymptotic relations for partitions, *Trans. Amer. Math. Soc., 219* (1976), 379–385.

[335] J. Riordan, *Introduction to Combinatorial Analysis*, John Wiley, New York, 1958.

[336] J. Riordan, *Combinatorial Identities*, Wiley, 1968.

[337] K. F. Roth and G. Szekeres, Some asymptotic formulae in the theory of partitions, *Quart. J. Math. Oxford Ser., 5* (1954), pp. 241–259.

[338] V. N. Sachkov, *Probabilistic Methods in Combinatorial Analysis* (in Russian), Nauka, Moscow, 1978.

[339] E. Schmutz, Asymptotic expansions for the coefficients of $e^{P(z)}$, *Bull. London Math. Soc. 21* (1989), pp. 482–486.

[340] A. Schrijver, A short proof of Minc's conjecture, *J. Comb. Theory (A), 25* (1978), 80–83.

[341] R. Sedgewick, Data movement in odd–even merging, *SIAM J. Comput., 7* (1978), pp. 239–272.

[342] L. A. Shepp and S. P. Lloyd, Ordered cycle lengths in a random permutation, *Trans. Amer. Math. Soc., 121* (1966), pp. 340–357.

[343] J. A. Shohat and J. D. Tamarkin, *The Problem of Moments*, Amer. Math. Soc., 1943.

[344] L. Sirovich, *Techniques of Asymptotic Analysis*, Springer Verlag, 1971.

[345] N. J. A. Sloane, *A Handbook of Integer Sequences*, Academic Press, 1973. A revised and expanded edition is in press.

[346] N. J. A. Sloane, *The New Book of Integer Sequences*, Freeman, 1994, to be published.

[347] I. N. Sneddon, *The Uses of Integral Transforms*, McGraw, 1972.

[348] J. Spencer, *Ten Lectures on the Probabilistic Method*, SIAM, 1987.

[349] R. P. Stanley, Generating Functions, in *Studies in Combinatorics*, M.A.A. Studies in Mathematics, Vol. 17., G–C. Rota, ed., Math. Ass. of America, 1978, pp. 100–141.

[350] R. P. Stanley, Differentiably finite power series, *European J. Combinatorics, 1* (1980), 175–188.

[351] R. P. Stanley, *Enumerative Combinatorics*, Wadsworth and Brooks/Cole, Monterey, 1986.

[352] R. P. Stanley, Log-concave and unimodal sequences in algebra, combinatorics, and geometry, pp. 500–535 in *Graph Theory and its Applications: East and West*, *Annals New York Acad. Sci.*, no. 576, 1989.

[353] K. B. Stolarsky, Power and exponential sums of digital sums related to binomial coefficient parity, *SIAM J. Appl. Math., 32* (1977), 717–730.

[354] G. Szegö, *Orthogonal Polynomials*, Amer. Math. Soc. Coll. Publ., vol. 23, rev. ed., Amer. Math. Soc., New York, 1959.

[355] G. Szekeres, Some asymptotic formulae in the theory of partitions. II *Quart. J. Math. Oxford*, Ser. 2, *4* (1953), pp. 96–111.

[356] G. Szekeres, Regular iteration of real and complex functions, *Acta Math.*, *100* (1958), pp. 103–258.

[357] G. Szekeres, Distribution of labelled trees by diameter, pp. 392–397 in *Combinatorial Mathematics X*, Proc. 10-th Australian Conf. Comb. Math., Lecture Notes in Mathematics, Springer, 1982.

[358] G. Szekeres, Asymptotic distribution of the number and size of parts in unequal partitions, *Bull. Australian Math. Soc. 36* (1987), 89–97.

[359] G. Szekeres, Asymptotic distribution of partitions by number and size of parts, pp. 527–538 in *Number Theory*, vol. I, K. György and G. Halász, eds., Colloq. Math. Soc. J. Bolyai, No. 51, North-Holland, 1990.

[360] W. Szpankowski, The evaluation of an alternative sum with applications to the analysis of some data structures, *Inform. Proc. Letters, 28* (1988), 13–19.

[361] L. Takács, On the number of distinct forests, *SIAM J. Discr. Math., 3* (1990), 574–581.

[362] L. Takács, A Bernoulli excursion and its various applications, *Adv. Appl. Prob., 23* (1991), 557–585.

[363] N. M. Temme, Asymptotic estimates of Stirling numbers, *Studies Appl. Math. 89* (1993), 233–243.

[364] E. C. Titchmarsh, *The Theory of Functions*, 2nd ed., Oxford University Press, London, 1939.

[365] E. C. Titchmarsh, *Fourier Integrals*, 2nd ed., Oxford Univ. Press, 1948.

[366] W. J. Trjitzinsky, Analytic theory of linear q-difference equations, *Acta Math., 61* (1933), 1–38.

[367] W. J. Trjitzinsky, Analytic theory of linear differential equations, *Acta math., 62* (1933), 167–226.

[368] V. S. Varadarajan, Meromorphic differential equations, *Expositiones Math.. 9* (1991), 97–188.

[369] R. C. Vaughan, *The Hardy-Littlewood Method*, Cambridge Univ. Press, 1981.

[370] A. M. Vershik and C. V. Kerov, Asymptotics of the Plancherel measure of the symmetric group and a limiting form for Young tableau, *Dokl. Akad. Nauk USSR, 233* (1977), 1024–1027. (In Russian.)

[371] J. Vitter and P. Flajolet, Analysis of algorithms and data structures, *Handbook of Theoretical Computer Science*, Vol. A: Algorithms and Complexity, Ch. 9, pp. 432–524, J. Van Leeuwen, ed., North Holland, 1990.

[372] W. Wasow, *Asymptotic Expansions for Ordinary Differential Equations*, Wiley, 1965.

[373] W. D. Wei, Y. Z. Cai, C. L. Liu, and A. M. Odlyzko, Balloting labelling and personnel assignment, *SIAM J. Alg. Discr. Methods, 7* (1986), pp. 150–158

[374] E. A. Whitehead, Jr., *Four-discordant permutations*, J. Austral. Math. Soc. (Ser. A) 28 (1979), 369–377.

[375] E. T. Whittaker and G. N. Watson, *A Course of Modern Analysis*, 4th ed., Cambridge University Press, Cambridge, 1927.

[376] H. S. Wilf, The asymptotics of $e^{P(z)}$ and the number of elements of each order in S_n, *Bull. Am. Math. Soc.*, *15* (1986), pp. 228–232.

[377] H. S. Wilf, *Generatingfunctionology*, Academic Press, 1990.

[378] H. S. Wilf, The asymptotic behavior of the Stirling numbers of the first kind, *J. Comb. Theory Ser. A*, to appear.

[379] H. S. Wilf and D. Zeilberger, Rational functions certify combinatorial identities, *J. Amer. Math. Soc.*, *3* (1990), pp. 147–158.

[380] H. S. Wilf and D. Zeilberger, An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities, *Inventiones math., 108* (1992), 575–633.

[381] R. Wilson, The coefficient theory of integral functions with dominant exponential parts, *Quart. J. Math. Oxford, ser. 2, 4* (1953), 142–149.

[382] J. Wimp, *Computation with Recurrence Relations*, Pitman, Boston, 1984.

[383] J. Wimp, Current trends in asymptotics: some problems and some solutions, *J. Computational Appl. Math., 35* (1991), 53–79.

[384] J. Wimp and D. Zeilberger, Resurrecting the asymptotics of linear recurrences, *J. Math. Anal. Appl.*, *111* (1985), pp. 162–176.

[385] R. Wong, *Asymptotic Approximations of Integrals*, Academic Press, 1989.

[386] R. Wong and M. Wyman, The method of Darboux, *J. Approx. Theory, 10* (1974), pp. 159–171.

[387] E. M. Wright, Asymptotic partition formulae, I: Plane partitions, *Quart. J. Math. Oxford Ser., 2* (1931), pp. 177–189.

[388] E. M. Wright, The coefficients of a certain power series, *J. London Math. Soc.*, *7* (1932), pp. 256–262.

[389] E. M. Wright, On the coefficients of power series having exponential singularities, *J. London Math. Soc.*, *24* (1949), pp. 304–309.

[390] E. M. Wright, Partitions of large bipartities, *Amer. J. Math., 80* (1958), 643–658.

[391] E. M. Wright, The asymptotic behavior of the generating functions of partitions of multipartities, *Quart. J. Math. Oxford (2) 10* (1959), 60–69.

[392] E. M. Wright, Partitions into k parts, *Math. Annalen*, *142* (1961), pp. 311–316.

[393] E. M. Wright, A relationship between two sequences, *Proc. London Math. Soc. 17* (1967), 296–304, 547–552.

[394] E. M. Wright, Asymptotic relations between enumerative functions in graph theory, *Proc. London Math. Soc. (3), 20* (1970) pp. 558–572.

[395] E. M. Wright, Graphs on unlabelled nodes with a given number of edges, *Acta Math., 126* (1971), pp. 1–9.

[396] E. M. Wright, The number of strong digraphs, *Bull. London Math. Soc., 3* (1971), 348–350.

[397] E. M. Wright, Graphs on unlabelled nodes with a large number of edges, *Proc. London Math. Soc. (3), 28* (1974), 577–594.

[398] E. M. Wright and B. G. Yates, The asymptotic expansion of a certain integral, *Quarterly J. Mathematics Oxford*, *1* (1950) pp. 41–53.

[399] R. A. Wright, L. B. Richmond, A. M. Odlyzko, and B. D. McKay, Constant time generation of free trees, *SIAM J. Comp., 15* (1986), pp. 540–548.

[400] M. Wyman, The asymptotic behavior of the Laurent coefficients, *Canad. J. Math., 11* (1959), pp. 534–555.

[401] M. Wyman, The method of Laplace, *Trans. Royal Soc. Canada*, *2* (1964), pp. 227–256.

[402] D. Zeilberger, Solutions of exponential growth to systems of partial differential equations, *J. Differential Eq.*, *31* (1979), pp. 287–295.

[403] D. Zeilberger, The algebra of linear partial difference operators and its applications, *SIAM J. Math. Analysis*, *11* (1980), pp. 919–932.

[404] D. Zeilberger, Six etudes in generating functions, *Intern. J. Computer Math.*, *29* (1989), pp. 201–215.

[405] D. Zeilberger, A holonomic approach to special function identities, *J. Comput. Appl. Math., 32* (1990), 321–368.

Fig. 1. Domain $\Delta(r, \phi, \eta)$ of Section 11.1 and the integration contour Γ.

Contents

A STRENGTHENING OF THE ASSMUS-MATTSON THEOREM*

A. R. Calderbank

Mathematical Sciences Research Center
AT&T Bell Laboratories
Murray Hill, NJ 07974, USA

P. Delsarte

Philips Research Laboratories
Avenue Albert Einstein 4
B-1348 Louvain-la-Neuve, Belgium

N. J. A. Sloane

Mathematical Sciences Research Center
AT&T Bell Laboratories
Murray Hill, NJ 07974, USA

DEDICATED TO THE MEMORY OF JESSIE MACWILLIAMS (1917-1990)

ABSTRACT

Let $w_1 = d, w_2, \ldots, w_s$ be the weights of the nonzero codewords in a binary linear $[n, k, d]$ code C, and let $w_1', w_2', \ldots, w_{s'}'$ be the nonzero weights in the dual code $C^\perp$. Let t be an integer in the range $0 < t < d$ such that there are at most $d-t$ weights w_i' with $0 < w_i' \le n-t$. Assmus and Mattson proved that the words of any weight w_i in C form a t-design. We show that if $w_2 \ge d+4$ then either the words of any nonzero weight w_i form a $(t+1)$-design or else the codewords of minimal weight d form a $\{1, 2, \ldots, t, t+2\}$-design. If in addition C is self-dual with all weights divisible by 4 then the codewords of any given weight w_i form either a $(t+1)$-design or a $\{1, 2, \ldots, t, t+2\}$-design. The special case of this result for codewords of minimal weight in an extremal self-dual code with all weights divisible by 4 also follows from a theorem of Venkov and Koch; however our proof avoids the use of modular forms.

---

* This paper appeared in *IEEE Trans. Inform. Theory*, **37** (1991), pp. 1261-1268.

1. A strengthened Assmus-Mattson theorem

Let C be a binary, linear $[n, k, d]$ code with nonzero weights $w_1 = d, w_2, \ldots, w_s$, and let $w_1', \ldots, w_{s'}'$ be the nonzero weights in the dual code $C^\perp$. Our starting point is the following theorem.

Theorem 1 (Assmus and Mattson [2]). *Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w_i' with $0 < w_i' \le n - t$. Then the codewords of any weight w_i in C form a t-design.*

Venkov [21], answering a question raised in [20], showed that this theorem has an analogue for extremal even unimodular lattices in Euclidean space of dimension $24m$. The expected analogue was that the lattice vectors of any fixed nonzero length would form a spherical 11-design. Venkov proved this and more: he showed that these vectors possess an additional symmetry, forming what he called a spherical 11½-design. His proof uses the theory of modular forms.

Venkov [21] also announced that similar results could be obtained for self-dual codes. These results are stated by Koch [15] (see also [14], [16]). In particular, Venkov and Koch show that, in any extremal binary self-dual doubly-even code C, the set $\mathtt{P}$ of minimal weight words has the property that a certain linear form associated with $\mathtt{P}$ is constant on $(t+2)$-sets. Here $t = 5$ if the length n of the code is a multiple of 24, $t = 3$ if $n \equiv 8 \pmod{24}$, and $t = 1$ if $n \equiv 16 \pmod{24}$. To prove their result they associate a unimodular lattice with C and again apply the theory of modular forms.

Our strengthened version of Theorem 1 involves the concept of a T-design, defined as follows (cf. [8]). Let Ω be the set of all d-subsets of the n-set $[1,n] = \{1, \ldots, n\}$, with $d \le n/2$. We identify Ω with the set of all points $\xi = (\xi_1, \ldots, \xi_n)$ in $\mathtt{R}^n$ that satisfy $\xi_p \in \{0,1\}$ for all p and $\Sigma_{p=1}^n \xi_p = d$. The vector space $\mathtt{R}^\Omega$ of mappings from Ω to $\mathtt{R}$ is invariant under the natural

action of the symmetric group S_n. The irreducible S_n-invariant subspaces of $\mathtt{R}^{\Omega}$ are the *harmonic spaces* $\mathrm{harm}(i)$, $i = 0, 1, \ldots, d$. (These spaces are described in detail in Section 2, where in particular we give an explicit basis for $\mathrm{harm}(i)$.)

Let $\mathtt{P}$ be a subset of Ω, i.e. a constant weight code, and let $\pi(\mathtt{P}) \in \mathtt{R}^{\Omega}$ be the corresponding characteristic vector. The importance of the harmonic space $\mathrm{harm}(i)$ is that if the projection of $\pi(\mathtt{P})$ onto $\mathrm{harm}(i)$ is zero, then there is some regularity in the way the vectors of $\mathtt{P}$ meet an arbitrary i-subset of $[1, n]$. In particular (see [10]), $\mathtt{P}$ is a t-design if and only if, for all $i = 1, 2, \ldots, t$, the inner product $\langle \pi(\mathtt{P}), f \rangle = 0$ for all $f \in \mathrm{harm}(i)$. As in [8] we extend the definition of a design to subsets $T \subseteq [1, n]$ other than $[1, t]$ by saying that a collection $\mathtt{P}$ is a *T-design* if, for all $i \in T$, the inner product $\langle \pi(\mathtt{P}), f \rangle = 0$ for all $f \in \mathrm{harm}(i)$. (In case $0 \in T$, a T-design is defined to be a T'-design with $T' = T \setminus \{0\}$.)

When combined with the results of Section 3 of the present paper (in particular Theorem 7), the Venkov-Koch result mentioned above implies that the codewords of minimal weight in an extremal self-dual doubly-even code C form a $\{1, 2, \ldots, t, t+2\}$-design. (For in this case the linear form in Theorem 7 reduces to Venkov's form, given on page 461 of Koch [15].)

The purpose of the present paper is to give a similar generalization of the Assmus-Mattson theorem that does not assume the code is self-dual and whose proof avoids the use of modular forms. Our main theorem is the following.

Theorem 2. *Let C be a binary $[n, k, d]$ code with nonzero weights $w_1 = d, w_2, \ldots, w_s$, and let $w_1', \ldots, w_{s'}'$ be the nonzero weights in the dual code $C^{\perp}$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w_i' with $0 < w_i' \le n - t$. If $w_2 \ge d + 4$ then either the codewords in C of any nonzero weight w_i form a $(t+1)$-design or else the codewords of minimal weight d form a $\{1, 2, \ldots, t, t+2\}$-design.*

The proof is given in Section 4. In one important special case we can prove slightly more.

Theorem 3. *If, in addition to the hypotheses of Theorem* 2, *C is self-dual with all weights divisible by* 4 *then the codewords of any given weight* w_i *form either a* $(t+1)$*-design or a* $\{1, 2, \ldots, t, t+2\}$*-design.*

The proof is given in Section 5.

A list of the known extremal codes is given in [6, p. 194] and [7]. We may conclude for example that the codewords of minimal weight in the [24, 12, 8] Golay code and the [48, 24, 12] extended quadratic residue code form $\{1, 2, 3, 4, 5, 7\}$-designs. The minimal weight codewords in any of the five [32, 16, 8] self-dual doubly-even codes ([5], [7]) or in the extremal self-dual codes of lengths 56, 80 and 104 form $\{1, 2, 3, 5\}$-designs, and the minimal weight words in the extremal self-dual codes of lengths 16, 40, 64, 88 and 136 form $\{1, 3\}$-designs. Other examples are given in Section 4.

The invariant linear forms associated with codes are further investigated in [3], [4]. Generalizations to nonlinear codes and other fields are considered in [3].

2. The harmonic space harm(i)

In this section we give a more precise definition of and an explicit basis for the harmonic space $\mathrm{harm}(i)$.

We first define the *homogeneous space* $\mathrm{hom}(i)$ $(0 \le i \le n)$. This is the subspace of $\mathbb{R}^{\Omega}$ represented by homogeneous polynomials $f(z) = f(z_1, \ldots, z_n)$ of total degree i and degree at most 1 in each variable z_p. Note that, since these functions are defined on Ω, z_p^2 and z_p $(1 \le p \le n)$ represent the same function, and $z_1 + z_2 + \cdots + z_p$ is the constant function d. The latter assertion implies that $\mathrm{hom}(j)$ is a subspace of $\mathrm{hom}(i)$ for $0 \le j \le i$.

The monomials $z_{p_1} z_{p_2} \cdots z_{p_i}$ are linearly independent and span hom(i). Thus the dimension of hom(i) is $\binom{n}{i}$ (cf. [10]).

The Laplacian Δ is the differential operator given by

$$\Delta f(z) = \sum_{p=1}^{n} \frac{\partial f(z)}{\partial z_p} .$$

This maps hom(i) onto hom($i-1$), and the kernel is the *harmonic space* harm(i). In [10] it is shown that there is an orthogonal decomposition

$$\text{hom}(i) = \text{harm}(i) \oplus \text{hom}(i-1) , \qquad (1 \le i \le n) ,$$

with respect to the inner product $\langle f, g \rangle = \sum_{\xi \in \Omega} f(\xi) g(\xi)$, from which it follows that the dimension of harm(i) is $\binom{n}{i} - \binom{n}{i-1}$ Hom(0) = harm(0) is the 1-dimensional space of constant functions.

Theorem 4. *For any i-subset* $\{q_1, \ldots, q_i\}$ *of* $[1,n]$ *we define an element* ϕ *of* $\mathbb{R}^{\Omega}$ *by*

$$\phi(z_1, \ldots, z_n) = \sum_{j=0}^{i} (-1)^j \binom{i}{j} \binom{d-j}{i-j} \binom{n-i+1}{j}^{-1} \sigma_j(z_{q_1}, \ldots, z_{q_i}) , \tag{1}$$

where $\sigma_j(z_{q_1}, \ldots, z_{q_i})$ *is the sum of the characteristic functions* $z_{p_1} z_{p_2} \cdots z_{p_j}$ *of all j-subsets* $\{p_1, \ldots, p_j\}$ *of* $\{q_1, \ldots, q_i\}$. *Then the set of all* $\binom{n}{i}$ *such* ϕ*'s spans* harm(i).

Proof. Consider a monomial $m(z)$ in hom(i). Without loss of generality we may take

$$m(z) = z_1 z_2 \cdots z_i .$$

For an integer $u \in [0, i]$ we define $\phi_u(z) \in \text{hom}(i)$ to be the sum of all monomials of degree i having exactly u variables z_p in common with $m(z)$. We first show that

$$\Delta \phi_u(z) = (i-u+1) g_{u-1}(z) + (n-2i+u+1) g_u(z) , \tag{2}$$

where $g_j(z) \in \hom(i-1)$ is the sum of all monomials of degree $i-1$ having exactly j variables in common with $m(z)$. We write $z = (x, y)$, where $x = (z_1, \ldots, z_i)$ and $y = (z_{i+1}, \ldots, z_n)$. Then by definition,

$$\phi_u(z) = \sigma_u(x)\sigma_{i-u}(y) , \qquad g_j(z) = \sigma_j(x)\sigma_{i-j-1}(y) , \tag{3}$$

where $\sigma_j(w) = \sigma_j(w_1, \ldots, w_r) = \Sigma\, w_{p_1} w_{p_2} \cdots w_{p_j}$ denotes the elementary symmetric function of degree j in the variables $w_1, \ldots, w_r$. Note that $\sigma_j(x)$ is the sum of all monomials of degree j dividing $m(z)$. Equation (2) follows from the identities

$$\Delta\sigma_u(x) = (i-u+1)\sigma_{u-1}(x) \quad \text{and} \quad \Delta\sigma_r(y) = (n-i-r+1)\sigma_{r-1}(y) .$$

We now define

$$\phi(z) = \sum_{u=0}^{i} (-1)^u \binom{i}{u}^{-1} \binom{n-2i+u}{u} \phi_u(z) . \tag{4}$$

It follows readily from (2) that $\phi(z)$ is a solution of the Laplace equation $\Delta\phi(z) = 0$. Thus we have associated an eigenfunction $\phi \in \mathrm{harm}(i)$ with the given monomial $m \in \hom(i)$.

We next prove that $\phi(z)$ satisfies Eq. (1). First a simple counting argument yields

$$\sigma_u(x)\sigma_l(x) = \sum_{j=\max\{u,l\}}^{u+l} \binom{j}{u}\binom{u}{j-l} \sigma_j(x) , \tag{5}$$

for all u and l with $u + l \le i$. We then obtain the identity

$$\sigma_r(y) = \sum_{l=0}^{r} (-1)^l \binom{d-l}{r-l} \sigma_l(x) , \tag{6}$$

for $r \le i$. This can be proved by induction on r, as follows. We use the two relations

$$\sigma_1(y) = d - \sigma_1(x)$$

(which is the case $r = 1$ of (6)) and

$$\sigma_1(\cdot)\sigma_l(\cdot) = l\,\sigma_l(\cdot) + (l+1)\sigma_{l+1}(\cdot)$$

(which is a special case of (5)) together with (6) to obtain

$$(r+1)\sigma_{r+1}(y) = \sum_{l=0}^{r+1}(-1)^l\left[(d-r-l)\binom{d-l}{r-l} + l\binom{d+1-l}{r+1-l}\right]\sigma_l(x)\ ,$$

which is (6) with r replaced by $r+1$.

Using (3)-(5) and the combinatorial identity

$$\sum_l (-1)^l \binom{d-l}{i-u-l}\binom{u}{j-l} = (-1)^{j-u}\binom{d-j}{i-j}$$

(which follows from [13], p. 58, Eq. (24)), we obtain a representation for $\phi_u(z)$ in the simple form

$$\phi_u(z) = \sum_{j=u}^{i}(-1)^{j-u}\binom{j}{u}\binom{d-j}{i-j}\sigma_j(x)\ . \tag{7}$$

Equation (1) now follows from (4) and (7), after applying the classical identity $\sum_u \binom{i-u}{j-u}\binom{n-2i+u}{u} = \binom{n-i+1}{j}$ ([12], Eq. (3.2)), together with $\binom{i}{j}\binom{j}{u} = \binom{i}{u}\binom{i-u}{j-u}$.

The set of all $\phi(z)$ associated with monomials m of degree i spans the whole space harm(i). For by construction the linear space spanned by these functions is invariant under the symmetric group S_n; and as the harmonic spaces harm(j) are the *irreducible* S_n-invariant subspaces of $\mathbb{R}^{\Omega}$, this implies that the space in question coincides with harm(i). This completes the proof of Theorem 4.

We conclude this section with an application of Theorem 4. (A stronger result will be given in Section 3.)

Theorem 5. *A classical* $(l-2)$*-design* $\mathbb{P}$ *is also an* $\{l\}$*-design if and only if for any l-subset x of* $[1,n]$ *the quantity*

$$L_x = \{l(d-l+1) - (n-2l+2)\}\mu_{l,x} + (d-l+1)\mu_{l-1,x}\,, \tag{8}$$

where $\mu_{j,x}$ *is the number of blocks in* $\mathtt{P}$ *that have exactly j points in common with x, is independent of the choice of x.* (*We shall therefore call* L_x *an invariant linear form.*)

Proof. Let λ_j $(0 \le j \le l-2)$ be the number of blocks of $\mathtt{P}$ containing a particular set of j points. If x is any l-subset of $[1,n]$ then since $\mathtt{P}$ is an $(l-2)$-design we have

$$\langle \pi(\mathtt{P}),\, \sigma_j(x)\rangle = \begin{bmatrix} l \\ j \end{bmatrix}_2 \lambda_j\,, \qquad j = 0, 1, \ldots, l-2\,,$$

$$\langle \pi(\mathtt{P}),\, \sigma_{l-1}(x)\rangle = l\,\mu_{l,x} + \mu_{l-1,x}\,,$$

$$\langle \pi(\mathtt{P}),\, \sigma_l(x)\rangle = \mu_{l,x}\,.$$

Now $\mathtt{P}$ is an $\{l\}$-design if and only if $\langle \pi(\mathtt{P}),\, f\rangle = 0$ for all $f \in \mathrm{harm}(l)$, or equivalently (from Theorem 4) if and only if $\langle \pi(\mathtt{P}),\, \phi(x)\rangle = 0$ for all l-subsets x of $[1,n]$. Using (1) with $i = l$, and the trivial calculation that

$$\frac{(-1)^l \begin{bmatrix} d-l \\ 0 \end{bmatrix}_2 \begin{bmatrix} n-l+1 \\ l \end{bmatrix}_2 \begin{bmatrix} l \\ l \end{bmatrix}_2}{(-1)^{l-1} \begin{bmatrix} d-l+1 \\ 1 \end{bmatrix}_2 \begin{bmatrix} n-l+1 \\ l-1 \end{bmatrix}_2 \begin{bmatrix} l \\ l-1 \end{bmatrix}_2} = -\,\frac{n-2l+2}{d-l+1}$$

we see that $\langle \pi(\mathtt{P}),\, \phi(x)\rangle = 0$ for all x implies that L_x is independent of x. Conversely, if L_x is independent of x, the inner product

$$\langle \pi(\mathtt{P}),\, \sum_{j=0}^{l} (-1)^j \begin{bmatrix} l \\ j \end{bmatrix}_2^{-1} \begin{bmatrix} d-j \\ l-j \end{bmatrix}_2 \begin{bmatrix} n-l+1 \\ j \end{bmatrix}_2 \sigma_j(x)\rangle = A\,,$$

for some constant A independent of x. Since

$$\sum_x \sigma_j(x) \in \mathrm{hom}(0)\,, \qquad \text{for all} \quad j\,,$$

$$\sum_{j=0}^{l} (-1)^j \begin{bmatrix} l \\ j \end{bmatrix}_2^{-1} \begin{bmatrix} d-j \\ l-j \end{bmatrix}_2 \begin{bmatrix} n-l+1 \\ j \end{bmatrix}_2 \sigma_j(x) \in \mathrm{harm}(l)\,, \qquad \text{for all} \quad x\,,$$

we have

$$\sum_x \sum_{j=0}^{l} (-1)^j \binom{l}{j}^{-1} \binom{d-j}{l-j} \binom{n-l+1}{j} \sigma_j(x) \in \hom(0) > \mathrm{harm}(l) = \{0\} ,$$

and so $A = 0$. This completes the proof.

3. Invariant linear forms

Any S_n-invariant subspace ζ of $\mathbb{R}^{\Omega}$ is the sum of harmonic subspaces:

$$\zeta = \sum_{i \in T} \mathrm{harm}(i) , \tag{9}$$

where T is a well-defined subset of $\{0, 1, \ldots, d\}$, and $\sum$ denotes an orthogonal sum. There are 2^{d+1} such subspaces ζ.

Let $\mathbb{P}$ be a subset of Ω. A subspace ζ of $\mathbb{R}^{\Omega}$ will be said to be $\mathbb{P}$*-regular* if

$$\langle \pi(\mathbb{P}), \psi \rangle = \frac{|\mathbb{P}|}{|\Omega|} \langle \pi(\Omega), \psi \rangle , \quad \text{for all} \quad \psi \in \zeta . \tag{10}$$

Note that since $\pi(\Omega)$ is the function 1 (which spans harm(0)), the inner product $\langle \pi(\Omega), \psi \rangle$ vanishes for all $\psi \in \mathrm{harm}(j)$ with $j \geq 1$.

Theorem 6. *A non-empty subset $\mathbb{P} \subseteq \Omega$ is a T-design if and only if the subspace ζ defined by* (9) *is $\mathbb{P}$-regular.*

Proof. If ζ is $\mathbb{P}$-regular it follows from (9) and (10) that

$$\langle \pi(\mathbb{P}), \psi \rangle = 0, \quad \text{for all} \quad \psi \in \mathrm{harm}(j) \quad \text{with} \quad j \in T,\ j \neq 0 , \tag{11}$$

i.e. $\mathbb{P}$ is a T-design. Conversely, if $\mathbb{P}$ is a T-design with $0 \notin T$ then

$$\pi(\mathbb{P}) \in \sum_{i \notin T} \mathrm{harm}(i) \tag{12}$$

and so $\zeta = \sum_{i \in T} \mathrm{harm}(i)$ is $\mathbb{P}$-regular.

We can now give the generalization of Theorem 5 that will be used to prove the main

theorem. We replace (8) by a more general invariant form, (13).

Theorem 7. *Let* $\mathcal{P}$ *be a non-empty subset of* Ω. *Suppose that for some integer* l *with* $1 \le l \le d$ *there exist real numbers* a, b, c, *not all zero, such that*

$$a\,\mu_{l,x} + b\,\mu_{l-1,x} = c \tag{13}$$

for all l*-subsets* x *of* $\{1, 2, \ldots, n\}$ ($\mu_{j,x}$ *was defined in Theorem* 5). *Then*

$$\begin{aligned} &\mathcal{P} \text{ is an } \{l\}\text{-design}, \qquad \text{if} \quad a \ne lb, \\ &\mathcal{P} \text{ is an } \{l-1\}\text{-design}, \quad \text{if} \quad a = lb. \end{aligned} \tag{14}$$

In particular, if $\mathcal{P}$ *is not an* $\{l-1\}$*-design then* $\mathcal{P}$ *is an* $\{l\}$*-design.*

Proof. For a given l-set $x = \{p_1, \ldots, p_l\}$ let us define a function $\psi_x \in \mathbb{R}^{\Omega}$ by

$$\begin{aligned} \psi_x(\xi_1, \ldots, \xi_n) &= a\,\xi_{p_1}\xi_{p_2}\cdots\xi_{p_l} + \\ b[(1-\xi_{p_1})\xi_{p_2}\cdots\xi_{p_l} &+ \xi_{p_1}(1-\xi_{p_2})\xi_{p_3}\cdots\xi_{p_l} + \cdots + \xi_{p_1}\cdots\xi_{p_{l-1}}(1-\xi_{p_l})]. \end{aligned} \tag{15}$$

The assumption (13) can be written as

$$\langle \pi(\mathcal{P}), \psi_x \rangle = c, \quad \text{for all} \quad l\text{-sets} \quad x. \tag{16}$$

The value of c can be deduced from a and b by summing (13) over all l-sets x; this yields

$$\left[a\binom{d}{l} + b\binom{d}{l-1}\binom{n-d}{1}\right]|\mathcal{P}| = c\binom{n}{l}. \tag{17}$$

Now $\langle \pi(\Omega), \psi_x \rangle$ is clearly constant, and this constant, c' say, is given by

$$\left[a\binom{d}{l} + b\binom{d}{l-1}\binom{n-d}{1}\right]|\Omega| = c'\binom{n}{l}. \tag{18}$$

It follows from (17), (18) that (16) amounts to

$$\langle \pi(\mathcal{P}), \psi_x \rangle = \frac{|\mathcal{P}|}{|\Omega|}\,\langle \pi(\Omega), \psi_x \rangle, \quad \text{for all} \quad l\text{-sets} \quad x. \tag{19}$$

Consider the linear space ζ spanned by the functions ψ_x (for all l-sets x). By definition, ζ is S_n-invariant. Furthermore it follows from (19) that ζ is $\mathtt{P}$-regular. Hence $\mathtt{P}$ is a T-design with respect to the set T defined from the harmonic decomposition (9) of ζ. In view of (15) we have

$$\psi_x(\xi) = (a - lb)\xi_{p_1} \cdots \xi_{p_l} + \theta_{l-1}, \tag{20}$$

where θ_{l-1} is a member of $\hom(l-1)$. Hence ζ is a subspace of $\hom(l)$, and ζ is a subspace of $\hom(l-1)$ if and only if $a = lb$. Furthermore it is easily seen from (15) that (assuming a, b, c are not all zero) ζ is not a subspace of $\hom(l-2)$. (This is obvious if $a \neq lb$. When $a = lb$,

$$\begin{aligned}
\sum_{\substack{x = \{1, \ldots, l-1, i\} \\ \text{where } i = l, \ldots, n}} \psi_x(\xi) &= b \sum_{i=l}^{n} [\xi_2 \xi_3 \cdots \xi_{l-1} + \xi_1 \xi_3 \cdots \xi_{l-1} + \cdots + \xi_1 \xi_2 \cdots \xi_{l-2}]\xi_i \\
&\quad + b(n-l+1)\xi_1 \cdots \xi_{l-1} \\
&= b[\xi_2 \cdots \xi_{l-1} + \cdots + \xi_1 \cdots \xi_{l-2}]\left(\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{l-1} \xi_i\right) \\
&\quad + b(n-l+1)\xi_1 \cdots \xi_{l-1} \\
&= b(n-2l+2)\xi_1 \cdots \xi_{l-1} \\
&\quad + b(d-l+2)[\xi_2 \cdots \xi_{l-1} + \cdots + \xi_1 \cdots \xi_{l-2}],
\end{aligned}$$

and since $n - 2l + 2$ is not zero, this sum cannot belong to $\hom(l-2)$ unless b, and hence a and c, are zero.) Thus if $a \neq lb$ then $\mathtt{P}$ is an $\{l\}$-design, and if $a = lb$ then $\mathtt{P}$ is an $\{l-1\}$-design. This completes the proof.

4. Proof of Theorem 2

Suppose C satisfies the hypotheses of Theorem 2. By Theorem 1 the codewords of any weight w_i in C form a t-design. If $k = \dim C = 1$, only the repetition code yields a t-design. In this case $C^{\perp}$ consists of all even weight vectors and gives trivial designs. So from now on we assume $k > 1$.

It is easy to see (the argument is given on page 165 of [17]) that there are no codewords of $C^{\perp}$ with weight w' satisfying $n-t < w' < n$, and hence that there are two cases: (i) C is even, $w'_{s'} = n$, $s' = d-t+1$, or (ii) C is not even, $w'_{s'} \neq n$, $s' = d-t$. Thus we can write

$$s' = d - t + 1 - \delta , \tag{21}$$

where $\delta = 0$ if C is even, $\delta = 1$ if C is not even.

We work in the framework of the Hamming association scheme $H(n,2)$ – see [8], [9], [11], [17, Chap. 21] for background. The *Krawtchouk polynomial* of degree i is defined to be

$$P_i(\xi) = \sum_{j=0}^{i} (-1)^j \binom{\xi}{j} \binom{n-\xi}{i-j} \qquad (0 \le i \le n) ,$$

and the *annihilator polynomial* of C is

$$\alpha(\xi) = 2^{n-k} \prod_{i=1}^{s'} \left(1 - \frac{\xi}{w'_i}\right).$$

Let us expand

$$\xi^m \, \alpha(\xi) = \sum_{i=0}^{s'+m} \alpha_i^{(m)} \, P_i(\xi), \qquad m = 0, 1, \ldots .$$

We set $\alpha_i^{(0)} = \alpha_i$. Note that $\alpha_{s'+m}^{(m)} \neq 0$ for all m.

It was shown in [9] that for all $x \in \mathbb{F}_2^n$,

$$\sum_{i=0}^{s'+m} \alpha_i^{(m)} \, b_i(x) = \begin{cases} 1, & m = 0, \\ 0, & m \ge 1, \end{cases} \tag{22}$$

where $b_i(x)$ is the number of codewords in C at distance i from x.

We next prove a lemma.

Lemma 8. *Let C be a binary $[n, k, d]$ code with nonzero weights $w_1 = d, w_2, \ldots, w_s$, and let $w'_1, \ldots, w'_{s'}$ be the nonzero weights in the dual code $C^{\perp}$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \le n - t$, and suppose $w_2 \ge d + 4$. If the codewords of minimal weight form a $(t+1)$-design then so do the codewords of any nonzero weight w_i.*

Proof. Let x be an arbitrary subset of $\{1, 2, \ldots, n\}$ of size $l = t + 1$. Setting $m = w_2 - d - 2 + \delta > 0$ in (22) we obtain

$$\sum_{i=0}^{w_2 - t - 1} \alpha_i^{(m)} \, b_i(x) = 0 . \tag{23}$$

The zero codeword contributes to the sum in (23) if and only if $l \le w_2 - t - 1$. The contributions from the codewords of weight d are independent of x, since by hypothesis these words form a $(t+1)$-design. Codewords of weight greater than w_2 do not contribute to the sum at all, since

$$w_3 - l > w_2 - l = w_2 - t - 1 . \tag{24}$$

We now consider the contributions from the codewords c of weight w_2. Suppose c intersects x in j points. Then

$$\mathrm{dist}(c, x) = w_2 + l - 2j \le w_2 - t - 1 , \tag{25}$$

implying $j = t + 1$, i.e. codewords of weight w_2 contribute to the sum in (23) if and only if they contain x. Therefore (23) implies that the number of codewords of weight w_2 containing x is independent of x, or in other words the codewords of weight w_2 form a $(t+1)$-design. Similarly, by taking $m = w_j - d - 2 + \delta$ in (22), we find that the words of weight w_j form a $(t+1)$-design. This proves the lemma.

We now complete the proof of Theorem 2. The set of minimal weight words in C will be

denoted by $\mathtt{P}$, and $\mu_{j,x}$ is the number of words in $\mathtt{P}$ that have exactly j points in common with a given l-set x.

Case (i), C even, $s' = d - t + 1$. Suppose first that there is a smallest integer f in the range $0 \le f \le [(d-t)/2]$ such that $\alpha_{d-t-2f} \neq 0$. Let x be an arbitrary subset of $\{1, 2, \ldots, n\}$ of size $l = t + 2f$. Since C is even, the distances from x to C are all congruent to t (modulo 2), and from (22) we have

$$\sum_{\substack{i=0 \\ i \equiv t \pmod 2}}^{d-t-2f} \alpha_i \, b_i(x) \;=\; 1 \,. \tag{26}$$

Proceeding as in the proof of the lemma, we find that only the zero codeword and the codewords of weight d contribute to the sum in (26), and the words of weight d contribute if and only if they contain x. Equation (26) then reads

$$\alpha_{d-t-2f} \, \mu_{t+2f,x} \;=\; 1 \;-\; \alpha_{t+2f} \, \varepsilon_{d-2t-2f-2} \,, \tag{27}$$

where we set

$$\varepsilon_p \;=\; \begin{cases} 0 & p < 0, \\ 1 & p \ge 0. \end{cases}$$

If $f \ge 1$ we conclude from (27) that $\mathtt{P}$ is a $(t+2f)$-design, in particular a $(t+1)$-design, and therefore by Lemma 8 that the codewords of every nonzero weight form $(t+1)$-designs.

On the other hand suppose $f = 0$. We take x to have weight $l = t + 2$, and find that (22) becomes

$$\alpha_{d-t-2} \, \mu_{t+2,x} \;+\; \alpha_{d-t} \, \mu_{t+1,x} \;=\; 1 \;-\; \alpha_{t+2} \, \varepsilon_{d-2t-2} \,, \tag{28}$$

where both coefficients on the left side are nonzero. From Theorem 7 we conclude that $\mathtt{P}$ is a $\{t+1\}$-design or a $\{t+2\}$-design, and hence either a $(t+1)$-design or a $\{1, \ldots, t, t+2\}$-design. In the former case Lemma 8 extends this to codewords of every nonzero weight.

The third possibility is that no such f exists, and all coefficients α_{d-t-2i} are zero. But in this case taking x in (22) to have weight t leads to a contradiction (that left side of (26) vanishes but the right side does not).

Case (ii), C not even, $s' = d-t$. Let x have weight $t+2$. Equation (22) implies

$$\alpha_{d-t-2}\,\mu_{t+2,x} + \alpha_{d-t}\,\mu_{t+1,x} = 1 - \alpha_{t+2}\,\varepsilon_{d-2t-2}\ ,$$

where $\alpha_{d-t} \neq 0$. From Theorem 7 we conclude that $\mathcal{P}$ is a $\{t+1\}$-design or a $\{t+2\}$-design, and Lemma 8 completes the proof.

An alternative proof of Theorem 2. The above argument shows only that an invariant linear form of the type (13) exists; by Theorem 7 this is enough to prove the desired result. However it is possible to give a proof in which a ''computation miracle'' produces an explicit invariant linear form. We give this direct proof in the case when C is even. We suppose that $\mathcal{P}$ is not a $(t+1)$-design.

By applying (22) with $m=0$ and 1 to a $(t+1)$-set x we obtain

$$\alpha_{d-t-1}\,\mu_{t+1,x} + \alpha_{d-t+1}\,\mu_{t,x} = 1 - \alpha_{t+1}\,\varepsilon_{d-2t}\ , \tag{29}$$

$$\alpha^{(1)}_{d-t-1}\,\mu_{t+1,x} + \alpha^{(1)}_{d-t+1}\,\mu_{t,x} = -\,\alpha^{(1)}_{t+1}\,\varepsilon_{d-2t+1}\ , \tag{30}$$

where $\alpha_{d-t+1} \neq 0$. Since $\mathcal{P}$ is a t-design,

$$(t+1)\,\mu_{t+1,x} + \mu_{t,x} = (t+1)\,\lambda_t\ , \tag{31}$$

where λ_t is the number of blocks through t given points. Since $\mathcal{P}$ is not a $(t+1)$-design, the left sides of (29)-(31) must be proportional (or else $m_{t+1,x}$ would be independent of x). Therefore

$$\alpha_{d-t-1} = (t+1)\,\alpha_{d-t+1}\ , \tag{32}$$

$$\alpha^{(1)}_{d-t-1} = (t+1)\,\alpha^{(1)}_{d-t+1}\ , \tag{33}$$

and so $\alpha_{d-t-1} \neq 0$. From the Krawtchouk recurrence [17, p. 152]

$$(i+1)\ P_i(\xi) = (n-2\xi)\ P_i(\xi) - (n-i+1)\ P_{i-1}(\xi)$$

($i \geq 1$), with $P_0(\xi) = 1$, $P_1(\xi) = n-2\xi$, we obtain

$$2\alpha_i^{(1)} = -(n-i)\alpha_{i+1} + n\alpha_i - i\alpha_{i-1} \tag{34}$$

($i \geq 1$). In particular,

$$2\alpha_{d-t+1}^{(1)} = n\alpha_{d-t+1} - (d-t+1)\alpha_{d-t}\ , \tag{35}$$

$$2\alpha_{d-t-1}^{(1)} = -(n-d+t+1)\alpha_{d-t} + n\alpha_{d-t-1} - (d-t-1)\alpha_{d-t-2}\ . \tag{36}$$

Furthermore $\alpha_{d-t} \neq 0$, or else (as shown in the first proof) $\mathtt{P}$ is a $(t+1)$-design. From (32), (33), (35), (36) we obtain

$$\alpha_{d-t-2} = \frac{(t+2)(d-t-1) - (n-2t-2)}{d-t-1}\ \alpha_{d-t}\ . \tag{37}$$

We now apply (22) with $m=0$ to a $(t+2)$-set x and find

$$\{(t+2)(d-t-1) - (n-2t-2)\}\ \mu_{t+2,x} + (d-t-1)\mu_{t+1,x}$$
$$= \frac{d-t-1}{\alpha_{d-t}}\ (1-\alpha_{t+2}\ \varepsilon_{d-2t-1})\ . \tag{38}$$

The left-hand side of (38) is the desired linear form, independent of x. Theorem 7 and Lemma 8 complete the proof. The most interesting aspect of this argument is the leverage provided by the assumption that $\mathtt{P}$ is *not* a $(t+1)$-design.

Examples. An example with $t=5$ is provided by the set of 759 minimal weight words in the [24, 12, 8] Golay code. In this case we have the identity $\mu_{7,x} + \mu_{6,x} = 1$ for any 7-set x. (There are only two possibilities, $(\mu_{7,x}, \mu_{6,x}) = (0,1)$ or $(1,0)$, corresponding to the two kinds of 7-subsets of [1, 24] under the action of the Mathieu group M_{24} – cf. [6, Fig. 10.1].) The 759 words form a {1, 2, 3, 4, 5, 7}-design.

A second example with $t=5$ is provided by the 17296 minimal weight words in the $[48, 24, 12]$ extended quadratic-residue code (or in any self-dual doubly even $[48, 24, 12]$ code). In this case we have the identity $\mu_{7,x} + \mu_{6,x} = 8$ for any 7-set x. (There are only two possibilities: $(\mu_{7,x}, \mu_{6,x}) = (0,8)$ or $(1,7)$.) Again the minimal weight words form a $\{1, 2, 3, 4, 5, 7\}$-design.

A more trivial example with $t=1$ is provided by the $[n=2m, 2, m]$ code $\{0^{2m}, 0^m 1^m, 1^m 0^m, 1^{2m}\}$. The two words of weight m form a $\{1,3\}$-design.

A further example: complementation. The $\{1, 2, \ldots, l, l+2\}$-design property is preserved when the blocks of $\mathcal{P}$ are complemented. To see this, let $\overline{\mathcal{P}} = \{[1,n] \setminus B \mid B \in \mathcal{P}\}$, and let $\nu_{j,x}$ be the number of blocks in $\overline{\mathcal{P}}$ meeting a given $(l+2)$-set x in exactly j points. Then $\nu_{j,x} = \mu_{l+2-j,x}$, and we must therefore show that

$$\bar{a}\,\mu_{0,x} + \bar{b}\,\mu_{1,x} = \bar{c} \tag{39}$$

for all x, for suitable real numbers $\bar{a}, \bar{b}, \bar{c}$ not all zero. Since $\overline{\mathcal{P}}$ is a $\{1, 2, \ldots, l, l+2\}$-design we have invariant linear forms

$$a\,\mu_{l+2,x} + b\,\mu_{l+1,x} = c\,, \quad \text{where} \quad b \neq 0\,, \tag{40}$$

$$\sum_{i=j}^{l+2} \binom{i}{j} \mu_{i,x} = \binom{l+2}{j} \lambda_j\,, \quad j=0, 1, \ldots, l\,, \tag{41}$$

where λ_j is the number of blocks of $\mathcal{P}$ through j given points. Equations (40), (41) form a triangular system of $l+2$ equations in the $l+3$ quantities $\mu_{j,x}$, $j=0, \ldots, l+2$. From this we obtain

$$\mu_{0,x} = \alpha\,\mu_{l+2,x} + \beta\,, \quad (\alpha, \beta \text{ not both zero})\,,$$

$$\mu_{1,x} = \gamma\,\mu_{l+2,x} + \delta\,, \quad (\gamma, \delta \text{ not both zero})\,,$$

for suitable real numbers $\alpha, \beta, \gamma, \delta$, and Equation (39) follows.

5. Extension to codewords of higher weight and the proof of Theorem 3

Lemma 8 shows that if the codewords of minimal weight form a $(t+1)$-design then so do the codewords of any nonzero weight. To extend the $\{1, 2, \ldots, t, t+2\}$-design property to codewords of higher weight it is necessary to make some assumptions about the gap sizes $w_i - w_{i-1}$ for $i \geq 3$. In the sequel we shall only consider self-dual codes with all weights divisible by 4, even though the arguments apply to a wider class of codes.

We begin with an example, the $[24, 12, 8]$ Golay code. The annihilator polynomial is

$$\begin{aligned} \alpha(\xi) &= 2^{12} \left(1 - \frac{\xi}{8}\right)\left(1 - \frac{\xi}{12}\right)\left(1 - \frac{\xi}{16}\right)\left(1 - \frac{\xi}{24}\right) \\ &= \sum_{i=0}^{3} P_i(\xi) + \frac{1}{6} P_4(\xi) \, . \end{aligned} \tag{42}$$

Given an arbitrary 7-set x, let $M_{j,x}^{w}$ be the number of codewords of weight w that meet x in exactly j points. From (38), (41) we obtain the invariant linear forms

$$M_{7,x}^{8} + M_{6,x}^{8} \, , \tag{43}$$

$$21 M_{7,x}^{8} + 6 M_{6,x}^{8} + M_{5,x}^{8} \, . \tag{44}$$

Next we apply (22) with $m = 1$ to obtain the invariant form

$$\alpha_1^{(1)} \, M_{7,x}^{8} + \alpha_3^{(1)} \, M_{6,x}^{8} + \alpha_5^{(1)} \, M_{5,x}^{8} + \alpha_5^{(1)} \, M_{7,x}^{12} \, . \tag{45}$$

Before calculating the shifted Krawtchouk coefficients $\alpha_j^{(1)}$ we can see that there are two possibilities. The first is that the form

$$\alpha_1^{(1)} \, M_{7,x}^{8} + \alpha_3^{(1)} \, M_{6,x}^{8} + \alpha_5^{(1)} \, M_{5,x}^{8} \tag{46}$$

is a linear combination of (43) and (44). Since $\alpha_5^{(1)} \neq 0$, we may conclude that in this case the codewords of weight 12 form a 7-design. The second possibility is that (43), (44), (46) form a basis for the space of linear forms in the variables $M_{j,x}^{8}$ $j = 5, 6, 7$. Now we understand the Golay

code well enough to know that the first possibility does not occur, but it is precisely this argument that we will apply to an arbitrary doubly-even code. We may in fact calculate the shifted Krawtchouk coefficients from (34), finding that $\alpha_0^{(1)} = \alpha_1^{(1)} = \alpha_2^{(1)} = 0$, $\alpha_3^{(1)} = \frac{35}{4}$, $\alpha_4^{(1)} = 0$, $\alpha_5^{(1)} = -\frac{5}{12}$, so (45) becomes

$$21\, M_{6,x}^8 - M_{5,x}^8 - M_{7,x}^{12}\,. \tag{47}$$

Next we apply (22) with $m = 3$ to obtain the invariant form

$$\alpha_1^{(3)}\, M_{7,x}^8 + \alpha_3^{(3)}\, M_{6,x}^8 + \alpha_5^{(3)}\, M_{5,x}^8 + \alpha_7^{(3)}\, M_{4,x}^8 + \alpha_5^{(3)}\, M_{7,x}^{12} + \alpha_7^{(3)}\, M_{6,x}^{12}\,, \tag{48}$$

where $\alpha_7^{(3)} \neq 0$. From (41) we have a second invariant form involving the new variable $M_{4,x}^8$, namely

$$35\, M_{7,x}^8 + 15\, M_{6,x}^8 + 5\, M_{5,x}^8 + M_{4,x}^8\,. \tag{49}$$

Since (43), (44), (46), (47) are a basis for the space of linear forms in the variables $M_{j,x}^8$ $j = 4, 5, 6, 7$, we may eliminate these variables from (48) and obtain an invariant form

$$a\, M_{7,x}^{12} + b\, M_{6,x}^{12}\,,$$

of type (13). In this case $a/b = 5$, and so the codewords of weight 12 in the Golay code form a $\{1, 2, 3, 4, 5, 7\}$-design.

The proof of Theorem 3 is a straightforward generalization of this example. From Theorem 1 the codewords of any given weight w_p form a t-design, so (generalizing (41)) we have invariant linear forms

$$L_{w_p,\,j} = \sum_{h=1}^{t+2} \binom{h}{j} M_{h,x}^{w_p}, \quad j = 0, 1, \ldots, t, \quad p = 1, \ldots, d-t\,, \tag{50}$$

where x is an arbitrary $(t+2)$-subset of $[1, n]$. From (22) we also have invariant forms (generalizing (45) and (48)):

$$H_m = \sum_{\substack{w_i, j \\ w_i+t+2-2j \le d-t+1+m}} \alpha^{(m)}_{w_i+t+2-2j} \, M^{w_i}_{j,x} , \quad m = 1, 3, 5, \ldots . \tag{51}$$

Finally Theorem 2 provides an invariant form

$$a \, M^d_{t+2,x} + b \, M^d_{t+1,x} , \quad b \neq 0 . \tag{52}$$

The theorem is proved by induction. For $i = 2, \ldots$, let $\Gamma(i)$ be the linear system in the variables $\{M^{w_p}_{j,x} : p < i, w_p + t + 2 - 2j < w_i - t - 2\}$ consisting of (52) and the linear forms

$$L_{w_p, j} \quad \text{for} \quad p < i, \quad w_p + t + 2 - 2j < w_i - t - 2 , \quad \text{and}$$

$$H_m \quad \text{for} \quad m < w_i - d - 3 , \quad m \quad \text{odd} .$$

The inductive hypothesis is that the corank of the linear system $\Gamma(i)$ is at most 1. This is certainly true for $i = 2$, since $\Gamma(2)$ includes the triangular system consisting of (52) and $L_{d,j}$ for $d + t + 2 - 2j < w_2 - t - 2$.

The linear system $\Gamma(i+1)$ involves variables $M^{w_p}_{j,x}$ that do not appear in $\Gamma(i)$. For each new variable $M^{w_p}_{j,x}$ with $w_p < w_{i+1}$ we have a linear form $L_{w_p, f}$, so these new variables do not change the corank. The linear form

$$H_{w_{i+1}-d-3} - \alpha^{(w_{i+1}-d-3)}_{w_{i+1}-t-2} \, M^{w_{i+1}}_{t+2,x} \tag{53}$$

only involves variables $M^{w_p}_{j,x}$ with $w_p < w_{i+1}$. We distinguish two cases.

The first is that (53) is a linear combination of forms from $\Gamma(i)$ and forms $L_{w_p, f}$ involving variables $M^{w_p}_{j,x}$ not appearing in $\Gamma(i)$. Then $M^{w_{i+1}}_{t+2,x}$ is independent of x, that is the codewords of weight w_{i+1} form a $(t+2)$-design. Now $\Gamma(i+1)$ includes the triangular system

$$M^{w_{i+1}}_{t+2,x}, \; (t+2) \, M^{w_{i+1}}_{t+2,x} + M^{w_{i+1}}_{t+1,x}, \; L_{w_{i+1}, j}$$

in the variables $M^{w_p}_{j,x}$, so the corank of $\Gamma(i+1)$ is at most 1.

The second case is that the linear form (53), together with the forms in $\Gamma(i)$ and the forms $L_{w_p, f}$ involving variables $M_{j,x}^{w_p}$ not appearing in $\Gamma(i)$, form a basis for the space of linear forms in the variables appearing in (53). Now consider $H_{w_{i+1}-d-1}$. We may eliminate variables from $H_{w_{i+1}-d-1}$ to obtain a linear form

$$a\, M_{t+2,x}^{w_{i+1}} + b\, M_{t+1,x}^{w_{i+1}}, \tag{54}$$

where $b \neq 0$. By Theorem 7 we may conclude that the codewords of weight w_{i+1} form a $(t+1)$-design or a $\{1, 2, \ldots, t, t+2\}$-design. The rank of $\Gamma(i+1)$ restricted to variables $M_{j,x}^{w_p}$ for $p < i+1$ is full. Since $\Gamma(i+1)$ includes the triangular system $\{(54), L_{w_{i+1}, j}\}$ in the variables $M_{j,x}^{w_{i+1}}$, the corank of $\Gamma(i+1)$ is at most 1.

Remarks. The proof leaves open the possibility that the codewords of weight w_i might form a $(t+1)$-design while the codewords of weight w_j $(j \neq i)$ form a $\{1, \ldots, t, t+2\}$-design.

Acknowledgements

We thank the referees for several helpful comments.

REFERENCES

[1] M. Abramowitz and I. A. Stegun, ‘‘Handbook of Mathematical Functions’’, National Bureau of Standards Appl. Math. Series, vol. 55, U.S. Dept. Commerce, Wash. D.C., 1972.

[2] E. F. Assmus, Jr., and H. F. Mattson, Jr., ‘‘New 5-designs’’, J. Comb. Theory, vol. 6 (1969), 122-151.

[3] A. R. Calderbank and P. Delsarte, ‘‘On error-correcting codes and invariant linear forms’’, SIAM J. Discrete Math., to appear.

[4] A. R. Calderbank and P. Delsarte, ‘‘The concept of a (t, r)-regular design as an extension of the classical concept of a t-design’’, preprint.

[5] J. H. Conway and V. Pless, ‘‘On the enumeration of self-dual codes’’, J. Comb. Theory, vol. 28A (1980), 26-53.

[6] J. H. Conway and N. J. A. Sloane, ‘‘Sphere Packings, Lattices and Groups’’, Springer-Verlag, N.Y. 1988.

[7] J. H. Conway and N. J. A. Sloane, ‘‘A new upper bound on the minimal distance of self-dual codes’’, IEEE Trans. Information Theory, vol. 36 (1990), 1319-1333.

[8] P. Delsarte, ‘‘An algebraic approach to the association schemes of coding theory’’, Philips Research Reports Supplements, vol. 10 (1973).

[9] P. Delsarte, ‘‘Four fundamental parameters of a code and their combinatorial significance’’, Info. Control, vol. 23 (1973), 407-438.

[10] P. Delsarte, ‘‘Hahn polynomials, discrete harmonics, and t-designs’’, SIAM J. Appl. Math., vol. 34 (1978), 157-166.

[11] J.-M. Goethals, ‘‘Association schemes’’, in ‘‘Algebraic Coding Theory and Applications’’, edited by G. Longo, CISM Courses and Lectures 258, Springer-Verlag, Vienna, 1979, 243-283.

[12] H. W. Gould, ‘‘Combinatorial Identities’’, Morgantown, W. Va., Revised edition, 1972.

[13] D. E. Knuth, ‘‘The Art of Computer Programming’’, vol. 1, 2nd edition, Addison-Wesley, Reading, Mass., 1973.

[14] H. V. Koch, ‘‘On self-dual, doubly-even codes of length 32’’, Report R-Math-32/84, Institüt f. Math., Akad. Wiss. DDR, Berlin, 1984.

[15] H. Koch, ‘‘Unimodular lattices and self-dual codes’’, in ‘‘Proc. Intern. Congress Math., Berkeley 1986’’, Amer. Math. Soc., Providence R.I., 1987, vol. 1, pp. 457-465.

[16] H. Koch and B. B. Venkov, ‘‘Über ganzzahlige euklidische Gitter’’, J. Reine Angew. Math., vol. **398** (1989), 144-168.

[17] F. M. MacWilliams and N. J. A. Sloane, ‘‘The Theory of Error-Correcting Codes’’, North Holland, Amsterdam, 1979.

[18] C. L. Mallows and N. J. A. Sloane, ‘‘An upper bound for self-dual codes’’, Inform. Control, vol. 22 (1973), 188-200.

[19] N. J. A. Sloane, ‘‘A Handbook of Integer Sequences’’, Academic Press, N.Y. 1973.

[20] N. J. A. Sloane, ‘‘Binary codes, lattices and sphere packings’’, in ‘‘Combinatorial Surveys’’, edited P. J. Cameron, Academic Press, N.Y. 1977, pp. 117-164.

[21] B. B. Venkov, ‘‘On even unimodular extremal lattices’’ (in Russian), Trudy Mat. Inst. Steklov, vol. 165 (1984), 43-48. English translation in Proc. Steklov Inst. Math., vol. 165 (1984), 47-52.

To appear, Theoretical Computer Science, Nov. 1992

The Ring of k-Regular Sequences

*Jean-Paul Allouche**
C. N. R. S. (U. R. A. 226)
Mathématiques et Informatique
33405 Talence Cedex
France
allouche%frbdx11.bitnet

Jeffrey Shallit§
Dept. of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada
shallit@watdragon.waterloo.edu

Abstract.

The *automatic sequence* is the central concept at the intersection of formal language theory and number theory. It was introduced by Cobham, and has been extensively studied by Christol, Kamae, Mendès France and Rauzy, and other writers. Since the range of automatic sequences is finite, however, their descriptive power is severely limited.

In this paper, we generalize the concept of automatic sequence to the case where the sequence can take its values in a (possibly infinite) ring R; we call such sequences *k–regular*. (When R is finite, we obtain automatic sequences as a special case.) We argue that k–regular sequences provide a good framework for discussing many "naturally–occurring" sequences, and we support this contention by exhibiting many examples of k–regular sequences from numerical analysis, topology, number theory, combinatorics, analysis of algorithms, and the theory of fractals.

We investigate the closure properties of k–regular sequences. We prove that the set of k–regular sequences forms a ring under the operations of term-by-term addition and convolution. Hence the set of associated formal power series in $R[[X]]$ also forms a ring.

We show how k–regular sequences are related to $\mathbb{Z}$–rational formal series. We give a machine model for the k–regular sequences. We prove that all k–regular sequences can be computed quickly.

Let the *pattern sequence* $e_P(n)$ count the number of occurrences of the pattern P in the base–k expansion of n. Morton and Mourant showed that every sequence over $\mathbb{Z}$ has a unique expansion as a sum of pattern sequences. We prove that this "Fourier" expansion maps k–regular sequences to k–regular sequences. (This can be viewed as a generalization of results of Choffrut and Schützenberger, and previous results of Allouche, Morton, and Shallit.) In particular, the coefficients in the expansion of $e_P(an+b)$ form a k–automatic sequence.

Many natural examples and some open problems are given.

* Research supported in part by "PICS: Théorie des nombres et ordinateurs"
§ Research supported in part by NSF Grant CCR-8817400, the Wisconsin Alumni Research Foundation, and a Walter Burke award from Dartmouth College.

I. Introduction.

Let $\{S(n)\}_{n\geq 0}$ be a sequence with values chosen from a finite set Σ. Then $\{S(n)\}_{n\geq 0}$ is said to be k–automatic if, informally speaking, $S(n)$ is a finite-state function of the base–k expansion of n.

Automatic sequences have been studied by Cobham [Cob], Christol, Kamae, Mendès France and Rauzy [CKMR], and others. (For example, see [DMFP], [Mau], and the survey paper of Allouche [A1].) There are many other ways to characterize automatic sequences. For example, consider the following

Definition 1.1.

The *k–kernel* of a sequence is the set of all subsequences of the form $\{S(k^e n + a)\}_{n\geq 0}$, where $e \geq 0$ and $0 \leq a < k^e$.

Cobham [Cob] proved the following

Theorem 1.2.

A sequence is k–automatic if and only if its k–kernel is finite.

Unfortunately, the range of automatic sequences is necessarily finite, and this restricts their descriptive power.

In this paper, we are concerned with a natural generalization of automaticity to the case where the sequence $\{S(n)\}_{n\geq 0}$ takes its values in a (possibly infinite) ring; we call such sequences k–regular. (Another generalization of automatic sequences was already given by Allouche [A4].) We use an analogue of Theorem 1.2 as our *definition*. We show that k–regular sequences provide an excellent framework for describing many "naturally occurring" sequences, such as the numerators of the left endpoints of the Cantor set, the sequence $\{\nu_p(n!)\}_{n\geq 0}$, which counts the number of times a prime p divides a factorial, binary Gray code, numerators of entries of the Stern-Brocot tree, multiplicative-cost addition chains, etc.

We prove that k–regular sequences have nice closure properties. By associating a formal power series with each sequence, we prove that the set of k–regular sequences forms a ring, but not a field, under the usual power series operations.

We explore the connection with a machine model of Schützenberger [Sch], which includes finite automata with counters as a special case. This allows us to prove that the n-th term of a k–regular sequence can be computed in time polynomial in $\log n$.

We introduce the *pattern sequences* $e_P(n)$, which count the number of occurrences of the string P in the base–k expansion of n. Morton and Mourant [MM] showed that every sequence $\{S(n)\}_{n\geq 0}$ over $\mathbb{Z}$ has a unique expansion as a sum of pattern sequences. In analogy with the Fourier transform, we call this sequence of coefficients $\{\hat{S}(n)\}_{n\geq 0}$ the *pattern transform* of $\{S(n)\}_{n\geq 0}$. We show that a sequence is k–regular if and only if its pattern transform is k–regular. This can be viewed as a generalization of results of Choffrut and Schützenberger [CS] and previous results of the authors and P. Morton [AMS].

Finally, we give many examples and some open problems.

II. k–regular sequences: definition and properties.

Let R' be a commutative Noetherian ring, i. e. a ring in which every ideal is finitely generated. (Examples of such rings include all finite rings, $\mathbb{Z}$, all fields K, and the polynomial rings $K[X]$.) Let R be a ring containing R'.

Let $\mathcal{S}(R)$ denote the set of sequences with values in R. Let $\{S(n)\}_{n\geq 0}$ be a sequence with values in R, and let k be an integer ≥ 2.

Definition 2.1.

We say $\{S(n)\}_{n\geq 0}$ is (R',k)–*regular* if there exist a finite number of sequences $S_1, S_2, \ldots, S_j$ with values in R, such that each sequence in the k–kernel of $\{S(n)\}_{n\geq 0}$ is an R'–linear combination of the S_i.

Let $\mathcal{K}$ denote the k–kernel of $\{S(n)\}_{n\geq 0}$. Then $\{S(n)\}_{n\geq 0}$ is (R',k)–regular means that

$$\langle \mathcal{K} \rangle \subseteq \langle S_1, S_2, \ldots S_n \rangle,$$

i. e. $\langle \mathcal{K} \rangle$ is a sub-module of a finitely generated R'–module. By a well-known theorem (see, e. g., [Lan, pp. 142–144]), it follows that $\langle \mathcal{K} \rangle$ itself is finitely generated.

Thus Definition 2.1 can be restated as follows: a sequence $\{S(n)\}_{n\geq 0}$ with values in R is (R',k)–regular if the R'–module generated by its k–kernel is a *finitely generated* R'–submodule of $\mathcal{S}(R)$.

If the context is clear, we usually write just k–regular.

Note that if R' is a finite ring, then we recover the case of k–automatic sequences. For if every subsequence in the k–kernel can be written as an R'–linear combination of a finite set of sequences, then there are only a finite number of distinct elements of the k–kernel. In fact, the same holds for sequences that take on only finitely many values (see Theorem 2.3 below).

The reader may now wish to look at Section VII for some examples of k–regular sequences.

Our first theorem gives several alternative characterizations of k–regular sequences:

Theorem 2.2.

The following are equivalent:

(a) $\{S(n)\}_{n\geq 0}$ is (R',k)–regular;

(b) The R'–module generated by the k–kernel of $\{S(n)\}_{n\geq 0}$ is generated by a finite number of its subsequences of the form $S(k^{f_i}n + b_i)$ where $0 \leq b_i < k^{f_i}$;

(c) There exists an integer E such that for all $e_j > E$, each subsequence $S(k^{e_j}n + a_j)$ with $0 \leq a_j < k^{e_j}$ can be expressed as an R'–linear combination

$$S(k^{e_j}n + a_j) = \sum_i c_{ij} S(k^{f_{ij}}n + b_{ij}),$$

where $f_{ij} \leq E$ and $0 \leq b_{ij} < k^{f_{ij}}$;

(d) There exist an integer r and r sequences $S = S_1, S_2, \ldots, S_r$, such that for $1 \leq i \leq r$, the k sequences $\{S_i(kn + a)\}_{n\geq 0}$, $0 \leq a < k$, are R'–linear combinations of the S_i;

(e) There exist an integer r, r sequences $S = S_1, S_2, \ldots, S_r$, and k matrices $B_0, B_1, \ldots, B_{k-1}$ in $M_{r,r}(R')$ such that if

$$V(n) = \begin{pmatrix} S_1(n) \\ \vdots \\ S_r(n) \end{pmatrix},$$

one has $V(kn + a) = B_a V(n)$ for $0 \leq a < k$.

Proof.

(a) $\Rightarrow$ (b): Let $\mathcal{K}$ denote the k–kernel of $S(n)$. Then $\langle \mathcal{K} \rangle$, the module generated by $\mathcal{K}$, is finitely generated, so there exist sequences $S_1, S_2, \ldots, S_k$ such that

$$\langle \mathcal{K} \rangle = \langle S_1, S_2, \ldots, S_k \rangle.$$

But then each S_i is necessarily a finite linear combination of elements from $\mathcal{K}$, and there are only finitely many S_i, so $\langle \mathcal{K} \rangle$ is generated by only finitely many members of $\mathcal{K}$.

(b) $\Rightarrow$ (c): Let the k–kernel of $\{S(n)\}_{n \geq 0}$ be generated by a finite set of its subsequences of the specified form, say

$$S(k^{f_i} n + b_i)$$

for $1 \leq i \leq i'$. Let $E = \max_{1 \leq i \leq i'} f_i$. Then for all $e_j > E$, we can write

$$S(k^{e_j} n + a_j) = \sum_i c_{ij} S(k^{f_{ij}} n + b_{ij}),$$

where $f_{ij} \leq E$ and $0 \leq b_{ij} < k^{f_{ij}}$.

(c) $\Rightarrow$ (d): Take as the r sequences the set $\mathcal{K}$ of subsequences $S_i(n) = S(k^{f_i} n + b_i)$ with $0 \leq f_i \leq E$ and $0 \leq b_i < k^{f_i}$. Then

$$S_i(kn + a) = S(k^{f_i}(kn + a) + b_i) = S(k^{f_i + 1} n + a k^{f_i} + b_i),$$

which, if $f_i + 1 \leq E$, is an element of $\mathcal{K}$, and if $f_i + 1 > E$, is a linear combination of elements of $\mathcal{K}$.

(d) $\Rightarrow$ (e): Follows trivially.

(e) $\Rightarrow$ (a): We need to see that $S(k^e n + a)$ is a linear combination of the S_i. Express a in base k (possibly with leading zeroes) as

$$\sum_{0 \leq i < e} a_i k^i;$$

then it is easy to see that

$$V(k^e n + a) = B_{a_0} B_{a_1} \cdots B_{a_{e-1}} V(n),$$

and this expresses $S(k^e n + a)$ as a linear combination of the S_i. ■

Remarks.

• Note that in parts (d) and (e) of the theorem, the sequences S_i can be taken to be in the k–kernel of S.

• Part (e) of the theorem gives a substitution-like definition, which can be compared to the linear k–substitutions of Liardet [Li], which generate exactly the k–automatic sequences.

• The *dimension* of the R'–module generated by the k–kernel is an invariant that may be interpreted as a measure of complexity of the sequence $\{S(n)\}_{n\geq 0}$.

• We note that every ultimately periodic sequence is (R', k)–regular for all R' and k.

Our next theorem illustrates a connection between k–regular sequences and k–automatic sequences:

Theorem 2.3.

A sequence is (R', k)–regular and takes on only finitely many values if and only if it is k–automatic.

Proof.

If a sequence is k–automatic, it is by definition finitely valued, and since its k–kernel is finite, it generates a finitely generated module.

Now suppose $S(n)$ is k–regular and takes on finitely many values. From Theorem 2.2 (e), there exist sequence $S = S_1, S_2, \ldots, S_r$ (which can be taken in the k–kernel of S) and matrices $B_0, B_1, \ldots B_{k-1}$ such that

$$V(n) = \begin{pmatrix} S_1(n) \\ S_2(n) \\ \vdots \\ S_r(n) \end{pmatrix}$$

satisfies $V(kn + a) = B_a V(n)$ for $0 \leq a < k$ and $n \geq 0$. Let $\mathcal{V}$ be the (finite) set of values of $\{V(n)\}_{n\geq 0}$, and define the k–homomorphism σ by $\sigma(v) = w_0 w_1 \cdots w_{k-1}$, where $v \in \mathcal{V}$ and $w_a = B_a v$ for $0 \leq a < k$. Then the infinite word

$$V(0)V(1)V(2)\cdots$$

is a fixed point of σ and $S_1(n)$ is an image of this fixed point. Hence $S(n)$ is k–automatic. ■

Corollary 2.4.

If $S(n)$ is $(\mathbb{Z}, k)$–regular, then for all $m \geq 1$, $\{S(n) \bmod m\}_{n\geq 0}$ is k-automatic.

Remark.

The converse does not hold. Let $S(n) = 2^n$ and use Theorem 2.11 below.

We now investigate the closure properties of k–regular sequences:

Theorem 2.5.

Let $\{S(n)\}_{n\geq 0}$ and $\{T(n)\}_{n\geq 0}$ be k–regular sequences. Then so are $S+T = \{S(n)+T(n)\}_{n\geq 0}$, $\alpha S = \{\alpha S(n)\}_{n\geq 0}$, and $ST = \{S(n)T(n)\}_{n\geq 0}$.

Proof.

Let $S_1 = S, S_2, \ldots, S_r$ (respectively $T_1 = T, T_2, \ldots, T_r$) be a system of generators for the module generated by the k–kernel of S (respectively T). Then it is easy to see that the $r + r'$ sequences $S_1, \ldots, S_r, T_1, \ldots, T_{r'}$ generate the module generated by the k–kernel of $S + T$. Similarly, the rr' sequences $S_i T_j$, $1 \leq i \leq r$, $1 \leq j \leq r'$ generate the module generated by the k–kernel of ST. Finally, the sequences αS_i, $1 \leq i \leq r$, generate the module generated by the k–kernel of αS. ■

Remarks.

We observe that some simple transformations do *not* preserve k–regularity.

• Let $S(n), T(n)$ be $(\mathbb{Z}, k)$–regular sequences with $T(n) \neq 0$ for all n. Then the sequence $S/T = \{S(n)/T(n)\}_{n\geq 0}$ need not even be $(\mathbb{Q}, k)$–regular.

For example, define $T(2n) = n + 1$, $T(2n + 1) = T(n) + 1$ for $n \geq 0$. Define $T_j(n) = T(2^j n + 2^{j-1} - 1)$. Then it is easy to see that $T_j(n) = n + j$ for $j \geq 1$.

Suppose $1/T(n)$ were $(\mathbb{Q}, 2)$–regular. Then the module generated by the sequences

$$1/T_1(n), 1/T_2(n), 1/T_3(n), \ldots$$

would have finite rank. Then for some $m \geq 1$, the $m \times m$ matrix M_{ij} defined by

$$M_{ij} = 1/T_j(i-1) = 1/(i+j-1),$$

$1 \leq i, j \leq m$, would have determinant 0. But M_{ij} is a Hilbert matrix and is well-known to have nonzero determinant, a contradiction, and the conclusion follows.

• We note that k–regular sequences are not closed under absolute value (and hence not closed under max and min). Consider the function $f(n) = e_0(n) - e_1(n)$, where $e_0(n)$ counts the number of 0's in the binary expansion of n, and $e_1(n)$ counts the number of 1's in the binary expansion of n. It is easily verified that $e_0(n)$ and $e_1(n)$ are k–regular; hence so is $f(n)$. But $|f(n)|$ is not k–regular. For we have

$$f(2^j n) = |e_0(n) - e_1(n) + j|$$

for $n \geq 1$ and $j \geq 0$. Now suppose there were a linear dependency among these subsequences; i. e. there exist a, b such that

$$|n + a| = \sum_{a+1 \leq i \leq b} c_i |x + i|$$

for all integers n. For $n \geq -(a + 1)$ the right side is of the form $An + B$ and hence monotone; but the left side is not, a contradiction.

• We also note that k–regular sequences are not closed under composition. As mentioned above, $e_1(n)$, the number of 1's in the binary expansion of n, is 2–regular, as is the

function $f(n) = n^2$. However, the composition $e_1(f(n)) = e_1(n^2)$ is not 2–regular; if it were, then by Corollary 2.4, $e_1(n^2)$ mod 2 would be 2–automatic. However, $e_1(n^2)$ mod 2 is not 2–automatic, by results of Allouche [A2].

In the next theorem, we show that if a sequence is k–regular, then so is the subsequence obtained by periodic indexing:

Theorem 2.6.

Let $\{S(n)\}_{n\geq 0}$ be a k–regular sequence. Then for $a \geq 1, b \geq 0$, the sequence $\{S(an + b)\}_{n\geq 0}$ is k–regular.

Proof.

Define $T(n) = S(an + b)$.

Suppose $S(n)$ is k–regular. Then the module generated by its k–kernel is generated by $S_1(n), S_2(n), \ldots, S_r(n)$. We claim that each sequence in the k–kernel of $T(n)$ can be expressed as a linear combination of $S_i(an + c)$, for $1 \leq i \leq r$ and $0 \leq c < a + b$.

Proof: Take an element of the k–kernel of $T(n)$, say $T(k^e n + j)$, $0 \leq j < k^e$. Write $ja + b = d \cdot k^e + f$, where $0 \leq f < k^e$. Then

$$\begin{aligned} T(k^e n + j) &= S(a(k^e n + j) + b) \\ &= S(k^e(an + d) + f), \end{aligned}$$

Notice that since $0 \leq j < k^e$, we have $0 \leq d < a + b$. Now the module generated by the k–kernel of $\{S(n)\}_{n\geq 0}$ is finitely generated, so $S(k^e m + f) = \sum_j c_j S_j(m)$ for constants c_j. Hence it follows that

$$S(k^e(an + d) + f) = \sum_j c_j S_j(an + d),$$

and the result follows. ■

Remark.

Let us define S indexed by negative arguments to be 0. For example, $\{S(n - 1)\}_{n\geq 0}$ is the sequence $\{S(n)\}_{n\geq 0}$ with a 0 tacked on the front.

Then it is easy to see that the preceding theorem holds even when $b < 0$.

Theorem 2.7.

Let $\{S(n)\}_{n\geq 0}$ be a sequence such that there exists an $a \geq 2$ such that $\{S(an+i)\}_{n\geq 0}$ is k–regular for $0 \leq i < a$. Then $\{S(n)\}_{n\geq 0}$ is k–regular.

Proof.

For $0 \leq i < a$, define

$$T_i(n) = \begin{cases} S(n), & \text{if } n \equiv i \pmod{a}; \\ 0, & \text{if } n \not\equiv i \pmod{a}. \end{cases}$$

Also, write $S_i(n) = S(an+i)$. Then it is easy to see that each sequence $T_i(n)$ is k–regular; indeed, $T_i(k^j n + c)$ is either the 0-sequence or the sequence $S_i(k^j n + c')$ interspersed with

groups of $a/\gcd(a,k^j)-1$ zeros. Hence the k–kernel of $T_i(n)$ is finitely generated. Finally, we see that

$$S(n) = \sum_{0 \le i < a} T_i(n),$$

which shows that $\{S(n)\}_{n\ge 0}$ is k–regular. ■

Remark.

From Theorems 2.6 and 2.7 it follows that if $S(n)$ is k–regular, and r is a rational number, then $S(\lceil rn \rceil)$ and $S(\lfloor rn \rfloor)$ are also k-regular.

Many sequence transformations from the literature preserve regularity. For example, let $\{S(n)\}_{n\ge 0}$ be a sequence, and consider its *Toeplitz transformation* $\{S'(n)\}_{n\ge 0}$ defined by $S'(2n) = S(n)$ and $S'(2n+1) = S'(n)$ for $n \ge 0$. (See [JK], [Pro]). Then we have the following, which generalizes the case of automatic sequences [A3]:

Theorem 2.8.

$\{S(n)\}_{n\ge 0}$ is 2–regular if and only if $\{S'(n)\}_{n\ge 0}$ is 2–regular.

Proof.

Suppose $\{S(n)\}_{n\ge 0}$ is 2–regular. Then the module generated by its 2–kernel is finitely generated, say by $S_1(n), \ldots, S_k(n)$. Now consider the module

$$M = \langle S'(n), S_1(n), \ldots, S_k(n) \rangle.$$

Note that $S_i(2n)$ and $S_i(2n+1)$ are linear combinations of the S_j. Also, $S'(2n) = S(n)$ and $S'(2n+1) = S'(n)$. Thus by Theorem 2.2 (d), $\{S'(n)\}_{n\ge 0}$ is 2–regular.

Now assume $\{S'(n)\}_{n\ge 0}$ is 2–regular. Then by Theorem 2.6, $S'(2n)$ is 2–regular. But $S'(2n) = S(n)$, and the result follows. ■

Theorem 2.9

Let f be an integer ≥ 1. Then $\{S(n)\}_{n\ge 0}$ is k^f–regular if and only if $\{S(n)\}_{n\ge 0}$ is k–regular.

Proof.

Suppose $\{S(n)\}_{n\ge 0}$ is k–regular. Then the module generated by its k–kernel is finitely generated and contains its k^f–kernel. Hence the module generated by its k^f–kernel is also finitely generated.

To prove the other direction, assume $\{S(n)\}_{n\ge 0}$ is k^f–regular.

We now show there exists a B such that for all $b > B$, each subsequence $S(k^b n + c)$ can be expressed as a linear combination

$$S(k^b n + c) = \sum_i d_i S(k^{b_i} n + c_i)$$

with $b_i \le B$ and $0 \le c_i < k^{b_i}$. The result will then follow from Theorem 2.2 (c).

For let us write $b = fr + s$, $0 \le s < f$ and $c = qk^{fr} + t$, $0 \le t < k^{fr}$. Then, by Theorem 2.2 (c), there exists E such that for all $r > E$ we can express

$$S((k^f)^r m + t) = \sum_i d_i S((k^f)^{r_i} m + t_i),$$

where $r_i \le E$ and $0 \le t_i < k^{fr_i}$.

Now put $m = k^s n + q$; we find

$$S((k^f)^r m + t) = S(k^b n + c) = \sum_i d_i S(k^{fr_i + s} n + q k^{fr_i} + t_i) = \sum_i d_i S(k^{b_i} n + c_i),$$

where $b_i = fr_i + s$ and $c_i = qk^{fr_i} + t_i$. Notice that $b_i < fE + f$. Also, $q \le k^s - 1$, so

$$\begin{aligned} c_i = qk^{fr_i} + t_i &\le (k^s - 1)k^{fr_i} + t_i \\ &\le k^{fr_i + s} - k^{fr_i} + t_i < k^{fr_i + s} = k^{b_i}; \end{aligned}$$

thus we may take $B = f(E + 1)$. Hence $\{S(n)\}_{n \ge 0}$ is also k–regular. ■

C. Choffrut and C. Reutenauer have pointed out that we may obtain alternative proofs of Theorems 2.6–2.9 using the notion of rational transduction [SS] and Theorem 4.3 below.

Theorem 2.10.

Let $\{S(n)\}_{n \ge 0}$ be a k–regular sequence with values in $\mathbb{C}$, the complex numbers. Then there exists a constant c such that $S(n) = O(n^c)$.

Proof.

We use the characterization of Theorem 2.2 (e). Let the base–k expansion of n be

$$a_{j-1} a_{j-2} \cdots a_1 a_0;$$

then $j \le 1 + \log_k n$. Then

$$V(n) = B_{a_0} B_{a_1} \cdots B_{a_{j-1}} V(0).$$

If v is a d-dimensional vector, define

$$\|v\| = \sum_{1 \le i \le d} |v_i|;$$

if M is a $d \times d$-matrix, define

$$\|M\| = \max_{1 \le i \le d} \sum_{1 \le j \le d} |M_{ij}|.$$

Then it is easy to see that $\|Mv\| \le \|M\| \|v\|$.

Thus we see

$$S(n) \leq \|V(n)\| \leq \|B_{a_0}\|\|B_{a_1}\|\cdots\|B_{a_{j-1}}\|\|V(0)\|.$$

Now let $c = \max_{0\leq i\leq k-1}\|B_i\|$, and $d = \|V(0)\|$. Then we have

$$S(n) \leq c^{1+\log_k n} d \leq d' n^{c'},$$

and the result follows. ■

Thus we see, for example, that $\{2^n\}_{n\geq 0}$ is not $(\mathbb{Z}, k)$–regular.

Theorem 2.11.

Let R be a Noetherian ring without zero divisors, and let $a \in R$. Then the sequence of powers $\{a^n\}_{n\geq 0}$ is (R,k)–regular if and only if $a = 0$ or a is a root of unity.

Proof.

One direction is simple, since if a is 0 or a root of of unity, then the sequence of powers is ultimately periodic, hence k–regular.

Now assume $\{a^n\}_{n\geq 0}$ is (R,k)–regular. Then there exist $r < \infty$ and λ_j, $0 \leq j < r$ such that

$$\sum_{0\leq j<r} \lambda_j a^{k^j \cdot n} = 0$$

for all $n \geq 0$.

Now recall the following identity for the Vandermonde determinant:

$$\begin{pmatrix} 1 & b_0 & b_0^2 & \dots & b_0^m \\ 1 & b_1 & b_1^2 & \dots & b_1^m \\ \vdots & \vdots & \vdots & \ddots & \dots \\ 1 & b_m & b_m^2 & \dots & b_m^m \end{pmatrix} = \prod_{i>j}(b_i - b_j).$$

From this, we see that the sequences $\{b_j^n\}_{n\geq 0}$ are linearly independent if and only if the numbers $b_1, b_2, \dots, b_m$ are distinct.

Hence the numbers $1, a^k, a^{k^2}, \dots, a^{k^r}$ are not all distinct and we must have

$$a^{k^j} = a^{k^l}$$

for some $j \neq l$. Since R has no zero-divisors, either $a = 0$ or a is a root of unity. ■

III. The ring of k–regular sequences.

Associated to every k–regular sequence $\{S(n)\}_{n\geq 0}$ is the formal power series in $R[[X]]$ defined by

$$\sum_{n\geq 0} S(n) X^n,$$

where X is an indeterminate. We call such a power series *k–regular*. In this section we show that the set of all k–regular power series forms a ring (but not a field).

Recall that the *convolution* $S \star T$ of two sequences $S(n)$ and $T(n)$ is defined as follows:

$$(S \star T)(n) = \sum_{i+j=n} S(i)T(j).$$

Theorem 3.1.

The set of k–regular sequences is closed under convolution.

Proof.

For simplicity we prove this only in the case $k = 2$.

Let us agree to write $\{A(2n)\}$ as shorthand for the sequence $\{A(2n)\}_{n\geq 0}$.

Let A and B be 2–regular sequences. The modules generated by their 2–kernels are generated by sequences $a_1, a_2, \ldots, a_{i'}$ and $b_1, b_2, \ldots, b_{j'}$, respectively. We want to find a basis for C, the module generated by the 2–kernel of $A \star B$. We write $u_{ij} = a_i \star b_j$ for $1 \leq i \leq i', 1 \leq j \leq j'$. We claim that the set $\mathcal{M}$ of $2i'j'$ sequences $\{u_{ij}(n)\}_{n\geq 0}$ and $\{u_{ij}(n-1)\}_{n\geq 0}$ generates the module C. (As in the previous section, we define $u_{ij}(-1) = 0$.)

It is clear that $\mathcal{M}$ contains all sequences of the form

$$(\{A(2^e n + i)\} \star \{B(2^f n + j)\})(n) \tag{1}$$

and

$$(\{A(2^e n + i)\} \star \{B(2^f n + j)\})(n-1). \tag{2}$$

Thus it suffices to show how to write all the sequences of the form

$$\{(A \star B)(2^g n + a)\}$$

as a linear combination of the sequences in (1) and (2).

This is done using the following formula:

$$\begin{aligned}(A \star B)(2^g n + a) = &\sum_{0\leq i\leq a} (\{A(2^g n + i)\} \star \{B(2^g n + a - i)\})(n) \\ &+ \sum_{a<j<2^g} (\{A(2^g n + j)\} \star (\{B(2^g n + 2^g + a - j)\})(n-1).\end{aligned}$$

Hence the result follows. ■

(Note: it is apparently impossible to obtain Theorem 3.1 using the standard tools of rational series, such as rational transductions.)

It follows from Theorem 3.1 that if the sequence $\{S(n)\}_{n\geq 0}$ is k–regular, then so is its running sum $\{\sum_{0\leq j\leq n} S(j)\}_{n\geq 0}$.

Since the convolution of sequences is equivalent to (ordinary) multiplication of the associated power series, we have:

Corollary 3.2.

The set of k–regular power series forms a ring.

Remark.

The set of k–regular power series does not form a field. This follows from the identity

$$\frac{1}{1-2X} = 1 + 2X + 4X^2 + 8X^3 + \cdots$$

and the fact that $\{2^n\}_{n\geq 0}$ is not k–regular (Theorem 2.10).

Theorem 3.3.

Let F be an algebraically closed field (e. g., $\mathbb{C}$). Let $\{S(n)\}_{n\geq 0}$ be a sequence with values in F. Let $f(X) = \sum_{n\geq 0} S(n)X^n$ be a formal power series in $F[[X]]$. Assume that $f(X)$ represents a rational function of X; i. e. there exist polynomials $p(X)$, $q(X)$ such that $f(X) = p(X)/q(X)$. Then $\{S(n)\}_{n\geq 0}$ is k–regular if and only if the poles of f are roots of unity.

Proof.

Note that by assumption, 0 is not a pole of f.

Suppose the poles of f are roots of unity. Then using expansion by partial fractions, we can write

$$f(X) = \sum_i \frac{c_i}{(1-\zeta_i X)^{e_i}}$$

where $c_i \in F$, the e_i are non-negative integers, and each ζ_i is a root of unity. To prove the coefficients of f form a k–regular sequence, it clearly suffices to show that $(1-\zeta_i X)^{-1}$ is k–regular. But this power series has periodic coefficients and so is k–regular.

Now suppose $f(X) = p(X)/q(X)$ for polynomials p, q, and f is k–regular. Let $1/\zeta$ be one of the poles of f; we may assume $\zeta \neq 0$. We can then write

$$f(X) = \frac{p(X)}{q(X)} = \frac{r(X)}{s(X)(1-\zeta X)^e},$$

where $r(x), s(X)$ are polynomials and $r(X)$ and $1-\zeta X$ are relatively prime. Then there exist two polynomials $u(X), v(X)$ such that

$$u(X)r(X) + v(X)(1-\zeta X)^e = 1.$$

Now $u(X)f(X)s(X) + v(X)$ is also a k–regular power series, and we have

$$u(X)f(X)s(X) + v(X) = (1-\zeta X)^{-e}. \tag{3}$$

Thus $(1-\zeta X)^{-e}$ is k–regular. But $(1-\zeta X)^{e-1}$ is a polynomial and hence a k–regular power series, so its product with (3) is k–regular and thus $(1-\zeta X)^{-1}$ is k–regular. But the coefficients of this power series are ζ^n, which by Theorem 2.11 is k–regular if and only if ζ is a root of unity.

This completes the proof. ■

Remarks.

• We note that Theorem 3.3 gives us the following characterization of k–regular sequences associated with rational formal power series: they must be linear recurrences whose characteristic polynomial is a product of cyclotomic polynomials. See, for example, Section VII, Example 18.

• Also note that if $R = \mathbb{Q}$, then (using Corollary 4.2 below) the radius of convergence of a k–regular power series is 1, and such a series either represents a rational function or has the unit circle as a natural boundary.

IV. Rational series and k–regular sequences.

At first glance, it might seem that there is no relationship between k–regular power series and the theory of $\mathbb{Z}$–rational formal series, as described in [SS], [BR] [E, Chap. V]. For $\sum_{n\geq 0} 2^n X^n$ is $\mathbb{Z}$–rational, but is not k–regular. Similarly, $\sum_{n\geq 0} e_1(n)X^n$ is k–regular, but is not $\mathbb{Z}$–rational. (Here $e_1(n)$ counts the number of 1's in the base–k expansion of n).

Nevertheless, there *is* a relationship which can be roughly described as follows: 2–regular power series are the "binary" analogue of $\mathbb{Z}$–rational formal series in one variable. Alternatively, $\mathbb{Z}$–rational series in one variable are the "unary" analogue of k–regular power series.

In this section, we develop this relationship between k–regular sequences and $\mathbb{Z}$–rational formal series. From this, we get a machine model for the k–regular sequences. This model plays the same role as the ordinary finite automaton does for k–automatic sequences. We also prove that all k–regular sequences can be computed quickly.

We introduce some notation that will be used throughout this section. Let k be fixed and define $\Sigma = \{0, 1, \ldots, k-1\}$. We need a way to uniquely associate integers with strings giving their base–k representation. If

$$n = \sum_{0\leq i<e} a_i k^i,$$

and $a_{e-1} \neq 0$, then we say that the string $a_{e-1}a_{e-2}\ldots a_1 a_0$ is the *standard base–k representation* of n. Note that the standard representation of 0 is the empty string. The set of *all* standard representations is just $\epsilon + (\Sigma - 0)\Sigma^*$.

First, we prove a useful lemma:

Lemma 4.1.

Let $\{S(n)\}_{n\geq 0}$ be a sequence with entries in R. Then $\{S(n)\}_{n\geq 0}$ is (R', k)–regular if and only if there exist matrices M_0, $M_1, \ldots, M_{k-1}$ with entries in R' and vectors λ, κ with entries in R such that

$$S(n) = \lambda M_{a_0} M_{a_1} \ldots M_{a_{e-1}} \kappa,$$

where $a_{e-1}a_{e-2}\ldots a_1 a_0$ is the standard base–k representation of n.

Proof.

Suppose $S(n)$ is k–regular. Then by Theorem 2.2 (e), we know that there exist matrices $M_0, \ldots M_{k-1}$ such that

$$V(kn+a) = M_a V(n),$$

where

$$V(n) = \begin{pmatrix} S_1(n) \\ \vdots \\ S_r(n) \end{pmatrix},$$

and $S(n) = S_1(n)$. Hence by setting $\kappa = V(0)$ and $\lambda = [1\ 0\ 0 \cdots 0]$, we see that

$$V(n) = \lambda M_{a_0} M_{a_1} \ldots M_{a_{e-1}} \kappa$$

for all $n \geq 0$.

Now suppose $S(n) = \lambda M_{a_0} \ldots M_{a_{e-1}} \kappa$ for all $n \geq 0$, where $a_{e-1} \cdots a_0$ is the standard base–k representation of n. Define $V(n) = M_{a_0} \cdots M_{a_{e-1}} \kappa$ and

$$V(n) = \begin{pmatrix} v_1(n) \\ \vdots \\ v_r(n) \end{pmatrix}.$$

Then

$$V(kn+a) = M_a M_{a_0} \cdots M_{a_{e-1}} \kappa = M_a V(n),$$

except possibly when $n = 0$ and $a = 0$. (This special case arises because the standard representation of kn is the string $a_{e-1} \cdots a_1 a_0 0$, for $n \geq 1$, but not for $n = 0$.) In this case, by setting $v' = V(0) - M_0 V(0)$ we see

$$V(kn) = M_0 V(n) + U(n) v'$$

for all $n \geq 0$, where $U(n)$ denotes the sequence that is 1 when $n = 0$ and 0 otherwise.

Then by Theorem 2.2 (d), we see that each of the sequences $v_1(n), \ldots, v_r(n)$ is k–regular. But then $S(n) = \lambda V(n)$ is k–regular, by Theorem 2.5. ■

Corollary 4.2. *Suppose $\{S(n)\}_{n\geq 0}$ is a $(\mathbb{Z}, k)$–regular sequence with values in $\mathbb{Q}$. Then there exist an integer r and a $(\mathbb{Z}, k)$–regular sequence $\{T(n)\}_{n\geq 0}$ with values in $\mathbb{Z}$ such that $S(n) = T(n)/r$.*

Proof. By Lemma 4.1, we have

$$S(n) = \lambda M_{a_0} M_{a_1} \cdots M_{a_{e-1}} \kappa$$

where $a_{e-1} \cdots a_1 a_0$ is the standard base–k representation of n. The matrices M_i have integral entries, and the vectors λ and κ have rational entries. Let g be the least common

multiple of the denominators of entries in λ, and g' be the least common multiple of the denominators of entries in κ. Then $T(n) = (g\lambda)M_{a_0}M_{a_1}\cdots M_{a_{e-1}}(g'\kappa)$ is a $(\mathbb{Z}, k)$–regular sequence with values in $\mathbb{Z}$. The result follows by putting $r = gg'$. ■

Now we show how k–regular sequences are related to $\mathbb{Z}$–rational formal series. Let $x_0, x_1, \ldots, x_{k-1}$ be non-commuting variables. If $w = w_1 \cdots w_r \in \Sigma^*$, then define $x_w = x_{w_1} \cdots x_{w_r}$. Let τ be the map that sends n to $x_{a_0}x_{a_1}\ldots x_{a_{e-1}}$, where the standard base–k representation of x is the string $a_{e-1}\cdots a_1 a_0$.

Theorem 4.3.
$\{S(n)\}_{n\geq 0}$ is k–regular if and only if the formal series

$$\sum_{n\geq 0} S(n)\tau(n)$$

is $\mathbb{Z}$–rational.

For example, in the case $k = 2$ we have

$$\sum_{n\geq 0} S(n)\tau(n) = S(0) + S(1)x_1 + S(2)x_0x_1 + S(3)x_1x_1 + S(4)x_0x_0x_1 + \cdots.$$

Proof. Suppose $\{S(n)\}_{n\geq 0}$ is k–regular. Then by Lemma 4.1, there exist matrices $M_0, \ldots, M_{k-1}$ such that

$$S(n) = \lambda M_{a_0} \cdots M_{a_{e-1}} \kappa.$$

But by the fundamental theorem for $\mathbb{Z}$–rational formal series (see, e.g. [SS, Theorem 2.3]),

$$T = \sum_{w\in\Sigma^*} \lambda M_w \kappa x_w$$

is $\mathbb{Z}$–rational. This is essentially the series $\sum_{n\geq 0} S(n)\tau(n)$, but it also contains terms that correspond to non-standard base–k representations of n. Let A be the set of standard base–k representations (e. g. those not beginning with a 0). Then as above, $A = \epsilon + (\Sigma - 0)\Sigma^*$, and so A is regular. Let A^R denote the set of reversals of strings in A; then A^R is also regular. Now

$$U = \text{char } A^R = \sum_{w\in A^R} x_w$$

is a $\mathbb{Z}$–rational formal series (see, e. g. [SS, Corollary 5.4 (iii)]). Then $T \odot U$ (the Hadamard product) is equal to $\sum_{n\geq 0} S(n)\tau(n)$, and since $\mathbb{Z}$–rational series are closed under $\odot$ (see, e. g. [SS, Theorem 4.4]), the result follows.

Now suppose $\sum_{n\geq 0} S(n)\tau(n)$ is $\mathbb{Z}$–rational. Then again by the definition of τ and the fundamental theorem we have $S(n) = \lambda M_w \kappa$, where $w = a_0 a_1 \cdots a_{e-1}$, and $a_{e-1}\cdots a_1 a_0$ is the standard base–k representation of n. This completes the proof. ■

Theorem 4.3 allows us to use the well-developed theory of $\mathbb{Z}$–rational series to discuss the properties of k–regular sequences, at least in some cases. We continue this below in Section V. Now, however, we sketch a description of our machine model.

This model is essentially the same as that first given by Schützenberger [Sch]. However, we repeat the description for completeness.

Let us define what we call a *matrix machine*. It is a finite-state machine with auxiliary storage in the form of a column vector $v \in R_{j1}$ for some $j > 0$. Here is how the machine operates: Suppose we are in state q. Upon reading a symbol a from the input, the machine first replaces v with Mv, where $M = M(q, a)$ is a $j \times j$ matrix. Then the machine moves to a new state $\delta(q, a)$. The output is determined as follows: when the last input symbol is read, we are in state q'. There is a row vector $\lambda(q')$, and the output is the scalar $\lambda(q')v$.

Now consider the case where the input is the base–k representation of an integer n, starting with the most significant digit, and the matrix machine computes $S(n)$. We claim this is precisely the class of k–regular sequences. By Lemma 4.1, this equivalence is easily seen in the case of 1-state machines. Thus to prove the equivalence it suffices to prove the following

Theorem 4.4 (Schützenberger).

A matrix machine with r states can be simulated by a matrix machine with 1 state.

Proof.

To simplify the exposition we show how to do this in the case where j, the size of the vectors and matrices involved, equals 1.

The idea is to replace the single element v by a vector v' of size r. All of the entries of v' will be zero, except for a single entry which equals v. We code the current state by the position of v inside v'; if it is in position i, we are currently in state i. Instead of multiplying by $M(q, a)$ we multiply by the matrix PQ, where $Q_{ii} = M(q_i, a)$, $0 \le i \le r-1$, and P is a permutation matrix defined as follows:

$$P_{ij} = \begin{cases} 1, & \text{if } \delta(q_j, a) = q_i \\ 0, & \text{otherwise.} \end{cases}$$

Finally, $\lambda(q_i)$ is the vector consisting of all ones.

The correctness of the construction is left to the reader. To extend this proof to the case $j > 1$, we replace all entries by block matrices. ■

Corollary 4.5.

The n-th term of a k–regular sequence can be computed using $O(\log n)$ operations, where an operation is an addition or multiplication of elements in the ring R.

Corollary 4.6.

The n-th term of a k–regular sequence over $\mathbb{Z}$ can be computed in time polynomial in $\log n$.

Remarks.

• At first glance, our matrix machines would also seem to be similar to the linear sequential machines (LSM) of Harrison [Har1]. This is not the case, however. Our input symbols a are chosen from an arbitrary alphabet Σ, while the LSM model uses k-tuples chosen from a field. Our model allows a different $n \times n$ matrix $M(q,a)$ for every state q and input symbol a, whereas the LSM model uses exactly two matrices A and B and defines a transition by

$$\delta(q,a) = Aq + Ba.$$

Our model allows the matrices to contain arbitrary ring elements, whereas the LSM model uses a field. Finally, in our model we are only interested in the output associated with the final state, rather than the string of outputs associated with each state visited.

• We mention a connection between $(\mathbb{Z},k)$–regular sequences with values in $\mathbb{Z}$ and the group $\Gamma_k(\mathbb{Z})$ of Morton and Mourant [MM]. Indeed, every sequence $\{S(n)\}_{n\geq 0}$ in $\Gamma_k(\mathbb{Z})$ is k–regular, as it is easily seen that $\{S(n)\}_{n\geq 0} \in \Gamma_k(\mathbb{Z})$ if and only if the sequence $\{S(n) - S(\lfloor n/k \rfloor)\}_{n\geq 0}$ is periodic.

V. The zero-set of a k–regular sequence.

Let $\{S(n)\}_{n\geq 0}$ be a k–regular sequence. In this section, we discuss the set

$$Q = \{n \mid S(n) = 0\},$$

or, more precisely, the set $Z(S)$ of strings of the standard base–k representations of elements of Q. We call this set the *zero-set* of the sequence $\{S(n)\}_{n\geq 0}$.

We also discuss the set $\overline{Z}(S)$, the set of strings of the standard base–k representations of n such that $S(n) \neq 0$. (This set is essentially the *support* of the associated $\mathbb{Z}$–rational power series.) Note that

$$Z(S) + \overline{Z}(S) = \epsilon + (\Sigma - 0)\Sigma^*,$$

where $\Sigma = \{0, 1, \ldots, k-1\}$.

Theorem 5.1. *The set $Z(S)$ is simultaneously in logarithmic space and polynomial time. The set $Z(S)$ is also in the complexity class* NC.

Proof.

The first statement follows immediately from results of Lipton and Zalcstein [LZ].

The second statement is left to the reader. ■

Theorem 5.2. *For fixed $k \geq 2$, it is undecidable if a given k–regular sequence $\{S(n)\}_{n\geq 0}$ has a zero term. In other words, it is undecidable if $Z(S)$ is nonempty.*

Proof.

To specify the k–regular sequence $S(n)$, it is necessary to agree on a representation. We assume we have been given the matrices in Lemma 4.1 or Theorem 2.2 (e).

As in [SS, Theorem 12.1], we reduce the problem of determining whether or not an arbitrary multivariate polynomial equation

$$p(x_1, x_2, \ldots x_r)$$

has a solution in non-negative integers (Hilbert's tenth problem) to the problem of whether $Z(S)$ is nonempty. The result will then follow by the celebrated result of Davis-Matijacevič-Putnam-Robinson [Dav].

Suppose we are given $p(x_1, x_2, \ldots, x_r)$. We encode this equation as a k–regular sequence as follows. First, we choose f such that $k^f \geq r+1$. We now represent the variable x_j by $e_j(n)$, the number of j's in the base–k^f expansion of n. Clearly for each r-tuple of non-negative integers $(b_1, b_2, \ldots, b_r)$, there exists an n for which

$$(e_1(n), \ldots, e_r(n)) = (b_1, \ldots, b_r).$$

Now $S(n) = p(e_1(n), e_2(n), \ldots, e_r(n))$ is k^f–regular and its matrix representation can be computed with a recursive algorithm. But by Theorem 2.9, $S(n)$ is also k–regular; furthermore, the corresponding matrices are effectively determinable. Clearly $Z(S)$ is nonempty if and only if $p(x_1, x_2, \ldots, x_r)$ has a solution in non-negative integers. ■

As in [SS, p. 124], we can also give a more explicit example:

Theorem 5.3. *There exists a k–regular sequence $\{S(n)\}_{n\geq 0}$ such that neither $Z(S)$ nor $\overline{Z}(S)$ are context-free.*

Proof.

Define $S(n) = e_1(n)^2 - e_0(n)$. It is not hard to verify that $\{S(n)\}_{n\geq 0}$ is k–regular. (Indeed, it will follow from Theorem 6.1.)

Now suppose $Z(S)$ is context-free. Then $Z(S) \cap 1^+0^* = \{1^n 0^{n^2} \mid n \geq 1\}$ would also be context-free. But this can easily seen to be false, using the pumping lemma.

Now suppose $\overline{Z}(S)$ is context-free. Then $L_1 = Z(S) \cap 1^+0^* = \{1^n 0^r \mid n \geq 1, r \neq n^2\}$ would be context-free. By a theorem of Ginsburg and Spanier [GS, Theorem 6.2, Corollary 2], $L_2 = 1^*0^* - L$ would be context free. But $L_2 \cap 1^+0^+ = \{1^n 0^{n^2} \mid n \geq 1\}$, which is not context-free, a contradiction. ■

VI. Some "Fourier" expansions.

For simplicity, all results and proofs in this section assume $k = 2$.

We introduce some notation that will be used throughout this section. Let $n_{(2)}$ denote the string in $A = \epsilon + 1(0+1)^*$ that represents n in base 2. If s is a string in A, let $v(S)$ denote the integer represented by s. Let $|s|$ denote the length of the string s. Let $\lambda(n)$ be the integer obtained from n by deleting the most significant bit of its base–2 expansion. Let m and n be integers; we write m suff n for the relation: the string $m_{(2)}$ is a suffix of the string $n_{(2)}$. Define $E = 1(0+1)^*$. Let $P \in E$, and let $e_P(n)$ denote the number of

(possibly overlapping) occurrences of P in the base-k expansion of n. Let $x_P(n)$ be the function that takes the value 1 if P is a suffix of $n_{(2)}$, and 0 otherwise.

Morton and Mourant proved [MM] that any sequence $\{S(n)\}_{n\geq 0}$ taking values in $\mathbb{Z}$ has a unique expansion as an infinite sum, as follows:

$$S(n) = S(0) + \sum_{P \in E} \hat{S}(v(P)) e_P(n).$$

Here the "Fourier" coefficients $\hat{S}(m)$ are integers. We define $\hat{S}(0) = S(0)$, and call the sequence $\{\hat{S}(n)\}_{n\geq 0}$ the *pattern transform* of $\{S(n)\}_{n\geq 0}$.

In this section, we prove the following result: a sequence is 2–regular if and only if its pattern transform is 2–regular. First, however, we show that the sequences e_P themselves are 2–regular.

Theorem 6.1.

The sequence $\{e_P(n)\}_{n\geq 0}$ is 2–regular for any pattern $P \in E$.

Proof.

Let us introduce the following notation: if $w = w_1 w_2 \cdots w_{j'}$ is a string and $j \leq j'$, then

$$\text{take}(j, w) = w_1 w_2 \cdots w_j.$$

We claim that each element of the 2–kernel can be written as a linear combination of the sequences $e_P(2^f n + a)$ for $0 \leq f < |P|$ and $0 \leq a < 2^f$ and the constant sequence 1.

Proof: Consider an element of the 2–kernel, $e_P(2^f n + a), 0 \leq a < 2^f$. Then if $f \leq |P| - 1$, this sequence is already in the list above. Otherwise, $f \geq |P|$. Then $2^f n + a$ can be written in base 2 as

$$n_{(2)} a'$$

where $|a'| = f$ and $v(a') = a$. Then

$$e_P(2^f n + a) = e_P(2^{|P|-1} n + c) + e_P(a),$$

where $c = v(\text{take}(|P| - 1, a'))$.

Now the first term on the right is in the list above, and the second term is a constant multiple of the constant sequence 1. Hence $e_P(2^f n + a)$ is a $\mathbb{Z}$–linear combination of elements in the list, and this completes the proof. ■

Corollary 6.2.

$\{e_P(an + b)\}_{n\geq 0}$ is 2–regular for all $a, b \geq 0$.

Theorem 6.3.

$\{S(n)\}_{n\geq 0}$ is 2–regular if and only if $\{\hat{S}(n)\}_{n\geq 0}$ is 2–regular.

First we prove two lemmas.

Lemma 6.4.
For all $n \geq 0$ we have

$$S(2n) = S(n) + \sum_{\substack{m \geq 1 \\ m \text{ suff } n}} \hat{S}(2m)$$

and

$$S(2n+1) = S(n) + \hat{S}(1) + \sum_{\substack{m \geq 1 \\ m \text{ suff } n}} \hat{S}(2m+1).$$

Proof.

$$\begin{aligned}
S(2n) &= S(0) + \sum_P \hat{S}(v(P)) e_P(2n) \\
&= S(0) + \hat{S}(v(1)) e_1(2n) + \sum_P \hat{S}(v(P0)) e_{P0}(2n) + \sum_P \hat{S}(v(P1)) e_{P1}(2n) \\
&= S(0) + \hat{S}(1) e_1(n) + \sum_P \hat{S}(v(P0)) e_{P0}(n) + \sum_P \hat{S}(v(P0)) x_P(n) + \sum_P \hat{S}(v(P1)) e_{P1}(n) \\
&= S(0) + \sum_P \hat{S}(v(P)) e_P(n) + \sum_P \hat{S}(v(P0)) x_P(n) \\
&= S(n) + \sum_P \hat{S}(v(P0)) x_P(n) \\
&= S(n) + \sum_{\substack{m \geq 1 \\ m \text{ suff } n}} \hat{S}(2m).
\end{aligned}$$

The formula for $S(2n+1)$ is proved similarly; the extra term $\hat{S}(1)$ comes from the fact that

$$\hat{S}(v(1)) e_1(2n+1) = \hat{S}(1)(e_1(n) + 1) = \hat{S}(1) e_1(n) + \hat{S}(1).$$

This completes the proof. ■

Lemma 6.5.
For all $n \geq 1$ we have

$$\hat{S}(2n) = S(2n) - S(n) - S(2\lambda(n)) + S(\lambda(n)).$$

For all $n \geq 1$ we have

$$\hat{S}(2n+1) = S(2n+1) - S(n) - S(2\lambda(n)+1) + S(\lambda(n)).$$

Proof.

Notice first that

$$\begin{aligned}\{m \geq 1 \mid m \text{ suff } n\} &= \{m \geq 1 \mid (m \text{ suff } n) \text{ and } (m \neq n)\} \cup \{n\} \\ &= \{m \geq 1 \mid m \text{ suff } \lambda(n)\} \cup \{n\},\end{aligned}$$

the unions being disjoint.

Hence, using Lemma 6.4, we find

$$\begin{aligned}S(2n) - S(n) &= \sum_{\substack{m \geq 1 \\ m \text{ suff } n}} \hat{S}(2m) \\ &= \hat{S}(2n) + \sum_{\substack{m \geq 1 \\ m \text{ suff } \lambda(n)}} \hat{S}(2m),\end{aligned}$$

which can be rewritten as:

$$\hat{S}(2n) = (S(2n) - S(n)) - (S(2\lambda(n)) - S(\lambda(n))).$$

The second formula is obtained in a slightly different manner:

$$\begin{aligned}S(2n+1) - S(n) &= \hat{S}(1) + \sum_{\substack{m \geq 1 \\ m \text{ suff } n}} \hat{S}(2m+1) \\ &= \hat{S}(2n+1) + \hat{S}(1) + \sum_{\substack{m \geq 1 \\ m \text{ suff } \lambda(n)}} \hat{S}(2m+1),\end{aligned}$$

which can be rewritten as:

$$\begin{aligned}\hat{S}(2n+1) &= (S(2n+1) - S(n) - \hat{S}(1)) - (S(2\lambda(n)+1) - S(\lambda(n)) - \hat{S}(1)) \\ &= S(2n+1) - S(n) - S(2\lambda(n)+1) + S(\lambda(n)).\end{aligned}$$

This completes the proof of Lemma 6.5. ■

We are now ready to prove Theorem 6.3:

Proof.

Suppose first that S is 2–regular and let $\{S_1 = S, S_2, \ldots, S_r\}$ be a finite set of generators for the $\mathbb{Z}$–module generated by its 2–kernel. Define

$$U(n) = \begin{cases} 1, & \text{if } n = 0; \\ 0, & \text{otherwise.} \end{cases}$$

Consider the $\mathbb{Z}$–module $\mathcal{M}$ generated by the $S_j(n)$, the $S_j(\lambda(n))$, and $U(n)$; i. e.

$$\mathcal{M} = \langle S_1, S_2, \ldots, S_r, S_1 \circ \lambda, S_2 \circ \lambda, \ldots, S_r \circ \lambda, U, \hat{S} \rangle.$$

To prove that $\hat{S}$ is 2–regular, it suffices to prove that for each sequence $V(n)$ contained in the list of generators for $\mathcal{M}$, the subsequences $\{V(2n)\}_{n\geq 0}$ and $\{V(2n+1)\}_{n\geq 0}$ are in $\mathcal{M}$.

For $S_1, \ldots, S_r$, this follows from the 2–regularity of S. By Lemma 6.5,

$$\hat{S}(2n) = S(2n) - S(n) - S(2\lambda(n)) + S(\lambda(n)) + \hat{S}(0)U(n),$$

and so $\mathcal{M}$ contains $\{\hat{S}(2n)\}_{n\geq 0}$. Similarly, Lemma 6.5 implies that

$$\hat{S}(2n+1) = S(2n+1) - S(n) - S(2\lambda(n)+1) + S(\lambda(n)) + \hat{S}(1)U(n);$$

hence $\mathcal{M}$ contains $\{\hat{S}(2n+1)\}_{n\geq 0}$.

For $S_i \circ \lambda$, we have $S_i(\lambda(2n)) = S_i(2\lambda(n))$. Similarly,

$$\begin{aligned} S_i(\lambda(2n+1)) &= \begin{cases} S_i(2\lambda(n)+1), & \text{if } n \geq 1; \\ S_i(0), & \text{if } n = 0, \end{cases} \\ &= S_i(2\lambda(n)+1) + (S_i(0) - S_i(1))U(n), \end{aligned}$$

showing that $S_i(2\lambda(n)+1)$ can be written as a linear combination of generators.

Finally, we see that $U(2n) = U(n)$, and $U(2n+1) = 0$, for all $n \geq 0$.

Now suppose that $\hat{S}$ is 2–regular. We wish to see that S is 2–regular.

Let $\hat{S}_1 = \hat{S}, \hat{S}_2, \ldots, \hat{S}_t$ be a finite set of generators for the $\mathbb{Z}$–module generated by the 2–kernel of $\hat{S}$. Then there exist integers a_{ij} and b_{ij} such that

$$\hat{S}_i(2n) = \sum_{1\leq j\leq t} a_{ij}\hat{S}_j(n),$$

and

$$\hat{S}_i(2n+1) = \sum_{1\leq j\leq t} b_{ij}\hat{S}_j(n).$$

Define

$$T_i(n) = \sum_{\substack{m\geq 1 \\ m \text{ suff } n}} \hat{S}_i(m),$$

and consider the $\mathbb{Z}$–module $\mathcal{N}$ generated by $S, T_1, T_2, \ldots, T_t$, and the constant sequence 1.

This module contains S. We must prove for each of the generators V, the sequences $\{V(2n)\}_{n\geq 0}$ and $\{V(2n+1)\}_{n\geq 0}$ are in $\mathcal{N}$.

For S, this follows from Lemma 6.4:

$$\begin{aligned} S(2n) &= S(n) + \sum_{\substack{m\geq 1 \\ m \text{ suff } n}} \hat{S}(2m) \\ &= S(n) + \sum_{\substack{m\geq 1 \\ m \text{ suff } n}} \sum_{1\leq j\leq t} a_{1j}\hat{S}(m) \\ &= S(n) + \sum_{1\leq j\leq t} a_{1j}T_j(n); \end{aligned}$$

similarly,

$$S(2n+1) = S(n) + \hat{S}(1) + \sum_{1 \le j \le t} b_{1j} T_j(n).$$

For T_i, we have:

$$\begin{aligned} T_i(2n) &= \sum_{\substack{m \ge 1 \\ m \text{ suff } 2n}} \hat{S}_i(m) \\ &= \sum_{\substack{k \ge 1 \\ k \text{ suff } n}} \hat{S}_i(2k) \\ &= \sum_{1 \le j \le t} a_{ij} T_j(n), \end{aligned}$$

and

$$\begin{aligned} T_i(2n+1) &= \sum_{\substack{m \ge 1 \\ m \text{ suff } 2n+1}} \hat{S}_i(m) \\ &= \hat{S}_i(1) + \sum_{\substack{k \ge 1 \\ k \text{ suff } n}} \hat{S}_i(2k+1) \\ &= \hat{S}_i(1) + \sum_{1 \le j \le t} b_{ij} T_j(n). \end{aligned}$$

The result for the constant sequence 1 is left to the reader! ■

Remarks.

• C. Reutenauer has pointed out the following simple proof of Theorem 6.3: Let $A = \{\epsilon\} \cup \{1, 2, \ldots, k-1\}\Sigma^*$. Then

$$S = (S, \epsilon) + \sum_{P \in E} (\hat{S}, P) \underline{A}\ \underline{P}\ \underline{\Sigma}^*,$$

where $\underline{L}$ is the characteristic series of L. We have

$$\begin{aligned} S - (S, \epsilon) &= \underline{A}\ \hat{S}\ \underline{\Sigma}^* \\ &= \underline{A}\hat{S}(\epsilon - \Sigma)^{-1}, \end{aligned}$$

and so

$$\hat{S} = \underline{A}^{-1}(S - (S, \epsilon))(\epsilon - \Sigma).$$

However, it is not immediately clear how to obtain the explicit formula in Lemma 6.5 from this observation.

• It is possible to view Theorem 6.3 as a generalization of results of Choffrut and Schützenberger [CS]. They discussed counting functions similar to our sum

$$\sum_{P \in E} \hat{S}(v(P)) e_P(n).$$

However, because they restricted their attention to finite automata with counters, they were forced to put restrictions on the set E.

• Theorem 6.3 is also a generalization of previous results of Allouche, Morton, and Shallit [AMS].

Our last result concerns the pattern transform of $\{e_P(an+b)\}_{n\geq 0}$. We prove that, in this case, the coefficients $\hat{S}(m)$ are bounded and in fact, are k–automatic.

Theorem 6.6.

Let

$$e_P(an+b) = \hat{S}(0) + \sum_{P\in E} \hat{S}(v(P))e_P(n).$$

Then $\hat{S}(m)$ is a 2–automatic sequence.

Proof.

By Corollary 6.2, we know that $S(n) = e_P(an+b)$ is 2–regular. Hence by Theorem 6.3, $\hat{S}(n)$ is 2–regular. By Theorem 2.3, it suffices to show that $\hat{S}$ takes only finitely many values. By Lemma 6.5 it suffices to prove that $S(n) - S(\lambda(n))$ takes only finitely many values.

If $n \neq 0$ and $s = |n_{(2)}|$, one has $(an+b) - (a\lambda(n)+b) = a(n-\lambda(n)) = a2^{s-1}$. Hence $an+b$ and $a\lambda(n)+b$ have the same $s-1$ final digits. Let x be fixed such that $\max(a,b) < 2^x$; then $an+b < 2^{x+s} + 2^x < 2^{x+s+1}$; hence $an+b$ has at most $x+s+1$ digits.

Finally, the numbers $an+b$ and $a\lambda(n)+b$ differ in at most $(x+s+1)-(s-1) = x+2$ digits. Hence, for every P, $|e_P(an+b) - e_P(a\lambda(n)+b)|$ is bounded by $x+2$, and the result follows. ■

VII. Some examples.

Unless otherwise indicated, we assume $k = 2$ in the examples that follow. Sequence numbers refer to Sloane's book [Sl].

Example 1.

By Theorem 6.1, we know the sequence $\{e_1(n)\}_{n\geq 0}$ is 2–regular. In fact, it satisfies the relations $e_1(2n) = e_1(n)$; $e_1(2n+1) = e_1(n)+1$. Hence its 2–kernel is generated by $e_1(n)$ and the constant sequence 1. (This is Sloane's sequence #41.)

Example 2.

Define $A(n) = \sum_{1\leq j\leq n} e_1(j)$, the total number of 1's in the base–2 expansion of the first n integers. Then $A(n)$ is 2–regular by the remark after Theorem 3.1. $A(n)$ has been extensively studied in the literature ([BS], [CL], [CY]). It is Sloane's sequence #360.

Example 3.

Consider the sequence

$$\{c(n)\}_{n\geq 0} = 0, 2, 6, 8, 18, 20, 24, 26, 54, 56, \ldots,$$

which lists the numerators of the left endpoints of the Cantor set. (Alternatively, these are the integers whose base–3 representations contain no 1's; see [MFP].) Then it is easy to see that $c(2n) = 3c(n)$ and $c(2n+1) = 3c(n)+2$. Hence it is 2–regular. (Note, however, that its characteristic sequence $101000101\cdots$ is actually 3–automatic.)

For a more general perspective on such sequences, see [Mah2].

Example 4.

The sequence $e_1(3n)$ has been studied extensively by Newman, Slater, and Coquet ([N], [NS], [Coq]). By Corollary 5.2 it is 2–regular. By Theorem 6.6 it has a 2–automatic pattern transform. In fact, we find

$$\begin{aligned} e_1(3n) &= 2e_1(n) - 2e_{11}(n) + e_{111}(n) - 2e_{1011}(n) + e_{11011}(n) - 2e_{101011}(n) + e_{1101011}(n) - \cdots \\ &= 2e_1(n) - 2\sum_{i\geq 0} e_{(10)^i 11}(n) + \sum_{i\geq 0} e_{11(01)^i 1}(n). \end{aligned}$$

This expansion gives an alternative explanation to the observation [N] that the first few values of $e_1(3n)$ are almost all even. See [AMS].

Example 5.

Let j be an integer ≥ 0. The sequence $\{n^j\}_{n\geq 0}$ is 2–regular, as the module generated by its 2–kernel is generated by the constant sequence 1 and the sequences $\{n\}_{n\geq 0}$, $\{n^2\}_{n\geq 0}$, $\ldots$, $\{n^j\}_{n\geq 0}$.

From Theorem 6.3, we know the corresponding pattern transforms are 2–regular. Using Lemma 6.5, we find:

$$\begin{aligned} n = {} & e_1(n) + e_{10}(n) + e_{11}(n) + 2(e_{100}(n) + \cdots + e_{111}(n)) \\ & + 4(e_{1000} + \cdots + e_{1111}(n)) + 8(e_{10000}(n) + \cdots + e_{11111}(n)) + \cdots \end{aligned}$$

Example 6.

Let w^R denote the reverse of the string w. Consider the map which takes every integer to the integer represented by the reverse of its base–2 representation, i. e. $r(n) = v(n_{(2)}^R)$. Then it is not difficult to show that [IMO] $r(2n) = r(n)$, $r(4n+3) = 3r(2n+1) - 2r(n)$, $r(8n+1) = 3r(4n+1) - 2r(2n+1)$, and $r(8n+5) = 5r(2n+1) - 4r(n)$. Hence it follows that the module generated by the 2–kernel of $\{r(n)\}_{n\geq 0}$ is generated by its subsequences $\{r(n)\}_{n\geq 0}$, $\{r(2n+1)\}_{n\geq 0}$, and $\{r(4n+1)\}_{n\geq 0}$.

Example 7.

Let $d(0) = 0$, $d(1) = 1$, $d(2n) = d(n)$, and $d(2n+1) = d(n) + d(n+1)$. This sequence forms the numerators of the entries in the *Stern-Brocot tree* (see [St], [GKP]). It was also studied by de Rham [R] and is Sloane's sequence #56. The first few terms are

$$0, 1, 1, 2, 1, 3, 2, 3, 1, 4, 3, 5, 2, 5, 3, 4, \ldots.$$

It is easy to see that $d(4n+1) = d(n) + d(2n+1)$, and $d(4n+3) = 2d(2n+1) - d(n)$, and it follows that d is 2–regular. Also see [Dij, pp. 215–216, 230–232].

A similar sequence is given by $a(0) = 0$, $a(1) = 1$, $a(2n) = a(n)$, and $a(2n+1) = a(n+1) - a(n)$. It satisfies $a(4n+1) = a(2n+1) - a(n)$ and $a(4n+3) = a(n)$ and hence is 2–regular. See [Rez1], [Rez2].

Example 8.

Define $\nu_2(n)$ to be the exponent of the highest power of 2 that divides n. (This is essentially Sloane's sequence #51.) Then if $h(n) = \nu_2(n+1)$, we see that $h(2n) = 0$ and $h(2n+1) = h(n) + 1$. Thus $\{h(n)\}_{n\geq 0}$ is 2–regular.

Using Lemma 6.4, we find

$$\nu_2(n+1) = e_1(n) - (e_{10}(n) + e_{110}(n) + e_{1110}(n) + \cdots).$$

Example 9.

Using the remark after Theorem 3.1, we see that $\nu_2(n!) = \sum_{1\leq j\leq n} \nu_2(j)$ is 2–regular.

Example 10.

Let the binary expansion of an integer n be written as

$$\sum_{i\geq 0} b_i(n) 2^i,$$

where $b_i \in \{0, 1\}$. Define $g(n) = \sum_{i\geq 0}(i+1)b_i(n)$. Then it is easy to see that $g(2n) = g(n) + e_1(n)$ and $g(2n+1) = g(n) + e_1(n) + 1$. Hence $\{g(n)\}_{n\geq 0}$ is 2–regular.

Using Lemma 6.4, we can compute the pattern transform of $g(n)$. We find

$$g(n) = \sum_{P\in 1(0+1)^*} e_P(n).$$

Example 11.

Let $f(n) = |n_{(2)}|$, i. e.

$$f(n) = \begin{cases} 0, & \text{if } n = 0; \\ 1 + \lfloor \log_2 n \rfloor, & \text{if } n \geq 1. \end{cases}$$

Then we easily see that $f(2n+1) = f(n)+1$, $f(4n) = 2f(2n)-f(n)$, and $f(4n+2) = f(n)+2$. Hence the module generated by its 2-kernel is generated by $\{f(n)\}_{n\geq 0}$, $\{f(2n)\}_{n\geq 0}$, and the constant sequence 1. Using Lemma 6.4, we find

$$f(n) = e_1(n) + e_{10}(n) + e_{100}(n) + e_{1000}(n) + \cdots.$$

Example 12.

Let $\{B(n)\}_{n\geq 0}$ be the sequence $0, 3, 5, 6, 9, 10, 12, 15, \ldots$, the integers whose base–2 representation contains an even number of 1's. (This is Sloane's sequence #952). Let $u(n) = e_1(n) \bmod 2$; then $u(n)$ is the classical Thue-Morse sequence and $1 - u(n)$ is the characteristic sequence of $B(n)$. We easily prove that $B(2n) = 2B(n) - u(n)$ and $B(2n+1) = 2B(n) + 3(1 - u(n))$; hence B is 2–regular.

Example 13.

Let $\{C(n)\}_{n\geq 0}$ be the sequence of *Moser-de Bruijn* ([Mos], [B2]): $0, 1, 4, 5, 16, 17, 20, 21, \ldots$ It consists of integers that can be written as the sum of distinct powers of 4. This is Sloane's sequence #1315. Note that $C(2n) = 4C(n)$ and $C(2n+1) = 4C(n) + 1$; hence C is 2–regular. See [LMP]. In [BM] it is shown that the characteristic sequence of $C(n)$ gives a binary number such that its binary expansion and the binary expansion of its reciprocal are explicitly known. Its continued fraction is also explicitly known.

Similarly, the sequence of *Loxton-van der Poorten* [LP1]

$$0, 1, 3, 4, 5, 11, 12, 13, 15, 16, 17, 19, 20, 21, 43, 44, \ldots$$

of positive integers that can be represented in base 4 using only the digits $-1, 0, 1$ is 3–regular.

Example 14.

Let $G(n) = 2^{e_1(n)}$. This is *Gould's sequence* [G], and Sloane's sequence #109. It satisfies $G(2n) = G(n)$; $G(2n+1) = 2G(n)$ and hence is 2–regular.

Glaisher [Gl] showed that $G(n)$ counts the number of odd binomial coefficients in row n of Pascal's triangle.

More generally, let p be a prime and let $G_p(n)$ be the number of binomial coefficients in row n of Pascal's triangle which are not divisible by p. Then Fine [Fi] showed that

$$G_p(n) = \prod_{0\leq i\leq e} (a_i + 1)$$

where the base–p expansion of n is $a_e a_{e-1} \cdots a_1 a_0$. Of course, $G_p(n)$ is p–regular.

Now put $H_p(n) = \sum_{0\leq k\leq n} G_p(k)$. Then $H_p(n)$ is also p–regular. The sequences $H_2(n)$,

$$1, 3, 5, 9, 11, 15, 19, 27, 29, 33, 37, 45, 49, 57, \ldots$$

and $H_3(n)$,

$$1, 3, 6, 8, 12, 18, 21, 27, 36, 38, 42, 48, 52, 60, 72, \ldots$$

appear in [LM]. Also see [HLVVM], [LMVV].

Example 15.

Let $\{b(n)\}_{n\geq 0}$ be the sequence of numbers represented by binary Gray code [Gr], [Gi]:

$$0, 1, 3, 2, 6, 7, 5, 4, 12, 13, 15, 14, 10, 11, 9, 8, \cdots$$

Then it is easy to see that $b(4n) = 2b(2n)$, $b(4n+1) = 2b(2n)+1$, $b(4n+2) = 2b(2n+1)+1$, and $b(4n+3) = 2b(2n+1)$. Hence $\{b(n)\}_{n\geq 0}$ is 2–regular.

Similarly, if $\gamma(n)$ denotes the sum of the bits in the Gray code representation of n, then we find $\gamma(2n+1) = 2\gamma(n) - \gamma(2n) + 1$; $\gamma(4n) = \gamma(2n)$; and $\gamma(4n+2) = \gamma(2n+1)+1$. Hence $\{\gamma(n)\}_{n\geq 0}$ is 2–regular. See [FR].

Example 16.

Consider the sequence of lattice points $(x(n), y(n))$ traced out by paperfolding curves with an ultimately periodic sequence of unfolding instructions [DMFP, MFS]. Then $\{x(n)\}_{n\geq 0}$ and $\{y(n)\}_{n\geq 0}$ are 2–regular.

For example, consider the sequence of lattice points $(x(n), y(n))$ traced out by the space-filling curve with unfolding instructions RLRLRL$\cdots$.

$$\begin{array}{rl} n & = 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15\ \cdots \\ x(n) & = 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 2\ 2\ 3\ 3\ 2\ 2\ 1\ 1\ \cdots \\ y(n) & = 0\ 1\ 1\ 2\ 2\ 3\ 3\ 2\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 4\ \cdots \end{array}$$

Then the sequences satisfy the identities $x(0) = 0$, $x(2) = 1$, $x(2n+1) = x(2n)$, $x(4n) = 2x(n)$, $x(8n+2) = -2x(n) + 2x(2n) + x(4n+2)$, $x(16n+6) = 2x(n) + x(4n+2)$, $x(16n+14) = 2x(2n) + 2x(4n+2) - x(8n+6)$, and $y(0) = 0$, $y(1) = 1$, $y(4n) = 2y(n)$, $y(4n+1) = y(4n+2) = 2y(n) - y(2n) + y(2n+1)$, $y(8n+3) = y(8n+7) = 2y(2n+1)$.

Example 17.

Van der Corput's sequence $\varphi_2(n)$ is defined as follows [Cor]: if

$$n = \sum_{i\geq 0} b_i(n) 2^i,$$

where $b_i \in \{0, 1\}$, then

$$\varphi_2(n) = \sum_{i\geq 0} b_i(n) 2^{-i-1}.$$

We see that $\varphi_2(0) = 0$, $\varphi_2(2n) = \frac{1}{2}\varphi_2(n)$, and $\varphi_2(2n+1) = \frac{1}{2} + \frac{1}{2}\varphi_2(n)$. Hence the sequence of rational numbers $\varphi_2(n)$ is $(\mathbb{Q}, 2)$–regular.

Also note that $\varphi_2(n) = r(n)/2^{f(n)}$, where $r(n)$ is the sequence of Example 6 and $f(n)$ is the sequence of Example 11.

Halton [Hal] generalized van der Corput's sequence to bases $b \geq 2$.

Example 18.

Let

$$\frac{1}{(1-X)(1-X^2)\cdots(1-X^j)} = \sum_{n\geq 0} P_j(n) X^n.$$

Then $P_j(n)$ enumerates the number of partitions of n into j or fewer parts. The sequence $P_3(n)$ is Sloane's sequence #186; $P_4(n)$ is sequence #229; $P_5(n)$ is sequence #237, and $P_6(n)$ is sequence #243.

By Theorem 3.3, $P_j(n)$ is k–regular for all $j \geq 1$ and all $k \geq 2$. Note, however, that the function $P_\infty(n) = \lim_{j\to\infty} P_j(n)$, which counts the number of unrestricted partitions, is not k–regular, as it grows too quickly. Hence k–regular sequences are not closed under taking simple limits.

Example 19.

Let $w = w_0 w_1 w_2 \cdots$ be an infinite word over a finite alphabet, and define $s_w(n)$ to be the number of distinct subwords (European terminology: factors) of length n in w. Then $s_w(n) - s_w(n-1)$ is frequently k–automatic, and hence in these cases, $s_w(n)$ is k–regular. For example, this is true when w is the fixed point of the Toeplitz substitution given by $0 \to 0010$ and $1 \to 1010$ [Rau]; when w is the infinite word of Thue-Morse, the fixed point of the substitution given by $0 \to 01$ and $1 \to 10$ [Brl] [LV]; and in a more general class of infinite words given by iterated homomorphisms discussed by Tapsoba [Tap].

Example 20.

It is well-known that n is a sum of three squares if and only if n is not of the form $4^a(8k+7)$. It is easily seen that the sequence

$$t(n) = \begin{cases} 0, & \text{if } n = 4^a(8k+7); \\ 1, & \text{otherwise,} \end{cases}$$

is 2–automatic. Hence the sequence

$$Q(n) = \sum_{1 \leq k \leq n} t(k),$$

which counts the number of positive integers $\leq n$ that are the sum of three squares, is 2–regular. See [Sh], [OS], [W].

Example 21.

An *addition chain to* n is a sequence of pairs of positive integers

$$(a_1, b_1), (a_2, b_2), \ldots, (a_r, b_r)$$

where (i) $a_r + b_r = n$ and (ii) for all s, either $a_s = 1$, or $a_s = a_i + b_i$ for some $i < s$, and the same holds for b_s. The cost of the addition chain is $\sum_{1 \leq i \leq r} a_i b_i$. Denote the cost of the minimum cost addition chain to n as $c(n)$. Then it can be shown [GYY] that $c(1) = 0$, and $c(2n) = c(n) + n^2$, $c(2n+1) = c(n) + n^2 + 2n$ for $n \geq 1$. Hence $c(n)$ is 2–regular.

Example 22.

Define $b(d; n)$ as the number of representations

$$n = \sum_{i \geq 0} \epsilon_i 2^i,$$

where $0 \le \epsilon_i < d$. Then $b(2;n) = 1$, $b(3;n) = d(n+1)$, where $d(n)$ is the sequence of Example 7, and $b(4;n) = 1 + \lfloor n/2 \rfloor$. (See [Rez3]). It is possible to show that $b(d;n)$ is 2–regular for all $d \ge 1$. However, $b(\infty;n) = \lim_{d\to\infty} b(d;n)$ is not k–regular, as it is known that

$$\log b(\infty;n) \sim \frac{1}{\log 4}(\log n)^2.$$

(See [Mah], [B1], [Kn1]). Note that if $f(X) = \sum_{n\ge 0}(-1)^{e_1(n)}X^n$, and $g(X) = \sum_{n\ge 0} b(\infty;n)X^n$, then $1/f(X) = g(X^2)(1+X)$, which shows that $f(X)$ is not invertible in the ring of 2–regular power series. P. Dumas has pointed out [Dum] that the sequence $\{b(\infty;n) \bmod 2^M\}_{n\ge 0}$ is 2–automatic for all $M \ge 0$.

Example 23.

Let $\nu_3(n)$ denote the exponent of the highest power of 3 that divides n, and $s_3(n)$ denote the sum of the digits of n when expressed in base 3.

Define $r(n) = \sum_{0\le i<n} \binom{2i}{i}$. Then $\nu_3(r(n))$ is 3–regular. This follows from the (not-so-trivial) fact that

$$\nu_3(r(n)) = \nu_3(\binom{2n}{n}) + 2\nu_3(n)$$

and the (trivial) fact that

$$\nu_3(\binom{2n}{n}) = s_3(n) - \frac{1}{2}s_3(2n).$$

See [SS2].

Example 24.

As in Section VI, let

$$\lambda(n) = \begin{cases} 0, & \text{if } n = 0; \\ n - 2^{\lfloor \log_2 n \rfloor}, & \text{if } n > 0. \end{cases}$$

Then A. Liao (personal communication) asked for the solution $T(n)$ to the recurrence

$$T(n) = \lambda(n) + T(\lambda(n)),$$

where $f(0) = 0$. We see that $T(n)$ is 2–regular, as the identities $T(2n) = 2T(n)$, $T(4n+1) = 2T(n) + T(2n+1)$, and $T(4n+3) = -2T(n) + 3T(2n+1) + 1$ can easily be verified by induction.

$T(n)$ also has the following pleasant expansion as a sum of pattern sequences:

$$T(n) = \sum_{v(P)\ge 3} \left\lceil \frac{\lambda(P)}{2} \right\rceil e_P(n).$$

Example 25.

The earliest reference to a non-trivial class of k–regular sequences we have found is from an 1822 paper of Charles Babbage [Bab], in which he discusses sequences such as

$$\Delta^2 u_n = u_{n+1} \bmod 10,$$

"which is one of a class of equations never hitherto integrated." By considering both sides (mod 10), we see that the sequence $\{u_n \bmod 10\}_{n\geq 0}$ is ultimately periodic and therefore u_n is k–regular for all $k \geq 2$. This sequence was deemed to have "no intrinsic mathematical significance" by Dubbey [Dub, pp. 182].

Example 26.

Let $e(0) = 0$, $e(1) = 1$, and define $e(n)$ to be the least integer greater than $e(n-1)$ such that the sequence $e(0), \ldots, e(n)$ contains no three terms in arithmetic progression. The first few terms of this sequence are

$$0, 1, 3, 4, 9, 10, 12, 13, 27, 28, 30, 31, 36, 37, 39, 40, 81, \ldots$$

and in general the sequence consists of numbers that can be written as distinct powers of 3. (Compare Example 13.) We have $e(2n) = 3e(n)$ and $e(2n+1) = 3e(n)+1$, and so $e(n)$ is 2–regular. See [ET] and [Guy, p. 114].

Example 27.

Let k be an integer ≥ 2, and put

$$f_k(n) = \sum_{1\leq i\leq n} \lfloor \log_k i \rfloor.$$

Then $f_k(n)$ is k–regular. In fact, we have

$$f_k(n) = (n+1)\lfloor \log_k n \rfloor - \frac{k^{\lfloor \log_k n \rfloor + 1} - k}{k-1}.$$

See [Kn3, Section 1.2.4, Exercise 42 (b)].

The number of comparisons required to sort n items in many sorting algorithms forms a 2–regular sequence. The following examples illustrate this:

Example 28.

Merge sort, given a list of n integers, proceeds as follows: first the left half of the list is sorted (recursively), then the right half is sorted, and finally the two halves are merged together. The number of comparisons needed to merge sort n items is given by $T(1) = 0$, and

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n - 1,$$

for $n \geq 2$, and it is not difficult to see that $T(n)$ is a 2–regular sequence.

The resulting sequence

$$0, 1, 3, 5, 8, 11, 14, 17, 21, \ldots$$

is Sloane's sequence #963. It was discussed by Levitt, Green, and Goldberg [LGG], who gave the following closed form:

$$T(n) = n\lceil \log_2 n \rceil - 2^{\lceil \log_2 n \rceil} + 1.$$

Also see [Kn2, Section 5.3.1, Equation (3)].

Example 29.

Let $c(n)$ denote the number of key comparisons used to sort n elements by Batcher's method (see, for example, [Knu2, Section 5.2.2]). The first few terms of this sequence are

$$0, 1, 3, 5, 9, 12, 16, 19, 26, 31, 37, 41, 48, 53, 59, 63, \ldots$$

Define $a(n) = c(n+1) - c(n)$. Then it is shown in [Knu2, Section 5.2.2, Exercises 14, 15] that $a(2n) = a(n) + \lfloor \log_2(2n) \rfloor$; $a(2n+1) = a(n) + 1$ and hence $a(n)$ is 2–regular. Hence $c(n)$ is 2–regular.

Example 30.

Let $F(n)$ denote the number of key comparisons in Ford-Johnson sorting. Here are the first few values of this sequence:

$$0, 1, 3, 5, 7, 10, 13, 16, 19, 22, 26, 30, 34, 38, 42, 46, 50, \ldots$$

It is Sloane's sequence #954. A. Hadian showed that

$$F(n) = \sum_{1 \le k \le n} \lceil \log_2 \frac{3n}{4} \rceil;$$

see [Kn2, Section 5.3.1]. It is easy to show that $\lceil \log_2 n \rceil$ is a 2–regular sequence. Then by Theorem 2.6 $\lceil \log_2 3n \rceil - 2 = \lceil \log_2 3n/4 \rceil$ is 2–regular. Finally, by Theorem 3.1, $F(n)$ must be 2–regular. Knuth [Kn2, Section 5.3.1, Exercise 14] gives the following "closed form" for $F(n)$:

$$F(n) = n\lceil \log_2 \frac{3n}{4} \rceil - \lfloor 2^{\lfloor \log_2 6n \rfloor}/3 \rfloor + \lfloor \frac{1}{2} \log_2 6n \rfloor.$$

Example 31.

Let $k(n)$ denote the maximum number of key comparisons used by list-merge sorting; see [Kn2, Section 5.2.4]. Here are the first few terms of this sequence

$$0, 1, 3, 5, 9, 11, 14, 17, 25, 27, 30, 33, 38, 41, 45, 49, \ldots$$

Then it is known that if the binary representation of n is

$$2^{e_1} + 2^{e_2} + \cdots + 2^{e_t},$$

then

$$k(n) = 1 - 2^{e_t} + \sum_{1 \le k \le t} (e_k + k - 1)2^{e_k}.$$

(See [Kn2, Section 5.2.4, Exercises 14]).

From this it is easy to see that $k(2n) = 2k(n) + 2n - 1$ for $n \ge 1$ and $k(2n+1) = k(2n) + e_1(n) + 2^{\nu_2(n)+1} - 1$ for $n \ge 1$. Hence $k(n)$ is a 2–regular sequence.

Example 32.

Similarly, the number of comparisons needed in many merging algorithms forms a 2–regular sequence. For example, let $M(m,n)$ denote the minimum number of comparisons to merge m things with n. Then

$$M(1,n) = \lceil \log_2(n+1) \rceil$$

and

$$M(2,n) = \lceil \log_2 \frac{7}{12}(n+1) \rceil + \lceil \log_2 \frac{14}{17}(n+1) \rceil.$$

(See [Kn2, Section 5.3.2].) While $M(1,n)$ is easily seen to be 2–regular, we can prove that $M(2,n)$ is 2–regular using Theorems 2.6 and 2.7.

Example 33.

In analysis of a greedy heuristic for a matching problem, Reingold and Tarjan [RT] define a function $f(n)$ for positive even arguments, and write

$$f(n) = \min_{\substack{2 \le t \le n-2 \\ t \text{ even} \\ \alpha \ge 1-\alpha-\beta > 0 \\ \beta \ge 1-\alpha-\beta > 0}} \{\alpha f(t) + \beta f(n-t)\}.$$

Later they show that

$$f(2n) = \begin{cases} \frac{2}{3} f(n), & \text{if } n \text{ is even;} \\ \frac{1}{3} f(n+1) + \frac{1}{3} f(n-1), & \text{if } n \text{ is odd.} \end{cases}$$

They also give the following explicit form for $f(2n)$:

$$f(2n) = 1 - \sum_{2 \le i \le n} 3^{-\lceil \log_2 i \rceil}.$$

It follows from this that $f(2n)$ is a $(\mathbb{Q}, 2)$–regular sequence. The first few values of this sequence are:

$$1, 2/3, 5/9, 4/9, 11/27, 10/27, 1/3, 8/27, \cdots$$

Example 34.

The *Josephus problem* is as follows: the numbers from 1 to n are written in a circle. Starting with the number 1, every $2nd$ number that remains is crossed off until only one is left. The "survivor" is denoted $J(n)$. The first few values of $J(n)$ are as follows:

$$1, 1, 3, 1, 3, 5, 7, 1, 3, 5, 7, 9, 11, 13, 15, \cdots.$$

This problem was discussed by Graham, Knuth, and Patashnik [GKP, pp. 8–16], who observed that $J(2n) = 2J(n) - 1$ and $J(2n+1) = 2J(n) + 1$ for $n \geq 1$. It follows that $J(n)$ is 2–regular.

The same problem, where 2 is replaced by k and the result is the first uncrossed-off number encountered when there are only $k-1$ numbers left, does not appear to be k-regular in general. See [GKP, pp. 79–81].

We are grateful to P. Dumas for pointing out this example.

Example 35.

We show that the sequence of primes $\{p(n)\}_{n\geq 0}$

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, \ldots$$

is not k–regular. Suppose it were. Then using Lemma 4.1, we see that $\{p(k^n)\}_{n\geq 0}$ must satisfy a linear recurrence. Then if

$$\lim_{n\to\infty} \frac{p(k^n)}{nk^n}$$

exists, it must be an algebraic number. But from the prime number theorem,

$$\lim_{n\to\infty} \frac{p(k^n)}{nk^n} = \log k,$$

which is transcendental, a contradiction.

VIII. Some Open Problems.

1. Let $R' = \mathbb{Q}$. Prove that if $\{S(n)\}_{n\geq 0}$ is k_1-regular and k_2-regular, and k_1 and k_2 are multiplicatively independent, then the associated power series $\sum_{n\geq 0} S(n)X^n \in \mathbb{Q}[[X]]$ is a rational function. In the case where R' is finite, this is a result of Cobham [Cob2].

2. Determine all the units of the ring of k-regular power series.

3. Obtain transcendence results for the real numbers $\sum_{n\geq 0} S(n)p^{-n}$, where $S(n)$ is p-regular and $\sum_{n\geq 0} S(n)X^n$ is not a rational function. See [LP2].

IX. Acknowledgments.

We are grateful to O. Salon for suggesting the term k–kernel [Sa].

Part of this work was done while the first author was visiting Dartmouth College.

Part of this work was done while the second author was a visiting scientist at the University of Waterloo and a visiting professor at the University of Wisconsin, Madison.

A preliminary version of this paper was presented at the Symposium on Theoretical Aspects of Computer Science (STACS) in Rouen, France, on February 24, 1990.

The second author acknowledges with thanks conversations with E. Bach, C. Choffrut, L. Dickey, J. Driscoll, G. Frandsen, D. Joseph, S. Kurtz, R. Lipton, A. Lubiw, D. Passman, E. Reingold, N. J. A. Sloane, and C. Reutenauer.

References

[A1] J.-P. Allouche, Automates finis en théorie des nombres, *Expo. Math.* **5** (1987), 239-266.

[A2] J.-P. Allouche, Somme des chiffres et transcendance, *Bull. Soc. Math. France* **110** (1982), 279-285.

[A3] J.-P. Allouche, Suites infinies à répétitions bornées, *Séminaire de Théorie des Nombres de Bordeaux*, 1983-1984, Exposé no. 20.

[A4] J.-P. Allouche, Note sur un article de Sharif et Woodcock, *Sém. de Théorie des Nombres de Bordeaux*, 2^e série, **1** (1989), 163-187.

[AMS] J.-P. Allouche, P. Morton, and J. Shallit, Pattern spectra, substring enumeration, and automatic sequences, preprint.

[B1] N. G. de Bruijn, On Mahler's partition problem, *Indag. Math.* **10** (1948), 210-220

[B2] N. G. de Bruijn, Some direct decompositions of the set of integers, *Math. Tables Aids Comput.* **18** (1964), 537-546.

[Bab] C. Babbage, Observations on the application of machinery to the computation of mathematical tables, *Memoirs of the Astronomical Society* **1** (1822), 311-314; reprinted in *Babbage's Calculating Engines*, Tomash Publishers, Los Angeles, 1982.

[BM] A. Blanchard and M. Mendès France, Symétrie et transcendance, *Bull. Sci. Math.* **106** (1982), 325-335.

[BR] J. Berstel and C. Reutenauer, *Rational Series and Their Languages*, Springer-Verlag, 1988.

[Brl] S. Brlek, Enumeration of factors in the Thue-Morse word, *Disc. Appl. Math.* **24** (1989), 83-96.

[BS] R. Bellman and H. N. Shapiro, On a problem in additive number theory, *Ann. Math.* **49** (1948), 333-340.

[CKMR] G. Christol, T. Kamae, M. Mendès France and G. Rauzy, Suites algébriques, automates et substitutions, *Bull. Soc. Math. France* **108** (1980), 401-419.

[CL] G. Clements and B. Lindström, A sequence of (± 1)-determinants with large values, *Proc. Amer. Math. Soc.* **16** (1965), 548-550.

[Cob] A. Cobham, Uniform tag sequences, *Math. Systems Theory* **6** (1972), 164-192.

[Cob2] A. Cobham, On the base-dependence of sets of numbers recognizable by finite automata, *Math. Systems Theory* **3** (1969), 186-192.

[Coq] J. Coquet, A summation formula related to the binary digits, *Invent. Math.* **73** (1983), 107-115.

[Cor] J. C. van der Corput, Verteilungsfunktionen, *Proc. Ned. Akad. v. Wet.* **38** (1935), 813-821.

[CS] C. Choffrut and M. P. Schützenberger, Counting with rational functions, *Theor. Comput. Sci.* **58** (1988), 81-101.

[CY] P. Cheo and S. Yien, A problem on the k-adic representations of positive integers, *Acta. Math. Sinica* **5** (1955), 433-438.

[Dav] M. Davis, Hilbert's tenth problem is unsolvable, *Amer. Math. Monthly* **80** (1973), 233-269.

[Dij] E. W. Dijkstra, *Selected Writings on Computing: a Personal Perspective*, *Springer-Verlag*, New York, 1982.

[DMFP] M. Dekking, M. Mendès France, and A. van der Poorten, FOLDS!, *Math. Intell.* **4** (1982), 130-138; 173-181; 190-195. (Errata in *Math. Intell.* **5** (1983), 5.)

[Dub] J. M. Dubbey, *The Mathematical Work of Charles Babbage*, Cambridge University Press, 1978.

[Dum] P. Dumas, Suite automatiques à valeurs dans un anneau commutatif, preprint.

[E] S. Eilenberg, *Automata, Languages, and Machines*, Volume A, Academic Press, 1974.

[ET] P. Erdös and P. Turán, On some sequences of integers, *J. Lond. Math. Soc.* **11** (1936), 261-264.

[Fi] N. J. Fine, Binomial coefficients modulo a prime, *Amer. Math. Monthly* **54** (1947), 589-592.

[FR] P. Flajolet and L. Ramshaw, A note on Gray code and odd-even merge, *SIAM J. Comput.* **9** (1980), 142-158.

[G] H. W. Gould, Exponential binomial coefficient series, Technical Report 4, Department of Mathematics, W. Virginia Univ., September 1961.

[Gi] E. Gilbert, Gray codes and paths on the n-cube, *Bell Sys. Tech. J.* **37** (1958), 815-826.

[GKP] R. Graham, D. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison-Wesley, 1989.

[Gl] J. W. L. Glaisher, On the residue of a binomial-theorem coefficient with respect to a prime modulus, *Quart. J. Pure Appl. Math.* **30** (1899), 150-156.

[Gr] F. Gray, U. S. patent 2,632,058, March 17, 1953 (Filed November 13, 1947).

[GS] S. Ginsburg and E. H. Spanier, Bounded ALGOL-like languages, *Trans. Amer. Math. Soc.* **113** (1964), 333-368.

[Guy] R. K. Guy, *Unsolved Problems in Number Theory*, Springer-Verlag, 1981.

[GYY] R. Graham, A. Yao, and F. Yao, Addition chains with multiplicative cost, *Disc. Math.* **23** (1978), 115-119.

[Hal] J. H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numer. Math.* **2** (1960), 84-90.

[Har1] M. Harrison, *Lectures on Linear Sequential Machines*, Academic Press, 1969.

[HLVVM] N. S. Holter, A. Lakhtakia, V. K. Varadan, V. V. Varadan, and R. Messier, On a new class of planar fractals: the Pascal-Sierpinski gaskets, *J. Phys. A: Math. Gen.* **19** (1986), 1753-1759.

[IMO] 29th International Math Olympiad–Solutions, *Math. Mag.* **62** (1989), 212.

[JK] K. Jacobs and M. Keane, 0-1-Sequences of Toeplitz type, *Z. Wahrscheinlichkeitstheorie verw. Geb.* **13** (1969), 123-131.

[Kn1] D. E. Knuth, An almost linear recurrence, *Fib. Quart.* **4** (1966), 117-128.

[Kn2] D. E. Knuth, *Sorting and Searching*, The Art of Computer Programming, V. 3, Addison-Wesley, 1973.

[Kn3] D. E. Knuth, *Fundamental Algorithms*, The Art of Computer Programming, V. 1, Addison-Wesley, 1973.

[Lan] S. Lang, *Algebra*, Addison-Wesley, 1971.

[LGG] K. N. Levitt, M. W. Green, and J. Goldberg, A study of the data commutation problems in a self-repairable multiprocessor, *Proc. AFIPS Conf.* **32** (1968), 515-527.

[Li] P. Liardet, Automata and generalized Rudin-Shapiro sequences, *Sem. Salzburg Universität*, 1986; Publication 23, U. R. A. #225, C. N. R. S., Marseille, 1989.

[LM] A. Lakhtakia and R. Messier, Self-similar sequences and chaos from Gauss sums, *Comput. & Graphics* **13** (1989), 59-62.

[LMP] D. H. Lehmer, K. Mahler, and A. J. van der Poorten, Integers with digits 0 or 1, *Math. Comp.* **46** (1986) 683-689.

[LMVV] A. Lakhtakia, R. Messier, V. K. Varadan, and V. V. Varadan, Fractal sequences derived from the self-similar extensions of the Sierpinski gasket, *J. Phys. A: Math. Gen.* **21** (1988), 1925-1928.

[LP1] J. H. Loxton and A. J. van der Poorten, An awful problem about integers in base four, *Acta Arithmetica* **49** (1987), 193-203.

[LP2] J. H. Loxton and A. J. van der Poorten, Arithmetic properties of automata: regular sequences, *J. reine angew. Math.* **392** (1988), 57-69.

[LV] A. de Luca and S. Varricchio, Some combinatorial properties of the Thue-Morse sequence and a problem in semigroups, *Theor. Comput. Sci.* **63** (1989), 333-348.

[LZ] R. Lipton and Y. Zalcstein, Word problems solvable in logspace, *J. ACM* **24** (1977), 522-526.

[Mah] K. Mahler, On a special functional equation, *J. Lond. Math. Soc.* **15** (1940), 115-123.

[Mah2] K. Mahler, On the generating function of the integers with a missing digit, *J. Indian Math. Soc.* **15** (1951), 33-40.

[Mau] C. Mauduit, Substitutions et ensembles normaux, Habilitation, Marseille, 1989.

[MFP] M. Mendès France and A. J. van der Poorten, From geometry to Euler identities, *Theor. Comput. Sci.* **65** (1989), 213-220.

[MFS] M. Mendès France and J. Shallit, Wire bending, *J. Combinatorial Theory, A* **50** (1989) 1-23.

[MM] P. Morton and W. Mourant, Paper folding, digit patterns, and groups of arithmetic fractals, *Proc. Lond. Math. Soc.* **59** (1989), 253-293.

[Mos] L. Moser, An application of generating series, *Math. Mag.* **35** (1962), 37-38.

[N] D. J. Newman, On the number of binary digits in a multiple of three, *Proc. Amer. Math. Soc.* **21** (1969), 719-721.

[NS] D. J. Newman and M. Slater, Binary digit distribution over naturally defined sequences, *Trans. Amer. Math. Soc.* **213** (1975), 71-78.

[OS] A. H. Osbaldestin and P. Shiu, A correlated digital sum problem associated with sums of three squares, *Bull. Lond. Math. Soc.* **21** (1989), 369-374.

[Pro] H. Prodinger, Non-repetitive sequences and Gray code, *Disc. Math.* **43** (1983), 113-116.

[R] G. de Rham, Un peu de mathématiques à propos d'une courbe plane, *Elem. Math.* **2** (1947), 73-77; 89-97.

[Rau] G. Rauzy, Suites à termes dans un alphabet fini, *Sém. Théorie des Nombres de Bordeaux*, 2^e série, 1982-3, 25.01–25.16.

[Rez1] B. Reznick, A new sequence with many properties, *Abs. Amer. Math. Soc.* **5** (1984), 16.

[Rez2] B. Reznick, Some extremal problems for continued fractions, *Ill. J. Math.* **29** (1985), 261-279.

[Rez3] B. Reznick, Some binary partition functions, in B. C. Berndt, H. G. Diamond, H. Halberstam, and A. Hildebrand, eds., *Analytic Number Theory (Proceedings of a Conference in Honor of Paul T. Bateman)*, Birkhäuser, Boston, 1990, pp. 451-477.

[RT] E. M. Reingold and R. E. Tarjan, On a greedy heuristic for complete matching, *SIAM J. Comput.* **10** (1981), 676-681.

[Sa] O. Salon, Quelles tuiles! (Pavages apériodiques du plan et automates bidimensionnels), *Sém. de Théorie des Nombres de Bordeaux*, (2) **1** (1989), 1-25.

[Sch] M. P. Schützenberger, On the definition of a family of automata, *Information and Control* **4** (1961), 245-270.

[Sh] P. Shiu, Counting sums of three squares, *Bull. Lond. Math. Soc.* **20** (1988), 203-208.

[Sl] N. J. A. Sloane, *A Handbook of Integer Sequences*, Academic Press, 1973.

[SS] A. Salomaa and M. Soittola, *Automata-theoretic Aspects of Formal Power Series*, Springer-Verlag, 1978.

[SS2] N. Strauss and J. Shallit, Advanced Problem 6625, *Amer. Math. Monthly* **97** (1990), 252.

[St] M. A. Stern, Über eine zahlentheoretische Funktion, *J. reine angew. Math.* **55** (1858), 193-220.

[Tap] T. Tapsoba, Complexité de suites automatiques, Thèse de troisième cycle, Université d'Aix-Marseille II, 1987.

[W] S. S. Wagstaff, Jr., The Schnirelmann density of the sums of three squares, *Proc. Amer. Math. Soc.* **52** (1975), 1-7.

Last revision: May 21, 1991

A PASCAL DIAMOND

William F. Klostermeyer, Michael E. Mays, and George Trapp

West Virginia University

January 1996

Abstract. A variation of Pascal's triangle, which is called a Pascal diamond, is introduced and some resulting number sequences are analyzed. Numbers in the Fibonacci sequence arise naturally as alternating sums of row elements. Explicit formulas are presented and some problems and conjectures are discussed.

INTRODUCTION. A generalization of Pascal's triangle can be defined using the following recurrence scheme. Given two rows of values, we compute a new row by adding together the four numbers in the diamond above the value to be computed. A sample diamond is given in figure 1. The value 16 is the sum of the four numbers above it in the diamond configuration.

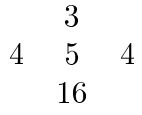


Figure 1: Sample Diamond

For our first application we start with one 1 in the first row and three 1's in the second row. The recurrence then determines the subsequent rows. The first few rows of the array are given in figure 2. We assume all blank positions are zero. So, for example, when calculating the second entry in the third row the 3 zeros are assumed to be up two places and up one and to the left. We call this array of

numbers a Pascal's diamond. We consider some properties of the diamond array below. For now we note that each row contains two more entries than the previous row, and each row is symmetric around the center line.

				1				
			1	1	1			
		1	2	4	2	1		
	1	3	8	9	8	3	1	
1	4	13	22	29	22	13	4	1

FIGURE 2: THE FIRST FIVE ROWS OF THE DIAMOND

This pattern generation scheme arose while studying a switch setting problem [1]. Given an n by m arrangement of switches, some on and some off, the goal is to achieve an all on configuration of the switches. Many puzzles and computer games are built using this idea. The operation available involves activating a particular switch, causing it and its rectilinearly adjacent neighbors to change states. We found that we could solve the switch setting problem by determining the null-space of a particular matrix defined by the switch arrangement topology. To determine the null-space we begin with an initial (row) vector containing one 1 and a second vector containing the three 1's under the initial vector's 1. We then "grow" the null-space vector by applying the diamond rule recursively. Our work on the switches differed in two ways from the recursion algorithm presented above. First, the rows are bounded by a certain fixed length and are not allowed to grow outward without bound on either the left or the right, and second, since the switches (in the simplest case) have only two states, all of the arithmetic is done modulo 2.

SOME PROPERTIES OF THE DIAMOND. Let $[n, k]$ represent the kth value of the nth row. The row numbering begins at 0 and the elements in a row also are numbered beginning at 0. We have $[0, 0] = 1$, and $[n, 0] = [n, 2n] = 1$ for

all n. The diamond then is the array beginning

$$\begin{array}{ccccccccc} & & & & [0,0] & & & & \\ & & & [1,0] & [1,1] & [1,2] & & & \\ & & [2,0] & [2,1] & [2,2] & [2,3] & [2,4] & & \\ & [3,0] & [3,1] & [3,2] & [3,3] & [3,4] & [3,5] & [3,6], & \end{array}$$

and the diamond defining recurrence relation can be written as

$$[n+1,k] = [n,k] + [n,k-1] + [n,k-2] + [n-1,k-2]. \tag{1}$$

Letting $k = 1$ in (1) gives the following relationship for the second entry of each row:

$$[n+1,1] = [n,1] + [n,0] = [n,1] + 1 \tag{2}$$

Two of the terms are missing in (2) because $k-2$ is -1, and the array values for negative k are taken to be 0. It follows directly from (2) that $[n,1] = n$ for all n. Writing down the recurrences for subsequent terms and solving them gives rise to the following formulas:

$$\begin{aligned} [n,2] &= (n^2 + 3n - 2)/2 \\ [n,3] &= (n^3 + 9n^2 - 22n + 12)/3! \\ [n,4] &= (n^4 + 18n^3 - 49n^2 + 6n + 48)/4! \\ [n,5] &= (n^5 + 30n^4 - 45n^3 - 570n^2 + 1904n - 1680)/5! \\ [n,6] &= (n^6 + 45n^5 + 55n^4 - 2865n^3 + 12184n^2 - 18780n + 8640)/6! \end{aligned}$$

We state the general result below.

Theorem 1. *$[n,k]$ is a polynomial in n of degree k, such that $k![n,k]$ is monic with integer coefficients.*

Proof. First rewrite (1) as

$$[n,k] - [n-1,k] = [n-1,k-1] + [n-1,k-2] + [n-2,k-2].$$

Treat this as an identity in the variable n and constant k, and sum over n. The least value of n to use is the last non-zero entry in the appropriate diagonal. It can be written as $\lfloor (k+1)/2 \rfloor$ to account for parity of k. Then

$$\begin{aligned}
[n,k] &= \sum_{i=\lfloor \frac{k+1}{2} \rfloor}^{n} ([i,k-1]+[i,k-2]+[i-1,k-2]) \\
&\quad + [\lfloor \frac{k+1}{2} \rfloor, k] \\
&= \sum_{i=\lfloor \frac{k+1}{2} \rfloor}^{n} ([i,k-1]+2[i,k-2]) \\
&\quad + [\lfloor \frac{k+1}{2} \rfloor - 1, k-2] - [n-1,k-2] + [\lfloor \frac{k+1}{2} \rfloor, k]
\end{aligned}$$

The sequence of polynomials thus continues, with the general recurrence establishing by induction that $[n,k]$ is a polynomial in n of degree k, such that $k![n,k]$ is monic with integer coefficients.

Theorem 2. *Let T_n be the sum of the elements in row n of the Pascal diamond. Then $lim_{n\to\infty} T_{n+1}/T_n = (3+\sqrt{13})/2$.*

Proof. Using the recurrence in (1) we have that

$T_{n+1} = 3T_n + T_{n-1}$. Now let $x = T_{n+1}/T_n$ and we see that x satisfies the quadratic equation $x = 3 + 1/x$, equivalently $x^2 - 3x - 1 = 0$. So we have that the ratio of sums of consecutive rows of the diamond approaches $(3+\sqrt{13})/2 = 3.3027756\ldots$

The first few values of the row sums are 1, 3, 10, 33, 109, 360, 1189, 3927, 12970, 42837, 141481, and 467280. This sequence has arisen several times in the literature, with several references available in the Encylopedia of Integer Sequences [2]. Another famous sequence also arises in connection with row sums, as the next theorem notes.

Theorem 3. $\sum_{i=0}^{n} (-1)^i [n,i] = F_{n+1}$, *the $n+1$st Fibonacci number.*

Proof. This follows immediately by induction.

A combinatorial problem to which the numbers in the Pascal diamond provide an insight is the focus of the next theorem.

Theorem 4. *Define an infinite directed graph $G(V, E)$ by using as the vertex set V points corresponding to the non-zero entries $[n, k]$ of the Pascal diamond array, and creating directed edges in E from the vertex $[n, k]$ to the vertices $[n + 1, k]$, $[n + 1, k + 1]$, $[n + 1, k + 2]$, and $[n + 2, k + 2]$. Then the number of distinct paths from $[0, 0]$ to $[n, k]$ is given by the value of $[n, k]$.*

Proof. Again, an easy proof is available by induction.

A SECOND RECURRENCE. In the switch setting problem, null-space vectors were built using the diamond rule modified to use a leftmost column entries that remained zero. In this case an array arises that is left justified: the only new non-zero values in successive rows appear on the right. This results in an array as shown in figure 3.

1					
1	1				
3	2	1			
6	7	3	1		
16	18	12	4	1	
40	53	37	18	5	1

FIGURE 3: ANOTHER RECURRENCE

In this left bounded array, each row contains one more element than the previous row. Clearly the last element of each row is 1 and the next to last element is n. We choose a notation that makes it easiest to describe a unimodal property of the rows of this array, rather than highlighting the correspondence between this array and the Pascal diamond. Thus we index from the center to keep track of these elements as

$\langle 1, 1\rangle$				
$\langle 2, 1\rangle$	$\langle 2, 2\rangle$			
$\langle 3, 1\rangle$	$\langle 3, 2\rangle$	$\langle 3, 3\rangle$		
$\langle 4, 1\rangle$	$\langle 4, 2\rangle$	$\langle 4, 3\rangle$	$\langle 4, 4\rangle$	
$\langle 5, 1\rangle$	$\langle 5, 2\rangle$	$\langle 5, 3\rangle$	$\langle 5, 4\rangle$	$\langle 5, 5\rangle$

It is natural to look for connections between this array and the Pascal diamond. Then the second array arises to the right of a central column of zeros, with the same array of negative numbers arising to the right. An engine for generating connections is the identity (which can be easily verified inductively)

$$[n,k]-[n,k-2]=\langle n+1,k-n+1\rangle. \tag{3}$$

Identity (3) can be used to extend the left bounded array leftward beyond the (implicit) column of zeros. Since the Pascal diamond is symmetrical, what is generated is a mirror image of the left bounded array, except that all the entries are negative. In fact, one way of obtaining the left bounded array is to start the Pascal diamond using the original recurrence, but with the two initial rows 0 and -1 0 1. Identity (3) also applies to provide an analogous result to Theorem 1, giving a second family of monic polynomials in n with integer coefficients. The first few values are listed below.

$$\langle n,n-2\rangle=(n^2+n-6)/2!$$
$$\langle n,n-3\rangle=(n^3+6n^2-43n+48)/3!$$
$$\langle n,n-4\rangle=(n^4+14n^3-109n^2+142n+24)/4!$$
$$\langle n,n-5\rangle=(n^5+25n^4-175n^3-385n^2+3534n-4920)/5!$$
$$\langle n,n-6\rangle=(n^6+39n^5-185n^4-3075n^3+23584n^2-56364n+43200)/6!$$

The row sums for the left bounded array,

$$\sum_{i=1}^{n}\langle n,i\rangle,$$

generating $\{U_n\}=\{1,2,6,17,51,154,473,1464,\dots\}$, are more difficult to analyze than the row sums in the Pascal diamond. Nevertheless, the same limiting value of ratios of successive rows exists.

Theorem 5. *Let U_n be the sum of the elements in row n of the left bounded array. Then $lim_{n\to\infty} U_{n+1}/U_n = (3+\sqrt{13})/2$.*

Proof. The row sum recurrence is $U_{n+1} = 3U_n + U_{n-1} - \langle n, n\rangle$, so it is enough to show that $\langle n, n\rangle = o(U_n)$ as $n \to \infty$ in order to make the argument of Theorem 2 apply. We show more, that the rows of the left bounded array are unimodal with a maximum value that moves ever rightward. First, note that path counting result of Theorem 4 applies to the left bounded array as well, with the change that the infinite directed graph has a restricted set of edges down the left column (thus the vertices labeled $\langle n, 1\rangle$ have only three outgoing edges, and the other vertices have four).

Now we need to establish the "more".

We believe the maximum value in row n occurs in position $O(\log n)$ or greater, perhaps as much as $O(n)$. The latter conjecture is based on an analogy with Pascal's triangle explored in the next section.

A CONNECTION WITH PASCAL'S TRIANGLE. A left bounded array can be constructed using the recurrence for Pascal's triangle as well:

	1							
0		1						
	1		1					
0		2		1				
	2		3		1			
0		5		4		1		
	5		9		5		1	

FIGURE 3: LEFT BOUNDED PASCAL'S TRIANGLE

This time, the analog of (3) holds to give these table entries as differences of binomial coefficients. Hence the maximum value in row n of this array occurs at the k value that gives the maximum value of the difference in binomial coefficients in row n of Pascal's triangle. But as n grows, by the classical limit theorem of De Moivre and Laplace the binomial distribution approaches a normal distribution,

and the maximum (absolute) derivative of this function occurs at a fixed value of x. Scaling n to x gives the value of k where the maximum difference occurs, a value that increases linearly with n.

References

1. J. Goldwasser, W. Klostermeyer, G. Trapp and C. Q. Zhang, *Setting switches in a grid*, Technical Report 95-20, Dept. of Stat. and Computer Science, West Virginia University.
2. N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, 1995.

Department of Mathematics
Department of Statistics & Computer Science
West Virginia University, Morgantown, WV 26506-6330
e-mail: mays@math.wvu.edu, {wfk, trapp}@cs.wvu.edu

Unknown Journal Series
Volume **000**, Number 0, 0000

Arrays, Numeration Systems and Frankenstein Games

Aviezri S. Fraenkel

Department of Applied Mathematics and Computer Science
Weizmann Institute of Science
Rehovot 76100, Israel
fraenkel@wisdom.weizmann.ac.il
http://www.wisdom.weizmann.ac.il/~fraenkel

Abstract. We define an infinite array $\mathcal{A}$ of nonnegative integers based on a linear recurrence, whose second row provides basis elements of an exotic ternary numeration system. Using the numeration system we explore many properties of $\mathcal{A}$. Further, we propose and analyze a family *Frankenstein* of 2-player pebbling games played on a semi-infinite strip, and present a winning strategy based on certain subarrays of $\mathcal{A}$. Though the strategy looks easy, it is actually computationally hard. The numeration system is then used to decide whether the family has an efficient strategy or not.

1. Introduction

Consider a doubly infinite array (matrix) $\mathcal{A} = \{A_j^n : 0 \le j,\ n \le \infty\}$ of nonnegative integers whose first few entries are displayed in Table 1. To define its formation rule, we introduce a little notation.

Denote by $\mathbb{Z}$, $\mathbb{Z}^0$ and $\mathbb{Z}^+$ the set of integers, nonnegative integers and positive integers respectively. If S is any *proper* subset of $\mathbb{Z}^0$, i.e., $S \neq \mathbb{Z}^0$, denote by $\operatorname{mex} S$ the least nonnegative integer in the complement of S with respect to $\mathbb{Z}^0$, i.e., the least nonnegative integer not occurring in S. Note that the mex of the empty set is 0. The term mex, introduced in [BCG1982], stands for Minimum EXcluded value.

For $n \in \mathbb{Z}^0$, the entries of the array are defined as follows.

$$A_0^n = \operatorname{mex}\{A_j^i : 0 \le i < n,\ j \ge 0\}, \tag{1}$$

???-????/00 $1.00 + $.25 per page

TABLE 1: A DOUBLY INFINITE ARRAY OF NONNEGATIVE INTEGERS.

n	A_0^n	A_1^n	A_2^n	A_3^n	A_4^n	A_5^n	A_6^n	
0	0	0	0	0	0	0	0	
1	1	3	8	21	55	144	377	
2	2	6	16	42	110	288	754	
3	4	11	29	76	199	521	1364	
4	5	14	37	97	254	665	1741	...
5	7	19	50	131	343	898	2351	
6	9	24	63	165	432	1131	2961	
7	10	27	71	186	487	1275	3338	
8	12	32	84	220	576	1508	3948	
9	13	35	92	241	631	1652	4225	
10	15	40	105	275	720	1885	4935	
		⋮						

$$A_1^n = 2A_0^n + n \ (n \geq 0), \quad A_j^n = 3A_{j-1}^n - A_{j-2}^n \ \ (j \geq 2, \ n \geq 0). \tag{2}$$

It can be seen, by induction on n, that the set on the right hand side of (1) is indeed a proper subset of $\mathbb{Z}^0$.

We further introduce a special ternary numeration system $\mathcal{U}$. Its basis elements are defined by $u_0 = 1$, $u_1 = 3$, $u_i = 3u_{i-1} - u_{i-2}$ $(i \geq 2)$.

Theorem I. *Every positive integer n has a unique representation over $\mathcal{U}$, in the form $n = \sum_{i \geq 0} d_i u_i$, where the digits d_i assume values in $\{0, 1, 2\}$, subject to the following special condition: if for some $0 \leq j < l$, $d_j = d_l = 2$, then there exists k satisfying $j < k < l$ (so actually $l - j \geq 2$), such that $d_k = 0$.*

Theorem I is a special case of Theorem 4, stated and proved in [Fra1985, §4]. The representation of the first few positive integers over $\mathcal{U}$ is given in Table 2. We write the representation of n both in terms of its basis elements, $n = \sum_{i=0}^{m} d_i u_i$, and in its "ternary" form $n = d_m \ldots d_0$, the same as is customary for more conventional numeration systems, such as decimal or binary ($528 = 8 \times 10^0 + 2 \times 10^1 + 5 \times 10^2$). Table 2 shows, for example, that $41 = 1211$; and $42 = 2000$ rather than 1212, because of the special condition. Similarly, $55 = 10000$, not 2112.[1]

[1]Some of my best friends are nonsemitic, among them referees and readers of my articles. A number of them have commented to me that in a table such as Table 2, the basis elements $1, 3, 8, 21, 55$ should be written from left to right rather than from right to left. I disagree. The "ternary" number $n = d_m \ldots d_0$, now easily readable from the table, would be reversed! There is a discrepancy in nonsemitic languages, often ignored, between text, including mathematical

TABLE 2: A SPECIAL TERNARY REPRESENTATION OF INTEGERS n.

55	21	8	3	1	n	21	8	3	1	n
	1	1	0	2	31				1	1
	1	1	1	0	32				2	2
	1	1	1	1	33			1	0	3
	1	1	1	2	34			1	1	4
	1	1	2	0	35			1	2	5
	1	1	2	1	36			2	0	6
	1	2	0	0	37			2	1	7
	1	2	0	1	38		1	0	0	8
	1	2	0	2	39		1	0	1	9
	1	2	1	0	40		1	0	2	10
	1	2	1	1	41		1	1	0	11
	2	0	0	0	42		1	1	1	12
	2	0	0	1	43		1	1	2	13
	2	0	0	2	44		1	2	0	14
	2	0	1	0	45		1	2	1	15
	2	0	1	1	46		2	0	0	16
	2	0	1	2	47		2	0	1	17
	2	0	2	0	48		2	0	2	18
	2	0	2	1	49		2	1	0	19
	2	1	0	0	50		2	1	1	20
	2	1	0	1	51	1	0	0	0	21
	2	1	0	2	52	1	0	0	1	22
	2	1	1	0	53	1	0	0	2	23
	2	1	1	1	54	1	0	1	0	24
1	0	0	0	0	55	1	0	1	1	25
1	0	0	0	1	56	1	0	1	2	26
1	0	0	0	2	57	1	0	2	0	27
1	0	0	1	0	58	1	0	2	1	28
1	0	0	1	1	59	1	1	0	0	29
1	0	0	1	2	60	1	1	0	1	30

formulas, and "digital" numbers. Though all of these are both written and read from left to right, the basis elements of the latter, which are usually implicit but here explicit, nevertheless increase from right to left. (There is an even greater discrepancy when embedding formulas and digital numbers in semitic language texts, but it is well-known and acknowledged. Moreover, word processors have long since learned to overcome it; human beings still have difficulties with it.)

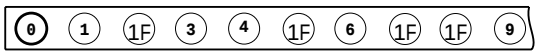


FIGURE 1.1. A position in *Frankenstein* with 1Fr. coins.

Lastly, we define a two-person pebbling game called *Frankenstein*[2], played on a semi-infinite strip with a finite number of pebbles, say coins, at most one per square. The squares are numbered with the nonnegative integers $0, 1, 2, \ldots$ from the left end of the strip, as in Fig. 1. There is a hole at square 0: a coin landing on it falls through the hole, disappearing from the play. The empty strip is denoted by ϕ. A single coin on the strip is a *spinster*. A legal move is to shift a number of coins from their present squares to *any* unoccupied squares with a lower number (a left shift), avoiding a spinster: we never permit a spinster position. Every move of ≥ 2 coins involves a *sequential* shifting of coins: an arbitrary coin is first shifted. Then a coin to its left is shifted, then a coin to its left, and so on. Every coin is shifted *at most once* in a single move. Also new coins can be created. Specifically, the moves from a position with say k $(k \geq 2)$ coins on squares

$$X = (x_0, \ldots, x_{k-1}), \quad 0 < x_0 < \cdots < x_{k-1}, \tag{3}$$

are of two types.

I (a) Shift a positive number of at most $k-1$ tokens, at least one of them to a *positive* numbered square. (b) A coin on precisely one square m may be shifted to 0 and new coins be placed on the unoccupied squares $j_1, \ldots, j_\ell$ if and only if $0 < \sum_{i=1}^{\ell} j_i < m$. A move consists of either (a) or (b) (or both).

II Shift all of the tokens by say, $0 < n_0 \leq \cdots \leq n_{k-1}$ squares, either preserving k or resulting in Φ. Moreover, n_{k-1} should not be too large; namely,

$$n_{k-1} \leq 2n_{k-2} + n_{k-3} + \cdots + n_0. \tag{4}$$

The player first unable to move loses, and the opponent wins. Notice that in every position there is at most one coin per square, and the only end position is Φ. A spinster is never permitted. In a type **II** move, either all coins are removed, or none. The number of coins can decrease or increase during play; but the sum of the occupied square numbers decreases at each move. Therefore play ends, and no game position is repeated.

Examples.

(i) Let $X = (1, 3)$. A move $X \to (0, 0)$ is inconsistent with (4). Also a move to 1 or 3 is not permitted, since they are spinsters (and also by the second part of **I**(a)). Thus the only possible move is to $(1, 2)$. Then player II can move to $(0, 0)$, winning.

(ii) From the position $(1, 3, 7)$ player I can move to Φ winning instantly, because the move $7 \to 0$, $3 \to 0$, $1 \to 0$ satisfies 4 (with equality).

[2]The game is played with coins called Francs (in Belgium or France) and Franks (in Switzerland). Alternatively, it may be played with pebbles or stones. Hence the name of the game.

(iii) Given the initial position $X = (1, 3, 8)$. A move $X \to (1, 3)$ is not permitted by the second part of **I**(a). It can be seen that if only the coin at 8 is shifted, then player II can move to Φ in the next move. We leave it to the reader to verify that X is a position from which player II can win, either by moving directly to Φ or by moving first to $(1, 3)$.

(iv) The winning move $(6, 8, 100) \to (1, 3, 8)$ involves (a): $100 \to 1$, $6 \to 3$ (or $100 \to 3$, $6 \to 1$).

(v) The winning move $(8, 19) \to (1, 3, 8)$ is of type (b): $19 \to (1, 3)$.

(vi) Show that $(55, 56, 200) \to (1, 3, 8, 21, 55)$ is a winning move $\big($involving both (a): $200 \to 8$ and (b): $(56 \to (1, 3))\big)$.

(vii) Verify that player II can win from the position $(2, 6)$.

We shall show that certain subarrays of the array $\mathcal{A}$ are the so-called "losing positions" of Frankenstein. For proving this it is helpful to use some of the properties of $\mathcal{A}$. Essentially, $\mathcal{A}$ is a splitting of $\mathbb{Z}^+$, but to state the result precisely, some further notions will first be introduced.

Define the operators L (*L*eft shift) and R (*R*ight shift) on representations over $\mathcal{U}$: if $n = \sum_{i=0}^{m} d_i u_i$ for some $n \in \mathbb{Z}^+$, then $L(n) = \sum_{i=0}^{m} d_i L(u_i) = \sum_{i=0}^{m} d_i u_{i+1}$; and $R(n) = \sum_{i=1}^{m} d_i R(u_i) = \sum_{i \geq 1} d_i u_{i-1}$ is defined if $d_0 = 0$. In other words, if $n = d_m \dots d_0$, then $L(n) = d_m \dots d_0 0$, and, if $i \geq 1$ (i.e., $d_0 = 0$), then $R(d_m \dots d_1 0) = d_m \dots d_1$. In particular: $L(u_i) = u_{i+1}$ $(i \geq 0)$; and $R(u_i) = u_{i-1}$ $(i \geq 1)$.

The j-th column of $\mathcal{A}$, excluding the 0 in the first row, is denoted by $A_j = \bigcup_{n=1}^{\infty} A_j^n$, $j \geq 0$; and the n-th row is $A^n = \bigcup_{j=0}^{\infty} A_j^n$, $n \geq 1$. If $n = \sum_{i \geq 0} d_i u_i$ with $d_0 \neq 0$, we say that n is *reduced*. A reduced number n has no right shift. The *golden section* is the positive root ϕ of the polynomial equation $x^2 - x - 1 = 0$, so $\phi = (1 + \sqrt{5})/2$ and $\phi^2 = \phi + 1$.

In §2 we prove,

Theorem 1. *The array $\mathcal{A}$ is a splitting of $\mathbb{Z}^+$: every positive integer appears precisely once in $\mathcal{A}$. Moreover, for every $j \geq 0$, the column A_j consists precisely of all positive integers whose representation ends in j 0s. In particular, A_0^n is reduced for all $n \in \mathbb{Z}^+$.*

The proof leans heavily on properties of the special ternary numeration system $\mathcal{U}$, which are also explored in §2. The system $\mathcal{U}$ is even more useful: the winning strategy for Frankenstein, based on subarrays of $\mathcal{A}$, is inefficient (exponential). The system $\mathcal{U}$ enables one to decide whether there is or there isn't a different, efficient (polynomial) strategy. This is taken up in §4. Some further remarkable properties of $\mathcal{A}$ are listed in Theorem 2, also proved in §2.

Let $f_0 = 1$, $f_1 = 2$, $f_n = f_{n-1} + f_{n-2}$ $(n \geq 2)$ be the sequence of Fibonacci numbers. (It is easily seen that the numeration basis elements u_i defined above, which constitute the second row of $\mathcal{A}$, are precisely the "even" Fibonacci numbers, i.e., $u_i = f_{2i}$ for all $i \geq 0$. Also the other rows of $\mathcal{A}$ are "even Fibonacci numbers" with different initial conditions, but these facts are not needed here.)

Theorem 2.

(i) *For $j, n \in \mathbb{Z}^+$, $A_j^n = \lfloor A_{j-1}^n \phi^2 \rfloor + 1 = L(A_{j-1}^n)$.*

(ii) *For all $n \geq 1$, $A_0^n = \lfloor (n-1)\phi \rfloor + 1$ is reduced.*

(iii) *For all $n \geq 0$, all $j \geq 0$ we have, $A_j^{n+1} - A_j^n \in \{f_{2j}, f_{2j+1}\}$, and for fixed j, each of f_{2j}, f_{2j+1} is assumed for infinitely many n. Moreover, for all n for which $A_0^{n+1} - A_0^n = f_0$ (respectively f_1), we also have for all j, $A_j^{n+1} - A_j^n = f_{2j}$ (respectively f_{2j+1}).*

(iv) *Let $j \geq 1$. There are no real numbers α, γ, such that for all $n \geq 1$, $A_j^n = \lfloor n\alpha + \gamma \rfloor$.*

Properties of $\mathcal{A}$ are also presented in Lemmas 1 and 2 in §2. The formulation of a winning strategy for Frankenstein needs a few technical concepts, so is best postponed to §3, where the precise result is stated and proved. A sum up is presented in the final §5.

2. Some Properties of the Array

We begin with a simple result.

Lemma 1. *For all $j, n \in \mathbb{Z}^+$, $A_j^n = 2A_{j-1}^n + A_{j-2}^n + \cdots + A_0^n + n$.*

Proof. Induction on j, for arbitrary but fixed n. By the first part of (2), the assertion holds for $j = 1$. Suppose it holds for some $j \geq 1$. By the second part of (2),

$$A_{j+1}^n = 2A_j^n + (A_j^n - A_{j-1}^n) = 2A_j^n + A_{j-1}^n + \cdots + A_0^n + n. \quad \blacksquare$$

The following is the main lemma used for proving both Theorem 1 and Theorem 2.

Lemma 2. *Let $n \geq 1$, $\mathcal{S}_n = \bigcup_{m<n} \bigcup_{j=0}^{\infty} A_j^m$. In every row A^n of $\mathcal{A}$, the element A_0^n is the smallest reduced element not in $\mathcal{S}_n$, and A_{j+1}^n is the left shift of A_j^n for all $j \geq 0$.*

Proof. Since $u_1 = 2u_0 + 1$ and $u_i = 3u_{i-1} - u_{i-2}$ $(i \geq 2)$, the same as the recurrence (2), and $A_0^1 = 1 = u_0$, the row A^1 consists of the basis elements of $\mathcal{U}$, for which the statement clearly holds. Suppose it holds for all $m < n$ $(n \geq 2)$. If A_0^n would not be reduced, then $R(A_0^n)$ would be a smaller element than A_0^n. Moreover, $R(A_0^n) \notin \mathcal{S}_n$, otherwise also $A_0^n = LR(A_0^n)$ would be in $\mathcal{S}_n$, by the induction hypothesis, contradicting (1). Thus A_0^n is the smallest reduced element not in $\mathcal{S}_n$.

Let $A_0^{n-1} = \sum_{i \geq 0} d_i u_i$ be the representation of A_0^{n-1} over $\mathcal{U}$. By the induction hypothesis, $A_1^{n-1} = L(A_0^{n-1}) = \sum_{i \geq 0} d_i u_{i+1}$ and A_0^{n-1} is reduced. In particular, $d_0 \neq 0$. We consider two cases.

(i) There exists $j \geq 1$ such that $d_i = 1$ for all $i < j$, but $d_j = 0$. Then $A_0^{n-1} + 1 = \sum_{i \geq 1} d_i u_i + (d_0 + 1)u_0$ is reduced (by Theorem I, with least significant digit 2), so the first part of the proof implies $A_0^{n-1} + 1 = A_0^n$. Now,

$$\begin{aligned} A_1^n &= 2A_0^n + n = 2(A_0^{n-1} + 1) + n = 2A_0^{n-1} + (n-1) + 3 = A_1^{n-1} + 3 \\ &= \sum_{i \geq 0} d_i u_{i+1} + u_1 = \sum_{i \geq 1} d_i u_{i+1} + (d_0 + 1)u_1 = L(A_0^n). \end{aligned}$$

(ii) There exists $j \geq 0$ such that $d_i = 1$ for all $i < j$, but $d_j = 2$. By Theorem I, $d_{j+1} \leq 1$. By Lemma 1 with $n = 1$, $A_0^{n-1} + 1 = \sum_{i \geq j+2} d_i u_i + (d_{j+1} + 1) u_{j+1}$ is not reduced, but $A_0^{n-1} + 2 = \sum_{i \geq j+2} d_i u_i + (d_{j+1} + 1) u_{j+1} + u_0 = A_0^n$ is reduced. Then by Lemma 1 ($n = 1$),

$$\begin{aligned} A_1^n &= 2A_0^n + n = 2(A_0^{n-1} + 2) + n = 2A_0^{n-1} + (n-1) + 5 \\ &= A_1^{n-1} + 5 = \sum_{i \geq 0} d_i u_{i+1} + 5 = \sum_{i \geq j+2} d_i u_{i+1} + \sum_{i=0}^{j+1} d_i u_{i+1} + 5 \\ &= \sum_{i \geq j+2} d_i u_{i+1} + d_{j+1} u_{j+2} + (2u_{j+1} + u_j + \cdots + u_1) + (u_0 + 1 + 3) \\ &= \sum_{i \geq j+2} d_i u_{i+1} + (d_{j+1} + 1) u_{j+2} + u_1 = L(A_0^n). \end{aligned}$$

It remains only to show that $A_{j+1}^n = L(A_j^n)$ for all $j \geq 1$. Suppose we already showed this for all $j < m$. For $m = 1$ this was just done. So consider A_{m+1}^n. Let $A_{m-1}^n = \sum_{i \geq 0} d_i u_i$ be the representation of A_{m-1}^n. By the induction hypothesis and (2),

$$A_{m+1}^n = 3A_m^n - A_{m-1}^n = \sum_{i \geq 0} d_i (3u_{i+1} - u_i) = \sum_{i \geq 0} d_i u_{i+2} = L(A_m^n). \quad \blacksquare$$

Proof of Theorem 1. By Lemma 2, $A_{j+1}^n = L(A_j^n)$. Therefore the representation of A_{j+1}^n has one additional 0 at its tail end than that of A_j^n. Since the representations of positive integers over $\mathcal{U}$ are unique (Theorem I), all entries in $\mathcal{A}$ are indeed distinct. Finally, every positive integer appears in $\mathcal{A}$ in view of (1). $\blacksquare$

For proving the left shift part of Theorem 2(i), we prove, more generally,

Lemma 3. *Let $m \in \mathbb{Z}^+$, $n = \lfloor m\phi^2 \rfloor + 1$. Then $n = L(m)$.*

Proof. Let $m = \sum_{i=0}^r d_i u_i$ be the representation of m, for suitable $r \in \mathbb{Z}^0$. We have to show: $n = \sum_{i=0}^r d_i u_{i+1}$. It suffices to show that $0 < m\phi^2 + 1 = \sum_{i=0}^r d_i u_{i+1} + \rho$, for some $0 < \rho < 1$. So it suffices to show that $0 < \sum_{i=0}^r d_i (u_{i+1} - u_i \phi^2) < 1$.

The characteristic equation of the second recurrence of (2) is $x^2 - 3x + 1 = 0$, with solutions $\phi^2 = (3 + \sqrt{5})/2$ and conjugate $\phi^{-2} = (3 - \sqrt{5})/2$. From this it follows that for $n \geq 0$,

$$u_n = \frac{\phi^{2n+2} - \phi^{-(2n+2)}}{\sqrt{5}}. \tag{5}$$

Then $u_{i+1} - u_i \phi^2 = \phi^{-2i}(1 - \phi^{-4})/\sqrt{5} > 0$. Note that, due to the special condition of Theorem I, $\sum_{i=0}^r d_i (u_{i+1} - u_i \phi^2)$ is largest when $d_0 = 2$ and $d_i = 1$ for all $i \geq 1$. Thus,

$$0 < \sum_{i=0}^r d_i (u_{i+1} - u_i \phi^2) < \frac{(1 - \phi^{-4})}{\sqrt{5}} (1 + \sum_{i=0}^{\infty} \phi^{-2i}) = \frac{(1 - \phi^{-4})}{\sqrt{5}} \phi^2 = 1. \quad \blacksquare$$

For proving (iv) of Theorem 2, we prove a technical result.

Lemma 4. *Let $\alpha > 0$, γ be real numbers. Letting $N_n = \lfloor (n+1)\alpha + \gamma \rfloor - \lfloor n\alpha + \gamma \rfloor$, we have*

$$\lfloor \alpha \rfloor \leq N_n \leq \lceil \alpha \rceil. \tag{6}$$

Moreover, each of the values $\lfloor \alpha \rfloor$ and $\lceil \alpha \rceil$ is assumed for infinitely many n.

Proof. The definition of N_n implies (6) directly. If $\alpha = p/q$ with gcd $(p, q) = 1$ is rational, then we may clearly assume, without loss of generality, that $\gamma = r/q$ ($p \in \mathbb{Z}^0$, $q \in \mathbb{Z}^+$, $r \in \mathbb{Z}$). The congruence $xp \equiv q - r \pmod{q}$ has a solution $x = n_0$, $0 \leq n_0 < q$, so $n_0 p = kq - r$ for some $k \in \mathbb{Z}$. It is then easily verified that $N_{n_0 - 1} = \lceil \alpha \rceil$, and $N_{n_0} = \lfloor \alpha \rfloor$. Since the above congruence has the general solution $n = n_0 + sq$, $s \in \mathbb{Z}$, each of the values $\lfloor \alpha \rfloor$ and $\lceil \alpha \rceil$ is assumed infinitely often.

If α is irrational, then the fractional values $(n\alpha)$ are dense in $(0, 1)$ (Kronecker's Theorem; see e.g., [HaWr1989], Ch. 23). Hence each of $\lfloor \alpha \rfloor$ and $\lceil \alpha \rceil$ is assumed infinitely often also in this case. ■

Proof of Theorem 2. From Lemma 3 we have, in particular, $\lfloor A^n_{j-1}\phi^2 \rfloor + 1 = L(A^n_{j-1})$. By Lemma 2, this is also the same as A^n_j for all $j \geq 1$, proving (i).

Since $\phi^{-1} + \phi^{-2} = 1$, it follows from Theorem II of [Fra1969] that if $S = \bigcup_{n=1}^{\infty}(\lfloor n\phi \rfloor + 1)$, $T = \bigcup_{n=1}^{\infty}(\lfloor n\phi^2 \rfloor + 1)$, then S, T are 2-*upper complementary*, i.e., $S \cup T = \mathbb{Z}^+ \setminus \{1\}$ and $S \cap T = \emptyset$. By Lemma 3, T contains only non reduced numbers. Hence S consists of precisely all the reduced numbers > 1, and T of all the non reduced numbers. Replacing n by $n - 1$, (ii) follows from (1).

For establishing (iii), we use induction on j. For $j = 0$, the claim follows directly from Lemma 4 and (ii), with $\alpha = \phi$, $\gamma = 1$. For $j = 1$ we have by (2), $A^{m+1}_1 - A^m_1 = 2(A^{m+1}_0 - A^m_0) + 1 \in \{2f_0 + 1, 2f_1 + 1\} = \{f_2, f_3\}$; and f_2 (respectively f_3) is assumed precisely when $2(A^{m+1}_0 - A^m_0) = f_0$ (f_1 respectively). Suppose it holds for all $i < j$ ($j \geq 2$). By (2), $A^{m+1}_j - A^m_j = 3(A^{m+1}_{j-1} - A^m_{j-1}) - (A^{m+1}_{j-2} - A^m_{j-2})$. This is either $3f_{2j-2} - f_{2j-4} = f_{2j}$ or f_{2j+1}, according to whether in the previous column $(j - 1)$ the result was $f_{2(j-1)}$ or f_{2j-1}. We have demonstrated the validity of (iii).

By Lemma 4, a necessary condition for the existence of real α, γ with α positive and irrational such that $A_j = \lfloor n\alpha + \gamma \rfloor$, is that $A^{n+1}_j - A^n_j \in \{\lfloor \alpha \rfloor, \lceil \alpha \rceil\}$. In particular, $A^{n+1}_j - A^n_j$ has to assume two consecutive integer values. But by (iii), the two assumed values are f_{2j} and f_{2j+1}; which are consecutive if and only if $j = 0$. This proves (iv). ■

Remark. Theorem 2 can be used to give an independent proof of Theorem 1, because the former implies, using the uniqueness of representation (Theorem I), that all entries of $\mathcal{A}$ are distinct, $A_{j+1} = L(A_j)$, and also every positive integer is assumed.

3. A Winning Strategy for Frankenstein

Informally, a position u in a game such as Frankenstein is called a P-position, if the *P*revious player can win, i.e., the player who moved to u. It is an N-position, if the *N*ext player can win, i.e., the player moving from u. The position Φ is a P-position, since player I (the player called upon to move from the the given position), cannot even make a move, so the opponent, player II, wins by default. By $F(u)$ we

denote the set of all immediate followers of u, i.e., the set of all positions reachable from u by a single move . Note that $F(u) = \emptyset$ if u is a *leaf*, i.e., an end position.

Denote by $\mathcal{P}$ the set of all P-positions, and by $\mathcal{N}$ the set of all N-positions. The informal definition of P- and N-positions implies,

$$u \in \mathcal{P} \iff F(u) \subseteq \mathcal{N}, \quad u \in \mathcal{N} \iff F(u) \cap \mathcal{P} \neq \emptyset. \tag{7}$$

All of these things can be done formally. See [Fra≥2001].

For the sake of compactness of discussion, we will be talking about reducing integers, rather than shifting coins on squares numbered with those integers. In terms of this convention, we state the main result of this section.

Theorem 3. *The P-positions of the game Frankenstein are given by*

$$\mathcal{P} = \bigcup_{n=0}^{\infty} \bigcup_{k=2}^{\infty} (A_0^n, \ldots, A_{k-1}^n).$$

Proof. Let $W = \bigcup_{n=0}^{\infty} \bigcup_{k=2}^{\infty} (A_0^n, \ldots, A_{k-1}^n)$. As was pointed out in §1, the empty strip Φ is a leaf, i.e., $F(\Phi) = \emptyset$, and so is a P-position by (7). It turns out that in view of (7), it suffices to demonstrate the following two properties for all positions.

(A) Every move from a position in W produces a position not in W.
(B) From every position not in W there exists a move to a position in W.

(A) Let $(A_0^n, \ldots, A_{k-1}^n) \in W$. For a move of type **I**, there is a number A_j^n which remains fixed, and a number L which is either reduced or replaced by a collection of smaller numbers. In either case, the resulting position contains A_j^n and a number $L \neq A_i^n$ for all $i \geq 0$, so it is not in W.

Now consider a move of type **II**. Suppose there is a move $X = (A_0^n, \ldots, A_{k-1}^n) \to (A_0^m, \ldots, A_{j-1}^m) \in W$. If $m > n$ $\big($such as $(2, 6, 16) \to (4, 11)\big)$, the move involves $A_0^n \to 0$, contrary to the requirement of preserving k. Clearly we cannot have $m = n$. So $m < n$, $j \leq k$. Suppose first that $j < k$. If $m = 0$ (so $(A_0^m, \ldots, A_{j-1}^m) = \Phi$), we have, using Lemma 1,

$$A_{k-1}^n = 2A_{k-2}^n + \sum_{i=0}^{k-3} A_i^n + n > 2A_{k-2}^n + \sum_{i=0}^{k-3} A_i^n, \tag{8}$$

contradicting condition (4). This contradiction holds a fortiori if $m > 0$, because then the terms to the right of A_{k-1}^n in (8) are even smaller, but the left side is still A_{k-1}^n if $j < k$. We conclude that $j = k$.

The presumed move is thus $(A_0^n, \ldots, A_{k-1}^n) \to (A_0^m, \ldots, A_{k-1}^m)$. By Lemma 1,

$$\begin{aligned} A_{k-1}^n - A_{k-1}^m &= 2(A_{k-2}^n - A_{k-2}^m) + \sum_{i=0}^{k-3} (A_i^n - A_i^m) + n - m \\ &> 2(A_{k-2}^n - A_{k-2}^m) + \sum_{i=0}^{k-3} (A_i^n - A_i^m). \end{aligned} \tag{9}$$

This contradicts (4), since Theorem 2(iii) implies that for every $n > m > 0$ and all $j \geq 0$, $A^n_{j+1} - A^m_{j+1} > A^n_j - A^m_j$, so $A^n_{k-1} - A^m_{k-1} = \max_{0 \leq i \leq k-1}(A^n_i - A^m_i)$.

(B) Given a position $X = (x_0, \ldots, x_{k-1}) \notin W$ of the form (3), with $k \geq 2$. We show that there is a single move to a position in W. By complementarity (Theorem 1), $x_0 = A^n_{j-1}$ for some $j, n \in \mathbb{Z}^+$.

Assume first $j > 1$. Since $k \geq 2$, there is $x_1 > x_0$. By Lemma 1, $x_1 > x_0 = 2A^n_{j-2} + \sum_{i=0}^{j-3} A^n_i + n$. If $k \geq j$, we reduce $(x_1, \ldots, x_{j-1}) \to (A^n_0, \ldots, A^n_{j-2})$, and put $x_\ell \to 0$ for all $\ell \geq j$, if any. If $k < j$, we reduce $(x_1, \ldots, x_{k-2}) \to (A^n_0, \ldots, A^n_{k-3})$, and then split a suitably reduced x_{k-1} into $(A^n_{k-2}, \ldots, A^n_{j-2})$. In particular, if $k = 2$, then x_1 is reduced and split into $(A^n_0, \ldots, A^n_{j-2})$. We have made a type **I** move to $(A^n_0, \ldots, A^n_{j-1}) \in W$.

We may thus assume $x_0 = A^n_0$. Then there exists $j \geq 2$ such that $x_i = A^n_i$ for $i < j - 1$, but $x_{j-1} \neq A^n_{j-1}$. If $x_{j-1} > A^n_{j-1}$, move $x_{j-1} \to A^n_{j-1}$ and put $x_i \to 0$ for all $i > j - 1$.

So we may assume $x_{j-1} < A^n_{j-1}$. We consider the following cases.

(i) $j = k$, so $x_{j-1} = x_{k-1}$. We have $x_{k-1} = A^n_{k-1} - t$ for some $t \geq 1$. We claim that $X = (A^n_0, \ldots, A^n_{k-2}, x_{k-1}) \to (A^{n-t}_0, \ldots, A^{n-t}_{k-2}, A^{n-t}_{k-1}) \in W$ is a legal type **II** move for $t < n$; and $X \to \Phi$, for $t \geq n$.

For $t < n$ we have by (9) (with $m = n - t$), $x_{k-1} - A^{n-t}_{k-1} = A^n_{k-1} - A^{n-t}_{k-1} - t > A^n_{k-2} - A^{n-t}_{k-2}$. Then by Lemma 1,

$$x_{k-1} - A^{n-t}_{k-1} = A^n_{k-1} - A^{n-t}_{k-1} - t = 2(A^n_{k-2} - A^{n-t}_{k-2}) + \sum_{i=0}^{k-3}(A^n_i - A^{n-t}_i),$$

which satisfies (4). If $t \geq n$, then by Lemma 1, $x_{k-1} \leq A_{k-1} - n = 2A^n_{k-2} + \sum_{i=0}^{k-3} A^n_i$, so $X \to \Phi$ satisfies (4).

(ii) $j < k$. (Recall that $x_{j-1} < A^n_{j-1}$.) We first dispose of two subcases.

a. If there is $r > j - 1$ with $x_r > A^n_{j-1}$ and $x_i \neq A^n_{j-1}$ for all $i \geq 0$, then make the type **I** move $x_r \to A^n_{j-1}$ and $x_i \to 0$ for all $i \geq j - 1$, $i \neq r$, resulting in $(A^n_0, \ldots, A^n_{j-1}) \in W$.

b. If $x_i \leq A^n_{j-1}$ for all $i > j - 1$, then $X = (A^n_0, \ldots, A^n_{j-2}, x_{j-1}, \ldots, x_{k-1}) \to \Phi$ is a legal move. Indeed, $x_{j-1} > A^n_{j-2}$; and Lemma 1 implies $A^n_{j-2} \geq n$. Hence,

$$x_{k-1} \leq A^n_{j-1} = 2A^n_{j-2} + \sum_{i=0}^{j-3} A^n_i + n < 2x_{j-1} + \sum_{i=0}^{j-3} A^n_i \leq 2x_{k-2} + \sum_{i=0}^{k-3} x_i,$$

is a legal type **II** move by (4).

So we may assume that X has the form

$$X = (A^n_0, \ldots, A^n_{j-2}, x_{j-1}, \ldots, A^n_{j-1}, \ldots, A^n_{j+s}, x_t, \ldots, x_{k-1}),$$

where each A^n_{j+i} appears for all $i \leq s$, $s \geq -1$, and possibly also some intermediate $x_i \neq A^n_r$, but A^n_{j+s+1} does not appear. Here are the two final subcases.

c. $x_{k-1} > A^n_{j+s+1}$. Then move, $x_{k-1} \to A^n_{j+s+1}$, $x_{j-1}, \ldots, x_t, \ldots, x_{k-2} \to 0$ (type **I** move), resulting in the position $(A^n_0, \ldots, A^n_{j+s+1}) \in W$.

d. $x_{k-1} < A^n_{j+s+1}$. Then $X \to \Phi$ is a legal type **II** move. Indeed, $x_{j-1} > A^n_{j-2} \geq n$, so

$$x_{k-1} < A^n_{j+s+1} = 2A^n_{j+s} + \sum_{i=0}^{j+s-1} A^n_i \leq 2x_{k-2} + \sum_{i=0}^{k-3} x_i.$$

We have shown that $W = \mathcal{P}$. ■

4. Does Frankenstein have a Polynomial Strategy?

The *statement* of Theorem 3 enables one to decide whether any given position X of the form (3) of Frankenstein is a P-position or an N-position, and the *proof* clearly indicates a winning move from any N-position. These two things together constitute a winning strategy for the game.

Given any position X of the form (3) of Frankenstein. To decide whether $X \in \mathcal{P}$ or $X \in \mathcal{N}$, we have to compute the entries of $\mathcal{A}$ only up to the first encounter of x_0. Thus it is readily seen that Theorem 2(ii) implies that A^n_j has to be computed only for $n \leq x_0(\phi - 1)$; and (5) implies that $j < \frac{1}{2}\log_\phi(\sqrt{5}(x_0 + 1)) - 1$. So the array has to be computed only up to $\Theta(x_0)$, which implies a strategy computation linear in x_0, which looks good.

However, the input size for Frankenstein is $\Theta(\sum_{i=0}^{k-1} \log x_i)$. So unless either k or x_{k-1} are exponentially larger than x_0, the indicated strategy is actually exponential. But only the construction of the table needs exponential time and, in fact, exponential space. The rest of the algorithm embodied in the proof of Theorem 3 is polynomial. A winning strategy is polynomial only if both of its parts are polynomial.

It follows from [Fra1985] that the computation of the representation of a positive integer N over the numeration system $\mathcal{U}$ can be done by a greedy Euclidean algorithm, namely always dividing the remainder r (initially: $r = N$), by the largest basis element $u_n \leq r$. This is a polynomial process. In particular, expressing a game position X of the form (3) over $\mathcal{U}$ can be done in polynomial time. It can then be observed in linear time whether or not x_0 is reduced, and all the other steps of the winning algorithm indicated in the proof of Theorem 3 can also be done in polynomial time. Thus the numeration system $\mathcal{U}$ actually enables us to formulate a polynomial strategy for Frankenstein — not only to decide whether it has or doesn't have one.

The game Frankenstein proposed here belongs to the family of *succinct* games, i.e., their input size is logarithmic. Normally an extra effort is required for showing that such games have a polynomial strategy. Different families of succinct games seem to require different methods of strategy computations.

For example, in *octal* games, invented by Guy and Smith [GuSm1956], a linearly ordered string of beads may be split and or reduced according to rules encoded in octal. See also [BCG1982, Ch. 4], [Con1976, Ch. 11]. The standard method for showing that an octal game is polynomial, is to demonstrate that its *Sprague-Grundy* function (the 0s of which constitute the set of P-positions) is periodic.

Periodicity has been established for a number of octal games. Some of the periods and or preperiods may be very large; see [GaPl1989]. Another way to establish polynomiality is to show that the Sprague-Grundy function values obey some other simple rule, such as forming an arithmetic sequence, as for Nim.

For the present class of pebbling games, polynomiality was established by a non-standard method. An arithmetic procedure, based on a class of special numeration systems, was the key to polynomiality. In [Fra1998] a game was proposed and analysed, and another numeration system was used there to establish polynomiality. For Wythoff's game [Wyt1907], [Cox1953], [YaYa1967], the Zeckendorf numeration system [Zec1972] can be used to establish polynomiality. But for Wythoff's game, this can be done also using the integer value function. From Theorem 2(iv) it follows that this cannot be done for Frankenstein. In [Fra1998] it was also proved that the integer value function cannot be used to establish polynomiality for the game defined there. But the question remains whether there or here, there is some *polynomial* algorithm not based on numeration systems.

5. Epilogue

We recap the main properties of the array $\mathcal{A}$.

(a)

$$A_0^n = \operatorname{mex}\{A_j^i : 0 \le i < n,\ j \ge 0\}\ (n \ge 0),$$

$$A_1^n = 2A_0^n + n\ (n \ge 0), \quad A_j^n = 3A_{j-1}^n - A_{j-2}^n\ \ (j \ge 2,\ n \ge 0). \quad \text{(The definition.)}$$

(b) For all $j, n \in \mathbb{Z}^+$, $A_j^n = 2A_{j-1}^n + A_{j-2}^n + \cdots + A_0^n + n$. (Lemma 1.)

(c) $\mathcal{A}$ is a splitting of $\mathbb{Z}^+$: every positive integer appears precisely once in $\mathcal{A}$. Moreover, for every $j \ge 0$, the column A_j consists precisely of all positive integers whose representation ends in j 0s. In particular, A_0^n is reduced for all $n \in \mathbb{Z}^+$. (Theorem 1.)

(d) (i) For $j, n \in \mathbb{Z}^+$, $A_j^n = \lfloor A_{j-1}^n \phi^2 \rfloor + 1 = L(A_{j-1}^n)$. (ii) For all $n \ge 1$, $A_0^n = \lfloor (n-1)\phi \rfloor + 1$ is reduced. (iii) For all $n \ge 0$, all $j \ge 0$ we have, $A_j^{n+1} - A_j^n \in \{f_{2j}, f_{2j+1}\}$, and for fixed j, each of f_{2j}, f_{2j+1} is assumed for infinitely many n. Moreover, for all n for which $A_0^{n+1} - A_0^n = f_0$ (respectively f_1), we also have for all j, $A_j^{n+1} - A_j^n = f_{2j}$ (respectively f_{2j+1}). (iv) Let $j \ge 1$. There are no real numbers α, γ, such that for all $n \ge 1$, $A_j^n = \lfloor n\alpha + \gamma \rfloor$.

The numeration system $\mathcal{U}$ was used both for proving the most important of these properties, and for deciding the polynomiality question of the strategy of Frankenstein.

The reason our title contains the term "arrays", whereas we have presented only a single array, is that we allude to an infinite family of arrays, based on some linear recurrence of the form

$$u_0 = 1, \quad u_n = b_1 u_{n-1} + \cdots + b_m u_m, \tag{10}$$

where the b_i are constants, except that $b_1 = b_1(n)$ may depend on n, with given initial integer values $u_{-m+1}, \ldots, u_{-1}$. If

$$1 \le b_m \le \cdots \le b_1, \tag{11}$$

then there is also an associated numeration system [Fra1985]. Replacing in (10) the elements u_j by columns A_j the recurrence is used to construct $\mathcal{A}$ (with possibly a special construction for the first initial values of j).

A first — to my knowledge — "Fibonacci array" has been defined in [Sto1977]. Other "Stolarsky arrays" were defined in papers such as [Kim1995] and [FrKi1994], and there are infinitely many such arrays. But we have not seen any applications of these arrays. Perhaps the present use for a winning strategy to a new class of games is the first application? Is there a natural infinite family of combinatorial games, matching the infinite family of arrays? And what's the nature of these arrays and their uses if (11) is violated?

It seems that the array defined here was not given before. Its antidiagonal hasn't appeared in [Slo1998] until we sent it in there recently; and its columns A_j and its rows A^n do not seem to appear in it for $j > 1$ and $n > 3$. As we remarked just prior to the statement of Theorem 2, the rows of the present array are "even Fibonacci numbers".

Several comments can be made about recurrences such as (2). We shall briefly relate to two items.

(I) The second recurrence of (2) can be considered to be the recurrence of the convergents of the *quasiregular* (or *semiregular*— halbregelmässig) continued fraction

$$3 + \cfrac{-1}{3 + \cfrac{-1}{3 + \cfrac{-1}{\ddots}}}$$

In [Per1950, Ch. 5] it is shown that every quasiregular continued fraction converges. In the present case it converges to ϕ^2. Many of the above properties of $\mathcal{A}$ can be deduced from this observation; also other properties not mentioned above, such as $u_n^2 - u_{n-1}u_{n+1} = 1$ for all n, and somewhat more complicated identities for elements in the other rows of $\mathcal{A}$.

(II) In [BBDD1998], the authors quote [BSS1993]: "...the recurrence $f_{n+1} = 6f_n - f_{n-1}$ cries out for a combinatorial interpretation. Finding this interpretation is an open problem." [BBDD1998] gives such an interpretation. We remark that in [Fra1985, §4] a class of regular (simple) continued fractions is defined whose convergents satisfy recurrences including the above. In particular, the numerators of the even-indexed convergents of the simple continued fraction

$$\sqrt{2} = [1, 2, 2, \ldots] = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{\ddots}}}$$

constitute the sequence $1, 7, 41, 239, \ldots$ with initial values $f_1 = 1$, $f_2 = 7$ considered in [BBDD1998]. Needless to say that each such recurrence also defines an exotic numeration system. Perhaps these facts constitute a "combinatorial interpretation".

The game Frankenstein is superficially reminiscent of the game of *Welter*, analyzed in [Con1976, Ch. 13]. The terminology "spinster" was introduced there. Welter is played on a semi-infinite strip with a finite number of coins, at most one per square, and the squares are numbered with the nonnegative integers $0, 1, 2, \ldots$ from the left end of the strip. A move consists of selecting a single coin and shifting it to an unoccupied square with lower number. The player first unable to move loses, and the opponent wins. The winning strategy is intricate. Moreover, it seems very difficult to generalize Welter. The game proposed here is not a generalization of Welter, but the moves are reminiscent of several moves of Welter taken simultaneously.

Acknowledgment

At the FUN conference I presented a paper entitled "Heap games and numeration systems". I sent it for publication two weeks before the conference, since at that time there were no plans for a special conference issue, to the best of my knowledge. It appeared in expanded form, with a modified title, in [Fra1998]. The present paper, though new in content, is nevertheless close in spirit to the earlier one. I thank the editors for considering it for the special issue.

References

1. [BBDD1998] E. Barcucci, S. Brunetti, A. Del Lungo and F. Del Ristoro [1998], A combinatorial interpretation of the recurrence $f_{n+1} = 6f_n - f_{n-1}$, *Discrete Math.* **190**, 235–240.

2. [BCG1982] E.R. Berlekamp, J.H. Conway and R.K. Guy [1982], *Winning Ways* (two volumes), Academic Press, London.

3. [BSS1993] J. Bonin, L. Shapiro and R. Simion [1993], Some q-analogues of the Schröder numbers arising from combinatorial statistics on lattice paths, *J. Statist. Plann. Inference* **34**, 35–55.

4. [Con1976] J.H. Conway [1976], *On Numbers and Games*, Academic Press, London.

5. [Cox1953] H.S.M. Coxeter [1953], The golden section, phyllotaxis and Wythoff's game, *Scripta Math.* **19**, 135–143.

6. [Fra1969] A.S. Fraenkel [1969], The bracket function and complementary sets of integers, *Canadian J. Math.* **21**, 6–27.

7. [Fra1985] A.S. Fraenkel [1985], Systems of numeration, *Amer. Math. Monthly* **92**, 105–114.

8. [Fra1998] A.S. Fraenkel [1998], Heap games, numeration systems and sequences, *Ann. of Combinatorics* **2**, 197–210.

9. [Fra$\geq$2001] A.S. Fraenkel [$\geq$2001], *Adventures in Games and Computational Complexity*, to appear in *Graduate Studies in Mathematics*, Amer. Math. Soc., Providence, RI.

10. [FrKi1994] A.S. Fraenkel and C. Kimberling [1994], Generalized Wythoff arrays, shuffles and interspersions, *Discrete Mathematics* **126**, 137–149.

11. [GaPl1989] A. Gangolli and T. Plambeck [1989], A note on periodicity in some octal games, *Internat. J. Game Theory* **18**, 311–320.

12. [GuSm1956] R.K. Guy and C.A.B. Smith [1956], The G-values of various games, *Proc. Camb. Phil. Soc.* **52**, 514–526.

13. [HaWr1989] G.H. Hardy [1989], *An Introduction to the Theory of Numbers*, 5th ed., Clarendon Press, Oxford.

14. [Kim1995] C. Kimberling [1995], Stolarsky interspersions, *Ars Combinatoria* **39**, 129–138.

15. [Per1950] O. Perron [1950], *Die Lehre von den Kettenbrüchen*, Chelsea, New York.

16. [Slo1998] N.J.A. Sloane [1998], Sloane's On-Line Encyclopedia of Integer Sequences, http://www.research.att.com/~njas/sequences/ .

17. [Sto1977] K.B. Stolarsky [1977], A set of generalized Fibonacci sequences such that each natural number belongs to exactly one, *Fibonacci Quart.* **15**, 224.

18. [Wyt1907] W.A. Wythoff [1907], A modification of the game of Nim, *Nieuw Arch. Wisk.* **7**, 199–202.

19. [YaYa1967] A.M. Yaglom and I.M. Yaglom [1967], *Challenging Mathematical Problems with Elementary Solutions*, translated by J. McCawley, Jr., revised and edited by B. Gordon, Vol. II, Holden-Day, San Francisco.

20. [Zec1972] E. Zeckendorf [1972], Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas, *Bull. Soc. Roy. Sci. Liège* **41**, 179–182.

The algebra of an age

Peter J. Cameron

Abstract

Associated with any oligomorphic permutation group G, there is a graded algebra $\mathcal{A}^G$ such that the dimension of its nth homogeneous component is equal to the number of G-orbits on n-sets. I show that the algebra is a polynomial algebra (free commutative associative algebra) in some cases, and pose some questions about transitive extensions.

1 The algebra

Let Ω be an infinite set. Let $\binom{\Omega}{n}$ denote the set of n-element subsets of Ω, V_n the vector space of functions from $\binom{\Omega}{n}$ to $\mathbb{Q}$. Set $\mathcal{A} = \bigoplus_{n\geq 0} V_n$, with multiplication defined as follows: for $f \in V_n$, $g \in V_m$, and $X \in \binom{\Omega}{n+m}$,

$$(fg)(X) = \sum_{Y \in \binom{X}{n}} f(Y)g(X \setminus Y).$$

This is the *reduced incidence algebra* of the poset of finite subsets of Ω (Rota [13]). It is a commutative and associative algebra with identity, but is far from an integral domain: any function with finite support is nilpotent.

Now, if G is any permutation group on Ω, let $\mathcal{A}^G = \bigoplus_{n\geq 0} V_n^G$, where V_n^G consists of the functions in V_n which are G-invariant (where G acts on V_n in the natural way: $f^g(X) = f(Xg^{-1})$). Now a function in V_n is fixed by G if and only if it is constant on the G-orbits. So, if G is *oligomorphic* (that is, G has only finitely many orbits on n-sets for all n), then $\dim(V_n^G) = f_n(G)$ is the number of orbits of G on $\binom{\Omega}{n}$.

If G has a finite orbit, then $\mathcal{A}^G$ contains non-zero nilpotents. I *conjecture* that conversely, if G has no finite orbits, then $\mathcal{A}^G$ is an integral domain. This question arose originally in studying the rate of growth of the numbers $f_n(G)$ for oligomorphic groups. The only evidence for it, apart from the fact that no counterexamples are known, is the following observation. Let $f \in V_n$ and $g \in V_m$ be such that $fg \neq 0$. Let X and Y be sets in the support of f and g respectively. By the Separation Lemma (Neumann [10], Lemma 2.3), if G has no finite orbits, then there is a translate Y' of Y such that $X \cap Y' = \emptyset$. Now we have a non-zero contribution to $(fg)(X \cup Y')$, though this may be cancelled out by other terms in the sum.

There is a stronger form of the conjecture, as follows. Let e be the constant function in V_1 with value 1. It is known that e is a non-zero-divisor in $\mathcal{A}$, and lies

in $\mathcal{A}^G$ for any group G. (This implies that multiplication by e is a monomorphism from V_n^G to V_{n+1}^G, and hence that $f_{n+1}(G) \geq f_n(G)$ for any n: see Cameron [1].) I *conjecture* that, if G has no finite orbits, then e is prime in $\mathcal{A}^G$, in the sense that if $e|fg$ then $e|f$ or $e|g$. This would imply that $\mathcal{A}^G$ is an integral domain.

There is a combinatorial version of this algebra, defined as follows. Let $\mathcal{C}$ be a class of finite relational structures closed under isomorphism and under taking induced substructures. Let $V_n(\mathcal{C})$ be the vector space of functions from the isomorphism types of n-element structures in $\mathcal{C}$ to $\mathbb{Q}$, and $\mathcal{A}(\mathcal{C}) = \bigoplus_{n\geq 0} V_n(\mathcal{C})$, with multiplication defined just as before.

The *age* of a relational structure M on Ω is the class of all finite structures embeddable in M as induced substructures. M is *homogeneous* if every isomorphism between finite induced substructures of M extends to an automorphism of M. Now we have:

- If $\mathcal{C}$ is the age of a relational structure M on Ω, then $\mathcal{A}(\mathcal{C})$ is a subalgebra of the reduced incidence algebra $\mathcal{A}$ on Ω (and this is equivalent to $\mathcal{C}$ having the *joint embedding property*, that is, any two members of $\mathcal{C}$ can be simultaneously embedded in a member of $\mathcal{C}$).

- If $\mathcal{C}$ is the age of a homogeneous relational structure M on Ω, then $\mathcal{A}(\mathcal{C}) = \mathcal{A}^G$, where $G = \mathrm{Aut}(M)$ (and this is equivalent to $\mathcal{C}$ having the *amalgamation property*, that is, any amalgam of two members of $\mathcal{C}$ with a common substructure can be embedded in a member of $\mathcal{C}$).

See, for example, Cameron [3] for discussion.

2 Polynomial algebras

There are only two techniques I know for determining the structure of the algebras $\mathcal{A}^G$ or $\mathcal{A}(\mathcal{C})$. The first is based on the simple observation that, regarding $G \times H$ as a permutation group on the disjoint union of the sets on which G and H act, we have

$$\mathcal{A}^{G\times H} = \mathcal{A}^G \otimes_{\mathbb{Q}} \mathcal{A}^H.$$

Let S denote the symmetric group on an infinite set. Then $\mathcal{A}^S$ is a polynomial ring in one variable (generated by the element e). Hence $\mathcal{A}^{S^n}$ is a polynomial algebra in n variables.

Now let H be a finite permutation group of degree n. Then the *wreath product* $S\mathrm{Wr}H$ is the semidirect product of S^n by H, and so $\mathcal{A}^{S\mathrm{Wr}H}$ consists of the invariants of H in the polynomial algebra (in the classical sense, where H acts as a linear group by permutation matrices). For example, if H is the symmetric group S_n, then $\mathcal{A}^{S\mathrm{Wr}S_n}$ is the polynomial algebra generated by the n elementary symmetric functions, by Newton's Theorem. (Note that $\mathcal{A}^{S\mathrm{Wr}H}$ is always an integral domain, but almost never a polynomial algebra.)

In this case, the numbers $f_n(S \operatorname{Wr} H)$ can be calculated by Molien's Theorem, which turns out to be a special case of a "cycle index theory" for oligomorphic permutation groups (see [3]).

The second approach requires that the class $\mathcal{C}$ has a "good notion of connectedness", as follows. I will give an axiomatic treatment, since in one of the examples below, words like "connected" and "involvement" have meanings quite different from their usual ones. We require

- a distinguished subclass of $\mathcal{C}$ consisting of "connected" structures;
- a partial order $\leq$ called "involvement" on the class of n-element structures for each n;
- a binary, commutative and associative "composition" $\circ$ such that, if X and Y are structures with n and m points respectively, then $X \circ Y$ is a structure with $n+m$ points.

Assume that the following conditions hold:

A1 Let S be a structure which is partitioned into disjoint induced substructures $S_1, S_2, \ldots$. Then $S_1 \circ S_2 \circ \ldots \leq S$.

A2 Any structure has a unique representation as a composition of connected structures.

Theorem 2.1 *If all the above conditions hold, then $\mathcal{A}(\mathcal{C})$ is a polynomial algebra, generated by the characteristic functions of the connected structures.*

Proof. If $|S| = n$, then S is a disjoint union $S_1 \cup S_2 \cup \ldots$ of connected structures; so we have a bijection between characteristic functions χ_S (the basis elements of $V_n(\mathcal{C})$) and monomials $\phi_S = \chi_{S_1}\chi_{S_2}\ldots$ of total weight n. Consider the matrix expressing the monomials ϕ_S in terms of the basis elements $\chi_{S'}$. The coefficient of χ_S in the row corresponding to ϕ_S is non-zero. Suppose that $\chi_{S'}$ also has non-zero coefficient. Then S' can be partitioned into induced substructures isomorphic to $S_1, S_2, \ldots$; so $S = S_1 \circ S_2 \circ \ldots \leq S'$. Thus the matrix is upper triangular with non-zero diagonal, and hence invertible. So the monomials of weight n form a basis for $V_n(\mathcal{C})$, and the theorem is proved.

Example 1. Let M be the countable "random graph" [4], whose age $\mathcal{C}$ is the class of all finite graphs. Let "connected" have its usual meaning, "involvement" mean "spanning subgraph", and "composition" be disjoint union (with no edges between the parts). Then A1 and A2 hold, and so $\mathcal{A}(\mathcal{C}) = \mathcal{A}^{\mathrm{Aut}(M)}$ is a polynomial algebra, whose generators correspond to the finite connected graphs.

This method works for many other ages, both of homogeneous structures (for example, the class of K_n-free graphs for fixed n [8]), and not (for example, bipartite graphs, N-free graphs [5]).

Example 2. Let $\mathcal{C}$ be the age of a homogeneous structure M, and let $G = \mathrm{Aut}(M)$. Let $\mathcal{C}'$ be the class of structures over a language with the relation symbols for $\mathcal{C}$ and one new binary symbol E, in which E is an equivalence relation each of whose classes carries a $\mathcal{C}$-structure (with no instances of relations holding between points in different E-classes). Then $\mathcal{C}'$ is the age of a homogeneous structure consisting of the disjoint union of countably many copies of M, with automorphism group $G \,\mathrm{Wr}\, S$, where S is the symmetric group of countable degree. Now let "connected" mean "only one E-class", "involvement" mean "inclusion of all relations", and "composition" mean "disjoint union". Then A1 and A2 hold.

The conclusion is that $\mathcal{A}^{G\mathrm{Wr}S}$ is always a polynomial algebra; the number of generators of degree n is equal to the number of orbits of G on n-sets.

Example 3. Let A be a fixed alphabet of finite size q, and let $\mathcal{C} = A^*$ be the set of words in A. (Here a word of length n is regarded as an n-set carrying a total order and q unary relations $R_1, \ldots, R_q$, where each element of the set satisfies exactly one of the unary relations; the word $a_1 a_2 \ldots a_q$ corresponds to the n-set $\{x_1, \ldots, x_n\}$, with $x_1 < x_2 < \ldots < x_n$ and in which x_i satisfies R_{a_i}.) The algebra $\mathcal{A}(A^*)$ is the *shuffle algebra* which arises in the theory of free Lie algebras [12]. The name comes from the fact that the product of two words is the sum of all words which can be obtained by "shuffling" them together, with appropriate multiplicities. For example,

$$(aab) \cdot (ab) = abaab + 3aabab + 6aaabb.$$

Also, A^* is the age of a homogeneous relational structure $M(q)$ which is order-isomorphic to $\mathbb{Q}$ and in which the set of elements satisfying each relation R_i is dense; in other words, a partition of $\mathbb{Q}$ into q dense subsets. Such a partition is unique up to order-isomorphism of $\mathbb{Q}$. Let $G(q) = \mathrm{Aut}(M(q))$.

Take a total order on A, and define the *lexicographic order* on A^* in the usual way: that is, $a_1 \ldots a_m < b_1 \ldots b_n$ if and only if *either*

- $m < n$, and $a_i = b_i$ for $i = 1, \ldots, m$; *or*
- for some $l < \min\{m, n\}$, we have $a_i = b_i$ for $i = 1, \ldots, l$, and $a_{l+1} < b_{l+1}$.

A non-empty word $w \in A^*$ is a *Lyndon word* if, whenever $w = xy$ with x, y non-empty, we have $w < y$; that is, w is less than any proper cyclic shift of itself. The number of Lyndon words of length n is $(1/n)\sum_{d|n} \mu(d) q^{n/d}$, where μ is the Möbius function. (This well-known number counts several other things, for example, irreducible polynomials over $\mathbb{F}_q$ if q is a prime power; see [12].) The following combinatorial properties hold for Lyndon words:

Lemma 2.2 *(i) Any word w has a unique expression in the form $w = w_1 w_2 \ldots$, where $w_1, w_2, \ldots$ are Lyndon words with $w_1 \geq w_2 \geq \ldots$.*

(ii) Given Lyndon words $w_1, w_2, \ldots$ with $w_1 \geq w_2 \geq \ldots$, the lexicographically greatest shuffle of these words is the concatenation $w_1 w_2 \ldots$.

Hence, if we let "connected" mean "Lyndon word", "involvement" mean "lexicographic order reversed", and "composition" mean "concatenation in decreasing lexicographic order", then A1 and A2 hold, and we conclude that $\mathcal{A}(A^*) = \mathcal{A}^{G(q)}$ is a polynomial algebra generated by the Lyndon words (a result of Radford [11]).

3 Transitive extensions

Not much is known in general about how the algebra $\mathcal{A}^G$ is affected by group-theoretic or model-theoretic constructions (direct products with product action, wreath products, covers and quotients, etc.). This section contains some comments about transitive extensions.

The permutation group H on Ω is a *transitive extension* of G if H is transitive and the stabiliser H_α of the point α, acting on $\Omega \setminus \{\alpha\}$, is isomorphic to G as permutation group. Note that, in this situation, H is closed if and only if G is closed.

A general question: *Let H be a transitive extension of G. What is the relation between $\mathcal{A}^H$ and $\mathcal{A}^G$?*

We can regard the group induced on Ω by G as the direct product of G (in its given action) with the trivial group of degree 1. For the latter group (K, say), the algebra $\mathcal{A}^K$ is generated by an element k of degree 1 with $k^2 = 0$. In other words, $\mathcal{A}^K \cong T(\mathbb{Q})$, the algebra of 2×2 upper triangular matrices with constant diagonal over $\mathbb{Q}$. Hence, using G^+ for the group induced on Ω by G, we have

$$\mathcal{A}^{G^+} \cong \mathcal{A}^G \otimes_{\mathbb{Q}} T(\mathbb{Q}) \cong T(\mathcal{A}^G).$$

However, we can only say that, since $G^+ \leq H$, the algebra $\mathcal{A}^H$ is a subalgebra of $T(\mathcal{A}^G)$. This does not seem to help to decide, for example, whether $\mathcal{A}^H$ is an integral domain.

There is a special class of transitive extensions for which a bit more can be said. We say that the transitive extension H of G is *curious* if H has a transitive subgroup (on the whole of Ω) which is isomorphic to G. In the case where G and H are closed, this means that H is a reduct of G. If H is a curious transitive extension of G, then $\mathcal{A}^H$ is a subalgebra of $\mathcal{A}^G$; in particular, $\mathcal{A}^H$ is an integral domain if $\mathcal{A}^G$ is. Perhaps it is possible to weave together the embeddings of $\mathcal{A}^H$ in $\mathcal{A}^G$ and in $T(\mathcal{A}^G)$ to get better information.

Example 1 (continued). A *two-graph* on Ω is a set T of 3-element subsets of Ω such that any 4-subset contains an even number of members of T (Seidel [14]).

Given a graph Γ on Ω, let $T(\Gamma)$ be the set of *odd triples* of Γ (those containing an odd number of edges). Then $T(\Gamma)$ is a two-graph on Ω. Every two-graph arises in this way.

Let R be the random graph on Ω_0. Take a new point ∞, and define T to be the two-graph on $\Omega = \Omega_0 \cup \{\infty\}$ derived from R (with ∞ as an isolated vertex). Then $\mathrm{Aut}(T)$ is a transitive extension of $\mathrm{Aut}(R)$. Moreover, it is curious; for the two-graph derived from R without an isolated vertex is clearly a reduct of R, and

is isomorphic to T. (In fact, T is the unique countable universal homogeneous two-graph.) See Thomas [16].

Problem. Is $\mathcal{A}^{\mathrm{Aut}(T)}$ a polynomial algebra?

Remark. Mallows and Sloane [9] showed that the numbers of two-graphs and *even graphs* (graphs with all valencies even) on n points are equal. Hence, if $\mathcal{A}^{\mathrm{Aut}(T)}$ is a polynomial algebra, then its generators are in one-to-one correspondence (preserving degree) with the finite *Eulerian graphs* (the connected even graphs).

Example 3 (continued). Let $G(q)$ be as in Example 3 in the preceding section. Then $G(q)$ has a transitive extension $H(q)$ defined as follows.

On the set of complex roots of unity, put $z_1 \equiv z_2$ if $z_2 z_1^{-1}$ is a qth root of unity. Let Ω be a dense subset containing exactly one member of each equivalence class of this relation. (Such a set is unique up to permutation preserving the cyclic order. If we choose a random member of each class, the resulting set almost surely has this property.) Now define binary relations $R_1, R_2, \ldots, R_q$ by $(z_1, z_2) \in R_j$ if and only if

$$\frac{2\pi(j-1)}{q} < \arg(z_2 z_1^{-1}) < \frac{2\pi j}{q}.$$

The structure $N(q)$ consists of the circular order and the relations $R_1, R_2, \ldots, R_q$. It is $\aleph_0$-categorical. Note that, if $z_1 \neq z_2$, then $(z_1, z_2) \in R_j$ for a unique value of j; and the converse of R_j is R_{q+1-j}. Let $H(q) = \mathrm{Aut}(N(q))$.

Now take $z \in \Omega$. Define a map $\phi : \Omega \setminus \{z\} \to (0,1)$ by letting $\phi(w)$ be the fractional part of $\frac{q}{2\pi}\arg(zw^{-1})$. Then $\phi(\Omega \setminus \{z\}) = (0,1) \cap \mathbb{Q}$. If we give $\phi(w)$ the colour j if $(z,w) \in R_j$, then each colour class is dense. Moreover, the structure $N(q)$ can be recovered uniquely from this information. So $H(q)$ is a transitive extension of $G(q)$.

This extension is also curious. If we repeat the above construction, but with z a point on the unit circle which is not a root of unity, we obtain a bijection from all of Ω to a countable dense subset of $(0,1)$ partitioned into q dense subsets.

Problem. Is $\mathcal{A}^{H(q)}$ a polynomial algebra?

Remark. For $q = 2$, the relations R_1 and R_2 are a converse pair of tournaments, each of which is isomorphic to the countable universal homogeneous *local order* [2], *locally transitive tournament* [7], or *vortex-free tournament* [6]: these are three alternative names for a tournament having no subtournament consisting of a directed 3-cycle dominating or dominated by a vertex. This structure is further discussed in the lectures of Evans, Ivanov and Macpherson.

Orbits of $H(q)$ on n-sets are parametrised by two-way infinite "shift register sequences" (x_i) with elements in $\{1, \ldots, q\}$ satisfying $x_i + n \equiv x_i + 1 \pmod{q}$ for all i. For $q = 2$, the sequences counting these orbits is listed as M0324 in the *Encyclopedia of Integer Sequences* [15], where further references can be found.

On the assumption that $\mathcal{A}^{H(2)}$ is a polynomial algebra, it is possible to compute the numbers of generators of each degree. The resulting sequence appears to be "unknown"; in particular, it is not in the Encyclopedia [15].

The group $H(2)$ does not have a transitive extension. Nevertheless, the following occurrence is suggestive.

Knuth [6] defines a *CC-structure* to be a set with a ternary relation satisfying five universal axioms, of which the first three assert that the induced structure on any 3-set is a circular order. The letters CC stand for "counter-clockwise"; and, given a set Ω of points in the Euclidean plane with no three collinear, the relation R such that $R\alpha\beta\gamma$ holds if and only if the points α, β, γ occur in the counter-clockwise sense, is a CC-structure. Such a CC-structure is called *representable*. There is a countable universal representable CC-structure, defined by choosing a countable dense set of points in the Euclidean plane with no three collinear. It is not homogeneous; indeed, the class of CC-structures (or of representable CC-structures) does not have the amalgamation property.

Given a ternary relation R on Ω whose restriction to any 3-set is a circular order, there is a derived tournament R_α on $\Omega \setminus \{\alpha\}$ defined by $R_\alpha\beta\gamma \Leftrightarrow R\alpha\beta\gamma$. Knuth's fifth axiom for CC-structures implies that R_α is a local order for any point α. Indeed, if we take the universal representable CC-structure above, and project $\Omega \setminus \{\alpha\}$ radially onto the unit circle with centre α, we obtain the homogeneous local order $N(2)$.

Problem. Do there exist countable CC-structures (or representable ones) with large automorphism groups, or with other nice model-theoretic properties?

Acknowledgment. I am grateful to R. A. Bailey, R. M. Bryant and D. G. Fon-Der-Flaass for their help with the contents of this paper.

References

[1] P. J. Cameron, Transitivity of permutation groups on unordered sets, *Math. Z.* **48** (1976), 127–139.

[2] P. J. Cameron, Orbits of permutation groups on unordered sets, II, *J. London Math. Soc.* (2) **23** (1981), 249–265.

[3] P. J. Cameron, *Oligomorphic Permutation Groups*, London Math. Soc. Lecture Notes **152**, Cambridge University Press, Cambridge, 1990.

[4] P. J. Cameron, The random graph, pp. 333–351 in *The Mathematics of Paul Erdős, II* (ed. R. L. Graham and J. Nešetřil), Algorithms and Combinatorics **14**, Springer, Berlin, 1997.

[5] J. Covington, A universal structure for N-free graphs, *Proc. London Math. Soc.* (3), **58** (1989), 1–16.

[6] D. E. Knuth, *Axioms and Hulls*, Lecture Notes in Computer Science **606**, Springer, Berlin, 1992.

[7] A. H. Lachlan, Countable homogeneous tournaments, *Trans. Amer. Math. Soc.* **284**, 431–461.

[8] A. H. Lachlan and R. E. Woodrow, Countable ultrahomogeneous undirected graphs, *Trans. Amer. Math. Soc.* **262** (1980), 51–94.

[9] C. L. Mallows and N. J. A. Sloane, Two-graphs, switching classes, and Euler graphs are equal in number, *SIAM J. Appl. Math.* **28** (1975), 876–880.

[10] P. M. Neumann, The lawlessness of finitary permutation groups, *Arch. Math.* **26** (1975), 561–566.

[11] D. E. Radford, A natural ring basis for the shuffle algebra and an application to group schemes, *J. Algebra* **58** (1979), 432–454.

[12] C. Reutenauer, *Free Lie Algebras*, London Math. Soc. Monographs (New Series) **7**, Oxford University Press, 1993.

[13] G.-C. Rota, On the foundations of combinatorial theory, I: Theory of Möbius functions, *Z. Wahrscheinlichkeitstheorie* **2** (1964), 340–368.

[14] J. J. Seidel, A survey of two-graphs, pp. 481–511 in *Proc. Int. Colloq. Teorie Combinatorie*, Accad. Naz. Lincei, Roma, 1977.

[15] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, New York, 1995.

[16] S. R. Thomas, Reducts of the random graph, *J. Symbolic Logic* **56** (1991) 176–181.

Author's address:
School of Mathematical Sciences,
Queen Mary and Westfield College,
London E1 4NS,
England.
e-mail: P.J.Cameron@qmw.ac.uk

Aesthetics for the Working Mathematician

Jonathan M. Borwein, FRSC

Prepared for
Queens University Symposium
on
Beauty and the Mathematical Beast:
Mathematics and Aesthetics
April 18, 2001

Shrum Professor of Science & Director

CECM
Centre for Experimental & Constructive Mathematics

Simon Fraser University, Burnaby, BC Canada

URL: www.cecm.sfu.ca/~jborwein/talks.html

Revised: September 7, 2001

BLAKE

- Songs of Innocence and Experience (1825)

ABSTRACT.

"If my teachers had begun by telling me that mathematics was pure play with presuppositions, and wholly in the air, I might have become a good mathematician. But they were overworked drudges, and I was largely inattentive, and inclined lazily to attribute to incapacity in myself or to a literary temperament that dullness which perhaps was due simply to lack of initiation." (George Santayana)

"Persons and Places", 1945, pp. 238-9

- Most research mathematicians neither think deeply about nor are terribly concerned about either pedagogy or the philosophy of mathematics. Nonetheless, as I hope to indicate, aesthetic notions have always permeated (pure and applied) mathematics.

- I shall argue for aesthetics before utility.

- Through examples, I aim to illustrate how and what that means at the research mine face. I also will argue that the opportunities to tie research and teaching to aesthetics are almost boundless — at all levels of the curriculum. This is in part due to the increasing power and sophistication of visualization, geometry, algebra and other mathematical software.

"The mathematician does not study pure mathematics because it is useful; he studies it because he delights in it and he delights in it because it is beautiful." (Henri Poincaré)

- The transparencies, and other resources, for this presentation are available at

`www.cecm.sfu.ca/personal/jborwein/talks.html`
`www.cecm.sfu.ca/personal/jborwein/mathcamp00.html`
and
`www.cecm.sfu.ca/ loki/Papers/Numbers/`

GAUSS

Gauss once confessed,

> *"I have the result, but I do not yet know how to get it."*

("Asimov's Book of ... Quotations," p. 115)

• One of Gauss's greatest discoveries, in 1799, was the relationship between the lemniscate sine function and the arithmetic-geometric mean iteration. This was based on a purely computational observation. The young Gauss wrote in his diary that the result

> *"will surely open up a whole new field of analysis."*

◇ He was right, as it pried open the whole vista of nineteenth century elliptic and modular function theory.

• Gauss's specific discovery, based on tables of integrals provided by Stirling (1692-1770), was that the reciprocal of the integral

$$\frac{2}{\pi}\int_0^1 \frac{dt}{\sqrt{1-t^4}}$$

agreed numerically with the limit of the rapidly convergent iteration given by $a_0 := 1$, $b_0 := \sqrt{2}$ and computing

$$a_{n+1} := \frac{a_n + b_n}{2}, \qquad b_{n+1} := \sqrt{a_n b_n}$$

◇ The sequences a_n, b_n have a common limit
1.1981402347355922074....

• Which is familiar, which is elegant — then and now?

◇ Aesthetic criteria change: 'closed forms' versus 'recursion'. 'Biology envy' replaces 'the blind watchmaker'.

GAUSS and HADAMARD

> *The object of mathematical rigor is to sanction and legitimize the conquests of intuition, and there was never any other object for it.* (J. Hadamard, 1865-1963)

In Borel, "Lecons sur la theorie des fonctions," 1928.

• Perhaps the greatest mathematician to think deeply and seriously about cognition in mathematics (*"... in arithmetic, until the seventh grade, I was last or nearly last."*).

◇ Author of "The psychology of invention in the mathematical field" (1945) and co-prover of the Prime Number Theorem (1896):

> *"The number of primes less than n tends to ∞ as does $\frac{n}{\log n}$."*

AESTHETIC(s) in WEBSTER

aesthetic, adj 1. pertaining to a sense of the beautiful or to the science of aesthetics.

2. having a sense of the beautiful; characterized by a love of beauty.

3. pertaining to, involving, or concerned with pure emotion and sensation as opposed to pure intellectuality.

4. a philosophical *theory or idea of what is aesthetically valid at a given time and place*: the clean lines, bare surfaces, and sense of space that bespeak the machine-age aesthetic.

5. aesthetics.

6. Archaic. the study of the nature of sensation.

Also, esthetic. Syn 2. discriminating, cultivated, refined.

aesthetics, noun 1. the branch of philosophy dealing with such notions as the beautiful, the ugly, the sublime, the comic, etc., as applicable to the fine arts, with a view to establishing the meaning and validity of critical judgments concerning works of art, and the principles underlying or justifying such judgments.

2. *the study of the mind and emotions in relation to the sense of beauty.*

• **JMB**: (unexpected) simplicity or organization in apparent complexity or chaos.

† We need to integrate this into mathematics education — to capture minds not only for utilitarian reasons. Detachment is important, — curtains, stages and picture frames.

RESEARCH MOTIVATIONS

INSIGHT – demands speed ≡ parallelism

- For rapid verification.
- For validation; proofs *and* refutations. For 'monster barring'.

† What is 'easy' changes — merging disciplines, levels and collaborators.

- Marry theory & practice, history & philosophy, proofs & experiments.
- Match elegance and balance to utility and economy.
- In analysis, algebra, geometry & topology.

AND GOALS

- Towards an Experimental M*a*thodology — philosophy and practice.
- Intuition is acquired — mesh computation and mathematics.
- Visualization — three is a lot of dimensions (pictures and sounds).
- 'Caging' and 'Monster-barring' (Lakatos).
  - graphic checks: compare $2\sqrt{y} - y$ and $\sqrt{y}\ln(y)$, $0 < y < 1$
  - randomized checks: equations, linear algebra, primality

PART of OUR 'METHODOLOGY'

1. (*High Precision*) computation of object(s).
2. *Pattern Recognition* of *Real Numbers* (Inverse Calculator and 'RevEng')*, or *Sequences* (Salvy & Zimmermann's 'gfun', Sloane and Plouffe's Encyclopedia).
3. Extensive use of 'Integer Relation Methods': *PSLQ* & *LLL* and FFT.†
   - Exclusion bounds are especially useful.
   - Great test bed for "Experimental Math".
4. Some automated theorem proving (Wilf-Zeilberger etc).

*ISC space limits: from 10Mb in 1985 to 10Gb today.

†Top Ten "Algorithm's for the Ages," Random Samples, Science, Feb. 4, 2000.

FOUR EXPERIMENTS

• 1. **Kantian** example: generating "the classical non-Euclidean geometries (hyperbolic, elliptic) by replacing Euclid's axiom of parallels (or something equivalent to it) with alternative forms."

• 2. The **Baconian** experiment is a contrived as opposed to a natural happening, it "is the consequence of 'trying things out' or even of merely messing about."

• 3. **Aristotelian** demonstrations: "apply electrodes to a frog's sciatic nerve, and lo, the leg kicks; always precede the presentation of the dog's dinner with the ringing of a bell, and lo, the bell alone will soon make the dog dribble."

• 4. The most important is **Galilean**: "a critical experiment – one that discriminates between possibilities and, in doing so, either gives us confidence in the view we are taking or makes us think it in need of correction."

◇ It is also the only one of the four forms which will make Experimental Mathematics a serious enterprise.

• From Peter Medawar's *Advice to a Young Scientist*, Harper (1979).

MILNOR

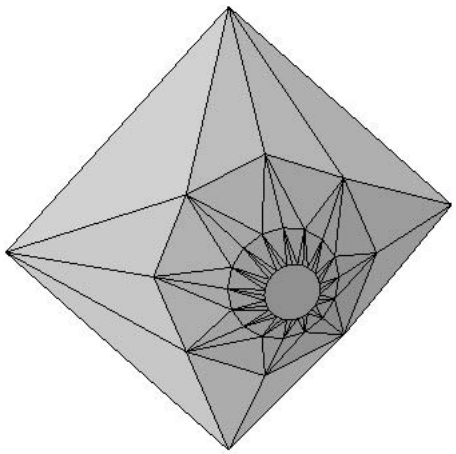
> *"If I can give an abstract proof of something, I'm reasonably happy. But if I can get a concrete, computational proof and actually produce numbers I'm much happier. I'm rather an addict of doing things on the computer, because that gives you an explicit criterion of what's going on. I have a visual way of thinking, and I'm happy if I can see a picture of what I'm working with."*

...

• Consider the following images of zeroes of 0/1 polynomials: `www.cecm.sfu.ca/interfaces/`

◇ But symbols are often more reliable than pictures.

A MISLEADING PICTURE



• LetsDoMath : `www.mathresources.com`

◇ Challenging students honestly? (Life)
◇ Making things tangible (Platonic solids)

De MORGAN & SYLVESTER

"Considerable obstacles generally present themselves to the beginner, in studying the elements of Solid Geometry, from the practice which has hitherto uniformly prevailed in this country, of never submitting to the eye of the student, the figures on whose properties he is reasoning, but of drawing perspective representations of them upon a plane. ... I hope that I shall never be obliged to have recourse to a perspective drawing of any figure whose parts are not in the same plane."

(Augustus De Morgan, 1806-71, First LMS President.)

• Adrian Rice, "What Makes a Great Mathematics Teacher?" MAA Monthly, June 1999, p. 540.

SYLVESTER'S THEOREM

† JavaViewLib : `www.cecm.sfu.ca/interfaces/` is Polthier's modern version of Felix Klein's (1840-1928) models.
† A modern version of Euclid: Cinderella.de : `personal/jborwein/circle.html` & Sketchpad.

"The early study of Euclid made me a hater of geometry."

(James Joseph Sylvester, 1814-97, Second LMS President)

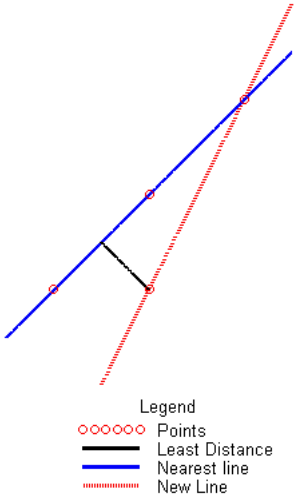
• In D. MacHale, "Comic Sections" (1993)

But discrete (now 'computational') geometry was different:

THEOREM. Given N non-collinear points in the plane there is a *proper* line through only two points.*

*Posed in *The Educational Times* **59** (1893).

KELLY'S "PROOF FROM 'THE BOOK' "



◇ It was forgotten for 50 years?

• First solved ("badly") by Gallai (1943). Also by Erdos who named 'the book'.

◇ Kelly's proof was published by Coxeter (1948)!

- Two more examples from the book:
  - Niven's 1947 proof that π is irrational (`personal/jborwein/pi.pdf`); and
  - Snell's law — travelling between Physics and the Calculus. (To or from?)

GEORGE LAKOFF & RAFAEL E. NUNEZ

"Recent Discoveries about the Nature of Mind.

In recent years, there have been revolutionary advances in cognitive science — advances that have a profound bearing on our understanding of mathematics.* Perhaps the most profound of these new insights are the following:

1. The embodiment of mind. The detailed nature of our bodies, our brains and our everyday functioning in the world structures human concepts and human reason. This includes mathematical concepts and mathematical reason.

*More serious curricular insights should come from neuro-biology (Dehaene et al., "Sources of Mathematical Thinking: Behavioral and Brain-Imaging Evidence," *Science*, May 7, 1999).

2. The cognitive unconscious. Most thought is unconscious — not repressed in the Freudian sense but simply inaccessible to direct conscious introspection. We cannot look directly at our conceptual systems and at our low-level thought processes. This includes most mathematical thought.

3. Metaphorical thought. For the most part, human beings conceptualize abstract concepts in concrete terms, using ideas and modes of reasoning grounded in sensory-motor systems. The mechanism by which the abstract is comprehended in terms of the concept is called *conceptual metaphor*. Mathematical thought also makes use of conceptual metaphor, as when we conceptualize numbers as points on a line."

- "Where Mathematics Comes From," Basic Books, 2000. (p. 5)

- They later observe:

"What is particularly ironic about this is that it follows from the empirical study of numbers as a product of mind that it is natural for people to believe that numbers are not a product of mind!" (Lakoff and Nunez, p. 81)

...

- Compare a more traditional view:

"The price of metaphor is eternal vigalence." (Arturo Rosenblueth and Norbert Wiener)

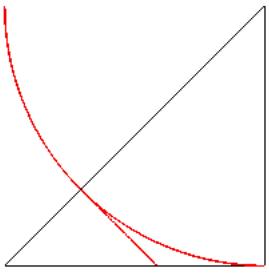
Quoted by R. C. Leowontin
in *Science* p. 1264, Feb 16, 2001
(The *Human Genome* Issue)

TWO THINGS ABOUT $\sqrt{2}$

- *A. Irrationality.*

- Tom Apostol's new geometric proof* of the irrationality of $\sqrt{2}$.

PROOF. Consider the *smallest* right-angled isoceles integral with integer sides:



◇ the smaller triangle is integral ···

*MAA, November 2000, pp. 241-242

• *B. Rationality.*

◇ $\sqrt{2}$ also makes things rational:

$$\begin{aligned}\left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} &= \\ \sqrt{2}^{(\sqrt{2}\cdot\sqrt{2})} &= \sqrt{2}^{\,2} = 2.\end{aligned}$$

• Hence there are irrational numbers α and β with α^β rational. But which ones?

† Compare: $\alpha := \sqrt{2}$, $\beta := 2\ln_2(3)$, which Maples says yields $\alpha^\beta = 3$.

† There are eight possible (ir)rational triples:

$$\alpha^\beta = \gamma.$$

TWO INTEGRALS

Even Maple knows

• A. $\pi \neq \frac{22}{7}$.

$$\int_0^1 \frac{(1-x)^4 x^4}{1+x^2}\,dx = \frac{22}{7} - \pi,$$

...

but struggles with

• *B. The sophomore's dream.*

$$\int_0^1 \frac{1}{x^x}\,dx = \sum_{n=1}^{\infty} \frac{1}{n^n}.$$

PARTIAL FRACTIONS & CONVEXITY

• We consider a network *objective function* p_N given by

$$p_N(\vec{q}) = \sum_{\sigma\in S_N} \left(\prod_{i=1}^{N} \frac{q_{\sigma(i)}}{\sum_{j=i}^{N} q_{\sigma(j)}}\right)\left(\sum_{i=1}^{N} \frac{1}{\sum_{j=i}^{N} q_{\sigma(j)}}\right)$$

summed over *all* $N!$ permutations; so a typical term is

$$\left(\prod_{i=1}^{N} \frac{q_i}{\sum_{j=i}^{N} q_j}\right)\left(\sum_{i=1}^{N} \frac{1}{\sum_{j=i}^{n} q_j}\right) .$$

◇ For $N = 3$ this is

$$q_1q_2q_3\left(\frac{1}{q_1+q_2+q_3}\right)\left(\frac{1}{q_2+q_3}\right)\left(\frac{1}{q_3}\right)$$
$$\times\left(\frac{1}{q_1+q_2+q_3} + \frac{1}{q_2+q_3} + \frac{1}{q_3}\right) .$$

• We wish to show p_N is *convex* on the positive orthant. First we try to simplify the expression for p_N.

• The *partial fraction decomposition* gives:

$$\begin{aligned} p_1(x_1) &= \frac{1}{x_1}, \\ p_2(x_1,x_2) &= \frac{1}{x_1} + \frac{1}{x_2} - \frac{1}{x_1+x_2}, \\ p_3(x_1,x_2,x_3) &= \frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} \\ &\quad - \frac{1}{x_1+x_2} - \frac{1}{x_2+x_3} - \frac{1}{x_1+x_3} \\ &\quad + \frac{1}{x_1+x_2+x_3}. \end{aligned}$$

So we predict the 'same' for $N = 4$ and are rewarded with:

CONJECTURE. For each $N \in \mathbb{N}$

$$p_N(x_1,\ldots,x_N) := \int_0^1 \left(1 - \prod_{i=1}^{N} (1-t^{x_i})\right)\frac{dt}{t}$$

is convex, indeed 1/concave.

• One can check $N < 5$ via a large symbolic Hessian computation. But not $N = 5$!

PROOF. A year later, analysis of *joint expectations* gave a convex integrand:

$$p_N(\vec{x}) = \int_{\mathbb{R}^n_+} e^{-(y_1+\cdots+y_n)} \max\left(\frac{y_1}{x_1}, \ldots, \frac{y_n}{x_n}\right) dy$$

◇ See *SIAM Electronic Problems and Solutions*.

• Computing adds reality, making concrete the abstract, and some hard things simple.

† Pascal's Triangle : `www.cecm.sfu.ca/interfaces/`

BERLINSKI

"The computer has in turn changed the very nature of mathematical experience, suggesting for the first time that mathematics, like physics, may yet become an empirical discipline, a place where things are discovered because they are seen."

...

"The body of mathematics to which the calculus gives rise embodies a certain swashbuckling style of thinking, at once bold and dramatic, given over to large intellectual gestures and indifferent, in large measure, to any very detailed description. But the era in thought that the calculus made possible is coming to an end. *Everyone feels this is so and everyone is right."*

π and FRIENDS

A: (*A quartic algorithm* (1984).) Set $a_0 = 6 - 4\sqrt{2}$ and $y_0 = \sqrt{2} - 1$. Iterate

$$(1) \qquad y_{k+1} = \frac{1 - (1 - y_k^4)^{1/4}}{1 + (1 - y_k^4)^{1/4}}$$

$$(2) \qquad \begin{aligned} a_{k+1} &= a_k(1 + y_{k+1})^4 \\ &- 2^{2k+3} y_{k+1}(1 + y_{k+1} + y_{k+1}^2) \end{aligned}$$

Then a_k converges *quartically* to $1/\pi$.

◇ 19 pairs of simple algebraic equations (1, 2) that *fit on one page* differ from π only after 700 billion digits. After 17 years, this still gives me an aesthetic buzz!

• Used since 1986, with Salamin-Brent scheme, by Bailey (LBL) and Kanada (Tokyo).

• In 1997, Kanada computed over 51 billion digits on a Hitachi supercomputer (18 iterations, 25 hrs on 2^{10} cpu's). His present world record is 2^{36} digits in April 1999.

◇ A billion (2^{30}) digit computation has been performed on a single Pentium II PC in under 9 days.

◇ The 50 billionth decimal digit of π *or* of $\frac{1}{\pi}$ is 042 !

• And after 18 billion digits, 0123456789 has finally appeared (Brouwer's famous intuitionist example *now* converges!).

Details at: www.cecm.sfu.ca/personal/jborwein/pi_cover.html.

B: (*'Pentium farming' for binary digits.*) Bailey, P. Borwein and Plouffe (1996) discovered a series for π (and some other *polylogarithmic constants*) which a startlingly allows one to compute hex–digits of π *without* computing prior digits.

• The algorithm needs very little memory and no multiple precision. The running time grows only slightly faster than linearly in the order of the digit being computed.

• The key, found by 'PSLQ' (below) is:

$$\pi = \sum_{k=0}^{\infty} \left(\frac{1}{16}\right)^k \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

• Knowing an algorithm would follow they spent several months hunting for such a formula (PSLQ).

◇ Once found, easy to prove in Mathematica, Maple or by hand.

◇ A most successful case of

REVERSE MATHEMATICAL ENGINEERING

...

This is entirely practicable, God reaches her hand deep into π:

...

• (Sept 97) Fabrice Bellard (INRIA) used a variant of this formula to compute 152 binary digits of π, starting at the *trillionth position* (10^{12}). This took 12 days on 20 work-stations working in parallel over the Internet.

PERCIVAL ON THE WEB

• (August 98) Colin Percival (SFU, age 17) finished a similar "embarrassingly parallel" computation of *five trillionth bit* (using 25 machines at about 10 times the speed). In *Hex*:

$$\underline{0}7E45733CC790B5B5979$$

The binary digits of π starting at the 40 trillionth place are

$$\underline{0}0000111110011111.$$

• (September 00) The quadrillionth bit is '0' (using 250 cpu years on 1734 machines from 56 countries).

Starting at the 999, 999, 999, 999, 997th bit of π one has:

$$111\mathbf{0}0011000100001011010110000011\mathbf{0}$$

FORM FOLLOWS FUNCTION

• A century after biology started to think physically:

> *"The waves of the sea, the little ripples on the shore, the sweeping curve of the sandy bay between the headlands, the outline of the hills, the shape of the clouds, all these are so many riddles of form, so many problems of morphology, and all of them the physicist can more or less easily read and adequately solve."*
>
> (D'Arcy Thompson, "On Growth and Form" 1917)

• In Philip Ball's "The Self-Made Tapestry: Pattern Formation in Nature,"

`http://scoop.crosswinds.net/books/tapestry.html`

• How will mathematics follow?

> *"The idea that we could make biology mathematical, I think, perhaps is not working, but what is happening, strangely enough, is that maybe mathematics will become biological!"*
>
> (Greg Chaitin, Interview, 2000)

- Consider
  - simulated annealing ('folding')
  - genetic algorithms ('scheduling')
  - neural networks ('training')
  - DNA computation ('traveling')
  - quantum computing ('sorting').

PARTITIONS and PATTERNS

• The number of *additive partitions* of n, $p(n)$, is generated by

$$P(q) := \prod_{n\geq 1} (1-q^n)^{-1}.$$

◇ Thus $p(5) = 7$ since

$$5 = 4+1 = 3+2 = 3+1+1 = 2+2+1$$
$$= 2+1+1+1 = 1+1+1+1+1.$$

QUESTION. How hard is $p(n)$ to compute — in 1900 (for MacMahon) and in 2000 (for Maple)?

...

• Algorithmic analysis uncovers *Euler's pentagonal number theorem*:

$$\boxed{\prod_{n\geq 1} (1-q^n) = \sum_{n=-\infty}^{\infty} (-1)^n q^{(3n+1)n/2}.}$$

◇ Ramanujan used MacMahon's table to find remarkable and deep congruences such as

$$p(5n+4) \equiv 0 \mod 5, \quad p(7n+5) \equiv 0 \mod 7$$
and
$$p(11n+6) \equiv 0 \mod 11,$$

from data like

$$\begin{aligned} P(q) \quad &= 1 + q + 2\,q^2 + 3\,q^3 + 5\,q^4 + 7\,q^5 + 11\,q^6 \\ &+15\,q^7 + 22\,q^8 + 30\,q^9 + 42\,q^{10} + 56\,q^{11} \\ &+77\,q^{12} + 101\,q^{13} + 135\,q^{14} + 176\,q^{15} \\ &+231\,q^{16} + 297\,q^{17} + 385\,q^{18} + 490\,q^{19} \\ &+627\,q^{20} + 792\,q^{21} + 1002\,q^{22} + 1255\,q^{23} \\ &+\cdots \end{aligned}$$

◇ We can recognize the *pentagonal numbers* in *Sloane's* on-line 'Encyclopedia of Integer Sequences'. And much more: `www.research.att.com/personal/njas/sequences/eisonline.html`.

• Keith Devlin: *Mathematics: he Science of Patterns* (1997).

A TASTE of RAMANUJAN

• G. N. Watson, discussing his response to such formulae of the wonderful Indian mathematical genius Ramanujan (1887-1920), describes:

> *"a thrill which is indistinguishable from the thrill I feel when I enter the Sagrestia Nuovo of the Capella Medici and see before me the austere beauty of the four statues representing 'Day,' 'Night,' 'Evening,' and 'Dawn' which Michelangelo has set over the tomb of Guiliano de'Medici and Lorenzo de'Medici."*
>
> (G. N. Watson, 1886-1965)

One of these is his remarkable formula

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801}\sum_{k=0}^{\infty}\frac{(4k)!(1103+26390k)}{(k!)^4 396^{4k}}$$

Each term of this series produces an additional *eight* correct digits in the result. Gosper used this formula to compute 17 million terms of the continued fraction for π in 1985.

• That said, Ramanujan prefers explicit forms such as

$$\frac{\log(640320^3)}{\sqrt{163}} = 3.14159265358979\underline{30164} \approx \pi.$$

◇ The number e^{π} is the easiest transcendental to fast compute (by elliptic methods). One 'differentiates' $e^{-t\pi}$ to obtain π (the AGM).

HARDY'S APOLOGY

"All physicists and a good many quite respectable mathematicians are contemptuous about proof." (G.H. Hardy, 1877-1947)

◇ Hardy's "A Mathematician's Apology" is a spirited defense of beauty over utility: *"Beauty is the first test. There is no permanent place in the world for ugly mathematics."*
His *"Real mathematics ... is almost wholly 'useless'"* has been overplayed and is dated: *"If the theory of numbers could be employed for any practical and obviously honourable purpose ..."*

◇ The Apology is one of Amazon's best sellers.

• "Hardy asked *'What's your father doing these days. How about that esthetic measure of his?'* I replied that my father's book was out. He said, *'Good, now he can get back to real mathematics'*." (Garret Birkhoff).

• Hardy, in "Ramanujan, Twelve Lectures ...," page 15, gives 'Skewes number' as a *"striking example of a false conjecture"*. The integral

$$\mathrm{li}\, x = \int_0^x \frac{dt}{\log t}$$

is a very good approximation to $\pi(x)$, the number of primes not exceeding x. Thus,
$\mathrm{li}\, 10^8 = 5,761,455$ while $\pi(10^8) = 5,762,209$.

• It was conjectured that

$$\mathrm{li}\, x > \pi(x)$$

and indeed it so for many x. Skewes (1933) showed the first explicit crossing $10^{10^{10^{34}}}$ — now reduced merely to 10^{1167}.

INTEGER RELATION DETECTION

The USES of LLL and PSLQ

• A vector $(x_1, x_2, \cdots, x_n)$ of reals *possesses an integer relation* if there are integers a_i not all zero with

$$0 = a_1x_1 + a_2x_2 + \cdots + a_nx_n.$$

PROBLEM: Find a_i if such exist. If not, obtain lower bounds on the size of possible a_i.

- $(n = 2)$ *Euclid's algorithm* gives solution.
- $(n \geq 3)$ Euler, Jacobi, Poincaré, Minkowski, Perron, others sought method.
- *First general algorithm* in 1977 by Ferguson & Forcade. Since '77: **LLL** (in Maple), HJLS, PSOS, **PSLQ** ('91, *parallel* '99).

• Integer Relation Detection was recently ranked among "the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century." J. Dongarra, F. Sullivan, *Computing in Science & Engineering* **2** (2000), 22–23.

Also: Monte Carlo, Simplex, Krylov Subspace, QR Decomposition, Quicksort, ..., FFT, Fast Multipole Method.

ALGEBRAIC NUMBERS

Compute α to sufficiently high precision ($O(n^2)$) and apply LLL to the vector

$$(1, \alpha, \alpha^2, \cdots, \alpha^{n-1}).$$

- Solution integers a_i are coefficients of a polynomial likely satisfied by α.

- If no relation is found, exclusion bounds are obtained.

FINALIZING FORMULAE

◇ If we know or suspect an identity exists integer relations are very powerful.

- (*Machin's Formula*) We try `lin_dep` on

  $[\arctan(1), \arctan(1/5), \arctan(1/239)]$

  and recover `[1, -4, 1]`. That is,

  $$\frac{\pi}{4} = 4\arctan(\frac{1}{5}) - \arctan(\frac{1}{239}).$$

  (Used on all serious computations of π from 1706 (100 digits) to 1973 (1 million).)

- (*Dase's Formula*). We try `lin_dep` on

  $[\pi/4, \arctan(1/2), \arctan(1/5), \arctan(1/8)]$

  and recover `[-1, 1, 1, 1]`. That is,

  $$\frac{\pi}{4} = \arctan(\frac{1}{2}) + \arctan(\frac{1}{5}) + \arctan(\frac{1}{8}).$$

  (Used by Dase to compute 200 digits of π in his head.$\cdots$)

JOHANN MARTIN ZACHARIAS DASE

• History at: `www-history.mcs.st-andrews.ac.uk`

"Zacharias Dase (1824-1861) had incredible calculating skills but little mathematical ability. He gave exhibitions of his calculating powers in Germany, Austria and England. While in Vienna in 1840 he was urged to use his powers for scientific purposes and he discussed projects with Gauss and others.

Dase used his calculating ability to calculate to 200 places in 1844. This was published in Crelle's Journal for 1844. Dase also constructed 7 figure log tables and produced a table of factors of all numbers between 7 000 000 and 10 000 000.

Gauss requested that the Hamburg Academy of Sciences allow Dase to devote himself full-time to his mathematical work but, although they agreed to this, Dase died before he was able to do much more work."

KUHN

"The issue of paradigm choice can never be unequivocally settled by logic and experiment alone.

...

in these matters neither proof nor error is at issue. The transfer of allegiance from paradigm to paradigm is a conversion experience that cannot be forced."

(Thomas Kuhn)

• In *Who got Einstein's Office?* by Ed Regis. A 1986 history of the Institute for Advanced Study.

And PLANCK

"... a new scientific truth does not triumph by convincing its opponents and making them see the light, but rather because its opponents die and a new generation grows up that's familiar with it."

(Albert Einstein quoting Max Planck)

• From "The Quantum Beat," by F.G. Major, Springer (1998)

HERSH

• Whatever the outcome of these discourses, mathematics is and will remain a uniquely human undertaking. Indeed Reuben Hersh's arguments for a humanist philosophy of mathematics, as paraphrased below, become more convincing in our setting:

1. Mathematics is human. It is part of and fits into human culture. It does not match Frege's concept of an abstract, timeless, tenseless, objective reality.

2. Mathematical knowledge is fallible. As in science, mathematics can advance by making mistakes and then correcting or even re-correcting them. The "fallibilism" of mathematics is brilliantly argued in Lakatos' *Proofs and Refutations*.

3. There are different versions of proof or rigor. Standards of rigor can vary depending on time, place, and other things. The use of computers in formal proofs, exemplified by the computer-assisted proof of the four color theorem in 1977, is just one example of an emerging nontraditional standard of rigor.

4. Empirical evidence, numerical experimentation and probabilistic proof all can help us decide what to believe in mathematics. Aristotelian logic isn't necessarily always the best way of deciding.

5. Mathematical objects are a special variety of a social-cultural-historical object. Contrary to the assertions of certain post-modern detractors, mathematics cannot be dismissed as merely a new form of literature or religion. Nevertheless, many mathematical objects can be seen as shared ideas, like Moby Dick in literature, or the Immaculate Conception in religion.

◇ From "Fresh Breezes in the Philosophy of Mathematics", *American Mathematical Monthly*, August-Sept 1995, 589–594.

• The recognition that "quasi-intuitive" analogies may be used to gain insight in mathematics can assist in the learning of mathematics. And honest mathematicians will acknowledge their role in discovery as well.

We should look forward to what the future will bring.

SANTAYANA

"When we have before us a fine map, in which the line of the coast, now rocky, now sandy, is clearly indicated, together with the winding of the rivers, the elevations of the land, and the distribution of the population, we have the simultaneous suggestion of so many facts, the sense of mastery over so much reality, that we gaze at it with delight, and need no practical motive to keep us studying it, perhaps for hours altogether. A map is not naturally thought of as an aesthetic object...

† My earliest, and still favourite, encounter with aesthetics.*

*Jerry Fodor: "... it is no doubt important to attend to the eternally beautiful and true. But it is more important not to be eaten." In Kieran Egan's, *Getting it Wrong from the Beginning*).

And yet, let the tints of it be a little subtle, let the lines be a little delicate, and the masses of the land and sea somewhat balanced, and we really have a beautiful thing; a thing the charm of which consists almost entirely in its meaning, but which nevertheless pleases us in the same way as a picture or a graphic symbol might please. Give the symbol a little intrinsic worth of form, line and color, and it attracts like a magnet all the values of things it is known to symbolize. It becomes beautiful in its expressiveness." (George Santayana)

- From "The Sense of Beauty," 1896.

A FEW CONCLUSIONS

- Draw your own! – perhaps ···
- Proofs are often out of reach – understanding, even certainty, is not.
- Packages can make concepts accessible (Maple, Cinderella).
- Progress is made 'one funeral at a time' (Niels Bohr (?)).
- 'We are Pleistocene People' (Kieran Egan).
- 'You can't go home again' (Thomas Wolfe).

***

REFERENCES

- D.H. Bailey and J.M. Borwein, "Experimental Mathematics: Recent Developments and Future Outlook," *Mathematics Unlimited — 2∞1 and Beyond*, B. Engquist and W. Schmid (Eds.), Springer–Verlag, 2000. [CECM Preprint 99:143]
- J.M. Borwein and P.B. Borwein, "Challenges for Mathematical Computing," Computing in Science & Engineering, 2001. [CECM Preprint 01:160].
- Jonathan M. Borwein and Robert Corless, "Emerging tools for experimental mathematics," *American Mathematical Monthly*, **106** (1999), 889–909. [CECM Preprint 98:110]
- J.M. Borwein and P. Lisoněk, "Applications of Integer Relation Algorithms," *Discrete Mathematics* (Special issue for FPSAC 1997), in press, 2000. [CECM Research Report 97:104]

- These and other references are available at www.cecm.sfu.ca/preprints/

◇ Quotations at jborwein/quotations.html

On *abab*-free and *abba*-free set partitions

Martin Klazar

Department of Applied Mathematics of Charles University
Malostranské náměstí 25
118 00 Praha 1
Czech Republic
klazar@kam.mff.cuni.cz

Set partitions

Martin Klazar
Department of Applied Mathematics of Charles University
Malostranské náměstí 25
118 00 Praha 1
Czech Republic
klazar@kam.mff.cuni.cz

Abstract

These are partitions of $[l] = \{1, 2, \ldots, l\}$ into n blocks such that no four term subsequence of $[l]$ induces the mentioned pattern and each k consecutive numbers of $[l]$ fall into different blocks. These structures are motivated by Davenport-Schinzel sequences. We summarize and extend known enumerative results for the pattern $p = abab$ and give an explicit formula for the number $p(abab, n, l, k)$ of such partitions. Our main tool are generating functions. We determine the corresponding generating function for $p = abba$ and $k = 1, 2, 3$. For $k = 2$ there is a connection with the number of directed animals. We solve exactly two related extremal problems.

1 Introduction and notation

A *partition* P of $[l] = \{1, 2, \ldots, l\}$ is a collection $(B_1, B_2, \ldots, B_n)$ of nonempty disjoint subsets of $[l]$, called *blocks*, whose union is $[l]$ and which are listed in the increasing order of their least elements. We define $|P| = l$ and $\|P\| = n$. Empty partition is denoted by $\emptyset$. Any partition P can be written in the *canonical sequential form* $P = a_1 a_2 \ldots a_l$ where $i \in B_{a_i}$. One can use any set of n symbols to express P this way. We call it *sequential form* and we call the set of symbols *alphabet* of P. For instance, 123242151 is the canonical sequential form of $P_0 = (\{1, 7, 9\}, \{2, 4, 6\}, \{3\}, \{5\}, \{8\})$. One of possible sequential forms is *ctrtdtcwc*, the alphabet is $\{c, t, r, d, w\}$. We are interested in enumeration of pattern-free partitions and therefore we will use often the sequential form.

A partition P is *k-regular*, $k \geq 1$, if $x, y \in B_i, x > y$, implies $x - y \geq k$. In other words, each k or less consecutive elements in the sequence are mutually different. The partition P_0 is not 3-regular but is 2-regular. 1-regularity poses no restriction. We say that P is *abab-free* if $x, y \in B_i$ and $z, t \in B_j$ for no four numbers $x < z < y < t$ and two different blocks B_i, B_j. Similarly, P is *abba-free* if $x, y \in B_i$ and $z, t \in B_j$ for no four numbers $x < z < t < y$ and two different blocks. In other words, no four term subsequence of the type *abab*, resp. *abba*, is present. It is easy to check that P_0 above is *abab*-free but not *abba*-free.

Suppose $p = abab$ or $p = abba$. By $p(p, n, l, k)$ we denote the cardinality of the set $\mathcal{P}(p, n, l, k)$ of k-regular and p-free partitions of $[l]$ with n blocks. $P(p, k)$ stands for the bivariate generating function

$$P(p, k) = P(p, k)(x, y) = \sum_{n,l \geq 0} p(p, n, l, k) x^n y^l.$$

By $p(p, n, \cdot, k)$, resp. $\mathcal{P}(p, n, \cdot, k)$, we mean $\sum_{l \geq 0} p(p, n, l, k)$, resp. $\bigcup_{l \geq 0} \mathcal{P}(p, n, l, k)$. Similarly for n replaced by the dot. Obviously $p(p, n, \cdot, 1) = \infty$ but it is not difficult to see that $p(p, n, \cdot, k) < \infty$ for $k \geq 2$. We define, for $k \geq 2$ and $n \geq 0$, $Ex(p, n, k)$ to be the maximum l such that $\mathcal{P}(p, n, l, k)$ is nonempty.

The sets $\mathcal{P}(abab, n, l, 1)$ appeared first in Kreweras [11] under the name of *noncrossing partitions*. The sets $\mathcal{P}(abab, n, \cdot, 2)$ were introduced by Davenport and Schinzel [3] when they studied $Ex(abab, n, 2)$ as a special case of a more general extremal function. The function $Ex(abab, n, 2)$ is often denoted as $\lambda_2(n)$ and is a special case of maximum lengths of *Davenport-Schinzel sequences* (we determine $Ex(abab, n, k)$ in Theorem 2.2). What is $\lambda_3(n)$ then? $Ex(ababa, n, 2)$, this function is far more difficult to handle. See [7], [15], [1], and [8] for more information and referrences.

The next section contains strenghtenings and generalizations of several known enumerative results concerning $\mathcal{P}(abab, \cdot, \cdot, \cdot)$. We determine the generating function $P(abab, k)$ and use it to generalize in Theorem 2.5 an identity of Simion and Ullman and to derive a general explicit formula for $p(abab, n, l, k)$. Nice formulas for these numbers are summarized in Theorem 2.7. Various specializations lead to Catalan, Motzkin, Narayana, and Schröder numbers. In the third section we determine in Theorem 3.1 the function $Ex(abba, n, k)$ and in Theorem 3.5 we derive an identity for $P(abba, k)$. Then we proceed to determine $P(abba, k)$ for $k = 1, 2, 3$. A specialization leads to numbers of directed animals with one root. In the last section we pose several problems.

2 *abab*-free partitions

The set of k-regular partitions of length $< k - 1$ is simply $\mathcal{C}(k) = \{\emptyset, x_1, x_1 x_2, \ldots, x_1 x_2 \ldots x_{k-2}\}$. The symbol X^j means the cartesian product $X \times X \times \ldots \times X$ j times. Here $A \times \emptyset = A$. Consider the mapping

$$F : \bigcup_{j \geq 1} (\mathcal{P}(abab, \cdot, \cdot, k) \backslash \mathcal{C}(k))^{j-1} \times \mathcal{P}(abab, \cdot, \cdot, k) \to \mathcal{P}(abab, \cdot, \cdot, k) \backslash \{\emptyset\}$$

defined by $F(u_1, u_2, \ldots, u_j) = xu_1xu_2x \ldots xu_j$ where the partitions u_i are interpreted as sequences with disjoint alphabets and x is a completely new symbol. The following easy lemma is crucial for handling *abab*-free partitions.

Lemma 2.1 *F is a bijection and if $F(u_1, u_2, \ldots, u_j) = u$ then $\sum \|u_i\| = \|u\| - 1$ and $\sum |u_i| = |u| - j$.*

Proof. It is easy to see that F is defined correctly and preserves lengths and numbers of blocks in the stated manner. Take a $u \in \mathcal{P}(abab, \cdot, \cdot, k)$, $u \neq \emptyset$, and consider the unique decomposition $u = xu_1xu_2x \ldots xu_j$ given by the occurrences of the first symbol. Note that the alphabets of u_is are disjoint. Obviously $F(u_1, u_2, \ldots, u_j) = u$ and we see that F is bijective. □

The following theorem generalizes the result $Ex(abab, n, 2) = 2n - 1$ of Davenport and Schinzel [3].

Theorem 2.2 *Suppose $k \geq 2$. For $0 \leq n \leq k-1$ we have $Ex(abab, n, k) = n$. For $n \geq k-1$ we have $Ex(abab, n, k) = 2n - k + 1$ and, for $k \geq 3$, only one partition realizing this length:*

$$u(n, k) = a_1a_2 \ldots a_{n-k+1}b_1b_2 \ldots b_{k-1}a_{n-k+1}a_{n-k} \ldots a_2a_1.$$

Proof. The first equality is trivial. We prove the rest by induction on n. For $n = k-1$ it is true. We show first $Ex(abab, n, k) \leq 2n - k + 1$. Suppose $n > k-1$ and take a $u \in \mathcal{P}(abab, n, \cdot, k)$. If no symbol in u repeats we are done. Otherwise consider the shortest interval I in u starting and ending with the same symbol. Clearly $|I| \geq k+1$ and, except for the end elements, no symbol in I repeats. The inner symbols of I cannot apppear elsewhere. Deleting the first two elements of I we get a partition v in $\mathcal{P}(abab, n-1, \cdot, k)$. So $|u| = |v| + 2 \leq 2n - 2 - k + 1 + 2 = 2n - k + 1$ and we conclude that $Ex(abab, n, k) = 2n - k + 1$.

Now suppose, in addition, that u attains the maximum length. Consider the decomposition $u = xu_1xu_2x \ldots xu_j$ of Lemma 2.1. $j = 1$ is impossible for then x could be added to the end of u. So $j \geq 2$. If u_j is nonempty and has no repetition then it can be added before u in the opposite order. If u_j is nonempty and has a repetition then x again can be added to the end of u. So u_j is empty. If $j > 2$ we get a contradiction $|u| = \sum_{i=1}^{j-1} |xu_i| + 1 \leq \sum_{i=1}^{j-1} (2\|xu_i\| - k) + 1 = 2n + 2(j-2) - (j-1)k + 1 = 2n - (j-1)(k-2) - 1 < 2n - k + 1$. So $j = 2$ and $u_2 = \emptyset$. Applying the induction assumption on u_1 we conclude that $u = u(n, k)$. □

For $k = 2$ the longest partition is not unique, actually $p(abab, n, 2n-1, 2) = \binom{2n-2}{n-1}/n$. This was proved first by Mullin and Stanton [12]. The following is both generalization and simplification of the argument of Gardy and Gouyou-Beauchamps [5] ($k = 2$).

Theorem 2.3 *For any $k \geq 1$,*

$$P(abab, k)(x, y) = \frac{1}{2y}\left(1 + y + yC(k) - xy - \sqrt{(1 + y + yC(k) - xy)^2 - 4y(1 + yC(k))}\right)$$

where $C(k) = C(k)(x, y) = 1 + xy + (xy)^2 + \ldots + (xy)^{k-2}$ ($C(2) = 1, C(1) = 0$) is the generating function for $\mathcal{C}(k)$.

Proof. Lemma 2.1 translates directly to generating functions:

$$P(abab, k) = 1 + x\sum_{j \geq 1} y^j (P(abab, k) - C(k))^{j-1} P(abab, k) = 1 + \frac{xyP(abab, k)}{1 + yC(k) - yP(abab, k)}.$$

Thus we have the quadratic equation $yP(abab, k)^2 - (1 + y + yC(k) - xy)P(abab, k) + 1 + yC(k) = 0$. Taking $P(abab, k)(0, 0) = 1$ into account we get the above solution. □

Some specializations of $P(abab,k)(x,y)$ generate standard sequences of numbers. Several special cases of $P(abab,k)(x,y)$ were also investigated before. Setting $k=1$ and $x=1$ we get the generating function $\frac{1}{2y}\left(1-\sqrt{1-4y}\right)$ of the sequence $\{p(abab,\cdot,l,1)\}_{l\geq 1}=\{1,2,5,14,42,132,429,\ldots\}$ of notorious *Catalan numbers*, A0108 in [E]. For $k=2$ and $y=1$ we get the generating function $\frac{1}{2}\left(3-x-\sqrt{1-6x+x^2}\right)$ of $\{p(abab,n,\cdot,2)\}_{n\geq 1}=\{1,2,6,22,90,394,1806,\ldots\}$. These are twice the *Schröder numbers*, A1003 in [E], which appeared first in [14]. The generating function $(P(abab,2)(x,1)-1+x)/2$ was derived in [12]. The specialization $k=2$ and $x=1$ yields $\frac{1}{2y}\left(1+y-\sqrt{1-2y-3y^2}\right)$ generating $\{p(abab,\cdot,l,2)\}_{l\geq 1}=\{1,1,2,4,9,21,51,\ldots\}$ which are *Motzkin numbers*, A1006 in [E], see [4]. For $k\geq 4$ the sequences $\{p(abab,n,\cdot,k)\}_{n\geq 1}$ seem new, for instance $\{p(abab,n,\cdot,4)\}_{n\geq 3}=\{1,2,5,13,35,97,275,\ldots\}$. For $k=3$ we get Catalan number once again. $\{p(abab,\cdot,l,k)\}_{l\geq 1}$ for $k\geq 3$ are not new, $\{p(abab,\cdot,l,3)\}_{l\geq 3}=\{1,2,4,8,17,37,82,\ldots\}$ is the sequence A4148 in [E]. These sequences were investigated in [17] by Stein and Waterman who, motivated by the secondary structure of the molecules of nucleic acids, introduced there the sets $\mathcal{P}(abab,\cdot,l,k)$. They mentioned without proof the result of C. J. Everett which we restate as the second half of the following theorem. We omit the proof as well.

Theorem 2.4

$$\lim_{k\to\infty}\lim_{n\to\infty}p(abab,n,\cdot,k)^{1/n}=\frac{3+\sqrt{5}}{2}\ \textit{and}\ \lim_{k\to\infty}\lim_{l\to\infty}p(abab,\cdot,l,k)^{1/l}=2.$$

The following theorem refines the identity of [16] where the version with two parameters k and l can be found (the proof there is combinatorial).

Theorem 2.5 *For any $n,l\geq 1$, $k\geq 2$, it is true that $p(abab,n,l,k)=p_{\leq 2}(abab,n-1,l-1,k-1)$. The subscript ≤ 2 means that we consider only the partitions with all blocks of size at most 2. Briefly, $xyP_{\leq 2}(abab,k-1)=P(abab,k)-1$.*

Proof. The generating function $P_{\leq 2}(abab,k)(x,y)$ is defined in the obvious manner. The relation for it differs from the one for $P(abab,k)$ only in that j may now attain only the values 1 and 2. So $P_{\leq 2}(abab,k)=1+x(yP_{\leq 2}(abab,k)+y^2(P_{\leq 2}(abab,k)-C(k))P_{\leq 2}(abab,k))$ and we get the equation $y(xyP_{\leq 2}(abab,k))^2-(1+xy^2C(k)-xy)(xyP_{\leq 2}(abab,k))+xy=0$. Thus

$$xyP_{\leq 2}(abab,k-1)(x,y)=\frac{1}{2y}\left(1+xy^2C(k-1)-xy-\sqrt{(1+xy^2C(k-1)-xy)^2-4xy^2}\right).$$

Taking $xy^2C(k-1)=yC(k)-y$ into account and comparing with the expression in Theorem 2.3 we get $xyP_{\leq 2}(abab,k-1)=P(abab,k)-1$. The identity is verified. □

Example

$$\mathcal{P}(abab,\cdot,5,2)=\{12345,12343,12342,12341,12324,12321,12314,12134,12131\}$$

and

$$\mathcal{P}_{\leq 2}(abab,\cdot,4,1)=\{1122,1123,1223,1233,1234,1221,1231,1232,1213\}.$$

To give an explicit formula for $p(abab,n,l,k)$ we need first to recall a well known bijection. It matches the elements of the sets $\mathcal{P}_{=2}(abab,n,2n,1)$ and $\mathcal{T}(n+1)$. Here $=2$ indicates partitions with all blocks of size 2 and $\mathcal{T}(n)$ is the set of all rooted plane trees with n vertices. Recursively: one vertex tree corresponds to $\emptyset$ and a general T corresponds to $x_1u_1x_1x_2u_2x_2\ldots x_ju_jx_j$ where u_i corresponds to the ith (counted from left) principal subtree of T and j is the degree of the root of T. The sequences u_i have disjoint alphabets and the symbols x_i are new and mutually different.

Recall that $|\mathcal{T}(n+1)| = c_n = \binom{2n}{n}/(n+1)$ is the nth Catalan number. Recall the formula

$$n(a,b) = n(a,a-b) = \frac{1}{a-b}\binom{a-1}{b}\binom{a-2}{b-1} = \frac{1}{a-1}\binom{a-1}{b}\binom{a-1}{b-1}$$

of Narayana [13] for the number of rooted plane trees with a vertices and b leaves.

Theorem 2.6 *For $k \geq 2$ and $n \leq l \leq \max(2n-k+1, n)$ we have*

$$p(abab, n, l, k) = \sum_{b=1}^{*} \frac{1}{l-n+1-b}\binom{l-n}{b}\binom{l-n-1}{b-1}\binom{l-1-b(k-2)}{2l-2n}$$

where $* = \min(l-n, \lfloor \frac{2n-l-1}{k-2} \rfloor)$ and the empty sum is equal to 1.

Proof. By Theorem 2.5 it is enough to count the number $p_{\leq 2}(abab, n-1, l-1, k-1)$ of partitions $u \in \mathcal{P}_{\leq 2}(abab, n-1, l-1, k-1)$. Each such u has $s = 2n-l-1$ singletons, symbols with one occurrence, and $d = l-n$ doubletons with two occurrences. The doubleton part of u corresponds, by the bijection, to a tree $T \in \mathcal{T}(d+1)$. By the $k-1$-regularity inside of each doubleton of u corresponding to a leaf of T there are at least $k-2$ singletons, in particular $b \leq (2n-l-1)/(k-2)$ for the number b of leaves of T. Besides this requirement singletons may be located arbitrarily in the $2d+1$ gaps of the doubleton part. The number of such u is therefore

$$\sum_{b=1}^{*} n(d+1,b)\binom{2d+1+s-b(k-2)-1}{s-b(k-2)}.$$

This is the general formula. □

In several instances one can give closed formulas.

Theorem 2.7 *For $n, l \geq 1$,*

$$p(abab, n, l, 1) = n(l+1, n) = \frac{1}{l-n+1}\binom{l}{n}\binom{l-1}{n-1},$$

$$p(abab, n, l, 2) = c_{l-n}\binom{l-1}{2l-2n} = \frac{1}{l-n+1}\binom{2l-2n}{l-n}\binom{l-1}{2l-2n},$$

$$p(abab, n, l, 3) = n(n, l-n+1) = \frac{1}{l-n+1}\binom{n-1}{2n-l-1}\binom{n-2}{2n-l-2}.$$

Thus $p(abab, n, l, 1) = p(abab, l-n+1, l, 1)$, $p(abab, n, l, 3) = p(abab, n, 3n-2-l, 3)$, $p(abab, n, l, 3) = p(abab, l-n+1, n-1, 1)$, and

$$p(abab, \cdot, n-1, 1) = p(abab, n, 2n-1, 2) = p(abab, n, \cdot, 3) = c_{n-1} = \frac{1}{n}\binom{2n-2}{n-1}.$$

Proof. The generating function for Narayana numbers $n(a,b)$ is

$$N(x,y) = \sum_{a,b \geq 1} n(a,b)x^a y^b = \frac{1-x+xy-\sqrt{(1-x+xy)^2-4xy}}{2}$$

where we put $n(1,1) = 1$. This formula can be easily derived by considerations similar to those in the proof of Theorem 2.3 and is well known. Consider the first three formulas. The formulas for $k = 1$ and $k = 3$ are consequences of the identities $P(abab, 1)(x,y) = N(y,x)/y - x + 1$ and $P(abab, 3)(x,y) =$

$N(xy,y)/y+1$ which can be readily checked. The formula for $k=2$ is a special case of Theorem 2.6 since for $k=2$

$$\sum_{b=1}^{*} n(d+1,b)\binom{2d+1+s-1}{s} = \binom{l-1}{s}\sum_{b=1}^{d} n(d+1,b) = \binom{l-1}{2l-2n}.c_d.$$

The remaining formulas follow from the symetry $n(a,b)=n(a,a-b)$ and from $\sum_b n(a,b)=c_{a-1}$. □

The formula for $p(abab,n,l,1)$ is contained implicitly already in the Narayana's result since one can prove it by an easy bijection matching partitions with trees. The formula for $p(abab,n,l,2)$ was derived in [5] directly extracting the coefficient from $P(abab,2)$. Although our counting relies on generating functions too, it indicates a bijective proof which is worked out in [9]. We have not seen the formula for $p(abab,n,l,3)$ before.

The closed formulas for $k=2,3$ are indicated by the presence of only small prime factors in the numbers $p(abab,n,l,k)$ when calculated by the general formula of Theorem 2.6. For $k\geq 4$ we get typicly factorizations as $p(abab,20,26,4)=2.13.330641$ or $p(abab,20,30,5)=5.31.2003$ which seems to exclude simple closed forms.

A sequence of numbers is called *unimodal* if it can be split into two parts, the initial one nondecreasing and the final one nonincreasing. The sequences $\{p(abab,n,l,1)\}_{n=1}^{l}$ and $\{p(abab,n,l,3)\}_{l=n}^{2n-2}$ are unimodal and symmetric. Examining the ratio $p(abab,n,l,2)/p(abab,n,l+1,2)$ one can prove easily that $\{p(abab,n,l,2)\}_{l=n}^{2n-1}$ is also unimodal and attains its maximum for $l=\lfloor n(1+\sqrt{1-1/n}/\sqrt{2})\rfloor$. Similarly, $\{p(abab,n,l,k)\}_{n=*}^{l}$, $*=\lceil (l+k-1)/2\rceil$, $k=2,3$, are unimodal for any $l\geq 2$.

Conjecture 2.8 *We conjecture that the sequences* $\{p(abab,n,l,k)\}_{l=n}^{2n-k+1}$ *and* $\{p(abab,n,l,k)\}_{n=*}^{l}$ *are unimodal for any* $n,l\geq k-1$ *and* $k\geq 2$.

3 *abba*-free partitions

Theorem 3.1 *Let* $k\geq 2$. *For* $1\leq n\leq k-1$ *again* $Ex(abba,n,k)=n$. *For* $n\geq k$ *we have* $Ex(abba,n,k)=2n+\lfloor\frac{n-1}{k-1}\rfloor-1$. *The longest partition is unique iff* $n-1$ *is divisible by* $k-1$. *In particular* $Ex(abba,n,2)=3n-2$ *and the longest partition*

$$1212323434545\ldots(n-1)n(n-1)n$$

is unique for any $n\geq 1$.

Proof. We prove first by induction on n the general upper bound. It is true for $n=k$ giving the value $2k$. Let $v\in\mathcal{P}(abba,n,\cdot,k)$ and $n>k$.

Claim 1 *One can suppose that no symbol appears in* v *more than three times.*

In the contrary case take four occurrences of a symbol a and consider the second and the third of them. A symbol $b\neq a$ must appear between them and b may have only one occurrence in v, for otherwise v is not *abba*-free. We delete the b-appearance plus possibly one a-appearance, the k-regularity is not violated. By induction $|v|\leq 2(n-1)+\lfloor\frac{n-2}{k-1}\rfloor-1+2\leq 2n+\lfloor\frac{n-1}{k-1}\rfloor-1$ and we are done in this case.

Let S_2 be the set of the symbols which appear in v at most twice and let S_3 consist of those appearing exactly three times. Let $|S_2|=n_2$ and $|S_3|=n_3$. Thus $n=n_2+n_3$.

Claim 2 $n_3(2k-4)+2\leq 2n_2-2(k-1)$

By a 3-*interval* we mean an interval I in v which begins and ends with an a-occurrence and which has one a-occurrence inside. There are n_3 3-intervals, one for each $a\in S_3$, no two of them are comparable by inclusion and no three of them intersect.

For any 3-interval I corresponding to an $a\in S_3$ there are at least $2k-2$ distinct symbols appearing in I which are distinct to a. Only at most 2 of those symbols can belong to S_3 and hence any I contributes

by at least $2k-4$ elements to S_2. On the other hand any $x \in S_2$ can appear only in at most two 3-intervals. This gives roughly the inequality in Claim 2, the corrections $+2$ and $-2(k-1)$ are caused by the first and by the last 3-interval — each contributes by at least $2k-3$ elements to S_2 and for each there are at least $k-1$ elements of S_2 which appear only in it.

Therefore $n_2 \geq n_3(k-2)+k = (n-n_2)(k-2)+k$ and $n_2 \geq n - \frac{n-1}{k-1} + 1$. Finally,

$$|v| \leq 3n_3 + 2n_2 = 3n - n_2 \leq 2n + \frac{n-1}{k-1} - 1.$$

To prove that this cannot be improved we express $n \geq k$ in the form $n-1 = m(k-1)+i, 0 \leq i < k-1$, and we consider the sequence (partition) $v(n,k) = B_1 B_2 \dots B_{m-1} B_m$ where the jth segment B_j, $1 \leq j \leq m-1$, is of the form

$$B_j = j\ x_1^j x_2^j \dots x_{k-2}^j\ (j+1)j\ x_1^j x_2^j \dots x_{k-2}^j$$

and the mth segment is of the form

$$B_m = m\ x_1^m \dots x_{k-2}^m\ (m+1)m\ y_1 y_2 \dots y_i x_1^m \dots x_{k-2}^m\ (m+1) y_1 y_2 \dots y_i.$$

The n element alphabet here is

$$\{1, 2, \dots, m+1, y_1, y_2, \dots, y_i\} \cup \{x_q^p \mid p = 1 \dots m, q = 1 \dots k-2\}.$$

An easy check reveals that the k-regular $v(n,k)$ is $abba$-free and that the length of v is

$$m(2k-1) + 2i + 1 = 2(n-1) + m + 1 = 2n + \lfloor \frac{n-1}{k-1} \rfloor - 1.$$

Thus $Ex(abba, n, k) = 2n + \lfloor \frac{n-1}{k-1} \rfloor - 1$. If $i > 0$ then the symbols $y_1, \dots, y_i$ can be placed in $v(n,k)$ differently than it is indicated above and we get several longest partitions.

It remains to prove that for $n-1$ divisible by $k-1$ there is no other longest partition than $v(n,k)$. For $n = k$ this is true. Let $n-1 > k-1$ be divisible by $k-1$ and let $u \in \mathcal{P}(abba, n, \cdot, k)$ be of the maximum length and in the canonical form. Since the length is maximum we have only symbols appearing two times or three times and no singletons. The sequence u starts with $u = 12 \dots k \dots$ and each of the symbols $1, 2, \dots, k-1$ appears in u only twice, in the contrary case we would have singletons. Thus $u = 12 \dots k-1\ k \dots 1 \dots 2 \dots$. The second 1 must follow immediately after k, in the contrary case we could delete 1's without violating k-regularity and get a sequence longer than $Ex(abba, n-1, k)$. The case $u = 12 \dots k1x \dots 2 \dots$ reduces by the switching $u = 12 \dots kx1 \dots 2 \dots$ to the previous case. So $u = 123 \dots k123 \dots k \dots$. Now k must appear three times for otherwise by deleting the initial segment of length $2k$ we would decrese n by k but l only by $2k$. Deleting the intial segment of length $2k-1$ and applying the induction assumption on the rest we conclude that $u = v(n,k)$. □

To enumerate the sets $\mathcal{P}(abba, n, l, k)$ we start with definitions and with an analogy of Lemma 2.1. Again, the subscript ≤ 2 indicates partitions with no block of size 3 or more. The set $\mathcal{I}(k)$ (resp. $\mathcal{E}(k)$) of *initial segments* (resp. *end segments*) consists of all partitions u where $u \in \mathcal{P}_{\leq 2}(abba, \cdot, \cdot, k)$ and the last (resp. the first) element of u is a doubleton. *Middle segments* $\mathcal{M}(k)$ are partitions $u \in \mathcal{P}_{\leq 2}(abba, \cdot, \cdot, k)$ such that the first and the last elements of u differ and are doubletons. Finally, *simple segments* $\mathcal{S}(k)$ are k-regular partitions u beginning and ending with a in which no symbol, except for a, repeats. Consider the mapping

$$G : \mathcal{I}(k) \times \mathcal{S}(k) \times \bigcup_{j \geq 1} (\mathcal{M}(k) \times \mathcal{S}(k))^{j-1} \times \mathcal{E}(k) \to \mathcal{P}(abba, \cdot, \cdot, k) \backslash \mathcal{P}_{\leq 2}(abba, \cdot, \cdot, k)$$

defined by $G(u_1, u_2, \dots, u_{2j+1}) = u = u_1 u_2 \dots u_{2j+1}+$ identification. This means that u_is are concatenated as sequences with disjoint alphabets and then the neighboring end elements of these segments are identified.

Lemma 3.2 *G is a bijection and $\sum |u_i| = |u| + 2j$, $\sum \|u_i\| = \|u\| + 2j$.*

Proof. The mapping G is defined correctly and preserves lengths and numbers of symbols in the stated manner. Take a $u \in \mathcal{P}(abba, \cdot, \cdot, k) \backslash \mathcal{P}_{\leq 2}(abba, \cdot, \cdot, k)$. Consider the splitting $u = v_1 a v_2 a \dots a v_m, m \geq 4$, of u by the occurrences of the first symbol a which appears more than twice. Obviously $v_1 a v_2 a \in \mathcal{I}(k)$ and $a v_3 a \dots a v_{m-2} a \in \mathcal{S}(k)$. If in $a v_{m-1} a v_m$ no symbol appears more than twice we are done since then $a v_{m-1} a v_m \in \mathcal{E}(k)$. Otherwise let $a v_{m-1} a v_m = a w_1 b w_2 b \dots b w_r$ be the splitting where b is the first symbol appearing $r \geq 3$ times and w_1 contains one a-appearance and one b-appearance. Then $a w_1 b \in \mathcal{M}(k)$ and $b w_2 b \dots b w_{r-2} b \in \mathcal{S}(k)$. Now we are left with the last segment $b w_{r-1} b w_r$. Continuing this way until the last segment falls into $\mathcal{P}_{\leq 2}(abba, \cdot, \cdot, k)$ we get a unique decomposition of u into segments. These segments have disjoint alphabets, except for the symbols $a, b, \dots$, and decompose u as described in the definition of G. Therefore G is bijective. □

We introduce the generating functions $S(k)(x,y)$, $I(k)(x,y)$, $E(k)(x,y)$, and $M(k)(x,y)$ which count the numbers of simple segments, initial segments, end segments, and middle segments with a given length and number of blocks, respectively. Clearly $I(k) = E(k)$.

Lemma 3.3 *For any $k \geq 1$,*

$$P(abba, k) = \frac{I^2(k) S(k)}{x^2 y^2 - M(k) S(k)} + P_{\leq 2}(abba, k).$$

Proof. Translating the decomposition Lemma 3.2 we get

$$P(abba, k) = P_{\leq 2}(abba, k) + I(k) S(k) [\sum_{j \geq 1} (M(k) S(k))^{j-1} (xy)^{-2j}] I(k).$$

The rest is a routine simplification using the geometric series formula. □

Lemma 3.4 *For any $k \geq 1$,*

1. $S(k)(x,y) = \frac{xy(1-xy)}{1-xy-y(xy)^{k-1}}$.
2. $I(k)(x,y) = E(k)(x,y) = (1-xy) P_{\leq 2}(abba, k)(x,y) - 1$.
3. $M(k)(x,y) = (1-xy)^2 P_{\leq 2}(abba, k)(x,y) - \frac{y(xy)^k}{1-xy} - 1 + xy$.

Proof. To build up a simple segment means to take a sequence of $m \geq 1$ a's, to put $k-1$ (mutually different) singletons into each of the $m-1$ gaps and then to add $r \geq 0$ new singletons into these gaps. Hence

$$S(k) = \sum_{m \geq 1} x y^m (xy)^{(m-1)(k-1)} \sum_{r \geq 0} \binom{m-1+r-1}{r} (xy)^r.$$

The inner sum equals, by a well known identity, $1/(1-xy)^{m-1}$. Using the geometric series formula we get the expression.

The number of initial segments of length l with n blocks equals to $p_{\leq 2}(abba, n, l, k) - p_{\leq 2}(abba, n-1, l-1, k)$, we are subtracting the partitions ending with a singleton. We have to subtract also the empty partition.

Similarly, the number of middle segments of length l with n blocks is $p_{\leq 2}(abba, n, l, k) - 2p_{\leq 2}(abba, n-1, l-1, k) + p_{\leq 2}(abba, n-2, l-2, k) - 1$ (modulo some adjustment for very small numbers n, l) which corresponds to the subtraction of the partitions beginning or ending with a singleton and the only partition beginning and ending with the same symbol. □

Putting it all together we get the following unexpected result.

Theorem 3.5 *For any $k \geq 1$,*

$$P(abba, k) = \frac{(1-2xy)P_{\leq 2}(abba, k) - 1}{(1-xy)^2 P_{\leq 2}(abba, k) - 1}.$$

Proof. Just substitute the expressions from Lemma 3.4 into the equation of Lemma 3.3. The terms with k will disappear during simplifications. □

It is surprising that the relation between $P_{\leq 2}(abba, k)$ and $P(abba, k)$ is independent on k. Theorem 3.5 is a counterpart of the relation $xyP_{\leq 2}(abab, k-1) = P(abab, k) - 1$ of Theorem 2.5.

We proceed to determine the functions $P_{\leq 2}(abba, k)$ and $P(abba, k)$ for $k = 1, 2, 3$. We know $P_{\leq 2}(abba, 1)$ already:

Lemma 3.6

$$P_{\leq 2}(abba, 1) = P_{\leq 2}(abab, 1) = \frac{P(abab, 2) - 1}{xy} = \frac{1 - xy - \sqrt{(1-xy)^2 - 4xy^2}}{2xy^2}.$$

Proof. The ultimate equality is a consequence of Theorem 2.3 and the penultimate equality is an instance of Theorem 2.5. We show by a simple bijection that $p_{\leq 2}(abba, n, l, 1) = p_{\leq 2}(abab, n, l, 1)$ for any $n, l \geq 0$ which proves the first equality.

We start with a bijection between $\mathcal{P}_{=2}(abba, n, 2n, 1)$ and $\mathcal{T}(n+1)$. Empty sequence is represented by a single vertex. Let $u \in \mathcal{P}_{=2}(abba, n, 2n, 1)$. The root of the tree T representing u will have degree m where $v = 12 \ldots m$, $u = vw$, is the maximal initial interval of u without repetitions. Consider the same decomposition $u' = v'w'$, $v' = m+1 \ldots m+r$, of the sequence u' that arises from u by deleting the $2m$ appearances of $1, \ldots, m$. Note that w starts with 1 and decomposes into $w = 1w_1 2w_2 \ldots mw_m$.

T is defined as follows. Suppose that the tree U representing u' has the principal subtrees, from left to right, $U_1, U_2, \ldots, U_r$, with roots $r(1), r(2), \ldots, r(r)$. Let $|v' \cap w_i| = l_i$, $l_1 + \ldots + l_m = r$, and let $l_0 = 0$. We delete the root of U and we join the l_j vertices $r(l_0 + \ldots + l_{j-1} + 1), \ldots, r(l_0 + \ldots + l_{j-1} + l_j)$, $j = 1, 2, \ldots, m$, to a new vertex v_j. Finally, we join the vertices v_j to a common vertex, the root of T. It is not difficult to check that this is indeed a bijection.

Now it is easy to give a bijection between $\mathcal{P}_{\leq 2}(abba, n, l, 1)$ and $\mathcal{P}_{\leq 2}(abab, n, l, 1)$. Let u lie in the former set. Consider the doubleton part of u and the tree T corresponding to it by the bijection we have just described. Replace the doubleton part by the sequence $v \in \mathcal{P}_{=2}(abab, \cdot, \cdot, 1)$ corresponding to T by the bijection described before Theorem 2.6. □

Theorem 3.7

$$P(abba, 1) = \frac{(1-xy)^2 - y - x^2y^3 P_{\leq 2}(abab, 1)}{(1-xy)^3 - y} = \frac{2 - 2y - 5xy + 3x^2y^2 + xy\sqrt{(1-xy)^2 - 4xy^2}}{2(1-xy)^3 - 2y}$$

$$P(abba, 1)(1, y) = \frac{1 - 3y + y^2 - y^3 P_{\leq 2}(abab, 1)(1, y)}{1 - 4y + 3y^2 - y^3} = \frac{-2 + 7y - 3y^2 - y\sqrt{1 - 2y - 3y^2}}{-2 + 8y - 6y^2 + 2y^3}$$

Proof. By the proof of Theorem 2.5 and by the previous lemma, the function $P_{\leq 2}(abba, 1)$ satisfies the quadratic equation $xy^2P^2 + (xy - 1)P + 1 = 0$. Thus the identity

$$((1-xy)^2 P - 1)\left(P\,\frac{xy^2}{(1-xy)^2} + \frac{xy^2}{(1-xy)^4} + \frac{1}{xy-1}\right) = -1 - \frac{1}{xy-1} - \frac{xy^2}{(1-xy)^4}$$

by which we rationalize the denominator of the expression in Theorem 3.5. Simplifying and substituting the explicit form of $P_{\leq 2}(abba, 1)$ we get the final result. The second formula arises by specialization. □

$P(abba, 1)(1, y)$ generates the sequence $\{p(abba, \cdot, l, 1)\}_{l \geq 1} = \{1, 2, 5, 14, 41, 123, 374, \ldots\}$ which seems new.

Lemma 3.8

$$P_{\leq 2}(abba,2)=\frac{P_{\leq 2}(abba,1)+1}{2+xy^2-xy}$$

Proof. Take a $u\in\mathcal{P}_{\leq 2}(abba,\cdot,\cdot,1)\backslash\mathcal{P}_{\leq 2}(abba,\cdot,\cdot,2)$ and consider the first violation of the 2-regularity $u=vaaw$. Thus $v\in\mathcal{P}_{\leq 2}(abba,\cdot,\cdot,2)$ and v and w have disjoint alphabets. Translated to generating functions, $P_{\leq 2}(abba,1)=P_{\leq 2}(abba,2).xy^2.P_{\leq 2}(abba,1)+P_{\leq 2}(abba,2)$. The solution for $P_{\leq 2}(abba,2)$ is

$$P_{\leq 2}(abba,2)=\frac{P_{\leq 2}(abba,1)}{xy^2P_{\leq 2}(abba,1)+1}.$$

Rationalizing the denominator as in the proof of Theorem 3.7 we get the desired relation. □

Setting $y=1$ in the previous lemma we get the following identity.

Consequence 3.9 *For any $n\geq 1$ it is true that $p_{\leq 2}(abba,n,\cdot,2)=p_{\leq 2}(abba,n,\cdot,-2)$. The minus sign indicates partitions which are* not *2-regular.*

Example

$$\mathcal{P}_{\leq 2}(abba,3,\cdot,2)=\{123,1213,12123,12132,121323,1231,1232,12312,12313,123123,12323\}$$

and

$$\mathcal{P}_{\leq 2}(abba,3,\cdot,-2)=\{1123,1223,1233,11233,12233,11223,112233,12133,121233,11232,112323\}.$$

Theorem 3.10

$$P(abba,2)=\frac{1-x(2y+3y^2+y^3)+x^2(y^2+y^3)-x^2y^3P_{\leq 2}(abab,1)}{1-x(3y+3y^2+y^3)+x^2(3y^2+2y^3)-x^3y^3}$$

$$P(abba,2)(1,y)=\frac{1-2y-2y^2-y^3P_{\leq 2}(abab,1)(1,y)}{1-3y}$$

$$P(abba,2)(x,1)=\frac{1-6x+2x^2-x^2P_{\leq 2}(abab,1)(x,1)}{1-7x+5x^2-x^3}$$

Proof. The expression for $P_{\leq 2}(abba,2)$ from the previous lemma is substituted in the formula of Theorem 3.5. The denominator of the resulted fraction is rationalized as in Theorem 3.7. Specializations lead to the other two formulas. □

The function $P(abba,2)(x,1)$ generates $\{p(abba,n,\cdot,2)\}_{n\geq 1}=\{1,3,15,85,501,3007,18235,\ldots\}$ which seems new. The sequence $\{p(abba,\cdot,l,2)\}_{l\geq 2}=\{1,2,5,13,35,96,267,\ldots\}$ generated by $P(abba,2)(1,y)$ is the sequence A5773 in [E]. Recall that a *directed animal with one root* is a finite set X of lattice points in the plane containing the origin and such that each point of X can be reached from the origin by a path lying completely in X and making only east or north unit steps. For more details consult [6].

Consequence 3.11 *For any $l\geq 2$ it is true that $p(abba,\cdot,l,2)$ is the same as the number of directed animals with one root and $l-1$ points.*

Proof. Simplifying the formula for $P(abba,2)(1,y)$ further we get a compact expression

$$P(abba,2)(1,y) = 1 + \frac{y}{2}\ \left(1 + \sqrt{\frac{1+y}{1-3y}}\right)$$

which equals $yQ + 1 + y$ where Q is the generating function for directed animals with one root, see [6]. □

Lemma 3.12

$$P_{\leq 2}(abba,3) = \frac{P_{\leq 2}(abba,2)}{(1-xy^2)^2 + xy^2(2-xy+x^2y^3-x^2y^4)P_{\leq 2}(abba,2)}$$

Proof. The idea is the same as in Lemma 3.8. Take a $u \in \mathcal{P}_{\leq 2}(abba,\cdot,\cdot,2)\backslash\mathcal{P}_{\leq 2}(abba,\cdot,\cdot,3)$ and consider the first violation of the 3-regularity by $u = vabaw$. Thus $v \in \mathcal{P}_{\leq 2}(abba,\cdot,\cdot,3)$ and v and w have disjoint alphabets. Now we have to distinguish three possibilities. For the sake of brevity we use P for $P_{\leq 2}(abba,3)$ and Q for $P_{\leq 2}(abba,2)$.

1) b is a singleton. The number of such u is counted by the coefficient in Px^2y^3Q.

2) b appears once more in w. The number of such u is counted by the coefficient in $P(x^2y^4Q + xy^2E(2))$. The first term counts the u's with the structure $u = vababw'$. If the second b does not follow immediately after the second a then bw is an end segment (see the beginning of Section 3) and such u's are counted by the second term.

3) b appears in v. Consider the interval I spanned by the two b appearances. Clearly $|I| \geq 4$. In the case $|I| > 4$ we are done as well as in the case when $|I| = 4$ but the other symbol in I different from a, say c, is a singleton. The bad situation is when $u = v'bcabaw$ and c appears in v'. Then consider the interval J spanned by the two c's. The bad situation is when $u = v''cdbcabaw$ and d appears in v''. Continuing this way we get a unique decomposition $u = v^*a_1sa_2a_1a_3a_2a_4a_3\ldots abaw$ where either $|s| \geq 2$ or s is a singleton. In the former case $v^*a_1sa_1$ is an initial segment in $\mathcal{I}(3)$ and such u's are accounted for in $I(3)[\sum_{m\geq 1}(xy^2)^m]Q$. In the latter case we have the splitting $v^*\ a_1sa_2a_1a_3a_2a_4a_3\ldots aba\ w$ of u into three segments with disjoint alphabets and so we account for such u in $P[\sum_{m\geq 2}(xy^2)^m]xyQ$.

We have the equation $Q = P[1 + x^2y^3Q + x^2y^4Q + xy^2(1-xy)Q - xy^2 + \frac{x^3y^5}{1-xy^2}Q + (1-xy)\frac{xy^2}{1-xy^2}Q] - \frac{xy^2}{1-xy^2}Q$ which solves for P by the stated formula. □

Theorem 3.13

$$P(abba,3) =$$

$$\frac{1 - x(2y+4y^2) + x^2(y^2+2y^4-y^5) + x^3(y^4-y^5+2y^7) - x^4(y^7-y^8+y^9) - (x^2y^3 - 2x^3y^5 + x^4y^7)F}{1 - x(3y+4y^2) + x^2(3y^2+3y^3+2y^4-y^5) - x^3(y^3+3y^5-2y^7) + x^4(y^6-y^7+y^8-y^9)}$$

where $F = P_{\leq 2}(abab,1) = (1 - xy - \sqrt{(1-xy)^2 - 4xy^2})/2xy^2$. *The specializations are*

$$P(abba,3)(x,1) = \frac{1 - 6x + 2x^2 + 2x^3 - x^4 - (x^2 - 2x^3 + x^4)P_{\leq 2}(abab,1)(x,1)}{1 - 7x + 7x^2 - 2x^3}\ \textit{and}$$

$$P(abba,3)(1,y) = \frac{1 - 2y - 3y^2 + 3y^4 - 2y^5 + y^7 + y^8 - y^9 - (y^3 - 2y^5 + y^7)P_{\leq 2}(abab,1)(1,y)}{1 - 3y - y^2 + 2y^3 + 2y^4 - 4y^5 + y^6 + y^7 + y^8 - y^9}$$

Proof. This is again only a manipulation with rational functions. First we substitute in the expression of Lemma 3.12 the formula for $P_{\leq 2}(abba,2)$ from Lemma 3.8 and express this way $P_{\leq 2}(abba,3)$ in terms of $P_{\leq 2}(abab,1)$:

$$P_{\leq 2}(abba,3) = \frac{m_1(x,y) + m_2(x,y)P_{\leq 2}(abab,1)}{m_3(x,y) + m_4(x,y)P_{\leq 2}(abab,1)}$$

where $m_1(x,y) = 1 + xy - xy^2 + x^2y^3$, $m_2(x,y) = -1 + 2xy + 2xy^2 - x^2y^3 + x^3y^5 - x^3y^6$, $m_3(x,y) = m_1(x,y) - x^2y^2$, and $m_4(x,y) = m_2(x,y) - x^2y^2$. Rationalizing the denominator we get the stated formula. □

The first specialization generates the sequence $\{p(abba,n,\cdot,3)\}_{n\geq 2} = \{1,4,19,95,448,2553,13537,\ldots\}$ and the second one the sequence $\{p(abba,\cdot,l,3)\}_{l\geq 3} = \{1,2,5,14,38,102,276,\ldots\}$, both of them seem new. Now we list the beginnings of the expansions of the functions $P(abba,k)(x,y)$ for $k = 1,2,3$.

$$P(abba,1)(x,y) = 1 + xy + (x+x^2)y^2 + (x+3x^2+x^3)y^3 + (x+6x^2+6x^3+x^4)y^4 +$$
$$+(x+9x^2+20x^3+10x^4+x^5)y^5 + (x+12x^2+44x^3+50x^4+15x^5+x^6)y^6 +$$
$$+(x+15x^2+77x^3+154x^4+105x^5+21x^6+x^7)y^7 + (x+18x^2+119x^3+350x^4+434x^5+196x^6+28x^7+x^8)y^8 +$$
$$+(x+21x^2+170x^3+663x^4+1260x^5+1050x^6+336x^7+36x^8+x^9)y^9 +$$
$$+(x+24x^2+230x^3+1120x^4+2907x^5+3822x^6+2268x^7+540x^8+45x^9+x^{10})y^{10} + \ldots$$

$$P(abba,2)(x,y) = 1 + yx + (y^2+y^3+y^4)x^2 + (y^3+3y^4+6y^5+4y^6+y^7)x^3 + (y^4+6y^5+20y^6+29y^7 +$$
$$+21y^8+7y^9+y^{10})x^4 + (y^5+10y^6+50y^7+119y^8+154y^9+111y^{10}+45y^{11}+10y^{12}+y^{13})x^5 +$$
$$+(y^6+15y^7+105y^8+364y^9+714y^{10}+837y^{11}+605y^{12}+274y^{13}+78y^{14}+13y^{15}+y^{16})x^6 +$$
$$+(y^7+21y^8+196y^9+924y^{10}+2520y^{11}+4257y^{12}+4642y^{13} +$$
$$+3354y^{14}+1638y^{15}+545y^{16}+120y^{17}+16y^{18}+y^{19})x^7 + \ldots$$

$$P(abba,3)(x,y) = 1 + yx + y^2x^2 + (y^3+y^4+y^5+y^6)x^3 + (y^4+3y^5+6y^6+7y^7+2y^8)x^4 +$$
$$+(y^5+6y^6+20y^7+34y^8+25y^9+8y^{10}+y^{11})x^5 + (y^6+10y^7+50y^8+124y^9+157y^{10}+106y^{11}+36y^{12}$$
$$+4y^{13})x^6 + (y^7+15y^8+105y^9+364y^{10}+687y^{11}+748y^{12}+465y^{13}+148y^{14}+19y^{15}+y^{16})x^7 + \ldots$$

4 Concluding remarks

We demonstrated in the paper that the structure $\mathcal{P}(p,n,l,k)$ leads to interesting extremal and enumerative results, we emphasized here the latter. Our solution for the pattern $p = abba$ is not completely satisfactory since we gave the explicit formula for $P(abba,k)$ only for the first three values of k.

Problem 1 What can be said about the generating function $P(abba,k)(x,y)$ for $k \geq 4$?

A field for exploration opens when one tries other patterns p. Methods yielding strong upper bounds on $Ex(p,n,k)$ were developed in [8], [10] but we do not know many nontrivial exact values of this function.

Problem 2 What is $Ex(abcabc,n,k), k \geq 3$? It is not too difficult to give the upper bound $6n$ on $Ex(abcabc,n,3)$ but we do not know the exact value. What can be said about the numbers $p(abcabc,n,l,k)$?

Consider the pattern *ababa*. It contains three appearances of a, thus each partition from $\mathcal{P}_{\leq 2}(\cdot,\cdot,\cdot)$ avoids it. In consequence the numbers $p(ababa,\cdot,l,k)$ and $p(ababa,n,\cdot,k)$ grow superexponentially for any fixed k and exponential rather than ordinary generating function is in place. The function $Ex(ababa,n,2)$ grows superlinearly (see [7]) and it seems very difficult to describe completely the structure of *ababa*-free sequences. Any enumerative result concerning $p = ababa$ would be of great interest.

Problem 3 What can be said about the numbers $p(ababa,n,l,k,)$?

We omitted here the first order asymptotics of the numbers $p(p,n,\cdot,k)$ and $p(p,\cdot,l,k)$, $p = abab, abba$. Knowing the explicit form of the generating function, the asymptotics can be found more or less routinely by methods described in [2]. The reader may wish to consult [17] where the asymptotics of the numbers $p(abab,\cdot,l,k)$, $k = 1,2,3$ is worked out this way.

Acknowledgments The work on this paper was done during the author's stay as a TA in the Department of Mathematics of Arizona State University, Tempe. I want to thank for the possibility to use the computer and other facilities. I thank prof. H. Kierstead for his support during my stay. Last but not least, the phenomenal database [E] of N. J. A. Sloane was very helpful.

References

[1] P. K. Agarwal, M. Sharir and P. Shor, Sharp upper and lower bounds on the lengths of general Davenport-Schinzel sequences, *J. Combin. Theory A* **52** (1989), 228–274.

[2] E. Bender, Asymptotic methods in enumeration, *Siam Review* **16** (1974), 485–515; Errata **18** (1976), 292.

[3] H. Davenport and A. Schinzel, A combinatorial problem connected with differential equations, *Amer. J. Math.* **87** (1965), 684–694.

[4] R. Donaghey and L. Shapiro, Motzkin numbers, *J. Combin. Theory A* **23** (1977), 291–301.

[5] D. Gardy and D. Gouyou-Beauchamps, Enumerating Davenport-Schinzel sequences, *Informatique théorique et Applications/Theoretical Informatics and Applications* **26** (1992), 387-402.

[6] D. Gouyou-Beauchamps and G. Viennot, Equivalence of the two-dimensional directed animal problem to a one-dimensional path problem, *Advances in Appl. Math.* **9** (1988), 334–357.

[7] S. Hart and M. Sharir, Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes, *Combinatorica* **6** (1986), 151–177.

[8] M. Klazar, Combinatorial aspects of Davenport-Schinzel sequences, thesis.

[9] M. Klazar, On the numbers of Davenport-Schinzel sequences, submitted.

[10] M. Klazar and P. Valtr, Generalized Davenport-Schinzel sequences, to appear in *Combinatorica*,

[11] G. Kreweras, Sur les partitions non croisées d'un cycle, *Discrete Math.* **1** (1972), 333–350.

[12] R. C. Mullin and R. G. Stanton, A map-theoretic approach to Davenport-Schinzel sequences, *Pacific J. Math.* **40** (1972), 167–172.

[13] V. T. Narayana, A partial order and its application to probability, *Sankhyá* **21** (1959), 91–98.

[14] E. Schröder, Vier combinatorische Probleme, *Zeitschrift für Mathematik und Physik* **15** (1870), 361–376.

[15] M. Sharir and P. K. Agarwal, *Davenport-Schinzel sequences and their geometric applications*, Cambridge University Press, in press.

[16] R. Simion and D. Ullman, On the structure of the lattice of noncrossing partitions, *Discrete Math.* **98** (1991), 193–206.

[E] N. J. A. Sloane and collaborators, On-line Encyclopedia of Integer Sequences, email: sequences@research.att.com, superseeker@research.att.com.

[17] P. R. Stein and M. S. Waterman, On some new sequences generalizing the Catalan and Motzkin numbers, *Discrete Math.* **26** (1979), 261–272.

EXPLICIT M/G/1 WAITING-TIME DISTRIBUTIONS FOR A CLASS OF LONG-TAIL SERVICE-TIME DISTRIBUTIONS

by

Joseph Abate[1]
AT&T retired

Ward Whitt[2]
AT&T Labs

February 5, 1998

Operations Research Letters 25 (1999) 25–31

[1]900 Hammond Road, Ridgewood, NJ 07450-2908

[2]Room A117, AT&T Labs, 180 Park Avenue, Building 103, Florham Park, NJ 07932-0971; email: wow@research.att.com

Abstract

O. J. Boxma and J. W. Cohen recently obtained an explicit expression for the M/G/1 steady-state waiting-time distribution for a class of service-time distributions with power tails. We extend their explicit representation from a one-parameter family of service-time distributions to a two-parameter family. The complementary cumulative distribution function (ccdf's) of the service times all have the asymptotic form $F^c(t) \sim \alpha t^{-3/2}$ as $t \to \infty$, so that the associated waiting-time ccdf's have asymptotic form $W^c(t) \sim \beta t^{-1/2}$ as $t \to \infty$. Thus the second moment of the service time and the mean of the waiting time are infinite. Our result here also extends our own earlier explicit expression for the M/G/1 steady-state waiting-time distribution when the service-time distribution is an exponential mixture of inverse Gaussian distributions (EMIG). The EMIG distributions form a two-parameter family with ccdf having the asymptotic form $F^c(t) \sim \alpha t^{-3/2} e^{-\eta t}$ as $t \to \infty$. We now show that a variant of our previous argument applies when the service-time ccdf is an undamped EMIG, i.e., with ccdf $G^c(t) = e^{\eta t} F^c(t)$ for $F^c(t)$ above, which has the power tail $G^c(t) \sim \alpha t^{-3/2}$ as $t \to \infty$. The Boxma-Cohen long-tail service-time distribution is a special case of an undamped EMIG.

Keywords: M/G/1 queue, waiting-time distribution, Pollaczek-Khintchine formula, long-tail distributions, power-tail distributions, exponential mixture of inverse Gaussian distributions.

1. Introduction

The steady-state waiting-time distribution in the M/G/1 queue is available via the classical Pollaczek-Khintchine transform. It can be readily computed by numerical transform inversion, when the service-time Laplace transform is available, e.g., as shown in Abate and Whitt [1]. Nevertheless it is interesting to have explicit formulas. When the service-time distribution has a rational transform, so does the waiting-time distribution, and the transform can be inverted analytically. More generally, the transform can be inverted analytically, yielding the Beneš formula, which is an infinite series containing n-fold convolutions of the service-time stationary-excess distribution for all n; e.g., see 4.82 on p. 255 of Cohen [8]. Because of the complexity of the Beneš formula, however, it is natural to look for more explicit formulas.

A more explicit formula for a non-rational service-time distribution was evidently first obtained for the gamma service-time distribution with shape parameter 1/2 in (9.21) of Abate and Whitt [1]. This result was extended in Proposition 8.2 of Abate and Whitt [3] to all exponential mixtures of inverse Gaussian (EMIG) service-time distributions. These service-time distributions have probability densities with asymptotics of the form $f(t) \sim \alpha t^{-3/2} e^{-\eta t}$ as $t \to \infty$, where $f(t) \sim g(t)$ as $t \to \infty$ means that $f(t)/g(t) \to 1$. Because of the $e^{-\eta t}$ term, these EMIG distributions do not have a long (a heavy) tail. However, recently, Boxma and Cohen [7] obtained an explicit expression for the M/G/1 waiting-time distribution for a class of long-tail service-time distributions. In this paper, we extend Boxma and Cohen's result to a larger class of long-tail service-time distributions. In particular, we extend our result in [3] to undamped EMIGs, i.e., to distributions with complementary cumulative distribution functions (ccdf's) $G^c(t) \equiv 1 - G(t) = e^{\eta t} F^c(t)$, where $F^c(t)$ is an EMIG ccdf. The Boxma-Cohen service-time distributions are a subclass.

Here is how the rest of this paper is organized. In Section 2 we give the explicit solution for the steady-state waiting-time distribution. In Section 3 we show that the service-time distributions used in Section 2 can be represented as undamped EMIGs. In Section 4 we show that both EMIGs and undamped EMIGs are completely monotone (mixtures of exponentials) and give their mixing densities. In Section 5 we give the asymptotic behavior of undamped EMIGs as $t \to 0$ and as $t \to \infty$. We apply that result to give the first two terms of the asymptotic expansion for the waiting-time ccdf in Section 2, which agrees with Boxma and Cohen [7]. In Section 6 we discuss the heavy-traffic approximation due to Boxma and Cohen [7]. For the service-time distributions considered here, we derive their limit from the explicit

waiting-time ccdf. We conclude in Section 7 by discussing other service-time distributions for which explicit representations of the waiting-time distribution are possible, but the greater complexity make them of dubious value.

2. The Explicit Solution

Consider a service-time probability density function (pdf) $g(t)$ with Laplace transform

$$\hat{g}(s) \equiv \int_0^\infty e^{-st} g(t) dt = 1 - \frac{s}{(\mu + \sqrt{s})(1 + \sqrt{s})} , \tag{2.1}$$

which has mean $m_1(g) = \mu^{-1}$ and all higher moments infinite. The pdf g has two-parameters, the displayed μ and the scale, which has been omitted. Both can range over the positive reals.

The Pollaczek-Khintchine formula involves the associated stationary-excess pdf $g_e(t) \equiv \mu G(t)$, $t \geq 0$. Its Laplace transform has the nice form

$$\hat{g}_e(s) \equiv \frac{1 - g(s)}{s m_1(g)} = \frac{\mu}{(\mu + \sqrt{s})(1 + \sqrt{s})} . \tag{2.2}$$

For $\mu \neq 1$,

$$\hat{g}_e(s) = \left(\frac{\mu}{1-\mu}\right) \left(\frac{1}{\mu + \sqrt{s}} - \frac{1}{1 + \sqrt{s}}\right) , \tag{2.3}$$

so that, by 29.3.37 of Abramowitz and Stegun [6],

$$g_e(t) = \mu G^c(t) = \left(\frac{\mu}{1-\mu}\right) (\psi(t) - \mu \psi(\mu^2 t)), \quad t \geq 0 \tag{2.4}$$

where

$$\psi(t) \equiv e^t \, erfc(\sqrt{t}) \sim \frac{1}{\sqrt{\pi t}} \quad \text{as} \quad t \to \infty , \tag{2.5}$$

with $erfc$ being the complementary error function, i.e.,

$$erfc(t) \equiv \frac{2}{\sqrt{\pi}} \int_t^\infty e^{-u^2} du \equiv 2\Phi^c(\sqrt{2}t) , \tag{2.6}$$

where $\Phi^c(t) \equiv 1 - \Phi(t)$ is the standard (mean 0, variance 1) normal complementary cumulative distribution function (ccdf); see 7.1.1 and 26.2.29 of Abramowitz and Stegun [6]. We will establish further properties of G and G_e in the next section.

The case $\mu = 1$ was considered by Boxma and Cohen [7]. The case $\mu = 1$ also corresponds to a subclass of beta mixtures of exponential (BME) pdf's considered by Abate and Whitt [4]; we will discuss this connection further in the next section. Boxma and Cohen show that the service-time ccdf when $\mu = 1$ is

$$G^c(t) = (2t + 1)\psi(t) - 2\sqrt{t/\pi}, \quad t \geq 0 , \tag{2.7}$$

for ψ in (2.5). In the next section we will show that the associated stationary-excess ccdf is

$$G_e^c(t) = 2\sqrt{t/\pi} - (2t-1)\psi(t), \quad t \geq 0 \ . \tag{2.8}$$

We now consider the steady-state waiting-time distribution in the M/G/1 queue with arrival rate λ. It has an atom of $1-\rho$ at the origin, assuming that $\rho \equiv \lambda/\mu < 1$, but otherwise a pdf. The Laplace transform of the ccdf is

$$\hat{W}^c(s) = \frac{\rho}{s}(1 - \hat{w}_\rho(s)) \ , \tag{2.9}$$

where $\hat{w}_\rho(s)$ is the Laplace transform of the conditional waiting time pdf, given that there is a positive wait, i.e.,

$$\hat{w}_\rho(s) = \frac{(1-\rho)\hat{g}_e(s)}{1-\rho\hat{g}_e(s)} \ . \tag{2.10}$$

Paralleling Propoosition 8.2 of Abate and Whitt [3], we can find an explicit expression for $\hat{W}^c(s)$ and analytically invert it. From (2.2)–(2.10), we deduce the following.

Theorem 2.1. *For the service-time pdf $g(t)$ with Laplace transform $\hat{g}(s)$ in (2.1),*

$$\hat{w}_\rho(s) = \frac{(1-\rho)\mu}{\nu_1 - \nu_2}\left(\frac{1}{\nu_2 + \sqrt{s}} - \frac{1}{\nu_1 + \sqrt{s}}\right) \tag{2.11}$$

and

$$\hat{W}^c(s) = \frac{\rho}{\nu_1 - \nu_2}\left(\frac{\nu_1}{\sqrt{s}(\nu_2 + \sqrt{s})} - \frac{\nu_2}{\sqrt{s}(\nu_1 + \sqrt{s})}\right) \ , \tag{2.12}$$

so that

$$W^c(t) = \frac{\rho}{\nu_1 - \nu_2}\left(\nu_1\psi(\nu_2^2 t) - \nu_2\psi(\nu_1^2 t)\right) \ , \tag{2.13}$$

where ψ is given in (2.5) and

$$\nu_{1,2} = \frac{1+\mu}{2} \pm \sqrt{\left(\frac{1+\mu}{2}\right)^2 - (1-\rho)\mu} \ . \tag{2.14}$$

Proof. Algebra yields (2.11) and (2.12). The Laplace transform (2.12) is easy to invert using 29.3.43 of Abramowitz and Stegun [6]. ∎

The case $\mu = 1$ (with $v_1 = 1 + \sqrt{\rho}$ and $v_2 = 1 - \sqrt{\rho}$) was obtained by Boxma and Cohen [7]. They included an atom at the origin in the service-time distribution, which we could do as well. The atom at the origin simply gets absorbed in ρ, i.e., corresponds to changing the arrival rate λ. This property is most easily seen from the virtual waiting time, which has the same distribution as the actual waiting time in M/G/1. A customer with 0 service time causes no change in the virtual waiting-time process upon its arrival. By the Poisson thinning property,

the arrival process of customers with positive service times is also a Poisson process but with reduced arrival rate $\lambda(1-\eta)$, where η is the atom at 0 in the service-time distribution. Hence, having an atom of mass η at 0 in the service-time distribution is equivalent to changing the arrival rate to $\lambda(1-\eta)$ and considering the service-time distribution without the atom, i.e., the conditional service-time distribution given that it is positive.

3. Undamped EMIGs

We obtain the service-time transform $\hat{g}(s)$ in (2.1) by undamping an *exponential mixture of inverse Gaussian* (EMIG) ccdf's. The EMIGs were discussed in Section 8 of [3].

Introducing a slight change of notation, we start with the Laplace transform of an EMIG pdf

$$\hat{f}(s) = \frac{\mu+1}{\mu+\sqrt{1+s}} \ . \tag{3.1}$$

Formula (3.1) is obtained from (8.9) of [3] by first replacing μ by $\mu+1$ and then introducing the scale parameter $\omega \equiv 1/2(\mu+1)$; i.e., $\hat{f}(s) = \hat{\rho}(s;\omega,\mu+1) \equiv \hat{\rho}(\omega s,1,\mu+1)$ for that ω. Paralleling $\hat{g}(s)$ in (2.1), an extra scale parameter can be added to $\hat{f}(s)$ in (3.1).

The moments of the pdf with transform in (3.1) can be derived from the inverse Gaussian moments by using (8.3) and (8.10) of [3] (r should be n in (8.3)). They are

$$m_1(F) = \frac{1}{2(\mu+1)}, \quad m_{n+1}(F) = \frac{1}{(2+2\mu)^{n+1}} \sum_{k=0}^{n} \frac{(n+1-k)(n+k)!}{k!} \left(\frac{\mu+1}{2}\right)^k \tag{3.2}$$

and squared coefficient of variation (variance divided by the mean) $c^2 = \mu+2$. For the case $\mu = 1$, (3.1) is the BME transform $\hat{v}(1/2,3/2;s)$ studied in [4] and the moments in this case are $m_n = n!\beta_n/(n+1)$ where $\beta_n = \binom{2n}{n}4^{-n}$.

Paralleling (8.13) and (8.14) of [3], the ccdf has the Laplace transform

$$\hat{F}^c(s) = \frac{1-\hat{f}(s)}{s} = \frac{1}{(\mu+\sqrt{1+s})(1+\sqrt{1+s})} \tag{3.3}$$

$$= \frac{1}{\mu-1}\left(\frac{1}{1+\sqrt{1+s}} - \frac{1}{\mu+\sqrt{1+s}}\right) \ , \quad \mu \neq 1 \ . \tag{3.4}$$

From (3.4) we see that EMIG stationary-excess pdf is

$$f_e(t) = \frac{\mu+1}{\mu-1} v(1/2,3/2;t) - \frac{2}{\mu-1} f(t) \ , \tag{3.5}$$

from which we obtain the simple moment recurrence for $\mu \neq 1$

$$m_{n+1}(F) = \frac{n!\beta_n}{2(\mu-1)} - \frac{n+1}{\mu^2-1} m_n(F) \ . \tag{3.6}$$

The recurrence formula (3.6) is recommended over (3.2) to calculate the moments. It is noteworthy that the moments $m_n(F)$ are always integer sequences when μ is an integer and they are scaled by the factor $(2+2\mu)^n$. Except for the cases $\mu = 0$ and 1, none of these integer sequences are found in Sloane and Plouffe [12]. For example, the moment sequence for $\mu = 2$ is 1, 5, 51, 807, 17445, 479565, ...

From (3.1) and 29.3.37 of Abramowitz and Stegun [6],

$$f(t) = (\mu+1)\left(\frac{e^{-t}}{\sqrt{\pi t}} - \mu e^{(\mu^2-1)t}\, erfc\,(\mu\sqrt{t})\right), \quad t \geq 0\ , \tag{3.7}$$

Going from (3.7) to (3.2) is surprisingly difficult. It can be done by applying the Gosper-Zeilberger algorithm, e.g., see Section 5.8, especially p. 236, of Graham, Knuth and Patashnik [10] or Petkovsek, Wilf and Zeilberger [11]. The associated EMIG pdf in [3], which unfortunately was inadvertently omitted from (8.10) of [3], is

$$\rho(t;1,\nu) = \frac{\nu e^{-t/2\nu}}{\sqrt{2\pi\nu t}} - 2^{-1}(\nu-1)e^{(\nu-2)t/2}\, erfc((\nu-1)\sqrt{t/2\nu})\ . \tag{3.8}$$

To obtain (3.7) and (3.8), first scale t by the factor $2v$, then let $\nu = \mu + 1$.

Similarly, from (3.4), we have for $\mu \neq 1$,

$$F^c(t) = \frac{1}{\mu-1}(\mu e^{(\mu^2-1)t}\, erfc\,(\mu\sqrt{t}) - \ erfc\,(\sqrt{t})), \quad t \geq 0\ , \tag{3.9}$$

whereas for $\mu = 1$, we invert $(1+\sqrt{1+s})^{-2}$ to get

$$F^c(t) = (1+2t)\, erfc\,(\sqrt{t}) - 2\sqrt{\pi/t}e^{-t}, \quad t \geq 0\ . \tag{3.10}$$

In the case $\mu = 1$, the pdf $f(t)$ in (3.7) coincides with the beta mixture of exponentials (BME) pdf $v(1/2, 3/2; t)$ in Abate and Whitt [4], which in turn coincides with the RBM first-moment pdf $h_1(t)$; see Table 3 in [4]. The associated cdf in (3.10) is $v(3/2, 3/2; t)/4$. (See the next section for further discussion.)

For all $\mu > 0$, the asymptotic expansion for $F^c(t)$ is

$$F^c(t) \sim \frac{e^{-t}}{\sqrt{\pi t}} \sum_{n=1}^{\infty} (-1)^{n+1} k_n(\mu) n! \beta_n t^{-n} \quad \text{as} \quad t \to \infty\ , \tag{3.11}$$

where $\beta_n = \binom{2n}{n} 4^{-n}$ is the moment sequence of the gamma pdf $\gamma(t) = e^{-t}/\sqrt{\pi t}$ as in Table 3 of [4] and

$$k_n(\mu) = \sum_{k=0}^{2n-1} \mu^k = \frac{1}{\mu-1}\left(1 - \frac{1}{\mu^{2n}}\right)\ , \tag{3.12}$$

drawing on 7.1.23 of Abramowitz and Stegun [6]. Note that $k_n(1) = 2n$.

As in our construction of B_2ME ccdf's from BME ccdf's in [4], we define the ccdf G^c associated with $\hat{g}(s)$ in (2.1) by undamping the ccdf $F^c(t)$, i.e., by letting

$$G^c(t) = e^t F^c(t), \quad t \geq 0 \ . \tag{3.13}$$

Combining (3.3) and (3.13), we obtain

$$\hat{G}^c(s) = \hat{F}^c(s-1) = \frac{1}{(\mu + \sqrt{s})(1+\sqrt{s})} \tag{3.14}$$

and

$$\hat{g}(s) = 1 - s\hat{G}^c(s) = 1 - \frac{s}{(\mu + \sqrt{s})(1+\sqrt{s})} \ , \tag{3.15}$$

just as in (2.1). Moreover,

$$\hat{G}_e^c(s) \equiv \frac{1-\hat{g}_e(s)}{s} = \left(\frac{\mu+1}{\mu}\right)\frac{1}{\sqrt{s}(1+\sqrt{s})} + \left(\frac{1}{\mu(1-\mu)}\right)\frac{1}{1+\sqrt{s}} - \left(\frac{1}{\mu(1-\mu)}\right)\frac{1}{\mu+\sqrt{s}} \ , \tag{3.16}$$

so that, by 29.3.37 and 29.3.43 of Abramowitz and Stegun [6],

$$G_e^c(t) = \frac{\mu}{1-\mu}(\mu^{-1}\psi(\mu^2 t) - \psi(t)), \quad t \geq 0 \ , \tag{3.17}$$

for ψ in (2.5).

In the case $\mu = 1$, we can apply the BME and B_2ME calculus in [4], in particular, (1.20), (1.7) and Table 3, to get

$$\begin{aligned} g_e(t) = G^c(t) &= V_2^c(1/2, 3/2; t) = e^t V(1/2, 3/2; t) \\ &= (1/4)e^t v(3/2, 3/2; t) \\ &= (2t+1)\psi(t) - 2\sqrt{t/\pi} \end{aligned} \tag{3.18}$$

and

$$\begin{aligned} G_e^c(t) &= V_2^c(3/2, 1/2; t) = e^t V^c(3/2, 1/2; t) \\ &= (3/4)e^t v(5/2, 1/2; t) \\ &= 2\sqrt{t/\pi} - (2t-1)\psi(t) \ , \end{aligned} \tag{3.19}$$

as given in (2.8).

4. Representation as a Mixture of Exponentials

We now show that EMIGs and undamped EMIGs are both completely monotone; i.e., can be expressed as mixtures of exponentials. As a consequence, they can be approximated arbitrarily closely by hyperexponential (finite mixtures of exponential) distributions; see Feldmann

and Whitt [9]. Of course, the hyperexponential approximations never match the asymptotic tail behavior. Nevertheless, the associated M/G/1 waiting-time distributions are also matched arbitrarily closely; see [9].

Theorem 4.1. *An EMIG is completely monotone; in particular, the ccdf can be expressed as*

$$F^c(t) = \int_0^1 e^{-t/y} w(y) dy \ , \tag{4.1}$$

where

$$w(y) = \frac{\mu + 1}{\pi \sqrt{y}} \left(\frac{\sqrt{1-y}}{1 + (\mu^2 - 1)y} \right), \quad 0 \le y \le 1 \ . \tag{4.2}$$

Proof. We regard the Laplace transform $\hat{F}^c(s)$ in (3.4) as the Stieltjes transform of the spectral density; i.e., initially assuming that

$$F^c(t) = \int_0^\infty e^{-xt} \phi(x) dy \ , \tag{4.3}$$

we obtain

$$\hat{F}^c(s) = \int_0^\infty \frac{1}{s+x} \phi(x) dx \ . \tag{4.4}$$

We can then calculate the alleged spectral density $\phi(x)$ by inverting its Stieltjes transform, p. 126 of Widder [14]; i.e.,

$$\phi(x) = -\frac{\mathrm{Im}\, \hat{F}^c(-x)}{\pi} = \frac{1}{\pi(\mu - 1)} \left(\frac{\sqrt{x-1}}{x} - \frac{\sqrt{x-1}}{x + \mu^2 - 1} \right) = \frac{(\mu+1)\sqrt{x-1}}{\pi x (x + \mu^2 - 1)}, \quad x > 1 \ . \tag{4.5}$$

The mixing density $w(y)$ is related to the spectral density $\phi(x)$ by $w(y) = y^{-2} \phi(y^{-1})$. Hence, from (4.5) we obtain (4.2). ∎

We can combine (3.13) and Theorem 4.1 to obtain a corresponding result for undamped EMIGS.

Corollary 1. *An undamped EMIG is also completely monotone, i.e.,*

$$G^c(t) = \int_0^1 e^{-t(1-y)/y} w(y) dy \tag{4.6}$$

$$= \int_0^\infty e^{-t/z} w(z/(z+1))(1+z)^{-2} dz \tag{4.7}$$

for $w(y)$ in (4.2).

In two special cases the EMIG is a beta mixture of exponentials (BME), as considered in [4]. Recall that the beta density is

$$b(p, q; y) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} y^{p-1} (1-y)^{q-1}, \quad 0 \le y \le 1 \ . \tag{4.8}$$

Corollary 2. *For $\mu = 0$, $w(y) = b(1/2, 1/2; y)$; for $\mu = 1$, $w(y) = b(1/2, 3/2; y)$.*

Hence, in the notation of [4], the EMIG in (3.1) is $\nu(1/2, 1/2; t)$ when $\mu = 0$ and $\nu(1/2, 3/2; t)$ when $\mu = 1$. For those cases additional properties are given in [4]. Recall that the special case considered by Boxma and Cohen [7] is $\mu = 1$. Thus their case is the B_2ME pdf $\nu_2(1/2, 3/2; t)$. By Theorem 8 of [4], it can also be expressed as a gamma mixture of Pareto distributions.

More generally, we can express the mixing pdf $w(y)$ in (4.2) as a linear combination of beta pdf's. To do so, we expand $(1 + (\mu^2 - 1)y)^{-1}$ in (4.2) in a power series.

Theorem 4.2. *For $\mu > 0$ with $\mu \neq 1$,*

$$w(y) = \frac{\mu + 1}{2} \sum_{n=0}^{\infty} (1 - \mu^2)^n \frac{\beta_n}{n+1} b\left(\frac{2n+1}{2}, 3/2; y\right) \ . \tag{4.9}$$

where $\beta_n \equiv \binom{2n}{n} 4^{-n}$, the moments of $b(1/2, 1/2; y)$.

5. Time Asymptotics

Combining (3.9) and (3.13), we obtain the undamped EMIG ccdf $G^c(t)$. From that form, we can obtain the asymptotics as $t \to 0$ and as $t \to \infty$. In particular, from (3.11),

Theorem 5.1. *For the undamped EMIG distribution,*

$$G^c(t) \sim 1 - 2(\mu + 1)\sqrt{t/\pi} \quad \text{as} \quad t \to 0 \ , \tag{5.1}$$

$$G^c(t) \sim \left(\frac{\mu + 1}{2\mu^2}\right) \frac{1}{\sqrt{\pi t^3}} \quad \text{as} \quad t \to \infty \ , \tag{5.2}$$

and

$$G_e^c(t) \sim \left(\frac{\mu + 1}{\mu}\right) \frac{1}{\sqrt{\pi t}} \quad \text{as} \quad t \to \infty \ . \tag{5.3}$$

Similarly, we obtain the large-time asymptotics for $W^c(t)$ from (2.13). For other M/G/1 waiting-time asymptotics, see Willekens and Teugels [15], Abate, Choudhury and Whitt [5] and Boxma and Cohen [7].

Theorem 5.2. *with the undamped EMIG service-time pdf transform $\hat{g}(s)$ in (2.1),*

$$W^c(t) \sim \frac{\rho}{1 - \rho} G_e^c(t) \left[1 - \frac{(1 + \mu)^2 - 2(1 - \rho)\mu}{2(1 - \rho)^2 \mu^2 t}\right] \quad \text{as} \quad t \to \infty \ . \tag{5.4}$$

Formula (5.4) here agrees with formula (3.12) of Boxma and Cohen [7] for the case $\mu = 1$.

6. Heavy-Traffic Asymptotics

Boxma and Cohen [7] establish general heavy-traffic limits and approximations as $\rho \to 1$. We obtain their result for our special case directly from the explicit representation in Section 2.

Theorem 6.1. *If $\rho \to 1$, then $\nu_1 \to 1 + \mu$, $\nu_2/(1-\rho) \to \mu/(1+\mu)$ and*

$$W^c(t/\alpha)\psi(t) \tag{6.1}$$

for $\psi(t)$ in (2.5), where

$$\alpha = \frac{(1-\rho)^2}{\rho^2}\left(\frac{\mu}{1+\mu}\right)^2 \ . \tag{6.2}$$

Based on (6.1), we would use the approximation

$$W^c(t) \approx \psi(\alpha t) = e^{\alpha t}\, erfc(\sqrt{\alpha t}) \tag{6.3}$$

for α in (6.2). Since $\rho^2 \to 1$ as $\rho \to 1$, the factor ρ^2 in (6.2) plays no role in the heavy-traffic limit. However, it makes the heavy-traffic approximation (6.3) asymptotically correct as $t \to \infty$ for each ρ as well. We could further simplify the right side of (6.3) by replacing $erfc(\sqrt{\alpha t})$ by its asymptotic form as $\alpha \to 0$, but the numerics performed by Boxma and Cohen [7] show that it is better to keep the error function. This phenomenon very closely parallels our asymptotic normal approximation for the M/G/1 busy-period distribution in Abate and Whitt [2]. Indeed, the same approximating functions are involved.

7. Other Explicit Expressions

Smith [13] first observed that if the service-time distribution has rational Laplace transform, then so does the M/G/1 steady-state waiting-time distribution, so that at least in principle it can be inverted analytically. This is easy to see in two steps: (1) going from the service-time cdf G to its associated stationary-excess cdf G_e and (2) going from G_e to the waiting-time cdf exploiting the Pollaczek-Khintchine formula. The other explicit representations obtained so far can be viewed as generalizations of this result. If the service-time distribution has a Laplace transform that is a rational function of $s^{1/n}$, then it is easy to see that so does the M/G/1 steady-state waiting-time distribution. For general n, this property seems difficult to exploit, but for $n = 2$, we can exploit it, because we can relate the transform involving $\sqrt{s}$ to the error function.

For example, at least in principle, we can obtain the explicit $M/G/1$ waiting-time distribution when the service-time distribution is a mixture of k undamped EMIGs. By the usual partial fraction expansion (assuming no multiple roots), we can represent the waiting-time distribution as a linear combination of undamped EMIGs. However, the additional complexity seems to make this approach unattractive.

References

[1] J. Abate and W. Whitt, The Fourier-series method for inverting transforms of probability distributions, *Queueing Systems* **10** (1992), 5–88.

[2] J. Abate and W. Whitt, Limits and approximations for the busy-period distribution in single-server queues, *Prob. Eng. Inf. Sci.* **9** (1995), 581–602.

[3] J. Abate and W. Whitt, An operational calculus for probability distributions via Laplace transforms, *Adv. Appl. Prob.* **28** (1996), 75–113.

[4] J. Abate and W. Whitt, Beta mixtures of exponential distributions, 1997, submitted.

[5] J. Abate, G. L. Choudhury and W. Whitt, Waiting-time tail probabilities in queues with long-tail service-time distributions, *Queueing Systems* **16** (1994), 311–338.

[6] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, Washington, D.C., 1972.

[7] O. J. Boxma and J. W. Cohen, The M/G/1 queue with heavy-tailed service-time distribution. CWI, Amsterdam, 1997.

[8] J. W. Cohen, *The Single Server Queue*, second ed., North-Holland, Amsterdam, 1982.

[9] A. Feldmann and W. Whitt, Fitting mixtures of exponentials to long-tail distributions to analyze network performance models, *Performance Evaluation* **31** (1997), 245–279.

[10] R. L. Graham, D. E. Knuth and O. Patashnik, *Concrete Mathematics*, second ed., Addison-Wesley, Reading, MA, 1994.

[11] N. Petkovsek, H. Wilf and D. Zeilberger, $A = B$, Peters, Wellesley, MA, 1996.

[12] N. J. A. Sloane and S. Plouffe, *Encyclopedia of Integer Sequences*, Academic, New York, 1995.

[13] W. L. Smith, On the distribution of queueing times, *Proc. Camb. Phil. Soc.* **49** (1953), 449–461.

[14] D. V. Widder, *An Introduction to Transform Theory*, Academic Press, New York, 1971.

[15] J. E. Willekens and J. L. Teugels, Asymptotic expansions for waiting time probabilities in an M/G/1 queue with long-tailed service time, *Queueing Systems* **10** (1992), 295–312.

IS $\pi(6521) = 6! + 5! + 2! + 1!$ UNIQUE?

CHRIS K. CALDWELL
University of Tennessee at Martin
Martin, TN 38238 USA
caldwell@utm.edu

G. L. HONAKER, JR.
Bristol, VA 24201 USA
sci-tchr@3wave.com

The first author is a professor of mathematics at UT Martin. He lives on a small "farm" in rural northwest Tennessee with his wife, five children, two cats, and numerous chickens. The second author is a schoolteacher and amateur number theorist. He is an avid chess player.

The prime counting function, $\pi(x)$, counts exactly how many primes there are less than or equal to x. The second author discovered the following "curio" (see [1]):

$$\pi(6521) = 6! + 5! + 2! + 1!.$$

If we write the positive integer x in base 10:

$$x = a_k \ldots a_2 a_1 a_0 \qquad (\text{with } a_k \geq 0)$$

are there any other prime solutions to

$$f(x) := \sum_{i=0}^{k} a_i! = \pi(x) \; ? \tag{1}$$

How many solutions could be generated if we allow x to be composite? Is there an upper bound on how far we would need to look? What if we work in a base other than 10 or use other functions? Below we provide answers to these questions, and then pose new areas for further investigation.

Searching for another

By the prime number theorem [2, pp. 225-227], the prime counting function $\pi(x)$ is asymptotic to $x / \ln x$. In fact, Dusart [3] has shown that, when $x \geq 599$,

$$\frac{x}{\ln x}\left(1 + \frac{0.992}{\ln x}\right) < \pi(x) < \frac{x}{\ln x}\left(1 + \frac{1.2762}{\ln x}\right). \tag{2}$$

The factorial $a_i!$ is at most 9! for each of the [1+log x] digits of x, so any solution x to (1) must satisfy

$$\frac{x}{\ln x}\left(1+\frac{0.992}{\ln x}\right) < \pi(x) = f(x) \le 9!\left[1+\frac{\ln x}{\ln 10}\right]. \qquad (3)$$

This statement is false for $x > 48{,}657{,}759$, so this is an upper bound for solutions. If x is an eight-digit solution beginning with 4, then the second digit is at most 8 and we can use the tighter bound

$$f(x) \le 4! + 8! + 9!\,6 < \pi(40{,}000{,}000) = 2{,}433{,}654$$

to see that there are no such solutions. Now we know $x < 40{,}000{,}000$. After checking to see that 39,999,999 does not work, we note that for $N_1 = (3.8)10^7 \le x < 39{,}999{,}999$ we have

$$f(x) \le 3! + 8! + 9!\,6 < \pi(N_1) = 2{,}318{,}966.$$

Similarly for $N_2 = (3.6)10^7 \le x < N_1$ we have

$$f(x) \le 3! + 7! + 9!\,6 < \pi(N_2) = 2{,}204{,}262.$$

Therefore there are no solutions with $x \ge N_2$.

For $N_3 = (3.0)10^7 \le x < N_2$, first we check the cases where x ends in six '9's individually; then for the remaining integers x we have

$$f(x) \le 3! + 5! + 8! + 9!\,5 < \pi(N_3) = 1{,}857{,}859.$$

A check of the integers $x \le N_3$ using the public domain program UBASIC [4] shows the following 23 solutions:

6500, 6501, 6510, 6511, **6521**, 12066, 50372, 175677, 553783, **5224903**, 5224923, 5246963, 5302479, 5854093, 5854409, 5854419, 5854429, 5854493, 5855904, 5864049, 5865393, 10990544, 11071599 [5, seq. A049529].

Of these, only 6,521 and 5,224,903 are prime [6, p. 11].

Bases other than 10

We can write x in a base B other than 10

$$x = b_k \ldots b_2 b_1 b_0 \qquad \text{(with } b_k > 0\text{)}$$

and ask whether the equation

$$g(x) := \sum_{i=0}^{k} b_i! = \pi(x) \tag{4}$$

has any solutions. Now $b_i! \le (B-1)!$ so we can replace the inequality (3) with

$$\frac{x}{\ln x} < \pi(x) = g(x) \le (B-1)!\left[1+\frac{\ln x}{\ln B}\right]. \tag{5}$$

Omitting the factor $1+0.992/\ln x$ from (3) ensures that the leftmost inequality holds for $x \ge 11$ rather than $x \ge 599$.

For each value of B the right side of (5) grows like a multiple of $\ln x$, whereas the left-hand side grows like $x/\ln x$, therefore the inequality is false for all large x. So there is a value $x_o(B)$ such that any solution satisfies $x \le x_o(B)$. We will show that we can take $x_o(B) = 2\ B\ B!\ \ln B$ for all bases $B > 2$. Since (5) is already false at $x = 13$ for $B = 2$, we may take $x_o(2) = 13$.

First note for any solution x we have $x \ge B$ (otherwise $x! = \pi(x)$), so (5) yields

$$\frac{x}{\ln x} < (B-1)!\left(1+\frac{\ln x}{\ln B}\right) \le \frac{2\ (B-1)!\ \ln x}{\ln B}. \tag{6}$$

We next show that $x < B^B$ (for $B \ge 3$). Otherwise, since $x/(\ln x)^2$ is an increasing function for $x > e^2$, the inequality above divided by $\ln x$ gives:

$$\frac{B^B}{B^2(\ln B)^2} \le \frac{x}{(\ln x)^2} < \frac{2\ (B-1)!}{\ln B} < \frac{2B}{\ln B}\left(\frac{B}{e}\right)^{B-1}.$$

The last inequality comes from $\ln (n-1)! \le n \ln n - n + 1$ (see [7, p. 79]). But this reduces to

$$e^{B-1} < 2B^2 \ln B,$$

which is false for $B \ge 6$. For the remaining bases 3, 4 and 5, we can verify $x < B^B$ individually using (5).

Finally, upon multiplying (6) by $\ln x$ and using our result $\ln x < B \ln B$, we have

$$x < 2\ (B-1)!\ B^2 \ln B,$$

which is the desired bound.

We used UBASIC and a slightly sharpened form of the bound above to lists all of the solutions for various small bases, the result of this search is in Table 1.

Insert Table 1 near here

Alternately we could choose an integer x and ask if there is any base B for which the equation (4) has a solution. Clearly $x \geq B$. If we find the least integer n such that $n! \geq \pi(x)$, then we know $b_0 = (x \bmod B) \leq n$, so B is a divisor of $x-i$ for some $i \leq n$. For each x we then have a relative short list of possible bases. In this way we find all of the prime integers $x \leq 160{,}000{,}000$ such that (4) holds (x and B are written in base 10):

(x,B) = (3,2), (3,3), (5,2), (5,3), (17,14), (19,4), (19,8), (97,24), (97,93), (101,5), (103,9), (229,5), (661,132), (661,656), (673,334), (701,232), (5449,908), (5449,5443), (5501,7), (6473,1078), (6521,10), (6719,7), (6733,7), (49037,49030), (49043,24518), (49277,7039), (56809,9467), (64921,8), (114599,8), (484061,484053), (485909,60738), (495491,9), (560437,9), (5222447,5222438), (5222501,2611246), (5222837,1305707), (5224451,580494), (5224903,10), (5378437,15), (6480811,15), (61194733,61194723), (61285057,6128505), (62009933,11) and (67717891,7524209).

There are infinitely many such solutions! To see this, let p_n be the nth prime, then $(x,B) = (p_{n!+1}, p_{n!+1}-n)$ is a solution to (4).

The multifactorials

Instead of the factorial function, we could use the double factorial function $n!!$ [8, p. 258] or its generalization—the multifactorial function. These are defined for integers n as follows.

$n! = 1$	for $n \leq 1$,	otherwise	$n! = n \cdot (n-1)!$	(n factorial)
$n!! = 1$	for $n \leq 1$,	otherwise	$n!! = n \cdot (n-2)!!$	(n double-factorial)
$n!!! = 1$	for $n \leq 1$,	otherwise	$n!!! = n \cdot (n-3)!!!$	(n triple-factorial)

and in general

$n!_k = 1$	for $n \leq 1$,	otherwise	$n!_k = n \cdot (n-k)!_k$	(n k-factorial).

For example, $13!!! = 13!_3 = 13 \cdot 10 \cdot 7 \cdot 4 \cdot 1$ and $23!_4 = 23 \cdot 19 \cdot 15 \cdot 11 \cdot 7 \cdot 3$.

The approach above can also be used to bound the integers to check for the multi-factorials. Using the double factorial function, we have four solutions: 34, 6288, 10982, and 11978. For the triple factorial function, we have these four solutions: 45, 117, 127, and 2199. If we restrict ourselves to prime solutions, then there are only two additional solutions provided by all of the multifactorial functions:

$$\pi(127) = 1!!! + 2!!! + 7!!!$$

and

$$\pi(97) = 9!_7 + 7!_7.$$

Other functions

If we just count the digits, there is one solution: 2 ($\pi(2) = 1$, and 2 has 1 digit). If we add the digits then there are four solutions: 0, 15, 27, and 39 (none of which is prime). Using higher powers, we find the following prime solutions:

$$\pi(93701) = 9^4 + 3^4 + 7^4 + 0^4 + 1^4$$

$$\pi(1776839) = 1^5 + 7^5 + 7^5 + 6^5 + 8^5 + 3^5 + 9^5$$

$$\pi(1264061) = 1^6 + 2^6 + 6^6 + 4^6 + 0^6 + 6^6 + 1^6$$

$$\pi(\mathbf{34543}) = 3^3 + 4^4 + 5^5 + 4^4 + 3^3.$$

Note that 34543, found by the first author, is also palindromic [9].

Questions for the reader

Why add the terms corresponding to each digit? We could multiply:

$$\pi(1321) = 1^3 \cdot 3^3 \cdot 2^3 \cdot 1^3$$

or alternate signs:

$$\pi(19) = -1 + 9$$

$$\pi(53) = 5^2 - 3^2, \quad \pi(227) = 2^2 - 2^2 + 7^2, \quad \pi(929) = 9^2 - 2^2 + 9^2$$

$$\pi(47501) = -4! + 7! - 5! + 0! - 1!.$$

How about backwards exponentiation: $\pi(17) = 7^1$ and $\pi(23) = 3^2$?

Exploring other functions such as the sum of divisors function, may also prove interesting. In all such cases, the authors would be pleased to hear of your results.

References

1. C. Caldwell and G. L. Honaker, Jr., "*Prime Curios!*," http://www.utm.edu/research/primes/curios/.
2. P. Ribenboim, *The New Book of Prime Number Records*, 3rd Edition, Springer-Verlag, New York, 1995.
3. P. Dusart, "The k^{th} prime is greater than $k(\ln k+\ln\ln k-1)$ for $k\geq 2$," *Math. Comp.*, **68**:225 (January 1999) 411-415.
4. C. Caldwell, "UBASIC," *J. Recreational Math.*, **25**:1 (1993) 47-54.
5. N. J. A. Sloane, "*The On-Line Encyclopedia of Integer Sequences*," http://www.research.att.com/~njas/sequences/SA.html.
6. M. Ecker, *Recreational & Educational Computing*, Issue #96 (2000) Volume 14, Number 4.
7. S. Lang, Undergraduate Analysis, Springer-Verlag, New York, 1983.
8. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions – with Formulas, Graphs, and Mathematical Tables*, Dover Pub., New York, 1974.
9. C. Caldwell, *The Prime Glossary: palindromic prime*, http://www.utm.edu/research/primes/glossary/PalindromicPrime.html.

Table 1: Solutions in other bases

base B	solutions written in base 10 (primes in boldface)
2	**3**, **5**, 6, 8, 9, 10
3	**3**, 4, **5**, 6, 8
4	4, 6, 10, **19**, 27, 63
5	**101**, **229**, 374
6	18, 20, 134, 731, 737, 789, 1547
7	**5501**, 5690, 6530, **6719**, 6726, **6733**, 13180, 14395
8	**19**, 844, 5530, 13174, 49336, 49337, 58341, 58348, **64921**, 106108, **114599**
9	21, **103,** 364, 851, 105712, 105721, 105730, 493832, 494055, 494056, **495491,** 495524, 550620, 550622, 550654, **560437,** 1029375, 1029376, 1029459, 1031285, 1041084, 1041085, 1041128, 1041411
11	5704, 5715, 6705, 106022, 107114, 5456695, 5927793, 5927804, 5927815, 5927825, 16981728, 61924436, 61934787, **62009933**, 63370216, 67733027, 67733038, 129294118, 134549464, 134549475, 134549486, 134551268, 136058582, 136058583, 197958265

CONTINUED FRACTIONS OF TAILS OF HYPERGEOMETRIC SERIES

JONATHAN MICHAEL BORWEIN, KWOK-KWONG STEPHEN CHOI AND WILFRIED PIGULLA

1. Motivation

The tails of the Taylor series for many standard functions such as arctan and log can be expressed as continued fractions in a variety of ways. A surprising side effect is that some of these continued fractions provide a dramatic acceleration for the underlying power series. These investigations were motivated by a surprising observation about Gregory's series. Gregory's series for π, truncated at $500,000$ terms gives to forty places

$$4\sum_{k=1}^{500,000}\frac{(-1)^{k-1}}{2k-1} = 3.14159\underline{0}6535897932\underline{40}4626433832\underline{6}9502884197\cdots. \tag{1}$$

To one's initial surprise only the underlined digits are wrong — differ from those of π. This is explained, ex post facto, by setting N equal to one million in the result below:

Theorem 1. *For integer N divisible by 4 the following asymptotic expansion holds:*

$$\begin{aligned}\frac{\pi}{2} - 2\sum_{k=1}^{N/2}\frac{(-1)^{k-1}}{2k-1} &\sim \sum_{m=0}^{\infty}\frac{E_{2m}}{N^{2m+1}} \\ &= \frac{1}{N} - \frac{1}{N^3} + \frac{5}{N^5} - \frac{61}{N^7} + \cdots,\end{aligned} \tag{2}$$

where the numerators $1, -1, 5, -61, 1385, -50521, \cdots$ are the Euler numbers E_0, E_2, E_4, E_6, E_8, $E_{10}, \cdots$.

The observation (1) arrived in the mail from Roy North in 1987. After verifying its truth numerically (which is much quicker today), it was an easy matter to generate a large number of the "errors" to high precision. The authors of [1] then recognized the sequence of errors in (1) as the Euler numbers — with the help of Sloane's 'Handbook of Integer Sequences'. The presumption that (1) is a form of Euler-Maclaurin summation is now formally verifiable for any fixed N in Maple. This allowed them to determine that (1) is equivalent to a set of identities between Bernoulli and Euler numbers that could with considerable effort have been established. Secure in the knowledge that (1) holds it is easier, however, to use the *Boole Summation formula* which applies directly to alternating series and *Euler*

Date: March 24, 2003.
1991 *Mathematics Subject Classification.* Primary .
Research supported by NSERC and by the Canada Research Chair Programme.

numbers (see [1]). Because N was a power of ten, the asymptotic expansion was obvious on the computer screen.

This is a good example of a phenomenon which really does not become apparent without working to reasonably high precision (who recognizes 2, -2, 10 ?), and which highlights the role of pattern recognition and hypothesis validation in experimental mathematics.

It was an amusing additional exercise to compute Pi to $5,000$ digits from (1). Indeed, with $N = 200,000$ and correcting using the first thousand even Euler numbers, Borwein and Limber [2] obtained $5,263$ digits of Pi (plus 12 guard digits). Thus, while the alternating Gregory series is very slowly convergent, the errors are highly predictable.

2. Three Continued Fraction Classes

We will discuss three classes of continued fractions: Euler, Gauss and Perron in this section.

2.1. **Euler's Continued Fraction.** Using the following notation for continued fraction:

$$\frac{a_1}{b_1} \pm \frac{a_2}{b_2} \pm \frac{a_3}{b_3} \pm \cdots = \cfrac{a_1}{b_1 \pm \cfrac{a_2}{b_2 \pm \cfrac{a_3}{b_3 \pm \ddots}}},$$

identities such as

$$a_0 + a_1 + a_1a_2 + a_1a_2a_3 + a_1a_2a_3a_4 = a_0 + \cfrac{a_1}{1 \;-\; \cfrac{a_2}{1+a_2 \;-\; \cfrac{a_3}{1+a_3 \;-\; \cfrac{a_4}{1+a_4}}}}$$

are easily verified symbolically. The general form

$$\begin{aligned}(3)\quad a_0 + a_1 + a_1a_2 + a_1a_2a_3 + \cdots + a_1a_2a_3\cdots a_N \\ = a_0 + \cfrac{a_1}{1 \;-\; \cfrac{a_2}{1+a_2 \;-\; \cfrac{a_3}{1+a_3 \;-\; \cdots \;-\; \cfrac{a_N}{1+a_N}}}}\end{aligned}$$

can then be obtained by substituting $a_N + a_N\, a_{N+1}$ for a_N and checking that the shape of the right hand side is preserved. This allows many series to be re-expressed as continued fractions. For example, with $a_0 = 0, a_1 = z, a_2 = -z^2/3, a_3 = -3z^2/5, \cdots,$

$$\arctan(z) = z - \frac{z^3}{3} + \frac{z^5}{5} - \frac{z^7}{7} + \frac{z^9}{9} - \cdots$$

we obtain, in the limit, the continued fraction for arctan due to Euler:

$$\arctan(z) = \cfrac{z}{1 + \cfrac{z^2}{3 - z^2 + \cfrac{9z^2}{5 - 3z^2 + \cfrac{25z^2}{7 - 5z^2 + \cdots}}}}.$$

When $z = 1$, this becomes the first infinite continued fraction, given by Lord Brouncker (1620-1684):

$$\frac{4}{\pi} = 1 + \frac{1}{2} \underset{+}{} \frac{9}{2} \underset{+}{} \frac{25}{2} \underset{+}{} \frac{49}{2} \underset{+}{} \cdots . \tag{4}$$

If we let $a_0 = \sum_1^N b_k$ be the initial segment of a similar series we may use (3) to replace the remaining terms by a continued fraction. For example, if we put

$$a_0 = \sum_{n=1}^{N} \frac{(-1)^{n-1} z^{2n-1}}{2n-1}, a_1 = \frac{(-1)^N z^{2N+1}}{2N+1}, a_2 = -\frac{2N+1}{2N+3} z^2, a_3 = -\frac{2N+3}{2N+5} z^2, \cdots$$

then we get

$$\arctan(z) = \sum_{n=1}^{N} (-1)^{n-1} \frac{z^{2n-1}}{2n-1} + \frac{(-1)^N z^{2N+1}}{2N+1} \underset{+}{} \frac{(2N+1)^2 z^2}{(2N+3) - (2N+1)z^2} \underset{+}{} \frac{(2N+3)^2 z^2}{(2N+5) - (2N+3)z^2} \underset{+}{} \frac{(2N+5)^2 z^2}{(2N+7) - (2N+5)z^2} \underset{+}{} \cdots . \tag{5}$$

2.2. **Gauss's Continued Fraction.** A rich vein lies in Gauss's continued fraction for the ratio of two hypergeometric functions $\dfrac{\mathrm{F}(a, b+1; c+1; z)}{\mathrm{F}(a, b; c; z)}$, see [5]. Recall that within its radius of convergence, the Gaussian hypergeometric function is defined by

$$\begin{aligned} \mathrm{F}(a, b; c; z) &= 1 + \frac{ab}{c} z + \frac{a(a+1)b(b+1)}{2!c(c+1)} z^2 \\ &+ \frac{a(a+1)(a+2)b(b+1)(b+2)}{3!c(c+1)(c+2)} z^3 + \cdots . \end{aligned} \tag{6}$$

The general continued fraction is developed by a reworking of the *contiguity relation*

$$\mathrm{F}(a, b; c; z) = \mathrm{F}(a, b+1; c+1; z) - \frac{a(c-b)}{c(c+1)} z \, \mathrm{F}(a+1, b+1; c+2; z), \tag{7}$$

and formally at least is quite easy to derive. Convergence and convergence estimates are more delicate. We therefore have

$$\frac{\mathrm{F}(a, b+1; c+1; z)}{\mathrm{F}(a, b; c; z)} = \left(1 - \frac{a(c-b)}{c(c+1)} z \frac{\mathrm{F}(a+1, b+1; c+2; z)}{\mathrm{F}(a, b+1; c+1; z)}\right)^{-1}$$

and this yields the recursive process for the continued fraction. In the limit, for $b = 0$ and replacing c by $c - 1$, this process yields

$$\mathrm{F}(a, 1; c; z) = \frac{1}{1} \underset{-}{} \frac{a_1 z}{1} \underset{-}{} \frac{a_2 z}{1} \underset{-}{} \frac{a_3 z}{1} \underset{-}{} \cdots \tag{8}$$

which is the case of present interest. Here

$$a_{2l+1} = \frac{(a+l)(c-1+l)}{(c+2l-1)(c+2l)} \qquad a_{2l+2} = \frac{(l+1)(c-a+l)}{(c+2l)(c+2l+1)}$$

for $l = 0, 1, \cdots$. We also let

$$F_M(a, 1; c; z) = \frac{1}{1} \underset{-}{} \frac{a_1 z}{1} \underset{-}{} \frac{a_2 z}{1} \underset{-}{} \cdots \underset{-}{} \frac{a_{M-1} z}{1}$$

denote the Mth convergent of the continued fraction to $F(a,1;c;z)$.

It is well known and easy to verify that $\log(1+z) = z\,\mathrm{F}(1,1;2;-z)$. It is then a pleasant surprise to discover that $\log(1+z) - z = -\frac{1}{2}z^2\,\mathrm{F}(2,1;3;-z)$, $\log(1+z) - z + \frac{1}{2}z^2 = \frac{1}{3}z^3\,\mathrm{F}(3,1;4;-z)$ and to conjecture that

$$\log(1+z) + \sum_{n=1}^{N-1} \frac{(-1)^n z^n}{n} = -\frac{(-1)^N z^N}{N}\,\mathrm{F}(N,1;N+1;-z). \tag{9}$$

This is easy to first verify for a few cases and then confirm rigorously. As always, a formula for log leads correspondingly to one for arctan:

$$\arctan(z) - \sum_{n=0}^{N-1} \frac{(-1)^n z^{2n+1}}{2n+1} = \frac{(-1)^N z^{2N+1}}{2N+1}\,\mathrm{F}\left(N+\frac{1}{2},1;N+\frac{3}{2};-z^2\right). \tag{10}$$

Happily, in both cases (8) is applicable — as it is for a variety of other functions such as $\log\left(\frac{1+z}{1-z}\right)$, $(1+z)^k$, and $\int_0^z (1+t^n)^{-1}\,dt = z\,\mathrm{F}\left(\frac{1}{n},1;1+\frac{1}{n};-z^n\right)$. Note that this last function recaptures $\log(1+z)$ and $\arctan(z)$ for $n=1$ and 2 respectively.

We next give the explicit continued fractions for (9) and (10).

Theorem 2. *Gauss's continued fractions for (9) and (10) are:*

$$\begin{aligned} &\log(1+z) + \sum_{n=1}^{N-1} \frac{(-1)^n z^n}{n} \\ &= \frac{(-1)^{N+1}z^N}{N} + \frac{N^2 z}{N+1} + \frac{1^2 z}{N+2} + \frac{(N+1)^2 z}{N+3} + \frac{2^2 z}{N+4} + \cdots \end{aligned} \tag{11}$$

and

$$\begin{aligned} &\arctan(z) - \sum_{n=0}^{N-1} \frac{(-1)^n z^{2n+1}}{2n+1} \\ &= \frac{(-1)^N z^{2N+1}}{2N+1} + \frac{(2N+1)^2 z^2}{2N+3} + \frac{2^2 z^2}{2N+5} + \frac{(2N+3)^2 z^2}{2N+7} + \frac{4^2 z^2}{2N+9} + \cdots. \end{aligned} \tag{12}$$

Suppose we return to Gregory's series, but add a few terms of the continued fraction for (10). One observes numerically that if the results are with $N = 500,000$, adding only six terms of the continued fraction has the effect of increasing the precision by 40 digits.

Example 3.

Let

$$E_1(N,M,z) := \log(1+z) - \left(-\sum_{n=1}^{N} \frac{(-z)^n}{n} - \frac{(-z)^{N+1}}{N+1}\mathrm{F}_M(N+1,1;N+2;-z)\right)$$

and

$$E_2(N,M,z) := \arctan(z) - \left(\sum_{n=0}^{N-1} \frac{(-1)^n z^{2n+1}}{2n+1} + \frac{(-1)^N z^{2N+1}}{2N+1}\mathrm{F}_M(N+\frac{1}{2},1;N+\frac{3}{2};-z^2)\right).$$

Then $E_1(N,M,z)$ and $E_2(N,M,z)$ measure the precision of the approximations to $\log(1+z)$ and $\arctan(x)$ obtained by computing the first N terms of Taylor series and then adding M terms of their continued fractions respectively. Tables 1, 2,

		5×10	5×10^2	5×10^3	5×10^4
M	0	0.48×10^{-4}	0.13×10^{-25}	0.15×10^{-232}	0.13×10^{-2292}
	1	0.43×10^{-4}	0.11×10^{-25}	0.14×10^{-232}	0.11×10^{-2292}
	2	0.40×10^{-8}	0.11×10^{-31}	0.14×10^{-240}	0.11×10^{-2302}
	3	0.34×10^{-8}	1.00×10^{-32}	0.12×10^{-240}	0.10×10^{-2302}
	4	0.12×10^{-11}	0.40×10^{-37}	0.50×10^{-248}	0.41×10^{-2312}
	5	0.10×10^{-11}	0.35×10^{-37}	0.45×10^{-248}	0.37×10^{-2312}
	6	0.78×10^{-15}	0.31×10^{-42}	0.40×10^{-255}	0.33×10^{-2321}

TABLE 1. Error $|E_1(N,M,0.9)|$ for $N=5\times 10^k (1\le k\le 4)$ and $0\le M\le 6$.

		5×10	5×10^2	5×10^3	5×10^4	5×10^5	5×10^6
M	0	0.99×10^{-2}	1.00×10^{-3}	1.00×10^{-4}	1.00×10^{-5}	1.00×10^{-6}	1.00×10^{-7}
	1	0.97×10^{-2}	1.00×10^{-3}	1.00×10^{-4}	1.00×10^{-5}	1.00×10^{-6}	1.00×10^{-7}
	2	0.91×10^{-6}	1.00×10^{-9}	1.00×10^{-12}	1.00×10^{-15}	1.00×10^{-18}	1.00×10^{-21}
	3	0.86×10^{-6}	1.00×10^{-9}	1.00×10^{-12}	1.00×10^{-15}	1.00×10^{-18}	1.00×10^{-21}
	4	0.31×10^{-9}	0.39×10^{-14}	0.40×10^{-19}	0.40×10^{-24}	0.40×10^{-29}	0.40×10^{-34}
	5	0.28×10^{-9}	0.39×10^{-14}	0.40×10^{-19}	0.40×10^{-24}	0.40×10^{-29}	0.40×10^{-34}
	6	0.22×10^{-12}	0.34×10^{-19}	0.36×10^{-26}	0.36×10^{-33}	0.36×10^{-40}	0.36×10^{-47}

TABLE 2. Error $|E_1(N,M,1)|$ for $N=5\times 10^k (1\le k\le 6)$ and $0\le M\le 6$.

		5×10	5×10^2	5×10^3	5×10^4	5×10^5	5×10^6
M	0	0.50×10^{-2}	0.50×10^{-3}	0.50×10^{-4}	0.50×10^{-5}	0.50×10^{-6}	0.50×10^{-7}
	1	0.49×10^{-2}	0.50×10^{-3}	0.50×10^{-4}	0.50×10^{-5}	0.50×10^{-6}	0.50×10^{-7}
	2	0.47×10^{-6}	0.50×10^{-9}	0.50×10^{-12}	0.50×10^{-15}	0.50×10^{-18}	0.50×10^{-21}
	3	0.44×10^{-6}	0.49×10^{-9}	0.50×10^{-12}	0.50×10^{-15}	0.50×10^{-18}	0.50×10^{-21}
	4	0.16×10^{-9}	0.20×10^{-14}	0.20×10^{-19}	0.20×10^{-24}	0.20×10^{-29}	0.20×10^{-34}
	5	0.15×10^{-9}	0.19×10^{-14}	0.20×10^{-19}	0.20×10^{-24}	0.20×10^{-29}	0.20×10^{-34}
	6	0.12×10^{-12}	0.17×10^{-19}	0.18×10^{-26}	0.18×10^{-33}	0.18×10^{-40}	0.18×10^{-47}

TABLE 3. Error $|E_2(N,M,1)|$ for $N=5\times 10^k (1\le k\le 6)$ and $0\le M\le 6$.

3 and 4 record those data for the approximations to $\log(1.9), \log(2), \arctan(1)$ and $\arctan(1/2)+\arctan(1/5)+\arctan(1/8)$ respectively. Note that

$$\frac{\pi}{4}=\arctan(\frac{1}{2})+\arctan(\frac{1}{5})+\arctan(\frac{1}{8})$$

is a formula of Machin type used by Johann Dase to compute 205 digits of π in his head in 1844.

After some further numerical experimentation it is clear that for large a, c the continued fraction $\mathrm{F}(a,1,c;z)$ is rapidly convergent. And indeed the rough rate is apparent.

This is part of the content of the next theorem:

		5×10	5×10^2
M	0	0.31×10^{-32}	0.37×10^{-304}
	1	0.19×10^{-33}	0.23×10^{-305}
	2	0.11×10^{-37}	0.15×10^{-311}
	3	0.26×10^{-38}	0.37×10^{-312}
	4	0.56×10^{-42}	0.92×10^{-318}
	5	0.13×10^{-42}	0.23×10^{-318}
	6	0.59×10^{-46}	0.13×10^{-323}

TABLE 4. Error $|E_2(N+1,M,1/2)+E_2(N+1,M,1/5)+E_2(N+1,M,1/8)|$ for $N=5\times 10^k (1\le k\le 2)$ and $0\le M\le 6$.

Theorem 4. *Suppose* $2 \le a, a+1 \le c \le 2a$ *and* $M \ge 2$. *Then for* $-1 \le z < 0$ *one has*

$$\begin{aligned} & |\mathrm{F}(a,1;c;z) - \mathrm{F}_M(a,1;c;z)| \\ \le\ & \frac{\Gamma(n+1)(n+a)\Gamma(n+c-a)\Gamma(a)\Gamma(c)}{\Gamma(n+a)\Gamma(n+c)a\Gamma(c-a)} \left(\frac{2a}{(c-2)\left(1-\frac{2}{z}\right)+(2a-c)} \right)^M \end{aligned}$$

where $n = [M/2]$ *and* $\mathrm{F}_M(a,1;c;z)$ *is the* M*-th convergent of the continued fraction to* $F(a,1,c;z)$.

The proof of Theorem 4 will be given in the Appendix below.

In [5] one can find listed many explicit continued fractions which can be derived from Gauss's continued fraction or various of its limiting cases. These include $\exp, \tanh, \tan$ and various less elementary functions. One especially attractive fraction is that for $J_{n-1}(z)/J_n(z)$ and $I_{n-1}(z)/I_n(z)$ where J and I are *Bessel functions of the first kind*. In particular,

$$\frac{J_{n-1}(2z)}{J_n(2z)} = \frac{n}{z} - \cfrac{\frac{z}{(n+1)}}{1} - \cfrac{\frac{z^2}{(n+1)(n+2)}}{1} - \cfrac{\frac{z^2}{(n+2)(n+3)}}{1} - \cdots . \tag{13}$$

Setting $z = i$ and $n = 1$ leads to the very beautiful continued fraction

$$\frac{I_1(2)}{I_0(2)} = [1,2,3,4,\cdots] .$$

In general, arithmetic simple continued fractions correspond to such ratios.

An example of a more complicated situation is:

$$\frac{(2\,z)^{2\,N+1}\,\mathrm{F}\left(N+\frac{1}{2},\frac{1}{2};N+\frac{3}{2};z^2\right)}{(N+1)\binom{2\,N+2}{N+1}\mathrm{F}\left(\frac{1}{2},-\frac{1}{2};\frac{1}{2};z^2\right)} = \frac{\arcsin(z)}{\sqrt{1-z^2}} - \sigma_{2N}(z) \tag{14}$$

where σ_{2N} is the $2N$-th Taylor polynomial for $\frac{\arcsin(z)}{\sqrt{1-z^2}}$. Only for $N=0$ is this precisely of the form of Gauss's continued fraction.

2.3. **Perron's Continued Fraction.** Another continued fraction expansion is based on Stieltjes work on the moment problem (see Perron [4]) and leads to similar acceleration. In volume 2, page 18 of [4] one finds a beautiful continued fraction for

$$\frac{1}{z^\mu}\int_0^z \frac{t^\mu}{1+t}\,dt = \frac{z}{\mu+1} + \frac{(\mu+1)^2 z}{(\mu+2)-(\mu+1)z} + \frac{(\mu+2)^2 z}{(\mu+3)-(\mu+2)z} + \cdots. \tag{15}$$

valid for $\mu > -1, -1 < z \leq 1$. One may deduce this as a consequence of Euler's continued fraction if we write

$$\frac{1}{z^\mu}\int_0^z \frac{t^\mu}{1+t}\,dt = \frac{z}{\mu+1} - \frac{z^2}{\mu+2} + \frac{z^3}{\mu+3} - \frac{z^4}{\mu+4} + \cdots$$

and observe that (15) follows from (3) in the limit.

Since

$$\frac{z^{\mu+1}}{\mu+1}\,\mathrm{F}\left(\mu+1, 1; \mu+2; -z\right) = \int_0^z \frac{t^\mu}{1+t}\,dt, \tag{16}$$

$$\frac{z^{2\,\mu+1}}{2\,\mu+1}\,\mathrm{F}\left(\mu+\frac{1}{2}, 1; \mu+\frac{3}{2}; -z^2\right) = \int_0^z \frac{t^{2\,\mu}}{1+t^2}\,dt, \tag{17}$$

for $\mu > 0$, on examining (9) and (10) this is immediately applicable to provide Euler continued fractions for the tail of the log and arctan series. Explicitly, we obtain:

Theorem 5. *Perron's continued fractions for (9) and (10) are:*

$$\log(1+z) + \sum_{n=1}^{N-1} \frac{(-1)^n z^n}{n} = \frac{(-1)^{N+1} z^N}{N} + \frac{N^2 z}{(N+1)-Nz} + \frac{(N+1)^2 z}{(N+2)-(N+1)z} + \cdots \tag{18}$$

and

$$\arctan(z) - \sum_{n=0}^{N-1} \frac{(-1)^n z^{2\,n+1}}{2\,n+1} = \frac{(-1)^N z^{2N+1}}{2N+1} + \frac{(2N+1)^2 z^2}{(2N+3)-(2N+1)z^2} + \frac{(2N+3)^2 z^2}{(2N+5)-(2N+3)z} + \cdots. \tag{19}$$

Moreover, while the Gauss and Euler/Perron continued fractions obtained are quite distinct the convergence behaviour is very similar to that of the previous section. Note also the coincidence of (19) and (5). Indeed as we have seen Theorem 5 coincides with a special case of (3).

3. Appendix

Recall that Gauss's continued fraction for $\mathrm{F}(a, 1; c; z)$ is

$$\mathrm{F}(a, 1; c; z) = \frac{1}{1} - \frac{a_1 z}{1} - \frac{a_2 z}{1} - \frac{a_3 z}{1} - \cdots$$

where

$$a_{2l+1} = \frac{(a+l)(c-1+l)}{(c+2l-1)(c+2l)} \qquad a_{2l+2} = \frac{(l+1)(c-a+l)}{(c+2l)(c+2l+1)}$$

for $l = 0, 1, \cdots$. Let

$$\frac{A_n(z)}{B_n(z)} = \frac{1}{1} - \frac{a_1 z}{1} - \frac{a_2 z}{1} - \cdots - \frac{a_{n-1} z}{1} = \mathrm{F}_n(a, 1; c; z)$$

be the n-th convergent of the continued fraction. It can be proved by induction that $A_1(z) = A_2(z) = B_1(z) = 1, B_2(z) = 1 - a_1 z$ and

$$A_k(z) = A_{k-1}(z) - a_{k-1} z A_{k-2}(z),$$

and

$$B_k(z) = B_{k-1}(z) - a_{k-1} z B_{k-2}(z),$$

for $k \geq 3$. Hence for $k \geq 2$, we have

$$A_k(z) B_{k-1}(z) - A_{k-1}(z) B_k(z) = a_1 \cdots a_{k-1} z^{k-1}.$$

Using the estimation in Theorem 8.9 of [3], we find that if $a_i > 0$ for all i, then

$$\left| \mathrm{F}(a, 1; c; z) - \frac{A_n(z)}{B_n(z)} \right| \leq \left| \frac{A_n(z)}{B_n(z)} - \frac{A_{n-1}(z)}{B_{n-1}(z)} \right| = \left| \frac{a_1 \cdots a_{n-1} z^{n-1}}{B_n(z) B_{n-1}(z)} \right|$$

One may verify that $B_n(z)$ are hypergeometric polynomials (see [5]) and explicitly

$$B_{2k}(z) = F(-k, 1 - a - k, 2 - c - 2k; z)$$

and

$$B_{2k+1}(z) = F(-k, -a - k, 1 - c - 2k; z).$$

These may also be written in terms of Jacobi Polynomials so that

$$B_{2k}(z) = \binom{2k + c - 2}{k}^{-1} (-z)^k P_k^{(a-1, c-a-1)} \left(1 - \frac{2}{z}\right)$$

and

$$B_{2k+1}(z) = \binom{2k + c - 1}{k}^{-1} (-z)^k P_k^{(a, c-a-1)} \left(1 - \frac{2}{z}\right).$$

We let

$$E_n := E_n(a, c, z) = \frac{a_1 a_2 \cdots a_n z^n}{B_n(z) B_{n+1}(z)} \qquad \text{and} \qquad F_n := \mathrm{F}_n(a, c, z) = \frac{E_{n+1}}{E_n}.$$

Then we get

$$F_{2n} = \frac{a_{2n+1} z B_{2n}(z)}{B_{2n+2}(z)} = \frac{(n + a)}{(n + 1)} \frac{P_n^{(a-1, c-a-1)}}{P_{n+1}^{(a-1, c-a-1)}} \left(1 - \frac{2}{z}\right)$$

and

$$F_{2n-1} = \frac{a_{2n} z B_{2n-1}(z)}{B_{2n+1}(z)} = \frac{(n + c - a - 1)}{(n + c - 1)} \frac{P_n^{(a, c-a-1)}}{P_{n+1}^{(a, c-a-1)}} \left(1 - \frac{2}{z}\right).$$

We need the following estimation. Assume $0 \leq \beta \leq \alpha, 1 \leq \alpha, 1 \leq n$ and $0 < x \leq 1$. We shall show

$$\frac{P_n^{(\alpha, \beta)}}{P_{n-1}^{(\alpha, \beta)}} \left(1 + \frac{2}{x}\right) \geq \frac{(n + \alpha - 1) \left((\alpha + \beta) \left(1 + \frac{2}{x}\right) + (\alpha - \beta)\right)}{2n\alpha}. \tag{20}$$

The Jacobi polynomials satisfy the recurrence relation

$$\begin{aligned}(21)\quad & 2n(n+\alpha+\beta)(2n+\alpha+\beta-2)P_n^{(\alpha,\beta)}(x)\\ &= (2n+\alpha+\beta-1)\left((2n+\alpha+\beta)(2n+\alpha+\beta-2)x+\alpha^2-\beta^2\right)P_{n-1}^{(\alpha,\beta)}(x)\\ &\qquad -2(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)P_{n-2}^{(\alpha,\beta)}(x)\end{aligned}$$

for $n=2,3,\cdots$ where

$$P_0^{(\alpha,\beta)}(x)\equiv 1 \qquad P_1^{(\alpha,\beta)}(x)=\frac{1}{2}(\alpha+\beta+2)x+\frac{1}{2}(\alpha-\beta).$$

We let

$$R_n := \frac{P_n^{(\alpha,\beta)}}{P_{n-1}^{(\alpha,\beta)}}\left(1+\frac{2}{x}\right)$$

and

$$T_n := \frac{(n+\alpha-1)\left((\alpha+\beta)\left(1+\frac{2}{x}\right)+(\alpha-\beta)\right)}{2n\alpha}.$$

For $n=1$,

$$\begin{aligned}R_1 &= \frac{1}{2}(\alpha+\beta+2)\left(1+\frac{2}{x}\right)+\frac{1}{2}(\alpha-\beta)\\ &\geq \frac{(\alpha+\beta)\left(1+\frac{2}{x}\right)+(\alpha-\beta)}{2}=T_1.\end{aligned}$$

So (20) is true for $n=1$. By the recurrence relation (21), we get

$$\begin{aligned}R_n &= \frac{(2n+\alpha+\beta-1)\left\{(2n+\alpha+\beta)(2n+\alpha+\beta-2)\left(1+\frac{2}{x}\right)+\alpha^2-\beta^2\right\}}{2n(n+\alpha+\beta)(2n+\alpha+\beta-2)}\\ &\qquad -\frac{(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)}{n(n+\alpha+\beta)(2n+\alpha+\beta-2)}\frac{1}{R_{n-1}}\\ &:= \alpha_n-\beta_n\frac{1}{R_{n-1}}\end{aligned}$$

for $n\geq 2$. Suppose (20) is true for $n-1$. Then

$$R_n \geq \alpha_n-\frac{\beta_n}{T_{n-1}}.$$

For convenience, we write $f(n,\alpha,\beta,x)$ for the numerator of the expression $\alpha_n-\frac{\beta_n}{T_{n-1}}-T_n$ after simplification to a fractional form, that is

$$\begin{aligned}&\frac{f(n,\alpha,\beta,x)}{x(n-2+\alpha)(\alpha x+\alpha+\beta)n(n+\alpha+\beta)(2n+\alpha+\beta-2)\alpha}\\ :=\ & \alpha_n-\frac{\beta_n}{T_{n-1}}-T_n.\end{aligned}$$

The function $f(n,\alpha,\beta,x)$ is a polynomial in n of degree 4 and can be shown that subject to our conditions on α,β and x, that it is increasing on n and $f(1,\alpha,\beta,x)>0$. It follows that $\alpha_n-\frac{\beta_n}{T_{n-1}}>T_n$ and $R_n\geq T_n$. This proves (20).

In view of (20), we have

$$F_{2n}\leq \frac{(n+a)}{(n+a-1)}\frac{2(a-1)}{\left((c-2)\left(1-\frac{2}{x}\right)+(2a-c)\right)}$$

and

$$F_{2n-1} \leq \frac{n(n+c-a-1)}{(n+c-1)(n+a-1)} \frac{2a}{\left((c-1)\left(1-\frac{2}{z}\right)+(2a-c+1)\right)}.$$

Thus for $n \geq 1$,

$$F_{2n}F_{2n-1} \leq \frac{(n+a)n(n+c-a-1)}{(n+c-1)(n+a-1)^2} \left\{ \frac{2a}{(c-2)\left(1-\frac{2}{z}\right)+(2a-c)} \right\}^2.$$

We are now ready to estimate E_n. Note that

$$\begin{aligned} E_{2n+1} &= E_1 F_{2n} \cdots F_1 \\ &= E_1 \left\{ \prod_{i=1}^{n} \frac{(i+a)i(i+c-a-1)}{(i+c-1)(i+a-1)^2} \right\} \left(\frac{2a}{(c-2)\left(1-\frac{2}{z}\right)+(2a-c)} \right)^{2n} \\ &\leq \frac{\Gamma(n+1)(n+a)\Gamma(n+c-a)\Gamma(a)\Gamma(c)}{\Gamma(n+a)\Gamma(n+c)a\Gamma(c-a)} \left(\frac{2a}{(c-2)\left(1-\frac{2}{z}\right)+(2a-c)} \right)^{2n+1} \end{aligned}$$

as claimed, because

$$E_1 = \frac{a_1 z}{B_1(z)B_2(z)} \leq \frac{2a}{(c-2)\left(1-\frac{2}{z}\right)+(2a-c)}.$$

The bound for E_{2n} can be obtained similarly. This proves Theorem 4. **QED**

References

[1] Jonathan Borwein and Peter Borwein and K. Dilcher, "Pi, Euler Numbers and Asymptotic Expansions," *American Mathematical Monthly*, **96** (1989), 681–687.

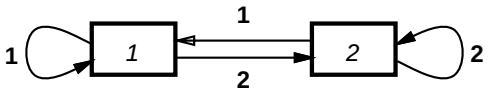
[2] Jonathan M. Borwein and Mark A. Limber, "Maple as a high precision calculator, *Maple News Letter*, **8** (1992), 39-44, and `www.cecm.sfu.ca/ preprints/1998pp.html`.

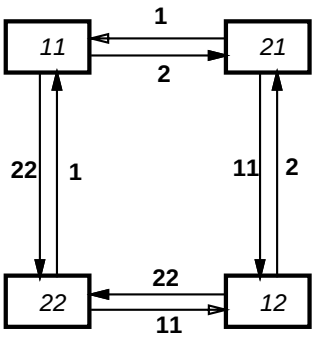
[3] W. B. Jones and W. J. Thron *Continued Fractions- Analytic Theory and Applications*, Encyclopedia of Mathematics and Its Applications, Vol 11, Addison-Wesley, Massachusetts, 1980.

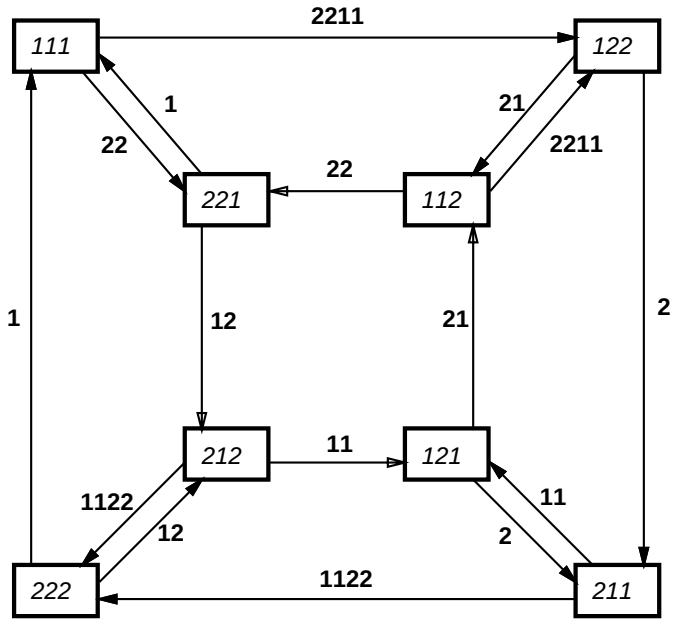
[4] Oskar Perron, *Die Lehre von den Kettenbrüchen*, Chelsea, New York, 1950.

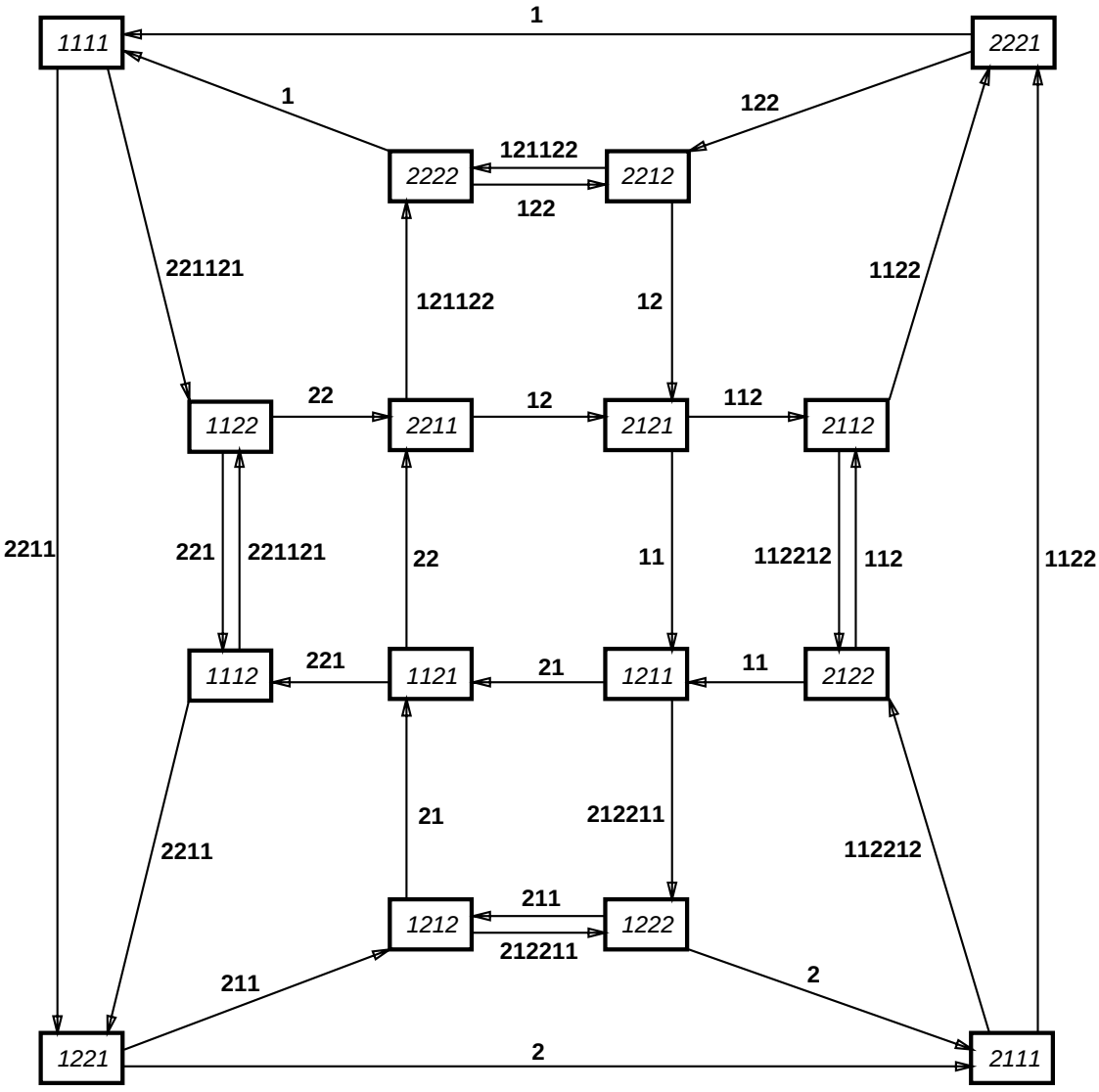
[5] H. S. Wall, *Analytic Theory of Continued Fractions*, Chelsea, New York, 1948.

CECM, Department of Mathematics, Simon Fraser University, Burnaby B.C., Canada, V5A 1S6. Email: `jborwein@cecm.sfu.ca`, `kkchoi@cecm.sfu.ca`


1
1
2
1
2
2


1
11
21
2
22
1
11
2
22
12
11


2211
111
122
1
21
22
2211
22
221
112
1
12
21
2
212
11
121
1122
12
11
2
222
1122
211


1111
2221
2222
2212
1122
2211
2121
2112
1112
1121
1211
2122
1212
1222
1221
2111
1
1
122
121122
122
221121
121122
12
1122
22
12
112
2211
221
221121
22
11
112212
112
1122
221
21
11
2211
21
212211
112212
211
211
212211
2
2

The NumbersWithNames Program

Simon Colton
Mathematical Reasoning Group
Division of Informatics
University of Edinburgh, UK
Email: simonco@dai.ed.ac.uk

Louise Dennis
School of Computer Science
and Information Technology
University of Nottingham, UK
Email: lad@cs.nott.ac.uk

Abstract

We present the NumbersWithNames program which performs data-mining on the Encyclopedia of Integer Sequences to find interesting conjectures in number theory. The program forms conjectures by finding empirical relationships between a sequence chosen by the user and those in the Encyclopedia. Furthermore, it transforms the chosen sequence into another set of sequences about which conjectures can also be formed. Finally, the program prunes and sorts the conjectures so that the most plausible ones are presented first. We describe here the many improvements to the previous Prolog implementation which have enabled us to provide NumbersWithNames as an online program. We also present some new results from using NumbersWithNames, including details of an automated proof plan of a conjecture NumbersWithNames helped to discover.

1 Introduction

The Encyclopedia of Integer Sequences[1] is one of the most useful and popular mathematics resources available on the internet. With the help of many mathematicians, Neil Sloane has collected over 60,000 sequences of integers along with information about them such as definitions, links, computer algebra code, etc. Because of the number and range of sequences in the Encyclopedia, there have been occasions when coincidences arising from its use have led to a connection between two different areas of mathematics (and other sciences) being made. For instance, in [Slo98], Sloane relates how a sequence which arose in connection with a quantization problem was linked via the Encyclopedia with a sequence arising from the study of three-dimensional quasi-crystals.

As part of the HR project [Col01], we have attempted to increase the possibility of such research conjectures being made. The HR program enables this by data-mining the Encyclopedia to find empirical relationships between sequences. This initial approach is detailed in [CBW00], where the emphasis was on producing conjectures about sequences which HR had also invented. For instance, HR invented the concept of integers for which the number of divisors is a prime number, and through data-mining, it also conjectured that numbers where the *sum* of divisors is a prime number have a prime number of divisors — a fact we were able to prove. Further results from this initial approach are given in [Col99].

[1] Available here: http://www.research.att.com/~njas/sequences

The previous Prolog implementation within HR was very basic. We have now changed the emphasis so that the user can choose the sequence about which to form conjectures from any in the Encyclopedia (or indeed, any they care to invent). We have also improved the way in which the program makes conjectures, and made it available as the 'NumbersWithNames' Java program which can be used online at: `http://www.machine-creativity.com/programs/nwn.`

Given a sequence S, chosen by the user, NumbersWithNames performs a four step process:

1. it identifies and invents sequences related to S
2. it makes conjectures about S and the related sequences
3. it prunes any uninteresting conjectures
4. it sorts the conjectures in order of decreasing plausibility

In §2, we detail how NumbersWithNames makes conjectures by finding relationships between the chosen sequence (and transformations of it) and those in the Encyclopedia. In §3, we detail a new measure of plausibility for these conjectures which has been generalised from two previous measures. This measure is used to both prune implausible conjectures and to sort those remaining so that the user can view the most plausible first. In §4, we present some new results from the program, and detail how we have used the $\lambda Clam$ proof planner [RSG98] to find a proof plan for a generalised conjecture suggested by results from NumbersWithNames.

2 Making Conjectures in NumbersWithNames

The Encyclopedia contains many different types of sequence. In particular, there are around 1000 number types, such as prime numbers, even numbers, odd numbers, etc. in the Encyclopedia which are sufficiently important to have been given a name in the mathematical literature, and NumbersWithNames works with these. This design consideration was for various reasons. First, all the sequences are downloaded as part of a Java archive file, so having 1,000 rather than 60,000 to download was preferable. Second, searching through 1,000 sequences repeatedly to find conjectures is possible in an acceptable time limit, but searching through 60,000 repeatedly is not. Third, and most importantly, conjectures about number types can be stated in a natural way, for instance: prime numbers are not multiples of four, or: odd refactorable numbers are square numbers (refactorable numbers are such that the number of divisors is itself a divisor [Col99]).

2.1 Finding Empirical Relationships

For a sequence S, chosen by the user, NumbersWithNames searches through the 1,000 number types, trying to find sequences, T, which are empirically related to S. The relationships it looks for are:

• **Subsequences,** i.e., all members of T that are within the range of S are actually in S. (The range of S is the part of the number line it occupies). For example, suppose S was the perfect numbers (equal to the sum of their proper divisors) and T was the even numbers. All the perfect numbers stored in the Encyclopedia which are in the range of the even numbers (in the Encyclopedia the range of the even

numbers is 0 to 120) are themselves even. Hence NumbersWithNames would make the conjecture that all perfect numbers are even (a well known open conjecture).

• **Supersequences,** i.e., all members of S that are within the range of T are actually in T.

• **Disjoint sequences,** i.e., no member of S is a member of T. For example, none of the entries in the perfect numbers sequence are found in the odd number sequence, so the program conjectures that there are no odd perfect numbers.

• **Moonshine sequences,** i.e., there is a large integer (greater than 10,000) which is found in both S and T. NumbersWithNames notices if any large integer in S matches to within $\pm$ 2 with one in T. This is inspired by the 'monstrous moonshine' theorem relating group theory and elliptic modular functions [CN79]. This theorem — the proof of which gained Richard Borcherds a Fields Medal — was discovered when the numbers 196883 and 196884 were found in seemingly distinct areas of mathematics.

2.2 Transforming the Given Sequence

Often, there may be a very interesting conjecture about a sequence closely related to the sequence of interest. As a trivial example, the conjecture: "all prime numbers are odd" is not true. However, the conjecture: "all prime numbers except 2 are odd" is true. Hence, transforming the sequence into a set of closely related ones may produce more interesting conjectures. To find concepts related to a chosen sequence S, NumbersWithNames first looks for ones with similar names in the database. For instance, if S was prime numbers, it would find sequences such as Mersenne prime numbers, Mills prime numbers, additive prime numbers and so on, and make conjectures about those also.

Following this, NumbersWithNames invents concepts by both transforming S and by combining it with others. The transformations are limited at the moment to:

• **Adding one and taking one from the sequence,** e.g., the sequences of primes-plus-one: $3, 4, 6, 8, \ldots$

• **Monster-barring,** e.g., the sequences of primes-except-2: $3, 5, 7, 11, \ldots$

• **Finding difference sequences,** i.e., taking the differences between consecutive terms in the sequence.

• **Finding the binomial sequence,** i.e., taking the first term of the difference sequence, the first term of the difference-of-differences sequence and so on (known as the binomial transformation).

NumbersWithNames also combines S with those which have been assigned the keyword: "core" in the Encyclopedia of Integer Sequences[2]. It has two ways to combine a pair of sequences:

• **Conjunction,** e.g., combining the sequences of odd numbers and prime numbers into the sequence: "odd prime numbers".

• **Indexing,** e.g., combining the sequences of odd numbers and prime numbers by taking the prime numbers which have an index in the sequence which is an odd number, i.e., p_1, p_3, p_5, etc. where p_i is the i-th prime number.

The user decides the number of additional sequences the program introduces. They are given three options: the first invents no additional sequences, the second invents all additional sequences except those produced by combination with the core sequences, and the third invents all additional sequences.

[2] Sequences in the Encyclopedia all have associated keywords, including "core", "nice" and "hard".

3 Pruning and Sorting Conjectures

NumbersWithNames employs pruning techniques to reduce the number of uninteresting and trivial conjectures produced. Firstly, the program discards conjectures which follow from the definitions of the sequences involved. For instance, it throws away the conjecture that odd prime numbers are prime numbers, because, using the names of these sequences, it assumes that odd prime numbers are, by definition, a specialisation of prime numbers. Secondly, it discards a conjecture if it has already made a stronger conjecture. For instance, it throws away conjectures such as: "e-perfect numbers are refactorable numbers" if it has already made (or makes later) the conjecture: "e-perfect numbers are *even* refactorable numbers". This is because the first conjecture is subsumed by the second, stronger, conjecture. This functionality is similar to the 'echo' heuristic employed by the Graffiti program (see §5). Finally, the user is able to prune the conjectures further, by supplying text which must be (or must not be) in the definition/keywords of the sequences in the conjecture.

Even after pruning, the program often produces a plethora of conjectures. Therefore, we enabled it to present the most plausible ones first. The plausibility is calculated as the probability that the conjecture is not a coincidence. Given sequence S with terms $s_1, \ldots, s_k$ and sequence T, the plausibility of the conjecture that T is a subsequence of S is calculated as:

$$1 - \left(\frac{k}{s_k - s_1}\right)^X$$

where X is the number of terms of T in the range of S. For example, suppose S was the powers of two, which has these terms in the Encyclopedia:

$$1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144,$$
$$524288, 1048576, 2097152, 4194304, 8388608, 16777216, 33554432, 67108864, 134217728,$$
$$268435456, 536870912, 1073741824, 2147483648, 4294967296, 8589934592$$

Further suppose that NumbersWithNames conjectures that superperfect[3] numbers:

$$2, 4, 16, 64, 4096, 65536, 262144, 1073741824, 1152921504606846976,$$

are powers of two. There are 34 powers of two recorded in the Encyclopedia, ranging over the numbers 1 to 8589934592, so the probability of a number between 1 and 8589934592 being a power of two is $(34/8589934592) \approx 0.000000004$. We do not know whether 1152921504606846976 is a power of two or not, because the sequence of powers of two stops before it gets that far. However, the other 8 superperfect numbers certainly are powers of two, and the probability of this happening by coincidence is therefore 0.000000004^8, which is approximately 6×10^{-68}. Therefore, the conjecture that superperfect numbers are powers of two is extremely plausible, because the probability of it happening as a coincidence is very small indeed. NumbersWithNames calculates this plausibility measure for the supersequence and subsequence conjectures and a similar one for the disjoint conjectures. It then presents the conjectures in decreasing plausibility. This measure supersedes two previous measures, namely the 'term overlap' (number of

[3] Superperfect numbers are integers n such that $\sigma(\sigma(n)) = 2n$, with $\sigma(n)$ defined as the sum of the divisors of n.

terms shared by the sequence and sub-sequence) and 'range overlap' (proportion of the number line occupied by either sequence which is actually occupied by both), which are described in [CBW00]. We found that the plausibility measure, in addition to being easier to use than the two previous measures, also highlighted more interesting conjectures. For instance, if three small terms overlap in a sequence and subsequence, the conjecture scores 3 for term overlap. If however, three large terms overlap — a potentially more significant result — the conjecture still only scores 3 for term overlap. In contrast, the plausibility measure is low for the small overlapping terms and high for the large overlapping terms.

4 Results

We chose 10 sequences at random and used all the functionality of NumbersWithNames to form as many conjectures about each one as possible. The average time to complete the task using the Java Virtual Machine (JVM) in Internet Explorer version 5 on a 1000Mhz laptop, was around 90 seconds per sequence. As this is not an excessive time to wait, we have not investigated any more sophisticated algorithms for finding conjectures. There is, however, a problem with the JVM implemented in versions of Netscape, and we have experienced between 4 and 5 times slower execution with Netscape. This appears to be a problem that Sun Microsystems are aware of.

NumbersWithNames, while relatively new as an online program, is in fact the conclusion of a project which has been ongoing for a number of years, namely using automated techniques within the HR program to find conjectures about sequences in the Encyclopedia of Integer Sequences. There have been many interesting results from this analysis along the way. Some interesting theorems HR discovered (which we proved ourselves) include:

• If the sum of divisors of an integer is prime, then the number of divisors is prime.

• Refactorable numbers (as described in §2 above) are congruent to 0, 1, 2 or 4 mod 8.

• Every even[4] perfect number can be written in the form $lcm(a, \sigma(a))$ and in the form $\phi(b)(\sigma(b) - b)$ for some a and b [where $lcm(x, y)$ is the lowest common multiple of x and y, $\sigma(a)$ is the sum of divisors of a and $\phi(b)$ is the number of integers less than or equal to b which are co-prime to it].

Also, there are many new conjectures produced by NumbersWithNames which are awaiting investigation (i.e., a proof or disproof), for example:

• e-perfect numbers (where the sum of the exponential divisors of n equals 2n) are even refactorable numbers (where the number of divisors is itself a divisor).

Also, NumbersWithNames has made some moonshine conjectures, such as pointing out the unlikely coincidence that:

• 1073741823 is a Stirling number, 1073741824 is a superperfect number (and power of two), and 1073741825 is a Jacobsthal-Lucas number.

Despite encouragement by ourselves within the mathematical community, we not encountered a mathematician using NumbersWithNames as a research tool. We present three successful investigations below,

[4] Note that the conjecture as to whether there exists an odd perfect number is still open.

in the hope that such success will encourage researchers to use NumbersWithNames to supply conjectures about sequences they are interested in.

4.1 Pernicious Numbers

Jeremy Gow invented the notion of pernicious numbers, namely integers n where the number of 1s in the binary representation of n is a prime number. This continues in the tradition of odious numbers (odd number of 1s) and evil numbers (even number of 1s). We wished to find something of interest about these numbers, but with the previous implementation of the data-mining within HR, we only discovered that powers of two are *not* pernicious. This is trivially true, because powers of two in binary form are a one followed by zeros.

However, when we looked for conjectures about pernicious numbers with NumbersWithNames later, it produced 165 subsequence conjectures. We pruned these by keeping only the 'core' sequences conjectured to be a subsequence of pernicious numbers. This reduced the number to seven and only one of these was true (we found counterexamples to the others using the GAP computer algebra system [Gap00]). The true conjecture was very interesting, though:

Perfect numbers are pernicious.

Perfect numbers — equal to the sum of their proper divisors — are of great interest in number theory, and any result about their nature may be important. The reason this conjecture was not made previously by HR is because we always used a term overlap minimum of four or more for the conjectures. That is, we instructed HR to discard any subsequence conjectures where the two sequences shared fewer than four terms. The perfect numbers are: $6, 28, 496, 8128, \ldots$ and the largest pernicious number in the Encyclopedia is 100, hence the conjecture that perfect numbers are pernicious was discarded, because the empirical evidence for it amounted to only two terms: 6 and 28. In NumbersWithNames, however, this conjecture was given a plausibility of 38%. Hence, as only conjectures with 0% plausibility are discarded, the conjecture was observed in the output.

It was not obvious to us that perfect numbers — defined in terms of the sum of their divisors — should show any special characteristics when written in binary. On writing the perfect numbers in binary, we noticed the following pattern:

$$6 = 110, \quad 28 = 11100, \quad 496 = 111110000, \quad 8128 = 1111111000000$$

To cross-check, we later used NumbersWithNames to provide conjectures about perfect numbers, and it conjectured not only that perfect numbers are pernicious, but also that they are nialpdrome numbers of type 2 (such that, in binary, they are 1s followed by 0s). Hence, taken together, NumbersWithNames had made the conjecture that perfect numbers, when written in binary, comprise a prime number of 1s followed by zeros, and we see that in the examples above.

It turns out to be fairly easy to prove this theorem, given a result found in Hardy and Wright's standard number theory text [HW38], that even perfect numbers are of the form: $2^{n-1}(2^n - 1)$ where $2^n - 1$ is a prime (called a Mersenne prime). It is fairly easy to show that multiplying a number of the

form 2^{n-1} with a number of the form $2^n - 1$ (with n the same in each), produces a number which, when written in binary, is n ones followed by $n-1$ zeros. On presenting this to an 'integer sequence fans' mailing list, the overall impression was that, while it was a pleasing result they had not seen before, because it followed easily from Hardy and Wright's theorem, it was just an example of how related the concepts in number theory are, and was unlikely to be of importance. This should not detract, however, from the fact that NumbersWithNames made us aware of this theorem, which added to the value of pernicious numbers, and that we are unlikely to have found it ourselves.

4.2 Zeitz Numbers

NumbersWithNames can provide insight which may help solve problems. As an illustrative example, we looked at a problem posed in [Zei99]:

• Show that numbers of the form $n(n+1)(n+2)(n+3)$ are never square numbers.

Zeitz suggests 'plugging and chugging', i.e., putting numbers into the formula and seeing if the results suggest anything which may help solve the problem. We used NumbersWithNames to help with the discovery part. Putting $n = 1, 2, 3$ and 4 into the formula above resulted in: $24, 120, 360$ and 840. We then added this as a new sequence to NumbersWithNames (which it is possible to do online, without having to recompile the program), and called this number type: 'zeitz numbers'. Then we asked for conjectures about this sequence.

The first four conjectures, sorted in terms of plausibility, about zeitz numbers were:

1. zeitz numbers are highly composite numbers
2. zeitz numbers are super-abundant numbers
3. zeitz numbers are minimal(1) numbers
4. zeitz-plus-one numbers are square numbers

The first three conjectures did not help us solve the problem, but the fourth one states that adding one to zeitz numbers produces square numbers. This implies that zeitz numbers can never be square numbers, because square numbers are never 1 apart on the number line. Thus, if the conjecture made by NumbersWithNames is true, then the problem is solved. In [Zei99], Zeitz says that making this conjecture is the most important part of solving the problem, and the rest follows easily from this Eureka step, namely showing that $n(n+1)(n+2)(n+3)$ can be written as $(n^2+3n+1)^2 - 1$.

4.3 Sqrt(n)-Rough Numbers

To encourage the 'integer sequence fans' to use NumbersWithNames, we have periodically used it to form some conjectures about sequences that were currently being discussed on that mailing list. On one occasion, discussion centred around a sequence invented by Knuth and Greene [GK90]: integers where the largest prime factor is less than the square root. For example, 8 is the first such number, because the largest prime factor is 2, which is less than $\sqrt{8}$. These are called sqrt(n)-rough numbers. NumbersWithNames made a series of conjectures that interested us, including:

• centred square numbers (of the form $2n(n+1)+1$) are sqrt(n)-rough-plus-one numbers

- hex numbers (of the form $3n(n+1)+1$) are sqrt(n)-rough-plus-one numbers
- star numbers (of the form $6n(n+1)+1$) are sqrt(n)-rough-plus-one numbers

All of these conjectures had plausibility 99% or 100%, and we note that if NumbersWithNames hadn't invented the sequence of sqrt(n)-rough-plus-one (by adding one to the original sequence), this series of conjectures would not have been brought to our attention. This highlights the need for the concept formation part of NumbersWithNames. We generalised this result to the following:

Given any two integers k and n such that $n^2 > k > 1$,
then the number $kn(n+1)$ will be a sqrt(n)-rough number.

We proved this result, and found that, unusually, we did not have to appeal to any results from number theory other than some simple facts about inequalities and square roots.

λClam [RSG98] is a higher-order proof planning system. It is a descendent of the *Clam* [BvHHS90] series, and is specialised for proof by induction, but is also intended to allow the rapid prototyping of automated theorem proving strategies. *λClam* works by using depth-first planning with *proof methods*. Each node in the search tree is a subgoal under consideration at that point[5]. The planner checks the preconditions for the available proof methods at each node and applies those whose preconditions succeed to create the child nodes. The plan produced is then a record of the sequence of method applications that lead to a trivial subgoal.

Proof methods are intended to act as partial tactic specifications for tactics in some object-level theorem prover. In *Clam* this was the *Oyster* constructive type-theory system, which was based on Nuprl. At present, *λClam* has no associated theorem prover. However the plans produced by *λClam* are at an equivalent level to "pen and paper" proofs produced by mathematicians. *λClam*'s proof methods are believed to be sound although they are not currently reducible to sequences of inference rule applications in some logic. Thus a *λClam* plan of the conjecture above would represent an equivalent guarantee of correctness to that provided by the hand proof already in existence. Proof method applications are governed by their preconditions (which may be either legal or heuristic in nature) and by a *proof strategy* which restricts the possible proof methods available depending on the progress through the proof. For instance, when involved in rewriting a goal using a selection of definitions and lemmas, we generally wish to attempt to rewrite as much as possible (i.e. simplify the goal as much as possible) by applying our rewriting method exhaustively before considering other procedures such as checking for tautologies.

λClam is, at present, ill-equipped to deal with problems which rely on a large body of previous results for their proof, but is better able to deal with problems where the proof follows primarily from the definitions of the concepts involved. As this was the case with the sqrt(n)-rough conjecture, we decided to investigate whether *λClam* could prove the result. Our aim was to show that *λClam* could be incorporated into the process to provide — for simple conjectures at least — this sort of proof automatically. Combination of systems such as *λClam* and NumbersWithNames are important both for the advancement of Artificial Intelligence and in order to attract mathematicians to use automated

[5] More accurately each node is the partial plan at that point but viewing this as the current subgoal is sufficient for this application.

tools, i.e., if researchers are supplied not with conjectures, but rather proved theorems, then this might encourage them to invest some time applying and even developing such automated techniques.

Proof methods and proof strategies are devised by observing common patterns in families of proofs. They are intended to represent generic mathematical processes applicable across a range of problems. $\lambda Clam$ had not previously been applied to problems like the theorem about sqrt(n)-rough numbers shown above, and so we had to create a new proof strategy. The original hand proof of the result was used as a guide to the proof procedures involved but abstracted to a number of generic steps. These were perceived to be the use of rewriting with definitions and lemmas and the use of reasoning based on transitivity. Rewriting is a standard procedure, and methods supporting this were already available in $\lambda Clam$. Reasoning about transitivity had not, however, been tackled previously by the system.

We extended $\lambda Clam$ with a simple proof method for reasoning about transitivity. The proof method's preconditions were as follows:

- We wish to prove $H \vdash A < B$ where H is a list of hypotheses and A and B are terms.
- $C < B'$ appears in the hypothesis list, H, or $C < B'$ is a known lemma and there exists a substitution σ on the free variables of B and B' such that $\sigma(B) = \sigma(B')$.

If these preconditions succeeded then the planner would add the subgoal $\sigma(H) \vdash \sigma(A) < \sigma(C)$ to the search tree. There is an equivalent case for replacing A by a new value. We prototyped a proof strategy which attempted to repeatedly apply symbolic evaluation (rewriting) and this transitivity method, backtracking where necessary. In order to make the search tractable, we had to modify the transitivity method so that it did not attempt to prove $H \vdash A < 1$ at any point. An expert $\lambda Clam$ user was able to put together the transitivity method and the prototype proof strategy (interleave rewriting and transitivity reasoning exhaustively) in less than 2 days of work.

With this machinery, $\lambda Clam$ automatically found a plan for the conjecture:

$$\forall k.\, \forall n.\, ((1 < \sqrt{k}) \wedge (k < n^2) \wedge (\sqrt{k} < n)) \rightarrow sr(k * (n * (n + 1)))$$

(where $sr(x)$ means that x is sqrt(n)-rough). A number of standard lemmas about squares etc. were assumed to make this possible. Note that the condition $1 < \sqrt{k}$ explicitly rules out those cases where $k = 2$ or $k = 3$, where a different style of reasoning, based on substituting values into the theorem would have been required.

Ideally, we would like to test our proof strategy on a number of conjectures produced by NumbersWithNames, to see if it is sufficiently general to automatically plan a range of problems. We have not attempted this, as we believe the strategy to be limited in its abilities. However, we are currently developing techniques in $\lambda Clam$ for the rapid combination of decision procedure techniques [JB01]. The proof for the theorem planned by $\lambda Clam$ requires a combination of linear arithmetic with some rewriting. This is a major application of the decision procedure work we are involved in and we hope that a generic strategy for performing this task will be available in $\lambda Clam$ shortly. We believe that, while our proof strategy provided a proof of concept that $\lambda Clam$ could be used for conjectures from Number-

sWithNames, it would be more robust in the long term to use a proof strategy based on our decision procedure work.

Clearly, the nature of proof strategy development requires a family of conjectures with proofs which may be examined for common patterns. There is work in the automated development of proof strategies [JKB00] which may, in future, allow proof planning systems to make a contribution even in new domains. The theorem proving work reported here is very preliminary in nature, but we feel that the speed (relative to other proof strategy developments) with which a proof strategy for a NumbersWithNames conjecture could be developed shows that $\lambda Clam$ could be usefully used when conjectures are being formed in well-understood domains.

5 Related Work

Compared to systems for proving theorems automatically, there have been relatively few programs designed to automatically make research conjectures of real interest to mathematicians. The Graffiti program [Faj88] has, to the best of our knowledge, been the only program which has successfully produced many conjectures of sufficient difficulty and importance to come to the attention of research mathematicians. Graffiti has been designed by mathematician Siemion Fajtlowicz to make conjectures of a numerical nature in graph theory. Given a set of well known graph theory invariants, such as the diameter, independence number, rank and chromatic number, Graffiti uses a database of graphs to empirically check whether one sum of invariants is less than another sum of invariants. If a conjecture passes the empirical test and Fajtlowicz cannot prove it easily, he forwards it to interested graph theorists.

As an example, conjecture 18 produced by Graffiti stated that, for any graph G:

$$\begin{array}{ccc} \text{chromatic\_number}(G) & & \text{maximum\_degree}(G) \\ + & \leq & + \\ \text{radius}(G) & & \text{frequency\_of\_maximum\_degree}(G) \end{array}$$

This was passed to some graph theorists, one of whom found a counterexample. These types of conjecture are of substantial interest to graph theorists because they are easy to understand, yet they often provide a significant challenge to resolve. The conjectures are also useful because calculating invariants is often expensive and bounds on sums of invariants may help bring computation time down.

In terms of adding to mathematical knowledge, Graffiti has been extremely successful. The conjectures it has produced have attracted the attention of scores of mathematicians, including many luminaries from the world of graph theory. There are over 60 graph theory papers published which investigate Graffiti's conjectures. Graffiti owes some of its success to the fact that the inequality conjectures it makes are of a difficult and important type, and that Fajtlowicz himself uses Graffiti and prunes and disseminates the results to many interested parties. In contrast to NumbersWithNames, to our knowledge, Graffiti is not available for mathematicians to experiment with themselves.

6 Conclusions and Future Work

We have described the NumbersWithNames program which produces interesting conjectures about sequences of integers in number theory. We have described the numerous advances over the version in HR, including a generalised plausibility measure, the ability to transform the given sequence into related ones to find conjectures about, and the ability to form moonshine conjectures. There are many more improvements we hope to make, including additional transformations of sequences, and enabling NumbersWithNames to interact with computer algebra systems to further empirically check the conjectures it makes using the computer algebra code supplied with some of the sequences in the Encyclopedia.

We have also reported new results from this data-mining approach. In contrast to Graffiti, however, where the main user is interested in the results, we are not number theorists, and hence, not only do we have less interest in the results, we are also not in a position to assess the implications, applications or importance of the conjectures NumbersWithNames produces. For this reason, we have made the program available to run online in the hope that research mathematicians and recreational mathematicians will use it. Hence, the next stage of the project is to attract mathematicians to work both with the program and with us. We have started this process by making conjectures about some sequences being discussed on the sequence fans mailing list, and hope to continue this approach by targeting various researchers with conjectures about sequences of particular interest to them. Finally, we have described how $\lambda Clam$ has been used to plan a proof for a generalised conjecture which arose from a series of conjectures made by NumbersWithNames, and we hope to pursue this interaction. In particular, we intend to use the decision procedure techniques soon to be available.

In a seminal 1958 paper [SN58], Newell and Simon made the prediction that:

'Within ten years a digital computer will discover and prove an important mathematical theorem.'

In our opinion — while some important mathematical theorems have been proved by automated means, for example the Robbins algebra problem [McC97] — for various reasons this prediction has not yet come true. Furthermore, only through interaction between conjecture making programs such as NumbersWithNames, HR and Graffiti, and theorem provers/planners such as $\lambda Clam$, will Newell and Simon's prediction be fulfilled. We cannot even claim that NumbersWithNames and $\lambda Clam$ have discovered and proved a theorem autonomously, not to mention an important theorem. However, this is a goal of our project, and one which we believe is within the grasp of modern computational techniques.

Acknowledgments

Simon Colton is also affiliated with the Department of Computer Science at the University of York, UK. We would like to thank the anonymous reviewers whose comments helped improve the final version of this paper. This work has been supported by EPSRC grants GR/M98012 and GR/M45030.

References

[BvHHS90] A Bundy, F van Harmelen, C Horn, and A Smaill. The Oyster-Clam system. In *Proceedings of CADE-10*, pages 647–648. Springer-Verlag, 1990.

[CBW00] S Colton, A Bundy, and T Walsh. Automatic invention of integer sequences. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 558–563, 2000.

[CN79] J Conway and S Norton. Monstrous moonshine. *Bulletin of the London Mathematical Society*, 11:308 – 339, 1979.

[Col99] S Colton. Refactorable numbers - a machine invention. *Journal of Integer Sequences,* `http://www.research.att.com/~njas/sequences/JIS`, 2, 1999.

[Col01] S Colton. *Automated Theory Formation in Pure Mathematics.* PhD thesis, Division of Informatics, University of Edinburgh, 2001.

[Faj88] S Fajtlowicz. On conjectures of Graffiti. *Discrete Mathematics 72*, 23:113–118, 1988.

[Gap00] Gap. *GAP Reference Manual.* The GAP Group, School of Mathematical and Computational Sciences, University of St. Andrews, 2000.

[GK90] D Greene and D Knuth. *Mathematics for the Analysis of Algorithms.* Birkhäuser, 1990.

[HW38] G Hardy and E Wright. *The Theory of Numbers.* Oxford University Press, 1938.

[JB01] P Janičić and A Bundy. A general setting for combining and integrating decision procedures into theorem provers. *Journal of Automated Reasoning*, 2001. To appear.

[JKB00] M Jamnik, M Kerber, and C Benzmüller. Towards learning new methods in proof planning. In *Proceedings of the 2000 Calculemus Symposium: Systems for Integrated Computation and Deduction*, 2000.

[McC97] W McCune. Solution of the Robbins problem. *Journal of Automated Reasoning*, 19(3):263–276, 1997.

[RSG98] J Richardson, A Smaill, and I Green. System description: proof planning in higher-order logic with λCLAM. In *Proceedings of CADE-15*, pages 129–133. Springer-Verlag, 1998.

[Slo98] N J A Sloane. My favorite integer sequences. In *Proceedings of the International Conference on Sequences and Applications*, 1998.

[SN58] H Simon and A Newell. Heuristic problem solving: The next advance in operations research. *Operations Research*, 6(1):1–10, 1958.

[Zei99] P Zeitz. *The Art and Craft of Problem Solving.* John Wiley and Sons, 1999.

A classification of plane and planar 2-trees

Gilbert Labelle, Cédric Lamathe, Pierre Leroux*
LaCIM, Département de Mathématiques, UQÀM

January 31, 2002

Abstract

We present new functional equations for the species of plane and of planar (in the sense of Harary and Palmer, 1973) 2-trees and some associated pointed species. We then deduce the explicit molecular expansion of these species, *i.e.* a classification of their structures according to their stabilizers. There result explicit formulas in terms of Catalan numbers for their associated generating series, including the asymmetry index series. This work is closely related to the enumeration of polyene hydrocarbons of molecular formula C_nH_{n+2}.

1 Introduction

We define recursively the class $\mathcal{A}$ of *2-dimensional trees* (in brief 2-*trees*) as the smallest class of simple graphs such that

1. the single edge is in $\mathcal{A}$,
2. if a simple graph G has a vertex x of degree 2 whose neighbors are adjacent and such that $G - x$ is in $\mathcal{A}$, then G is in $\mathcal{A}$.

One can see that a 2-tree is essentially composed of triangles (complete graph on 3 vertices) glued together along edges in a tree-like fashion.

Note that all 2-trees are planar simple graphs. However, by a *planar* 2-*tree*, we mean here a 2-tree admitting an embedding in the plane in such a way that all faces (except possibly the outer face) are triangles, and we call *plane* 2-*tree* a 2-tree equipped with such an embedding. This terminology agrees with Harary and Palmer [8]. In Figure 3, we show a correspondence between plane 2-trees and (unrooted) triangulations of polygons in the plane which is also a correspondence between planar 2-trees and (unrooted) triangulations of polygons in space (no orientation), also known as triangulations of the disc, see [4] . Figure 1 gives an example of an unlabelled and a triangle-labelled planar 2-tree, Figure 2 shows two different plane 2-trees which are in fact the same planar 2-tree since they are isomorphic simple graphs. We point out the work of Palmer and Read, [15], who enumerate plane embeddings of 2-trees without any condition on the faces, and which they also call plane 2-trees. Planar 2-trees (in our sense) are closely related to acyclic polyene hydro-carbons of molecular formula C_nH_{n+2} (planar trees in the hexagonal lattice); see [5].

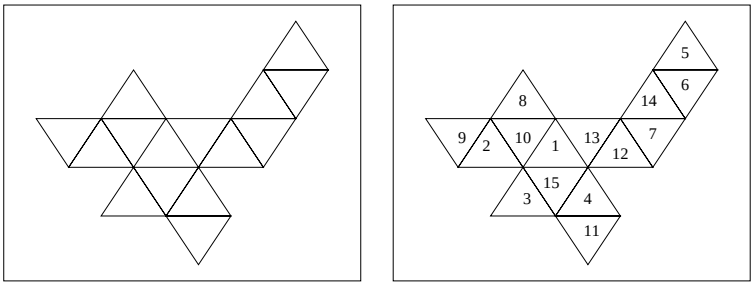


Figure 1: An unlabelled plane 2-tree and one of its labellings

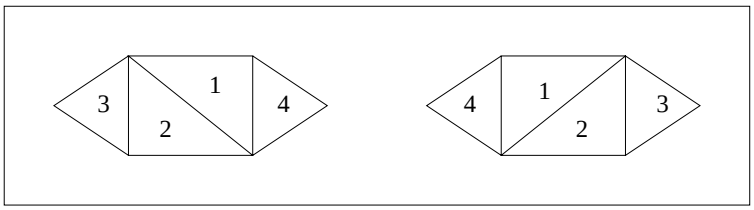


Figure 2: Two different plane 2-trees, one planar 2-tree

We follow the approach of Fowler and al. in [6, 7] for general 2-trees. However, we go further here, giving explicitly the molecular expansion of plane and planar 2-trees, which could not be done in the general case. This is a stronger result than simple labelled and unlabelled enumeration since it gives a classification of the different structures according to stabilizers. For instance, it permits us to have an explicit enumeration of the symmetric and asymmetric parts of these species. Moreover, we obtain closed formulas for all coefficients appearing in these expansions.

To derive these results we use functional equations in the context of the combinatorial theory of species and deduce the molecular expansions and all the associated series. In the following, we label 2-trees at triangles and we denote by X the species of singletons, *i.e.* of simple triangles. Recall that a combinatorial species is a class of finite labelled structures, closed under relabelling along bijections. To each species F we associate series : $F(x)$, the exponential generating series of labelled structures; $\widetilde{F}(x)$, the ordinary generating series of unlabelled structures; $\overline{F}(x)$, the generating series of unlabelled asymmetric structures; Z_F and Γ_F, the cycle and asymmetry index series. The usual shapes of these series for any species F are as follows

$$F(x) = \sum_{n\geq 0} f_n \frac{x^n}{n!}, \tag{1}$$

$$\widetilde{F}(x) = \sum_{n\geq 0} \widetilde{f}_n x^n, \quad \overline{F}(x) = \sum_{n\geq 0} \overline{f}_n x^n, \tag{2}$$

$$Z_F(x_1, x_2, \ldots) = \sum_{n_1,n_2,\ldots} f_{n_1,n_2,\ldots} \frac{x_1^{n_1} x_2^{n_2} \ldots}{1^{n_1} n_1! 2^{n_2} n_2! \ldots}, \tag{3}$$

$$\Gamma_F(x_1, x_2, \ldots) = \sum_{n_1,n_2,\ldots} f^*_{n_1,n_2,\ldots} \frac{x_1^{n_1} x_2^{n_2} \ldots}{1^{n_1} n_1! 2^{n_2} n_2! \ldots}, \tag{4}$$

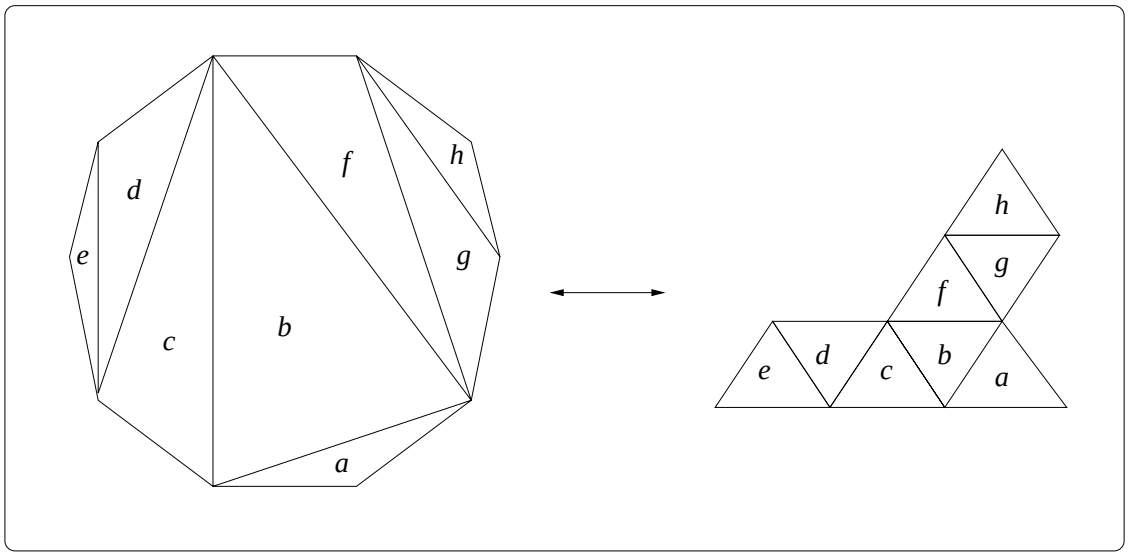


Figure 3: Correspondence between triangulations of a polygon and plane 2-trees

where f_n, $\widetilde{f}_n$ and $\overline{f}_n$ are the numbers of labelled, unlabelled and unlabelled asymmetric F-structures respectively, over an n-element set, and $f_{n_1,n_2,\ldots}$ is the number of F-structures left fixed under a given permutation of cycle type $1^{n_1}2^{n_2}\ldots$. For a definition of the asymmetry index series, see [3].

To illustrate the notion of molecular expansion, we give here the first few terms of this decomposition for the species a_π of plane 2-trees (Eq. (5) and Figure 4) and a_{p} of planar 2-trees (Eq. (6) and Figure 5). As usual, E_n denotes the species of n-element sets and C_3, of 3-element (oriented) cycles. For complete explicit expansions see Theorem 7 for plane 2-trees and Theorem 12 for planar 2-trees.

$$a_\pi = a_\pi(X) = 1 + X + E_2(X) + X^3 + XC_3(X) + 2E_2(X^2) + X^4 + 6X^5 + \cdots \qquad (5)$$

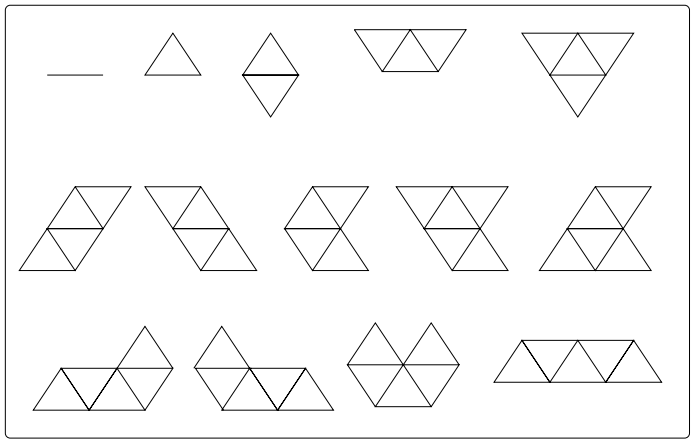


Figure 4: First terms of the molecular expansion of the species a_π of plane 2-trees

$$a_{\rm p} = a_{\rm p}(X) = 1 + X + E_2(X) + XE_2(X) + XE_3(X) + 2E_2(X^2) + 2X^5 + 2XE_2(X^2)$$
$$+X^2E_2(X^2) + \cdots + P_4^{bic}(X,X) + \cdots + XC_3(X^2) + \cdots + XP_6^{bic}(X,X) + \cdots . \qquad (6)$$

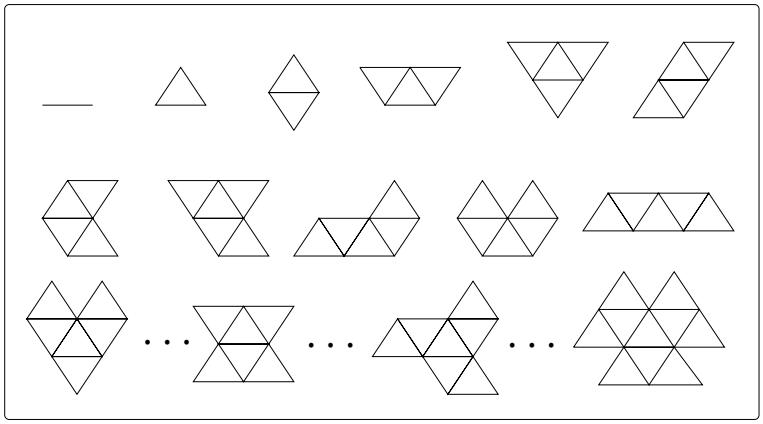


Figure 5: First terms of the molecular expansion of the species $a_{\rm p}$ of planar 2-trees

The expansion of $a_{\rm p}$ involves species $P_4^{bic}(X,Y)$ and $P_6^{bic}(X,Y)$ that are described in Section 2. They are two-sort variants of the species of P_{2n}^{bic} introduced by J. Labelle in [14].

In this paper, we call *degree* of an edge of a 2-tree, the number (less than or equal to 2) of triangles to which it belongs. Let us introduce the auxiliary species A which can be defined as follows:

- A represents the species of plane 2-trees pointed at an external edge, *i.e.* an edge of degree at most 1,
- A is isomorphic to the species of planar 2-trees pointed at an external edge equipped with an orientation,
- A is characterized by the functional equation

$$A = 1 + XA^2 , \qquad (7)$$

illustrated in Figure 6.

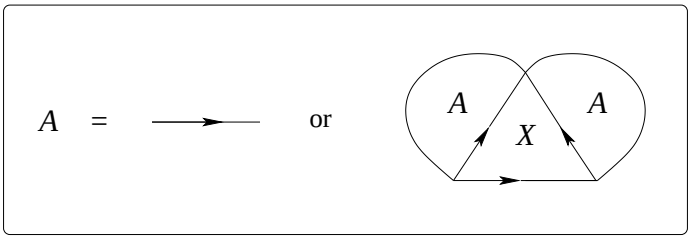


Figure 6: $A = 1 + XA^2$

Note that the species A can also be viewed as the species of rooted triangulations of polygons. This species is fundamental for the following and we will use it several times. We can see that it is asymmetric, *i.e.* the automorphism group of each of its structures is trivial; thus the molecular expansion and the associated series have the same coefficients in their expression. As expected, these coefficients are the Catalan numbers.

Proposition 1. The molecular expansion of the species $A = A(X)$ is

$$A(X) = \sum_{n\in\mathbb{N}} \mathbf{c}_n X^n, \tag{8}$$

where $\mathbf{c}_n = \frac{1}{n+1}\binom{2n}{n}$ (*Catalan numbers*). More generally, if $A^k(X) = \sum_{n\in\mathbb{N}} \mathbf{c}_n^{(k)} X^n$, $k \geq 1$, then

$$\begin{aligned} \mathbf{c}_n^{(k)} &= \sum_{i=0}^{\lfloor\frac{k-1}{2}\rfloor} (-1)^i \binom{k-1-i}{i} \mathbf{c}_{n+k-1-i}, & (9)\\ &= \frac{k}{n}\binom{2n-1+k}{n-1}. & (10) \end{aligned}$$

Proof. The formula for $\mathbf{c}_n$ follows directly from a simple application of the Lagrange inversion on the relation (7). It can also be computed by expanding in series the algebraic solution $A(X) = (1-\sqrt{1-4X})/2X$ of (7). For the $\mathbf{c}_n^{(k)}$, we work with the unlabelled generating series. First, we remark that

$$A^k(x) = \sum_{i=0}^{\lfloor\frac{k-1}{2}\rfloor} (-1)^i \binom{k-1-i}{i} \frac{A(x)}{x^{k-1-i}} + \sum_{i=o}^{\lfloor\frac{k-2}{2}\rfloor} (-1)^{i+1} \binom{k-2-i}{i} \frac{1}{x^{k-1-i}}, \tag{11}$$

where $\lfloor\cdot\rfloor$ represents the floor function. This formula is easily shown by recurrence on k distinguishing two cases depending on the parity of k and using the fact that $A^2(x) = \frac{1}{x}(A(x)-1)$, which follows from (7). Next, extracting the coefficient of x^n in this expression gives the result. The second expression for $\mathbf{c}_n^{(k)}$ is obtained by a simple application of the composite Lagrange inversion formula on equation (7). ∎

For instance, for k from 1 up to 6, we have

$$\begin{aligned} \mathbf{c}_n^{(1)} &= \mathbf{c}_n = \frac{1}{n}\binom{2n}{n-1}, \\ \mathbf{c}_n^{(2)} &= \mathbf{c}_{n+1} = \frac{2}{n}\binom{2n+1}{n-1}, \\ \mathbf{c}_n^{(3)} &= \mathbf{c}_{n+2} - \mathbf{c}_{n+1} = \frac{3}{n}\binom{2n+2}{n-1}, \\ \mathbf{c}_n^{(4)} &= \mathbf{c}_{n+3} - 2\mathbf{c}_{n+2} = \frac{4}{n}\binom{2n+3}{n-1}, \\ \mathbf{c}_n^{(5)} &= \mathbf{c}_{n+4} - 3\mathbf{c}_{n+3} + \mathbf{c}_{n+2} = \frac{5}{n}\binom{2n+4}{n-1}, \\ \mathbf{c}_n^{(6)} &= \mathbf{c}_{n+5} - 4\mathbf{c}_{n+4} + 3\mathbf{c}_{n+3} = \frac{6}{n}\binom{2n+5}{n-1}. \end{aligned} \tag{12}$$

In order to lighten notations, we slighty extend the definition of the Catalan numbers as follows:

$$\mathbf{c}_n = \frac{1}{n+1}\binom{2n}{n}\chi(n \in \mathbb{N}). \tag{13}$$

In other words, $\mathbf{c}_n$ is the ususal Catalan number if n is a nonnegative integer, and 0 otherwise.

We will use two dissymmetry formulas, analogous to the case of classical 2-trees (see Fowler and al. in [6, 7]); the same proof applies in the case of plane and planar 2-trees and is omitted.

Theorem 1. DISSYMMETRY THEOREM FOR PLANE AND PLANAR 2-TREES. The species a_π of plane 2-trees and a_{p} of planar 2-trees satisfy the following isomorphisms of species

$$a_\pi^- + a_\pi^\Delta = a_\pi + a_\pi^{\underline{\Delta}}, \tag{14}$$

and

$$a_{\mathrm{p}}^- + a_{\mathrm{p}}^\Delta = a_{\mathrm{p}} + a_{\mathrm{p}}^{\underline{\Delta}}, \tag{15}$$

where the exponents $-$, Δ and $\underline{\Delta}$ represent the pointing of 2-trees at an edge (Figure 7a), at a triangle (Figure 7b) and at a triangle with one of its edges distinguished (Figure 7c).

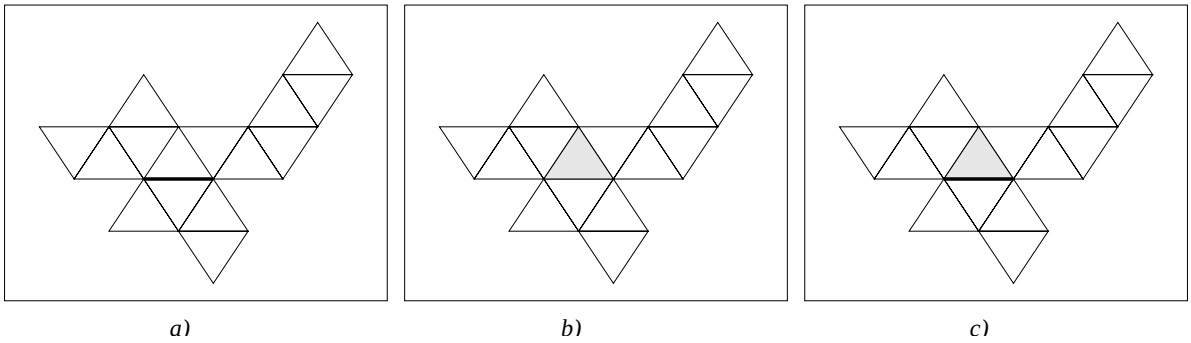


Figure 7: Examples of the exponents: a) $-$, b) Δ and c) $\underline{\Delta}$

The rest of the paper is organized as follows. In the next section, we introduce and study the auxiliary two-sort species $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$ which are needed for the expression of the species a_{p}^- and a_{p}^Δ in terms of A. In Section 3, we give addition formulas for the substitution of an asymmetric species $Y = B(X)$ into the species $E_2(Y)$, $C_3(Y)$, $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$. These results are put together in Section 4 to give the molecular expansion of the species a_π and a_{p}. All the coefficients that occur in the expressions are given explicitly in terms of Catalan numbers. Finally, the labelled, unlabelled and asymmetric enumeration of plane and planar 2-trees is carried out in Section 5.

2 The auxiliary molecular species $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$

This section is devoted to the study of some particular molecular species. A *molecular species* M is a species having only one isomorphy type. In other words, any two M-structures are

isomorphic. A molecular species is characterized by the fact that it is indecomposable under the combinatorial sum :

$$M \text{ is molecular} \quad \Leftrightarrow \quad (M = F + G \Rightarrow F = 0 \text{ or } G = 0). \tag{16}$$

It is often very useful to write a molecular species in the form

$$M = \frac{X^n}{H}, \tag{17}$$

where X^n represents the species of lists of length n and H is a subgroup of the symmetric group $\mathbb{S}_n$. We write $H \leq \mathbb{S}_n$. In fact, H is the stabilizer of some M-structure on $[n] = \{1, 2 \ldots n\}$ and n is called the *degree* of the species M. Two molecular species of degree n, X^n/H and X^n/K, are equal (*i.e.* isomorphic as species) if and only if H and K are conjugate subgroups of $\mathbb{S}_n$.

Here are some examples of molecular species

- when $H = 1$, then $X^n/1 = X^n$,
- when $H = <\rho>$, where ρ is the circular permutation $\rho = (1, 2, \ldots n)$, then $X^n/<\rho> = C_n$, the species of oriented cycles of length n,
- if now the group H is $\mathbb{S}_n$, then we have $X^n/\mathbb{S}_n = E_n$, the species of sets of size n.

We denote by $\mathcal{M}$ the set of molecular species. We can see easily that the first elements of this set, up to degree 3, are

$$\mathcal{M} = \{1, X, X^2, E_2, X^3, XE_2, E_3, C_3(X), \ldots\}. \tag{18}$$

Moreover, each species F can be expressed as a (possibly infinite) linear combination with integer coefficients of molecular species as follows,

$$F = \sum_{M \in \mathcal{M}} f_M M, \tag{19}$$

where $f_M \in \mathbb{N}$ represents the number of subspecies of F isomorphic to M. This development is unique and it is called *molecular expansion* of the species F.

It is also possible to extend the notion of molecular species to the case of multi-sort species. For instance, for two-sort species, where X and Y represent the two sorts, any molecular species can be written as

$$M(X, Y) = \frac{X^n Y^m}{H}, \tag{20}$$

where $H \leq \mathbb{S}_n^X \times \mathbb{S}_m^Y$ is the stabilizer of an M-structure. Here, $\mathbb{S}_n^X$ represents the symmetric group of degree n for the points of sort X.

We can now introduce the auxiliary species $Q(X, Y)$ and $S(X, Y)$ which will be important in our analysis of planar 2-trees. They can be defined by Figures 8 a) and 8 b) respectively, where X stands for the sort of triangles and Y, of directed edges.

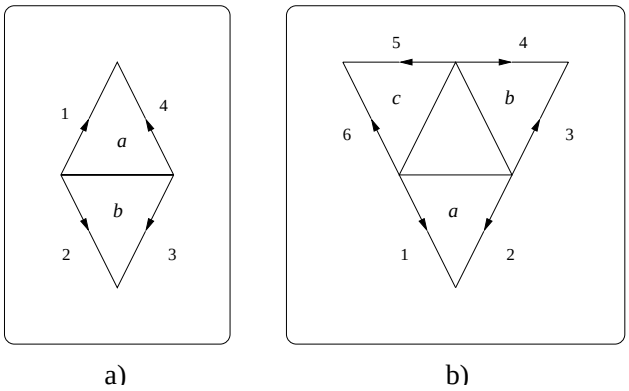


Figure 8: Structures belonging to the species $Q(X,Y)$ and $S(X,Y)$

These two molecular species are related to known species:

$$Q(X,Y) = P_4^{\text{bic}}(X,Y), \quad S(X,Y) = P_6^{\text{bic}}(X,Y), \tag{21}$$

where the species $P_n^{\text{bic}}(X)$, for n an even integer, represents the species of (vertex labelled) bicolored n-gons (see J. Labelle [14]). More precisely, the edges are colored with a set of two colors, $\{0,1\}$, in such a way that incident edges have different colors. We can then generalize to the two-sort species $P_n^{\text{bic}}(X,Y)$ where X represents the sort of edges of color 1 (dotted lines) and Y stands for the sort of vertices, as shown by Figure 9 for $n=4$ and $n=6$. This Figure also establishes (21).

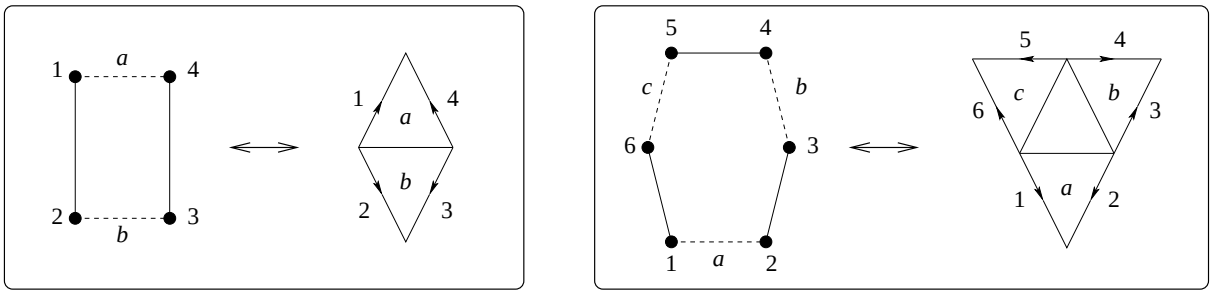


Figure 9: $P_4^{\text{bic}}(X,Y)$ and $P_6^{\text{bic}}(X,Y)$

In order to completly describe the species Q and S, we have to identify their stabilizers, and so we write them in the form (20). We have

$$P_4^{\text{bic}}(X,Y) = \frac{X^2Y^4}{D_2}, \quad P_6^{\text{bic}}(X,Y) = \frac{X^3Y^6}{S_3} \tag{22}$$

where the two groups D_2 and S_3 are characterized by their action on the labelled structures of Figure 9 :

1. $D_2 =< h, v > \leq \mathbb{S}_2^X \times \mathbb{S}_4^Y$, with

$$h = (a,b)(1,2)(3,4) \quad \text{and} \quad v = (a)(b)(1,4)(2,3).$$

Note that $h^2 = 1,\ v^2 = 1,\ hv = vh$, and $D_2 \cong \mathbb{Z}_2 \times \mathbb{Z}_2$.

2. $S_3 =< s, w > \leq \mathbb{S}_3^X \times \mathbb{S}_6^Y$, where

$$s = (a)(b,c)(1,2)(3,6)(4,5) \quad \text{and} \quad w = (a,b,c)(1,3,5)(2,4,6).$$

Note that $s^2 = 1,\ w^3 = 1,\ sws = w^2$, and $S_3 \cong \mathbb{S}_3$.

Here are the formulas giving the cycle index series and the asymmetry index series of a molecular two-sort species.

Theorem 2. [3, 10, 12] Let $M(X,Y) = X^n Y^m/H$ be a molecular species on two sorts, with $H \leq \mathbb{S}_n^X \times \mathbb{S}_m^Y$. Then, the cycle index series of M is given by

$$Z_M(x_1, x_2, \ldots; y_1, y_2, \ldots) = \frac{1}{|H|} \sum_{h \in H} x_1^{c_1(h)} x_2^{c_2(h)} \ldots y_1^{d_1(h)} y_2^{d_2(h)} \ldots, \tag{23}$$

where $c_i(h)$ (resp. $d_i(h)$), for $i \geq 1$, denotes the number of cycles of lenght i of the permutation on X-points (resp. Y-points) induced by the element $h \in H$. Furthermore, the asymmetry index series of M is given by

$$\Gamma_M(x_1, x_2, \ldots; y_1, y_2, \ldots) = \frac{1}{|H|} \sum_{V \leq H} \mu(\{1\}, V) x_1^{c_1(V)} x_2^{c_2(V)} \ldots y_1^{d_1(V)} y_2^{d_2(V)} \ldots, \tag{24}$$

where the sum is taken over all subgroups V of H, $\{1\}$ is the identity subgroup of H, $\mu(\{1\}, V)$ denotes the value of the Möbius function in the lattice of subgroup of H and $c_i(V)$ (resp. $d_i(V)$), represents the number of orbits with i elements of sort X (resp. Y) with respect to the natural action of V on $[n]$ (resp. $[m]$).

Proposition 2. The cycle index of the species $P_4^{\text{bic}}(X,Y)$ and $P_6^{\text{bic}}(X,Y)$ are given by

$$Z_{P_4^{\text{bic}}}(x_1, x_2, \ldots; y_1, y_2, \ldots) = \frac{1}{4}(x_1^2 y_1^4 + 2x_2 y_2^2 + x_1^2 y_2^2), \tag{25}$$

$$Z_{P_6^{\text{bic}}}(x_1, x_2, \ldots; y_1, y_2, \ldots) = \frac{1}{6}(x_1^3 y_1^6 + 2x_3 y_3^2 + 3x_1 x_2 y_2^3). \tag{26}$$

Proof. This is an easy exercise, using (23) and writing explicitly the elements of the group D_2 and S_3 : $D_2 = \{1, h, v, h \cdot v\}$ and $S_3 = \{1, s, \omega, \omega^2, s \cdot \omega, s \cdot \omega^2\}$. ∎

Proposition 3. The asymmetry index series of the two species $P_4^{\text{bic}}(X,Y)$ and $P_6^{\text{bic}}(X,Y)$ are given by

$$\Gamma_{P_4^{\text{bic}}}(x_1, x_2, \ldots; y_1, y_2, \ldots) = \frac{1}{4}(x_1^2 y_1^4 - x_1^2 y_2^2 - 2x_2 y_2^2 + 2x_2 y_4), \tag{27}$$

$$\Gamma_{P_6^{\text{bic}}}(x_1, x_2, \ldots; y_1, y_2, \ldots) = \frac{1}{6}(x_1^3 y_1^6 - x_3 y_3^2 - 3x_1 x_2 y_2^3 + 3x_3 y_6). \tag{28}$$

Proof. It suffices to determine the lattice of subgroups of D_2 and S_3 and to apply (24). Details are left to the reader. ∎

The cycle index series of a species encompasses the two other classical enumerative series, namely the exponential generating function of labelled structures and the ordinary generating function of unlabelled structures. In a similar way, the asymmetry index series contains other series as specializations, in particular the asymmetry generating series. For the two-sort case, these series are related as follows :

Theorem 3. ([3]). For any two-sort species F, we have

$$F(x,y) = Z_F(x,0,\ldots;y,0,\ldots) = \Gamma_F(x,0,\ldots;y,0,\ldots), \tag{29}$$
$$\widetilde{F}(x,y) = Z_F(x,x^2,\ldots;y,y^2,\ldots), \tag{30}$$
$$\overline{F}(x,y) = \Gamma_F(x,x^2,\ldots;y,y^2,\ldots). \tag{31}$$

We then confirm the expressions of the generating series of the species $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$.

Remark 1. We have

$$P_4^{\mathrm{bic}}(x,y) = \frac{1}{4}x^2y^4, \quad \widetilde{P}_4^{\mathrm{bic}}(x,y) = x^2y^4, \quad \overline{P}_4^{\mathrm{bic}}(x,y) = 0, \tag{32}$$
$$P_6^{\mathrm{bic}}(x,y) = \frac{1}{6}x^3y^6, \quad \widetilde{P}_6^{\mathrm{bic}}(x,y) = x^3y^6, \quad \overline{P}_6^{\mathrm{bic}}(x,y) = 0. \tag{33}$$

The fact that $\overline{P}_4^{\mathrm{bic}}(x,y)$ and $\overline{P}_6^{\mathrm{bic}}(x,y)$ equals 0, means that these two species are purely symmetric, *i.e.* , their asymmetric part is reduced to the empty set.

Note that if we put $Y := X^k$, for $k \geq 1$, in the species $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$, the resulting one-sort species are molecular. Indeed, the substitution of a molecular species in another one remains molecular. These two species $P_4^{\mathrm{bic}}(X,X^k)$ and $P_6^{\mathrm{bic}}(X,X^k)$, for $k \geq 1$, will be essential in order to obtain the molecular expansion of planar 2-trees. Besides, we remark the fact that

$$P_4^{\mathrm{bic}}(X,1) = E_2(X), \quad P_6^{\mathrm{bic}}(X,1) = E_3(X), \tag{34}$$

since, in Figure 8, setting $Y = 1$ corresponds to unlabelling the directed edges.

To end this section, let us give the derivative of the two-sort species $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$.

Proposition 4. The partial derivatives of $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$ are given by

$$\frac{\partial}{\partial X}P_4^{\mathrm{bic}}(X,Y) = XE_2(Y^2), \quad \frac{\partial}{\partial Y}P_4^{\mathrm{bic}}(X,Y) = X^2Y^3, \tag{35}$$
$$\frac{\partial}{\partial X}P_6^{\mathrm{bic}}(X,Y) = E_2(XY^3), \quad \frac{\partial}{\partial Y}P_6^{\mathrm{bic}}(X,Y) = X^3Y^5. \tag{36}$$

Proof. Let $F(X,Y)$ be a two-sort species and U and V be two sets representing the two sorts. Then, the partial derivatives, with respect to X and Y are defined by

$$\frac{\partial F}{\partial X}[U,V] = F[U+\{*\},V], \quad \frac{\partial F}{\partial Y}[U,V] = F[U,V+\{*\}],$$

where $*$ is a supplementary element which is used in the construction of the F-structures. From this definition, it is easy to obtain (35) et (36). ∎

3 Addition formulas

In this section, we prove some addition formulas which will be necessary to obtain the explicit molecular expansions for plane and planar 2-trees.

Proposition 5. Let B be an asymmetric species whose molecular expansion is given by

$$B(X) = \sum_{k\geq 0} b_k X^k \,.$$

Then, we have the following addition formulas relative to the species E_2 of two-element sets and C_3 of oriented 3-cycles :

$$E_2(B(X)) = \sum_{k\geq 1} b_k E_2(X^k) + \sum_{k\geq 0} \alpha_k X^k, \tag{37}$$

$$C_3(B(X)) = \sum_{k\geq 1} b_k C_3(X^k) + \sum_{k\geq 0} \beta_k X^k, \tag{38}$$

with

$$\alpha_0 = \frac{1}{2}(b_0^2 + b_0), \quad \beta_0 = \frac{1}{3}(b_0^3 + 2b_0)\,, \tag{39}$$

$$\alpha_k = \frac{1}{2}\sum_{i+j=k} b_i b_j - \frac{1}{2}\chi(2|k) b_{\frac{k}{2}}, \quad k \geq 1\,, \tag{40}$$

$$\beta_k = \frac{1}{3}\sum_{l+m+n=k} b_l b_m b_n - \frac{1}{3}\chi(3|k) b_{\frac{k}{3}}, \quad k \geq 1\,, \tag{41}$$

where, for $a, b \in \mathbb{N}$, $\chi(a|b) = 1$, if a divides b, and 0, otherwise.

Proof. First note that for any species F, the constant (*i.e.* of degree 0) term $F(b_0)$ of $F(B)$ is given by $Z_F(b_0, b_0, \ldots)$, in virtue of Polya's theorem. This yields (39). An analysis of the different shapes of molecular species which can arise in $E_2(B)$, permits us to write the following relation

$$E_2(B) = \sum_{k\geq 1} \gamma_k E_2(X^k) + \sum_{k\geq 0} \alpha_k X^k. \tag{42}$$

We now have to compute α_k and γ_k, for all $k \geq 1$. Note that we can order, in the species B, the b_k copies of the molecule X^k, for each $k \geq 1$. Then, to obtain an $E_2(X^k)$-structure from $E_2(B)$, we must take twice the same copy of X^k among the b_k available; otherwise the pair of B-structures will be asymmetric. Hence $\gamma_k = b_k$, for all $k \geq 1$. In order to compute α_k, we could perform a direct enumeration. However, we introduce a different method which will prove very useful in other situations. Differentiating the two members of (42), we get

$$BB' = \sum_{k\geq 1} k b_k X^{2k-1} + \sum_{k\geq 1} k\alpha_k X^{k-1}.$$

Integrating back this last relation, in the realm of formal power series in X, leads us to

$$\frac{1}{2}B^2 = \frac{1}{2}\sum_{k\geq 1} b_k X^{2k} + \sum_{k\geq 0} \alpha_k X^k + \text{const}\,.$$

Identifying coefficients of X^n in both sides of the last equality gives us the relation (40). To obtain (41), we first write

$$C_3(B) = \sum_{k\geq 1} \delta_k C_3(X^k) + \sum_{k\geq 0} \beta_k X^k. \tag{43}$$

The same argument as used above implies $\delta_k = b_k$, $k \geq 1$, and the same technique of differentiating-integrating equation (43) gives the announced formula for β_k. In the process, we use the fact that

$$(C_3(B))' = L_2(B)B' = B^2 B'$$

where L_2 represents the species of two-element lists. ∎

As a particular case, we have

$$E_2(1+X) = 1 + X + E_2(X), \tag{44}$$
$$C_3(1+X) = 1 + X + X^2 + C_3(X). \tag{45}$$

When $B = A$, formulas (39)–(41) take a simpler form because of the convolutive properties of Catalan numbers, as seen in Proposition 1. For this case, the coefficients α_k and β_k are given by $\alpha_0 = \beta_0 = 1$ and, for $k \geq 1$,

$$\alpha_k = \frac{1}{2}(\mathbf{c}_{k+1} - \mathbf{c}_{\frac{k}{2}}), \tag{46}$$
$$\beta_k = \frac{1}{3}(\mathbf{c}_{k+2} - \mathbf{c}_{k+1} - \mathbf{c}_{\frac{k}{3}}). \tag{47}$$

We now give the main result of this section, addition formulas for the species $P_4^{\mathrm{bic}}(X,Y)$ and $P_6^{\mathrm{bic}}(X,Y)$. Let $b_k^{(n)}$ denotes the coefficient of X^k in the species $B^n(X)$, with the convention that $b_x^{(n)} = 0$ if the index x is fractional, for all $n, k \geq 1$.

Theorem 4. Let B be an asymmetric species whose molecular expansion is given by

$$B(X) = \sum_{n\geq 0} b_k X^k.$$

Then,

$$P_4^{\mathrm{bic}}(X,B) = \sum_{k\geq 3} a_k^{'} X^k + \sum_{k\geq 2} a_k^{''} E_2(X^k) + \sum_{k\geq 1} a_k^{'''} X^2 E_2(X^k) + \sum_{k\geq 0} a_k^{iv} P_4^{\mathrm{bic}}(X, X^k), \tag{48}$$

where

$$a_k^{'} = \frac{1}{4} b_{k-2}^{(4)} - \frac{3}{4} b_{\frac{k-2}{2}}^{(2)} + \frac{1}{2} b_{\frac{k-2}{4}}, \tag{49}$$
$$a_k^{''} = b_{k-1}^{(2)} - b_{\frac{k-1}{2}}, \tag{50}$$
$$a_k^{'''} = \frac{1}{2}(b_k^{(2)} - b_{\frac{k}{2}}), \tag{51}$$
$$a_k^{iv} = b_k. \tag{52}$$

Proof. We proceed in a similar way as in Proposition 5, beginning with an analysis of the different symmetries which can appear in structures belonging to the species $P_4^{\mathrm{bic}}(X, B(X))$. This permits us to write (48) where all coefficients have to be determined. We first note that $a_k^{iv} = \mathbf{c}_k$ since the only way to build a $P_4^{\mathrm{bic}}(X, X^k)$-structure from the species $P_4^{\mathrm{bic}}(X, B)$ is to take four times the same copy of the molecule X^k among the b_k available copies. This gives (52). Next, we consider $E_2(X^k)$-structures. In order to obtain such a structure from the species $P_4^{\mathrm{bic}}(X, B)$, we can take two non isomorphic $X^{\frac{k-1}{2}}$-structures α and β from the species B, and put them in the two differents ways shown in Figure 10 a) and 10 b). This contributes for a term of

$$2\sum_l \binom{b_l}{2} E_2(X^{2l+1}),$$

remembering that the two internal triangles also contribute for one X each. We can also take an X^i-structure α and an X^j-structure β such that $i + j = k - 1$ and $i \neq j$, and put them in the two different configurations drawn in Figure 10 a) and b). In the molecular

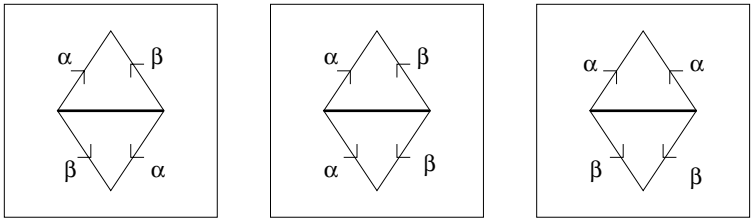


Figure 10: Symmetries of order 2 in $P_4^{\mathrm{bic}}(X, B)$

expansion of the species $Q(X, B)$ this stands for

$$2 \sum_{\substack{i+j=k-1 \\ i<j}} b_i b_j E_2(X^k).$$

It leads to (50), *i.e.*

$$a_k^{''} = 2 \sum_{\substack{i+j=k-1 \\ i<j}} b_i b_j + \binom{b_{\frac{k-1}{2}}}{2} = b_{k-1}^{(2)} - b_{\frac{k-1}{2}}.$$

Let us now turn to the coefficient $a_k^{'''}$ of $X^2 E_2(X^k)$ in the relation (48). The configurations belonging to an $X^2 E_2(X^k)$ are shown in Figure 10 c). We then have

$$a_k^{'''} = \sum_{\substack{i+j=k \\ i<j}} b_i b_j + \binom{b_{\frac{k}{2}}}{2} = \frac{1}{2}(b_k^{(2)} - b_{\frac{k}{2}})$$

types of $X^2 E_2(X^k)$-structures. It remains to determine the asymmetric part of the species $Q(X, B)$, *i.e.* the coefficient $a_k^{'}$ of X^k in the molecular expansion (48), for all k. To find it, we differentiate the relation (48) and we identify the coefficient of X^k in each side. It gives the expression (49), which completes the proof. Note that we use the combinatorial derivative of a composite species $F(X, B(X))$. As in calculus, we have

$$(F(X, B(X)))' = \frac{\partial F(X, Y)}{\partial X}|_{Y:=B} + \frac{\partial F(X, Y)}{\partial Y}|_{Y:=B} \cdot B', \tag{53}$$

and we can use Proposition 4. ∎

Remark 2. We can perform a precise classification separating rotational and reflectional symmetries. Indeed, the symmetries illustrated by Figure 10 are rotational for the case a), vertically reflectional for case b) and horizontally reflectional for c).

Remark also that we could obtain the expression of $a_k^{'''}$ by identifying the coefficient of $XE_2(X^k)$ after deriving (48).

Theorem 5. For all asymmetric species B whose molecular expansion is

$$B(X) = \sum_{k\geq 0} b_k X^k\,,$$

we have

$$P_6^{\mathrm{bic}}(X,B) = \sum_{k\geq 4} d_k^{'} X^k + \sum_{k\geq 2} d_k^{''} XE_2(X^k) + \sum_{k\geq 2} d_k^{'''} C_3(X^k) + \sum_{k\geq 0} d_k^{iv} P_6^{\mathrm{bic}}(X,X^k), \tag{54}$$

where

$$\begin{aligned}
d_k^{'} &= \frac{1}{6}b_{k-3}^{(6)} - \frac{1}{2}b_{\frac{k-3}{2}}^{(3)} + \frac{1}{3}b_{\frac{k-3}{3}}^{(2)} + \frac{2}{3}b_{\frac{k-3}{6}}, && (55)\\
d_k^{''} &= b_{k-1}^{(3)} - b_{\frac{k-1}{3}}, && (56)\\
d_k^{'''} &= \frac{1}{2}(b_{k-1}^{(2)} - b_{\frac{k-1}{2}}), && (57)\\
d_k^{iv} &= b_k, && (58)
\end{aligned}$$

where $b_k^{(n)}$ represents the coefficient of X^k in $B^n(X)$.

Proof. A precise analysis of the different symmetries arising in the species $P_6^{\mathrm{bic}}(X,B)$ permit us to write the expansion (54). We then compute all coefficients of this expression by the same method as for the species $P_4^{\mathrm{bic}}(X,B)$. ∎

When we put $B = A$ in the two previous theorems, the coefficients appearing in the molecular expansions of the species P_4^{bic} and P_6^{bic} are simpler. In fact, by Proposition 1 we get the following expressions for a_k^i and d_k^i, for $i \in \{\prime, \prime\prime, \prime\prime\prime, iv\}$

$$\begin{aligned}
a_k^{'} &= \frac{1}{4}\mathbf{c}_{k+1} - \frac{1}{2}\mathbf{c}_k - \frac{3}{4}\mathbf{c}_{\frac{k}{2}} + \frac{1}{2}\mathbf{c}_{\frac{k-2}{4}},\\
a_k^{''} &= \mathbf{c}_k - \mathbf{c}_{\frac{k-1}{2}}, && (59)\\
a_k^{'''} &= \frac{1}{2}(\mathbf{c}_{k+1} - \mathbf{c}_{\frac{k}{2}}),\\
a_k^{iv} &= \mathbf{c}_k,
\end{aligned}$$

$$\begin{aligned}
d_k^{'} &= \frac{1}{6}\mathbf{c}_{k+2} - \frac{2}{3}\mathbf{c}_{k+1} + \frac{1}{2}\mathbf{c}_k - \frac{1}{2}\mathbf{c}_{\frac{k+1}{2}} + \frac{1}{2}\mathbf{c}_{\frac{k-1}{2}} - \frac{1}{6}\mathbf{c}_{\frac{k}{3}} + \frac{1}{2}\mathbf{c}_{\frac{k-3}{6}},\\
d_k^{''} &= \mathbf{c}_{k+1} - \mathbf{c}_k - \mathbf{c}_{\frac{k-1}{3}}, && (60)\\
d_k^{'''} &= \frac{1}{2}(\mathbf{c}_k - \mathbf{c}_{\frac{k-1}{2}}),\\
d_k^{iv} &= \mathbf{c}_k.
\end{aligned}$$

4 Molecular expansion of plane and planar 2-trees

In this part, we use the dissymmetry theorem and the results of the previous section to obtain an explicit form for the molecular expansion of the species of plane 2-trees and of planar 2-trees.

4.1 Plane 2-trees

Recall that plane 2-trees are 2-trees that are embedded (drawn) in the plane in such a way that all internal faces are triangles. The dissymetry theorem gives an expression for the species a_π in terms of the pointed species a_π^-, a_π^Δ and $a_\pi^{\dot\Delta}$, namely

$$a = a_\pi^- + a_\pi^\Delta - a_\pi^{\dot\Delta}. \tag{61}$$

Here, we can use the orientation of the plane to obtain simple expressions for the pointed species as function of the species A defined in the introduction, as shown in Figure 11 :

Theorem 6. The species arising in the dissymmmetry theorem for plane 2-trees satisfy

$$a_\pi^- = E_2(A), \tag{62}$$
$$a_\pi^\Delta = XC_3(A), \tag{63}$$
$$a_\pi^{\dot\Delta} = A_+ \cdot A, \tag{64}$$

where $A_+ = A - 1$.

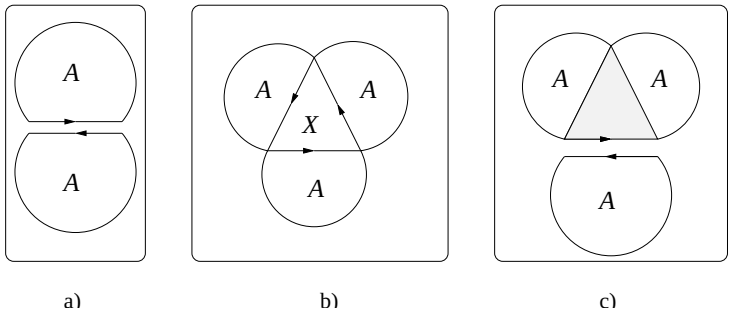


Figure 11: The species $E_2(A)$, $XC_3(A)$ and $A_+ \cdot A$

Using the expansion formulas for $E_2(A)$ and $C_3(A)$, given in Section 3, we can now compute the molecular expansion of the species a_π.

Theorem 7. The molecular expansion of the species a_π of plane 2-trees is given by

$$a_\pi = a_\pi(X) = 1 + X + \sum_{k\geq 2} b_k X^k + \sum_{k\geq 1} c_k E_2(X^k) + \sum_{k\geq 1} d_k XC_3(X^k), \tag{65}$$

where

$$b_k = \frac{2}{3}\mathbf{c}_k - \frac{1}{6}\mathbf{c}_{k+1} - \frac{1}{2}\mathbf{c}_{\frac{k}{2}} - \frac{1}{3}\mathbf{c}_{\frac{k-1}{3}}, \tag{66}$$
$$c_k = d_k = \mathbf{c}_k, \tag{67}$$

where X^k represents the species of k-lists of triangles and $\mathbf{c}_k$ are the usual Catalan numbers with the convention that $\mathbf{c}_r = 0$ if r is not an integer; see (13).

To conclude this section we write the asymmetric part, in the sense of G. Labelle [11], of the species of plane 2-trees :

$$\overline{a}_\pi(X) = 1 + X + \sum_{k\geq 2} b_k X^k, \tag{68}$$

where b_k, for $k \in \mathbb{N}$, is given by the formula (66). The species $\overline{a}_\pi$ is not to be confused with the pointed species a_π^-.

4.2 Planar 2-trees

This subsection is devoted to planar 2-trees, *i.e.* 2-trees admitting an embedding in the plane in such a way that all internal faces are triangles. The difference here is that the embedding is not explicitely given and that reflexive symmetries are possible. In other words, planar 2-trees are viewed as simple graphs. The dissymetry theorem for the species $a_{\rm p}$ of planar 2-trees yields

$$a_{\rm p} = a_{\rm p}^- + a_{\rm p}^\Delta - a_{\rm p}^{\underline{\Delta}}. \tag{69}$$

Moreover, we have the following expressions for the pointed species $a_{\rm p}^-$, $a_{\rm p}^\Delta$ and $a_{\rm p}^{\underline{\Delta}}$, in terms of the auxiliary species $P_4^{\rm bic}(X,Y)$ and $P_6^{\rm bic}(X,Y)$ introduced in Section 2.

Theorem 8. The species of pointed planar 2-trees $a_{\rm p}^-$, $a_{\rm p}^\Delta$ and $a_{\rm p}^{\underline{\Delta}}$ satisfy the following isomorphisms of species :

$$a_{\rm p}^-(X) = 1 + XE_2(A) + P_4^{\rm bic}(X,Y)|_{Y:=A}, \tag{70}$$

$$a_{\rm p}^\Delta(X) = X + X^2E_2(A) + XE_2(A_+) + XP_6^{\rm bic}(X,Y)|_{Y:=A}, \tag{71}$$

$$a_{\rm p}^{\underline{\Delta}}(X) = XE_2(A) + X^2E_2(A^2). \tag{72}$$

Proof. We obtain the functional equations (70) and (72) by analyzing the structures according to the degree of the distinguished edge. For example, the three terms on the right hand side of (70) correspond respectively to the degrees 0, 1 and 2 of the pointed edge. This isomorphism is described in Figure 12. In (71), the four terms correspond to the four possibilities for the number of edges of degree 2 in the pointed triangle, from 0 to 3; see Figure 13. For (72), see Figure 14. ∎

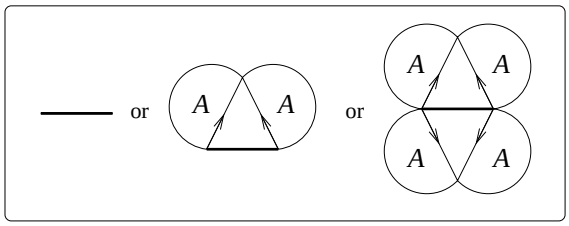


Figure 12: The species $a_{\rm p}^-$

Combining the molecular expansion of the quotient species $P_4^{\rm bic}(X,A)$ and $P_6^{\rm bic}(X,A)$ established in Section 3 with Proposition 1 and Proposition 5, gives the molecular expansion of the species $a_{\rm p}^-$ and $a_{\rm p}^\Delta$. Note that we use the same notation for the coefficients of the different molecular expansions in the four following theorems.

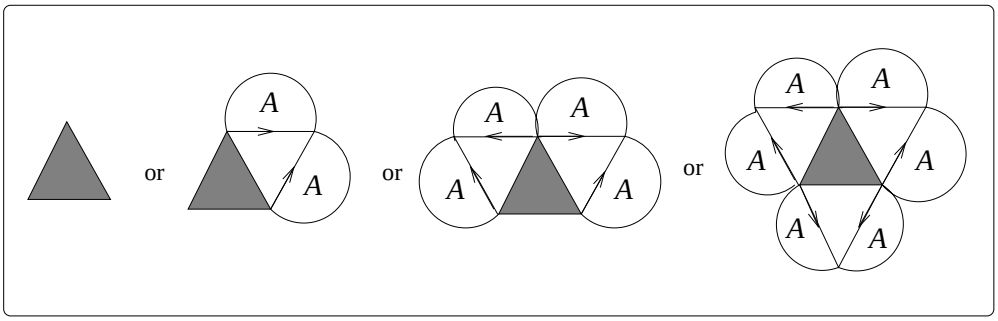


Figure 13: The species a_{p}^{Δ}

Theorem 9. The molecular expansion of the species a_{p}^{-} of edge pointed planar 2-trees is given by

$$\begin{aligned} a_{\mathrm{p}}^{-}(X) &= 1+\sum_{k\geq 0} a_k^1 X^k + \sum_{k\geq 1} a_k^2 E_2(X^k) + \sum_{k\geq 1} a_k^3 X E_2(X^k) \\ &\quad + \sum_{n\geq 1} a_k^4 X^2 E_2(X^k) + \sum_{k\geq 1} a_k^5 P_4^{\mathrm{bic}}(X, X^k), \end{aligned} \tag{73}$$

where

$$\begin{aligned} a_k^1 &= \frac{1}{4}\mathbf{c}_{k+1} - \frac{3}{4}\mathbf{c}_{\frac{k}{2}} - \frac{1}{2}\mathbf{c}_{\frac{k-1}{2}} + \frac{1}{2}\mathbf{c}_{\frac{k-2}{4}}, \\ a_k^2 &= \mathbf{c}_k - \mathbf{c}_{\frac{k-1}{2}}, \\ a_k^3 &= a_k^5 = \mathbf{c}_k, \\ a_k^4 &= \frac{1}{2}(\mathbf{c}_{k+1} - \mathbf{c}_{\frac{k}{2}}). \end{aligned} \tag{74}$$

Theorem 10. The molecular expansion of the species a_{p}^{Δ} is given by

$$\begin{aligned} a_{\mathrm{p}}^{\Delta}(X) &= 1+\sum_{k\geq 0} a_k^1 X^k + \sum_{k\geq 1} a_k^2 X\cdot E_2(X^k) + \sum_{k\geq 2} a_k^3 X^2 E_2(X^k) \\ &\quad + \sum_{k\geq 2} a_k^4 X C_3(X^k) + \sum_{k\geq 2} a_k^5 X P_6^{\mathrm{bic}}(X, X^k), \end{aligned} \tag{75}$$

where

$$\begin{aligned} a_k^1 &= \frac{1}{6}(\mathbf{c}_{k+1} - \mathbf{c}_k) - \frac{1}{2}\mathbf{c}_{\frac{k}{2}} - \mathbf{c}_{\frac{k-2}{2}} - \frac{1}{2}\mathbf{c}_{\frac{k-1}{2}} - \frac{1}{6}\mathbf{c}_{\frac{k-1}{3}} + \frac{1}{2}\mathbf{c}_{\frac{k-4}{6}}, \\ a_k^2 &= a_k^5 = \mathbf{c}_k, \\ a_k^3 &= \mathbf{c}_{k+1} - \mathbf{c}_{\frac{k-1}{3}}, \\ a_k^4 &= \frac{1}{2}(\mathbf{c}_k - \mathbf{c}_{\frac{k-1}{2}}). \end{aligned} \tag{76}$$

Proposition 1 and Proposition 5 also allow us to obtain the molecular expansion of the species a_{p}^{Δ}.

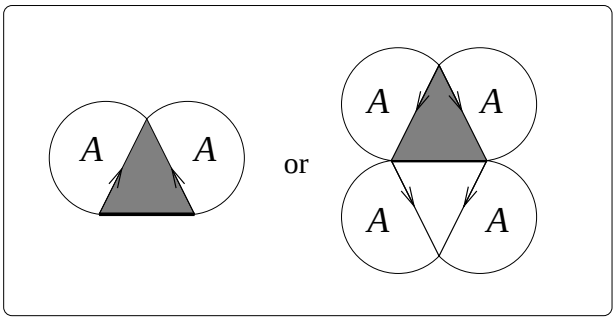


Figure 14: The species a_{p}^{Δ}

Theorem 11. The molecular expansion of the species a_{p}^{Δ} of planar 2-trees pointed at a triangle with a distinguished edge is given by

$$a_{\mathrm{p}}^{\Delta}(X) = \sum_{k\geq 0} a_k^1 X^k + \sum_{k\geq 1} a_k^2 X E_2(X^k) + \sum_{k\geq 1} a_k^3 X^2 E_2(X^k), \tag{77}$$

where

$$\begin{aligned} a_k^1 &= \frac{1}{2}\Big(\mathbf{c}_{k+1} - \mathbf{c}_k - \mathbf{c}_{\frac{k-1}{2}} - \mathbf{c}_{\frac{k}{2}}\Big), \\ a_k^2 &= \mathbf{c}_k, \\ a_k^3 &= \mathbf{c}_{k+1}. \end{aligned} \tag{78}$$

Using the dissymetry theorem, we are now able to put together relations (73)-(75)-(77) and give an explicit form of the molecular expansion of the species a_{p} of planar 2-trees.

Theorem 12. The molecular expansion of the species a_{p} of planar 2-trees is given by the following formula

$$\begin{aligned} a_{\mathrm{p}}(X) &= 1 + \sum_{k\geq 1} a_k^1 X^k + \sum_{k\geq 1} a_k^2 E_2(X^k) + \sum_{k\geq 1} a_k^3 X E_2(X^k) + \sum_{k\geq 2} a_k^4 X^2 E_2(X^k) \\ &\quad + \sum_{k\geq 2} a_k^5 X C_3(X^k) + \sum_{k\geq 0} a_k^6 P_4^{\mathrm{bic}}(X, X^k) + \sum_{k\geq 0} a_k^7 X P_6^{\mathrm{bic}}(X, X^k), \end{aligned} \tag{79}$$

where

$$\begin{aligned} a_k^1 &= -\frac{1}{12}\mathbf{c}_{k+1} + \frac{1}{3}\mathbf{c}_k - \frac{3}{4}\mathbf{c}_{\frac{k}{2}} - \frac{1}{2}\mathbf{c}_{\frac{k-1}{2}} - \frac{1}{6}\mathbf{c}_{\frac{k-1}{3}} + \frac{1}{2}\mathbf{c}_{\frac{k-2}{4}} + \frac{1}{2}\mathbf{c}_{\frac{k-4}{6}}, \\ a_k^2 &= \mathbf{c}_k - \mathbf{c}_{\frac{k-1}{2}}, \\ a_k^3 &= a_k^6 = a_k^7 = \mathbf{c}_k, \\ a_k^4 &= \frac{1}{2}(\mathbf{c}_{k+1} - \mathbf{c}_{\frac{k}{2}}) - \mathbf{c}_{\frac{k-1}{3}}, \\ a_k^5 &= \frac{1}{2}(\mathbf{c}_k - \mathbf{c}_{\frac{k-1}{2}}). \end{aligned} \tag{80}$$

5 Enumeration formulas

5.1 Enumeration of plane 2-trees

Before obtaining the explicit enumeration of plane 2-trees, we recall some basic formulas involving index series of the species of 2-element sets (E_2) and of oriented 3-cycles (C_3) :

$$Z_{E_2}(x_1,x_2,\ldots) = \frac{1}{2}(x_1^2+x_2), \qquad \Gamma_{E_2}(x_1,x_2,\ldots) = \frac{1}{2}(x_1^2-x_2), \tag{81}$$

$$Z_{C_3}(x_1,x_2,\ldots) = \frac{1}{3}(x_1^3+2x_3), \qquad \Gamma_{C_3}(x_1,x_2,\ldots) = \frac{1}{3}(x_1^3-x_3). \tag{82}$$

We will also use some substitutional laws of the theory of species : for any species F and G such that $G(0)=0$ (G has no structure on the empty set), we have

$$(F\circ G)(x) = F(G(x)), \tag{83}$$

$$(F\circ G)^{\sim}(x) = Z_F(\widetilde{G}(x),\widetilde{G}(x^2),\ldots)., \tag{84}$$

$$(\overline{F\circ G})(x) = \Gamma_F(\bar{G}(x),\bar{G}(x^2),\ldots), \tag{85}$$

$$Z_{F\circ G} = Z_F\circ Z_G, \tag{86}$$

$$\Gamma_{F\circ G} = \Gamma_F\circ\Gamma_G, \tag{87}$$

where $\circ$ denotes the plethystic composition on the right hand side of (86) and (87).

If the species G has some structures on the empty set, *i.e.* $G(0)=g_0\neq 0$, formulas (84)–(86) remain valid. However, formula (83) should then be replaced by

$$(F\circ G)(x) = Z_F(G(x),g_0,g_0,\ldots), \tag{88}$$

and there is no known general formula for Γ. Here, we only need the following formulas

$$\Gamma_{E_2(G)}(x_1,x_2,\ldots) = g_0+\frac{1}{2}(\Gamma_G^2(x_1,x_2,\ldots)-\Gamma_G(x_2,x_4,\ldots)), \tag{89}$$

$$\Gamma_{C_3(G)}(x_1,x_2,\ldots) = g_0+\frac{1}{3}(\Gamma_G^3(x_1,x_2,\ldots)-\Gamma_G(x_3,x_6,\ldots)). \tag{90}$$

We now give the explicit enumerative formulas provided directly by the molecular expansion of the species of plane 2-trees.

Theorem 13. The numbers $a_{\pi,n}$, $\widetilde{a}_{\pi,n}$ and $\overline{a}_{\pi,n}$ of labelled, unlabelled and unlabelled asymmetric plane 2-trees on n triangles, $n\geq 2$, are given by

$$a_{\pi,n} = n!(\frac{2}{3}\mathbf{c}_n-\frac{1}{6}\mathbf{c}_{n+1}), \tag{91}$$

$$\widetilde{a}_{\pi,n} = \frac{2}{3}\mathbf{c}_n-\frac{1}{6}\mathbf{c}_{n+1}+\frac{1}{2}\mathbf{c}_{\frac{n}{2}}+\frac{2}{3}\mathbf{c}_{\frac{n-1}{3}}, \tag{92}$$

$$\overline{a}_{\pi,n} = \frac{2}{3}\mathbf{c}_n-\frac{1}{6}\mathbf{c}_{n+1}-\frac{1}{2}\mathbf{c}_{\frac{k}{3}}-\frac{1}{3}\mathbf{c}_{\frac{n-1}{3}}. \tag{93}$$

To obtain these enumerating formulas, we can also use the expressions (62)–(64) which lead to closed formulas for the associated series of the three pointed species : the exponential generating series of labelled structures,

$$\begin{aligned} a_{\pi}^{-}(x) &= \frac{1}{2}(1+A^2(x)), \\ a_{\pi}^{\Delta}(x) &= \frac{x}{3}(2+A^3(x)), \\ a_{\pi}^{\underline{\Delta}}(x) &= A^2(x)-A(x), \end{aligned} \tag{94}$$

the ordinary generating series of unlabelled structures

$$\begin{aligned} \widetilde{a}_{\pi}^{-}(x) &= \frac{1}{2}(A^2(x)+A(x^2)), \\ \widetilde{a}_{\pi}^{\Delta}(x) &= \frac{x}{3}(A^3(x)+2A(x^3)), \\ \widetilde{a}_{\pi}^{\underline{\Delta}}(x) &= A^2(x)-A(x), \end{aligned} \tag{95}$$

the cycle index series

$$\begin{aligned} Z_{a_{\pi}^{-}}(x_1,x_2,\ldots) &= \frac{1}{2}\Big(A^2(x_1)+A(x_2)\Big), \\ Z_{a_{\pi}^{\Delta}}(x_1,x_2,\ldots) &= \frac{x_1}{3}\Big(A^3(x_1)+2A(x_3)\Big), \\ Z_{a_{\pi}^{\underline{\Delta}}}(x_1,x_2,\ldots) &= A^2(x_1)-A(x_1), \end{aligned} \tag{96}$$

the asymmetry cycle index series

$$\begin{aligned} \Gamma_{a_{\pi}^{-}}(x_1,x_2,\ldots) &= 1+\frac{1}{2}\Big(A^2(x_1)-A(x_2)\Big), \\ \Gamma_{a_{\pi}^{\Delta}}(x_1,x_2,\ldots) &= x_1+\frac{x_1}{3}\Big(A^3(x_1)-A(x_3)\Big), \\ \Gamma_{a_{\pi}^{\underline{\Delta}}}(x_1,x_2,\ldots) &= A^2(x_1)-A(x_1). \end{aligned} \tag{97}$$

We emphasize the fact, used above, that since the species A is asymmetric we have the following relations

$$A(x)=\widetilde{A}(x) \;=\; \overline{A}(x) \quad \text{and} \quad Z_A(x_1,x_2,\ldots)=A(x_1)=\Gamma_A(x_1,x_2,\ldots). \tag{98}$$

We then deduce easily (thanks to the dissymmetry theorem) the expressions of the series associated with the species of plane 2-trees

Proposition 6. The series associated to the species a_{π} of plane 2-trees are given by

$$\begin{aligned} a_{\pi}(x) &= \frac{1}{2}+\frac{2}{3}x+A(x)-\frac{1}{2}A^2(x)+\frac{x}{3}A^3(x), \\ \widetilde{a}_{\pi}(x) &= 1+x+A(x)+\frac{x}{3}A^3(x)-\frac{1}{2}A(x^2)-\frac{x}{3}A(x^3)-\frac{1}{2}A^2(x), \\ \bar{a}_{\pi}(x) &= A(x)+\frac{x}{3}A^3(x)-A(x^2)-A(x^3)-\frac{1}{2}A^2(x), \\ Z_{a_{\pi}}(x_1,x_2,\ldots) &= A(x_1)+\frac{1}{2}A(x_2)+\frac{2}{3}x_1A(x_3)-\frac{1}{2}A^2(x_1)+\frac{x_1}{3}A^3(x_1), \\ \Gamma_{a_{\pi}}(x_1,x_2,\ldots) &= 1+x_1+A(x_1)+\frac{x_1}{3}A^3(x_1)-\frac{1}{2}A(x_2)-\frac{x_1}{3}A(x_3)-\frac{1}{2}A^2(x_1). \end{aligned} \tag{99}$$

To recover the formulas (92), we can use the dissymmetry theorem and the next proposition giving the enumeration of the different pointed plane 2-trees.

Proposition 7. The coefficients $a^{\bullet}_{\pi,n}$, $a^{\overset{\Delta}{\bullet}}_{\pi,n}$, $a^{\overset{\Delta}{\underline{\bullet}}}_{\pi,n}$ representing the numbers of labelled structures with n triangles for the different pointings, $\widetilde{a}^{\bullet}_{\pi,n}$, $\widetilde{a}^{\overset{\Delta}{\bullet}}_{\pi,n}$, $\widetilde{a}^{\overset{\Delta}{\underline{\bullet}}}_{\pi,n}$ for the numbers of unlabelled structures, and $\overline{a}^{\bullet}_{\pi,n}$, $\overline{a}^{\overset{\Delta}{\bullet}}_{\pi,n}$, $\overline{a}^{\overset{\Delta}{\underline{\bullet}}}_{\pi,n}$ for unlabelled asymmetric structures, are given, for $n \geq 2$, by

$$\begin{aligned} a^{\bullet}_{\pi,n} &= \frac{n!}{2}\mathbf{c}_{n+1}, \\ a^{\overset{\Delta}{\bullet}}_{\pi,n} &= \frac{n!}{3}(\mathbf{c}_{n+1} - \mathbf{c}_n), \\ a^{\overset{\Delta}{\underline{\bullet}}}_{\pi,n} &= n!(\mathbf{c}_{n+1} - \mathbf{c}_n), \end{aligned} \tag{100}$$

$$\begin{aligned} \widetilde{a}^{\bullet}_{\pi,n} &= \frac{1}{2}(\mathbf{c}_{n+1} + \mathbf{c}_{\frac{n}{2}}), \\ \widetilde{a}^{\overset{\Delta}{\bullet}}_{\pi,n} &= \frac{1}{3}(\mathbf{c}_{n+1} - \mathbf{c}_n + 2\mathbf{c}_{\frac{n-1}{3}}), \\ \widetilde{a}^{\overset{\Delta}{\underline{\bullet}}}_{\pi,n} &= \mathbf{c}_{n+1} - \mathbf{c}_n, \end{aligned} \tag{101}$$

and

$$\begin{aligned} \bar{a}^{\bullet}_{\pi,n} &= \frac{1}{2}(\mathbf{c}_{n+1} - \mathbf{c}_{\frac{n}{2}}), \\ \bar{a}^{\overset{\Delta}{\bullet}}_{\pi,n} &= \frac{1}{3}(\mathbf{c}_{n+1} - \mathbf{c}_n - \mathbf{c}_{\frac{n-1}{3}}), \\ \bar{a}^{\overset{\Delta}{\underline{\bullet}}}_{\pi,n} &= \mathbf{c}_{n+1} - \mathbf{c}_n. \end{aligned} \tag{102}$$

Proof. To obtain these coefficients, we simply use relations (94), (95) and (97). ∎

We now give the explicit expressions for the cycle index series of the species of plane 2-trees.

Proposition 8. The cycle index series and the asymmetric index series of the species of plane 2-trees are

$$Z_{\mathcal{A}_\pi}(x_1, x_2, \ldots) = 1 + \sum_{n\geq 1}(\frac{2}{3}\mathbf{c}_n - \frac{1}{6}\mathbf{c}_{n+1})x_1^n + \frac{1}{2}\sum_{n\geq 1}\mathbf{c}_n x_2^n + \frac{2}{3}x_1\sum_{n\geq 0}\mathbf{c}_n x_3^n, \tag{103}$$

$$\Gamma_{\mathcal{A}_\pi}(x_1, x_2, \ldots) = 1 + x_1 + \sum_{n\geq 1}(\frac{2}{3}\mathbf{c}_n - \frac{1}{6}\mathbf{c}_{n+1})x_1^n - \frac{1}{2}\sum_{n\geq 1}\mathbf{c}_n x_2^n - \frac{1}{3}x_1\sum_{n\geq 0}\mathbf{c}_n x_3^n. \tag{104}$$

Proof. We first express the cycle index series given by the relations (96) in powers of x_1, x_2, ...

$$\begin{aligned} Z_{\mathcal{A}^{\bullet}_\pi}(x_1, x_2, \ldots) &= \frac{1}{2}\sum_{n\geq 0}\mathbf{c}_{n+1}x_1^n + \frac{1}{2}\sum_{n\geq 0}\mathbf{c}_n x_2^n, \\ Z_{\mathcal{A}^{\overset{\Delta}{\bullet}}_\pi}(x_1, x_2, \ldots) &= \frac{1}{3}\sum_{n\geq 1}(\mathbf{c}_{n+1} - \mathbf{c}_n)x_1^n + \frac{2}{3}x_1\sum_{n\geq 0}\mathbf{c}_n x_3^n, \\ Z_{\mathcal{A}^{\overset{\Delta}{\underline{\bullet}}}_\pi}(x_1, x_2, \ldots) &= \sum_{n\geq 1}(\mathbf{c}_{n+1} - \mathbf{c}_n)x_1^n. \end{aligned} \tag{105}$$

We also have

$$\begin{aligned}
\Gamma\, a_{\pi}^{-}(x_1,x_2,\ldots) &= 1+\frac{1}{2}\sum_{n\geq 0}\mathbf{c}_{n+1}x_1^n-\frac{1}{2}\sum_{n\geq 0}\mathbf{c}_n x_2^n,\\
\Gamma\, a_{\pi}^{\Delta}(x_1,x_2,\ldots) &= x_1+\frac{1}{3}\sum_{n\geq 1}(\mathbf{c}_{n+1}-\mathbf{c}_n)x_1^n-\frac{1}{3}x_1\sum_{n\geq 0}\mathbf{c}_n x_3^n,\\
\Gamma\, a_{\pi}^{\underline{\Delta}}(x_1,x_2,\ldots) &= \sum_{n\geq 1}(\mathbf{c}_{n+1}-\mathbf{c}_n)x_1^n.
\end{aligned} \tag{106}$$

It suffices then to use the dissymmetry theorem to obtain the stated result. ∎

5.2 Enumeration of planar 2-trees

We now give all associated series of the species a_{p}^{-}, a_{p}^{Δ} and $a_{\mathrm{p}}^{\underline{\Delta}}$ using substitutionnal laws of the theory of species. After this, we will be able to give all coefficients arising in these differents series, and, with the dissymetry theorem, we obtain the number of labelled and unlabelled planar 2-trees on n triangles as well as the coefficients of its cycle and asymmetry index series.

Theorem 14. The exponential generating function of labelled stuctures for the species a_{p}^{-}, a_{p}^{Δ} and $a_{\mathrm{p}}^{\underline{\Delta}}$ of planar pointed 2-trees are given, in terms of the species A, by

$$\begin{aligned}
a_{\mathrm{p}}^{-}(x) &= 1+\frac{x}{2}(1+A^2(x))+\frac{1}{4}x^2A^4(x),\\
a_{\mathrm{p}}^{\Delta}(x) &= x+\frac{x^2}{2}(1+A^2(x))+\frac{x}{2}A_+^2(x)+\frac{x^4}{6}A^6(x), labelgf1\\
a_{\mathrm{p}}^{\underline{\Delta}}(x) &= \frac{x}{2}(1+A^2(x))+\frac{x^2}{2}(1+A^4(x)).
\end{aligned} \tag{107}$$

Moreover, the ordinary generating series of unlabelled structures of these species are given by

$$\begin{aligned}
\widetilde{a}_{\mathrm{p}}^{-}(x) &= 1+xA(x)+\frac{x}{2}(A^2(x)+A(x^2))+\frac{x^2}{4}(A^4(x)+3A^2(x^2)),\\
\widetilde{a}_{\mathrm{p}}^{\Delta}(x) &= x+\frac{x^2}{2}(A^2(x)+A(x^2))+\frac{x}{2}(A_+^2(x)+A_+(x^2))\\
&\qquad +\frac{x^4}{6}(A^6(x)+2A^2(x^3)+3A^3(x^2)),\\
\widetilde{a}_{\mathrm{p}}^{\underline{\Delta}}(x) &= \frac{x}{2}(A^2(x)+A(x^2))+\frac{x^2}{2}(A^4(x)+A^2(x^2)).
\end{aligned} \tag{108}$$

Corollary 1. The exponential and the ordinary generating functions of the species of planar 2-trees are given, in terms of A, by

$$\begin{aligned}
a_{\mathrm{p}}(x) &= 1+x+\frac{x}{2}A_+^2(x)+\frac{x^2}{2}A^2(x)-\frac{x^2}{4}A^4(x)-\frac{x^4}{6}A^6(x),\\
\widetilde{a}_{\mathrm{p}}(x) &= 1+x+\frac{x}{2}(A_+^2(x)+A_+(x^2))+\frac{x^2}{2}A(x^2)+\frac{x^2}{2}(A^2(x)-A^2(x^2))\\
&\qquad -\frac{x^2}{4}A^4(x)+\frac{x^4}{6}(A^6(x)+2A^2(x^3)+3A^3(x^2)).
\end{aligned} \tag{109}$$

A simple extraction of coefficients in Theorem 14, combined with Proposition 1, yields the following corollary.

Corollary 2. The numbers $a^{-}_{\mathrm{p},n}$, $a^{\Delta}_{\mathrm{p},n}$ and $a^{\underline{\Delta}}_{\mathrm{p},n}$ of labelled planar 2-trees on n triangles pointed respectively at an edge, at a triangle, and at a triangle pointed at one of its edges, are given by

$$\begin{aligned} a^{-}_{\mathrm{p},n} &= \frac{n!}{4}\mathbf{c}_{n+1}, \\ a^{\Delta}_{\mathrm{p},n} &= \frac{n!}{6}(\mathbf{c}_{n+1}-\mathbf{c}_n), \\ a^{\underline{\Delta}}_{\mathrm{p},n} &= \frac{n!}{2}(\mathbf{c}_{n+1}-\mathbf{c}_n). \end{aligned} \tag{110}$$

Moreover, for the same pointed series, the numbers of unlabelled structures on n triangles $\widetilde{a}^{-}_{\mathrm{p},n}$, $\widetilde{a}^{\Delta}_{\mathrm{p},n}$ and $\widetilde{a}^{\underline{\Delta}}_{\mathrm{p},n}$ have the following expressions :

$$\begin{aligned} \widetilde{a}^{-}_{\mathrm{p},n} &= \frac{1}{4}\mathbf{c}_{n+1}+\frac{1}{2}\mathbf{c}_{\frac{n-1}{2}}+\frac{3}{4}\mathbf{c}_{\frac{n}{2}}, \\ \widetilde{a}^{\Delta}_{\mathrm{p},1} &= 1,\ \widetilde{a}^{\Delta}_{\mathrm{p},2}=1,\ \widetilde{a}^{\Delta}_{\mathrm{p},3}=2,\ \widetilde{a}^{\Delta}_{\mathrm{p},4}=6, \\ \widetilde{a}^{\Delta}_{\mathrm{p},n} &= \frac{1}{6}(\mathbf{c}_{n+1}-\mathbf{c}_n)+\frac{1}{2}\Big(\mathbf{c}_{\frac{n-1}{2}}+\mathbf{c}_{\frac{n}{2}}\Big)+\frac{1}{3}\mathbf{c}_{\frac{n-1}{3}}, \quad n\geq 5 \\ \widetilde{a}^{\underline{\Delta}}_{\mathrm{p},n} &= \frac{1}{2}(\mathbf{c}_{n+1}-\mathbf{c}_n)+\mathbf{c}_{\frac{n-1}{2}}+\mathbf{c}_{\frac{n}{2}}. \end{aligned} \tag{111}$$

Hence, the dissymmetry theorem leads us to enumeration formulas for labelled and unlabelled planar 2-trees as follows. For the unlabelled asymmetric enumeration, we use directly the molecular decomposition of the species a_{p}.

Theorem 15. The numbers $a_{\mathrm{p},n}$, $\widetilde{a}_{\mathrm{p},n}$ and $\bar{a}_{\mathrm{p},n}$ of labelled, unlabelled and unlabelled asymmetric planar 2-trees on n triangles, are given by the following formulas

$$a_{\mathrm{p},n} = n!(\frac{1}{3}\mathbf{c}_n-\frac{1}{12}\mathbf{c}_{n+1}), \tag{112}$$

$$\widetilde{a}_{\mathrm{p},n} = \frac{1}{3}\mathbf{c}_n-\frac{1}{12}\mathbf{c}_{n+1}+\frac{1}{2}\mathbf{c}_{\frac{n-1}{2}}+\frac{1}{3}\mathbf{c}_{\frac{n-1}{3}}+\frac{3}{4}\mathbf{c}_{\frac{n}{2}}, \tag{113}$$

$$\bar{a}_{\mathrm{p},n} = -\frac{1}{12}\mathbf{c}_{n+1}+\frac{1}{3}\mathbf{c}_n-\frac{3}{4}\mathbf{c}_{\frac{n}{2}}-\frac{1}{2}\mathbf{c}_{\frac{n-1}{2}}-\frac{1}{6}\mathbf{c}_{\frac{n-1}{3}}+\frac{1}{2}\mathbf{c}_{\frac{n-2}{4}}+\frac{1}{2}\mathbf{c}_{\frac{n-4}{6}}. \tag{114}$$

Finally, we give the expression of the asymmetry index series of the species a_{p} of planar 2-trees obtained directly from the molecular expansion of the species a_{p}.

Proposition 9. The asymmetry index series of the species of planar 2-trees is given by

$$\begin{aligned} \Gamma a_{\mathrm{p}}(x_1,x_2,\ldots) = 1+x_1+\sum_n \gamma^1_n x_1^n+\sum_n \gamma^2_n x_2^n+\sum_n \gamma^3_n x_1x_2^n+\sum_n \gamma^4_n x_1^2x_2^n+ \\ +\sum_n \gamma^5_n x_1x_3^n+\sum_n \gamma^6_n x_2x_4^n+\sum_n \gamma^7_n x_1x_3x_6^n, \end{aligned} \tag{115}$$

where

$$\begin{aligned}
\gamma_n^1 &= -\frac{1}{12}\mathbf{c}_{n+1} + \frac{1}{3}\mathbf{c}_n, \\
\gamma_n^2 &= \gamma_n^3 = -\frac{1}{2}\mathbf{c}_n, \\
\gamma_n^4 &= -\frac{1}{4}\mathbf{c}_{n+1}, \\
\gamma_n^5 &= -\frac{1}{6}\mathbf{c}_n, \\
\gamma_n^6 &= \gamma_n^7 = \frac{1}{2}\mathbf{c}_n.
\end{aligned} \tag{116}$$

5.3 Another method for the unlabelled enumeration

In order to obtain the unlabelled enumeration of plane and planar 2-trees, we can also use the approach of Palmer and Read in [15]. Remark first that, for any species F, we can write

$$F = \sum_{k\geq 1} F_{(k)}, \tag{117}$$

where for $k \geq 1$, $F_{(k)}$ represents the symmetric part of F of order k, *i.e.* the subspecies consisting of F-structures whose stabilizer is of order k exactly. In particular, $F_{(1)} = \overline{F}$, the asymmetric part of F.

Also note that, for $G = F_{(k)}$, $k \geq 1$, we have $G(x) = \frac{1}{k}\widetilde{G}(x)$, since an unlabelled $F_{(k)}$-structure of degree n can be labelled in $n!/k$ ways. Hence

$$\widetilde{F}(x) = F(x) + \sum_{k\geq 2} \frac{k-1}{k}\widetilde{F}_{(k)}(x). \tag{118}$$

For plane 2-trees, we have

$$a_\pi = \overline{a}_\pi + a_{\pi,(2)} + a_{\pi,(3)}, \tag{119}$$

and for planar 2-trees,

$$a_{\rm p} = \overline{a}_{\rm p} + a_{{\rm p},(2)} + a_{{\rm p},(3)} + a_{{\rm p},(4)} + a_{{\rm p},(6)}. \tag{120}$$

Hence, we can write

$$\widetilde{a}_\pi(x) = a_\pi(x) + \frac{1}{2}\widetilde{a}_{\pi,(2)}(x) + \frac{2}{3}\widetilde{a}_{\pi,(3)}(x), \tag{121}$$

and

$$\widetilde{a}_{\rm p}(x) = a_{\rm p}(x) + \frac{1}{2}\widetilde{a}_{{\rm p},(2)}(x) + \frac{2}{3}\widetilde{a}_{{\rm p},(3)}(x) + \frac{3}{4}\widetilde{a}_{{\rm p},(4)}(x) + \frac{5}{6}\widetilde{a}_{{\rm p},(6)}(x). \tag{122}$$

After identifying all terms appearing in (121), we then deduce

$$\widetilde{a}_\pi(x) = a_\pi(x) + \frac{1}{2}A(x^2) + \frac{2}{3}xA(x^3), \tag{123}$$

for the plane case. For planar 2-trees, we have

$$\begin{aligned}
\widetilde{a}_{\mathrm{p},(2)}(x) &= \frac{3}{2}(A(x^2) - x^2A(x^4)) + xA(x^2) - x^4A(x^6), \\
\widetilde{a}_{\mathrm{p},(3)}(x) &= \frac{1}{2}(xA(x^3) - x^4A(x^6)), \\
\widetilde{a}_{\mathrm{p},(4)}(x) &= x^2A(x^4), \quad \widetilde{a}_{\mathrm{p},(6)} = x^4A(x^6),
\end{aligned} \tag{124}$$

which yields

$$\widetilde{a}_{\mathrm{p}}(x) = a_{\mathrm{p}}(x) + \frac{1}{2}xA(x^2) + \frac{1}{3}xA(x^3) + \frac{3}{4}A(x^2). \tag{125}$$

It remains to extract the coefficients of x^n in equations (123) and (125) to find the numbers of unlabelled plane and planar 2-trees over n triangles, given by (92) and (113).

References

[1] P. Auger, G. Labelle, and P. Leroux, *Combinatorial addition formulas*, Proceedings FPSAC'01, Tempe, Arizona, May 21-25 2001, H. Barcelo and V. Welker, Eds, pp 19–26.

[2] P. Auger, G. Labelle, and P. Leroux, *Combinatorial addition formulas and applications*, Advances in Applied Mathematics, to appear.

[3] F. Bergeron, G. Labelle, and P. Leroux, *Combinatorial Species and tree-like structures*, Encyclopedia of Mathematics and it'ss Applications, vol. 67, Cambridge University Press, (1998).

[4] W. G. Brown, *Enumeration of triangulations of the disk*, Proc. London Math. Soc. **14**, 746-768, (1964).

[5] S. J. Cyvin, J. Brunvoll, E. Brensdal, B. N. Cyvin and E. K. Lloyd, *Enumeration of Polyene Hydrocarbons : A Complete Mathematical Solution*, J. Chem. Inf. Comput. Sci., **35**, 743-751, (1995).

[6] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *Specifying 2-trees*, Proceedings FPSAC'00, Moscou, 26-30 juin 2000, 202-213.

[7] T. Fowler, I. Gessel, G. Labelle, P. Leroux, *The Specification of 2-trees*, Advances in Applied Mathematics, to appear.

[8] F. Harary and E. Palmer, *Graphical Enumeration*, Academic Press, New York, (1973).

[9] F. Harary, E. Palmer and R. Read, *On the cell-growth problem for arbitrary polygons*, Discrete Mathematics, 11, 371–389, (1975).

[10] A. Kerber, *Enumeration under Finite Group Action: Symmetry Classes of Mappings*, Combinatoire énumérative, Proceedings, Montréal, Québec, Lectures Notes in Mathematics, vol. 1234, Springer-Verlag, New-York/Berlin, 160–176, (1985).

[11] G. Labelle, *On Asymmetric Structures*, Discrete Mathematics, 99, 141-162, (1992).

[12] G. Labelle, J. Labelle and K. Pineau, *Sur une généralisation des séries indicatrices d'espèces*, J. of Comb. Theory, Series A, **69**, No. 1, 17–35, (1995).

[13] G. Labelle, C. Lamathe and P. Leroux, *Développement moléculaire de l'espèce des 2-arbres planaires*, Proceedings GASCom 2001, 41–46, (2001).

[14] J. Labelle, *Quelques espèces sur les ensembles de petite cardinalité*, Annales des Sciences Mathématiques du Québec, **11**, 31-58, (1985).

[15] E. Palmer and R. Read, *On the Number of Plane* 2*-trees*, J. London Mathematical Society **6**, 583-592, (1973).

[16] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego, (1995).

E-mail address : **{gilbert, lamathe, leroux}@lacim.uqam.ca**

* Corresponding author : Pierre Leroux
LaCIM, Département de Mathématiques, UQÀM
C. P. 8888, succursale Centre-Ville
Montréal (Qc) Canada H3C 3P8

Enumeration of matchings in polygraphs*

Per Håkan Lundow

Abstract

The 6-cube has a total of 7174574164703330195841 matchings of which 16332454526976 are perfect. This was computed with a transfer matrix method associated with polygraphs. For polygraphs of type $G \times P_m$ we present a method for compression of the transfer matrix. This compression gives a substantial reduction of the order of the transfer matrix by exploiting the automorphisms of the graph G. We compute and tabulate matching polynomials of various polygraphs, such as the $4 \times 4 \times m$-grid. A Mathematica package, GrafPack, is demonstrated and used for computation of matching polynomials, permanents and for generating transfer matrices.

1 Introduction

A simple graph is denoted $G = (V, E)$ where V is the set of vertices and E is the set of edges. A matching M is a set of independent edges in G, i.e. no pair of edges in M have a vertex in common. A k-matching is a matching on k edges and a perfect matching is a matching that covers all the vertices in G. The matching polynomial of a graph G on n vertices is defined as

$$\mu(G;\, x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k p(G,k) x^{n-2k}$$

where $p(G,k)$ denotes the number of k-matchings in G and we define $p(G,0) = 1$. We overload the notation and define

$$\mu(G) = \sum_{k=0}^{\lfloor n/2 \rfloor} p(G,k)$$

i.e. $\mu(G)$ is the number of matchings in G. A 1-factor is a spanning 1-regular subgraph. The edges of a 1-factor then form a perfect matching and the number of 1-factors in a graph G is denoted $\Phi(G)$. In general it is a $\#P$-complete problem to compute $\mu(G;\, x)$ and also $\Phi(G)$, though there are families of graphs such as paths, cycles and complete graphs, for which these functions can be simply expressed. Apart from these instances, general expressions are scarce. It is well-known however, that $\Phi(G)$ can be computed in polynomial time for planar graphs. Computing the matching polynomial is still harder, becoming

*Revised and updated version of "Computation of the matching polynomial and the number of 1-factors in polygraphs", Research reports, No 12, 1996.

$\#P$-complete even for planar graphs. More information on these matters can be found in Godsil [4] and Lovász and Plummer [14]. For more on complexity classes, see Welsh [22].

In the next section we will state some of the applications of matching theory to physics and chemistry. This is followed by a quick introduction to the subject of actually computing the matching polynomial, the number of matchings and the number of 1-factors in a graph. A family of graphs of interest in chemistry, polygraphs, is presented together with a transfer matrix method to compute their matching polynomials. We then present a new result, a compression of the matrices, which allows us to make these matrices considerably smaller. The algorithms described have been implemented in Mathematica. Some of the Mathematica routines are demonstrated and we give tables of the resulting numbers for some polygraphs along with some recurrence relations.

2 Applications of matching theory

There are several connections between matching theory and statistical physics and also chemistry. For example, adsorption of oxygen and hydrogen on a metallic surface can be modelled by a system of monomers-dimers. The question is whether adsorption undergoes a phase transition at some critical temperature. The surface is represented as a grid and it is exposed to a gas consisting of monomers and dimers. Dimers could here correspond to oxygen molecules which cover adjacent vertices on the grid. A set of dimers forms a matching on the grid and the state of the system is then represented by this matching. As partition function one takes the matching polynomial with non-negative coefficients. The paper by Heilmann and Lieb [6] contains a detailed study of this problem.

The Ising model is concerned with the phenomenon of spontaneous magnetization. If a magnetic material is placed in a hot environment it becomes unmagnetized, although below a certain critical temperature the material will regain a degree of its magnetism. We then have a phase transition at this critical temperature. The partition function of the Ising model can be expressed in terms of the 1-factors of a graph with weighted edges, the weight of a 1-factor being the product of its edge-weights. Again we refer the reader to [6] and also Kasteleyn [12]. A nice introduction to the Ising model is given by Cipra [3].

In mathematical chemistry, molecules are viewed as graphs and chemists refer to 1-factors as Kekulé structures. It turns out that the stability of some families of molecules is closely related to the number of 1-factors in their graphs. Several types of polynomials, partition functions and invariants of interest in chemistry have been suggested, many of which are expressed in terms of the numbers $p(G,k)$. For example, $\mu(G)$ is also known as the Hosoya index and has been used to model physicochemical properties such as the boiling point of hydrocarbons. See for example Hosoya [7], Rouvray [17] and Trinajstić [21]. A more general account of combinatorics in statistical physics and chemistry can be found in Chapter 37 and 38 of The Handbook of Combinatorics [5].

3 Computation methods

3.1 The matching polynomial

To compute the matching polynomial of a graph G we need the facts below. We will just state them and refer the reader who requires proofs to [4]. First of all

$$\mu(G;\, x) = \mu(G - e;\, x) - \mu(G - u - v;\, x)$$

where $e = \{u, v\}$ is an edge of G. If G and H are disjoint graphs then

$$\mu(G \cup H;\, x) = \mu(G;\, x)\, \mu(H;\, x)$$

Let P_n, C_n and K_n denote the path, cycle and complete graph respectively on n vertices. The complementary graph of G is denoted by $\overline{G}$, thus $\overline{K_n}$ is the empty graph on n vertices. We have

$$\begin{aligned}
\mu(P_n;\, x) &= \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \binom{n-k}{k} x^{n-2k} \\
\mu(C_n;\, x) &= \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \frac{n}{n-k} \binom{n-k}{k} x^{n-2k} \\
\mu(K_n;\, x) &= \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \frac{n!}{(2k)!!(n-2k)!} x^{n-2k} \\
\mu(\overline{K_n};\, x) &= x^n
\end{aligned}$$

We can now give a simple recursive algorithm for computation of $\mu(G;\, x)$: if the maximum degree of the graph is at most 2, then the graph is a union of vertex-disjoint paths and cycles and we can compute the product of their respective matching polynomials. Otherwise, pick a pair of adjacent vertices of high degree, delete these vertices and the edge and make the recursive calls. Though recursive, the method works well for smaller graphs. The running time of the algorithm depends on the number of edges of G, meaning that dense graphs could be a problem. However, the following formula takes care of that

$$\mu(G;\, x) = \sum_{k=0}^{\lfloor n/2 \rfloor} p(\overline{G};\, k)\, \mu(K_{n-2k};\, x)$$

Thus, if G is dense (has more than $n^2/4$ edges, say), then use the algorithm above on $\overline{G}$ and apply the last formula. To extract $\Phi(G)$ and $\mu(G)$ from the matching polynomial we observe that $\Phi(G) = |\mu(G;\, 0)|$ and $\mu(G) = |\mu(G;\, \mathbf{i})|$, where $\mathbf{i}$ is the imaginary unit. In the next section we describe a better way to compute $\Phi(G)$ when G is bipartite.

3.2 The permanent

For bipartite graphs, there is a simple non-recursive method to compute Φ. Let $G = (V \cup W, E)$ be a bipartite graph on $2n$ vertices with bipartition (V, W),

where $V = \{ v_1, \ldots, v_n \}$ and $W = \{ w_1, \ldots, w_n \}$. The biadjacency matrix $B = (b_{i,j})_{n \times n}$ is defined to have entries

$$b_{i,j} = \begin{cases} 1 & \text{if } \{v_i, w_j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

The permanent of an $n \times n$-matrix B is defined as

$$\operatorname{per}(B) = \sum_{\pi} \prod_{i=1}^{n} b_{i,\pi(i)}$$

where the sum is taken over all permutations π of $\{ 1, \ldots, n \}$. If B is the matrix defined above, then

$$\Phi(G) = \operatorname{per}(B).$$

Thus, counting the 1-factors in a bipartite graph is equivalent to evaluating the permanent of its biadjacency matrix. The permanent, looking deceptively similar to the determinant, shares few of its nice properties. Particularly the property $\det(AB) = \det(A)\det(B)$ does not hold for permanents. Also, whereas the determinant can be computed in $O(n^3)$ time, no polynomial-time algorithm is known for the permanent. In fact, it has been shown to be a $\#P$-hard problem, making computation of $\Phi(G)$ a $\#P$-complete problem for bipartite graphs as well. A detailed survey on the permanent is found in Minc [15] and a proof of the $\#P$-hardness result is sketched in [22].

Evaluation of the permanent, as formulated above, would require $n \cdot n!$ arithmetic operations. It was shown by Ryser [18] that

$$\operatorname{per}(B) = (-1)^n \sum_{S \subseteq [n]} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j \in S} b_{i,j}$$

where $[n] = \{ 1, \ldots, n \}$. This reduces the number of operations required to about $n^2\, 2^{n-1}$. Nijenhuis and Wilf [16] devised and implemented a method to reduce the number of operations by a factor n. Their main trick is to order the sets in the first sum in Gray-code order, i.e., so that consecutive sets differ in exactly one element. As it stands then, the permanent can be computed with about $n2^{n-1}$ operations. Counting the 1-factors in the 6-cube (64 vertices) is thus quite feasible, but the 7-cube (128 vertices) would require immense computer resources with this approach.

There are inequalities for permanents of doubly stochastic matrices (having row and column sums equal to 1) that can be applied to regular bipartite graphs, see [14]. If the bipartite graph G above is k-regular then

$$n! \left(\frac{k}{n}\right)^n \leq \Phi(G) \leq (k!)^{n/k}$$

Applied to the 7-cube we get $3.9280 \cdot 10^{27} \leq \Phi(Q^7) \leq 7.0924 \cdot 10^{33}$.

3.3 Estimating the number of 1-factors

We finish this section by describing a simple probabilistic method for estimating $\Phi(G)$, proved in [14]. The adjacency matrix $A = (a_{i,j})_{n \times n}$ of an oriented graph

$\overrightarrow{G}$ on the vertices $\{ v_1, \ldots, v_n \}$ has entries

$$a_{i,j} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ -1 & \text{if } (v_j, v_i) \in E \\ 0 & \text{otherwise} \end{cases}$$

Give the graph G an orientation by randomly orienting every edge with probability $1/2$ in either direction. It turns out that the expected value of $\det(A(\overrightarrow{G}))$ is $\Phi(G)$. This implies a probabilistic method to estimate $\Phi(G)$. Just compute

$$\frac{1}{p} \sum_{i=1}^{p} \det(A(\overrightarrow{G}))$$

where the sum is taken over p independently chosen orientations of G. When G is bipartite we can gain a factor 8 in running time. Give G a random orientation $\overrightarrow{G}$ by letting each non-zero entry of the biadjacency matrix B be positive or negative with equal probability. Observe that if G is bipartite then

$$A(\overrightarrow{G}) = \begin{pmatrix} 0 & B(\overrightarrow{G}) \\ -B(\overrightarrow{G})^T & 0 \end{pmatrix}$$

and the reader may verify that

$$\det(A(\overrightarrow{G})) = (\det(B(\overrightarrow{G}))^2$$

This method is also called the Godsil-Gutman estimator. The major drawback with the method is that the number p which gives a small relative error with a large probability is not necessarily polynomially bounded in n. Only for a few families of graphs is this known to be the case. However, the very simplicity of the method makes it a first candidate for computing a rough estimate of $\Phi(G)$, or at least the number of digits of $\Phi(G)$. Karmarkar et al. [13] contains an analysis of the Godsil-Gutman estimator and describes a slightly improved version of it. An implementation of the estimator in Fortran was applied to the 7-cube with $p = 10^7$ and resulted in the estimate $\Phi(Q^7) \approx 3.89 \cdot 10^{29}$.

4 Polygraphs

So far we have not discussed how to take advantage of symmetries or recurring structures in a graph when computing matching polynomials. As an example, the reader may have in mind the $2 \times 2 \times m$-grid, $m \geq 1$, when reading this section. This is just the 2×2-grid, recurring m times, linked together by edges. Graphs of this kind belong to a family of graphs of interest in theoretical chemistry and are called polygraphs, see Figure 1. They were introduced by Babic et al. [1] who also gave a matrix method for computing their matching polynomials. A polygraph consists of a set of disjoint graphs $G_1, \ldots, G_m$ and a set of binary relations $X_1, \ldots, X_m$. Let $X_i \subseteq V(G_i) \times V(G_{i+1})$ for $i = 1, \ldots, m-1$ and $X_m \subseteq V(G_m) \times V(G_1)$. For consistency we define X_0 to be identical to X_m. The polygraph Ω_m has vertices $V(G_1) \cup \cdots \cup V(G_m)$ and edges $E(G_1) \cup X_1 \cup \cdots \cup E(G_m) \cup X_m$. Let Γ_m be the graph Ω_m without the edges X_m. If $G_1 = \cdots = G_m = G$ and $X_1 = \cdots = X_m = X$ we denote Ω_m by ω_m and call it a

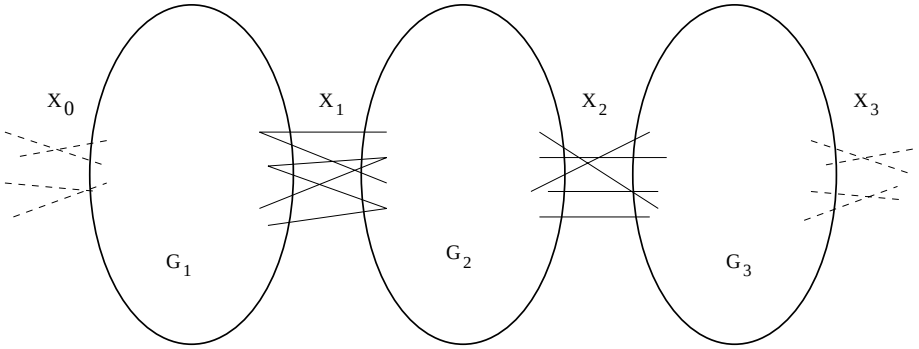


Figure 1: The structure of a polygraph

rotagraph on (G, X). Likewise, we denote Γ_m by γ_m and call it a fasciagraph on (G, X). Let $M(X)$ be the set of all matchings in X. We index these matchings with numbers $1, 2, \ldots, |M(X)|$ and adopt the convention of letting the first matching be the empty set. Let $W_i^{(k)}$ denote the ith element in $M(X_k)$. If $W \in M(X)$, let $D(W)$ and $R(W)$ be the domain and range respectively of W. Define $\mu(G - A - B;\, x) = 0$ if $A \cap B \neq \emptyset$, where $A, B \subseteq V(G)$. Define matrices $T_k = T_k(G_k, X_{k-1}, X_k)$, $k = 1, \ldots, m$ with entries

$$T_k(i,j) = (-1)^{|W_j^{(k)}|}\, \mu(G_k - R(W_i^{(k-1)}) - D(W_j^{(k)});\, x) \tag{1}$$

where the notation $T_k(i,j)$ refers to the entry in the ith row and jth column of the matrix T_k. Below we repeat some of the results in [1].

$$\begin{aligned}
[T_1 \cdots T_m](i,j) &= (-1)^{|W_j^{(m)}|}\, \mu(\Gamma_m - R(W_i^{(m)}) - D(W_j^{(m)});\, x) \\
[T_1 \cdots T_m](1,1) &= \mu(\Gamma_m;\, x) \\
\operatorname{tr}(T_1 \cdots T_m) &= \mu(\Omega_m;\, x)
\end{aligned}$$

For rota- and fasciagraphs, we have that $T_1 = \cdots = T_m = T$ where

$$T(i,j) = (-1)^{|W_j|} \mu(G - R(W_i) - D(W_j);\, x) \tag{2}$$

We then have

$$\begin{aligned}
T^m(i,j) &= (-1)^{|W_j|}\, \mu(\Gamma_m - R(W_i^{(m)}) - D(W_j^{(m)});\, x) \\
T^m(1,1) &= \mu(\gamma_m;\, x) \\
\operatorname{tr}(T^m) &= \mu(\omega_m;\, x)
\end{aligned}$$

The formulae become really simple if we want the special cases $G \times P_m$ or $G \times C_m$. Then, for all $A_i, A_j \subseteq V(G)$ we let

$$T(i,j) = (-1)^{|A_j|}\, \mu(G - A_i - A_j;\, x) \tag{3}$$

and so, if we let $A_1 = \emptyset$,

$$\begin{aligned}
T^m(1,1) &= \mu(G \times P_m;\, x) \\
\operatorname{tr}(T^m) &= \mu(G \times C_n;\, x)
\end{aligned}$$

Of course, after the obvious adjustments, these formulae also holds if we want the number of 1-factors (i.e. Φ) or the number of matchings (i.e. μ), simply delete the sign in front of the entries. Having defined the transfer matrix we can construct recurrence relations for the matching polynomial of ω_m and γ_m. Denote the characteristic polynomial of the matrix T by

$$\Xi(T,\lambda) = \det(\lambda I - T) = \sum_{k=0}^{N} a_k \lambda^{N-k}$$

where $N = |M(X)|$ (which is also the order of T). Application of the Cayley-Hamilton theorem gives that $\Xi(T,T) = \mathbf{0}$, where the $\mathbf{0}$ represents a zero-matrix of order N. From this we derive the recursive formulae of order N

$$\sum_{k=0}^{N} a_k \operatorname{tr}(T^{m-k}) = 0$$

$$\sum_{k=0}^{N} a_k \, T^{m-k}(1,1) = 0$$

where $m \geq N$. Note that when we are determining $\mu(\omega_m; x)$ and $\mu(\gamma_m; x)$, the coefficients a_k will be polynomials in x.

5 Compression

Let T be the transfer matrix for a fasciagraph as defined by Equation (2). Of course we wish the order of T to be as small as possible, to make matrix computations easy and the recurrence relations short. Unfortunately, though the method described in the previous section *does* take advantage of the recurring structure of the rota- and fasciagraphs, any symmetry in the graph G is *not* exploited. For example, if the edges in X are all independent, the matrix T has order $2^{|X|}$, no matter what graph G we use, empty or complete. In this section we will address this problem. In fact, in a special case we may reduce the order of the matrices by almost a factor the size of the automorphism group of G. First some notation though.

If G and H are graphs, then the Cartesian product $G \times H$ is defined as the graph having vertices $V(G) \times V(H)$ and where (v,w) is adjacent to (v',w') if and only if

$$v = v' \text{ and } \{w,w'\} \in E(H), \text{ or, } w = w' \text{ and } \{v,v'\} \in E(G)$$

For example, $P_m \times P_n$ is the $m \times n$-grid, $C_m \times P_n$ is a cylinder and $C_m \times C_n$ is a torus.

Let $\mathrm{Aut}(G)$ be the group of automorphisms of G and let A be a subset of $V(G)$ such that $\alpha(A) = A$ for all $\alpha \in \mathrm{Aut}(G)$. The case we are aiming for is the fasciagraph γ_m on (G,X) where we let $X = \{\,(v,v) : v \in A\,\}$. Note that if $A = V(G)$ then $\gamma_m = G \times P_m$.

We will now classify the subsets of A into equivalence classes under the automorphism group according to the following; let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_r$ be the equivalence classes of subsets of A. That is to say, every $I \subseteq A$ belongs to some $\mathcal{A}_k$, and $I, J \in \mathcal{A}_k$ if and only if $J = \alpha(I)$ for some $\alpha \in \mathrm{Aut}(G)$. As a convention

we let $\mathcal{A}_1 = \{\emptyset\}$. We can now define the compressed matrix C in terms of the matrix T. Since the edges in X are independent, no confusion will arise when we write $T(I,J)$ instead of $T(i,j)$ where $I = D(W_i)$ and $J = R(W_j)$.

Definition 5.1. The compressed transfer matrix C is the $r \times r$-matrix with entries

$$C(i,j) = \sum_{J \in \mathcal{A}_j} T(I,J) \quad \text{where } I \in \mathcal{A}_i \text{ and } i,j = 1,\ldots,r. \tag{4}$$

When calculating $C(i,j)$ we have to pick a set $I \in \mathcal{A}_i$. The following lemma says that it doesn't matter which set we pick, i.e. the matrix C is well-defined.

Lemma 5.2. *Let $I_1, I_2 \in \mathcal{A}_i$. Then*

$$\sum_{J \in \mathcal{A}_j} T(I_1,J) = \sum_{J \in \mathcal{A}_j} T(I_2,J) \quad \textit{for } i,j = 1,\ldots,r$$

Proof. Since $I_1, I_2 \in \mathcal{A}_i$ we can assume that $I_2 = \alpha(I_1)$ for some permutation $\alpha \in \mathrm{Aut}(G)$. It suffices to show that the sets in $\{I_1 \cup J : J \in \mathcal{A}_j\}$ are equal to the sets in $\{I_2 \cup J : J \in \mathcal{A}_j\}$ in some, possibly permuted, order. It follows by the definition of the set $\mathcal{A}_j$ that for all $\alpha \in \mathrm{Aut}(G)$ and $J \in \mathcal{A}_j$ there is a $J' \in \mathcal{A}_j$ such that $J' = \alpha(J)$. Thus, for all $J \in \mathcal{A}_j$ there is a $J' \in \mathcal{A}_j$ such that

$$I_2 \cup J = \alpha(I_1) \cup \alpha(J') = \alpha(I_1 \cup J')$$

and the lemma follows. □

Theorem 5.3. *If $I \in \mathcal{A}_i$ then*

$$C^m(i,j) = \sum_{J \in \mathcal{A}_j} T^m(I,J) \quad \textit{for } m \geq 1 \textit{ and } i,j = 1,\ldots,r.$$

Proof. By induction on m. The case $m = 1$ follows from the definition of the matrix C. Assume the theorem to be true for $m-1$ and show it for $m > 1$. We have

$$\sum_{J \in \mathcal{A}_j} T^m(I,J) = \sum_{J \in \mathcal{A}_j} \sum_{K \subseteq A} T^{m-1}(I,K)\, T(K,J) =$$

$$= \sum_{J \in \mathcal{A}_j} \sum_{k=1}^{r} \sum_{K \in \mathcal{A}_k} T^{m-1}(I,K)\, T(K,J) =$$

$$= \sum_{k=1}^{r} \sum_{K \in \mathcal{A}_k} T^{m-1}(I,K) \sum_{J \in \mathcal{A}_j} T(K,J)$$

By the lemma and the definition this is

$$\sum_{k=1}^{r} \sum_{K \in \mathcal{A}_k} T^{m-1}(I,K)\, C(k,j)$$

and the induction hypothesis allows us to write this as

$$\sum_{k=1}^{r} C^{m-1}(i,k)\, C(k,j) = C^m(i,j)$$

and by the principle of induction the theorem follows. □

Corollary 5.4. *If C is defined on the matrix T in Equation (2) then*

$$C^m(1,1) = \mu(\gamma_m;\, x) \quad \textit{for } m \geq 1.$$

Proof. Recall that $\mathcal{A}_1 = \{\emptyset\}$.

$$C^m(1,1) = \sum_{J \in \mathcal{A}_1} T^m(\emptyset, J) = T^m(\emptyset,\emptyset) = T^m(1,1) = \mu(\gamma_m;\, x)$$

□

Comparing the orders of C and T, how much did we gain? The order of T is $N = 2^{|A|}$ since all edges in X are independent. If we denote by r the order of C, then r is (usually) slightly larger than $N/|\mathrm{Aut}(G)|$ which is a lower bound on the number of equivalence classes. The exact number can be determined with Polya's Enumeration Theorem:

$$r = \frac{1}{|\mathrm{Aut}(G)|} \sum_{\pi \in \mathrm{Aut}(G)} 2^{c(\pi, A)}$$

where $c(\pi, A)$ is the number of cycles in the permutation π that contain elements from A. In Broersma and Xueliang [2] a reduction of almost a factor 2 of the order of T was accomplished. They laid slightly less strong restrictions on the binary relation X (independent edges, though), but the graph G was restricted to having vertex-set $\{1, 2, \ldots, 2p\}$ and an automorphism $i \leftrightarrow p + i$, for $i = 1, \ldots, p$. The compression described here puts no restrictions on G, and works better the more automorphisms G has. Unfortunately we pay with information, since the trace of C no longer has the meaning it had for T.

6 Further reductions

We assume that we just want to count the 1-factors in γ_m. The order of the matrix C may then at least be halved to obtain a new, smaller, matrix $\hat{C}$. The simplest reduction stems from the fact that a graph on an odd number of vertices does not have a 1-factor. As before we let r denote the order of C. Renumber the families of sets that resulted from the classification procedure such that $\mathcal{A}_1, \ldots, \mathcal{A}_s$ contain the subsets of A of even size, and the remaining classes $\mathcal{A}_{s+1}, \ldots, \mathcal{A}_r$ contain the subsets of odd size. If $|V(G)|$ is even then $C(i,j) = 0$ if $i \leq s$ and $j > s$, or, $i > s$ and $j \leq s$. If $|V(G)|$ is odd, then $C(i,j) = 0$ if $i, j \leq s$ or $i, j > s$. The matrix C will then look like

$$\begin{pmatrix} P & 0 \\ 0 & Q \end{pmatrix} \quad \text{for even } |V(G)|, \qquad \begin{pmatrix} 0 & R \\ S & 0 \end{pmatrix} \quad \text{for odd } |V(G)|. \tag{5}$$

Here P is an $s \times s$-matrix, Q an $(r-s) \times (r-s)$-matrix, R an $s \times (r-s)$-matrix and S an $(r-s) \times s$-matrix. Assume that $|V(G)|$ is even and define

$$\hat{C}(i,j) = C(i,j) \quad \text{for } i, j = 1, 2, \ldots, s \tag{6}$$

Then $\hat{C}$ is the upper block P on the diagonal of C. The other blocks in C will not affect this matrix during matrix multiplication, since C is block diagonal. We have then proved the following

Proposition 6.1.

$$\hat{C}^m(i,j) = C^m(i,j) \quad \textit{for } m \geq 1$$

We continue with the case when $|V(G)|$ is odd and define

$$\hat{C}(i,j) = C^2(i,j) \quad \text{for } i,j = 1,2,\ldots,s \tag{7}$$

This means that $\hat{C}$ is the block product RS. Note that the upper left block in C^m will be a zero matrix when m is odd. A proposition similar to the one above follows.

Proposition 6.2.

$$\hat{C}^m(i,j) = C^{2m}(i,j) \quad \textit{for } m \geq 1$$

In both the odd and the even case we end up with an $s \times s$-matrix, where s is the number of even non-equivalent subsets of A. If $|A|$ is odd then $s = r/2$ and if $|A|$ is even then $s \approx r/2$. Roughly then, the order of $\hat{C}$ is half that of C.

The last case, finally, is when G is bipartite. Note that a bipartite graph on two sets of unequal size does not contain a 1-factor. Restrict G to be a bipartite graph on $2n$ vertices with bipartition (V,W) and let $|V| = |W| = n$. Again we renumber the classes, but this time such that for all $I \subseteq A$ we have that $I \in \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_s$ if and only if $|I \cap V| = |I \cap W|$, that is, I is a balanced subset of $V \cup W$. Then $C(i,j) = 0$ if $i \leq s$ and $j > s$, or, $i > s$ and $j \leq s$. The matrix C will then look like the matrix in Equation (5) (in the even case) and so we define

$$\hat{C}(i,j) = C(i,j) \quad \text{for } i,j = 1,2,\ldots,s \tag{8}$$

Correspondingly, Proposition 6.1 follows.

How much did this reduce the order of C? If we let $a_v = |A \cap V|$ and $a_w = |A \cap W|$, then the number of sets to classify is

$$a = \sum_{k=0}^{\min(a_v,a_w)} \binom{a_v}{k}\binom{a_w}{k}$$

The order of $\hat{C}$ is then approximately $\frac{ar}{N}$. For the special case when $A = V \cup W$, the above sum is

$$a = \sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n} \sim \frac{2^{2n}}{\sqrt{\pi n}}$$

by Stirlings formula. We can then estimate the order of $\hat{C}$ to approximately $r/\sqrt{\pi n}$.

Henceforth, when we refer to $\hat{C}$ we mean that the appropriate reduction method has been applied. If G is bipartite as above, then we apply the reduction described for the bipartite case, and not merely the reduction in the even case.

7 Examples

In this section we apply the methods described above. What the examples also should demonstrate is that the method of polygraphs is very general and unless we can use a compression technique it does not give us good, i.e. short, recursion formulae. It does, however, deliver the *specific* polynomials and numbers

we desire, making tabulations of them fairly easy to carry through, even for rotagraphs, where the compression technique does not work.

At the same time we give a short demonstration of some of the functions in a Mathematica package, GrafPack, that are relevant to this article. The package is available on the web site `www.math.umu.se`. Download the entire GrafPack-directory, put it where Mathematica can see it (e.g. under ExtraPackages), start up Mathematica and type `<<GrafPack‘Master‘`. For an introduction to Mathematica, see [23]. The book by Skiena [19] is also recommended.

Example 7.1. To compute the matching polynomial of a graph, we use the recursive method described in Section 3.1. The matching polynomial of the 4-cube is produced with the command

```
MatchingPolynomial[Hypercube[4], x]
```

where `x` is a variable. This returns the polynomial

$$272 - 3712\,x^2 + 11648\,x^4 - 14208\,x^6 + 8256\,x^8 \\ -2496\,x^{10} + 400\,x^{12} - 32\,x^{14} + x^{16}$$

The number of matchings in the 4-cube, 41025, is returned by the command

```
NumberOfMatchings[Hypercube[4]]
```

To obtain the number of 1-factors in the 4-cube, type

```
NumberOfOneFactors[Hypercube[4]]
```

and we receive the constant term, 272, of the polynomial above. Since the 4-cube is bipartite the function computes the permanent of the biadjacency matrix. Had we entered a non-bipartite graph, the function would have used the recursive method of Section 3.1.

The permanent of a square matrix is computed with the Nijenhuis-Wilf method, see Section 3.2. This gives the permanent of the 10×10-matrix with zeroes on the diagonal and ones off the diagonal

```
Permanent[1 - IdentityMatrix[10]]
```

If we want to estimate the number of 1-factors in a fairly large graph, the probabilistic algorithm of Section 3.3 can be used. The command

```
EstimateNumberOfOneFactors[Hypercube[6], 1000]
```

takes the average of 1000 determinants of oriented (bi-)adjacency matrices. The integer should be chosen with care, as large as possible to get a reliable result, modulo how long the user is prepared to wait. In this example, the graph is bipartite so the function will orient only the bi-adjacency matrix. A run returned the estimate $1.8051 \cdot 10^{13}$. Being a probabilistic algorithm though, we will receive different results at different runs.

Example 7.2. We compute the matching polynomial and the number of 1-factors in the fasciagraph $\gamma_m = C_4 \times P_m$ using the compression technique. The subsets of $A = V(C_4) = \{\,1, 2, 3, 4\,\}$ sorts into 6 classes under the automorphism group of C_4 and the compressed matrix C then has order 6. Type

```
g = Cycle[4];
aut = Automorphisms[g];
orb = Orbits[aut, 2];
mat = CompressedTransferMatrixMP[g, orb, x]
```

The variable `orb` contains lists of isomorphic 2-colourings (their ranks to be precise) of the graph. The compressed matrix C, defined by Equation (4), is returned

$$\begin{pmatrix} 2-4x^2+x^4 & 8x-4x^3 & -4+4x^2 & 2x^2 & -4x & 1 \\ -2x+x^3 & 2-3x^2 & 2x & x & -1 & 0 \\ -1+x^2 & -2x & 1 & 0 & 0 & 0 \\ x^2 & -2x & 0 & 1 & 0 & 0 \\ x & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We continue the previous sequence of commands:

```
rec = RecursionCoefficients[mat];
r = Length[rec];
Clear[f];
Evaluate[Array[f, r]] = MatrixPower[mat, r, 1, 1, All];
f[m_] := f[m] = Sum[Expand[rec[[i]]*f[m-i]], {i, 1, r}];
```

If we try e.g. `f[7]` then $\mu(C_4 \times P_7; x)$ is returned.

The matrix for enumeration of matchings is given by

```
mat = CompressedTransferMatrixM[g, orb]
```

If we want $\Phi(\gamma_m)$, observe that the graph $C_4 = (V \cup W, E)$ is bipartite with $|V| = |W| = 2$. So we only need to classify those subsets $I \subseteq V \cup W$ such that $|I \cap V| = |I \cap W|$. There are only 6 such sets and they sort into 3 classes. Thus, the matrix $\hat{C}$ has order 3. This is all taken care of by the next function

```
mat = CompressedTransferMatrix1F[g, orb]
```

The matrix $\hat{C}$, defined by Equation (8), is returned

$$\begin{pmatrix} 2 & 4 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

To get a recursive formula for $\Phi(\gamma_m)$ we proceed as above and receive the following recursive formula

$$\Phi(\gamma_m) = 3\,\Phi(\gamma_{m-1}) + 3\,\Phi(\gamma_{m-2}) - \Phi(\gamma_{m-3})$$

We could of course solve this recursive relation to get an explicit formula for $\Phi(\gamma_m)$, but we leave this to the enthusiastic reader.

The recursive formulae above corresponds exactly to those obtained by Hosoya and Motoyama [9]. They also gave a recursive formula for $\Phi(P_2 \times P_3 \times P_m)$. Typing the last command sequence with `g=GridGraph[2,3]` will return exactly the same formula, namely

$$\begin{aligned} \Phi(\gamma_m) = 6\,\Phi(\gamma_{m-1}) + 21\,\Phi(\gamma_{m-2}) - 42\,\Phi(\gamma_{m-3}) \\ -89\,\Phi(\gamma_{m-4}) + 68\,\Phi(\gamma_{m-5}) + 89\,\Phi(\gamma_{m-6}) - 42\,\Phi(\gamma_{m-7}) \\ -21\,\Phi(\gamma_{m-8}) + 6\,\Phi(\gamma_{m-9}) + \Phi(\gamma_{m-10}) \end{aligned}$$

The authors of [9] estimated the order of the recursive formula for the matching polynomial to be approximately 20. This method would return one of order 24 which suits fairly well to their estimate.

We finish this example with a word of warning. Suppose that we replace the graph used above, C_4, with an odd graph, such as P_3, and generate the matrix $\hat{C}$. Then $\hat{C}^m(1,1) = \Phi(P_3 \times P_{2m})$ (!). Note also that the `RecursionCoefficients`-function returns the coefficients $\{5, -5, 1\}$, which should be interpreted as

$$\Phi(P_3 \times P_{2m}) = 5\,\Phi(P_3 \times P_{2m-2}) - 5\,\Phi(P_3 \times P_{2m-4}) + \Phi(P_3 \times P_{2m-6})$$

Example 7.3. Let $G = C_4$ and $X = \{(1,1),(2,2),(3,3),(4,4)\}$. Then $\omega_m = C_4 \times C_m$. To compute $\mu(\omega_4;\, x) = \mu(Q^4;\, x)$ type

```
g = Cycle[4];
rel = Table[{i,i},{i, 1, Order[g]}];
mat = TransferMatrixMP[g, rel, rel, x];
Sum[MatrixPower[mat, 4, i, i], {i, 1, Length[mat]}]
```

Here `rel` is the binary relation of edges between the graphs. Note that the built-in function `MatrixPower` has been extended to return particular entries. We could of course obtain recursive formulae for $\Phi(\omega_m)$ and $\mu(\omega_m;\, x)$ as above, but they would be unnecessarily long since they would both have order 16. In [9] a recursive formula for $\Phi(\omega_m)$ of order 8 was given, and the recursive formula for $\mu(\omega_m;\, x)$, was estimated to have order 10.

Example 7.4. In this example we scrutinize the 3-dimensional grids $P_4 \times P_4 \times P_m$. Let us first view it as the fasciagraph γ_m on $P_4 \times P_4$ with relation $X = \{\,(1,1),\ldots,(16,16)\,\}$. The matrix T has order 65536, which would require an enormous amount of computer memory to store. However, T will be very sparse. Since 16 vertices overlap in X only 3^{16} of the entries are non-zero and, if we only want 1-factors, fewer still are non-zero. The use of typical sparse matrix methods for computations of powers of T is of course a justified approach. Compression works well here, the automorphism group of $P_4 \times P_4$ has 8 elements and the order of C is 8548. This is still a trifle too big when we are storing polynomials in a computer. The matrix $\hat{C}$ on the other hand has order 1723, as computations have shown, and this is not too big to treat easily. Note that only the elements $\hat{C}^m(1,1)$ are desired, and so only vector-matrix multiplication needs to be performed. This approach does not bring us the matching polynomials of γ_m, but for smaller m we can use a rotagraph approach. For the case $m = 4$ we let $G = P_2 \times P_2 \times P_4$ and $X = \{(3,3),(4,2),(7,7),(8,6),(11,11),(12,10),(15,15),(16,14)\}$, see Figure 2. The rotagraph on (G, X) is the cubic grid $P_4 \times P_4 \times P_4$. The matrix T has order 256, which is fairly easily treated. The polynomial is listed in the Tables section. To compute it type

```
g = GridGraph[2, 2, 4];
rel = {{3,3},{4,2},{7,7},{8,6},{11,11},{12,10},{15,15},{16,14}};
mat = TransferMatrixMP[g, rel, rel, x, Verbose->True];
Sum[MatrixPower[mat, 4, i, i], {i, 1, Length[mat]}]
```

Note that adding the option `Verbose->True` as a last argument of the function `TransferMatrixMP` shows the progress of the computations. This makes the waiting for the computations to finish more bearable.

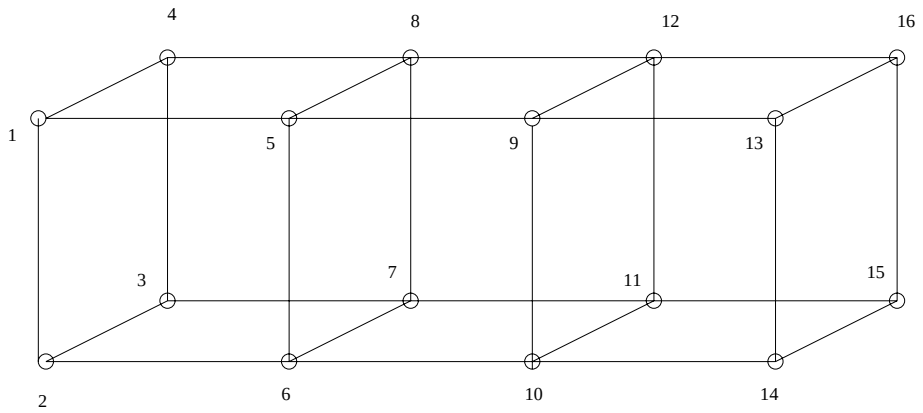


Figure 2: The $2 \times 2 \times 4$-grid

Example 7.5. We continue here the rotagraph approach from the previous example and describe a method for computing the entries in the transfer matrix. Let $G = P_2 \times P_2 \times P_4$ and X be the relation given earlier. We will view G as a fasciagraph on $H = P_2 \times P_2$ with the relation $Y = \{(1,1),(2,2),(3,3),(4,4)\}$ between each copy of H, refer to these copies as $H_1, \ldots, H_4$. Let $A \subseteq R(X)$ and $B \subseteq D(X)$ and say that this pair of sets corresponds to the (i,j)th entry in the transfer matrix T that we are aiming for. If $A \cap B \neq \emptyset$ then $T(i,j) = 0$, otherwise we wish to compute $T(i,j) = \Phi(G - A - B)$. We will do this with transfer matrices though we will forbid the vertices $A \cup B$. To do this we define a family of transfer matrices, one for each possible set of vertices that intersect $V(H_k)$. Let $U_k = (A \cup B) \cap V(H_k)$ for $k = 1, \ldots, 4$. Since $A \cup B$ intersects each H_k in at most 3 vertices there are only 2^3 different sets U_k. To compute $\Phi(G)$, we would normally use the matrix in Equation (3). Instead we define a modified matrix as follows; for all $A_i, A_j \subseteq V(H)$ let

$$S_U(i,j) = \begin{cases} \Phi(H - U - A_i - A_j), & \text{if } U \cap (A_i \cup A_j) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Now it is easy to see that $T(i,j) = [S_{U_1} \cdots S_{U_4}](1,1)$. If we scale our problem to $G = P_3 \times P_3 \times P_6$ then we let $H = P_3 \times P_3$ and produce the necessary 2^5 matrices S in advance, each a 512×512 matrix. These matrices will be extremely sparse so sparse matrix methods are very beneficial and there will be no problem in storing them on a computer. This approach was implemented in Fortran to compute $\Phi(P_6 \times P_6 \times P_n)$ for $n = 1, \ldots, 5$, (so the case with $n = 6$ is still difficult) and $\mu(P_5 \times P_5 \times P_n)$ for $n = 1, \ldots, 5$, see the Tables section.

Example 7.6. The n-cube, denoted Q^n, is the graph having the set of binary strings of length n as vertices. Two vertices are adjacent if their binary strings differ in exactly one position. Note that $Q^n = Q^{n-1} \times P_2$ and $Q^n = Q^{n-2} \times C_4$. We will view Q^6 as the rotagraph $Q^4 \times C_4$ and proceed to compute $\Phi(Q^6)$ and $\mu(Q^6)$. Note that a transfer matrix for this rotagraph has order $2^{16} = 65\,536$. However, the transfer matrix for counting 1-factors has only $5\,494\,273$ non-zero entries and the matrix for counting matchings has $3^{16} = 43\,046\,721$ non-zero

entries. Thus storage in a computer memory is possible on a larger workstation by using standard sparse matrix methods. Recall that $\mathrm{tr}(T^4)$ is the desired number. Again we may use the automorphisms of Q^4 to reduce the amount of work. Let $\mathcal{A}_1, \ldots, \mathcal{A}_{402}$ be the equivalence classes of $V = V(Q^4)$ and note that every row (and column) of T corresponds to a subset of V. Let A_i be a member of $\mathcal{A}_i$ for $i = 1, \ldots, 402$. We have

$$\mathrm{tr}(T^4) = \sum_{I \subseteq V} T^4(I, I) = \sum_{i=1}^{402} |\mathcal{A}_i|\, T^4(A_i, A_i)$$

Fortran implementations of this approach gave $\Phi(Q^6) = 16332454526976$ and $\mu(Q^6) = 7174574164703330195841$. A smaller example of the sum above is given by the following computation of $\Phi(Q^4)$:

```
g = Hypercube[2];
rel = Table[{i, i}, {i, 1, Order[g]}];
aut = Automorphisms[g];
orb = Orbits[aut, 2];
mat = TransferMatrix1F[g, rel, rel];
Sum[
    i = 1 + orb[[k]];
    Length[orb[[k]]]*MatrixPower[mat, 4, i, i],
    {k, 1, Length[orb]}
]
```

Note that the ranks of the 2-colourings are counted from zero but the indices of the matrix are counted from one, which explains the definition of `i`. The number of matchings and the matching polynomials can also be computed this way.

We should remark that the matching polynomial of the 6-cube, for completeness listed in the Tables-section, was computed with a rather different approach; first compute the Ising partition function in two variables and extract the matching polynomial from it. This method will be described in some future paper.

8 Tables

"This process of reduction to cipher is the highest effort man or woman is capable of making. It is the only effort worth making, and it is possible only through ever-increasing self-restraint..."

Gandhi, 1927.

The matching polynomials and the number of 1-factors has been extensively tabulated for various grids, cylinders and tori. General expressions exist for the number of 1-factors in graphs such as $P_m \times P_n$, $P_m \times C_n$, $C_m \times C_n$, $P_2 \times P_3 \times P_m$. The papers by Hosoya et al. [7, 8, 9, 10, 11] contain plenty of tables and general expressions, to which we refer the reader. Fans of integer sequences might want to consult the book by Sloane and Plouffe [20], which also can be reached on the Internet as a searchable database at `http://www.research.att.com/~njas/sequences/`. Below is listed tables of

$p(G,k)$, $\Phi(G)$, $\mu(G)$ and recurrence relations for some fasciagraphs on smaller cycles, grids and hypercubes. They were generated by running a precursor of GrafPack on a Power Macintosh 8100/80. In the tables of $p(G,k)$, integers being the number of 1-factors are printed in bold. To simplify the recurrence relations we let μ_m denote $\mu(\gamma_m; x)$ and Φ_m denote $\Phi(\gamma_m)$. Let also r denote the order of the compressed matrix C for matching polynomials and $\hat{r}$ the order of the compressed (and reduced) matrix $\hat{C}$ for 1-factors.

Table 1: Order of compressed matrices for some $G \times P_m$

G	r	$\hat{r}$	G	r	$\hat{r}$	G	r	$\hat{r}$
$P_2 \times P_3$	24	10	P_2	3	2	C_3	4	2
$P_2 \times P_4$	76	27	P_3	6	3	C_4	6	3
$P_2 \times P_5$	288	82	P_4	10	5	C_5	8	4
$P_2 \times P_6$	1072	268	P_5	20	10	C_6	13	6
$P_3 \times P_3$	102	51	P_6	36	14	C_7	18	9
$P_3 \times P_4$	1120	274	P_7	72	36	C_8	30	11
$P_4 \times P_4$	8548	1723	P_8	136	43	C_9	46	23
$C_3 \times C_3$	26	13	P_9	272	136	C_{10}	78	26
Q^3	22	9	P_{10}	528	142	C_{11}	126	63
Q^4	402	93	P_{11}	1056	528	C_{12}	224	62

Table 2: $P_5 \times P_5 \times P_m$

m	μ
1	2810694
2	423657524608288
3	4212722192548586089679 2
4	4435122353330774501960785797973
5	463310369790129032480118384076035223552

Table 3: $P_6 \times P_6 \times P_m$

m	Φ
1	6728
2	53786626921
3	57248060375968384
4	123115692449982216049513
5	216388579168758145017797108072

Table 4: $C_3 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$
0	1	1	1	1	1	1	1	1	1
1	3	9	15	21	27	33	39	45	51
2		18	69	156	279	438	633	864	1131
3		**4**	107	501	1399	3017	5571	9277	14351
4			36	672	3558	11613	29049	61374	115392
5				285	4338	25029	92109	259956	615348
6				**19**	2100	28557	175363	709740	2214051
7					276	15072	190575	1226919	5363931
8						2880	106824	1284651	8582760
9						**91**	25978	752716	8726408
10							1818	216951	5289783
11								23754	1730235
12								**436**	255239
13									11085
μ	4	32	228	1655	11978	86731	627960	4546684	32919766

Table 5: $C_4 \times P_m = Q^2 \times P_m = P_2 \times P_2 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
0	1	1	1	1	1	1	1	1
1	4	12	20	28	36	44	52	60
2	**2**	42	142	306	534	826	1182	1602
3		44	440	1672	4248	8680	15480	25160
4		**9**	588	4863	19774	56333	129644	258907
5			288	7416	55200	235132	728840	1840836
6			**32**	5470	91200	637914	2810312	9294734
7				1620	84984	1112668	7465728	33741064
8				**121**	40553	1208714	13541312	88199495
9					8204	771436	16397296	164774936
10					**450**	261500	12752616	216370582
11						39080	5986432	194313364
12						**1681**	1532336	114468886
13							178272	41514628
14							**6272**	8380100
15								788536
16								**23409**
μ	7	108	1511	21497	305184	4334009	61545775	873996300

Table 6: $C_5 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$
0	1	1	1	1	1	1	1
1	5	15	25	35	45	55	65
2	5	75	240	505	870	1335	1900
3		145	1125	3910	9495	18880	33065
4		95	2710	17725	64660	173020	382305
5		**11**	3227	48193	286799	1081285	3103896
6			1645	77405	839930	4723695	18237825
7			240	69510	1612685	14550495	78786505
8				31060	1975730	31488555	251718625
9				5360	1465295	47151280	593631680
10				**176**	598928	47476226	1023782605
11					113015	30669915	1268978075
12					6625	11778955	1100004130
13						2360195	639919835
14						191480	234612615
15						**2911**	49020224
16							4885170
17							153830
μ	11	342	9213	253880	6974078	191668283	5267252351

Table 7: $C_6 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$
0	1	1	1	1	1	1	1
1	6	18	30	42	54	66	78
2	9	117	363	753	1287	1965	2787
3	**2**	336	2290	7562	17874	34954	60530
4		420	8139	46938	160887	414792	894189
5		192	16446	187530	987834	3472752	9527094
6		**20**	18141	487241	4241321	21158661	75753275
7			9870	813486	12846774	95402040	458907006
8			2148	843342	27359544	320645463	2143757547
9			**108**	509542	40372976	803176510	7768505882
10				160653	40170300	1489152993	21861085377
11				21438	25795320	2015817270	47616569682
12				**725**	9980480	1949485107	79675739431
13					2078160	1304474898	101182136226
14					188832	576346062	95821362789
15					**4480**	156728330	66035085642
16						23429940	32011697004
17						1566180	10405152504
18						**28561**	2112964124
19							239567604
20							12371220
21							**179928**
μ	18	1104	57536	3079253	164206124	8761336545	467431319920

Table 8: $P_3 \times P_3 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
0	1	1	1	1	1	1
1	12	33	54	75	96	117
2	44	436	1260	2525	4231	6378
3	56	2984	16736	50552	113684	215393
4	18	11434	140322	672126	2085694	5054442
5		24766	778452	6277198	27731168	87622530
6		29180	2913096	42480118	276805102	1164755616
7		16984	7361472	211846420	2120333560	12163620462
8		3993	12381180	784200907	12634826746	101433879357
9		**229**	13428840	2154366513	59027097072	682916407521
10			8893248	4362041263	216913695094	3738673165242
11			3278784	6419477292	626708528128	16712392258753
12			568344	6716664818	1417900872204	61103060700766
13			31344	4835018662	2493032893120	182629834939538
14				2281569082	3367348279396	445089189580448
15				655842108	3437515277416	880370659944042
16				101934041	2593501127101	1403576812451606
17				6870327	1402515949328	1786799130667754
18				**117805**	520871037067	1793930275383832
19					124842772364	1397774304403158
20					17531745326	827727493314932
21					1217704320	362423901173076
22					28613174	113077255268116
23						23878571601956
24						3164202873629
25						233176559173
26						7654682266
27						**64647289**
μ	131	90040	49793133	28579431833	16294017491392	9303034425177393

Table 9: $C_3 \times C_3 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
0	1	1	1	1	1	1
1	18	45	72	99	126	153
2	99	810	2241	4401	7290	10908
3	180	7518	39678	116316	257106	481731
4	72	38709	442575	2039814	6188463	14778099
5		110817	3254724	25088310	107856216	334725885
6		167448	16056147	223066398	1409411676	5808709002
7		117900	53046918	1456699500	14108774220	79104051891
8		29520	115246440	7029374175	109615427955	858999657429
9		**1120**	158653112	25022727081	665714322238	7517635432505
10			129944880	65127684555	3168417127554	53381488744872
11			56958480	121909424148	11801137694058	308693456717967
12			10992408	159953324046	34221545160489	1455432762661803
13			585792	141626935710	76569860426940	5588494400657529
14				80001899586	130436645000040	17417917114151796
15				26440161960	166051546684152	43821565164155937
16				4418860545	154011257081100	88290020235183381
17				278666595	100510188513840	140932058555779443
18				**2861029**	43956690488688	175746115986201690
19					11993327746128	168125848472949201
20					1823418619560	120495553386274359
21					126181749120	62707121963709243
22					2535163200	22712557651235100
23						5392873133377065
24						767195930393457
25						56362288663467
26						1606470279210
27						**7537209013**
mu	370	473888	545223468	633518934269	735463713700160	853881267896192137

Table 10: $P_4 \times P_4 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$
0	1	1	1	1
1	24	64	104	144
2	224	1816	4992	9768
3	1044	30208	146940	415368
4	2593	328214	2972395	12430848
5	3388	2456736	43888740	278659560
6	2150	13022504	490410658	4862322484
7	552	49492032	4243096376	67752463152
8	**36**	135062729	28849000711	767471193606
9		262610832	155554203920	7157834054584
10		357580896	668490123332	55469187090396
11		331384336	2293235516668	359485412847192
12		200032432	6270624556725	1956911884067608
13		73483328	13607937421412	8971759857716256
14		14707328	23264863112266	34682805390128328
15		1308928	31002090496224	113035590354067768
16		**32000**	31731778597928	310146213937970487
17			24460558393664	714514530994393464
18			13831123293040	1376672261486529068
19			5534768640848	2206488832067036760
20			1490639531680	2921624380278645192
21			250915666208	3168204916452408416
22			23455372800	2783182424023411992
23			980808000	1953962180835361272
24			**10885344**	1077824850339404286
25				457155298292389608
26				144991813332269700
27				33134934405040272
28				5183929033351776
29				515240510630328
30				28894756833940
31				736291240776
32				**5051532105**
μ	10012	1441534384	154620656140976	17312701462385916505

Table 11: $Q^4 \times P_m = C_4 \times C_4 \times P_m$

k	$m=1$	$m=2$	$m=3$	$m=4$
0	1	1	1	1
1	32	80	128	176
2	400	2840	7568	14600
3	2496	59120	274560	759584
4	8256	803580	6848000	27822084
5	14208	7517264	124694656	763504368
6	11648	49715240	1718209088	16311133584
7	3712	235146480	18327675008	278274362192
8	**272**	795862790	153549653616	3858979023370
9		1910146160	1019460142080	44051088838656
10		3190117800	5389069021056	417676281992856
11		3594554960	22710637612800	3310348880868432
12		2605908220	76162736983680	22024174794317232
13		1129177840	202303330851072	123313091919432144
14		259084440	422310466869504	581630577946974072
15		25108944	685115567624704	2310324639457748096
16		**589185**	850667743539584	7715963153250311251
17			792016077516800	21604808702631926656
18			538003442426880	50504855552895180056
19			256874061012992	98016417871417039760
20			81810395008768	156788269717168962800
21			16087147553792	204849983435540593552
22			1725682248704	216149310892878810872
23			80406638592	181614258291882122496
24			**930336768**	119387717864796680906
25				60042777844937606416
26				22443085396359803280
27				5999543286903760304
28				1087639382471943076
29				123724794351752480
30				7805441127361896
31				217782023223920
32				**1545853411969**
μ	41025	13803794944	3952450882750401	1149377449671217283137

Table 12: $Q^6 = C_4 \times C_4 \times C_4$

k	$p(Q^6, k)$
0	1
1	192
2	17376
3	986240
4	39408480
5	1179696384
6	27488385408
7	511416198144
8	7732531647360
9	96216012236800
10	994137263758848
11	8583228570909696
12	62184244929659648
13	378969619199569920
14	1944655398731796480
15	8398980067449999360
16	30480925212093104640
17	92675048634081607680
18	235053748112782356480
19	494482501391128289280
20	856482708316893954048
21	1210188907641505775616
22	1378948882982541631488
23	1249011213103104491520
24	883258965992225095680
25	476635207372408553472
26	190551239146197909504
27	54258655709480353792
28	10420946627414016000
29	1246585402333593600
30	81808261704974336
31	2333280165691392
32	**16332454526976**
μ	7174574164703330195841

8.1 Recursion formulae

$$\Phi(C_3 \times P_{2m}) = 5\,\Phi_{2m-2} - \Phi_{2m-4}$$

$$\mu(C_3 \times P_m) = 6\,\mu_{m-1} + 9\,\mu_{m-2} - 1\,\mu_{m-4}$$

$$\mu(C_3 \times P_m;\, x) = (-5\,x + x^3)\mu_{m-1} + (-5 + 3\,x^2 - x^4)\mu_{m-2} + (x + x^3)\mu_{m-3} - \mu_{m-4}$$

$$\Phi(C_4 \times P_m) = 3\,\Phi_{m-1} + 3\,\Phi_{m-2} - \Phi_{m-3}$$

$$\mu(C_4 \times P_m) = 14\,\mu_{m-1} + 6\,\mu_{m-2} - 46\,\mu_{m-3} + 18\,\mu_{m-4} + 2\,\mu_{m-5} - 1\,\mu_{m-6}$$

$$\begin{aligned}\mu(C_4 \times P_m;\, x) = {} & (6 - 7x^2 + x^4)\,\mu_{m-1} + (-7 - 6x^2 + 6x^4 - x^6)\,\mu_{m-2} \\ & + (-8 + 26x^2 - 10x^4 + 2x^6)\,\mu_{m-3} + (9 - 6x^2 + 2x^4 - x^6)\,\mu_{m-4} \\ & + (2 + x^2 + x^4)\,\mu_{m-5} - \mu_{m-6}\end{aligned}$$

$$\Phi(C_5 \times P_{2m}) = 19\,\Phi_{2m-2} - 41\,\Phi_{2m-4} + 19\,\Phi_{2m-6} - \Phi_{2m-8}$$

$$\begin{aligned}\mu(C_5 \times P_m) = {} & 25\,\mu_{m-1} + 76\,\mu_{m-2} - 209\,\mu_{m-3} - 159\,\mu_{m-4} + 119\,\mu_{m-5} \\ & + 40\,\mu_{m-6} - 3\,\mu_{m-7} - 1\,\mu_{m-8}\end{aligned}$$

$$\begin{aligned}\mu(C_5 \times P_m;\, x) = {} & (15\,x - 9\,x^3 + x^5)\mu_{m-1} + (-19 + 19\,x^2 - 27\,x^4 + 10\,x^6 - x^8)\mu_{m-2} \\ & + (34\,x - 85\,x^3 + 69\,x^5 - 19\,x^7 + 2\,x^9)\mu_{m-3} + (-41 + 95\,x^2 - 39\,x^4 - 9\,x^6 \\ & + 6\,x^8 - x^{10})\mu_{m-4} + (2\,x - 65\,x^3 + 39\,x^5 - 11\,x^7 + 2\,x^9)\mu_{m-5} \\ & + (-19 + 11\,x^2 - 7\,x^4 + 2\,x^6 - x^8)\mu_{m-6} + (3\,x + x^3 + x^5)\mu_{m-7} - \mu_{m-8}\end{aligned}$$

$$\Phi(C_6 \times P_m) = 4\,\Phi_{m-1} + 16\,\Phi_{m-2} - 6\,\Phi_{m-3} - 16\,\Phi_{m-4} + 4\,\Phi_{m-5} + \Phi_{m-6}$$

$$\begin{aligned}\mu(C_6 \times P_m) = {} & 53\,\mu_{m-1} + 66\,\mu_{m-2} - 2616\,\mu_{m-3} + 5076\,\mu_{m-4} + 5806\,\mu_{m-5} \\ & - 14388\,\mu_{m-6} + 1276\,\mu_{m-7} + 6022\,\mu_{m-8} - 1420\,\mu_{m-9} - 424\,\mu_{m-10} \\ & + 90\,\mu_{m-11} + 5\,\mu_{m-12} - 1\,\mu_{m-13}\end{aligned}$$

$$\begin{aligned}\mu(C_6 \times P_m;\, x) = {} & (-12 + 29\,x^2 - 11\,x^4 + x^6)\mu_{m-1} + (-32 + 12\,x^2 + 47\,x^4 - 49\,x^6 \\ & + 13\,x^8 - x^{10})\mu_{m-2} + (71 - 568\,x^2 + 948\,x^4 - 714\,x^6 + 266\,x^8 - 46\,x^{10} + 3\,x^{12})\mu_{m-3} \\ & + (313 - 983\,x^2 + 1261\,x^4 - 1339\,x^6 + 848\,x^8 - 283\,x^{10} + 46\,x^{12} - 3\,x^{14})\mu_{m-4} \\ & + (40 + 924\,x^2 - 2103\,x^4 + 1956\,x^6 - 812\,x^8 + 97\,x^{10} + 34\,x^{12} - 11\,x^{14} + x^{16})\mu_{m-5} \\ & + (-601 + 2884\,x^2 - 4334\,x^4 + 3559\,x^6 - 1903\,x^8 + 823\,x^{10} - 241\,x^{12} + 40\,x^{14} \\ & - 3\,x^{16})\mu_{m-6} + (-311 + 1132\,x^2 - 470\,x^4 + 161\,x^6 + 259\,x^8 - 351\,x^{10} + 153\,x^{12} \\ & - 32\,x^{14} + 3\,x^{16})\mu_{m-7} + (368 - 892\,x^2 + 1743\,x^4 - 1764\,x^6 + 968\,x^8 - 265\,x^{10} \\ & + 26\,x^{12} + 3\,x^{14} - x^{16})\mu_{m-8} + (251 - 529\,x^2 + 575\,x^4 - 205\,x^6 - 60\,x^8 + 59\,x^{10} \\ & - 18\,x^{12} + 3\,x^{14})\mu_{m-9} + (-47 - 172\,x^4 + 130\,x^6 - 58\,x^8 + 14\,x^{10} - 3\,x^{12})\mu_{m-10} \\ & + (-40 + 28\,x^2 - 11\,x^4 + 9\,x^6 - x^8 + x^{10})\mu_{m-11} + (-5\,x^2 - x^4 - x^6)\mu_{m-12} + \mu_{m-13}\end{aligned}$$

References

[1] D. Babic et al., *The matching polynomial of a polygraph*, Discrete Appl. Math. **15** (1986) 11–24

[2] H.J. Broersma and Li Xueliang, *On 'The matching polynomial of a polygraph'*, Discrete Appl. Math. **46** (1993) 79–86

[3] B.A. Cipra, *An introduction to the Ising model*, Amer. Math. Monthly **94** 937–959

[4] C.D. Godsil, *Algebraic combinatorics*, Chapman and Hall, 1993

[5] R. Graham, M. Grötschel and L. Lovasz (editors), *The Handbook of Combinatorics*, Elsevier Science B. V., 1995

[6] O.J. Heilmann and E.H. Lieb, *Theory of monomer-dimer systems*, Commun. Math. Phys. **25** (1972) 190–232

[7] H. Hosoya, *On some counting polynomials in chemistry*, Discrete Appl. Math. **19** (1988) 239–257

[8] H. Hosoya, *Matching and symmetry of graphs*, Comp. and Maths. with Appls. **12B** (1986) 271–290

[9] H. Hosoya and A. Motoyama, *An effective algorithm for obtaining polynomials for dimer statistics. Application of operator technique on the topological index to two- and three-dimensional rectangular and torus lattices*, J. Math. Phys. **26** (1985) 157–167

[10] H. Hosoya et al., *Generalized expression of the perfect matching number for* $2 \times 3 \times n$ *lattices*, J. Math. Phys. **34** (1993) 1043–1051

[11] H. Hosoya et al., *Generalized expression for the number of perfect matchings of cylindrical* $m \times n$ *graphs*, J. Math. Phys. **32** (1991) 1885–1889

[12] P.W. Kasteleyn, *Graph theory and crystal physics*, Graph theory and theoretical physics, ed. F. Harary, Academic Press, London, 1967.

[13] N. Karmarkar et al., *A Monte-Carlo algorithm for estimating the permanent*, SIAM J. Comput. **22** (1993) 284–293

[14] L. Lovasz and M.D. Plummer, *Matching theory*, North-Holland, 1986

[15] H. Minc, *Permanents*, Addison-Wesley, 1978

[16] A. Nijenhuis and H. Wilf, *Combinatorial algorithms*, Academic Press, 1978

[17] D. H. Rouvray, *The modeling of chemical phenomena using topological indices*, J. Comput. Chem., **8**, (1987) 470–480

[18] H.J. Ryser, *Combinatorial mathematics*, The Carus mathematical monographs, 1963

[19] S.S. Skiena, *Implementing discrete mathematics: Combinatorics and graph theory with Mathematica*, Addison-Wesley, 1990

[20] N.J.A. Sloane and S. Plouffe, *The encyclopedia of integer sequences*, Academic Press, 1995

[21] N. Trinajstić, *Chemical graph theory*, CRC Press, 1992

[22] D. J. A. Welsh, *Complexity: Knots, colourings and counting*, Cambridge university press, 1993

[23] S. Wolfram, *Mathematica: a system for doing mathematics by computer*, Addison-Wesley, 1991

Thesis for the Degree of Doctor of Philosophy

Generalized Patterns in Words and Permutations

Sergey Kitaev

CHALMERS | GÖTEBORG UNIVERSITY

Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, January 2003

Generalized patterns in words and permutations
Sergey Kitaev
ISBN 91-628-5521-2

Cover: "Counting occurrences of generalized patterns",
designed and created by Toufik Mansour.

Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, Sweden, January 2003

Abstract

The thesis consists of the following nine papers:

I *Multi-avoidance of generalized patterns.* (*Discrete Mathematics*, to appear) Recently, Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. We investigate simultaneous avoidance of two or more 3-patterns without internal dashes, that is, where the pattern corresponds to a contiguous subword in a permutation.

II *Generalized pattern avoidance with additional restrictions.* (Séminaire Lotharingien de Combinatoire, to appear) We consider n-permutations that avoid the generalized pattern 1-32 and whose k rightmost letters form an increasing subword. The number of such permutations is a linear combination of Bell numbers. We find a bijection between these permutations and all partitions of an $(n-1)$-element set with one subset marked that satisfy certain additional conditions. Also we find the e.g.f. for the number of permutations that avoid a generalized 3-pattern with no dashes and whose k leftmost or k rightmost letters form either an increasing or decreasing subword. Moreover, we find a bijection between n-permutations that avoid the pattern 132 and begin with the pattern 12 and increasing rooted trimmed trees with $n+1$ nodes.

III *Simultaneous avoidance of generalized patterns* (joint work with Toufik Mansour). In [Kit1] Kitaev considered simultaneous avoidance (multi-avoidance) of two or more 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. There either an explicit or a recursive formula was given for all but one case of simultaneous avoidance of more than two patterns. In this paper we find the exponential generating function for the remaining case. Also we consider permutations that avoid a pattern of the form x-yz or xy-z and begin with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $23\dots k1$, $(k-1)(k-2)\dots 1k$ or end with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $1k(k-1)\dots 2$, $k12\dots(k-1)$. For each of these cases we find either the ordinary or exponential generating functions or a precise formula for the number of such permutations. Besides we generalize some of the obtained results as well as some of the results given in [Kit3]: we consider permutations avoiding certain generalized 3-patterns and beginning (ending) with an arbitrary pattern having either the greatest or the least letter as its rightmost (leftmost) letter.

IV *On multi-avoidance of generalized patterns* (joint work with Toufik Mansour). In [Kit1] Kitaev discussed simultaneous avoidance of two 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. In three essentially different cases, the numbers of such n-permutations are 2^{n-1}, the number of involutions

in $\mathcal{S}_n$, and $2E_n$, where E_n is the n-th Euler number. In this paper we give recurrence relations for the remaining three essentially different cases.

To complete the descriptions in [Kit3] and [KitMans1], we consider avoidance of a pattern of the form x-y-z (a classical 3-pattern) and beginning or ending with an increasing or decreasing pattern. Moreover, we generalize this problem: we demand that a permutation must avoid a 3-pattern, begin with a certain pattern and end with a certain pattern simultaneously. We find the number of such permutations in case of avoiding an arbitrary generalized 3-pattern and beginning and ending with increasing or decreasing patterns.

V *Partially Ordered Generalized Patterns.* (*Discrete Mathematics*, to appear) We introduce partially ordered generalized patterns (POGPs), which further generalize the generalized permutation patterns (GPs) introduced by Babson and Steingrímsson. A POGP p is a GP some of whose letters are incomparable. Thus, in an occurrence of p in a permutation π, two letters that are incomparable in p pose no restrictions on the corresponding letters in π. We describe many relations between POGPs and GPs and give general theorems about the number of permutations avoiding certain classes of POGPs. These theorems have several known results as corollaries but also give many new results. We also give the generating function for the entire distribution of the maximum number of non-overlapping occurrences of a pattern p with no dashes, provided we know the e.g.f. for the number of permutations that avoid p.

VI *Partially ordered generalized patterns and k-ary words* (joint work with Toufik Mansour). We study the generating functions (g.f.) for the number of k-ary words avoiding some POGPs. We give analogues, extend and generalize several known results, as well as get some new results. In particular, we give the g.f. for the entire distribution of the maximum number of non-overlapping occurrences of a pattern p with no dashes (that allowed to have repetition of letters), provided we know the g.f. for the number of k-ary words that avoid p.

VII *Counting the occurrences of generalized patterns in words generated by a morphism* (joint work with Toufik Mansour). We count the number of occurrences of certain patterns in given words. We choose these words to be the set of all finite approximations of a sequence generated by a morphism with certain restrictions. The patterns in our considerations are either classical patterns 1-2, 2-1, 1-1-$\cdots$-1, or arbitrary generalized patterns without internal dashes, in which repetitions of letters are allowed. In particular, we find the number of occurrences of the patterns 1-2, 2-1, 12, 21, 123 and 1-1-$\cdots$-1 in the words obtained by iterations of the morphism $1 \to 123$, $2 \to 13$, $3 \to 2$, which is a classical example of a morphism generating a nonrepetitive sequence.

VIII *The Peano curve and counting occurrences of some patterns* (joint work with Toufik Mansour). We introduce *Peano words*, which are words corresponding to finite approximations of the Peano space filling curve. We then find the number of occurrences of certain patterns in these words.

IX *The sigma-sequence and counting occurrences of some patterns, subsequences and subwords.* We consider *sigma-words*, which are words used by Evdokimov in the construction of the sigma-sequence [Evdok1]. We then find the number of occurrences of certain patterns, subsequences and subwords in these words.

Key words and phrases. Generalized pattern avoidance, partially ordered generalized patterns, occurrence of a pattern in a word or permutation, iterated morphism, Peano curve, sigma-sequence, Dragon curve

AMS 2000 subject classification: 05A05, 05A15, 05A18, 68R15

Acknowledgements

First of all I would like to thank my advisor Einar Steingrímsson for his knowledge, enthusiasm, generosity and sense of humour. I would like to thank him for being ready to help me at a moment's notice with professional advice, and for making me feel very welcome during my stay in Sweden. Without doubt he is one of the greatest people I have ever met. Thank you Einar very much for everything you did for me!

I would like to thank Alexander Evdokimov for suggesting interesting unsolved problems to me, his advice, his permanent interest in my work and his encouragement and support during my student life. Also, my thanks go to Toufik Mansour for being such a great collaborator, for fruitful discussions, interesting, sometimes crazy, ideas, and for being the first person who has read this thesis.

I thank all the people at the Department of Mathematics at Chalmers University of Technology and Göteborg University, all the people at the Laboratory of Discrete Analysis at the Sobolev Institute of Mathematics and everybody who plays traditional Friday football with me for creating such a friendly atmosphere.

Special thanks to all my friends for making my life full of fun.

I would like to thank my mother, father, grandmother and grandfather (even though he is not among us anymore) for their invaluable support throughout all my life. My thanks go to the Shuiskii family, especially to Sergei Ivanovich, for interesting debates and helpful advice. Also, I would like to thank my aunt Valentina Semenova for being such a nice person.

The final acknowledgement goes to my wonderful wife Daria, for her love and inspiration. It is to her that I dedicate this thesis.

Sergey Kitaev
Göteborg, December 2002

Contents

In mathematics, if a pattern occurs, we can go on to ask, Why does it occur? What does it signify? And we can find answers to these questions. In fact, for every pattern that appears, a mathematician feels he ought to know why it appears.

– W. W. Sawyer

Introduction

In the last decade a wealth of papers has been written on the subject of pattern avoidance in permutations, also known as the study of "restricted permutations" and "permutations with forbidden subsequences." This topic is the main focus of the present thesis (the first five papers are about this). In the sixth paper, which extends and generalizes the fifth paper, we study certain patterns in k-ary words. The last three papers are dedicated to counting occurrences of certain patterns in certain words related to sequences generated by *morphisms*, the *Peano curve* and the *sigma-sequence*, respectively.

0.1 Permutation patterns

We write permutations as words $\pi = a_1 a_2 \cdots a_n$, whose letters are distinct and usually consist of the integers $1, 2, \ldots, n$.

An occurrence of a pattern τ in a permutation π is "classically" defined as a subsequence in π (of the same length as τ) whose letters are in the same relative order as those in τ. Formally speaking, for $r \leq n$, we say that a permutation σ in the symmetric group $\mathcal{S}_n$ has an occurrence of the pattern $\tau \in \mathcal{S}_r$ if there exist $1 \leq i_1 < i_2 < \cdots < i_r \leq n$ such that $\tau = \sigma(i_1)\sigma(i_2)\ldots\sigma(i_r)$ in reduced form. The *reduced form* of a permutation σ on a set $\{j_1, j_2, \ldots, j_r\}$, where $j_1 < j_2 < \cdots < j_r$, is the permutation σ_1 obtained by renaming the letters of the permutation σ so that j_i is renamed i for all $i \in \{1, \ldots, r\}$. For example, the reduced form of the permutation 3651 is 2431.

We denote by $\mathcal{S}_n(\tau)$ the set of all permutations in $\mathcal{S}_n$ which *avoid* τ, that is have no occurrences of τ. If $R = \{\tau_1, \tau_2, \ldots, \tau_k\}$, we let

$$\mathcal{S}_n(R) = \bigcap_{1 \leq i \leq k} \mathcal{S}_n(\tau_i).$$

The *reverse* $\mathcal{R}(\pi)$ of a permutation $\pi = a_1 a_2 \ldots a_n$ is the permutation $a_n a_{n-1} \ldots a_1$. The *complement* $C(\pi)$ is the permutation $b_1 b_2 \ldots b_n$ where $b_i = n + 1 - a_i$. Also, $\mathcal{R} \circ C$ is the composition of $\mathcal{R}$ and C. For example, $\mathcal{R}(13254) = 45231$, $C(13254) = 53412$ and $\mathcal{R} \circ C(13254) = 21435$. We call these bijections of $\mathcal{S}_n$ to itself *trivial*, and it is easy to see that for any pattern τ the number $|\mathcal{S}_n(\tau)|$ of permutations avoiding the pattern τ is the same as for the patterns $\mathcal{R}(\tau)$, $C(\tau)$ and $\mathcal{R} \circ C(\tau)$. For example, the number of permutations that avoid the pattern 132 is the same as the number of permutations that avoid the pattern 231. This property holds for sets of patterns as well. If we apply one of the trivial bijections to all patterns of a set R, then we get a set R' for which $|\mathcal{S}_n(R')|$ is equal to $|\mathcal{S}_n(R)|$. For example, the number of permutations avoiding $\{123, 132\}$ equals the number of those avoiding $\{321, 312\}$ because the second set is obtained from the first one by complementing each pattern.

Fundamental questions are to determine $|\mathcal{S}_n(R)|$ viewed as a function of n, and if $|\mathcal{S}_n(R)| = |\mathcal{S}_n(R')|$ to find an explicit bijection between $\mathcal{S}_n(R)$ and $\mathcal{S}_n(R')$. It is also interesting to find relations between $\mathcal{S}_n(R)$ and other combinatorial

structures. By determining $|\mathcal{S}_n(R)|$ we mean finding an explicit formula, or ordinary or exponential generating functions (*g.f.* and *e.g.f.* respectively).

In cases when one does not succeed in finding $|\mathcal{S}_n(R)|$, there appear other questions. For example, does there exist a constant c such that $|\mathcal{S}_n(R)| < c^n$? (see [Bona3]). One more example is the following question: is $|\mathcal{S}_n(R)|$ *P-recursive*? We recall that a function $f : \mathbb{N} \to \mathbb{C}$ is called *P-recursive* if there exist polynomials $P_0, P_1, \ldots, P_k \in \mathbb{C}[n]$, so that

$$P_k(n)f(n+k) + P_{k-1}(n)f(n+k-1) + \cdots + P_0(n)f(n) = 0$$

for all $n \in \mathbb{N}$ (see [Bona4, NooZeil]). However, in the present thesis we only deal with the fundamental questions.

The most studied case has been to forbid a single pattern of length 3. Because of obvious symmetry arguments, namely the trivial bijections, there are only two essentially distinct cases to enumerate, $|\mathcal{S}_n(123)|$ and $|\mathcal{S}_n(132)|$. As it happens, these two functions are equal to the nth Catalan number, $C_n = \frac{1}{n+1}\binom{2n}{n}$, which was shown by Knuth [Knuth]. The first bijection between the two cases was presented by Simion and Schmidt [SimSch], a second one was given by Richards [Rich]; West described in [West1] a construction using trees; and recently, Krattenthaler [Krat] connected the 123-avoiding and 132-avoiding permutations via Dyck paths.

While there are 24 permutation patterns of length 4, for many of them the sequences $|\mathcal{S}_n(\tau)|$ are identical. In fact, there are only three different classes of patterns from this point of view [West, Stank]. The patterns 1342, 1234 and 1324 are distinct representatives of these classes. Table 1 shows the present state of research on permutations avoiding given patterns of length 4, where

$$(\star) = 2\sum_{k=0}^{n} \binom{2k}{k}\binom{n}{k}^2 \frac{3k^2+2k+1-n-2kn}{(k+1)^2(k+2)(n-k+1)} \quad \text{and}$$

$$(\star\star) = \frac{7n^2-3n-2}{2}\cdot(-1)^{n-1} + 3\sum_{i=2}^{n} 2^{i+1}\cdot\frac{(2i-4)!}{i!(i-2)!}\binom{n-i+2}{2}(-1)^{n-i}.$$

The second column there corresponds to the question of existence of a constant c such that $|\mathcal{S}_n(\tau)| < c^n$. Stanley and Wilf conjectured that such a constant exists for any pattern τ.

For the patterns of length greater than 4, the following result by Regev [Regev] is worth mention.

Theorem 1. *For all n, the number $N_n(12\ldots k)$ of permutations in $\mathcal{S}_n$ that avoid the pattern $12\ldots k$ is asymptotically equal to*

$$\lambda_k \frac{(k-1)^{2n}}{n^{(k^2-2k)/2}}.$$

Here

$$\lambda_k = {\gamma_k}^2 \int_{x_1\geq x_2}\int_{x_2\geq x_3}\cdots\int_{x_{k-1}\geq x_k} [D(x_1,x_2,\ldots,x_k)\cdot e^{-(k/2)x^2}]^2\, dx_1 dx_2 \ldots dx_k,$$

pattern p	$\lvert\mathcal{S}_n(p)\rvert < c^n$	**formula for** $N_n(p)$	P**-recursive**
1234	yes Regev [Regev]	$(\star)$ Gessel [Gessel]	yes Zeilberger [Zeil]
1342	yes Bóna [Bona]	$(\star\star)$ Bóna [Bona1]	yes Bóna [Bona1]
1324	yes Bóna [Bona]	open	open

Table 1: Present state of research on avoidance of patterns of length 4

where $D(x_1, x_2, \ldots, x_k) = \prod_{i<j}(x_i - x_j)$, *and* $\gamma_k = (1/\sqrt{2\pi})^{k-1} \cdot k^{k^2/2}$.

Another general result, involving generating functions, is due to Gessel [Gessel].

Theorem 2. *Let* $\ell_k(n) = |\mathcal{S}_n(12\ldots k)|$*; then*

$$L_k(x) = \sum_{n\geq 0} \ell_k(n)\frac{x^{2n}}{n!} = det(I_{|i-j|}(x))_{1\leq i,j\leq k},$$

where $I_i(x)$ *is a Bessel function:*

$$I_i(x) = \sum_{n\geq 0} \frac{x^{2n+i}}{n!(n+i)!} = \sum_{n\geq 0} \binom{2n+i}{n} \frac{x^{2n+i}}{(2n+i)!}.$$

This result was later explained in terms of lattice walks by Gessel, Weinstein and Wilf [GWW].

A natural question is the consideration of those permutations that avoid two or more patterns simultaneously. This problem was solved completely for the patterns from $\mathcal{S}_3$ (see [SimSch]). We summarize some of the results from that paper in Table 2. The trivial bijections break the set of all possibilities into 12 classes of equivalence; we pick one representative from each class.

For the case of simultaneous avoidance of two patterns τ_1 and τ_2, where $\tau_1 \in \mathcal{S}_3$ and $\tau_2 \in \mathcal{S}_4$ see [West2]. We summarize the known results in Table 3.

The results in Table 4 were given by West.

For the case of simultaneous avoidance of two patterns in $\mathcal{S}_4$, see [Bona2, Kremer] and references therein. Several recent papers [ChowWest, MV1, Krat,

patterns	enumeration
$\{123, 132\}$	2^{n-1}
$\{123, 231\}$	$\binom{n}{2} + 1$
$\{123, 321\}$	zero for $n > 4$
$\{132, 213\}$	2^{n-1}
$\{132, 231\}$	2^{n-1}
$\{132, 312\}$	2^{n-1}
$\{123, 132, 213\}$	Fibonacci numbers
$\{123, 132, 231\}$	n
$\{123, 132, 312\}$	n
$\{123, 132, 321\}$	zero for $n > 4$
$\{123, 231, 312\}$	n
$\{132, 213, 231\}$	n

Table 2: Simultaneous avoidance of patterns of length 3 ([SimSch])

restrictions	formula	author
$S_n(123, 4321)$	0	West
$S_n(123, 3421)$	$\binom{n}{4} + 2\binom{n}{3} + n$	West
$S_n(132, 4321)$	$2\binom{n}{4} + \binom{n}{3} + \binom{n}{2} + 1$	West
$S_n(123, 4231)$	$\binom{n}{5} + 2\binom{n}{4} + \binom{n}{3} + \binom{n}{2} + 1$	West
$S_n(123, 3241)$	$3 \cdot 2^{n-1} - \binom{n+1}{2} - 1$	West
$S_n(123, 3412)$	$2^{n+1} - \binom{n+1}{3} - 2n - 1$	Stanley
$S_n(132, 4231)$	$1 + (n-1)2^{n-2}$	Guibert
$S_n(132, 3421)$	$1 + (n-1)2^{n-2}$	West
$S_n(132, 3214)$	g.f.: $\frac{(1-x)^3}{1-4x+5x^2-3x^3}$	West

Table 3: Simultaneous avoidance of a 3-pattern and a 4-pattern

restrictions	restrictions	formula
$S_n(123, 2143)$	$S_n(312, 1342)$	
$S_n(123, 2413)$	$S_n(312, 3241)$	
$S_n(132, 2314)$	$S_n(312, 3214)$	
$S_n(132, 2341)$	$S_n(123, 3214)$	F_{2n}
$S_n(312, 2314)$	$S_n(312, 4321)$	(Fibonacci number)
$S_n(132, 3412)$	$S_n(312, 3421)$	
$S_n(312, 1432)$	$S_n(132, 3241)$	
$S_n(3142, 2413)$	$S_n(4132, 4231)$	the $(n-1)$-st Schröder number g.f.: $\frac{1-x-\sqrt{1-6x+x^2}}{2x}$

Table 4: Some results given by West

MV3, MV2] deal with the case $\tau_1 \in S_3$, $\tau_2 \in S_k$ for various pairs τ_1, τ_2. Erdős and Szekeres [ErdSze] gave the following general result.

Theorem 3. *For all* $n \geq (\ell - 1)(m - 1) + 1$,

$$|\mathcal{S}_n(12 \ldots \ell, m \ldots 21)| = 0.$$

0.2 Generalized permutation patterns

In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns* that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. In order to avoid confusion we write a "classical" pattern, say 231, as 2-3-1, and if we write, say 2-31, then we mean that if this pattern occurs in a permutation π, then the letters in π that correspond to 3 and 1 are adjacent. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, whereas the pattern 2-3-1 occurs, in addition, in the subwords 562 and 563. If we use "[" in a pattern, for example if we write $p = [1\text{-}2)$, we indicate that in an occurrence of p, the letter corresponding to the 1 must be the first letter of the permutation, whereas if we write, say, $p = (1\text{-}2]$, then the letter corresponding to 2 must be the last (rightmost) letter of the permutation. Thus, a parenthesis at either end of a pattern corresponds to a dash, and a square bracket corresponds to the absence of a dash. However, when a pattern begins *and* ends with a parenthesis, we omit these parentheses, writing simply 123 instead of (123).

The motivation for introducing these patterns in [BabStein] was the study of Mahonian statistics. A number of interesting results on generalized patterns were obtained in [Claes]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were

patterns P	$\lvert\mathcal{S}_n(P)\rvert$	**description**
1-23	B_n	partitions of $[n]$
1-32	B_n	partitions of $[n]$
2-13	C_n	Dyck paths of length $2n$
1-23, 12-3	$B_n^\star$	non-overlapping partitions of $[n]$
1-23, 1-32	I_n	involutions in $\mathcal{S}_n$
1-23, 13-2	M_n	Motzkin paths of length n

Table 5: Generalized pattern avoidance ([Claes])

shown there. The main results from that paper are given in Table 5, where B_n is the n-th Bell number, C_n is the n-th Catalan number, and $B_n^\star$ is the n-th Bessel number.

For some other results on generalized permutation patterns see [ClaesMans1, ClaesMans2, Kit1, Kit2, Kit3, KitMans1, KitMans2]

Paper I. In Paper I ([Kit1]) we consider 3-patterns without internal dashes, that is, generalized patterns of the form xyz. Thus, such patterns correspond to contiguous subwords anywhere in a permutation. For example the permutation $\pi = 12345$ has 3 occurrences of the pattern 123 but 10 occurrences of the classical pattern 1-2-3. Patterns without internal dashes were considered by Elizalde and Noy in [ElizNoy]. In that paper, there is a number of results on the distribution of several classes of patterns without internal dashes. In particular, formulas are given for the bivariate exponential generating functions that count permutations by the number of occurrences of any given 3-pattern.

As in the paper by Simion and Schmidt [SimSch], dealing with the classical patterns, Claesson [Claes] considered a number of cases when permutations have to avoid two or more generalized patterns simultaneously (see Table 5). However, except for the simultaneous avoidance of the patterns 123 and 132, and three more pairs each of which is essentially equivalent to one of these, there were no other results for multi-avoidance of the patterns without internal dashes. In Paper I we give either an explicit formula or a recursive formula for almost all cases of simultaneous avoidance of more than two patterns. We also mention what is known about double restrictions. There are 18 classes of equivalence. As we did before, we choose a representative from each class and record all the known results in Table 6, where we define the *double factorial* $n!!$ by $0!! = 1$, and, for $n > 0$,

$$n!! = \begin{cases} n \cdot (n-2) \cdots 3 \cdot 1, & \text{if } n \text{ is odd,} \\ n \cdot (n-2) \cdots 4 \cdot 2, & \text{if } n \text{ is even.} \end{cases}$$

Besides, in order to complete the description of simultaneous avoidance of two generalized patterns without internal dashes, we put in the same table some

results from papers III ([KitMans1]) and IV ([KitMans2]).

Paper II. In Paper II ([Kit3]) we consider avoidance of some generalized 3-patterns with additional restrictions. The restrictions consist of demanding that a permutation begin or end with the pattern $12\dots k$ or the pattern $k(k-1)\dots 1$. We observe that avoidance of some pattern with the additional restrictions described above in fact is equivalent to simultaneous avoidance of several patterns. For example, beginning with the pattern 12 is equivalent to the avoidance of the pattern [21) in the Babson-Steingrímsson notation. Thus avoidance of the pattern 132 and beginning with the pattern 12 is equivalent to simultaneous avoidance of the patterns 132 and [21). Also, ending with the pattern 123 is equivalent to simultaneously avoiding the patterns (132], (213], (231], (312] and (321]. So, demanding that a permutation must begin or end with some pattern is equivalent to simultaneous avoidance of a set of generalized patterns. A motivation for considering additional restrictions such as beginning or ending with some patterns is their connection to some classes of trees mentioned below.

It turns out that the number of permutations that avoid the pattern 1-32 and end with the pattern $12\dots k$ is a linear combination of the Bell numbers. We find a bijection between these permutations and all partitions of an $(n-1)$-element set with one subset marked that satisfy certain additional conditions. In particular, we get that the total number of partitions of an $(n-1)$-element set with one part marked, is equal to the number of (1-32)-avoiding n-permutations that end with a 12-pattern. Also, we get an identity involving the Bell numbers and the Stirling numbers of the second kind, which seems to be new. Besides, we prove that the number of 132-avoiding n-permutations that begin with the pattern 12 is equal to the number of *increasing rooted trimmed trees* with $n+1$ nodes. In an increasing rooted tree, the nodes are numbered and the numbers increase as we move away from the root. A trimmed tree is a tree where no node has a single leaf as a child (every leaf has a sibling).

In Sections 4–7 of Paper II, we give a complete description, in terms of exponential generating functions, for the number of permutations that avoid a pattern of the form xyz and begin or end with the pattern $12\dots k$ or the pattern $k(k-1)\dots 1$. We record all the results concerning these e.g.f. in Table 7. The case $k=1$ is equivalent to the absence of the additional restriction. This case was considered in [ElizNoy] and Paper I.

Paper III. As mentioned above, Paper II dealt with the avoidance of a generalized 3-pattern p with no dashes and, at the same time, beginning or ending with an increasing or decreasing pattern. Theorem 2 in Paper III ([KitMans1]) generalizes some of these results to the case of beginning (resp. ending) with an arbitrary pattern p that has the greatest or least letter as the rightmost (resp. leftmost) letter. To write down this theorem, we need the following definitions. Let $E_q^p(x)$ denote the exponential generating function for the number of permutations that avoid the pattern q and begin with the pattern p. Also, Γ_k^{min} (resp. Γ_k^{max}) denotes the set of all k-patterns with no dashes such that the least

class	restrictions	formula
1	123, 321, 231, 213	2
2	321, 213, 231, 312	2
3	132, 231, 213, 312	2
4	123, 321, 132, 231	2 if $n = 3$; 0 if $n > 3$
5	231, 312, 213, 123	$n - 1$
6	123, 321, 132, 213	$2C_k$, if $n = 2k + 1$ $C_k + C_{k-1}$, if $n = 2k$
7	231, 312, 321	$\binom{n}{\lfloor n/2 \rfloor}$
8	123, 213, 312	n
9	132, 213, 312	$1 + 2^{n-2}$
10	123, 213, 231	recursive formula: $A(0) = 1;\ A(1) = 1;$ $A(n) = \sum_i \binom{n-i-1}{i} A(n-2i-1) + ((n+1) \bmod 2)$ the first few numbers: 1, 1, 2, 3, 6, 13, 29, 72, 185 . . .
11	123, 321, 231	$(n-1)!! + (n-2)!!$
12	123, 231, 312	e.g.f.: $1 + x(\sec x + \tan x)$, Paper III
13	321, 132	recurrence relation, Paper IV
14	213, 231	recurrence relation, Paper IV
15	132, 213	recurrence relation, Paper IV
16	123, 321	$2E_n$, where E_n is the n-th Euler number
17	321, 231	the number of involutions in $\mathcal{S}_n$, Claesson [Claes]
18	132, 231	2^{n-1}

Table 6: Simultaneous avoidance of generalized 3-patterns (mostly Paper I)

	avoid	begin	end	e.g.f.
	123	$12\dots k$	–	
1	123	–	$12\dots k$	$\frac{\sqrt{3}}{2}\frac{e^{x/2}}{\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})}$, if $k=1$
	321	$k\dots 21$	–	$\frac{\sqrt{3}}{2}e^{x/2}\sec(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})-\frac{1}{2}-\frac{\sqrt{3}}{2}\tan(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})$, if $k=2$
	321	–	$k\dots 21$	0, if $k\geq 3$
	123	$k\dots 21$	–	
2	123	–	$k\dots 21$	$\frac{\sqrt{3}}{2}\frac{e^{x/2}}{\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})}$, if $k=1$
	321	$12\dots k$	–	
	321	–	$12\dots k$	$\frac{e^{x/2}\int_0^x e^{-t/2}t^{k-1}\sin(\frac{\sqrt{3}}{2}t+\frac{\pi}{6}))\ dt}{(k-1)!\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})}$, if $k\geq 2$
	132	$12\dots k$	–	
3	213	–	$12\dots k$	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
	312	$k\dots 21$	–	$e^{-x^2/2}(1-\int_0^x e^{-t^2/2}\ dt)^{-1}-x-1$, if $k=2$
	231	–	$k\dots 21$	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_2}(e^{-t_1^2/2}-$
				$(t_1+1)(1-\int_0^{t_1}e^{-t^2/2}\ dt))dt_1dt_2\cdots dt_{k-2}$, if $k\geq 3$
	132	$k\dots 21$	–	
4	213	–	$k\dots 21$	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
	312	$12\dots k$	–	$\frac{1}{(k-1)!(1-\int_0^x e^{-t^2/2}\ dt)}\int_0^x t^{k-1}e^{-t^2/2}\ dt$, if $k\geq 2$
	231	–	$12\dots k$	
	213	$12\dots k$	–	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
5	132	–	$12\dots k$	
	231	$k\dots 21$	–	$\int_0^x\int_0^t\frac{s^{k-2}e^{T(t)-T(s)}}{(k-2)!(1-\int_0^t e^{-m^2/2}dm)}\ dsdt$, if $k\geq 2$, where
	312	–	$k\dots 21$	$T(x)=-x^2/2+\int_0^x\frac{e^{-t^2/2}}{1-\int_0^t e^{-s^2/2}ds}\ dt$
	213	$k\dots 21$	–	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
6	132	–	$k\dots 21$	$-\frac{x^{k-1}}{(k-1)!}+\sum_{n=0}^{k-2}\int_0^x\int_0^{t_n}\cdots\int_0^{t_1}\frac{C_{k-n}(t)+\delta_{n,k-2}}{1-\int_0^t e^{-m^2/2}dm}dtdt_1\cdots dt_n$,
	231	$12\dots k$	–	if $k\geq 2$, where $C_k(x)=e^{T(x)}\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_1}e^{-T(t)}$.
	312	–	$12\dots k$	$\left(\frac{e^{-t^2/2}}{1-\int_0^t e^{-m^2/2}dm}-t-1\right)dtdt_1\cdots dt_{k-2}$ and $T(x)$ as above

Table 7: Avoiding a pattern xyz with additional restrictions (Paper II)

(resp. greatest) letter of a pattern is the rightmost letter. Now, we formulate Theorem 2 from Paper III:

Theorem 4. *Suppose* $p_1, p_2 \in \Gamma_k^{min}$ *and* $p_1 \in S_k(132)$, $p_2 \in S_k(123)$. *Thus, the complements* $C(p_1), C(p_2) \in \Gamma_k^{max}$ *and* $C(p_1) \in S_k(312)$, $C(p_2) \in S_k(321)$. *Then, for* $k \geq 2$,

$$E_{132}^{p_1}(x) = E_{312}^{C(p_1)}(x) = \frac{\int_0^x t^{k-1} e^{-t^2/2}\, dt}{(k-1)!(1 - \int_0^x e^{-t^2/2}\, dt)}$$

and

$$E_{123}^{p_2}(x) = E_{321}^{C(p_2)}(x) = \frac{e^{x/2} \int_0^x e^{-t/2} t^{k-1} \sin(\frac{\sqrt{3}}{2}t + \frac{\pi}{6}))\, dt}{(k-1)! \cos(\frac{\sqrt{3}}{2}x + \frac{\pi}{6})}.$$

Propositions 4–15 (resp. 16–27) in Paper III give a complete description for the number of permutations avoiding a pattern of the form x-yz or xy-z and beginning with one of the patterns $12\ldots k$ or $k(k-1)\ldots 1$ (resp. $23\ldots k1$ or $(k-1)(k-2)\ldots 1k$). For each of these cases we find either the ordinary or exponential generating function or a precise formula for the number of such permutations. Theorem 28 in Paper III generalizes some of these results:

Theorem 5. *Suppose* $p_1, p_2 \in \Gamma_k^{min}$ *and* $p_1 \in S_k(1\text{-}23)$, $p_2 \in S_k(1\text{-}32)$. *Thus, the complements* $C(p_1), C(p_2) \in \Gamma_k^{max}$ *and* $C(p_1) \in S_k(1\text{-}23)$, $C(p_2) \in S_k(3\text{-}12)$. *Then, we have*

$$E_{1\text{-}23}^{p_1}(x) = E_{3\text{-}21}^{C(p_1)}(x) = E_{1\text{-}32}^{p_2}(x) = E_{3\text{-}12}^{C(p_2)}(x) =$$

$$\begin{cases} (e^{e^x}/(k-1)!) \int_0^x t^{k-1} e^{-e^t + t}\, dt, & \text{if } k \geq 2, \\ e^{e^x - 1}, & \text{if } k = 1. \end{cases}$$

Moreover, the results from Propositions 4–27 in Paper III give a complete description for the number of permutations that avoid a pattern of the form x-yz or xy-z and end with one of the patterns $12\ldots k$, $k(k-1)\ldots 1$, $1k(k-1)\ldots 2$ and $k12\ldots(k-1)$. To get the last one of these we only need to apply the reverse operation defined above.

Paper IV. In Paper IV ([KitMans2]) we continue consideration of generalized pattern avoidance with additional restriction. In Section 4 of Paper IV, we consider avoidance of a pattern x-y-z, and beginning or ending with an increasing or decreasing pattern. This completes the results given in Paper III, which concerns the number of permutations that avoid a generalized 3-pattern and begin or end with an increasing or decreasing pattern.

In Sections 5–8 of Paper IV, we consider stronger restrictions, which generalize many results from Papers II, III, IV. Namely, we give enumeration for the number of permutations that avoid a generalized 3-pattern, and begin *and* end with increasing or decreasing patterns. We record our results in terms of either generating functions, or exponential generating functions, or formulas for the numbers in question.

In Section 9 of Paper IV, we discuss possible directions for generalization of the results from Sections 5–8. The first direction is to consider avoidance of more than one pattern, beginning with some pattern and ending with another pattern. The second direction concerns permutations in $\mathcal{S}_n$ containing a pattern τ exactly r times, beginning with some pattern and ending with another pattern.

0.3 Partially ordered generalized patterns

Suppose we are interested in finding the number of permutations that avoid all patterns from the set $\{12\text{-}4\text{-}3, 13\text{-}4\text{-}2, 23\text{-}4\text{-}1\}$ simultaneously. There is a way to code these three patterns into one pattern, and instead of considering three patterns to consider one. This is done by allowing some letters of a pattern to be incomparable. Thus the set of patterns above can be replaced by the pattern $p = 1'2'\text{-}3\text{-}1''$, where in an occurrence of p in a permutation π the letter corresponding to the $1''$ in p can be either larger or smaller than the letters corresponding to $1'2'$, but all of them must be less than the letter corresponding to the 3 in p. Such patterns are discussed in Papers V ([Kit2]) and VI ([KitMans3]). These patterns allow us to determine the distribution of *non-overlapping* occurrences of patterns without internal dashes.

Paper V. In Paper V ([Kit2]) we introduce a further generalization of generalized patterns (GPs)—namely *partially ordered generalized patterns (POGP).* A POGP is a GP some of whose letters are incomparable. For instance, if we write $p = 1\text{-}1'2'$ then we mean that in an occurrence of p in a permutation π the letter corresponding to the 1 in p can be either larger or smaller than the letters corresponding to $1'2'$. Thus, the permutation 31254 has three occurrences of p, namely 3-12, 3-25, and 1-25.

We consider two particular classes of POGPs—*shuffle patterns* and *multi-patterns.* A multi-pattern is of the form $p = \sigma_1\text{-}\sigma_2\text{-}\cdots\text{-}\sigma_k$ and a shuffle pattern is of the form $p = \sigma_0\text{-}a_1\text{-}\sigma_1\text{-}a_2\text{-}\cdots\text{-}a_k\text{-}\sigma_k$, where for any i and j, the letter a_i is greater than any letter of σ_j and for any $i \neq j$ each letter of σ_i is incomparable to any letter of σ_j These patterns are investigated in Sections 4 and 5. A corollary to Theorem 13 is the result of Claesson [Claes, Proposition 2] that the number of n-permutations that avoid the pattern 12-3 is the n-th Bell number.

Let p and q be two patterns. An occurrence of p *overlaps* an occurrence of q in a permutation π if these two occurrences share a letter in π. For example, if $p = 123$, $q = 231$ and $\pi = 623514$ then 235 and 351, being occurrences of the patterns p and q respectively, overlap.

Let $p = \sigma_1\text{-}\sigma_2\text{-}\cdots\text{-}\sigma_k$ be an arbitrary multi-pattern and let $A_i(x)$ be the exponential generating function (e.g.f.) for the number of permutations that avoid σ_i for each i. In Theorem 28 we find the e.g.f., in terms of the $A_i(x)$, for the number of permutations that *avoid* p.

Theorem 6. *Let $p = \sigma_1\text{-}\sigma_2\text{-}\cdots\text{-}\sigma_k$ be a multi-pattern and let $A_i(x)$ be the e.g.f. for the number of permutations that avoid σ_i. Then the e.g.f. $B(x)$ for*

the number of permutations that avoid p is

$$B(x) = \sum_{i=1}^{k} A_i(x) \prod_{j=1}^{i-1}((x-1)A_j(x)+1).$$

In fact, this allows us to find the e.g.f. for the *entire distribution* of the maximum number of non-overlapping occurrences of a pattern p with no dashes, if we only know the e.g.f. for the number of permutations that avoid p:

Theorem 7. *Let p be a GP with no dashes. Let $A(x)$ be the e.g.f. for the number of permutations that avoid p. Let $D(x,y) = \sum_{\pi} y^{N(\pi)} \frac{x^{|\pi|}}{|\pi|!}$ where $N(\pi)$ is the maximum number of non-overlapping occurrences of p in π. Then*

$$D(x,y) = \frac{A(x)}{1 - y((x-1)A(x)+1)}.$$

In many cases, this theorem gives nice generating functions. The following two examples are corollaries to Theorem 7. We recall that a descent in a permutation $\pi = a_1 a_2 \dots a_n$ is an i such that $a_i > a_{i+1}$. Two descents i and j *overlap* if $j = i+1$.

Example 1. If we consider descents then $A(x) = e^x$, hence the distribution of the maximum number of non-overlapping descents is given by the formula

$$D(x,y) = \frac{e^x}{1 - y(1 + (x-1)e^x)}.$$

The reader might want to compare this result with some known results related to descents. To this end we recall the following. The number of descents in a permutation π is denoted $\operatorname{des}\pi$ (and is equivalent to the generalized pattern 21). Any statistic with the same distribution as des is said to be *Eulerian*. The *Eulerian numbers* $A(n,k)$ count permutations in the symmetric group $\mathcal{S}_n$ with k descents and they are the coefficients of the *Eulerian polynomials* $A_n(t)$ defined by $A_n(t) = \sum_{\pi \in \mathcal{S}_n} t^{1+\operatorname{des}\pi}$. The e.g.f. for Eulerian polynomials is given by

$$\sum_{n \geq 0} A_n(t) \frac{x^n}{n!} = \frac{t(1-t)e^x}{e^{xt} - te^x}.$$

Example 2. If we consider the maximum number of non-overlapping occurrences of the pattern 132 then the distribution of these numbers is given by the formula

$$D(x,y) = \frac{1}{1 - yx + (y-1)\displaystyle\int_0^x e^{-t^2/2}\, dt}.$$

We will talk about *bivariate generating functions*, or *b.g.f.*, exclusively as generating functions of the form

$$A(u,z) = \sum_{\pi} u^{p(\pi)} \frac{z^{|\pi|}}{|\pi|!} = \sum_{n,k\geq 0} A_{n,k} u^k \frac{x^n}{n!},$$

were $A_{n,k}$ is the number of n-permutations with k occurrences of the pattern p.

In order to apply the last two theorems, as well as some other results from Paper V, we need to know how many patterns avoid a given ordinary GP with no dashes. We are also interested in different approaches to studying these patterns. There is a number of results on the distribution of several classes of patterns with no dashes. These results can be used as building blocks for some of the results in Paper V. The most important of these is the following result by Elizalde and Noy:

Theorem 8. ([ElizNoy, Theorem 3.4]) *Let m and a be positive integers with $a \leq m$, let $\sigma = 12\ldots a\tau(a+1) \in \mathcal{S}_{m+2}$, where τ is any permutation of the letters $\{a+2, a+3, \ldots, m+2\}$, and let $A(u,z)$ be the b.g.f. for permutations where u marks the number of occurrences of σ and z marks the length of the permutation. Then $A(u,z) = 1/w(u,z)$, where w is the solution of*

$$w^{a+1} + (1-u)\frac{z^{m-a+1}}{(m-a+1)!}w' = 0$$

with $w(0) = 1$, $w'(0) = -1$ and $w^{(k)}(0) = 0$ for $2 \leq k \leq a$. In particular, the distribution does not depend on τ.

In Paper V we give alternative proofs, using inclusion-exclusion, of some of the results of Elizalde and Noy [ElizNoy]. Our proofs result in explicit formulas for the coefficients of the e.g.f. whereas Elizalde and Noy obtained differential equations for the same e.g.f..

Paper VI. From now on we are not discussing permutations and generalized permutation patterns. Instead we consider k-ary words and occurrences of patterns in them. First of all we need some definitions, most of which are intuitively clear from the preceding discussion.

Let $[k]^n$ denote the set of all the words of length n over the (totally ordered) alphabet $[k] = \{1, 2, \ldots, k\}$. We refer to these words as *n-long k-ary words*. A *generalized pattern* τ is a word in $[\ell]^m$ (possibly with dashes between some letters) that contains each letter from $[\ell]$ (possibly with repetitions). We say that the word $\sigma \in [k]^n$ *contains* a generalized pattern τ if σ contains a subsequence order-isomorphic to τ in which the entries corresponding to consecutive entries of τ that are not separated by a dash must be adjacent. Otherwise, we say that σ *avoids* τ and write $\sigma \in [k]^n(\tau)$. Thus, $[k]^n(\tau)$ denotes the set of all the words in $[k]^n$ that avoid τ. Moreover, if P is a set of generalized patterns then $|[k]^n(P)|$ denotes the set all the words in $[k]^n$ that avoid all patterns from P simultaneously. For example, a word $\pi = a_1 a_2 \ldots a_n$ avoids the pattern 13-2 if

π has no subsequence $a_i a_{i+1} a_j$ with $j > i+1$ and $a_i < a_j < a_{i+1}$. Also, π avoids the pattern 121 if it has no subword $a_i a_{i+1} a_{i+2}$ such that $a_i = a_{i+2} < a_{i+1}$.

Burstein [Burstein] considered patterns without repeated letters on words instead of permutations. In particular, he found the number $|[k]^n(P)|$ of words of length n in a k-letter alphabet that avoid all patterns from a set $P \subseteq \mathcal{S}_3$ simultaneously. Burstein and Mansour [BurMans1] (resp. [BurMans2, BurMans3]) considered forbidden patterns (resp. generalized patterns) with repeated letters.

In Paper VI ([KitMans3]) we introduce a further generalization of the generalized patterns, namely *partially ordered generalized patterns in words* (*POGPs*), which are analogues of POGPs in permutations [Kit2]. A POGP is a generalized pattern some of whose letters are incomparable. For example, if we write $\tau = 1\text{-}1'2'$, then we mean that in an occurrence of τ in a word $\sigma \in [k]^n$ the letter corresponding to the 1 in τ can be either larger than, smaller than, or equal to the letters corresponding to $1'2'$. Thus, the word $113425 \in [5]^6$ contains seven occurrences of τ, namely 113, 134 twice, 125 twice, 325, and 425.

Following Paper V, we consider two particular classes of POGPs— *shuffle patterns* and *multi-patterns*, which allows us to give an analogue for all the main results of [Kit2] for k-ary words.

Let $\tau = \tau_0\text{-}\tau_1\text{-}\cdots\text{-}\tau_s$ be an arbitrary multi-pattern and let $A_{\tau_i}(x;k)$ be the ordinary generating function (g.f.) for the number of words in a k-letter alphabet that avoid τ_i for each i. In Theorem 4.7 of Paper VI we find the g.f., in terms of the $A_{\tau_i}(x;k)$, for the number of k-ary words that avoid τ:

Theorem 9. *Let $\tau = \tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_s$ be a multi-pattern. Then*

$$A_\tau(x;k) = \sum_{j=1}^{s} A_{\tau_j}(x;k) \prod_{i=1}^{j-1} ((kx-1)A_{\tau_i}(x;k)+1).$$

In particular, this allows us to find the g.f. for the entire distribution of the maximum number of non-overlapping occurrences of a pattern τ with no dashes, if we only know the g.f. for the number of k-ary words that avoid τ:

Theorem 10. *Let τ be a generalized pattern with no dashes. Then, for all $k \geq 1$,*

$$\sum_{n\geq 0} \sum_{\sigma\in[k]^n} y^{N_\tau(\sigma)} x^n = \frac{A_\tau(x;k)}{1 - y((kx-1)A_\tau(x;k)+1)},$$

where $N_\tau(\sigma)$ is the maximum number of non-overlapping occurrences of τ in σ.

Thus, in order to apply our results from the last two theorems we need to know how many k-ary words avoid a given ordinary generalized pattern with no dashes. This question was examined, for instance, in [BurMans1, Sections 2 and 3], [BurMans2, Section 3] and [BurMans3, Section 3.3].

All of the following examples are corollaries to Theorem 10.

Example 3. If we consider rises (the pattern 12) then $A_{12}(x;k) = \frac{1}{(1-x)^k}$ (see [BurMans2]), hence the distribution of the maximum number of non-overlapping

descents is given by the formula:

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_{12}(\sigma)}x^n = \frac{1}{(1-x)^k + y(1-kx-(1-x)^k)}.$$

Example 4. The distribution of the maximum number of non-overlapping occurrences of the pattern 122 is given by the formula:

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_{122}(\sigma)}x^n = \frac{x}{(1-x^2)^k + x - 1 + y(1-kx^2-(1-x^2)^k)},$$

since, according to [BurMans3, Theorem 3.10], $A_{122}(x;k) = \frac{x}{(1-x^2)^k-(1-x)}$.

Example 5. If we consider the pattern 212 then $A_{212}(x;k) = \left(1 - x\sum_{j=0}^{k-1}\frac{1}{1+jx^2}\right)^{-1}$ (see [BurMans3, Theorem 3.12]), hence the distribution of the maximum number of non-overlapping occurrences of the pattern 212 is given by the formula:

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_{212}(\sigma)}x^n = \frac{1}{1 - x\sum_{j=0}^{k-1}\frac{1}{1+jx^2} + xy\left(\sum_{j=0}^{k-1}\frac{1}{1+jx^2} - k\right)}.$$

0.4 Counting occurrences of certain patterns in certain words

The most attention, in the papers on classical or generalized patterns, in particular in Papers I–VI, is paid to obtaining exact formulas and/or generating functions for the number of words or permutations avoiding, or having k occurrences of, certain patterns. In Papers VII–IX we suggest another problem, namely counting the occurrences of certain patterns in certain words. These words were chosen to be the set of all finite approximations of certain sequences.

In Paper VII ([KitMans4]) this is a sequence generated by a morphism (a system of substitutions, to be defined below) with certain restrictions. In Paper VIII ([KitMans5]) the sequence is obtained from the *Peano curve*. The Peano curve was studied by the Italian mathematician Giuseppe Peano in 1890 as an example of a continuous space filling curve. Finally, in Paper IX ([Kit4]) this sequence is the *sigma-sequence*, which was used by Evdokimov [Evdok1] to construct chains of maximal length in the n-dimensional unit cube.

Independent interest in the sigma-sequence appears in connection with the well-known *Dragon curve*, discovered by the physicist John E. Heighway and defined as follows: Fold a sheet of paper in half, then fold in half again (so that the folds are parallel), and again, etc. and then unfold in such a way that each crease created by the folding process is opened out into a 90-degree angle. The "curve" refers to the shape of the partially unfolded paper as seen edge on (see

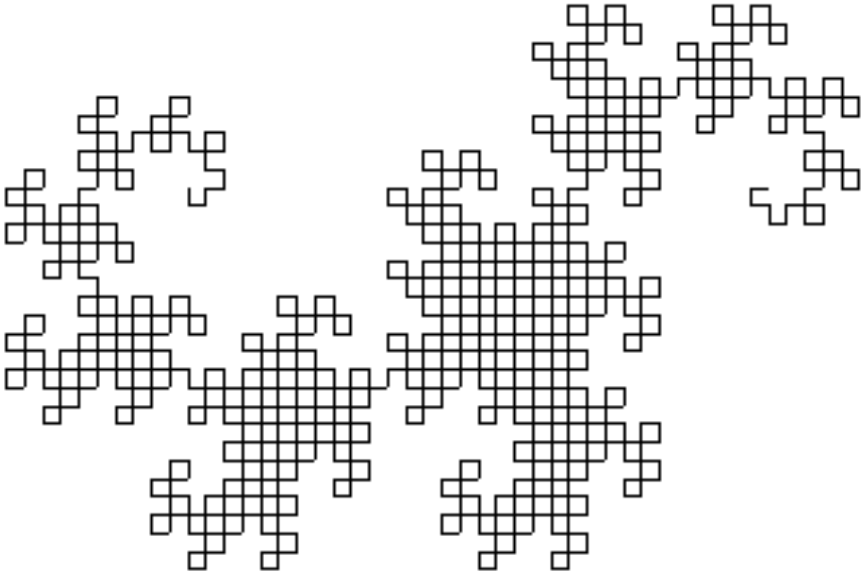


Figure 1: Dragon curve

Figure 1). If one travels along the curve, some of the creases will represent turns to the left and others turns to the right. Now if 1 indicates a turn to the right, and 2 to the left, and we start travelling along the curve indicating the turns, we get the sigma-sequence [Evdok].

Paper VII. In Paper VII ([KitMans4]) we count the occurrences of certain patterns in certain words. We choose these words to be a set of all finite approximations (to be defined below) of a sequence generated by a morphism with certain restrictions. The motivation is to study classes of sequences and words that are defined by iterative schemes [Lothaire, Salomaa]. The pattern τ in our considerations is either a classical pattern (with repeated letters allowed) from the set $\{1\text{-}2, 2\text{-}1, 1\text{-}1\text{-}\cdots\text{-}1\}$, or an arbitrary generalized pattern without internal dashes, in which repetitions of letters are allowed. In particular, we find that there are $(3 \cdot 4^{n-1} + 2^n)$ occurrences of the pattern 1-2 in the n-th finite approximation of the sequence w defined below, which is a classical example of a nonrepetitive sequence.

Let Σ be an alphabet and $\Sigma^\star$ the set of all words over Σ. A map $\varphi : \Sigma^\star \to \Sigma^\star$ is called a *morphism* if we have $\varphi(uv) = \varphi(u)\varphi(v)$ for any $u, v \in \Sigma^\star$. It is easy to see that a morphism φ can be defined by defining $\varphi(i)$ for each $i \in \Sigma$. The set of all rules $i \mapsto \varphi(i)$ is called a *substitution system*. We create words by starting with a letter from the alphabet Σ and iterating the substitution system. Such a substitution system is called a *D0L* (*Deterministic, with no*

context Lindenmayer) *system* [LindRoz]. D0L systems are classical objects of formal language theory. They are interesting from a mathematical point of view [Frid], but also have applications in theoretical biology [Lind]. Let $|X|$ denote the length of a word X, that is the number of letters in X.

Suppose a word $\varphi(a)$ begins with a for some $a \in \Sigma$, and that the length of $\varphi^k(a)$ increases without bound. The symbolic sequence $\lim\limits_{k\to\infty} \varphi^k(a)$ is said to be *generated* by the morphism φ. In particular, $\lim\limits_{k\to\infty} \varphi^k(a)$ is a *fixed point* of φ. However, in this paper we are only interested in the *finite approximations* of $\lim\limits_{k\to\infty} \varphi^k(a)$, that is in the words $\varphi^k(a)$ for $k = 1, 2, \ldots$.

An example of a sequence generated by a morphism is the following sequence w. We create words by starting with the letter 1 and iterating the substitution system ϕ_w: $1 \mapsto 123$, $2 \mapsto 13$, $3 \mapsto 2$. Thus, the initial letters of w are 123132123213.... This sequence was constructed in connection with the problem of constructing a nonrepetitive sequence on a 3-letter alphabet, that is, a sequence that does not contain any subwords of the type XX, where X is any non-empty word over a 3-letter alphabet. The sequence w has that property. The question of the existence of such a sequence, as well as the questions of the existence of sequences avoiding other kinds of repetitions, were studied in algebra [Adian, Justin, Kol], discrete analysis [Carpi, Dekk, Evdok2, Ker, Pleas] and in dynamical systems [MorseHedl]. In Examples 2.2, 2.6 and 3.3 of Paper VII we give the number of occurrences of the patterns 1-2, 2-1, 1-1-$\cdots$-1, 12, 123 and 21 in the finite approximations of w.

Suppose $N_\phi^\tau(n)$ denotes the number of occurrences of a pattern τ in a word generated by some morphism ϕ after n iterations. Suppose $W = AXBYC$, where A, X, B, Y, and C are some subwords. We say that an occurrence of a pattern τ in W is *external* for the pair of words (X, Y), if this occurrence starts somewhere in X and ends somewhere in Y. Also, an occurrence of τ in W is *internal* for the word X if this occurrence is a subsequence of X. For example, if $W = 12324265$, $A = 1$, $X = 23$, $B = 2$ and $Y = 426$ then an occurrence of the generalized pattern 213, namely 324 is external for (X, Y). On the other hand, the word $X = 231$ has two internal occurrences of the pattern 2-1, namely 21 and 31.

The following theorem was proved in Paper VII.

Theorem 11. *Let $\mathcal{A} = \{1, 2, \ldots, k\}$ be an alphabet, where $k \geq 2$ and a pattern $\tau \in \{1\text{-}2, 2\text{-}1\}$. Let X_1 begins with the letter 1 and consists of ℓ copies of each letter $i \in \mathcal{A}$ ($\ell \geq 1$). Let a morphism ϕ be such that*

$$1 \to X_1,\ 2 \to X_2,\ 3 \to X_3, \ldots, k \to X_k,$$

where we allow X_i to be the empty word ϵ for $i = 2, 3, \ldots, k$ (that is, ϕ may be an erasing morphism), $\sum\limits_{i=2}^{k} |X_i| = k \cdot d$, and each letter from $\mathcal{A}$ appears in the word $X_2 X_3 \ldots X_k$ exactly d times. Besides, let $e_{i,j}$ (resp. e_i) be the number of external occurrences of τ for (X_i, X_j) (resp. (X_i, X_i)), where $i \neq j$. We assume

that $e_{i,j} = e_{j,i}$ for all i and j. Let s_i be the number of internal occurrences of τ in X_i. In particular, $s_i = e_i = e_{i,j} = e_{j,i} = 0$, whenever $X_i = \epsilon$; also, $e_i = |X_i| \cdot (|X_i| - 1)/2$, whenever there are no repetitive letters in X_i. Then $N_\phi^\tau(1) = s_1$ and for $n \geq 2$, $N_\phi^\tau(n)$ is given by

$$\ell \cdot (d+\ell)^{n-2} \sum_{i=1}^{k} s_i + \binom{\ell \cdot (d+\ell)^{n-2}}{2} \sum_{i=1}^{k} e_i + \ell^2 \cdot (d+\ell)^{2n-4} \sum_{1 \leq i < j \leq k} e_{i,j}.$$

Paper VIII. Let us define the Peano curve and the Peano words. We follow [GelbOlm] and present a description of a curve that fills the unit square $S = [0,1] \times [0,1]$, given in 1891 by D. Hilbert.

As indicated in Figure 2, the idea is to subdivide S and the unit interval $I = [0,1]$ into 4^n closed subsquares and subintervals, respectively, and to set up a correspondence between subsquares and subintervals so that inclusion relationships are preserved (at each stage of subdivision, if a square corresponds to an interval, then its subsquares correspond to subintervals of that interval).

We now define the continuous mapping f of I onto S: If $x \in I$, then at each stage of subdivision x belongs to *at least* one closed subinterval. Select either one (if there are two) and associate it to the corresponding square. In this way a decreasing sequence of closed squares is obtained corresponding to a decreasing sequence of closed intervals. This sequence of closed squares has the property that there is exactly one point belonging to all of them. This point is defined to be $f(x)$. It can be shown that the point $f(x)$ is well-defined, that is, independent of any choice of intervals containing x; the range of f is S; and f is continuous.

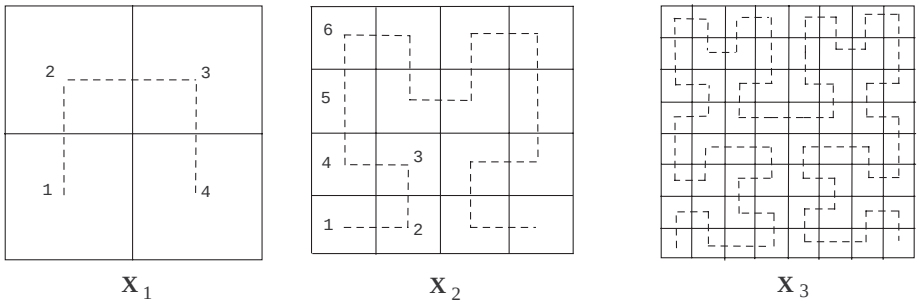


Figure 2: The Peano words

The following discrete analogue of the Peano curve was given by Evdokimov [Evdok]. For subdivision stage (iteration) n we construct a word X_n as follows: Go through the curve inside S starting at the point 1 (see Figure 2), and coding any movement "up" by 1, "right" by 2, "down" by 3, "left" by 4. Thus, we start

x	y	$N_{\tau_1(x,y)}(X_{2k+1})$	$N_{\tau_2(x,y)}(X_{2k+1})$	$N_{\tau_1(x,y)}(X_{2k+2})$	$N_{\tau_2(x,y)}(X_{2k+2})$
1	1	$\binom{4^{2k}-1}{\ell}$	$\binom{4^{2k}-1}{\ell}$	$\binom{4^{2k+1}+2^{2k+1}-1}{\ell}$	$\binom{4^{2k+1}+2^{2k+1}-1}{\ell}$
1	2	S_1	$\binom{4^{2k}}{\ell}+\binom{4^{2k}+2^{2k}-1}{\ell}$	S_2	$\binom{4^{2k+1}}{\ell}$
2	1	0	$\binom{4^{2k}-2^{2k}}{\ell}$	$\binom{4^{2k+1}}{\ell}$	S_2

Table 8: Generalized patterns having 2 letters (Paper VIII)

with the first iteration $X_1 = 123$, the second iteration is $X_2 = 214112321233432$. More generally, it is easy to see that the n-th iteration is given by

$$X_n = \varphi_1(X_{n-1})1X_{n-1}2X_{n-1}3\varphi_2(X_{n-1}),$$

where the function $\varphi_1(A)$ reverses the letters in the word A and makes the substitution corresponding to the permutation 4123, that is, 1 becomes 4 etc. The function φ_2 does the same, except with 4123 replaced by 2341. In this paper, we are interested in the words X_n, for $n = 1, 2, \ldots$, which appear as the subdivision stages of the Peano curve. We call these words the Peano words.

In Paper VIII ([KitMans5]) we consider the Peano words and find the number of occurrences of the patterns

$$12,\ 21,\ 1^\ell,\ \tau_1(x,y) = [x\text{-}y^\ell),\ \tau_2(x,y) = (x^\ell\text{-}y] \text{ and } \tau_3(x,y,z) = [x\text{-}y^\ell\text{-}z],$$

where $x, y, z \in \{1, 2, 3\}$, $y^\ell = y\text{-}y\text{-}\cdots\text{-}y$ (ℓ times), and we recall that "[" in $p = [x - w)$ indicates that in an occurrence of p, the letter corresponding to the x must be the first letter of the word. For example, the number of occurrences of the pattern 12 in X_n, according to Theorem 4 in Paper VIII, is equal to either $\frac{2}{5}(4 \cdot 16^k + 1)$ or $\frac{2}{5}(16^{k+1} - 1)$ depending on whether n is odd or even.

Let $N_\tau(W)$ denote the number of occurrences of the pattern τ in the word W. Let S_1 and S_2 denote the following:

$$S_1 = \binom{4^{2k} - 2^{2k}}{\ell} + \binom{4^{2k}}{\ell} + \binom{4^{2k} + 2^{2k} - 1}{\ell}, \quad S_2 = \binom{4^{2k+1}}{\ell} + \binom{4^{2k+1} - 2^{2k+1}}{\ell}.$$

Tables 8 and 9 give all the results concerning the patterns $\tau_1(x, y)$, $\tau_2(x, y)$ and $\tau_3(x, y, z)$ except those triples (x, y, z), for which $N_{\tau_3(x,y,z)}(X_n) = 0$ for all n.

Paper IX. Let us define the sigma-sequence and the sigma-words. In [Evdok1, Yab], Evdokimov constructed chains of maximal length in the n-dimensional unit cube using the *sigma-sequence*. The sigma-sequence w_σ was defined there by the following recursive scheme:

x	y	z	$N_{\tau_3(x,y,z)}(X_{2k+1})$	$N_{\tau_3(x,y,z)}(X_{2k+2})$
1	1	1	0	$\binom{4^{2k+1}-2}{\ell}$
1	1	2	$\binom{4^{2k}-1}{\ell}$	0
1	2	1	0	S_2
1	2	2	$\binom{4^{2k}-1}{\ell}$	0
2	1	2	0	$\binom{4^{2k+1}}{\ell}$
1	2	3	$\binom{4^{2k}+2^{2k}-1}{\ell}$	0
1	3	2	$\binom{4^{2k}-2^{2k}}{\ell}$	0

Table 9: Generalized patterns having 3 letters (Paper VIII)

$$C_1 = 1, \qquad D_1 = 2$$

$$C_{k+1} = C_k 1 D_k, \qquad D_{k+1} = C_k 2 D_k$$

$$k = 1, 2, \ldots$$

and $w_\sigma = \lim_{k\to\infty} C_k$. Thus, the initial letters of w_σ are 11211221112212.... We call the words C_k the *sigma words*. The first four values of the sequence $\{C_k\}_{k\geq 1}$ are 1, 112, 1121122, 112112211122122.

In [Kit] an equivalent definition of w_σ was given: any natural number $n \neq 0$ can be presented unambiguously as $n = 2^t(4s + \sigma)$, where $\sigma < 4$, and t is the greatest natural number such that 2^t divides n. If n runs through the natural numbers then σ runs through some sequence consisting of 1s and 3s. If we substitute 2 for 3 in this sequence, we get w_σ.

In Paper IX ([Kit4]) we give either an explicit formula or recurrence relation for the number of occurrences for some classes of patterns, subwords and subsequences in the sigma-words. In particular, Theorem 4 allows us to find the number of occurrences of an arbitrary generalized pattern without internal dashes of length ℓ, provided we know certain four numbers that can be easily calculated for the words C_k, D_k, C_{k+1} and D_{k+1}, where $k = \lceil \log_2 \ell \rceil$. Theorem 9 gives a recurrence relation for counting occurrences of patterns of the form τ_1-τ_2. In Section 6 we discuss occurrences of patterns of the form τ_1-τ_2-$\cdots$-τ_k, where the pattern τ_i does not overlap with the patterns τ_{i-1} and τ_{i+1} for $i = 1, 2, \ldots, k-1$. Finally, Section 7 deals with patterns of the form $[\tau_1$-τ_2-$\cdots$-$\tau_k]$, $[\tau_1$-τ_2-$\cdots$-$\tau_k)$ and $(\tau_1$-τ_2-$\cdots$-$\tau_k]$ in the Babson-Steingrímsson notation.

To formulate some of the results from Paper IX we need the following definitions.

Suppose a word $W = AaB$, where A and B are some words of *the same length*, and a is a single letter. We define the *kernel of order k* for the word

W to be the subword consisting of the $k-1$ rightmost letters of A, the letter a, and the $k-1$ leftmost letters of B. We denote it by $\mathcal{K}_k(W)$. For example, $\mathcal{K}_3(111211221) = 12112$. If $|A| < k-1$ then we set $\mathcal{K}_k(W) = \epsilon$, that is the kernel in this case is the empty word. Also, $m_k(\tau, W)$ denotes the number of occurrences of the pattern τ in $\mathcal{K}_k(W)$.

The following theorems are proved in Paper IX.

Theorem 12. *Let $\tau = \tau_1\tau_2\ldots\tau_\ell$ be an arbitrary generalized pattern without internal dashes that consists of 1s and 2s. Suppose $k = \lceil \log_2 \ell \rceil$, $a = m_\ell(\tau, D_k1C_k)$, and $b = m_\ell(\tau, D_k2C_k)$. Then for $n > k+1$, we have*

$$c_n^\tau = (a + b + c_{k+1}^\tau + d_{k+1}^\tau) \cdot 2^{n-k-2} - b,$$

$$d_n^\tau = (a + b + c_{k+1}^\tau + d_{k+1}^\tau) \cdot 2^{n-k-2} - a.$$

Theorem 13. *Let $p = \tau_1\text{-}\tau_2$ be a generalized pattern such that $|\tau_1| = k_1$ and $|\tau_2| = k_2$. Suppose $k = \lceil \log_2(k_1 + k_2 - 1) \rceil$. Let the following denote the number of occurrences of the subwords τ_1 and τ_2 in the kernels (recall that by definitions $|C_n| = |D_n|$):*

$$\begin{array}{ll} a_{\tau_1} = m_{k_1}(\tau_1, D_k1C_k) & a_{\tau_2} = m_{k_2}(\tau_2, D_k1C_k) \\ b_{\tau_1} = m_{k_1}(\tau_1, D_k2C_k) & b_{\tau_2} = m_{k_2}(\tau_2, D_k2C_k) \end{array}$$

Also, let r_1^a (resp. r_2^a, r_1^b, r_2^b) denote the number of occurrences of overlapping subwords τ_1 and τ_2 in the word D_k1C_k (resp. D_k1C_k, D_k2C_k, D_k2C_k), where $\tau_1 \in \mathcal{K}_{k_1}(D_k1C_k)$ and $\tau_2 \in C_k$ (resp. $\tau_1 \in D_k$ and $\tau_2 \in \mathcal{K}_{k_2}(D_k1C_k)$, $\tau_1 \in \mathcal{K}_{k_1}(D_k2C_k)$ and $\tau_2 \in C_k$, $\tau_1 \in D_k$ and $\tau_2 \in \mathcal{K}_{k_2}(D_k2C_k)$).

Besides, we assume that we know $c_n^{\tau_i}$ and $d_n^{\tau_i}$ for $n > n_i$, $i = 1, 2$. Then for $n > \max(k+1, n_1+1, n_2+1)$, c_n^τ and d_n^τ are given by the following recurrence:

$$\begin{pmatrix} c_n^\tau \\ d_n^\tau \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_{n-1}^\tau \\ d_{n-1}^\tau \end{pmatrix} + \begin{pmatrix} \alpha_n \\ \beta_n \end{pmatrix},$$

where

$$\alpha_n = (c_{n-1}^{\tau_1} + a_{\tau_1} - r_1^a)d_{n-1}^{\tau_2} + (a_{\tau_2} - r_2^a)c_{n-1}^{\tau_1}$$

and

$$\beta_n = (c_{n-1}^{\tau_1} + b_{\tau_1} - r_1^b)d_{n-1}^{\tau_2} + (b_{\tau_2} - r_2^b)c_{n-1}^{\tau_1}.$$

Theorem 14. *Let $\tau = \tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k$ be a generalized pattern such that $|\tau_i| = k_i$ for $i = 1, 2, \ldots, k$. We assume that for $i = 1, 2, \ldots, k-1$, the subword τ_i does not overlap with the subwords τ_{i-1} and τ_{i+1} in the following sense: no suffix of τ_{i-1} is a prefix of τ_i and no suffix of τ_i is a prefix of τ_{i+1}.*

Suppose $\ell_i = \lceil \log_2 k_i \rceil$, $\ell = \max_i \ell_i$, *and for the subwords* τ_i *we have* $a_i = m_{k_i}(\tau_i, D_{\ell_i} 1 C_{\ell_i})$ *and* $b_i = m_{k_i}(\tau_i, D_{\ell_i} 2 C_{\ell_i})$, *for* $i = 1, 2, \ldots, k$.

We assume that we know $c_{n-1}^{\tau_1\text{-}\cdots\text{-}\tau_i}$ *and* $d_{n-1}^{\tau_{i+1}\text{-}\cdots\text{-}\tau_k}$ *for each* $1 \le i \le k-1$ *and for all* $n > n^\star$. *Then for all* $n > \max(\ell + 1, n^\star + 1)$, c_n^τ *and* d_n^τ *are given by the following recurrence:*

$$\begin{pmatrix} c_n^\tau \\ d_n^\tau \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_{n-1}^\tau \\ d_{n-1}^\tau \end{pmatrix} + \sum_{i=1}^{k-1} \begin{pmatrix} c_{n-1}^{e(i)} \cdot d_{n-1}^{f(i)} \\ c_{n-1}^{e(i)} \cdot d_{n-1}^{f(i)} \end{pmatrix} + \sum_{i=1}^{k} \begin{pmatrix} a_i \cdot c_{n-1}^{e(i-1)} \cdot d_{n-1}^{f(i)} \\ b_i \cdot c_{n-1}^{e(i-1)} \cdot d_{n-1}^{f(i)} \end{pmatrix},$$

where $e(i) = \tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_i$ *and* $f(i) = \tau_{i+1}\text{-}\tau_{i+2}\text{-}\cdots\text{-}\tau_k$.

Theorem 15. *Suppose* τ_1 *and* τ_2 *are two patterns without internal dashes such that* $|\tau_1| = k_1$ *and* $|\tau_2| = k_2$. *Also, suppose* $\ell_1 = \log_2(k_1 + 1)$, $\ell_2 = \log_2(k_2 + 1)$ *and* $\ell = \log_2(k_1 + k_2 + 1)$.

Let $a(\tau_1, \tau_2)$ *be the number of overlapping subwords* τ_1 *and* τ_2 *in* C_ℓ *such that* τ_1 *consists of the* k_1 *leftmost letters of* C_ℓ; $b(\tau_1, \tau_2)$ *is the number of overlapping subwords* τ_1 *and* τ_2 *in* C_ℓ *such that* τ_2 *consists of the* k_2 *rightmost letters of* C_ℓ.

We assume that we know $c_n^{\tau_i}$ *and* $d_n^{\tau_i}$ *for* $i = 1, 2$ *and for all* $n > n^\star$.

i. For $n \ge \max(\ell_1, n^\star)$,

$$c_n^{[\tau_1\text{-}\tau_2)} = \begin{cases} c_n^{\tau_2} - a(\tau_1, \tau_2), & \text{if } C_{\ell_1} \text{ begins with } \tau_1, \\ 0, & \text{otherwise.} \end{cases}$$

ii. For $n \ge \max(\ell_2, n^\star)$,

$$c_n^{(\tau_1\text{-}\tau_2]} = \begin{cases} c_n^{\tau_1} - b(\tau_1, \tau_2), & \text{if } C_{\ell_2} \text{ ends with } \tau_2, \\ 0, & \text{otherwise.} \end{cases}$$

iii. For $n \ge \ell$,

$$c_n^{[\tau_1\text{-}\tau_2]} = \begin{cases} 1, & \text{if } C_\ell \text{ begins with } \tau_1 \text{ and ends with } \tau_2, \\ 0, & \text{otherwise.} \end{cases}$$

iv. For $n \ge \max(\ell_1, n^\star)$,

$$d_n^{[\tau_1\text{-}\tau_2)} = \begin{cases} d_n^{\tau_2} - a(\tau_1, \tau_2), & \text{if } D_{\ell_1} \text{ begins with } \tau_1, \\ 0, & \text{otherwise.} \end{cases}$$

v. For $n \geq \max(\ell_2, n^\star)$,

$$d_n^{(\tau_1\text{-}\tau_2]} = \begin{cases} d_n^{\tau_1} - b(\tau_1, \tau_2), & \textit{if } D_{\ell_2} \textit{ ends with } \tau_2, \\ 0, & \textit{otherwise.} \end{cases}$$

vi. For $n \geq \ell$,

$$d_n^{[\tau_1\text{-}\tau_2]} = \begin{cases} 1, & \textit{if } D_\ell \textit{ begins with } \tau_1 \textit{ and ends with } \tau_2, \\ 0, & \textit{otherwise.} \end{cases}$$

So, in Paper IX we count occurrences of certain patterns, subsequences and subwords in the sigma-words, which are particular initial subwords of w_σ. However, the challenging question is to find the number of occurrences of patterns, subsequences and subwords in an arbitrary initial subword of w_σ, or more generally, in a subword of w_σ starting in position i and ending in position j.

Bibliography

[Adian] Adian S. I.: *The Burnside problem and identities in groups.* Translated from the Russian by John Lennox and James Wiegold. Ergebnisse der Mathematik und ihrer Grenzgebiete [Results in Mathematics and Related Areas], 95. Springer-Verlag, Berlin-New York, (1979).

[BabStein] Babson E., Steingrímsson E.: Generalized permutation patterns and a classification of the Mahonian statistics, Sém. Lothar. Combin. **44** (2000), Art. B44b, 18 pp.

[Bona] Bóna M.: Permutations avoiding certain patterns; The case of length 4 and generalizations. *Discrete Math.* **175** (1997), 55–67.

[Bona1] Bóna M.: Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A* **80** (1997), no. 2, 257–272.

[Bona2] Bóna M.: The permutation classes equinumerous to the smooth class. *Electron. J. Combin.* **5** (1998), no. 1, Research Paper 31, 12 pp. (electronic).

[Bona3] Bóna M.: The Solution of a Conjecture of Wilf and Stanley for all layered patterns. *J. Combin. Theory Ser. A* **85** (1999), 96–104.

[Bona4] Bóna M.: The number of permutations with exactly r 132-subsequences is P-recursive in the size!. *Adv. Appl. Math.* **18** (1997), 510–522.

[Burstein] Burstein A., Enumeration of words with forbidden patterns, Ph.D. thesis, University of Pennsylvania, 1998.

[BurMans1] Burstein A., Mansour T.: Words restricted by patterns with at most 2 distinct letters, *Electron. J. Combin.* **9**, no. 2, #R3 (2002).

[BurMans2] Burstein A., Mansour T.: Words restricted by 3-letter generalized multipermutation patterns, preprint CO/0112281[1].

[BurMans3] Burstein A., Mansour T.: Counting occurrences of some subword patterns, preprint CO/0204320.

[1] References of type CO/xxxxxxx refer to preprints in Archive http://arxiv.org/

[Carpi] Carpi A.: On the number of abelian square-free words on four letters, Discrete Appl. Math. **81** (1998), 155–167.

[ChowWest] Chow T., West J.: Forbidden subsequences and Chebyshev polynomials. *Discrete Math.* **204** (1999), no. 1–3, 119–128.

[Claes] Claesson A.: Generalised Pattern Avoidance, European J. Combin. **22** (2001), no. 7, 961–971.

[ClaesMans1] Claesson A., Mansour T.: Permutations avoiding a pair of generalized patterns of length three with exactly one dash, preprint CO/0107044.

[ClaesMans2] Claesson A., Mansour T.: Counting Occurrences of a Pattern of Type (1,2) or (2,1) in Permutations, *Adv. in Appl. Math.* **29** (2002), 293–310.

[Dekk] Dekking F. M.: Strongly non-repetitive sequences and progression-free sets, Journal Com. Theory, Vol. **27-A**, no. 2 (1979), 181–185.

[ElizNoy] Elizalde S., Noy M.: Enumeration of Subwords in Permutations, Proceedings of FPSAC 2001.

[ErdSze] Erdös P., Szekeres G.: A combinatorial problem in geometry, *Compositio Mathematica* **2** (1935), 463–470.

[Evdok] Evdokimov A. A.: Private communication (2001).

[Evdok1] Evdokimov A. A.: On the Maximal Chain Length of an Unit n-dimensional Cube, Maths Notes **6**, no. 3 (1969), 309–319. (Russian)

[Evdok2] Evdokimov A. A.: Strongly asymmetric sequences generated by a finite number of symbols, Dokl. Akad. Nauk SSSR **179** (1968), 1268–1271. (Russian) English translation in: Soviet Math. Dokl. **9** (1968), 536–539.

[Frid] Frid A. E.: On the frequency of factors in a D0L word, J. Automata, Languages and Combinatorics, Otto-von-Guericke-Univ., Magdeburg **3**, no. 1 (1998), 29–41.

[GelbOlm] Gelbaum B., Olmsted J.: *Counterexamples in Analysis*, Holden-Day, San Francisco, London, Amsterdam, (1964).

[Gessel] Gessel I. M.: Symmetric functions and P-recursiveness, *J. Combin. Theory Ser. A* **53** (1990), 257–285.

[GWW] Gessel I. M., Weinstein J., Wilf H. S.: Lattice walks in Z^d and permutations with no long ascending subsequences, *Electron. J. Combin.* **5**, no. 1, #R2 (1998).

[GoulJack] Goulden I. P., Jackson D. M.: *Combinatorial Enumeration*, A Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons Inc., New York, (1983).

[Justin] Justin J.: Characterization of the repetitive commutative semigroups, *J. Algebra* **21** (1972), 87–90.

[Ker] Keränen V.: Abelian squares are avoidable on 4 letters, In W. Kuich, editor, Proc. ICALP'92, *Lecture Notes in Comp. Sci.* **623** (1992), 41–52.

[Kit] Kitaev S.: There are no iterated morphisms that define the Arshon sequence and the sigma-sequence, *J. Automata, Languages and Combinatorics*, to appear.

[Kit1] Kitaev S.: Multi-avoidance of generalised patterns, *Discrete Math.*, to appear.

[Kit2] Kitaev S.: Partially ordered generalized patterns, *Discrete Math.*, to appear.

[Kit3] Kitaev S.: Generalized pattern avoidance with additional restrictions, preprint math.CO/0205215.

[Kit4] Kitaev S.: The sigma-sequence and counting occurrences of some patterns, preprint CO/0211260.

[KitMans1] Kitaev S., Mansour T.: Simultaneous avoidance of generalized patterns, preprint CO/0205182.

[KitMans2] Kitaev S., Mansour T.: On multi-avoidance of generalized patterns, preprint CO/0209340.

[KitMans3] Kitaev S., Mansour T.: Partially Ordered generalized patterns and k-ary words, preprint CO/0210023

[KitMans4] Kitaev S., Mansour T.: Counting the occurrences of generalized patterns in words generated by a morphism, preprint CO/0210170.

[KitMans5] Kitaev S., Mansour T.: The Peano curve and counting occurrences of some patterns, preprint CO/0210268.

[Knuth] Knuth D. E.: *The Art of Computer Programming*, 2nd ed. Addison-Wesley, Reading, MA (1973).

[Kol] Kolotov A. T.: Aperiodic sequences and functions of the growth of algebras, Algebra i Logika **20** (1981), no. 2, 138–154. (Russian)

[Krat] Krattenthaler C.: Permutations with restricted patterns and Dyck paths, *Adv. in Appl. Math.* **27** (2001), 510–530.

[Kremer] Kremer D.: Permutations with forbidden subsequences and a generalized Schröder number, *Discrete Math.* **218** (2000), 121–130.

[Lind] Lindenmayer A.: Mathematical models for cellular interaction in development, Parts I and II, *J. Theoretical Biology*, **18** (1968), 280–315.

[LindRoz] Lindenmayer A., Rozenberg G.: *Automata, languages, development*, North-Holland Publishing Co., Amsterdam-New York-Oxford (1976).

[Lothaire] Lothaire M.: *Combinatorics on Words*, Encyclopedia of Mathematics, Vol. **17**, Addison-Wesley (1986). Reprinted in the *Cambridge Mathematical Library*, Cambridge University Press, Cambridge UK (1997).

[MacMah] MacMahon P. A.: *Combinatory Analysis.* Chelsea, New York, 1960 (Originally published by Camb. Univ. Press, London, 1915/16).

[Mans1] Mansour T.: Continued fractions and generalized patterns, *European J. Combin.* **23**, no. 3 (2002), 329–344.

[Mans2] Mansour T.: Continued fractions, statistics, and generalized patterns, *Ars Combin.*, to appear.

[Mans3] Mansour T.: Restricted 1-3-2 permutations and generalized patterns, *Ann. Comb.* **6**, no. 1 (2002), 65–76.

[MV1] Mansour T., Vainshtein A.: Restricted permutations, continued fractions, and Chebyshev polynomials, *Electron. J. Combin.* **7**, no. 1 (2000), Research Paper 17, 9 pp. (electronic).

[MV2] Mansour T., Vainshtein A.: Restricted 132-avoiding permutations, *Adv. in Appl. Math.* **126** (2001), no. 3, 258–269.

[MV3] Mansour T., Vainshtein A.: Layered restrictions and Chebyshev polynomials, *Ann. Comb.* **5** (2001), 451–458.

[MV4] Mansour T., Vainshtein A.: Restricted permutations and Chebyshev polyonomials, *Sém. Lothar. Combin.* **47** (2002), Article B47c.

[MorseHedl] Morse M., Hedlung G.: Unending chess, symbolic dynamics and a problem in semigroups, Duke Math. Journal, Vol. **11**, no. 1 (1944), 1–7.

[NooZeil] Noonan J., Zeilberger D.: The enumeration of permutations with a prescribed number of "forbidden" patterns, *Adv. Appl. Math.* **17** (1996), 381–407.

[Pleas] Pleasants P.: Non-repetitive sequences, *Proc. Camb. Phil. Soc.*, Vol. **68** (1970), 267–274.

[Regev] Regev A.: Asymptotic values for degrees associated with strips of Young diagrams, *Adv. in Math.* **41** (1981), 115–136.

[Rich] Richards D.: Ballot sequences and restricted permutations. Ars Combin. (1988), 25:83–86.

[Rob] Robertson A.: Permutations containing and avoiding 123 and 132 patterns, *Discrete Math. Theor. Comput. Sci.* **3** (1999), no. 4, 151–154 (electronic).

[RWZ] Robertson A., Wilf H.: and D. Zeilberger: Permutation patterns and continued fractions, *Electron. J. Combin.* **6** (1999), no. 1, Research Paper 38, 6 pp. (electronic).

[Salomaa] Salomaa A.: *Jewels of Formal Language Theory*, Computer Science Press (1981).

[SimSch] Simion R., Schmidt F.: Restricted permutations, *European J. Combin.* **6**, no. 4 (1985), 383–406.

[SloPlo] Sloane N, J. A. and Plouffe S.: *The Encyclopedia of Integer Sequences*, Academic Press, (1995). http://www.research.att.com/∼njas/sequences/.

[Stank] Stankova Z.: Forbidden Subsequences, *Discrete Math.* **132** (1994), 291–316.

[Stanley] Stanley R. P.: Enumerative Combinatorics, Vol. 1, Cambridge University Press, (1997).

[West] West J.: Permutations with forbidden subsequences; and, Stack sortable permutations, PHD-thesis, Massachusetts Institute of Technology, (1990).

[West1] West J.: Generating trees and the Catalan and Schröder numbers, *Discrete Math.* **146** (1995), 247–262.

[West2] West J.: Generating trees and forbidden subsequences, *Discrete Math.* **157** (1996), 363–372.

[Zeil] Zeilberger D.: Holonomic systems for special functions, *J. Computational and Applied Mathematics*, **32** (1990), 321–368.

[Yab] Yablonsky S. V.: *Discrete mathematics and mathematical problems of cybernetics*, Nauka, Vol. **1**, Moscow (1974), 112–116. (Russian)

Paper I

Multi-avoidance of generalised patterns

Multi-Avoidance of Generalised Patterns

Sergey Kitaev [1]

Abstract

Recently, Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. We investigate simultaneous avoidance of two or more 3-patterns without internal dashes, that is, where the pattern corresponds to a contiguous subword in a permutation.

1.1 Introduction and Background

We write permutations as words $\pi = a_1 a_2 \cdots a_n$, whose letters are distinct and usually consist of the integers $1, 2, \ldots, n$.

An occurrence of a pattern p in a permutation π is "classically" defined as a subsequence in π (of the same length as the length of p) whose letters are in the same relative order as those in p. Formally speaking, for $r \leq n$, we say that a permutation σ in the symmetric group $\mathcal{S}_n$ has an occurrence of the pattern $p \in \mathcal{S}_r$ if there exist $1 \leq i_1 < i_2 < \cdots < i_r \leq n$ such that $p = \sigma(i_1)\sigma(i_2)\ldots\sigma(i_r)$ in reduced form. The *reduced form* of a permutation σ on a set $\{j_1, j_2, \ldots, j_r\}$, where $j_1 < j_2 < \cdots < j_r$, is a permutation σ_1 obtained by renaming the letters of the permutation σ so that j_i is renamed i for all $i \in \{1, \ldots, r\}$. For example, the reduced form of the permutation 3651 is 2431.

In [1] Babson and Steingrímsson introduced *generalised permutation patterns* that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. In order to avoid confusion we write a "classical" pattern, say 231, as 2-3-1, and if we write, say 2-31, then we mean that if this pattern occurs in the permutation, then the letters in the permutation that correspond to 3 and 1 are adjacent. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, whereas the pattern 2-3-1 occurs, in addition, in the subwords 562 and 563.

The motivation for introducing these patterns in [1] was the study of Mahonian statistics. A number of interesting results on generalised patterns were obtained in [5]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were shown there.

In this paper we consider 3-patterns without internal dashes, that is, generalised patterns of the form xyz. Thus, such patterns correspond to contiguous subwords anywhere in a permutation. For example the permutation $\pi = 12345$ has 3 occurrences of the pattern 123 but 10 occurrences of the classical pattern 1-2-3. Patterns without internal dashes were considered by Elizalde and Noy

[1] Matematiska Institutionen, Chalmers tekniska högskola and Göteborgs universitet, S-412 96 Göteborg, Sweden; E-mail: kitaev@math.chalmers.se

in [6]. In that paper, there is a number of results on the distribution of several classes of patterns without internal dashes. In particular, formulas are given for the bivariate exponential generating functions that count permutations by the number of occurrences of any given 3-pattern. Those formulas give rise to the exponential generating functions for the number of permutations that avoid any 3-pattern.

As in the paper by Simion and Schmidt [11], dealing with the classical patterns, one can consider the case when permutations have to avoid two or more generalised patterns simultaneously. A number of such cases were considered in [5]. However, except for the simultaneous avoidance of the patterns 123 and 132, and three more pairs that are essentially equivalent to this, there are no other results for multi-avoidance of the patterns without internal dashes. In this paper we give either an explicit formula or a recursive formula for almost all cases of simultaneous avoidance of more than two patterns. We also mention what is known about double restrictions.

1.2 Preliminaries

Since we only treat patterns of length 3, and permutations of length 1 or 2 avoid all such patterns, we always assume that our permutations have length $n \geq 3$.

Obviously, no permutation avoids all six patterns of length three. Only the increasing permutation $12\dots n$ avoids all 3-patterns but 123, and only the decreasing permutation avoids all but 321.

Consider now permutations that avoid all but one 3-pattern, different from 123 and 321, Obviously, there is exactly one such 3-permutation. However, for $n \geq 4$ there is no such permutation. Indeed, if the permutation $\pi = a_1a_2\dots a_n$ avoids the patterns 123 and 321, then the letters of π alternate in size. That means that $a_1a_2a_3$ and $a_2a_3a_4$ form different patterns and thus π has an occurrence of a forbidden pattern.

There are, of course, $\binom{6}{k}$ sets consisting of k different 3-patterns, so we have 15 sets of two 3-patterns, 20 with three 3-patterns and 15 with four. So we have 50 different sets having more than one restriction. But we can simplify our work by partitioning the sets into equivalence classes in the way shown below and it will be enough to consider only 18 sets of restrictions.

The *reverse* $R(\pi)$ of a permutation $\pi = a_1a_2\dots a_n$ is the permutation $a_na_{n-1}\dots a_1$. The *complement* $C(\pi)$ is the permutation $b_1b_2\dots b_n$ where $b_i = n+1-a_i$. Also, $R\circ C$ is the composition of R and C. For example, $R(13254) = 45231$, $C(13254) = 53412$ and $R\circ C(13254) = 21435$. We call these bijections of S_n to itself *trivial*, and it is easy to see that for any pattern p the number $A_p(n)$ of permutations avoiding the pattern p is the same as for the patterns $R(p)$, $C(p)$ and $R\circ C(p)$. For example, the number of permutations that avoid the pattern 132 is the same as the number of permutations that avoid the pattern 231. This property holds for sets of patterns as well. If we apply one of the trivial bijections to all patterns of a set G, then we get a set G' for which $A_{G'}(n)$ is equal to $A_G(n)$. For example, the number of permutations avoiding

$\{123, 132\}$ equals the number of those avoiding $\{321, 312\}$ because the second set is obtained from the first one by complementing each pattern.

So up to equivalence modulo the trivial bijections we need to investigate 18 sets of restrictions that are represented in the table below.

We define the *double factorial* $n!!$ by $0!! = 1$, and, for $n > 0$,

$$n!! = \begin{cases} n \cdot (n-2) \cdots 3 \cdot 1, & \text{if } n \text{ is odd,} \\ n \cdot (n-2) \cdots 4 \cdot 2, & \text{if } n \text{ is even.} \end{cases}$$

Recall that the *n-th Catalan number* is defined by

$$C_n = \frac{1}{n+1}\binom{2n}{n}.$$

Instead of writing $A_G(n)$ for a set G of patterns, we will write $A(n)$ since it will be unambiguous what set of patterns is under consideration.

Class	**Restrictions**	**Formula**
1	123, 321, 132, 312 123, 321, 231, 213	2
2	123, 312, 132, 213 321, 213, 231, 312 123, 231, 231, 132 321, 132, 312, 231	2
3	132, 231, 213, 312	2
4	123, 321, 132, 231 123, 321, 312, 213	2, if $n = 3$ 0, if $n > 3$
5	132, 213, 312, 321 231, 312, 213, 123 213, 132, 231, 321 312, 231, 132, 123	$n - 1$
6	123, 321, 132, 213 123, 321, 231, 312	$2C_k$, if $n = 2k + 1$ $C_k + C_{k-1}$, if $n = 2k$
7	123, 132, 213 231, 312, 321	$\binom{n}{\lfloor n/2 \rfloor}$
8	123, 132, 231 123, 213, 312 132, 231, 321 213, 312, 321	n

Class	Restrictions	Formula
9	132, 213, 231 132, 213, 312 132, 231, 312 213, 231, 312	$1 + 2^{n-2}$
10	123, 132, 312 123, 213, 231 132, 312, 321 213, 231, 321	Recursive Formula: $A(0) = 1;\ A(1) = 1;$ $A(n) = \sum_i \binom{n-i-1}{i} A(n-2i-1) + ((n+1) \bmod 2)$ The first few numbers: 1, 1, 2, 3, 6, 13, 29, 72, 185 . . .
11	123, 321, 132 123, 321, 231 123, 321, 312 123, 321, 213	$(n-1)!! + (n-2)!!$
12	123, 231, 312 132, 213, 321	?
13	123, 231 321, 132 321, 213 123, 312	?
14	213, 231 312, 132	?
15	132, 213 231, 312	?
16	123, 321	$2E_n$, where E_n is the n-th Euler number
17	123, 132 321, 231 321, 312 123, 213	the number of involutions in S_n (Claesson, [5])
18	132, 231 312, 213	2^{n-1}

We now give proofs and comments for the results represented in the table.

1.3 Proofs, remarks, comments

From now on, when talking about class **i**, we mean the first set of patterns in the equivalence class **i** according to the table above. Thus, for instance, **8** will be taken to refer to the set of patterns $\{123, 132, 231\}$.

Let us consider class **1**. There are only two patterns, namely 231 and 213, that are *allowed* to occur. Suppose a permutation $\pi = a_1 a_2 \dots a_n$ avoids the patterns from **1**. If $a_1 a_2 a_3$ forms a 231-pattern then $a_2 a_3 a_4$ has to form a 213-pattern since $a_2 > a_3$. It is easy to see that $a_3 a_4 a_5$ has to form the pattern

231 and so on. Moreover, if we consider the letters in even positions from left to right then we get an increasing sequence any element of which is greater then any element in an odd position; letters in odd positions form a decreasing sequence when read from left to right. From this we see that there is a unique such permutation in which the letters $\{1, 2, \ldots, \lfloor (n+1)/2 \rfloor\}$ are in the odd positions in decreasing order, and all other letters are in the even positions in increasing order.

By the same argument there is only one permutation that avoids **1** and begins with a 213-pattern. Thus, in this case $A(n) = 2$.

For class **2** there are only two permutations that avoid it, namely $\pi_1 = n(n-1)(n-2)\ldots 1$ and $\pi_2 = (n-1)n(n-2)(n-3)\ldots 1$. This is because n has to be either in the leftmost position or in the second position from the left, for otherwise we have either an occurrence of the pattern 123 or of the pattern 213 that involves n. To the right of n we have to have decreasing order because otherwise we have an occurrence of a 312- or a 213-pattern. Moreover, if n is in the second position from the left then in the leftmost position we must have the letter $(n-1)$ because otherwise $(n-1)$ must be in the third place and the first three letters form a 132-pattern.

There are obviously only two permutations that avoid class **3**. They are $\pi_1 = 12\ldots n$ and $\pi_2 = n(n-1)\ldots 1$.

For class **4**, only the patterns 213 and 312 are allowed. Obviously, for $n = 3$ we have $A(n) = 2$. Suppose $n > 3$. If a permutation $\pi = a_1 a_2 \ldots a_n$ avoids **4**, then it has to be that $a_2 < a_3$, because $a_1 a_2 a_3$ forms either a 213- or a 312-pattern. But this means that $a_2 a_3 a_4$ cannot form a 213- or a 312-pattern, whence $A(n) = 0$.

For class **5**, n has to be either in the rightmost position or in the second position from the right, for otherwise we have an occurrence of a 312- or a 321-pattern. Moreover we must have increasing order to the left of n because otherwise we have an occurrence of a 213- or a 312-pattern. Thus there is only one permutation with n in the rightmost position.
If n is in the second position from the right then $(n-1)$ cannot be in the rightmost position, because in this case we have an occurrence of a 132-pattern that involves n and $(n-1)$. So in this case $(n-1)$ has to be in the third position from the right, and we can put any letter i other than $n-1$ and n in the rightmost position. This means that $A(n) = 1 + (n-2) = n-1$.

Class **6** will be considered in Theorem 2 below.

Theorem 1. *For class* **7** *we have* $A(n) = \binom{n}{\lfloor n/2 \rfloor}$.

Proof. Let us construct a permutation that avoids class **7** by inserting the numbers $1, 2, \ldots, n$ into n slots and observing the following:

The number 1 can be placed either in the rightmost slot or in the second slot from the right, since otherwise, independently of what we have to the right of 1 in the permutation, we get either a 123- or a 132-pattern, which is prohibited. If 1 has already been placed then 2 must be placed in such way that:

1. The two slots immediately to the right of 2 are not both empty, for otherwise we will get an occurrence of either a 123- or a 132-pattern involving 2;

2. If 1 is not in the rightmost slot then 2 cannot be immediately to the left of 1, because in this case we will get an occurrence of a 213-pattern involving the letters 1 and 2.

In general it is easy to see that if i letters have been placed then for some j such that $0 \leq j \leq i$ the rightmost j slots are non-empty and the $2 \cdot (i-j)$ slots immediately to the left of these j slots are alternatingly empty and non-empty. By an argument analogous to the above we can only place the letter $(i+1)$ into either

0) the rightmost empty slot or

1) the second empty slot to the left of the leftmost non-empty slot.

If we place 1 next to the rightmost slot we assume that we use option 1).

Let us call the leftmost two slots *critical* slots. When we fill one of the critical slots, there is only one way to place the remaining letters, using option 0), since in this case, option 1) can not be applied any more.

So any permutation with the right properties can be written as a sequence of 0s and 1s according to which option we use in placing the ith letter $(i = 1, 2, \ldots)$ and we stop writing a (0,1)-sequence whenever we reach one of the critical slots.

Let us call the (0,1)-sequences thus constructed *legal sequences*.

Example 1. Let $n = 6$. The (0,1)-sequence 01101 is a legal sequence that corresponds to the permutation 5736241. But 1111 is not a legal sequence, because after 3 steps, namely 111, we are already in a critical slot and must stop writing the (0,1)-sequence.

Since obviously there is a bijection between legal sequences and permutations in class **7**, our problem is to count all possible legal sequences. We prove by induction on n that the number of such sequences is equal to $\binom{n}{\lfloor n/2 \rfloor}$.

It is easy to check this for $n = 3$.

Assuming that for all $i < n$ we have $A(i) = \binom{i}{\lfloor i/2 \rfloor}$, we prove the statement for $A(n)$. We consider separately the cases when n is even and odd.

Suppose n is even. The number of legal sequences that begin with 0 is obviously equal to

$$A(n-1) = \binom{n-1}{\lfloor (n-1)/2 \rfloor} = \binom{n-1}{(n-2)/2}.$$

Now we prove that the number of legal sequences beginning with 1 is equal to the number of legal sequences beginning with 0. We shall show that a bijection between these legal sequences is given by the correspondence $0X \leftrightarrow 1X$, where $0X$ is any legal (0,1)-sequence of length ℓ, $\frac{n}{2} \leq \ell \leq n-1$, that starts with 0. From this it follows that

$$A(n) = 2A(n-1) = \tbinom{n-1}{(n-2)/2} + \tbinom{n-1}{(n-2)/2} =$$

$$= \tbinom{n-1}{(n-2)/2} + \tbinom{n-1}{n/2} = \tbinom{n}{n/2} = \tbinom{n}{\lfloor n/2 \rfloor}.$$

So the problem is to prove that $0X \leftrightarrow 1X$ is a bijection.

We use induction on even n. If $n = 2$ then we only have the critical slots and thus there are only two legal sequences possible, namely 0 and 1. In this case $X = \emptyset$ and we have that $0X \leftrightarrow 1X$ is a bijection.

Suppose for all even m less than n the correspondence $0X \leftrightarrow 1X$ is a bijection. We consider the case $m = n$. Recall that n is even.

By *n-permutation* we mean a permutation of elements $1, 2, \ldots, n$.

A (0,1)-sequence $p_0 = 00X'$ is a legal sequence that corresponds to some n-permutation avoiding **7** if and only if $p_0' = X'$ is a legal sequence that corresponds to some $(n-2)$-permutation. To see this we observe that after the first two steps, p_0 fills in the two rightmost slots. We can strike them and forget about the first two steps of p_0; by this, we are left with the (0,1)-sequence X' that can be investigated (if it is a legal sequence) with respect to $(n-2)$-permutations.

By the same reasoning, a (0,1)-sequence $p_1 = 10X'$ is a legal sequence that corresponds to some n-permutation avoiding **7** if and only if $p_1' = X'$ is a legal sequence that corresponds to some $(n-2)$-permutation.

From these arguments we conclude, that if $X = 0X'$ then the correspondence $0X \leftrightarrow 1X$ is a bijection.

For any natural number k, we write (k) instead of writing k consecutive letters 1. In particular $(0) = \emptyset$.

Suppose $X = (k)0X'$ and $k \geq 1$. Reasoning as before, $p_0 = 0(k)0X'$ is a legal sequence with respect to n-permutations if and only if $p_0' = 0(k-1)X'$ is a legal sequence with respect to $(n-2)$-permutations. Also, $p_1 = 1(k)0X'$ is a legal sequence with respect to n-permutations if and only if $p_1' = 1(k-1)X'$ is a

legal sequence with respect to $(n-2)$-permutations. By induction, for $(n-2)$-permutations, the correspondence $0Y \leftrightarrow 1Y$ between legal sequences $0Y$ and $1Y$ is a bijection, thus the correspondence $0X \leftrightarrow 1X$, when $X = (k)0X'$, is a bijection for n-permutations as well.

The last thing we need to observe is that since n is even, $p_0 = 0(k)$ is a legal sequence if and only if $p_1 = 1(k)$ is a legal sequence.

This proves that the correspondence $0X \leftrightarrow 1X$ is a bijection.

Suppose n is odd. If a legal sequence begins with 0, then we obviously have that there are $A(n-1) = \binom{n-1}{(n-1)/2}$ such legal sequences. So to prove the statement we need to prove that the number of legal sequences that begin with 1 is equal to $\binom{n-1}{(n+1)/2}$ because if it is so then we have

$$A(n) = \binom{n-1}{(n-1)/2} + \binom{n-1}{(n+1)/2} = \binom{n}{(n-1)/2} = \binom{n}{\lfloor n/2 \rfloor}.$$

If a legal sequence begins with 1 then either

i) the number of 1s always exceeds the number of 0s, or

ii) at some point the number of 1s is equal to the number of 0s.

Let us consider case **i)**. Here we deal with Catalan numbers, which, among many other things, count the *Dyck paths*. A Dyck path of length $2n$ is a lattice path from $(0,0)$ to $(2n,0)$ with steps $(1,1)$ and $(1,-1)$ that never goes below the x-axis. Let us explain why in case **i)** we have $\frac{1}{(n-1)/2}\binom{n-3}{(n-3)/2}$ legal sequences with the right properties.

We can see that the number of ones is fixed in this case and equal to $(n-1)/2$. We can complete our (0,1)-sequence with 0s if necessary (in order to complete a Dyck path that corresponds to the (0,1)-sequence under consideration). Moreover, we can forget about the leftmost letter 1 because we know that it is followed by another letter 1, so we have $(n-3)/2$ ones. We thus substitute $k = (n-3)/2$ in the formula for the Catalan numbers, $C_k = \frac{1}{k+1}\binom{2k}{k}$, which completes the consideration of **i)**.

In case **ii)** we apply induction. Let us consider the first time, say step i, when the number of 0s is equal to the number of 1s. Obviously it can occur at any even step (and not at any odd one). Moreover, because it is the first such time, if we consider initial subsequences of length less then i, we always have that in such subsequences the number of 1s exceeds the number of 0s. So in case **ii)**, if we apply the induction hypothesis to the $A(n-i)$, the number of legal sequences is equal to

$$\sum_{\substack{i=2 \\ i \text{ is even}}}^{n-3} \frac{1}{i/2}\binom{i-2}{(i-2)/2}A(n-i) = \sum_{\substack{i=2 \\ i \text{ is even}}}^{n-3} \frac{1}{i/2}\binom{i-2}{(i-2)/2}\binom{n-i}{(n-i-1)/2}.$$

So to complete the case when n is odd we need only check the following equality:

$$\binom{n-1}{(n+1)/2} = \sum_{\substack{i=2 \\ i \text{ is even}}}^{n-3} \frac{1}{i/2}\binom{i-2}{(i-2)/2}\binom{n-i}{(n-i-1)/2} + \frac{1}{(n-1)/2}\binom{n-3}{(n-3)/2}.$$

The last term can be moved inside the sum. Since n is odd, we have $n = 2m+1$ and the equation above can be rewritten as

$$\binom{2m}{m+1} = \sum_{i=1}^{m} \frac{1}{i}\binom{2(i-1)}{i-1}\binom{2(m-i)+1}{m-i}.$$

We give a combinatorial proof of this identity. We observe that the left hand side of it counts the number of all lattice paths from $(0,0)$ to $(2m,-2)$ with steps $(1,1)$ and $(1,-1)$.

The i-th term in the right hand side counts the number of such paths whose first step below the x-axis is just after step $2(i-1)$. Now the first $2(i-1)$ steps of any such path determine a Dyck path of length $2(i-1)$. So there are $\binom{2(i-1)}{i-1}/i$ possibilities for a such path to pass the point $(2(i-1),0)$ and come to the point $(2i-1,-1)$ with the $(1,-1)$ step. We multiply this number with $\binom{2(m-i)+1}{m-i}$ which counts the number of all lattice paths from $(2i-1,-1)$ to $(2m,-2)$ with steps $(1,1)$ and $(1,-1)$. Thus, the right hand side counts the same paths as the left hand side.

This completes the case when n is odd and thereby the proof. □

Example 2. For $n = 4$ there are indeed $\binom{4}{2} = 6$ permutations avoiding class **7**. In the table below we show these permutations and legal sequences that correspond to them.

Permutation	**Corresponding legal sequence**
4321	0000
3421	001
4231	01
4312	100
3412	101
2413	11

Theorem 2. *For class* **6** *we have*

$$A(n) = \begin{cases} 2C_k, & \text{if } n = 2k+1, \\ C_k + C_{k-1}, & \text{if } n = 2k, \end{cases}$$

where C_k is the k-th Catalan number.

Proof. We consider n empty slots. If we fill the slots successively with the letters $1, 2, \ldots, n$ then we always have one or two possibilities, namely, either

0) we place the current number in the rightmost empty slot, or

1) we place it in the second empty slot left of the leftmost non-empty slot.

Observe that we can use option 0), except in the first step, only if there is a non-empty slot to the left of the rightmost empty slot. This is a crucial difference between classes **6** and **7**.

As in the proof of Theorem 1 we can consider the critical slots as well as (0,1)-sequences that appear in the obvious way (we have always one or two possibilities until we reach a critical slot and uniquely place all remaining numbers). After that we can associate the (0,1)-sequences with Dyck paths and apply the formula for the number of Dyck paths.

The number of legal sequences that correspond to the permutations avoiding class **6**, whose rightmost letter is 1, is equal to

$$\frac{1}{\lfloor (n-1)/2 \rfloor + 1} \binom{2 \cdot \lfloor (n-1)/2 \rfloor}{\lfloor (n-1)/2 \rfloor}.$$

The number of legal sequences that correspond to the permutations avoiding class **6**, with the second letter from the right equals 1, is equal to

$$\frac{1}{\lfloor n/2 \rfloor + 1} \binom{2 \cdot \lfloor n/2 \rfloor}{\lfloor n/2 \rfloor}.$$

From these facts we have that

$$A(n) = \frac{1}{\lfloor n/2 \rfloor + 1} \binom{2 \cdot \lfloor n/2 \rfloor}{\lfloor n/2 \rfloor} + \frac{1}{\lfloor (n-1)/2 \rfloor + 1} \binom{2 \cdot \lfloor (n-1)/2 \rfloor}{\lfloor (n-1)/2 \rfloor}.$$

Substituting n by $2k+1$ and $2k$, respectively, completes the proof. □

For class **8**, 1 must be either in the rightmost position or in the second position from the right. It is easy to see that the letters to the left of 1 must be in decreasing order. So there are n ways to choose the rightmost element of a permutation and all other elements can be placed uniquely, so there are n permutations avoiding **8**.

For class **9**, if 1 is in the rightmost position then we must place all other letters in decreasing order, so in this case we have the permutation $\pi = n(n-1)\ldots 21$ that avoids class **9**.

Assume that 1 is not in the rightmost position. The letters to the left of 1 must be in decreasing order. On the other hand it is easy to see that the letters to the right of 1 must be in increasing order (the set of such elements is non-empty). But 2 can not be to the left of 1 since in this case we obviously have an occurrence of a 213-pattern in the permutation that involves the letters 1 and

2. So 2 is immediately right of 1. Thus, to determine a permutation in class **9** is equivalent to partitioning the letters $\{3,4,\ldots,n\}$ into two blocks. There are 2^{n-2} ways of doing it. One of the blocks is all elements of a permutation to the right of 12, and the other one is all elements to the left of 12. So there are $1+2^{n-2}$ permutations avoiding class **9**.

Let us consider class **10**. We explain how to get a recurrence relation for $A(n)$ in this case.

It is easy to see that 1 is either in the rightmost position or in the second position from the right. In the first case there are $A(n-1)$ permutations that avoid **10**. In the second case we can place the letter 2 either in the position immediately left of 1 or in the second position left of 1.

In the first of these cases we choose from the remaining $(n-2)$ letters a candidate for the rightmost position. One can do this in $(n-2)$ ways. Then we multiply this by $A(n-3)$ since three of rightmost positions do not affect to placement of all other letters in a permutation.

So we need to consider the case when 2 is in the second position left of 1. In general, we need to consider the case when the letters $1,2,\ldots,i$ have been already placed in such way that $2i$ rightmost positions are alternatingly empty and non-empty, the rightmost position is empty, and these i letters are in decreasing order from the left to the right. If we place $(i+1)$ immediately left of the leftmost non-empty position then we choose i elements from the remaining $(n-i-1)$ elements in order to fill in i of rightmost empty positions. We observe that we must fill in the chosen elements in increasing order from the left to the right, otherwise we get an occurrence of a 312-pattern that is prohibited. Then we multiply this by $A(n-2i-1)$ because in this case the $(2i+1)$ rightmost letters do not affect the placement of the other letters in the permutation. So we need to consider the case when $(i+1)$ is in the second position left of i and so on.

So we have

$$A(n)=\sum_i \binom{n-i-1}{i} A(n-2i-1)+((n+1) \bmod 2).$$

The last term appears because if n is odd we have to consider the permutation

$$\pi=\frac{n+1}{2}\frac{n-1}{2}\frac{n+3}{2}\frac{n-3}{2}\ldots 2(n-1)1n,$$

which avoids **10** and which is not counted in the sum.

As initial conditions one can take $A(0)=1$, $A(1)=1$.

Theorem 3. *For class* **11** *we have* $A(n)=(n-1)!!+(n-2)!!$.

Proof. Since the patterns 123 and 321 can not occur in the permutations avoiding class **11**, such permutations are *alternating* or *reverse alternating*, that is, of the form $a_1>a_2<a_3>\cdots$ or $a_1<a_2>a_3<\cdots$, with one more restriction. One can easily see that 1 is either in the rightmost position or next to this

position, for otherwise we have an occurrence of a 123- or 132-pattern. If we go from the right to the left starting from 1 and jumping over one element then we get an increasing sequence of letters because otherwise we have an occurrence of the pattern 132.

Let $P_1(n)$ be the number of permutations having 1 in the rightmost position and let $P_2(n)$ be the number of permutations having 1 in the next to the rightmost position. Then obviously

$$A(n) = P_1(n) + P_2(n).$$

It is easy to see that

$$\begin{aligned} P_1(n) &= P_2(n-1), \\ P_2(n) &= (n-1)P_2(n-2) \end{aligned}$$

whence $P_1(n) = (n-2)!!$ and $P_2(n) = (n-1)!!$. □

Class **16** is a classically studied object. Permutations that avoid **16** are the alternating and the reverse alternating permutations. It is well known that the exponential generating function for the number of such permutations is $2(\tan x + \sec x)^2$. The initial values for $A(n)$ are $1, 2, 4, 10, 32, 122, 544, 2770, \ldots$

For the result on class **17** we refer the reader to Porism 10 in [5].

Finally, for class **18** we can observe that to the left of 1 in such a permutation we must have a decreasing subword and to the right of 1 we must have an increasing subword, since otherwise we have either a 132- or a 231-pattern. Thus we can choose the elements to the right of 1 from the set $\{2, 3, \ldots, n\}$ in 2^{n-1} ways and then arrange uniquely the right hand side and the left hand side (elements of a permutation to the left of 1). So there are 2^{n-1} permutations that avoid class **18**.

Bibliography

[1] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, Sém. Lothar. Combin. **44** (2000), Art. B44b, 18 pp.

[2] M. Bóna: Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps, J. Combin. Theory Ser. A **80** (1997), no. 2, 257–272.

[3] M. Bóna: Permutations avoiding certain patterns: the case of length 4 and some generalisations, Discrete Math. **175** (1997), no. 1–3, 55–67.

[4] M. Bóna: Permutations with one or two 132-subsequences, Discrete Math. **181** (1998), no. 1–3, 267–274.

[5] A. Claesson: Generalised Pattern Avoidance, European J. Combin. **22** (2001), no. 7, 961–971.

[6] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, Proceedings of FPSAC 2001.

[7] M. Klazar: Counting pattern-free set partitions. I. A generalisation of Stirling numbers of the second kind, European J. Combinatorics **21** (2000), no. 3, 367–378.

[8] J. Noonan, D. Zeilberger: The enumeration of permutations with a prescribed number of "forbidden" patterns, Adv. in Appl. Math. **17** (1996), no. 4, 381–407.

[9] N. J. A. Sloane and S. Plouffe. The Encyclopedia of Integer Sequences, Academic Press, (1995)
http://www.research.att.com/~njas/sequences/.

[10] R. P. Stanley: Enumerative Combinatorics, Vol. 1, Cambridge University Press, (1997).

[11] F. W. Schmidt, R. Simion: Restricted permutations, European J. Combin. **6** (1985), no. 4, 383–406.

Paper II

Generalized pattern avoidance with additional restrictions

Generalized Pattern Avoidance with Additional Restrictions

Sergey Kitaev

E-mail: kitaev@math.chalmers.se
Matematik, Chalmers tekniska högskola och Göteborgs universitet,
S-412 96 Göteborg, Sweden

Abstract

Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. We consider n-permutations that avoid the generalized pattern $1-32$ and whose k rightmost letters form an increasing subword. The number of such permutations is a linear combination of Bell numbers. We find a bijection between these permutations and all partitions of an $(n-1)$-element set with one subset marked that satisfy certain additional conditions. Also we find the e.g.f. for the number of permutations that avoid a generalized 3-pattern with no dashes and whose k leftmost or k rightmost letters form either an increasing or decreasing subword. Moreover, we find a bijection between n-permutations that avoid the pattern 132 and begin with the pattern 12 and increasing rooted trimmed trees with $n+1$ nodes.

2.1 Introduction and Background

All permutations in this paper are written as words $\pi = a_1 a_2 \cdots a_n$, where the a_i consist of all the integers $1, 2, \ldots, n$.

A *pattern* is a word on some alphabet of letters, where some of the letters may be separated by dashes. In our notation, the classical permutation patterns, first studied systematically by Simion and Schmidt [SchSim], are of the form $p = 1 - 3 - 2$, the dashes indicating that the letters in a permutation corresponding to an occurrence of p do not have to be adjacent. In the classical case, an occurrence of a pattern p in a permutation π is a subsequence in π (of the same length as the length of p) whose letters are in the same relative order as those in p. For example, the permutation 264153 has only one occurrence of the pattern $1-2-3$, namely the subsequence 245. Note that a classical pattern should, in our notation, have dashes at the beginning and end. Since most of the patterns considered in this paper satisfy this, we suppress these dashes from the notation.

In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns* (*GPs*) where two adjacent letters in a pattern may be required to be adjacent in the permutation. Such an adjacency requirement is indicated by the absence of a dash between the corresponding letters in the pattern. Thus, a pattern with no dashes corresponds to a contiguous subword anywhere in a

permutation. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, but the pattern 2-3-1 occurs also in the subwords 562 and 563. The motivation for introducing these patterns in [BabStein] was the study of Mahonian statistics.

A number of interesting results on GPs were obtained by Claesson [Claes]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were shown there. In [Kit1] the present author investigated simultaneous avoidance of two or more 3-letter GPs with no dashes. Also there is a number of works concerning GPs by Mansour (see for example [Mans1, Mans2]).

In this paper we consider avoidance some generalized 3-patterns with additional restrictions. The restrictions consist of demanding that a permutation begin or end with the pattern $12\dots k$ or the pattern $k(k-1)\dots 1$.

It turns out that the number of permutations that avoid the pattern $1-32$ and end with the pattern $12\dots k$ is a linear combination of the Bell numbers. The n-th Bell number is the number of ways a set of n elements can be partitioned into nonempty subsets. We find a bijection between these permutations and all partitions of an $(n-1)$-element set with one subset marked that satisfy certain special conditions. In particular, in Theorem 1, we investigate the case $k = 2$. We get that the total number of partitions of an $(n-1)$-element set with one part marked, is equal to the number of $(1-32)$-avoiding n-permutations that end with a 12-pattern. Lemma 1 gives us an identity involving the Bell numbers and the Stirling numbers of the second kind, which seems to be new. In Theorem 3 we prove that the number of 132-avoiding n-permutations that begin with the pattern 12 is equal to the number of increasing rooted trimmed trees with $n+1$ nodes.

In Sections $4-7$, we give a complete description (in terms of *exponential generating functions (e.g.f.)*) for the number of permutations that avoid a pattern of the form xyz and begin or end with the pattern $12\dots k$ or the pattern $k(k-1)\dots 1$. We record all the results concerning these e.g.f. in the table in Section 7. The case $k = 1$ is equivalent to the absence of the additional restriction. This case was considered in [ElizNoy] and [Kit2].

We observe that avoidance of some pattern with the additional restrictions described above, in fact is equivalent to simultaneous avoidance of several patterns. For example, beginning with the pattern 12 is equivalent to the avoidance of the pattern [21) in the Babson-Steingrímsson notation. Thus avoidance of the pattern 132 and beginning with the pattern 12 is equivalent to simultaneous avoidance of the patterns 132 and [21). Also, ending with the pattern 123 is equivalent to simultaneously avoiding the patterns (132], (213], (231], (312] and (321].

2.2 Set partitions and pattern avoidance

We recall some basic definitions.

A *partition* of a set S is a family, $\pi = \{A_1, A_2, \dots, A_k\}$, of pairwise disjoint

non-empty subsets of S such that $S = \cup_i A_i$. The total number of partitions of an n-element set is called a *Bell number* and is denoted B_n.

The *Stirling number of the second kind* $S(n,k)$ is the number of ways a set with n elements can be partitioned into k disjoint, non-empty subsets.

Proposition 1. *Let $P(n,k)$ be the number of n-permutations that avoid the pattern $1-32$ and end with the pattern $12\ldots k$. Then*

$$P(n,k) = \sum_{i=0}^{n-k} \binom{n-1}{i} B_i.$$

Proof. Suppose a permutation $\pi = \sigma 1 \tau$ avoids the pattern $1-32$ and ends with the pattern $12\ldots k$. The letters of τ must be in increasing order, since otherwise we have an occurrence of the pattern $1-32$ involving 1. Also, σ must avoid $1-32$. If $|\sigma| = i$ then obviously $0 \le i \le n-k$ and we can choose the letters of σ in $\binom{n-1}{i}$ ways. By [Claes, Proposition 5], the number of i-permutations that avoid the pattern $1-32$ is equal to B_i, hence there are B_i ways to form σ. □

Lemma 1. *We have* $\displaystyle\sum_{i=0}^{n-1} \binom{n}{i} B_i = \sum_{i=0}^{n} i \cdot S(n,i).$

Proof. The identity can be proved from the recurrences for $S(n,k)$ and B_n, but we give a combinatorial proof.

The left-hand side of the identity is the number of ways to choose i elements from an n-element set, and then to make all possible partitions of the chosen elements.

The right-hand side is the number of ways to partition a set with n elements into i disjoint non-empty subsets ($i = 1, 2, \ldots, n$) and mark one of the subsets. For example if $n = 4$ then $\overline{1} - 24 - 3$ and $1 - \overline{24} - 3$ are two different partitions, where the marked subset is overlined.

A bijective correspondence between these combinatorial interpretations is given by the following: For the left-hand side, after partitioning the i chosen elements, let the remaining $n-i$ elements form the marked subset in the partition. □

The formula for $P(n,k)$ in Proposition 1, applied to $k = 2$, and Lemma 1 now give the following theorem:

Theorem 1. *The total number of partitions of an $(n-1)$-element set with one part marked, is equal to the number of $(1-32)$-avoiding n-permutations that end with the pattern 12.*

We give now a direct combinatorial proof of this theorem.

Proof. Suppose $P = S_1 - S_2 - \cdots - S_k$ is a partition of an $(n-1)$-element set into k subsets with one marked subset and T_i is the word that consists of all elements of S_i in increasing order. We may, without loss of generality, assume that $\min(S_i) < \min(S_j)$ if $i > j$. In particular, $1 \in S_k$. There are two cases possible:

1) $S_k = \{1\}$ (S_k is not marked set);

2) Either $S_k = \overline{1}$ or $1 \in S_k$ and $|S_k| \geq 2$.

In the first case, to a partition $P = S_1 - S_2 - \cdots - \overline{S_i} - \cdots - S_{k-1} - 1$ we associate the permutation $\pi(P) = nT_1T_2 \ldots T_{i-1}T_{i+1} \ldots T_{k-1}1T_i$, which is $(1-32)$-avoiding and ends with the pattern 12 since $S_i \neq \emptyset$. For example $4 - \overline{23} - 1 \mapsto 54123$.

In the second case we adjoin n to a marked subset, and then consider the permutation $\pi(P) = T_1T_2 \ldots T_k$. This permutation is obviously $(1-32)$-avoiding since $\min(S_i) < \min(S_j)$ if $i > j$ and the letters in T_i are in increasing order. Also it ends with the pattern 12. For example $5 - \overline{34} - 12 \mapsto 534612$, and $5 - 234 - \overline{1} \mapsto 523416$.

Obviously in both cases we have an injection.

Now it is easy to see that the correspondence above is a surjection as well. Indeed, for any $(1-32)$-avoiding permutation π that ends with the pattern 12, we can check if π begins with n or not and according to this we have either case 1) or 2). In the first case, we remove n, then read π from left to right and consider all maximal increasing intervals. The elements of each such interval correspond to some subset, and we let all the letters to the right of 1 constitute the marked subset. In the second case, we divide π into maximal increasing intervals, and let the letters of each interval correspond to a subset. Then we let the interval containing n be the marked subset. Thus we have a surjection. So the correspondence is a bijection and the theorem is proved. □

The following theorem generalizes Theorem 1.

Theorem 2. *Let $P = S_1 - S_2 - \cdots - S_\ell$ be a partition of $\{1, 2, \ldots, n-1\}$ into ℓ subsets with subset S_i marked. We assume also that $1 \in S_\ell$. Then $P(n,k)$ counts all possible marked partitions of $\{1, 2, \ldots, n-1\}$ that satisfy the following conditions:*

1) if $i = \ell$ (the last subset is marked) then $|S_\ell| \geq k-1$;

2) if $i \neq \ell$ and $|S_\ell| \neq 1$ then $|S_\ell| \geq k$;

3) if $i \neq \ell$ and $|S_\ell| = 1$ then $|S_i| \geq k-1$.

Proof. A proof of this theorem is similar to the proof of Theorem 1. We assume that $\min(S_i) < \min(S_j)$ for $i > j$ and consider three cases.

If a partition satisfies 1), that is $P = S_1 - S_2 - \cdots - \overline{S_\ell}$ and $|S_\ell| \geq k-1$, then adjoining n to S_ℓ guarantees that the permutation $\pi(P) = T_1T_2 \ldots T_\ell$, which is $(1-32)$-avoiding, ends with k letters in increasing order.

In case 2), we adjoin n to the marked subset and consider $\pi(P) = T_1T_2 \ldots T_\ell$. This permutation avoids the pattern $1-32$ and ends with the pattern $12 \ldots k$ since $|S_\ell| \geq k$.

In case 3), to a partition $P = S_1 - S_2 - \cdots - \overline{S_i} - \cdots - S_{k-1} - 1$ we associate the permutation $\pi(P) = nT_1T_2 \ldots T_{i-1}T_{i+1} \ldots T_{k-1}1T_i$, which is $(1-32)$-avoiding and ends with at least k letters in increasing order since $|S_i| \geq k-1$.

That this correspondence is a bijection can be shown in a way similar to the proof of Theorem 1. □

2.3 Increasing rooted trimmed trees and pattern avoidance

In an *increasing rooted tree*, nodes are numbered and the numbers increase as we move away from the root. A *trimmed tree* is a tree where no node has a single leaf as a child (every leaf has a sibling).

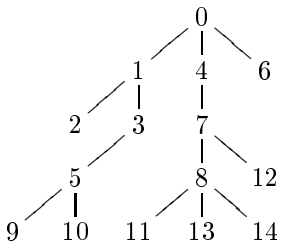
Theorem 3. *Let A_n denote the set of all n-permutations that avoid the pattern* 132 *and begin with the pattern* 12. *The number of permutations in A_n is equal to the number of increasing rooted trimmed trees (IRTTs) with $n+1$ nodes.*

Proof. A *right-to-left minimum* of a permutation π is an element a_i such that $a_i < a_j$ for every $j > i$.

We describe a bijective correspondence F between the permutations in A_n and IRTTs with $n+1$ nodes.

Suppose $\pi \in A_n$ and $\pi = P_0 a_0 P_1 a_1 \dots P_k a_k$, where a_i are the right-to-left minima of π and P_j are (possibly empty) subwords of π. We construct a IRTT $T = F(\pi)$ with $n+1$ nodes as follows. The root of T is labelled by 0 and $a_0, a_1, \dots, a_k$ are the labels of the root's children if we read them from left to right. Then we let the right-to-left minima of P_i be the labels of the children of a_i and so on. It is easy to see that, since π avoids 132 and begins with 12, T avoids limbs of length 2. Also, T is an increasing rooted tree and hence T is a IRTT. For instance,

$F(2\ 9\ 10\ 5\ 3\ 1\ 11\ 13\ 14\ 8\ 12\ 7\ 4\ 6) =$



Obviously, the correspondence F is an injection.

To see, that F is a surjection, we show how to construct the permutation $\pi \in A_n$ that corresponds to a given IRTT T. The main rule is the following: If a_i and a_j are siblings, and $a_i < a_j$, then the labels of the nodes of the subtree below a_j, are all the letters in π between a_i and a_j, that is, $a_{i+1}, a_{i+2}, \dots, a_{j-1}$. If a_i is a single child, then the labels of the nodes of the subtree below a_i appear immediately left of a_i in π. That is, if there are k nodes in the subtree below a_i then the k corresponding labels form the subword $a_{i-k} a_{i-k+1} \dots a_{i-1}$. We now start from the first level of T, which consists of the root's children, and apply this rule. After that we consider the second level and so on. The fact that T is

a IRTT ensures that π avoids the pattern 132 and begins with the pattern 12. Thus, F is a bijection. □

2.4 Avoiding 132 and beginning with $12\dots k$ or $k(k-1)\dots 1$

Let $E_q^p(x)$ denote the e.g.f. for the number of permutations that avoid the pattern q and begin with the pattern p.

If $k = 1$, then there is no additional restriction, that is, we are dealing with avoidance of the pattern 132 (no dashes) and thus

$$E_{132}^1(x) = \frac{1}{1 - \int_0^x e^{-t^2/2}\, dt}, \tag{2.1}$$

since this result is a special case of [ElizNoy, Theorem 4.1] and [Kit2, Theorem 12].

Theorem 4. *We have*

$$E_{132}^{12}(x) = \frac{e^{-x^2/2}}{1 - \int_0^x e^{-t^2/2}\, dt} - x - 1,$$

and for $k \geq 3$

$$E_{132}^{12\dots k}(x) = E_{132}^1(x) \int_0^x \int_0^{t_{k-2}} \cdots \int_0^{t_2} \left(e^{-t_1^2/2} - \frac{t_1 + 1}{E_{132}^1(t_1)} \right) dt_1 dt_2 \cdots dt_{k-2}.$$

Proof. Let $E_{n,k}$ denote the number of n-permutations that avoid the pattern 132 and begin with an increasing subword of length $k > 0$. Let π be such a permutation of length $n+1$. Also, suppose $k \neq 2$. If $\pi = \sigma 1 \tau$ then either $\sigma = \epsilon$ or $\sigma \neq \epsilon$ where ϵ denotes the empty word. If $\sigma = \epsilon$ then τ must avoid 132 and begin with an increasing subword of length $k-1$. Otherwise σ must avoid 132 and begin with an increasing subword of length k, whereas τ must begin with the pattern 12, or be a single letter (there are n ways to choose this letter), or be ϵ. This leads to the following:

$$E_{n+1,k} = E_{n,k-1} + \sum_{i \geq 0} \binom{n}{i} E_{i,k} E_{n-i,2} + n E_{n-1,k} + E_{n,k}. \tag{2.2}$$

Multiplying both sides of the equality with $x^n/n!$ and summing over all n we get the following differential equation

$$\frac{d}{dx} E_{132}^{12\dots k}(x) = (E_{132}^{12}(x) + x + 1) E_{132}^{12\dots k}(x) + E_{132}^{12\dots(k-1)}(x), \tag{2.3}$$

with the initial conditions $E_{132}^{12\dots k}(0) = 0$ for $k \geq 3$.

Observe that equality (2.3) is not valid for $k = 2$. Indeed, if $k = 2$, then it is incorrect to add the term $E_{n,k-1} = E_{n,1}$ in (2.2), since this term counts

the number of permutations $\pi = 1\tau$ with the only restriction for τ that it must avoid 132. The absence of an additional restriction for τ means that the 3 leftmost letters of π could form the pattern 132. However, we can use (2.3) to find $E_{132}^{12}(x)$ by letting k equal 1. In this case we have

$$\frac{d}{dx}E_{132}^{1}(x) = (E_{132}^{12}(x) + x + 1)E_{132}^{1}(x),$$

which gives

$$E_{132}^{12}(x) = \frac{e^{-x^2/2}}{1 - \int_0^x e^{-t^2/2}\, dt} - x - 1. \tag{2.4}$$

For the case $k \geq 3$, it is convenient to write $E_{132}^{12}(x)$ in the form

$$E_{132}^{12}(x) = B'(x) - x - 1,$$

where $B(x) = -\ln(1 - \int_0^x e^{-t^2/2}\, dt)$ and thus $B'(x) = \exp(B(x) - \frac{x^2}{2})$. So (2.3) is equivalent to the differential equation

$$\frac{d}{dx}E_{132}^{12\ldots k}(x) = B'(x)E_{132}^{12\ldots k}(x) + E_{132}^{12\ldots(k-1)}(x)$$

which has the solution

$$E_{132}^{12\ldots k}(x) = e^{B(x)} \int_0^x e^{-B(t)} E_{132}^{12\ldots(k-1)}(t)\, dt =$$

$$E_{132}^{1}(x) \int_0^x \frac{E_{132}^{12\ldots(k-1)}(t)}{E_{132}^{1}(t)} dt =$$

$$E_{132}^{1}(x) \int_0^x \int_0^{t_2} \frac{E_{132}^{12\ldots(k-2)}(t_1)}{E_{132}^{1}(t_1)} dt_1 dt_2 =$$

$$E_{132}^{1}(x) \int_0^x \int_0^{t_{k-2}} \cdots \int_0^{t_2} \frac{E_{132}^{12}(t_1)}{E_{132}^{1}(t_1)} dt_1 dt_2 \cdots dt_{k-2}.$$

Using (2.1) and (2.4) we now get the desired result. □

Using the formula for $E_{132}^{12\ldots k}(x)$ in Theorem 4 one can derive, in particular, that

$$E_{132}^{123}(x) = -\frac{1}{2} - x - \frac{x^2}{2} + \frac{\left(1 + \frac{x}{2}\right) e^{-x^2/2} - \frac{1}{2}}{1 - \int_0^x e^{-t^2/2}\, dt}.$$

Theorem 5. *For $k \geq 2$*

$$E_{132}^{k(k-1)\cdots 1}(x) = \frac{E_{132}^{1}(x)}{(k-1)!}\int_0^x t^{k-1}e^{-t^2/2}\ dt.$$

Proof. We proceed as in the proof of Theorem 4.

Let $R_{n,k}$ denote the number of n-permutations that avoid the pattern 132 and begin with a decreasing subword of length $k > 1$ and let π be such a permutation of length $n+1$. Suppose also that $\pi = \sigma 1 \tau$. If $\tau = \epsilon$ then, obviously, there are $R_{n,k}$ ways to choose σ. If $|\tau| = 1$, that is, 1 is in the second position from the right in π, then there are n ways to choose the rightmost letter in π and we multiply this by $R_{k,n-1}$, which is the number of ways to choose σ. If $|\tau| > 1$ then τ must begin with the pattern 12, otherwise the letter 1 and the two leftmost letters of τ form the pattern 132, which is forbidden. So, in this case there are $\sum_{i\geq 0}\binom{n}{i}R_{i,k}E_{n-i,2}$ such permutations with the right properties, where i indicates the length of σ and $E_{n-i,2}$ is defined in the proof of Theorem 4. In the last formula, of course, $R_{i,k} = 0$ if $i < k$. Finally we have to consider the situation when 1 is in the k-th position. In this case we can choose the letters of σ in $\binom{n}{k-1}$ ways, write them in decreasing order and then choose τ in $E_{n-k+1,2}$ ways. Thus

$$R_{n+1,k} = R_{n,k} + nR_{n-1,k} + \sum_{i\geq 0}\binom{n}{i}R_{i,k}E_{n-i,2} + \binom{n}{k-1}E_{n-k+1,2}. \qquad (2.5)$$

We observe that (2.5) is not valid for $n = k-1$ and $n = k$. Indeed, if 1 is in the k-th position in these cases, the term $\binom{n}{k-1}E_{n-k+1,2}$, which counts the number of such permutations, is zero, whereas there is one "good" $(n+1)$-permutation in the case $n = k-1$ and n "good" $(n+1)$-permutations in case $n = k$. Multiplying both sides of the equality with $x^n/n!$, summing over n and using the observation above (which gives the term $x^{k-1}/(k-1)! + kx^k/k!$ in the right-hand side of Equalion (2.6)), we get

$$\frac{d}{dx}E_{132}^{k(k-1)\cdots 1}(x) = (E_{132}^{12}(x) + x + 1)\left(E_{132}^{k(k-1)\cdots 1}(x) + \frac{x^{k-1}}{(k-1)!}\right), \qquad (2.6)$$

with the initial condition $E_{k(k-1)\cdots 1}^{132}(0) = 0$. We solve the equation in the way proposed in Theorem 4 and get

$$E_{132}^{k(k-1)\cdots 1}(x) = \frac{E_{132}^{1}(x)}{(k-1)!}\int_0^x \frac{(E_{132}^{12}(t)+t+1)t^{k-1}}{E_{132}^{1}(t)}\,dt = \frac{E_{132}^{1}(x)}{(k-1)!}\int_0^x t^{k-1}e^{-t^2/2}\,dt.$$

□

For instance,

$$E_{132}^{21}(x) = \frac{1-e^{-x^2/2}}{1-\int_0^x e^{-t^2/2}\ dt} \quad \text{and} \quad E_{132}^{321}(x) = \frac{1}{2}\left(-1 + \frac{1-xe^{-x^2/2}}{1-\int_0^x e^{-t^2/2}\ dt}\right).$$

Moreover, the integral $\int_0^x t^{k-1}e^{-t^2/2}\,dt$ from the formula for $E_{132}^{k(k-1)\dots1}(x)$ can be solved to show that $E_{132}^{k(k-1)\dots1}(x)$ equals

$$\frac{(k/2-1)!2^{k/2-1}}{(k-1)!(1-\sqrt{\frac{\pi}{2}}\operatorname{erf}(x))}\left(1-e^{-x^2/2}\sum_{i=0}^{k/2-1}\frac{x^{2i}}{2^i i!}\right),$$

if k is even, and

$$\frac{1}{(k-1)!!}\left(-1+\frac{1}{1-\sqrt{\frac{\pi}{2}}\operatorname{erf}(x)}\left(1-e^{-x^2/2}\sum_{i=0}^{(k-3)/2}\frac{x^{2i+1}}{(2i+1)!!}\right)\right)$$

if k is odd.

In the formula above, $\operatorname{erf}(x)$ is the *error function*:

$$\operatorname{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt.$$

2.5 Avoiding 123 and beginning with $k(k-1)\dots1$ or $12\dots k$

If $k=1$, we have no additional restrictions and, according to [ElizNoy, Theorem 4.1],

$$E_{123}^{1}(x)=\frac{\sqrt{3}}{2}\frac{e^{x/2}}{\cos\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)}.$$

Theorem 6. *For $k\geq 2$*

$$E_{123}^{k(k-1)\dots1}(x)=\frac{e^{x/2}}{(k-1)!\cos\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)}\int_0^x e^{-t/2}t^{k-1}\sin\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{3}\right)\,dt.$$

In particular,

$$E_{123}^{21}(x)=\frac{\sqrt{3}}{2}\tan\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)-x-\frac{1}{2}.$$

Proof. Let $P_{n,k}$ denote the number of n-permutations that avoid the pattern 123 and begin with a decreasing subword of length k. We observe that we can use arguments similar to the proof of Theorem 5 to get the recurrence formula for $P_{n,k}$. Indeed, we only need to write the letter P instead of R and E in (2.5):

$$P_{n+1,k}=P_{n,k}+nP_{n-1,k}+\sum_{i\geq 0}\binom{n}{i}P_{i,k}P_{n-i,2}+\binom{n}{k-1}P_{n-k+1,2}. \qquad (2.7)$$

This formula is valid for $k > 1$. Multiplying both sides of the equality with $x^n/n!$, summing over n and reasoning as in the proof of Theorem 5, we get:

$$\frac{d}{dx}E_{123}^{k(k-1)\ldots1}(x) = (E_{123}^{21}(x) + x + 1)\left(E_{123}^{k(k-1)\ldots1}(x) + \frac{x^{k-1}}{(k-1)!}\right), \qquad (2.8)$$

with the initial condition $E_{123}^{k(k-1)\ldots1}(0) = 0$. To solve (2.8), we need to know $E_{123}^{21}(x)$. To find it, we consider the case $k = 1$. In this case we have almost the same recurrence as we have in (2.7), but we must remove the last term in the right-hand side:

$$P_{n+1,1} = P_{n,1} + nP_{n-1,1} + \sum_{i\geq 0}\binom{n}{i}P_{i,k}P_{n-i,2}.$$

After multiplying both sides of the last equality with $x^n/n!$ and summing over n, we have

$$\frac{d}{dx}E_{123}^{1}(x) = (E_{123}^{21}(x) + x + 1)E_{123}^{1}(x),$$

and thus

$$E_{123}^{21}(x) = \frac{\frac{d}{dx}E_{123}^{1}(x)}{P_1(x)} - x - 1 = \frac{\sqrt{3}}{2}\tan\left(\frac{\sqrt{3}}{2}x + \frac{\pi}{6}\right) - x - \frac{1}{2}.$$

Now we solve (2.8) in the way we solved Equalion (2.6) and get

$$E_{123}^{k(k-1)\ldots1}(x) = \frac{e^{x/2}}{(k-1)!\cos\left(\frac{\sqrt{3}}{2}x + \frac{\pi}{6}\right)}\int_0^x e^{-t/2}t^{k-1}\sin\left(\frac{\sqrt{3}}{2}t + \frac{\pi}{3}\right)\,dt.$$

□

The following theorem is straightforward to prove.

Theorem 7. *We have* $E_{123}^{12\ldots k}(x) = 0$ *for* $k \geq 3$ *and*

$$E_{123}^{12}(x) = E_{123}^{1}(x) - E_{123}^{21}(x) = \frac{\sqrt{3}}{2}\frac{e^{x/2}}{\cos\left(\frac{\sqrt{3}}{2}x + \frac{\pi}{6}\right)} - \frac{1}{2} - \frac{\sqrt{3}}{2}\tan\left(\frac{\sqrt{3}}{2}x + \frac{\pi}{6}\right).$$

2.6 Avoiding 213 and beginning with $k(k-1)\ldots1$ or $12\ldots k$

If $k = 1$, then by [ElizNoy, Theorem 4.1] or [Kit2, Theorem 12]

$$E_{213}^{1}(x) = \frac{1}{1 - \int_0^x e^{-t^2/2}\,dt}.$$

Theorem 8. *For $k \geq 2$*

$$E_{213}^{12\ldots k}(x) = \int_0^x \int_0^t \frac{s^{k-2} e^{T(t)-T(s)}}{(k-2)!(1-\int_0^t e^{-m^2/2}dm)}\, dsdt,$$

where $T(x) = -x^2/2 + \int_0^x \frac{e^{-t^2/2}}{1-\int_0^t e^{-s^2/2}ds}\, dt$.

Proof. Let A_n denote the number of n-permutations that avoid the pattern 213 and let B_n denote the number of n-permutations that avoid 213 and begin with the pattern $12\ldots k$. Let C_n denote the number of n-permutation that avoid 213, begin with the pattern $12\ldots k$ and end with the pattern 12 and let D_n denote the number of n-permutations that avoid 213 and end with the pattern 12. Also, let $A(x)$, $B(x)$, $C(x)$ and $D(x)$ denote the e.g.f. for the numbers A_n, B_n, C_n and D_n respectively.

We observe, that

$$D(x) = E_{132}^{12}(x) = e^{-x^2/2}/(1-\int_0^x e^{-t^2/2}dt) - x - 1,$$

since, by using the reverse and complement discussed in the next section, there are as many permutations that avoid the pattern 213 and end with the pattern 12 as those that avoid the pattern 132 and begin with the pattern 12. Also, $A(x) = E_{312}^{1}(x)$ and $B(x) = E_{213}^{12\ldots k}(x)$.

Suppose now that $\pi = \sigma(n+1)\tau$ is an $(n+1)$-permutation that avoids the pattern 213 and begins with the pattern $12\ldots k$. So σ must avoid 213, begin with $12\ldots k$, but also end with the pattern 12 since otherwise the two rightmost letters of σ together with the letter $(n+1)$ form the pattern 213, which is forbidden. For τ, there is only one restriction — avoidance of 213. So if $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways, which gives $\sum_{i\geq 0}\binom{n}{i}C_iA_i$ permutations that avoid the pattern 213 and begin with the pattern $12\ldots k$. Moreover, it is possible for $(n+1)$ to be in the kth position, in which case we choose the letters of σ in $\binom{n}{k-1}$ ways and arrange them in increasing order. Thus

$$B_{n+1} = \sum_{i\geq 0}\binom{n}{i}C_iA_{n-i} + \binom{n}{k-1}A_{n-(k-1)}.$$

Multiplying both sides of this equality with $x^n/n!$ and summing over n, we get

$$B'(x) = \left(C(x) + \frac{x^{k-1}}{(k-1)!}\right)A(x), \tag{2.9}$$

with the initial condition $B(0) = 0$.

To solve (2.9) we need to find $C(x)$. Let $\pi = \sigma(n+1)\tau$ be an $(n+1)$-permutation that avoids the pattern 213, begins with the pattern $12\ldots k$ and ends with the pattern 12. Reasoning as above, σ must avoid the pattern 213, begin with the pattern $12\ldots k$ and end with the pattern 12, whereas τ must

avoid 213 and end with the pattern 12. This gives $\sum_{i\geq 0}\binom{n}{i}C_iD_{n-i}$ permutations counted by C_{n+1}. Also, the letter $(n+1)$ can be in the kth position, which gives $\binom{n}{k-1}D_{n-(k-1)}$ permutations, and this letter can be in the $(n+1)$st position, which gives C_n permutations that avoid the pattern 213, begin with the pattern $12\ldots k$ and end with the pattern 12. Also, if $n+1=k$ and all the letters are arranged in increasing order, then $(n+1)$ is in the $(n+1)$st position, but this permutation is not counted by C_n above. So

$$C_{n+1}=\sum_{i\geq 0}\binom{n}{i}C_iD_{n-i}+\binom{n}{k-1}D_{n-(k-1)}+C_n+\delta_{n,k-1},$$

where $\delta_{n,k}$ is the Kronecker delta, that is,

$$\delta_{n,k}=\begin{cases}1, & \text{if } n=k,\\ 0, & \text{else.}\end{cases}$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get

$$C'(x)=(D(x)+1)C(x)+(D(x)+1)\frac{x^{k-1}}{(k-1)!}. \tag{2.10}$$

To solve (2.10), it is convenient to introduce the function $T(x)$ such that $T'(x)=D(x)+1$. Thus

$$T(x)=x+\int_0^x D(t)dt=-x^2/2+\int_0^x\frac{e^{-t^2/2}}{1-\int_0^t e^{-s^2/2}ds}\,dt,$$

and we need to solve the equation

$$C'(x)=T'(x)C(x)+T'(x)\frac{x^{k-1}}{(k-1)!},$$

with $C(0)=0$.

The solution to this equation is given by

$$C(x)=e^{T(x)}\int_0^x e^{-T(t)}T'(t)\frac{t^{k-1}}{(k-1)!}dt=-\frac{x^{k-1}}{(k-1)!}+e^{T(x)}\int_0^x e^{-T(t)}\frac{t^{k-2}}{(k-2)!}dt.$$

Now we substitute $C(x)$ into (2.9) to get the desired result. □

Theorem 9. *For $k\geq 2$*

$$E_{213}^{k(k-1)\ldots 1}(x)=-\frac{x^{k-1}}{(k-1)!}+\sum_{n=0}^{k-2}\int_0^x\int_0^{t_n}\cdots\int_0^{t_1}\frac{C_{k-n}(t)+\delta_{n,k-2}}{1-\int_0^t e^{-m^2/2}dm}dtdt_1\cdots dt_n,$$

where

$$C_k(x)=e^{T(x)}\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_1}e^{-T(t)}\left(\frac{e^{-t^2/2}}{1-\int_0^t e^{-m^2/2}dm}-t-1\right)dtdt_1\cdots dt_{k-2},$$

with $T(x)=-x^2/2+\int_0^x\frac{e^{-t^2/2}}{1-\int_0^t e^{-s^2/2}ds}\,dt$.

Proof. Let A_n denote the number of n-permutations that avoid the pattern 213 and let $B_{n,k}$ denote the number of n-permutations that avoid 213 and begin with the pattern $k(k-1)\dots 1$ for $k \geq 2$. Let $C_{n,k}$ denote the number of n-permutation that avoid 213, begin with $k(k-1)\dots 1$ for $k \geq 2$ and end with the pattern 12 and let D_n denote the number of n-permutations that avoid 213 and end with the pattern 12. Also, let $A(x)$, $B_k(x)$, $C_k(x)$ and $D(x)$ denote the e.g.f. for the numbers A_n, $B_{n,k}$, $C_{n,k}$ and D_n respectively. In the proof of Theorem 8 it was shown that $D(x) = e^{-x^2/2}/(1 - \int_0^x e^{-t^2/2}dt) - x - 1$ and $A(x) = E_{312}^1(x)$. Moreover, $B_k(x) = E_{213}^{k(k-1)\dots 1}(x)$.

Suppose now that $\pi = \sigma(n+1)\tau$ is an $(n+1)$-permutation that avoids 213 and begins with the pattern $k(k-1)\dots 1$. So σ must avoid 213, begin with $k(k-1)\dots 1$, but also end with the pattern 12 if $|\sigma| \geq 2$, since otherwise the two rightmost letters of σ together with the letter $(n+1)$ form the pattern 213 which is forbidden. For τ, there is only one restriction - avoidance of 213. So if $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways, which gives $\sum_{i\geq 0}\binom{n}{i}C_{i,k}A_i$ permutations counted by $B_{n+1,k}$. Also, it is possible for $(n+1)$ to be the leftmost letter, in which case the remaining letters must form a n-permutation that avoids 213 and begins with the pattern $(k-1)(k-2)\dots 1$. Thus

$$B_{n+1,k} = \sum_{i\geq 0}\binom{n}{i}C_{i,k}A_{n-i} + B_{n,k-1}. \tag{2.11}$$

However, this formula is not valid when $k = 2$ and $n = 0$. Indeed, since $B_{0,1} = A_0 = 1$, it follows from the formula that $B_{1,2} = 1$, which is not true, since $B_{1,2}$ must be 0. So, in the right-hand side of (2.11), we need to subtract the term

$$\gamma_{n,k} = \begin{cases} 1, & \text{if } n = 0 \text{ and } k = 2, \\ 0, & \text{else.} \end{cases}$$

Multiplying both sides of the obtained equality by $x^n/n!$ and summing over n, we get, that for $k \geq 3$

$$\frac{d}{dx}B_k(x) = C_k(x)A(x) + B_{k-1}(x), \tag{2.12}$$

with the initial condition $B_k(0) = 0$, and

$$\frac{d}{dx}B_2(x) = C_2(x)A(x) + B_1(x) - 1, \tag{2.13}$$

with the initial condition $B_2(0) = 0$.

The solution to differential equations (2.12) and (2.13) is given by

$$B_k(x) = -\frac{x^{k-1}}{(k-1)!} + \sum_{n=0}^{k-2}\int_0^x\int_0^{t_n}\cdots\int_0^{t_1}\frac{C_{k-n}(t)+\delta_{n,k-2}}{1-\int_0^t e^{-m^2/2}dm}dtdt_1\cdots dt_n.$$

So, to prove the theorem, we only need to find $C_k(x)$.

Suppose $\pi = \sigma(n+1)\tau$ be an $(n+1)$-permutation that avoids the pattern 213, begins with the pattern $k(k-1)\dots 1$ and ends with the pattern 12. It is clear that σ must avoid 213, begin with the pattern $k(k-1)\dots 1$ and end with the pattern 12, whereas τ must avoid 213 and end with the pattern 12. There are $\sum_{i\geq 0}\binom{n}{i}C_{i,k}D_{n-i}$ permutations with these properties. Also, the letter $(n+1)$ can be in the leftmost position, which gives $C_{n,k-1}$ permutations, and $(n+1)$ can be in the rightmost position, which gives $C_{n,k}$ permutations, since in this case, two letters immediately to the left of $(n+1)$ cannot form a descent. So,

$$C_{n+1,k} = \sum_{i\geq 0}\binom{n}{i}C_{i,k}D_{n-i} + C_{n,k-1} + C_{n,k}.$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get the following differential equation

$$C_k'(x) = (D(x)+1)C_k(x) + C_{k-1}(x). \tag{2.14}$$

As when solving Equation (2.10), it is convenient to introduce the function $T(x)$ such that $T'(x) = D(x)+1$. Moreover, Equation (2.14) is similar to Equation (2.3) and we can solve it in the same way. Also we observe that from the definitions, $C_1(t) = D(t)$, and thus

$$C_k(x) = e^{T(x)}\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_1} e^{-T(t)}C_1(t)dtdt_1\cdots dt_{k-2} =$$

$$e^{T(x)}\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_1} e^{-T(t)}\left(\frac{e^{-t^2/2}}{1-\int_0^t e^{-m^2/2}dm} - t - 1\right)dtdt_1\cdots dt_{k-2}.$$

□

2.7 Summarizing the results from sections 2.4, 2.5 and 2.6

We recall that the *reverse* $R(\pi)$ of a permutation $\pi = a_1a_2\dots a_n$ is the permutation $a_na_{n-1}\dots a_1$ and the *complement* $C(\pi)$ is the permutation $b_1b_2\dots b_n$ where $b_i = n+1-a_i$. Also, $R\circ C$ is the composition of R and C. We call these bijections of $\mathcal{S}_n$ to itself *trivial*. Let ϕ be an arbitrary trivial bijection. It is easy to see that, for example, there are as many permutations avoiding the pattern 132 as those avoiding the pattern $\phi(132)$. Moreover if, for instance, a permutation π begins with a decreasing pattern of length k, then depending on ϕ, $\phi(\pi)$ either begins with an increasing pattern, or ends with either a decreasing or increasing pattern of length k. This allows us to apply Theorems $6-11$ to a number of other cases. We summarize all the obtained results concerning avoidance of a generalized 3-pattern with no dashes and beginning or ending with either increasing or decreasing subword, in the table below.

	avoid	begin	end	e.g.f.
	123	$12\dots k$	–	
1	123	–	$12\dots k$	$\frac{\sqrt{3}}{2}\frac{e^{x/2}}{\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})}$, if $k=1$
	321	$k\dots 21$	–	$\frac{\sqrt{3}}{2}\frac{e^{x/2}}{\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})}-\frac{1}{2}-\frac{\sqrt{3}}{2}\tan(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})$, if $k=2$
	321	–	$k\dots 21$	0, if $k\geq 3$
	123	$k\dots 21$	–	
2	123	–	$k\dots 21$	$\frac{\sqrt{3}}{2}\frac{e^{x/2}}{\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})}$, if $k=1$
	321	$12\dots k$	–	
	321	–	$12\dots k$	$\frac{e^{x/2}\int_0^x e^{-t/2}t^{k-1}\sin(\frac{\sqrt{3}}{2}t+\frac{\pi}{6}))\ dt}{(k-1)!\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})}$, if $k\geq 2$
	132	$12\dots k$	–	
3	213	–	$12\dots k$	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
	312	$k\dots 21$	–	$e^{-x^2/2}(1-\int_0^x e^{-t^2/2}\ dt)^{-1}-x-1$, if $k=2$
	231	–	$k\dots 21$	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_2}(e^{-t_1^2/2}-$
				$(t_1+1)(1-\int_0^{t_1}e^{-t^2/2}\ dt))dt_1dt_2\cdots dt_{k-2}$, if $k\geq 3$
	132	$k\dots 21$	–	
4	213	–	$k\dots 21$	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
	312	$12\dots k$	–	$\frac{1}{(k-1)!(1-\int_0^x e^{-t^2/2}\ dt)}\int_0^x t^{k-1}e^{-t^2/2}\ dt$, if $k\geq 2$
	231	–	$12\dots k$	
	213	$12\dots k$	–	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
5	132	–	$12\dots k$	
	231	$k\dots 21$	–	$\int_0^x\int_0^t\frac{s^{k-2}e^{T(t)-T(s)}}{(k-2)!(1-\int_0^t e^{-m^2/2}dm)}\ dsdt$, if $k\geq 2$, where
	312	–	$k\dots 21$	$T(x)=-x^2/2+\int_0^x\frac{e^{-t^2/2}}{1-\int_0^t e^{-s^2/2}ds}\ dt$
	213	$k\dots 21$	–	$(1-\int_0^x e^{-t^2/2}\ dt)^{-1}$, if $k=1$
6	132	–	$k\dots 21$	$-\frac{x^{k-1}}{(k-1)!}+\sum_{n=0}^{k-2}\int_0^x\int_0^{t_n}\cdots\int_0^{t_1}\frac{C_{k-n}(t)+\delta_{n,k-2}}{1-\int_0^t e^{-m^2/2}dm}dtdt_1\cdots dt_n$,
	231	$12\dots k$	–	if $k\geq 2$, where $C_k(x)=e^{T(x)}\int_0^x\int_0^{t_{k-2}}\cdots\int_0^{t_1}e^{-T(t)}$.
	312	–	$12\dots k$	$\left(\frac{e^{-t^2/2}}{1-\int_0^t e^{-m^2/2}dm}-t-1\right)dtdt_1\cdots dt_{k-2}$ and $T(x)$ as above

Bibliography

[BabStein] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, Séminaire Lotharingien de Combinatoire, B44b:18pp, 2000.

[Bon] M. Bóna: Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. J. Combin. Theory Ser. A **80** (1997), no. 2, 257–272.

[Claes] A. Claesson: Generalised Pattern Avoidance, European J. Combin. **22** (2001), 961-971.

[ClaesMans] A. Claesson and T. Mansour: Permutations avoiding a pair of generalized patterns of length three with exactly one dash, preprint CO/0107044.

[ElizNoy] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, Proceedings of FPSAC 2001.

[Kit1] S. Kitaev: Multi-avoidance of generalised patterns, to appear in Discrete Mathematics.

[Kit2] S. Kitaev: Partially ordered generalized patterns, to appear in Discrete Mathematics.

[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Loth] M. Lothaire: *Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, **17**, Addison-Wesley Publishing Co., Reading, Mass. (1983).

[Mans1] T. Mansour: Continued fractions and generalized patterns, European J. Combin., to appear (2002), math.CO/0110037.

[Mans2] T. Mansour: Continued fractions, statistics, and generalized patterns, to appear in Ars Combinatorica (2002), preprint CO/0110040.

[SloPlo] N. J. A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, (1995). `http://www.research.att.com/~njas/sequences/`

[Stan] R. Stanley: *Enumerative Combinatorics*, Volume **1**, Cambridge University Press, (1997).

[SchSim] R. Simion, F. Schmidt: Restricted permutations, European J. Combin. **6** (1985), no. 4, 383–406.

Paper III

Simultaneous avoidance of generalized patterns

Simultaneous avoidance of generalized patterns

Sergey Kitaev[1] and Toufik Mansour [2]

Abstract

In [BabStein] Babson and Steingrímsson introduced generalized permutation patterns that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. In [Kit1] Kitaev considered simultaneous avoidance (multi-avoidance) of two or more 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. There either an explicit or a recursive formula was given for all but one case of simultaneous avoidance of more than two patterns. In this paper we find the exponential generating function for the remaining case. Also we consider permutations that avoid a pattern of the form $x-yz$ or $xy-z$ and begin with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $23\dots k1$, $(k-1)(k-2)\dots 1k$ or end with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $1k(k-1)\dots 2$, $k12\dots(k-1)$. For each of these cases we find either the ordinary or exponential generating functions or a precise formula for the number of such permutations. Besides we generalize some of the obtained results as well as some of the results given in [Kit3]: we consider permutations avoiding certain generalized 3-patterns and beginning (ending) with an arbitrary pattern having either the greatest or the least letter as its rightmost (leftmost) letter.

3.1 Introduction and Background

Permutation patterns: All permutations in this paper are written as words $\pi = a_1 a_2 \dots a_n$, where the a_i consist of all the integers $1, 2, \dots, n$. Let $\alpha \in S_n$ and $\tau \in S_k$ be two permutations. We say that α *contains* τ if there exists a subsequence $1 \le i_1 < i_2 < \cdots < i_k \le n$ such that $(\alpha_{i_1}, \dots, \alpha_{i_k})$ is order-isomorphic to τ, that is, for all j and m, $\tau_j < \tau_m$ if and only if $a_{i_j} < a_{i_m}$; in such a context τ is usually called a *pattern*. We say that α *avoids* τ, or is τ-*avoiding*, if α does not contain τ. The set of all τ-avoiding permutations in S_n is denoted by $S_n(\tau)$. For an arbitrary finite collection of patterns T, we say that α avoids T if α avoids each $\tau \in T$; the corresponding subset of S_n is denoted by $S_n(T)$.

While the case of permutations avoiding a single pattern has attracted much attention, the case of multiple pattern avoidance remains less investigated. In particular, it is natural, as the next step, to consider permutations avoiding pairs of patterns τ_1, τ_2. This problem was solved completely for $\tau_1, \tau_2 \in S_3$ (see [SchSim]), for $\tau_1 \in S_3$ and $\tau_2 \in S_4$ (see [W]), and for $\tau_1, \tau_2 \in S_4$ (see [B, K]

[1]Matematik, Chalmers tekniska högskola och Göteborgs universitet, S-412 96 Göteborg, Sweden. E-mail: kitaev@math.chalmers.se

[2]LaBRI, Université Bordeaux 1, 351 cours de la Libération 33405 Talence Cedex, France. E-mail:toufik@labri.fr

and references therein). Several recent papers [CW, MV1, Kr, MV3, MV2] deal with the case $\tau_1 \in S_3$, $\tau_2 \in S_k$ for various pairs τ_1, τ_2.

Generalized permutation patterns: In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns* (*GPs*) where two adjacent letters in a pattern may be required to be adjacent in the permutation. Such an adjacency requirement is indicated by the absence of a dash between the corresponding letters in the pattern. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, but the pattern 2-3-1 occurs also in the subwords 562 and 563. Note that a classical pattern should, in our notation, have dashes at the beginning and end. Since most of the patterns considered in this paper satisfy this, we suppress these dashes from the notation. Thus, a pattern with no dashes corresponds to a contiguous subword anywhere in a permutation. The motivation for introducing these patterns was the study of Mahonian statistics. A number of results on GPs were obtained by Claesson, Kitaev and Mansour. See for example [Claes], [Kit1, Kit2, Kit3] and [Mans1, Mans2, Mans3].

As in [SchSim], dealing with the classical patterns, one can consider the case when permutations have to avoid two or more generalized patterns simultaneously. A complete solution for the number of permutations avoiding a pair of 3-patterns of type (1,2) or (2,1), that is, the patterns having one internal dash, is given in [ClaesMans]. In [Kit1] Kitaev gives either an explicit or a recursive formula for all but one case of simultaneous avoidance of more than two patterns. This is the case of avoiding the GPs 123, 231 and 312 simultaneously. In Theorem 1 we find the exponential generating function (e.g.f.) for the number of such permutations.

As it was discussed in [Kit3], if a permutation begins (resp. ends) with the pattern $p = p_1 p_2 \dots p_k$, that is, the k leftmost (resp. rightmost) letters of the permutation form the pattern p, then this is the same as avoidance of $k! - 1$ patterns simultaneously. For example, beginning with the pattern 123 is equivalent to the simultaneous avoidance of the patterns (132], (213], (231], (312] and (321] in the Babson-Steingrímsson notation. Thus demanding that a permutation must begin or end with some pattern, in fact, we are talking about simultaneous avoidance of generalized patterns. The motivation for considering additional restrictions such as beginning or ending with some patterns is their connection to some classes of trees. An example of such a connection can be found in [Kit3, Theorem 5]. There it was shown that there is a bijection between n-permutations avoiding the pattern 132 and beginning with the pattern 12 and *increasing rooted trimmed trees* with $n+1$ nodes. We recall that a trimmed tree is a tree where no node has a single leaf as a child (every leaf has a sibling) and in an increasing rooted tree, nodes are numbered and the numbers increase as we move away from the root. The avoidance of a generalized 3-pattern p with no dashes and, at the same time, beginning or ending with an increasing or decreasing pattern was discussed in [Kit3]. Theorem 2 generalizes some of these results to the case of beginning (resp. ending) with an arbitrary pattern avoiding p and having the greatest or least letter as the rightmost (resp. leftmost) letter.

Propositions 4 – 15 (resp. 16 – 27) give a complete description for the number of permutations avoiding a pattern of the form $x - yz$ or $xy - z$ and beginning with one of the patterns $12\ldots k$ or $k(k-1)\ldots 1$ (resp. $23\ldots k1$ or $(k-1)(k-2)\ldots 1k$). For each of these cases we find either the ordinary or exponential generating functions or a precise formula for the number of such permutations. Theorem 3 generalizes some of these results. Besides, the results from Propositions 4–27 give a complete description for the number of permutations that avoid a pattern of the form $x - yz$ or $xy - z$ and end with one of the patterns $12\ldots k$, $k(k-1)\ldots 1$, $1k(k-1)\ldots 2$ and $k12\ldots(k-1)$. To get the last one of these we only need to apply the reverse operation discussed in the next section. The results of Theorems 2 and 3 can also be used to get the case of ending with a pattern from the sets Δ_k^{min} or Δ_k^{max} introduced in the next section.

Except for the empty permutation, every permutation ends and begins with the pattern $p = 1$. To simplify the discussion we assume that the empty permutation also begin with the pattern 1. This does not course any harm since, to count the generating functions in question for this, we need only subtract 1 from the generating functions obtained in this paper.

3.2 Preliminaries

The *reverse* $R(\pi)$ of a permutation $\pi = a_1 a_2 \ldots a_n$ is the permutation $a_n a_{n-1} \ldots a_1$. The *complement* $C(\pi)$ is the permutation $b_1 b_2 \ldots b_n$ where $b_i = n + 1 - a_i$. Also, $R \circ C$ is the composition of R and C. For example, $R(13254) = 45231$, $C(13254) = 53412$ and $R \circ C(13254) = 21435$. We call these bijections of S_n to itself *trivial*, and it is easy to see that for any pattern p the number $A_p(n)$ of permutations avoiding the pattern p is the same as for the patterns $R(p)$, $C(p)$ and $R \circ C(p)$. For example, the number of permutations that avoid the pattern 132 is the same as the number of permutations that avoid the pattern 231. This property holds for sets of patterns as well. If we apply one of the trivial bijections to all patterns of a set G, then we get a set G' for which $A_{G'}(n)$ is equal to $A_G(n)$. For example, the number of permutations avoiding $\{123, 132\}$ equals the number of those avoiding $\{321, 312\}$ because the second set is obtained from the first one by complementing each pattern.

In this paper we denote the nth Catalan number by C_n; the generating function for these numbers by $C(x)$; the nth Bell number by B_n.

Also, $N_q^p(n)$ denotes the number of permutations that avoid the pattern q and begin with the pattern p; $G_q^p(x)$ (resp. $E_q^p(x)$) denotes the ordinary (resp. exponential) generating function for the number of such permutations. Besides, Γ_k^{min} (resp. Γ_k^{max}) denotes the set of all k-patterns with no dashes such that the least (resp. greatest) letter of a pattern is the rightmost letter; Δ_k^{min} (resp. Δ_k^{max}) denotes the set of all k-patterns with no dashes such that the least (resp. greatest) letter of a pattern is the leftmost letter.

Recall the following properties of $C(x)$:

$$C(x) = \frac{1-\sqrt{1-4x}}{2x} = \frac{1}{1-xC(x)}. \tag{3.1}$$

3.3 Simultaneous avoidance of 123, 231 and 312

The *Entringer numbers* $E(n,k)$ (see [SloPlo, Sequence A000111/M1492]) are the number of permutations on $1,2,\ldots,n+1$, starting with $k+1$, which, after initially falling, alternately fall then rise. The Entringer numbers (see [Ent]) are given by

$$E(0,0) = 1, \qquad E(n,0) = 0,$$

together with the recurrence relation

$$E(n,k) = E(n,k+1) + E(n-1,n-k).$$

The numbers $E(n) = E(n,n)$, are the secant and tangent numbers given by the generating function

$$\sec x + \tan x.$$

The following theorem completes the consideration of multi-avoidance of more than two generalized 3-patterns with no dashes made in [Kit1].

Theorem 1. *Let $E(x)$ be the e.g.f. for the number of permutations that avoid 123, 231 and 312 simultaneously. Then*

$$E(x) = 1 + x(\sec(x) + \tan(x)).$$

Proof. Let $s(n; i_1, \ldots, i_m)$ denote the number of permutations $\pi \in S_n(123, 231, 312)$ such that $\pi_1\pi_2\ldots\pi_m = i_1 i_2 \ldots i_m$ and $f : S_n \to S_n$ be a map defined by

$$f(\pi_1\pi_2\ldots\pi_n) = (\pi_1+1)(\pi_2+1)\ldots(\pi_n+1),$$

where the addition is modulo n. Using f one can see that for all $a = 1, 2, \ldots, n-1$,

$$s(n;a) = s(n;a+1). \tag{3.2}$$

Thus, $|S_n(123,231,312)| = ns(n;1)$ and we only need to prove that $s(n;1) = E_{n-1}$, where E_n is the nth Euler number (see [SloPlo, Sequence A000111/M1492]).

Suppose $\pi \in S_n(123,231,312)$ is an n-permutation such that $\pi_1 = 1$ and $\pi_2 = t$. Since π avoids 123, we get $\pi_3 \le t-1$ and it is easy to see that

$$s(n;1,t) = \sum_{j=2}^{t-1} s(n;1,t,j) = \sum_{j=1}^{t-2} s(n-1;t-1,j),$$

so

$$s(n;1,t+1) = s(n;1,t) + \sum_{j=1}^{t-1} s(n-1;t,j) - \sum_{j=1}^{t-2} s(n-1;t-1,j).$$

Using (3.2) we get

$$s(n;1,t+1) = s(n;1,t)+s(n-1;t,1)+\sum_{j=2}^{t-1} s(n-1;t-1,j-1)-\sum_{j=1}^{t-2} s(n-1;t-1,j),$$

and by (3.2) again, we have for all $t = 2, 3, \ldots, n-1$,

$$s(n;1,t+1) = s(n;1,t) + s(n-1;1,n-t+1).$$

Besides, by the definition, it is easy to see that $s(n;1,2) = 0$ for all $n \geq 3$, hence using the definition of Entringer numbers [Ent] we get $s(n;1) = \sum_{t=2}^{n} s_{n;1,t} = E_{n-1}$, as required. □

3.4 Avoiding a 3-pattern with no dashes and beginning with a pattern whose rightmost letter is the greatest or smallest

The following theorem generalizes Theorems 7 and 8 in [Kit3]. Recall the definition of $E_q^p(x)$ in Section 3.2.

Theorem 2. *Suppose $p_1, p_2 \in \Gamma_k^{min}$ and $p_1 \in S_k(132)$, $p_2 \in S_k(123)$. Thus, the complements $C(p_1), C(p_2) \in \Gamma_k^{max}$ and $C(p_1) \in S_k(312)$, $C(p_2) \in S_k(321)$. Then, for $k \geq 2$,*

$$E_{132}^{p_1}(x) = E_{312}^{C(p_1)}(x) = \frac{\int_0^x t^{k-1} e^{-t^2/2}\, dt}{(k-1)!(1-\int_0^x e^{-t^2/2}\, dt)}$$

and

$$E_{123}^{p_2}(x) = E_{321}^{C(p_2)}(x) = \frac{e^{x/2}\int_0^x e^{-t/2} t^{k-1} \sin(\frac{\sqrt{3}}{2}t + \frac{\pi}{6}))\, dt}{(k-1)!\cos(\frac{\sqrt{3}}{2}x + \frac{\pi}{6})}.$$

Proof. To prove the theorem, it is enough to copy the proofs of Theorems 7 and 8 in [Kit3], since the fact that the first $k-1$ letters of p are possibly not in decreasing order is immaterial for the proofs of that theorems. Thus we can get the formula for $E_{132}^p(x)$ and $E_{123}^p(x)$, and automatically, using properties of the complement, the formula for $E_{312}^{C(p)}(x)$ and $E_{321}^{C(p)}(x)$, directly from these theorems. However we give here a proof of the formula for $E_{132}^p(x)$ and refer to [Kit3, Theorem 8] for a proof of the formula for $E_{123}^p(x)$.

If $k = 1$, we have no additional restrictions, that is, we are dealing only with the avoidance of 132 and, according to [ElizNoy, Theorem 4.1] or [Kit2, Theorem 12],

$$E_{132}^1(x) = \frac{1}{1-\int_0^x e^{-t^2/2}\, dt}.$$

Also, according to [Kit3, Theorem 6],

$$E^{12}_{132}(x) = \frac{e^{-x^2/2}}{1 - \int_0^x e^{-t^2/2}\, dt} - x - 1.$$

Let $R_{n,k}$ (resp. $F_{n,k}$) denote the number of n–permutations that avoid the pattern 132 and begin with a decreasing (resp. increasing) subword of length $k > 1$ and let π be such a permutation of length $n+1$. Suppose $\pi = \sigma 1 \tau$. If $\tau = \epsilon$ then, obviously, there are $R_{n,k}$ ways to choose σ. If $|\tau| = 1$, that is, 1 is in the second position from the right, then there are n ways to choose the rightmost letter in π and we multiply this by $R_{k,n-1}$, which is the number of ways to choose σ. If $|\tau| > 1$ then τ must begin with the pattern 12, otherwise the letter 1 and the two leftmost letters of τ form the pattern 132, which is forbidden. So, in this case there are $\sum_{i \geq 0} \binom{n}{i} R_{i,k} F_{n-i,2}$ such permutations with the right properties, where i indicates the length of σ. In the last formula, of course, $R_{i,k} = 0$ if $i < k$. Finally we have to consider the situation when 1 is in the k-th position. In this case we can choose the letters of σ in $\binom{n}{k-1}$ ways, write them in decreasing order and then choose τ in $F_{n-k+1,2}$ ways. Thus

$$R_{n+1,k} = R_{n,k} + nR_{n-1,k} + \sum_{i \geq 0} \binom{n}{i} R_{i,k} F_{n-i,2} + \binom{n}{k-1} F_{n-k+1,2}. \qquad (3.3)$$

We observe that (3.3) is not valid for $n = k-1$ and $n = k$. Indeed, if 1 is in the k-th position in these cases, the term $\binom{n}{k-1} F_{n-k+1,2}$, which counts the number of such permutations, is zero, whereas there is one "good" $(n+1)$–permutation in case $n = k-1$ and n good $(n+1)$–permutations in case $n = k$. Multiplying both sides of the equality with $x^n/n!$, summing over n and using the observation above (which gives the term $x^{k-1}/(k-1)! + kx^k/k!$ in the right-hand side of equality (3.4)), we get

$$\frac{d}{dx} E^p_{132}(x) = (E^{12}_{132}(x) + x + 1) E^p_{132}(x) + (E^{12}_{132}(x) + x + 1) \frac{x^{k-1}}{(k-1)!}, \qquad (3.4)$$

with the initial condition $E^p_{132}(0) = 0$. We solve this equation and get

$$E^p_{132}(x) = \frac{E^1_{132}(x)}{(k-1)!} \int_0^x \frac{(E^{12}_{132}(t) + t + 1) t^{k-1}}{E^1_{132}(t)}\, dt = \frac{E^1_{132}(x)}{(k-1)!} \int_0^x t^{k-1} e^{-t^2/2}\, dt.$$

□

Remark 1. *It is obvious that if in the previous theorem $p_1 \notin S_k(132)$ and $p_2 \notin S_k(123)$, then $E^{p_1}_{132}(x) = E^{p_2}_{123}(x) = 0$.*

3.5 Avoiding a pattern x-yz and beginning with an increasing or decreasing pattern

In this section we consider avoidance of one of the patterns $1-23$, $1-32$, $2-31$, $2-13$, $3-12$ and $1-32$ and beginning with a decreasing pattern. We get all the other cases, that is, avoidance of one of these patterns and beginning with an increasing pattern, by the complement operation. For instance, we have $E_{1-23}^{k(k-1)\cdots 1}(x) = E_{3-21}^{12\cdots k}(x)$.

Proposition 1. *We have*

$$E_{1-23}^{k(k-1)\cdots 1}(x) = E_{1-32}^{k(k-1)\cdots 1}(x) = \begin{cases} (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. We prove the statement for the pattern $1-23$. All the arguments we give for this pattern are valid for the pattern $1-32$. The only difference is that instead of decreasing order in τ (see below), we have increasing order.

Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n–permutations that avoid the pattern $1-23$ and begin with a decreasing subword of length k. Suppose $\pi = \sigma 1 \tau$ be one of such permutations of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of $1-23$ in π starting from the letter 1. If $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Besides, if $|\sigma| = k-1$ and letters of σ are in decreasing order, we get $\binom{n}{k-1}$ additional possibilities to choose π. Thus

$$B_{n+1,k} = \sum_{i\geq 0} \binom{n}{i} B_{i,k} + \binom{n}{k-1}.$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get the differential equation

$$\frac{d}{dx}E_{1-23}^{k(k-1)\cdots 1}(x) = (E_{1-23}^{k(k-1)\cdots 1}(x) + \frac{x^{k-1}}{(k-1)!})e^x$$

with the initial condition $E_{1-23}^{k(k-1)\cdots 1}(0) = 0$. The solution to this equation is given by

$$E_{1-23}^{k(k-1)\cdots 1}(x) = (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt. \tag{3.5}$$

If $k = 1$, then there is no additional restriction. According to [Claes, Prop. 2] (resp. [Claes, Prop. 5]), the number of n-permutations that avoid the pattern 1-23 (resp. 1-32) is the nth Bell number and the e.g.f. for the Bell numbers is e^{e^x-1}. However, all the arguments used for $k \geq 2$ remain the same for the case $k = 1$ except for the fact that we do not count the empty permutation, which,

of course, avoids 1-23. So, if $k = 1$, we need to add 1 to the right-hand side of (3.5):

$$E^1_{1-23}(x) = e^{e^x} \int_0^x e^{-e^t+t}\, dt + 1 = e^{e^x-1}.$$

□

Proposition 2. *We have*

$$E^{k(k-1)\dots1}_{3-12}(x) = \begin{cases} e^{e^x} \int_0^x e^{-e^t} \sum\limits_{n\geq k-1} \dfrac{t^n}{n!}\, dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n–permutations that avoid the pattern $3-12$ and begin with a decreasing subword of length k. Suppose $\pi = \sigma(n+1)\tau$ be such a permutation of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of the pattern $3-12$ in π starting from the letter $(n+1)$. If $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Besides, if $n \geq k-1$, then π can be decreasing, that is, $(n+1)$ can be in the leftmost position. Thus

$$B_{n+1,k} = \sum_{i\geq 0} \binom{n}{i} B_{i,k} + \delta_{n,k},$$

where

$$\delta_{n,k} = \begin{cases} 1, & \text{if } n \geq k-1, \\ 0, & \text{else.} \end{cases}$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get the differential equation

$$\frac{d}{dx} E^{k(k-1)\dots1}_{3-12}(x) = e^x E^{k(k-1)\dots1}_{3-12}(x) + \sum_{n\geq k-1} \frac{x^n}{n!}$$

with the initial condition $E^{k(k-1)\dots1}_{3-12}(0) = 0$. The solution to this equation is given by

$$E^{k(k-1)\dots1}_{3-12}(x) = e^{e^x} \int_0^x e^{-e^t} \sum_{n\geq k-1} \frac{t^n}{n!}\, dt. \tag{3.6}$$

If $k = 1$, then there is no additional restriction. In [Claes, Prop. 5] it is shown that $E^1_{1-32}(x) = e^{e^x-1}$. Using the complement, the number of n-permutations that avoid $1-32$ is equal to the number of n-permutations that avoid $3-12$. We get that $E^1_{3-12}(x) = e^{e^x-1}$. However, all the arguments used for the case $k \geq 2$ remain the same for the case $k = 1$ except the fact that we do not count the empty permutation, which avoids 3-12. So, if $k = 1$, we need to add 1 to the right-hand side of (3.6):

$$E^1_{3-12}(x) = e^{e^x} \int_0^x e^{-e^t} e^t\, dt + 1 = e^{e^x-1}.$$

□

Proposition 3. *We have*

$$E_{3-21}^{k(k-1)\dots1}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ e^{e^x} \int_0^x e^{-e^t}(e^t - 1)\ dt, & \text{if } k = 2, \\ e^{e^x - 1}, & \text{if } k = 1. \end{cases}$$

Proof. For $k \geq 3$, the statement is obviously true. If $k = 1$, then the statement follows from [Claes, Prop. 2] and the fact that there are as many n-permutations avoiding the pattern $1-23$, as n-permutations avoiding the pattern $3-21$. For the case $k = 2$, we can use exactly the same arguments as those in the proof of Proposition 2 to get the same recurrence relation and thus the same formula, which, however, is valid only for $k = 2$. □

Recall the definition of N_q^p in Section 3.2.

Proposition 4. *We have*

$$N_{2-13}^{k(k-1)\dots1}(n) = \begin{cases} C_{n-k+1}, & \text{if } n \geq k, \\ 0, & \text{else.} \end{cases}$$

Proof. If $k = 1$, then the statement follows from [Claes, Prop. 22]. Suppose $k \geq 2$ and let $\pi = \sigma n \tau$ be an n-permutation avoiding $2-31$ and beginning with the pattern $k(k-1)\dots1$. Suppose, without loss of generality that σ consists of the letters $1, 2, \dots, \ell$. Now ℓ must be the rightmost letter of σ, since otherwise ℓ, the rightmost letter of σ and n form the pattern $2-13$. Also, the letter $(\ell-1)$ must be next to the rightmost letter of σ since otherwise the letter $(\ell-1)$, next to the rightmost letter of σ and the letter ℓ form the pattern $2-13$. And so on. Thus σ must be increasing, which contradicts the fact that π must begin with a decreasing pattern of length greater than 1. So $|\sigma| = 0$ and τ must begin with the pattern $(k-1)(k-2)\dots1$. Now, we can consider the letter $(n-1)$ and, by the same reasoning, get that it must be in the second position of π. Then we consider $(n-2)$, and so on up to the letter $(n-k+2)$. Finally, we get that $\pi = n(n-1)\dots(n-k+2)\pi'$, where π' must avoid the pattern $2-13$ and thus, there are C_{n-k+1} ways to choose π ([Claes, Prop. 22]). □

Recall that $C(x)$ is the generating function for the Catalan numbers. Also recall the definition of G_q^p in Section 3.2.

Proposition 5. *We have*

$$G_{2-31}^{k(k-1)\dots1}(x) = \begin{cases} x^k C^{k+1}(x), & \text{if } k \geq 2 \\ C(x), & \text{if } k = 1. \end{cases}$$

Proof. If $k = 1$, then there is no additional restriction, and thus $G_{2-31}^1(x) = C(x)$ (applying the complement operation to [Claes, Prop. 22]).

Suppose $k \geq 2$. Using the reverse, we see that beginning with $k(k-1)\dots1$ and avoiding $2-31$ is equivalent to ending with $12\dots k$ and avoiding $13-2$, which by [Claes] is equivalent to ending with $12\dots k$ and avoiding $1-3-2$.

Suppose $\pi = \pi' n \pi''$ ends with $12\dots k$ and avoids $1-3-2$. Each letter of π' must be greater than any letter of π'', since otherwise we have an occurrence of the pattern $1-3-2$ involving the letter n. Also, π' and π'' avoid the pattern $1-3-2$, and π'' ends with the pattern $12\dots k$. In terms of generating functions (the generating function for the number of permutations ending with $12\dots k$ and avoiding $1-3-2$ is, of course, $G_{2-31}^{k(k-1)\dots1}(x)$) this means that

$$G_{2-31}^{k(k-1)\dots1}(x) = xC(x)G_{2-31}^{k(k-1)\dots1}(x) + xG_{2-31}^{(k-1)\dots1}(x), \tag{3.7}$$

where the rightmost term corresponds to the case when π'' is empty. Now, (3.1) and (3.7) give

$$G_{2-31}^{k(k-1)\dots1}(x) = x^k C(x)/(1-xC(x))^k = x^k C^{k+1}(x).$$

□

3.6 Avoiding a pattern xy-z and beginning with an increasing or decreasing pattern

First of all we state the following well-known binomial identity

$$\sum_{i=1}^{n-m-k+1} \binom{n-m-i}{k-1}\binom{m+i-1}{m} = \binom{n}{m+k}. \tag{3.8}$$

Let $s_q(n)$ denote the cardinality of the set $S_n(q)$ and $s_q(n; i_1, i_2, \dots i_m)$ denote the number of permutations $\pi \in S_n(q)$ with $\pi_1\pi_2\dots\pi_m = i_1 i_2 \dots i_m$.

In this section we consider avoidance of one of the patterns 12-3, 13-2 and 23-1 and beginning with an increasing or decreasing pattern. We get all the other cases, which are avoidance of one of the patterns 32-1, 31-2 and 21-3 and beginning with an increasing or decreasing pattern, by the complement operation. For instance, we have $N_{13-2}^{12\dots k}(n) = N_{31-2}^{k(k-1)\dots1}(n)$.

3.6.1 The pattern $12-3$

We first consider beginning with the pattern $p = k\dots21$. In [ClaesMans, Lemma 9] it was proved that

$$s_{12-3}(n;i) = \sum_{j=0}^{i-1}\binom{i-1}{j}s_{12-3}(n-2-j),$$

together with $s_{12-3}(n;n) = s_{12-3}(n;n-1) = s_{12-3}(n-1)$.

On the other hand, from the definitions, it is easy to see that

$$N_{12-3}^{k(k-1)\dots1}(n) = \sum_{i=1}^{n-k+1} \binom{n-i}{k-1} s_{12-3}(n-k+1;i).$$

Hence, using (3.8) and the fact shown in [Claes] that $s_{12-3}(n)$ equals B_n, we get the following proposition.

Proposition 6. *For all $n \geq k+1$, we have*

$$N_{12-3}^{k(k-1)\dots1}(n) = (k+1)B_{n-k} + \\ + \sum_{j=0}^{n-k-2} \left(\tbinom{n}{k+j} - k\tbinom{n-k-1}{j} - \tbinom{n-k}{j} \right) B_{n-k-1-j},$$

together with $N_{12-3}^{k(k-1)\dots1}(k) = 1$ and $N_{12-3}^{k(k-1)\dots1}(n) = 0$ for all $n \leq k-1$.

Now, let us consider beginning with the pattern $p = 12\dots k$. From the definitions, it is easy to see that $N_{12-3}^{12\dots k}(n) = 0$ for all n, where $k \geq 3$, and $N_{12-3}^{1}(n) = s_{12-3}(n) = B_n$ (see [ClaesMans, Prop. 10]). Thus, we only need to consider the case $k = 2$.

Suppose $\pi \in S_{12-3}(n)$ is a permutation with $\pi_1 < \pi_2$. It is easy to see that $\pi_2 = n$. Hence $N_{12-3}^{12}(n) = (n-1)s_{12-3}(n-2)$, for all $n \geq 2$, and by [ClaesMans, Prop. 10], we get the truth of the following

Proposition 7. *We have*

$$E_{12-3}^{12\dots k}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ x^2 \displaystyle\sum_{j=0}^{k} (1-jx)^{-1} \sum_{d\geq 0} \frac{x^d}{(1-x)(1-2x)\dots(1-dx)}, & \text{if } k = 2, \\ \displaystyle\sum_{d\geq 0} \frac{x^d}{(1-x)(1-2x)\dots(1-dx)}, & \text{if } k = 1. \end{cases}$$

3.6.2 The pattern $13-2$

Let us introduce an object that plays an important role in the proof of the main result in this case. For $n \geq m+1 \geq 0$, we define

$$A(n;m) = \sum_{1\leq i_m < \dots < i_2 < i_1 < n-1} s_{1-3-2}(n; i_1, i_2, \dots, i_m).$$

We extend this definition to $m = 0$ by $A(n;0) = s_{1-3-2}(n)$.

Lemma 1. *For all $n \geq m \geq 0$,*

$$A(n;m) = \sum_{j\geq 0} (-1)^j \binom{m+1-j}{j} s_{1-3-2}(n-j).$$

Proof. For $m = 0$ the lemma holds by definitions. Let $m \geq 0$; so

$$\begin{aligned}
A(n;m) &= \sum_{1\leq i_m<\cdots<i_2<i_1<n-1} \sum_{j=1}^{n} s_{1-3-2}(n; i_1, i_2, \ldots, i_m, j),\\
&= A(n;m+1) + \sum_{1\leq i_m<\cdots<i_2<i_1<n-1} s_{1-3-2}(n; i_1, i_2, \ldots, i_m, n),\\
&= A(n;m+1) + \sum_{1\leq i_m<\cdots<i_2<i_1<n-1} s_{1-3-2}(n-1; i_1, i_2, \ldots, i_m),\\
&= A(n;m+1) + \sum_{1\leq i_m<\cdots<i_2<n-2} s_{1-3-2}(n-1; n-1, i_2, \ldots, i_m) +\\
&\qquad + \sum_{1\leq i_m<\cdots<i_2<i_1<n-2} s_{1-3-2}(n-1; i_1, i_2, \ldots, i_m)
\end{aligned}$$

$$= A(n;m+1) + A(n-1;m) + \sum_{1\leq i_{m-1}<\cdots<i_1<n-2} s_{1-3-2}(n-2; i_1, \ldots, i_{m-1}),$$

$$= \cdots = A(n;m+1) + A(n-1;m) + \cdots + A(n-m-1;0).$$

Hence, using induction on m, we get

$$\begin{aligned}
A(n;m+1) = &\sum_{j\geq 0} (-1)^j \binom{m+1-j}{j} s_{1-3-2}(n-j)\\
&- \sum_{d=0}^{m} \sum_{j\geq 0} (-1)^j \binom{m-d+1-j}{j} s_{1-3-2}(n-1-d-j).
\end{aligned}$$

Using the identity $\binom{r}{0} - \binom{r}{1} + \cdots + (-1)^s \binom{r}{s} = \binom{r-1}{s}$, we get

$$\begin{aligned}
A(n;m+1) = &\sum_{j\geq 0} (-1)^j \binom{m+1-j}{j} s_{1-3-2}(n-j)\\
&- \sum_{d=0}^{m} (-1)^d \binom{m-d}{d} s_{1-3-2}(n-1-d).
\end{aligned}$$

Now using the identity $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$, we get

$$A(n;m+1) = \sum_{j\geq 0} (-1)^j \binom{m+2-j}{j} s_{1-3-2}(n-j),$$

which means that the lemma holds for $m+1$. □

Now we find $N_{13-2}^{k(k-1)\ldots 1}(n)$.

Proposition 8. *Let $k \geq 1$. For all $n \geq 0$,*

$$N_{13-2}^{k(k-1)\dots1}(n) = C_{n+1-k} + \sum_{d=0}^{k-2} \sum_{j \geq 0} (-1)^j \binom{k+1-d-j}{j} C_{n-d-j}.$$

Proof. Claesson [Claes] proved that the set of permutations that avoid the pattern $13-2$ is the same as the set of permutations that avoid the pattern $1-3-2$, hence

$$N_{13-2}^{k(k-1)\dots1}(n) = N_{1-3-2}^{k(k-1)\dots1}(n). \tag{3.9}$$

If the leftmost letter of a permutation avoiding 13-2 and beginning with the pattern $k(k-1)\dots1$ is n, then, obviously, there are $N_{1-3-2}^{(k-1)(k-2)\dots1}(n-1)$ such permutations. Otherwise, it is easy to see that there are $A(n;k)$ such permutations. So, by Lemma 1 and the considerations above, also using the fact that the number of $(1-3-2)$-avoiding n-permutations in S_n is C_n, we get

$$N_{13-2}^{k(k-1)\dots1}(n) = N_{13-2}^{(k-1)(k-2)\dots1}(n-1) + \sum_{j \geq 0} (-1)^j \binom{k+1-j}{j} C_{n-j}.$$

Moreover, using the definitions and Equation (3.9), we have $N_{13-2}^{1}(n) = s_{1-3-2}(n)$ $= C_n$, for all $n \geq 0$. Hence, by induction on k, the proposition holds. □

Now, let us consider the case of $N_{13-2}^{12\dots k}(n)$.

Proposition 9. *Let $k \geq 1$. For all $n \geq k$, we have*

$$N_{13-2}^{12\dots k}(n) = C_{n+1-k}.$$

Proof. Suppose $\pi = \pi' n \pi''$ is a permutation in $S_n(13-2) = S_n(1-3-2)$ (see (3.9)), such that $\pi_1 < \pi_2 < \cdots < \pi_k$. It is easy to see that there exists an m such that

$$\pi = (m+1)(m+2)\dots(m+k-1)\beta n \pi'',$$

where β is a $1-3-2$-avoiding permutation on the letters $m+k, m+k+1, \dots, n-1$, and $\pi'' \in S_m(1-3-2)$. Hence, in terms of generating functions, we get

$$\sum_{n \geq 0} N_{13-2}^{12\dots k}(n) x^n = x^k C^2(x).$$

The rest is easy to check using the identity $xC^2(x) = C(x) - 1$. □

3.6.3 The pattern $23-1$

We first consider beginning with the pattern $p = k(k-1)\dots1$.

Proposition 10. *For all $k \geq 1$,*

$$E_{23-1}^{k(k-1)\dots1}(x) = x^{k-1} \left(\sum_{d \geq 0} \frac{x^d}{(1-x)(1-2x)\cdots(1-dx)} - 1 \right).$$

Proof. Let $\pi \in S_n(23-1)$ be a permutation such that $\pi_1 < \pi_2 < \cdots < \pi_k$. Since π avoids $23-1$, we have $\pi_j = j$, for each $j = 1, 2, \ldots, k-1$. Hence $\pi = 12\ldots(k-1)\pi'$, where π' is a non-empty $23-1$-avoiding permutation in S_{n+1-k}. The rest is easy to get by using [ClaesMans, Prop. 17]. □

Now let us consider beginning with the pattern $p = 12\ldots k$.

Proposition 11. *Suppose $k \geq 1$. For all $n \geq k+1$,*

$$N_{23-1}^{12\ldots k}(n) = \left(1 + \binom{n-1}{k-1}\right) B_{n-k} + \sum_{j=0}^{n-k-2} \left[\binom{n-1}{k+j} - \binom{n-k-1}{j}\right] B_{n-k-1-j},$$

with $N_{23-1}^{12\ldots k}(k) = 1$.

Proof. In [ClaesMans, Lemma 16] proved that for all $2 \leq i \leq n-1$,

$$s_{23-1}(n;i) = \sum_{j=0}^{i-2} \binom{i-2}{j} s_{23-1}(n-2-j),$$

together with $s_{23-1}(n;n) = s_{23-1}(n;1) = s_{23-1}(n-1) = B_{n-1}$.

On the other hand, by the definitions, it is easy to see that

$$N_{23-1}^{12\ldots k}(n) = \sum_{i=1}^{n-k+1} \binom{n-i}{k-1} s_{23-1}(n-k+1;i).$$

Hence, using (3.8) and the fact that [Claes] $s_{23-1}(n)$ is given by B_n, we get the desired result. □

3.7 Avoiding a pattern xy-z and beginning with the pattern $(k-1)(k-2)\ldots 1k$ or $23\ldots k1$

In this section we consider avoidance of one of the patterns $12-3$, $13-2$, $23-1$, $21-3$, $31-2$ and $13-2$ and beginning with the pattern $(k-1)(k-2)\ldots 1k$. The case when a permutation begins with the pattern $23\ldots k1$ and avoids a pattern $xy-z$ can be obtained then by the complement operation.

3.7.1 Avoiding $12-3$ and beginning with $(k-1)(k-2)\ldots 1k$

Proposition 12. *We have*

$$N_{12-3}^{(k-1)(k-2)\ldots 1k}(n) = \binom{n-1}{k-1} B_{n-k}.$$

Proof. Suppose $\pi = \pi' n \pi''$ avoids the pattern $12-3$ and begins with the pattern $(k-1)(k-2)\ldots 1k$. We have that π' must be decreasing, since otherwise we have an occurrence of the pattern $12-3$ involving the letter n, and π'' must

avoid $12-3$. Also, since π begins with $(k-1)\dots 21k$, the length of π' is $k-1$. Hence, by [Claes] (the number of permutations in $S_n(12-3)$ is given by B_n), we have

$$N_{12-3}^{(k-1)(k-2)\dots 1k}(n) = \binom{n-1}{k-1} B_{n-k}.$$

□

3.7.2 Avoiding $13-2$ and beginning with $(k-1)(k-2)\dots 1k$

By [Claes], a permutation π avoids the pattern $13-2$ if and only if π avoids $1-3-2$.

Suppose $\pi = \pi' n \pi''$ is an n-permutation avoiding $1-3-2$ and beginning with $(k-1)(k-2)\dots 1k$. Obviously, π' and π'' avoid $1-3-2$ and each letter of π' is greater than any letter of π'', since otherwise we have an occurrence of the pattern $1-3-2$ involving the letter n. Also, π' begins with the pattern $(k-1)(k-2)\dots 1k$ or $\pi' = (k-1)(k-2)\dots 1$.

By [Knuth], the generating function for the number of permutations that avoid $1-3-2$ is $C(x)$, hence, using the considerations above,

$$G_{13-2}^{(k-1)(k-2)\dots 1k}(x) = xG_{13-2}^{(k-1)(k-2)\dots 1k}(x)C(x) + x^k C(x).$$

Therefore, by (3.1), we get the following.

Proposition 13. *We have*

$$G_{13-2}^{(k-1)(k-2)\dots 1k}(x) = x^k C^2(x).$$

Hence

$$N_{13-2}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} C_{n-(k-1)}, & \text{if } n \geq k \\ 0, & \text{else.} \end{cases}$$

3.7.3 Avoiding $21-3$ and beginning with $(k-1)(k-2)\dots 1k$

If $k \geq 3$ then, by the definitions, we have $N_{21-3}^{(k-1)(k-2)\dots 1k}(n) = 0$.

If $k = 1$ then, by the definitions and [Claes], we have $N_{21-3}^{1}(n) = B_n$.

Suppose $k = 2$ and $\pi = \pi' n \pi''$ is an n-permutation avoiding the pattern $21-3$ and beginning with the pattern $(k-1)(k-2)\dots 1k = 12$. It is easy to see that π' must be increasing, and the length of π' is at least 1. Thus, using the fact that the number of permutations in $S_n(21-3)$ is given by B_n (see [Claes]), we have

$$N_{21-3}^{(k-1)(k-2)\dots 1k}(n) = \sum_{j=1}^{n-1} \binom{n-1}{j} B_{n-1-j}. \tag{3.10}$$

Since $B_n = \displaystyle\sum_{j=0}^{n-1} \binom{n-1}{j} B_{n-1-j}$, equality (3.10) gives that

$$N_{21-3}^{(k-1)(k-2)\dots 1k}(n) = B_n - B_{n-1}.$$

Thus we have proved the following.

Proposition 14.

$$N_{21-3}^{(k-1)(k-2)\ldots 1k}(n) = \begin{cases} 0, & \text{if } k \geq 3 \\ B_n - B_{n-1}, & \text{if } k = 2, \\ B_n, & \text{if } k = 1. \end{cases}$$

3.7.4 Avoiding $23-1$ and beginning with $(k-1)(k-2)\ldots 1k$

Proposition 15. *We have that* $N_{23-1}^{(k-1)(k-2)\ldots 1k}(n)$ *is given by*

$$\begin{cases} B_{n-k} + \displaystyle\sum_{t=2}^{n-k+2} \binom{t+k-3}{k-2} \sum_{j=0}^{t-2} \binom{t-2}{j} B_{n-k-1-j}, & \text{if } k \geq 3 \\ B_{n-1}, & \text{if } k = 2, \\ B_n, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k = 2$. We are interested in the permutations $\pi \in S_n(23{-}1)$ that begin with the pattern 12. It is easy to see that $\pi_1 = 1$, hence $B_{12}^{23-1}(n) = B_{n-1}$ for all $n \geq 2$.

Suppose $k \geq 3$. We recall that $s_{23-1}(n;t)$ is the number of permutations in $S_n(23-1)$ having t as the first letter. By [ClaesMans], $s(n;1) = B_{n-1}$ and for $t \geq 2$, we have

$$s(n;t) = \sum_{j=0}^{t-2} \binom{t-2}{j} B_{n-2-j}.$$

On the other hand, if a permutation $\pi = \pi' 1 t \pi''$ avoids $23-1$ and begins with the pattern $(k-1)(k-2)\ldots 1k$, then π' is decreasing of length $k-2$, and using $s(n;t)$, we get

$$N_{23-1}^{(k-1)(k-2)\ldots 1k}(n) = B_{n-k} + \sum_{t=2}^{n-k+2} \binom{t+k-3}{k-2} \sum_{j=0}^{t-2} \binom{t-2}{j} B_{n-k-1-j}.$$

□

3.7.5 Avoiding $31-2$ and beginning with $(k-1)(k-2)\ldots 1k$

By [Claes], a permutation π avoids the pattern $31-2$ if and only if π avoids the pattern $3-1-2$.

Suppose $\pi = \pi' 1 \pi''$ is an n-permutation avoiding $3-1-2$ and beginning with $(k-1)(k-2)\ldots 1k$. Obviously, π' and π'' avoid $3-1-2$ and each letter of π' is smaller than any letter of π'', since otherwise we have an occurrence of the pattern $3-1-2$ involving the letter 1. Also, π' begins with the pattern $(k-1)(k-2)\ldots 1k$ or $\pi' = (k-1)(k-2)\ldots 2$ and π'' is not empty. So, using the generating function for the number of permutations avoiding the pattern $3-1-2$, which is $C(x)$ ([Knuth]), we get

$$G_{31-2}^{(k-1)(k-2)\ldots 1k}(x) = xG_{31-2}^{(k-1)(k-2)\ldots 1k}(x)C(x) + x^{k-1}(C(x)-1).$$

Therefore, using (3.1), we get the following.

Proposition 16. *We have*

$$G_{31-2}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} x^k C^3(x), & \text{if } k \geq 2, \\ C(x), & \text{if } k = 1. \end{cases}$$

Hence

$$N_{31-2}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} C_{n-k+2} - C_{n-k+1}, & \text{if } k \geq 2, \\ C_n, & \text{if } k = 1. \end{cases}$$

3.7.6 Avoiding $32-1$ and beginning with $(k-1)(k-2)\dots 1k$

Proposition 17.

$$N_{32-1}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} 0, & \text{if } k \geq 4 \\ B_{n-1} - (n-2)B_{n-3}, & \text{if } k = 3 \text{ and } n \geq 3, \\ B_n - (n-1)B_{n-2}, & \text{if } k = 2 \text{ and } n \geq 2, \\ B_n, & \text{if } k = 1. \end{cases}$$

Proof. Using the definitions and [Claes], it is easy to see that the statement is true for $k = 1, 2$ and $k \geq 4$.

Suppose now that $k = 3$ and $\pi = \pi' 1 \pi''$ is an n-permutation avoiding the pattern $32-1$ and beginning with the pattern $(k-1)(k-2)\dots 1k = 213$. We have that π' must be increasing, since otherwise we have an occurrence of the pattern $32-1$ involving the letter 1, and π'' must avoid $32-1$. Moreover, since π begins with 213, the length of π is 1 and the rightmost letter of π'' is greater than the letter of π'. Also, it is easy to see that the number of permutations in $S_{n-1}(32-1)$ beginning with the pattern 12 is the same as the number of permutations in $S_n(32-1)$ beginning with the pattern 213 (one can see it by placing 1 in the second position). Hence $N_{32-1}^{(k-1)\dots 21k}(n) = B_{n-1} - (n-2)B_{n-3}$ for all $n \geq 3$. □

3.8 Avoiding a pattern x-yz and beginning with the pattern $(k-1)(k-2)\dots 1k$ or $23\dots k1$

In this section we consider avoidance of one of the patterns $1-23$, $1-32$, $2-31$, $2-13$, $3-12$ and $1-32$ and beginning with the pattern $(k-1)(k-2)\dots 1k$. The case when a permutation begins with the pattern $23\dots k1$ and avoids a pattern $x-yz$ can be obtained by the complement operation.

Proposition 18. *We have*

$$E_{1-32}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} e^{e^x} \int_0^x e^{-e^t} \displaystyle\sum_{n \geq k-1} \frac{t^n}{n!}\, dt, & \text{if } k \geq 2, \\ e^{e^x - 1}, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n–permutations that avoid the pattern $1-32$ and begin with the pattern $(k-1)(k-2)\dots 1k$. Suppose $\pi = \sigma 1 \tau$ is such a permutation of length $n+1$. Obviously, the letters of τ must be in increasing order, since otherwise we have an occurrence of the pattern $1-32$ in π starting from the letter 1. If $|\sigma| = i$, then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in increasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Also, if $n \geq k-1$, then 1 can be in the $(k-1)$th position, and in this case, since π begins with the pattern $(k-1)(k-2)\dots 1k$, it must be that $\pi = (k-1)(k-2)\dots 21k(k+1)\dots(n+1)$. Thus, in the last case we have only one permutation. This leads to the recurrence relation

$$B_{n+1,k} = \sum_{i \geq 0} \binom{n}{i} B_{i,k} + \delta_{n,k},$$

where

$$\delta_{n,k} = \begin{cases} 1, & \text{if } n \geq k-1, \\ 0, & \text{else.} \end{cases}$$

This recurrence relation is identical to the one given in the proof of Proposition 2, so using this proof we get the desired result. □

Proposition 19. *We have*

$$E_{1-23}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} e^{e^x} \int_0^x \int_0^t \frac{r^{k-2}}{(k-2)!} e^{r-e^t}\, drdt, & \text{if } k \geq 2, \\ e^{e^x - 1}, & \text{if } k = 1. \end{cases}$$

Proof. If $k = 1$, then the statement is true due to Proposition 1.

Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n-permutations that avoid the pattern $1-23$ and begin with the pattern $(k-1)(k-2)\dots 1k$. Suppose $\pi = \sigma 1 \tau$ is such a permutation of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of the pattern $1-23$ in π starting from the letter 1. If $|\sigma| = i$, then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in the decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Besides, if $n \geq k-1$, then 1 can be in the $(k-1)$th position, and in this case, since π begins with the pattern $(k-1)(k-2)\dots 1k$ and τ is decreasing, it must be that the kth letter of π is $(n+1)$ and there are $\binom{n-1}{k-2}$ ways to choose the letters of σ and then write them in decreasing order. Thus,

$$B_{n+1,k} = \sum_{i \geq 0} \binom{n}{i} B_{i,k} + \binom{n-1}{k-2}.$$

Multiplying both sides of the equality with $x^n/n!$ and summing over n, we get the differential equation

$$\frac{d}{dx} E_{1-23}^{(k-1)(k-2)\dots 1k}(x) = E_{1-23}^{(k-1)(k-2)\dots 1k} e^x + \sum_{n \geq 0} \binom{n-1}{k-2} \frac{x^n}{n!},$$

with the initial condition $E_{1-23}^{(k-1)(k-2)\ldots 1k}(0) = 0$. If $F(x)$ denotes the last term, then it is easy to see that $F'(x) = \frac{x^{k-2}}{(k-2)!}e^x$, and thus

$$F(x) = \int_0^x \frac{t^{k-2}}{(k-2)!}e^t \, dt.$$

Now, the solution to the equation above is given by

$$E_{1-23}^{(k-1)(k-2)\ldots 1k}(x) = e^{e^x}\int_0^x e^{-e^t}F(t)\, dt = e^{e^x}\int_0^x\int_0^t \frac{r^{k-2}}{(k-2)!}e^{r-e^t}\, drdt. \tag{3.11}$$

For example, if $k = 2$, then $(k-1)(k-2)\ldots 1k = 12$ and (3.11) gives

$$E_{1-23}^{12} = e^{e^x}\int_0^x e^{-e^t}(e^t - 1)\, dt,$$

which is a particular case of Proposition 3, since the number of n-permutations that avoid the pattern $3-21$ and begin with the pattern 21 is equal to the number of n-permutations that avoid the pattern $1-23$ and begin with the pattern 12 by applying the complement. □

Proposition 20. *We have*

$$G_{2-13}^{(k-1)(k-2)\ldots 1k}(x) = \begin{cases} 0, & \text{if } k \geq 3 \\ x^2C^3(x), & \text{if } k = 2 \\ C(x), & \text{if } k = 1. \end{cases}$$

Hence

$$N_{2-13}^{(k-1)(k-2)\ldots 1k}(n) = \begin{cases} 0, & \text{if } k \geq 3 \\ C_{n-1} - C_{n-2}, & \text{if } k = 2 \\ C_n, & \text{if } k = 1. \end{cases}$$

Proof. For the case $k = 1$, see Proposition 4. If $k \geq 3$, then the statement is true, since in this case the pattern $(k-1)(k-2)\ldots 1k$ does not avoid $2-13$.

Suppose now that $k = 2$. Using the reverse, we see that beginning with the pattern 12 and avoiding $2-13$ is equivalent to ending with the pattern 21 and avoiding $31-2$, which by [Claes] is equivalent to ending with the pattern 21 and avoiding the pattern $3-1-2$.

Let $\pi = \pi'1\pi''$ be an n-permutation avoiding $3-1-2$ and ending with the pattern 21. Obviously, π' and π'' avoid $3-1-2$ and each letter of π' is less than any letter of π'', since otherwise we have an occurrence of $3-1-2$ involving the letter 1. Also, π'' ends with the pattern 21 or $|\pi''| = 1$. So, using the generating function for the number of permutations avoiding $3-1-2$, which is $C(x)$ ([Knuth]), we have

$$G_{2-13}^{12}(x) = xG_{2-13}^{12}(x)C(x) + x(C(x) - 1).$$

Therefore, using (3.1), we get the desired result. □

Proposition 21. *We have*

$$G_{2-31}^{(k-1)(k-2)\dots 1k}(x) = x^k C^2(x).$$

Hence

$$N_{2-31}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} C_{n-(k-1)}, & \text{if } n \geq k \\ 0, & \text{else.} \end{cases}$$

Proof. Using the reverse, we see that beginning with the pattern $(k-1)(k-2)\dots 1k$ and avoiding the pattern $2-31$ is equivalent to ending with the pattern $k12\dots(k-1)$ and avoiding the pattern $13-2$, which, by [Claes], is equivalent to ending with the pattern $k12\dots(k-1)$ and avoiding the pattern $1-3-2$.

Let $\pi = \pi' n \pi''$ be an n-permutation avoiding the pattern $1-3-2$ and ending with the pattern $k12\dots(k-1)$. Obviously, π' and π'' avoid the pattern $1-3-2$ and each letter of π' is greater than any letter of π'', since otherwise we have an occurrence of the pattern $1-3-2$ involving the letter n. Also, π'' ends with the pattern $k12\dots(k-1)$ or $\pi'' = 12\dots(k-1)$.

Using the reverse operation, the generating function for the number of permutations ending with the pattern $k12\dots(k-1)$ and avoiding $1-3-2$ is equal to $G_{2-31}^{(k-1)(k-2)\dots 1k}(x)$. In terms of generating functions, the considerations above lead to

$$G_{2-31}^{(k-1)(k-2)\dots 1k}(x) = xG_{2-31}^{(k-1)(k-2)\dots 1k}(x)C(x) + x^k C(x).$$

Therefore, by (3.1), we get the desired result. □

Proposition 22. *We have*

$$E_{3-12}^{(k-1)(k-2)\dots 1k}(x) = \begin{cases} (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. Suppose $k \geq 2$. Let $B_{n,k}$ denote the number of n-permutations that avoid the pattern $3-12$ and begin with a decreasing subword of length k. Let $\pi = \sigma(n+1)\tau$ be such a permutation of length $n+1$. Obviously, the letters of τ must be in decreasing order since otherwise we have an occurrence of $3-12$ in π starting from the letter (n+1). If $|\sigma| = i$ then we can choose the letters of σ in $\binom{n}{i}$ ways. Since the letters of τ are in decreasing order, they do not affect σ and thus there are $B_{i,k}$ possibilities to choose σ. Also, if $|\sigma| = k-1$ and the letters of σ are in decreasing order, we get $\binom{n}{k-1}$ additional ways to choose π. Thus

$$B_{n+1,k} = \sum_{i\geq 0}\binom{n}{i}B_{i,k} + \binom{n}{k-1}.$$

This recurrence relation is identical to the one given in the proof of Proposition 1, and we get the desired result using that proof. □

Proposition 23.

$$E_{3-21}^{(k-1)(k-2)\dots 1k}(n) = \begin{cases} 0, & \text{if } k \geq 4 \\ (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k = 2 \text{ or } k = 3, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof. If $k \geq 4$ then the statement is true, since in this case the pattern $(k-1)(k-2)\dots 1k$ does not avoid the pattern $3-21$. In the other cases, we use the same arguments as we have in the proof of Proposition 22. The only difference is that instead of decreasing order in τ, we have increasing order. □

3.9 Conclusions

The goal of our paper is to give a complete description for the numbers of permutations avoiding a pattern of the form $x-yz$ or $xy-z$ and either beginning with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $23\dots k1$, $(k-1)(k-2)\dots 1k$, or ending with one of the patterns $12\dots k$, $k(k-1)\dots 1$, $1k(k-1)\dots 2$, $k12\dots(k-1)$. This description is given in Sections 5–8. However, some of our results can be generalized to beginning with a pattern belonging to Γ_k^{min} or Γ_k^{max}, and thus to the ending with a pattern belonging to Δ_k^{min} or Δ_k^{max} (see Section 3.2 for definitions). An example of such a generalisation is given in Theorem 3 below. This theorem generalizes Propositions 1 and 22 and can be proved by using the same considerations as we do in the proofs of these propositions.

Theorem 3. *Suppose* $p_1, p_2 \in \Gamma_k^{min}$ *and* $p_1 \in S_k(1-23)$, $p_2 \in S_k(1-32)$. *Thus, the complements* $C(p_1), C(p_2) \in \Gamma_k^{max}$ *and* $C(p_1) \in S_k(1-23)$, $C(p_2) \in S_k(3-12)$. *Then, we have*

$$E_{1-23}^{p_1}(x) = E_{3-21}^{C(p_1)}(x) = E_{1-32}^{p_2}(x) = E_{3-12}^{C(p_2)}(x) =$$

$$\begin{cases} (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Bibliography

[BabStein] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, *Séminaire Lotharingien de Combinatoire*, B44b:18pp, 2000.

[Bon] M. Bóna: Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A* **80** (1997), no. 2, 257–272.

[B] M. Bóna: The permutation classes equinumerous to the smooth class. *Electron. J. Combin. 5* (1998), no. **1**, Research Paper 31, 12 pp. (electronic).

[CW] T. Chow and J. West: Forbidden subsequences and Chebyshev polynomials. *Discrete Math.* **204** (1999), no. 1-3, 119–128.

[Claes] A. Claesson: Generalised Pattern Avoidance, *European J. Combin.* **22** (2001), 961-971.

[ClaesMans] A. Claesson and T. Mansour: Permutations avoiding a pair of generalized patterns of length three with exactly one dash, preprint CO/0107044.

[ElizNoy] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, *Proceedings of FPSAC 2001.*

[Ent] R. Entringer: A Combinatorial Interpretation of the Euler and Bernoulli Numbers, *Nieuw. Arch. Wisk.* **14** (1966), 241–246.

[Kit1] S. Kitaev: Multi-avoidance of generalised patterns, *Discrete Math.*, to appear (2002).

[Kit2] S. Kitaev: Partially ordered generalized patterns, *Discrete Math.*, to appear (2002).

[Kit3] S. Kitaev: Generalized pattern avoidance, preprint.
`http://www.math.chalmers.se/~kitaev/papers.html`

[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Kr] C. Krattenthaler: Permutations with restricted patterns and Dyck paths, *Adv. in Appl. Math.* **27** (2001), 510–530.

[K] D. Kremer: Permutations with forbidden subsequnces and a generalized Schröder number, *Discrete Math.* **218** (2000), 121–130.

[Loth] M. Lothaire: *Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, **17**, Addison-Wesley Publishing Co., Reading, Mass. (1983).

[Mans1] T. Mansour: Continued fractions and generalized patterns, *European J. Combin.*, to appear (2002), math.CO/0110037.

[Mans2] T. Mansour: Continued fractions, statistics, and generalized patterns, to appear in Ars Combinatorica (2002), preprint CO/0110040.

[Mans3] T. Mansour: Restricted 1-3-2 permutations and generalized patterns, *Annals of Combinatorics* **6** (2002), 1–12.

[MV1] T. Mansour and A. Vainshtein: Restricted permutations, continued fractions, and Chebyshev polynomials, *Electron. J. Combin.* **7** (2000) no. 1, Research Paper 17, 9 pp. (electronic).

[MV2] T. Mansour and A. Vainshtein: Restricted 132-avoiding permutations, *Adv. in Appl. Math.* **126** (2001), no. 3, 258–269.

[MV3] T. Mansour and A. Vainshtein: Layered restrictions and Chebyshev polynomials, *Annals of Combinatorics* **5** (2001), 451–458.

[MV4] T. Mansour and A. Vainshtein: Restricted permutations and Chebyshev polyonomials, *Séminaire Lotharingien de Combinatoire* **47** (2002), Article B47c.

[R] A. Robertson: Permutations containing and avoiding 123 and 132 patterns, *Discrete Math. Theor. Comput. Sci.* **3** (1999), no. 4, 151–154 (electronic).

[RWZ] A. Robertson, H. Wilf, and D. Zeilberger: Permutation patterns and continued fractions, *Electron. J. Combin.* **6** (1999), no. 1, Research Paper 38, 6 pp. (electronic).

[SloPlo] N. J. A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, (1995). `http://www.research.att.com/~njas/sequences/`

[Stan] R. Stanley: *Enumerative Combinatorics*, Vol. **1**, Cambridge University Press, (1997).

[SchSim] R. Simion, F. Schmidt: Restricted permutations, *European J. Combin.* **6** (1985), no. 4, 383–406.

[W] J. West: Generating trees and forbidden subsequences, *Discrete Math.* **157** (1996), 363–372.

Paper IV

On multi-avoidance of generalized patterns

On multi-avoidance of generalized patterns

Sergey Kitaev and Toufik Mansour [1]

Matematik, Chalmers tekniska högskola och Göteborgs universitet,
S-412 96 Göteborg, Sweden
kitaev@math.chalmers.se, toufik@math.chalmers.se

Abstract

In [Kit1] Kitaev discussed simultaneous avoidance of two 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. In three essentially different cases, the numbers of such n-permutations are 2^{n-1}, the number of involutions in $\mathcal{S}_n$, and $2E_n$, where E_n is the n-th Euler number. In this paper we give recurrence relations for the remaining three essentially different cases.

To complete the descriptions in [Kit3] and [KitMans], we consider avoidance of a pattern of the form x-y-z (a classical 3-pattern) and beginning or ending with an increasing or decreasing pattern. Moreover, we generalize this problem: we demand that a permutation must avoid a 3-pattern, begin with a certain pattern and end with a certain pattern simultaneously. We find the number of such permutations in case of avoiding an arbitrary generalized 3-pattern and beginning and ending with increasing or decreasing patterns.

4.1 Introduction and Background

Permutation patterns: All permutations in this paper are written as words $\pi = a_1 a_2 \dots a_n$, where the a_i consist of all the integers $1, 2, \dots, n$. Let $\alpha \in S_n$ and $\tau \in S_k$ be two permutations. We say that α *contains* τ if there exists a subsequence $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ such that $(\alpha_{i_1}, \dots, \alpha_{i_k})$ is order-isomorphic to τ; in such a context τ is usually called a *pattern*. We say that α *avoids* τ, or is τ-*avoiding*, if such a subsequence does not exist. The set of all τ-avoiding permutations in S_n is denoted by $S_n(\tau)$. For an arbitrary finite collection of patterns T, we say that α avoids T if α avoids any $\tau \in T$; the corresponding subset of S_n is denoted by $S_n(T)$.

While the case of permutations avoiding a single pattern has attracted much attention, the case of multiple pattern avoidance remains less investigated. In particular, it is natural, as the next step, to consider permutations avoiding pairs of patterns τ_1, τ_2. This problem was solved completely for $\tau_1, \tau_2 \in S_3$ (see [SchSim]), for $\tau_1 \in S_3$ and $\tau_2 \in S_4$ (see [W]), and for $\tau_1, \tau_2 \in S_4$ (see [B, K] and references therein). Several recent papers [CW, MV1, Kr, MV3, MV2] deal with the case $\tau_1 \in S_3$, $\tau_2 \in S_k$ for various pairs τ_1, τ_2.

[1] Research financed by EC's IHRP Programme, within the Research Training Network "Algebraic Combinatorics in Europe", grant HPRN-CT-2001-00272

Generalized permutation patterns: In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns* (*GPs*) where two adjacent letters in a pattern may be required to be adjacent in the permutation. Such an adjacency requirement is indicated by the absence of a dash between the corresponding letters in the pattern. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, but the pattern 2-3-1 occurs also in the subwords 562 and 563. Note that a classical pattern should, in our notation, have dashes at the beginning and end. Since most of the patterns considered in this paper satisfy this, we suppress these dashes from the notation. Thus, a pattern with no dashes corresponds to a contiguous subword anywhere in a permutation. The motivation for introducing these patterns was the study of Mahonian statistics. A number of results on GPs were obtained by Claesson, Kitaev and Mansour. See for example [Claes], [Kit1, Kit2, Kit3] and [Mans1, Mans2, Mans3].

As in [SchSim], dealing with the classical patterns, one can consider the case when permutations have to avoid two or more generalized patterns simultaneously. A complete solution for the number of permutations avoiding a pair of 3-patterns of type (1,2) or (2,1), that is the patterns having one internal dash, is given in [ClaesMans1]. In [Kit1] Kitaev discussed simultaneous avoidance of two 3-patterns with no internal dashes, that is, where the patterns correspond to contiguous subwords in a permutation. In three essentially different cases, the numbers of such n-permutations are 2^{n-1}, the number of involutions in $\mathcal{S}_n$, and $2E_n$, where E_n is the n-th Euler number. The remaining cases are avoidance of 123 and 231, 213 and 231, 132 and 213. In Section 4.3 we give recurrence relations for these cases.

In Section 4, we consider avoidance of a pattern x-y-z, and beginning or ending with increasing or decreasing pattern. This completes the results made in [KitMans], which concerns the number of permutations that avoid a generalized 3-pattern and begin or end with an increasing or decreasing pattern.

In Sections 5–8, we give enumeration for the number of permutations that avoid a generalazed 3-pattern, begin *and* end with increasing or decreasing patterns. We record our results in terms of either *generating functions*, or *exponential generating functions*, or formulas for the numbers appeared.

In Section 4.9, we discuss possible directions of generalization of the results from Sections 5–8.

4.2 Preliminaries

The *reverse* $R(\pi)$ of a permutation $\pi = a_1a_2\dots a_n$ is the permutation $a_n\dots a_2a_1$. The *complement* $C(\pi)$ is the permutation $b_1b_2\dots b_n$ where $b_i = n+1-a_i$. Also, $R\circ C$ is the composition of R and C. For example, $R(13254) = 45231$, $C(13254) = 53412$ and $R\circ C(13254) = 21435$. We call these bijections of S_n to itself *trivial*, and it is easy to see that for any pattern p the number $A_p(n)$ of permutations avoiding the pattern p is the same as for the patterns $R(p)$, $C(p)$ and $R\circ C(p)$. For example, the number of permutations that avoid the pattern

132 is the same as the number of permutations that avoid the pattern 231. This property holds for sets of patterns as well. If we apply one of the trivial bijections to all patterns of a set G, then we get a set G' for which $A_{G'}(n)$ is equal to $A_G(n)$. For example, the number of permutations avoiding $\{123, 132\}$ equals the number of those avoiding $\{321, 312\}$ because the second set is obtained from the first one by complementing each pattern.

In this paper we denote the nth Catalan number by C_n; the generating function for these numbers by $C(x)$; the nth Bell number by B_n.

Also, $N_p^q(n)$ denotes the number of permutations that avoid the pattern p and begin with the pattern q; $G_p^q(x)$ (respectively, $E_p^q(x)$) denotes the ordinary (respectively, exponential) generating function for the number of such permutations. Besides, $N_p^{q,r}(n)$ denotes the number of permutations that avoid the pattern p, begin with the pattern q and end with the pattern r; $G_p^{q,r}(x)$ (respectively, $E_p^{q,r}(x)$) denotes the ordinary (respectively, exponential) generating function for the number of such permutations.

Recall the following properties of $C(x)$:

$$C(x) = \frac{1 - \sqrt{1-4x}}{2x} = \frac{1}{1 - xC(x)}. \tag{4.1}$$

4.3 Simultaneous avoidance of two 3-patterns with no dashes

4.3.1 Avoidance of patterns 123 and 231 simultaneously

We first consider the avoidance of the patterns 123 and 231 simultaneously.

Let $a(n; i_1, i_2, \ldots, i_m)$ denote the number of permutations $\pi \in S_n(123, 231)$ such that $\pi_1\pi_2 \ldots \pi_m = i_1 i_2 \ldots i_m$ and let $a(n) = |S_n(123, 231)|$.

By the definitions, we get that $a(n) = \sum_{j=1}^{n} a(n; j)$ and $a(n; n) = a(n-1)$. Hence

$$a(n) = a(n-1) + a(n;1) + a(n;2) + \cdots + a(n; n-1). \tag{4.2}$$

Also, by the definitions, for all $1 \leq i \leq n-1$, we get

$$a(n; i) = \sum_{j=1}^{i-1} a(n; i, j) + \sum_{j=i+1}^{n} a(n; i, j). \tag{4.3}$$

Suppose $\pi \in S_n(123, 231)$ is such that $\pi_1 = i$ and $\pi_2 = j$. If $i > j$ then there is no occurrence of the pattern 123 or 231 that contains π_1, so $a(n; i, j) = a(n-1; j)$. If $i < j$ then since π avoids 123 and 231, we get that $i < \pi_3 < j$, and thus in this case $a(n; i, j) = a(n-2; i) + a(n-2; i+1) + \cdots + a(n-2; j-2)$.

Hence, using (4.2) and (4.3), we get the following theorem.

Proposition 1. *Let $s_n = |S_n(123, 231)|$. For all $n \geq 3$,*

$$s_n = s_{n-1} + s_n(1) + s_n(2) + \cdots + s_n(n-1),$$

where for all $1 \leq i \leq n$,

$$s_n(i) = \sum_{j=1}^{i-1} s_{n-1}(j) + \sum_{j=i}^{n-2} (n-1-j)s_{n-2}(j),$$

and $s_3(1) = 1$, $s_3(2) = 1$, $s_3(3) = 2$.

Using this theorem, we get quickly the first values of the sequence $|S_n(123, 231)|$ for $n = 0, 1, 2, \ldots, 10$:

n	0	1	2	3	4	5	6	7	8	9	10
$\lvert S_n(123,231)\rvert$	1	1	2	4	11	39	161	784	4368	27260	189540

4.3.2 Avoidance of patterns 132 and 213 simultaneously

We consider avoidance of the patterns 132 and 213 simultaneously.

Let $b(n; i_1, i_2, \ldots, i_m)$ denote the number of permutations $\pi \in S_n(132, 213)$ such that $\pi_1\pi_2 \ldots \pi_m = i_1 i_2 \ldots i_m$ and let $b(n) = |S_n(132, 213)|$.

Suppose $\pi \in S_n(132, 213)$ is such that $\pi_1 = i$ and $\pi_2 = j$. If $i > j$ then, since π avoids 213, we get $\pi_3 \leq i - 1$. Thus

$$b(n; i, j) = \sum_{k=1,\ k \neq j}^{i-1} b(n-1; j, k). \tag{4.4}$$

If $i < j$ then, since π avoids 132, we get $\pi_3 \leq i - 1$ or $\pi_3 \geq j + 1$. Thus

$$b(n; i, j) = \sum_{k=1}^{i-1} b(n-1; j-1, k) + \sum_{k=j}^{n-1} b(n-1; j-1, k). \tag{4.5}$$

Using (4.4) and (4.5), we get the following theorem.

Proposition 2. *Let* $s_n = |S_n(132, 213)|$. *Then* $s_n = \sum_{i,j=1}^{n} s(n; i, j)$ *with*
$s(n; i, i) = 0$ *for all* $n, i \geq 1$;
$s(n; i, j) = \sum_{k=1}^{i-1} s(n-1; j, k)$ *if* $i > j$;
$s(n; i, j) = \sum_{k=1}^{i-1} s(n-1; j-1, k) + \sum_{k=j}^{n-1} s(n-1; j-1, k)$ *if* $i < j$;
and $s(2; 1, 2) = s(2; 2, 1) = 1$, $s(2; 1, 1) = s(2; 1, 1) = 0$.

Using this theorem, we get

n	0	1	2	3	4	5	6	7	8	9	10
$\lvert S_n(132,213)\rvert$	1	1	2	4	11	37	149	705	3814	23199	156940

4.3.3 Avoidance of the patterns 213 and 231 simultaneously

We now consider avoidance of the patterns 213 and 231 simultaneously. This case is equivalent to avoidance of the patterns 132 and 312 by applying the reverse operation.

Let $c(n; i_1, i_2, \ldots, i_m)$ denote the number of permutations $\pi \in S_n(132, 312)$ such that $\pi_1\pi_2 \ldots \pi_m = i_1 i_2 \ldots i_m$ and let $c(n) = |S_n(132, 312)|$. We proceed as in the previous case. For $n \geq i > j \geq 1$, we have

$$c(n; i, j) = \sum_{k=1}^{j-1} c(n-1; j, k) + \sum_{k=i}^{n-1} c(n-1; j, k). \tag{4.6}$$

For $1 \leq i < j \leq n$, we have

$$c(n; i, j) = \sum_{k=1}^{i-1} c(n-1; j-1, k) + \sum_{k=j}^{n-1} c(n-1; j-1, k). \tag{4.7}$$

Using (4.6) and (4.7), we get the following theorem.

Proposition 3. *Let $s_n = |S_n(132, 312)|$. Then $s_n = \sum_{i,j=1}^{n} s(n; i, j)$ with*
$s(n; i, i) = 0$ for all $n, i \geq 1$;
$s(n; i, j) = \sum_{k=1}^{j-1} s(n-1; j, k) + \sum_{k=i}^{n-1} s(n-1; j, k)$ if $i > j$;
$s(n; i, j) = \sum_{k=1}^{i-1} s(n-1; j-1, k) + \sum_{k=j}^{n-1} s(n-1; j-1, k)$ if $i < j$;
and $s(2; 1, 2) = s(2; 2, 1) = 1$, $s(2; 1, 1) = s(2; 1, 1) = 0$.

Using this theorem, we get

n	0	1	2	3	4	5	6	7	8	9	10
$\lvert S_n(132, 312)\rvert$	1	1	2	4	10	30	108	454	2186	11840	71254

4.4 Avoiding a pattern x-y-z and beginning or ending with certain patterns

Recall the definitions of $G_q^p(x)$, $N_q^p(n)$, $C(x)$ and C_n in Section 4.2.

Proposition 4. *We have*

$$G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C^2(x).$$

Proof. Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 < \pi_2 < \cdots < \pi_k$ and $\pi_j = n$. It is easy to see that π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \ldots, n$, and $\pi'' \in S_{n-j}(1\text{-}3\text{-}2)$. If we now consider two cases, namely $j = k$ and $j \geq k+1$, we get

$$G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C(x) + x G_{1\text{-}3\text{-}2}^{12\ldots k}(x) C(x).$$

Thus, $G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C(x)/(1 - xC(x))$ and, using (4.1), we get the desired result. □

Proposition 5. *We have*

$$G_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x) = x^k C^{k+1}(x).$$

Proof. Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 > \pi_2 > \cdots > \pi_k$ and $\pi_j = n$. It is easy to see that π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \dots, n$, and $\pi'' \in S_{n-j}(1\text{-}3\text{-}2)$. If we consider separately the cases $j = 1$ and $j \geq 2$, we get

$$G_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x) = xG_{1\text{-}3\text{-}2}^{(k-1)(k-2)\dots1}(x) + xG_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x)C(x).$$

Hence,

$$G_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x) = xG_{1\text{-}3\text{-}2}^{(k-1)(k-2)\dots1}(x)/(1-xC(x))$$

and, using (4.1), we get $G_{1\text{-}3\text{-}2}^{k(k-1)\dots1}(x) = xC(x)G_{1\text{-}3\text{-}2}^{(k-1)(k-2)\dots1}(x)$. By induction on k, using the fact that $G_{1\text{-}3\text{-}2}^{1}(x) = C(x) - 1 = xC^2(x)$, we get the desired result. □

Proposition 6. *We have*

$$G_{2\text{-}1\text{-}3}^{12\dots k}(x) = x^k C^{k+1}(x).$$

Proof. One can use the same considerations as we have in the proof of Proposition 5, by considering a permutation $\pi = \pi' 1 \pi'' \in S_n(2\text{-}1\text{-}3)$ such that $\pi_1 < \pi_2 < \cdots < \pi_k$ and $\pi_j = 1$. □

Proposition 7. *We have*

$$G_{2\text{-}1\text{-}3}^{k(k-1)\dots1}(x) = x^k C^2(x).$$

Proof. One can use the same considerations as we have in the proof of Proposition 4, by considering a permutation $\pi = \pi' 1 \pi'' \in S_n(2\text{-}1\text{-}3)$ such that $\pi_1 > \pi_2 > \cdots > \pi_k$ and $\pi_j = 1$. □

Let $s_n(i_1, \dots, i_m)$ denote the number of permutations $\pi \in S_n(1\text{-}2\text{-}3)$ such that $\pi_1 \pi_2 \dots \pi_m = i_1 i_2 \dots i_m$. It is easy to see that

$$s_n(n) = s_n(n-1) = C_{n-1}, \tag{4.8}$$

and

$$s_n(t) = s_n(t,n) + \sum_{j=1}^{t-1} s_n(t,j) = s_{n-1}(t) + \sum_{j=1}^{t-1} s_{n-1}(j). \tag{4.9}$$

Now, (4.8) and (4.9) with induction on t give

$$s_n(n-t) = \sum_{j=0}^{t} (-1)^j \binom{t-j}{j} C_{n-j} \tag{4.10}$$

Let us prove the following proposition.

Proposition 8. *We have*

$$G_{1\text{-}2\text{-}3}^{12\ldots k}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ x^2C^2(x), & \text{if } k = 2, \\ xC^2(x), & \text{if } k = 1\ . \end{cases}$$

Proof. For $k \geq 3$, the statement is obviously true. If $k = 1$ then $G_{1\text{-}2\text{-}3}^{1}(x) = C(x) - 1 = xC^2(x)$.

Suppose now that $k = 2$. From the definitions, for all $n \geq 2$, we have

$$N_{1\text{-}2\text{-}3}^{12}(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s_n(i, j).$$

In this formula, j can only be equal to n, since otherwise we have an occurrence of the pattern 1-2-3. Using this fact with (4.8) and (4.9), we get for $n \geq 2$,

$$N_{1\text{-}2\text{-}3}^{12}(n) = \sum_{i=1}^{n-1} s_n(i, n) = \sum_{i=1}^{n-1} s_{n-1}(i) = C_{n-1}.$$

Hence, $G_{1\text{-}2\text{-}3}^{12}(x) = x(C(x) - 1) = x^2C^2(x)$. □

Proposition 9. *We have*

$$N_{1\text{-}2\text{-}3}^{k(k-1)\ldots1}(n) = \sum_{t=1}^{n+1-k} \binom{n-t}{k-1} \sum_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j}.$$

Proof. From the definitions, we have

$$N_{1\text{-}2\text{-}3}^{k(k-1)\ldots1}(n) = \sum_{i_1=k}^{n} \sum_{i_2=1}^{i_1-1} \cdots \sum_{i_k=1}^{i_{k-1}-1} s_n(i_1, \ldots, i_k) = \sum_{t=1}^{n+1-k} \binom{n-t}{k-1} s_n(t).$$

Using (4.10), we get

$$N_{1\text{-}2\text{-}3}^{k(k-1)\ldots1}(n) = \sum_{t=1}^{n+1-k} \binom{n-t}{k-1} \sum_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j}.$$

□

4.5 Avoiding a pattern x-y-z, beginning and ending with certain patterns simultaneously

Recall the definitions of $G_q^{p,r}(x)$ and $N_q^{p,r}(n)$ in Section 4.2.

Proposition 10. *We have*

(i) $G_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots\ell}(x) = x^{k+\ell-1}C^{\ell+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(ii) $G_{1\text{-}3\text{-}2}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) = x^{k+\ell-1}C^2(x)$.

(iii) $G_{1\text{-}3\text{-}2}^{k(k-1)\ldots 1,\ell(\ell-1)\ldots 1}(x) = x^{k+\ell-1}C^{k+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$, *where* $m = \max(k,\ell)$.

(iv) *the generating function* $G_{1\text{-}3\text{-}2}(x,y,z) = \sum_{k,\ell\geq 0} G_{1\text{-}3\text{-}2}^{k(k-1)\ldots 1,12\ldots\ell}(x)y^k z^\ell$ *for the sequence* $\{G_{1\text{-}3\text{-}2}^{k(k-1)\ldots 1,12\ldots\ell}(x)\}_{k,\ell\geq 0}$ (*where* k *and* ℓ *go through all natural numbers*) *is*

$$\frac{1}{1-x(y+z)}\left(x(y+z+yz) + \frac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right).$$

Proof.

Beginning with $12\ldots k$ and ending with $\ell(\ell-1)\ldots 1$: Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 < \pi_2 < \cdots < \pi_k$, $\pi_n < \pi_{n-1} < \cdots < \pi_{n-\ell+1}$ and $\pi_j = n$. It is easy to see that π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \ldots, n$, and $\pi'' \in S_{n-j}(1\text{-}3\text{-}2)$. We now consider three cases, namely $j = k$, $k+1 \leq j \leq n-\ell$ and $j = n-\ell+1$. In terms of generating functions, we have

$$G_{1\text{-}3\text{-}2}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) = x^k G_{2\text{-}1\text{-}3}^{\ell(\ell-1)\ldots 1}(x) + xG_{1\text{-}3\text{-}2}^{12\ldots k}(x)G_{2\text{-}1\text{-}3}^{\ell(\ell-1)\ldots 1}(x) + x^\ell G_{1\text{-}3\text{-}2}^{12\ldots k}(x) + x^{k+\ell-1},$$

where we observed that to avoid 1-3-2 and end with $\ell(\ell-1)\ldots 1$ is the same as to avoid 2-1-3 and begin with $\ell(\ell-1)\ldots 1$ by applying the reverse and complement operations. Also, we added the term $x^{k+\ell-1}$, since when $j = k = n-\ell+1$, we have one "good" $(k+\ell-1)$-permutation, which is not counted by our three cases.

From Propositions 4 and 7, we have that

$$G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C^2(x) \text{ and } G_{2\text{-}1\text{-}3}^{\ell(\ell-1)\ldots 1}(x) = x^\ell C^2(x).$$

Thus, using the fact that $xC^2(x) = C(x) - 1$, we get

$$\begin{aligned} G_{1\text{-}3\text{-}2}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) &= x^{k+\ell}C^2(x)(2 + xC^2(x)) + x^{k+\ell-1} \\ &= x^{k+\ell-1}(C(x)-1)(C(x)+1) + x^{k+\ell-1} = x^{k+\ell-1}C^2(x). \end{aligned}$$

Beginning with $12\ldots k$ and ending with $12\ldots\ell$: Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 < \pi_2 < \cdots < \pi_k$, $\pi_n > \pi_{n-1} > \cdots > \pi_{n-\ell+1}$ and $\pi_j = n$. As above, π avoids 1-3-2 if and only if π' is a 1-3-2-avoiding permutation on the letters $n-j+1, n-j+2, \ldots, n$, and $\pi'' \in S_{n-j}(1\text{-}3\text{-}2)$. We consider the cases $j = k$, $k+1 \leq j \leq n-\ell$ and $j = n$. In terms of generating functions, the first approximation for the function $G_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots\ell}(x)$ is

$$G_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots\ell}(x) \approx x^k G_{2\text{-}1\text{-}3}^{12\ldots\ell}(x) + xG_{1\text{-}3\text{-}2}^{12\ldots k}(x)G_{2\text{-}1\text{-}3}^{12\ldots\ell}(x) + xG_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots(\ell-1)}(x),$$

where we observed that to avoid 1-3-2 and end with $12\ldots\ell$ is the same as to avoid 2-1-3 and begin with $12\ldots\ell$ by applying the reverse and complement

operations. We use the sign "$\approx$" because there are some "good" permutations, which are not counted by our considerations. We discuss them below.

From Propositions 4 and 6, we have that $G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C^2(x)$ and $G_{2\text{-}1\text{-}3}^{12\ldots\ell}(x) = x^\ell C^{\ell+1}(x)$. Thus, using the fact that $xC^2(x) = C(x) - 1$ and $G_{1\text{-}3\text{-}2}^{12\ldots k,1}(x) = G_{1\text{-}3\text{-}2}^{12\ldots k}(x) = x^k C^2(x)$ (Proposition 4), we get

$$
\begin{aligned}
&G_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots\ell}(x) \\
&\quad \approx x^{k+\ell}C^{\ell+1}(x) + x^{k+\ell+1}C^{\ell+3}(x) + xG_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots(\ell-1)}(x) \\
&\quad = x^{k+\ell}C^{\ell+2}(x) + xG_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots(\ell-1)}(x) \\
&\quad = x^{k+\ell}C^{\ell+2}(x) + x^{k+\ell}C^{\ell+1}(x) + x^2G_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots(\ell-2)}(x) \\
&\quad = \cdots = x^{k+\ell}C^4(x)(C^{\ell-2}(x) + C^{\ell-3}(x) + \cdots + 1) + x^{k+\ell-1}C^2(x) \\
&\quad = x^{k+\ell-1}(C(x)-1)C^2(x)\tfrac{1-C^{\ell-1}(x)}{1-C(x)} + x^{k+\ell-1}C^2(x) = x^{k+\ell-1}C^{\ell+1}(x).
\end{aligned}
$$

To complete the proof of this case, we observe that in our considerations above, we do not count increasing permutations of length $m = \max(k,\ell), m+1, \ldots, k+\ell-2$, which satisfy all our restrictions. We did not count them because the k-beginning and ℓ-ending in these permutations overlap in more than one letter. So, to get the desired result, we need to add the term

$$x^m + x^{m+1} + \ldots + x^{k+\ell-2} = (x^m - x^{k+\ell-1})/(1-x)$$

to the approximate value of $G_{1\text{-}3\text{-}2}^{12\ldots k,12\ldots\ell}(x)$. For example, expanding $G_{1\text{-}3\text{-}2}^{12,123}(x)$, we have, in particular, that there are 2002 10-permutations that avoid 1-3-2, begin with the pattern 12 and end with the pattern 123.

Beginning with $k(k-1)\ldots1$ and ending with $\ell(\ell-1)\ldots1$: If $\ell = 1$ then, by Proposition 5, $G_{1\text{-}3\text{-}2}^{k(k-1)\ldots1,1}(x) = x^k C^{k+1}(x)$. Suppose $\ell \geq 2$, and $\pi = \pi' 1 \pi'' \in S_n(1\text{-}3\text{-}2)$ is such that $\pi_1 > \pi_2 > \cdots > \pi_k$, $\pi_n < \pi_{n-1} < \cdots < \pi_{n-\ell+1}$ and $\pi_j = 1$. Obviously, π'' is the empty word, since otherwise we have an occurrence of the pattern 1-3-2 starting from the letter 1. Thus, the first approximation for the function $G_{1\text{-}3\text{-}2}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}$ is

$$G_{1\text{-}3\text{-}2}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) \approx xG_{1\text{-}3\text{-}2}^{k(k-1)\ldots1,(\ell-1)(\ell-2)\ldots1}(x) = \cdots = x^{k+\ell-1}C^{k+1}(x).$$

Like in the previous case, we did not count decreasing permutations of length $m = max(k,\ell)$, $m+1, \ldots, k+\ell-2$, which satisfy all our restrictions. Thus, to get the desired result, we add the term $(x^m - x^{k+\ell-1})/(1-x)$ to the approximate value of $G_{1\text{-}3\text{-}2}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$.

Beginning with $k(k-1)\ldots1$ and ending with $12\ldots\ell$: Suppose $\pi = \pi' n \pi'' \in S_n(1\text{-}3\text{-}2)$. Any letter of π' is greater than any letter of π'', since otherwise we have an occurrence of the pattern 1-3-2 in π containing the letter

n which is forbidden. Also, π' and π'' avoid 1-3-2. If π begins with $k(k-1)\dots 1$, ends with $12\dots\ell$ and π' and π'' are not empty, then π' must begin with $k(k-1)\dots 1$ and π'' must end with $12\dots\ell$. If π' is empty then π'' must begin with $(k-1)(k-2)\dots 1$ and end with $12\dots\ell$. If π'' is empty then π' must begin with $k(k-1)\dots 1$ and end with $12\dots(\ell-1)$. In terms of generating functions, the discussion above leads to the following:

$$G_{1\text{-}3\text{-}2}^{k(k-1)\dots 1,12\dots\ell}(x) \approx$$

$$xG_{1\text{-}3\text{-}2}^{k(k-1)\dots 1}(x)G_{2\text{-}1\text{-}3}^{12\dots\ell}(x) + xG_{1\text{-}3\text{-}2}^{(k-1)\dots 1,12\dots\ell}(x) + xG_{1\text{-}3\text{-}2}^{k(k-1)\dots 1,12\dots(\ell-1)}(x),$$

where we observed that to avoid 1-3-2 and end with $12\dots\ell$ is the same as to avoid 2-1-3 and begin with $12\dots\ell$. However, to put the sign "=" instead of "$\approx$", we have to correct the right-hand side of the recurrence relation by observing that when either $k=1$ and $\ell=0$, or $k=0$ and $\ell=1$, or $k=1$ and $\ell=1$, the formula do not count the permutation $\pi=1$ which satisfies all the conditions needed. Thus, if we make correction of the right-hand side, then multiply both parts of the obtained equality by $x^k y^\ell$ and sum over all natural k and ℓ we get (recall the definition of $G_{1\text{-}3\text{-}2}(x,y,z)$ in the statement of the theorem):

$$G_{1\text{-}3\text{-}2}(x,y,z) = x\sum_{k,\ell\geq 0} G_{1\text{-}3\text{-}2}^{k(k-1)\dots 1}(x)G_{2\text{-}1\text{-}3}^{12\dots\ell}(x)y^k z^\ell + x(y+z)G_{1\text{-}3\text{-}2}(x,y,z) + x(y+z+yz).$$

From Propositions 5 and 6, $G_{1\text{-}3\text{-}2}^{k(k-1)\dots 1}(x)G_{2\text{-}1\text{-}3}^{12\dots\ell}(x) = x^{k+\ell}C^{k+\ell+2}(x)$, and thus

$$\begin{aligned} G_{1\text{-}3\text{-}2}(x,y,z) &= \tfrac{1}{1-x(y+z)}\left(x(y+z+yz) + \sum_{k,\ell\geq 0} x^{k+\ell}C^{k+\ell+2}(x)y^k z^\ell\right) \\ &= \tfrac{1}{1-x(y+z)}\left(x(y+z+yz) + zC^2(z)\sum_{k\geq 0}(xyC(x))^k \sum_{\ell\geq 0}(xzC(x))^\ell\right) \\ &= \tfrac{1}{1-x(y+z)}\left(x(y+z+yz) + \tfrac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right), \end{aligned}$$

where we used that $xC^2(x) = C(x)-1$. □

Proposition 11. *We have*

(i) $G_{2\text{-}1\text{-}3}^{12\dots k,12\dots\ell}(x) = x^{k+\ell-1}C^{k+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(ii) $G_{2-1-3}^{k(k-1)\dots 1,12\dots\ell}(x) = x^{k+\ell-1}C^2(x)$.

(iii) $G_{2\text{-}1\text{-}3}^{k(k-1)\dots 1,\ell(\ell-1)\dots 1}(x) = x^{k+\ell-1}C^{\ell+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$, *where* $m = \max(k,\ell)$.

(iv) *the generating function* $G_{2\text{-}1\text{-}3}(x,y,z) = \sum_{k,\ell\geq 0} G_{2\text{-}1\text{-}3}^{12\dots k,\ell(\ell-1)\dots 1}(x)y^k z^\ell$ *for the sequence* $\{G_{2\text{-}1\text{-}3}^{12\dots k,\ell(\ell-1)\dots 1}(x)\}_{k,\ell\geq 0}$ (*where* k *and* ℓ *go through all natural numbers*) *is*

$$\frac{1}{1-x(y+z)}\left(x(y+z+yz) + \frac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right).$$

Proof. We apply the reverse and complement operations and then use the results of Proposition 10. For example, to avoid 2-1-3, begin with $12\dots k$ and end with $12\dots\ell$ is the same as to avoid 1-3-2, begin with $12\dots\ell$ and end with $12\dots k$. □

Let $h_n^{k,\ell}(t;s)$ denote the number of 1-2-3-avoiding n-permutations such that $\pi_k = t$, $\pi_{n-\ell+1} = s$, $\pi_1 > \pi_2 > \cdots > \pi_k$, and $\pi_{n-\ell+1} > \pi_{n-\ell+2} > \cdots > \pi_n$. Also, we define $g_n(i_1, i_2, \dots, i_m; b)$ to be the number of 1-2-3-avoiding n-permutations such that $\pi_1\pi_2\cdots\pi_m = i_1 i_2 \dots i_m$ and $\pi_n = b$. We need the following two lemmas to prove Proposition 12.

Lemma 1. *For all $n \geq 2$,*

$$g_n(a;b) = \begin{cases} 0, & 2 \leq a+1 < b \leq n, \\ \binom{n-2}{a-1}, & 1 \leq a \leq n-1, \\ \sum_{j=0}^{n-a} (-1)^j \binom{n-a-j}{j} \left(\sum_{i=0}^{b-1} (-1)^i \binom{b-1-i}{i} C_{n-2-j-i} \right), & 1 \leq b < a \leq n. \end{cases}$$

Proof. By definitions we have

(1) $g_n(a;b) = 0$ for all $2 \leq a+1 < b \leq n$;

(2) $g_n(a;a+1) = g_n(a,1;a+1) + \ldots + g_n(a,a-1;a+1) + g_n(a,a+2;a+1) + \ldots + g_n(a,n-1;a+1) + g_n(a,n;a+1)$. Using the fact that no there exists a permutation $\pi \in S_n$(1-2-3) such that $\pi_1 = a$, $\pi_2 \leq a-2$, and $\pi_n = a+1$ we get

$$g_n(a;a+1) = g_n(a,a-1;a+1) + g_n(a,a+2;a+1) + \ldots + g_n(a,n;a+1).$$

Using the fact that no there exists a permutation $\pi \in S_n$(1-2-3) such that $\pi_1 = a$ and $a \leq \pi_2 \leq n-1$ we get $g_n(a;a+1) = g_n(a,a-1;a+1) + g_n(a,n;a+1)$. On the other hand, it is easy to see that $g_n(a,a-1;a+1) = g_{n-1}(a-1;a)$ and $g_n(a,n;a+1) = g_{n-1}(a;a+1)$. Hence,

$$g_n(a;a+1) = g_{n-1}(a-1;a) + g_{n-1}(a;a+1).$$

Using induction we get that $g_n(a;a+1) = \binom{n-2}{a-1}$ for all $n \geq 2$ and $1 \leq a \leq n-1$.

(3) Similarly as (2) we have for all $a > b$,

$$g_n(a;b) = g_{n-1}(b-1;b) + g_{n-1}(b+1;b) + g_{n-1}(b+2;b) + \cdots + g_{n-1}(a;b).$$

Using Equation (4.10) we get

$$g_n(a;1) = g_n(a;2) = s_{n-1}(a-1) = \sum_{j=0}^{n-a} (-1)^j \binom{n-a-j}{j} C_{n-2-j}.$$

Using induction on b, we get

$$g_n(a;b) = \sum_{j=0}^{n-a} (-1)^j \binom{n-a-j}{j} \left(\sum_{i=0}^{b-1} (-1)^i \binom{b-1-i}{i} C_{n-2-j-i} \right).$$

□

Lemma 2. *We have*

$$h_n^{k,\ell}(t;s) = \begin{cases} \binom{n-t}{k-1}\binom{s-1}{\ell-1} g_{t-(\ell-1);s-(\ell-1)}(n+2-k-\ell), & \text{if } 1 \le s < t \le n; \\ h_n^{k,\ell}(t+1;t), & \text{if } s = t+1; \\ h_{n-1}^{k,\ell-1}(t;s-1) + h_{n-1}^{k-1;\ell}(t;s-1), & \text{if } 2 \le t+1 < s \le n. \end{cases}$$

Proof. (1) Let $n \ge t > s \ge 1$; so by definitions we get

$$h_n^{k,\ell}(t;s) = \binom{n-t}{k-1}\binom{s-1}{\ell-1} g_{t-(\ell-1);s-(\ell-1)}(n-(k-1)-(\ell-1)).$$

(2) Let $s = t+1$; so it is easy to see $h_n^{k,\ell}(t;t+1) = h_n^{k,\ell}(t+1;t)$;

(3) Let $2 \le t+1 < s \le n$. Let π any permutations in S_n(1-2-3) such that $\pi_k = t$ and $\pi_{n+1-\ell} = s$ where $\pi_1 > \cdots > \pi_k$ and $\pi_{n+1-\ell} > \cdots > \pi_n$; so there two possibilities either $\pi_{n+2-\ell} = s-1$ or $\pi_j = s-1$ where $j \le k-1$. In this first case we get that there exist $h_{n-1}^{k,\ell-1}(t;s-1)$ permutations, and in the second case we have that there exist $h_{n-1}^{k-1;\ell}(t;s-1)$ permutations. (we extend the number $h_n^{k,l}(a;b)$ as 0 for any $\ell \le 0$ or $k \le 0$. □

We recall that the Kronecker delta $\delta_{n,k}$ is defined to be

$$\delta_{n,k} = \begin{cases} 1, & \text{if } n = k, \\ 0, & \text{else.} \end{cases}$$

Proposition 12. *We have*

(i) $G_{1\text{-}2\text{-}3}^{12\ldots k,12\ldots\ell}(x) = \begin{cases} 0, & \text{if } k \ge 3 \text{ or } \ell \ge 3 \\ xC^2(x), & \text{if } k = 1 \text{ and } \ell = 1 \end{cases}$,

$N_{1\text{-}2\text{-}3}^{12,12}(n) = \begin{cases} 0, & \text{if } n = 3 \\ C_{n-2}, & \text{else} \end{cases}$, *and*

$N_{1\text{-}2\text{-}3}^{12,1}(n) = N_{1-2-3}^{1,12}(n) = C_{n-1}$.

(ii) $N_{1\text{-}2\text{-}3}^{k(k-1)\ldots1,12\ldots\ell}(n) =$

$$\begin{cases} 0, & \text{if } \ell \ge 3, \\ \sum_{t=1}^{n-k} \binom{n-t-1}{k-1} \sum_{j=0}^{n-t-1} (-1)^j \binom{n-t-j-1}{j} C_{n-t-j-1} + (k-1)\delta_{n,k+1}, & \text{if } \ell = 2, \\ \sum_{t=1}^{n+1-k} \binom{n-t}{k-1} \sum_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j}, & \text{if } \ell = 1. \end{cases}$$

(iii) $N_{1\text{-}2\text{-}3}^{12\ldots k,\ell(\ell-1)\ldots1}(n) =$

$$\begin{cases} 0, & \text{if } k \ge 3, \\ \sum_{t=1}^{n-\ell} \binom{n-t-1}{\ell-1} \sum_{j=0}^{n-t-1} (-1)^j \binom{n-t-j-1}{j} C_{n-t-j-1} + (\ell-1)\delta_{n,\ell+1}, & \text{if } k = 2, \\ \sum_{t=1}^{n+1-\ell} \binom{n-t}{\ell-1} \sum_{j=0}^{n-t} (-1)^j \binom{n-t-j}{j} C_{n-t-j}, & \text{if } k = 1. \end{cases}$$

(iv) $N_{1-2-3}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x) = \sum_{t=1}^{n-k+1}\sum_{s=\ell}^{n} h_n^{k,\ell}(t;s)$, *where* $h_n^{k,\ell}(t;s)$ *is given in Lemma 2.*

Proof. **Beginning with** $12\dots k$ **and ending with** $12\dots\ell$**:** If $k \geq 3$ or $\ell \geq 3$, the statement is obvious, since in this case $12\dots k$ or $12\dots\ell$ does not avoid the pattern 1-2-3. If $k = 1$ or $\ell = 1$, we get the statement from Proposition 8 (in the first of these cases we apply the reverse and complement operations). Suppose now that $k = 2$, $\ell = 2$, and an n-permutation π avoids 1-2-3, begins with the pattern 12 and ends with the pattern 12. The letter n must be next to the leftmost letter, since otherwise two leftmost letters and n form the pattern 1-2-3. Also, the letter 1 must be next to the rightmost letter, since otherwise 1 and two rightmost letters form the pattern 1-2-3. It is easy to see now that there are C_{n-2} possibilities to choose π, since we can take any 1-2-3-avoiding permutation on the letters $\{2,3,\dots,n-1\}$ (there are C_{n-2} such permutations), then let the letters n and 1 be in the second and $(n-1)$-st positions respectively. These considerations are fail only when $n = 3$, since in this case the second and $(n-1)$-st positions coincide. However, in this case we obviously have no permutations with the good properties.

Beginning with $k(k-1)\dots1$ **and ending with** $12\dots\ell$**:** The statement is true for $\ell \geq 3$, since in this case $12\dots\ell$ does not avoid 1-2-3. For the case $\ell = 1$ we use Proposition 9. Suppose now that $\ell = 2$, and an n-permutation π avoids 1-2-3, begins with the pattern $k(k-1)\dots1$ and ends with the pattern 12. The letter 1 must be next to the rightmost letter, since otherwise 1 and two rightmost letters form the pattern 1-2-3. So, to form π we can take any $(n-1)$-permutation on the letters $\{2,3,\dots,n\}$ that avoid 1-2-3 and begin with the pattern $k(k-1)\dots1$ (the number of such permutations is given by Proposition 9), and then let the letter 1 be in the $(n-1)$-st position. Also, we observe that in the case $n = k+1$ we have $k-1$ extra permutations, which are obtained from the $(n-1)$-permutations having the $k-1$ leftmost letters in decreasing order and two rightmost letters in increasing order.

Beginning with $12\dots k$ **and ending with** $\ell(\ell-1)\dots1$**:** By the reverse and complement operations, to avoid 1-2-3, begin with the pattern $12\dots k$ and end with the pattern $\ell(\ell-1)\dots1$ is the same as to avoid 1-2-3, begin with the pattern $\ell(\ell-1)\dots1$ and end with the pattern $12\dots k$, so we can apply the results of the previous case.

Beginning with $k(k-1)\dots1$ **and ending with** $\ell(\ell-1)\dots1$**:** The statement is straitforward to prove. □

4.6 Avoiding a pattern xyz, beginning and ending with certain patterns simultaneously

Recall the definitions of $E_q^{p,r}(x)$ in Section 4.2.

Proposition 13. *We have*

(i) $E_{213}^{12\ldots k,12\ldots\ell}(x) = \begin{cases} E_{132}^{12\ldots\ell}(x), & \text{if } k=1 \\ E_{213}^{12\ldots k}(x), & \text{if } \ell=1 \end{cases}$, *where* $E_{132}^{12\ldots\ell}(x)$ *and* $E_{213}^{12\ldots k}(x)$ *are given in* [Kit3, Theorem 6] *and* [Kit3, Theorem 10] *respectively. For* $k,\ell \geq 2$, $E_{213}^{12\ldots k,12\ldots\ell}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{213}^{12\ldots k,12\ldots\ell}(x) = E_{213}^{12\ldots k,12\ldots(\ell-1)}(x)+\left(E_{213}^{12\ldots k,12}(x)+\frac{x^{k-1}}{(k-1)!}\right)E_{132}^{12\ldots\ell}(x).$$

(ii) $E_{213}^{12\ldots k,\ell(\ell-1)\ldots1}(x) = \begin{cases} E_{132}^{\ell(\ell-1)\ldots1}(x), & \text{if } k=1 \\ E_{213}^{12\ldots k}(x), & \text{if } \ell=1 \end{cases}$, *where* $E_{132}^{\ell(\ell-1)\ldots1}(x)$ *and* $E_{213}^{12\ldots k}(x)$ *are given in* [Kit3, Theorem 7] *and* [Kit3, Theorem 10] *respectively. For* $k,\ell \geq 2$, $E_{213}^{12\ldots k,\ell(\ell-1)\ldots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{213}^{12\ldots k,\ell(\ell-1)\ldots1}(x) = \frac{x^{\ell-1}}{(\ell-1)!}E_{213}^{12\ldots k}(x)+$$

$$\left(E_{213}^{12\ldots k,12}(x)+\frac{x^{k-1}}{(k-1)!}\right)E_{132}^{\ell(\ell-1)\ldots1}(x)+\binom{k+\ell-2}{k-1}\frac{x^{k+\ell-2}}{(k+\ell-2)!}.$$

(iii) $E_{213}^{k(k-1)\ldots1,12\ldots\ell}(x) = \begin{cases} E_{132}^{12\ldots\ell}(x), & \text{if } k=1 \\ E_{213}^{k(k-1)\ldots1}(x), & \text{if } \ell=1 \end{cases}$, *where* $E_{132}^{12\ldots\ell}(x)$ *and* $E_{213}^{k(k-1)\ldots1}(x)$ *are given in* [Kit3, Theorem 6] *and* [Kit3, Theorem 11] *respectively. For* $k,\ell \geq 2$, $E_{213}^{k(k-1)\ldots1,12\ldots\ell}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{213}^{k(k-1)\ldots1,12\ldots\ell}(x) = E_{213}^{(k-1)\ldots1,12\ldots\ell}(x)+$$

$$E_{213}^{k(k-1)\ldots1,12}(x)E_{132}^{12\ldots\ell}(x)+E_{213}^{k(k-1)\ldots1,12\ldots(\ell-1)}(x).$$

(iv) $E_{213}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) = \begin{cases} E_{132}^{\ell(\ell-1)\ldots1}(x), & \text{if } k=1 \\ E_{213}^{k(k-1)\ldots1}(x), & \text{if } \ell=1 \end{cases}$, *where* $E_{132}^{\ell(\ell-1)\ldots1}(x)$ *and* $E_{213}^{k(k-1)\ldots1}(x)$ *are given in* [Kit3, Theorem 7] *and* [Kit3, Theorem 11] *respectively. For* $k,\ell \geq 2$, $E_{213}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{213}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) = E_{213}^{(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)+$$

$$\left(E_{132}^{\ell(\ell-1)\ldots1}(x)+\frac{x^{\ell-1}}{(\ell-1)!}\right)E_{213}^{k(k-1)\ldots1,12}(x).$$

Proof.

Beginning with $12\ldots k$ and ending with $\ell(\ell-1)\ldots1$: The statement is obviously true when $k=1$ and $\ell=1$. Suppose now that $k\geq 2$, $\ell\geq 2$ and an $(n+1)$-permutation π avoids 213, begins with the pattern $12\ldots k$ and ends with the pattern $12\ldots\ell$. The letter $(n+1)$ can only be in the position k, or in the position i, where $(k+1)\leq i\leq n-\ell+1$, or in the position $n-\ell+2$.

In the first case, we choose the $(k-1)$ leftmost letters in $\binom{n}{k-1}$ ways, rearrange them into the increasing order, and observe, that the letters of π to the right of $(n+1)$ must form an $(n-k+1)$-permutation, that avoids 213 and ends with the pattern $\ell(\ell-1)\dots1$ (the number of such permutations, using the reverse and complement operation, is equal to the number of $(n-k+1)$-permutations that avoid 132 and begin with the pattern $\ell(\ell-1)\dots1$). In the third case, we choose the $(\ell-1)$ rightmost letters in $\binom{n}{\ell-1}$ ways, rearrange them into the decreasing order, and observe, that the letters of π to the right of $(n+1)$ must form an $(n-\ell+1)$-permutation, that avoids 213, begins with the pattern $12\dots k$, and ends with the pattern 12 (if it ends with the pattern 21, the letter $(n+1)$ and two letters immediately to the left of it form the pattern 213). In the second case, we choose the letters of π to the left of $(n+1)$ in $\binom{n}{i-1}$ ways and observe, that these letters must form a $(i-1)$-permutation that avoids 213, begins with the pattern $12\dots k$ and ends with the pattern 12. In the same time, the letters to the right of $(n+1)$ must form an $(n-i+2)$-permutation that avoids 213 and ends with the pattern $\ell(\ell-1)\dots1$. Besides, we observe that if $n=k+\ell-2$, that is $|\pi|=k+\ell-1$, and first k-letters of π are rearranged into the increasing order, whereas the last ℓ letters are rearranged in the decreasing order, we have a number of extra "good" permutations. The number of such permutations is the number of ways of choosing the first $(k-1)$ letters, that is $\binom{k+\ell-2}{k-1}$. This discussion leads to the following:

$$N_{213}^{12\dots k,\ell(\ell-1)\dots1}(n+1)=\tbinom{n}{k-1}N_{132}^{\ell(\ell-1)\dots1}(n-k+1)+\tbinom{n}{\ell-1}N_{213}^{12\dots k}(n-\ell+1)$$
$$+\sum_{i=0}^{n}\binom{n}{i}N_{213}^{12\dots k,12}(i)N_{132}^{\ell(\ell-1)\dots1}(n-i)+\binom{k+\ell-2}{k-1}\delta_{n,k+\ell-2},$$

where $\delta_{n,k+\ell-2}$ is the Kronecker delta. We get the desirable result by multiplying both sides of the last equality by $x^n/n!$ and summing over n.

Beginning with $12\dots k$ and ending with $12\dots\ell$: The statement is obviously true when $k=1$ and $\ell=1$. Suppose now that $k\geq2$, $\ell\geq2$ and an $(n+1)$-permutation π avoids 213, begins with the pattern $12\dots k$ and ends with the pattern $12\dots\ell$. The letter $(n+1)$ can only be in the position k, or in the position i, where $(k+1)\leq i\leq n-\ell$, or in the $(n+1)$-th position. In the last case, the number of such permutations is obviously $N_{213}^{12\dots k,12\dots\ell-1}(n)$. In the first case, we choose the $(k-1)$ leftmost letters in $\binom{n}{k-1}$ ways, rearrange them into increasing order, and observe, that the letters of π to the right of $(n+1)$ must form an $(n-k+1)$-permutation, that avoids 213 and ends with the pattern $12\dots\ell$ (the number of such permutations, using the reverse and complement operation, is equal to the number of $(n-k+1)$-permutations that avoid 132 and begin with the pattern $12\dots\ell$). In the second case, we choose the letters of π to the left of $(n+1)$ in $\binom{n}{i-1}$ ways and observe, that these letters must form a $(i-1)$-permutation that avoids 213, begins with the pattern $12\dots k$ and ends with the pattern 12 (if it ends with the pattern 21, the letter $(n+1)$ and two letters immediately to the left of it form the pattern 213). In the same time, the letters to the right of $(n+1)$ must form an $(n-i+2)$-permutation that avoids

213 and ends with the pattern $12\dots\ell$. This discussion leads to the following:

$$N_{213}^{12\dots k,12\dots\ell}(n+1) = N_{213}^{12\dots k,12\dots\ell-1}(n) + \sum_{i=0}^{n}\binom{n}{i}N_{213}^{12\dots k,12}(i)N_{132}^{12\dots\ell}(n-i) + \binom{n}{k-1}N_{132}^{12\dots\ell}(n-k+1).$$

We get the desirable result by multiplying both sides of the last equality by $x^n/n!$ and summing over n.

Beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$ or with $\ell(\ell-1)\dots1$: We proceed in the same way as we do under considering the previous case. □

Proposition 14. *We have*

(i) $E_{132}^{12\dots k,12\dots\ell}(x) = \begin{cases} E_{213}^{12\dots\ell}(x), & \text{if } k=1 \\ E_{132}^{12\dots k}(x), & \text{if } \ell=1 \end{cases}$, *where* $E_{213}^{12\dots\ell}(x)$ *and* $E_{132}^{12\dots k}(x)$ *are given in* [Kit3, Theorem 10] *and* [Kit3, Theorem 6] *respectively. For* $k,\ell\geq 2$, $E_{132}^{12\dots k,12\dots\ell}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{132}^{12\dots k,12\dots\ell}(x) = E_{132}^{12\dots k-1,12\dots\ell}(x)+\left(E_{132}^{12,12\dots\ell}(x)+\frac{x^{\ell-1}}{(\ell-1)!}\right)E_{132}^{12\dots k}(x).$$

(ii) $E_{132}^{12\dots k,\ell(\ell-1)\dots1}(x) = \begin{cases} E_{132}^{12\dots k}(x), & \text{if } \ell=1 \\ E_{213}^{\ell(\ell-1)\dots1}(x), & \text{if } k=1 \end{cases}$, *where* $E_{132}^{12\dots k}(x)$ *and* $E_{213}^{\ell(\ell-1)\dots1}(x)$ *are given in* [Kit3, Theorem 6] *and* [Kit3, Theorem 11] *respectively. For* $k,\ell\geq 2$, $E_{132}^{12\dots k,\ell(\ell-1)\dots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{132}^{12\dots k,\ell(\ell-1)\dots1}(x) = E_{132}^{12\dots k,(\ell-1)\dots1}(x)+$$
$$E_{132}^{12,\ell(\ell-1)\dots1}(x)E_{132}^{12\dots k}(x) + E_{132}^{12\dots(k-1),\ell(\ell-1)\dots1}(x).$$

(iii) $E_{132}^{k(k-1)\dots1,12\dots\ell}(x) = \begin{cases} E_{213}^{12\dots\ell}(x), & \text{if } k=1 \\ E_{132}^{k(k-1)\dots1}(x), & \text{if } \ell=1 \end{cases}$, *where* $E_{213}^{12\dots\ell}(x)$ *and* $E_{132}^{k(k-1)\dots1}(x)$ *are given in* [Kit3, Theorem 10] *and* [Kit3, Theorem 7] *respectively. For* $k,\ell\geq 2$, $E_{132}^{k(k-1)\dots1,12\dots\ell}(x)$ *satisfies*

$$\tfrac{\partial}{\partial x}E_{213}^{k(k-1)\dots1,12\dots\ell}(x) = \tfrac{x^{k-1}}{(k-1)!}E_{213}^{12\dots\ell}(x) + \left(E_{132}^{12,12\dots\ell}(x)+\tfrac{x^{\ell-1}}{(\ell-1)!}\right)E_{132}^{k(k-1)\dots1}(x) + \tbinom{k+\ell-2}{\ell-1}\tfrac{x^{k+\ell-2}}{(k+\ell-2)!}.$$

(iv) $E_{132}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x) = \begin{cases} E_{213}^{\ell(\ell-1)\dots1}(x), & \text{if } k=1 \\ E_{132}^{k(k-1)\dots1}(x), & \text{if } \ell=1 \end{cases}$, *where* $E_{132}^{k(k-1)\dots1}(x)$ *and* $E_{213}^{\ell(\ell-1)\dots1}(x)$ *are given in* [Kit3, Theorem 7] *and* [Kit3, Theorem 11] *respectively. For* $k,\ell\geq 2$, $E_{132}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{132}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x) = E_{132}^{(\ell-1)\dots1,k(k-1)\dots1}(x)+$$

$$\left(E_{132}^{k(k-1)\ldots1}(x)+\frac{x^{k-1}}{(k-1)!}\right)E_{132}^{12,\ell(\ell-1)\ldots1}(x).$$

Proof. We apply the reverse and complement operations and then use the results of Proposition 14. For example, to avoid 213, begin with $12\ldots k$ and end with $12\ldots\ell$ is the same as to avoid 132, begin with $12\ldots\ell$ and end with $12\ldots k$. □

Proposition 15. *We have*

(i) $E_{123}^{12\ldots k,12\ldots\ell}(x) =$

$$\begin{cases} 0, & \text{if } k\geq 3 \text{ or } \ell\geq 3,\\ x-\frac{1}{2}-\frac{\sqrt{3}}{2}\tan\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)+ & \\ \sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)\left(\frac{\sqrt{3}}{2}\left(e^{x/2}+e^{-x/2}\right)-\sin\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{3}\right)\right), & \text{if } k=2 \text{ and } \ell=2,\\ \frac{\sqrt{3}}{2}e^{x/2}\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)-1, & \text{if } k=1 \text{ and } \ell=1,\\ \frac{\sqrt{3}}{2}e^{x/2}\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)-\frac{1}{2}-\frac{\sqrt{3}}{2}\tan\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right), & \text{else;}\end{cases}$$

(ii) $E_{123}^{12\ldots k,\ell(\ell-1)\ldots1}(x) =$

$$\begin{cases} 0, & \text{if } k\geq 3,\\ \Phi_\ell(x)=\frac{e^{x/2}}{(\ell-1)!}\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)\int_0^x e^{-t/2}t^{\ell-1}\sin\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{3}\right)\,dt, & \text{if } k=1,\\ \int_0^x \sec\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{6}\right)\left(\sin\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{3}\right)-\frac{\sqrt{3}}{2}e^{-t/2}\right)\left(\Phi_\ell(t)+\frac{t^{\ell-1}}{(\ell-1)!}\right)\,dt, & \text{if } k=2;\end{cases}$$

(iii) $E_{123}^{k(k-1)\ldots1,12\ldots\ell}(x) =$

$$\begin{cases} 0, & \text{if } \ell\geq 3,\\ \Phi_k(x)=\frac{e^{x/2}}{(k-1)!}\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)\int_0^x e^{-t/2}t^{k-1}\sin\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{3}\right)\,dt, & \text{if } \ell=1,\\ \int_0^x \sec\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{6}\right)\left(\sin\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{3}\right)-\frac{\sqrt{3}}{2}e^{-t/2}\right)\left(\Phi_k(t)+\frac{t^{k-1}}{(k-1)!}\right)\,dt, & \text{if } \ell=2;\end{cases}$$

(iv) $E_{123}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) = \begin{cases} E_{123}^{\ell(\ell-1)\ldots1}(x), & \text{if } k=1,\\ E_{123}^{k(k-1)\ldots1}(x), & \text{if } \ell=1,\\ E_{123}^{k(k-1)\ldots1}(x)-E_{123}^{k(k-1)\ldots1,12}(x), & \text{if } \ell=2;\end{cases}$

For $k\geq 2$ and $\ell\geq 3$, $E_{123}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$ satisfies

$$\frac{\partial}{\partial x}E_{123}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)=\left(E_{123}^{\ell(\ell-1)\ldots1}(x)+\frac{x^{\ell-1}}{(\ell-1)!}\right)E_{123}^{k(k-1)\ldots1,21}(x)+E_{123}^{(k-1)\ldots1,\ell(\ell-1)\ldots1}(x),$$

where $E_{123}^{k(k-1)\ldots1}(x)$ is given in [KitMans, Theorem 2]:

$$E_{123}^{k(k-1)\ldots1}(x)=\frac{e^{x/2}\int_0^x e^{-t/2}t^{k-1}\sin(\frac{\sqrt{3}}{2}t+\frac{\pi}{6}))\,dt}{(k-1)!\cos(\frac{\sqrt{3}}{2}x+\frac{\pi}{6})},$$

and $E_{123}^{k(k-1)\ldots1,12}$ is given in this theorem above.

Proof.

Beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$: If $\ell \geq 3$ then the pattern $12\dots\ell$ does not avoid 123, thus the statement is true. If $\ell = 1$, the statement is true according to [Kit3, Theorem 8] and the observation that if $k = 1$ then this formula gives the expression

$$\frac{\sqrt{3}}{2}e^{x/2}\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)-1,$$

which is true according to [ElizNoy, Theorem 4.1] and the assumption that the empty permutation does not begin or end with the pattern $p = 1$. So, we need only to consider the case $\ell = 2$. Recall the definitions of $E_q^p(x)$ in Section 4.2.

Let $P_k(n)$ denote the number of n-permutations that avoid the pattern 123, begin with a decreasing subword of length k and end with the pattern 12. Also, let $R_k(n)$ denote the number of n-permutations that avoid the pattern 123 and begin with a decreasing subword of length k. Let $\pi = \pi_1 1 \pi_2$ be an $(n+1)$-permutation that avoids the pattern 123, begins with the pattern $k(k-1)\dots1$ and ends with the pattern 12. We observe that π_1 avoids 123 and begins with $k(k-1)\dots1$; π_2 ends with the pattern 12 and $|\pi_2| > 0$ since otherwise π cannot end with the pattern 12; if $|\pi_2| > 1$ then π_2 must begin with the pattern 21 since otherwise we have an occurrence of the pattern 123 beginning from the letter 1. If $|\pi_1| = i$ then the letters of π_1 can be chosen in $\binom{n}{i}$ ways. So, there are at least

$$\sum_{i\geq0}\binom{n}{i}R_k(i)P_2(n-i)+nR_k(n-1)$$

$(n+1)$-permutations with the good properties, where the first term corresponds to the case $|\pi_2| > 1$ and the second term to the case $|\pi_2| = 1$. By this formula, we do not count the permutations having $|\pi_1| = k-1$, although in this case π begins with the pattern $k(k-1)\dots1$. So, we can choose the letters of π_1 in $\binom{n}{k-1}$ ways, and according to whether $|\pi| \geq 1$ or $|\pi| = 1$, we have two terms:

$$\binom{n}{k-1}P_2(n-k+1)+k\delta_{n,k},$$

where $\delta_{n,k}$ is the Kronecker delta. Thus,

$$P_k(n+1)=\sum_{i\geq0}\binom{n}{i}R_k(i)P_2(n-i)+nR_k(n-1)+\binom{n}{k-1}P_2(n-k+1)+k\delta_{n,k}.$$

After multiplying both sides of the last equality with $x^n/n!$ and summing over n, we have

$$\frac{d}{dx}E_{123}^{k(k-1)\dots1,12}(x)=(E_{123}^{21,12}(x)+x)\left(E_{123}^{k(k-1)\dots1}(x)+\frac{x^{k-1}}{(k-1)!}\right), \qquad (4.11)$$

with the initial condition $E_{123}^{k(k-1)\dots1,12}(0) = 0$. Since

$$E_{123}^{k(k-1)\dots1}(x) = E_{123}^{k(k-1)\dots1,1}(x) =$$

$$\frac{e^{x/2}}{(k-1)!}\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)\int_0^x e^{-t/2}t^{k-1}\sin\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{3}\right)\,dt,$$

to solve (4.11), we only need to know $E_{123}^{21,12}(x)$. To find it, we set $k=2$ into (4.11) and solve this equation. For an example how to solve such an equation, we refer to [Kit3, Theorem 6]. We get

$$E_{123}^{21,12}(x)=-x+\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)e^{-x/2}\int_0^x e^{t/2}\cos\left(\frac{\sqrt{3}}{2}t+\frac{\pi}{6}\right)\,dt.$$

Now, we put the formula for $E_{123}^{21,12}(x)$ into (4.11) and solve this differential equation to get the desired result.

Beginning with $12\dots k$ and ending with $\ell(\ell-1)\dots 1$: By the reverse and complement operations, to avoid 123, begin with the pattern $12\dots k$ and end with the pattern $\ell(\ell-1)\dots 1$ is the same as to avoid 123, begin with the pattern $\ell(\ell-1)\dots 1$ and end with the pattern $12\dots k$, so we can apply the results of the previous case.

Beginning with $12\dots k$ and ending with $12\dots\ell$: The statement is obvious if $k\geq 3$ or $\ell\geq 3$. If $k=1$ and $\ell=1$ then the statement is true according to [ElizNoy, Theorem 4.1] (but we need to subtract 1, since by our assumption the empty permutation does not begin or end with the pattern $p=1$). If $\ell=1$ and $k=2$, the statement is true according [Kit3, Theorem 9]. If $k=1$ and $\ell=2$, we apply the reverse and complement operations, and use again [Kit3, Theorem 9]. So, we only need to consider the case $k=2$ and $\ell=2$. It is easy to see that

$$E_{123}^{12,12}(x)=E_{123}^{1,12}(x)-E_{123}^{21,12}(x),$$

and from the previous cases

$$E_{123}^{1,12}(x)=\frac{\sqrt{3}}{2}e^{x/2}\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)-\frac{1}{2}-\frac{\sqrt{3}}{2}\tan\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right),$$

and

$$E_{123}^{21,12}(x)=-x+\sec\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{6}\right)\left(\sin\left(\frac{\sqrt{3}}{2}x+\frac{\pi}{3}\right)-\frac{\sqrt{3}}{2}e^{-x/2}\right).$$

Beginning with $k(k-1)\dots 1$ and ending with $\ell(\ell-1)\dots 1$: If $\ell=1$, the statement is trivial. If $k=1$, we get the statement by using the reverse and complement operations. For the case $\ell=2$, we observe that the number of n-permutations that avoid the pattern 123, begin with the pattern $k(k-1)\dots 1$ and end with the pattern 21 is equal to the number of n-permutation that avoid 123 and begin with the pattern $k(k-1)\dots 1$ minus the number of n-permutations that avoid the pattern 123, begin with the pattern $k(k-1)\dots 1$ and end with the pattern 12. Suppose now that $k\geq 2$ and $\ell\geq 3$ and an $(n+1)$-permutation π avoids 123, begins with $k(k-1)\dots 1$ and ends with $\ell(\ell-1)\dots 1$.

It is easy to see that the letter $(n+1)$ can be either in the first position, or in the position i, where $(k+1) \le i \le (n-\ell)$, or in the position $(n-\ell+1)$. In the first of these cases, obviously we have $N_{123}^{(k-1)\dots1,\ell(\ell-1)\dots1}(n)$ permutations. In the second case, we choose the letters of π to the left of $(n+1)$ in $\binom{n}{i-1}$ ways. These letters must form a permutation that avoids 123, begins with the pattern $k(k-1)\dots1$, and ends with the pattern 21 (if the last condition does not hold, the letter $(n+1)$ and two letters to the left of it form a 123-pattern. In the same time, the letters to the right of $(n+1)$ form a permutation that avoids 123 and ends with the pattern $\ell(\ell-1)\dots1$. In the third case, we can choose the letters to the right of $(n+1)$ in $\binom{n}{\ell-1}$ ways, rearrange them into the decreasing order, and form from the letters to the left of $(n+1)$ a permutation that avoids 123, begins with the pattern $k(k-1)\dots1$ and ends with the pattern 21 (by the same reasons as above) in $N_{123}^{k(k-1)\dots1,21}(n-\ell+1)$ ways. Thus,

$$N_{123}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(n+1) = N_{123}^{(k-1)\dots1,\ell(\ell-1)\dots1}(n)$$
$$+\sum_{i=0}^{n}\binom{n}{i}N_{123}^{k(k-1)\dots1,21}(i)N_{123}^{\ell(\ell-1)\dots1}(n-i) + \binom{n}{\ell-1}N_{123}^{k(k-1)\dots1,21}(n-\ell+1),$$

where we observed, that to avoid 123 and end with $\ell(\ell-1)\dots1$ is the same as to avoid 123 and begin with $\ell(\ell-1)\dots1$ using the reverse and complement. Now, we multiply both sides of the equality by $x^n/n!$ and sum over n to get the desirable result. □

4.7 Avoiding a pattern x-yz, beginning and ending with certain patterns simultaneously

Proposition 16. *We have*

(i) $E_{1\text{-}32}^{12\dots k,1}(x) = E_{1\text{-}32}^{12\dots k}(x) = \begin{cases} e^{e^x}\int_0^x e^{-e^t}\sum\limits_{n\ge k-1}\dfrac{t^n}{n!}\,dt, & \text{if } k\ge 2 \\ e^{e^x-1}, & \text{if } k=1 \end{cases}$.

For $\ell \ge 2$, $E_{1\text{-}32}^{12\dots k,12\dots\ell}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{1\text{-}32}^{12\dots k,12\dots\ell}(x) = \left(e^x - \sum_{i=0}^{\ell-2}\frac{x^i}{i!}\right)E_{1\text{-}32}^{12\dots k}(x) + e^x x^{max(\ell,k)-1}.$$

(ii) $E_{1\text{-}32}^{12\dots k,\ell(\ell-1)\dots1}(x)$ *satisfies*

$$\frac{\partial^{\ell-1}}{\partial x^{\ell-1}}E_{1\text{-}32}^{12\dots k,\ell(\ell-1)\dots1}(x) = \begin{cases} e^{e^x}\int_0^x e^{-e^t}\sum\limits_{n\ge k-1}\dfrac{t^n}{n!}\,dt, & \text{if } k\ge 2, \\ e^{e^x-1}, & \text{if } k=1. \end{cases}$$

(iii)

$$E_{1\text{-}32}^{k(k-1)\dots1,1}(x) = E_{1\text{-}32}^{k(k-1)\dots1}(x) =$$

$$\begin{cases} (e^{e^x}/(k-1)!)\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k \geq 2 \\ e^{e^x-1}, & \text{if } k = 1 \end{cases}$$

For $\ell \geq 2$, $E_{1\text{-}32}^{k(k-1)\ldots1,12\ldots\ell}(x)$ *satisfies*

$$\frac{\partial}{\partial x}E_{1\text{-}32}^{k(k-1)\ldots1,12\ldots\ell}(x) = \left(e^x - \sum_{i=0}^{\ell-2}\frac{x^i}{i!}\right)E_{1\text{-}32}^{k(k-1)\ldots1}(x) + \left(e^x - \sum_{i=0}^{\ell-2}\frac{x^i}{i!}\right)\frac{x^{k-1}}{(k-1)!}.$$

(iv) $E_{1\text{-}32}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$ *satisfies*

$$\frac{\partial^{\ell-1}}{\partial x^{\ell-1}}\left(E_{1\text{-}32}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) - \frac{x^{max(k,\ell)} - x^{k+\ell-1}}{1-x}\right) =$$

$$\begin{cases} \frac{e^{e^x}}{(k-1)!}\int_0^x t^{k-1}e^{-e^t+t}\,dt, & \text{if } k \geq 2, \\ e^{e^x-1}, & \text{if } k = 1. \end{cases}$$

Proof.

Beginning with $12\ldots k$ **and ending with** $\ell(\ell-1)\ldots1$**:** If $\ell = 1$ then the result follows from [KitMans, Proposition 5], since to avoid 1-32 and begin with $12\ldots k$ is the same as to avoid 3-12 and begin with $k(k-1)\ldots1$. Suppose now that $\ell \geq 2$ and a permutation π avoids the pattern 1-32, begins with the pattern $12\ldots k$ and ends with the pattern $\ell(\ell-1)\ldots1$. Since $\ell \geq 2$, we have that the letter 1 must be in the rightmost position since otherwise, this letter and two rightmost letters of π form the pattern 1-32, which is forbidden. Thus,

$$N_{1\text{-}32}^{12\ldots k,\ell(\ell-1)\ldots1}(n) = N_{1\text{-}32}^{12\ldots k,(\ell-1)(\ell-2)\ldots1}(n-1) = \cdots = N_{1\text{-}32}^{12\ldots k,1}(n-\ell+1).$$

Multiplying both sides of the equality $N_{1\text{-}32}^{12\ldots k,\ell(\ell-1)\ldots1}(n) = N_{1\text{-}32}^{12\ldots k,1}(n-\ell+1)$ by $x^{n-\ell+1}/(n-\ell+1)!$ and summing over n, we get

$$\frac{\partial^{\ell-1}}{\partial x^{\ell-1}}E_{1\text{-}32}^{12\ldots k,\ell(\ell-1)\ldots1}(x) = E_{1\text{-}32}^{12\ldots k}(x),$$

where $E_{1\text{-}32}^{12\ldots k}(x)$ is given in [KitMans, Proposition 5], since to avoid 1-32 and begin with $12\ldots k$ is the same as to avoid 3-12 and begin with $k(k-1)\ldots1$.

Beginning with $k(k-1)\ldots1$ **and ending with** $\ell(\ell-1)\ldots1$**:** We use the same arguments as those given under consideration of the previous case, but instead of [KitMans, Proposition 5] we use [KitMans, Proposition 4]. However, we observe, that when we use the argument

$$N_{1\text{-}32}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(n) = N_{1\text{-}32}^{k(k-1)\ldots1,(\ell-1)(\ell-2)\ldots1}(n-1) = \cdots = N_{1\text{-}32}^{k(k-1)\ldots1,1}(n-\ell+1)$$

for $k, \ell \geq 2$, we do not count the decreasing permutations of length $max(k,\ell)$, $max(k,\ell)+1, \ldots, k+\ell-2$, since in this case, the patterns $k(k-1)\ldots1$ and $\ell(\ell-1)\ldots1$ overlap in more than one letter, which causes the observation. So, we need to consider additionally the term

$$x^{max(k,\ell)} + x^{max(k,\ell)+1} + \cdots + x^{k+\ell-2} = \frac{x^{max(k,\ell)} - x^{k+\ell-1}}{1-x},$$

which vanishes if $k = 1$ or $\ell = 1$.

Beginning with $12\dots k$ and ending with $12\dots\ell$: The only interesting case here is the case $k \geq 2$ and $\ell \geq 2$. Using the reverse and complement, instead of considering avoiding 1-32, beginning with $12\dots k$ and ending with $12\dots\ell$, we consider avoiding 1-32, beginning with $12\dots\ell$ and ending with $12\dots k$. Suppose an n-permutation π satisfies all the conditions. We observe, that the letter n can be in the position i, where $\ell \leq i \leq n-k$. Also, n can be in the rightmost position if $n \geq max(\ell, k)$. In any case, the letters of π to the left of n must be in the increasing order, since otherwise we have an occurrence of the pattern 21-3. This means that in the second case we have the only one permutation. In the first case, the letters of π to the right of n must avoid 21-3 and end with the pattern $12\dots k$. The number of such permutations, using the reverse and complement, is given by $N_{1\text{-}32}^{12\dots k}(n-i)$. Thus, for $n \geq max(\ell, k)$,

$$N_{21\text{-}3}^{12\dots\ell,12\dots k}(n) = \sum_{i=\ell}^{n-k} \binom{n-1}{i-1} N_{1\text{-}32}^{12\dots k}(n-i) + 1.$$

This gives

$$N_{21\text{-}3}^{12\dots\ell,12\dots k}(n) = \sum_{i=1}^{n} \binom{n-1}{i-1} N_{1\text{-}32}^{12\dots k}(n-i) - \sum_{i=1}^{\ell-1} \binom{n-1}{i-1} N_{1\text{-}32}^{12\dots k}(n-i) + 1,$$

which leads to the desirable result after multiplying both sides of the last equality by $x^n/n!$ and summing over n.

Beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$: The only interesting case here is the case $k \geq 2$ and $\ell \geq 2$. Using the reverse and complement, instead of considering avoiding 1-32, beginning with $k(k-1)\dots1$ and ending with $12\dots\ell$, we consider avoiding 1-32, beginning with $12\dots\ell$ and ending with $k(k-1)\dots1$. Suppose an n-permutation π satisfies all the conditions. We observe, that the letter n can only be in the position i, where $\ell \leq i \leq n-k$, or in position $(n-k+1)$ (in the case $n \geq k+\ell-1$). In the first case, it is easy to see that the letters of π to the left of n must be in the increasing order, and the letters of π to the right of n must avoid 21-3 and end with the pattern $k(k-1)\dots1$. Using the reverse and complement, the total number of permutations counted in the first case is $\sum_{i=\ell}^{n-k} \binom{n-1}{i-1} N_{1\text{-}32}^{k(k-1)\dots1}(n-i)$. In the second case, the letters to the left of n are in the increasing order, whereas the letters to the right of n are in decreasing order. The number of such permutations is $\binom{n-1}{k-1}$, which is the number of ways to choose the last $k-1$ letters. Thus,

$$N_{21\text{-}3}^{12\dots\ell,k(k-1)\dots1}(n) = \sum_{i=\ell}^{n-k} \binom{n-1}{i-1} N_{1\text{-}32}^{k(k-1)\dots1}(n-i) + \binom{n-1}{k-1}.$$

Multiplying both parts of the equality by $x^{n-1}/(n-1)!$ and summing over n,

we get

$$\frac{\partial}{\partial x}E_{21\text{-}3}^{12\ldots\ell,k(k-1)\ldots1}(x) = \sum_{n\geq k+\ell}\binom{n-1}{k-1}\frac{x^{n-1}}{(n-1)!}$$
$$+\sum_{n\geq 0}\left(\sum_{i=1}^{n-1}\binom{n-1}{i-1}N_{1\text{-}32}^{k(k-1)\ldots1}(n-i) - \sum_{i=1}^{\ell-1}\binom{n-1}{i-1}N_{1\text{-}32}^{k(k-1)\ldots1}(n-i)\right)\frac{x^{n-1}}{(n-1)!},$$

which leads to the desirable result. □

Proposition 17. *We have*

(i) $G_{2\text{-}13}^{12\ldots k,12\ldots\ell}(x) = x^{k+\ell-1}C^{k+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$.

(ii) $G_{2\text{-}13}^{k(k-1)\ldots1,12\ldots\ell}(x) = x^{k+\ell-1}C^2(x)$.

(iii) $G_{2\text{-}13}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) = x^{k+\ell-1}C^{\ell+1}(x) + \frac{x^m - x^{k+\ell-1}}{1-x}$, *where* $m = \max(k,\ell)$.

(iv) *the generating function* $G_{2\text{-}13}(x,y,z) = \sum_{k,\ell\geq 0} G_{2\text{-}13}^{12\ldots k,\ell(\ell-1)\ldots1}(x)y^k z^\ell$ *for the sequence* $\{G_{2\text{-}13}^{12\ldots k,\ell(\ell-1)\ldots1}(x)\}_{k,\ell\geq 0}$ (*where* k *and* ℓ *go through all natural numbers*) *is*

$$\frac{1}{1-x(y+z)}\left(x(y+z+yz) + \frac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right).$$

Proof. By [Claes], to avoid the pattern 2-13 is the same as to avoid the pattern 2-1-3. Thus we can apply the results of Proposition 11. □

Proposition 18. *We have*

(i) $E_{1\text{-}23}^{12\ldots k,12\ldots\ell}(x) = \begin{cases} 0, & \text{if } k\geq 3 \text{ or } \ell\geq 3,\\ E_{1\text{-}23}^{12\ldots k}(x), & \text{if } \ell = 1,\\ E_{12\text{-}3}^{12\ldots\ell}(x), & \text{if } k = 1,\\ \int_0^x tE_{12\text{-}3}^{12}(t)\,dt + \frac{x^2}{2!}, & \text{if } k=2 \text{ and } \ell = 2, \end{cases}$

where $E_{12\text{-}3}^{12\ldots k}(x)$ *and* $E_{1\text{-}23}^{12\ldots k}(x)$ *are given by* [KitMans, Proposition 10] *and* [KitMans, Proposition 6] *respectively:*

$$E_{12\text{-}3}^{12\ldots k}(x) = \begin{cases} 0, & \text{if } k\geq 3,\\ x^2\sum_{j=0}^{k}(1-jx)^{-1}\sum_{d\geq 0}\frac{x^d}{(1-x)(1-2x)\ldots(1-dx)}, & \text{if } k=2,\\ \sum_{d\geq 0}\frac{x^d}{(1-x)(1-2x)\ldots(1-dx)}, & \text{if } k=1; \end{cases}$$

$$E_{1\text{-}23}^{12\ldots k}(x) = E_{3-21}^{k(k-1)\ldots1}(x) = \begin{cases} 0, & \text{if } k\geq 3,\\ e^{e^x}\int_0^x e^{-e^t}(e^t-1)\,dt, & \text{if } k=2,\\ e^{e^x-1}, & \text{if } k=1. \end{cases}$$

(ii) $N_{1\text{-}23}^{12\ldots k,\ell(\ell-1)\ldots1}(n) =$

$$\begin{cases} 0, & \text{if } k \geq 3, \\ 0, & \text{if } k = 2 \text{ and } n \leq \ell, \\ 1 + N_{1\text{-}23}^{12,(\ell-1)(\ell-2)\ldots1}(n-1) + \sum_{j=\ell+1}^{n-2} \binom{n-1}{j-1} N_{1\text{-}23}^{12}(n-j), & \text{if } k = 2 \text{ and } n \geq \ell+1, \\ N_{12\text{-}3}^{\ell(\ell-1)\ldots1}(n), & \text{if } k = 1, \end{cases}$$

where the numbers $N_{12\text{-}3}^{\ell(\ell-1)\ldots1}(n)$ *are given in* [KitMans, Proposition 9]*, and the numbers* $N_{1\text{-}23}^{12}(n)$ *are given by expending the exponential generating functions in* [KitMans, Proposition 6].

(iii)

$$E_{1\text{-}23}^{k(k-1)\ldots1,12\ldots\ell}(x) =$$

$$\begin{cases} 0, & \text{if } \ell \geq 3 \\ \frac{1}{(k-1)!} \int_0^x \int_0^t t m^{k-1} e^{e^t - e^m + m} \, dm dt + \frac{k x^{k+1}}{(k+1)!}, & \text{if } \ell = 2 \\ (e^{e^x}/(k-1)!) \int_0^x t^{k-1} e^{-e^t + t} \, dt, & \text{if } \ell = 1 \end{cases},$$

where $E_{1\text{-}23}^{k(k-1)\ldots1,1}(n) = E_{1\text{-}23}^{k(k-1)\ldots1}(n)$ *is given by* [KitMans, Proposition 4]*, and* $N_{1\text{-}23}^{1,\ell(\ell-1)\ldots1}(n) = N_{12\text{-}3}^{\ell(\ell-1)\ldots1}(n)$ *is given by* [KitMans, Proposition 9]*;*

(iv) *For* $k \geq 2$ *and* $\ell \geq 2$, $E_{1\text{-}23}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x} E_{1\text{-}23}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) = E_{1\text{-}23}^{k(k-1)\ldots1,(\ell-1)\ldots1}(x) + \left(e^x - \sum_{i=0}^{\ell-1} \frac{x^i}{i!} \right) \left(E_{1\text{-}23}^{k(k-1)\ldots1}(x) + \frac{x^k}{(k-1)!} \right).$$

Proof.

Beginning with $k(k-1)\ldots1$ **and ending with** $12\ldots\ell$**:** If $\ell \geq 3$ then $E_{1\text{-}23}^{k(k-1)\ldots1,12\ldots\ell}(x) = 0$, since in this case the pattern $12\ldots\ell$ does not avoid 1-23. If $\ell = 1$ then we use [KitMans, Proposition 4], since in this case the only restrictions to the permutations are avoiding 1-23 and beginning with the pattern $k(k-1)\ldots1$. Suppose now that $\ell = 2$ and an $(n+1)$-permutation π avoids 1-23, begins with $k(k-1)\ldots1$ and ends with the pattern 12. The letter 1 must be in next to the rightmost position, since otherwise this letter and two rightmost letters form the pattern 1-23. We can choose the rightmost letter of π in n ways, and the letters to the left of 1 must form a 1-23-avoiding permutation that begins with $k(k-1)\ldots1$. Besides, if $n = k$, and the $k-1$ letters to the right of 1 are in the decreasing order, we get n extra permutations that satisfy our restrictions. Thus,

$$N_{1\text{-}23}^{k(k-1)\ldots1,12}(n+1) = n N_{1\text{-}23}^{k(k-1)\ldots1}(n) + n\delta_{n,k},$$

where $\delta_{n,k}$ is the Kronecker delta. Multiplying both sides of the equality by $x^n/n!$ and summing over n we get

$$E_{1\text{-}23}^{k(k-1)\ldots1,12}(x) = \int_0^x tE_{1\text{-}23}^{k(k-1)\ldots1}(t)\ dt + \frac{kx^{k+1}}{(k+1)!}.$$

Using the formula for $E_{1\text{-}23}^{k(k-1)\ldots1}(t)$ in [KitMans, Proposition 4], we get the desirable result.

Beginning with $12\ldots k$ and ending with $12\ldots\ell$: The first three cases are easy to prove in the same manner as we do in the proves of previous propositions. The only interesting case is when $k = 2$ and $\ell = 2$. Using the reverse and complement operations, instead of considering avoiding 1-23, beginning with 12 and ending with 12, we consider avoiding 12-3, beginning with 12 and ending with 12, which we find to be more easy. Suppose an $(n+1)$-permutation π satisfies all the restrictions. It is easy to see that $|\pi| \neq 1$ and $|\pi| \neq 3$, as well as if $|\pi| = 2$ (that is $n = 1$) then π must be 12. Suppose $|\pi| \geq 4$. Since π begins with the pattern 12, it is impossible for the letter $(n+1)$ to be somewhere to the right of the second letter of π or to be the leftmost letter. Thus, $(n+1)$ must be in the second position. We can choose the leftmost letter of π in n ways, since any choice of this letter will not lead to an occurrence of the pattern $12-3$ beginning with two leftmost letters. If $\pi = a(n+1)\pi'$ then π' must avoid 12-3 and end with the pattern 12. The number of such permutations, using the reverse and complement, is given by $N_{1\text{-}23}^{12}(n-1)$. Thus,

$$N_{12\text{-}3}^{12,12}(n+1) = nN_{1\text{-}23}^{12}(n-1).$$

Multiplying both sides of the equality by $x^n/n!$ and summing over all n, we get

$$(E_{12\text{-}3}^{12,12}(x))' = xE_{1\text{-}23}^{12}(x) + x,$$

where the term x corresponds to the permutation 12. We have the desirable result by integrating both sides of the last equality.

Beginning with $12\ldots k$ and ending with $\ell(\ell-1)\ldots1$: All the cases but $k = 2$ and $n \geq \ell + 1$ are easy to prove. Let us consider this case. Using the reverse and complement operations, instead of considering avoiding 1-23, beginning with 12 and ending with $\ell(\ell-1)\ldots1$, we consider avoiding 12-3, beginning with $\ell(\ell-1)\ldots1$ and ending with 12, which we find to be more easy. Let an n-permutation π satisfies all the conditions. We observe, that the letter n is either in the first position, or in position j, where $k+1 \leq j \leq n-2$, or in the last position. Obviously, in the first of these cases the number of "good" permutations is given by $N_{12\text{-}3}^{(\ell-1)(\ell-2)\ldots1,12}(n-1)$, which is equivalent to $N_{1\text{-}23}^{12,(\ell-1)(\ell-2)\ldots1}(n-1)$ by using the reverse and complement. In the second case, we choose the letters to the left of n in $\binom{n-1}{j-1}$ ways, rearrange them to the decreasing order (we do it since otherwise we have an occurrence of the pattern 12-3 having the letter n). After that, the letters to the right of n must form a permutation that avoid 12-3 and end with the pattern 12. Using the reverse and complement, there are $N_{1\text{-}23}^{12}(n-j)$ such permutations. So, totally, in the

second case there are $\sum_{j=\ell+1}^{n-2}\binom{n-1}{j-1}N_{1\text{-}23}^{12}(n-j)$ permutations. Finally, if n is at the last position, we have the only one such permutation, since the other letters must be in the decreasing order.

Beginning with $k(k-1)\dots1$ and ending with $\ell(\ell-1)\dots1$: The only interesting case here is the case $k\geq 2$ and $\ell\geq 2$. Using the reverse and complement operations, instead of considering avoiding 1-23, beginning with $k(k-1)\dots1$ and ending with $\ell(\ell-1)\dots1$, we consider avoiding 12-3, beginning with $\ell(\ell-1)\dots1$ and ending with $k(k-1)\dots1$, which we find to be more easy. Let an n-permutation π satisfies all the conditions. We observe, that the letter n is either in the first position, or in position j, where $\ell+1\leq j\leq n-k$, or in the last position $n-k+1$. We proceed as in the previous case to get the following

$$N_{12\text{-}3}^{\ell(\ell-1)\dots1,k(k-1)\dots1}=N_{12\text{-}3}^{(\ell-1)\dots1,k(k-1)\dots1}+\sum_{i=\ell+1}^{n-k}\binom{n-1}{i-1}N_{1\text{-}23}^{k(k-1)\dots1}(n-i)+\binom{n-1}{k-1},$$

where three terms in the right-hand side correspond to the three cases described above. We now multiply both sides of the equality by $x^n/n!$, sum over n and observe the following detail. We cannot write instead of $i=\ell+1$ (in the sum above) $i=1$ as we did in most of the cases above, since, for instance, the case $i=1$ do not necessarily make the term of summation equal 0 as it was before. Thus, instead of the factor e^x, we have the factor $\left(e^x-\sum_{i=0}^{\ell-1}\frac{x^i}{i!}\right)$ □

4.8 Avoiding a pattern xy-z, beginning and ending with certain patterns simultaneously

Proposition 19. *We have*

(i) $G_{13\text{-}2}^{12\dots k,12\dots\ell}(x)=x^{k+\ell-1}C^{\ell+1}(x)+\frac{x^m-x^{k+\ell-1}}{1-x}$.

(ii) $G_{13\text{-}2}^{12\dots k,\ell(\ell-1)\dots1}(x)=x^{k+\ell-1}C^2(x)$.

(iii) $G_{13\text{-}2}^{k(k-1)\dots1,\ell(\ell-1)\dots1}(x)=x^{k+\ell-1}C^{k+1}(x)+\frac{x^m-x^{k+\ell-1}}{1-x}$, *where* $m=\max(k,\ell)$.

(iv) *the generating function* $G_{13-2}(x,y,z)=\sum_{k,\ell\geq0}G_{13\text{-}2}^{k(k-1)\dots1,12\dots\ell}(z)y^kz^\ell$ *for the sequence* $\{G_{13\text{-}2}^{k(k-1)\dots1,12\dots\ell}(x)\}_{k,\ell\geq0}$ (*where* k *and* ℓ *go through all natural numbers*) *is*

$$\frac{1}{1-x(y+z)}\left(x(y+z+yz)+\frac{C(x)-1}{(1-xyC(x))(1-xzC(x))}\right).$$

Proof. We apply the reverse and complement operations and then use the results of Proposition 17. For example, to avoid 2-13, begin with $12\dots k$ and end with $12\dots\ell$ is the same as to avoid 13-2, begin with $12\dots\ell$ and end with $12\dots k$. □

Proposition 20. *We have*

(i) $E_{21\text{-}3}^{12\ldots k,1}(x) = E_{21\text{-}3}^{12\ldots k}(x)$ *is given by* [KitMans, Proposition 14]. *For* $\ell \geq 2$, $E_{21\text{-}3}^{12\ldots k,12\ldots \ell}(x)$ *satisfies*

$$\frac{\partial}{\partial x} E_{21\text{-}3}^{12\ldots k,12\ldots \ell}(x) = \left(e^x - \sum_{i=0}^{k-2} \frac{x^i}{i!} \right) E_{1\text{-}32}^{12\ldots \ell}(x) + e^x x^{max(\ell,k)-1}.$$

where $E_{1\text{-}32}^{12\ldots \ell}(x) = E_{3\text{-}12}^{\ell(\ell-1)\ldots 1}(x)$ *is given by* [KitMans, Proposition 5].

(ii) *For* $\ell \geq 2$, $E_{21\text{-}3}^{12\ldots k,\ell(\ell-1)\ldots 1}(x)$ *satisfies*

$$\frac{\partial}{\partial x} E_{21\text{-}3}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) = \left(e^x - \sum_{i=0}^{k-2} \frac{x^i}{i!} \right) E_{1\text{-}32}^{\ell(\ell-1)\ldots 1}(x) + \left(e^x - \sum_{i=0}^{k-2} \frac{x^i}{i!} \right) \frac{x^{\ell-1}}{(\ell-1)!},$$

where $E_{1\text{-}32}^{\ell(\ell-1)\ldots 1}(x)$ *is given by* [KitMans, Proposition 4].

(iii) $E_{21\text{-}3}^{k(k-1)\ldots 1,12\ldots \ell}(x)$ *satisfies*

$$\frac{\partial^{k-1}}{\partial x^{k-1}} E_{21\text{-}3}^{k(k-1)\ldots 1,12\ldots \ell}(x) = \begin{cases} e^{e^x} \int_0^x e^{-e^t} \sum\limits_{n \geq \ell-1} \dfrac{t^n}{n!}\, dt, & \text{if } \ell \geq 2, \\ e^{e^x - 1}, & \text{if } \ell = 1. \end{cases}$$

(iv) $E_{21\text{-}3}^{k(k-1)\ldots 1,\ell(\ell-1)\ldots 1}(x)$ *satisfies*

$$\frac{\partial^{k-1}}{\partial x^{k-1}} \left(E_{2\text{-}13}^{k(k-1)\ldots 1,\ell(\ell-1)\ldots 1}(x) - \frac{x^{max(k,\ell)} - x^{k+\ell-1}}{1-x} \right) =$$

$$\begin{cases} \frac{e^{e^x}}{(\ell-1)!} \int_0^x t^{\ell-1} e^{-e^t + t}\, dt, & \text{if } \ell \geq 2, \\ e^{e^x - 1}, & \text{if } \ell = 1. \end{cases}$$

Proof. We apply the reverse and complement operations and then use the results of Proposition 16. For example, to avoid 1-32, begin with $12\ldots k$ and end with $12\ldots \ell$ is the same as to avoid 21-3, begin with $12\ldots \ell$ and end with $12\ldots k$. □

Proposition 21. *We have*

(i) $E_{12\text{-}3}^{12\ldots k,12\ldots \ell}(x) = \begin{cases} 0, & \text{if } k \geq 3 \text{ or } \ell \geq 3, \\ E_{12\text{-}3}^{12\ldots k}(x), & \text{if } \ell = 1, \\ E_{1\text{-}23}^{12\ldots \ell}(x), & \text{if } k = 1, \\ \int_0^x t E_{1\text{-}23}^{12}(t)\, dt + \frac{x^2}{2!}, & \text{if } k = 2 \text{ and } \ell = 2, \end{cases}$

where $E_{12\text{-}3}^{12\ldots k}(x)$ *and* $E_{1\text{-}23}^{12\ldots k}(x)$ *are given in Proposition 18.*

(ii) $E_{1\text{-}23}^{12\ldots k,\ell(\ell-1)\ldots 1}(x) = \begin{cases} 0, & \text{if } k \geq 3, \\ \frac{1}{(\ell-1)!} \int_0^x \int_0^t t m^{\ell-1} e^{e^t - e^m + m}\, dm dt + \frac{\ell x^{\ell+1}}{(\ell+1)!}, & \text{if } k = 2, \\ (e^{e^x}/(\ell-1)!) \int_0^x t^{\ell-1} e^{-e^t + t}\, dt, & \text{if } k = 1; \end{cases}$

(iv) $N_{12\text{-}3}^{k(k-1)\ldots1,12\ldots\ell}(n) =$

$$\begin{cases} 0, & \text{if } \ell \geq 3, \\ 0, & \text{if } \ell = 2 \text{ and } n \leq k, \\ 1 + N_{12\text{-}3}^{(k-1)(k-2)\ldots1,12}(n-1) + \displaystyle\sum_{j=k+1}^{n-2} \binom{n-1}{j-1} N_{3\text{-}21}^{21}(n-j), & \text{if } \ell = 2 \text{ and } n \geq k+1, \\ N_{12\text{-}3}^{k(k-1)\ldots1}(n), & \text{if } \ell = 1, \end{cases}$$

where the numbers $N_{12\text{-}3}^{k(k-1)\ldots1}(n)$ *are given in* [KitMans, Proposition 9]*, and the numbers* $N_{3\text{-}21}^{21}(n)$ *are given by expending the exponential generating functions in* [KitMans, Proposition 6].

(iv) $N_{12\text{-}3}^{k(k-1)\ldots1,1}(n) = N_{12\text{-}3}^{k(k-1)\ldots1}(n)$ *is given by* [KitMans, Proposition 9]*, and* $N_{12\text{-}3}^{1,\ell(\ell-1)\ldots1}(n) = N_{1\text{-}23}^{\ell(\ell-1)\ldots1}(n)$ *is given by* [KitMans, Proposition 4]. *For* $k \geq 2$ *and* $\ell \geq 2$, $E_{12\text{-}3}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x)$ *satisfies*

$$\frac{\partial}{\partial x} E_{12\text{-}3}^{k(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) =$$

$$E_{12\text{-}3}^{(k-1)\ldots1,\ell(\ell-1)\ldots1}(x) + \left(e^x - \sum_{i=0}^{k-1} \frac{x^i}{i!}\right)\left(E_{1\text{-}23}^{\ell(\ell-1)\ldots1}(x) + \frac{x^\ell}{(\ell-1)!}\right).$$

Proof. We apply the reverse and complement operations and then use the results of Proposition 18. For example, to avoid 1-23, begin with $12\ldots k$ and end with $12\ldots\ell$ is the same as to avoid 12-3, begin with $12\ldots\ell$ and end with $12\ldots k$. □

4.9 Further results

In this section, we propose two directions of generalization of the results from the previous sections. The first one is a consideration of avoiding more than one pattern, beginning with some pattern and ending with another pattern. For example, suppose that $v = 12\text{-}3$, $w = 21\text{-}3$, $p = 12\ldots k$, $q = 12\ldots\ell$, and $E_{v,w}^{p,q}(x)$ denotes the exponential generating function for the number of permutations that avoid the patterns v and w simultaneously, begin with the pattern p and end with the pattern q. It is easy to see that if $k \geq 3$ or $\ell \geq 3$ then $E_{12\text{-}3,21\text{-}3}^{12\ldots k,12\ldots\ell}(x) = 0$. For the other k and ℓ, one can prove the following theorem:

Theorem 1. *We have*

(i) $E_{12\text{-}3,21\text{-}3}^{1,1}(x) = e^{x+x^2/2} - 1$.

(ii) $E_{12\text{-}3,21\text{-}3}^{1,12}(x) = e^{x+x^2/2}\left(1 - \int_0^x e^{-t-t^2/2}dt\right) - 1$.

(iii) $E_{12\text{-}3,21\text{-}3}^{12,1}(x) = \int_0^x te^{t+t^2/2}dt$.

(iv) $E_{12\text{-}3,21\text{-}3}^{12,12}(x) = \frac{1}{2}x^2 + \int_0^x \left[e^{t+t^2/2}\left(1 - \int_0^t e^{-r-r^2/2}dr\right) - 1\right]dt$.

The second direction is a consideration of permutations in S_n containing a pattern v exactly r times, beginning with some pattern and ending with another pattern. For example, suppose that $v = 12\text{-}3$, $r = 1$, $p = 1 \dots k$, $q = 1$, and $N_{v;r}^{p,q}(n)$ denotes the number of n-permutations that contain the pattern v exactly r times, begin with the pattern p, and end with the pattern q. It is easy to see that the only interesting case is $1 \leq k \leq 3$, since otherwise $N_{12\text{-}3;1}^{12\dots k,1}(n) = 0$. Moreover, one can prove the following theorem:

Theorem 2. *Let F_n denote the number of n-permutations containing* 12-3 *exactly once. Then, for all $n \geq 3$,*

$$N_{12\text{-}3;1}^{1,1}(n) = F_n N_{12\text{-}3;1}^{12,1}(n) = (n-1)F_{n-1} + (n-2)B_{n-2},$$
$$N_{12\text{-}3;1}^{123,1}(n) = (n-2)B_{n-3},$$

where B_n is the nth Bell number, and F_n is given by [ClaesMans2, Corollarly 13].

Bibliography

[BabStein] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, *Séminaire Lotharingien de Combinatoire*, B44b:18pp, 2000.

[Bon] M. Bóna: Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A* **80** (1997), no. 2, 257–272.

[B] M. Bóna: The permutation classes equinumerous to the smooth class. *Electron. J. Combin. 5* (1998), no. **1**, Research Paper 31, 12 pp. (electronic).

[CW] T. Chow and J. West: Forbidden subsequences and Chebyshev polynomials. *Discrete Math.* **204** (1999), no. 1-3, 119–128.

[Claes] A. Claesson: Generalised Pattern Avoidance, *European J. Combin.* **22** (2001), 961-971.

[ClaesMans1] A. Claesson and T. Mansour: Permutations avoiding a pair of generalized patterns of length three with exactly one dash, preprint CO/0107044.

[ClaesMans2] A. Claesson and T. Mansour, Counting Occurrences of a Pattern of Type (1,2) or (2,1) in Permutations, *Advances in Applied Mathematics*, to appear (2002).

[ElizNoy] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, *Proceedings of FPSAC 2001.*

[Ent] R. Entringer: A Combinatorial Interpretation of the Euler and Bernoulli Numbers, *Nieuw. Arch. Wisk.* **14** (1966), 241–246.

[Kit1] S. Kitaev: Multi-avoidance of generalised patterns, *Discrete Math.*, to appear (2002).

[Kit2] S. Kitaev: Partially ordered generalized patterns, *Discrete Math.*, to appear (2002).

[Kit3] S. Kitaev: Generalized pattern avoidance with additional restrictions, preprint math.CO/0205215.

[KitMans] S. Kitaev and T. Mansour: Simultaneous avoidance of generalized patterns, preprint math.CO/0205182.

[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Kr] C. Krattenthaler: Permutations with restricted patterns and Dyck paths, *Adv. in Appl. Math.* **27** (2001), 510–530.

[K] D. Kremer: Permutations with forbidden subsequnces and a generalized Schröder number, *Discrete Math.* **218** (2000), 121–130.

[Loth] M. Lothaire: *Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, **17**, Addison-Wesley Publishing Co., Reading, Mass. (1983).

[Mans1] T. Mansour: Continued fractions and generalized patterns, *European Journal of Combinatorics*,**23:3** (2002), 329–344.

[Mans2] T. Mansour: Continued fractions, statistics, and generalized patterns, to appear in Ars Combinatorica (2002), preprint CO/0110040.

[Mans3] T. Mansour: Restricted 1-3-2 permutations and generalized patterns, *Annals of Combinatorics* **6** (2002), 65–76.

[MV1] T. Mansour and A. Vainshtein: Restricted permutations, continued fractions, and Chebyshev polynomials, *Electron. J. Combin.* **7** (2000) no. 1, Research Paper 17, 9 pp. (electronic).

[MV2] T. Mansour and A. Vainshtein: Restricted 132-avoiding permutations, *Adv. in Appl. Math.* **126** (2001), no. 3, 258–269.

[MV3] T. Mansour and A. Vainshtein: Layered restrictions and Chebyshev polynomials, *Annals of Combinatorics* **5** (2001), 451–458.

[MV4] T. Mansour and A. Vainshtein: Restricted permutations and Chebyshev polyonomials, *Séminaire Lotharingien de Combinatoire* **47** (2002), Article B47c.

[R] A. Robertson: Permutations containing and avoiding 123 and 132 patterns, *Discrete Math. Theor. Comput. Sci.* **3** (1999), no. 4, 151–154 (electronic).

[RWZ] A. Robertson, H. Wilf, and D. Zeilberger: Permutation patterns and continued fractions, *Electron. J. Combin.* **6** (1999), no. 1, Research Paper 38, 6 pp. (electronic).

[SloPlo] N. J. A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, (1995).

[Stan] R. Stanley: *Enumerative Combinatorics*, Vol. **1**, Cambridge University Press, (1997).

[SchSim] R. Simion, F. Schmidt: Restricted permutations, *European J. Combin.* **6** (1985), no. 4, 383–406.

[W] J. West: Generating trees and forbidden subsequences, *Discrete Math.* **157** (1996), 363–372.

Paper V

Partially Ordered Generalized Patterns

Partially Ordered Generalized Patterns

Sergey Kitaev

E-mail: kitaev@math.chalmers.se
Matematik, Chalmers tekniska högskola och Göteborgs universitet,
S-412 96 Göteborg, Sweden

Abstract

We introduce partially ordered generalized patterns (POGPs), which further generalize the generalized permutation patterns (GPs) introduced by Babson and Steingrímsson [BabStein]. A POGP p is a GP some of whose letters are incomparable. Thus, in an occurrence of p in a permutation π, two letters that are incomparable in p pose no restrictions on the corresponding letters in π. We describe many relations between POGPs and GPs and give general theorems about the number of permutations avoiding certain classes of POGPs. These theorems have several known results as corollaries but also give many new results. We also give the generating function for the entire distribution of the maximum number of non-overlapping occurrences of a pattern p with no dashes, provided we know the e.g.f. for the number of permutations that avoid p.

5.1 Introduction and Background

All permutations in this paper are written as words $\pi = a_1 a_2 \cdots a_n$, where the a_i consist of all the integers $1, 2, \ldots, n$.

We will be concerned with *patterns* in permutations. A pattern is a word on some alphabet of letters, where some of the letters may be separated by dashes. In our notation, the classical permutation patterns, first studied systematically by Simion and Schmidt [SchSim], are of the form $p = 1 - 3 - 2$, the dashes indicating that the letters in a permutation corresponding to an occurrence of p don't have to be adjacent. In the classical case, an occurrence of a pattern p in a permutation π is a subsequence in π (of the same length as the length of p) whose letters are in the same relative order as those in p. For example, the permutation 41352 has only one occurrence of the pattern $1-2-3$, namely the subword 135.

Note that a classical pattern should, in our notation, have dashes at the beginning and end. Since all patterns considered in this paper satisfy this, we suppress these dashes from the notation. Thus, a pattern with no dashes corresponds to a contiguous subword anywhere in a permutation.

In [BabStein] Babson and Steingrímsson introduced *generalized permutation patterns (GPs)* where two adjacent letters in a pattern may be required to be adjacent in the permutation. Such an adjacency requirement is indicated by the absence of a dash between the corresponding letters in the pattern. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern

2-31, namely the subword 564, but the pattern 2-3-1 occurs also in the subwords 562 and 563. The motivation for introducing these patterns in [BabStein] was the study of Mahonian statistics.

A number of interesting results on GPs were obtained by Claesson in [Claes]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were shown there. In [Kit] the present author investigated simultaneous avoidance of two or more 3-letter GPs with no dashes. This work is of particular interest here since avoidance of the patterns considered in this paper has a close connection to simultaneous avoidance of two or more GPs with no dashes. Also important here is the work of Elizalde and Noy [ElizNoy] where they find the distribution of several patterns with no dashes.

In this paper we introduce a further generalization of GPs — namely *partially ordered generalized patterns (POGP)*. A POGP is a GP some of whose letters are incomparable. For instance, if we write $p = 1 - 1'2'$ then we mean that in an occurrence of p in a permutation π the letter corresponding to the 1 in p can be either larger or smaller than the letters corresponding to $1'2'$. Thus, the permutation 13425 has four occurrences of p, namely 134, 125, 325 and 425.

We consider two particular classes of POGPs — *shuffle patterns* and *multi-patterns*. A multi-pattern is of the form $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ and a shuffle pattern is of the form $p = \sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - a_k - \sigma_k$, where for any i and j, the letter a_i is greater than any letter of σ_j and for any $i \neq j$ each letter of σ_i is incomparable with any letter of σ_j These patterns are investigated in Sections 5.4 and 5.5. A corollary to one of our theorems (Theorem 5) about the shuffle patterns is the result of Claesson [Claes, Proposition 2] that the number of n-permutations that avoid the pattern $12 - 3$ is the n-th Bell number.

Let $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ be an arbitrary multi-pattern and let $A_i(x)$ be the exponential generating function (e.g.f.) for the number of permutations that avoid σ_i for each i. In Theorem 11 we find the e.g.f., in terms of the $A_i(x)$, for the number of permutations that avoid p. In particular, this allows us to find the e.g.f. for the entire *distribution* of the maximum number of non-overlapping occurrences of a pattern p with no dashes, if we only know the e.g.f. for the number of permutations that *avoid* p. In many cases, this gives nice generating functions.

We also give alternative proofs, using inclusion-exclusion, of some of the results of Elizalde and Noy [ElizNoy]. Our proofs result in explicit formulas for the e.g.f. in terms of infinite series whereas Elizalde and Noy obtained differential equations for the same e.g.f..

5.2 Definitions and Preliminaries

A *partially ordered generalized pattern (POGP)* is a GP where some of the letters can be incomparable.

Example 1. *The simplest non-trivial example of a POGP that differs from the ordinary GPs is* $p = 1' - 2 - 1''$*, where the second letter is the greatest one and*

the first and the last letters are incomparable to each other. The permutation 3142 has two occurrences of p, namely, the subwords 342 and 142.

It is easy to see that the number of permutations that avoid p in Example 1 is equal to 2^{n-1}. Indeed, if $\pi = a_1 \dots a_n$ and a_i is the leftmost letter in π that is smaller than its successor, then all letters to the right of a_i must be in increasing order. So any permutation π avoiding p can be written as $\pi_1 1 \pi_2$, where π_1 is decreasing and π_2 is increasing and there are 2^{n-1} ways to pick the permutation π_1, which determines π.

Definition 1. *If the number of permutations in S_n, for each n, that avoid a POGP p is equal to the number of permutations that avoid a POGP q, then p and q are said to be* equivalent *and we write $p \equiv q$ in this case.*

If A_n is the number of n-permutations that avoid a pattern p, then the *exponential generating function*, or *e.g.f.*, of the class of such permutations is

$$A(x) = \sum_{n \geq 0} A_n \frac{x^n}{n!}.$$

We will talk about *bivariate generating functions*, or *b.g.f.*, exclusively as generating functions of the form

$$A(u,x) = \sum_{\pi} u^{p(\pi)} \frac{x^{|\pi|}}{|\pi|!} = \sum_{n,k \geq 0} A_{n,k} u^k \frac{x^n}{n!},$$

were $A_{n,k}$ is the number of n-permutations with k occurrences of the pattern p.

The *reverse* $R(\pi)$ of a permutation $\pi = a_1 a_2 \dots a_n$ is the permutation $a_n a_{n-1} \dots a_1$. The *complement* $C(\pi)$ is the permutation $b_1 b_2 \dots b_n$ where $b_i = n+1-a_i$. Also, $R \circ C$ is the composition of R and C. For example, $R(13254) = 45231$, $C(13254) = 53412$ and $R \circ C(13254) = 21435$. We call these bijections of $\mathcal{S}_n$ to itself *trivial*, and it is easy to see that any pattern p is equivalent to the patterns $R(p)$, $C(p)$ and $R \circ C(p)$. For example, the number of permutations that avoid the pattern 132 is the same as the number of permutations that avoid the patterns 231, 312 and 213, respectively.

It is convenient to introduce the following definition.

Definition 2. *Let p be a GP without internal dashes. A permutation π* quasi-avoids *p if π has exactly one occurrence of p and this occurrence consists of the $|p|$ rightmost letters of π.*

For example, the permutation 51342 quasi-avoids the pattern $p = 231$, whereas the permutations 54312 and 45231 do not. Indeed, 54312 ends with 312, which is not an occurrence of the pattern p, and 45231 has an occurrence of p, namely 452, in a forbidden place.

Proposition 1. *Let p be a non-empty GP with no dashes. Let $A(x)$ (resp. $A^\star(x)$) be the e.g.f. for the number of permutations that avoid (resp. quasi-avoid) p. Then*

$$A^\star(x) = (x-1)A(x) + 1.$$

Proof. We first show that

$$A_n^\star = nA_{n-1} - A_n. \tag{5.1}$$

If we consider all $(n-1)$-permutations that avoid p and all possible extending of these permutations to the n-permutations by writing one more letter to the right, then the number of obtained permutations will be nA_{n-1}. Obviously, the set of these permutations is a disjoint union of the set of all n-permutations that avoid p and the set of all n-permutations that quasi-avoid p. Thus we get (5.1). Multiplying both sides of (5.1) with $x^n/n!$ and summing over all natural numbers n, observing that $A_0^\star = 0$, we get the desired result. □

Definition 3. *Suppose $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes and $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ where each letter of σ_i is incomparable with any letter of σ_j whenever $i \neq j$. We call such POGPs* multi-patterns.

Definition 4. *Suppose $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes and $a_1 a_2 \ldots a_k$ is a permutation of k letters. We define a* shuffle *pattern to be a pattern of the form*

$$\sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - \sigma_{k-1} - a_k - \sigma_k,$$

where for any i and j, the letter a_i is greater than any letter of σ_j and for any $i \neq j$ each letter of σ_i is incomparable with any letter of σ_j. We also allow σ_0 and σ_k, but not the other σ_i, to be empty patterns.

The pattern from Example 1 is an example of a shuffle pattern. It follows from the definitions that we can get a multi-pattern from a shuffle pattern by removing all the a_i.

Let $\mathcal{S}_\infty$ denote the disjoint union of the $\mathcal{S}_n$ for all $n \in \mathbb{N}$. The POGPs (which include the GPs, as well as the classical patterns), can be considered as functions from $\mathcal{S}_\infty$ to $\mathbb{N}$ that count the number of occurrences of the pattern in a permutation in $\mathcal{S}_\infty$. This allows us to write a POGP (as a function) as a linear combination of GPs. For example,

$$1' - 2 - 1'' \;=\; (1 - 3 - 2) \;+\; (2 - 3 - 1),$$

from which, in particular, we see that to avoid $1' - 2 - 1''$ is the same as to avoid simultaneously the patterns $1 - 3 - 2$ and $2 - 3 - 1$. A straightforward argument leads to the following proposition.

Proposition 2. *For any POGP p there exists a set S of GPs such that a permutation π avoids p if and only if π avoids all the patterns in S.*

The following theorem can be easily proved by induction on k:

Theorem 1. *Let $p_1 = \sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - \sigma_{k-1} - a_k - \sigma_k$ (resp. $p_2 = \sigma_0 - \sigma_1 - \cdots - \sigma_k$) be an arbitrary shuffle pattern (resp. multi-pattern) with*

$|\sigma_i| = \ell_i$ for all $i = 0, \ldots, k$. Then to avoid the pattern p_1 (resp. p_2) is the same as to avoid

$$\prod_{i=1}^{k} \binom{\ell_0 + \ell_1 + \cdots + \ell_i}{\ell_i} = \binom{\ell_0 + \ell_1}{\ell_1} \binom{\ell_0 + \ell_1 + \ell_2}{\ell_2} \cdots \binom{\ell_0 + \ell_1 + \cdots + \ell_k}{\ell_k}$$

ordinary GPs.

Example 2. *Let $p = 1'2'-3-1''$. That is $\sigma = 12$ and $\tau = 1$. By Theorem 1, to avoid p is the same as to avoid $\binom{3}{2} = 3$ GPs simultaneously, namely $12-4-3$, $13-4-2$ and $23-4-1$.*

There is a number of results on the distribution of several classes of patterns with no dashes. These results can be used as building blocks for some of the results in the present paper. The most important of these is the following result by Elizalde and Noy [ElizNoy]:

Theorem 2. *[ElizNoy, Theorem 3.4] Let m and a be positive integers with $a \leq m$, let $\sigma = 12 \ldots a\tau(a+1) \in \mathcal{S}_{m+2}$, where τ is any permutation of $\{a+2, a+3, \ldots, m+2\}$, and let $P(u,z)$ be the b.g.f. for permutations where u marks the number of occurrences of σ. Then $P(u,z) = 1/w(u,z)$, where w is the solution of*

$$w^{a+1} + (1-u)\frac{z^{m-a+1}}{(m-a+1)!}w' = 0$$

with $w(0) = 1$, $w'(0) = -1$ and $w^{(k)}(0) = 0$ for $2 \leq k \leq a$. In particular, the distribution does not depend on τ.

5.3 GPs with no dashes

In order to apply our results in what follows we need to know how many patterns avoid a given ordinary GP with no dashes. We are also interested in different approaches to studying these patterns. The theorems in this section can be proved using an inclusion-exclusion argument similar to the one given in the proof of Theorem 12 and we omit these proofs. This allows us to get explicit formulas for the e.g.f. in terms of infinite series instead of having to solve differential equations as done by Elizalde and Noy [ElizNoy] for the same e.g.f.. However, in particular cases, we use certain differential equations to simplify our series.

Theorem 3. *[GoulJack] Let $A_k(x)$ be the e.g.f. for the number of permutations avoiding the pattern $p = 123 \ldots k$. Then*

$$A_k(x) = 1/F_k(x),$$

where $F_k(x) = \sum_{i \geq 0} \frac{x^{ki}}{(ki)!} - \sum_{i \geq 0} \frac{x^{ki+1}}{(ki+1)!}$.

For some k it is possible to simplify the function $F_k(x)$ in the theorems above. Indeed, $F_k(x)$ satisfies the differential equation $F_k^{(k)}(x) = F_k(x)$ with the k initial conditions $F_k(0) = 1$, $F_k'(0) = -1$ and $F_k^{(i)}(0) = 0$ for all $i = 2, 3, \ldots, k-1$. For instance, if $k = 4$ then

$$F_4(x) = \frac{1}{2}(\cos x - \sin x + e^{-x}).$$

Theorem 4. *Let k and a be positive integers with $a < k$, let $p = 12\ldots a\tau(a+1) \in \mathcal{S}_{k+1}$, where τ is any permutation of the elements $\{a+2, a+3, \ldots, k+1\}$, and let $A_{k,a}(x)$ be the e.g.f. for the number of permutations that avoid p. Let*

$$F_{k,a}(x) = \sum_{i\geq 1} \frac{(-1)^{i+1}x^{ki+1}}{(ki+1)!} \prod_{j=2}^{i} \binom{jk-a}{k-a}.$$

Then

$$A_{k,a}(x) = 1/(1 - x + F_{k,a}(x)).$$

If $k = 2$ and $a = 1$ in the previous theorem, corresponding to the pattern $p = 132$, then from Theorem 4 the function $F_{2,1}(x)$, which is the same for the patterns p, 231, 312 and 213 because of the trivial bijections, can be written as:

$$F_{2,1}(x) = \sum_{i\geq 1} \frac{(-1)^{i+1}x^{ki+1}}{i!(k!)^i(ki+1)} = x - \int_0^x e^{-t^2/2}\, dt.$$

That is

$$A_{2,1} = \frac{1}{1 - \displaystyle\int_0^x e^{-t^2/2}\, dt},$$

which is a special case of Theorem 4.1 in [ElizNoy].

5.4 The Shuffle Patterns

We recall that according to Definition 4, a shuffle pattern is a pattern of the form $\sigma_0 - a_1 - \sigma_1 - a_2 - \cdots - \sigma_{k-1} - a_k - \sigma_k$, where $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes, $a_1a_2\ldots a_k$ is a permutation of k letters, for any i and j the letter a_i is greater than any letter of σ_j and for any $i \neq j$ each letter of σ_i is incomparable with any letter of σ_j.

Let us consider a shuffle pattern that in fact is an ordinary generalized pattern. This pattern is $p = \sigma - k$, where σ is an arbitrary pattern with no dashes that is built on elements $1, 2, \ldots, k-1$. So the last element of p is greater than any other element.

Theorem 5. *Let $p = \sigma - k$ and let $A(x)$ (resp. $B(x)$) be the e.g.f. for the number of permutations that avoid σ (resp. p). Then $B(x) = e^{F(x,A(y))}$, where*

$$F(x, A(y)) = \int_0^x A(y)\, dy.$$

Proof. Suppose that $\pi \in \mathcal{S}_{n+1}$ and that π avoids p. Suppose the letter $(n+1)$ is in the i-th position and $\pi = \pi_1(n+1)\pi_2$, where π_1 and π_2 might be empty.

Since π is p-avoiding, π_1 must be σ-avoiding, because otherwise an occurrence of σ in π_1 together with the letter $(n+1)$ gives an occurrence of p in π. But if π_1 is σ-avoiding then there is no interaction between π_1 and π_2, that is, if π_2 is p-avoiding and π_1 is σ-avoiding then π is p-avoiding. To see this it is enough to see that if an occurrence of σ in π contains the letter $(n+1)$, then this occurrence of σ can not lead to an occurrence of $p = \sigma - k$ containing the letter $(n+1)$.

From the above, considering all possible positions of $(n+1)$, we get the recurrence relation

$$B_{n+1} = \sum_i \binom{n}{i} A_i B_{n-i},$$

where B_j (resp. A_j) is the number of j-permutations that avoid p (resp. σ), because we can choose the elements of π_1 in $\binom{n}{i}$ ways.

Multiplying both sides of the equality by $x^n/n!$ we get

$$\frac{B_{n+1}}{n!}x^n = \sum_i \frac{A_i}{i!}x^i \frac{B_{n-i}}{(n-i)!}x^{n-i}.$$

Taking the sum over all natural numbers n leads us to

$$B'(x) = A(x)B(x)$$

where the derivative of B is with respect to x. Since $B(0) = 1$, the solution of the differential equation is $B(x) = e^{F(x,A(y))}$. □

Example 3. *Let $p = 1-2$. Here $\sigma = 1$, so $A(x) = 1$ since $A_n = 0$ for all $n \geq 1$ and $A_0 = 1$. So*

$$B(x) = e^{F(x,1)} = e^x.$$

This corresponds to the fact that for each $n \geq 1$ there is exactly one permutation that avoids the pattern p, namely $\pi = n(n-1)\ldots 1$.

Example 4. *Suppose $p = 12-3$. Here $\sigma = 12$, so $A(x) = e^x$, since there is exactly one permutation that avoids the pattern σ. So*

$$B(x) = \sum_{n\geq 0} \frac{B_n}{n!}x^n = e^{F(x,e^y)} = e^{e^x-1}.$$

According to [Claes, Proposition 2], for all $n \geq 1$, B_n is the n-th Bell number and the e.g.f. for the Bell numbers is e^{e^x-1}.

The table below gives the initial values of B_n for several patterns $p = \sigma - k$. These numbers were obtained by expanding the corresponding $B(x)$. The functions $A(x)$ are taken from the previous section.

pattern	initial values for B_n
132-4	1, 2, 6, 23, 107, 585, 3671, 25986, 204738
123-4	1, 2, 6, 23, 108, 598, 3815, 27532, 221708
1234-5	1, 2, 6, 24, 119, 705, 4853, 38142, 336291
12345-6	1, 2, 6, 24, 120, 719, 5022, 40064, 359400

Theorem 6. *Let p be the shuffle pattern $\sigma - k - \tau$. So k is the greatest letter of the pattern, and each letter of σ is incomparable with any letter of τ. Let $A(x)$, $B(x)$ and $C(x)$ be the e.g.f. for the number of permutations that avoid σ, τ and p respectively. Then $C(x)$ is the solution of the differential equation*

$$C'(x) = (A(x) + B(x))C(x) - A(x)B(x),$$

with $C(0) = 1$.

Proof. As before, we consider the symmetric group $\mathcal{S}_{n+1}$ and a permutation $\pi \in \mathcal{S}_{n+1}$ that avoids p. Suppose the letter $(n+1)$ is in the i-th position and $\pi = \pi_1(n+1)\pi_2$, where π_1 and π_2 might be empty.

There are exactly four mutually exclusive possibilities:

1) π_1 does not avoid σ, π_2 does not avoid τ.

2) π_1 avoids σ, π_2 does not avoid τ;

3) π_1 does not avoid σ, π_2 avoids τ;

4) π_1 avoids σ, π_2 avoids τ;

Obviously, the situation 1) is impossible, since an occurrence of σ in π_1 with $(n+1)$ and with an occurrence of τ in π_2 gives us an occurrence of p in π. On the other hand, if p occurs in π then it is easy to see that the letter $(n+1)$ cannot be one of the letters in the occurrences of σ or τ, so all p-avoiding permutations are described by the possibilities 2)–4). We count these permutations in the following way.

In $\binom{n}{i}$ ways we choose first i elements from the letters $1, 2 \ldots n$, that is, the elements of π_1. Let A_i, B_i and C_i be the number of i-permutations that avoid σ, τ and p respectively.

If π_1 is σ-avoiding, we let π_2 be any p-avoiding permutation of the remaining $(n-i+1)$ letters. This accounts for all "good" permutations from the possibilities 2) and 4). There are $\binom{n}{i}A_iC_{n-i}$ such permutations.

If π_2 is τ-avoiding, we let π_1 be any p-avoiding permutation of chosen i letters. This covers all "good" permutations from 3) and 4). There are $\binom{n}{i}B_iC_{n-i}$ such permutations.

But we have counted p-avoiding permutations that correspond to 4) twice, so we must subtract $\binom{n}{i}A_iB_{n-i}$, which is the number of such permutations.

So we have

$$C_{n+1} = \sum_i \binom{n}{i} (A_iC_{n-i} + B_iC_{n-i} - A_iB_{n-i}).$$

Multiplying both sides of the equality by $x^n/n!$ we get

$$\frac{C_{n+1}}{n!}x^n = \sum_i \left(\frac{A_i + B_i}{i!}x^i \frac{C_{n-i}}{(n-i)!}x^{n-i} - \frac{A_i}{i!}x^i \frac{B_{n-i}}{(n-i)!}x^{n-i} \right),$$

so

$$C'(x) = (A(x) + B(x))C(x) - A(x)B(x).$$

□

Example 5. *Let $p = 1'-2-1''$. That is, $\sigma = 1$ and $\tau = 1$. So $A(x) = B(x) = 1$ and we need to solve the equation*

$$C'(x) = 2C(x) - 1$$

with $C(0) = 1$. The solution of this equation is $C(x) = \frac{1}{2}(e^{2x} + 1)$, so for all $n \geq 1$ we have $C_n = 2^{n-1}$, as in Example 1.

In the table below we record the initial values of C_n for several patterns $p = \sigma - k - \tau$.

σ	τ	**initial values for C_n**
1	12	1, 2, 6, 21, 82, 354, 1671, 8536, 46814
1	132	1, 2, 6, 24, 116, 652, 4178, 30070, 240164
1	123	1, 2, 6, 24, 116, 657, 4260, 31144, 253400
1	1234	1, 2, 6, 24, 120, 715, 4946, 38963, 344350
12	12	1, 2, 6, 24, 114, 608, 3554, 22480, 152546
12	132	1, 2, 6, 24, 120, 710, 4800, 36298, 302780
12	123	1, 2, 6, 24, 120, 710, 4815, 36650, 308778
12	1234	1, 2, 6, 24, 120, 720, 5025, 39926, 355538
123	123	1, 2, 6, 24, 120, 720, 5020, 39790, 352470
123	132	1, 2, 6, 24, 120, 720, 5020, 39755, 351518
132	132	1, 2, 6, 24, 120, 720, 5020, 39720, 350496

Remark 2. *The pattern $p = \sigma - k$ from Theorem 5 is a particular case of the pattern $p = \sigma - k - \tau$ from Theorem 6 when τ is the empty word. The e.g.f. for the number of permutations that avoid the empty word is zero, because no permutation avoids the empty word. So if τ is empty, we can use Theorem 6 to get Theorem 5. Indeed, $B(x) = 0$, and after renaming C with B we get in Theorem 6 exactly the same differential equation as we have in Theorem 5.*

We now give two corollaries to Theorem 6.

Corollary 1. *Suppose we have the shuffle pattern $p = \sigma - k - \tau$. We consider the pattern $\varphi(p) = \varphi_1(\sigma) - k - \varphi_2(\tau)$, where φ_1 and φ_2 are any trivial bijections. Then $p \equiv \varphi(p)$.*

Proof. We just observe that if $A(x)$ (resp. $B(x)$) is the e.g.f. for the number of permutations that avoid σ (resp. τ) then $A(x)$ (resp. $B(x)$) is the e.g.f. for the number of permutations that avoid $\varphi_1(\sigma)$ (resp. $\varphi_2(\tau)$). □

Corollary 2. *We have* $\sigma - k - \tau \equiv \tau - k - \sigma$.

Proof. This follows directly from the differential equation of Theorem 6 ($A(x)$ and $B(x)$ are symmetric in that equation), but we can also obtain this as a corollary to Corollary 1. By Corollary 1, the pattern $\sigma - k - \tau$ is equivalent to the pattern $\sigma - k - R(\tau)$. Reversing the pattern $\sigma - k - R(\tau)$, we obtain the pattern

$$R(\sigma - k - R(\tau)) = R(R(\tau)) - k - R(\sigma) = \tau - k - R(\sigma),$$

which thus is equivalent to $\sigma - k - \tau$. Finally, we use Corollary 1 one more time to get

$$\tau - k - R(\sigma) \equiv \tau - k - R(R(\sigma)) = \tau - k - \sigma.$$

□

5.5 The Multi-Patterns

We recall that according to Definition 3, a multi-pattern is a pattern $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$, where $\{\sigma_0, \sigma_1, \ldots, \sigma_k\}$ is a set of GPs with no dashes and each letter of σ_i is incomparable with any letter of σ_j whenever $i \neq j$.

We first discuss patterns of the type $p = \sigma - \tau$ which are a particular case of the multi-patterns to be treated in this section.

If σ or τ is the empty word then we are dealing with ordinary GPs with no dashes, some of which were investigated in [ElizNoy] and Section 3. The analysis of the case when σ or τ is equal to 1 can also be reduced to the analysis of ordinary GPs. For example, suppose that $\sigma = 1$, that is, $p = 1 - \tau$, and we want to count the number of permutations in $\mathcal{S}_n$ that avoid p. We can choose the leftmost letter of a permutation avoiding p in n ways, then the remainder of the permutation must avoid τ, so we multiply n by the number of permutations in $\mathcal{S}_{n-1}$ that avoid τ. For instance, if $p = 1 - 1'2'$ then the number of permutations in $\mathcal{S}_n$ avoiding p is exactly n.

Theorem 7. *Let* $p = \sigma - \tau$ *and* $q = \varphi_1(\sigma) - \varphi_2(\tau)$, *where* φ_1 *and* φ_2 *are any of the trivial bijections. Then* p *and* q *are equivalent.*

Proof. The theorem is equivalent to the following statement:

Let $p = \sigma - \tau$ and $q = \sigma - \varphi(\tau)$, where φ is a trivial bijection. Then p and q are equivalent.

It is obvious that the statement follows from Theorem 7. Conversely, suppose we have $p = \sigma - \tau$. We observe that any two trivial bijections commute, that is for any trivial bijection ψ, we have $\psi(R(x)) = R(\psi(x))$. This observation, the statement and the fact that $x \equiv R(x)$ give

$$p = \sigma - \tau \equiv \sigma - \varphi_2(\tau) \equiv R(\varphi_2(\tau)) - R(\sigma) \equiv R(\varphi_2(\tau)) - \varphi_1(R(\sigma)) \equiv$$

$$R(\varphi_2(\tau)) - R(\varphi_1(\sigma)) \equiv \varphi_1(\sigma) - \varphi_2(\tau).$$

So to prove the theorem we now prove the statement.

Let $p = \sigma - \tau$ and $q = \sigma - \varphi(\tau)$, where φ is a trivial bijection. Let A_n (resp. B_n) be the number of n-permutations that avoid p (resp. q). We are going to prove that $A_n = B_n$.

Suppose π avoids p and $\pi = \pi_1\sigma'\pi_2$, where $\pi_1\sigma'$ has exactly one occurrence of the pattern σ, namely σ'. Then π_2 must avoid τ, $\varphi(\pi_2)$ must avoid $\varphi(\tau)$ and $\pi_\varphi = \pi_1\sigma'\varphi(\pi_2)$ avoids q. The converse is also true, that is, if π_φ has no occurrences of q then π has no occurrences of p. If π has no occurrences of σ then π has no occurrences of p as well as no occurrences of q. Since any permutation either avoids σ or can be factored as above, we have a bijection between the class of permutations that avoid p and the class of permutations that avoid q. Thus $A_n = B_n$. □

We get the following corollary to Theorem 7:

Corollary 3. *The pattern $\sigma - \tau$ is equivalent to the pattern $\tau - \sigma$.*

Proof. We proceed as in the proof of Corollary 2. From Theorem 7 we have:

$$\sigma - \tau \equiv \sigma - R(\tau) \equiv R(R(\tau)) - R(\sigma) \equiv \tau - R(R(\sigma)) \equiv \tau - \sigma.$$

□

We observe that the presence of the dash in the patterns in Theorem 7 is essential. That is, generally speaking, the pattern $\sigma\tau$ is not equivalent to the pattern $\varphi_1(\sigma)\varphi_2(\tau)$ for any trivial bijections φ_1 and φ_2. For example, there are 66 permutations in $\mathcal{S}_5$ that avoid the pattern $122'1'$ but only 61 that avoid $121'2'$. In Section 6 we investigate the pattern $122'1'$.

Theorem 8 and Corollary 4 generalise Theorem 7 and Corollary 3:

Theorem 8. *Suppose we have multi-patterns $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ and $q = \tau_1 - \tau_2 - \cdots - \tau_k$, where $\tau_1\tau_2\ldots\tau_k$ is a permutation of $\sigma_1\sigma_2\ldots\sigma_k$. Then p and q are equivalent.*

Proof. We proceed by induction on k. If $k = 2$ then the statement is true by Corollary 3. Suppose the statement is true for all $k' < k$. Suppose p has exactly k blocks. If a permutation π avoiding p has no occurrences of σ_1 then it obviously avoids both p and q. Otherwise we factor π as $\pi = \pi_1\sigma_1'\pi_2$ where $\pi_1\sigma_1'$ has exactly one occurrence of the pattern σ_1, namely σ_1'. Then π_2 must avoid $\sigma_2 - \cdots - \sigma_k$. Moreover it is irrelevant from which letters $\pi_1\sigma_1'$ is built and therefore we can apply the inductive hypothesis. We can rearrange $\sigma_2'\ldots\sigma_k'$ of $\sigma_2\ldots\sigma_k$ in such a way that the blocks in $\tau_1\tau_2\ldots\tau_k$ corresponding to $\sigma_2, \ldots, \sigma_k$ are arranged in the same order as the τ's. Now we consider separately two cases: $\tau_k \neq \sigma_1$ and $\tau_k = \sigma_1$. In the first case we use the following equivalences:

$$p = \sigma_1 - \sigma_2 - \cdots - \sigma_k \equiv \sigma_1 - \sigma_2' - \cdots - \sigma_k' \equiv R(\sigma_k') - \cdots - R(\sigma_2') - R(\sigma_1).$$

For the pattern $R(\sigma'_k) - \cdots - R(\sigma'_2) - R(\sigma_1)$ we use the factorisation of a permutation π avoiding this pattern, where the role of σ_1 is played by $R(\sigma'_k)$. So by the inductive hypothesis we put the pattern $R(\sigma_1)$ in the right place somewhere to the left of $R(\sigma'_2)$ and apply R to get that $p \equiv q$.

In the second case we have:

$$p \equiv R(\sigma'_k) - \cdots - R(\sigma'_2) - R(\sigma_1) \equiv R(\sigma'_k) - \cdots - R(\sigma_1) - R(\sigma'_2) \equiv$$

$$\sigma'_2 - \sigma_1 - \cdots - \sigma'_k \equiv \sigma'_2 - \cdots - \sigma'_k - \sigma_1 = q$$

The first equivalence here is taken from the considerations above; the second one uses the inductive hypothesis; then we use the fact that $R(R(x)) = x$ and apply the inductive hypothesis again. □

Corollary 4. *Suppose we have multi-patterns $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ and $q = \varphi_1(\sigma_1) - \varphi_2(\sigma_2) - \cdots - \varphi_k(\sigma_k)$, where each φ_i is an arbitrary trivial bijection. Then p and q are equivalent.*

Proof. We use induction on k, Theorem 8 and the factorisation of permutations, which is discussed in the proof of Theorem 8. If $k = 2$ then the statement is true by Theorem 7. Suppose the statement is true for all $k' < k$. Then

$$p = \sigma_1 - \sigma_2 - \cdots - \sigma_k \equiv \sigma_1 - \varphi_2(\sigma_2) - \cdots - \varphi_k(\sigma_k) \equiv$$

$$\varphi_2(\sigma_2) - \sigma_1 - \cdots - \varphi_k(\sigma_k) \equiv \varphi_2(\sigma_2) - \varphi_1(\sigma_1) - \cdots - \varphi_k(\sigma_k) \equiv$$

$$\varphi_1(\sigma_1) - \varphi_2(\sigma_2) - \cdots - \varphi_k(\sigma_k) = q,$$

where first we apply the inductive hypothesis then Theorem 8 then the inductive hypothesis and finally Theorem 8 again. □

Theorem 9. *Suppose $p = \sigma - p'$, where p' is an arbitrary POGP, and the letters of σ are incomparable to the letters of p'. Let $C(x)$ (resp. $A(x)$, $B(x)$) be the e.g.f. for the number of permutations that avoid p (resp. σ, p'). Moreover let $A^\star(x)$ be the e.g.f. for the number of permutations that quasi-avoid σ. Then*

$$C(x) = A(x) + B(x)A^\star(x).$$

Proof. Let A_n, B_n, C_n be the number of n-permutations that avoid the patterns σ, p' and p respectively. Also $A^\star_n$ is the number of n-permutations that quasi-avoid σ. If a permutation π avoids σ then it avoids p. Otherwise we find the leftmost occurrence of σ in π. We assume that this occurrence consists of the $|\sigma|$ rightmost letters among the i leftmost letters of π. So the subword of π beginning at the $(i+1)$st letter must avoid p'. From this we conclude

$$C_n = A_n + \sum_{i=|\sigma|}^{n} \binom{n}{i} A^\star_i B_{n-i}.$$

We observe that we can change the lower bound in the sum above to 0, because $A^\star_i = 0$ for $i = 0, 1, \ldots, |\sigma| - 1$. Multiplying both sides by $x^n/n!$ and taking the sum over all n we get the desired result. □

Corollary 5. *Suppose $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ is a multi-pattern where $|\sigma_i| = 2$ for all i, so each σ_i is equal to either 12 or 21. If $B(x)$ is the e.g.f. for the number of permutations that avoid p then*

$$B(x) = \frac{1 - (1 + (x-1)e^x)^k}{1 - x}.$$

Proof. We use Theorem 9, induction on k and the fact that $A(x) = e^x$ and $A^\star(x) = 1 + (x-1)e^x$. □

The following corollary to Corollary 5 can be proved combinatorially.

Theorem 10. *There are $(n-2)2^{n-1} + 2$ permutations in $\mathcal{S}_n$ that avoid the pattern $p = 12 - 1'2'$ or, according to Theorem 7, the pattern $p = 12 - 2'1'$.*

One more corollary to Theorem 9 is the following theorem that is the basis for calculating the number of permutations that avoid a multi-pattern, and therefore is the main result for multi-patterns in this paper.

Theorem 11. *Let $p = \sigma_1 - \sigma_2 - \cdots - \sigma_k$ be a multi-pattern and let $A_i(x)$ be the e.g.f. for the number of permutations that avoid σ_i. Then the e.g.f. $B(x)$ for the number of permutations that avoid p is*

$$B(x) = \sum_{i=1}^{k} A_i(x) \prod_{j=1}^{i-1} ((x-1)A_j(x) + 1).$$

Proof. We use Theorem 9 and prove by induction on k that

$$B(x) = \sum_{i=1}^{k} A_i(x) \prod_{j=1}^{i-1} A_j^\star(x).$$

Then we use Proposition 1 to get the desired result. □

Remark 3. *One can consider the function $B(x)$ from Theorem 11 as a function in k variables $B(x) = B(A_1(x), A_2(x), \ldots, A_k(x))$. Then, by Theorem 8, this function is symmetric in the variables $A_1(x), A_2(x), \ldots, A_k(x)$. That means that we can rename the variables, which may simplify the calculation of $B(x)$.*

5.6 Patterns of the Form $\sigma\tau$

Theorem 12. *Let $B(x)$ be the e.g.f. for the number of permutations that avoid the pattern $p = 122'1'$. Then*

$$B(x) = \frac{1}{2} + \frac{1}{4}\tan x(1 + e^{2x} + 2e^x \sin x) + \frac{1}{2}e^x \cos x.$$

Proof. Let B_n be the number of n-permutations that avoid p and A_n be the number of n-permutations that avoid p and begin with the pattern 12. Let also $A(x)$ be the e.g.f. for the numbers A_n. We set $B_0 = A_0 = A_1 = 1$. Suppose π is a $(n+1)$-permutation that avoids p. There are three mutually exclusive possibilities:

1) $\pi = (n+1)\pi_2$;

2) $\pi = \pi_1(n+1)$;

3) $\pi = \pi_1(n+1)\pi_2$ and $\pi_1, \pi_2 \neq \varepsilon$.

Obviously, in 1) and 2) the letter $(n+1)$ does not affect the rest of the permutation π, and therefore in each of these cases we have B_n permutations that avoid p. In 3), it is easy to see that if π_1 has more than one letter then π_1 must end with a 21 pattern whereas if π_2 has more than one letter then π_2 must begin with a 12 pattern. The key observation is that the number of n-permutations that avoid p and end with a 21 pattern is the same as the number of n-permutations that avoid p and begin with a 12 pattern. To see this it is enough to apply the reverse function to any n-permutation π that begins with 12-pattern and avoids p and observe that $R(p) = p$, that is, $R(\pi)$ avoids p and ends with a 21 pattern. Obviously this is a bijection. So if $|\pi_1| = i$ then we can choose the letters of π_1 in $\binom{n}{i}$ ways and then choose a permutation π_1 in A_i ways and a permutation π_2 in A_{n-i} ways, since the letters of π_1 and π_2 do not affect each other. From all this we get

$$B_{n+1} = 2B_n + \sum_{i=1}^{n-1} \binom{n}{i} A_i A_{n-i} = 2B_n + \sum_{i=0}^{n} \binom{n}{i} A_i A_{n-i} - 2A_n.$$

We multiply both sides of the last equality by $x^n/n!$ to get

$$B_{n+1}\frac{x^n}{n!} = 2B_n\frac{x^n}{n!} + \sum_{i=0}^{n} \frac{A_i}{i!}x^i \frac{A_{n-i}}{(n-i)!}x^{n-i} - 2A_n\frac{x^n}{n!}.$$

Summing both sides over all natural numbers n we get:

$$B'(x) = 2B(x) + A^2(x) - 2A(x). \tag{5.2}$$

To solve this differential equation with the initial condition $B(0) = 1$, we need to determine $A(x)$. One can observe that if a permutation π avoids p and begins with the pattern 12 then π has the structure $\pi = a_1 b_1 a_2 b_2 a_3 b_3 \cdots$, where $a_i < b_i$ for all i. Moreover, if $b_1 < a_2$ then we must have $a_1 < b_1 < a_2 < b_2 < a_3 < \cdots$ since otherwise we obviously have an occurrence of the pattern p. A first approximation is that $A_n = \binom{n}{2} A_{n-2}$, because we can choose $a_1 b_1$ in π in $\binom{n}{2}$ ways and then pick an arbitrary $(n-2)$-permutation that avoids p and begins with the pattern 12, to be $a_2 b_2 a_3 b_3 \ldots$, in A_{n-2} ways. But it is possible that $b_1 < a_2$ in which case $b_1 a_2 b_2 a_3$ can be an occurrence of p in π, and it is an occurrence

of p unless $a_2 < b_2 < a_3 < \cdots$. So in order to avoid this we must subtract the number of permutations of the form $abcd\pi'$, where $a < b < c < d$ and π' is any $(n-4)$-permutation that avoids p, from the first approximation of A_n. Thus the second approximation is that $A_n = \binom{n}{2}A_{n-2} - \binom{n}{4}A_{n-4}$. We observe that in the second approximation we do not count the increasing permutation $123\ldots n$. Moreover, among the permutations counted by $\binom{n}{4}A_{n-4}$, there are the permutations that begin with 6 increasing letters. Except for the increasing permutation, such permutations are not counted by $\binom{n}{2}A_{n-2}$. We must therefore add the number of such permutations. So the third approximation is that $A_n = \binom{n}{2}A_{n-2} - \binom{n}{4}A_{n-4} + \binom{n}{6}A_{n-6}$ and so on. That is,

$$A_n = \binom{n}{2}A_{n-2} - \binom{n}{4}A_{n-4} + \binom{n}{6}A_{n-6} - \binom{n}{8}A_{n-8} + \cdots = \sum_{i\geq 1}(-1)^{i+1}\binom{n}{2i}A_{n-2i}. \tag{5.3}$$

We observe that if $n = 4k$ or $n = 4k+1$ then we do not count the increasing permutation in our sum. This, together with Equation 5.3, gives us

$$\sum_{i\geq 0}(-1)^i\binom{n}{2i}A_{n-2i} = \begin{cases} 1, & \text{if } n = 4k \text{ or } n = 4k+1, \\ 0, & \text{if } n = 4k+2 \text{ or } n = 4k+3. \end{cases}$$

Multiplying both sides of the equality with $x^n/n!$ and summing over all natural numbers n we get

$$(A_0 + A_1 x + \frac{A_2}{2!}x^2 + \cdots)(1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots) = \sum_{k=0}^{\infty}\left(\frac{x^{4k}}{(4k)!} + \frac{x^{4k+1}}{(4k+1)!}\right).$$

The left hand side of this equality is equal to $A(x)\cos x$. Let $F(x)$ be the function in the right hand side of the equality. Then it is easy to see that $F(x)$ is the solution to the differential equation $F^{(4)}(x) = F(x)$ with the initial conditions $F(0) = F'(0) = 1$, $F^{(2)}(0) = F^{(3)}(0) = 0$. So $F(x) = \frac{1}{2}(\cos x + \sin x + e^x)$ and

$$A(x) = \frac{1}{2}\left(1 + \tan x + \frac{e^x}{\cos x}\right).$$

Now we solve the differential equation (5.2) and get

$$B(x) = \frac{1}{2} + \frac{1}{4}\tan x(1 + e^{2x} + 2e^x \sin x) + \frac{1}{2}e^x \cos x.$$

□

Remark 4. *The series expansion of $B(x)$ in Theorem 12 begins with*

$$B(x) = 1 + x + x^2 + x^3 + \frac{3}{4}x^4 + \frac{11}{20}x^5 + \frac{7}{20}x^6 + \frac{7}{30}x^7 + \frac{103}{720}x^8 + \cdots.$$

That is, the initial values for B_n are 1, 2, 6, 18, 66, 252, 1176, 5768.

5.7 The Distribution of Non-Overlapping GPs

A descent in a permutation $\pi = a_1 a_2 \dots a_n$ is an i such that $a_i > a_{i+1}$. The number of descents in a permutation π is denoted des π (and is equivalent to the generalized pattern 21). Any statistic with the same distribution as des is said to be *Eulerian*. The *Eulerian numbers* $A(n,k)$ count permutations in the symmetric group $\mathcal{S}_n$ with k descents and they are the coefficients of the *Eulerian polynomials* $A_n(t)$ defined by $A_n(t) = \sum_{\pi \in \mathcal{S}_n} t^{1+\text{des } \pi}$. The Eulerian polynomials satisfy the identity

$$\sum_{k \geq 0} k^n t^k = \frac{A_n(t)}{(1-t)^{n+1}}.$$

Two descents i and j *overlap* if $j = i+1$. We define a new statistic, namely the *maximum number of non-overlapping descents*, or MND, in a permutation. For instance, MND(321) = 1 whereas MND(41532) = 2. One can find the distribution of this new statistic by using Corollary 5. This distribution is given in Example 6. However, we prove a more general theorem:

Theorem 13. *Let p be a GP with no dashes. Let $A(x)$ be the e.g.f. for the number of permutations that avoid p. Let $D(x,y) = \sum_{\pi} y^{N(\pi)} \frac{x^{|\pi|}}{|\pi|!}$ where $N(\pi)$ is the maximum number of non-overlapping occurrences of p in π. Then*

$$D(x,y) = \frac{A(x)}{1 - y((x-1)A(x)+1)}.$$

Proof. We fix the natural number k and consider an auxiliary multi-pattern $P_k = p - p - \cdots - p$ with k copies of p. If a permutation avoids P_k then it has at most $k-1$ non-overlapping occurrences of p. From Theorem 11, the e.g.f. $B_k(x)$ for the number of permutations avoiding P_k is equal to $\sum_{i=1}^{k} A(x) \prod_{j=1}^{i-1}((x-1)A(x)+1)$. If we subtract $B_k(x)$ from the e.g.f. $B_{k+1}(x) = \sum_{i=1}^{k+1} A(x) \prod_{j=1}^{i-1}((x-1)A(x)+1)$ for the number of permutations avoiding P_{k+1}, which is obtained by applying Theorem 11 to the pattern P_{k+1}, then we get the e.g.f. $D_k(x)$ for the number of permutations that have exactly k non-overlapping occurrences of the pattern p. So

$$D_k(x) = \sum_n D_{n,k} \frac{x^n}{n!} = B_{k+1}(x) - B_k(x) = A(x)((x-1)A(x)+1)^k.$$

Now

$$D(x,y) = \sum_{n,k \geq 0} D_{n,k} y^k \frac{x^n}{n!} = \sum_k D_k(x) y^k = \frac{A(x)}{1 - y((x-1)A(x)+1)}.$$

□

All of the following examples are corollaries to Theorem 13.

Example 6. *If we consider descents then* $A(x) = e^x$*, hence the distribution of* MND *is given by the formula:*

$$D(x,y) = \frac{e^x}{1 - y(1 + (x-1)e^x)}.$$

Example 7. *Theorems 3 and 13 give the distribution of the maximum number of non-overlapping occurrences of the increasing subword of length* k *(the pattern* $123\ldots k$*), which is equal to*

$$D(x,y) = \frac{1}{(1-x)y + (1-y)F_k(x)},$$

were $F_k(x) = \sum_{i\geq 0} \frac{x^{ki}}{(ki)!} - \sum_{i\geq 0} \frac{x^{ki+1}}{(ki+1)!}$.

Example 8. *If we consider the maximum number of non-overlapping occurrences of the pattern 132 then the distribution of these numbers is given by the formula*

$$D(x,y) = \frac{1}{1 - yx + (y-1)\int_0^x e^{-t^2/2}\, dt}.$$

Example 9. *The distribution of the maximum number of non-overlapping occurrences of the pattern from Theorem 4 is given by the formula:*

$$D(x,y) = \frac{1}{1 - x + (1-y)F_{k,a}(x)},$$

where $F_{k,a}(x) = \sum_{i\geq 1} \frac{(-1)^{i+1}x^{ki+1}}{(ki+1)!} \prod_{j=2}^{i} \binom{jk-a}{k-a}$.

Bibliography

[BabStein] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, Séminaire Lotharingien de Combinatoire, B44b:18pp, 2000.

[Bon1] M. Bóna: Exact enumeration of 1342-avoiding permutations; A close link with labeled trees and planar maps, Journal of Combinatorial Theory, Series A, **80** (1997) 257-272.

[Bon2] M. Bóna: Permutations avoiding certain patterns; The case of length 4 and generalisations, Discrete Mathematics **175** (1997), 55-67.

[Bon3] M. Bóna: Permutations with one or two 132-subsequences, Discrete Mathematics **181** (1998), 267-274.

[Claes] A. Claesson: Generalised Pattern Avoidance, European Journal of Combinatorics **22** (2001), 961-971.

[ClaesMans] A. Claesson and T. Mansour: Permutations avoiding a pair of generalized patterns of length three with exactly one dash, preprint CO/0107044.

[ElizNoy] S. Elizalde and M. Noy: Enumeration of Subwords in Permutations, Proceedings of FPSAC 2001.

[GoulJack] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, A Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons Inc., New York, (1983).

[Kit] S. Kitaev: Multi-avoidance of generalised patterns, preprint. `http://www.math.chalmers.se/~kitaev/papers.html`

[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Mans] T. Mansour: Restricted 1-3-2 permutations and generalized patterns, preprint CO/0110039.

[SloPlo] N. J. A. Sloane and S. Plouffe: *The Encyclopedia of Integer Sequences*, Academic Press, (1995). http://www.research.att.com/∼njas/sequences/.

[Stan] R. Stanley: *Enumerative Combinatorics*, Volume **1**, Cambridge University Press, (1997).

[SchSim] R. Simion, F. Schmidt: Restricted permutations, European J. Combin. **6** (1985), no. 4, 383–406.

Paper VI

Partially ordered generalized patterns and k-ary words

Partially Ordered generalized patterns and k-ary words

Sergey Kitaev and Toufik Mansour [1]

Matematik, Chalmers tekniska högskola och Göteborgs universitet,
S-412 96 Göteborg, Sweden
kitaev@math.chalmers.se, toufik@math.chalmers.se

Abstract

Recently, Kitaev [Ki2] introduced partially ordered generalized patterns (POGPs) in the symmetric group, which further generalize the generalized permutation patterns introduced by Babson and Steingrímsson [BS]. A POGP p is a GP some of whose letters are incomparable. In this paper, we study the generating functions (g.f.) for the number of k-ary words avoiding some POGPs. We give analogues, extend and generalize several known results, as well as get some new results. In particular, we give the g.f. for the entire distribution of the maximum number of non-overlapping occurrences of a pattern p with no hyphens (that allowed to have repetition of letters), provided we know the g.f. for the number of k-ary words that avoid p.

6.1 Introduction

Let $[k]^n$ denote the set of all the words of length n over the (totally ordered) alphabet $[k] = \{1, 2, \ldots, k\}$. We call these words by n-long k-ary words. A *generalized pattern* τ is a word in $[\ell]^m$ (possibly with hyphens between some letters) that contains each letter from $[\ell]$ (possibly with repetitions). We say that the word $\sigma \in [k]^n$ *contains* a generalized pattern τ, if σ contains a subsequence isomorphic to τ in which the entries corresponding to consecutive entries of τ, which are not separated by a hyphen, must be adjacent. Otherwise, we say that σ *avoids* τ and write $\sigma \in [k]^n(\tau)$. Thus, $[k]^n(\tau)$ denotes the set of all the words in $[k]^n$ that avoid τ. Moreover, if P is a set of generalized patterns then $|[k]^n(P)|$ denotes the set all the words in $[k]^n$ that avoid each pattern from P simultaneously.

Example 1. *A word $\pi = a_1 a_2 \ldots a_n$ avoids the pattern* 13-2 *if π has no subsequence $a_i a_{i+1} a_j$ with $j > i+1$ and $a_i < a_j < a_{i+1}$. Also, π avoids the pattern* 121 *if it has no subword $a_i a_{i+1} a_{i+2}$ such that $a_i = a_{i+2} < a_{i+1}$.*

Classical patterns are generalized patterns with all possible hyphens (say, 2-1-3-4), in other words, those that place no adjacency requirements on σ. The first case of classical patterns studied was that of permutations avoiding a pattern of length 3 in $\mathcal{S}_3$. Knuth [Knuth] found that, for any $\tau \in \mathcal{S}_3$, $|\mathcal{S}_n(\tau)| = C_n$,

[1] Research financed by EC's IHRP Programme, within the Research Training Network "Algebraic Combinatorics in Europe", grant HPRN-CT-2001-00272

the nth Catalan number. Later, Simion and Schmidt [SS] determined the number $|\mathcal{S}_n(P)|$ of permutations in $\mathcal{S}_n$ simultaneously avoiding any given set of patterns $P \subseteq \mathcal{S}_3$. Burstein [Bu] extended this to $|[k]^n(P)|$ with $P \subseteq \mathcal{S}_3$. Burstein and Mansour [BM1] considered forbidden patterns with repeated letters. Also, Burstein and Mansour [BM2, BM3] considered forbidden generalized patterns with repeated letters.

Generalized permutation patterns were introduced by Babson and Steingrímsson [BS] with the purpose of the study of Mahonian statistics. Claesson [C] and Claesson and Mansour [CM] considered the number of permutations avoiding one or two generalized patterns with one hyphen. Kitaev [Ki1] examined the number of $|\mathcal{S}_n(P)|$ of permutations in $\mathcal{S}_n$ simultaneously avoiding any set of generalized patterns with no hyphens. Besides, Kitaev [Ki2] introduced a further generalization of the generalized permutation patterns namely *partially ordered generalized patterns*.

In this paper we introduce a further generalization of the generalized patterns namely *partially ordered generalized patterns in words* (*POGPs*), which is an analogue of POGPs in permutations [Ki2]. A POGP is a generalized pattern some of whose letters are incomparable. For example, if we write $\tau = 1\text{-}1'2'$, then we mean that in occurrence of τ in a word $\sigma \in [k]^n$ the letter corresponding to the 1 in τ can be either larger, smaller, or equal to the letters corresponding to $1'2'$. Thus, the word $113425 \in [5]^6$ contains seven occurrence of τ, namely 113, 134 twice, 125 twice, 325, and 425.

Following [Ki2], we consider two particular classes of POGPs – *shuffle patterns* and *multi-patterns*, which allows us to give an analogue for all the main results of [Ki2] for k-ary words. A multi-pattern is of the form $\tau = \tau^0\text{-}\tau^1\text{-}\cdots\text{-}\tau^s$ and a shuffle pattern of the form $\tau = \tau^0\text{-}a_1\text{-}\tau^1\text{-}a_2\text{-}\cdots\text{-}\tau^{s-1}\text{-}a_s\text{-}\tau^s$, where for any i and j, the letter a_i is greater than any letter of τ^j and for any $i \neq j$ each letter of τ^i is incomparable with any letter of τ^j. These patterns are investigated in Sections 6.3 and 6.4.

Let $\tau = \tau^0\text{-}\tau^1\text{-}\cdots\text{-}\tau^s$ be an arbitrary multi-pattern and let $A_{\tau^i}(x;k)$ be the generating function (g.f.) for the number of words in k-letter alphabet that avoid τ^i for each i. In Theorem 6 we find the g.f., in terms of the $A_{\tau^i}(x;k)$, for the number of k-ary words that avoid τ. In particular, this allows us to find the g.f. for the entire *distribution* of the maximum number of non-overlapping occurrences of a pattern τ with no hyphens, if we only know the g.f. for the number of k-ary words that avoid τ. Thus, in order to apply our results in what follows we need to know how many k-ary words avoid a given ordinary generalized pattern with no hyphens. This question was examined, for instance, in [BM1, Sections 2 and 3], [BM2, Section 3] and [BM3, Section 3.3].

6.2 Definitions and Preliminaries

A *partially ordered generalized pattern* (*POGP*) is a generalized pattern where some of the letters can be incomparable.

Example 2. *The simplest non-trivial example of a POGP that differs from*

the ordinary generalized patterns is $\tau = 1'$-2-$1''$*, where the second letters is the greatest one and the first and the last letters are incomparable to each other. The word* $\sigma = 31421$ *has five occurrences of* τ*, namely* 342, 341, 142, 141, *and* 121.

Let $A_\tau(x;k) = \sum_{n\geq 0} a_\tau(n;k)x^n$ denote the *generating function* (*g.f.*) for the numbers $a_\tau(n;k)$ of words in $[k]^n$ avoiding the pattern τ. For $\tau = 1'$-2-$1''$, we have

$$A_{1'\text{-}2\text{-}1''}(x;k) = \frac{1}{(1-x)^{2k-1}} - \sum_{j=1}^{k-1} \frac{x}{(1-x)^{2j}}. \tag{6.1}$$

Indeed, if $\sigma \in [k]^n$ avoids τ, and σ contains $s > 0$ copies of the letter k, then the letters k appear as leftmost or rightmost letters of σ. If σ contains no k then $\sigma \in [k-1]^n$. So, for all $n \geq 0$, we have

$$a_\tau(n;k) = a_\tau(n;k-1)+2a_\tau(n-1;k-1)+3a_\tau(n-2;k-1)+\cdots+(n+1)a_\tau(0;k-1),$$

since there are $(i+1)a_\tau(n-i;k-1)$ possibilities to place i letters k into σ, for $0 \leq i \leq n$. Hence, for all $n \geq 2$,

$$a_\tau(n;k) - 2a_\tau(n-1;k) + a_\tau(n-2;k) = a_\tau(n;k-1),$$

together with $a_\tau(0,k) = 1$ and $a_\tau(1,k) = k$. Multiplying both sides of the recurrence above with x^n and summing over all $n \geq 2$, we get Equation 6.1.

Definition 5. *If the number of words in* $[k]^n$*, for each* n*, that avoid a POGP* τ *is equal to the number of words that avoid a POGP* ϕ*, then* τ *and* ϕ *are said to be* equivalent *and we write* $\tau \equiv \phi$.

The *reverse* $R(\sigma)$ of a word $\sigma = \sigma_1\sigma_2\ldots\sigma_n$ is the word $\sigma_n\ldots\sigma_2\sigma_1$. The *complement* $C(\sigma)$ is the word $\theta = \theta_1\theta_2\ldots\theta_n$ where $\theta_i = k+1-\sigma_i$ for all $i = 1,2,\ldots,n$. For example, if $\sigma = 123331 \in [3]^6$, then $R(\sigma) = 133321$, $C(\sigma) = 321113$, and $R(C(\sigma)) = 311123$. We call these bijections of $[k]^n$ to itself *trivial*. For example, the number of words that avoid the pattern 12-2 is the same as the number of words that avoid the patterns 2-21, 1-12, and 21-1, respectively.

Following [Ki2], it is convenient to introduce the following definition.

Definition 6. *Let* τ *be a generalized pattern without hyphens. A word* σ quasi-avoids τ *if* σ *has exactly one occurrence of* τ *and this occurrence consists of the* $|\tau|$ *rightmost letters of* σ*, where* $|\tau|$ *denotes the number of letters in* τ.

For example, the word 5112234 quasi-avoids the pattern 1123, whereas the words 5223411 and 1123345 do not.

Proposition 1. *Let* τ *be a non-empty generalized pattern with no hyphens. Let* $A^*_\tau(x;k)$ *denote the g.f. for the number of words in* $[k]^n$ *that quasi-avoid* τ*. Then*

$$A^*_\tau(x;k) = (kx-1)A_\tau(x;k) + 1. \tag{6.2}$$

Proof. Using the similar arguments as those in the proof of [Ki2, Proposition 4], we get that, for $n \geq 1$,

$$a^*_\tau(n;k) = k a_\tau(n-1;k) - a_\tau(n;k),$$

where $a^*_\tau(n;k)$ denotes the number of words in $[k]^n$ that quasi-avoid τ. Multiplying both sides of the last equality by x^n and summing over all natural numbers n, we get the desired result. □

Definition 7. *Suppose* $\{\tau^0, \tau^1, \ldots, \tau^s\}$ *is a set of generalized patterns with no hyphens and*

$$\tau = \tau^0\text{-}\tau^1\text{-}\cdots\text{-}\tau^s,$$

where each letter of τ^i *is incomparable with any letter of* τ^j *whenever* $i \neq j$*. We call such POGPs* multi-patterns.

Definition 8. *Suppose* $\{\tau^0, \tau^1, \ldots, \tau^s\}$ *is a set of generalized patterns with no hyphens and* $a_1 a_2 \ldots a_s$ *is a word of* s *letters. We define a* shuffle *pattern to be a pattern of the form*

$$\tau = \tau^0\text{-}a_1\text{-}\tau^1\text{-}a_2\text{-}\cdots\text{-}\tau^{s-1}\text{-}a_s\text{-}\tau^s,$$

where each letter of τ^i *is incomparable with any letter of* τ^j *whenever* $i \neq j$*, and the letter* a_i *is greater than any letter of* τ^j *for any* i *and* j*.*

For example, $1'$-2-$1''$ is a shuffle pattern, and $1'$-$1''$ is a multi-patterns. From definitions, we obtain that we can get a multi-pattern from a shuffle pattern by removing all the letters a_i.

There is a connection between multi-avoidance of the generalized patterns and the POGPs. In particular, to avoid $1'$-2-$1''$ is the same as to avoid simultaneously the patterns 1-2-1, 1-3-2, and 2-3-1. A straightforward argument leads to the following proposition.

Proposition 2. *For any POGP* τ *there exists a set* T *of generalized patterns such that a word* σ *avoids* τ *if and only if* σ *avoids all the patterns in* T*.*

For example, if $\tau = 1'2'$-3-$1''$, then to avoid τ is the same to avoid 5 patterns, 12-3-1, 12-3-2, 12-4-3, 13-4-2, and 23-4-1. Moreover, the following proposition holds:

Proposition 3. *Suppose* $\tau = \tau_1\text{-}a\text{-}\tau_2$ *(resp.* $\phi = \phi_1\text{-}\phi_2$*) is a shuffle pattern (resp. a multi-pattern) such that* τ_1, $\phi_1 \in [r_1]^{\ell_1}$, τ_2, $\phi_2 \in [r_2]^{\ell_2}$ *and each letter of* $[r_1]$ *is incomparable with any letter of* $[r_2]$*. Also, without lose the generality, suppose* $r_1 \geq r_2$*. Then to avoid* τ *(resp.* ϕ*) is the same as to avoid* $\sum_{i=0}^{r_2} \binom{r_1}{i}\binom{r_2}{i}\binom{r_1+r_2-i}{r_1}$ *generalazed patterns. In particular, the number of generalized patterns does not depend on the lengths* ℓ_1 *and* ℓ_2*.*

Proof. Obviously, to prove the statement, we need to find the number of ways to make a total order on $[r_1] \cup [r_2]$ (the letter a does not play any roll, since it is always the greatest letter). Any total order on $[r_1] \cup [r_2]$ is an alphabet that can consist of $r_1 + r_2 - i$ letters, where i is the number of letters in $[r_2]$ that supposed to coincide with some letters in $[r_1]$. Clearly, $0 \leq i \leq r_2$ and we can choose coinciding letters in $\binom{r_1}{i}\binom{r_2}{i}$ ways. Now, after choosing the coinciding letters, we can make a total order in $\binom{r_1+r_2-i}{r_1}$ ways, which is given by [Ki2, Theorem 8]. □

6.3 The shuffle patterns

We recall that according to Definition 8, a shuffle pattern is a pattern of the form

$$\tau = \tau^0\text{-}a_1\text{-}\tau^1\text{-}a_2\text{-}\cdots\text{-}\tau^{s-1}\text{-}a_s\text{-}\tau^s,$$

where $\{\tau^0, \tau^1, \ldots, \tau^s\}$ is a set of generalized patterns with no hyphens, $a_1 a_2 \ldots a_s$ is a word of s letters, for any i and j the letter a_i is greater than any letter of τ^j and for any $i \neq j$ each letter of τ^i is incomparable with any letter of τ^j.

Let us consider the shuffle pattern $\phi = \tau\text{-}\ell\text{-}\tau$, where ℓ is the greatest letter in ϕ and letters each letter in the left τ is incomparable with any letter in the right τ.

Theorem 1. *Let ϕ be the shuffle pattern τ-ℓ-τ described above. Then for all $k \geq \ell$,*

$$A_\phi(x;k) = \frac{1}{(1 - xA_\tau(x;k-1))^2}\Big(A_\phi(x;k-1) - xA_\tau^2(x;k-1)\Big).$$

Proof. We show how to get a recurrence relation on k for $A_\phi(x;k)$, which is the g.f. for the number of words in $[k]^n(\phi)$. Suppose $\sigma \in [k]^n(\phi)$ is such that it contains exactly d copies of the letter k. If $d = 0$ then the g.f. for the number of such words is $A_\phi(x;k-1)$. Assume that $d \geq 1$. Clearly, σ can be written in the following form:

$$\sigma = \sigma^0 k \sigma^1 k \cdots k \sigma^d,$$

where σ^j is a ϕ-avoiding word on $k-1$ letters, for $j = 0, 1, \ldots, d$. There are two possibilities: either σ^j avoids τ for all j, or there exists j_0 such that σ^{j_0} contains τ and for any $j \neq j_0$, the word σ^j avoids τ. In the first case, the number of such words is given by the g.f. $x^d A_\tau^{d+1}(x;k-1)$, whereas in the second case, by $(d+1)x^d A_\tau^d(x;k-1)(A_\phi(x;k-1) - A_\tau(x;k-1))$. In the last expression, the multiple $(d+1)$ is the number of ways to choose j, such that σ^j has an occurrence of τ, and $A_\phi(x;k-1) - A_\tau(x;k-1)$ is the g.f. for the number of words avoiding ϕ and containing τ.

Therefore,

$$A_\phi(x;k) = A_\phi(x;k-1) +$$
$$\sum_{d\geq 1}(d+1)x^d A_\tau^d(x;k-1)A_\phi(x;k-1) - \sum_{d\geq 1} d x^d A_\tau^{d+1}(x;k-1),$$

equivalently,

$$A_\phi(x;k) = A_\phi(x;k-1)+$$

$$A_\phi(x;k-1)\frac{2xA_\tau(x;k-1) - x^2A_\tau^2(x;k-1)}{(1-xA_\tau(x;k-1))^2} - \frac{xA_\tau^2(x;k-1)}{(1-xA_\tau(x;k-1))^2}.$$

The rest is easy to check. □

Example 3. *Let $\phi = 1'$-2-$1''$. Here $\tau = 1$, so $A_\tau(x;k) = 1$ for all $k \geq 1$, since only the empty word avoids τ. Hence, according to Theorem 1, we have*

$$A_\phi(x;k) = \frac{A_\phi(x;k-1) - x}{(1-x)^2},$$

which together with $A_\phi(x;1) = \frac{1}{1-x}$ (for any n only the word $\underbrace{11\ldots1}_{n \text{ times}}$ avoids ϕ) gives Equation 6.1.

More generally, we consider a shuffle pattern of the form τ^0-ℓ-τ^1, where ℓ is the greatest element of the pattern.

Theorem 2. *Let ϕ be the shuffle pattern τ-ℓ-ν. Then for all $k \geq \ell$, $A_\phi(x;k) =$*

$$\frac{1}{(1-xA_\tau(x;k-1))(1-xA_\nu(x;k-1))}\Big(A_\phi(x;k-1) - xA_\tau(x;k-1)A_\nu(x;k-1)\Big).$$

Proof. We proceed as in the proof of Theorem 1. Suppose $\sigma \in [k]^n(\phi)$ is such that it contains exactly d copies of the letter k. If $d = 0$ then the g.f. for the number of such words is $A_\phi(x;k-1)$. Assume that $d \geq 1$. Clearly, σ can be written in the following form:

$$\sigma = \sigma^0 k \sigma^1 k \cdots k \sigma^d,$$

where σ^j is a ϕ-avoiding word on $k-1$ letters, for $j = 0, 1, \ldots, d$. There are two possibilities: either σ^j avoids τ for all j, or there exists j_0 such that σ^{j_0} contains τ, σ^j avoids τ for all $j = 0, 1, \ldots, j_0 - 1$ and σ^j avoids ν for any $j = j_0 + 1, \ldots, d$. In the first case, the number of such words is given by the g.f. $x^d A_\tau^{d+1}(x;k-1)$. In the second case, we have

$$x^d \sum_{j=0}^{d} A_\tau^j(x;k-1)A_\nu^{d-j}(x;k-1)(A_\phi(x;k-1) - A_\tau(x;k-1)).$$

Therefore, we get

$$A_\phi(x;k) = A_\phi(x;k-1) + A_\phi(x;k-1)\sum_{d\geq 1} x^d \sum_{j=0}^{d} A_\tau^j(x;k-1)A_\nu^{d-j}(x;k-1)$$
$$- \sum_{d\geq 1} x^d \sum_{j=1}^{d} A_\tau^j(x;k-1)A_\nu^{d+1-j}(x;k-1),$$

equivalently,

$$A_\phi(x;k) = (A_\phi(x;k-1) - xA_\tau(x;k-1)A_\nu(x;k-1)) \sum_{d\geq 0} x^d \sum_{j=0}^{d} A_\tau^j(x;k-1)A_\nu^{d-j}(x;k-1).$$

Hence, using the identity $\displaystyle\sum_{n\geq 0} x^n \sum_{j=0}^{n} p^j q^{n-j} = \frac{1}{(1-xp)(1-xq)}$ we get the desired result. □

We now give two corollaries to Theorem 2.

Corollary 6. *Let $\phi = \tau^0$-ℓ-τ^1 be a shuffle pattern, and let $f(\phi) = f_1(\tau^0)$-ℓ-$f_2(\tau^1)$, where f_1 and f_2 are any trivial bijections. Then $\phi \equiv f(\phi)$.*

Proof. Using Theorem 2, and the fact that the number of words in $[k]^n$ avoiding τ (resp. ν) and $f_1(\tau)$ (resp. $f_2(\nu)$) have the same generating functions, we get the desired result. □

Corollary 7. *For any shuffle pattern τ-ℓ-ν, we have*

$$\tau\text{-}\ell\text{-}\nu \equiv \nu\text{-}\ell\text{-}\tau.$$

Proof. Corollary 6 yields that the shuffle pattern τ-ℓ-ν is equivalent to the pattern τ-ℓ-$R(\nu)$, which is equivalent to the pattern $R(\tau\text{-}\ell\text{-}R(\nu)) = \nu$-$\ell$-$R(\tau)$. Finally, we use Corollary 6 one more time to get the desired result. □

6.4 The multi-patterns

We recall that according to Definition 7, a multi-pattern is a pattern of the form $\tau = \tau^0$-τ^1-$\cdots$-τ^s, where $\{\tau^0, \tau^1, \ldots, \tau^s\}$ is a set of generalized patterns with no hyphens and each letter of τ^i is incomparable with any letter of τ^j whenever $i \neq j$.

The simplest non-trivial example of a multi-pattern is the multi-pattern $\phi = 1\text{-}1'2'$. To avoid ϕ is the same as to avoid the patterns 1-12, 1-23, 2-12, 2-13, and 3-12 simultaneously. To count the number of words in $[k]^n(1\text{-}1'2')$, we choose the leftmost letter of $\sigma \in [k]^n(1\text{-}1'2')$ in k ways, and observe that all the other letters of σ must be in a non-increasing order. Using [BM1], for all $n \geq 1$, we have

$$|[k]^n(1\text{-}1'2')| = k \cdot \binom{n+k-2}{n-1}.$$

The following theorem is an analogue to [Ki2, Theorem 21].

Theorem 3. *Let $\tau = \tau^0$-τ^1 and $\phi = f_1(\tau^0)$-$f_2(\tau^1)$, where f_1 and f_2 are any of the trivial bijections. Then $\tau \equiv \phi$.*

Proof. First, let us prove that the pattern $\tau = \tau^0\text{-}\tau^1$ is equivalent to the pattern $\phi = \tau^0\text{-}f(\tau^1)$, where f is a trivial bijection. Suppose that $\sigma = \sigma^1\sigma^2\sigma^3 \in [k]^n$ avoids τ and $\sigma^1\sigma^2$ has exactly one occurrence of τ^0, namely σ^2. Then σ^3 must avoid τ^1, so $f(\sigma^3)$ avoids $f(\tau^3)$ and $\sigma_f = \sigma^1\sigma^2 f(\sigma^3)$ avoids ϕ. The converse is also true, if σ_f avoids ϕ then σ avoids τ. Since any word either avoids τ^0 or can be factored as above, we have a bijection between the class of words avoiding τ and the class of words avoiding ϕ. Thus $\tau \equiv \phi$.

Now, we use the considerations above as well as the properties of trivial bijections to get

$$\tau \equiv \tau^0\text{-}f_2(\tau^1) \equiv R(\tau^0\text{-}f_2(\tau^1)) \equiv R(f_2(\tau^1))\text{-}R(\tau^0) \equiv \\ R(f_2(\tau^1))\text{-}f_1(R(\tau^0)) \equiv R(f_2(\tau^1))\text{-}R(f_1(\tau^0)) \equiv f_1(\tau^0)\text{-}f_2(\tau^1).$$

□

Using Theorem 3, we get the following corollary, which is an analogue to [Ki2, Corollary 22].

Corollary 8. *The multi-pattern $\tau^0\text{-}\tau^1$ is equivalent to the multi-pattern $\tau^1\text{-}\tau^0$.*

Proof. From Theorem 3, using the properties of the trivial bijection R, we get

$$\tau^0\text{-}\tau^1 \equiv \tau^0\text{-}R(\tau^1) \equiv R(R(\tau^1))\text{-}R(\tau^0) \equiv \tau^1\text{-}R(R(\tau^0)) \equiv \tau^1\text{-}\tau^0.$$

□

Using induction on s, Corollary 8, and proceeding in the way proposed in [Ki2, Theorem 23], we get

Theorem 4. *Suppose we have multi-patterns $\tau = \tau^0\text{-}\tau^1\text{-}\cdots\text{-}\tau^s$ and $\phi = \phi^0\text{-}\phi^1\text{-}\cdots\text{-}\phi^s$, where $\tau^1\tau^2\cdots\tau^s$ is a permutation of $\phi^1\phi^2\cdots\phi^s$. Then $\tau \equiv \phi$.*

The last theorem is an analogue to [Ki2, Theorem 23]. As a corollary to Theorem 4, using Theorem 3 and the idea of the proof of [Ki2, Corollary 24], we get the following corollary which is an analogue to [Ki2, Corollary 24].

Corollary 9. *Suppose we have multi-patterns $\tau = \tau^0\text{-}\tau^1\text{-}\cdots\text{-}\tau^s$ and $\phi = f_0(\tau^0)\text{-}f_1(\tau^1)\text{-}\cdots\text{-}f_s(\tau^s)$, where f_i is an arbitrary trivial bijection. Then $\tau \equiv \phi$.*

The following theorem is a good auxiliary tool for calculating the g.f. for the number of words that avoid a given POGP. For particular POGPs, it allows to reduce the problem to calculating the g.f. for the number of words that avoid another POGP which is shorter. We recall that $A^*_\tau(x;k)$ is the generating function for the number of words in $[k]^n$ that quasi-avoid the pattern τ.

Theorem 5. *Suppose $\tau = \tau^0\text{-}\phi$, where ϕ is an arbitrary POGP, and the letters of τ^0 are incomparable to the letters of ϕ. Then for all $k \geq 1$, we have*

$$A_\tau(x;k) = A_{\tau^0}(x;k) + A_\phi(x;k)A^*_{\tau^0}(x;k).$$

Proof. Suppose $\sigma = \sigma^1\sigma^2\sigma^3 \in [k]^n$ avoids the pattern τ, where $\sigma^1\sigma^2$ quasi-avoids the pattern τ^0, and σ^2 is the occurrence of τ^0. Clearly, σ^3 must avoid ϕ. To find $A_\tau(x;k)$, we observe that there are two possibilities: either σ avoids τ^0, or σ does not avoid τ^0. In these cases, the g.f. for the number of such words is equal to $A_{\tau^0}(x;k)$ and $A_\phi(x;k)A^*_{\tau^0}(x;k)$ respectively (the second term came from the factorization above). Thus, the statement is true. □

Corollary 10. *Let $\tau = \tau^1\text{-}\tau^2\text{-}\cdots\text{-}\tau^s$ be a multi-pattern such that τ^j is equal to either 12 or 21, for $j = 1, 2, \ldots, s$. Then*

$$A_\tau(x;k) = \frac{1 - \left(1 + \frac{kx-1}{(1-x)^k}\right)^s}{1-kx}.$$

Proof. According to [BM2], $A_{12}(x;k) = A_{21}(x;k) = \frac{1}{(1-x)^k}$. Using Theorem 5, Proposition 1 and induction on s, we get the desired result. □

More generally, using Theorem 5 and Proposition 1, we get the following theorem that is the basis for calculating the number of words that avoid a multi-pattern, and therefore is the main result for multi-patterns in this paper.

Theorem 6. *Let $\tau = \tau^1\text{-}\tau^2\text{-}\cdots\text{-}\tau^s$ be a multi-pattern. Then*

$$A_\tau(x;k) = \sum_{j=1}^{s} A_{\tau^j}(x;k) \prod_{i=1}^{j-1}((kx-1)A_{\tau^i}(x;k)+1).$$

6.5 The distribution of non-overlapping generalized patterns

A descent in a word $\sigma \in [k]^n$ is an i such that $\sigma_i > \sigma_{i+1}$. Two descents i and j *overlap* if $j = i+1$. We define a new statistics, namely the *maximum number of non-overlapping descents,* or MND, in a word. For example, MND(33211) = 1 whereas MND(13211143211) = 3. One can find the distribution of this new statistic by using Corollary 10. This distribution is given in Example 4. However, we prove a more general theorem:

Theorem 7. *Let τ be a generalized pattern with no hyphens. Then for all $k \geq 1$,*

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_\tau(\sigma)}x^n = \frac{A_\tau(x;k)}{1-y((kx-1)A_\tau(x;k)+1)},$$

where $N_\tau(\sigma)$ is the maximum number of non-overlapping occurrences of τ in σ.

Proof. We fix the natural number s and consider the multi-pattern $\Phi_s = \tau\text{-}\tau\text{-}\cdots\text{-}\tau$ with s copies of τ. If a word avoids Φ_s then it has at most $s-1$ non-overlapping occurrences of τ. Theorem 6 yields

$$A_{\Phi_s}(x;k) = \sum_{j=1}^{s} A_\tau(x;k) \prod_{i=1}^{j-1}((kx-1)A_\tau(x;k)+1).$$

So, the g.f. for the number of words that has exactly s non-overlapping occurrences of the pattern τ is given by

$$A_{\Phi_{s+1}}(x;k) - A_{\Phi_s}(x;k) = A_\tau(x;k)((kx-1)A_\tau(x;k)+1)^s.$$

Hence,

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_\tau(\sigma)}x^n = \sum_{s\geq 0} A_\tau(x;k)((kx-1)A_\tau(x;k)+1)^s = \frac{A_\tau(x;k)}{1-y((kx-1)A_\tau(x;k)+1)}.$$

□

All of the following examples are corollaries to Theorem 7.

Example 4. *If we consider descents (the pattern* 12*) then* $A_{12}(x;k) = \frac{1}{(1-x)^k}$ *(see [BM2]), hence the distribution of* MND *is given by the formula:*

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_{12}(\sigma)}x^n = \frac{1}{(1-x)^k + y(1-kx-(1-x)^k)}.$$

Example 5. *The distribution of the maximum number of non-overlapping occurrences of the pattern* 122 *is given by the formula:*

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_{122}(\sigma)}x^n = \frac{x}{(1-x^2)^k + x - 1 + y(1-kx^2-(1-x^2)^k)},$$

since according to [BM3, Theorem 3.10], $A_{122}(x;k) = \frac{x}{(1-x^2)^k-(1-x)}$.

Example 6. *If we consider the pattern* 212 *then* $A_{212}(x;k) = \left(1 - x\sum_{j=0}^{k-1}\frac{1}{1+jx^2}\right)^{-1}$ *(see [BM3, Theorem 3.12]), hence the distribution of the maximum number of non-overlapping occurrences of the pattern* 212 *is given by the formula:*

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_{212}(\sigma)}x^n = \frac{1}{1 - x\sum_{j=0}^{k-1}\frac{1}{1+jx^2} + xy\left(\sum_{j=0}^{k-1}\frac{1}{1+jx^2} - k\right)}.$$

Example 7. *Using [BM3, Theorem 3.13], the distribution of the maximum number of non-overlapping occurrences of the pattern* 123 *is given by the formula:*

$$\sum_{n\geq 0}\sum_{\sigma\in[k]^n} y^{N_{123}(\sigma)}x^n = \frac{1}{\sum_{j=0}^{k} a_j\binom{k}{j}x^j + y\left(1-kx-\sum_{j=0}^{k} a_j\binom{k}{j}x^j\right)},$$

where $a_{3m} = 1$, $a_{3m+1} = -1$, *and* $a_{3m+2} = 0$, *for all* $m \geq 0$.

Bibliography

[BS] E. Babson, E. Steingrímsson: Generalized permutation patterns and a classification of the Mahonian statistics, Séminaire Lotharingien de Combinatoire, B44b:18pp, (2000).

[Bu] A. Burstein, Enumeration of words with forbidden patterns, Ph.D. thesis, University of Pennsylvania, 1998.

[BM1] A. Burstein and T. Mansour, Words restricted by patterns with at most 2 distinct letters, *Electronic J. of Combinatorics*, to appear (2002).

[BM2] A. Burstein and T. Mansour, Words restricted by 3-letter generalized multipermutation patterns, preprint CO/0112281.

[BM3] A. Burstein and T. Mansour, Counting occurrences of some subword patterns, preprint CO/0204320.

[C] A. Claesson: Generalised Pattern Avoidance, European Journal of Combinatorics **22** (2001), 961-971.

[CM] A. Claesson and T. Mansour, Enumerating Permutations Avoiding a Pair of Babson-Steingrímsson Patterns, preprint CO/0107044.

[Ki1] S. Kitaev, Multi-avoidance of generalised patterns, Discr. Math., to appear (2002).

[Ki2] S. Kitaev, Partially ordered generalized patterns, Discr. Math., to appear (2002).

[Knuth] D. E. Knuth: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[SS] R. Simion, F. Schmidt: Restricted permutations, European J. Combin. **6**, no. 4 (1985), 383–406.

Paper VII

Counting the occurrences of generalized patterns in words generated by a morphism

Counting the occurrences of generalized patterns in words generated by a morphism

Sergey Kitaev and Toufik Mansour [1]

Matematik, Chalmers tekniska högskola och Göteborgs universitet,
S-412 96 Göteborg, Sweden
kitaev@math.chalmers.se, toufik@math.chalmers.se

Abstract

We count the number of occurrences of certain patterns in given words. We choose these words to be the set of all finite approximations of a sequence generated by a morphism with certain restrictions. The patterns in our considerations are either classical patterns 1-2, 2-1, 1-1-$\cdots$-1, or arbitrary generalized patterns without internal dashes, in which repetitions of letters are allowed. In particular, we find the number of occurrences of the patterns 1-2, 2-1, 12, 21, 123 and 1-1-$\cdots$-1 in the words obtained by iterations of the morphism $1 \to 123$, $2 \to 13$, $3 \to 2$, which is a classical example of a morphism generating a nonrepetitive sequence.

7.1 Introduction and Background

We write permutations as words $\pi = a_1 a_2 \cdots a_n$, whose letters are distinct and usually consist of the integers $1, 2, \ldots, n$.

An occurrence of a pattern p in a permutation π is "classically" defined as a subsequence in π (of the same length as the length of p) whose letters are in the same relative order as those in p. Formally speaking, for $r \leq n$, we say that a permutation σ in the symmetric group $\mathcal{S}_n$ has an occurrence of the pattern $p \in \mathcal{S}_r$ if there exist $1 \leq i_1 < i_2 < \cdots < i_r \leq n$ such that $p = \sigma(i_1)\sigma(i_2)\ldots\sigma(i_r)$ in reduced form. The *reduced form* of a permutation σ on a set $\{j_1, j_2, \ldots, j_r\}$, where $j_1 < j_2 < \cdots < j_r$, is a permutation σ_1 obtained by renaming the letters of the permutation σ so that j_i is renamed i for all $i \in \{1, \ldots, r\}$. For example, the reduced form of the permutation 3651 is 2431. The first case of classical patterns studied was that of permutations avoiding a pattern of length 3 in $\mathcal{S}_3$. Knuth [Knuth] found that, for any $\tau \in \mathcal{S}_3$, the number $|\mathcal{S}_n(\tau)|$ of n-permutations avoiding τ is C_n, the nth Catalan number. Later, Simion and Schmidt [SimSch] determined the number $|\mathcal{S}_n(P)|$ of permutations in $\mathcal{S}_n$ simultaneously avoiding any given set of patterns $P \subseteq \mathcal{S}_3$.

In [BabStein] Babson and Steingrímsson introduced *generalised permutation patterns* that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. In order to avoid confusion we write a "classical" pattern, say 231, as 2-3-1, and if we write, say 2-31, then we mean that if

[1] Research financed by EC's IHRP Programme, within the Research Training Network "Algebraic Combinatorics in Europe", grant HPRN-CT-2001-00272

this pattern occurs in the permutation, then the letters in the permutation that correspond to 3 and 1 are adjacent. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, whereas the pattern 2-3-1 occurs, in addition, in the subwords 562 and 563. A motivation for introducing these patterns in [BabStein] was the study of Mahonian statistics. A number of interesting results on generalised patterns were obtained in [Claes]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were shown there.

Burstein [Burstein] considered words instead of permutations. In particular, he found the number $|[k]^n(P)|$ of words of length n in k-letter alphabet that avoid each pattern from a set $P \subseteq \mathcal{S}_3$ simultaneously. Burstein and Mansour [BurMans1] (resp. [BurMans2, BurMans3]) considered forbidden patterns (resp. generalized patterns) with repeated letters.

The most attention, in the papers on classical or generalized patterns, is paid to counting exact formulas and/or generating functions for the number of words or permutations avoiding, or having k occurrences of, certain pattern. In this paper we suggest another problem, namely counting the number of occurrences of a particular pattern τ in given words. We choose these words to be a set of all finite approximations (to be defined below) of a sequence generated by a morphism with certain restrictions. A motivation for such a choice is big interest in studying classes of sequences and words that are defined by iterative schemes [Lothaire, Salomaa]. The pattern τ in our considerations is either a classical pattern from the set $\{1\text{-}2, 2\text{-}1, 1\text{-}1\text{-}\cdots\text{-}1\}$, or an arbitrary generalized pattern without internal dashes, in which repetitions of letters are allowed. In particular, we find that there are $(3 \cdot 4^{n-1} + 2^n)$ occurrences of the pattern 1-2 in the n-th finite approximation of the sequence w defined below, which is a classical example of a nonrepetitive sequence.

Let Σ be an alphabet and $\Sigma^\star$ be the set of all words of Σ. A map $\varphi : \Sigma^\star \to \Sigma^\star$ is called a *morphism*, if we have $\varphi(uv) = \varphi(u)\varphi(v)$ for any $u, v \in \Sigma^\star$. It is easy to see that a morphism φ can be defined by defining $\varphi(i)$ for each $i \in \Sigma$. The set of all rules $i \to \varphi(i)$ is called a *substitution system*. We create words by starting with a letter from the alphabet Σ and iterating the substitution system. Such a substitution system is called a *D0L (Deterministic, with no context Lindenmayer) system* [LindRoz]. D0L systems are classical objects of formal language theory. They are interesting from mathematical point of view [Frid], but also have applications in theoretical biology [Lind]. Let $|X|$ denote the length of a word X, that is the number of letters in X.

Suppose a word $\varphi(a)$ begins with a for some $a \in \Sigma$, and that the length of $\varphi^k(a)$ increases without bound. The symbolic sequence $\lim\limits_{k\to\infty} \varphi^k(a)$ is said to be *generated* by the morphism φ. In particular, $\lim\limits_{k\to\infty} \varphi^k(a)$ is a *fixed point* of φ. However, in this paper we are only interesting in the *finite approximations* of $\lim\limits_{k\to\infty} \varphi^k(a)$, that is in the words $\varphi^k(a)$ for $k = 1, 2, \ldots$.

An example of a sequence generated by a morphism can be the following sequence w. We create words by starting with the letter 1 and iterating the substitution system ϕ_w: $1 \to 123$, $2 \to 13$, $3 \to 2$. Thus, the initial letters of

w are 123132123213.... This sequence was constructed in connection with the problem of constructing a nonrepetitive sequence on a 3-letter alphabet, that is, a sequence that does not contain any subwords of the type $XX = X^2$, where X is any non-empty word over a 3-letter alphabet. The sequence w has that property. The question of the existence of such a sequence, as well as the questions of the existence of sequences avoiding other kinds of repetitions, were studied in algebra [Adian, Justin, Kol], discrete analysis [Carpi, Dekk, Evdok, Ker, Pleas] and in dynamical systems [MorseHedl]. In Examples 1, 4 and 5 we give the number of occurrences of the patterns 1-2, 2-1, 1-1-$\cdots$-1, 12, 123 and 21 in the finite approximations of w.

To proceed further, we need the following definitions. Let $N_\phi^\tau(n)$ denote the number of occurrences of the pattern τ in a word generated by some morphism ϕ after n iterations. We say that an occurrence of τ is *external* for a pair of words (X, Y), if this occurrence starts in X and ends in Y. Also, an occurrence of τ for a word X is *internal*, if this occurrence starts and ends in this X.

7.2 Patterns 1-2, 2-1 and 1-1-$\cdots$-1

Theorem 1. *Let $\mathcal{A} = \{1, 2, \ldots, k\}$ be an alphabet, where $k \geq 2$ and a pattern $\tau \in \{1\text{-}2, 2\text{-}1\}$. Let X_1 begins with the letter 1 and consists of ℓ copies of each letter $i \in \mathcal{A}$ ($\ell \geq 1$). Let a morphism ϕ be such that*

$$1 \to X_1,\ 2 \to X_2,\ 3 \to X_3, \ldots, k \to X_k,$$

where we allow X_i to be the empty word ϵ for $i = 2, 3, \ldots, k$ (that is, ϕ may be an erasing morphism), $\sum_{i=2}^{k} |X_i| = k \cdot d$, and each letter from $\mathcal{A}$ appears in the word $X_2 X_3 \ldots X_k$ exactly d times. Besides, let $e_{i,j}$ (resp. e_i) be the number of external occurrences of τ for (X_i, X_j) (resp. (X_i, X_i)), where $i \neq j$. We assume that $e_{i,j} = e_{j,i}$ for all i and j. Let s_i be the number of internal occurrences of τ in X_i. In particular, $s_i = e_i = e_{i,j} = e_{j,i} = 0$, whenever $X_i = \epsilon$; also, $e_i = |X_i| \cdot (|X_i| - 1)/2$, whenever there are no repetitive letters in X_i. Then $N_\phi^\tau(1) = s_1$ and for $n \geq 2$, $N_\phi^\tau(n)$ is given by

$$\ell \cdot (d+\ell)^{n-2} \sum_{i=1}^{k} s_i + \binom{\ell \cdot (d+\ell)^{n-2}}{2} \sum_{i=1}^{k} e_i + \ell^2 \cdot (d+\ell)^{2n-4} \sum_{1 \leq i < j \leq k} e_{i,j}.$$

Proof. We assume that $\tau = 1\text{-}2$. All the considerations for this τ remain the same for the case $\tau = 2\text{-}1$.

If $n = 1$ then the statement is trivial.

Suppose $n \geq 2$. Using the fact that $X_1 X_2 X_3 \ldots X_k$, has exactly $d + \ell$ occurrences of each letter i, $i = 1, 2, \ldots, k$, one can prove by induction on n, that the word $\phi^n(1)$ is a permutation of $\ell \cdot (d+\ell)^{n-2}$ copies of each word X_i, where $i = 1, 2, \ldots, k$. This implies, in particular, that $|\phi^n(1)| = k \cdot \ell \cdot (d+\ell)^{n-1}$.

An occurrence of τ in $\phi^n(1)$ can be either internal, that is when τ occurs inside a word X_i, or external, which means that τ begins in a word X_i and ends in another word X_j. In the first of these cases, since there are $\ell\cdot(d+\ell)^{n-2}$ copies of each X_i, we have $\ell\cdot(d+\ell)^{n-2}\sum_{i=1}^{k} s_i$ possibilities. In the second case, either $i=j$, which gives $\binom{\ell\cdot(d+\ell)^{n-2}}{2}\sum_{i=1}^{k} e_i$ possibilities, or $i\neq j$, in which case there are $\ell\cdot(d+\ell)^{n-2}$ possibilities to choose X_i (resp. X_j) among $\ell\cdot(d+\ell)^{n-2}$ copies of X_i (resp. X_j), and using the fact that $e_{i,j}=e_{j,i}$ (the order in which the words X_i and X_j occur in $\phi^n(1)$ is unimportant), we have $\ell^2\cdot(d+\ell)^{2n-4}\sum_{1\le i<j\le k} e_{i,j}$ possibilities. Summing all the possibilities, we finish the proof. □

Let s (resp. e) denote the vector $(s_1,s_2,\ldots,s_k)$ (resp. $(e_1,e_2,\ldots,e_k)$), where s_i and e_j are defined in Theorem 1. All of the following examples are corollaries to Theorem 1.

Example 1. *If we consider the morphism ϕ_w defined in Section 7.1 and the pattern $\tau=$ 1-2 then $d=\ell=1$, $s=(3,1,0)$, $e=(3,1,0)$ and $e_{1,2}=e_{2,1}=2$, $e_{1,3}=e_{3,1}=1$, $e_{2,3}=e_{3,2}=1$. Hence, the number of occurrences of τ is given by $N_{\phi_w}^{1\text{-}2}(1)=3$ and, for $n\ge 2$, $N_{\phi_w}^{1\text{-}2}(n)=(3\cdot 4^{n-1}+2^n)/2$. If $\tau=$ 2-1 then $s=(0,0,0)$, $e=(3,1,0)$ and $e_{1,2}=e_{2,1}=2$, $e_{1,3}=e_{3,1}=1$, $e_{2,3}=e_{3,2}=1$. Hence, $N_{\phi_w}^{2\text{-}1}(1)=0$ and, for $n\ge 2$, $N_{\phi_w}^{2\text{-}1}(n)=(3\cdot 4^{n-1}-2^n)/2$.*

Example 2. *If we consider the morphism ϕ: $1\to 1324$, $2\to\epsilon$, $3\to 14$, and $4\to 23$ then for the pattern $\tau=$ 1-2, we have $d=\ell=1$, $s=(5,0,1,1)$, $e=(6,0,1,1)$, and $e_{i,j}$, for $i\neq j$, are elements of the matrix*

$$\begin{pmatrix} - & 0 & 3 & 3\\ 0 & - & 0 & 0\\ 3 & 0 & - & 2\\ 3 & 0 & 2 & - \end{pmatrix}.$$

Hence, $N_{\phi}^{1\text{-}2}(1)=5$ and, for $n\ge 2$, $N_{\phi}^{1\text{-}2}(n)=3\cdot 4^{n-1}+11\cdot 2^{n-2}$.

Example 3. *If we consider the morphism ϕ: $1\to 13542$, $2\to 423$, $3\to\epsilon$, $4\to 5115$, and $5\to 234$ then for the pattern $\tau=$ 1-2, we have $\ell=1$, $d=2$, $s=(6,1,0,2,3)$, $e=(10,3,0,4,3)$, and $e_{i,j}$, for $i\neq j$, are elements of the matrix*

$$\begin{pmatrix} - & 6 & 0 & 8 & 6\\ 6 & - & 0 & 6 & 3\\ 0 & 0 & - & 0 & 0\\ 8 & 6 & 0 & - & 6\\ 6 & 3 & 0 & 6 & - \end{pmatrix}.$$

Hence, $N_{\phi}^{1\text{-}2}(1)=6$ and, for $n\ge 2$, $N_{\phi}^{1\text{-}2}(n)=5\cdot 9^{n-1}+2\cdot 3^{n-2}$.

Using the proof of Theorem 1, we have the following.

Theorem 2. *Let a morphism ϕ satisfy all the conditions in the statement of Theorem 1 and the pattern $\tau=\underbrace{1\text{-}1\text{-}\cdots\text{-}1}_{r \text{ times}}$. Then, for $n\ge 2$, the number of*

occurrences of τ *in* $\phi^n(1)$ *is given by* $k \cdot \binom{\ell \cdot (d+\ell)^{n-1}}{r}$, *whereas for* $n = 1$, *by* $k \cdot \binom{\ell}{r}$.

Proof. From the proof of Theorem 1, we have that if $n \geq 2$ (resp. $n = 1$) then $\phi^n(1)$ has exactly $\ell \cdot (d+\ell)^{n-1}$ (resp. ℓ) copies of each letter from $\mathcal{A}$. We can choose r of them in $\binom{\ell \cdot (d+\ell)^{n-1}}{r}$ (resp. $\binom{\ell}{r}$) ways to form the pattern τ. The rest is clear. □

The following example is a corollary to Theorem 2.

Example 4. *If we consider the morphism* ϕ_w *defined in Section 7.1 and the pattern* τ = 1-1-1-1 *then* $d = \ell = 1$, $r = 4$, *hence the number of occurrences of* τ *in* $\phi^n(1)$ *is 0, whenever* $n = 1$ *or* $n = 2$, *and* $3 \cdot \binom{2^{n-1}}{4}$ *otherwise.*

7.3 Patterns without internal dashes

In what follows we need to extend the notion of an external occurrence of a pattern. Suppose $W = AXBYC$, where A, X, B, Y and C are some subwords. We say that an occurrence of τ in W is external for a pair of words (X, Y), if this occurrence starts in X, ends in Y and is allowed to have some of its letters in B. For instance, if $W = 12324245$, where $A = 1$, $X = 23$, $B = 2$ and $Y = 424$ then an occurrence of the generalized pattern 213, namely the subword 324 is an external occurrence for (X, Y).

Theorem 3. *Let* $\mathcal{A} = \{1, 2, \ldots, k\}$ *be an alphabet and a generalized pattern* τ *has no internal dashes. Let* X_1 *begins with the letter 1 and consists of* ℓ *copies of each letter* $i \in \mathcal{A}$ ($\ell \geq 1$). *Let a morphism* ϕ *be such that*

$$1 \to X_1,\ 2 \to X_2,\ 3 \to X_3, \ldots, k \to X_k,$$

where we allow X_i *to be the empty word* ϵ *for* $i = 2, 3, \ldots, k$ (*that is,* ϕ *may be an erasing morphism*), $\sum_{i=2}^{k} |X_i| = k \cdot d$, *and each letter from* $\mathcal{A}$ *appears in the word* $X_2 X_3 \ldots X_k$ *exactly* d *times. Besides, we assume that there are no external occurrences of* τ *in* $\phi^n(1)$ *for the pair* (X_i, X_j) *for each* i *and* j. *Let* s_i *be the number of internal occurrences of* τ *in* X_i. *In particular,* $s_i = 0$, *whenever* $X_i = \epsilon$. *Then* $N_\phi^\tau(1) = s_1$ *and for* $n \geq 2$,

$$N_\phi^\tau(n) = \ell \cdot (d+\ell)^{n-2} \sum_{i=1}^{k} s_i.$$

Proof. The theorem is straightforward to prove by observing that for $n \geq 2$, $\phi^n(1)$ has $\ell \cdot (d+\ell)^{n-2}$ occurrences of each word X_i (see the proof of Theorem 1). □

Remark 5. *In order to use Theorem 3, we need to control the absence of external occurrences of a pattern τ for given τ (without internal dashes) and a morphism ϕ. To do this, we need, for any pair (X_i, X_j), to consider all the words $X_i W X_j$, where $|W| < |\tau| - 1$, and W is a permutation of a number of words from the set $\{X_1, X_2, \ldots, X_k\}$.*

The following examples are corollaries to Theorem 3.

Example 5. *If we consider the morphism ϕ_w defined in Section 7.1 and the pattern $\tau = 12$ then all the conditions of Theorems 3 hold. In this case $d = \ell = 1$ and $s = (2, 1, 0)$. Hence, the number of occurrences of the patterns 12, that is the number of rises, is given by $N_{\phi_w}^{12}(1) = 2$ and, for $n \geq 2$, $N_{\phi_w}^{12}(n) = 3 \cdot 2^{n-2}$. If $\tau = 123$ then we can apply the theorem to get that for $n \geq 2$, $N_{\phi_w}^{123}(n) = 2^{n-2}$.*

If we want to count the number of occurrences of the pattern $\tau = 21$, that is the number of descents, then we cannot apply Theorem 1, since for instance, the pair $(X_1, X_2) = (123, 13)$ has an external occurrence of τ. However, it is obvious that the number of descents in $\phi^n(1)$ is equal to $|\phi^n(1)| - N_{\phi_w}^{12}(1) - 1 = 3 \cdot 2^{n-2} - 1$.

Example 6. *If we consider the morphism ϕ: $1 \to 1243$, $2 \to 3$, $3 \to \epsilon$, and $4 \to 124$ then for the pattern $\tau = 123$, all the conditions of Theorems 3 hold. In this case $d = \ell = 1$, $s = (1, 0, 0, 1)$. Hence, for $n \geq 1$, $N_{\phi}^{123}(n) = 2^{n-1}$. For $\tau = 321$ we cannot apply Theorem 3, since the pair (X_4, X_1) has an external occurrence of τ (look at $X_4 X_2 X_1 = 12\mathbf{431}243$). Consideration of the words $X_4 X_2$ and $X_4 X_1$ implies that the theorem cannot be apply for the patterns 132 and 231 respectively. However, we can apply the theorem to the pattern 213 to prove that it does not occur in $\phi^n(1)$ for any n.*

Acknowledgement: The final version of this paper was written during the second author's (T.M.) stay at Haifa University, Haifa 31905, Israel. T.M. wants to express his gratitude to Haifa University for the support.

Bibliography

[Adian] Adian S. I.: *The Burnside problem and identities in groups.* Translated from the Russian by John Lennox and James Wiegold. Ergebnisse der Mathematik und ihrer Grenzgebiete [Results in Mathematics and Related Areas], 95. Springer-Verlag, Berlin-New York, (1979). xi+311 pp.

[BabStein] Babson E., Steingrímsson E.: Generalized permutation patterns and a classification of the Mahonian statistics, Sém. Lothar. Combin. **44** (2000), Art. B44b, 18 pp.

[Burstein] Burstein A., Enumeration of words with forbidden patterns, Ph.D. thesis, University of Pennsylvania, 1998.

[BurMans1] Burstein A. and Mansour T., Words restricted by patterns with at most 2 distinct letters, *Electronic J. of Combinatorics*, to appear (2002).

[BurMans2] Burstein A. and Mansour T., Words restricted by 3-letter generalized multipermutation patterns, preprint CO/0112281.

[BurMans3] Burstein A. and Mansour T., Counting occurrences of some subword patterns, preprint CO/0204320.

[Carpi] Carpi A.: On the number of abelian square-free words on four letters, Discrete Appl. Mathematics, Elsevier, **81** (1998), 155–167.

[Claes] A. Claesson: Generalised Pattern Avoidance, European J. Combin. **22** (2001), no. 7, 961–971.

[Dekk] Dekking F. M.: Strongly non-repetitive sequences and progression-free sets, Journal Com. Theory, Vol. **27-A**, No. 2 (1979), 181–185.

[Evdok] Evdokimov A. A.: Strongly asymmetric sequences generated by a finite number of symbols, Dokl. Akad. Nauk SSSR, 179 (1968), 1268–1271. (Russian) English translation in: Soviet Math. Dokl., 9 (1968), 536–539.

[Frid] Frid A. E.: On the frequency of factors in a DOL word, J. Automata, Languages and Combinatorics, Otto-von-Guericke-Univ., Magdeburg **3(1)** (1998), 29–41.

[Justin] Justin J.: Characterization of the repetitive commutative semigroups, Journal of Algebra (1972), no. 21, 87–90.

[Ker] Keränen V.: Abelian squares are avoidable on 4 letters, In W. Kuich, editor, Proc. ICALP'92, Lecture Notes in Comp. Sci., **623**, Springer-Verlag, Berlin (1992), 41–52.

[Knuth] Knuth D. E.: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Kol] Kolotov A. T.: Aperiodic sequences and functions of the growth of algebras, Algebra i Logika **20** (1981), no. 2, 138–154. (Russian)

[Lind] Lindenmayer A.: Mathematical models for cellular interaction in development, Parts I and II, Journal of Theoretical Biology, 18 (1968), 280–315.

[LindRoz] Lindenmayer A., Rozenberg G.: *Automata, languages, development*, North-Holland Publishing Co., Amsterdam-New York-Oxford (1976), viii+529 pp.

[Lothaire] Lothaire M.: *Combinatorics on Words*, Encyclopedia of Mathematics, Vol. **17**, Addison-Wesley (1986). Reprinted in the *Cambridge Mathematical Library*, Cambridge University Press, Cambridge UK, (1997).

[MorseHedl] Morse M., Hedlung G.: Unending chess, symbolic dynamics and a problem in semigroups, Duke Math. Journal, Vol. **11**, No. 1 (1944), 1–7.

[Pleas] Pleasants P.: Non-repetitive sequences, Proc. Camb. Phil. Soc., Vol. **68** (1970), 267–274.

[Salomaa] Salomaa A.: *Jewels of Formal Language Theory*, Computer Science Press, (1981).

[SimSch] R. Simion, F. Schmidt: Restricted permutations, European J. Combin. **6**, no. 4 (1985), 383–406.

Paper VIII

The Peano curve and counting occurrences of some patterns

The Peano curve and counting occurrences of some patterns

Sergey Kitaev and Toufik Mansour [1]

Matematik, Chalmers tekniska högskola och Göteborgs universitet,
S-412 96 Göteborg, Sweden
kitaev@math.chalmers.se, toufik@math.chalmers.se

Abstract

We introduce *Peano words*, which are words corresponding to finite approximations of the Peano space filling curve. We then find the number of occurrences of certain patterns in these words.

8.1 Introduction and Background

We write permutations as words $\pi = a_1 a_2 \cdots a_n$, whose letters are distinct and usually consist of the integers $1, 2, \ldots, n$.

An occurrence of a pattern p in a permutation π is "classically" defined as a subsequence in π (of the same length as the length of p) whose letters are in the same relative order as those in p. Formally speaking, for $r \leq n$, we say that a permutation σ in the symmetric group $\mathcal{S}_n$ has an occurrence of the pattern $p \in \mathcal{S}_r$ if there exist $1 \leq i_1 < i_2 < \cdots < i_r \leq n$ such that $p = \sigma(i_1)\sigma(i_2)\ldots\sigma(i_r)$ in reduced form. The *reduced form* of a permutation σ on a set $\{j_1, j_2, \ldots, j_r\}$, where $j_1 < j_2 < \cdots < j_r$, is a permutation σ_1 obtained by renaming the letters of the permutation σ so that j_i is renamed i for all $i \in \{1, \ldots, r\}$. For example, the reduced form of the permutation 3651 is 2431. The first case of classical patterns studied was that of permutations avoiding a pattern of length 3 in $\mathcal{S}_3$. Knuth [Knuth] found that, for any $\tau \in \mathcal{S}_3$, the number $|\mathcal{S}_n(\tau)|$ of n-permutations avoiding τ is C_n, the nth Catalan number. Later, Simion and Schmidt [SimSch] determined the number $|\mathcal{S}_n(P)|$ of permutations in $\mathcal{S}_n$ simultaneously avoiding any given set of patterns $P \subseteq \mathcal{S}_3$.

In [BabStein] Babson and Steingrímsson introduced *generalised permutation patterns* that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. In order to avoid confusion we write a "classical" pattern, say 231, as 2-3-1, and if we write, say 2-31, then we mean that if this pattern occurs in the permutation, then the letters in the permutation that correspond to 3 and 1 are adjacent. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, whereas the pattern 2-3-1 occurs, in addition, in the subwords 562 and 563. A motivation for introducing these patterns in [BabStein] was the study of Mahonian

[1] Research financed by EC's IHRP Programme, within the Research Training Network "Algebraic Combinatorics in Europe", grant HPRN-CT-2001-00272

statistics. A number of interesting results on generalised patterns were obtained in [Claes]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were shown there.

Burstein [Burstein] considered words instead of permutations. In particular, he found the number $|[k]^n(P)|$ of words of length n in a k-letter alphabet that avoid all patterns from a set $P \subseteq \mathcal{S}_3$ simultaneously. Burstein and Mansour [BurMans1] (resp. [BurMans2, BurMans3]) considered forbidden patterns (resp. generalized patterns) with repeated letters.

The most attention, in the papers on classical or generalized patterns, is paid to finding exact formulas and/or generating functions for the number of words or permutations avoiding, or having k occurrences of, certain patterns. In [KitMans] the present authors suggested another problem, namely counting the number of occurrences of certain patterns in certain words. These words were chosen to be the set of all finite approximations of a sequence generated by a *morphism* with certain restrictions. A motivation for this choice was the interest in studying classes of sequences and words that are defined by iterative schemes [Lothaire, Salomaa].

In the present paper we also study the number of occurrences of certain patterns in certain words. But here we choose these words to be the discrete analogue given by Evdokimov, of subdivision stages from which the *Peano curve* is obtained. We call these words the *Peano words*. The Peano curve was studied by the Italian mathematician Giuseppe Peano in 1890 as an example of a continuous space filling curve. We consider the Peano words and find the number of occurrences of the patterns 12, 21, 1^ℓ, [x-y^ℓ), (x^ℓ-y] and [x-y^ℓ-z], where $x, y, z \in \{1, 2, 3\}$, $y^\ell = y$-y-$\cdots$-y (ℓ times), and "[" in $p = [x - w)$ indicates that in an occurrence of p, the letter corresponding to the x must be the first letter of the word.

8.2 The Peano curve and the Peano words

We follow [GelbOlm] and present a description (of a curve that fills the unit square $S = [0,1] \times [0,1]$) given in 1891 by the German mathematician D. Hilbert.

As indicated in Figure 8.3, the idea is to subdivide S and the unit interval $I = [0,1]$ into 4^n closed subsquares and subintervals, respectively, and to set up a correspondence between subsquares and subintervals so that inclusion relationships are preserved (at each stage of subdivision, if a square corresponds to an interval, then its subsquares correspond to subintervals of that interval).

We now define the continuous mapping f of I onto S: If $x \in I$, then at each stage of subdivision x belongs to *at least* one closed subinterval. Select either one (if there are two) and associate the corresponding square. In this way a decreasing sequence of closed squares is obtained corresponding to a decreasing sequence of closed intervals. This sequence of closed squares has the property that there is exactly one point belonging to all of them. This point is by definition $f(x)$. It can be shown that the point $f(x)$ is well-defined, that is, independent of any choice of intervals containing x; the range of f is S; and f

is continuous.

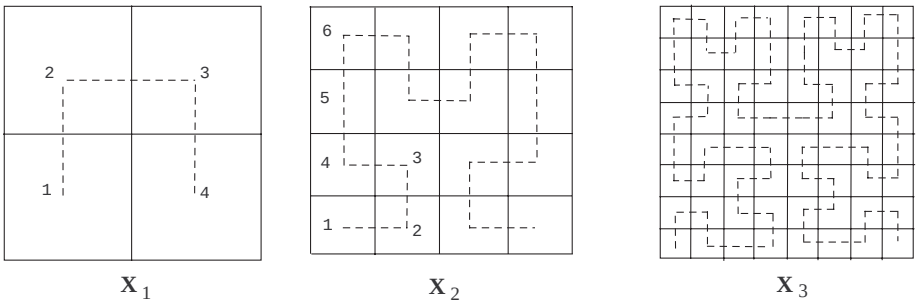


Figure 8.3: the Peano words

The following discrete analogue of the Peano curve was given by Evdokimov [Evdok]. We consider a subdivision stage (an iteration), go through the curve inside S starting in the point 1 (see Figure 8.3), and coding any movement "up" by 1, "right" by 2, "down" by 3, "left" by 4. Thus, we start with the first iteration $X_1 = 123$, the second iteration is $X_2 = 214112321233432$. More generally, it is easy to see that the n-th iteration is given by

$$X_n = \varphi_1(X_{n-1})1X_{n-1}2X_{n-1}3\varphi_2(X_{n-1}),$$

where the function $\varphi_1(A)$ reverses the letters in the word A and makes the substitution corresponding to the permutation 4123, that is, 1 becomes 4 etc. The function φ_2 does the same, except with 4123 replaced by 2341. In this paper, we are interested in the words X_n, for $n = 1, 2, \ldots$, which appear as the subdivision stages of the Peano curve. We call these words the Peano words.

8.3 The main results

It is easy to see that the length of the curve after the n-th iteration is $|X_n| = 4^n - 1$. Moreover, the following lemma holds.

Lemma 1. *The number of occurrences of the letters* 1, 2, 3 *and* 4 *in* X_n *is given by* 4^{n-1}, $4^{n-1} + 2^{n-1} - 1$, 4^{n-1} *and* $4^{n-1} - 2^{n-1}$ *respectively.*

Proof. Suppose d_1^n (resp. d_2^n, d_3^n, d_4^n) denote the number of occurrences of the letter 1 (resp. 2,3,4) in the word X_n. It is easy to see, using the way we construct X_n, that

$$\begin{pmatrix} d_1^n \\ d_2^n \\ d_3^n \\ d_4^n \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} d_1^{n-1} \\ d_2^{n-1} \\ d_3^{n-1} \\ d_4^{n-1} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

Using the diagonalization of the matrix in the identity above, namely the fact that

$$\begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix} =$$
$$\begin{pmatrix} -1 & -1 & 0 & 1 \\ 1 & 0 & -1 & 1 \\ -1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} -1/4 & 1/4 & -1/4 & 1/4 \\ -1/2 & 0 & 1/2 & 0 \\ 0 & -1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix},$$

we get that the vector $(d_1^n, d_2^n, d_3^n, d_4^n)'$ is equal to the vector

$$(4^{n-1}, 4^{n-1} + 2^{n-1} - 1, 4^{n-1}, 4^{n-1} - 2^{n-1})'.$$

□

As a corollary to Lemma 1 we have the following.

Corollary 11. *The number of occurrences of the pattern* $\tau = \underbrace{1\text{-}1\text{-}\cdots\text{-}1}_{\ell} = 1^\ell$ *in* X_n *is equal to*

$$\binom{4^{n-1} - 2^{n-1}}{\ell} + 2\binom{4^{n-1}}{\ell} + \binom{4^{n-1} + 2^{n-1} - 1}{\ell}.$$

Proof. The number of occurrences of a subsequence $\underbrace{i\text{-}i\text{-}\cdots\text{-}i}_{\ell}$ in X_n, for $i = 1, 2, 3, 4$, is obviously given by $\binom{d_i^n}{\ell}$, where d_i^n is defined and determined in the proof of Lemma 1. The rest is easy to see. □

Definition 9. *Let* $r(A)$ *(resp.* $d(A)$*) denote the number of occurrences of the pattern* 12 *(resp.* 21*), that is the number of rises (resp. descents), in a word* A*.*

Lemma 2. *Suppose* $A = 1X3$ *and* $B = 2Y2$ *for some words* X *and* Y*. Then* $r(\varphi_1(A)) = d(A) + 1$*,* $d(\varphi_1(A)) = r(A) - 1$*,* $r(\varphi_2(B)) = d(B)$ *and* $d(\varphi_2(B)) = r(B)$*.*

Proof. If $\bar{A}$ and $\bar{B}$ denote the reverses of A and B then $r(\bar{A}) = d(A)$, $d(\bar{A}) = r(A)$, $r(\bar{B}) = d(B)$, and $d(\bar{B}) = r(B)$.

We consider two factorizations of each word $\bar{A}$ and $\bar{B}$. We can write $\bar{A}$ as

$$\bar{A} = 3A_1 \underbrace{1 \dots 1}_{i_1} A_2 \underbrace{1 \dots 1}_{i_2} A_3 \dots A_k \underbrace{1 \dots 1}_{i_k},$$

where A_i, for $i = 1, 2, \dots, k$ is a word over the alphabet $\{2, 3, 4\}$, only A_1 can be the empty word ϵ, and $i_j \geq 1$ for $j = 1, 2 \dots, k$. Also, we can write $\bar{A}$ as

$$\bar{A} = 3A_0' \underbrace{4 \dots 4}_{i_1'} A_1' \underbrace{4 \dots 4}_{i_2'} A_2' \dots A_{k-1}' \underbrace{4 \dots 4}_{i_k'} A_k' 1,$$

where $A_i^{'}$, for $i = 0, 1, \ldots, k$ is a word over the alphabet $\{1, 2, 3\}$, only $A_0^{'}$ and $A_k^{'}$ can be ϵ, and $i_j^{'} \geq 1$ for $j = 1, 2 \ldots, k$.

The word $\bar{B}$ can be written as

$$\bar{B} = 2B_0 \underbrace{1 \ldots 1}_{j_1} B_1 \underbrace{1 \ldots 1}_{j_2} B_2 \ldots B_{\ell-1} \underbrace{1 \ldots 1}_{j_\ell} B_\ell 2,$$

where B_i, for $i = 0, 1, \ldots, \ell$, is a word over the alphabet $\{2, 3, 4\}$, only B_0 and B_ℓ can be ϵ, and $j_i \geq 1$ for $i = 1, 2 \ldots, \ell$. Also, $\bar{B}$ can be written as

$$\bar{B} = 2B_0^{'} \underbrace{4 \ldots 4}_{j_1^{'}} B_1^{'} \underbrace{4 \ldots 4}_{j_2^{'}} B_2^{'} \ldots B_{\ell-1}^{'} \underbrace{4 \ldots 4}_{j_\ell^{'}} B_\ell^{'} 2,$$

where $B_i^{'}$, for $i = 0, 1, \ldots, \ell$, is a word over the alphabet $\{1, 2, 3\}$, only $B_0^{'}$ and $B_\ell^{'}$ can be ϵ, and $j_i^{'} \geq 1$ for $i = 1, 2 \ldots, \ell$.

It follows from the definitions that $\varphi_1(A)$ and $\varphi_1(B)$ (resp. $\varphi_2(A)$ and $\varphi_2(B)$) are obtained from $\bar{A}$ and $\bar{B}$ by permuting the letters with the function π_1 (resp. π_2) that acts as the permutation 4123 (resp. 2341).

We now consider the first factorizations of $\bar{A}$ and $\bar{B}$, and the function π_1. It is easy to see that if W is equal to A_i, or B_i, or $3A_1$, or $2B_0$, or $B_\ell 2$, then $r(W) = r(\pi_1(W))$ and $d(W) = d(\pi_1(W))$, since π_1 is an order-preserving function when it acts from the set $\{2, 3, 4\}$ to the set $\{1, 2, 3\}$. From the other hand, occurrences of the rises 12, 13 and 14 (resp. the descents 41, 31 and 21) in $\bar{A}$ and $\bar{B}$, give occurrences of the descents 41, 42 and 43 (resp. the rises 34, 24 and 14) in $\pi_1(\bar{A})$ and $\pi_1(\bar{B})$ respectively. If we now read the first factorizations of $\bar{A}$ and $\bar{B}$ from the left to the right, then the occurrences of the subwords $a1$ alternate with the occurrences of the subwords $1b$, where $a, b \in \{2, 3, 4\}$. Moreover, in the factorization of $\bar{A}$, we begin and end with the subword $a1$ for some $a \in \{2, 3, 4\}$, which gives that $d(A) + 1 = r(\bar{A}) + 1 = r(\pi_1(\bar{A})) = r(\varphi_1(A))$ and $r(A) - 1 = d(\bar{A}) - 1 = d(\pi_1(\bar{A})) = d(\varphi_1(A))$; in the factorization of $\bar{B}$, we begin with the subword $a1$ and end with the subword $1b$ for some $a, b \in \{2, 3, 4\}$, which gives that $d(B) = r(\bar{B}) = r(\pi_1(\bar{B})) = r(\varphi_1(B))$ and $r(B) = d(\bar{B}) = d(\pi_1(\bar{B})) = d(\varphi_1(B))$.

If we consider the second factorizations of $\bar{A}$ and $\bar{B}$, and the function π_2, one can see that if W is equal to $A_i^{'}$, or $B_i^{'}$, or $3A_0^{'}$, or $A_k^{'}1$, or $2B_0^{'}$, or $B_\ell^{'}2$, then $r(W) = r(\pi_2(W))$ and $d(W) = d(\pi_2(W))$, since π_2 is an order-preserving function when it acts from the set $\{1, 2, 3\}$ to the set $\{2, 3, 4\}$. From the other hand, occurrences of the rises 14, 24 and 34 (resp. the descents 41, 42 and 43) in $\bar{A}$ and $\bar{B}$, give occurrences of the descents 21, 31 and 41 (resp. the rise 12, 13 and 14) in $\pi_2(\bar{A})$ and $\pi_2(\bar{B})$ respectively. If we now read the second factorizations of $\bar{A}$ and $\bar{B}$ from the left to the right, then the occurrences of the subwords $a4$ alternate with the occurrences of the subwords $4b$, where $a, b \in \{1, 2, 3\}$. Moreover, in both cases, we begin with the subword $a4$ and end with the subword $4b$ for some $a, b \in \{1, 2, 3\}$, which gives that $d(A) = r(\bar{A}) = r(\pi_2(\bar{A})) = r(\varphi_2(A))$, $r(A) = d(\bar{A}) = d(\pi_2(\bar{A})) = d(\varphi_2(A))$, $d(B) = r(\bar{B}) = r(\pi_2(\bar{B})) = r(\varphi_2(B))$ and $r(B) = d(\bar{B}) = d(\pi_2(\bar{B})) = d(\varphi_2(B))$. □

Theorem 1. *Let r_n (resp. d_n) be the number of occurrences of the pattern* 12 *(resp.* 21*) in X_n. Then for all $k \geq 0$,*

$$\begin{aligned} r_{2k+1} &= \tfrac{2}{5}(4 \cdot 16^k + 1), \\ r_{2k+2} &= \tfrac{2}{5}(16^{k+1} - 1), \\ d_{2k+1} &= \tfrac{8}{5}(16^k - 1), \\ d_{2k+2} &= \tfrac{2}{5}(16^{k+1} - 1). \end{aligned}$$

Proof. Using the properties of φ_1 and φ_2, as well as the way we construct X_n, it is easy to check by induction, that X_{2k+1} and X_{2k+2} can be factorized as follow:

$$X_{2k+1} = \underbrace{1X^{(1)}1}_{\varphi_1(X_{4k})}1\underbrace{2Y^{(1)}2}_{X_{4k}}2\underbrace{2Y^{(1)}2}_{X_{4k}}3\underbrace{3Z^{(1)}3}_{\varphi_2(X_{4k})},$$

$$X_{2k+2} = \underbrace{2X^{(2)}4}_{\varphi_1(X_{4k+1})}\ 1\underbrace{1Y^{(2)}3}_{X_{4k+1}}2\underbrace{1Y^{(2)}3}_{X_{4k+1}}3\ \underbrace{4Z^{(2)}2}_{\varphi_2(X_{4k+1})}\ ,$$

where $X^{(i)}$, $Y^{(i)}$ and $Z^{(i)}$ are some words for $i = 1, 2$.

Suppose we know r_{2k+1} and d_{2k+1} for some k. Since $X_{2k+1} = 1W3$ for some word W, using Lemma 2 and the factorization of the word X_{2k+2}, we can find r_{2k+2} and d_{2k+2}. Indeed, $\varphi_1(X_{4k+1})$ has $d_{2k+1}+1$ rises and $r_{2k+1}-1$ descents; $\varphi_2(X_{4k+1})$ has d_{2k+1} rises and r_{2k+1} descents; two subwords X_{2k+1} give $2r_{2k+1}$ rises and $2d_{2k+1}$ descents. Besides, we have some extra rises and descents appeared between different blocks of the decomposition. They are one extra rise between the letter 3 and the subword $\varphi_2(X_{4k+1})$, and 3 extra descents between the subword $\varphi_1(X_{4k+1})$ and the letter 1, the subword X_{4k+1} and the letter 2, the letter 2 and the subword X_{4k+1}. Thus, $r_{2k+2} = 2r_{2k+1} + 2d_{2k+1} + 2$ and $d_{2k+2} = 2r_{2k+1} + 2d_{2k+1} + 2$, which shows, in particular, that for even n, in X_n, the number of rises is equal to the number of descents.

We now analyze the factorization of X_{2k+3}, which is similar to that of X_{2k+1}. Using the fact that $X_{2k+2} = 2W'2$ for some word W' and Lemma 2, we can find r_{2k+3} and d_{2k+3}. Indeed, we can use the similar considerations as above to get $r_{2k+3} = 2r_{2k+2} + 2d_{2k+2} + 2 = 8r_{2k+1} + 8d_{2k+1} + 10$ and $d_{2k+3} = 2r_{2k+2} + 2d_{2k+2} = 8r_{2k+1} + 8d_{2k+1} + 8$. Thus, if x_k denote the vector $(r_{2k+1}, d_{2k+1})'$ then

$$x_{k+1} = \begin{pmatrix} 8 & 8 \\ 8 & 8 \end{pmatrix} x_k + \begin{pmatrix} 10 \\ 8 \end{pmatrix},$$

with $x_0 = (2, 0)$, since in $X_1 = 123$, there are two rises and no descents. This recurrence relation, using diagonalization of the matrix in it, leads us to

$$x_k = (\frac{2}{5}(4 \cdot 16^k + 1), \frac{8}{5}(16^k - 1))'.$$

Finally, $r_{2k+2} = d_{2k+2} = 2r_{2k+1} + 2d_{2k+1} + 2 = \frac{2}{5}(16^{k+1} - 1)$. □

Let $N_\tau(W)$ denote the number of occurrences of the pattern τ in the word W.

Using Lemma 1 and the proof of Theorem 1, we can count, for X_n, the number of occurrences of the patterns $\tau_1(x,y) = [x\text{-}y^\ell)$, $\tau_2(x,y) = x^\ell\text{-}y]$ and $\tau_3(x,y,z) = [x\text{-}y^\ell\text{-}z]$, where $x,y,z \in \{1,2,3\}$, $y^\ell = y\text{-}y\text{-}\cdots\text{-}y$ (ℓ times), and "[" in $p = [x - w)$ indicates that in an occurrence of p, the letter corresponding to the x must be the first letter of the word, whereas "]" in $\tau_3(x,y,z)$ indicates that in an occurrence of $\tau_3(x,y,z)$, the letter corresponding to the z must be the last (rightmost) letter of the word.

If we consider, for instance, the pattern $\tau_1(1,2) = [1\text{-}2^\ell)$ then the letter 1 in this pattern must correspond to the leftmost letter of the word X_n. Now if $n = 2k+1$ then from the proof of Theorem 1 $X_n = 1W$ for some word W, which means that to the sequence 2^ℓ there can correspond any subsequence i^ℓ in X_n, where $i = 2,3,4$. Thus, using Lemma 1 and the way we prove Corollary 11, there are $\binom{4^{2k}-2^{2k}}{\ell} + \binom{4^{2k}}{\ell} + \binom{4^{2k}+2^{2k}-1}{\ell}$ occurrences of the pattern $\tau_1(1,2)$ in X_{2k+1}. If $n = 2k+2$ then $X_n = 2W$ for some word W and for the sequence 2^ℓ there correspond any subsequence i^ℓ in X_n, where $i = 3,4$. Thus, $N_{\tau_1(1,2)}(X_{2k+2}) = \binom{4^{2k}-2^{2k}}{\ell} + \binom{4^{2k}}{\ell}$.

In the example above, as well as in the following considerations, we assume ℓ to be greater then 0. If $\ell = 0$ then obviously $N_{\tau_1(x,y)}(X_n) = N_{\tau_2(x,y)}(X_n) = 1$, whereas $N_{\tau_3(x,y,z)}(X_n)$ is equal to 1 if $x < z$ and $n = 2k+1$, or $x = z$ and $n = 2k+2$, and it is equal to 0 otherwise.

When we consider $\tau_3(x,y,z)(X_n)$, we observe that since $X_{2k+2} = 2W2$ for some W, $N_{\tau_3(x,y,z)}(X_{2k+2}) = 0$, whenever $x \neq z$. Also, since $X_{2k+1} = 1W3$ for some W, $N_{\tau_3(x,y,z)}(X_{2k+1}) = 0$, whenever $x \geq z$.

Let us consider the pattern $\tau_3(2,1,3) = [2\text{-}1^\ell\text{-}3]$. As it was mentioned before, $N_{\tau_3(2,1,3)}(X_{2k+2}) = 0$. But, if we consider $X_{2k+1} = 1W3$, then it is easy to see that $N_{\tau_3(2,1,3)}(X_{2k+1}) = 0$, since the leftmost letter of X_{2k+1} is the least letter, which means that it cannot correspond to the letter 2 in the pattern. As one more example, we can consider the pattern $\tau_3(1,1,2) = [1\text{-}1^\ell\text{-}2]$. We are only interested in case $X_n = X_{2k+1}$, since $N_{\tau_3(1,1,2)}(X_{2k+2}) = 0$. The number of occurrences of the pattern is obviously given by the number of ways to choose ℓ letters among $4^{2k} - 1$ letters 1 (totally, there are 4^{2k} letters 1 according to Lemma 1, but we cannot consider the leftmost 1 since it corresponds to the leftmost 1 in the pattern). Thus, $N_{\tau_3(1,1,2)}(X_{2k+1}) = \binom{4^{2k}-1}{\ell}$.

All the other cases of x, y, z in the patterns $\tau_1(x,y)$, $\tau_2(x,y)$ and $\tau_3(x,y,z)$ can be considered in the same way. Let S_1 and S_2 denote the following:

$$S_1 = \binom{4^{2k}-2^{2k}}{\ell} + \binom{4^{2k}}{\ell} + \binom{4^{2k}+2^{2k}-1}{\ell}, \quad S_2 = \binom{4^{2k+1}}{\ell} + \binom{4^{2k+1}-2^{2k+1}}{\ell}.$$

The tables below give all the results concerning the patterns under consideration, except those triples (x,y,z), for which $N_{\tau_3(x,y,z)}(X_n) = 0$ for all n.

x	y	$N_{\tau_1(x,y)}(X_{2k+1})$	$N_{\tau_2(x,y)}(X_{2k+1})$	$N_{\tau_1(x,y)}(X_{2k+2})$	$N_{\tau_2(x,y)}(X_{2k+2})$
1	1	$\binom{4^{2k}-1}{\ell}$	$\binom{4^{2k}-1}{\ell}$	$\binom{4^{2k+1}+2^{2k+1}-1}{\ell}$	$\binom{4^{2k+1}+2^{2k+1}-1}{\ell}$
1	2	S_1	$\binom{4^{2k}}{\ell} + \binom{4^{2k}+2^{2k}-1}{\ell}$	S_2	$\binom{4^{2k+1}}{\ell}$
2	1	0	$\binom{4^{2k}-2^{2k}}{\ell}$	$\binom{4^{2k+1}}{\ell}$	S_2

x	y	z	$N_{\tau_3(x,y,z)}(X_{2k+1})$	$N_{\tau_3(x,y,z)}(X_{2k+2})$
1	1	1	0	$\binom{4^{2k+1}-2}{\ell}$
1	1	2	$\binom{4^{2k}-1}{\ell}$	0
1	2	1	0	S_2
1	2	2	$\binom{4^{2k}-1}{\ell}$	0
2	1	2	0	$\binom{4^{2k+1}}{\ell}$
1	2	3	$\binom{4^{2k}+2^{2k}-1}{\ell}$	0
1	3	2	$\binom{4^{2k}-2^{2k}}{\ell}$	0

Bibliography

[BabStein] Babson E., Steingrímsson E.: Generalized permutation patterns and a classification of the Mahonian statistics, Sém. Lothar. Combin. **44** (2000), Art. B44b, 18 pp.

[Burstein] Burstein A., Enumeration of words with forbidden patterns, Ph.D. thesis, University of Pennsylvania, 1998.

[BurMans1] Burstein A., Mansour T.: Words restricted by patterns with at most 2 distinct letters, *Electronic J. of Combinatorics*, to appear (2002).

[BurMans2] Burstein A., Mansour T.: Words restricted by 3-letter generalized multipermutation patterns, preprint CO/0112281.

[BurMans3] Burstein A., Mansour T.: Counting occurrences of some subword patterns, preprint CO/0204320.

[Claes] A. Claesson: Generalised Pattern Avoidance, European J. Combin. **22** (2001), no. 7, 961–971.

[GelbOlm] Gelbaum B., Olmsted J.: *Counterexamples in Analysis*, Holden-day, San Francisco, London, Amsterdam, (1964).

[KitMans] Kitaev S., Mansour T.: Counting the occurrences of generalized patterns in words generated by a morphism, preprint CO/0210170.

[Knuth] Knuth D. E.: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Lothaire] Lothaire M.: *Combinatorics on Words*, Encyclopedia of Mathematics, Vol. **17**, Addison-Wesley (1986). Reprinted in the *Cambridge Mathematical Library*, Cambridge University Press, Cambridge UK (1997).

[Salomaa] Salomaa A.: *Jewels of Formal Language Theory*, Computer Science Press (1981).

[SimSch] Simion R., Schmidt F.: Restricted permutations, European J. Combin. **6**, no. 4 (1985), 383–406.

Paper IX

The sigma-sequence and counting occurrences of some patterns, subsequences and subwords

The sigma-sequence and counting occurrences of some patterns, subsequences and subwords

Sergey Kitaev [1]

Abstract

We consider *sigma-words*, which are words used by Evdokimov in the construction of the sigma-sequence [Evdok]. We then find the number of occurrences of certain patterns, subsequences and subwords in these words.

9.1 Introduction and Background

We write permutations as words $\pi = a_1 a_2 \cdots a_n$, whose letters are distinct and usually consist of the integers $1, 2, \ldots, n$.

An occurrence of a pattern p in a permutation π is "classically" defined as a subsequence in π (of the same length as the length of p) whose letters are in the same relative order as those in p. Formally speaking, for $r \leq n$, we say that a permutation σ in the symmetric group $\mathcal{S}_n$ has an occurrence of the pattern $p \in \mathcal{S}_r$ if there exist $1 \leq i_1 < i_2 < \cdots < i_r \leq n$ such that $p = \sigma(i_1)\sigma(i_2)\ldots\sigma(i_r)$ in reduced form. The *reduced form* of a permutation σ on a set $\{j_1, j_2, \ldots, j_r\}$, where $j_1 < j_2 < \cdots < j_r$, is a permutation σ_1 obtained by renaming the letters of the permutation σ so that j_i is renamed i for all $i \in \{1, \ldots, r\}$. For example, the reduced form of the permutation 3651 is 2431. The first case of classical patterns studied was that of permutations avoiding a pattern of length 3 in $\mathcal{S}_3$. Knuth [Knuth] found that, for any $\tau \in \mathcal{S}_3$, the number $|\mathcal{S}_n(\tau)|$ of n-permutations avoiding τ is C_n, the nth Catalan number. Later, Simion and Schmidt [SimSch] determined the number $|\mathcal{S}_n(P)|$ of permutations in $\mathcal{S}_n$ simultaneously avoiding any given set of patterns $P \subseteq \mathcal{S}_3$.

In [BabStein] Babson and Steingrímsson introduced *generalised permutation patterns* that allow the requirement that two adjacent letters in a pattern must be adjacent in the permutation. In order to avoid confusion we write a "classical" pattern, say 231, as 2-3-1, and if we write, say 2-31, then we mean that if this pattern occurs in the permutation, then the letters in the permutation that correspond to 3 and 1 are adjacent. For example, the permutation $\pi = 516423$ has only one occurrence of the pattern 2-31, namely the subword 564, whereas the pattern 2-3-1 occurs, in addition, in the subwords 562 and 563. A motivation for introducing these patterns in [BabStein] was the study of Mahonian statistics. A number of interesting results on generalised patterns were obtained in [Claes]. Relations to several well studied combinatorial structures, such as set partitions, Dyck paths, Motzkin paths and involutions, were shown there.

[1] Matematiska Institutionen, Chalmers tekniska högskola and Göteborgs universitet, S-412 96 Göteborg, Sweden; E-mail: kitaev@math.chalmers.se

Burstein [Burstein] considered words instead of permutations. In particular, he found the number $|[k]^n(P)|$ of words of length n in a k-letter alphabet that avoid all patterns from a set $P \subseteq \mathcal{S}_3$ simultaneously. Burstein and Mansour [BurMans1] (resp. [BurMans2, BurMans3]) considered forbidden patterns (resp. generalized patterns) with repeated letters.

The most attention, in the papers on classical or generalized patterns, is paid to finding exact formulas and/or generating functions for the number of words or permutations avoiding, or having k occurrences of, certain patterns. In [KitMans1] the authors suggested another problem, namely counting the number of occurrences of certain patterns in certain words. These words were chosen to be the set of all finite approximations of a sequence generated by a *morphism* with certain restrictions. A motivation for this choice was the interest in studying classes of sequences and words that are defined by iterative schemes [Lothaire, Salomaa]. In [KitMans2] the authors also studied the number of occurrences of certain patterns in certain words. But there they choose these words to be the subdivision stages from which the *Peano curve* is obtained. The authors called these words the *Peano words*. The Peano curve was studied by the Italian mathematician Giuseppe Peano in 1890 as an example of a continuous space filling curve.

In the present paper we consider the *sigma-words*, which are words used by Evdokimov in construction of the *sigma-sequence* [Evdok]. Evdokimov used this sequence to construct chains of maximal length in the n-dimensional unit cube. Independent interest to the sigma-sequence appears in connection with the well-known *Dragon curve*, discovered by physicist John E. Heighway and defined as follows: we fold a sheet of paper in half, then fold in half again, and again, etc. and then unfold in such way that each crease created by the folding process is opened out into a 90-degree angle. The "curve" refers to the shape of the partially unfolded paper as seen edge on. If one travels along the curve, some of the creases will represent turns to the left and others turns to the right. Now if 1 indicates a turn to the right, and 2 to the left, and we start travelling along the curve indicating the turns, we get the sigma-sequence [Evdokimov]. In [Kitaev] the sigma-sequence was studied from another point of view. It was proved there that this sequence cannot be defined by iterated morphism.

Since the sigma-sequence w_σ is a sequence in a 2-letter alphabet, we consider patterns in 2-letter alphabets. Moreover, the patterns in a 1-letter alphabet (for example 1-1-1) correspond to two subsequences (for this example, these subsequences are 1-1-1 and 2-2-2), whereas the patterns in a 2-letter alphabet (with at least one letter 2) uniquely determine the subsequences in w_σ that correspond to them, and conversely. For example, an occurrence of the pattern 1-2-1 is an occurrence of the subsequence 1-2-1, whereas an occurrence of the subsequence (subword) 211 is an occurrence of the pattern 211. Thus, any our result for a pattern, can be interpreted in term of subsequences or subwords, depending on the context, and conversely.

In our paper we give either an explicit formula or recurrence relation for the number of occurrences for some classes of patterns, subwords and subsequences in the sigma-words. In particular, Theorem 1, allows to find the number of

occurrences of an arbitrary generalized pattern without internal dashes of length ℓ, provided we know four certain numbers that can be easily calculated for the sigma-words C_k, D_k, C_{k+1} and D_{k+1} (to be defined below), where $k = \lceil \log_2 \ell \rceil$. Theorem 2 gives a recurrence relation for counting occurrences of patterns of the form τ_1-τ_2. In Section 9.6 we discuss occurrences of patterns of the form τ_1-τ_2-$\cdots$-τ_k, where the pattern τ_i does not overlap with the patterns τ_{i-1} and τ_{i+1} for $i = 1, 2, \ldots, k-1$. Finally, Section 9.7 deals with patterns of the form $[\tau_1$-τ_2-$\cdots$-$\tau_k]$, $[\tau_1$-τ_2-$\cdots$-$\tau_k)$ and $(\tau_1$-τ_2-$\cdots$-$\tau_k]$ in Babson and Steingrímsson notation, which means that we use "[x" in a pattern p to indicate that in an occurrence of p, the letter corresponding to the x must be the first letter of a word under consideration, whereas if we use "y]", we mean that the letter corresponding to y must be the last (rightmost) letter in the word.

9.2 Preliminaries

In [Evdok, Yab], Evdokimov constructed chains of maximal length in the n-dimensional unit cube using the *sigma-sequence*. The sigma-sequence w_σ was defined there by the following inductive scheme:

$$C_1 = 1, \quad D_1 = 2$$
$$C_{k+1} = C_k 1 D_k, \quad D_{k+1} = C_k 2 D_k$$
$$k = 1, 2, \ldots$$

and $w_\sigma = \lim\limits_{k \to \infty} C_k$. Thus, the initial letters of w_σ are 11211221112212.... We call the words C_k the *sigma words*. The first four values of the sequence $\{C_k\}_{k \geq 1}$ are 1, 112, 1121122, 112112211122122.

In [Kitaev] an equivalent definition of w_σ was given: any natural number $n \neq 0$ can be presented unambiguously as $n = 2^t(4s + \sigma)$, where $\sigma < 4$, and t is the greatest natural number such that 2^t divides n. If n runs through the natural numbers then σ runs through some sequence consisting of 1 and 3. If we substitute 3 by 2 in this sequence, we get w_σ.

In this paper we count occurrences of patterns in the sigma-words, which are particular initial subwords of w_σ. However, the challenging question is to find the number of occurrences of patterns or subwords in an arbitrary initial subword of w_σ, or more generally, in a subword of w_σ starting in the position i and ending in the position j.

It turns out that for counting occurrences of certain patterns or subwords in C_n, one needs to know the number of occurrences of certain patterns in D_n. So, in the most cases, we give results for both C_n and D_n. However, our main purpose is the words C_n for $n \geq 1$, and in some propositions and examples we do not consider D_n.

In what follows, we give initial values for the words C_i and D_i:

$C_1 = 1$ $D_1 = 2$

$C_2 = 112$ $D_2 = 122$

$C_3 = 1121122$ $D_3 = 1122122$

$C_4 = 112112211122122$ $D_4 = 112112221122122$

$C_5 = 1121122111221221112112221122122$

$D_5 = 1121122111221222112112221122122$

We now give some other definitions.

A *descent* (resp. *rise*) in a word $w = a_1 a_2 \dots a_n$ is an i such that $a_i > a_{i+1}$ (resp. $a_i < a_{i+1}$). It follows from the definitions that an occurrence of a descent (resp. rise) is an occurrence of the pattern 21 (resp. 12).

Let c_n^τ (resp. d_n^τ) denote the number of occurrences of the pattern τ in C_n (resp. D_n).

Suppose a word $W = AaB$, where A and B are some words of the same length, and a is a letter. We define the *kernel of order k* for the word W to be the subword consisting of the $k-1$ rightmost letters of A, the letter a, and the $k-1$ leftmost letters of B. We denote it by $\mathcal{K}_k(W)$. For example, $\mathcal{K}_3(111211221) = 12112$. If $|A| < k-1$ then we assume $\mathcal{K}_k(W) = \epsilon$, that is, the kernel in this case is the empty word. Also, $m_k(\tau, W)$ denotes the number of occurrences of the pattern (or the word, or the subsequence depending on the context) τ in $\mathcal{K}_k(W)$.

We denote x-x-$\cdots$-x (ℓ times) by x^ℓ. Also, $\lceil a \rceil$ denotes the least natural number b such that $a \le b$.

9.3 Patterns 1-1-$\cdots$-1, 1-2 and 2-1

It is easy to see that $|C_n| = |D_n| = 2^n - 1$. The following lemma gives the number of the letters 1 and 2 in C_n and D_n.

Lemma 1. *The number of 1s (resp. 2s) in C_n is 2^{n-1} (resp. $2^{n-1}-1$). The number of 1s (resp. 2s) in D_n is $2^{n-1}-1$ (resp. 2^{n-1}).*

Proof. It is enough to find the number of 1s c_n and d_n in C_n and D_n respectively, since the number of 2s in C_n and D_n are obviously equal to $|C_n| - c_n$ and $|D_n| - d_n$ respectively.

It is easy to see from the structure of C_n and D_n that

$$\begin{cases} c_n = c_{n-1} + d_{n-1} + 1, \\ d_n = c_{n-1} + d_{n-1}, \end{cases}$$

together with $c_1 = 1$ and $d_1 = 0$. The solution to this recurrence is $c_n = 2^{n-1}$ and $d_n = 2^{n-1} - 1$. □

Proposition 1. *The number occurrences of the subsequence* 1^k *(resp.* 2^k*) in* C_n *is* $\binom{2^{n-1}}{k}$ *(resp.* $\binom{2^{n-1}-1}{k}$*). Thus, the number of occurrences of the pattern* 1^k *in* C_n *is equal to*

$$c_n^{1^k} = \binom{2^{n-1}}{k} + \binom{2^{n-1}-1}{k} = \frac{2^n - k}{2^{n-1} - k}\binom{2^{n-1}-1}{k}.$$

Proof. From Lemma 1, there are 2^{n-1} (resp. $2^{n-1}-1$) occurrences of the letter 1 (resp. 2) in C_n, and thus there are $\binom{2^{n-1}}{k}$ (resp. $\binom{2^{n-1}-1}{k}$) occurrences of the subsequence 1^k (resp. 2^k) there. □

Proposition 2. *We have that for all* $n \geq 2$, $c_n^{1\text{-}2} = d_n^{1\text{-}2} = 2\cdot 4^{n-2} + (n-2)\cdot 2^{n-2}$, *and* $c_n^{2\text{-}1} = d_n^{2\text{-}1} = 2 \cdot 4^{n-2} - n \cdot 2^{n-2}$.

Proof. Let us first consider the pattern 1-2. An occurrence of this pattern in $C_n = C_{n-1}1D_{n-1}$ is either inside C_{n-1}, or inside D_{n-1}, or the letter 1 is from the word $C_{n-1}1$, whereas the letter 2 is from the word D_{n-1}. Thus

$c_n^{1\text{-}2} = c_{n-1}^{1\text{-}2} + d_{n-1}^{1\text{-}2} + \{$ (the number of 1s in C_{n-1}) $+ 1\}\cdot\{$ the number of 2s in $D_{n-1}\}$.

Using the same considerations for $D_n = C_{n-1}2D_{n-1}$, one can get

$d_n^{1\text{-}2} = c_{n-1}^{1\text{-}2} + d_{n-1}^{1\text{-}2} + \{$ the number of 1s in $C_{n-1}\}\cdot\{$ (the number of 2s in D_{n-1}) $+ 1\}$.

The number of 1s and 2s in C_{n-1} and D_{n-1} is given in Lemma 1. So,

$$\left\{ \begin{array}{l} c_n^{1\text{-}2} = c_{n-1}^{1\text{-}2} + d_{n-1}^{1\text{-}2} + 2^{n-2}\cdot(2^{n-2}+1) \\ d_n^{1\text{-}2} = c_{n-1}^{1\text{-}2} + d_{n-1}^{1\text{-}2} + 2^{n-2}\cdot(2^{n-2}+1) \end{array} \right. \Leftrightarrow$$

$$\begin{pmatrix} c_n^{1\text{-}2} \\ d_n^{1\text{-}2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_{n-1}^{1\text{-}2} \\ d_{n-1}^{1\text{-}2} \end{pmatrix} + \begin{pmatrix} 2^{n-2}\cdot(2^{n-2}+1) \\ 2^{n-2}\cdot(2^{n-2}+1) \end{pmatrix} \qquad (9.1)$$

together with $c_2^{1\text{-}2} = 2$ and $d_2^{1\text{-}2} = 2$. Here, and several times in what follows, we need to solve recurrence relations of the form

$$x_n = Ax_{n-1} + b,$$

where A is a matrix, and x_n, x_{n-1} and b are some vectors, where b sometimes depends on n. We recall from linear algebra that such relations can be solved by diagonalization of the matrix A, that is, by writing $A = VDV^{-1}$, where D is a diagonal matrix consisting of eigenvalues of A, and the columns of V are eigenvectors of A. For example, if A is a 2×2 matrix that consists of 1s, then we use

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

for computing powers of A, and thus for solving the recurrence relations. For the recurrence 9.1, we get that for all $n \geq 2$, $c_n^{1\text{-}2} = d_n^{1\text{-}2} = 2\cdot 4^{n-2} + (n-2)\cdot 2^{n-2}$.

In the same manner, we can get that for the pattern 2-1,

$$\begin{cases} c_n^{2\text{-}1} = c_{n-1}^{2\text{-}1} + d_{n-1}^{2\text{-}1} + 2^{n-2} \cdot (2^{n-2} - 1), \\ d_n^{2\text{-}1} = c_{n-1}^{2\text{-}1} + d_{n-1}^{2\text{-}1} + 2^{n-2} \cdot (2^{n-2} - 1), \end{cases}$$

together with $c_3^{2\text{-}1} = 2$ and $d_3^{2\text{-}1} = 2$. This gives, that for all $n \geq 2$, $c_n^{2\text{-}1} = d_n^{2\text{-}1} = 2 \cdot 4^{n-2} - n \cdot 2^{n-2}$. □

Proposition 2 shows that asymptotically, the numbers of occurrences of the patterns, or the subsequences, 1-2 and 2-1 in C_n or D_n are equal.

9.4 Patterns without internal dashes

Recall the definitions in Section 9.2.

Theorem 1. *Let $\tau = \tau_1 \tau_2 \dots \tau_\ell$ be an arbitrary generalized pattern without internal dashes that consists of 1s and 2s. Suppose $k = \lceil \log_2 \ell \rceil$, $a = m_\ell(\tau, D_k 1 C_k)$, and $b = m_\ell(\tau, D_k 2 C_k)$. Then for $n > k+1$, we have*

$$c_n^\tau = (a + b + c_{k+1}^\tau + d_{k+1}^\tau) \cdot 2^{n-k-2} - b,$$

$$d_n^\tau = (a + b + c_{k+1}^\tau + d_{k+1}^\tau) \cdot 2^{n-k-2} - a.$$

Proof. Suppose $n > k+1$. In this case, $C_n = C_{n-1} 1 D_{n-1} = W_1 \mathcal{K}_\ell(D_k 1 C_k) W_2$, for some words W_1 and W_2 such that $|W_1| = |W_2|$. Because of the definition of the kernel $\mathcal{K}_\ell(D_k 1 C_k)$, an occurrence of the pattern τ in C_n is in either C_{n-1}, or D_{n-1}, or $\mathcal{K}_k(D_k 1 C_k)$ (from the definitions $|C_{n-1} \cap \mathcal{K}_k(D_k 1 C_k)| = |D_{n-1} \cap \mathcal{K}_k(D_k 1 C_k)| = \ell - 1$ and thus these intersections cannot be an occurrence of τ). So,

$$c_n^\tau = c_{n-1}^\tau + d_{n-1}^\tau + a,$$

whereas in the same way, we can obtain that

$$d_n^\tau = c_{n-1}^\tau + d_{n-1}^\tau + b.$$

By solving these recurrence relations, we get the desirable. □

In particular, Theorem 1 is valid for $\ell = 1$, in which case the number of occurrences of τ in C_n (or D_n) is the number of letters in C_n (or D_n). Indeed, in this case, $k = 0$, $a = b = c_1^1 = d_1^1 = 1$, hence $c_n^1 = d_n^1 = 2^n - 1 = |C_n| = |D_n|$. Also, as a corollary to Theorem 1 we have, that if $a = b = c_{k+1}^\tau = d_{k+1}^\tau = 0$ for some pattern τ, then this pattern never appears in sigma-sequence.

All of the following examples are corollaries to Theorem 1.

Example 1. *Suppose $\tau = 12$. We have that $k = 1$, $a = m_2(12, D_1 1 C_1) = 0$ and $b = m_2(12, D_1 2 C_1) = 0$. Besides, $c_2^{12} = 1$ and $d_2^{12} = 1$. Thus using Theorem 1, for all $n > 2$, $c_n^{12} = 2^{n-2}$. So, the number of rises in C_n is equal to 2^{n-2}, for $n \geq 2$.*

If $\tau = 21$, again $k = 1$, but now $a = m_2(21, D_1 1 C_1) = 1$, $b = m_2(21, D_1 2 C_1) = 1$. Besides, $c_3^{21} = 1$ and $d_3^{21} = 1$. From Theorem 1, for all $n > 3$, $c_n^{21} = 2^{n-2} - 1$, which shows that the number of descents in C_n is one less than the number of rises.

Since in both cases $a = b$, using the recurrences in Theorem 1, we have that $c_n^{12} = d_n^{12} = 2^{n-2}$, whereas $c_n^{21} = d_n^{21} = 2^{n-2} - 1$.

Example 2. *Suppose $\tau = 112$. We have that $k = 2$, $a = m_3(112, D_2 1 C_2) = 0$, and $b = m_3(112, D_2 2 C_2) = 0$. Besides, $c_3^{112} = 2$ and $d_3^{112} = 1$. Now, from Theorem 1, we have that for all $n > 3$, $c_n^{112} = d_n^{112} = 3 \cdot 2^{n-4}$.*

Example 3. *Suppose $\tau = 221$. We have that $k = 2$, $a = m_3(221, D_2 1 C_2) = 1$, and $b = m_3(221, D_2 2 C_2) = 1$. Besides, $c_3^{221} = 0$ and $d_3^{221} = 1$. Now, from Theorem 1, we have that for all $n > 3$, $c_n^{221} = d_n^{221} = 3 \cdot 2^{n-4} - 1$.*

Example 4. *If $\tau = 2212221$ then $k = 3$, $a = m_7(221, D_3 1 C_3) = 0$, $b = m_7(221, D_3 2 C_3) = 1$, $c_4^{2212221} = 0$, and $d_4^{2212221} = 0$. Thus for $n \geq 4$, $c_n^{2212221} = 2^{n-4} - 1$.*

9.5 Patterns of the form τ_1-τ_2

Theorem 2. *Let $p = \tau_1$-τ_2 be a generalized pattern such that $|\tau_1| = k_1$ and $|\tau_2| = k_2$. Suppose $k = \lceil \log_2(k_1 + k_2 - 1) \rceil$. The following denote the number of occurrences of the subwords τ_1 and τ_2 in the kernels:*

$$\begin{array}{ll} a_{\tau_1} = m_{k_1}(\tau_1, D_k 1 C_k) & a_{\tau_2} = m_{k_2}(\tau_2, D_k 1 C_k) \\ b_{\tau_1} = m_{k_1}(\tau_1, D_k 2 C_k) & b_{\tau_2} = m_{k_2}(\tau_2, D_k 2 C_k) \end{array}$$

Also, let r_1^a (resp. r_2^a, r_1^b, r_2^b) denote the number of occurrences of overlapping subwords τ_1 and τ_2 in the word $D_k 1 C_k$ (resp. $D_k 1 C_k$, $D_k 2 C_k$, $D_k 2 C_k$), where $\tau_1 \in \mathcal{K}_{k_1}(D_k 1 C_k)$ and $\tau_2 \in C_k$ (resp. $\tau_1 \in D_k$ and $\tau_2 \in \mathcal{K}_{k_2}(D_k 1 C_k)$, $\tau_1 \in \mathcal{K}_{k_1}(D_k 2 C_k)$ and $\tau_2 \in C_k$, $\tau_1 \in D_k$ and $\tau_2 \in \mathcal{K}_{k_2}(D_k 2 C_k)$).

Besides, we assume that we know $c_n^{\tau_i}$ and $d_n^{\tau_i}$ for $n > n_i$, $i = 1, 2$. Then for $n > \max(k+1, n_1+1, n_2+1)$, c_n^τ and d_n^τ are given by the following recurrence:

$$\begin{pmatrix} c_n^\tau \\ d_n^\tau \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_{n-1}^\tau \\ d_{n-1}^\tau \end{pmatrix} + \begin{pmatrix} \alpha_n \\ \beta_n \end{pmatrix},$$

where

$$\alpha_n = (c_{n-1}^{\tau_1} + a_{\tau_1} - r_1^a) d_{n-1}^{\tau_2} + (a_{\tau_2} - r_2^a) c_{n-1}^{\tau_1}$$

and

$$\beta_n = (c_{n-1}^{\tau_1} + b_{\tau_1} - r_1^b) d_{n-1}^{\tau_2} + (b_{\tau_2} - r_2^b) c_{n-1}^{\tau_1}.$$

Proof. Suppose $n > \max(k+1, n_1+1, n_2+1)$. Let us find a recurrence for the number c_n^τ (one can use the same considerations for d_n^τ).

An occurrence of the pattern τ in $C_n = C_{n-1} 1 D_{n-1}$ is either inside C_{n-1}, or inside D_{n-1}, or begins in C_{n-1} or the letter 1 between C_{n-1} and D_{n-1} and

ends in D_{n-1} or the letter 1. The first two cases obviously give c_{n-1} and d_{n-1} occurrences of τ. To count the contribution of the last to cases, we work with words instead of patterns. We do it to take in account the situations when τ_1 or τ_2 consists of copies of only one letter. In this case, we cannot count occurrence of these patterns separately, and then use this information, since, for instance, occurrences of the pattern $\tau_1 = 111$ are subwords 111 and 222 (the last one of these subwords we do not need), whereas occurrences of the pattern $\tau_1 = 222$ are not defined at all (222 is not a pattern).

If an occurrence of τ_1-τ_2 does not entirely belong to C_{n-1} or D_{n-1} then we only have one of the following possibilities:

(a) the subword τ_1 entirely belongs to C_{n-1} and the subword τ_2 entirely belongs to D_{n-1};

(b) the subword τ_1 belongs entirely to C_{n-1} and the subword τ_2 belongs to the kernel $\mathcal{K}_{k_2}(D_k 1 C_k)$, where $k = \lceil \log_2(k_1 + k_2 - 1) \rceil$ is the least number that allow to control, in C_n $(n > k)$, overlapping occurrences of subwords τ_1 and τ_2 where τ_1 is entirely from C_{n-1} and $\tau_2 \in \mathcal{K}_{k_2}(D_k 1 C_k)$;

(c) the subword τ_2 belongs entirely to D_{n-1} and the subword τ_1 belongs to the kernel $\mathcal{K}_{k_1}(D_k 1 C_k)$.

In (a) we obviously have $c_{n-1}^{\tau_1} \cdot d_{n-1}^{\tau_2}$ possibilities.

In (b) we have $c_{n-1}^{\tau_1} \cdot a_{\tau_2} - c_{n-1}^{\tau_1} \cdot r_2^a$ possibilities, since we need to subtract those occurrences of τ_1 and τ_2 that overlap.

Analogically to (b), in (c) we have $d_{n-1}^{\tau_2} \cdot a_{\tau_1} - d_{n-1}^{\tau_2} \cdot r_1^a$ possibilities, which completes the proof. □

Remark 6. *For using Theorem 2, one needs to know c_n^τ and d_n^τ for patterns τ without internal dashes. These numbers could be obtained by using Theorem 1.*

The following corollary to Theorem 2 is straitforward to prove, using the fact that for non-overlapping patterns τ_1 and τ_2, $r_1^a = r_2^a = r_1^b = r_2^b = 0$.

Corollary 12. *We make the same assumptions as those in Theorem 2. Suppose additionally that the words τ_1 and τ_2 are not overlapping in the following sense: no one suffix of τ_1 is a prefix of τ_2. Then for $n > \max(k+1, n_1+1, n_2+1)$, c_n^τ and d_n^τ are given by the same recurrence as that in Theorem 2 with*

$$\alpha_n = (c_{n-1}^{\tau_1} + a_{\tau_1}) d_{n-1}^{\tau_2} + a_{\tau_2} c_{n-1}^{\tau_1}$$

and

$$\beta_n = (c_{n-1}^{\tau_1} + b_{\tau_1}) d_{n-1}^{\tau_2} + b_{\tau_2} c_{n-1}^{\tau_1}.$$

Remark 7. *Corollary 12 is valid under more weak assumptions, namely we only need the property of non-overlapping of the patterns τ_1 and τ_2 when one of them is in its kernel and the other one is not in its kernel. Example 7 deals with the pattern τ that has overlapping blocks τ_1 and τ_2, but Corollary 12 can be applied. However, from practical point of view, checking the fact if two subwords are non-overlapping is more easy than considering the kernels and checking the non-overlapping of the subwords there.*

Example 5. *Suppose* $\tau = 12\text{-}21$. *We have that* $|\tau_1| = |\tau_2| = 2$. *Now, in the statement of Theorem 2 we have that* $k = 2$, $a_{\tau_1} = 0$, $a_{\tau_2} = 1$, $b_{\tau_1} = 0$ *and* $b_{\tau_2} = 1$. *Also, since there are no overlapping occurrences of the subwords* 12 *and* 21 *in* $\mathcal{K}_3(1221112)$ *and* $\mathcal{K}_3(1222112)$, *we have* $r_1^a = 0$, $r_2^a = 0$, $r_1^b = 0$ *and* $r_2^b = 0$. *Besides, from example 1,* $c_n^{12} = d_n^{12} = 2^{n-2}$ *and* $c_n^{21} = d_n^{21} = 2^{n-2} - 1$. *Thus,* $\alpha_n = \beta_n = 4^{n-3}$. *Using the fact that* $c_3^{12\text{-}21} = 0$ *and* $d_3^{12\text{-}21} = 1$, *this allows us to get an explicit formula for* $c_n^{12\text{-}21}$ *and* $d_n^{12\text{-}21}$ *for* $n > 3$*:*

$$c_n^{12\text{-}21} = d_n^{12\text{-}21} = \frac{1}{2}4^{n-2} - 3 \cdot 2^{n-4}.$$

In particular $c_4^{12\text{-}21} = 5$.

Example 6. *Suppose* $\tau = 1\text{-}221$. *We have that* $|\tau_1| = 1$ *and* $|\tau_2| = 3$. *Moreover, the words* τ_1 *and* τ_2 *are not overlapping, hence we can use Corollary 12. We have that* $k = 2$, $a_{\tau_1} = 1$, $a_{\tau_2} = 1$, $b_{\tau_1} = 0$ *and* $b_{\tau_2} = 1$. *From example 3,* $d_n^{221} = 3 \cdot 2^{n-4} - 1$. *Also, the number of occurrences of the letter 1 (the subword* $\tau_1 = 1$*) is given by Lemma 1:* $c_n^1 = 2^{n-1}$. *So,* $\alpha_n = 6 \cdot 4^{n-4} + 3 \cdot 2^{n-5} - 1$ *and* $\beta_n = 6 \cdot 4^{n-4}$. *One can get now an explicit formula for* $c_n^{1\text{-}221}$ *and* $d_n^{1\text{-}221}$ *for* $n > 4$*:*

$$c_n^{1\text{-}221} = \tfrac{1}{2}4^{n-2} + 27 \cdot 2^{n-5} - n - 7,$$
$$d_n^{1\text{-}221} = \tfrac{1}{2}4^{n-2} + 21 \cdot 2^{n-5} - 8.$$

In particular, $c_5^{1\text{-}221} = 47$.

Example 7. *Suppose* $\tau = 112\text{-}21$. *We have that* $|\tau_1| = k_1 = 3$ *and* $|\tau_2| = k_2 = 2$. *The other parameters in Theorem 2 are* $k = 3$, $a_{\tau_1} = 0$, $a_{\tau_2} = 1$, $b_{\tau_1} = 0$, $b_{\tau_2} = 1$, $r_1^a = r_2^a = r_1^b = r_2^b = 0$. *From Example 2, for* $n \geq 4$, $c_n^{112} = 3 \cdot 2^{n-4}$, *and from Example 1,* $d_n^{21} = 2^{n-2} - 1$. *Thus, in Theorem 2,* $\alpha_n = \beta_n = c_{n-1}^{112}(d_{n-1}^{21} + 1) = 3 \cdot 4^{n-4}$. *Now, we solve the recurrence relation from the theorem to get, that for* $n > 3$

$$c_n^{112\text{-}21} = d_n^{112\text{-}21} = \frac{3}{2} \cdot 4^{n-3} - 2^{n-4}.$$

9.6 Counting occurrences of $\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k$

In this section we study the number of occurrences of a pattern $\tau = \tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k$, where τ_i are patterns without internal dashes. We say that τ consists of k blocks. We assume that for $i = 1, 2, \ldots, k-1$, the pattern τ_i does not overlap with the patterns τ_{i-1} and τ_{i+1}. In this case we give a recurrence relation for the number of occurrences of τ, provided we know the number of occurrences of certain patterns consisting of less than, or equal to, $k-1$ blocks, as well as $2k$ certain numbers which can be calculated by considering the words $D_\ell 1 C_\ell$ and $C_\ell 2 D_\ell$, where ℓ is the maximum number such that $\ell \leq \max_i \lceil \log_2 |\tau_i| \rceil$. The cases of $k = 1$ and $k = 2$ are studied in the previous sections; they give the bases for our calculations. However, the case of overlapping patterns τ_i is not solved, and it remains as a challenging problem, since an answer to this problem

gives the way to count occurrences of an arbitrary generalized pattern, or an arbitrary subsequence, in σ-words.

Theorem 3. *Let $\tau = \tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k$ be a generalized pattern such that $|\tau_i| = k_i$ for $i = 1, 2, \ldots, k$. We assume that for $i = 1, 2, \ldots, k-1$, the subword τ_i does not overlap with the subwords τ_{i-1} and τ_{i+1} in the following sense: no one suffix of τ_{i-1} is a prefix of τ_i and no one suffix of τ_i is a prefix of τ_{i+1}.*

Suppose $\ell_i = \lceil \log_2 k_i \rceil$, $\ell = \max_i \ell_i$, and for the subwords τ_i we have $a_i = m_{k_i}(\tau_i, D_{\ell_i} 1 C_{\ell_i})$ and $b_i = m_{k_i}(\tau_i, D_{\ell_i} 2 C_{\ell_i})$, for $i = 1, 2, \ldots, k$.

We assume that we know $c_{n-1}^{\tau_1\text{-}\cdots\text{-}\tau_i}$ and $d_{n-1}^{\tau_{i+1}\text{-}\cdots\text{-}\tau_k}$ for each $1 \le i \le k-1$ and for all $n > n^\star$. Then for all $n > \max(\ell + 1, n^\star + 1)$, c_n^τ and d_n^τ are given by the following recurrence:

$$\begin{pmatrix} c_n^\tau \\ d_n^\tau \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_{n-1}^\tau \\ d_{n-1}^\tau \end{pmatrix} + \sum_{i=1}^{k-1} \begin{pmatrix} c_{n-1}^{\tau_1\text{-}\cdots\text{-}\tau_i} \cdot d_{n-1}^{\tau_{i+1}\text{-}\cdots\text{-}\tau_k} \\ c_{n-1}^{\tau_1\text{-}\cdots\text{-}\tau_i} \cdot d_{n-1}^{\tau_{i+1}\text{-}\cdots\text{-}\tau_k} \end{pmatrix} + \sum_{i=1}^{k} \begin{pmatrix} a_i \cdot c_{n-1}^{\tau_1\text{-}\cdots\text{-}\tau_{i-1}} \cdot d_{n-1}^{\tau_{i+1}\text{-}\cdots\text{-}\tau_k} \\ b_i \cdot c_{n-1}^{\tau_1\text{-}\cdots\text{-}\tau_{i-1}} \cdot d_{n-1}^{\tau_{i+1}\text{-}\cdots\text{-}\tau_k} \end{pmatrix}.$$

Proof. We consider only c_n^τ, since the same arguments can be applied to d_n^τ. We use the considerations similar to those in Theorem 2.

An occurrence of the pattern τ in $C_n = C_{n-1}1D_{n-1}$ can be entirely in C_n or D_n. The first term counts such occurrences. Otherwise, we have two possibilities: either the letter 1 between the words C_{n-1} and D_{n-1} does not belong to an occurrence of τ, or it does do it, in which case there exist i (exactly one) such that the subword τ_i occurs in its kernel. The first sum in the statement is obviously responsible for the first of this cases, whereas the second sum is responsible for the second case (in the last case we use the fact that subwords τ_i are not overlapping). □

As a corollary to Theorem 3, we have Corollary 12.

The following example is another corollary to Theorem 3.

Example 8. *Suppose $\tau = 2-1-221$, that is, $\tau_1 = 2$, $\tau_2 = 1$ and $\tau_3 = 221$. So, parameters in Theorem 3 are the following: $k_1 = k_2 = 1$, $k_3 = 3$, $\ell_1 = \ell_2 = 1$, $\ell_3 = 2$, $\ell = 2$. From $D_1 1 C_1 = 211$ we obtain $a_1 = 0$, $a_2 = 1$. From $D_2 1 C_2 = 1221112$ we obtain $a_3 = 1$. From $D_1 2 C_1 = 221$ we get $b_1 = 1$, $b_2 = 0$. From $D_2 2 C_2 = 1222112$ we get $b_3 = 1$. Besides, from Proposition 2, Examples 3 and 6, we have*

$$c_n^{\tau_1\text{-}\tau_2} = c_n^{2\text{-}1} = 2 \cdot 4^{n-2} - n \cdot 2^{n-2}, \text{for } n > 1;$$

$$d_n^{\tau_3} = d_n^{221} = 3 \cdot 2^{n-4} - 1, \text{for } n > 3;$$

$$d_n^{\tau_2\text{-}\tau_3} = d_n^{1\text{-}221} = \tfrac{1}{2} \cdot 4^{n-2} + 21 \cdot 2^{n-5} - 8, \text{for } n > 4.$$

Also, the number of occurrences of the subword $\tau_1 = 2$ in C_n is given by Proposition 1: $c_n^{\tau_1} = c_n^2 = 2^{n-1} - 1$. So, the number of occurrences of the pattern τ

in C_n and D_n, for $n > 5$, satisfies the following recurrence relation:

$$\begin{pmatrix} c_n^\tau \\ d_n^\tau \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_{n-1}^\tau \\ d_{n-1}^\tau \end{pmatrix} + \begin{pmatrix} \frac{5}{1024}8^n + \frac{25-3n}{256}4^n - \frac{171}{64}2^n + 9 \\ \frac{5}{1024}8^n + \frac{21-3n}{256}4^n - 2^{n+1} \end{pmatrix},$$

with initial conditions $c_5^\tau = 70$ and $d_5^\tau = 74$.

9.7 Patterns of the form $[\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k]$, $[\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k)$ and $(\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k]$

We recall that according to Babson and Steingrímsson notation for generalized patterns, if we use "[" in a pattern, for example if we write $p = [1\text{-}2)$, we indicate that in an occurrence of p, the letter corresponding to the 1 must be the first letter of a word under consideration, whereas if we write, say, $p = (1\text{-}2]$, then the letter corresponding to 2 must be the last (rightmost) letter of the word.

In the theorems of this section, we assume that we can find the numbers $c_n^{\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k}$ and $d_n^{\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k}$ for any patterns τ_i, $i = 1, 2, \ldots, k$, without internal dashes. For certain special cases, these numbers can be obtained using the theorems of Sections 9.5 and 9.6.

Theorem 4. *Suppose τ_1 and τ_2 are two patterns without internal dashes such that $|\tau_1| = k_1$ and $|\tau_2| = k_2$. Also, suppose $\ell_1 = \log_2(k_1 + 1)$, $\ell_2 = \log_2(k_2 + 1)$ and $\ell = \log_2(k_1 + k_2 + 1)$.*

Let $a(\tau_1, \tau_2)$ be the number of overlapping subwords τ_1 and τ_2 in C_ℓ such that τ_1 consists of the k_1 leftmost letters of C_ℓ; $b(\tau_1, \tau_2)$ is the number of overlapping subwords τ_1 and τ_2 in C_ℓ such that τ_2 consists of the k_2 rightmost letters of C_ℓ.

We assume that we know $c_n^{\tau_i}$ and $d_n^{\tau_i}$ for $i = 1, 2$ and for all $n > n^\star$.

i. For $n \geq \max(\ell_1, n^\star)$,

$$c_n^{[\tau_1\text{-}\tau_2)} = \begin{cases} c_n^{\tau_2} - a(\tau_1, \tau_2), & \text{if } C_{\ell_1} \text{ begins with } \tau_1, \\ 0, & \text{otherwise.} \end{cases}$$

ii. For $n \geq \max(\ell_2, n^\star)$,

$$c_n^{(\tau_1\text{-}\tau_2]} = \begin{cases} c_n^{\tau_1} - b(\tau_1, \tau_2), & \text{if } C_{\ell_2} \text{ ends with } \tau_2, \\ 0, & \text{otherwise.} \end{cases}$$

iii. For $n \geq \ell$,

$$c_n^{[\tau_1\text{-}\tau_2]} = \begin{cases} 1, & \text{if } C_\ell \text{ begins with } \tau_1 \text{ and ends with } \tau_2, \\ 0, & \text{otherwise.} \end{cases}$$

iv. *For* $n \geq \max(\ell_1, n^\star)$,

$$d_n^{[\tau_1\text{-}\tau_2)} = \begin{cases} d_n^{\tau_2} - a(\tau_1, \tau_2), & \text{if } D_{\ell_1} \text{ begins with } \tau_1, \\ 0, & \text{otherwise.} \end{cases}$$

v. *For* $n \geq \max(\ell_2, n^\star)$,

$$d_n^{(\tau_1\text{-}\tau_2]} = \begin{cases} d_n^{\tau_1} - b(\tau_1, \tau_2), & \text{if } D_{\ell_2} \text{ ends with } \tau_2, \\ 0, & \text{otherwise.} \end{cases}$$

vi. *For* $n \geq \ell$,

$$d_n^{[\tau_1\text{-}\tau_2]} = \begin{cases} 1, & \text{if } D_\ell \text{ begins with } \tau_1 \text{ and ends with } \tau_2, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. We prove case i, all the other cases are then easy to see.

Clearly, if C_{ℓ_1} does not begin with τ_1 then C_n does not begin with τ_1 for all $n \geq \ell_1$, which means that $c_n^{[\tau_1\text{-}\tau_2)} = 0$ in this case. Otherwise, to count occurrences of the pattern $[\tau_1\text{-}\tau_2)$ is the same as to find the number of occurrences of the pattern τ_2 in C_n and then subtract the number of such occurrences of τ_2 that begin from the i-th letter of C_n, where $1 \leq i \leq k_1$. □

The following two examples are corollaries to Theorem 4.

Example 9. *Suppose we have the patterns* $\sigma_1 = [1122 - 21211)$ *and* $\sigma_2 = (21221 - 12]$. *From Theorem 4,* $c_n^{\sigma_1} = d_n^{\sigma_1} = 0$, *since* C_3 *does not begin with* 1122 *(*$\ell_1 = 3$*). Also,* $c_n^{\sigma_2} = d_n^{\sigma_2} = 0$, *since* C_3 *does not end with* 12 *(*$\ell_2 = 3$*).*

Example 10. *Suppose* $\tau = [112\text{-}21)$. *We have that* $k_1 = 3$, $\ell_1 = 2$ *and* C_2 *begins with the subword* 112. *Besides,* $a(112, 21) = 1$ *and, from Example 1,* $c_n^{21} = d_n^{21} = 2^{n-2} - 1$. *Theorem 4 now gives, that for* $n > 3$, *we have* $c_n^{[112\text{-}21)} = c_n^{\tau_2} - a(\tau_1, \tau_2) = 2^{n-2} - 2$.

The following theorem is straitforward to prove using the assumptions concerning non-overlapping of certain subwords.

Theorem 5. *Let* $\{\tau_1, \tau_2, \ldots, \tau_k\}$ *be a set of generalized patterns without internal dashes. Suppose* $|\tau_1| = s_1$, $|\tau_k| = s_k$, $\ell_1 = \log_2(s_1 + 1)$ *and* $\ell_k = \log_2(s_k + 1)$. *Also,* $\ell = \max(\ell_1, \ell_k)$.

i. *With the assumption that the subword* τ_1 *does not overlap with the subword* τ_2, *that is, no one suffix of* τ_1 *is a prefix of* τ_2, *we have*

(a)

$$c_n^{[\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k)} = \begin{cases} c_n^{\tau_2\text{-}\tau_3\text{-}\cdots\text{-}\tau_k}, & \text{if } C_{\ell_1} \text{ begins with } \tau_1, \\ 0, & \text{otherwise.} \end{cases}$$

(b)

$$d_n^{[\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k)} = \begin{cases} d_n^{\tau_2\text{-}\tau_3\text{-}\cdots\text{-}\tau_k}, & \text{if } D_{\ell_1} \text{ begins with } \tau_1, \\ 0, & \text{otherwise.} \end{cases}$$

ii. With assumption that the subword τ_{k-1} does not overlap with the subword τ_k, that is, no one suffix of τ_{k-1} is a prefix of τ_k, we have

(a)

$$c_n^{(\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k]} = \begin{cases} c_n^{\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_{k-1}}, & \text{if } C_{\ell_k} \text{ ends with } \tau_k, \\ 0, & \text{otherwise.} \end{cases}$$

(b)

$$d_n^{(\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k]} = \begin{cases} d_n^{\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_{k-1}}, & \text{if } D_{\ell_k} \text{ ends with } \tau_k, \\ 0, & \text{otherwise.} \end{cases}$$

iii. With the assumption that the subword τ_1 does not overlap with the subword τ_2, and the subword τ_{k-1} does not overlap with the subword τ_k, we have

(a)

$$c_n^{[\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k]} = \begin{cases} c_n^{\tau_2\text{-}\tau_3\text{-}\cdots\text{-}\tau_{k-1}}, & \text{if } C_\ell \text{ begins with } \tau_1 \text{ and ends with } \tau_k, \\ 0, & \text{otherwise.} \end{cases}$$

(b)

$$d_n^{[\tau_1\text{-}\tau_2\text{-}\cdots\text{-}\tau_k]} = \begin{cases} d_n^{\tau_2\text{-}\tau_3\text{-}\cdots\text{-}\tau_{k-1}}, & \text{if } D_\ell \text{ begins with } \tau_1 \text{ and ends with } \tau_k, \\ 0, & \text{otherwise.} \end{cases}$$

The following example is a corollary to Theorem 5.

Example 11. *Suppose τ = [112-1-221-22]. The parameters of Theorem 5 are $k_1 = 3$, $k_2 = 2$, $\ell_1 = 2$, $\ell_2 = 2$, $\ell = 2$. C_3 begins with the subword 112 and ends with the subword 22. Thus by Theorem 5 and Example 6, $c_n^{[112\text{-}1\text{-}221\text{-}22]} = c_n^{1\text{-}221} = \frac{1}{2}4^{n-2} + 27 \cdot 2^{n-5} - n - 7$.*

Bibliography

[BabStein] Babson E., Steingrímsson E.: Generalized permutation patterns and a classification of the Mahonian statistics, Sém. Lothar. Combin. **44** (2000), Art. B44b, 18 pp.

[Burstein] Burstein A., Enumeration of words with forbidden patterns, Ph.D. thesis, University of Pennsylvania, (1998).

[BurMans1] Burstein A., Mansour T.: Words restricted by patterns with at most 2 distinct letters, *Electronic J. of Combinatorics*, to appear (2002).

[BurMans2] Burstein A., Mansour T.: Words restricted by 3-letter generalized multipermutation patterns, preprint CO/0112281.

[BurMans3] Burstein A., Mansour T.: Counting occurrences of some subword patterns, preprint CO/0204320.

[Claes] A. Claesson: Generalised Pattern Avoidance, European J. Combin. **22** (2001), no. 7, 961–971.

[Evdok] Evdokimov A. A.: On the Maximal Chain Length of an Unit n-dimensional Cube, Maths Notes **6**, No. 3 (1969), 309–319. (Russian)

[Evdokimov] Private communication (2001).

[GelbOlm] Gelbaum B., Olmsted J.: *Counterexamples in Analysis*, Holden-day, San Francisco, London, Amsterdam, (1964).

[Kitaev] Kitaev S., There are no iterated morphisms that define the Arshon sequence and the sigma-sequence, to appear J. Automata, Languages and Combinatorics (2002).

[KitMans1] Kitaev S., Mansour T.: Counting the occurrences of generalized patterns in words generated by a morphism, preprint CO/0210170.

[KitMans2] Kitaev S., Mansour T.: The Peano curve and counting occurrences of some patterns, preprint CO/0210268.

[Knuth] Knuth D. E.: *The Art of Computer Programming*, 2nd ed. Addison Wesley, Reading, MA, (1973).

[Lothaire] Lothaire M.: *Combinatorics on Words*, Encyclopedia of Mathematics, Vol. **17**, Addison-Wesley (1986). Reprinted in the *Cambridge Mathematical Library*, Cambridge University Press, Cambridge UK (1997).

[Salomaa] Salomaa A.: *Jewels of Formal Language Theory*, Computer Science Press (1981).

[SimSch] Simion R., Schmidt F.: Restricted permutations, European J. Combin. **6**, no. 4 (1985), 383–406.

[Yab] Yablonsky S. V.: *Discrete mathematics and mathematical problems of cybernetics*, Nauka, Vol. **1**, Moscow (1974), 112–116. (Russian)